

Wydział Informatyki

Katedra Inżynierii Oprogramowania

Inżynieria oprogramowania i baz danych

Autorzy

Oleksandr Bondarchuk, 7164 Tomasz Chudobiński, 7332 Robert Cybulski, 7357 Monika Iwasaki, 7621 Przemysław Kłosiński, 7071 Tomasz Kowalski, 6589 Tomasz Krajewski, 7175 Rafał Kryński, 5026 Dawid Pacholczyk, 6144 Krzysztof Pałka, 3949 Mikołaj Pawilno-Pacewicz, 6837 Przemysław Paziewski, 6227 Bartłomiej Szczepocki, 7514 Przemysław Walerianczyk, 7143 Marcin Wilczak, 6800

Service Desk – generyczny system do obsługi zgłoszeń serwisowych

Promotor

Praca inżynierska dr inż. Mariusz Trzaska

Streszczenie

Celem niniejszej pracy inżynierskiej było stworzenie generycznego systemu zarządzania zgłoszeniami serwisowymi dla firm. Dokumentacja opisuje wszystkie etapy inżynierii tytułowego oprogramowania, począwszy od opisu konkurencyjnych rozwiązań, przez fazy analizy, projektowania, implementacji, a kończąc na testowaniu.

Projekt Service Desk składa się z dwóch aplikacji: administracyjnej i klienckiej. Pierwsza z nich nazywa się PJWSTK Workflow Designer i oferuje graficzny interfejs modelowania *workflow* (z ang. przepływ prac) z możliwością wykorzystania poniższych konstrukcji logicznych i funkcjonalności:

- warunków, na podstawie których system podąża odpowiednią ścieżką,
- przełączników, pozwalających na wybranie jednego z kilku różnych wariantów,
- pętli, realizowanych poprzez serię warunków,
- statusów, umożliwiających sprawdzić aktualny stan problemu,
- automatycznych emaili do klientów, dzięki czemu są na bieżąco informowani o ważnych zmianach w ich zgłoszeniach.

Gotowe przepływy można wykorzystać w webowej aplikacji klienckiej – Service Desk. Oprogramowanie to służy do obsługi klienta, zgłaszającego do serwisu wadliwy produkt lub usługę. Aplikacja pozwala zastąpić długie procedury interaktywnym interfejsem, który jest intuicyjny i nie wymaga długiego szkolenia konsultantów, by móc z nim pracować.

Projekt został napisany z wykorzystaniem .NET Framework, języka C#, wzorca architektonicznego Model View Controller, technologii Windows Workflow Foundation, bibliotek CSS Bootstrap oraz bazy danych Microsoft SQL Server.

Spis treści

St	reszczen	ie	2
Sp	ois treści		3
1.	Wste	p	8
	1.1.	Cel i zakres pracy	8
	1.2.	Struktura pracy	8
	1.3.	Struktura organizacyjna zespołu projektowego	9
2.	Apli	kacje wsparcia dla usług i urządzeń	. 11
	2.1.	Wprowadzenie do problematyki serwisowania	. 11
	2.1.1.	Czym jest workflow?	. 11
	2.1.2.	Cechy systemów do obsługi zgłoszeń z workflow	. 11
	2.2.	Service Desk - narzędzie wspierające obszar eksploatacji usług ITIL	. 12
	2.2.1.	Czym jest ITIL	. 12
	2.2.2.	Podstawowe pojęcia ITIL	. 13
	2.2.3.	Wdrożenie Service Desk, jako element dobrej praktyki ITIL	. 14
	2.2.4.	Wybrane punkty zgodność narzędzia Service Desk ze standardem ITIL	. 19
	2.3.	HP Open View Service Desk (OVSD)	. 20
	2.3.1.	Wymagania funkcjonalne	. 20
	2.3.2.	Rejestracja zgłoszenia	. 21
	2.3.3.	Wyszukiwanie osoby	. 23
	2.3.4.	Dodawanie załączników	. 23
	2.3.5.	Wyszukiwanie zgłoszeń/wyświetlanie stanu rejestracji zgłoszenia	. 24
	2.3.6.	Hasła	. 24
	2.3.7.	Zakończenie pracy	. 25
	2.3.8.	Wyświetlanie listy usług/klasyfikacji	. 25
	2.3.9.	E-maile	. 25
	2.3.10). Wady rozwiązania	. 29
	2.4.	Oracle RightNow	. 29
	2.4.1.	Oracle RightNow Contextual Workspaces Cloud Service	. 30
	2.4.2.	Projektowanie formularzy	. 30
	2.4.3.	Generowanie scenariuszy rozmów	. 31
	2.4.4.	Skryptowanie	. 33
	2.4.5.	Procesy pracy	. 34
	2.4.6.	Ocena oprogramowania	. 34
	2.5.	SolveDirect	. 35

2.5.1.	Moduł SD.Call	. 36
2.5.2.	Moduł SD.Inventory	. 38
2.5.3.	Moduł SD.Bridge	. 38
2.5.4.	Moduł SD. Solution	. 39
2.5.5.	Moduł SD.Report	. 40
2.5.6.	Moduł SD.Message	. 41
2.5.7.	Moduł SD.Mobile	. 42
2.5.8.	Podsumowanie	. 43
3. Prop	onowane rozwiązanie	. 44
3.1.	Diagram przypadków użycia	. 44
3.1.1.	Aktorzy	. 44
3.1.2.	Administracja danymi	. 45
3.1.3.	Wyszukiwanie i wyświetlanie zgłoszeń.	. 48
3.1.4.	Tworzenie i praca ze zgłoszeniem	. 50
3.2.	Diagram klas	. 52
3.3.	Wymagania funkcjonalne	. 53
3.4.	Wymagania niefunkcjonalne	. 59
4. Wyk	orzystane technologie	. 62
4.1.	.NET Framework	. 62
4.1.1.	Przedstawienie	. 62
4.1.2.	Historia rozwoju	. 62
4.1.3.	Język pośredni	. 63
4.1.4.	Wspólne Środowisko Uruchomieniowe	. 65
4.1.5.	Biblioteka Klas	. 65
4.1.6.	Visual Studio	. 65
4.1.7.	Podsumowanie	. 66
4.2.	C#	. 66
4.2.1.	Przedstawienie języka C#	. 66
4.2.2.	Historia powstania języka	. 66
4.2.3.	C# na tle innych języków	. 67
4.2.4.	Frameworki	. 68
4.2.5.	Common Language Runtime	. 69
4.2.6.	Ciekawe funkcje – LINQ	. 70
4.3.	Model View Controller	. 71
4.4.	Windows Workflow Foundation	. 72
4.4.1.	Technologia	. 72
4.4.2.	Workflow Designer	. 74

4.5.	Microsoft Entity Framework	75
4.5.1	. Zarys rozwiązania	76
4.5.2.	. Przykładowe mapowania	76
4.5.3	Nowe podejście do tworzenia aplikacji – <i>Code-First</i>	77
4.6.	Bootstrap	
4.6.1	. Przegląd technologii	79
4.6.2.	. Grid	81
4.6.3	Base CSS	82
4.6.4	Dostosowanie	83
4.6.5	. Przykłady	84
4.7.	Microsoft SQL Server	86
4.7.1	. Relacyjny model danych	86
4.7.2.	. Historia SQL	86
4.7.3	. Składnia SQL	86
4.7.4	. Transact SQL	88
4.7.5	. Historia MS SQL	88
4.7.6	. Opis Systemu	89
5. Proj	ekt aplikacji	
5.1.	Zastosowane rozwiązania	
5.2.	Opis encji	
6. Imp	lementacja	
6.1.	Wstęp	
6.2.	Code-first	
6.3.	Modele danych	
6.4.	Budowa klasy biznesowej	
6.5.	Relacje	
6.5.1	. Relacja jeden do wielu	
6.5.2.	. Relacja wiele do wielu	
6.6.	Dziedziczenie	
6.7.	Aktualizacja bazy danych – migracje	103
6.8.	Testowanie funkcjonalności	
6.8.1	. Faza 0 – CRUD	105
6.8.1	1. Obsługa obiektu umowa	
6.8.1	.2. Obsługa obiektu grupa	
6.8.1	.3. Obsługa obiektu producent	
6.8.1	.4. Obsługa obiektu osoba	
6.8.1	.5. Obsługa obiektu firma	

	6.8.1.6.	Obsługa obiektu lokalizacja	110
	6.8.1.7.	Obsługa obiektu pracownicy	112
	6.8.1.8.	Obsługa obiektu Produkt - Urządzenie	113
	6.8.1.9.	Obsługa obiektu Produkt - Usługa	114
	6.8.2.	Faza I - Weryfikacja wprowadzanych danych	116
	6.8.2.1.	Weryfikacja wprowadzenia danych – obiekt Firma	116
	6.8.2.2.	Weryfikacja wprowadzenia danych – obiekt Lokalizacja	116
	6.8.2.3.	Weryfikacja wprowadzenia danych – obiekt Grupa	117
	6.8.2.4.	Weryfikacja wprowadzenia danych – przypisanie osób do Grupy	118
	6.8.2.5.	Weryfikacja wprowadzenia danych – obiekt Osoba	118
	6.8.2.6.	Weryfikacja wprowadzenia danych – obiekt Pracownik	119
	6.8.2.7.	Weryfikacja wprowadzenia danych – obiekt Stanowisko	120
	6.8.2.8.	Weryfikacja wprowadzenia danych – obiekt Produkt Urządzenie	120
	6.8.2.9.	Weryfikacja wprowadzenia danych – obiekt Produkt Usługa	121
	6.8.2.10	. Weryfikacja wprowadzenia danych – obiekt Producent	121
	6.8.2.11	. Weryfikacja wprowadzenia danych – obiekt Zgłoszenie	122
	6.8.3.	Faza II - testy interfejsu Użytkownika	123
	6.8.3.1.	Rejestracja zgłoszenia przez Konsultanta	123
	6.8.3.2.	Dodanie pracowników do grupy	124
	6.8.3.3.	Automatyczne nadawanie numeru zgłoszeń	125
	6.8.3.4.	Przeszukiwanie bazy osób zgłaszających po dostępnych polach	126
	6.8.3.5.	Przeszukiwanie bazy listy lokalizacji po dostępnych polach	126
	6.8.3.6.	Wyświetlanie listy zgłoszeń	127
	6.8.3.7.	Wyszukiwanie zgłoszeń po numerze zgłoszenia lub numerze umowy	127
	6.8.3.8.	Wyświetlanie listy zgłoszeń względem kończących się czasów SLA	128
	6.8.3.9.	Aktualizacja statusu zgłoszeń	128
	6.8.3.10	. Samodzielne definiowanie kroków w workflow	129
	6.8.3.11	. Workflow drukarka nie drukuje przypadek 1 (brak papieru)	130
	6.8.3.12	. Workflow drukarka nie drukuje przypadek 2 (diody się nie świecą)	133
	6.8.3.13	. Workflow drukarka nie drukuje przypadek 3 (brak tuszu/tonera)	136
	6.8.3.14	. Workflow drukarka nie drukuje przypadek 4 (zacięty papier)	136
	6.8.3.15	. Workflow awaria samochodu przypadek 1 (kontrolki się nie świecą)	137
	6.8.3.16	. Workflow awaria samochodu przypadek 2 (kontrolki się świecą)	139
6	.9. Te	stowanie użyteczności	144
	6.9.1.	Definicja	144
	6.9.2.	Metody testowania użyteczności	144
	6.9.3.	Elementy badania	145

	6.9.4.	Analiza ekspercka – badanie, część właściwa145		
	6.9.4.1.	Układ strony 145		
	6.9.4.2.	Nawigacja147		
	6.9.4.3.	Czytelność 148		
	6.9.4.4.	Ocena Końcowa		
7.	Przykł	ady zastosowań151		
7.	1. Ir	nterfejs151		
7.2	2. W	<i>lorkflow</i> w teorii		
	7.2.1.	Legenda		
	7.2.2.	Workflow "Weryfikacja Klienta"		
	7.2.3.	Workflow "Awaria drukarki"		
	7.2.4.	Workflow "Awaria Monitora"		
	7.2.5.	Workflow "Awaria Komputera"		
	7.2.6.	Workflow "Awaria Samochodu"		
7.	3. И	<i>lorkflow</i> w praktyce164		
7.4	4. P	odsumowanie167		
8.	Podsu	mowanie i wnioski		
9.	Słown	ik pojęć 170		
10.	Biblio	grafia171		
11.	Wykaz	z tabel		
12.	Wykaz rysunków176			
13.	Wykaz listingów179			

1. Wstęp

Obsługa serwisowa jest niezbędnym elementem opieki klienta powiązanym ze sprzedażą produktów i usług. Jednocześnie stanowi jeden ze sposobów kontaktu z klientem. Dlatego też odgrywa bardzo ważną rolę w funkcjonowaniu przedsiębiorstw i budowaniu ich pozycji rynkowej.

Szybkość i jakość obsługi jest kluczowa dla klienta, jak i producenta / dostawcy, gdyż bezpośrednio przekłada się na koszty, a w efekcie na końcowy zysk. Nieprzerwana praca urządzeń i usług, która może zostać zapewniona przez sprawny serwis przekłada się na markę oraz zaufanie klientów.

Zagadnieniem wsparcia obsługi serwisowej, począwszy od zgłoszenia awarii, poprzez monitorowanie naprawy, po zasygnalizowanie wykonanej naprawy zajmuje się zaprojektowana aplikacja Service Desk. Oprogramowanie to służy do optymalizacji obsługi zgłoszeń serwisowych. Ułatwia monitorowanie czasu naprawy oraz umożliwia definiowanie własnych procedur i przebiegów prac (*workflow*) serwisowych zależnych od typu awarii, serwisowanego sprzętu lub usługi.

1.1. Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest stworzenie prototypu aplikacji wspierającej działalność firm, zajmujących się komercyjnym serwisowaniem produktów i usług. Projekt tworzony jest w ramach pracy inżynierskiej realizowanej na specjalizacji Inżynieria Oprogramowania i Baz Danych (rok akademicki 2012/2013) w Polsko Japońskiej Wyższej Szkole technik Komputerowych (oddział Warszawa).

1.2. Struktura pracy

Niniejsza praca inżynierska – będąca efektem i stanowiąca podsumowanie – pracy wieloosobowego zespołu, stanowi prezentację utworzonej aplikacji, produktów konkurencyjnych oraz koncepcji przebiegów (*workflow*).

Jej struktura oddaje kolejność oraz chronologię przeprowadzonych prac i zawiera odpowiednio:

- niezbędne wprowadzenie do problematyki workflow;
- prezentację istniejących na rynku rozwiązań;
- charakterystykę oraz przede wszystkim nowości będące cechami aplikacji stworzonej w ramach projektu inżynierskiego;
- opis zastosowanej technologii, a wreszcie
- opis implementacji, użyteczności i przeprowadzonych testów.

Praca zakończona jest słownikiem pojęć charakterystycznych dla poruszanego obszaru wiedzy, spisem cytowanej literatury oraz spisem tabel i rysunków. Załącznikiem do pracy jest (zgodnie z Zarządzeniem Dziekana Wydziału Informatyki PJWSTK) gotowa aplikacja oraz jej kod źródłowy i sama praca w formie elektronicznej.

1.3. Struktura organizacyjna zespołu projektowego

Rysunek 1 przedstawia strukturę organizacyjną zespołu projektowego.



Rysunek 1. Struktura organizacyjna zespołu projektowego. Źródło: Opracowanie własne.

Rada nadzorcza projektu – Zajmuje się nadzorem i kontrolą postępów prac, poczynionych przez cały zespół. Wspiera grupę merytoryczne w zakresie prowadzenia projektu oraz wykorzystywanych technologii. Podejmuje ostateczne decyzje dotyczące ocen pracy poszczególnych członków zespołu.

Kierownik projektu – Odpowiada za realizację projektu w wyznaczonym czasie, jego jakość i zgodność z wymaganiami. Zajmuje się ustaleniem harmonogramu prac, ich podziałem i wyznaczaniem terminów. Do jego obowiązków należy też powołanie, a następnie komunikacja z kierownikami zespołów, ocena ich pracy i zatwierdzanie wykonania zadań. Ponadto w oparciu o zgłoszenia i predyspozycje przydziela członków grupy do różnych zespołów i zarządza zmianami. Stara się identyfikować różne rodzaje ryzyka w projekcie i im przeciwdziałać. Kierownik motywuje uczestników do efektownej pracy i rozwiązuje ewentualne konflikty wewnątrz zespołu.

Kierownik zespołu – Jego rolą jest kierowanie działaniami konkretnego zespołu. Zajmuje się podziałem pracy i ustalaniem terminów wykonania przydzielonych zadań. Wspiera członków zespołu w realizowaniu powierzonych celów i ocenia, jakość wykonanej przez nich pracy.

Uczestnik projektu / zespołu – Jego głównym zadaniem jest realizacja prac związanych ze zleconymi mu zadaniami. Zgłasza on też wszelkie przypadki ryzyka i problemów mogących prowadzić do niewykonania zadania lub jego niewłaściwej realizacji. W razie wystąpienia możliwych opóźnień w wykonaniu pracy sygnalizuje to kierownikowi swojego zespołu w celu otrzymania od niego wsparcia.

Zespół analizy - ustalenie wszystkich tych czynników, które mogą wpłynąć na przebieg procesu projektowego i implementacyjnego. Zadaniem członków tego zespołu jest spisanie wymagań klienta, które musi spełniać oprogramowanie. Następnie opisanie wszystkich możliwych przypadków użycia i zbudowanie na ich podstawie logicznego modelu systemu. Musi on opisywać sposób realizacji postawionych wymagań abstrahując od szczegółów implementacyjnych. Stworzenie diagramów zgodnie z notacją UML: diagramu przypadków użycia, diagramu klas.

Zespół projektowania – Zadaniem tego zespołu jest transformacja wymagań na postać implementowalną. Zespół musi uwzględnić szczegóły implementacyjne takie jak: hardware, system operacyjny, język programowania, biblioteki i narzędzia programistyczne. Jego członkowie są odpowiedzialni za zaprojektowanie graficznego interfejsu użytkownika i stworzenie grafiki, wykorzystanej w programie. Muszą też stworzyć projekt bazy danych i fizycznej struktury systemu.

Zespół implementacji – Stworzenie kodu i implementacja zgodnie z założeniami zespołu analizy i wypracowanymi rozwiązaniami zespołu projektowego.

Zespół testowania – Członkowie tego zespołu zajmują się testowaniem wszystkich możliwych przypadków użycia systemu i zgłaszaniem wykrytych w trakcie błędów. Mają oni weryfikować oprogramowanie, czyli sprawdzać, czy wytwarzane oprogramowanie jest zgodne ze specyfikacją. Drugim aspektem ich pracy jest walidacja polegająca na sprawdzaniu, zgodności z oczekiwaniami użytkownika.

Zespół dokumentacji – odpowiedzialny za napisanie pełnej dokumentacji (spis treści, słownik pojęć, indeksy itp.) do pracy inżynierskiej we współpracy z wszystkimi pozostałymi zespołami. Jego członkowie zajmują się zbieraniem materiałów, redakcją i korektą językową otrzymanych tekstów. Utrzymują jednolitą konwencję nazw w projekcie.

2. Aplikacje wsparcia dla usług i urządzeń

Celem rozdziału jest przedstawienie funkcjonalności i szczegółowe opisanie aplikacji konkurencyjnych względem naszego projektu inżynierskiego "Serwis".

2.1. Wprowadzenie do problematyki serwisowania

Serwis może być rozumiany na wiele sposobów - jest to zagadnienie wymagające klaryfikacji. Każda firma inaczej rozumie serwis. Przykładowo, może to być obsługa sprzętu komputerowego, jaki posiada obecnie na stanie, czyli wspieranie działów odpowiedzialnych za inwentaryzację i zarządzenie sprzętem komputerowym.

Każde oprogramowanie tego typu powinno posiadać moduł *helpdesk*, czyli obsługę zgłoszeń, nazywanych również tercetami. Aplikacja wspierająca usługi tego typu powinna integrować w sobie funkcjonalności, umożliwiające zgłaszanie awarii w miarę łatwy sposób i możliwie szeroko definiowalny. Elastyczność jest w stanie zapewnić *workflow*, czyli możliwość ustalenia z góry, w jaki sposób zostanie obsłużone dane zgłoszenie. Należy tutaj uwzględnić zarówno negatywne jak i pozytywne przebiegi zdarzeń.

Oprogramowanie opracowywane w ramach tego projektu inżynierskiego zawiera wszystkie wyżej wymienione funkcjonalności.

2.1.1. Czym jest workflow?

Workflow (z ang. *workflow* – przepływ pracy) jest to wcześniej zdefiniowany sposób przepływu informacji pomiędzy różnymi uczestnikami, którzy biorą udział w procesie przetwarzania tych informacji.

Pojęcie *workflow* używa się często odnosząc do oprogramowania wspomagającego pracę w grupach. Pozwala ono przypisać role osobom uczestniczącym w przetwarzaniu dokumentów oraz pozwala na określenie stanów pośrednich dokumentów. Przepływ pracy pokazany jest na rysunku 90.

2.1.2. Cechy systemów do obsługi zgłoszeń z workflow

Systemy *workflow* mają za zadanie wspieranie pracy grupowej w przedsiębiorstwach. Wszelkie informacje, zadania lub dokumenty przepływają do kolejnych uczestników procesów według wcześniej zdefiniowanych procedur. Każdy uczestnik ma określoną rolę oraz przypisanie czynności, które może wykonać w określonym procesie.

Do większości systemów dostęp jest możliwy poprzez przeglądarkę stron WWW, dzięki czemu zyskujemy bardzo łatwy dostęp do obiegu informacji zarówno wewnątrz przedsiębiorstwa, jak i pomiędzy pracownikami współpracujących firm, czy też między pracownikami a klientami praktycznie z dowolnego miejsca na świecie, z którego jest dostęp do Internetu.

System powinien umożliwiać definiowanie dostępności do treści dla zarejestrowanych użytkowników lub też dla gości, czyniąc je poufnymi lub też dostępnymi publicznie. Administrator aplikacji może nadawać uprawnienia dostępu do danych użytkownikom lub też grupom użytkowników.

Do obsługi tego typu systemów nie jest wymagana wiedza informatyczna. Spójna i jednolita platforma, ten sam sposób realizowania dostępu do wszystkich rodzajów informacji ma zapewnić prostą i intuicyjną obsługę.

Aplikacje umożliwiają definiowanie alarmów, przypomnień, powiadomień e-mail lub SMS i ich automatyczne bądź ręczne wysyłanie oraz generowanie raportów, które pozwalają na bieżąco sprawdzać istotne parametry działalności.

Narzędzia pozwalają na importowanie i eksportowanie danych z oraz do zewnętrznych systemów z plików xls/csv. Możliwe jest także dołączanie załączników do informacji (wszelkiego rodzaju dokumentów, zdjęć, rysunków itp.).

Każda informacja przechowywana w systemie posiada historię zmian, dzięki czemu zawsze mamy możliwość określenia, kto i jakich zmian dokonywał.

Poszczególne systemy mogą też występować w różnych wersjach. Użytkownik może wybrać podstawową bardziej ekonomiczną wersje lub, co wiąże się z dodatkowymi kosztami, rozbudowaną o dodatkowe funkcjonalności, dzięki czemu każdy może dopasować wybieraną wersje aplikacji do swoich potrzeb oraz możliwości.

Producenci aplikacji gwarantują ciągły rozwój swoich produktów, dostosowywanie się do ewoluującej rzeczywistości oraz zmiennych w czasie potrzeb swoich klientów. Klient zyskuje dostęp do aktualizacji systemów.

2.2. Service Desk - narzędzie wspierające obszar eksploatacji usług ITIL

W związku z podjęciem decyzji o realizacji systemu obsługi zgłoszeń w naturalny sposób pojawiła się potrzeba, aby budowany system implementował pojęcia oraz dobre praktyki ITIL. Kluczowe w procesie analizy systemu było jasne zdefiniowanie pojęć oraz poruszanie się w obszarze procesów, zwłaszcza występujących w ramach cyklu eksploatacji usług. Wiedza ta pozwoliła jednoznacznie zdefiniować czym jest usługa oraz jakie elementy składowe posiada, co pozwoliło w sposób bardziej zrozumiały spojrzeć zespołowi na system Service Desk, jako system obsługi zgłoszeń dla usług świadczonych odbiorcom.

2.2.1. Czym jest ITIL

ITIL to standard zarządzania usługami informatycznymi, który przez ostatnie kilkanaście lat stał się powszechny na całym świecie. Znajomość i stosowanie dobrych praktyk pozwala eksploatować oraz utrzymywać usługi na zadawalającym poziomie. Wprowadzenie modelu usługowego pozwala zrozumieć organizacji IT świadczącej usługe informatyczna swoja role oraz nawiązać dialog w biznesie. W Polsce model usługowy w branży informatycznej staje się coraz bardziej popularny i śladami rynków bardziej dojrzałych jak Zachodnia Europa jest coraz częściej szeroko stosowany zarówno w środowisku komercyjnym jak i w administracji. W latach osiemdziesiątych w Wielkiej Brytanii rozpoczęto prace dotyczące standardu ITIL. Pierwszą wersje zbioru dokumentów ITIL nazywano GITIM, Government Information Technology Infrastructure Management. Dokumenty te określały standardy, tzw. framework dostarczania do organizacji rządowych usług IT i pomimo, że w dzisiejszych czasach ITIL znacznie ewoluował to koncepcyjnie jest podobny i skupia się na właściwej eksploatacji i utrzymaniu usług. Pierwsza publikacja ukazała się w 1989 roku pod nazwa "Help Desk". Zaproponowany standard został szybko zaadoptowany w Europie na początku 1990 roku i jest wykorzystywany zarówno w rządowych i poza rządowych dużych instytucjach. ITIL jest zarejestrowanym znakiem towarowym OGC. Kolejno w 2001 oraz 2007 roku powstały druga i trzecia wersja biblioteki ITIL. Na pierwszą składały się dwie główne publikacje, natomiast ITIL® V3 to 5 publikacji pokazujący wszystkie fazy życia usługi oraz ich procesy. Z końcem roku 2007 powstała aktualizacja tego standardu, a bieżąca premiera publikacji 2011 Edition przypadła na 29 lipca 2011 roku. W tym dniu zostały opublikowane zaktualizowane wszystkie 5 publikacji opisujące wszystkie fazy cyklu życia usługi:

- Strategia usług,
- Projektowanie usług,

- Przekazanie usług,
- Eksploatacja usług,
- Ustawiczne doskonalenie usług.

Głównym celem ITIL jest określenie i usprawnianie procesów występujących w kolejnych etapach życia usługi świadczonych przez organizację IT. Pełny cykl życia usługi wraz z procesami zarządzania opisane w ITIL® 2011 Edition przedstawia Rysunek 2.



Rysunek 2. Cykl życia usługi. Źródło: [1]

Wdrożenie w narzędziu cech ITIL pozwoliło we wczesnym etapie prac wprowadzić jednolitą terminologię i uporządkować sposób działania oprogramowania, aby jak najlepiej wspierało ono wybrane procesy fazy eksploatacji świadczonych usług w ramach organizacji.

2.2.2. Podstawowe pojęcia ITIL

Wprowadzenie jednolitego oraz precyzyjnego słownika pojęć między światem biznesu, a światem IT to jak wspomniano jedna z korzyści, jaka płynie z ITIL. Aby można było omówić sposób wdrożenia ITIL przy wykorzystaniu oprogramowania *Service Desk* konieczne jest przedstawienie podstawowych terminów z wybranych obszarów biblioteki ITIL.[2]

- 1. **Dostawca usług** używany jest często, jako skrót określający dostawcę usług informatycznych, który świadczy usługi informatyczne odbiorcom zewnętrznym i wewnętrznym.
- Incydent niezaplanowana przerwa działania usługi lub obniżenie jakości usługi informatycznej. Incydentem może być także awaria danego elementu usługi, który jeszcze nie wpłynął na przerwę w działaniu usługi. Przykładem może być awaria jednego noda w niezawodnościowym klastrze bazodanowym.
- 3. **Kategoria** używana jest do łączenia razem spraw. Jest to nazwana grupa czegoś, co ma wspólną cechę. Na przykład grupowanie incydentów o podobnym typie, jako kategoria incydentów, czy kategoria kosztów, jako grupa kosztów podobnego typu.
- 4. **Odbiorca/Klient** zakupuje świadczone usługi od dostawcy. Dla dostawcy usług klienci definiują oraz uzgadniają docelowy poziom świadczonych usług SLA.

- 5. **Problem** źródło przynajmniej jednego incydentu, którego przyczyna jest nie zawsze znana w chwili jego rejestracji. Dalsza analiza problemu jest realizowana przez proces zarządzania problemami.
- 6. **Proces biznesowy** należący do organizacji ma na celu ułatwić dostarczenie usługi lub produktu odbiorcy końcowemu. W praktyce wiele procesów bazuje opierając się właśnie o usługi informatyczne. Proces zakupu usługi lub towaru przez detalistę pozwalający mu świadczyć usługę odbiorcy końcowemu może stanowić przykład procesu biznesowego.
- 7. **Typ zgłoszenia** kategoria, która ma na celu podzielić rodzaje składanych zgłoszeń do centrum obsługi użytkowników *Service Desk*. Typowe zgłoszenia to: incydent, wniosek o usługę oraz reklamacja.
- 8. Usługa informatyczna jest to usługa świadczona przez dostawcę usług informatycznych, na rzecz jednego lub wielu odbiorców. Na usługę składa się połączenie technologii informatycznych, ludzi i procesów, które wspierają procesy biznesowe danego odbiorcy. Poziom świadczenia usługi dla procesu biznesowego powinien zostać zdefiniowany w umowie *SLA* (*Service Level Agrement*).
- 9. Wniosek o usługę jest prośbą składaną przez użytkownika do Service Desk i zarządzaną w ramach procesu realizacji wniosków. Przykładem takiej prośby może być wniosek o zmianę hasła, prośba o poradę czy udzielenie informacji. Często w sytuacji, gdy wymagana jest zmiana, wniosek o usługę powiązany jest z wnioskiem o zmianę (*RFC request for change*).
- 10. **Zarządzanie incydentem** oznacza jak najszybsze przewrócenie usługi do normalnej eksploatacji i organicznie niekorzystnego wpływu incydentu.
- 11. **Zarządzanie zdarzeniem** obejmuje cały okres ich eksploatacji, w ramach procesu można wyróżnić etapy: wystąpienie zdarzenia, wykrycie, filtrowanie, wykonanie czynności, jeśli są konieczne, analiza oraz zamknięcie.
- 12. Zdarzenie dowolne wydarzenie, które może wpływać na proces świadczenia usług informatycznych oraz możliwych odchyleń na usługi.
- 13. **Zgłoszenie** to rozmowa telefoniczna pomiędzy użytkownikiem, a centrum obsługi użytkowników. W efekcie przeprowadzonej rozmowy konsultant może zarejestrować w systemie incydent lub wniosek o usługę.
- 14. **Znany problem** jest to problem, dla którego znane jest obejście i przyczyna wystąpienia. Znane problemy mogą być również wykryte przez samych programistów lub dostawców.

2.2.3. Wdrożenie Service Desk, jako element dobrej praktyki ITIL

Ustanowienie procesu Zarządzania Poziomem Usług (*Service Level Management*) mającego na celu podpisanie możliwych do realizacji umów SLA oraz wdrożenie *Service Desk* przy pomocy zaprojektowanego oprogramowania i zapewnienie, że są one realizowane na zagwarantowanym poziomie może stanowić pierwszy krok do wdrożenia ITIL w organizacji. Jednak, aby można było mówić o samej fazie eksploatacji usług niezbędne jest posiadanie wiedzy o świadczonych usługach. Katalog Usług tworzony w fazie projektowania usług stanowi bazę danych na temat działający usług w organizacji. *Service Desk* i Zarządzanie Poziomem Usług są punktami styku miedzy klientami, a dostawcą, dlatego zdefiniowanie spójnego źródła informacji o świadczonych usługach oraz ich poziomie przez dostawcę jest szczególnie ważne z punktu widzenia ITIL.

Dalsze omówienie procesu Zarządzanie poziomem usług ma na celu przybliżyć kwestie związane z umowami SLA, a rozpoczęcie od zagadnień związanych z fazą projektowania jak właściwe zaprojektowanie Katalogu usług zaprezentowano, jako przepis na identyfikację usługi zgodnie z modelem ITIL. Powyższe zagadnienie zostaną omówione na przykładzie wdrożenia usługi *Service Desk*, jako centralnego punktu obsługi incydentów i zapytań dla usługi 'zarządzanie komputerami osobistymi.



Rysunek 3. Planowanie katalogu usług. Źródło: [3]

W pierwszym kroku wdrożenia musi nastąpić identyfikacja usług składających się na Katalogu usług. Katalog usług stanowi dokument lub bazę danych zawierającą ustrukturyzowaną treść o wszystkich produkcyjnych usługach informatycznych. Na skład portfela usług, który prezentowany jest odbiorcom końcowy składa się m.in. katalog usług określający zawarte kontrakty na świadczenie usług i informacje dotyczące sposobu ich świadczenia, np. zakresu obsługi, czy zgłaszania problemów z jej świadczeniem. W ramach Katalogu usług wchodzą wyłącznie usługi aktywne, czyli określone na etapie Eksploatacji usług, które dana organizacja obecnie oferuje oraz te usługi, które zostały nowowprowadzone dla klientów.



Rysunek 4. Katalog usług. Źródło: [3]

W trakcie identyfikowania usług nie interesują nas aspekty techniczne niezbędne do ich świadczenia, a skupiamy się jedynie na samym zidentyfikowaniu usług, jakie organizacja IT będzie świadczyć. Poniżej znajduje się przykładowy Katalog usług, który został wdrożony w ramach przebiegów *workflow* w naszej aplikacji realizującej funkcję *Service Desk*.

- 1. Internet (workflow Brak Internetu)
- 2. Zarządzanie komputerami osobistymi (workflow: Monitor nie działa)
- 3. Zarządzanie aplikacją (workflow Program nie działa)
- 4. Samochody (workflow Samochód się nie uruchamia)

5. Zarządzanie drukarkami (workflow Nie działa drukarka sieciowa)

Każda wymieniona usługa z katalogu powinna zostać szczegółowo opisana. Dlatego kolejnym krokiem rozwoju Katalogu usług jest zwiększanie stopnia szczegółów dotyczących opisu każdej świadczonej usługi.

Tabela 1. Przykład usługi zarządzanie komputerami osobistymi w Katalogu usług. Źródło: [4]

Nazwa usługi:	Zarządzanie komputerami osobistymi	
Właściciel usługi:	Firma MySupport świadczy usługę dla firmy NextGeneration.	
Odbiorca:	Wszyscy pracownicy firmy NextGeneration korzystający z komputerów osobistych RVT.	
Opis usługi:	Organizacja MySupport zapewnia użytkownikom kompleksowe wsparcie w ramach wykorzystania komputerów osobistych RVT w firmie NextGeneration.	

Wsparcie obejmuje rozwiązywanie bieżących Incydentów oraz Wniosków, zgłaszanych za pośrednictwem *Service Desk*:

- dostarczanie, podłączenie, wymiana, konserwacja i naprawy komputerów osobistych, monitorów i laptopów,
- instalacje i aktualizacje systemu operacyjnego,
- instalacje i aktualizacje oprogramowania antywirusowego,
- instalacje i aktualizacje oprogramowania biurowego (w tym pakietu biurowego oraz przeglądarki internetowej),
- zmiany w konfiguracji komputerów i oprogramowania pozwalające na prace z reszta infrastruktury,
- rozwiązywanie złożonych problemów.

W zależności od możliwości technicznych i decyzji pracownika MySupport ww. aktywności firma może przeprowadzać u klienta lub zdalnie. Dopuszcza się również za aprobatą Klienta przeprowadzanie działań zapobiegawczych.

Wykluczenia z usługi:	Usługa nie dotyczy zarządzania kolejkami drukarek,	
Godziny świadczenia usługi	Dni robocze 7:00 - 16:00. Użytkownicy laptopów także 10:00 - 18:00 w weekendy.	
Poziom świadczenia usług:	Osiemdziesiąt procent zgłoszeń dotyczących obsługi komputerów osobistych o priorytecie krytyczne będzie obsłużonych od momentu zgłoszenia w ciągu trzech godzin roboczych, pozostałe dwadzieścia procent musi zostać zrealizowane w przeciągu sześciu godzin	
Sposób przyjmowania	Zgłaszanie incydentów może odbywać się wyłącznie przez Service Desk, a zgłoszenia rejestrowane są wyłącznie elektronicznie. Dopuszcza się	

zgłoszeń:	zgłoszenia telefoniczne, które analogicznie będą zarejestrowane w	
	systemie informatycznym.	

Umowa SLA o możliwym do zagwarantowania poziomie świadczenia usługi jest negocjowana w ramach realizacji procesu Zarządzania Poziomem Świadczenia Usługi. Ma on również na celu zapewnić, że uzgodnione kontrakty SLA są realizowane. Ponieważ na możliwość realizacji usługi przez dostawcę na zadanym poziomie SLA wpływa wiele dodatkowych czynników proces ten odpowiada także, aby wszystkie procesy zarządzania usługami informatycznymi (ITSM, *IT Service Management*), umowy o gwarantowanym poziomie wsparcia (OLA, *Operational Level Agreement*) oraz kontrakty z dostawcami zewnętrznymi (UC, *Underpinning Contract*) były zgodne z wynegocjowanymi docelowymi poziomami świadczenia usług.

W ramach projektu przyjęto uproszczenie SLM zakładając, że wprowadzona do systemu umowa zawiera SLA na jedną usługę informatyczną, dodatkowo przyjęto, że dana usługa może być związana z jednym dostawcą zewnętrznym. W przedstawionym przykładzie umowa SLA pomiędzy MySupport, a NextGeneration zawarta została na jedną usługę zarządzanie komputerami osobistymi oraz dotyczy urządzeń jednego producenta zewnętrznego, firmy RVT dostarczającej numery seryjne urządzeń. Firma MySupport w celu zapewnienie właściwego poziomu świadczenia usługi może posiadać dodatkowy kontrakt z dostawcą zewnętrznym RVT, np. na dostarczenie nowych części lub urządzeń.

Katalog usług rozwijany jest wraz z kolejnymi umowami podpisanymi z klientem, które wraz z SLA są wprowadzane do systemu Service Desk.. Dzięki katalogowi usług możemy określić pracochłonność, co ma wpływ na koszt świadczonej usługi. Monitorując gwarantowany poziom świadczenia usługi otrzymujemy wiedzę nt. pracy, jaką ponosi organizacja świadcząc daną usługę dla poszczególnych klientów. Katalog usług jest podstawowym elementem dla procesów ITIL, a poprawnie wdrożony Service Desk stanowi klucz do właściwego świadczenia usług przez firmy outsourcingowe. Pełną umowę między klientem, a dostawcą usługi otrzymujemy po zakończeniu określania okien serwisowych oraz poziomu dostępności usług i uzupełnieniu Katalogu usług. Ustrukturyzowana forma Katalogu usług pozwala na określenie Jednostki Kosztu (CU, Cost Unit), pozwalającej na rozliczenia się w stosunku do realizacji danej usługi np. liczba utworzonych użytkowników w domenie na podstawie złożonych wniosków. Ważnym elementem jest wprowadzenie priorytetów w ramach gwarantowanego poziomu świadczonych usług na etapie specyfikowania warunków oraz przedziałów czasowych ich realizacji, np. "osiemdziesiat procent zgłoszeń dotyczących obsługi komputerów osobistych o priorytecie krytyczne będzie obsłużonych od momentu zgłoszenia w ciągu trzech godzin roboczych, pozostałe dwadzieścia procent musi zostać zrealizowane w przeciągu sześciu godzin". [5]

Service Desk pozwala zarządzać usługami IT w sposób kontrolowany oraz wydajny. Centralna ewidencja i obsługa zgłoszeń oraz możliwe dążenie do szybkiego ich zamykania, często w ramach pierwszej linii wsparcia wpływa znacząco na sposób postrzegania świadczonej usługi oraz większe zadowolenie klientów. Coraz więcej organizacji postrzega Service Desk, jako kluczową funkcję w procesie biznesowym i stara się, aby była ona dostępna bez przerwy. Niedostrzeganie kluczowej roli, jaką odgrywa Service Desk może powodować obniżenie, jakości świadczonych usług. Service Desk najczęściej realizowany jest w jednym z trzech modeli: Lokalny Service Desk, Centralny Service Desk, Wirtualny Service Desk.



Rysunek 5. Centralny Service Desk. Źródło: [6]

Oprogramowanie *Service Desk* wspiera realizację funkcji *Service Desk* zdefiniowaną w bibliotece dobrych praktyk ITIL pozwalając rejestrować oraz zarządzać incydentami oraz wnioskami o usługę. Właściwie wdrożenie wprowadza także pojedynczy punkt kontaktu między użytkownikiem usługi, a jej dostawcą

Zgodnie z ITIL Service Desk spełnia poniższe zadania: [7]

- przyjmowanie oraz rejestrowanie zgłoszeń,
- bezzwłoczne rozwiązywanie prostych incydentów i wniosków,
- podjęcie próby rozwiązania w ramach pierwszej linii, zebranie niezbędnych informacji do dalszej analizy, ewentualne przekazanie ich do kolejnej grupy wsparcie zgodnie z procedurą – zdefiniowaną w ramach *workflow*,
- eskalacja zgodnie z przyjętą polityką (pionowa / pozioma) oraz monitorowanie realizacji prac nad incydentem,
- przekazywanie stanu realizacji zgłoszenia użytkownikom,
- opracowywanie wykresów, raportów, itp. dla kierownictwa IT z realizacji.

Zatem możemy powiedzieć, że korzyści płynące z tego rozwiązania są następujące:

- dobrze funkcjonujący *Service Desk* to większa, jakości realizowanych usług IT co przekłada się na satysfakcję użytkowników,
- poprawa dostępności usług dla klientów dzięki udostępnionemu jednemu punktowi kontaktu,
- wyspecjalizowani pracownicy *Service Desk* to szybsze, bardziej profesjonalne rozwiązywanie zgłoszeń, według ITIL nawet 80% powinno być rozwiązywane przez *Service Desk*,
- wspólna praca w zespołach wsparcia nad zgłoszeniami oraz poprawa komunikacji,
- optymalnie kierowana i nadzorowana infrastruktura informatyczna,
- lepsze wykorzystanie zasobów: ludzi, sprzętu, oprogramowania,
- lepsze zarządzanie informacją.

2.2.4. Wybrane punkty zgodność narzędzia Service Desk ze standardem ITIL

Zgodnie z założeniem projektu głównym celem nie jest pełna zgodność z koncepcją ITIL oraz certyfikacja narzędzia, a osiągnięcie funkcjonalności obsługi zgłoszeń serwisowych produktów oraz usług. System powinien nawiązywać do zbioru dobrych praktyk ITIL, dzięki czemu odbiór wytworzonego oprogramowania dla użytkownika końcowego powinien być prostszy dzięki naśladowaniu uznanych standardów. Poniżej zostały przedstawione wybrane punkty zgodności aplikacji Service Desk z biblioteką ITIL w obszarze zarządzania incydentami oraz zarządzania wnioskami fazy eksploatacji usług.

Punkty zgodności w zakresie zarządzania incydentami (Incident Management) [8]

Podstawowy cel procesu zarządzanie incydentami (*Incident Management*) to odtworzenie działania usługi klientowi w maksymalnie krótkim czasie. Zarządzanie incydentami, więc powinno minimalizować wpływ na biznes. Incydenty są zgłaszane przez użytkowników *Service Desk* lub raportowane przez proces Zarządzania zdarzeniami (*Event Management*). Pracownicy *Service Desk* w ramach procesu Zarządzanie incydentami stosując obejścia (*workarounds*) w sposób skuteczny mogą wyeliminować znaczną część objawów awarii. Dalsza analiza w ramach procesu Zarządzanie problemami pozwala zająć się obsługą faktycznych przyczyn awarii. Poniżej zostały wymienione wybrane punkty zgodności w zakresie zarządzania incydentami dla *Service Desk*.

- wykorzystuje terminy ITIL takie jak incydent czy wniosek
- pozwala na kontrolę dostępu dotyczącą otwierania, modyfikowania i zamykania incydentów na z góry ustalonych warunkach, warunki te mogą być zdefiniowane m.in. przez *workflow*
- pozwala określić pola formularza incydentu, które będę wymagane przy jego obsłudze,
- incydenty są rejestrowane w systemie ręcznie przez konsultanta,
- dla każdego incydentu automatycznie przydzielany jest unikatowy identyfikator,
- w trakcie rejestracji zapisywana jest data incydentu oraz przy każdej jego zmianie data aktualizacji,
- w ramach rejestracji odnotowywana jest osoba zgłaszająca oraz sposób kontaktu: telefon, email,
- formularz rejestracji incydentu zawiera pole umożliwiające opis symptomów usterki,
- każdy zarejestrowany incydent zawiera pole opisujące jego status takie jak aktywny, oczekujący, zamknięty,
- stan incydentu w określonych okolicznościach może zostać zmieniony ręcznie,
- kluczowe atrybuty incydentu takie jak status, priorytet, przypisanie do kolejki mogą być ustawione wyłącznie przez *Service Desk*,
- wszyscy autoryzowani użytkownicy posiadają dostęp do podstawowych informacji incydentu jak status, priorytet, znacznik czasu

Punkty zgodności w zakresie realizacji wniosków (Request Fulfillment) [9]

Główny cel procesu realizacji wniosków (*Request Fulfillment*) to zagwarantowanie użytkownikom dostępu do udostępnianych standardowych usług informatycznych wraz z dostarczenie jednolitego ustandaryzowanego sposobu ich wnioskowania. Proces ten zarządza etapami wszystkich wniosków w tym również skargami. Podstawowe zrealizowane punkty zgodności w ramach *Service Desk* to m.in.

