



POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

Wydział Informatyki

Katedra Inżynierii Oprogramowania

Inżynieria Oprogramowania i Baz Danych

Marcin Bienias, s19685

Filip Bojanowski, s19962

Patryk Grzeszczak, s19676

Dominik Kabala, s18585

Mateusz Lajnert, s9817

Daniel Liebchen, s18529

Szymon Łaguna, s18788

Marcin Marciniak, s11981

Oskar Marszałek, s18324

Tuan Viet Nguyen, s18964

Tomasz Palczewski, s18969

Michał Pazur, s19023

Szerownia.pl

System umożliwiający wzajemne pożyczanie przedmiotów codziennego użytku.

Praca inżynierska
napisana pod kierunkiem:
Dr. inż. Mariusza Trzaski

Warszawa, luty 2022

Streszczenie

Aplikacja, która da jej użytkownikom możliwość realizacji wypożyczeń różnych przedmiotów, obraliśmy za cel prac programistycznych. Zamiast nadmiarowo kupować przedmiot, z którego skorzystamy kilka razy do roku, lepiej za drobną opłatą, pożyczyć go od kogoś z okolicy.

Idąc z duchem czasu i dostępem do coraz to nowszych urządzeń elektronicznych, aplikację stworzono w wersji przeglądarkowej oraz mobilnej w systemie Android. Podstawową technologią jest Java, dla interfejsu użytkownika użyto biblioteki React i JavaScript.

Słowa kluczowe

Wypożyczenie, wypożyczający, użyczający, ogłoszenie, ogłoszenie użyczenia, ogłoszenie poszukiwania, polisa, towarzystwo ubezpieczeniowe, szkoda, likwidator, rzeczoznawca, oględziny, ocena, ocena użyczenia, ocena użytkownika, zgłoszenie, moderator, administrator, zgłoszenie do moderacji, java, mysql, android, rest api, spring, jwt token, hibernate, bootstrap, zależności, postman, swagger, maven, materialui, javascript, react, selenium, intellij, teams, aplikacja przeglądarkowa, aplikacja mobilna.

Spis treści

STRESZCZENIE.....	2
SŁOWA KLUCZOWE.....	2
SPIS TREŚCI.....	3
4.10 Postman i Swagger.....	69
4.10.1 Postman	69
4.10.2 Swagger	70
4.11. Android SDK	72
4.12. Selenium - testy automatyczne	73
4.13. Testy jednostkowe.....	74
5.1. Wzorzec projektowy	77
5.1.1 Warstwa widoku	78
5.1.2 Warstwa kontrolera	82
5.2. Pozostałe wzorce projektowe.....	85
5.2.1 Fasada	85
5.2.2 Data Access Object.....	86
5.2.3 Data transfer Object.....	86
5.2.4 Wstrzykiwanie zależności.....	87
5.3. Komunikacja z bazą danych	87
5.4. Mechanizmy zapewniające bezpieczeństwo	88
5.4.1 Logowanie do systemu.....	88
5.4.2 JWT Token	89
5.4.3 Dostęp do funkcjonalności wymagających logowania	89
5.5. Testowanie aplikacji	91
5.5.1 Manualne testy integracyjne i systemowe.....	92
5.5.2 Manualne testy integracyjne i systemowe.....	94
5.5.3 Testy automatyczne.....	96
6.2. Użytkownik niezalogowany.....	99
6.3. Użytkownik zalogowany i premium	102
6.3.1 Użytkownik zalogowany.....	102
6.3.2 Użytkownik premium	107
6.4. Likwidator.....	111
6.5. Rzeczoznawca.....	115
6.6. Moderator.....	116

6.7. Administrator	117
6.8. Aplikacja mobilna	119
7.1 Wprowadzenie	125
7.2. Organizacja pracy	125
7.3. Uwagi o technologiach.....	126
7.4. Rozwój aplikacji	127
7.5. Podsumowanie	128
Bibliografia	130
Spis ilustracji.....	133
Spis tabel.....	136
Spis listingów.....	137

1. Wstęp

Spółczeństwu XXI wieku zależy na kompleksowej i bezkontaktowej obsłudze spraw za pośrednictwem Internetu. Jednocześnie świat jest pochłaniany przez wszechobecny konsumpcjonizm. Mając na celu zaspokojenie potrzeb klientów, a zarazem ograniczenie kupowania postanowiliśmy stworzyć platformę, która pozwoli na korzystanie z rzeczy właśnie wtedy, gdy są potrzebne. W momencie wybierania tematu niniejszej pracy na rynku polskim nie było takiego serwisu, dlatego można uznać ten projekt za innowacyjny. Rozwiązania dla wypożyczania przedmiotów zaczęły się pojawiać w Internecie w trakcie prac nad naszym projektem.

1.1. Cel pracy

Głównym zamiarem pracy inżynierskiej było utworzenie narzędzia, które ma za zadanie wspomóc wypożyczanie przedmiotów między użytkownikami korzystającymi z serwisu Szerownia.pl. Za pomocą stworzonych funkcjonalności system umożliwia wypożyczenie przedmiotu przez użytkownika, ale i nie tylko. W ten sposób użytkownicy mogą zaoszczędzić pieniądze, które mogą przeznaczyć na inne cele niż kupowanie drogich przedmiotów, które nie będą ponownie używane. W drugą stronę, osoby chcące dorobić mogą udostępniać swoje rzeczy, żeby wspomóc innych użytkowników.

1.2. Organizacja pracy

Dokument pracy dyplomowej przedstawia etapy wytwarzania oprogramowania oraz dokumentuje wkład pracy całego zespołu w realizację projektu. Kolejne podrozdziały zostały poświęcone zakresowi prac nad systemem Szerownia.pl. Praca została podzielona na odpowiednie rozdziały, które opisują pracę nad projektem.