• rozwiązanie pozwala na kontrolę dostępu dotyczącą otwierania, modyfikowania i zamykania wniosków na z góry ustalonych warunkach, warunki te mogą być zdefiniowane m.in. przez *workflow*,

- rozwiązanie pozwala określić pola formularza wniosku, które będę wymagane przy jego rejestracji,
- rozwiązanie umożliwia generowanie raportów zarządczych z rekordów wniosków historycznych,
- do wniosku automatycznie zapisywana jest data i czas utworzenia/aktualizacji rekordu
- każdy rejestrowany wniosek zawiera pole opisujące jego status takie jak otwarty, prace w toku (WIP), zamknięty,
- formularz rejestracji wniosku umożliwia wprowadzenie szczegółów wniosku takich, jak: kto zgłosił wniosek, do kogo wniosek zostanie przypisany, informację dodatkowe do zamknięcia,
- rozwiązanie umożliwia użytkownikowi obserwację statusu oraz postępu realizacji wniosku,
- rozwiązanie obsługi wniosków jest zintegrowane z rozwiązaniem obsługi incydentów w jednej aplikacji *Service Desk*

2.3. HP Open View Service Desk (OVSD)¹

HP OpenView Service Desk (OVSD) jest oprogramowaniem firmy Hewlett Packard - lidera z branży zaawansowanych technologii. Service Desk należy do grupy produktów OpenView. Jest rozwiązaniem, które służy do zarządzaniem incydentem, zmianą i innymi standardami procesu zarządzania, które są wspólnym punktem do komunikacji między użytkownikami końcowymi a organizacją IT. Aplikacja ta jest oparta na najlepszych praktykach ITIL® Service Desk. Głównymi zaletami takiego podejścia jest redukcja kosztów przez polepszenie skuteczności i efektywności procesów - nadaje się do zastosowania zarówno w średnich jak i dużych przedsiębiorstwach. Standaryzacja i kontrola procesów zmniejsza ryzyko oraz zapewnia dostosowanie do regulacji prawnych oraz wewnętrznych procedur organizacji, co korzystnie wpływa na efektywne i terminowe dostarczanie usług. Częścią tego rozwiązania jest katalog usług oraz portal dla użytkownika końcowego, które pomagają zautomatyzować i przyspieszyć zwykłe zadania z zakresu zarządzania usługami informatycznymi, w tym zarządzanie incydentem i wnioskiem. [10]

2.3.1. Wymagania funkcjonalne

- Pracownicy mogą samodzielnie rejestrować, klasyfikować i przeglądać zgłoszenia incydentów lub wniosków, dotyczących usług świadczonych przez dział informatyki, po zalogowaniu się do intranetowego systemu rejestracji zgłoszeń udostępnionego na portalu.
- Zgłoszenia w systemie mogą być rejestrowane tylko przez uprawnionych użytkowników posiadających w systemie konto, utworzone wcześniej przez administratora.
- Użytkownik może wyszukiwać w systemie osoby, dodawać do zgłoszenia załączniki, wyszukiwać zgłoszenia oraz sprawdzać ich status, wyświetlać i drukować listę usług.
- System umożliwia zmianę hasła lub wygenerowanie nowego, jeśli użytkownik zapomniał nie pamięta swojego hasła.
- System wylogowuje użytkownika z aplikacji, jeśli nie wykryje żadnej aktywności (po 21 minutach).
- System wysyła do użytkowników emaile o zdefiniowanej wcześniej treści.

¹ Ze względu na poufność informacji nie podano, w jakiej organizacji zostało zaimplementowane powyższe rozwiązanie. Poniższy opis rozwiązania został stworzony na podstawie dokumentu "Podręcznik użytkownika HP

2.3.2. Rejestracja zgłoszenia.

Po zalogowaniu do aplikacji należy wypełnić formatkę Nowe Zgłoszenie.

Wypeł-Lista **Obowią**niane Zawartość Nazwa pola wartości zkowe automatycznie Dotyczy osoby / Dane osoby/organizacji, której dotyczy zgłoszenie. Dotyczy Dane osoby muszą znajdować się w bazie osób Т Т uprawnionych do rejestracji. Zgłaszajacy Dane osoby/organizacji zgłaszającej. Dane Т zgłaszającego muszą znajdować się w bazie osób uprawnionych do rejestracji. Osoba do kontaktu Nazwisko i imię osoby wskazanej do kontaktu. Dane osoby wybierane są z bazy osób uprawnionych do rejestracji. Т Jeżeli zgłoszenie rejestrowane jest przez użytkownika spoza działu, dane Osoby do kontaktu należy wpisać "ręcznie". Adres e-mail Adres e-mail, na który zostanie dodatkowo wysłane powiadomienia o stanie realizacji zgłoszenia. Pokój (pole Nr pokoju osoby, której dotyczy zgłoszenie. Т dostępne dla pracowników) Telefon Nr telefonu pod którym można uzyskać dodatkowe informacje do zgłoszenia. Usługa / Typ Nazwa usługi, której dotyczy zgłoszenie. Lista Т Т zgłoszenia wartości uzależniona jest od Zgłaszającego. Klasyfikacja Klasyfikacja usługi - w liście wyświetlane sa Т Т wartości dedykowane dla wybranego Typu Zgłoszenia Temat Krótki opis najlepiej identyfikujący zagadnienia. Т Opis Szczegółowy opis zawierający wszystkie istotne Т informacje. Pole ograniczone do 1000 znaków.

Tabela 2. Opis pól formatki Nowe zgłoszenie. Źródło: Opracowanie własne.

Tabela 3. Opis pozostałych pól (nie wypełnianych przez użytkownika). Źródło: Opracowanie własne.

Nazwa pola	Zawartość
Numer zgłoszenia (ID)	Generowany automatycznie unikalny numer zgłoszenia w systemie.
Zarejestrowano	Data i godzina zarejestrowania zgłoszenia.

Status	Status zgłoszenia opisuje, na jakim etapie obsługi jest zgłoszenie. Pole to może przyjmować następujące wartości:		
	• Zarejestrowane – status nadawany automatycznie w momencie zarejestrowania zgłoszenia		
	• <i>Przekazane</i> - zgłoszenie przekazane do realizacji do osoby lub grupy wsparcia (kilka osób)		
	• <i>W trakcie realizacji</i> – zgłoszenie podjęte do realizacji przez osobę		
	• <i>Wstrzymane</i> – wstrzymana realizacja zgłoszenia. Status jest nadawany, gdy do rozwiązania zgłoszenia niezbędne są dodatkowe informacje od zgłaszającego (np. zrzut z ekranu, wynik skryptu) lub gdy rozwiązanie zgłoszenia zostało przekazane poza Departament Informatyki		
	• <i>Rozwiązane</i> – zgłoszenie rozwiązane		
	• <i>Zamknięte</i> – zgłoszenie jest automatycznie zamykane po upływie zdefiniowanego czasu oczekiwania na reklamację		
	 <i>Reklamowane</i> – jeżeli użytkownik zgłosi nieprawidłowości w rozwiązaniu. 		
Powód wstrzymania	Pole jest wypełniane, jeżeli <i>Status</i> zgłoszenia ma wartość <i>Wstrzymane</i> . Dopuszczalne wartości:		
	 Oczekiwanie na informację – jeżeli opis zgłoszenia jest niewystarczający do rozwiązania i konieczne jest pozyskanie dodatkowych informacji od Zgłaszającego np. dodatkowego opisu, komunikatu błędu, wyniku skryptu 		
	• <i>Przekazane na zewnątrz</i> - jeżeli konieczne jest przekazanie zgłoszenia poza Departament Informatyki np. do serwisu, do departamentu merytorycznego		
	• <i>Przyczyna po stronie użytkownika</i> - jeżeli rozwiązanie zgłoszenia nie jest możliwe z winy użytkownika, np. użytkownik przebywa na urlopie / na zwolnieniu / w delegacji, nie dostarczył wymaganych wniosków lub na dostarczonych wnioskach brak wymaganych podpisów.		
Kod zamknięcia	Informacja o sposobie rozwiązania zgłoszenia. Kod zamknięcia jest nadawany		
	przy rozwiązywaniu zgłoszenia. Dostępne wartości:		
	• Zrealizowane - podano docelowe rozwiązanie		
	• <i>Odrzucone</i> - zgłoszenie spoza świadczonego zakresu wsparcia		
	• Obejście - zastosowanie tymczasowego rozwiązania, jeżeli nie jest		

znane docelowe rozwiązanie	
łał Zgłoszenie lub stwierdził zanik	
realizacja zgłoszenia np. omyłkowa ko w tym przypadku użytkownik nie nienia o rozwiązaniu zgłoszenia.	
a w działaniu Usługi lub obniżenie jej	
vykonanie świadczenia w ramach nioskiem może być również zapytanie o	

Przy rejestracji szczególną uwagę należy zwrócić na właściwą klasyfikację zgłoszenia, ponieważ może mieć to wpływ na czas realizacji zgłoszenia (poprawna klasyfikacja spowoduje, że zgłoszenie szybciej trafi do właściwej grupy wsparcia, a co za tym idzie – zostanie szybciej rozwiązane). Istotna jest także treść wpisywana do pola *Temat*, ponieważ w dalszej pracy z systemem będzie ono często wykorzystywane, jako klucz przy wyszukiwaniu zgłoszenia. W zależności od wybranego *Typu Zgłoszenia* na formatce mogą pojawić się dodatkowe pola wymuszające wprowadzenie uzupełniających informacji do zgłoszenia, np. nazwy drukarki lub wersji systemu i wersji słowników centralnych. Po wypełnieniu pól obowiązkowych i naciśnięciu klawisza **Wyślij zgłoszenie** lub **Dodaj załącznik**, zgłoszenie zostanie automatycznie zarejestrowane w systemie i zostanie mu nadany unikalny numer identyfikacyjny ID. Do zgłoszenia można w każdym momencie rejestracji lub później np. podczas komunikacji z osobami realizującymi zgłoszenie, dodać załącznik lub uzupełnić opis zgłoszenia. Po zarejestrowaniu zgłoszenia, system automatycznie wysyła mailowe powiadomienie do osób z pól *Dotyczy osoby, Zgłaszający, Osoba do kontaktu, Adres e-mail* oraz na adresy mailowe wymienione w wewnętrznej procedurze organizacji.

2.3.3. Wyszukiwanie osoby

Istnieje możliwość wyszukiwania osób w bazie osób, uprawnionych do rejestracji. W celu wyszukania osoby należy nacisnąć symbol lupy przy jednym z pól: Dotyczy osoby lub Osoba do kontaktu. W nowym oknie, które się pojawi należy wypełnić pola *Login/Nazwa użytkownika* lub *Imię i nazwisko*, nacisnąć klawisz **Znajdź**, wybrać znalezioną osobę lub wybrać jednostkę organizacyjną, w której pracuje szukana osoba, wcisnąć przycisk Znajdź, wybrać osobę z listy osób znalezionych. Imię i Nazwisko wybranej osoby zostanie wpisane odpowiednio do pól *Dotyczy osoby* lub *Osoba do kontaktu*.

2.3.4. Dodawanie załączników

Do jednego zgłoszenia można dodać wiele załączników. Załączniki można dodawać poprzez naciśnięcie klawisza Dodaj załącznik: w trakcie rejestracji zgłoszenia - z pierwszego ekranu formularza *Nowe zgłoszenie*, po zarejestrowaniu zgłoszenia - z formularza *Przeglądaj/edytuj*

zgłoszenie serwisowe pojawiającego się po naciśnięciu klawisza Wyślij zgłoszenie, w trakcie przeglądania zgłoszenia - z widoku *Moje zgłoszenia serwisowe (aktywne)*.

Uzupełnienie opisu zgłoszenia o nową informację należy wprowadzić w pole tekstowe znajdujące się nad klawiszem Dodaj wpis do opisu zgłoszenia. Pole Opis zostanie uzupełnione o dokonany wpis po naciśnięciu klawisza Dodaj wpis do opisu zgłoszenia. Dodatkowo zapisana zostanie dokładna informacja, kiedy (data i godzina) i przez kogo wpis został dokonany. O nowych wpisach do zgłoszenia informowana jest osoba realizująca zgłoszenie - automatycznie wysyłane jest powiadomienie o nowym wpisie do zgłoszenia. Jeśli opis przekroczy 1000 znaków pojawi się komunikat o błędach "Pole Opis nie może zawierać więcej niż 1000 znaków. Należy skrócić treść w polu opisu, a pełny tekst dodać w formie załącznika."

Nie można samodzielnie usuwać wcześniej dodanych załączników, w tym celu należy przesłać dodatkową informację do zgłoszenia z wskazaniem, które pliki są nieaktualne lub pomyłkowo dołączone, wtedy pracownik obsługujący zgłoszenie może je usunąć.

2.3.5. Wyszukiwanie zgłoszeń/wyświetlanie stanu rejestracji zgłoszenia

Zarejestrowane w systemie zgłoszenie użytkownik może wyszukać a następnie obejrzeć wybierając z menu widok *Moje zgłoszenia serwisowe (wszystkie)* lub *Moje zgłoszenia serwisowe (aktywne)*. Lista wyświetlanych zgłoszeń jest ograniczona w zależności od zalogowanego użytkownika zgodnie z wewnętrzną procedurą. Domyślnie system wyświetla zgłoszenia zarejestrowane w ciągu ostatnich dwóch miesięcy (wypełniona data w polu *Zarejestrowano*). Listę wyświetlanych zgłoszeń można dodatkowo zawęzić podając kryteria wyszukiwania zgłoszeń:

- Nr Zgłoszenia podać nr zgłoszenia (ID)
- Dotyczy wyszukać osobę, której dotyczy zgłoszenie
- Zarejestrowano wybrać datę początkową i/lub końcową
- *Typ zgłoszenia* wybrać z listy typ zgłoszenia
- Status wybrać status zgłoszenia
- Temat wpisać cały temat zgłoszenia lub jego część
- Termin realizacji wybrać datę początkową i/lub końcową

Po wypełnieniu parametrów przeszukiwania bazy zgłoszeń, należy nacisnąć klawisz Szukaj. Wyświetlona zostanie lista zgłoszeń zgodna z określonymi parametrami. Listę tę można sortować po każdym z wyświetlanych pól, klikając myszą na nazwie kolumny. Pole, które jest aktualnie kluczem sortowania oznaczone jest znakiem "v" lub "^" w zależności od kierunku sortowania. Otwarcie zgłoszenia do edycji lub przeglądania nastąpi po kliknięciu myszą na wskazanym zgłoszeniu. Zgłoszenia zamknięte dostępne są w trybie przeglądania tzn. nie można dodawać do nich załączników ani dopisywać informacji. Gwiazdka (*) przy statusie zgłoszenia *Rozwiązane* lub *Wstrzymane* oznacza, że odpowiedź na zgłoszenie została już przygotowana, ale jest jeszcze w trakcie weryfikacji przez Pomoc Informatyczną. Dla statusów *Rozwiązane** i *Wstrzymane** pole *Rozwiązanie* nie jest dostępne dla użytkownika.

2.3.6. Hasła

Zmiana hasła następuje po pierwszym zalogowaniu oraz w przypadku ujawnienia swojego hasła użytkownik powinien zmienić hasło na unikatowe. W tym celu należy wybrać z menu opcję **Zmień** *hasło* i wpisać poprawnie dotychczasowe hasło w pole **Aktualne hasło**, następnie uzupełnić pola **Nowe hasło** i **Powtórz hasło.** Po naciśnięciu klawisza Zmień nowe hasło zostanie zapisane w aplikacji. Nowe hasło musi mieć przynajmniej 8 znaków.

W sytuacji, gdy użytkownik zapomniał swoje hasło system generuje nowe hasło i przesyła je użytkownikowi. W tym celu na ekranie logowania należy wybrać opcję: Nie pamiętam hasła. System najpierw poprosi o potwierdzenie tożsamości użytkownika przez podanie adresu e-mail podanego przy pierwszym logowaniu, a następnie na ten adres zostanie automatycznie wysłana wiadomość zawierająca informacje dotyczące użytkownika i jego hasło. Maile zawierają treść zgodnie z szablonem ustalonym w danej organizacji.

2.3.7. Zakończenie pracy

W celu zakończenia pracy z serwisem należy wylogować się wybierając z opcji w menu Wyloguj. Po wylogowaniu system zgłasza się z ekranem logowania (można ponownie się zalogować) lub zamknąć okno przeglądarki internetowej. Po 20 minutach bezczynności system wyświetla komunikat ostrzegający przed zamknięciem sesji. Jeśli w ciągu minuty nie zostanie wciśnięty przycisk ok, sesja zostanie automatycznie zamknięta. W przypadku dalszej pracy należy zalogować się ponownie.

2.3.8. Wyświetlanie listy usług/klasyfikacji

W celu wyświetlenia dostępnej użytkownikowi listy usług lub listy usług poszerzonej o klasyfikacje należy skorzystać odpowiednio z opcji w menu *Lista dostępnych usług* lub *Lista dostępnych usług z klasyfikacją.* Wybraną listę można wydrukować. Drukowanie listy nastąpi po naciśnięciu symbolu drukarki.

2.3.9. E-maile

Wiadomości są wysyłane na podstawie tabeli, w której określona jest treść wiadomości, adresaci oraz warunki wysłania.

Temat	Warunki wysłania	Adresaci
Zgłoszenie nr [ID] zostało zarejestrowane w systemie	 zarejestrowane w systemie Pomocy Informatycznej po rejestracji zgłoszenia gdy: wartość pola Wprowadzone przez osobę inna niż System administrator; wartość pola Zgłaszający wypełnione pole Termin realizacji. 	Osoby wskazane w polach: • Zgłaszający • Dotyczy • Kontakt • Inny e-mail
Informacja o terminie realizacji Zgłoszenia [ID]	 po modyfikacji terminu realizacji zgłoszenia gdy: wartość pola <i>Status</i> jest inna niż <i>Wstrzymane</i> lub <i>Rozwiązane</i> lub <i>Zamknięte</i>. 	Osoby wskazane w polach: • Zgłaszający; • Dotyczy • Kontakt • Inny e-mail
Rozwiązanie zgłoszenia nr [ID] zostało wstrzymane	po rejestracji lub modyfikacji zgłoszenia gdy:	Osoby wskazane w polach: • <i>Zgłaszający</i>

Tabela 4. Warunki wysłania maili. Źródło: Opracowanie własne.

	 wartość pola Status = Wstrzymane wartość pola Weryfikacja jest inna niż Weryfikacja obowiązkowa. 	 Dotyczy Kontakt Inny e-mail
Zgłoszenie nr [ID] otrzymuje status "Reklamowane"	po modyfikacji zgłoszenia gdy: • wartość pola <i>Status</i> = <i>Reklamowane</i> .	Osoby wskazane w polach: • Zgłaszający • Dotyczy • Kontakt • Inny e-mail
Zgłoszenie nr [ID] zostało rozwiązane	 po rozwiązaniu zgłoszenia nie wymagającego weryfikacji gdy: wartość pola <i>Kod zamknięcia</i> inna niż <i>Anulowane</i> lub po rozwiązaniu zgłoszenia wymagającego weryfikacji gdy: wartość pola <i>Kod zamknięcia</i> inna niż <i>Anulowane</i> pole<i>Rozwiązano</i> na zakładce <i>Terminy</i> jest puste. 	Osoby wskazane w polach: • Zgłaszający • Dotyczy • Kontakt • Inny e-mail
Dane konta użytkownika serwisu	po zaznaczeniu na stronie startowej aplikacji: "Nie pamiętam hasła" lub "Nie, jestem nowym użytkownikiem serwisu".	Użytkownik aplikacji.

Tabela 5. Maile do osoby/grupy wsparcia. Źródło: Opracowanie własne.

Temat	Warunki wysłania	Adresaci
Zgłoszenie nr [ID] przydzielono grupie [Grupa Wsparcia]	 po rejestracji lub modyfikacji zgłoszenia gdy: wartość pola <i>Status = Przekazane</i>; pole <i>Przydzielenie Do Osoby</i> jest puste; wypełnione pole <i>Przydzielenie Do Grupy Wsparcia</i>. 	Wszystkie osoby należące do wskazanej grupy wsparcia
Zgłoszenie nr [ID] przydzielono do [Osoby]	 po rejestracji lub modyfikacji zgłoszenia gdy: wartość pola <i>Status</i> równa <i>Przekazane</i>; wypełnione pole <i>Przydzielenie Do Osoby</i>. 	Osoba wskazana w polu • <i>Przydzielenie Do Osoby</i>
Zgłoszenie nr [ID] zostało zareklamowane	po modyfikacji zgłoszenia gdy:	Osoba wskazana w polu:

	• wartość pola <i>Status</i> = <i>Reklamowane</i> .	• Przydzielenie Do osoby
Zgłoszenie nr [ID] zostało zaktualizowane przez Zgłaszającego	 po dodaniu wpisu do pola <i>Opis</i> gdy: wypełnione pole <i>Przydzielenie Do Osoby</i> 	Osoby wskazane w polach: • Przydzielenie Do Osoby • Zgłaszający • Dotyczy • Kontakt • Inny e-mail
Zmiana w Zgłoszeniu [ID]	po modyfikacji zgłoszenia na stronie WWW przez użytkownika, gdy: wartość pola <i>Status</i> jest inna niż <i>Rozwiązane</i>	Osoba zarządzająca aplikacją

Tabela 6. Maile eskalacyjne (eskalacja hierarchiczna). Źródło: Opracowanie własne.

Temat	Warunki wysłania	Adresaci
Zgłoszenie [ID] nie zostało podjęte od 30 minut	 po upływie 30 minut od rejestracji zgłoszenia, gdy zgłoszenie nie zostało przydzielone do realizacji lub po upływie 30 minut od przydzielenia zgłoszenia <i>Do Grupy Wsparcia = Service Desk</i>, gdy zgłoszenie nie zostało rozwiązane i gdy: wartość pola <i>Zgłaszający</i> jest inna niż <i>Helpdesk</i> 	Osoba wskazana w polu <i>Przydzielenie Do Osoby</i>
Usługa - Przekroczono 70% czasu na realizację zgłoszenia nr [ID]	 po upływie 70% czasu na obsługę zgłoszenia, gdy zgłoszenie nie zostało rozwiązane i gdy: wartość pola <i>Status</i> równa <i>Przekazane</i> lub <i>W trakcie realizacji</i> wartość pola <i>Usługa</i> równa <i>Drukowanie</i> lub <i>Poczta elektroniczna</i> lub <i>Stanowisko komputerowe</i> 	Osoba wskazana w polu: • <i>Przydzielenie Do Osoby</i> oraz osoba zarządzająca aplikacją
Usługa - Przekroczono 100% czasu na realizację zgłoszenia nr [ID]	 po upływie 100% czasu na obsługę zgłoszenia, gdy zgłoszenie nie zostało rozwiązane i gdy: wartość pola <i>Status</i> równa <i>Przekazane</i> lub <i>W trakcie realizacji</i> wartość pola <i>Usługa</i> równa <i>Drukowanie</i> lub <i>Poczta elektroniczna</i> lub <i>Stanowisko komputerowe</i> 	Osoba wskazana w polu: Przydzielenie Do Osoby, osoba zarządzająca aplikacją, Adresat e- maila wpisany w pole Do Wiadomości (e-mail dla eskalacji) danej Grupy Wsparcia
Rozwiązanie Zgłoszenia [ID] nie zostało zweryfikowane	po 2 godzinach od rozwiązania zgłoszenia, gdy zgłoszenie nie zostało zweryfikowane i gdy:	Osoba zarządzająca aplikacją

	• wartość pola <i>Weryfikacja</i> = <i>Weryfikacja obowiązkowa</i> .	
Zgłoszenie [ID] jest "Wstrzymane" od 16 godzin	 po 16 godzinach od zmiany wartości pola <i>Status</i> na <i>Wstrzymane</i>, jeżeli zgłoszenie nie zostało rozwiązane i gdy; wartość pola <i>Przydzielenie Do grupy wsparcia</i> jest pusta lub równa <i>Service Desk</i> lub <i>Administrator ServiceDesk</i>. 	Osoba zarządzająca aplikacją
Zmiana w Rozwiązanym Zgłoszeniu [ID] dokonana przez serwis WWW	po dodaniu wpisu do pola <i>Opis</i> gdy: • wartość pola <i>Status = Rozwiązane</i>	Osoba zarządzająca aplikacją
Usługa - Zgłoszenie [ID] jest "Wstrzymane" od 10 dni	po upływie 10 dni od zmiany wartości pola Status na Wstrzymane	Osoba wskazana w polu: <i>Przydzielenie Do Osoby</i>
Usługa - Przekroczono 300% czasu na realizację zgłoszenia nr [ID]	 czasu na realizację zgłoszenia nr [ID] po upływie 300% czasu na obsługę zgłoszenia, gdy zgłoszenie nie zostało rozwiązane i gdy: wartość pola <i>Status</i> równa <i>Przekazane</i> lub <i>W trakcie realizacji</i> 	Osoba wskazana w polu: • <i>Przydzielenie Do Osoby</i> oraz osoba zarządzająca aplikacją, i osoby stojące wyżej hierarchii

Tabela 7. Maile dodatkowe. Źródło: Opracowanie własne.

Temat	Warunki wysłania	Adresaci				
Informacja o zgłoszeniu nr [ID] (dodatkowy mail)	po rejestracji lub modyfikacji zgłoszenia gdy: na zakładce <i>E-maile</i> jest zaznaczone pole <i>Wyślij wiadomość e-mail</i> .	Osoby wskazane w polu: <i>Lista</i> dystrybucyjna				
Usługa - Potencjalny incydent bezpieczeństwa dla zgłoszenia nr [ID]	 incydent bezpieczeństwa dla zgłoszenia nr [ID] po rejestracji lub modyfikacji zgłoszenia gdy: na zakładce <i>Inne</i> wartość pola <i>Incydent bezpieczeństwa</i> równa <i>Do</i> <i>weryfikacji</i>. 	osoba zarządzająca aplikacją				

2.3.10. Wady rozwiązania

Minusem rozwiązania jest staromodny wygląd webowej aplikacji, który prezentuje Rysunek 6. Na dodatek firma Hewlett Packard z dniem 31 grudnia 2012 roku zakończyła wsparcie techniczne dla

	HelpIT syst	tem Rejestracji Z	Zgłoszeń 🗌			
1	Osoba zalogowana:		Jednostka organizacyjna:			
Menu			N	we zgłoszenie		
Nowe zgłoszenie	Dotyczy osoby	Kowalski Jan	ø	Zgłaszający	Kowalski Jan	
Moje zgloszenia serwisowe (wszystkie)	Organizacja Telefon			Organizacja Pokój / Telefon		
Moje zgłoszenia serwisowe (aktywne)	Osoba do kontaktu		p			
Lista dostępnych usług	Pokój *			Telefon		
Lista dostępnych usług z klasyfikacją	Usługa/Typ zgłoszenia *		Q			
Zmień hasło	Klasyfikacja *		Q			
Wyloguj	Temat *					
		Dedictori				
Odnośniki	Wyślij zgłoszenie	Dodaj załącznik			Wyczyść	
erwsze logowanie do HelpIT odręcznik Użytkownika agenda						
o <u>rtal IT</u> Server: (3)						

Rysunek 6. System rejestracji zgłoszeń oparty na HP OpenView Service Desk Źródło: Aplikacja HP OpenView Service Desk

oprogramowania HP OpenView Service Desk [11]. Już dzisiaj powoduje to spore problemy w poprawnym funkcjonowaniu tej aplikacji w systemie Windows 8. Na chwilę obecną nie jest możliwe zakupienie tego oprogramowania, a wsparcie techniczne jest udzielane jedynie dotychczasowym klientom.

2.4. Oracle RightNow

Aplikacja Oracle RightNow posiada bogaty zestaw narzędzi oraz modułów wspierających obsługę i serwis.

Oracle RightNow wywodzi się w prostej linii z *Open Source*'owego oprogramowania rozwijanego w dwóch wersjach. Dwa lata temu RightNow był dostępny w wersji *Community Edition*, gdzie moduły można było stworzyć samodzielnie, ale nie posiadał wsparcia od producenta. Występowała również wersja w pełni komercyjna, posiadająca wsparcie i większe możliwości analityczne, takie jak: automatyzacje zadań oraz dodatki optymalizujące prace. Z racji, iż był to produkt udany i wdrożony w wielu korporacjach został wykupiony przez firmę Oracle.

W tym dokumencie został opisany produkt już dostępny i zmodyfikowany przez firmę Oracle.

2.4.1. Oracle RightNow Contextual Workspaces Cloud Service

Jest to podstawowa wersja oprogramowania, oferującego całego bogactwo narzędzi dodatkowych wspierających pracę użytkownika. Oferuje możliwość projektowania podstawowych przestrzeni roboczych dla każdego zagadnienia biznesowego, używanego przez konsultantów, pracujących z systemem.

Środowisko pracy dla każdego rekordu powinno zawierać wszystkie potrzebne informacje tak, aby konsultant mógł prawidłowo zająć się wszelkimi zdarzeniami, jakie może napotkać w czasie swojej pracy. Podczas interakcji konsultant musi również posiadać możliwość zanotowania użytecznej, w przyszłej analizie, informacji. Wiedza taka jest przydatna w celu udoskonalania produktu jak i dostarczania obsługi na odpowiednim poziomie.

2.4.2. Projektowanie formularzy

RightNow oferuje całe bogactwo różnych możliwości, w tym kreowania zasad *When/IF*. Tego typu udogodnienia niosą ze sobą nieocenioną korzyść biznesową, ponieważ pozwalają dostosować wymagania czy też zagadnienia biznesowe do danej firmy. Ułatwia to dopasowanie programu do potrzeb danego przedsiębiorstwa.

Home Insert Field Insert Control	Object Tools Design Rules	admin only Incide	eit Workspier 199011 - Workspace Desig	بم – – × 10 گ
Assigned & Rules triggered Current Selection + & Rules that Ref Delete & All Workspace	by this Object erence this Object tules	Add Rule	Edit Move Down	
	Add rule Rule Properties Name @ww.RMM Rule Definition When Editor Loaded or Record Reld is Changed	Add Rule	Edit Rule Move Down	Notes
Logged in ast General Administrator			Bionine :	

Rysunek 7. Stosowanie zasad When.IF, THEN, Else. Źródło: [12]

Kreowanie zasad When.IF, THEN, Else zaczyna się od właściwej identyfikacji typu posiadanych:

- Kontaktów
- Incydentów
- Zadań

- Możliwości, w tym też możliwości biznesowych
- Cech szczególnych danej organizacji

RightNow oferuje możliwość wybrania określonej grupy zdarzeń i połączenia ich z odpowiednimi interakcjami np. kategoriami produktów. Następnie można je pogrupować w odpowiednie panele, zakładki w środowisku, a następnie pogrupować zachodzące zdarzenia na zasadzie akcji i reakcji. Prowadzi to do powstania dynamicznego środowiska.

Po zakończeniu definiowania środowiska pracy i sprawdzeniu poprawności identyfikacji jego kluczowych elementów, Oracle RightNow oferuje możliwość dostarczenia wygenerowanego środowiska do odpowiednich osób lub też konsultantów. Dzięki czemu każdy otrzymuje dostęp do potrzebnych informacji i narzędzi, umożliwiających sprawną pracę, w krótkim czasie.

		EMEA Cust Support - cccoe -	RightNow CX			- = ×	
Home						ち 🕡 🖉 🗙	
Save Save & New Refrest	Delete	es Permissions Analytics	Audit N	otes			
Save Action	15	Show		infile.			
Configuration «	Profiles EMEA Custs	upport				4 P	
Configuration				Constanting			
Administration Start Page Staff Management	Profile - Edit			Workspaces/	Workflows rvice Mgr Test Inc icial Response Syst	dent Workspace	
Staff Accounts by Manager	*Name			Pin Conte	ACTS		
Logged In Staff Accounts	EMEA Cust Support			P Chat	occo.		
Prolie:				1 Contract	ote 2005		
Password Configuration	Interfaces			19 PO Your	ote 2000		
Workspaces/Workflows	Party Property lies		Contraction of the	Additional and a second			
Scripts Explorer	Interface Langui	ige "Label Aco	ess Navigatio	TSE Molecol	Providenting	workspaces	
Client Workflow Images	cccoe English	(US) EMEA Cust Support	CRM Navig	atic Product	Registration		
E Customizable Menus				ter Produ	act Registration - I	ncident	
🗉 🛅 Site Configuration	Workspaces/Workflo	Add-Ins		* Standard	d		
Control C		All discourse all statements		Encode	Leg Incident		
E C Sales	Editor	workspaces worknows		Upgrade	Upgraded Chat Workspaces		
🗄 🛅 Marketing	Answer	Answer		* Workflow	* 🗃 Workflow		
🖹 🛅 Database	Answer Multi-Edit	Answer Multi Edit		-			
Customize List	Contact	Administrator Contact Workspace	100005	_		Cite Cancel	
Recent items 🛛 🗧	Contact Multi-Edit	Contact Multi Edit		_	-		
Quick Search 🗧	Incident	Product Registration - Incident					
A Hama	Incident Multi-Edit	Incident Multi Edit			a	100	
Li nome	Opportunity	Lead - Opportunity Workspace			9	4.5	
G Analytics	Opportunity Multi-Edit	Opportunity Multi Edit			a		
191	Organization	Marketing Manager Organization \	Vorkspace 1000	110	9		
1 asks	Organization Multi-Edit	Organization Multi Edit			a	RightNow	
Communication Center	Quote	Marketing Manager Quote Worksp	ace 100012		4		
u 🥹 🤉 🔁 🤧 🚳 -							
Logged in as: Chris Morris						100% 🕤 🕘 💮 , ti	

Rysunek 8. Dostarczenie dynamicznych interfejsów. Źródło: [12]

Środowisko, w którym zaimplementowane są tego typu możliwości, jest w stanie pracować produktywniej. Każdy problem może być rozłożony w łatwy sposób na czynniki pierwsze, co pozwala na dostarczenie skrojonego na miarę interfejsu, który przyspiesza pracę.

Praca z Oracle RightNow ułatwia organizację pracą - każdy manager działu wie, jakimi interfejsami dysponuje i może je w odpowiedni sposób rozdzielić pomiędzy konsultantów. Minimalizuje to sytuacje, podczas których konsultant może być przytłoczony wyborem właściwego interfejsu spośród wielu dostępnych. Do dyspozycji ma wyłącznie zadania przydzielone lub zatwierdzone przez jego kierownika.

2.4.3. Generowanie scenariuszy rozmów

RightNow oferuje możliwość zarządzania wiedza, co jest istotne z perspektywy firmy i jej procesów. Realizowane jest to przez możliwość generowania scenariuszy, które pozwalają właściwie zarządzać danym problemem. Takie przewodniki mogą być wytworzone dla najczęstszych przypadków, z jakimi stykają się konsultanci i pracownicy firmy.



Rysunek 9. Dostarczenie scenariusza dla agentów. Źródło: [12]

Scenariusz może być kreowany na wiele sposobów. Najwygodniejszy wydaje się ten przedstawiony na powyższym rysunku, czyli za pomocą wbudowanego narzędzia graficznego do tworzenia drzewa decyzyjnego. Podczas jego tworzenia użytkownik ma dostęp do wielu opcji, w tym do możliwości dodania:

- Pytania
- Pytania wybieranego z listy pytań za pomocą rozwijanego menu
- Kontrolki 'Radio button'
- Przycisku z pytaniem
- Odpowiedzi
- Czy tez wyjaśnienia do danego tekstu
- Wielu, wielu więcej

Po wykreowaniu przewodników, będących odpowiedziami na najczęstsze problemy, z jakimi styka się dana firma, istnieje możliwość ich publikacji w bazie wiedzy. W ten sposób stają się dostępne dla wszystkich agentów i pracowników. Firma wyposażona w tego typu aplikację lepiej radzi sobie z problemami i może na nie szybciej reagować.

2.4.4. Skryptowanie

Oprócz kreowania przewodników, czyli typowych scenariuszy użytkowania aplikacji, Oracle RightNow oferuje możliwość skryptowania sytuacji lub zdarzeń, z jakimi stykają się użytkownicy.

Oprogramowanie Oracle pozwala wytworzyć gotowe odpowiedzi na trudne sytuacje, z którymi trzeba mierzyć się w codziennej pracy. Jednocześnie pozwala lepiej wypełniać skomplikowane procedury, w których, przez ich rozmiar i skomplikowanie, bardzo często łatwo jest ominąć kluczowe kroki.

(R)		age Tools	Object Tools CS	O-Script-Conti	act - userconfere	nce_devb91 - Rig	htNow CRM	_ = ×
(E)	Home Insert Field Insert Control Page	e Rules	Design					6 0 0 ×
Prev	iew Add Branch - Page Delete Pages Lal	• • • • • • • • • • • • • • • • • • •	Table Position	Padding + Margin + Size +	A DEX F		Validate 12 ST Validate 12 Pa Designer (otes how Outline ge Selector = Options
28	RightNow Media Bar							
6	Your Status Active Engagements		Session Controls		1	- 1 m		Supervisor
69 1	Logn Status: Unavailable		Frish Wrap Up	nate SS	Conference	Request Dia	Place On Hold	Monitor
9	Scripts Explorer Contact							4 Þ
2	Script Pages X							
à	Greeting	Okay, we car	n set up your new a	ccount over	the phone now	The set up pro	ocess should ta	ike just a
2		tew minutes.	to begin, thi need y	our basic co	ntact informatic	on.		
3	Contact Qualification 1							
0	Contact Qualification 2	First, I will nee	ed your address:		Agentins	tructions:		
91	B		-		7	Enter the addre	ss in the box to th tress or ZIP code	e left. • to healn
a	DDS Misk Eval	Input an address	, then press enter		3	Press -Enter- to	verify the addres	is with the
2	Go to Recommended Upsell				US Posta 4	V Service. Modfy address and	by again I needed.	
-								
	Go to Basic Product	Next I will ne	ed your first and las	t name:	-			
2	Please call other service number		va your mot and ido	indirite.				
~		First Name*						
	Recommended Upsell	Last Name*						
2	Go to Schedule Truck Roll				Continue			
**					controle			
20	Basic Product							
- A	Schedule Truck Roll							
411	(c)							
Logge	d in as: Neil Srinivasan					Online	100% 💮	0: 🔶t

Rysunek 10. Ekran do projektowania skryptów. Źródło: [12]

Kreowanie skryptów w Oracle RightNow jest proste i przypomina kreowanie formularzy z drobnymi różnicami.

- Skrypty są sekwencjami ekranów, utrzymanych razem liniowo lub z rozgałęzioną nawigacją. Zamiast projektować jeden ekran z mnóstwem informacji, co jest możliwe przy kreowaniu interfejsów, opisywanych kilka punktów wyżej, można projektować całą serię ekranów, która składa się na pojedynczy scenariusz.
- Skrypty mogą pobierać dane, które używane są później przy rozgałęzieniach. Usługa, zwana
 pytaniami skryptowymi, umożliwia agentom dokonywanie właściwych wyborów podczas
 interakcji. Należy mieć na uwadze, że dane z takiej interakcji nie są zapamiętywane na jej
 koniec. To jest zasadnicza cecha odróżniająca skrypty od projektów formularzy, gdzie cała
 interakcja i decyzje zapamiętywane są w bazie danych.
- Skrypty można osadzać w formularzach.

Skrypty są przede wszystkim uzupełnieniem funkcjonalności, jaką są formularze. Cześć procesów, czyli przywitanie klienta czy zadanie mu pewnych ściśle określonych pytań, może zostać zautomatyzowane poprzez osadzenie samych skryptów w formularzach. Integracja skryptów z formularzami pozwala znacznie wzbogacić interakcję klienta z konsultantem.

2.4.5. Procesy pracy

Jedną z najbardziej zaawansowanych możliwości, jakie oferuje RightNow jest możliwość generowania procesów pracy (*workflow*), czyli dynamicznego pulpitu agenta. Pozwala to przełączać kontekst tyle razy ile wymaga tego dany proces.

Żeby to osiągnąć należy wykonać nowy *workflow*. Następnie do każdego elementu *workflow* można dodać odpowiedni formularz, który został już wcześniej wytworzony czy też np. odpowiedni skrypt.



Rysunek 11. Ekran projektowania workflow. Źródło: [12]

Po zebraniu wszystkich formularzy i skryptów można zacząć je łączyć tak żeby tworzyły logiczną całość. Oraz żeby był możliwy przepływ od początku do końca. Możliwe jest przetestowanie *workflow* za pomocą odpowiedniego narzędzia. Dla zaawansowanego testowania można udostępnić dany *workflow* za pomocą specjalnego konta testera. Przez co nie będzie to miało destrukcyjnego wpływu na środowisko produkcyjne. Po właściwym przetestowaniu można już dostarczyć cały proces do agentów.

2.4.6. Ocena oprogramowania

Obecnie wszelkie oprogramowanie, jakie jest wytwarzane na rynek IT niezależnie czy dotyczy ono *helpdesku*, wsparcia klienta, czy też jest oprogramowaniem czysto biznesowym, jest najczęściej wykonywane w formie Webowej. Czyli powinno być zbudowane i zaprojektowane w taki sposób, aby była możliwa jego praca w przeglądarce internetowej. Prowadzi to do wielu udogodnień miedzy innymi administratorzy odpowiedzialni za aplikacji mogą ja łatwiej dostarczyć do klientów. Osoby odpowiedzialne za bezpieczeństwo nie muszą się obawiać, że tak naprawdę binaria programu znajdują się na komputerze użytkownika, co może prowadzić do wielu problemów – miedzy innymi możliwość

wglądu w pliki konfiguracyjne itp. Sami developerzy mają zdecydowanie prostsze zadanie podczas dostarczania uaktualnienia do programu muszą je dostarczyć do jednego miejsca.

Oczywiście forma webowa to nie tylko same plusy. Niewątpliwe wadą tego typu aplikacji jest konieczność zapewniania pełnej kompatybilności z przeglądarkami, w których aplikacja pracuje. Dodatkowo samo bezpieczeństwo też spoczywa na przeglądarce, wszelkie problemy, jakie ją dotykają mogą tez być niebezpieczne dla oprogramowania.

Dlatego projektując aplikacje należy rozważyć wszelkie za i przeciw ponieważ kiedy proces projektowania się zakończy zmiana decyzji może być bardzo kosztowna zarówno, jeżeli chodzi o środki finansowe jak i wkład intelektualny i co jest chyba najbardziej dotkliwe w obecnych czasach: czas stracony na złych decyzjach.

Oracle RightNow jest to zintegrowane narzędzie dostarczające wielu możliwości dla każdej firmy i instytucji, która boryka się z problemami podczas pracy. Oprogramowanie to pozwala automatyzować wszelkie procesy i odpowiednio je uregulować. Użytkownik dostaje cały wachlarz możliwości od kreowania prostych skryptów, które pozwolą na uproszczenie typowych operacji. Po tworzenie indywidualnych formularzy do wprowadzania istotnych informacji.

Wszystko to na koniec można ubrać w *workflow*, czyli pełną automatyzacje skomplikowanych procesów. Zapewnia to, że każdy proces zostanie wypełniony od początku do końca, tak ja przewiduje procedura bez pomijania kluczowych dla sprawy kroków.

2.5. SolveDirect

Oprogramowanie SolveDirect jest systemem wspierającym ITSM – *IT Service Management*, którego zadaniem jest:

- wspieranie partnerów biznesowych w codziennej pracy,
- stała poprawa jakości świadczonych usług, oraz
- raportowanie.

Wedle założeń twórców SolveDirect miał być tylko łącznikiem, mostem między innymi aplikacjami. Z racji stale rosnącej liczby programów uznano, że musi istnieć "coś", co pozwoli na łatwą integrację oprogramowania.

Obecna wersja SolveDirect jest dostarczana na zasadach *SaaS* (z ang. *Software as a Service*) - oprogramowanie jako usługa. Klient otrzymuje dane do logowania i nie musi się martwić o wydajność, pojemność, bezpieczeństwo danych czy wsparcie dla aplikacji. Cała administracja systemem odbywa się poprzez strony WWW.

SolveDirect podzielony jest na moduły, z których każdy wspiera inny obszar działalności klienta. Moduły opisane zostały w dalszej części tego rozdziału, a są to:

- *SD.Call* zgłoszenia serwisowe
- *SD.Inventory* katalog produktów
- *SD.Bridge* łącznik między systemowy
- SD.Solution baza rozwiązań
- *SD.Report* moduł raportowy
- *SD.Message* moduł automatycznych powiadomień
- *SD.Mobile* obsługa urządzeń mobilnych

2.5.1. Moduł SD.Call

Główny moduł do obsługi zgłoszeń serwisowych. Umożliwia wykonywanie podstawowych funkcji Service Desku. Użytkownikiem tego modułu mogą być zarówno operatorzy *helpdesku*, serwisanci jak również klient końcowy. Funkcje te umożliwiają otwarcie nowego zgłoszenia, wyświetlanie tych już istniejących, edytowanie i aktualizowanie zgłoszeń oraz finalnie ich zamknięcie.

Tworzenie zgłoszenia odbywa się na wcześniej przygotowanej przez administratora formatce (patrz Rysunek 12).

New Incident								
Create new call								
RequestType: CallState:		INC-Incident -						
CUSTOMER INCIDE	ENT							
Customer:								
Caller:	DQ	Servicedesk, Demo						
CallerEMail:								
CallerTel:		4315853555	-					
Description:		New Incident	=					
Contact:		Servicedesk, Demo						
CPFirstName:		Demo						
CPLastName:		Servicedesk						
CPTel:		4315853555						
CCPEMail:								
CustCalIID:								
CONTRACT AND S	LA							
Contract/SLA:	DQ							
PROVIDER								
Provider:								
SPCalIID:								
Helpdesk:	\supset	Servicedesk, Demo						
Editor:	\triangleright	Servicedesk, Demo						
Remarks:								
Diagnosis:								
CLASSIFICATION			Ŧ					

Rysunek 12. Przykładowa formatka nowego zgłoszenia. Źródło: [13]

Wyświetlanie zgłoszeń może odbywać się z zastosowaniem wszelkiego rodzaju filtrów i sortowań. Najczęściej stosowane ustawienia zostają zapisane przez administratora w formie listy wyszukiwania, która jest dostępna do wyboru dla użytkownika z listy rozwijalnej. Filtrami mogą być w zasadzie dowolne pola użyte na formatkach zgłoszenia. Dodatkowo niektóre pola umożliwiają dodatkowe funkcje wyszukiwania jak np. zastępowanie pojedynczego znaku w wyszukiwanej frazie znakiem "?" jaki ciągu znaków znakiem "*". Wyszukiwanie dzieli się na pola tekstowe i słownikowe, w polach tekstowych można korzystać z operatorów logicznych *and/or*, a w polach słownikowych z wyboru wielokrotnego. Wyświetloną listę zgłoszeń można w łatwy sposób wyeksportować do plików w dowolnym z poniższych formatów: xls, xml, csv.


Rysunek 13. Przykładowy *workflow* dla pierwszej linii wsparcia. Źródło: Opracowanie własne.

Edycja i aktualizacja zgłoszeń to kluczowe zadanie dla osób pracujących na *helpdesku* czy też w serwisie. Jest to wymagane do sprawnej obsługi klienta. Aby pracownik nie musiał się zastanawiać, co począć ze zgłoszeniem system umożliwia skonfigurowanie *workflow*, po którym będzie "przemieszczać" się dane zgłoszenie. Można to przyrównać do budowy torów kolejowych, administrator wyznacza szlak, a pracownik podejmuje decyzję, którym torem obsłużone zostanie dane

zgłoszenie. Z uwagi na charakter naszego projektu skupimy się tylko na obsłudze incydentów. Poniżej przedstawiona jest przykładowa konfiguracja przepływu pracy. *Workflow* ten przedstawia, listę możliwych przejść zgłoszenia dla pierwszej linii, czyli *helpdesku*.

Pracownik, aby zmienić status zgłoszenia z jednego stanu do drugiego wykonuje tzw. akcję, do której może być przypięta dowolna formatka.

2.5.2. Moduł SD.Inventory

Według założeń moduł *SD.Inventory* przechowuje dane konfiguracyjne urządzeń w bazie SolveDirect, oczywiście wszystko odbywa się za pośrednictwem interfejsu webowego. W razie potrzeby z bazą może pracować wielu użytkowników realizujących różne zadania jednocześnie, na co pozwala bezpieczny system autoryzacyjny. Zapewnia on dostęp do odpowiednich obszarów tylko tym pracownikom, którzy są do tego uprawnieni.

Moduł SD. Inventory daje łatwy dostęp do danych konfiguracyjnych takich jak:

- typ urządzenia,
- nr seryjny,
- warunki gwarancyjne,
- miejsce użytkowania.

Poprawne prowadzenie i aktualizowanie informacji w tym module, pozwala na wyszukiwanie i otwarcie zgłoszeń przypisanych do danego urządzenia. To znaczy, że pracownik po otrzymaniu zgłoszenia, może po numerze seryjnym wybrać urządzenie i wybrać je w celu otwarcia nowego zgłoszenia w systemie. Równocześnie kierownik firmy czy menadżer kontraktu może szybko otrzymać informacje, np. ile urządzeń jest pod opieką firmy, a ile jest w danej chwili w naprawie – czyli są wystawione zgłoszenia na dany sprzęt.

s⊙lvedirect	Search		× P User: De	emo Servicedesk	SDStandard	~	English 🗸	Service Manaç	er 🗸	Indianapolis (EST)	Switch to SD	2 Docu Logout
🗙 🛾 Service Desk 🖉 Service Request	Managem	ent Incid	ent Management	Problem Management	Change Mana	gement Ser	rvice Level Managemen	t Reports B2I	3 Monitoring	Administration	Customizing	About SolveDirect
Administration		Devices										*8
Users and Organizations	>>	KurzName	Hersteller	ProdKlasse	Typ/Modell	Name	SerNr	InvNr	Lieferant	Kunde	Vertrag/SL	A IstBestSL
Contracts		Desktop	DELL	Desktop	Dell OptiPlex/SX		123456789		Demo INC	Demo C1	INC.C1 INC	6
Devices and Locations	8 U I	Desktop	DELL	Desktop			456123		Demo CHG	Demo C1	CHG.C1 CH	iG 🥝
- Cocations												
- 💮 Manufacturers												
Device Types												
M _ Solutions												

Rysunek 14. Przykładowy wygląd modułu SD. Inventory. Źródło: [13]

2.5.3. Moduł SD.Bridge

Moduł *SD.Bridge* jest najstarszym dostępnym modułem, gdyż to od niego zaczęto tworzenie całego systemu SolveDirect. Jest narzędziem typu ETL (z ang. *extract, transform, load*). Jego głównym zadaniem jest łączenie ze sobą różnych systemów. Według konceptu platforma *SD.Bridge* składa się z predefiniowanych interfejsów do systemów zewnętrznych. Użytkownik platformy może wybrać najbardziej odpowiedni kanał transferu danych, a komunikacja z partnerem odbywa się przez ustaloną neutralną bramkę. Ujmując to inaczej, moduł *SD.Bridge* zajmuje się tłumaczeniem danych z jednego systemu na język, który zrozumie drugi system. Najlepiej przedstawia to rysunek nr 15.



Rysunek 15. Mapa możliwych ścieżek komunikacyjnych dla modułu SD. Bridge. Źródło: [14]

Moduł *SD.Bridge* to bardzo wszechstronne narzędzie. Często klienci decydowali się na wybór tylko tego modułu z racji dużego rozproszenia różnych systemów informatycznych. Dzięki temu nie było konieczne wdrażanie jednego dużego systemu, a w przypadku zmiany któregoś z mniejszych podsystemów nie było potrzeby przebudowywania całej mapy komunikacji "każdy z każdym". Wystarczy jedynie zmienić ścieżkę połączenia.

2.5.4. Modul SD. Solution

W każdej organizacji świadczącej wsparcie dla swoich partnerów biznesowych, po pewnym czasie napływające zgłoszenia stają się w dużej mierze powtarzalne. Z drugiej strony, przy dużej liczbie różnych zgłoszeń nie ma możliwości posiadania pełnej wiedzy na temat wszystkich możliwych rozwiązań. Dlatego też powstał moduł *SD.Solution*, który jest bazą rozwiązań gdzie szczególny nacisk został położony na rozwiązania z obszaru incydentów. Dzięki temu modułowi operator *helpdesku* może jeszcze szybciej reagować na napływające zgłoszenia.

Moduł ten został tak skonstruowany, aby użytkownik, po rozpoznaniu typowego błędu, mógł jeszcze szybciej reagować na potrzeby klienta podając rozwiązanie w czasie rzeczywistym np. w trakcie rozmowy z klientem.

Administrator ma możliwość przygotowania i opracowania gotowych rozwiązań na przyszłość.

Baza *SD.Solution* pozwala na inteligentne wyszukiwanie rozwiązań. Po zastosowaniu odpowiednich kryteriów, system wyświetli rozwiązania od najbardziej do najmniej trafnych. Natomiast po wyborze danego rozwiązania, jego pozycja w rankingu rozwiązań wzrasta.



Rysunek 16. Przykład zastosowania modułu SD. Solution. Źródło: [15]

2.5.5. Moduł SD.Report

Moduł *SD.Report* został przygotowany tak, aby dostarczyć szerokie możliwości raportowania, w szczególności monitorowanie wolumenu zgłoszeń, jak również poziom jakości serwisu. Zadanie to jest tym trudniejsze im większa jest organizacja i jej rozproszenie geograficzne, dlatego też moduł ten został wyposażony w wiele funkcji wyszukujących i grupujących zgłoszenia.

SD.Report czerpie informacje bezpośrednio z bazy danych SolveDirect. Po ustawieniu predefiniowanych kryteriów raporty dostępne są w czasie rzeczywistym, jako:

- strony HTML,
- pliki csv, xls lub pdf.

Moduł raportowy umożliwia przygotowanie raportów w formie tabelarycznej lub jako wykresy kołowe czy słupkowe.

Moduł ten umożliwia przeprowadzenie dogłębnej analizy sytuacji *helpdesku* na każdym szczeblu, co ułatwia podejmowanie trafnych decyzji biznesowych.

s⊙lvedirect	Search	×Р	lser: Dem	o Serviced	esk		SDSI	tandard	~ I			English	v		Service	Manag	ger 🗸		1	Indiana	apolis ((EST)	Switch to S	D² Doci	u Logout
Service Desk Service Request	Management 🛛 Ir	ncident Manage	ment	Problem	Manage	nent	Chan	ge Mar	nageme	nt S	ervice L	evel Ma	anager	nent Re	eports	B2	B Mor	nitorin	g	Admir	nistrat	ion	Customizing	About S	SolveDirect
Service Desk	Calls Op	en SL Respons	se																						* 8
Calls Open New Incident New Problem New Change New Service Request	▶List ▶Tab	le Graphic C	controls 5	10 15	20	25 3	0 35	40	45	50 55	Numt 60	oer 65	70 3	5 80	85	90	95	100	105	110	115	120	 InTime Overdue InProgress 	s	
Calls Open SL Recovery Calls Open SL Response More Service Level Management Reports New Call by User SD_Standard Workflows Reports.pdf		Incident						37										60							
		Service	4	8																					
		Change		1	5			22																	
				1	5	10																			

Rysunek 17. Wynik działania przykładowego raportu. Źródło: [13]

2.5.6. Moduł SD.Message

Moduł *SD.Message* oferuje zautomatyzowane, standardowe procedury powiadomień i eskalacji, dzięki czemu wszyscy uczestnicy procesu są uwzględnieni w przebiegu zdarzeń.

Po zdefiniowaniu powiadomienia, czyli określeniu jego treści odbiorców jak również warunków wysłania powiadomienia, są one automatycznie rozsyłane. Mogą być dystrybuowane poprzez e-mail, sms, protokół soap, https post lub ftp.

Wzorce powiadomień można dowolnie definiować za pomocą szablonów html, natomiast dane pobierane są z systemu za pomocą instrukcji języka xml.

Wśród cech i funkcji powiadomień modułu SD. Message możemy wyróżnić:

- powiadamianie serwisantów indywidualnie lub całych grup techników
- powiadomienia w przypadku zmiany statusu
- powiadomienia w przypadku braku działalności na zgłoszeniu, np. alert po...
- powiadomienia w przypadku wystąpienia zdarzenia
- różne typy komunikacji jak wspomniane powyżej sms, e-mail, etc
- definiowanie indywidualnych wzorców powiadomień

Poniższy rysunek nr przedstawia przykładową skonfigurowany alert modułu SD.Message:

Select message tr	iggers:						
Company CallSy	stem	IsActive					
Demo 💌 ALL							
√GO!							
Expand all / Collapse all List							
1 message trigger	s selected						
🖃 🗁 Trigger Error	When error occurs						
- 🗁 Outbound	Communication: Error Me	essage via Mail from SD					
📃 Template	: Content Error Message						
📃 Template	: Subject Error Message S	Subject					
- 🗁 Receiver							
Caller-M	ail from SD						

Rysunek 18. Przykładowy alert modułu SD. Message. Źródło: [13]

2.5.7. Moduł SD.Mobile

W dobie rosnących potrzeb bycia w ruchu i rozwiązywania problemów w drodze, nie sposób pominąć tak ważnej kwestii jak wersja mobilna. W chwili obecnej dostęp do urządzeń przenośnych jest na tyle szeroki, że naturalnym jest konieczność posiadania dedykowanej do tego celu wersji. Dlatego też powstał moduł *SD.Mobile*. Jest to alternatywny sposób dostępu do standardowej platformy SolveDirect i zawiera następujące funkcje:

- przeglądanie listy zgłoszeń,
- wyświetlanie szczegółów zgłoszenia,
- aktualizowanie i praca nad zgłoszeniem,
- zapisywanie czasu pracy,
- zamknięcie zgłoszenia.

Wszystkie kolumny i pola dostępne w interfejsie mobilnym mogą być łatwo modyfikowane i ustawiane przez administratora, tak aby optymalnie odzwierciedlić żądany przebieg procesu serwisowego.

1.00	nord Moour	liet	00 %				
- 19	Incidents I	ast Week: 6	(Page 1/1)				
	Refresh	NewCall	Select				
	INC-Incident			-			
	298680819						
	12-02-06 15:30	CET					
	SD_Ref Customer 1						
	SD_Ref Incident						
	Incident escalated to 3rd						
	¥						
	I have no intern	et access					
	INC-Incident						
	298680869						
	12-02-06 17:57	-CET					
	SD_Ref Custon	ner 1					
	SD_Ref Incider	NI					
	Service recover	red					
	÷						
	My SAP client of	does not boot					
2		+	00 G	ş			
	- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10	(1.1)	lean le	đ			

Rysunek 19. Przykładowy wygląd mobilnej wersji aplikacji SolveDirect. Źródło: [16]

2.5.8. Podsumowanie

Aplikacja SolveDirect to narzędzie, które znajdzie zastosowanie w każdej firmie świadczącej usługi serwisowe. Dzięki ogromnemu wachlarzowi narzędzi konfiguracyjnych aplikacja może być stosowana w wielu branżach.

Istotną zaletą jest dostarczanie jej na zasadach modelu *SaaS*, dzięki czemu użytkownicy nie muszą się martwić o zasoby, bezpieczeństwo danych, czy wydajność. Jednakże może też stać się to wadą. Zastosowanie takiego modelu dostarczania aplikacji, ogranicza możliwości rozbudowy programu, gdyż aplikacja jest dostarczana globalnie i wszelkie zmiany muszą uwzględniać potrzeby wszystkich klientów.

W porównaniu do naszego projektu, aplikacja SolveDirect nie daje możliwości dokładnego definiowania *workflow*, gdyż definiujemy tylko ścieżki, po których porusza się zgłoszenie. W SolveDirect nie ma możliwości definiowania *workflow* na zasadzie "krok po kroku" wraz z elementami decyzyjnymi.

Z racji długiego istnienia² na rynku interfejs programu SolveDirect odstaje od współczesnych standardów, jednakże z uwagi na rozległą sieć klientów radykalne zmiany w wyglądzie mogą być trudne do wprowadzenia.

² Produkt SolveDirect zadebiutował komercyjnie w 2000 roku.

3. Proponowane rozwiązanie

Proponowanym rozwiązaniem jest aplikacja zapewniająca wszechstronną i całkowitą obsługę zgłoszeń serwisowych. Które rozpoczynają się np. w momencie przyjęcia zgłoszenia, obejmuję np. przydzielenie zgłoszenia do określonego pracownika. Kończy się zaś mailowym powiadomieniem klienta o usunięciu usterki.

Ponadto aplikacja umożliwia definiowanie własnych przebiegów serwisowych polegających na przepływie informacji pomiędzy uczestnikami procesu, wykonywanie czynności warunkowych oraz uzależnianie kierunku działania od danych zewnętrznych.

3.1. Diagram przypadków użycia

Niniejszy podrozdział zawiera przypadki użycia ilustrujące użycie projektowanej aplikacji przez jej użytkowników w warunkach produkcyjnych – pracowników firmy serwisowej oraz klientów samodzielnie dokonujących zgłoszenia awarii.

3.1.1. Aktorzy



Rysunek 20. Aktorzy obecni w systemie. Źródło: Opracowanie własne.

Rysunek powyżej prezentuje aktorów – użytkowników wchodzących w interakcję z projektowanym systemem. Po stronie firmy korzystającej z aplikacji będą to konsultanci, kierownicy oraz administrator/administratorzy systemu. Do ich dziedziczonych ról i uprawnień należy:

- dla Konsultanta L2: praca w systemie tylko w obrębie swojego poziomu serwisowego
- dla Konsultanta L1: praca w systemie tylko w obrębie swojego poziomu serwisowego, rejestracja zgłoszeń, kontakt ze zgłaszającym oraz zamykanie zgłoszenia.
- dla Kierownika: wykonywanie akcji konsultantów L1 i L2 oraz tworzenie raportów
- dla Administratora: wszelkie czynności użytkowe oraz konfiguracyjne (posiada on maksymalne uprawnienia do systemu)

Pozostali aktorzy – nie będący pracownikami firmy korzystającej z aplikacji - obejmują następujące obiekty: Klient, System, Firma oraz Zleceniodawca. Bezpośrednie role dotyczą tylko

aktora klienta, i obejmują zgłoszenie problemu konsultantowi lub samodzielne zarejestrowanie go w systemie.



3.1.2. Administracja danymi

Rysunek 21. Aktywność aktorów: kierownik, konsultanci L1/L2. Źródło: Opracowanie własne.

Przypadek użycia zaprezentowany na rysunku nr 21 przedstawia aktywność w systemie trzech aktorów: kierownika (użytkownika uprzywilejowanego) oraz konsultantów L1 / L2.

Przykładowy scenariusz dla aktora "Kierownik":

Użytkownik uruchamia aplikację. Po pomyślnym zalogowaniu się do systemu przechodzi na stronę, umożliwiającą wprowadzenie osób zgłaszających usterki do bazy, równocześnie system sprawdza uprawnienia użytkownika. Na ww. stronie:

• w odpowiednich polach użytkownik wprowadza dane osoby do systemu i akceptuje przyciskiem "Zapisz" wprowadzone dane, co powoduje dodanie rekordu do bazy, lub

 użytkownik za pomocą przycisku "Wprowadź dane z pliku" powoduje wyświetlenie systemowego (zależnego od systemu operacyjnego) okna dialogowego służącego wskazaniu ścieżki do pliki zawierającego dane. Po naciśnięciu przycisku "Dodaj osoby do bazy" następuje wczytanie pliku i utworzenie odpowiednich rekordów.

W przypadku wskazania pliku niespełniającego kryteriów wyświetlony zostaje stosowny komunikat.

Powyższy scenariusz – zawierający dwa sposoby wprowadzania danych (ręczny oraz za pomocą pliku) dotyczy czynności polegającej na dodawaniu do bazy danych osób uprawnionych do informowania w imieniu klienta i zaistniałej awarii. Analogiczny (i w związku z tym nieopisywany i nieprezentowany na odrębnym rysunku) sposób działania przewidziany jest w przypadku dwóch innych czynności, tj.

- wprowadzania lokalizacji, oraz
- wprowadzania produktów objętych wsparciem serwisowym.



Rysunek 22. Przykładowy scenariusz dla aktora "Kierownik". Źródło: Opracowanie własne.

Scenariusze alternatywne dla aktora "Kierownik":

Wszystkie trzy scenariusze realizowane przez użytkownika uprzywilejowanego zawierają identyczne lub bardzo zbliżone scenariusze alternatywne:

- niepoprawne dane logowania efekt: komunikat informacyjny
- brak uprawnień do odwiedzania strony efekt: komunikat informacyjny
- niepoprawny format pliku lub błąd w trakcie importu danych efekt: komunikat informacyjny

Diagram aktywności dla ww. scenariusza przykładowego prezentowany jest na rysunku poniżej:

Przykładowy scenariusz dla aktora "Konsultant":

Użytkownik uruchamia aplikację, po pomyślnym zalogowaniu się do systemu przechodzi na stronę umożliwiającą przeszukanie bazy osób zgłaszających usterki, równocześnie system sprawdza uprawnienia użytkownika.

Po naciśnięciu przycisku "Wyszukaj" system wyświetla spis rekordów spełniających określone kryteria. Ze względu na minimalizację obciążenia serwera rekordy prezentowane są w grupach, przechodzenie pomiędzy poszczególnymi grupami możliwe jest za pomocą przycisków. Po naciśnięciu przycisku "Eksportuj" system prezentuje okno dialogowe umożliwiające wybór formatu a następnie lokalizacji i nazwy dla nowotworzonego pliku. Po akceptacji lokalizacji następuje utworzenie pliku.

Dwa pozostałe rodzaje aktywności użytkownika "Konsultant", tj. eksport listy lokalizacji oraz eksport listy produktów przebiegają według bardzo zbliżonego schematu, obejmującego wybór rekordów, wskazanie miejsca docelowego, formatu pliku i wreszcie zapis pliku z danymi.





Scenariusze alternatywne dla aktora "Konsultant":

Wszystkie trzy scenariusze realizowane przez użytkownika systemu zawierają identyczne lub bardzo zbliżone scenariusze alternatywne:

- niepoprawne dane logowania efekt: komunikat informacyjny
- brak uprawnień do odwiedzania strony efekt: komunikat informacyjny
- brak uprawnień aplikacji do zapisywania w danej lokalizacji lub niedozwolona nazwa pliku efekt: komunikat informacyjny

3.1.3. Wyszukiwanie i wyświetlanie zgłoszeń.

Rysunek 24 prezentuje czynności związane z wyświetleniem listy zgłoszeń, utworzonej na podstawie zadanych przez użytkownika kryteriów. I tak, po pomyślnym zalogowaniu się do systemu i weryfikacji przez system uprawnień do odwiedzenia określonej strony, użytkownik może zdefiniować kryteria określające jakie zgłoszenia mają być wyświetlone. Mogą być to m.in. status zgłoszenia, przypisany konsultant, opis słowny (zawierający określone słowa), czy produkt którego dotyczy zgłoszenie.

Zbiór zgłoszeń będący efektem wykonanego zapytania może być wyeksportowany do pliku, a same prezentowane rekordy mogą być sortowane względem czasu umowy SLA. Ponadto, dla każdego z wyświetlonych zgłoszeń możliwe jest wyświetlenie jego szczegółów (strony zawierającej dodatkowe dane), a także - zależnie od poziomu uprawnień użytkownika - możliwa jest też edycja szczegółów zgłoszenia.

Scenariusze alternatywne dotyczące ww. aktywności obejmują takie zdarzenia jak:

- niepoprawne dane logowania efekt: komunikat informacyjny
- brak uprawnień do odwiedzania strony efekt: komunikat informacyjny



Rysunek 24. Wyszukiwanie i wyświetlanie zgłoszeń. Źródło: Opracowanie własne.

3.1.4. Tworzenie i praca ze zgłoszeniem



Rysunek 25. Praca ze zgłoszeniami i tworzenie zgłoszeń. Źródło: Opracowanie własne.

Zgodnie z przeznaczeniem projektowanego systemu, główna część jego funkcjonalności dotyczy rejestracji oraz obsługi zgłoszeń serwisowych. Aktywność tego rodzaju prezentują rysunek powyżej prezentujący ciąg czynności niezbędnych do zarejestrowania zgłoszenia w systemie.

Scenariusz rejestracji zgłoszenia obejmuje i pociąga za sobą - oprócz takich czynności jak uwierzytelnienie użytkownika - następujące czynności:

- przypisanie produktu do zgłoszenia
- przypisanie konsultanta do zgłoszenia
- przypisanie grupy wsparcia

- przypisanie klienta (wraz z danymi dodatkowymi: lokalizacją i danymi zgłaszającymi)
- wybór typu zgłoszenia.

Opcjonalnie, wraz ze zgłoszeniem w systemie może zostać zapisany załącznik będący uzupełnieniem informacji.

Zgłoszenia zarejestrowany w systemie, mogą być modyfikowane przez użytkownika, a modyfikacja dotyczy zmiany stanu, priorytetu czy wreszcie przypisania zgłoszenia do kolejnej linii wsparcia.



Rysunek 26. Czynności administracyjne: zarządzanie użytkownikami. Źródło: Opracowanie własne.

Oprócz działań realizowanych przez użytkownika, sam system ma wpływ na istniejące zgłoszenia. Zautomatyzowane metody to:

- określenie priorytetu zgłoszenia
- nadanie numeru zgłoszenia, oraz
- po uwzględnieniu upływającego czasu wysłanie komunikatu ostrzegającego o niebezpieczeństwie przekroczenia czasu naprawy

Scenariusze alternatywne dla ww. aktywności obejmują (oprócz braku uwierzytelnienia użytkowników) niepoprawną identyfikację klienta, uwzględnienie nie istniejącej w systemie lokalizacji, czy wreszcie próbę rejestracji w systemie zgłoszenia dla produktu nieobjętego wsparciem. Wszystkie te czynności uniemożliwiają rejestrację zgłoszenia.

Czynności administracyjne (dotyczące zarządzanie użytkownikami) realizowane w trakcie obsługi systemu prezentowane są na rysunku powyżej. Obejmują one takie zadania jak:

- dodawanie użytkownika wraz z niezbędnymi danymi osobowymi
- przypisanie użytkownika do określonej grupy (np. kierownicy)
- określenie uprawnień i kwalifikacji użytkownika

W zakresie pracy z danymi dotyczącymi użytkownika obecnego już w systemie, to administrator (superużytkownik) ma możliwość wyszukiwania na podstawie określonych kryteriów użytkowników obecnych w systemie. Następnie - używając wyświetlonej listy - dla każdego z rekordów może przystąpić do edycji danych użytkownika, zmienić jego hasło lub wreszcie usunąć użytkownika.

3.2. Diagram klas

Rysunki *na* kolejnych stronach prezentują częściowo diagram klas przedstawiający logikę biznesową projektowanej aplikacji, czyli strukturę i związki pomiędzy poszczególnymi jej obiektami, ich atrybuty oraz dostarczane metody.

W omawianiu zaprezentowanego diagramu pomóc może logiczne rozdzielenie projektu aplikacji na klasy związane z osobą, zgłoszeniem oraz aktywnością realizowaną w ramach określonego *workflow*. I tak:

- Klasa "*osoba*" reprezentuje obiekty reprezentujące pracowników firmy zapewniającej obsługę serwisową (firmy będącej użytkownikiem systemu) oraz pracowników firm klientów. Jak widzimy z osobami ściśle powiązane jest (patrz akapit poniżej)
- "*zgłoszenie*", które łączy ze sobą pracowników firmy serwisującej oraz klientów a ponadto związane jest z takimi klasami jak lokalizacja wykonania naprawy, produkt którego dotyczy naprawa, czy wreszcie umowa będąca podstawą naprawy (patrz rysunek nr 27).
- Klasa "workflow" powiązana jest z określonymi rodzajami realizowanych aktywności. Co należy podkreślić rodzaj workflow determinowany jest i ściśle zależy od typu i właściwości zgłoszenia. Ponadto sama aktywność zachodząca w ramach workflow zajść może dopiero po spełnieniu określonych warunków.

W zakresie samej struktury klas i istniejących pomiędzy nimi relacjami model biznesowy determinujący projektowaną aplikację wymusił zastosowanie m.in. klas asocjacyjnych – widocznych na rysunku nr 28, jako uzupełnienie relacji pomiędzy pracownikiem a dokonywanym zgłoszeniem oraz pomiędzy umową a produktem, którego ona dotyczy.

Spowodował także powstanie związku o charakterze agregacji pomiędzy pracownikiem a grupą pracowników, do której przynależność określa uprawnienia systemowe. Ponadto należy zauważyć, że sama idea *workflow* spowodowała zaistnienie związku rekurencyjnego dotyczącego klasy reprezentującej poszczególne aktywności tworzące przebieg (*workflow*) – rysunek nr 27.



Rysunek 27. Fragment diagramu klas. Źródło: Opracowanie własne.

3.3. Wymagania funkcjonalne

Lp.	Nazwa wymagania	Opis wymagania	Ograniczenia	Aktorzy
1.0	Rejestracja zgłoszenia przez konsultanta	Konsultant wybiera przycisk "Rejestracja zgłoszenia". Następnie w otwartym formularzu wprowadza numer umowy. System wyświetla nazwę firmy i podpowiada listę przypisanych do niej lokalizacji do zatwierdzenia, gdzie wykonana ma być usługa.	Istnieje baza umów i klientów uprawnionych do wykonywania zgłoszeń.	Konsultant
1.1	Konieczność przypisania klienta do zgłoszenia	System automatycznie podpowiada listę osób, uprawnionych do zakładania zgłoszeń na podstawie umowy. Jeśli w umowie określono, że zgłoszenia może dokonać dowolna osoba z firmy, to konsultant wprowadza	Brak	System

Tabela 8. Wymagania funkcjonalne. Źródło: Opracowanie własne.

		imię, nazwisko, kontakt {telefon/email}.		
1.2	Wybór lokalizacji z bazy	Konsultant wybiera z listy lokalizację, w której ma nastąpić serwis. Lokalizacja zawiera następujące dane: kod pocztowy, miasto, nazwę ulicy oraz numer ulicy i lokalu.	Pole "Adres Wykonania Serwisu" jest obowiązkowe i musi zostać wprowadzone przez konsultanta. Istnieje w bazie adres, pod którym ma nastąpić serwis.	Konsultant
1.3	Wybranie typu zgłoszenia	Konsultant wybiera typ zgłoszenia (lista wartości jest z góry zdefiniowana). Na podstawie wybranego typu zgłoszenia {przykładowo incydent lub wniosek o usługę/wniosek o dostęp} następuje dynamiczna klasyfikacja do właściwego <i>workflow</i> . Dany <i>workflow</i> jest zdefiniowany przez administratora i ma z góry określone pola formularza.	Typ zgłoszenia: incydent, czyli nieplanowana przerwa w świadczeniu usługi informatycznej lub obniżenie jej jakości np. awaria sieci - pola zdefiniowane przez administratora to produkt (identyfikacja przez nr seryjny podany przez Zleceniodawcę) oraz pełny opis. Drugi typem zgłoszenia jest wniosek o usługę lub wniosek o dostęp, czyli ogólny opis różnego rodzaju żądań, które użytkownicy kierują do organizacji IT - pola zdefiniowane przez administratora to temat (informacje, czego dotyczy wniosek, np. instalacja oprogramowania, zmiana hasła, nadanie dostępu) oraz pełen opis (pełna treść wniosku). Administrator może określić dodatkowe przepływy, które będą zawierały wybrane pola. Zgłoszenie powinno być dodawane po przez kreator zgłoszeń. Kreator pozwala na wybranie dodatkowych pól definiujących zgłoszenie. Dynamiczna klasyfikacja.	Konsultant
1.4	Wprowadzenie słownego opisu zgłoszenia (komentarz / opis)	Do zgłoszenia można wpisać słowny opis w formie wolnego tekstu.	Maksymalnie 1000 znaków	Konsultant

1.5	Wyświetlenie produktu z bazy na podstawie numeru seryjnego	System wyszukuje produkt z bazy przy pomocy numeru seryjnego, który został wprowadzony przez Konsultanta. Dodatkowo na jego podstawie system podaje typ, model oraz opcjonalnie opis produktu. Numery seryjne produktów dostarcza Zleceniodawca.	System
1.6	Przypisanie osoby odpowiedzialnej do zgłoszenia	Wybór konsultanta zależy od przepływu pracy. Jeśli w <i>workflow</i> zdefiniowano wybór ręczny - konsultant wybiera osobę do której chce przypisać zgłoszenie. Jeśli w <i>workflow</i> zdefiniowano wybór automatyczny, to: a) gdy nie podano serwisanta powiązanego z firmą w umowie, to wybierany jest serwisant o najmniejszej zajętości. b) w przeciwnym wypadku automatycznie system wpisuje serwisanta przydzielonego do firmy.	Konsultant
1.7	Przypisanie do zgłoszenia klienta (osoby zgłaszającej) poprzez wybór z bazy	Osoba zgłaszająca (klient) jest wyszukiwana w bazie na podstawie numeru umowy	Konsultant
1.8	Możliwość przypisania do zgłoszenia właściwej umowy ramowej (SLA)	W trakcie wprowadzania zgłoszenia do systemu, Konsultant L1 może przypisać do zgłoszenia stosowną umowę SLA (<i>Service Level Agreement</i>) spośród umów zawartych z danym klientem. Umowa ma bezpośredni wpływ na planowany czas wykonania naprawy.	Konsultant
2.0	Dodanie pracowników do grupy	Administrator systemu może przypisać pracowników do grupy	Administrator

3.0	Automatyczne nadawanie numeru zgłoszeniom	Po wprowadzeniu nowego zgłoszenia do systemu (w dowolny sposób), System automatycznie nada unikalny numer zgłoszenia.	System
4.0	Przeszukiwanie bazy osób zgłaszających po dostępnych polach	Konsultant ma możliwość przeszukiwania bazy osób zgłaszających. W tym celu należy skorzystać z formularza wyszukiwania. Formularz wyszukiwania pozwala na podanie zawartości jednego lub więcej pól danych. Po podaniu kryteriów i wciśnięciu przycisku wyszukiwania, System wyświetli wszystkie osoby pasujące do zadanych kryteriów. System powinien akceptować wpisy z * (jako dowolny ciąg znaków) oraz ? jako dowolny pojedynczy znak.	Konsultant
5.0	Przeszukiwanie bazy listy lokalizacji po dostępnych polach	Konsultant z odpowiednimi uprawnieniami ma możliwość wyszukania listy lokalizacji znajdujących się bazie. Funkcjonalność ta powinna być dostępna z poziomu wyszukiwarki systemowej. Wyszukiwarka pozwoli na podanie zawartości jednego lub więcej pól danych. Po wybraniu przycisku wyszukaj, system wyświetli rezultaty wyszukiwania w postaci listy lokalizacji, spełniających kryteria podane w wyszukiwarce.	Konsultant
6.0	Przeszukiwanie bazy produktów po dostępnych polach	Konsultant z odpowiednimi uprawnieniami ma możliwość przeszukiwania bazy produktów po dostępnych	Konsultant

		polach	
7.0	Wyświetlanie listy zgłoszeń	System wyświetli aktualną listę zgłoszeń serwisowych, po tym jak Konsultant naciśnie na pulpicie pracy przycisk lista zgłoszeń. Na głównym ekranie listy będą wyświetlone podstawowe dane zgłoszenia (data, produkt, zgłaszający, status, odpowiedź do). Ikoną "+" Konsultant rozwinie szczegółowe informacje dla danego zgłoszenia. System domyślnie wyświetli zgłoszenia w realizacji.	Konsultant
8.0	Wyszukiwanie zgłoszeń po opisie, urządzeniu, lokalizacji, aktualnym statusie etc.	Konsultant będzie mógł przeszukać listę po wybranych kolumnach (data, status, produkt, zgłaszający) lub podając słowo kluczowe na formularzu wyszukiwania zgłoszeń	Konsultant
9.0	Możliwość wyświetlenia szczegółów zgłoszenia	Po wyborze zgłoszenia z listy, Konsultant powinien móc wyświetlić jego szczegółowe informacje	Konsultant
10.0	Wyświetlanie listy zgłoszeń względem kończących się czasów, wynikających z umów ramowych (SLA)	Konsultant ma możliwość sortowania malejąco i rosnąco według czasów SLA.	Konsultant
11.0	Aktualizacja statusu zgłoszeń	Konsultant ma możliwość zadecydowania, co dalej zrobić ze zgłoszeniem. Za pomocą wcześniej zdefiniowanych "kroków" w <i>workflow</i> może pokierować zgłoszenie do innego statusu.	Konsultant
12.0	Możliwość wysyłania	System powinien automatycznie wysłać	

	powiadomień o zmianach statusów zgłoszenia	powiadomienie do osoby zgłaszającej w chwili zmiany statusu zgłoszenia. Powiadomienie powinno zawierać podstawowe dane takie jak numer, opis, typ, itp.	
13.0	Administracja systemem		
13.1	Samodzielne definiowanie statusów zgłoszeń	Administrator ma możliwość zdefiniowania statusów zgłoszenia (kroków w <i>workflow</i>)	
13.2	Możliwość określania obowiązkowości wypełnia pól w "formatce zgłoszenia"	Administrator w ramach opracowania danego przepływu określa formatkę zgłoszenia przez podanie pól obowiązkowych. Formatka zostanie wyświetlona w ramach przepływu w chwili wyboru typu zgłoszenia, do którego jest przypisana. Warunkiem początkowym jest zaznaczenie wartości "widoczny" w trybie generowania formularza dla usługi SaaS (punkt 3.10) , wtedy pojawi się drugi <i>checkbox</i> "wymagany", przy pomocy, którego określa się czy wypełnienie danej pozycji jest obligatoryjne.	
13.3	Możliwość określania widoczności i edytowalności kolumn na "formatce" zgłoszenia	Jeśli w ramach umowy wymagana jest dostępność serwisu w modelu SaaS, administrator aplikacji może określić wymagane pola na formatce zgłoszeniowej. Proces kreowania rozpocznie się od edycji standardowej formatki zgłoszeniowej, gdzie w trybie edycji przy każdej z pozycji powinien widnieć <i>checkbox</i> "widoczny". Formatka zostanie wyświetlona w	

ramach przepływu w chwili wyboru typu zgłoszenia, do którego jest przypisana	
przypisana.	

3.4. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne stawianie przed projektowanym system obejmują jego cechy wydajnościowe, cechy dotyczące jego niezawodności, poziomu bezpieczeństwa oraz możliwości uruchamiania aplikacji na określonych przeglądarkach internetowych przez określoną - jako minimalną liczbę użytkowników. Zestawienie takich, postulowanych wymagań niefunkcjonalnych prezentowane jest w poniższych tabelach.

Tabela 9. Wymagania niefunkcjonalne – wydajność. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
1	Wydajność	Musi	Jednocześnie może przeglądać portal co najmniej 200 osób na maszynie testowej zgodnej ze specyfikacją:
			Technologia Sandy Bridge E Rdzenie: 6 Wątki: 12 Zegar: 3,2 GHz Pamięć RAM: 64GB DDR3 ECC Dysk: 2x 1TB SATA3 (Raid 1) Karta sieciowa: 1000 Mbit/s
2	Wydajność	Musi	Czas potrzebny od zalogowania do wyświetlenia listy zgłoszeń, przypisanych do użytkownika, nie powinien być dłuższy niż 5 sekund, przy założeniu 2 Mbit łącza na maszynie testowej zgodnej ze specyfikacją: Technologia Sandy Bridge E Rdzenie: 6 Wątki: 12 Zegar: 3,2 GHz Pamięć RAM: 64GB DDR3 ECC Dysk: 2x 1TB SATA3 (Raid 1) Karta siacjowa: 1000 Mbit/s

Tabela 10. Wymagania niefunkcjonalne – niezawodność. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara	
3	Niezawodność – po godzinach biznesowych	Musi	W godzinach 9:00 – 17:00 system powinien być dostępny przez 99% czasu	
4	Niezawodność – po godzinach biznesowych	Musi	W godzinach 17:00 – 9:00 pracy system powinier być dostępny przez 90% czasu	
5	Niezawodność –	Musi	System musi uniemożliwić jednoczesną pracę	

blokowanie obiektów	dwóch osób na tym samym obiekcie. Np. dwie osoby (= dwa loginy) nie mogą równocześnie pracować nad jednym zgłoszeniem.
	W takim przypadku system powinien wyświetlić komunikat dla osoby próbującej edytować obiekt właściwy dla sytuacji według klucza:
	< <obiekt>> zablokowany przez <<login>>.</login></obiekt>

Tabela 11. Wymagania niefunkcjonalne – bezpieczeństwo danych. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
6	Bezpieczeństwo danych	Musi	Wszystkie dane dot. klientów i ich zamówień muszą być zabezpieczone przed dostępem osób trzecich.
7	Bezpieczeństwo danych - login	Musi	Login do systemu powinien zawierać nazwisko i pierwszą literę imienia.
8	Bezpieczeństwo danych - hasło	Musi	Hasło musi składać się z min. 6 znaków w tym przynajmniej jednej (do wyboru) cyfry, znaku specjalnego, wielkiej litery.
9	Bezpieczeństwo danych – reset haseł	Musi	System musi pozwalać na określenie administratorowi częstotliwości zmiany haseł. Domyślnie – zmiana hasła wymuszona, co 3 miesiące.
10	Bezpieczeństwo danych - integralność danych (odporność na niszczenie danych)	Musi	Wszystkie dane muszą być zapisywane na serwerze zapasowym co najmniej raz na 24h.
11	Bezpieczeństwo danych – szyfrowanie bazy danych	Musi	Baza danych musi być zaszyfrowana mechanizmem TDE (<i>Transparent Data Encryption</i>).

Tabela 12. Wymagania niefunkcjonalne – łatwość użytkowania. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
12	Łatwość użytkowania - użytkownik	Powinien	Po 4 godzinnym szkoleniu, użytkownik powinien być w stanie samodzielnie pracować w systemie.
13	Łatwość użytkowania – kluczowy użytkownik	Powinien	Po 16 godzinnym szkoleniu, kluczowy użytkownik (np. kierownik) powinien być w stanie samodzielnie pracować w systemie.
14	Łatwość użytkowania – administrator	Powinien	Po dwutygodniowym (10 sesji po 8h) szkoleniu, administrator powinien być w stanie samodzielnie pracować w systemie.
15	Łatwość użytkowania – rozdzielczość	Powinien	W rozdzielczości ekranu 1024x768 użytkownik powinien widzieć minimum 10 zgłoszeń na liście.
16	Łatwość użytkowania –	Powinien	W rozdzielczości ekranu 1024x768 użytkownik

rozdzielczość	powinien widzieć minimum 20 pól zgłoszenia w
	widoku szczegółowym.

Tabela 13. Wymagania niefunkcjonalne – obsługiwane przeglądarki. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
17	Obsługiwane przeglądarki	Musi	Przynajmniej jedna z wymienionych przeglądarek musi być w 100% kompatybilna: Internet Explorer 7, Opera 9, Chrome 17, Firefox 10.
18	Obsługiwane przeglądarki	Powinien	Oprócz wybranej jednej 100% kompatybilnej przeglądarki, system powinien umożliwić pracę w stopniu podstawowym, w co najmniej dwóch przeglądarkach z listy poniżej: Internet Explorer 7, Opera 9, Chrome 17, Firefox 10.

Tabela 14. Wymagania niefunkcjonalne – możliwości rozbudowy. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
19	Możliwości rozbudowy	Powinien	System powinien umożliwiać łatwą rozbudowę o kolejne funkcjonalności, jak np. wersja mobilna.

Tabela 15. Wymagania niefunkcjonalne – standardy kodowania. Źródło: Opracowanie własne.

Lp.	Cecha	Priorytet	Miara
20	Standard kodowania - platforma	Musi	System musi być wykonany na platformie .NET.
21	Standard kodowania - język	Musi	System musi być wykonany w języku programowania C#.
22	Standard bazy danych – język	Musi	Bazy danych musi być kompatybilna z językiem SQL.
23	Standard bazy danych – dostęp do bazy	Musi	Dostęp do bazy danych musi dać się uzyskać za pomocą oprogramowania Microsoft SQL Server 2008 R2 Express with Service Pack1.

4. Wykorzystane technologie

W niniejszym rozdziale opisane są wszystkie ważniejsze technologie, które zostały wykorzystane do stworzenia prototypu oprogramowania "Service Desk", wraz z uzasadnieniem ich wyboru.

4.1. .NET Framework

W tej części dokumentu zostanie przedstawiono *framework* .NET 4, która został wykorzystany w projekcie.

4.1.1. Przedstawienie

Technologia .NET Framework (w skrócie .NET) została opracowana przez Microsoft. Pierwsze prace rozpoczęto pod koniec lat dziewięćdziesiątych dwudziestego wieku, natomiast pierwszą finalną wersje opublikowano 13 lutego 2002 roku.

.NET jest platformą programistyczną, której dwie podstawowe składowe to środowisko uruchomieniowe CLR (*Common Language Runtime*) oraz podstawowa biblioteka klas, która odpowiada za dostarczenie funkcjonalności dla aplikacji takich jak interfejs użytkownika, dostęp do danych, łączność z bazami danych, kryptografię, aplikacje internetowe, algorytmikę, komunikację sieciową. Programista tworząc aplikacje łączy swój kod źródłowy wraz z bibliotekami dostarczanymi przez .NET.

Do technologii .NET nie jest przypisany żaden konkretny język programistyczny. Pozwala na stosowanie różnych języków, które spełniają określone dla tej technologii standardy, na przykład C#, Visual Basic .NET, C++/CLI, J#, JScript .NET. Niektóre z nich w istotny sposób zmodyfikowano w celu dostosowania do nowej technologii dodając jednocześnie do ich nazw przyrostek .NET (Visual Basic .NET, Delphi.NET).

4.1.2. Historia rozwoju

Pierwsza wersja .NET 1.0 została wprowadzona wraz z Visual Studio .NET w 2002 r. Zawierała pierwszą wersję CLR (*Common Language Runtime*), oraz pierwszą podstawową bibliotekę klas.

Wersja 1.1 (Visual Studio .NET 2003, dystrybuowana wraz z Windows Server 2003) została wydana w kwietniu 2004 r. Zawierała uaktualnienie ASP.NET oraz ADO.NET. Wprowadzała wykonywanie "*side-by-side*", co umożliwiało aplikacjom uruchamianie wielu wersji CLR na jednym komputerze.

Wersja 2.0 (Visual Studio 2005, dystrybuowana wraz z Windows Server 2003 R2) została wydana w listopadzie 2005 r. Wprowadzono nową wersję CLR 2.0, rozszerzono bibliotekę klas o generyczność, generyczne kolekcje oraz znaczne rozszerzenie ASP.NET.

Wersja 3.0 (Visual Studio 2005, dystrybuowana wraz z Windows Vista, Windows Server 2008) została wydana w listopadzie 2006 r. Wersja ta w rzeczywistości była rozszerzoną wersją2.0 o technologie *Windows Presentation Foundation* (WPF), *Windows Communications Foundation* (WCF), *Windows Workflow Foundation* (WF), i *CardSpace*.