1.3. Zakres prac

Praca nad aplikacją odbywała się w dwunastoosobowym zespole projektowym, który został podzielony na pięć podzespołów:

1. Analiza,
2. Projektowanie,
3. Implementacja,
4. Testy,

5. Dokumentacja.

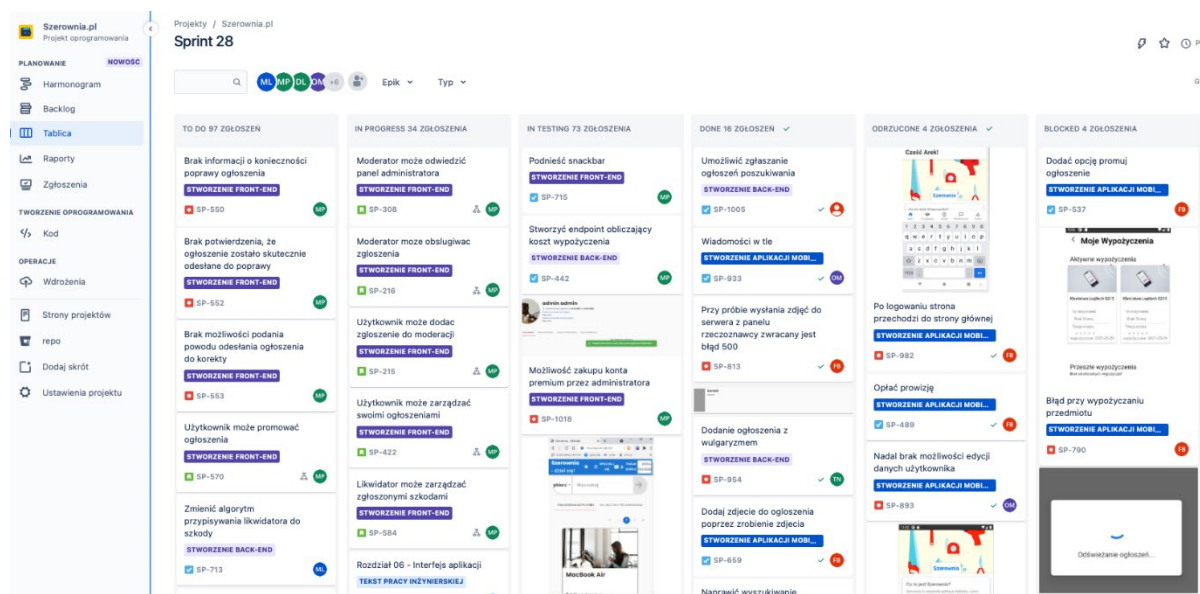
Nie było członka zespołu, który pracował tylko w ramach jednego podzespołu. Każdy wносił swoją pracą istotny wkład do całości.

Momentem, w którym odbywało się planowanie, przydzielanie i rozliczanie z zadań były zajęcia projektowe. Poszczególni członkowie zespołów spotykali się głównie online przy użyciu aplikacji MS Teams [1]. Natomiast narzędzie Jira Software było wsparciem podczas każdego z etapów wytwarzania oprogramowania [2]. Zadania utworzone do realizacji posiadały następujące statusy:

- Zablokowane,
- Do zrobienia,
- W trakcie,
- W testach,
- Zakończone,
- Odrzucone.

Dokumentacja, notatki z zajęć oraz uwagi promotora były umieszczane w Confluence [3]. Dzięki ogólnemu dostępowi do zasobu, członkowie zespołów zawsze mogli z nich korzystać.

Kierownik projektu był obowiązany do nadzorowania i weryfikacji postępu prac danego zespołu. Rysunek 1 przedstawia kolumny z zadaniami w Jira.



Rysunek 1. Zadania zaplanowane za pomocą narzędzie Jira. Źródło: Opracowanie własne.

Skład poszczególnych zespołów oraz kierowników pokazany jest w tabeli numer 1. Do uczestników projektu został przypisany wykaz zadań realizowanych przez nich.

Tabela 1. Podział podzespołów i przydział zadań. Źródło: Opracowanie własne.

Kierownik projektu Szerownia.pl	
Marcin Bienias	
Zespół analizy	
Kierownik	Marcin Bienias
Członkowie zespołu	Filip Bojanowski Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Daniel Liebchen Tuan Viet Nguyen Marcin Marciniak
Podział zadań	
Stan sztuki	Marcin Bienias Patryk Grzeszczak Tuan Viet Nguyen
Funkcjonalności systemu	Filip Bojanowski Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Daniel Liebchen
Przypadki użycia	Marcin Bienias Filip Bojanowski Mateusz Lajnert Daniel Liebchen
Diagram klas	Marcin Marciniak Filip Bojanowski Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Daniel Liebchen Tuan Viet Nguyen Marcin Marciniak
Diagramy aktywności	Mateusz Lajnert

	Daniel Liebchen Tuan Viet Nguyen Patryk Grzeszczak Dominik Kabala
--	--

Zespół projektowania	
Kierownik zespołu	Filip Bojanowski
Członkowie zespołu	Marcin Bienias Mateusz Lajnert Tomasz Palczewski Michał Pazur
Podział zadań	
Projektowanie ekranów aplikacji webowej	Michał Pazur
Projektowanie ekranów aplikacji mobilnej	Filip Bojanowski
Projektowanie panelu likwidatora	Mateusz Lajnert
Projektowanie panelu rzeczoznawcy	Mateusz Lajnert
Projektowanie panelu moderatora	Tomasz Palczewski
Projektowanie panelu administratora	Tomasz Palczewski
Projektowanie raportów	Filip Bojanowski
Walidacja wymagań i funkcjonalności	Michał Pazur
Logo aplikacji	Michał Pazur

Zespół implementacji	
Kierownik	Michał Pazur
Członkowie zespołu	Filip Bojanowski Szymon Łaguna Marcin Bienias Oskar Marszałek Tuan Viet Nguyen Tomasz Palczewski
Podział zadań	
Szkolenie REST API	Marcin Bienias
Szkolenie SpringBoot i Hibernate	Tomasz Palczewski
Szkolenie Android Studio	Oskar Marszałek