Wersja 3.5 (Visual Studio 2008, dystrybuowana wraz z Windows 7, Windows Server 2008 R2) została wydana w listopadzie 2007 r. Dodano do niej nowe funkcjonalności *AJAX-enabled* oraz *LINQ*. Dodatkowo w czerwcu 2009 r. wydano *service pack 1* tej wersji, który wprowadził funkcjonalność *Dynamic Data*, oraz niewielkie inne ulepszenia.

Wersja 4.0 (Visual Studio 2010) została wydana w kwietniu 2010 r. Zawierała kolejną wersję CLR 4, rozszerzała podstawową bibliotekę klas, wprowadzała nowe funkcjonalności *Managed Extensibility Framework* (MEF), *Dynamic Language Runtime* (DLR), i *code contracts*.

Wersja 4.5 (Visual Studio 2012, dystrybuowana wraz z Windows 8, Windows Server 2012) została wydana sierpniu 2012 r. Wprowadziła kolejną wersję CLR 4.5, wsparcie budowy aplikacji dla *Windows Store*, oraz uaktualnienia dla WPF, WCF, WF i ASP.NET.

Poniższy rysunek przestawia historię poszczególnych wersji technologii wraz ze wskazaniem wersji systemu Microsoft Windows, w której są instalowane kolejne wersje technologii .NET.



Rysunek 28. Historia rozwoju .NET. Źródło: [17]

4.1.3. Język pośredni

Technologia .NET umożliwia wykorzystywanie różnych języków programistycznych (np. C#, Visual Basic .NET, C++/CLI, J#, JScript .NET). Kod napisany w dowolnym z tych języków przed jego wykonaniem jest kompilowany do języka niższego poziomu *Common Intermediate Language* (w skrócie CIL, lub po prostu IL, z ang. język pośredni). Silnik .NET (CLR) wykorzystuje tylko kod języka pośredniego. Każdy język wykorzystywany w .NET jest oparty na języku pośrednim, co sprawia, że występują pewne podobieństwa tych języków np. Visual Basic i C# zapewniają praktycznie te same funkcje i wydajność. Dzięki temu aplikacja napisana w C# może korzystać z funkcji dostarczanych VB i odwrotnie.

Kompatybilność między różnymi językami formalizuje *Common Language Specification* (CLS). Przestrzeganie wytycznych CLS gwarantuje, że komponent napisany w jednym z języków .NET może być używany we wszystkich pozostałych. Jedną z części CLS jest *Common Type System* (CTS), który definiuje zasady dotyczące typów danych dzielonych przez wszystkie języki .NET. CTS definiuje także składniki obiektowe takie jak klasy czy metody.



Poniższy rysunek pokazuje w jaki sposób języki .NET są kompilowane do IL. Wszystkie pliki DLL lub EXE napisane w .NET zawierają kod IL.

Rysunek 29. Kompilacja kodu w .NET. Źródło: [18]

CLR uruchamia tylko kod języka pośredniego, z pominięciem informacji, w jakim języku oryginalnie został napisany. Kolejnym krokiem jest przekształcenie kodu na kod maszynowy

odpowiedni dla aktualnej platformy. Krok ten jest wykonywany po uruchomieniu aplikacji, tuż przed jej wykonaniem.

4.1.4. Wspólne Środowisko Uruchomieniowe

Common Language Runtime (CLR, z ang. Wspólne Środowisko Uruchomieniowe) jest to silnik wspierający wszystkie języki .NET. Każdy kod aplikacji .NET jest uruchamiany w CLR zarówno w aplikacjach dla Windows jak i aplikacjach internetowych. Np. gdy przeglądarka internetowa zażąda dostępu do aplikacji stworzonej w .NET serwis jest uruchamiany w środowisku CLR, po wykonaniu kodu otrzymana finalnie strona HTML jest wysyłana do przeglądarki.

CRL nie tylko wykonuje kod, ale dostarcza także szereg usług powiązanych, takich jak weryfikacja, optymalizacja czy zarządzanie obiektami:

- Głęboka integracja języków: kod napisany w VB czy C# oraz wszystkich językach .NET jest kompilowany do języka pośredniego. CLR nie ma dostępu do informacji, w jakim języku aplikacja została oryginalnie napisana. Jest to coś więcej niż kompatybilność języków – jest to integracja różnych języków programowania.
- Wykonanie typu side-by-side: CLR ma możliwość załadowania rożnych wersji tego samego komponentu w tym samym czasie. Można uaktualniać wielokrotnie dany komponent, a aplikacja będzie zawsze korzystać z jego właściwej wersji. W efekcie można mieć zainstalowanych wiele wersji środowiska .NET. W przypadku instalacji nowszych wersji starsza nie jest przez nią zastępowana, dzięki czemu nie ma potrzeby przepisywać od początku działającej na wcześniejszych wersjach środowiska aplikacji.
- Mniej błędów: CLR zapobiega pojawieniu się bardzo dużej ilości błędów, np. przeciwdziała wielu błędom pamięci, które mogą wystąpić w językach niskiego poziomi chociażby w C++.

4.1.5. Biblioteka Klas

.NET posiada olbrzymie repozytorium klas, które zapewniają szeroki zestaw prefabrykowanych funkcjonalności, począwszy od czytania plików XML po wysyłanie wiadomości mailowych. Biblioteka ta jest bardzo wszechstronna, dostępna dla każdego z języków .NET, co wspiera zachowanie ich spójności bez potrzeby instalowania dodatkowych komponentów na komputerze czy serwerze www.

Bibliotekę klas można postrzegać, jako zestaw narzędzi dobrze zaopatrzonego programisty. Filozofią Microsoftu jest zapewnienie szerokiej infrastruktury programistycznej pozostawiając programistom napisanie ściśle biznesowego kodu. Na przykład .NET wspiera rozwiązania dla dostępu do baz danych oraz transakcji współbieżnych zakładając, że setki użytkowników może zażądać dostępu do tej samej strony internetowej w tym samym czasie. Programiście pozostaje jedynie dołączenie danej funkcjonalności do jego aplikacji.

4.1.6. Visual Studio

Visual Studio jest narzędziem deweloperskim, zapewniającym bogate środowisko umożliwiające szybkie tworzenie zaawansowanych aplikacji. Oczywiście aplikacje .NET można tworzyć bez tego narzędzia (np. można pisać kod źródłowy w zwykłym edytorze tekstu, następnie skompilować kod kompilatorem .NET) jednak byłoby to zadanie bardzo żmudne, o wiele bardziej pracochłonne oraz narażone na błędy. Z tego powodu większość programistów .NET korzysta z Visual Studio.

Wybrane funkcje Visual Studio:

- Projektowanie strony: umożliwia stworzenie interaktywnej strony metodą przeciągnij-i-upuść przy pomocy zintegrowanego narzędzia projektowania formularzy stron, bez potrzeby znajomości HTML.
- Automatyczna detekcja błędów: Visual Studio pozwala zaoszczędzić wiele godzin pracy poprzez wykrywanie i zgłaszanie występujących w aplikacji błędów jeszcze zanim aplikacja zostanie uruchomiona. Potencjalne błędy są zgłaszane na bieżąco w czasie pisania kodu.
- Narzędzia do debugowania: pozwalające śledzić kod oraz wartości zmiennych w czasie działania aplikacji internetowej dzięki wbudowanemu w Visual Studio serwerowi www.

4.1.7. Podsumowanie

Wybór technologii .NET na potrzeby niniejszego projektu był spowodowany dużą popularnością tej technologii. Zdobyte doświadczenie może zaowocować w przyszłości przy pracy w komercyjnych projektach. Istotnym atutem tej technologii jest jej uniwersalność, która ułatwia współprace programistom korzystających z wielu różnych języków programowania.

4.2. C#

W tej części dokumentu zostanie przedstawiony język C#, który został wykorzystany w projekcie. Ze względu na dość szeroką i modułową budowę, dokument zostanie podzielony na kilka części.

4.2.1. Przedstawienie języka C#

Prace nad językiem C# zostały podjęte przez firmę Microsoft w celu stworzenia jednolitej platformy skoncentrowanej na programiście. Język ten miał być odpowiedzią na rozwiązanie konkurencji, które szybko zdobyło popularność wśród rzeszy programistów, czyli język Java. C#, tak samo jak jego konkurent, miał być językiem w pełni obiektowym a jednocześnie o spójnej przestrzeni nazw jak i dostępnych metod.

Skoncentrowano się przede wszystkim na zmniejszeniu liczby błędów, które popełnia programista podczas procesu tworzenia i projektowania aplikacji. Wcześniej programista, oprócz pisania własnych modułów, musiał odwoływać się do rozszerzeń pochodzących z Windows API. Niestety takie podejście nie było korzystne z punktu widzenia programisty jak i osób, które musiały administrować daną aplikacja.

Regularnie pojawiające się aktualizacje naprawiały dotychczasowe błędy, ale też przynosiły wiele nowych. Zmagać się z nimi musiał zarówno programista jak i administrator, który podczas wdrażania poprawek do systemów musiał mieć na uwadze oprogramowanie, które funkcjonuje na stacjach roboczych. Dlatego też, kiedy rozpoczynano prace nad językiem C#, wzięto ten problem wziąć pod uwagę.

4.2.2. Historia powstania języka

Historia powstania języka C# jest zawiła przede wszystkim, dlatego że miało na nią wpływ bardzo wiele różnych zdarzeń, mających miejsce podczas tworzenia języka. W styczniu 1999 r. został uformowany przez Andersa Hejlsbera [19][20] zespół mający zbudować i zaprojektować nowy język programowania.

Początkowo język miał nazywać się COOL, czyli "*C-like Object Oriented language*", ale ze względów prawnych pomysł został niestety zarzucony.



Rysunek 30. Notacja języka w muzyce. Źródło: [21]

Pomysł na nowa nazwę dla języka wziął się z inspiracji notacjami, jakie spotyka się w muzyce. Pisząc nuty muzyk używał znaku "Cis" w celu oznaczeniu dźwięku, który miał być odegrany o pół tonu wyżej. Miało to na celu wykazanie, że nowy język ma być następcą dobrze znanego, ale starego już języka C++, w którym znaki "++"oznaczają inkrementacje danej zmiennej o 1. Symbolizowało to rozwój względem języka C.

Informacje o tym, że trwają prace nad nowym językiem programowanie zostały przekazane do wiadomości publicznej w roku 2000 na konferencji *Proffesional Developers Conference*.

4.2.3. C# na tle innych języków

Podczas wyboru języka programowania do napisania aplikacji, która jest przedmiotem tej pracy inżynierskiej stanięto przed poważnym problemem. Sam zespół jak i wiedza była silnie podzielona. Zwolenników pisania w języku C# jak i tych, którzy byli temu przeciwni było równie wielu.

Ostatecznie wybrano język C#, jako ten który odpowiada na potrzeby projektu, jakie muszą być spełnione, czyli:

- szybkość pisania kodu,
- wiedza w zespole,
- dostęp do dokumentacji.

Język ten od momentu powstania wzbudzał wiele kontrowersji, między innymi przez wysokie podobieństwo do języka Java, mimo to sukcesywnie zyskuje na popularności wśród rzeszy programistów. Warto oprzeć się na statystkach, które jasną pokazują, że obecnie udział w światowym rynku języka C# wynosi około 6% [22]. Może nie wydaje się to duża liczba, ale należy mieć na uwadze, że ilość języków, jaka istnieje obecnie na świecie, dochodzi do setek.





Warto zauważyć, że C# ewidentnie znajduje się w kanale wzrostowym zyskując rynek przede wszystkim kosztem Javy, do której przynajmniej w początkowym okresie był bliźniaczo podobny.

4.2.4. Frameworki

Wszelkie funkcje języka C# sterowane są przez system *frameworków*[24]. Inaczej rzecz ujmując każda nowa wydana wersja C# jest połączona z dodatkowym zestawem bibliotek, które pozwalają znacznie przyśpieszyć prace nad oprogramowaniem, jak też znacznie poszerzają możliwości, jakie posiada język oprogramowania. Zamiast pisać jakaś metodę od podstaw, znacznie prościej jest wykorzystać już istniejącą we *frameworku* właśnie.

Zwiększenie efektywności w pisaniu kodu nie zawsze idzie w parze ze zwiększeniem wydajności samej aplikacji. Wiąże się to przede wszystkim z tym, że metody już zawarte we *frameworkach* nie muszą podchodzić do rozwiązania danego problemy w sposób optymalny. Czasami metoda napisana od podstaw, ale wykonująca od początku do końca tylko to co powinna, może być zdecydowanie wydajniejsza niż metoda znajdująca się we *frameworku*. [25]

W przypadku C# dostępna jest już szósta wersja języka, która została opatrzona numerem 4.5.

Wersja	Data	.NET Framework	Visual Studio
C# 1.0	Styczeń 2002	.NET Framework 1.0	Visual Studio .NET
C# 1.2	Maj 2003	.NET Framework 1.1	Visual Studio .NET 2003
C# 2.0	Listopad 2005	.NET Framework 2.0	Visual Studio 2005
C# 3.0	Listopad 2007	.NET Framework 2.0 (oprócz	Visual Studio 2008

Tabela 16. Historia wydań frameworków i języka C#. Źródło: [26][27]

		rozszerzenia LINQ/Query)	Visual Studio 2010
		.NET Framework 3.0 oprócz	
		rozszerzenia LINQ/Query)	
		.NET Framework 3.5	
C# 4.0	Maj 2010	.NET Framework 4	Visual Studio 2010
C# 5.0	Sierpień 2012	.NET Framework 4.5	Visual Studio 2012

Jak widać język rozwija się dość dynamicznie, a najnowsze jego wydanie pochodzi z 2012 roku, czyli jest dość świeże.

Warto wymienić nowe funkcje obecnej wersji języka C# [28] i jego bibliotek:

- Ulepszone środowisko uruchomieniowe CLR i biblioteka klas podstawowych BCL
 - Lepsza wydajność osiągana przez sprawniejszą obsługę procesorów wielordzeniowych
- Instrukcje Lambda
- Niejawna kontynuacja wiesza
- Poprawa w dostępie do danych jak i ulepszone ich mapowanie
 - o Entity Framework
 - \circ WCF
- Udoskonalona obsługa składnika, jakim jest WPF Windows Presentation Foundation
 - o Obsługa wielodotyku zaimplementowana w Windows 7
 - Obsługa SDK programu *Surface* 2.0
 - Ulepszony moduł przydatny do tworzenia aplikacji biznesowych, w tym nowe formanty LOB *Line of Bussines*
 - Udoskonalony moduł WF Workflow Foundation
 - o Większa paleta wbudowanych działań
- WCF Windows Comunication Foundation
 - Odnajdowanie usług
 - o Routing
 - Obsługa protokołu REST
- Łatwiejsze programowanie równoległe
 - Mechanizm TPL Task Parallel Library
 - PLINQ równoległe zapytania LINQ
 - CDS *Coordination Data Structures* umożliwia lepsze wykorzystanie procesorów wielordzeniowych

Przede wszystkim widać jest dążenie platformy do obsługi wielowątkowości, o czym świadczy asynchroniczne wywoływanie metod pomocne przy wielowątkowych programach.

Te i wiele innych funkcji zdecydowały o wyborze języka C#, jako języka, w którym została stworzony program opisywana praca inżynierska.

4.2.5. Common Language Runtime

Common Language Runtime (CLR) środowisko uruchomieniowe dla języka C# i całej platformy .NET. Należy to traktować, jako maszynę wirtualną, która pozwala na uruchomienie tego samego kodu na wielu różnych platformach.

Program napisany w C# powinien się uruchomić, a wszelkie odwołania powinien wykonywać bezpośrednio do tej maszyny wirtualnej. Stanowi ona swego rodzaju warstwę pośrednią tłumaczącą rozkazy systemu i procesora na informacje zrozumiale dla danej aplikacji.



Rysunek 32. Common Language Runtime. Źródło: [29]

W obecnych czasach bardzo ważne jest bezpieczeństwo danych jak i samej firmy. Oznacza to tylko jedno, dany program też powinien zapewniać i cechować się wysokim poziomem ochrony danych, na których pracuje, oraz to, że nie powinien posiadać słabych punktów, które nieuprawnionej osobie umożliwiłyby np. włamanie się do aplikacji i przechwycenie istotnych z punktu widzenia przedsiębiorstwa informacji. Żeby dodatkowo zabezpieczyć się przed takimi wypadkami, CLR ma wbudowane wiele różnych mechanizmów maksymalizujących ochronę danych. Dodatkowo sam w sobie posiada wiele gotowych modułów, które właściwie wykorzystane pozwalają podnieść poziom bezpieczeństwa pracującej aplikacji.

Są nimi:

- CAS [30][31] *Code Access Security* umożliwia zdefiniowanie w kodzie źródłowym programu odpowiednich uprawnień
- RBS [32][33] *Role-Based Security* operuje na danych jakie uzyskuje podczas uwierzytelnienia użytkownika miedzy innymi informacje o rolach jakie posiada dany użytkownik

Dodatkowo należy pamiętać o miesięcznych, dostarczanych przez Microsoft, biuletynach bezpieczeństwa [34], poprawkach i aktualizacjach do całego środowiska uruchomieniowego.

Ponownie C# udowadnia, że jest językiem, w którym pisanie nawet *wymagających aplikacji* jest słusznym wyborem, który może przynieść wiele korzyści w przyszłości.

4.2.6. Ciekawe funkcje – LINQ



Rysunek 33. Podstawowy schemat przedstawiający funkcje danych słów w LINQ. Źródło: [35]

Jest to unikalna funkcjonalność języka c# pozwala na wykonywanie zapytań do kolekcji obiektów za pomocą dedykowanego zapisu. Zapytania te przypominają swoją składnia język SQL i mają podobne możliwości.

Zamiast wykonywać zapytanie do bazy danych, które jest dość kosztowne obliczeniowo – wymaga ustanowienia połączenia, wysłania zapytania, następnie aplikacja musi oczekiwać na bazę danych aż ta jej zwróci wynik i dopiero, kiedy go dostanie jest w stanie pójść dalej w przetwarzaniu danych. Wiąże się to z często dużym obciążeniem sieci jak i dana infrastruktura IT w firmie musi być skonstruowana tak, aby opóźnienia były jak najniższe. Dużo lepiej to wygląda w LINQ on operuje na obiektach znajdujących się już w pamięci serwera, a pamięć RAM jest jedną z najszybszych i posiadających najniższe opóźnienia w porównaniu z dyskami twardymi. Bardzo często zdarza się że zapytanie wykonane w LINQ może wykonać się szybciej niż zapytanie do bazy danych.



Rysunek 34. Schemat zintegrowania LINQ z platforma .NET. Źródło: [35]

Nie bez znaczenia jest głęboka integracja z platforma .Net i ciągły rozwój tego języka. Z jego użyciem można napisać aplikacje, która w ogóle nie będzie musiała odwoływać się do bazy danych za pomocą tradycyjnego języka SQL.

4.3. Model View Controller

Model View Controller (MVC) został wymyślonych przez firmę Xerox w roku 1979. Za autora uznaje się Szweda Trygve Reenskaug. Pierwszy raz został wykorzystany również przez firmę Xerox do tworzenia interfejsów użytkownika powstałych w języku Smalltalk. Wzorzec został utworzony z myślą o aplikacjach i interfejsach sieciowych.[36]

MVC jest to wzorzec architektoniczny stosowany w projektowaniu systemów informatycznych dla oprogramowania wykorzystującego interfejs graficzny. Składa się on z 3 głównych części:

1. Model – reprezentuje model danych, fragment logiki. Najczęściej jest programistycznym odwzorowaniem struktury trwałości danych. Model jest samodzielnym bytem i w obrębie jednej aplikacji może występować wiele różnych modeli. Zaleca się, aby model był jedynym elementem oprogramowania, który ma związek z trwałością danych.

2. Kontroler – zarządza logiką aplikacji, reaguje na dane wejściowe i steruje procesami. Odpowiada za zarządzanie modelami oraz generowanie odpowiednich widoków. Można przyjąć, że jest on najważniejszą częścią, jednakże do poprawnego działania wzorzec musi zawierać wszystkie trzy elementy.

3. Widok – definiuje warstwę prezentacji. Widok może składać się z wielu mniejszych części składowych, co ma na celu wyeliminowanie redundancji kodu. Klasyczne wykorzystanie wzorca MVC zakłada, że widok komunikuje się bezpośrednio z modelem w celu pozyskania danych do prezentacji. Jednakże w realiach webowych przyjmuje się bardzo często wariację MVC, która polega na tym, że to kontroler odpowiada za pobranie niezbędnych danych z modeli i przekazuje je do widoku. Rolą widoku w takim wypadku jest wyłącznie wyświetlenie danych według zadanego schematu.[37]



Rysunek 35. Schemat MVC. Źródło: [38]

MVC został wykorzystany w niniejszej pracy z poniższych powodów:

- Zapewnia natywne wsparcie wzorca w projektach ASP .Net
- Pozwala na wygodny podział pracy. Wyznaczeni programiści nie muszą się przejmować warstwą prezentacji, odpowiadają tylko za mechanikę i na niej się skupiają. Osoby odpowiedzialne za interfejs mają zapewnione niezbędne dane, na których pracują. Bazodanowcy zapewniają spójność bazy z modelami, na których pracują pozostali. Pozwala to na zrównoleglenie prac, co przyspiesza tempo prac i zwiększa wydajność zespołu.
- Popularność wzorca pozwala na szybkie wdrożenie osób o mniejszych umiejętnościach i zasobie wiedzy.

4.4. Windows Workflow Foundation

W niniejszym rozdziale opisana została technologia Windows Workflow Foundation, która posłużyła za silnik dla modelowania *workflow* w projekcie.

4.4.1. Technologia

Windows Workflow Foundation (WF) został wydany wraz z pojawieniem się .NET Framework 3.0 w 2006 roku, a następnie zaktualizowany rok później w. NET Framework 3.5. Począwszy od wersji środowiska programistycznego Visual Studio 2005 jest jego integralną częścią. Technologia ta wspiera aplikacje tworzone dla systemów Windows oraz aplikacje internetowe. Tworzenie aplikacji z
wykorzystaniem WF pozwala na definiowanie przepływów informacji (*workflow*) w systemie w sposób przejrzysty i jednoznaczny.

Oczywiście, cały mechanizm sterowania przepływem informacji można stworzyć w kodzie aplikacji bez korzystania z WF, jednakże w przypadku częstych zmian w przepływach (np. dodanie dodatkowej osoby na ścieżce akceptacji) zaczyna to być problematyczne. Windows Workflow Foundation wspiera użytkowników w takich przypadkach przejrzystym zintegrowanym środowiskiem programistycznym (ang. *Integrated Development Environment*, IDE), gdzie za pomocą opcji przeciągnij-i-upuść możemy dodawać kolejne aktywności.





Rysunek 36. Przykład scenariusza workflow. Źródło: Opracowanie własne.

Na *workflow* składają się pojedyncze aktywności, połączone ze sobą ścieżkami. Budowanie tego typu przepływu sprowadza się do przeciągnięcia z menu narzędziowego aktywności i sparametryzowania ich zakładce właściwości. Może się zdarzyć, że będzie potrzebna dedykowana aktywność, której w danej chwili nie ma wśród predefiniowanych narzędzi. Nową aktywność można dopisać w kodzie i dołączyć ją do sekcji narzędzi, w celu jej wielokrotnego użycia przy kolejnych przepływach. Przypomina to tworzenie własnej kontrolki *UserControl*. Tworzenie nowych narzędzi wymaga umiejętności programistycznych, dlatego jest to zadanie dedykowane dla przyszłych administratorów aplikacji.



Rysunek 37. Przykład dedykowanej aktywności. Źródło: Opracowanie własne.

Podczas tworzenia *workflow*, opisywana technologia udostępnia takie aktywności, jak *If*, *Switch, While, ForEach, Try, Catch.* Dodatkowo istnieją także aktywności, które odpowiadają za transakcyjność, współbieżne przetwarzanie czy wysyłanie wiadomości e-mail. Do bardziej skomplikowanych zadań można wykonać fragment naszego kodu. Przy bardziej złożonych przepływach, które mogą trwać dłuższy okres czasu np. kilka dni, wykorzystywana jest maszyna stanu. Jej zadaniem jest przechowywanie np. w postaci usługi persystancji³ aktualnego stanu przepływu (np. przy obsłudze płatności, zamówień) tak aby po zajściu określonej czynności proces mógł przejść do kolejnego etapu.

Podsumowując, technologia Windows Workflow Foundation znacznie ułatwia budowanie przepływów pracy i późniejsze ich modyfikowanie, co ma pozytywny wpływ na jakość tworzonego oprogramowania.

4.4.2. Workflow Designer

Narzędzie Worklow Designer (WD) zostało wykorzystane na potrzeby tworzenia przepływów informacji. Jest to oddzielna aplikacja zbudowana na bazie Re-Hosting Workflow Designer. Jej niezależność polega na tym, że nie wymaga zainstalowania środowiska programistycznego Visual Studio. Administrator aplikacji Service Desk otrzymuje gotowe narzędzie dzięki któremu, w sposób identyczny jak to się odbywa w środowisku Visual Studio, buduje lub zmienia przepływy informacji. Narzędzie WD zostało zmodyfikowane na potrzeby projektu. Dodano dedykowane aktywności, które są używane w przepływach, zarządzanych przez aplikację Service Desk.

³ Persystancja - trwałe przechowywanie, które eliminuje ryzyko utraty informacji.

PJWSTK Workflow Designer	
Plik Widok Pomoc	
Toolbox	- ↓ ×
Search	
▲ Kontrolki	
💏 If	
🐒 DoWhile	
While	
Flowchart	
FlowDecision	
FlowSwitch<>	
E Formularz	
E-mail	
Pytanie tak/nie	
🚺 Komunikat	

Rysunek 38. Widok zmodyfikowanego *toolboxa* Workflow Designer. Źródło: Opracowanie własne.

Dodatkowa opcja [Plik] \rightarrow [Wyślij na serwer] umożliwia wysłanie stworzonego lub zmodyfikowanego przepływu bezpośrednio do aplikacji Service Desk. Tak utworzony przepływ może być od razu podpięty pod dany typ zgłoszenia.

4.5. Microsoft Entity Framework

Entity Framework jest technologią, której nadrzędnym celem jest wsparcie całego procesu wytwarzania oprogramowania. Jego użycie jest szczególnie wskazane dla aplikacji, których działanie opiera się w głównej mierze na operacjach na danych.

Dotychczas programiści i architekci aplikacji musieli stawiać czoła wielu wyzwaniom takim jak np. modelowanie powiązań, jakie występują pomiędzy tabelami w bazie danych, żeby móc później zaimplementować logikę biznesową. Innym wyzwaniem było odpowiednie zarządzanie kodem, który musiał odpowiednio komunikować się z wybranymi i skomplikowanymi silnikami, odpowiedzialnymi za transport danych pomiędzy samą aplikacją a bazą danych.

Rozwiązaniem powyższych kwestii i problemów z nimi związanych jest podejście do pracy z danymi, jakie oferuje programiście *Entity Framework*. Umożliwia on na prace bezpośrednio na obiektach i atrybutach, które posiada, przez co jest to inne podejście względem tradycyjnej pracy na tabelach i ich kolumnach połączonych zdefiniowanymi wcześniej relacjami.

Oznacza to tylko jedno - mniej kodu, który trzeba napisać, aby aplikacja była w stanie wykonywać to, czego chce programista. Często prowadzi to do szybszego powstawania aplikacji, co prowadzi do znacznego obniżenia kosztów wytwarzania oprogramowania.

4.5.1. Zarys rozwiązania

Entity Framework potrzebuje odpowiednich plików do właściwego pobrania danych z bazy i zmapowania ich na odpowiedni model. Wszelkie zapytania do modelu danych mogą być wykonywane w różny sposób:

- LINQ używany jest to generowania zapytań w modelu koncepcyjnym
- Entity SQL umożliwia wykonywane zapytań jak w tradycyjnym SQL. Różnice najlepiej ilustruje przykład:
 - Tradycyjnie SELECT a FROM b
 - Entity SQL SELECT b.a as A from B as b

Dostawcą danych w *Entity Framework* jest *Entity Client*, który zapewnia właściwą komunikację oraz tłumaczy wykonywane zapytania np. w LINQ na odpowiednie zapytania SQL, które mogą być wykonane na bazie danych.



Rysunek 39. Entity Framework - przepływ danych. Źródło: [39]

Klasy takie jak *System.Data.Objects.ObjectContext* i *System.Data.Entity.DbContext* odpowiedzialne są za właściwą reprezentacje kontenera encji w modelu koncepcyjnym. Klasy te oprócz odpowiedniego zarządzania m.in. relacjami odpowiedzialne są też za wykonanie *insertów*, *update* 'ów czy *delete* z bazy danych za pośrednictwem metody *SaveChanges*.

4.5.2. Przykładowe mapowania

Entity Framework zajmuje się odpowiednią interpretacją relacji, jakie zachodzą w bazie danych. Poniżej przedstawiono przykładową realizację mapowania wiele do wiele w bazie danych:



Rysunek 40. Asocjacja wiele do wiele w tradycyjnej bazie danych. Źródło: [40]



Jej zmapowana postać za pośrednictwem Entity Framework ma postać jak na rysunku 41.

Rysunek 41. Asocjacja wiele do wiele z użyciem Entity Framework. Źródło: [41]

Jak widać, tabela łącząca pracowników z projektami została ukryta i pokazywane są tylko główne tabele, czyli projekty i pracownicy. Programista operując tylko na obiektach może dostać się zarówno do wszystkich projektów danego pracownika jak i listy pracowników, do których został przypisany dany projekt.

4.5.3. Nowe podejście do tworzenia aplikacji – Code-First

Wcześniejsze metody tworzenia aplikacji opierały się na koncepcji *Database–First*, która oznaczała tworzenie na początku bazy danych. Wiąże się to z wieloma zaletami. Programista w momencie rozpoczęcia prac ma już cały obraz bazy danych, może także mieć już przygotowane zapytania i widoki, na których może operować. Jednak wszelkie modyfikacje są utrudnione np. modelu bazy danych. Konieczna jest właściwa komunikacja z osoba odpowiedzialna za bazę danych, a następnie poprawki w kodzie programu. Powoduje to liczne opóźnienia i problemy, przez co sama aplikacja powstaje dłużej niż zakładano.

Model *First* - ten rodzaj tworzenia aplikacji i bazy danych polega na zamodelowaniu najpierw modelu klas w *designerze*, który jest już zaimplementowany w *Entity Framework*. Następnie możliwe jest zmapowanie tego modelu na istniejącą bazę danych, ale też stworzenie bazy danych od podstaw na bazie stworzonego wzorca.

Alternatywnym podejściem do tworzenia aplikacji jest wykorzystanie *Code-First*, czyli tworzenia najpierw kodu. W takim przypadku nie ma potrzeby tworzenia odrębnego modelu danych w wyspecjalizowanym *designerze*. Jest to zarówno szybkie jak i elastyczne rozwiązanie. Programiści, tworząc obiekty w aplikacji, jednocześnie tworzą odpowiednie mapowania po stronie bazy danych.

Przykład takiego mapowania z kodu programu Service Desk przedstawiono w listingu nr 1.

Listing 1. Klasa *Employee* z mapowaniem po stronie bazy danych. Źródło: Opracowanie własne.

```
namespace PJWSTK.Serwis.Models
{
    /// <summary>
    /// Klasa reprezentująca pracownika
    /// </summary>
    public class Employee : Person
    {
        #region Properties
        /// <summary>
        /// PESEL pracownika
        /// </summary>
        [Display(Name = "PESEL")]
        [MaxLength(11)]
        [Required(ErrorMessageResourceType
                                                    typeof(ErrorResources),
                                            =
ErrorMessageResourceName = "Employee PeselRequired")]
        public string Pesel { get; set; }
    }
```

Ze względu na szybkość tworzenia oprogramowania, ten sposób kodowania wybrano podczas tworzenia tej pracy inżynierskiej. W przeciwnym razie wymagałoby to wcześniejszego utworzenia samej bazy danych lub też jej modelu, co w konsekwencji wymaga dodatkowego nakładu czasu.

4.6. Bootstrap

Bootstrap jest to zestaw narzędzi do tworzenia aplikacji oraz stron internetowych stworzony przez programistów Twittera Marka Otto i Jacoba Thorntona. Technologia została stworzona na wewnętrzne potrzeby firmy w celu wyeliminowania niespójności powstających przy stosowaniu różnych technologii w czasie tworzenia aplikacji. Twórcy początkowo podzielili się nowym narzędziem ze swoimi współpracownikami pokazując jak mogą oni przyspieszyć tworzenie nowych projektów. Następnie w sierpniu 2011 roku opublikowali pierwszą wersje, jako *open-source*.

Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.



By nerds, for nerds.

Built at Twitter by @mdo and @fat, Bootstrap utilizes LESS CSS, is compiled via Node, and is managed through GitHub to help nerds do awesome stuff on the web.



Made for everyone.

Bootstrap was made to not only look and behave

great in the latest desktop browsers (as well as IE7!),

but in tablet and smartphone browsers via responsive

CSS as well



Packed with features. A 12-column responsive grid, dozens of components, JavaScript plugins, typography, form controls, and even a web-based Customizer to make Bootstrap your own.

Rysunek 42. Strona tytułowa Bootstrap. Źródło: [42]

4.6.1. Przegląd technologii

Aby rozpocząć prace z technologią *Bootstrap* wymagana jest znajomość HTML oraz CSS. Wystarczy udać się na stronę projektu http://twitter.github.io/bootstrap/index.html i pobrać aktualną wersję. Po rozpakowaniu archiwum otrzymujemy logicznie pogrupowana zawartość z następującą strukturą plików.



Rysunek 43. Logicznie pogrupowana struktura plików Bootstrap. Źródło: [43]

Bootstrap jest wyposażony w wiele gotowych rozwiązań HTML, CSS oraz JS podzielonych na cztery sekcje:

- *Scaffolding* globalne style dla całej strony tj. tło, style linków, grid, oraz zawiera dwa proste layout'y
- *Base CSS* style dla typowych elementów HTML tj. tabele, formularze, przyciski oraz zestaw ikon.
- *Components* podstawowe style komponentów interfejsu tj. zakładki, paski nawigacyjne, nagłówki stron, alerty.
- *JavaScript plugins* interaktywne komponenty tj. okna modalne, dymki z podpowiedziami, wyskakujące okienka i wiele innych.

Lista komponentów wspólna dla Components i JavaScript plugins:

- Przyciski rozwijane
- Grupy przycisków
- Przyciski nawigacyjne
- Paski nawigacyjne
- Etykiety
- Oznaczenia
- Nagłówki
- Miniatury
- Powiadomienia
- Paski postępu
- Okna dialogowe
- Podpowiedzi
- Accordion
- Carousel
- Typeahead

Wdrożenie technologii *Bootstrap*, na przykładzie prostego szablonu strony HTML, wygląda jak na poniższym rysunku.

```
1.
      <!DOCTYPE html>
 2.
      <html>
        \langle head \rangle
4.
          <title>Bootstrap 101 Template</title>
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
 5.
        </head>
 7.
        <body>
8.
          <h1>Hello, world!</h1>
9.
          <script src="http://code.jquery.com/jquery.js"></script>
10.
        </body>
11. </html>
```

Rysunek 44. Przykładowy plik HTML z Bootstrap. Źródło: [43]

1.	html
2.	<html></html>
з.	<head></head>
4.	<title>Bootstrap 101 Template</title>
5.	<pre><meta content="width=device-width, initial-scale=1.0" name="viewport"/></pre>
6.	Bootstrap
7.	k href="css/bootstrap.min.css" rel="stylesheet" media="screen">
8.	
9.	<body></body>
10.	<h1>Hello, world!</h1>
11.	<script src="http://code.jquery.com/jquery.js"></script>
12.	<script src="js/bootstrap.min.js"></script>
13.	
14.	

Rysunek 45. Dodanie szablonu Bootstrap do pliku HTML. Źródło: [43]

Po dołączeniu właściwych plików CSS oraz JS można rozwijać dowolna stronę lub aplikację z technologią *Bootstrap*.

Marketing stuff!

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Get started today

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

View details »

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

View details »

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa.

∨iew details »

Rysunek 46. Przykładowy dostępny układ strony. Źródło: [43]

4.6.2. Grid

Bootstrap zawiera 12 kolumnowy *CSS grid* (stały lub płynny), oraz zestaw styli służący do ładnego wyświetlania całości na różnej wielkości ekranach (*Responsive design*). *Grid* dostosowuje się do szerokości pomiędzy 724 a 1170 pikseli w zależności od rozdzielczości ekranu.

Live grid example

The default Bootstrap grid system utilizes **12 columns**, making for a 940px wide container without responsive features enabled. With the responsive CSS file added, the grid adapts to be 724px and 1170px wide depending on your viewport. Below 767px viewports, the columns become fluid and stack vertically.



Basic grid HTML

For a simple two column layout, create a .row and add the appropriate number of .span* columns. As this is a 12-column grid, each .span* spans a number of those 12 columns, and should always add up to 12 for each row (or the number of columns in the parent).



Given this example, we have .span4 and .span8, making for 12 total columns and a complete row.

Rysunek 47. Przykładowy Bootstrap grid. Źródło: [44]

Aby uzyskać układ strony złożony z dwóch kolumn wystarczy złączyć odpowiednio np. 4 i 8 kolumn, pamiętając, aby ich suma nadal wynosiła 12.

Dzięki technologii *Response design* możemy uzyskać pożądany wygląd naszej strony czy aplikacji bez względu na to czy będzie wyświetlana na telefonie, tablecie czy ekranie monitora.



Rysunek 48. Technologia Response design na różnych urządzeniach. Źródło: [44]

W zależności o rozdzielczości ekranu będzie zmieniała się szerokość kolumn, elementy strony będą różnie grupowane. Wielkość nagłówków oraz tekstu zostanie zmieniona w zależności od urządzenia, na którym będzie wyświetlana.

4.6.3. Base CSS

Bootstrap zawiera szeroki zakres zdefiniowanych styli dla poszczególnych elementów HTML. Pozwala to tworzyć bardzo zaawansowane aplikacje bez konieczności budowania własnych struktur, wystarczy poszczególnym elementom nadać odpowiednią klasę. Zmiana wyglądu danego elementu sprowadza się do zmiany jego klasy.

Button sizes

Fancy larger or smaller buttons? Add .btn-large, .btn-small, or .btn-mini for additional sizes.

Examp	
	Large button
D	efault button Default button
Sr Mini	nall button Small button button Mini button
1	
- 1. 	<pre>chutton class="http://www.second.com/manu".tupe="hutton"slange.hutton://wuttons.</pre>
	<pre>chutton class="bth bth-large" type="button">Large button chutton class="bth bth-large" type="hutton">Large button</pre>
4.	
5.	(b)
6.	<pre><button class="btn btn-primary" type="button">Default button</button></pre>
7.	<pre><button class="btn" type="button">Default button</button></pre>
8.	
9.	
10.	 button class="btn btn-small btn-primary" type="button">Small button
11.	
12.	
13.	
14.	<pre><button class="btn btn-mini btn-primary" type="button">Mini button</button></pre>
15.	<pre><button class="btn btn-mini" type="button">Mini button</button></pre>
16.	



Input with warning	Something may have gone wrong
Input with error	Please correct the error
Input with info	Username is taken
Input with success	Woohoo!

Rysunek 50. Przykładowy efekt styli walidacji wprowadzanych danych. Źródło: [45]

4.6.4. Dostosowanie

Bootstrap umożliwia wygenerowanie sobie spersonalizowanej paczki, zawierającej tylko potrzebne nam elementy, a nawet przebudowany na nasze potrzeby CSS.

1. Choose components

Scaffolding	Components	Miscellaneous
✓ Normalize and reset	Button groups and dropdowns	💟 Wells
✓ Body type and links	✓ Navs, tabs, and pills	🛃 Close icon
🖌 Grid system	V Navbar	✓ Utilities
✓ Layouts	🕑 Breadcrumbs	✓ Component animations
Base CSS	✓ Pagination	Responsive
🛃 Headings, body, etc	✓ Pager	Visible/hidden classes
✔ Code and pre	Thumbnails	✓ Narrow tablets and below (<767p×)
✓ Labels and badges	✓ Alerts	▼ Tablets to desktops (767-979 px)
▼ Tables	✓ Progress bars	✓ Large desktops (>1200p×)
✔ Forms	🖌 Hero unit	🖌 Responsive navbar
V Buttons	🖌 Media component	
V Icons	JS Components	
	Tooltips	
	Popovers	
	V Modals	
	✓ Dropdowns	
	🔽 Collapse	
	🕑 Carousel	

2. Select jQuery plugins

Toggle all

✓ Transitions (required for any animation)	Affix	Heads up!
🖌 Modals	☑ Alert messages	All checked plugins will be compiled into a single file, bootstrap is. All plugins require
🔽 Dropdowns	Puttons	the latest version of jQuery to be included.

Rysunek 51. Dopasowanie komponentów Bootstrap. Źródło: [46]

4.6.5. Przykłady

Prostota i wygoda stosowania tej technologii może sprawić, że niebawem powstanie wiele stron o podobnym wyglądzie.



Rysunek 52. Przykład strony wykorzystującej technologię Bootstrap. Źródło: [47]

Slow Food®Columbus



Rysunek 53. Przykład strony wykorzystującej technologię Bootstrap. Źródło: [48]

4.7. Microsoft SQL Server

Microsoft SQL Server jest systemem służącym do zarządzania bazami danych. Został on stworzony w 1989 przez firmę Microsoft. Wykorzystuje język zapytań Transact SQL, który jest rozszerzeniem języka SQL (*Structured Query Language*).

SQL jest językiem służącym do tworzenia, zarządzania i modyfikowania relacyjnych baz danych.

4.7.1. Relacyjny model danych

Pierwszy raz określenie to zostało użyte przez Edgara Codda w 1970 roku [49]. Podstawowym elementem tego modelu danych jest tabela, która ma określoną liczbę kolumn. Każda kolumna ma osobną nazwę i typ danych, który określa, jakiego rodzaju wartości mogą być w niej umieszczane. Każdy wiersz jest jednym rekordem informacji.

Ważną rzeczą w relacyjnym modelu danych są klucze. Służą one do zachowania unikalności poszczególnych rekordów. Każda tabela musi posiadać klucz główny, który ma unikalną wartość, pozwalającą zidentyfikować cały wiersz. Tabele mogą też posiadać klucze obce, które są wskaźnikami do wierszy w innych tabelach.[50]



Rysunek 54. Przykładowy schemat bazy danych. Źródło: Opracowanie własne.

4.7.2. Historia SQL

Język SQL został stworzony w latach 70 w firmie IBM przez Donalda D. Chamberlaina i Raymonda F. Boyce. Początkowo nosił nazwę SEQUEL (*Structured English Query Language*), jednak w związku z problemami związanymi z prawami do nazwy, została ona zmieniona na SQL. Jeszcze w latach 70 został wykorzystany przez firmę Relational Software w Oracle V2. Później, w latach 80 stał się standardem w Amerykańskim Narodowym Instytucie Normalizacji (ANSI) i Międzynarodowej Organizacji Normalizacyjnej (ISO).

4.7.3. Składnia SQL

Tabela 17. Podstawowe klauzule. Źródło: Opracowanie na podstawie [51]

Klauzula	Opis		
FROM	Definiuje z jakiej tabeli pobieramy dane		
WHERE	Określa warunki pozwalające przefiltrować uzyskany wynik		
ORDER BY	Porządkuje dane względem kolumny		

GROUP BY	Grupuje wyniki, które w podanych kolumnach się powtarzają
HAVING	Pozwala filtrować pogrupowane dane

			,				
T 1 1	10	\mathbf{A}	77 / 11	\sim	•	1	F / 1 1
I abala	1 2	Imaratory	Irodio	Inno	001100010 00	nodatowno	1211
	10.		Z .I C IC I I C I	<u>()</u>	uuwanne na	1 IRRESIAWIE	
I uoviu	10.	operatory	LIUGIU.	Opra	comunic in		1211

Operator	Opis		
=	Równe		
<> lub !=	Nierówne		
>	Większe niż		
<	Mniejsze niż		
>=	Większe bądź równe		
<=	Mniejsze bądź równe		
BETWEEN	Pomiędzy danymi wartościami		
LIKE	Szuka danych pasujących do określonego wzorca		
IN	Pozwala podać wiele różnych wartości dla kolumny		

Zapytania w języku SQL standardowo pisane są wielkimi literami i kończy się je średnikiem. Dzielą się na pięć podstawowych typów: DML, DDL, DCL I DQL. [52]

- DML (Język manipulacji danymi *Data Modification Language*) dotyczy tworzenia, modyfikacji i kasowania rekordów w bazie. Przykładowe komendy to: *UPDATE*, *INSERT*, *DELETE*.
- DDL (Język definicji danych *Data Definition Language*) dotyczy tworzenia, modyfikacji i kasowania obiektów w bazie. Przykładowe komendy to: *ALTER VIEW*, *CREATE TABLE* czy *DROP*.
- DCL (Język kontroli nad danymi *Data Control Language*) komendy dotyczące kontroli uprawnień użytkowników do baz, tabel, obiektów i rekordów w bazie. Przykładowe komendy to: *GRANT*, *DENY*, *REVOKE*.
- TCL (Język kontroli transakcji *Transaction Control Language*) komendy dotyczące obsługi transakcji w bazie. Przykładowe komendy to: *BEGIN TRANSACTION*, *COMMIT* czy *ROLLBACK*.
- DQL (Język definiowania zapytań *Data Querying Language*) komendy dotyczące ekstrakcji danych z bazy (kwerend), a dokładnie polecenie *SELECT*.

Poniżej zaprezentowano przykłady pokazujące zastosowanie poszczególnych typów zapytań SQL. Do tego celu użyto tabeli *REQUEST*, posiadającej dwie kolumny z danymi: *STATUS* i *REQUEST_ID*.

DML jest używany do wprowadzania zmian w danych. Rysunek 55 przedstawia przykładowe

UPDATE REQUEST SET STATUS = 'A' WHERE REQUEST_ID = 1234;

zastosowanie instrukcji Update.

Rysunek 55. Przykład zapytania Update. Źródło: Opracowanie własne.

DDL pozwala tworzyć, kasować i zmieniać bazę danych. Rysunek 56 przedstawia przykładowe zastosowanie instrukcji *Create*.

CREATE TABLE REQUEST(REQUEST_ID INT STATUS VARCHAR(3) PRIMARY KEY (REQUEST_ID));

Rysunek 56. Przykład zapytania Create. Źródło: Opracowanie własne.

DCL służy do zarządzania uprawnieniami użytkowników (np. GRANT). Rysunek 57 przedstawia przykładowe zastosowanie instrukcji *Grant*.

GRANT SELECT ON REQUEST TO Userl;

Rysunek 57. Przykład zapytania Grant. Źródło: Opracowanie własne.

TCL umożliwia przeprowadzania skomplikowanych operacji na bazie bądź anulowania ich poprzez stosowną komendę zakończenia. *Commit* zatwierdza zmiany, *Rollback* zaś wycofuje je w razie potrzeby.

DQL jest podstawowym narzędziem do odpytywania bazy danych (SELECT). Rysunek 58 przedstawia przykładowe zastosowanie instrukcji *Select*.

SELECT * FROM REQUEST;

Rysunek 58. Przykład zapytania Select. Źródło: Opracowanie własne.

4.7.4. Transact SQL

Język Transact SQL jest rozwinięciem języka SQL. Daje możliwość tworzenia zmiennych, pętli i instrukcji warunkowych. Bardzo ważnymi funkcjami tego języka są też procedury i wyzwalacze. Procedura jest to niewielka część kodu wykonująca jakieś operacje. Wyzwalacz zaś jest procedurą uruchamianą w związku z jakimś wydarzeniem w bazie danych. W języku tym nie istnieje określona obowiązkowa konstrukcja kodu. Nie trzeba też kończyć go średnikiem.[53]

4.7.5. Historia MS SQL

Pierwsza wersja Microsoft SQL Server została wypuszczona na rynek w roku, 1989 jako konkurencja dla rozwiązań Oracle i IBM. Była ona zbudowana na podstawie rozwiązań dostarczonych w wydanym w 1984 systemie Sybase SQL Server przez firmę Sybase, od której Microsoft wykupił prawa. Od tamtej pory system jest ciągle rozwijany. Jak w wypadku większości swoich rozwiązań, Microsoft dostarcza system w wielu różnych odmianach, dzielących się na dwie grupy: dla zwykłego odbiorcy (np. *Enterprise, Standard*) i specjalistyczną (np. *Developer*).

Wersja	Rok	Nazwa	Nazwa Kodowa
1.0 (OS/2)	1989	SQL Server 1.0 (16 bit)	-
1.1 (OS/2)	1991	SQL Server 1.1 (16 bit)	-
4.21 (WinNT)	1993	SQL Server 4.21	SQLNT
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra

Tabela 19. Kolejne wydania SQL Server. Zrodło: 54	Tabela	19.	Kolejne	wydania	SQL	Server.	Zródło:	[54	1
---	--------	-----	---------	---------	-----	---------	---------	-----	---

7.0	1998	SQL Server 7.0	Sphinx
-	1999	SQL Server 7.0 OLAP Tools	Palato mania
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64-bit Edition	Liberty
9.0	2005	SQL Server 2005	Yukon
10.0	2008	SQL Server 2008	Katmai
10.25	2010	SQL Azure DB	CloudDatabase
10.5	2010	SQL Server 2008 R2	Kilimanjaro
11.0	2012	SQL Server 2012	Denali

4.7.6. Opis Systemu

MS SQL Server posiada architekturę klient-serwer, co oznacza, że zajmuje się on obsługą zgłaszających się do niego klientów. Pozwala na komunikację poprzez usługi internetowe (*web service*), w takim wypadku operacje są przekazywane do serwera za pomocą protokołu TDS (*Tabular Data System*) [53]. Służy do przesyłania danych pomiędzy serwerem bazy danych a użytkownikiem. Dane zwracane są, jako dokument XML. Pakiety TDS mogą być transportowane za pomocą normalnych protokołów przekazywania danych jak TCP/IP. Aby zmniejszyć częstotliwość używania dysku twardego używane dane są buforowane w pamięci RAM. Oferuje też możliwość równoczesnego korzystania z systemu przez kilku użytkowników. W takiej sytuacji w celu zachowania spójności danych można zastosować dwie metody: optymistyczną i pesymistyczną. Optymistyczna polega na dopuszczeniu modyfikacji tych samych danych w tym samym momencie przez różnych użytkowników poprzez trzymanie starszych wersji w tabeli tymczasowej. Przy stosowaniu metody pesymistycznej system blokuje aktualnie edytowane zasoby, aby nikt inny nie mógł ich w tym czasie zmienić.[55]

System przetrzymuje dane na temat bazy w plikach *.mdf i *.ndf. Dodatkowo jest też plik *.ldf, który jest dziennikiem transakcji i jest jednym z elementów koniecznych do posiadania aktualnej kopii zapasowej [56]. Dziennik transakcji jest uzupełniany automatycznie przy każdej zmianie na bazie danych. Najważniejsza jest pełna kopia pozwalająca przywrócić bazę danych. Jej utworzenie zabiera jednak bardzo dużo zasobów i jest kosztowne dla systemu, dlatego w przypadku dużych baz nie powinna być robiona zbyt często. Następnie używana jest kopia różnicowa, której sporządzenie jest mniej czasochłonne i powinno się ją tworzyć codziennie. Ostatnim elementem jest dziennik transakcji, który pozwala przywrócić wszystkie zmiany wprowadzone od ostatniej kopii zapasowej. Pozwala to na przywrócenie bazy po awarii i na wykonywanie *rollbacku*. System posiada też dziennik zdarzeń, który pozwala łatwo odnaleźć i wyeliminować przyczyny ewentualnej awarii.

W celu jak najszybszego wykonywania się zapytań jest dostępna funkcjonalność zakładania indeksów na kolumnach. Powinno się je zakładać na najczęściej używanych kolumnach o unikalnych wartościach (jak np. Id).

MS SQL Server pozwala na korzystanie z baz przez wielu różnych użytkowników i zarządzanie ich uprawnieniami. Administrator systemu może nadawać role, które określają, jakie uprawnienia ma dany użytkownik. Najczęściej większość użytkowników ma możliwość przeglądania bazy (wykonywania instrukcji *Select*), ale już nie mogą jej edytować (*Update*) [57]. Rysunek 59 przedstawia przykładową operację nadawania uprawnień użytkownikowi.

W projekcie decydowano się wykorzystać Microsoft SQL Server, ponieważ jest on bardzo dobrym rozwiązaniem do obsługi aplikacji webowych, pracujących w systemie Windows. Dodatkowo od edycji 2005 zawiera komponent CLR (*Common Language Runtime*), który pozwala na integrację z użytym w tym projekcie Microsoft .NET.

Database Properties - Libra	гуАрр						
elect a page	<u> S</u> cript 🔸 🚺 Help						
Ƴ General Ƴ Files Ƴ Filegroups Ƴ Options	<u>S</u> erver name: View server permissions	NET1	SQLEXPRES	S			
	D.L.	1.3					
Extended Properties	Database <u>n</u> ame:	Library	мрр				
67.0	Users or roles:						
	Name				Ty	pe	
	📥 student				Us	er	
	•						•
	Effective Per Explicit germissions for stu	missions			<u></u> dd	<u>B</u> err	I ⊳ Iove
onnection	Effective Per Explicit germissions for stu Permission	missions Ident: Grantor	Grant		Add	. Berr	ove
onnection Server:	Effective Per Explicit permissions for stu Permission Alter any application r	missions udent: Grantor dbo	Grant		Add	. <u>B</u> err	ove
onnection Server: IETT\SQLEXPRESS	Effective Per Explicit permissions for stu Permission Alter any application r Alter any assembly	missions Ident: Grantor dbo dbo	Grant	বা	Add	Berr	ove
onnection Server: VET1\SQLEXPRESS Connection:	Effective Per Explicit germissions for stu Permission Alter any application r Alter any assembly Alter any asymmetric	missions Ident: Grantor dbo dbo dbo	Grant	বাব	Add	Deny	ove
onnection Server: JETT\SQLEXPRESS Connection: JETT\thicks	Effective Per Explicit germissions for stu Permission Alter any application r Alter any assembly Alter any asymmetric Alter any certificate	missions Ident: Grantor dbo dbo dbo dbo	Grant	<u>र</u> र र	Add	Deny	ove
onnection Server: NET1\SQLEXPRESS Connection: VET1\thicks	Effective Per Explicit germissions for stu Permission Alter any application r Alter any assembly Alter any asymmetric Alter any certificate Alter any contract	missions Ident: Grantor dbo dbo dbo dbo dbo	Grant	ব ব ব ব	Add	Deny	
onnection Server: VET1\SQLEXPRESS Connection: VET1\thicks	Effective Per Explicit permissions for stu Permission Alter any application r Alter any assembly Alter any assembly Alter any certificate Alter any certificate Alter any contract Alter any database D	missions Ident: Grantor dbo dbo dbo dbo dbo dbo	Grant	ব ব ব ব ব	<u>A</u> dd	Deny	
onnection Server: VET1\SQLEXPRESS Connection: VET1\thicks View connection properties	Effective Per Explicit permissions for stu Permission Alter any application r Alter any assembly Alter any asymmetric Alter any certificate Alter any certificate Alter any contract Alter any database D Alter any database e	missions Ident: Grantor dbo dbo dbo dbo dbo dbo dbo dbo	Grant	रा रा रा रा	Add	Deny	
onnection Server: NET1\SQLEXPRESS Connection: NET1\thicks View connection properties Yiew connection properties Ready	Effective Per Explicit germissions for stu Permission Alter any application r Alter any assembly Alter any certificate Alter any certificate Alter any contract Alter any database D Alter any database e Alter any database e Alter any database	missions ident: Grantor dbo dbo dbo dbo dbo dbo dbo dbo	Grant	र र र र		Deny Deny	Dye
onnection Server: NET1\SQLEXPRESS Connection: NET1\thicks Yew connection properties Yogress Ready	Effective Per Explicit germissions for stu Permission Alter any application r Alter any asymmetric Alter any certificate Alter any certificate Alter any certificate Alter any database D Alter any database e Alter any database e Alter any database e Alter any database and Alter any database e Alter any database and A	missions dent: Grantor dbo dbo dbo dbo dbo dbo dbo dbo	Grant	र र र र	Add	Deny	

Rysunek 59. Nadawanie uprawnień w MS SQL Server. Źródło: Opracowanie własne.

5. Projekt aplikacji

Na etapie projektowania po przeanalizowaniu wielu wariantów podjęto decyzję, aby wykorzystać następujące technologie:

- framework ASP .NET MVC,
- języka C#,
- *framework* Razor (warstwa wizualna).

Wynikło to z faktu, iż Microsoft posiada świetną rozwiązanie zarządzania *workflow* (Windows Workflow Foundation) i dla maksymalnej kompatybilności zastosowany został szereg innych technologii tej firmy, czyli:

- C#
- ASP .NET
- Razor
- Entity Framework

Model View Controller (MVC) został wybrany jako naturalny wzorzec projektowy z racji webowej charakterystyki systemu, oraz faktu, że był on najszerzej znany wśród członków zespołu. Dodatkowo wyraźny podział MVC na trzy elementy:

- trwałość danych
- mechanika aplikacji
- warstwa wizualna

pozwolił na wytworzenie podzespołów, które pracowały równolegle, co zapewniło maksymalną efektywność pracy.

Tworzenie projektu systemu oparto o zasady zgodnych z DDD (*Domain Driven Design*), czyli metodologii, która skupia się wokół maksymalnego odwzorowania rzeczywistości. Jest to szczególnie ważne w zespole, w którym nie każdy miał styczność z budową rozbudowanych systemów, ale posiada wiedzę merytoryczną dotyczącą problematyki zagadnień związanych z funkcjonowaniem systemu. Wynika to z faktu, iż zarówno osoba projektująca, analityk czy programista rozmawiają o procesie / modelu, który faktycznie ma miejsce w rzeczywistości.

Celem fazy projektowania było zapewnienie możliwości zakodowania wszystkich funkcjonalności opracowanych na etapie analizy lub takie przygotowanie systemu, aby ewentualny rozwój istniejących możliwości systemu był jak najłatwiejszy.

5.1. Zastosowane rozwiązania

Rozwiązanie zaproponowanie w toku fazy projektowania udało się przygotować bez potrzeby znacznych zmian w stosunku do diagramu analitycznego. Dzięki czemu przejście między fazami analizy, projektowania i implementacji odbyło się płynnie. Rozwój systemu był cały czas kontrolowany z perspektywy analitycznej i projektowej - bez straty czasu na rozpoznawanie i rozumienie różnić pomiędzy tymi etapami. Projekt był kontrolowany iteracyjnie. Po implementacji danego fragmentu funkcjonalności regularnie odbywała się kontrola zgodności z modelem analitycznym, a następnie zgodności z projektem. Również sam projekt podlegał kilkukrotnemu sprawdzeniu poprawności względem ustaleń z etapu analizy (etapu pierwotnego jak i późniejszych iteracji).



Rysunek 60. Klasa asocjacyjna PrivilegeType. Źródło: Opracowanie własne.

Pewne niezbędne zmiany miały jednak miejsce. Dla zwiększenia wydajności i wygody obsługi zarządzania uprawnieniami klasa asocjacyjna, umiejscowiona pomiędzy klasą Pracownik a klasą Uprawnienia Systemowe stała się modelem danych w bezpośredniej asocjacji z modelem *Employee*. Wynika to z faktu, że uprawnienia systemowe mogą występować w różnych konstelacjach.

Dodatkowo bezpośredni dostęp do konfiguracji jest ważniejszy niż wgląd do opisu uprawnień, utworzonych na potrzeby zarządzania nimi.

Kwestię agregacji rozwiązano poprzez zastosowanie kolekcji w modelu *Employee*, będącym częścią grupy (model *Group*). Jak widać na rysunku nr 61, model *Employee*, który jest składową wielu grup użytkowników, posiada pole *GroupList*. Pole to przetrzymuje kolekcję grup do których należy. Tym samym zachowana została zasada, że całość nie zarządza czasem istnienia części.

Jedną z ważnych funkcjonalności systemu Service Desk jest generyczność. Ma ona szczególne znaczenie w elementach takich jak budowanie formularzy (przy typie zgłoszenia) i produktów.

Produkty dzielą się na urządzenia i usługi. Są to dwie, bardzo odmienne grupy. Produkty mają parametry fizyczne takie jak: wymiary, masa, żywotność baterii itp. Usługi zaś definiowane są przez bardziej umowne elementy jak: skuteczność czy niezawodność. Aby zapewnić możliwość obsługi tego mechanizmu, utworzona została opcja definiowania parametrów, które mogą opisywać produkty i przydzielenie ich do odpowiedniej grupy (urządzenia lub usługi). Przy każdym modelu wybierane są tylko te parametry, które faktycznie go opisują. W ten sposób każdy klient korzystający z systemu będzie widział jedynie te parametry, które faktycznie definiują jego produkty. W tym celu utworzony został zestaw własności produktów.



Rysunek 61. Klasa Employee. Źródło: Opracowanie własne.

Projekt analityczny musiał zostać zmodyfikowany, gdyż zakładał zwykłe dziedziczenie specjalistycznych własności z abstrakcyjnej, ogólnej własności. W celu maksymalizacji elastyczności zastosowano asocjację z modelem opisującym własności produktu, który to zawiera kolekcję własności (poprzez kolejną asocjację)



Rysunek 62. Projekt implementacyjny Produktów i ich cech. Źródło: Opracowanie własne.

Uproszczony schemat własności został zastosowany w przypadku generyczności formularzy dla danego typu zgłoszenia. W tym przypadku, konkretny typ zgłoszenia posiada kolekcję właściwości określonego typu. Ich kolejność definiuje porządek wyświetlania pól formularza. Można było

zastosować tu uproszczony (zgodny z założeniami analizy) schemat, ponieważ typy pól formularza html są z góry zdefiniowane i znane. Jedyne zmiana, która została tu dokonana polega na możliwości definiowania podtypów pól. Dla przykładu model *PropertyDate* jest niczym innym jak zwykłym polem tekstowym, jednak można wykorzystać podtyp i wygenerować odpowiednie skrypty do budowania kalendarza tzw. *datepickera*.



Rysunek 63. Projekt generyczności parametrów Produktów. Źródło: Opracowanie własne.

5.2. Opis encji



Rysunek 64. Projekt implementacji firmy. Źródło: Opracowanie własne.

Firma jest to główna jednostka opisująca klienta. Zawiera wszystkie informacje na jego temat. Najważniejszymi elementami w kontekście *workflow* są tu: Umowa (*Contract* - potwierdza nawiązanie współpracy), Lokalizacja (*Location* - lokalizacje w których może być dokonana usługa serwisowa), osoby zgłaszające (*RequestingPerson* - osoby upoważnione do zgłaszania i prowadzenia spraw usług).



Rysunek 65. Projekt implementacji produktu. Źródło: Opracowanie własne.

Produkt to rzecz/usługa, która może podlegać serwisowaniu lub naprawie. Opisany jest przez zbiór elementów, takich jak: skuteczność, wymiary, waga, funkcjonalności.



Rysunek 66. Projekt implementacji pracownika. Źródło: Opracowanie własne.

Uprawnienia dotyczą Pracowników systemu. Ich celem jest udostępnianie odpowiednim osobom tylko tych elementów, do których faktycznie potrzebują dostępu. Służy to ograniczeniu błędnych procedur wewnątrz firmy.



Rysunek 67. Projekt implementacji zgłoszenia – część 1 z 2. Źródło: Opracowanie własne.



Rysunek 68. Projekt implementacji zgłoszenia – część 2 z 2. Źródło: Opracowanie własne.

Zgłoszenie to jedna z najważniejszych jednostek całego systemu. Przyjmowane jest przez serwis. Zgłoszenie powiązane jest z umową, lokalizacją i produktem. Definiuje to: czy można, gdzie, co i dla kogo. Dodatkowo każde zgłoszenie jest powiązane bezpośrednio z typem zgłoszenia, który definiuje przebieg procesu obsługi.

6. Implementacja

6.1. Wstęp

Od początku zespół implementacji był w ścisłej współpracy z zespołami projektowania i testów. Zapewniło to płynną pracę oraz wykluczenie ewentualnych zatorów. Sam zespół został podzielony na mniejsze podzespoły w celu maksymalnego zrównoleglenia prac. Poszczególne podzespoły odpowiedzialne były za:

- kodowanie grafiki i szablonów,
- utworzenie mechanizmów do zarządzania rekordami (CRUD),
- implementację mechaniki workflowów oraz spięcie technologii z systemem Service Desk,
- utworzenie aplikacji administratorskiej do tworzenia *workflowów* wykorzystywanych przez system.

W zależności od etapu prac nacisk był kładziony na różne części. W początkowej potrzebowaliśmy wytworzyć maksymalną ilość CRUDów, aby można było zarządzać rekordami w bazie danych, niezbędnymi do testów. Zastosowano plan minimum z ograniczonymi funkcjonalnościami. Równocześnie tworzone były statyczne pliki html wraz ze wszystkimi skryptami css i js.

W drugim etapie prac zamknęliśmy prace dotyczące grafiki, zaś zasoby zostały przeniesione na wytwarzanie elementów powiązanych z *workflowami*. Dodatkowo zespół testów sprawdzał co już zostało wykonane i na bieżąco przygotowywał listę punktów zapalnych, które mogłyby mieć wpływ na ewentualne opóźnienie.

Kiedy prace nad *workflowami* dobiegały końca, część zasobów została przeniesiona na wykończenie kwestii zarządzania systemem. Było to możliwe, ponieważ prace nad każdym elementem były maksymalnie zrównoleglone. W przypadku, gdy w jednej części systemu zasoby ludzkie były w nadmiarze, względem pracy do wykonania, były przenoszone do części, gdzie nadal istniały braki zarówno funkcjonalne jak i zasobowe.

O jakość zadbano poprzez ciągłe testy i kontrolę zgodności z dokumentacją. W tym celu został wydzielony podzespół z ekipy testerów, który cały czas pilnował rozwoju systemu. Uniknęliśmy w ten sposób wielu dodatkowych błędów i nieplanowanych odstępstw od dokumentacji.

6.2. Code-first

W projekcie wykorzystano technologię *Code-First*, która jest strategią generowania struktury bazy danych za pomocą Microsoft .NET opartego na *frameworku* MVC 4. W myśl tej strategii najpierw implementowane są klasy biznesowe modelu danych, na podstawie którego generowana jest baza danych. Główne powody wyboru strategii *Code-First* to:

- pełna kontrola nad kodem aplikacji,
- brak konieczność projektowania bazy danych Entity Framework założy niezbędne tabele,
- brak konieczności ręcznej ingerencji w strukturę bazy danych *Entity Framework* wykona automatycznie aktualizacje.

6.3. Modele danych

Klasy biznesowe znajdują się w katalogu *Models* w panelu *Solution Explorer*. Na podstawie tych klas zbudowana została struktura bazy danych.

Solution Explorer 👻 🤻	×
Search Solution Explorer (Ctrl+;)) -
🔺 📹 Models	٠
C* AccountModels.cs	
C* Contract.cs	
C* Employee.cs	
C* Feature.cs	
C# Firm.cs	
C* Group.cs	
C* Location.cs	
C* Operate.cs	
C# Person.cs	
C* Position.cs	
C* Privileges.cs	
C* PrivilegeType.cs	
C# Problem.cs	
C* Producer.cs	
C* Product.cs	
C* ProductDevice.cs	
C# ProductDeviceModel.cs	
C# ProductFeature.cs	_
Solution Explorer Team Explorer	Ŧ

Rysunek 69. Okienko Solution Explorer, widok folderu Models. Źródło: Opracowanie własne.

6.4. Budowa klasy biznesowej

W ciałach klas znajdują się deklaracje pól oraz konstruktora (listing nr 2).

Listing 2. Deklaracja pól i konstruktora. Źródło: Opracowanie własne.

```
public string City { get; set; }
public Firm() { }
```

Dla czytelności kodu atrybuty klasy oraz konstruktory pogrupowane zostały w bloki zwane regionami. Rozpoczęcie bloku oznaczane jest przez słowo kluczowe '#region NazwaRegionu' oraz zamknięcie bloku '#endregion'.

Środowisko Microsoft Visual Studio wspiera programistę w przygotowywaniu dokumentacji do kodu. Po trzykrotnym wprowadzeniu znaku *slash* przed atrybutem lub metodą edytor generuje szablon komentarza.

Atrybuty klasy przyjmują różne typy danych w zależności od tego jakie informacje ma przechowywać atrybut i dzielą się na atrybuty wartościowe oraz referencyjne. Definicja typu atrybutu ma bezpośredni wpływ na sposób reprezentacji danych w bazie SQL, wygenerowanej za pomocą *Entity Framework*. Atrybuty wartościowe zostaną bezpośrednio zmapowane na odpowiadające im typy, dostępne w relacyjnej bazie danych.

W celu sprecyzowania atrybutu klasy, jego działania, czy też sposobu w jaki ma zostać odzwierciedlony w bazie danych, deklaracje atrybutów klasy poprzedza się *DataAnnotations*

umieszczonymi w nawiasach kwadratowych bezpośrednio przed atrybutem. Mają one wpływ na sposób projektowania bazy danych przez *Entity Framework* oraz na automatyczne generowanie formatek za pomocą narzędzia Razor.

[Column("ID_Firm")] – adnotacja ta ustawia nazwę dla kolumny w bazie danych dla następującego po nim atrybutu klasy. Wykorzystujemy ją gdy chcemy sami nadać nazwę kolumnie w tabeli np. z powodu ograniczeń nałożonych na schemat bazy danych.

[Key] – informuje, że następująca po nim deklaracja atrybutu klasy jest kluczem głównym (listing nr 3).

Listing 3. Deklaracja klasy Id. Źródło: Opracowanie własne.

```
public int Id { get; set; }
```

Aby każda reprezentacja klasy biznesowej w modelu relacyjnym miała klucz główny, najkorzystniej jest utworzyć klasę z zadeklarowanym polem klucza głównego (patrz listing nr 4).

Listing 4. Deklaracja klasy *SqlObject* z zadeklarowanych polem klucza głównego. Źródło: Opracowanie własne.

```
public class SqlObject
{
    #region Properties
    /// <summary>
    /// PK
    /// </summary>
    [Key]
    [Column("ID")]
    public int Id { get; set; }
    /// <summary>
    /// Data dodania
    /// </summary>
    [DataType(DataType.Date)]
    public DateTime InsertDate { get; set; }
    /// <summary>
    /// Data aktualizacji
    /// </summary>
    [DataType(DataType.Date)]
    public DateTime UpdateDate { get; set; }
    #endregion
```

}

Następnie wykorzystując dziedziczenie (listing nr 5) każda klasa biznesowa otrzymuje atrybut klucza głównego oraz dodatkowe przydatne pola: datę dodania oraz datę aktualizacji.

Listing 5. Deklaracja klasy *Firm* z dziedziczeniem z klasy *SqlObject*. Źródło: Opracowanie własne.

```
public class Firm : SqlObject
{ ... }
```

[ForeignKey("FirmId")] – informuje o tym, że następująca po nim deklaracja atrybutu typu referencyjnego ma być reprezentowana w bazie danych kluczem obcym w relacji jeden do jeden. Jako parametr podaje się nazwę atrybutu klasowego typu int, który będzie fizycznie kolumną klucza obcego. Do definicji pola złożonego dodajemy słowo kluczowe *virtual* (listing nr 6).

Listing 6. Deklaracja klasy Firm ze słowem kluczowym virtual. Źródło: Opracowanie własne.

public virtual Firm Firm { get; set; }

[Display(Name = "Adres lokalizacji")] – wykorzystywany jest do sterowania generowaniem widoku kontrolki przez narzędzie Razor dla atrybutu klasy znajdującego się poniżej. Parametr '*Name*' wykorzystuje się do przekazania narzędziu nazwy dla etykiety, która zostanie umieszczona obok kontrolki reprezentującej atrybut klasy.

[MaxLength(6)] – definiuje maksymalną ilość znaków znajdującego się pod nim atrybutu klasy. Narzędzie Razor generuje kod, który waliduje długość ciągu znaków wpisywanego do kontrolki reprezentującej atrybut klasy. Narzędzie generujące bazę danych wykorzystuje znacznik do zdefiniowania długości atrybutu w bazie danych. Bez tego znacznika narzędzie ustawi maksymalnie dostępną długość ciągu znaków lub liczby dla danego typu danych. Opcjonalnie po przecinku można dodać drugi parametr *ErrorMessage*, definiujący komunikat błędu (listing nr 7).

Listing 7. Definicja komunikatu błędu. Źródło: Opracowanie własne.

[MaxLength(6, ErrorMessage="Nie możesz wpisac więcej niż 6 znaków")]

[DataType(DataType.EmailAddress)] – jest informacją dla narzędzia, generującego kontroler klasy, że atrybut nad którym się znajduje jest typu adres email. Narzędzie Razor wygeneruje dodatkowy kod, który sprawdzi czy wprowadzony ciąg znaków do kontrolki jest adresem email.

[DataType(DataType.Date)] – informuje narzędzie generujące kontroler klasy, że znajdujący się pod nim atrybut jest typu daty. Narzędzie Razor wygeneruje dodatkowy kod, który umieści obok kontrolki atrybutu klasy ikonę kalendarza, po naciśnięciu której w wygodny sposób użytkownik aplikacji będzie mógł wybrać datę.

[DataType(DataType.Password)] – informuje narzędzie generujące kontroler klasy, że znajdujący się pod nim atrybut jest typu hasło. Wygenerowana kontrolka będzie maskować wpisywany do niej ciąg znaków.

[Required] – informacja oznacza, że znajdujący się pod nim atrybut klasy jest wymagany. Narzędzie generujące bazę danych wykorzysta tę informację, definiując kolumnę tabeli bazy danych, ustawiając jej właściwość '*Allow Nulls*' na '*false*'. Narzędzie *Razor*, natrafiając na ten znacznik przy generowaniu kontrolki, utworzy dodatkowy kod, sprawdzający czy to pole formularza zostało uzupełnione przed wysłaniem. Opcjonalnie można dodać parametr z komunikatem błędu (listing nr 8).

Listing 8. Definicja wymagalności atrybutu z komunikatem błędu. Źródło: Opracowanie własne.

[Required(ErrorMessage="Pol	e jest	wymagane")]	
-----------------------------	--------	-------------	--

Komunikaty błędów mogą również być przechowywane w osobnym pliku zasobów. W tym przypadku *Data Annotation* wygląda jak na listingu nr 9.

Listing 9. Definicja wymagalności atrybutu z komunikatem błędu w osobnym pliku. Źródło: Opracowanie własne.

[Required(ErrorMessageResourceType = typeof(ErrorResources), ErrorMessageResourceName = "Common_NameRequired")]

Każda klasa biznesowa powinna posiadać ekstensję w klasie dziedziczącej po *DbContext* i tak też zostało to zaimplementowane w projekcie (listing nr 10).

Listing 10. Implementacja ekstensji. Źródło: Opracowanie własne.

```
public class ServiceEntities : DbContext
{
    #region DbSet
    public DbSet<Contract> Contracts { get; set; }
    public DbSet<Employee> Employees { get; set; }
    (...)
}
```

6.5. Relacje

W tej części dokumentu opisana została implementacja relacji jeden do wielu i wiele do wielu.

6.5.1. Relacja jeden do wielu

Relacja jeden do wielu realizowana jest poprzez umieszczenie atrybutów relacji zarówno po stronie jeden jak i wiele. Implementacja relacji po stronie jeden realizowana jest poprzez użycie dwóch atrybutów (listing nr 11).

> Listing 11. Implementacja relacji jeden do wielu po stronie jeden. Źródło: Opracowanie własne.

```
[Column("ID_Location")]
public int LocationId { get; set; }
[ForeignKey("LocationId")]
public virtual Location Location { get; set; }
```

Implementacja relacji po stronie wiele (listing nr 12) realizowana jest za pomocą atrybutu kolekcji, który dziedziczy po klasie ICollection. Implementację atrybutu kolekcji realizuje się poprzez umieszczenie w ciele klasy dwóch atrybutów kolekcji. Pierwszy z nich jest atrybutem prywatnym - w nim przechowywana jest kolekcja. Drugi atrybut jest publiczny oraz opatrzony słowem kluczowym *virtual*. Posiada on dwie metody: *get* oraz *set*. Metoda *get* zwraca kolekcję (atrybut prywatny klasy), a

jeżeli kolekcja nie została wcześniej utworzona, wywołuje jej konstruktor. Taka konstrukcja zabezpiecza aplikacje przed pojawieniem się błędu czasu wykonania, gdyby aplikacja próbowała się odwołać do atrybutu, który nie został jeszcze utworzony i posiadał wartość *null*.

Listing 12. Implementacja relacji jeden do wielu po stronie wiele. Źródło: Opracowanie własne.

```
private ICollection<Location> _LocationList;
public virtual ICollection<Location> LocationList {
    get { return _LocationList ?? (_LocationList = new
        Collection<Location>()); }
        set { _LocationList = value; }
}
```

6.5.2. Relacja wiele do wielu

Relacja wiele do wielu jest realizowana za pomocą kolekcji w obu klasach stron asocjacji.

Entity framework nie generuje bezpośrednio tabeli pośredniczącej dla asocjacji. Należy jawnie w klasie dziedziczącej z DbContext w metodzie *OnModelCreating* zaimplementować kod, generujący klasę pośredniczącą asocjacji (listing nr 13).

Listing 13. Implementacja relacji wiele do wielu poprzez klasę asocjacyjną. Źródło: Opracowanie własne.

6.6. Dziedziczenie

Entity framework realizuje dziedziczenie w bazie danych poprzez utworzenie jednej tabeli dla klasy bazowej z jej atrybutami oraz umieszczając w kolejnych kolumnach wszystkie atrybuty klas dziedziczących. W ostatniej kolumnie o nazwie *Discriminator* umieszczona jest nazwa klasy dziedziczącej, której dotyczy rekord.

	ID	FirstName	LastName	Phone	Password	Discriminator
1	1	Jan	Dawes	211388994	NULL	RequestingPerson
2	2	Mateusz	Curtis	257129089	NULL	RequestingPerson

Rysunek 70. Widok ekranu programu Microsoft Management Studio ilustruje układ tabeli realizującej dziedziczenie. Źródło: Opracowanie własne.

6.7. Aktualizacja bazy danych – migracje

W trakcie implementacji może się zdarzyć konieczność modyfikacji klas biznesowych. Zmiana ta wpływa na konieczność przeprojektowania struktury bazy danych. W tym celu wykorzystuje się mechanizm migracji bazy danych, który zaktualizuje strukturę tabel o naniesione zmiany w klasach biznesowych. Aby skorzystać z narzędzia, uruchamiamy okno konsoli wybierając Menu \rightarrow TOOLS \rightarrow Library Package Manager \rightarrow Package Manager Console.

M	TOC	DLS	TEST	WINDOW	HELP									
Ċ.	o [©]	Atta	ich to Pro	ocess		Ctrl+Alt+P		, 8	⊨ (¶	to 🛠		위케	2個	Ŧ
Privi	*#	Con	inect to [Database				Proc	luct.cs		SqlObje	ect.cs		Firm.c
🔩 P	£	Cod	le Snippe	ets Manager		Ctrl+K, Ctrl+B)	1	Name					
		Cho	ose Tool	lbox Items										
		Libr	ary Pack	age Manager			•	C:\	Packag	je Mani	ager Co	onsole		
	¢	Exte	nsions a	nd Updates				ø	Packag	ge Mana	ager Se	ttings		
		Crez	ate GLIID					-			_		_	

Rysunek 71. Umiejscowienie 'Package Manager Console'. Źródło: Opracowanie własne.

Za pomocą narzędzia *Entity Framework Migrations* można w prosty sposób aktualizować bazę danych. Korzystanie z migracji należy poprzedzić wprowadzeniem do konsoli komendy '*Enable-Migrations*'.



Rysunek 72. Przygotowanie migracji. Źródło: Opracowanie własne.

W wyniku działania komendy utworzony zostanie folder w oknie *Solution Explorer* o nazwie *migrations*. W miarę dodawania nowych migracji folder ten będzie zapełniał się kolejnymi plikami migracji. W folderze tym znajduje się również plik Configuration.cs. Jest to klasa dziedzicząca z *DbMigrationsConfiguration*. Klasa ta posiada atrybut *AutomaticMigrationsEnabled*, za pomocą którego możemy określić czy migracje mają być przeprowadzone automatycznie. W klasie tej znajduję się również metoda 'Seed'. W ciele tej metody można umieścić deklaracje obiektów które

zostaną umieszczone w bazie danych po zakończeniu migracji. W ten sposób można w łatwy sposób zasilić bazę danych początkowymi danymi niezbędnymi do uruchomienia aplikacji i jej późniejszego testowania. Migrację przygotowuje się wpisując w okno konsoli polecenie '*add-migration*' (rysunek nr 72). Następnie należy podać nazwę dla pliku migracji.

Narzędzie umieszcza pliki migracji w folderze '*Migrations*'. Tak utworzony plik jest źródłem informacji dla *Entity Framework* w jaki sposób należy zmodyfikować bazę danych, w związku ze zmianami wprowadzonymi do implementacji klas biznesowych.

Package Manager Console 👻 👎	x
Package source: NuGet official package source 👻 🏟	**
PM> update-database Specify the '-Verbose' flag to view the SQL statements being applied to the target database. System.Data.ProviderIncompatibleException: An error occurred while getting provider information from the database. This can be caused by Entity Framework using an incorrect connection string. Check the inner exceptions for details and ensure that the connection string is correct> System.Data.ProviderIncompatibleException: The provider did not return a ProviderManifestToken string>	•
Package Manager Console Error List	



Aby uaktualnić bazę danych należy w oknie konsoli wpisać polecenie '*update-database*'. *Entity Framework* wykorzysta ostatnio utworzony plik migracji do aktualizacji bazy danych.

6.8. Testowanie funkcjonalności

Testy funkcjonalne zostały przeprowadzone z zachowaniem poniższych zasad i sposobu przeprowadzenia testów. Realizacja testów polegała na wykonaniu kolejnych czynności wymienionych w scenariuszach testowych. Scenariusze te składają się z kroków opisujących wykonanie pojedynczej czynności oraz wykazu kryteriów poprawności ich wykonania. Spełnienie kryteriów poprawności jest równoznaczne z pozytywną realizacją Scenariusza Testowego.

W przypadku stwierdzenia występowania potencjalnej usterki, osoba wykonująca testy rejestrowała zgłoszenie w aplikacji Mantis opisując problem, który został zauważony oraz wprowadziła informacje do niniejszego dokumentu. Tabela nr 20 przedstawia przyjęte statusy nadawane podczas realizacji poszczególnych scenariuszy.

Status	Opis Błędu
Błąd krytyczny (N)	Błąd uniemożliwiający realizację przypadku testowego lub powodujący realizację przypadku testowego niezgodnie ze zdefiniowanymi kryteriami jego poprawności.
Wada Inna (PW)	Błąd, który nie blokuje realizacji przypadku testowego oraz nie podlega pod ustalone kryteria poprawności dla wykonywanego przypadku testowego.
Uwaga (PU)	Wynik pozytywny po uwzględnieniu poprawek.

Tabela 20. Statusy błędów. Źródło: Opracowanie własne.

Wynik pozytywny (P)	Bez uwag
---------------------	----------

6.8.1. Faza 0 – CRUD

Pierwsza faza testów, oznaczona numerem zero, obejmowała testowanie obsługi wszystkich obiektów. Jej nazwa CRUD wzięła się od pierwszych liter słów *Create*, *Read*, *Update*, *Delete* (z ang. tworzyć, czytać, aktualizować, kasować).

6.8.1.1. Obsługa obiektu umowa

Tabela 21. Scenariusz testu: Obsługa obiektu umowa. Źródło: Opracowanie własne.

Nazwa scer	nariusza	Tworzenie, edycja, usuwanie obiektu kontrakt (umowa)				
Warunki po	czątkowe	Uruchomiony system Service Desk, Użytkownik z uprawnieniami Administratora Biznesowego				
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z lewego me umowę"	enu z sekcji umowa wybrać opcję "dodaj				
2	Wcisnąć prz a) Pone	ycisk "Anuluj" owić krok 1				
3	Wypełnić do i. Po v	ostępne pola vypełnieniu, zmienić ich zawartość				
4	Zakończyć k	ilikając przycisk "dodaj"	 Pole data końca umowy, według diagramu klas nie jest obowiązkowa. Nie można wybrać dat z przyszłości dla zawarcia umowy oraz daty końca umowy, pojawia się komunikat: "<i>The value '19-03- 2013' is not valid for</i> <i>Data zawarcia umowy</i>". Brakuje osób zgłaszających 			
5	Z lewego me umowę"	enu z sekcji umowa wybrać opcję "Wyszukaj				
6	Wyszukiwać wpisując zar	kolejno po wszystkich dostępnych polach, ówno dane istniejące jak i fałszywe				
7	Na dowolnie kolejno ze sl a) Edyt	e wybranej zarejestrowanej umowie wybierać krajnej prawej kolumny: uj	Raz wstawionej/wybranej daty nie da się			

	 i. Wcisnąć przycisk "Anuluj" ii. Ponowić krok a iii. Zmieniać po kolei wszystkie dostępne pola iv. Zakończyć wciskając "Edytuj b) Podgląd/szczegóły Zakończyć wciskając "powrót do listy" 	skasować
	i. Wcisnąć przycisk "anuluj umowę"	
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie umowy Wyszukanie umowy. Edycja umowy. Usunięcie umowy 	
Uzyskany rezultat	Zgodny z oczekiwanym	
Wynik testu	PU. Zgłoszone błędy. Numer zgłoszenia w Nrozwiązane w dniu 24/05/2013.	1antis – 53. Zgłoszenie

6.8.1.2. Obsługa obiektu grupa

Tabela 22. Scenariusz testu: Obsługa obiektu grupa. Źródło: Opracowanie własne.

Nazwa scena	ariusza	usza Tworzenie, edycja, usuwanie obiektu grupa				
Warunki poc	ownik z uprawnieniami					
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z lewego	menu z sekcji Grupy wybrać opcję "Dodaj"	Nie działa "anuluj"			
2	Wcisnąć p a) Po	przycisk "Anuluj" pnowić krok 1				
3	Wpisać na	azwę grupy				
4	Wcisnąć p	przycisk "Dodaj"				
5	Wykonać dla grupy	ponownie kroki 1-4 podając takie same dane	Powinniśmy zastanowić się czy nazwa grupy nie powinna być unikalna			
6	Z lewego grupę"	o menu wybrać z sekcji umowa "Wyszukaj	Brakuje wyszukiwarki grup			
7	Wyszukiw wpisując z	vać kolejno po wszystkich dostępnych polach, zarówno dane istniejące jak i fałszywe				
8	Na dowol kolejno ze a) E	Inie wybranej zarejestrowanej grupie wybierać e skrajnej prawej kolumny: dytuj i. Wcisnąć przycisk "Anuluj" ii. Ponowić krok a	W <i>details</i> i <i>delete</i> nagłówek nie mieści się w oknie			

	 iii. Zmieniać po kolei wszystkie dostępne pola iv. Zakończyć wciskając "Edytuj b) Podgląd/szczegóły i. Zakończyć wciskając "powrót do listy" c) Usuń i. Wcisnąć przycisk "anuluj umowę" 		
9	Wcisnąć przycisk "usuń"		
10	Dodatkowe uwagi	Brakuje metod dodaj osobę i usuń osobę	
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie grupy Wyszukanie grupy. Edycja grupy. Usunięcie grupy 		
Uzyskany rezultat	Zgodny z oczekiwanym		
Wynik testu	/ status PU. Zgłoszone błędy. Numer zgłoszenia w l rozwiązane w dniu 24/05/2013.	Mantis – 38. Zgłoszenie	

6.8.1.3. Obsługa obiektu producent

					,		
T 1 1 00	a .	. 01 1	1 . 1 .	1 /	77 / 11	<u> </u>	1
Tobolo 17	Noonomillog to	atur ()hahuaa	o biolztu ·	nroduoont	Irodio	(mrogoutionia	TTROCHO
	ACEDALINSZ IE		1 ())))(=)K []]]			() macowanne	W/1/2 STIP
1 u u u u u u u u u u u u u u u u u u u			ioononiu	producent.	LIUGIU.	Opracowanie	wiusiic.
		0		1		1	

Nazwa scenariusza		Tworzenie, edycja, usuwanie obiektu produce	ent	
Warunki początkowe		Uruchomiony system Service Desk, Uż Administratora Biznesowego	ytkownik z uprawnieniami	
Realizacja scenariusza testowego				
Krok		Opis	Uwagi	
1	Z lewego	menu z sekcji Producent wybrać opcję "Dodaj	"	
2	Wcisnąć j a) P			
3	Podać nazwę producenta			
4	Zakończyć wciskają "dodaj"			
5	Ponowić krok 1			
6	Spróbować ponownie dodać tego samego producenta			
7	Z lewego	menu wybrać z sekcji Producent "Dodaj"		
8	Wyszukiv wpisując	vać kolejno po wszystkich dostępnych polac zarówno dane istniejące jak i fałszywe	h,	
9	Na dowo wybierać a) E	olnie wybranym zarejestrowanym producenc kolejno ze skrajnej prawej kolumny: dytuj	ie W <i>edit</i> i <i>delete</i> nagłówek nie mieści się w oknie	

	 i. Wcisnąć przycisk "Anuluj" ii. Ponowić krok a iii. Zmieniać po kolei wszystkie dostępne pola iv. Zakończyć wciskając "Edytuj b) Podgląd/szczegóły i. Zakończyć wciskając "powrót do listy" c) Usuń 		
10	Wcisnąć przycisk "usuń"		
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie producenta Wyszukanie producenta. Edycja producenta. Usunięcie producenta 		
Uzyskany rezultat	Zgodny z oczekiwanym		
Wynik testu	/ status PU. Zgłoszone błędy. Numer zgłoszenia w Mantis – 48 i 70. Zgłoszenia rozwiązane w dniu 9/06/2013.		

6.8.1.4. Obsługa obiektu osoba

Tabela 24. Scenariusz testu: Obsługa obiektu osoba. Źródło: Opracowanie własne.

Nazwa scenariusza		Tworzenie, edycja, usuwanie obiektu osoba			
Warunki początkowe		Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami		
	Realizacja scenariusza testowego				
Krok		Opis	Uwagi		
1	Z lewego Osobę"	o menu z sekcji Osoby wybrać opcję "Dodaj	W wyszukiwaniu Osoby, jest błąd przy wciśnięciu <i>edit</i>		
2	Wcisnąć j a) P	przycisk "Anuluj" onowić krok 1			
3	Wypełnić a) P	c dostępne pola o wypełnieniu zmienić ich zawartość	E-mail powinien być unikalny		
4	Zakończy	ć wciskają "dodaj"			
5	Ponowić l	krok 1			
6	Ponownie	dodać tą samą osobę			
7	Z lewego	menu wybrać z sekcji Osoby "Dodaj"			
8	Wyszukiv wpisując z	vać kolejno po wszystkich dostępnych polach, zarówno dane istniejące jak i fałszywe	Nie ma żadnych pól, po których można wyszukiwać		
9	Na dowo	lnie wybranej zarejestrowanej osobie wybierać	<i>Details</i> i <i>delete</i> mają jakieś stare wersje		
	kolejno ze skrajnej prawej kolumny:	formatek			
--	---	----------	--		
	 a) Edytuj Wcisnąć przycisk "Anuluj" Ponowić krok a Zmieniać po kolei wszystkie dostępne pola Zakończyć wciskając "Edytuj b) Podgląd/szczegóły Zakończyć wciskając "powrót do listy" 				
10	Wcisnąć przycisk "usuń"				
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie osoby Wyszukanie osoby Edycja osoby Usunięcie osoby 				
Uzyskany rezultat	Zgodny z oczekiwanym				
Wynik testu	Wynik testu / status PU. Zgłoszone błędy. Numer zgłoszenia w Mantis – 29 i 71				

6.8.1.5. Obsługa obiektu firma

Tabela 25. Scenariusz testu: Obsługa obiektu firma. Źródło: Opracowanie własne.