Szkolenie JavaScript i React	Michał Pazur
Tworzenie i przydział zadań Jira	Michał Pazur
Konfiguracja serwera i wdrożenie aplikacji	Michał Pazur
Implementacja klas z diagramu i mapowanie Hibernate	Szymon Łaguna Marcin Bienias Tuan Viet Nguyen Tomasz Palczewski
Implementacja generycznej obsługi błędów zapytań REST	Tomasz Palczewski
Implementacja skryptu wypełniającego bazę danych	Tuan Viet Nguyen
Tworzenie widoków w React	Michał Pazur
Implementacja Back-End	Szymon Łaguna Marcin Bienias Tuan Viet Nguyen Tomasz Palczewski
Implementacja Front-End	Michał Pazur
Implementacja modułów systemu	Szymon Łaguna Marcin Bienias Tuan Viet Nguyen Tomasz Palczewski
Szyfrowanie danych logowania	Tomasz Palczewski
Rejestracja użytkowników	Marcin Bienias
Dodawanie ogłoszeń	Tomasz Palczewski
Nadawanie uprawnień użytkownikom wewnętrznym	Tuan Viet Nguyen
Analiza UX	Marcin Bienias Szymon Łaguna
Poprawa błędów UX	Michał Pazur
Analiza błędów back-end	Tuan Viet Nguyen Tomasz Palczewski
Poprawa błędów back-end	Szymon Łaguna Marcin Bienias Tuan Viet Nguyen Tomasz Palczewski
Budowanie aplikacji mobilnej	Filip Bojanowski Oskar Marszałek

API dla aplikacji mobilnej	Filip Bojanowski Oskar Marszałek
Ekrany aplikacji mobilnej	Filip Bojanowski Oskar Marszałek

Zespół testów	
Kierownik	Daniel Liebchen
Członkowie zespołu	Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Marcin Marciniak
Podział zadań	
Przygotowanie scenariuszy testowych i przypadków testowych	Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Daniel Liebchen Marcin Marciniak
Testy manualne funkcjonalności	Patryk Grzeszczak Dominik Kabala Mateusz Lajnert Daniel Liebchen Marcin Marciniak
Testy API	Patryk Grzeszczak Marcin Marciniak
Testy automatyczne	Dominik Kabala Daniel Liebchen

Zespół dokumentacji	
Kierownik	Mateusz Lajnert
Członkowie zespołu	Patryk Grzeszczak Dominik Kabala Daniel Liebchen

	Szymon Łaguna Marcin Marciniak
Podział zadań	
Streszczenie, słowa kluczowe	Mateusz Lajnert Marcin Marciniak
Rozdział 1: podrozdziały 1.1 i 1.2	Patryk Grzeszczak
Rozdział 1: podrozdział 1.3	Mateusz Lajnert
Rozdział 1: podrozdział 1.4	Dominik Kabala
Rozdział 2: podrozdziały: 2.1 i 2.2	Marcin Marciniak
Rozdział 2: podrozdziały: 2.3 i 2.4	Mateusz Lajnert
Rozdział 3: podrozdział 3.1	Patryk Grzeszczak Marcin Marciniak
Rozdział 3: podrozdział 3.2	Daniel Liebchen Szymon Łaguna
Rozdział 3: podrozdział 3.3	Dominik Kabala
Rozdział 3: podrozdziały 3.4 i 3.5	Mateusz Lajnert
Rozdział 3: podrozdział 3.6	Patryk Grzeszczak
Rozdział 3: podrozdział 3.7 i 3.12	Marcin Marciniak
Rozdział 3: podrozdział 3.8	Patryk Grzeszczak
Rozdział 4: podrozdziały 4.1 i 4.2	Mateusz Lajnert
Rozdział 4: podrozdziały 4.3 i 4.11	Marcin Marciniak
Rozdział 4: podrozdziały 4.4, 4.8 i 4.9	Patryk Grzeszczak
Rozdział 4: podrozdziały 4.5 i 4.7	Dominik Kabala
Rozdział 4: podrozdziały 4.6 i 4.13	Szymon Łaguna
Rozdział 4: podrozdziały 4.10 i 4.12	Daniel Liebchen
Rozdział 5: podrozdział 5.1	Marcin Marciniak Szymon Łaguna
Rozdział 5: podrozdziały 5.2 i 5.3	Daniel Liebchen
Rozdział 5: podrozdział 5.4	Dominik Kabala Patryk Grzeszczak
Rozdział 5: podrozdział 5.5	Patryk Grzeszczak Dominik Kabala Szymon Łaguna Marcin Marciniak

Rozdział 6: podrozdział 6.1	Daniel Liebchen Marcin Marciniak
Rozdział 6: podrozdział 6.2 i 6.6	Dominik Kabala
Rozdział 6: podrozdział 6.3	Patryk Grzeszczak
Rozdział 6: podrozdziały 6.4 i 6.5	Szymon Łaguna
Rozdział 6: podrozdział 6.7	Daniel Liebchen
Rozdział 6: podrozdział 6.8	Marcin Marciniak
Rozdział 7: podrozdziały 7.1 i 7.3	Marcin Marciniak
Rozdział 7: podrozdział 7.2	Dominik Kabala
Rozdział 7: podrozdział 7.4	Daniel Liebchen
Bibliografia, Spis ilustracji, Spis tabel, Spis listingów	Mateusz Lajnert

1.4. Kontekst prawny i codzienne funkcjonowanie

Niniejszy rozdział przybliży czytelnikowi regulacje prawne oraz charakter funkcjonowania wypożyczania przedmiotów.

1.4.1 Kontekst prawny

W Polsce istnieją przepisy prawne regulujące najem. Podstawą prawną najmu jest Tytuł XVII Najem i dzierżawa Kodeksu Cywilnego [4]. Opłaty są realizowane przez aplikację w formie szybkich płatności. Wynajmujący oraz najemca zawierają umowę wynajmu, w której określają warunki wynajmu oraz oznaczają czas wynajmu. Najemca jest zobowiązany do oddania przedmiotu w stanie niepogorszonym.

1.4.2 Ochrona danych osobowych

Niezwykle istotnym elementem aplikacji jest ochrona danych osobowych. Prawo polskie zobowiązuje nas jako dostawcy i administratora aplikacji do odpowiedniego zabezpieczenia danych użytkowników. Jako autorzy aplikacji jesteśmy bezpośrednio odpowiedzialni za wszystkie naruszenia ochrony danych osobowych. Cięży na nas obowiązek tworzenia umów o powierzeniu przetwarzania danych oraz informowania Urzędu Ochrony Danych Osobowych o wszelkich naruszeniach w terminie 72h od wykrycia naruszenia [5]. Dodatkowo każdy użytkownik może od nas zażądać udostępnienia oraz usunięcia wszystkich jego danych osobowych. Jako firma kontrolująca oraz przetwarzająca dane musimy wyznaczyć odpowiednią osobę na stanowisko Inspektora Ochrony Danych. Osoba ta musi

dysponować wiedzą ekspercką w zakresie ochrony danych osobowych. Dane przechowywane przez naszą aplikację nie mogą być przesyłane poza Unię Europejską bez zachowania odpowiedniego poziomu zabezpieczeń.

1.4.3 Codzienne funkcjonowanie

Większość przedmiotów dostępnych na stronie stanowią sprzęty elektroniczne i elektryczne. Określone kategorie pozwalają właścicielom zakwalifikować przedmioty do odpowiedniej kategorii. W przypadku, gdy jakiś przedmiot nie kwalifikuje się do żadnej podkategorii, powinna ona zostać dodana przez administratora, w celu ulepszenia możliwości filtrowania i wyszukiwania na stronie. Wyszukiwarka jest istotną funkcjonalnością na stronie, dzięki niej użytkownicy mogą wyszukiwać przedmioty wykorzystując w tym celu interesujące ich kategorie lub korzystając z zaawansowanego filtrowania przedmiotów.

Najważniejszymi elementami działania aplikacji są zapewnienie użytkownikom możliwości wystawiania ofert użyczenia oraz możliwości wypożyczenia tych przedmiotów. Dodatkową funkcją dostępną dla użytkowników jest ubezpieczenie wynajmu. Aby zapewnić te funkcjonalności musimy dać użytkownikom możliwość komunikacji między sobą. Będą do tego służyć wiadomości prywatne. Istotnym elementem funkcjonowania systemu jest odpowiedzialność za uszkodzenie lub utratę przedmiotu. W przypadku gdy na wypożyczenie zostało wykupione ubezpieczenie, odszkodowanie wypłaca ubezpieczyciel, natomiast jeżeli ubezpieczenie nie zostało wykupione wynajmujący jest obowiązany sam zapłacić za szkody. Właściciel przedmiotu sam określa wartość przedmiotu podając kwotę podczas tworzenia ogłoszenia. W systemie dajemy możliwość firmom ubezpieczeniowym sprzedaży polis ubezpieczeniowych oraz likwidacji szkód. Odbywa się to poprzez rolę Likwidatora, który ma za zadanie procesować likwidację szkody oraz Rzecznawcę, który jest odpowiedzialny za weryfikację i ocenę zniszczonego mienia.

Niezbędnymi elementami systemu są konta użytkowników i zarządzanie nimi. Użytkownik może stworzyć swoje konto. W przypadku utracenia hasła może je odzyskać korzystając ze swojego adresu e-mail. Użytkownik zarejestrowany może również zmieniać dane na swoim profilu oraz hasło, a także wystawiać opinie innym użytkownikom po zakończeniu wynajmu przedmiotu.

2. Istniejące systemy umożliwiające wypożyczenia

Na świecie funkcjonuje wiele aplikacji służących do ogłaszania sprzedaży, wynajmu czy usług. Przed przystąpieniem do projektowania i implementacji systemu będącego tematem niniejszej pracy, zespół zapoznał się z kilkoma propozycjami podobnych rozwiązań. W tym rozdziale zostaną przedstawione trzy propozycje:

- LokPo Lokalne Pożyczanie - polski portal, oferujący możliwość wystawiania ogłoszeń na pożyczanie przedmiotów,
- OLX - serwis ogłoszeniowy będący częścią Grupy OLX Sp. z o.o.,
- Plenti - kolejna polska firma z Warszawy oferująca elektronikę na wypożyczenie.

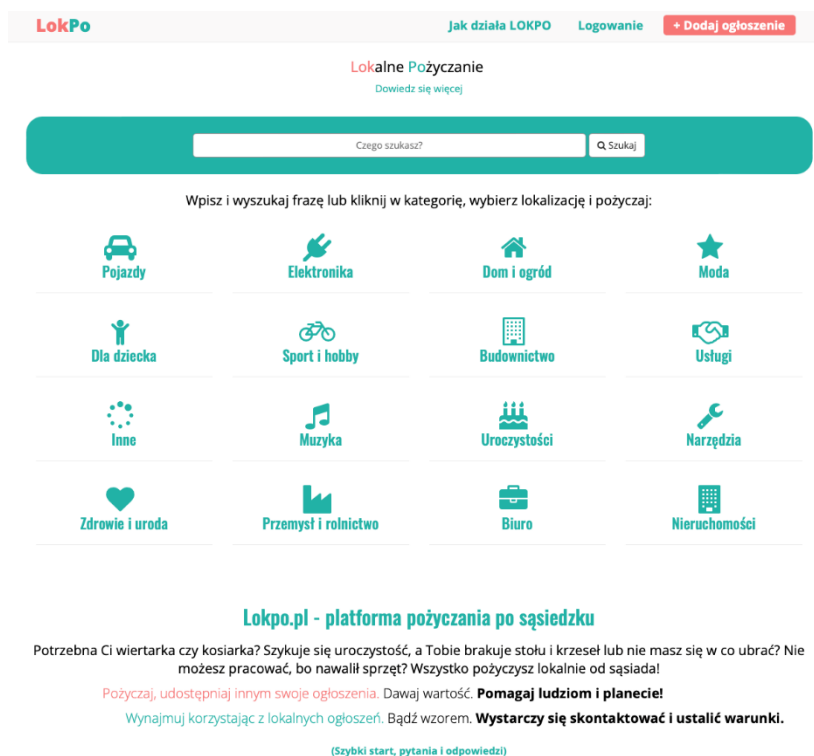
Wszystkie opisane aplikacje są wystawione na świat produkcyjnie i zostały przetestowane od strony użytkownika serwisu.