Nazwa scenariusza		Tworzenie, edycja, usuwanie obiektu firma	
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Z lewego firmę"	o menu z sekcji Firma wybrać opcję "Dodaj	W lewym panelu są dwie opcje dodaj firmę
			Z obu opcji nie da się zapisać dodania nowej firmy
			a. Błąd przy dodawaniu firmy z opcji pierwszej (z szarym tłem) patrz załącznik firma1.jpg
			b. Błąd przy dodawaniu firmy z opcji drugiej(z białym tłem) patrz załącznik firma2.jpg
2	Wcisnąć	przycisk "Anuluj"	
	a) P	onowić krok 1	

3	Wypełnić dostępne pola	
	a) Po wypełnieniu spróbować zmienić ich zawartość	
4	Zakończyć wciskają "dodaj"	
5	Ponowić krok 1	
6	Ponownie dodać tę samą firmę	
7	Z lewego menu wybrać z sekcji umowa "Wyszukaj firmę"	
8	Wyszukiwać kolejno po wszystkich dostępnych polach, wpisując zarówno dane istniejące jak i fałszywe	
9	 Na dowolnie wybranej zarejestrowanej firmie wybierać kolejno ze skrajnej prawej kolumny: a. Edytuj Wcisnąć przycisk "Anuluj" Ponowić krok a Zmieniać po kolei wszystkie dostępne pola Zakończyć wciskając "Edytuj" b. Podgląd/szczegóły Zakończyć wciskając "powrót do listy" Usuń 	
10	Wcisnąć przycisk "usuń"	
11	Dodatkowe uwagi	Brak metody DodajKlienta, która to metoda przypisywać będzie osoby uprawnione do zgłaszania incydentów/zgłoszeń
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie firmy Wyszukanie firmy Edycja firmy Usunięcie firmy 	
Uzyskany rezultat	Zgodny z oczekiwanym	
Wynik testu	/ status PU. Zgłoszone błędy. Numer zgłoszenia w Mant	tis – 31 i 72

6.8.1.6. Obsługa obiektu lokalizacja

Tabela 26. Scenariusz testu: Obsługa obiektu lokalizacja. Źródło: Opracowanie własne.

Nazwa scenariusza	Tworzenie, edg	ycja, usuv	wanie obie	ektu Lol	kalizacja		
Warunki początkowe	Uruchomiony	system	Service	Desk,	Użytkownik	Z	uprawnieniami

Administratora Biznesowego					
Realizacja scenariusza testowego					
Krok	Opis	Uwagi			
1	Z lewego menu z sekcji Lokalizacja wybrać opcję "Dodaj lokalizację"				
2	Weisnąć przycisk "Anuluj"				
2					
3	wypemic dostępne pola				
	a) Po wypełnieniu sprobować zmienić ich zawartość				
4	Zakończyć wciskają "dodaj"				
5	Ponowić krok 1				
6	Spróbować ponownie dodać tę samą lokalizację				
7	Z lewego menu wybrać z sekcji umowa "Wyszukaj lokalizację"				
8	Wyszukiwać kolejno po wszystkich dostępnych polach, wpisując zarówno dane istniejące jak i fałszywe	Brak kryteriów wyszukiwania			
9	 Na dowolnie wybranej zarejestrowanej lokalizacji wybierać kolejno ze skrajnej prawej kolumny: a) Edytuj Wcisnąć przycisk "Anuluj" Ponowić krok a Zmieniać po kolei wszystkie dostępne pola Zakończyć wciskając "Edytuj b) Podgląd/szczegóły Zakończyć wciskając "powrót do listy" c) Usuń 	Dodać ikony <i>details,</i> <i>edit, delete</i> tak jak np. dla firmy W <i>details</i> pole <i>update</i> powinno być z godziną i tylko do odczytu. Czyli, że po edycji lokalizacji ta data i godzina powinna ustawić się automatycznie na datę edycji (systemową) Zmienić nazwę z <i>UpdateDate</i> na "Data edycji" <i>InstertDate</i> powinno być tylko do odczytu Zmienić nazwę z <i>InstertDate</i> na "data dodania"			
10	Weisnąć przycisk "usuń"	Jeśli są wystawione zgłoszenia z użyciem danej lokalizacji to nie powinno dać się jej			

			usunąć. Po naciśnięciu		
			delete, lokalizacja		
			powinna się		
			dezaktywować.		
Oczekiwany	1. D	odanie lokalizacji			
rezultat	2. W	2. Wyszukanie lokalizacji			
(Kryteria akceptacji)	3. E 4. U	 Edycja lokalizacji Usunięcie lokalizacji 			
Uzyskany rezultat	Zgodny z oczekiwanym				
Wynik testu	/ status	PU. Zgłoszone błędy. Numer zgłoszenia w Ma numer 28 rozwiązane w dniu 12/05/2013. rozwiązane w dniu 10/06/2013	ntis – 28 i 73. Zgłosznie Zgłoszenie numer 73		

6.8.1.7. Obsługa obiektu pracownicy

					,		
T 1 1 07 0	• • • •	$\cap 1 1$	1 . 1 .	•	77 / 11	\sim ·	1
Tabala 11 Neg	anoriller toetile	(hehiro)	Obiol/fil	nracountou	/rodio:	Inrocomonia	WAGNA
-1 and $a \ge 1$. Not	JHAHUMA IDMU. Y	VDSiuga	UDDUKLU	DIACOWINCY		VDD aCOWAIIIC	wiashe
1 40014 271 500	Junius Costan	Cobiaga	00101104	prace mine ,	. Drogro.	opiacomanie	Tricolle.
		<u> </u>		1 2		1	

Nazwa scena	ariusza	Tworzenie, edycja, usuwanie obiektu Pracov	wnicy	7	
Warunki poc	zątkowe	Uruchomiony system Service Desk, U Administratora Biznesowego	Jżytko	wnik z uprawnieniami	
		Realizacja scenariusza testowego			
Krok		Opis		Uwagi	
1	Z leweg	o menu z sekcji pracownicy wybrać op icy" następnie "dodaj"	pcję		
2	Wcisnąć a) P	przycisk "Anuluj" onowić krok 1			
3	Wypełnić a) P	Wypełnić dostępne pola a) Po wypełnieniu spróbować zmienić ich zawartość			
4	Zakończy	Zakończyć wciskają "dodaj"			
5	Ponowić	Ponowić krok 1			
6	Ponownie	e dodać tego samego pracownika			
7	Z lewe	go menu wybrać z sekcji Pracowr nicy"	nicy	Brak przycisku wyszukaj pracownika	
				Nagłówek przy wyszukiwaniu pracowników zaciągany z umowy	

8	Wyszukiv wpisując	vać kolejno po wszystkich dostępnych polach, zarówno dane istniejące jak i fałszywe		
9	Na dowo wybierać a) E b) P c) U	 olnie wybranym zarejestrowanym pracowniku kolejno ze skrajnej prawej kolumny: dytuj Wcisnąć przycisk "Anuluj" Ponowić krok a Zmieniać po kolei wszystkie dostępne pola Zakończyć wciskając "Edytuj odgląd/szczegóły Zakończyć wciskając "powrót do listy" 	Widok <i>d</i> zupełnie poprawiając się na "Firm	<i>letails</i> jest nieczytelny, c wzorować na"
10	Wcisnąć przycisk "usuń"			
Oczekiwany rezultat (Kryteria akceptacji)	1. Dodanie pracownika2. Wyszukanie pracownika3. Edycja pracownika4. Usunięcie pracownika			
Uzyskany rezultat	zultat Zgodny z oczekiwanym			
Wynik testu	/ status	PU. Zgłoszone błędy. Numer zgłoszenia w l rozwiązane w dniu 24/05/2013.	Mantis – 37	. Zgłoszenie

6.8.1.8. Obsługa obiektu Produkt - Urządzenie

Tabela 28. Scenariusz testu: Obsługa obiektu Produkt - Urządzenie. Źródło: Opracowanie własne.

Nazwa scenariusza		Tworzenie, edycja, usuwanie obiektu Produkt - Urządzenie			
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami		
		Realizacja scenariusza testowego			
Krok		Opis	Uwagi		
1	Z lewego - urządzei	menu z sekcji produkty wybrać opcję "produkt nie" następnie "dodaj"	Brak pól do wyszukiwania		
2	Wcisnąć j a) P	przycisk "Anuluj" onowić krok 1	Przy próbie wciśnięcia anuluj występuje błąd aplikacji		
3	Wypełnić a) P	dostępne pola o wypełnieniu spróbować zmienić ich zawartość	Przy dodawaniu produkt urządzenie przetłumaczyć opisy pól na jęz. Polski		

		Przy dodawaniu produkt urządzenie, nie ma możliwości wyboru producenta, wstawia się jakiś domyślny		
4	Zakończyć wciskają "dodaj"			
5	Ponowić krok 1			
6	Ponownie dodać ten sam produkt-urządzenie			
7	Z lewego menu wybrać z sekcji produkty wybrać produkt-urządzenie			
8	Wyszukiwać kolejno po wszystkich dostępnych polach, wpisując zarówno dane istniejące jak i fałszywe			
9	Na dowolnie wybranym zarejestrowanym urządzeniu wybierać kolejno ze skrajnej prawej kolumny: a) Edytuj i. Wcisnąć przycisk "Anuluj" ii. Ponowić krok a iii. Zmieniać po kolei wszystkie dostępne pola iv. Zakończyć wciskając "Edytuj b) Podgląd/szczegóły i. Zakończyć wciskając "powrót do listy" c) Usuń	Przy próbie edycji produkt urządzenie i po wciśnięciu zapisz występuje błąd aplikacji		
10	Wcisnąć przycisk "usuń"	Ekran <i>Delete</i> odbiega od standardu aplikacji		
Oczekiwany rezultat (Kryteria akceptacji)	 Dodanie urządzenia (produkt - urządzenie) Wyszukanie urządzenia (produkt – urządzenie) Edycja urządzenia (produkt – urządzenie) Usunięcie urządzenia (produkt – urządzenie) 			
Uzyskany rezultat	Zgodny z oczekiwanym			
Wynik testu	/ status PU. Zgłoszone błędy. Numer zgłoszenia w M kroku 9 zarejestrowana w zgło Zgłoszenie numer 40 rozwiązane w dniu 24/05 74 rozwiązane w dniu 09/06/2013.	antis – 40 i 74 uwaga z szeniu numer 40. 5/2013. Zgłoszenie numer		

6.8.1.9. Obsługa obiektu Produkt - Usługa

Tabela 29. Scenariusz testu: Obsługa obiektu Produkt - Usługa. Źródło: Opracowanie własne.

Nazwa scenariusza	Tworzenie, edycja, usuwanie obiektu Produkt - Usługa
-------------------	--

Warunki początkowe Uruchomiony system Service Desk, Użytkownik z uprawnieni. Administratora Biznesowego							
Realizacja scenariusza testowego							
Krok		Opis	Uwagi				
1	Z lewego - usługa"	menu z sekcji produkty wybrać opcję "produkt następnie "dodaj"					
2	Wcisnąć a) Ponow	przycisk "Anuluj" ić krok 1					
3	Wypełnić a) Po wyt	e dostępne pola pełnieniu spróbować zmienić ich zawartość					
4	Zakończy	vć wciskając "dodaj"					
5	Ponowić	krok 1					
6	ponownie	e dodać ten sam produkt-usługę					
7	Z leweg produkt-u	o menu wybrać z sekcji produkty wybrać Isługa					
8	Wyszukiv wpisując	wać kolejno po wszystkich dostępnych polach, zarówno dane istniejące jak i fałszywe					
9	Na dowo kolejno z a) E b) P d. Us	 lnie wybranej zarejestrowanej usłudze wybierać e skrajnej prawej kolumny: Żdytuj Wcisnąć przycisk "Anuluj" Ponowić krok a Zmieniać po kolei wszystkie dostępne pola Zakończyć wciskając "Edytuj vodgląd/szczegóły Zakończyć wciskając "powrót do listy" 	Ekrany <i>details</i> i <i>delete</i> odbiegają od standardu aplikacji, przy poprawie wzorować się na "firma"				
10	Wcisnąć	przycisk "usuń"					
Oczekiwany rezultat (Kryteria akceptacji)	1. E 2. V 3. E 4. U	Dodanie usługi (produkt - usługa) Vyszukanie usługi (produkt - usługa) dycja usługi (produkt - usługa) Jsunięcie usługi (produkt - usługa)					
Uzyskany rezultat	Zgodny z	oczekiwanym					
Wynik testu / statusPU. Zgłoszone błędy. Numer zgłoszenia w Mantis – 40 i 74. Zgło numer 40 rozwiązane w dniu 24/05/2013. Zgłoszenie num rozwiązane w dniu 09/06/2013.							

6.8.2. Faza I - Weryfikacja wprowadzanych danych

W kolejnej fazie, oznaczonej numerem jeden, przetestowano czy dane wprowadzane do aplikacji są trwałe i zapisują się zgodnie z oczekiwaniami.

6.8.2.1. Weryfikacja wprowadzenia danych – obiekt Firma

Tabela 30. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Firma. Źródło: Opracowanie własne.

Nazwa scei	cenariusza Weryfikacja wprowadzenia danych – obiekt Firma					
Warunki po	czątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami			
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z lewego m firmę"	enu z sekcji firma wybrać opcję "Dodaj				
2	Wypełnić na a. Nazwa b. Dane kc c. NIP d. REGON e. Adres lo f. Kod poo g. Miasto h. Telefon i. Email	stępujące pola: ntaktowe [skalizacji 1 sztowy				
3	Wcisnąć prz	ycisk "Dodaj"				
Oczekiwany rezultat (Kryteria akceptacji)	Zarejestrowano nowy obiekt Firma w bazie danych					
Uzyskany rezultat	Zarejestrowano nowy obiekt Firma w bazie danych					
Wynik testu / status P						

6.8.2.2. Weryfikacja wprowadzenia danych – obiekt Lokalizacja

Tabela 31. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Lokalizacja. Źródło: Opracowanie własne.

Nazwa scenariusza	Weryfikacja wprowadzenia danych – obiekt Lokalizacja		
Warunki początkowe	Uruchomiony system Service Desk, Użytkownik z uprawnieniami Administratora Biznesowego		

Realizacja scenariusza testowego					
Krok		Opis	Uwagi		
1	Z lewego n "Dodaj Loka	nenu z sekcji lokalizacja wybrać opcję Ilizację"			
2	Wypełnić następujące pola: a. Firma b. Adres lokalizacji c. Kod pocztowy d. Miasto e. Jest główna lokalizacją f. Telefon g. Email				
3	Weisnąć przycisk "Zapisz"				
Oczekiwany rezultat (Kryteria	Zarejestrowano nowy obiekt Lokalizacja w bazie danych				
akceptacji)					
Uzyskany rezultat	Zarejestrowano nowy obiekt Lokalizacja w bazie danych				
Wynik test	u / status	PU.			

6.8.2.3. Weryfikacja wprowadzenia danych – obiekt Grupa

Tabela 32. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Grupa. Źródło: Opracowanie własne.

Nazwa scer	nariusza	Weryfikacja wprowadzenia danych – obiekt Grupa					
Warunki po	czątkowe	Użytkownik z uprawnieniami					
		Realizacja scenariusza testowego					
Krok		Opis	Uwagi				
1	Z lewego me	enu z sekcji Grupy wybrać opcję "Dodaj"					
2	Wypełnić wa a. Nazwa	szystkie dostępne pola na formularzu Grupy					
3	Weisnąć prz	ycisk "Dodaj"					
Oczekiwany rezultat (Kryteria	Zarejestrowa	ano nowy obiekt Grupa w bazie danych					
akceptacji) Uzyskany rezultat	Zarejestrowa	nno nowy obiekt Grupa w bazie danych					
Wynik test	u / status	Р					

6.8.2.4. Weryfikacja wprowadzenia danych – przypisanie osób do Grupy

Tabela 33. Scenariusz testu: Weryfikacja wprowadzenia danych - przypisanie osób do Grupy. Źródło: Opracowanie własne.

Nazwa scer	enariusza Weryfikacja wprowadzenia danych – przypisanie osób do Grupy					
Warunki po	początkowe Uruchomiony system Service Desk, Użytkownik z upraw Administratora Biznesowego					
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z lewego "Wyszukaj"	menu z sekcji Grupy wybrać opcję				
2	Wcisnąć ikonę ² " <i>GroupEmployees</i> " przy wybranej grupie					
3	Zaznaczyć osoby przypisywane do grupy					
4	Wcisnąć przycisk "Dodaj"					
Oczekiwany rezultat	Zarejestrowano nowy skład grupy w bazie danych					
(Kryteria akceptacji)						
Uzyskany rezultat	Zarejestrowano nowy skład grupy w bazie danych					
Wynik testu / status P						

6.8.2.5. Weryfikacja wprowadzenia danych – obiekt Osoba

Tabela 34. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Osoba. Źródło: Opracowanie własne.

Nazwa scei	nariusza	Weryfikacja wprowadzenia danych – obiekt Osoba						
Warunki początkowe		Uruchomiony system Service Desk, Użytkownik z upra Administratora Biznesowego		uprawnieniami				
		Realizacja	scenari	usza testo	owego			
Krok	Opis			Uwagi				
1	Z lewego menu z sekcji Osoby wybrać opcję "Dodaj osobe"							
2	Wypełnić na a. Imię b. Nazwisł c. Telefon d. E-mail	stępujące pola: co						

3	Weisnąć prz	ycisk "Dodaj"	
Oczekiwany rezultat	Zarejestrowa	no nowy obiekt Osoba w bazie danych	
(Kryteria akceptacji)			
Uzyskany rezultat	Zarejestrowa	no nowy obiekt Osoba w bazie danych	
Wynik test	u / status	Р	

6.8.2.6. Weryfikacja wprowadzenia danych – obiekt Pracownik

Tabela 35. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Pracownik. Źródło: Opracowanie własne.

Nazwa scer	nariusza	usza Weryfikacja wprowadzenia danych – obiekt Pracownik					
Warunki po	oczątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami				
		Realizacja scenariusza testowego					
Krok		Opis	Uwagi				
1	Z lewego n "Pracownicy	nenu z sekcji Pracownicy wybrać opcję					
2	Wcisnąć prz	ycisk "Dodaj"					
3	Wypełnić na a. PESEL b. <i>Position</i> c. Data zat d. Login e. Hasło f. Imię g. Nazwisł h. Telefon i. E-mail	stępujące pola: rudnienia	Nazwa pola <i>Position</i> powinna być przetłumaczona na język Polski				
4	Wcisnąć prz	ycisk "Dodaj"					
Oczekiwany rezultat (Kryteria	Zarejestrowano nowy obiekt Pracownik w bazie danych						
akceptacji)							
Uzyskany rezultat	Zarejestrowano nowy obiekt Pracownik w bazie danych						
Wynik testu / status P							

6.8.2.7. Weryfikacja wprowadzenia danych – obiekt Stanowisko

Tabela 36. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Stanowisko. Źródło: Opracowanie własne.

Nazwa scei	nariusza	iusza Weryfikacja wprowadzenia danych – obiekt Stanowisko				
Warunki po	Warunki początkowe Uruchomiony system Service Desk, Użytkownik z upraw Administratora Biznesowego					
		Realizacja scenariusza testowego	-			
Krok		Opis	Uwagi			
1	Z lewego n "Stanowiska	Z lewego menu z sekcji Pracownicy wybrać opcję "Stanowiska"				
2	Wcisnąć prz	Wcisnąć przycisk "Dodaj stanowisko"				
3	Wypełnić następujące pola a. Nazwa					
4	Wcisnąć prz	ycisk "Dodaj"				
Oczekiwany rezultat	Zarejestrowano nowy obiekt Stanowisko w bazie danych					
(Kryteria akceptacji)						
Uzyskany rezultat	Zarejestrowano nowy obiekt Stanowisko w bazie danych					
Wynik testu	ı / status	Р				

6.8.2.8. Weryfikacja wprowadzenia danych – obiekt Produkt Urządzenie

Tabela 37. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Produkt Urządzenie. Źródło: Opracowanie własne.

Nazwa scer	Weryfikacja w	prowadze	enia dany	ch – obi	ekt Produkt U	rząc	lzenie	
Warunki po	czątkowe	Uruchomiony system Service Desk, Administratora Biznesowego		Użytkownik	z	uprawnieniami		
		Realizacja	scenari	usza testo	owego			
Krok	Opis				Uw	vagi		
1	Z lewego menu z sekcji Produkty wybrać opcję "Produkt - urządzenie"							
2	Wcisnąć przycisk "Dodaj"							
3	Wypełnić następujące pola: a. Numer seryjny b. Model c. Cechy							

	d. Opis			
4	Wcisnąć przycisk "Dodaj"			
Oczekiwany rezultat (Kryteria akcentacii)	Zarejestrowano nowy obiekt Produkt - urządzenie w bazie danych			
Uzyskany rezultat	Zarejestrowano nowy obiekt Produkt - urządzenie w bazie danych			
Wynik test	u / status	Р		

6.8.2.9. Weryfikacja wprowadzenia danych – obiekt Produkt Usługa

Tabela 38. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Produkt Usługa. Źródło: Opracowanie własne.

Nazwa scenariusza		Weryfikacja wprowadzenia danych – obie	kt Produkt Usługa		
Warunki po	czątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami		
		Realizacja scenariusza testowego			
Krok		Opis	Uwagi		
1	Z lewego u "Produkt - u	Z lewego menu z sekcji Produkty wybrać opcję			
2	Wcisnąć prz	Weisnąć przycisk "Dodaj"			
3	Wypełnić następujące pola: a. Numer usługi b. Opis				
4	Wcisnąć przycisk "Dodaj"				
Oczekiwany rezultat (Kryteria akceptacji)	Zarejestrowano nowy obiekt Produkt - usługa w bazie danych				
Uzyskany rezultat	Zarejestrowa danych	no nowy obiekt Produkt - usługa w bazie			
Wynik testu / status		Р			

6.8.2.10. Weryfikacja wprowadzenia danych – obiekt Producent

Tabela 39. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Producent. Źródło: Opracowanie własne.

Nazwa scenariusza		Weryfikacja wprowadzenia danych – obiekt Producent				
Warunki po	czątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami			
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z lewego r "Dodaj"	Z lewego menu z sekcji Producent wybrać opcję "Dodaj"				
2	Wcisnąć prz	Wcisnąć przycisk "Dodaj"				
3	Wypełnić następujące pola: a. Nazwa					
4	Wcisnąć przycisk "Dodaj"					
Oczekiwany rezultat	Zarejestrowano nowy obiekt Producent w bazie danych					
(Kryteria akceptacji)						
Uzyskany rezultat	Zarejestrowano nowy obiekt Producent w bazie danych					
Wynik test	u / status	Р				

6.8.2.11. Weryfikacja wprowadzenia danych – obiekt Zgłoszenie

Tabela 40. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Zgłoszenie. Źródło: Opracowanie własne.

Nazwa scenariusza		Weryfikacja wprowadzenia danych – obie	kt Zgłoszenie
Warunki po	czątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Z lewego n "Rejestracja	nenu z sekcji Zgłoszenia wybrać opcję zgłoszenia"	
2	Wypełnić na a. Nr umo b. Osoba o c. Typ zgło	stępujące pola: wy dpowiedzialna oszenia	
3	Wcisnąć prz	ycisk "Dalej"	
4	Wypełnić wszystkie pola a. <i>Location</i> b. <i>Product</i> c. Opis		Nazwy pól <i>Location</i> i <i>Product</i> powinny być przetłumaczone na język Polski

5	Weisnąć prz	ycisk "Dodaj"
Oczekiwany rezultat	Zarejestrowa danych	no nowy obiekt Zgłoszenie w bazie
(Kryteria akceptacji)		
Uzyskany rezultat	Zarejestrowa danych	no nowy obiekt Zgłoszenie w bazie
Wynik test	u / status	PU. Zgłoszone uwagi. Numer Mantis - 49

6.8.3. Faza II - testy interfejsu Użytkownika

Ostatnia faza testów obejmowała bardziej skomplikowane czynności, jak chociażby rejestrację zgłoszenia, dodawanie nowych osób do bazy, mechanizm wyszukiwania, edycję workflow, czy też test działania gotowych przepływów.

6.8.3.1. Rejestracja zgłoszenia przez Konsultanta

Tabela 41. Scenariusz testu: Rejestracja zgłoszenia przez Konsultanta. Źródło: Opracowanie własne.

Nazwa scenariusza		Rejestracja zgłoszenia przez Konsultanta	
Warunki po	oczątkowe	Uruchomiony system Service Desk, Administratora Biznesowego	Użytkownik z uprawnieniami
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Z lewego r "rejestracja "rejestracja z	nenu z sekcji zgłoszenia wybrać opcję zgłoszenia" lub z górnej belki wybrać zgłoszenia	Nie działa przycisk rejestracja zgłoszenia z górnej belki
2	Wcisnąć prz a. Ponowie	ycisk "Anuluj" č krok 1	Nie działa przycisk anuluj.
3	 Wypełnić następujące pola: a. Numer umowy Wybsać ręcznie Wyszukać korzystając z "lupki" b. Wybrać osobę odpowiedzialną Wybrać Typ zgłoszenia Nacisnąć przycisk dalej 		 Przy wyszukiwaniu umowy: Brak możliwości wyszukiwania po numerze umowy na liście brak nazwy firmy. Po co są numerowane strony skoro jest tylko 6 umów. Godzina zawarcia umowy raczej nie potrzebna Nie ma żadnego typu zgłoszenia do wyboru

			Błąd zapisu odpow	aplikacji . Bra viedzialnej	przy ak	próbie osoby
4	Wypełnić na a. Wybrać b. Wybrać c. Podać o	Wypełnić następujące pola: a. Wybrać lokalizację z listy b. Wybrać produkt c. Podać opis				
5	Nacisnąć prz	Nacisnąć przycisk "dalej"				
Oczekiwany rezultat	Zarejestrowane zgłoszenie					
(Kryteria akceptacji)						
Uzyskany rezultat	Zarejestrowane zgłoszenie					
Wynik testu / status		PU. Zgłoszone błędy. Numer zgłoszen rozwiązane w dniu 01/06/2013.	ia w N	Iantis – 6	6, Zgł	oszenie

6.8.3.2. Dodanie pracowników do grupy

				,	
TT 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	a • • •	D 1 '	.1 / 1	77 / 11	\sim \cdot 1
Tabala / 7	Neanarillez factur	Lodonia nro/	numizou do	aring Irodia	· I Inracowania Włacna
\mathbf{I} and \mathbf{I}	OUTHAILUSZ ITSUL.	\mathbf{D}	$A \cup W \cup U \cup V \cup U \cup U$	21000×10000	
1 40 0 14 121		200000 prov			

Nazwa scena	ariusza	Dodanie pracowników grupy		
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Z lewego	Z lewego menu sekcji grupy wybrać opcję "wyszukaj"		
2	Przejść d poprzez w	Przejść do listy osób dla dowolnej wyszukanej grupy poprzez wybór ikony "osoby"		
3	Zaznaczenie flagi obok interesującej osoby.			
Oczekiwany rezultat (Kryteria akceptacii)	Pracownik został dodany do grupy			
Uzyskany rezultat	Pracownik został dodany do grupy			
Wynik testu	/ status	Р		

6.8.3.3. Automatyczne nadawanie numeru zgłoszeń

Tabela 43. Scenariusz testu: Automatyczne nadawanie numeru zgłoszeń. Źródło: Opracowanie własne.

Nazwa scenariusza		Automatyczne nadawanie numeru zgłoszeń		
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Z leweg "rejestrac "rejestrac	o menu z sekcji zgłoszenia wybrać opcję ja zgłoszenia" lub z górnej belki wybrać ja zgłoszenia		
2	Wypełnić a. Numo i. V ii. V b. Wybi c. Wybi	 Wypełnić następujące pola: a. Numer umowy Wpisać ręcznie Wyszukać korzystając z "lupki" b. Wybrać osobę odpowiedzialną Wybrać Typ zgłoszenia 		
3	Nacisnąć	przycisk dalej		
4	Wypełnić a. Adre: b. Produ c. Opis	e następujące pola: s ikt	Przejście do ekranu wyszukiwania zgłoszeń. Brak widocznych numerów zgłoszeń. Brak możliwości sprawdzenia, edycji bądź usunięcia zgłoszeń.	
5	Nacisnąć przycisk dalej			
Oczekiwany rezultat	Wyświetlenie nadawanego numeru zgłoszenia.			
(Kryteria akceptacji)				
Uzyskany rezultat	Przejście do ekranu wyszukiwania zgłoszeń.			
Wynik testu	/ status	PU. Zgłoszone błędy. Numer zgłoszenia w Mar rozwiązane w dniu 05/06/2013	ntis – 62 i 63. Zgłoszenia	

6.8.3.4. Przeszukiwanie bazy osób zgłaszających po dostępnych polach

Tabela 44. Scenariusz testu: Przeszukanie bazy osób zgłaszających po dostępnych polach. Źródło: Opracowanie własne.

Nazwa scenariusza		Przeszukiwanie bazy osób zgłaszających po dost	ępnych polach	
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytko Administratora Biznesowego	ownik z uprawnieniami	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Z leweg "wyszuki	Z lewego menu z sekcji osoby wybrać opcję "wyszukiwanie osoby".		
2	W odpowiednich polach wpisać dane filtrujące (np. Telefon).			
Oczekiwany rezultat (Kryteria akceptacji)	Wyświetlona lista osób zgłaszających po zastosowaniu filtra			
Uzyskany rezultat	Wyświetlona lista osób zgłaszających po zastosowaniu filtra			
Wynik testu	/ status	Р		

6.8.3.5. Przeszukiwanie bazy listy lokalizacji po dostępnych polach

Tabela 45. Scenariusz testu: Przeszukiwanie bazy listy lokalizacji po dostępnych polach. Źródło: Opracowanie własne.

Nazwa scenariusza		Przeszukiwanie bazy listy lokalizacji po dostępn	ych polach	
Warunki poc	zątkowe	Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Z lewego menu z sekcji lokalizacja wybrać opcję "wyszukiwanie lokalizacji".			
2	W odpowiednich polach wpisać dane filtrujące (np. Miasto).			
Oczekiwany rezultat	Wyświetlona lista lokalizacji po zastosowaniu filtra			
(Kryteria akceptacji)				
Uzyskany rezultat	Wyświetł			

Wynik testu / status	Р

6.8.3.6. Wyświetlanie listy zgłoszeń

Tabela 46. Scenariusz testu: Wyświetlanie listy zgłoszeń. Źródło: Opracowanie własne.

Nazwa scenariusza		Wyświetlanie listy zgłoszeń				
Warunki początkowe		Uruchomiony system Service Desk, Użytkownik z uprawnieniami Administratora Biznesowego				
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z leweg "realizow	Z lewego menu z sekcji zgłoszenia wybrać opcję "realizowane zgłoszenia"				
2	Z lewego menu z sekcji zgłoszenia wybrać "wyszukiwanie zgłoszeń"					
Oczekiwany rezultat (Kryteria akceptacji)	Wyświetlona lista zgłoszeń					
Uzyskany rezultat	Wyświetlona lista zgłoszeń					
Wynik testu	/ status	Р				

6.8.3.7. Wyszukiwanie zgłoszeń po numerze zgłoszenia lub numerze umowy

Tabela 47. Scenariusz testu: Wyszukiwanie zgłoszeń po dostępnych polach. Źródło: Opracowanie własne.

Nazwa scenariusza		Wyszukiwanie zgłoszeń po opisie, urządzeni statusie etc.	u, lokalizacji, aktualnym	
Warunki początkowe		Uruchomiony system Service Desk, Użytk Administratora Biznesowego	ownik z uprawnieniami	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Z leweg "wyszuki	o menu z sekcji zgłoszenia wybrać opcję wanie zgłoszeń"		
2	Wpisanie Numer un	danych, po których chcemy wyszukać np. nowy.		
3	Kliknąć przycisk Dalej.			
Oczekiwany rezultat (Kryteria	Wyświetl	one zgłoszenia po zastosowaniu filtra		

akceptacji)			
Uzyskany rezultat	Wyświetl	one zgłoszenia po zastosowaniu filtra	
Wynik testu / status		Р	

6.8.3.8. Wyświetlanie listy zgłoszeń względem kończących się czasów SLA

Tabela 48. Scenariusz testu: Wyświetlanie listy zgłoszeń względem kończących się czasów SLA. Źródło: Opracowanie własne.

Nazwa scenariusza		Wyświetlanie listy zgłoszeń względem kończących się czasów SLA					
Warunki początkowe		Uruchomiony system Service Desk, Użytko Administratora Biznesowego	ownik z uprawnieniami				
		Realizacja scenariusza testowego					
Krok		Opis	Uwagi				
1	Z lewego "realizow	Z lewego menu z sekcji zgłoszenia wybrać opcję "realizowane zgłoszenia" lub "wyszukiwanie zgłoszeń"					
2	Podać kryteria wyszukiwania - opcjonalne						
3	Kliknąć na nagłówek kolumny "Liczba dni do zamknięcia Sortowanie r zgłoszenia"						
Oczekiwany rezultat	Lista zgł czasu SLA	oszeń wyświetlona i posortowana względem A					
(Kryteria akceptacji)							
Uzyskany rezultat	Nieposort	owana lista					
Wynik testu	/ status	Zgłoszone błędy. Numer zgłoszenia w Mantis - 6	58				

6.8.3.9. Aktualizacja statusu zgłoszeń

Tabela 49. Scenariusz testu: Aktualizacja statusu zgłoszeń. Źródło: Opracowanie własne.

Nazwa scenariusza		Aktualizacja statusu zgłoszeń	
Warunki początkowe		Uruchomiony system Service Desk, Użytko Administratora Biznesowego	ownik z uprawnieniami
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Z leweg	o menu z sekcji zgłoszenia wybrać opcję	
	"icalizow	ane zgioszenia nuo "wyszukiwanie zgioszeni	

3	Wybrać zgłoszenie i wejść w tryb edycji poprzez kliknięcie "Edit"				
4	Z listy rozwijalnej zmienić status				
5	Nacisnąć przycisk "zapisz"				
Oczekiwany rezultat	Zaktualizowane zgłoszenie – zmieniony status				
(Kryteria akceptacji)					
Uzyskany rezultat	Zaktualizowane zgłoszenie – zmieniony status				
Wynik testu	Wynik testu / status P				

6.8.3.10. Samodzielne definiowanie kroków w workflow

Tabela 50. Scenariusz testu: Samodzielne definiowanie kroków w *workflow*. Źródło: Opracowanie własne.

Nazwa scenariusza		Samodzielne definiowanie kroków w workflow				
Warunki początkowe		Zainstalowana biblioteka AvalonDock, zainstalowana aplikacja PJWSTK Workflow Designer				
		Realizacja scenariusza testowego				
Krok		Opis	Uwagi			
1	Z górnego	o menu, wybrać Plik → Nowy workflow				
2	Z panelu	usunąć element Sequence				
3	Z sekcji Flowchar	Z sekcji Toolbox przeciągnij na panel kontrolkę Flowchart				
4	Z sekcji przepływ	Z sekcji <i>Toolbox</i> używaj kontrolek budując swój przepływ				
5	Na konie Wybierz	Na koniec z górnego menu, wybrać Plik → Zapisz. Wybierz katalog\PJWSTK.Serwis\Workflows\				
Oczekiwany rezultat	Kroki w <i>workflow</i> zdefiniowane					
(Kryteria akceptacji)						
Uzyskany rezultat	Uzyskany rezultat Kroki w <i>workflow</i> zdefiniowane					
Wynik testu	/ status	Р				

6.8.3.11. *Workflow* drukarka nie drukuje przypadek 1 (brak papieru)

Tabela 51. Scenariusz testu: *Workflow* "drukarka nie drukuje" - brak papieru, załadowanie papieru. Źródło: Opracowanie własne.

Nazwa scenariusza		<i>Workflow</i> , drukarka nie drukuje" - winny brak papieru w drukarce			
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świeca się diody".			
		Realizacja scenariusza testowego			
Krok		Opis	Uwagi		
1	Na pytanie odpowiedź: Tak	czy świecą się diody/ekran wybieramy			
2	Czy drukarka Nie	Czy drukarka ma załadowany papier? Wybór odpowiedzi: Nie			
3	Czy problem Tak				
4	Wysłanie ma	ila	Przy próbie wysłania pustego maila nic się nie dzieje - propozycja wyświetlenia alertu o niemożliwości wysłania maila bez nadawcy		
Oczekiwany rezultat (Kryteria	Zakończenie " <i>closed</i> " oraz	<i>workflow.</i> Przypisanie do zgłoszenia statusu wysłanie maila o wprowadzonej treści.			
Uzyskany rezultat	Zakończenie " <i>closed</i> " oraz	workflow. Przypisanie do zgłoszenia statusu wysłanie maila o wprowadzonej treści.			
Wynik testu / status		PU. Zgłoszone błędy. Numer Mantis – 60. Zgło 01/06/2013.	oszenie rozwiązane w dniu		

Tabela 52. Scenariusz testu: *Workflow* "drukarka nie drukuje" – brak papieru, wymiana drukarki. Źródło: Opracowanie własne.

Nazwa scenariusza		<i>Workflow</i> "drukarka nie drukuje" - drukarka nie możliwa do naprawy					
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się diody".					
		Re	ealizacja	scena	riusza testow	ego	
Krok			(Opis			Uwagi
1	Na pytanie odpowiedź: Tak	czy	świecą	się	diody/ekran	wybieramy	

2	Czy drukarka ma załadowany papier? Wybór odpowiedzi:				
	Nie				
3	Polecenie załadowania papieru.				
	Czy problem został rozwiązany? Wybór odpowiedzi: Nie				
4	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi:				
	Tak				
5	Infomacja o zawiadomieniu konsultantów L2 – e-mail automatyczny				
6	Czy rozwiązano problem? Wybór odpowiedzi:				
	Nie				
7	Czy szacunkowy koszt naprawy przekracza cenę nowej drukarki? Wybór odpowiedzi: Tak				
8	Komunikat o zamówieniu nowej drukarki				
9	Komunikat o wymianie drukarki przez konsultanta L2				
10	Zakończenie workflow z statusem "closed"				
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed".				
(Kryteria akceptacji)					
Uzyskany rezultat	Uzyskany rezultat Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> ".				
Wynik testu	I / status P				

Tabela 53. Scenariusz testu: *Workflow* ,,drukarka nie drukuje" - brak papieru, brak zgody na koszty. Źródło: Opracowanie własne.

Nazwa scer	nariusza	Workflow "drukarka nie drukuje" - drukarka nie możliwa do naprawy		
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się diody".		
Realizacja scenariusza testowego				
Krok	Opis Uwagi			
1	Na pytanie odpowiedź: Tak	czy świecą się diody/ekran wybieramy		
2	Czy drukarka Nie			
3	Polecenie zał	adowania papieru.		

	Czy problem Nie	został rozwiązany? Wybór odpowiedzi:				
4	Czy zgłaszaja Wybór odpov	acy ma abonament na wsparcie bezpośrednie? viedzi:				
	Nie					
5	Czy zgłaszaja odpowiedzi: Nie	ący akceptuje koszty obsługi L2? Wybór	Pytanie ma błąd językowy: jest: 'akceptujesz' powinno: 'akceptuje'			
6	Wysłanie ma	Wysłanie maila				
Oczekiwany rezultat	Zakończenie "closed". Wy	workflow. Przypisanie do zgłoszenia statusu słanie maila potwierdzającego rezygnacje				
(Kryteria akceptacji)						
Uzyskany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> ". Wysłanie maila potwierdzającego rezygnacje					
Wynik testu / status		PU. Zgłoszone błędy. Numer mantis – 60. Zgło 01/06/2013.	szenie rozwiązane w dniu			

Tabela 54. Scenariusz testu: *Workflow* "drukarka nie drukuje" - brak papieru, wymiana części. Źródło: Opracowanie własne.

Nazwa scenariusza		Workflow "drukarka nie drukuje" - drukarka możliwa do naprawy	
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świeca sie diody".	
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Na pytanie odpowiedź: Tak	czy świecą się diody/ekran wybieramy	
2	Czy drukarka ma załadowany papier ? Wybór odpowiedzi:		
	Nie		
3	Polecenie załadowania papieru.		
	Czy problem został rozwiązany? Wybór odpowiedzi: Nie		
4	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi:		
	Nie		
5	Czy zgłaszający akceptuje koszty obsługi L2? Wybór odpowiedzi: Tak		
6	Mail wygene	rowany przez system do konsultanta L2 z	

	informacją o przypisaniu zgłoszer	ia	
7	Czy rozwiązano problem? Wybór	odpowiedzi:	
9	Czy szacunkowy koszt naprawy przekracza cenę nowej drukarki? Wybór odpowiedzi: Nie		
10	Czy części są dostępne? Wybór o Tak	lpowiedzi:	
11	Wymiana uszkodzonych części Czy drukarka działa poprawnie? Wybór odpowiedzi: Tak		
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed".		
(Kryteria akceptacji)			
Uzyskany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed".		
Wynik testu / status P			

6.8.3.12. *Workflow* drukarka nie drukuje przypadek 2 (diody się nie świecą)

Tabela 55. Scenariusz testu: *Workflow* "drukarka nie drukuje" - diody się nie świecą, poprawienie kabla naprawia problem. Źródło: Opracowanie własne.

Nazwa scenariusza		Workflow "drukarka nie drukuje" - winny źle podłączony kabel	
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku workflow, na którym wyświetlane jest pytanie "Czy świeca się diody".	
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Na pytanie odpowiedź: Nie	czy świecą się diody/ekran wybieramy	
2	Polecenie sprawdzenia kabla zasilającego i uruchomienie drukarki poprzez jej przycisk.		
	Czy problem został rozwiązany? Wybór odpowiedzi: Tak		
3	Czy problem został rozwiązany? Wybór odpowiedzi: Tak		
4	Wysłanie maila		
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed" oraz wysłanie maila o wprowadzonej treści.		
(Kryteria akceptacji)			

Uzyskany	Zakończenie	workflow. Przypisanie do zgłoszenia statusu	
rezultat	"closed" oraz	z wysłanie maila o wprowadzonej treści.	
Wynik testu / status		Р	

Tabela 56. Scenariusz testu: *Workflow* "drukarka nie drukuje" - diody się nie świecą, konsultant L2 naprawia problem. Źródło: Opracowanie własne.

Nazwa scenariusza		<i>Workflow</i> "drukarka nie drukuje" - brak zapa wizyta konsultantów	llonych diod i potrzebna	
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszen "Drukarka nie drukuje". Rozpoczęcie testu nastę workflow, na którym wyświetlane jest pytanie "C	ie wybierając <i>workflow</i> puje od pierwszego kroku Czy świecą się diody".	
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Na pytanie odpowiedź: Nie	czy świecą się diody/ekran wybieramy		
2	Polecenie spi drukarki pop Czy problem	Polecenie sprawdzenia kabla zasilającego i uruchomienie drukarki poprzez jej przycisk. Wybór odpowiedzi: Czy problem został rozwiązany?		
3	Nie Czy problem został rozwiązany? Wybór odpowiedzi: Nie Nie			
4	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi:			
5	I dK	poinformowaniu konsultantów I 2		
6	Czy rozwiązano problem? Wybór odpowiedzi: Tak			
7	Automatyczny mail został wysłany.			
8	Zakończenie workflow z statusem "closed"			
Oczekiwany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> ". Wysłanie automatycznego maila.			
(Kryteria akceptacji)				
Uzyskany rezultat	Zakończenie "closed". Wy	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> ". Wysłanie automatycznego maila.		
Wynik testu	ı / status	Р		

Tabela 57. Scenariusz testu: *Workflow* "drukarka nie drukuje" - diody się nie świecą, wymiana drukarki. Źródło: Opracowanie własne.

Nazwa scenariusza <i>Workflow</i> "drukarka nie drukuje" - brak zapalonych diod i potrzebna wizyta konsultantów

Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszen "Drukarka nie drukuje". Rozpoczęcie testu nastę workflow, na którym wyświetlane jest pytanie "C	ie wybierając <i>workflow</i> puje od pierwszego kroku Czy świecą się diody".	
Realizacja scenariusza testowego				
Krok		Opis	Uwagi	
1	Na pytanie odpowiedź: Nie	czy świecą się diody/ekran wybieramy		
2	Polecenie spr drukarki pop Czy problem	rawdzenia kabla zasilającego i uruchomienie rzez jej przycisk. Wybór odpowiedzi: został rozwiązany?		
	Nie			
3	Czy problem Nie	został rozwiązany? Wybór odpowiedzi:		
4	Czy zgłaszaja Wybór odpov Nie	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi:		
5	Czy zgłaszający akceptuje koszty obsługi L2? Wybór odpowiedzi:			
6	Infomacia o zavviodomianiu konsultantów I 2			
8		Czy rozwiązano problem? Wybór odpowiedzi:		
/	Czy rozwiązano problem? wybor odpowiedzi: Nie			
8	Czy szacunkowy koszt naprawy przekracza cenę nowej drukarki? Wybór odpowiedzi: Nie			
9	Czy części są Nie	dostępne? Wybór odpowiedzi:		
10	Informacja o	potrzebie zamówienia nowej drukarki.		
11	Komunikat o	Komunikat o wymianie drukarki przez konsultanta L2		
12	Zakończenie workflow z statusem "closed"			
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed".			
(Kryteria akceptacji)				
Uzyskany rezultat	Zakończenie " <i>closed</i> ".	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed".		
Wynik testu	ı / status	Р		

6.8.3.13. *Workflow* drukarka nie drukuje przypadek 3 (brak tuszu/tonera)

Tabela 58. Scenariusz testu: *Workflow* "drukarka nie drukuje" - brak tuszu/tonera. Źródło: Opracowanie własne.