2.1. LokPo. Lokalne Pożyczanie

LokPo Lokalne Pożyczanie to portal zbliżony do Szerownia.pl. Użytkownicy mają możliwość ogłoszenia przedmiotów, które chcieliby użyczyć. Osoby, które nie są zainteresowane kupnem czegoś na własność, mogą wypożyczyć przedmiot na określony czas. Dotyczy to zarówno fizycznych przedmiotów jak i usług.

Serwis nie jest zbyt popularny. Przykładowo, kategoria Elektronika obejmuje jedynie około 70 ogłoszeń. Być może wynika to z tego, że powstał stosunkowo niedawno. Strona została utworzona w 2020 roku, prawdopodobnie jako odpowiedź na pandemię koronawirusa. Być może niska popularność jest również, skutkiem tego, że właściciele nie zdecydowali się na promocję serwisu poza województwem śląskim, skąd pochodzi większość ogłoszeń (głównie z Katowic).

Rysunek numer 2 przedstawia widok strony głównej serwisu LokPo.



Rysunek 2. Strona główna serwisu www.lokpo.pl. Źródło: Opracowanie własne.

Zgodnie z opisem dostępnym na stronie internetowej, serwis umożliwia przeglądanie ogłoszeń, wystawianie własnych przedmiotów lub usług oraz kontakt z osobą ogłaszającą przedmiot do użyczenia. Rejestracja w serwisie możliwa jest poprzez utworzenie konta. Brak możliwości korzystania z serwisu poprzez zalogowanie się kontem Googla lub Facebooka. Ciekawą funkcją jest automatyczne inicjowanie procedury zakładania konta, jeśli użytkownik nie był wcześniej zalogowany. Aplikacja nie umożliwia dokonywania płatności ani zamawiania dostawy, stanowi jedynie tablicę ogłoszeń. Serwis nie udostępnia również kalendarza wypożyczeń, czy narzędzia do wprowadzania rabatów. Tego typu informacje muszą zostać zawarte w części opisowej ogłoszenia.

Użytkownik może dodać nowe ogłoszenie. Aplikacja nie umożliwia wyboru cech produktu, niezależnie od kategorii przedmiotu. Można dodać zdjęcia (maksymalnie 4). Poza wyborem kategorii, obowiązkowe jest też podanie minimalnego kosztu wypożyczenia.

Rysunek numer 3 pokazuje formularz dodawania nowego ogłoszenia w serwisie LokPo.

Dodaj ogłoszenie:
To tylko 1 minuta!

Tytuł*

Kategoria*

Koszt od*

Opis (rozbudowany przyciągnie więcej chętnych)*
 Wprowadź opis - napisz kilka zdań - określ wymiary, funkcje, opis sposób używania itp. Podaj także warunki wynajmu np. sprecyzuj ceny - dla dłuższych okresów możesz dać rabaty itd.
 - Wymagany depozyt: (np. 100zł lub brak)
 - Wypożyczenie na: (np. dzień, weekend, 7 dni)

Zdjęcie (< 1 Mb) **Zdjęcie2** **Zdjęcie3** **Zdjęcie4**

Imię* **Email*** **Telefon***

Miejscowość* **Dzielnica** **Ulica** **Numer**

Dodaj ogłoszenie

Rysunek 3. Formularz dodawania nowego ogłoszenia. Źródło: Opracowanie własne.

W tabeli numer 2 zostały przedstawione moduły i elementy omawianej aplikacji.

Tabela 2. Obszary wspierane przez serwis LokPo. Źródło: Opracowanie własne.

Lp.	Obszar	Opis podstawowych funkcjonalności
1	Rejestracja i logowanie	- zarejestrowanie nowego konta w serwisie za pomocą adresu e mail, który jednak nie jest sprawdzany mailem weryfikującym - zalogowanie się do serwisu za pomocą numeru telefonu, konta Google lub Facebook nie jest możliwe
2	Kategorie	- przeglądanie kategorii znajdujących się aktualnie w serwisie - lista głównych kategorii na stronie głównej, brak podkategorii - przeglądanie ofert wypożyczenia po wybranej kategorii - wyszukiwanie ofert po nazwie przedmiotu - brak możliwości sortowania i filtrowania

3	Zamówienia	- przeglądanie ogłoszenia i zapoznanie ze szczegółami zawartymi w opisie - kontakt bezpośredni z ogłaszającym na podstawie podanych danych kontaktowych lub modułu do korespondencji
4	Promocje	- brak promocji związanych z wypożyczaniem - brak płatności na rzecz serwisu
5	Obsługa klienta	- skrzynka wiadomości
6	Panel użytkownika	- ogłoszenia użytkownika - wiadomości - ustawienia

LokPo jest prostym serwisem do wypożyczeń. Działa głównie na rynku lokalnym, choć zdarzają się ogłoszenia np. z Warszawy. Liczba funkcji jest bardzo ograniczona. Właściwie, można podsumować, że portal służy wyłącznie jako tablica ogłoszeń, specjalizująca się w wypożyczeniach.

2.2. OLX

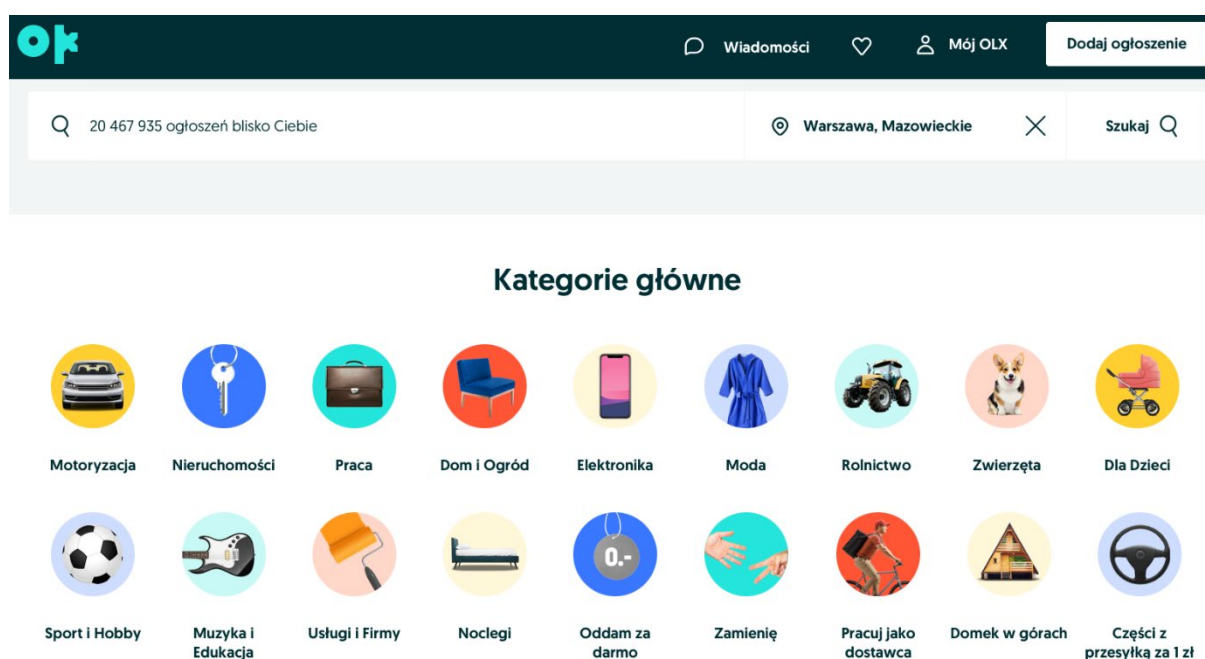
Portal OLX.pl to serwis ogłoszeń lokalnych. Pierwotnie funkcjonował pod nazwą szerlok.pl, później tablica.pl, kiedyś część grupy Allegro. Obecnie należy do grupy medialno-technologicznej Naspers. Serwis pierwotnie nie był przeznaczony do wypożyczeń, jednak właściciel platformy rozwija funkcjonalności w tym kierunku. Zmiany były promowane szeroko zakrojoną akcją marketingową. Funkcje, jakie udostępnia serwis to:

- przeglądanie ogłoszeń,
- dodawanie ogłoszeń,
- dodawanie zdjęć z poziomu aplikacji lub strony mobilnej,
- wyróżnienie ogłoszeń,
- wymiana wiadomości,
- wysyłanie CV za pośrednictwem formularza w kategorii Praca,
- obserwowanie wybranych ogłoszeń lub wyników wyszukiwania,
- dostęp do strony przy pomocy telefonu komórkowego: aplikacja i strona mobilna.

Rejestracja w serwisie możliwa jest poprzez utworzenie konta lub zalogowanie się przez konto na Facebooku lub Googla. Przy tworzeniu konta konieczna jest weryfikacja użytkownika poprzez prawidłowe wprowadzenie sześciocyfrowego hasła, przesłanego na wskazany numer telefonu.

Użytkownik (po zalogowaniu) może dodać nowe ogłoszenie. Serwis nie udostępnia wprost możliwości wypożyczenia przedmiotu. Możliwe opcje to: sprzedaż, oddanie za darmo, zamiana.

Jeśli chcemy wystawić przedmiot do wypożyczenia musimy zawrzeć taką informację w tytule ogłoszenia, a szczegółowe warunki opisać w treści. Ogólną cechą ogłoszeń w OLX jest to, że niewiele jest zamkniętych opcji do wykorzystania przy opisie przedmiotu, a większość informacji wprowadzana jest w formie tekstu. Rysunek numer 4 przedstawia widok strony głównej serwisu OLX.



Rysunek 4. Strona główna serwisu OLX. Źródło: Opracowanie własne.

Sam serwis kwestię wypożyczeń rozwiązał, dodając do kategorii “Usługi i firmy” podkategorie “Wypożyczalnia”, gdzie z kolei można zamieszczać ogłoszenia. Zdefiniować można: kategorię, cenę czy ogłoszenie jest prywatne, czy firmowe, stan przedmiotu oraz dodatkowe (niezbyt liczne) informacje dla poszczególnych kategorii, np. dla kategorii „bielizna” można zdefiniować, czy jest „damska” czy „męska”. Dla kategorii „sporty zimowe” brak jakichkolwiek określonych cech.

Tabela 3 przedstawia moduły i elementy w omawianej aplikacji.

Tabela 3. Obszary wspierane przez serwis OLX. Źródło: Opracowanie własne.

Lp.	Obszar	Opis podstawowych funkcjonalności
-----	--------	-----------------------------------

1	Rejestracja i logowanie	- zarejestrowanie nowego konta w serwisie za pomocą numeru telefonu i kodu weryfikacyjnego, który przychodzi w SMS na podany numer telefonu - zalogowanie się do serwisu za pomocą numeru telefonu i kodu weryfikacyjnego - logowanie za pomocą konta Google lub Facebook
2	Kategorie	- przeglądanie kategorii znajdujących się aktualnie w serwisie - lista kategorii na stronie głównej - po kliknięciu w odnośnik, rozwijają się drzewa kategorii szczegółowych - przeglądanie ofert wypożyczenia po wybranej kategorii - wyszukiwanie ofert po nazwie przedmiotu - sortowanie po cenie lub dacie publikacji
3	Zamówienia	- przeglądanie ogłoszenia i zapoznanie się ze szczegółami zawartymi w opisie - kontakt bezpośredni z ogłaszającym - możliwość skorzystania z przesyłki OLX
4	Promocje	- brak promocji związanych z wypożyczaniem - okresowe zniżki na ogłoszenia promowane - ogłoszenia promowane widoczne są u góry wyników wyszukiwania
5	Obsługa klienta	- skrzynka wiadomości - połączenie głosowe z konsultantem
6	Panel użytkownika	- ogłoszenia użytkownika - wiadomości - płatności - szukam pracy (pilotaż dla osób, które szukają zatrudnienia) - ustawienia - twoje przesyłki (panel dla kupujących i sprzedających z dostawą OLX)

OLX.pl nie jest typowym serwisem do wypożyczeń. Wykorzystanie go do tego celu jest problematyczne, ponieważ wiele warunków trzeba opisywać, brak zautomatyzowanego wsparcia dla takiego typu ogłoszeń. Zaletą tego rozwiązania jest jego prostota oraz fakt, że jest ono częścią

popularnego portalu, więc mimo niedogodności, wciąż wielu użytkowników znających ten serwis będzie wykorzystywać OLX do wypożyczeń.