Nazwa scen	nariusza	<i>Workflow</i> drukarka nie drukuje" – brak tuszu/to	nera	
Warunki początkowe		Użytkownik stworzył odpowiednie zgłoszenie wybierając <i>workflow</i> "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się diody".		
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Na pytanie odpowiedź: Tak	czy świecą się diody/ekran wybieramy		
2	Czy drukarka Tak	Czy drukarka ma załadowany papier? Wybór odpowiedzi: Tak		
3	Czy drukarka wskazuje na zacięcie się papieru? Wybór odpowiedzi: Nie			
4	Czy drukarka wskazuje na brak tonera komunikatem bądź zapaleniem się odpowiedniej diody? Wybór odpowiedzi: Tak			
5	Polecenie wymiany tonera. Czy problem został rozwiązany? Wybór odpowiedzi: Tak			
6	Wysłanie ma	Wysłanie maila		
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed" oraz wysłanie maila o wprowadzonej treści.			
(Kryteria akceptacji)				
Uzyskany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed" oraz wysłanie maila o wprowadzonej treści.			
Wynik testu	ı / status	Р		

6.8.3.14. *Workflow* drukarka nie drukuje przypadek 4 (zacięty papier)

Tabela 59. Scenariusz testu: *Workflow* "drukarka nie drukuje" - zacięty papier. Źródło: Opracowanie własne.

Nazwa scenariusza	Workflow "drukarka nie drukuje" – zacięty papier	
Warunki początkowe	 Workflow "drukarka nie drukuje" – zacięty papier Użytkownik stworzył odpowiednie zgłoszenie wybierając workflow "Drukarka nie drukuje". Rozpoczęcie testu następuje od pierwszeg kroku workflow, na którym wyświetlane jest pytanie "Czy świecą si diody". 	

Realizacja scenariusza testowego				
Krok		Opis	Uwagi	
1	Na pytanie odpowiedź: Tak	czy świecą się diody/ekran wybieramy		
2	Czy drukarka	a ma załadowany papier? Wybór odpowiedzi:		
	Tak			
3	Czy drukarka odpowiedzi: Tak	Czy drukarka wskazuje na zacięcie się papieru? Wybór odpowiedzi: Tak		
4	Polecenie otv papieru.	Polecenie otwarcia pokrywy drukarki i próby usunięcia papieru.		
	Czy problem został rozwiązany? Wybór odpowiedzi: Tak			
5	Wysłanie ma	uila		
Oczekiwany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie maila o wprowadzonej treści.			
(Kryteria akceptacji)				
Uzyskany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed" oraz wysłanie maila o wprowadzonej treści.			
Wynik testu	ı / status	Р		

6.8.3.15. *Workflow* awaria samochodu przypadek 1 (kontrolki się nie świecą)

Tabela 60. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się nie świecą, poprawienie klem naprawia problem. Źródło: Opracowanie własne.

Nazwa scer	nariusza	Workflow "awaria samochodu" - klemy źle założ	żone
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".	
		Realizacja scenariusza testowego	
Krok	Opis Uwagi		Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka ? Nie		
2	Sprawdzenie klem na akumulatorze. Czy problem został rozwiązany? Wybór odpowiedzi:		
	Tak		
3	Automatyczne wysłanie maila i zamkniecie zgłoszenia ze statusem "closed" Mail nie jest wysyłany		Mail nie jest wysyłany

Oczekiwany rezultat	Zakończenie "closed" oraz	<i>workflow</i> . Przypisanie do zgłoszenia statusu z wysłanie automatycznego maila.	
(Kryteria akceptacji)			
Uzyskany rezultat	Zakończenie "closed" oraz	<i>workflow</i> . Przypisanie do zgłoszenia statusu z wysłanie automatycznego maila.	
Wynik testu / status		PU. Zgłoszone błędy. Numer Mantis - 61	

Tabela 61. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się nie świecą, konsultant L2 naprawia problem. Źródło: Opracowanie własne.

Nazwa scenariusza		Workflow "awaria samochodu" - interwencja konsultanta L2	
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".	
	1	Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka ? Wybór odpowiedzi: Nie		
2	Sprawdzenie klem na akumulatorze. Czy problem został rozwiązany? Wybór odpowiedzi:		
3	Czy zgłaszający ma abonament na wsparcie bezpośrednie?		
	Wybór odpov	wiedzi:	
	Tak		
4	Wysłanie wsparcia L2		
5	Czy rozwiązano problem? Wybór odpowiedzi: Tak		
6	Automatyczne wysłanie maila i zamkniecie zgłoszenia ze statusem " <i>closed</i> "		
Oczekiwany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie automatycznego maila.		
(Kryteria akceptacji)			
Uzyskany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie automatycznego maila.		
Wynik testu / status		Р	

Tabela 62. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się nie świecą, naprawione w serwisie. Źródło: Opracowanie własne.

Nazwa scenariusza		Workflow "awaria samochodu" - naprawa w serwisie	
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".	
	1	Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka wybieramy odpowiedź: Nie		
2	Sprawdzenie klem na akumulatorze. Czy problem został rozwiązany? Wybór odpowiedzi:		
3	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi:		
4	Czy zgłaszający akceptuje koszty obsługi L2? Wybór odpowiedzi: Tak		
5	Wysłanie ws	parcia L2	
6	Czy rozwiązano problem? Wybór odpowiedzi: Nie		
7	Czy abonament zgłaszającego pokrywa koszty transportu do serwisu? Wybór odpowiedzi: Tak		
8	Konsultant L2 zabiera pojazd do serwisu		
9	Zamkniecie zgłoszenia ze statusem " <i>closed – Repaired by</i> service"		
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed – Repaired by service".		
(Kryteria akceptacji)			
Uzyskany rezultat	y Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu ,, <i>closed – Repaired by service</i> ".		
Wynik testu	ı / status	Р	

6.8.3.16. *Workflow* awaria samochodu przypadek 2 (kontrolki się świecą)

Tabela 63. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się świecą wciśnięcie sprzęgła naprawia problem. Źródło: Opracowanie własne.

Nazwa scenariusza	Workflow "awaria samochodu" - problem z rozrusznikiem
Ivazwa scenariusza	<i>workflow</i> , awaria samochodu - problem z rozruszinkiem

Warunki początkowe		Użytkownik ma utworzony dostęp do systen zgłoszenie, w którym wybiera <i>workflow</i> Rozpoczęcie testu następuje od pierwszego kro wyświetlane jest pytanie "Czy świecą się kontro po przekręceniu kluczyka?".	nu. Tworzy odpowiednie "Awaria samochodu". oku <i>workflow</i> , na którym olki na desce rozdzielczej
		Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka wybieramy odpowiedź: Tak		
2	Czy rozrusznik próbuje uruchomić silnik? Wybór odpowiedzi:		
	Nie		
3	Wciśnij sprzęgło w trakcie uruchamiania silnika. Czy problem został rozwiązany? Wybór odpowiedzi:		
	Tak		
4	Automatyczne wysłanie maila i zamknięcie zgłoszenia ze statusem " <i>closed</i> "		
Oczekiwany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie automatycznego maila.		
(Kryteria akceptacji)			
Uzyskany rezultat	Uzyskany rezultatZakończenie workflow. Przypisanie do zgłoszenia statusu "closed" oraz wysłanie automatycznego maila.		
Wynik testu	ı / status	Р	

Tabela 64. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się świecą, konsultant L2 naprawia problem. Źródło: Opracowanie własne.

Nazwa scenariusza		Workflow "awaria samochodu" - problem z rozrusznikiem II	
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".	
Realizacja scenariusza testowego			
Krok	Opis Uwagi		Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka wybieramy odpowiedź: Tak		
2	Czy rozrusznik próbuje uruchomić silnik? Wybór odpowiedzi:		
	Nie		
3	Wciśnij sprze Czy problem	gło w trakcie uruchamiania silnika. został rozwiązany? Wybór odpowiedzi:	

	Nie		
4	Czy zgłaszaja Wybór odpov Tak	acy ma abonament na wsparcie bezpośrednie? viedzi:	
5	Wysłanie ws	parcia L2	
6	Czy rozwiązano problem? Wybór odpowiedzi: Tak		
7	Automatyczne wysłanie maila i zamkniecie zgłoszenia z statusem " <i>closed</i> "		
Oczekiwany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie automatycznego maila.		
(Kryteria akceptacji)			
Uzyskany rezultat	Zakończenie <i>workflow</i> . Przypisanie do zgłoszenia statusu " <i>closed</i> " oraz wysłanie automatycznego maila.		
Wynik testu	ı / status	Р	

Tabela 65. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się świecą, brak akceptacji kosztów transportu. Źródło: Opracowanie własne.

Nazwa scenariusza		<i>Workflow</i> "awaria samochodu" - brak abonamentu i brak zgody na transport zepsutego samochodu		
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".		
		Realizacja scenariusza testowego		
Krok		Opis	Uwagi	
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka wybieramy odpowiedź: Tak			
2	Czy rozrusznik próbuje uruchomić silnik? Wybór odpowiedzi:			
	Tak			
3	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi: Tak			
4	Wysłanie wsparcia L2			
5	Czy rozwiązano problem? Wybór odpowiedzi: Nie			
6	Czy abonament zgłaszającego pokrywa koszty transportu do serwisu? Wybór odpowiedzi:			
	Nie			

7	Czy zgłaszają odpowiedzi: Nie	cy akceptuje koszty transportu? Wybór	
8	Zamkniecie z	głoszenia ze statusem "closed- Client resigned"	
Oczekiwany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed- Client resigned".		
(Kryteria akceptacji)			
Uzyskany rezultat	Zakończenie workflow. Przypisanie do zgłoszenia statusu "closed- Client resigned".		
Wynik testu / status P			

Tabela 66. Scenariusz testu: *Workflow* "awaria samochodu" - kontrolki się świecą, naprawa w serwisie. Źródło: Opracowanie własne.

Nazwa scenariusza		<i>Workflow</i> "awaria samochodu" - brak abonamentu na transport zepsutego samochodu, lecz klient akceptuje koszty	
Warunki początkowe		Użytkownik ma utworzony dostęp do systemu. Tworzy odpowiednie zgłoszenie, w którym wybiera <i>workflow</i> "Awaria samochodu". Rozpoczęcie testu następuje od pierwszego kroku <i>workflow</i> , na którym wyświetlane jest pytanie "Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka?".	
	-	Realizacja scenariusza testowego	
Krok		Opis	Uwagi
1	Czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka wybieramy odpowiedź: Tak		
2	Czy rozrusznik próbuje uruchomić silnik? Wybór odpowiedzi: Tak		
3	Czy zgłaszający ma abonament na wsparcie bezpośrednie? Wybór odpowiedzi: Tak		
4	Wysłanie ws	parcia L2	
5	Czy rozwiązano problem? Wybór odpowiedzi: Nie		
6	Czy abonament zgłaszającego pokrywa koszty transportu do serwisu? Wybór odpowiedzi:		
	Nie		
7	7 Czy zgłaszający akceptuje koszty transportu? Wybór odpowiedzi: Tak		
8	Konsultant L2 zabiera pojazd do serwisu		
9	Zamkniecie zgłoszenia ze statusem " <i>closed – Repaired by</i>		

	service"			
Oczekiwany rezultat	Zakończenie "closed – Rep	workflow. Przypisanie do zgłosz paired by service".	enia statusu	
(Kryteria akceptacji)				
Uzyskany rezultat	Zakończenie "closed – Rep	workflow. Przypisanie do zgłosz paired by service".	enia statusu	
Wynik testu	ı / status	Р		

6.9. Testowanie użyteczności

6.9.1. Definicja

Termin "*usability*" w języku polskim jest tłumaczony na dwa sposoby: "użyteczność", oraz "funkcjonalność". Wybór tłumaczenia został oparty o normę ISO 9126, w której następuje rozróżnienie tych dwóch terminów. Angielskie "*functionality*" określa ona jako dostępną liczbę funkcji w aplikacji, bądź serwisie internetowym, zapewniających zaspokojenie potrzeb jej użytkowników. Natomiast pojęcie "*usability*" zdefiniowano jako łatwość obsługi i łatwość zrozumienia tych funkcjonalności. W związku z powyższym, jak i tym że takie rozróżnienie jest najczęściej stosowane w branży informatycznej, na potrzeby tej pracy przyjęto używać tłumaczenia tego terminu jako "użyteczność".

Samo pojęcie *usability* jest pojęciem na tyle rozległym, że ciężko jest zdefiniować go jednoznacznie i wyczerpująco. W normie ISO 9241 stwierdzono, że użyteczność jest "*miarą wydajności, efektywności i satysfakcji użytkownika z jaką dany produkt może być używany do osiągnięcia określonych celów przez jego użytkowników"[58]. Co za tym idzie <i>usability* ocenia, czy dana aplikacja umożliwia proste i przyjemne korzystanie z jej funkcjonalności. Warto też wspomnieć o pięciu najważniejszych elementach tego terminu, wskazanych przez Jakoba Nielsena – jednego z najbardziej znanych i cenionych specjalistów od *usability*, który tematyką tą zajmuję się już od lat dziewięćdziesiątych[59]:

- "Przydatność praktyczna. Czy system wykonuje zadania, które odpowiadają potrzebom użytkownika?
- Łatwość nauki i obsługi. Jak szybko można nauczyć się obsługi systemu? Czy dla większości osób obsługa systemu jest wystarczająca łatwa?
- Skuteczność. Czy system zapewni wynik zadania w takiej postaci, jak oczekuje tego użytkownik?
- Efektywność. Czy pożądany wynik osiąga się przy umiarkowanym wysiłku ze strony użytkownika?
- Zadowolenie. Czy użytkownik lubi pracować z systemem i czy rekomendowałby go innym?"[60]

Reasumując aplikacja zgodna z *usability* ma umożliwiać łatwy, wydajny, intuicyjny i efektywny dostęp do jej funkcjonalności. W dalszej części odpowiedziano na pytanie, czy nasza aplikacja pozwala na łatwe i szybkie osiągniecie założonych celów przez jej użytkownika.

6.9.2. Metody testowania użyteczności

Badając *usability* najczęściej stosuje się dwie metody badawcze. Jedną z nich są testy z użytkownikami, w których uczestniczy moderator. Podczas testu, w którym zaproszeni użytkownicy wykonują zlecone im zadania, moderator stara się wyłapać, jakie elementy sprawiają im największą trudność i to głównie od jego obserwacji zależy efektywność tej metody. Zwolennikiem tej metody jest Steve Krug (por.[61]) - ekspert *usability* - który twierdzi, że właściwie przeprowadzone testy, prowadzone już od początku tworzenia aplikacji pozwalają na stworzenie produktu zgodnego z filozofią użyteczności [62]. Podobnego zdania jest wspomniany już wcześniej Jakob Nielsen, który dodatkowo w swoich badaniach określił, że grupa już jedynie 5 uczestników pozwala na wykrycie około 85% błędów/trudności w tworzonej aplikacji.[63]
Niestety pomimo wysokiej skuteczności tej metody, nie zastosowano jej w tej pracy gdyż przygotowanie rzetelnego harmonogramu testów, oraz jego realizacja wymaga dużej dozy doświadczenia, jak i nakładów finansowych.

Drugim sposobem oceny użyteczności jest tak zwana analiza ekspercka. Polega ona na tym, że ekspert ocenia każdy z poszczególnych elementów aplikacji pod kątem jego użyteczności. Zdecydowanym plusem tej metody jest możliwość oceny serwisu internetowego, bądź aplikacji przez jedną osobę. Niestety wiąże się to z faktem, że cześć oceny ma subiektywny wydźwięk. Jednakże, aby ocena była jak najbardziej miarodajna zostało wybranych kilka zasad *usability*, a następnie na ich podstawie oceniono badane elementy w skali 1-6 (gdzie jeden jest najniższą oceną, a sześć najwyższą). Na podstawie sumy z poszczególnych elementów określono czy aplikacją jest zgodna z zasadami *usability*, czy też posiada jakieś rażące błędy ograniczające jej użyteczność.

6.9.3. Elementy badania

Nasza aplikacja jest aplikacją webową, w związku z tym oceniono ją w znacznym stopniu tak jak gdyby była typową stroną WWW. Badaniu zostały poddane poniższe aspekty:

Układ strony – istotny element każdej witryny internetowej, gdyż to on bezpośrednio odpowiada za rozłożenie wszystkich części składowych strony. Oceniając układ strony zwrócono uwagę między innym na to, czy układ wpisuje się w powszechnie obowiązujące kanony, czy sekcja nawigacyjna jest łatwo rozróżnialna, czy też ergonomię strony.

Nawigacja – w tym podpunkcie zostały sprawdzone takie elementy jak menu aplikacji, czy też zastosowanie elementów nawigacji takich jak np. *breadcrumbs* (okruszki chleba – czyli ścieżka nawigowania, umożliwiająca użytkownikowi zorientowanie się w którym miejscu serwisu obecnie się znajduje) [64].

Czytelność i elementy *layoutu* – jest to element o tyle istotny, że w odbiorze przedstawianych w aplikacji treści istotną role odgrywa nie tylko sama wartość merytoryczna, ale też ważnym jest sposób w jaki treści te są przedstawione. Dlatego też w tym podpunkcie został oceniony dobór czcionek, rozmieszczenie tekstów, oraz dobór kolorów tła i czcionki (odpowiedni ich kontrast).

Żeby maksymalnie wyeliminować subiektywny aspekt oceny, każdy element strony został zestawiony ze standardami opisanymi w książkach: Marek Kasperski i Anna Boguska-Torbicz *Projektowanie stron WWW* [64] i *Steve Krug Nie każ mi myśleć!* [62].

6.9.4. Analiza ekspercka – badanie, część właściwa

6.9.4.1. Układ strony

W aplikacji do przedstawienia treści wykorzystano układ dwukolumnowy asymetryczny z dodatkowym menu lewostronnym, oraz charakterystyczną dla większości witryn orientacje pionową, co oznacza że informacje mają strukturę ciągnącą się od góry do dołu. Takie rozmieszczenie treści jest zgodne z obecnie obowiązującymi kanonami [64], dzięki czemu przyszyły odbiorca (opierając się o swoje doświadczenia z innymi aplikacjami/stronami webowymi) nie powinien mieć większych problemów z poruszaniem się w obrębie witryny. Równie istotnym jest zachowanie spójności pomiędzy stroną główną a podstronami, co też zostało zapewnione dzięki zastosowaniu jednakowego układu wszystkich stron aplikacji [62].

Service desk 🐗 Strona główna 🛛 O firmie Kor	ntakt 💿 🥜 Rejestracja z	gloszenia Panel admini:	stracyjny Moje konto - ME	NU GŁ(OWNE	Hasio	Zaloguj
		_		_	_	_	
				<u> </u>			
Dodaj firmę	Wyszukiwan	ie zgłoszenia					
Wyszukiwanie firmy							
LOKALIZAGJA	Nr. umowy						
Dodaj Lokalizację							
Dodaj lokalizację z pliku							
Wyszukiwanie Lokalizacji							Darksteinen
UMOWA							Lodaj nowe
Wyszukiwanie umów	Nr. umowy	Status zoloszenia	Liczba dni do zamkniecia zgłoszenia	Priorytet			
GRUPY	20140						
Dodaj	33119						60 60 60
Wyszukaj							
OSOBY							
Dodaj osobę							
Wyszkiwanie osoby							
PRACOWNECY							
Pracownicy							
Stanowiska							
PRODUKTY							
Produkt - urządzenie							
Produkt - usługa							
PRODUCENT							
Dodaj							

Rysunek 74. Układ strony z wyróżnionymi elementami składowymi (układ dwukolumnowy asymetryczny z dodatkowym menu lewostronnym). Źródło: Opracowanie własne.

Kolejnym pozytywnym elementem układu strony zastosowanym w tej aplikacji jest logo. Zostało ono umieszczone w lewym górnym rogu, a więc w miejscu w którym szuka go większość użytkowników serwisów internetowych [62]. Takie rozmieszczenie pozwala na łatwe zorientowanie się, że ciągle pozostajemy w obrębie naszej aplikacji. Początkowo logo od strony graficznej zostało stworzone w podobny sposób, co reszta elementów menu głównego, w wyniku czego nie wyróżniało się ono w sposób należyty. Zostało to jednak poprawione, a efekt zmiany przedstawiają zrzuty umieszczone poniżej:



Rysunek 75. Wycinek menu głównego, zawierający słabo wyróżniające się logo. Źródło: Opracowanie własne.



Rysunek 76. Wycinek menu głównego z poprawionym logo. Źródło: Opracowanie własne.

Analizując ergonomię aplikacji skupiono się na tym czy zagospodarowano w sposób efektywny przestrzeń aplikacji, oraz na tym czy rozkład przycisków jest prawidłowy. Większość podstron w sposób właściwy wykorzystuje dostępny obszar, aczkolwiek już samo rozmieszczenie przycisków nie należy do wzorcowych. Za przykład posłuży poniższy zrzut ekranowy przedstawiający proces dodawania nowej firmy na którym nie widać przycisków "Dodaj" oraz "Anuluj", gdyż zostały one umieszczone w prawym dolnym rogu aplikacji, który nie jest dostępny bez przewinięcia strony w dół.

Taki układ przycisków sterujących wprowadza pewną dezorientację użytkownika i należy wprowadzić poprawę w tym aspekcie aplikacji. Mimo, że problem ten nie występuje w przypadku formularza zawierającego mniejszą liczbę pól (przyciski wtedy mieszczą się na powierzchni aktualnie wyświetlanej), w celu zachowania spójności aplikacji należy umieścić wszystkie przyciski w jednym widocznym miejscu na przykład prawym górnym rogu formularza.

C C localhost:6937/firmlocation/create				100	0
vice desk # Strona glówna C	firmie Kontakt 🖋 Rejestracja zgłoszenia Panel administrac	yjny Moje konto •	Login	Haslo	Zalogu
	daj firme				
IRMA	Dodawanie firmy				
lodaj firmę					
Vyszukiwanie firmy	Dane ogólne	l okalizacie			
DKALIZACJA Jodaj Lokalizacio	Nazwa	Dadatkowa lakalizaria			
	IND. WO				
	Dava kustatisma	Lokanzacja 15			
AWOMI	Dane kontaktowe	Adres lokalizacji			
Vyszukiwanie umów		Kod pocztowy			
RUPY	NIP				
	REGON	Miasto			
eopy			🖂 lest akkuna lekalizasia		
lodaj osobe			 Jest growną iokaiizacją 		
Vyszkiwanie osoby		Telefon			
RACOWNICY					
		Email			
tanowiska		Opis			
RODUKTY					
RODUCENT					

Rysunek 77. Dodawanie firmy i niewidoczne przyciski. Źródło: Opracowanie własne.

Reasumując, układ treści w aplikacji można ocenić pozytywnie. Zastosowano rozłożenie elementów w sposób zgodny z obecnie obowiązującymi kanonami, co powinno pozwolić na łatwiejsze wdrażanie przyszłych użytkowników systemu. Również jeśli chodzi o zagospodarowanie wolnej przestrzeni, to wydaję się on być poprawne. Po wyeliminowaniu błędów związanych z rozmieszczeniem przycisków, układ strony został oceniony na 5 punktów (na 6 możliwych).

6.9.4.2. Nawigacja

Rolą nawigacji w aplikacjach (czy też witrynach internetowych), podzielonych na wiele podstron jest umożliwienie użytkownikowi zorientowanie się w którym jej miejscu obecnie się znajduje. Dobrze zaprojektowana umożliwi jej odbiorcy łatwą i szybką orientację w aplikacji, dzięki czemu będzie mógł on korzystać z aplikacji w sposób efektywny i przyjazny. Cały ten podrozdział będzie oparty o informacje zawarte w książce Steve Kruga[62].

Główną funkcjonalność nawigacji w analizowanej aplikacji zapewnia menu boczne umieszczone z jej lewej strony. Zostało ono podzielone na sekcje tematyczne, które pozwalają w prosty i szybki sposób przełączać się pomiędzy żądanymi treściami. W tym miejscu warto by się zastanowić czy całe menu powinno być rozwinięte tak jak ma to miejsce obecnie, czy może lepiej zastosować widok kategorii, które dopiero po rozwinięciu pokazywałyby szczegółowe opcje danej sekcji? Niestety nie ma jednej właściwej odpowiedzi na to pytanie, gdyż wszystko zależy od charakteru aplikacji, oraz od tego ile elementów faktycznie zawiera analizowane menu. W tym konkretnym przypadku liczba opcji jest niewielka, dzięki czemu można pozwolić sobie na to by było ono całkowicie rozwinięte. Zyskujemy dzięki temu bardzo szybki dostęp do szukanych treści, a niewielka ilość opcji nie utrudnia ich odnalezienia. Należy jednak pamiętać o tym, że w raz z rozbudowywaniem menu, kwestie tą będzie trzeba ponownie przeanalizować tak aby struktura nawigacji pozostawała ciągle użyteczną. Ostatnim aspektem badanego elementu jest brak wyróżnienia aktualnej pozycji w menu (w pewnym stopniu rekompensuje to zastosowanie breadcrumbs, ale o tym w dalszej części). Warto rozważyć wprowadzenie jakiegoś graficznego zaznaczenia miejsca, w którym obecnie użytkownik się znajduje, co sprawi że aplikacja będzie łatwiejsza w użytkowaniu i bardziej transparentna.

Następnym elementem nawigacji jest menu główne zawierające jedynie najbardziej przydatne funkcje, oraz Logo które jednoznacznie identyfikuje aplikację i jest odnośnikiem do strony głównej. W menu głównym w przeciwieństwie do menu bocznego nie zabrakło graficznego wyróżnienia aktualnej pozycji w której znajduję się użytkownik. Całość sprawie wrażenie dobrze zaprojektowanej a jedynym mankamentem jest niepotrzebny dodatkowy odnośnik do strony głównej, gdyż to samo zadanie pełni logo.



Rysunek 78. Niepotrzebny dodatkowy przycisk do strony głównej. Źródło: Opracowanie własne.

W aplikacji zastosowano również dodatkowy element nawigacji – *breadcrumbs*, pokazujący w którym miejscu aplikacji obecnie się znajdujemy. Poszczególne poziomy hierarchii zostały oddzielone od siebie separatorem '/' a sama aktualna pozycja została dodatkowo wyróżniona innym kolorem, co widać na poniższym zrzucie ekranowy.

Strona główna / Panel Administracyjny - firmy / Dodaj firme

Rysunek 79. Przykład *breadcrumbs* zastosowanych w aplikacji "Service desk". Źródło: Opracowanie własne.

Z racji charakteru tworzonego programu, nie zastosowano jednej głównej wyszukiwarki umieszczonej na przykład w menu głównym. Zastąpiono ją natomiast kilkoma o zawężonym spectrum poszukiwań, umożliwiającym wyszukiwanie chociażby umów, pod różnymi kryteriami (zrzut poniżej). Plusem zaprojektowanych wyszukiwarek jest możliwość zatwierdzenia wyszukiwania zarówno poprzez przycisk, jak i bezpośrednio z klawiatury wciskając klawisz "Enter".

Wyszukiwar	nie umów		
Nr umowy	Data zawarcia umowy	Typ usługi	
Nazwa firmy	Data rozwiązania umowy	Aktywna	
			Misorulaci

Rysunek 80. Przykładowa wyszukiwarka. Źródło: Opracowanie własne.

Biorąc pod uwagę całość, nawigacja została uznana za zgodną z ideą użyteczności i otrzymała ocenę 5pkt. Posiada ona właściwie zaprojektowane menu boczne i główne, również wyszukiwarki spełniają swoje zadania a dodatkowo dodane *breadcrumbs* pozwalają na jeszcze łatwiejszą orientacje, gdzie aktualnie się znajdujemy.

6.9.4.3. Czytelność

W aplikacji zostały wykorzystane czcionki bezszeryfowe (takie jak *Helvetica*, czy *Arial*), co oznacza brak zdobników poprawiających wygląd/dekoracyjność znaków. Jest to właściwy wybór, gdyż czcionki z rodziny bezszeryfowej znaczniej lepiej prezentują się na ekranach monitorów, niż czcionki szeryfowe które lepiej wypadają w przypadku ich wydruku[64].

Ważnym elementem wpływającym na czytelność wyświetlanych treści jest zastosowany kontrast pomiędzy kolorem czcionek a ich tłem. Do tego badania wykorzystany został program *Colour*

Contrast Analyser[65] zgodny z standardami W3C⁴, co powinno pozwolić na obiektywną ocenę. Za właściwe skontrastowanie tych elementów, program przyjmuje takie, które uzyskały wynik na poziomie 5,00 wzwyż[66]. Poniżej wyniki z wybranych elementów:

Tabela 67. Zestawienie wyników badania kontrastu pomiędzy kolorem tła a kolorem czcionki. Źródło: Opracowanie własne.

Element Badany(opis)	Kolor czcionki	Kolor tła	Element badany (fragment w formie zrzutu ekranowego)	Wynik
Menu Główne	#9999999	#2E0854	O firmie	5,76
Menu Główne zaznaczenie	#FFFFFF	#111111	Moje konto 👻	18,88
Menu boczne nagłówek	#9999999	#F5F5F5	LOKALIZACJA	2,61
Menu boczne	#FFFFFF	#AECC23	Dodaj firmę	1,83
Breadcrumbs	#CED530	#FFFFFF	Strona główna / Panel	1,59
			Dodawanie firmy	
Treści właściwa	#333333	#FFFFFF	Dane ogólne	12,63
			Nazwa PJWST	

Jak widać na powyższym zestawieniu, dobór kolorów nie zawsze jest odpowiedni. Cześć elementów stanowiących trzon nawigacji w aplikacji ma nieco zbyt mały kontrast pomiędzy czcionkami a ich tłem. W ramach wyeliminowania tego problemu zaleca się zwiększenie owego kontrastu na przykład poprzez przyciemnienie tła.

Podsumowując aspekt czytelności, można stwierdzić że jest on poprawny. Zostały zastosowane odpowiednie typy czcionek, które wygodnie czyta się na monitorze, a główne treści posiadają odpowiedni kontrast. Całościowo przyznano 4 punkty, gdyż prezentowane treści w subiektywnej ocenie wydają się być należycie czytelne. Dodatkowe punkty zostaną przyznane po poprawieniu kontrastu w elementach nawigacji.

6.9.4.4. Ocena Końcowa

Użyteczność ze względu na swój skomplikowany charakter bywa trudna do zmierzenia, a sama ocena często bywa subiektywną. Oznacza to, że nie zawsze można łatwo i jednoznaczniej ocenić, czy akurat takie rozwiązanie będzie lepsze od innych. Oceniając badaną aplikacje starano się odnosić w możliwie jak największym stopniu do obecnie obowiązujących standardów i w zestawieniu z nimi badany interfejs graficzny wypadł poprawnie. W trzech testowanych elementach przyznano 14

⁴ World Wide Web Consortium, w skrócie W3C – organizacja, zajmująca się kreowaniem standardów związanych z technologiami WWW.

punktów na 18 możliwych, gdyż w opinii badającego aplikacja w kwestii *usability* spełnia swoją rolę poprzez zapewnienie łatwości w jej poznaniu i obsłudze (realizowanej chociażby przez stosowanie powszechnie obowiązujących układów treści), czy też umożliwienie prostej realizacji zadań (między innym dzięki zastosowaniu właściwej nawigacji). Również w aspekcie mocno subiektywny, którym jest zadowolenie z pracy z danym oprogramowaniem, badający uznaje je za właściwie użyteczne [60].

7. Przykłady zastosowań

W poniższym rozdziale zaprezentowano owoc prac projektowych oraz przedstawiono jego zastosowanie na praktycznych przykładach.

7.1. Interfejs

Aplikacja webowa została zaprojektowana z myślą o użytkownikach, których stopień zaawansowania pracy z komputerem nie musi być na wysokim poziomie, aby mogli swobodnie poruszać się po aplikacji. Menu górne służy do nawigacji po panelu głównym. Stąd użytkownik może wybrać akcję jaką chce wykonać (rejestracja nowego zgłoszenia, przejście do panelu administracyjnego – dostępne dla administratora aplikacji, sprawdzenie przypisanych zgłoszeń, przegląd historii zgłoszeń, czy edytowanie danych).

Po wybraniu odpowiedniej akcji poniżej menu głównego znajduje się link z lokalizacją – ma to na celu lepszą orientację w którym miejscu w danej chwili jesteśmy jak i szybki powrót do poprzednich ekranów. Po lewej stronie wyświetla się menu boczne z akcjami jakie możemy wybrać w ramach danej grupy.

Service desk						Logir	1	Hasło	Zaloguj
	🕷 Strona główna			🖋 Rejestracja zgłoszenia		rjny Moje konto 🗸			
Strona główna / Panel Ad	dministracyjny - firmy	/ Dodaj firm	e						
FIRMA		Do	odaw	anie firmy	/				
Wyszukiwanie firmy									
LOKALIZACJA			Dane ogól	ine	L	.okalizacje			
Dodaj Lokalizację			Naz	wa		Dodatkowa lokalizacja			
Dodaj lokalizację z p Wyszukiwanie Lokal	oliku lizacji					okalizacja 1 »	-		
UMOWA		D	ane kontakto	we		Adres lokalizacj			
Wyszukiwanie umów	r					Kod pocztowy			
GRUPY									
Wyszukai			١	NIP		Miasto			
OSOBY Dodaj osobę			REG	И			📝 Jest główną lok	alizacją	
Wyszkiwanie osoby						Telefor			

Rysunek 81. Widok aplikacji - menu główne z widocznym panelem do nawigacji i menu bocznym. Źródło: Opracowanie własne

Jak widać formularz do dodawania nowej firmy jest czytelny: zawiera duży nagłówek, który zapobiega powstawaniu błędów typu wpisywanie danych do nieodpowiedniego formularza, nagłówki oddzielające sekcje są napisane pogrubioną czcionką, pola do wypełnienia danych zostały odpowiednio podpisane i oddzielone są od siebie ramkami, z zachowaniem znacznej odległości. Ma to na celu ułatwienie pracy z aplikacją i zmniejszenie błędów przy wpisywaniu danych do niewłaściwych pól.

Dodatkowo przyciski zatwierdzające daną czynność są duże i trójwymiarowe co pozwala na szybszą lokalizację przycisku oraz uniknięcie błędu – takich jak przypadkowe zatwierdzenie akcji. Przyciski, które anulują akcję są zawsze z tej samej strony umieszczone względem przycisku zatwierdzającego (po prawej stronie) oraz posiadają inny kolor (szary). Odróżnienie ich od siebie ma istotny wpływ na pracę. Chcemy, żeby użytkownik nie wybierał niewłaściwego przycisku przez pomyłkę. Dodatkowo przyciski te są umieszczone na jasnoszarym pasku na dole formularza. W ten sposób jest zaznaczony jego koniec.

Dodai	Anulu

Rysunek 82. Przyciski na formularzu do zatwierdzania i anulowania akcji. Źródło: Opracowanie własne

Wszelkie listy są przedstawione za pomocą tabelki – ułatwia to przeglądanie informacji w niej zawartej. Przy każdym wierszy, na samym końcu znajdują się ikonki symbolizujące akcje jakie można na danym rekordzie wykonać (edycja, podgląd, kasowanie).

v ice desk				Login	Hasło	Zaloguj
🖨 Strona główna		🖋 Rejestracja zgłoszenia		Moje konto 👻		
na główna > Osoby > Lista osob						
						odai nowa
FIRMA						sodaj noną
Dodaj firmę	Imię	Nazwisko	Telefon	E-mail		
wyszukiwanie firmy	Jan	Dawes	211388994	jadawes@gmail.com	C (Q x
Lokalizacja Dodaj Lokalizację	Mateusz	Curtis	257129089	macurtis@gmail.com	80	Q x
Dodaj lokalizację z pliku	Bartek	Truman	282850174	batruman@gmail.com	80	Q x
Wyszukiwanie Lokalizacji	Kacper	Barkley	217582269	kabarkley@gmail.com	ß	Q x
UMOWA	Michal	Grant	253313254	migrant@gmail.com	ß	Q×
GRUPY	Szymon	Washington	288044359	szwashington@gmail.com	ו ו <i>ו</i> נו	Q×
Dodaj	Antoni	Jones	214775444	aniones@gmail.com	C (Q ×
Wyszukaj	Filip	Adams	240506530	fiadams@gmail.com	20	0 ¥
O SOBY	Timp	Addins	245000005	nadams@gmail.com		~ ~
Dodaj osobę	Piotr	Martin	285248624	pimartin@gmail.com	6	Q×
Wyszkiwanie osoby	Maciej	Maxwell	210979719	mamaxwell@gmail.com	6	Q X
PRACOWNICY	Aleksander	Farewell	246700804	alfarewell@gmail.com	80	Q x

Rysunek 83. Lista osób – dodawanie, wejście w szczegóły danego rekordu, edycja i kasowanie. Źródło: Opracowanie własne

Wygląd aplikacji został oparty o *framework Bootstrap*, który został opisany w rozdziale 4.6. Gotowe elementy (menu, przyciski) są dostępne na stronie *http://twitter.github.io/bootstrap/*.

7.2. Workflow w teorii

W niniejszym podrozdziale opisano algorytm działania przepływów dla kilku sytuacji serwisowych, które mogą zdarzyć się naprawdę. Zapoznanie się z poniższymi *workflow* pozwala lepiej zrozumieć jak można wykorzystać aplikację Service Desk i jakie możliwości oferuje.

7.2.1. Legenda

Do przedstawienia workflow zastosowano poniższe oznaczenia:



Początek / koniec workflow.

Proces - opis czynności wykonywanych przez system.

 \bigcirc

Decyzja – od nich zależy dalszy przebieg workflow

7.2.2. Workflow "Weryfikacja Klienta"

Jest to *workflow* uruchamiany przy każdym nowym zgłoszeniu. W jego trakcie następuje sprawdzenie, czy dzwoniąca osoba jest klientem firmy i wybranie odpowiedniego do przyczyny kontaktu przepływu.



Rysunek 84. Workflow "Weryfikacja klienta". Źródło: Opracowanie własne.

Dla każdego zgłoszenia najpierw uruchamiany jest *workflow* "weryfikacja klienta". Konsultant prosi o informacje umożliwiające jednoznaczną identyfikację klienta, takie jak: imię i nazwisko, firmę, adres, numer telefonu itp. W przypadku niepoprawnej identyfikacji zgłoszenie jest zamykane ze statusem "*Closed - Client not verified*". W razie sukcesu konsultant wybiera stosowny typ zgłoszenia i odpowiedni *workflow* dla danego produktu / usługi.

7.2.3. Workflow "Awaria drukarki"

Workflow ten opisuje wszystkie możliwości przebiegu w przypadku awarii drukarki. Od pomocy zdalnej, w trakcie której podawane są pytania pomocnicze, przez wysłanie konsultanta, aż do zabrania urządzenia do serwisu w celu naprawy, bądź wymiany na nowy.

1. Użytkownik systemu otrzymuje telefon od klienta w sprawie awarii drukarki. W pierwszym kroku następuje weryfikacja produktu – klient musi podać firmę, model i nr seryjny. Opcjonalnie można też podać datę zakupu. Na tej podstawie system sprawdza, czy produkt jest uprawniony do naprawy. W przypadku niepowodzenia zamyka zgłoszenie. W przeciwnym razie przechodzi do serii pytań pomocniczych, mających na celu ustalenie rodzaju problemu i jego ewentualnego rozwiązania.

2. Zgłaszający jest proszony o weryfikację czy problem nie leży w braku papieru lub jego zacięciu, zużytym tonerze, czy konieczności ponownego uruchomienia sprzętu. Jeżeli któraś z proponowanych przez system sugestii okaże się właściwym rozwiązaniem, zostaje wysłany email do klienta, a zgłoszenie jest zamykane ze statusem "*Solved*". W przypadku braku właściwego rozwiązania, następuje przejście do kolejnej części *workflow*, mającej na celu udzielenie wsparcia bezpośredniego.

3. W drugiej części *workflow* sprawdzane jest czy klient ma wykupiony abonament na wsparcie bezpośrednie. Jeżeli nie posiada i nie akceptuje wynikających z tego kosztów, zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*". W przypadku aktywnego abonamentu lub akceptacji kosztów system wyszukuje konsultantów z odpowiednimi kompetencjami i informuje ich o dostępnym zgłoszeniu.

4. Kiedy konsultant przyjmie zgłoszenie, jest ono usuwane z kolejki dostępnych zgłoszeń, a jego status ustawiany jest na "*Assigned*". Jeżeli udało się naprawić sprzęt, zgłoszenie jest zamykane, jako "*Closed – Problem Solved*", a do klienta jest wysyłany email z potwierdzeniem. Jeśli jednak do naprawy potrzeba wymiany części urządzenia sprawdzane jest, czy abonament klienta pokrywa transport do serwisu. W sytuacji braku takiej opcji, klient jest proszony o akceptację związanych z tym kosztów, a w przypadku jej braku zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*", a do klienta zostaje wysłany email z tą informacją.

5. Po zabraniu do serwisu system oblicza koszty i na tej podstawie podejmowana jest decyzja o dalszej naprawie, bądź wymianie sprzętu na nowy. Wymiana nastąpi również w przypadku braku części zamiennych. Wtedy konsultant zabiera uszkodzoną maszynę, dostarcza nową i zamyka zgłoszenie w statusie "*Closed – New product provided*".

6. Jeżeli koszty naprawy są akceptowalne a części są dostępne to zostaje zlecona ich wymiana. Następnie jest sprawdzana poprawność działania produktu. Może się zdarzyć sytuacja, w której nadal są problemy z jakąś częścią. W takiej sytuacji następuje ponowne obliczenie kosztów jej wymiany i ponownie podejmowana decyzja odnośnie dalszej naprawy, bądź kupna nowego sprzętu.

7. Jeżeli wszystko działa jak trzeba przedmiot jest zwracany klientowi, a zgłoszenie zamykane ze statusem "*Closed – Repaired by service*".



Rysunek 85. Workflow "Awaria drukarki" część 1 z 2. Źródło: Opracowanie własne.



Rysunek 86. Workflow "Awaria drukarki" część 2 z 2. Źródło: Opracowanie własne.

7.2.4. Workflow "Awaria Monitora"

Workflow ten opisuje wszystkie możliwości przebiegu w przypadku awarii monitora. Od pomocy zdalnej, w trakcie której podawane są pytania pomocnicze, przez wysłanie konsultanta, aż do zabrania wyświetlacza do serwisu w celu naprawy, bądź wymiany na nowy.

1. Użytkownik systemu otrzymuje telefon od klienta w sprawie awarii monitora. W pierwszym kroku następuje weryfikacja produktu – klient musi podać firmę, model i nr seryjny. Opcjonalnie można też podać datę zakupu. Na tej podstawie system sprawdza, czy jest uprawniony do naprawy. W przypadku niepowodzenia zamyka zgłoszenie W przeciwnym razie przechodzi do serii pytań pomocniczych mających na celu ustalenia rodzaju problemu i jego ewentualnego rozwiązania.

2. Zgłaszający jest proszony o weryfikację czy problem nie leży w braku podłączenia do prądu, złych ustawieniach, bądź awarii komputera, do którego jest podłączony. Jeżeli któraś z proponowanych przez system sugestii okaże się właściwym rozwiązaniem, zostaje wysłany email do klienta, a zgłoszenie jest zamykane ze statusem "*Solved*". W przypadku braku właściwego rozwiązania następuje przejście do kolejnej części *workflow* mającej na celu udzielenie wsparcia bezpośredniego.

3. W drugiej części *workflow*, sprawdzane jest czy klient posiada abonament na wsparcie bezpośrednie. Jeżeli nie i nie akceptuje wynikających z tego kosztów, zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*". W przypadku aktywnego abonamentu, lub akceptacji kosztów system wyszukuje konsultantów z odpowiednimi kompetencjami o dostępnym zgłoszeniu.

4. Kiedy konsultant przyjmie zgłoszenie, jest ono usuwane z kolejki dostępnych zgłoszeń, a jego status ustawiany jest na "*Assigned*".Jeżeli udało się naprawić sprzęt zgłoszenie jest zamykane, jako "*Closed – Problem Solved*", a do klienta jest wysyłany email z informacją na ten temat. Jeśli jednak do naprawy potrzeba wymiany części monitora sprawdzane jest, czy abonament klienta pokrywa transport do serwisu. W sytuacji braku takiej opcji, klient jest proszony o akceptację związanych z tym kosztów. W przypadku jej braku zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*", a do klienta zostaje wysłany email z tą informacją.

5. Po zabraniu do serwisu system oblicza koszt naprawy i na tej podstawie podejmowana jest decyzja o dalszej naprawie, bądź wymianie sprzętu na nowy. Wymiana nastąpi również w przypadku braku części zamiennych. Wtedy konsultant zabiera uszkodzony wyświetlacz, dostarcza nowy i zamyka zgłoszenie w statusie "*Closed – New product provided*".