2.3. Plenti

Plenti to aplikacja webowa, która została stworzona przez firmę Plenti spółka z ograniczoną odpowiedzialnością z siedzibą w Warszawie. Aplikacja jest dostępna również na urządzenia z systemem Android i iOS.

Producent deklaruje, że aplikacja daje jej użytkownikom możliwość wypożyczania najnowszej generacji urządzeń elektrycznych takich jak konsole do gier, drony, gogle wirtualnej rzeczywistości, nawigacje samochodowe czy też ekspresy do kawy. W dodatku wypożyczenia przedmiotów odbywają się bez konieczności podpisania żadnej umowy.

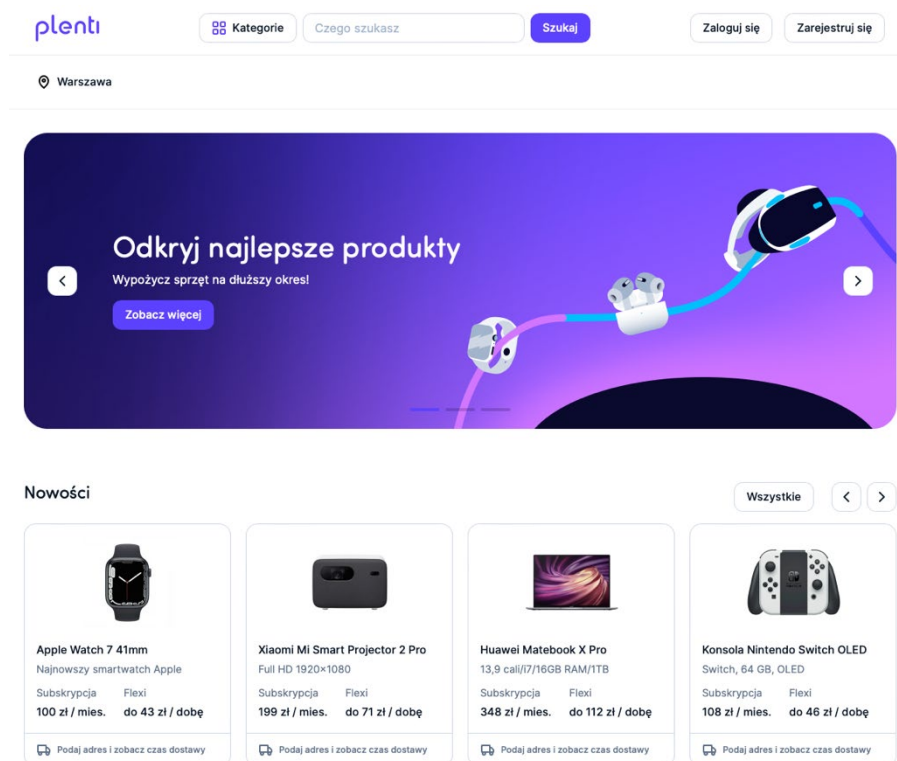
Aplikacja działa w modelu biznesowym wypożyczalni przedmiotów, których serwis jest właścicielem. Użytkownik rejestrujący się w serwisie nie ma możliwości wystawiania swoich ofert na wypożyczenie przedmiotu.

Poniżej wymienione są moduły znajdujące się w obecnej wersji serwisu:

1. Rejestracja i logowanie
2. Kategorie i wyszukiwanie
3. Zamówienia
4. Promocje na wypożyczenia
5. Obsługa klienta
6. Panel użytkownika

Do zarejestrowania się w serwisie wystarczy podanie numeru telefonu i wpisanie kodu weryfikacyjnego. Podstawowym zadaniem aplikacji jest przeprowadzenie użytkownika przez proces wypożyczenia interesującego go przedmiotu. Weryfikacja tożsamości klienta odbywa się na podstawie podanych danych takich jak imię, nazwisko, numeru telefonu i adres e-mail. Ostatnim krokiem do weryfikacji jest wykonanie swojego zdjęcia z dokumentem tożsamości i dodanie go w swoim profilu.

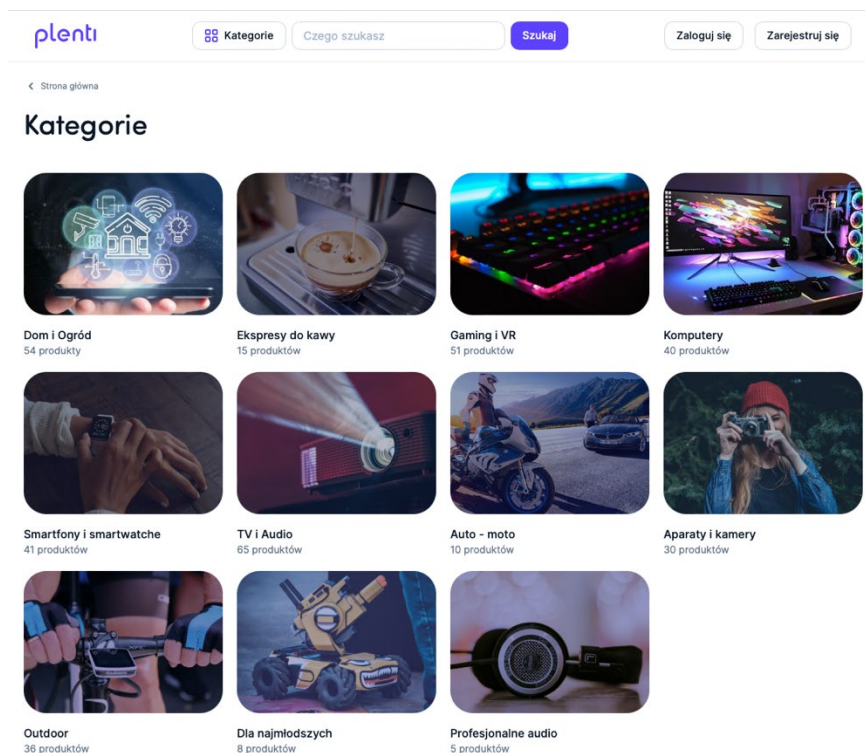
Rysunek numer 5 przedstawia widok strony głównej serwisu Plenti.