6. Jeżeli koszty naprawy są akceptowalne, a części są dostępne to zostaje zlecona ich wymiana, a następnie jest sprawdzana poprawność działania produktu. Może się zdarzyć, że nawet po naprawie, nadal są problemy z jakąś częścią. W takiej sytuacji następuje ponowne obliczenie kosztów naprawy i ponownie podejmowana jest decyzja odnośnie dalszej naprawy, bądź kupna nowego sprzętu.

7. Jeżeli wszystko działa jak trzeba monitor jest zwracany klientowi, a zgłoszenie zamykane ze statusem "*Closed – Repaired by service*".



Rysunek 87. Workflow "Awaria monitora" część 1 z 2. Źródło: Opracowanie własne.



Rysunek 88. Workflow "Awaria monitora" część 2 z 2. Źródło: Opracowanie własne.

7.2.5. Workflow "Awaria Komputera"

Poniżej przedstawiono przykładowy *workflow* opracowany w celu zaprezentowania funkcjonalności *switch*. Opisuje on przypadek zepsutego komputera i w zależności od wybranego systemu operacyjnego przechodzi inną ścieżkę przebiegu.



Rysunek 89. Workflow "Awaria komputera". Źródło: Opracowanie własne.

1. Po pomyślnej weryfikacji produktu konsultantowi zostaje wyświetlony formularz, w którym musi określić datę zakupu sprzętu i, przez zaznaczenie *checkboxa*, wybrać czy na produkt została wykupiona dodatkowa gwarancja. Następną opcją wyświetlaną konsultantowi jest *switch*, za pomocą którego wybierany jest system operacyjny. Opcja ta pozwala na pójście różnymi ścieżkami w zależności od dokonanego wyboru.

2. W przypadku systemu Mac OS wyświetlane są dane, które mają być wysłane do autoryzowanego serwisu. Konsultant zbiera je od zgłaszającego, weryfikuje, a następnie zatwierdza ich przesłanie. W takim przypadku następuje zakończenie tego *workflow*.

3. Dla systemu Linux od razu wysyłany jest konsultant drugiej linii wsparcia (L2). Jeżeli nie uda się rozwiązać problemu poprzez rekonfigurację oprogramowania, klient musi zdecydować czy zgadza się na ponowną instalację systemu. W przypadku negatywnej odpowiedzi *workflow* się kończy, jeżeli jednak klient się zgodzi, to konsultant L2 spróbuje naprawić usterkę w ten sposób. Po pomyślnej

reinstalacji następuje weryfikacja poprawności działania. Gdy problem jest rozwiązany workflow się zamyka.

4. Ostatnim systemem do wyboru jest Windows, w którego przypadku przy usterce zalecany jest restart komputera. Jeżeli to pomoże, to *workflow* zostaje zakończony. W przeciwnym przypadku wysyłany jest konsultant L2 i ścieżka postępowania jest taka sama jak w przypadku systemu Linux.

7.2.6. Workflow "Awaria Samochodu"

Workflow ten opisuje wszystkie możliwości przebiegu w przypadku awarii samochodu. Od pomocy zdalnej, w trakcie której podawane są pytania pomocnicze, przez wysłanie konsultanta, aż do zabrania pojazdu do serwisu w celu naprawy, bądź wymiany na nowy.

1. Użytkownik systemu otrzymuje telefon od klienta w sprawie awarii samochodu. W pierwszym kroku następuje weryfikacja produktu – klient musi podać numer rejestracyjny lub numer VIN pojazdu. Na tej podstawie system sprawdza, czy auto jest uprawnione do naprawy. W przypadku nieudanej weryfikacji zamyka zgłoszenie. W przeciwnym razie przechodzi do serii pytań pomocniczych, mających na celu ustalenie rodzaju problemu i jego ewentualnego rozwiązania.

2. Zgłaszający jest proszony o sprawdzenie czy świecą się kontrolki na desce rozdzielczej po przekręceniu kluczyka - jeśli tak, to czy rozrusznik próbuje uruchomić silnik. Jeżeli w trakcie rozmowy wstępnej uda się rozwiązać problem, system automatycznie wyśle email do klienta, a zgłoszenie jest zamykane ze statusem "*Solved*". W przypadku braku właściwego rozwiązania, następuje przejście do kolejnej części *workflow*, mającej na celu udzielenie wsparcia bezpośredniego.

3. W tej części *workflow* sprawdzane jest czy klient ma wykupiony abonament na wsparcie bezpośrednie. Jeżeli nie i nie akceptuje wynikających z tego kosztów, zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*". W przypadku aktywnego abonamentu lub akceptacji kosztów system wyszukuje konsultantów z odpowiednimi kompetencjami i informuje ich o dostępnym zgłoszeniu poprzez email.

4. Kiedy jeden z konsultantów drugiej linii wsparcia przyjmie zgłoszenie, jest ono usuwane z kolejki dostępnych zgłoszeń, a jego status jest ustawiany na "*Assigned*". Jeżeli udało się naprawić samochód, zgłoszenie jest zamykane ze statusem "*Closed – Problem Solved*", a do klienta jest wysyłany email z potwierdzeniem.

5. Jeśli jednak do naprawy potrzeba wymiany części pojazdu sprawdzane jest, czy abonament klienta pokrywa transport do serwisu. W sytuacji braku takiej opcji, klient jest proszony o akceptację związanych z tym kosztów. W przypadku braku zgody zgłoszenie jest zamykane ze statusem "*Closed – Client Resigned*", a do klienta zostaje wysłany email z potwierdzeniem tego faktu.

6. Po zabraniu pojazdu do serwisu i wykonanej naprawie, auto jest zwracane klientowi, a zgłoszenie zamykane ze statusem "*Closed – Repaired by service*".



Rysunek 90. Workflow "Awaria samochodu" część 1 z 2. Źródło: Opracowanie własne.



Rysunek 91. Workflow "Awaria samochodu" część 2 z 2. Źródło: Opracowanie własne.

7.3. *Workflow* w praktyce

W niniejszym rozdziale zaprezentowano aplikację administracyjną PJWSTK Workflow Designer, która jest jednym z owoców tego projektu. Narzędzie to pozwala na tworzenie i edycję przepływów, które później mogą zostać wykorzystane w aplikacji klienckiej.



Rysunek 92. Główne okno aplikacji PJWSTK Workflow Designer. Źródło: Opracowanie własne.

Rysunek 92 przedstawia główne okno aplikacji administratora. Po lewej stronie znajduje się panel z aktywnościami, na podstawie których Administrator buduje przepływ, wykorzystywany później w aplikacji klienckiej.

PJWSTK Workflow Designer	The second	
<u>Plik Widok Pomoc</u>		
Toolbox 👻 🖡 🗙	DrukarkaNieDrukuje.xaml = + ×	Properties 💌 🖡 🗙
Search	DrukarkaNieDrukuje Exoand All Collapse All	System.Activities.ActivityBuilder
 Kontrolki 		Clear
 If DoWhile While Flowchart FlowCartin FlowGwith <> Konie Formularz Formularz Formularz Formularz Formularz Komunikat 	Invitanta nie drukuje Sprawdž poprawne podłąc: Gry świeća się diody/etran Sprawdž poprawne podłąc: Image: Cry świeća się diody/etran Image: Cry świeća się diody/etran Image: Cry świeća się diody/etran Image: Cry świeća się diody/etran Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Image: Cry drukarka ma załadowa Im	Cerr L Cerr

Rysunek 93. Projektowanie workflow "Drukarka nie drukuje". Źródło: Opracowanie własne.

Na rysunku 93 przedstawiono przykład zbudowanego przepływu dla scenariusza "Drukarka nie drukuje". W centralnym oknie widnieją aktywności, powiązane ścieżkami przepływu. Przepływ determinowany jest przez aktywności *Decision*, które, w zależności od wybranej opcji przez operatora systemu, kierują przepływ dalej. Po prawej stronie znajduje się okno Właściwości. W tym miejscu Administrator ma możliwość konfiguracji poszczególnych aktywności.

swiecą się dlody/ekran	
	Expression Editor
+	Condition (Boolean)
	WorkflowContext.RequestContext.Get(Of String)(RequestContextKeys.CS_ANSWER) = QuestionYesNoCodeActivityKeys.CS_YES_ANSWER
Decision	
ukarka ma załadowai	OK Cancel

Rysunek 94. Przykład wyrażenia dla kontrolki Decision. Źródło: Opracowanie własne.

Wyrażenie na rysunku 94 sprawdza czy odpowiedź, zwrócona z przepływu (czyli wybrana przez użytkownika), jest równa odpowiedzi, zadeklarowanej jako warunek przez Administratora.



Rysunek 95. Przykład wyrażenia dla kontrolki E-mail. Źródło: Opracowanie własne.

Wyrażenie przedstawione na rysunku 95 informuje, że jest to e-mail końcowy, który zostanie wysyłany na zakończenie przepływu.

	Line	
	ActivitySpecificProperties	
	Answer	Odpowiedź na pytanie
	Message	"Załaduj papier do kasety/podajnika drukarki"
	Question	"Czy problem został rozwiązany?"
🕨 📮 Załaduj papier do kasety/p	CommonProperties	
	BookmarkName	"MSG2"
	Misc	
	DisplayName	Załaduj papier do kasety/podajnika drukarki

Rysunek 96. Konfiguracja aktywności "Tak / nie". Źródło: Opracowanie własne.

Aktywność z rysunku 96 ma za zadanie wyświetlić okno z pytaniem (właściwość *Question*). W tytule okna pojawi się informacja z właściwości *Message*. Użytkownik ma w tym momencie do wyboru zaznaczenie jednej z dwóch opcji: tak lub nie. Po zaznaczeniu odpowiedź wraca do przepływu i na tej podstawie zgłoszenie kierowane jest dalej na odpowiednią ścieżkę.

Ξ	ActivitySpecificProperties	
	FinishInfo	"Koniec. Problem został rozwiązany. Status closed. Repaired by service"
Ξ	CommonProperties	
	BookmarkName	"Finish4"
Ξ	Misc	
	DisplayName	Koniec. Problem został rozwiązany. Status closed.
1		
	📮 Koniec. Problem został	roz

Rysunek 97. Aktywność kończąca przepływ. Źródło: Opracowanie własne.

Na końcu każdej ścieżki znajduje się aktywność Koniec. Jej zadaniem jest wyświetlenie użytkownikowi informacji o statusie zakończenia przepływu. W tym przypadku we właściwości *FinishInfo* zawarto informacje, że zgłoszenie zakończyło się pozytywnie. Taki komunikat właśnie otrzyma użytkownik.

7.4. Podsumowanie

W powyższych przykładach zaprezentowaliśmy następujące funkcjonalności workflow:

- warunki, na podstawie których system podąża odpowiednią ścieżką,
- *switch*, który pozwala wybranie jednej z kilku różnych ścieżek przepływu pracy,
- pętlę, realizowaną poprzez serię warunków.
- statusy, pozwalające łatwo sprawdzić aktualny stan problemu.
- automatycznie wysyłane maile do klientów, dzięki którym klient jest na bieżąco informowany o ważnych zmianach w swoim zgłoszeniu (jego rozwiązaniu, bądź zamknięciu)

Powyższe funkcjonalności pozwalają tworzyć w łatwy sposób nawet skomplikowane przepływy pracy, dzięki czemu firma może zaoszczędzić na kosztach szkoleń i pisaniu długich, nieczytelnych procedur.

8. Podsumowanie i wnioski

Projekt Service Desk okazał się dużym wyzwaniem organizacyjnym. Zademonstrował wszystkie popularne problemy, na jakie może natrafić liczny zespół:

- zróżnicowanie osób,
- zróżnicowanie umiejętności,
- niekompetentne osoby.

Te – jak i inne – problemy były na bieżąco analizowane i adresowane w celu znalezienia rozwiązania oraz zakończenia prac w założonym czasie.

W trakcie całego projektu grupa pracowała jak dobre, prężne przedsiębiorstwo, zajmujące się kompleksową inżynierią oprogramowania. Grupa podeszła do tematu bardzo rzeczowo. Zgodnie z logiką, temat pracy został początkowo rozbity na prostsze elementy, a dopiero na ich podstawie została podjęta decyzja o wyborze technologii. Co ciekawe nie zdecydowano się na język programowania Java czy C++, które były wykładane w toku studiów, a na bardziej komercyjne rozwiązanie. Fachowe podejście do tematu zaowocowało wybraniem doskonałych narzędzi, a tym samym dostarczeniem oprogramowania tak dobrego jak tylko się dało.

Problemy jakie były stawiane przed każdym z członków zespołu sprawiały, że każdy z nich musiał wykazać się elastycznością w aspektach takich jak:

- zarządzanie czasem,
- podział pracy,
- współpraca w zespole,
- logiczne myślenie,
- wielozadaniowość.

Każda z powyższych rzeczy jest istotnym elementem w przyszłej karierze zawodowej inżyniera informatyki.

Jak w każdym zespole, tak i w tym były słabe ogniwa. Sprawiało to, że zarówno kierownicy zespołów jak i kierownik projektu musieli wykazać się elastycznością oraz szybką reakcją na zaistniałe problemy. Każdy słaby punkt należało wzmocnić lub wymienić. Fakt, że ktoś nie sprawdzał się w jednym rodzaju zadań nie znaczy, że należało go skreślić. Takie osoby najczęściej świetnie się sprawdzały na innym etapie prac.

W trakcie projektu wszyscy członkowie zespołu nauczyli się jak można wykorzystać różnice w sposobie myślenia i pracy różnych osób. Każdy zrozumiał, że odmienność można wykorzystać w bardzo pozytywny sposób, trzeba tylko wiedzieć gdzie i jak ją ukierunkować.

Oprogramowanie Service Desk jest prototypem, który pokazuje w jaki sposób można zbudować system, dbający o standard przepływu informacji. Era pełnej informatyzacji w wielu przypadkach niesie więcej chaosu niż ładu, dlatego standaryzacja jest bardzo ważną cechą systemów dla firm serwisowych. Projekt jest skierowany do spółek, obsługujących wielu klientów z różnorodnymi produktami i usługami. W takich firmach każda chwila przestoju, każda zagubiona lub niezidentyfikowana rzecz jest stratą, na którą nikt nie może sobie pozwolić. Technologia *Windows Workflow Foundation* świetnie wpisuje się w ten schemat - przynosi ład, porządek i standard. Pozwala na dynamiczne tworzenie nowych procedur oraz nieustanne dopracowywanie już istniejących w celu zwiększenia efektywności.

Jak już zostało wspomniane, Service Desk jest jedynie prototypem i, jak każdy prototyp, wymaga rozwoju, w celu wypuszczenia na rynek komercyjny. Dalsze prace powinny się skupić na trzech podstawowych elementach:

- interfejsie,
- statystykach,
- mobilności.

Interfejs – jest to temat rzeka, który nie posiada jedynego słusznego rozwiązania. Jednak w przypadku prototypu Service Desk podstawowe elementy, które warto rozwinąć, dotyczą płynniejszego i wygodniejszego poruszania się pomiędzy wieloma listami ogromnej ilości rekordów. Na dodatek przy dużych ilościach zgłoszeń podstawowe wyszukiwanie może okazać się zbyt mało wydajne.

Statystyka – w podsumowaniu kilkakrotnie padły pojęcia związane z wydajnością, stratą i zyskiem. Do prawidłowego monitorowania tych elementów niezbędny jest system, pozwalający na budowanie zaawansowanych statystyk na potrzeby kontrolingu. Każdy *workflow* powinien być regularnie analizowany w celu zidentyfikowania słabych punktów, a docelowo dopracowania przepływów do perfekcji. Ważnym elementem również jest informacja na temat awaryjności produktów czy usług. System do przyjmowania zgłoszeń jest idealnym miejscem na zbieranie tego typu informacji.

Platforma mobilna – jest to funkcjonalność skierowana w głównej mierze do kadry wyższego szczebla i pracowników linii wsparcia w terenie. Kadra wyższego szczebla powinna mieć stały dostęp z dowolnego miejsca w celu sprawowania jak najlepszej kontroli. Powinni móc w dowolnym momencie i miejscu skontrolować elementy za które odpowiadają, aby ich reakcja mogła być jak najszybsza. Z kolei pracownicy terenowi z racji charakteru swojej pracy nie mogą być uzależnieni od komputerów - z racji ich małej poręczności. Platforma mobilna zapewniłaby im dostęp do najważniejszych informacji w każdy momencie.

9. Słownik pojęć

Dostawca usług - używany jest często, jako skrót określający dostawcę usług informatycznych, który świadczy usługi informatyczne odbiorcom zewnętrznym i wewnętrznym.

Incydent – niezaplanowana przerwa działania usługi lub obniżenie jakości usługi informatycznej. Incydentem może być także awaria danego elementu usługi, który jeszcze nie wpłynął na przerwę w działaniu usługi. Przykładem może być awaria jednego węzła w niezawodnościowym klastrze bazodanowym.

Kategoria - używana jest do łączenia razem spraw. Jest to nazwana grupa czegoś, co ma wspólną cechę. Na przykład grupowanie incydentów o podobnym typie, jako kategoria incydentów, czy kategoria kosztów, jako grupa kosztów podobnego typu.

Odbiorca/Klient - zakupuje świadczone usługi od dostawcy. Dla dostawcy usług klienci definiują oraz uzgadniają docelowy poziom świadczonych usług SLA.

Problem - źródło przynajmniej jednego incydentu, którego przyczyna jest nie zawsze znana w chwili jego rejestracji. Dalsza analiza problemu jest realizowana przez proces zarządzania problemami.

Proces biznesowy - należący do organizacji ma na celu ułatwić dostarczenie usługi lub produktu odbiorcy końcowemu. W praktyce wiele procesów bazuje opierając się właśnie o usługi informatyczne. Proces zakupu usługi lub towaru przez detalistę pozwalający mu świadczyć usługę odbiorcy końcowemu może stanowić przykład procesu biznesowego.

Typ zgłoszenia – kategoria, która ma na celu podzielić rodzaje składanych zgłoszeń do centrum obsługi użytkowników Service Desk. Typowe zgłoszenia to: incydent, wniosek o usługę oraz reklamacja.

Usługa informatyczna - jest to usługa świadczona przez dostawcę usług informatycznych, na rzecz jednego lub wielu odbiorców. Na usługę składa się połączenie technologii informatycznych, ludzi i procesów, które wspierają procesy biznesowe danego odbiorcy. Poziom świadczenia usługi dla procesu biznesowego powinien zostać zdefiniowany w umowie *SLA (Service Level Agrement)*.

Wniosek o usługę – jest prośbą składaną przez użytkownika do Service Desk i zarządzaną w ramach procesu realizacji wniosków. Przykładem takiej prośby może być wniosek o zmianę hasła, prośba o poradę czy udzielenie informacji. Często w sytuacji, gdy wymagana jest zmiana, wniosek o usługę powiązany jest z wnioskiem o zmianę (*RFC - request for change*).

Zarządzanie incydentem - oznacza jak najszybsze przewrócenie usługi do normalnej eksploatacji i organicznie niekorzystnego wpływu incydentu.

Zarządzanie zdarzeniem - obejmuje cały okres ich eksploatacji, w ramach procesu można wyróżnić etapy: wystąpienie zdarzenia, wykrycie, filtrowanie, wykonanie czynności, jeśli są konieczne, analiza oraz zamknięcie.

Zdarzenie – dowolne wydarzenie, które może wpływać na proces świadczenia usług informatycznych oraz możliwych odchyleń na usługi.

Zgłoszenie - to rozmowa telefoniczna pomiędzy użytkownikiem, a centrum obsługi użytkowników. W efekcie przeprowadzonej rozmowy konsultant może zarejestrować w systemie incydent lub wniosek o usługę.

Znany problem – jest to problem, dla którego znane jest obejście i przyczyna wystąpienia. Znane problemy mogą być również wykryte przez samych programistów lub dostawców.

10. Bibliografia

Literatura i pozycje internetowe ponumerowane są w kolejności ich występowania w niniejszej pracy.

- [1]. Materiały OGC ITIL® na licencji OGC, http://www.ctpartners.pl/page/111030/item,1079/ITILreg_2011_EDITION.xhtml, 10.04.2013 r.
- [2]. Polski glosariusz ITIL®, wersja 1.0, z dnia 15 grudnia 2011 oparty na angielskim glosariuszu, wersja 1.0, 29.07.11 r.
- [3]. Model procesowy zarządzanie usługami ITIL cz. I, <u>http://pmanager.pl/index.php?option=com_content&view=article&id=229%3Amodel-procesowy-zarzdzanie-usugami-itil-czii&Itemid=59</u>, 10.04.2013 r.
- [4]. Mariusz Matrejek, *Tworzenie katalogu usług*, strona 3 http://www.itlife.pl/index2.php?option=content2&do_pdf=1&id=964, 10.04.2013 r.
- [5]. Mariusz Matrejek, *Tworzenie katalogu usług*, strona 7 <u>http://www.itlife.pl/index2.php?option=content2&do_pdf=1&id=964</u>, 10.04.2013 r.
- [6]. Materiały osiatis, <u>http://itil.osiatis.es/ITIL_course/it_service_management/service_desk/introduction_and_objective</u> <u>s_service_desk/structure_service_desk.php</u>, 10.04.2013 r.
- [7]. Funkcja Service Desk, http://itsm.itlife.pl/content/view/10059/104/, 10.04.2013 r.
- [8]. PinkVERIFY[™] 2011 Assessment Criteria Incident Management, http://www.pinkelephant.com/PinkVerify/SelfAssessment.htm, 10.04.2013 r.
- [9]. PinkVERIFY[™] 2011 Assessment Criteria Request Fulfillment, http://www.pinkelephant.com/PinkVerify/SelfAssessment.htm, 10.04.2013 r.
- [10]. HP ITSM best practices for HP OpenView Service Desk, 5982-4988EN, 03/2004 <u>http://h10130.www1.hp.com/itil/ITSM_Service_Desk.pdf</u>
- [11]. HP Service Desk (SD) software version 4.5 Obsolescence & Migrations, http://support.openview.hp.com/encore/sd4_5.jsp, 15.06.2013 r.
- [12]. Neil Srinivasan, Best Practices for Designing Contact Center Experiences with Oracle RightNow CX Cloud Service, <u>http://www.oracle.com/us/products/applications/best-practices-design-contactcnt-1583702.pdf</u>, 20.05.2013
- [13]. Wersja demo systemu SolveDirect, http://solvedirect.com, 26.05.2013 r.
- [14]. SolveDirect Green Book: Service Workflow Connection Use Cases, Version 5.7.3, data wydania 19.05.2011 r., <u>http://docu-archive.solvedirect.com/docu/upload/2011/7/</u> SD_UseCases_Workflow_Connections_5.7.3-18120544.pdf
- [15]. Archiwum dokumentów SolveDirect, <u>http://docu-archive.solvedirect.com/docu/upload/2009/11/SDSolutionsOverview-17105527.jpg</u>, 26.05.2013 r.
- [16]. SolveDirect Green Book: *Mobile PRE FINAL*, data wydania 28.02.2012 r., <u>http://docu-archive.solvedirect.com/docu/upload/2012/4/SD_Mobile_2012_02_27_mo-03104017.pdf</u>
- [17]. .NET Framework Versions and Dependencies, <u>http://msdn.microsoft.com/en-us/library/bb822049.aspx</u>, 21.05.2013 r.
- [18]. Matthew MacDonald. Beginning ASP.NET 4.5 in C#, Apress, 2012
- [19]. Mary Joe Foley "*What's Microsoft's father of C#'s next trick?*": <u>http://www.zdnet.com/whats-microsofts-father-of-cs-next-trick-7000004226</u>/, 21.05.2013 r.
- [20]. Mary Joe Foley "*Microsoft Big Brains: Anders Hejlsberg*" http://www.zdnet.com/blog/microsoft/microsoft-big-brains-anders-hejlsberg/1771, 21.05.2013 r.
- [21]. http://pl.wikipedia.org/wiki/Plik:C_Sharp_wordmark.svg, 20.05.2013 r.

- [22]. TIOBE Programming Community Index for June 2013 http://www.tiobe.com/content/paperinfo/tpci/index.html, 21.05.2013 r.
- [23]. TIOBE Programming Community Index for June 2013: *Long term trends*, <u>http://www.tiobe.com/content/paperinfo/tpci/images/tpci_trends.png</u>, 12.05.2013 r.
- [24]. .NET Framework, http://msdn.microsoft.com/en-us/vstudio/aa496123.aspx, 26.05.2013 r.
- [25]. Julian Templeman, David Vitter, *Visual Studio .NET: .NET Framework. Czarna księga*, ISBN 83-7197-733-6, data wydania 18.02.2003 r.
- [26]. .NET Framework Versions and Dependencies, <u>http://msdn.microsoft.com/en-us/library/bb822049.aspx</u>, 20.05.2013 r.
- [27]. List of .NET Framework versions, http://en.wikipedia.org/wiki/List_of_.NET_Framework_versions, 20.05.2013 r.
- [28]. Microsoft .NET Framework 4 Client Profile, <u>http://www.microsoft.com/pl-pl/download/details.aspx?id=17113</u>, 12.05.2013 r.
- [29]. http://en.wikipedia.org/wiki/File:CLR_diag.svg, 20.05.2013 r.
- [30]. UB "Understanding .NET Code Access Security", http://www.codeproject.com/Articles/5724/Understanding-NET-Code-Access-Security, 22.05.2013 r.
- [31]. "Code Access Security", <u>http://msdn.microsoft.com/en-us/library/930b76w0(v=vs.90).aspx</u>, 26.05.2013 r.
- [32]. Edgar Głowacki "*Role-Based Security*"<u>http://edu.pjwstk.edu.pl/wyklady/tbo/scb/lecture-04/lecture-04.html</u>, 26.05.2013 r.
- [33]. "*Role-Based Security*" <u>http://msdn.microsoft.com/en-us/library/52kd59t0(v=vs.90).aspx</u>, 26.05.2013 r.
- [34]. Microsoft Security Bulletin, <u>http://technet.microsoft.com/pl-pl/security/bulletin</u>, 26.05.2013 r.
- [35]. Amro Khasavneh "*Understanding LINQ (C#)*": http://www.codeproject.com/Articles/19154/Understanding-LINQ-C, 12.06.2007 r.
- [36]. Steve Burbeck, Ph.D. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC), <u>http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html</u>, 06.06.2013
 r.
- [37]. Dino Esposito. *Programming Microsoft ASP.NET MVC*, Microsoft Press, wydanie 15.05.2010 r., ISBN-10: 0735627142
- [38]. <u>http://www.cocoalab.com/media/labnotes/mvc.png</u>, 09.06.2013 r.
- [39]. "Entity Framework Overview" <u>http://msdn.microsoft.com/en-us/library/bb399567.aspx#MappingObjectsToData</u> 02.08.2012 r.
- [40]. Many To Many Mappings in Entity Framework, <u>http://weblogs.asp.net/zeeshanhirani/archive/2008/08/21/many-to-many-mappings-in-entity-framework.aspx</u> 20.05.2013 r.
- [41]. Many To Many Mappings in Entity Framework <u>http://weblogs.asp.net/zeeshanhirani/archive/2008/08/21/many-to-many-mappings-in-entity-framework.aspx</u> 20.05.2013 r.
- [42]. Bootstrap webpage, http://twitter.github.io/bootstrap/index.html, 20.04.2013 r.
- [43]. Bootstrap: Getting started, http://twitter.github.io/bootstrap/getting-started.html, 20.04.2013 r.
- [44]. Bootstrap: Scaffolding, http://twitter.github.io/bootstrap/scaffolding.html, 20.04.2013 r.
- [45]. Bootstrap: Base CSS, http://twitter.github.io/bootstrap/base-css.html, 20.04.2013 r.

- [46]. *Bootstrap: Customize and download*, <u>http://twitter.github.io/bootstrap/customize.html</u>, 20.04.2013 r.
- [47]. Soundready webpage, http://soundready.fm/, 20.04.2013 r.
- [48]. Slow Food Columbus webpage, http://slowfoodcolumbus.org/, 20.04.2013 r.
- [49]. Lech Banachowski, Elżbieta Mrówka Matejewska, Krzysztof Stencel. Systemy baz danych. Wykłady i ćwiczenia. Wydawnictwo Polsko-Japońskiej Wyższej Szkoły Technik Komputerowych, Warszawa 2004.
- [50]. Jakub Kasprzak. *Relacyjne bazy danych pojęcia podstawowe*. http://www.sqlpedia.pl/relacyjne-bazy-danych-pojecia-podstawowe/, 16.04.2013 r.
- [51]. Transact-SQL Reference (Transact-SQL), <u>http://msdn.microsoft.com/en-us/library/ms189826(v=sql.90).aspx</u>, 20.05.2013 r.
- [52]. Jakub Kasprzak. *Język SQL historia, standardy*. <u>http://www.sqlpedia.pl/jezyk-sql-historia-standardy/</u> 02.01.2013 r.
- [53]. Leonard Lobel, Andrew Burst. Programming Microsoft SQL Server 2012. ISBN 978-0-7356-5822-6, 2012
- [54]. Microsoft SQL Server, https://en.wikipedia.org/wiki/Microsoft_SQL_Server, 20.05.2013 r.
- [55]. Microsoft Releases SQL Server 2012 to Help Customers Manage "Any Data, Any Size, Anywhere", data publikacji 06.04.2012 r. <u>http://www.microsoft.com/en-us/news/press/2012/mar12/03-06SQLServer12PR.aspx</u>,
- [56]. Paweł Lenkiewicz. Administrowanie bazami danych na przykładzie Microsoft SQL Server 2005. ISBN: 978-83-89244-73-4, Warszawa 2008
- [57]. Dr. Thomas E. Hicks. *Tutorial: MSSQL- Windows Form Development*, <u>http://www.cs.trinity.edu/~thicks/Tutorials/MSSQL-Win-Form-App-Dev-Rec-Lock-1-LibraryApp/MSSQL-Win-Form-App-Dev-Rec-Lock-1-LibraryApp.html</u>, 06.06.2013 r.
- [58]. Tomasz Karwatka, *Co to jest usability*, <u>http://kursusability.pl/kurs-usability/co-to-jest-usability/co-to-jest-usability/co-to-jest-usability/co-to-jest-usability.pl/kurs-usability/co-to-jest-usability/co-to-jest-usability.pl/kurs-usability/co-to-jest-usability.pl/kurs-usability.pl/kurs-usability/co-to-jest-usability.pl/kurs-usability.pl/kurs-usability/co-to-jest-usability.pl/kurs-usability.pl/kurs-usability/co-to-jest-usability.pl/kurs-usability.pl/kurs-usability/co-to-jest-usability.pl/kurs-usabilit</u>
- [59]. Jakob Nielsen, http://pl.wikipedia.org/wiki/Jakob_Nielsen, 25.04.2013 r.
- [60]. Mariusz Trzaska, *Modelowanie i implementacja systemów informatycznych*, Wydawnictwo PJWSTK, Warszawa 2008, str. 223
- [61]. Advanced Common Sense: *About Steve Krug*, <u>http://www.sensible.com/about.html</u>, 25.04.2013 r.
- [62]. Steve Krug, Nie każ mi myśleć! Wydanie II, Wydawnictwo Helion, Gliwice 2006
- [63]. Maciej Lipiec, 6 mitów związanych z badaniami użyteczności, <u>http://uxdesign.pl/6-mitow-</u> zwiazanych-z-badaniami-uzytecznosci/, 25.04.2013 r.
- [64]. Marek Kasperski i Anna Boguska-Torbicz, *Projektowanie stron WWW*, Wydawnictwo Helion, Warszawa 2008
- [65]. Support Contrast Analyser development webpage, <u>http://www.paciellogroup.com/resources/contrastAnalyser</u>, 25.05.2013 r.
- [66]. Web Content Accessibility Guidelines 2.0, <u>http://www.w3.org/TR/2007/WD-WCAG20-20070517/#visual-audio-contrast-contrast</u>, 25.05.2013 r.

11. Wykaz tabel

Tabela 1. Przykład usługi zarządzanie komputerami osobistymi w Katalogu usług	16
Tabela 2. Opis pól formatki Nowe zgłoszenie	21
Tabela 3. Opis pozostałych pól (nie wypełnianych przez użytkownika)	21
Tabela 4. Warunki wysłania maili	25
Tabela 5. Maile do osoby/grupy wsparcia	26
Tabela 6. Maile eskalacyjne (eskalacja hierarchiczna)	27
Tabela 7. Maile dodatkowe	28
Tabela 8. Wymagania funkcjonalne	53
Tabela 9. Wymagania niefunkcjonalne - wydajność	59
Tabela 10. Wymagania niefunkcjonalne - niezawodność	59
Tabela 11. Wymagania niefunkcjonalne – bezpieczeństwo danych	60
Tabela 12. Wymagania niefunkcjonalne – łatwość użytkowania	60
Tabela 13. Wymagania niefunkcjonalne – obsługiwane przeglądarki	61
Tabela 14. Wymagania niefunkcjonalne – możliwości rozbudowy	61
Tabela 15. Wymagania niefunkcjonalne – standardy kodowania	61
Tabela 16. Historia wydań <i>frameworków</i> i języka C#	68
Tabela 17. Podstawowe klauzule	86
Tabela 18. Operatory	87
Tabela 19. Kolejne wydania SQL Server	88
Tabela 20. Statusy błędów	104
Tabela 21. Scenariusz testu: Obsługa obiektu umowa	105
Tabela 22. Scenariusz testu: Obsługa obiektu grupa	106
Tabela 23. Scenariusz testu: Obsługa obiektu producent	107
Tabela 24. Scenariusz testu: Obsługa obiektu osoba	108
Tabela 25. Scenariusz testu: Obsługa obiektu firma	109
Tabela 26. Scenariusz testu: Obsługa obiektu lokalizacja	110
Tabela 27. Scenariusz testu: Obsługa obiektu pracownicy	112
Tabela 28. Scenariusz testu: Obsługa obiektu Produkt - Urządzenie	113
Tabela 29. Scenariusz testu: Obsługa obiektu Produkt - Usługa	114
Tabela 30. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Firma	116
Tabela 31. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Lokalizacja	116
Tabela 32. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Grupa	117
Tabela 33. Scenariusz testu: Weryfikacja wprowadzenia danych - przypisanie osób do Grupy	118
Tabela 34. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Osoba	118
Tabela 35. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Pracownik	119
Tabela 36. Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Stanowisko	120

Tabela 37.	Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Produkt Urządzenie 120
Tabela 38.	Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Produkt Usługa 121
Tabela 39.	Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Producent 121
Tabela 40.	Scenariusz testu: Weryfikacja wprowadzenia danych - obiekt Zgłoszenie 122
Tabela 41.	Scenariusz testu: Rejestracja zgłoszenia przez Konsultanta 123
Tabela 42.	Scenariusz testu: Dodanie pracowników do grupy124
Tabela 43.	Scenariusz testu: Automatyczne nadawanie numeru zgłoszeń 125
Tabela 44.	Scenariusz testu: Przeszukanie bazy osób zgłaszających po dostępnych polach 126
Tabela 45.	Scenariusz testu: Przeszukiwanie bazy listy lokalizacji po dostępnych polach 126
Tabela 46.	Scenariusz testu: Wyświetlanie listy zgłoszeń 127
Tabela 47.	Scenariusz testu: Wyszukiwanie zgłoszeń po dostępnych polach 127
Tabela 48.	Scenariusz testu: Wyświetlanie listy zgłoszeń względem kończących się czasów SLA 128
Tabela 49.	Scenariusz testu: Aktualizacja statusu zgłoszeń 128
Tabela 50.	Scenariusz testu: Samodzielne definiowanie kroków w workflow
Tabela 51.	Scenariusz testu: Workflow "drukarka nie drukuje" - brak papieru, załadowanie papieru 130
Tabela 52.	Scenariusz testu: Workflow "drukarka nie drukuje" – brak papieru, wymiana drukarki 130
Tabela 53.	Scenariusz testu: Workflow "drukarka nie drukuje" - brak papieru, brak zgody na koszty131
Tabela 54.	Scenariusz testu: Workflow "drukarka nie drukuje" - brak papieru, wymiana części 132
Tabela 55	. Scenariusz testu: <i>Workflow</i> "drukarka nie drukuje" - diody się nie świecą, poprawienie kabla naprawia problem
Tabela 56	Scenariusz testu: <i>Workflow</i> "drukarka nie drukuje" - diody się nie świecą, konsultant L2 naprawia problem
Tabela 57	. Scenariusz testu: Workflow "drukarka nie drukuje" - diody się nie świecą, wymiana drukarki
Tabela 58.	Scenariusz testu: Workflow "drukarka nie drukuje" - brak tuszu/tonera
Tabela 59.	Scenariusz testu: Workflow "drukarka nie drukuje" - zacięty papier
Tabela 60	Scenariusz testu: <i>Workflow</i> "awaria samochodu" - kontrolki się nie świecą, poprawienie klem naprawia problem
Tabela 61.	Scenariusz testu: <i>Workflow</i> "awaria samochodu" - kontrolki się nie świecą, konsultant L2 naprawia problem
Tabela 62.	Scenariusz testu: Workflow "awaria samochodu" - kontrolki się nie świecą, naprawione w serwisie
Tabela 63.	Scenariusz testu: <i>Workflow</i> "awaria samochodu" - kontrolki się świecą wciśnięcie sprzęgła naprawia problem
Tabela 64	. Scenariusz testu: <i>Workflow</i> "awaria samochodu" - kontrolki się świecą, konsultant L2 naprawia problem
Tabela 65	. Scenariusz testu: <i>Workflow</i> "awaria samochodu" - kontrolki się świecą, brak akceptacji kosztów transportu
Tabela 66.	Scenariusz testu: Workflow "awaria samochodu" - kontrolki się świecą, naprawa w serwisie
Tabela 67.	Zestawienie wyników badania kontrastu pomiędzy kolorem tła a kolorem czcionki 149

12. Wykaz rysunków

Rysunek 1. Struktura organizacyjna zespołu projektowego	9
Rysunek 2. Cykl życia usługi	13
Rysunek 3. Planowanie katalogu usług	15
Rysunek 4. Katalog usług	15
Rysunek 5. Centralny Service Desk	
Rysunek 6. System rejestracji zgłoszeń oparty na HP OpenView Service Desk	29
Rysunek 7. Stosowanie zasad When.IF, THEN, Else	30
Rysunek 8. Dostarczenie dynamicznych interfejsów	31
Rysunek 9. Dostarczenie scenariusza dla agentów	32
Rysunek 10. Ekran do projektowania skryptów	33
Rysunek 11. Ekran projektowania workflow	34
Rysunek 12. Przykładowa formatka nowego zgłoszenia	36
Rysunek 13. Przykładowy workflow dla pierwszej linii wsparcia	37
Rysunek 14. Przykładowy wygląd modułu SD.Inventory	38
Rysunek 15. Mapa możliwych ścieżek komunikacyjnych dla modułu SD. Bridge	39
Rysunek 16. Przykład zastosowania modułu SD. Solution	40
Rysunek 17. Wynik działania przykładowego raportu	41
Rysunek 18. Przykładowy alert modułu SD.Message	
Rysunek 19. Przykładowy wygląd mobilnej wersji aplikacji SolveDirect	43
Rysunek 20. Aktorzy obecni w systemie	44
Rysunek 21. Aktywność aktorów: kierownik, konsultanci L1/L2	45
Rysunek 22. Przykładowy scenariusz dla aktora "Kierownik"	
Rysunek 23. Przykładowy scenariusz dla aktora "Konsultant"	47
Rysunek 24. Wyszukiwanie i wyświetlanie zgłoszeń	49
Rysunek 25. Praca ze zgłoszeniami i tworzenie zgłoszeń	50
Rysunek 26. Czynności administracyjne: zarządzanie użytkownikami	51
Rysunek 27. Fragment diagramu klas	53
Rysunek 28. Historia rozwoju .NET	63
Rysunek 29. Kompilacja kodu w .NET	64
Rysunek 30. Notacja języka w muzyce	67
Rysunek 31. Popularność różnych języków programowania na przestrzeni lat	68
Rysunek 32. Common Language Runtime	70
Rysunek 33. Podstawowy schemat przedstawiający funkcje danych słów w LINQ	70
Rysunek 34. Schemat zintegrowania LINQ z platforma .NET	71
Rysunek 35. Schemat MVC	72
Rysunek 36. Przykład scenariusza workflow	73

Rysunek 37.	Przykład dedykowanej aktywności	74
Rysunek 38.	Widok zmodyfikowanego toolboxa Workflow Designer	75
Rysunek 39.	Entity Framework – przepływ danych	76
Rysunek 40.	Asocjacja wiele do wiele w tradycyjnej bazie danych	77
Rysunek 41.	Asocjacja wiele do wiele z użyciem Entity Framework	77
Rysunek 42.	Strona tytułowa Bootstrap	79
Rysunek 43.	Logicznie pogrupowana struktura plików Bootstrap	79
Rysunek 44.	Przykładowy plik HTML z <i>Bootstrap</i>	80
Rysunek 45.	Dodanie szablonu <i>Bootstrap</i> do pliku HTML	81
Rysunek 46.	Przykładowy dostępny układ strony	81
Rysunek 47.	Przykładowy Bootstrap grid	82
Rysunek 48.	Technologia Response design na różnych urządzeniach	82
Rysunek 49.	Rozmiary przycisków	83
Rysunek 50.	Przykładowy efekt styli walidacji wprowadzanych danych	83
Rysunek 51.	Dopasowanie komponentów Bootstrap	84
Rysunek 52.	Przykład strony wykorzystującej technologię Bootstrap	85
Rysunek 53.	Przykład strony wykorzystującej technologię Bootstrap	85
Rysunek 54.	Przykładowy schemat bazy danych	86
Rysunek 55.	Przykład zapytania Update	87
Rysunek 56.	Przykład zapytania Create	88
Rysunek 57.	Przykład zapytania Grant	88
Rysunek 58.	Przykład zapytania Select	88
Rysunek 59.	Nadawanie uprawnień w MS SQL Server	90
Rysunek 60.	Klasa asocjacyjna PrivilegeType	92
Rysunek 61.	Klasa Employee	93
Rysunek 62.	Projekt implementacyjny Produktów i ich cech	93
Rysunek 63.	Projekt generyczności parametrów produktów	94
Rysunek 64.	Projekt implementacji firmy	94
Rysunek 65.	Projekt implementacji produktu	95
Rysunek 66.	Projekt implementacji pracownika	95
Rysunek 67.	Projekt implementacji zgłoszenia – część 1 z 2	96
Rysunek 68.	Projekt implementacji zgłoszenia – część 2 z 2	96
Rysunek 69.	Okienko Solution Explorer, widok folderu Models	98
Rysunek 70.	Widok ekranu programu Microsoft Management Studio ilustruje układ tabeli realizują dziedziczenie1	cej 03
Rysunek 71.	Umiejscowienie 'Package Manager Console'	03
- Rysunek 72.	Przygotowanie migracji	03
Rysunek 73.	Uaktualnienie bazy danych	04

Rysunek 74. Układ strony z wyróżnionymi elementami składowymi (układ dwukolumno asymetryczny z dodatkowym menu lewostronnym)	owy 146
Rysunek 75. Wycinek menu głównego, zawierający słabo wyróżniające się logo	146
Rysunek 76. Wycinek menu głównego z poprawionym logo	146
Rysunek 77. Dodawanie firmy i niewidoczne przyciski	147
Rysunek 78. Niepotrzebny dodatkowy przycisk do strony głównej	148
Rysunek 79. Przykład breadcrumbs zastosowanych w aplikacji "Service desk"	148
Rysunek 80. Przykładowa wyszukiwarka	148
Rysunek 81. Widok aplikacji - menu główne z widocznym panelem do nawigacji i menu bocznym	151
Rysunek 82. Przyciski na formularzu do zatwierdzania i anulowania akcji	152
Rysunek 83. Lista osób – dodawanie, wejście w szczegóły danego rekordu, edycja i kasowanie	152
Rysunek 84. Workflow "Weryfikacja klienta"	153
Rysunek 85. Workflow "Awaria drukarki" część 1 z 2	155
Rysunek 86. Workflow "Awaria drukarki" część 2 z 2	156
Rysunek 87. Workflow "Awaria monitora" część 1 z 2	158
Rysunek 88. Workflow "Awaria monitora" część 2 z 2	159
Rysunek 89. Workflow "Awaria komputera"	160
Rysunek 90. Workflow "Awaria samochodu" część 1 z 2	162
Rysunek 91. Workflow "Awaria samochodu" część 2 z 2	163
Rysunek 92. Główne okno aplikacji PJWSTK Workflow Designer	164
Rysunek 93. Projektowanie workflow "Drukarka nie drukuje"	165
Rysunek 94. Przykład wyrażenia dla kontrolki Decision	165
Rysunek 95. Przykład wyrażenia dla kontrolki E-mail	166
Rysunek 96. Konfiguracja aktywności "Tak / nie"	166
Rysunek 97. Aktywność kończąca przepływ	166

13. Wykaz listingów

Listing 1. Klasa Employee z mapowaniem po stronie bazy danych	
Listing 2. Deklaracja pól i konstruktora	
Listing 3. Deklaracja klasy Id	
Listing 4. Deklaracja klasy SqlObject z zadeklarowanych polem klucza głównego	
Listing 5. Deklaracja klasy Firm z dziedziczeniem z klasy SqlObject	100
Listing 6. Deklaracja klasy Firm ze słowem kluczowym virtual	100
Listing 7. Definicja komunikatu błędu	100
Listing 8. Definicja wymagalności atrybutu z komunikatem błędu	100
Listing 9. Definicja wymagalności atrybutu z komunikatem błędu w osobnym pliku	101
Listing 10. Implementacja ekstensji	101
Listing 11. Implementacja relacji jeden do wielu po stronie jeden	101
Listing 12. Implementacja relacji jeden do wielu po stronie wiele	102
Listing 13. Implementacja relacji wiele do wielu poprzez klasę asocjacyjną	102