Rysunek 5. Strona główna serwisu Plenti. Źródło: Opracowanie własne.

Oprogramowanie umożliwia dobranie dodatkowych akcesoriów pasujących do wypożyczanego przedmiotu, przeprowadzenie symulacji kosztu wypożyczenia czy też zweryfikowanie tożsamości wypożyczającego mające na celu obniżenie kosztu kaucji zwrotnej. Opisane jest również szczegółowo pudełko z zawartością tego co ma zostać wypożyczone a także dokładna specyfikacja zawierająca takie parametry jak np.: waga, wymiar, format zapisu filmów, rozdzielczość filmów.

Rysunek numer 6 pokazuje widok kategorii ogłoszeń dostępnych w serwisie Plenti.



Rysunek 6. Wszystkie kategorie serwisu Plenti. Źródło: Opracowanie własne.

Program umożliwia wygodne przeglądanie kategorii i wyszukiwanie przedmiotów. Kategorie można rozwinąć jako listę z górnego menu lub zobaczyć wszystkie dostępne kategorie klikając w przycisk “Wszystkie” na stronie głównej w sekcji z kategoriami. Rysunek 5 przedstawia widok po kliknięciu w wyżej wspomniany przycisk.

Kolejną podstawową funkcjonalnością jest panel użytkownika, w którym można dokonać weryfikacji tożsamości, dodać kartę płatniczą, zweryfikować trwające wynajmy czy też sprawdzić wynajmy historyczne.

Tabela 4 przedstawia moduły i elementy w omawianej aplikacji.

Tabela 4. Obszary wspierane przez serwis Plenti. Źródło: Opracowanie własne.

Lp.	Obszar	Opis podstawowych funkcjonalności
1	Rejestracja i logowanie	- zarejestrowanie nowego konta w serwisie za pomocą numeru telefonu i kodu weryfikacyjnego, który przychodzi w SMS na podany numer - zalogowanie się do serwisu za pomocą numeru telefonu i kodu weryfikacyjnego
2	Kategorie	- przeglądanie kategorii znajdujących się aktualnie w serwisie - lista rozwijana w menu głównym ze wszystkimi kategoriami - dedykowany ekran z graficzną prezentacją kategorii - przeglądanie ofert wypożyczenia po wybranej kategorii - wyszukiwanie ofert po nazwie przedmiotu - sortowanie po cenie i/lub dostarczenia do paczkomatu
3	Zamówienia	- przeglądanie ofert wypożyczenia i zapoznanie się ze szczegółami danej oferty - symulator kosztu wynajmu z informacją cena za dobę oraz całkowity koszty wypożyczenia za wybrany okres czasu - sekcja z odpowiedziami na często zadawane pytania - podanie adresu na jaki ma być dostawa - oszacowanie dnia i godziny dostarczenia przedmiotu - wybór metody płatności - dodanie komentarza do zamówienia - możliwość wpisania kodu promocyjnego na zamówienie
4	Promocje	- przeglądanie aktualnie dostępnych promocji na karuzeli - zobaczenie wszystkich dostępnych promocji - sortowanie ogłoszeń promocyjnych
5	Obsługa klienta	- czat z pracownikiem serwisu
6	Panel	- informacje ogólne o użytkowniku oraz weryfikacja tożsamości

	użytkownika	- poleć nas, czyli indywidualny kod dla użytkownika by polecał serwis znajomym i zdobywał profity - płatności, gdzie można dodać kartę płatniczą czy też kredytową do serwisu - wynajmy, gdzie użytkownik może sprawdzić swoje trwające wynajmy - historia, w której znajdują się wszystkie zakończone wynajmy - możliwość wpisania kodu promocyjnego - zakładka z informacjami o aktualnych promocjach użytkownika - ustawienia, gdzie użytkownik może dodać swoje zdjęcie, wybrać wersję językową portalu i wybrać, czy chce dostawać maila z informacjami marketingowymi.
--	-------------	--

Podsumowując aplikacja Plenti oferuje przejrzysty i nowoczesny layout oraz jest intuicyjna w użyciu. Posiada wygodny interfejs do przeglądania zarówno kategorii jak i konkretnych ofert. Ponadto daje użytkownikowi szeroki wachlarz możliwości w zarządzaniu swoim kontem. Niemniej ilość kategorii i znajdujących się w niej produktów może zostawić niedosyt u przeciętnego fana nowoczesnych technologii. Tym bardziej, szukając takich przedmiotów jak wiertarka czy inny sprzęt remontowo budowlany, spotkamy się ze sporym zawodem.

2.4. Podsumowanie

Wszystkie przedstawione aplikacje mają na celu wypożyczenie na pewien okres przedmiotu, którego nie chcemy lub nie potrzebujemy kupować na własność. Ich podstawowymi modułami są rejestracja, logowanie, kategorie, zamówienia, obsługa klienta czy też panel użytkownika.

W przypadku rejestracji i logowania, tylko Plenti nie daje możliwości korzystania z serwisu po zalogowaniu tradycyjnie adresem e-mail, który i tak do pełnej weryfikacji trzeba podać. To samo tyczy się social media, których nie można użyć do zalogowania. A trzeba przyznać, że taka możliwość jest bardzo wygodna.

Każdy z przedstawionych serwisów daje jej użytkownikom możliwość skorzystania z jakiegoś przedmiotu. W przypadku LokPo użytkownicy samodzielnie wystawiają ogłoszenia z ofertami. Natomiast w serwisie Plenti właścicielem wszystkich ogłoszonych przedmiotów jest również właściciel serwisu co znacząco zmienia model biznesowy w porównaniu do LokPo. Z kolei w serwisie OLX, mimo innego głównego celu istnienia aplikacji i również za namową spółki, użytkownicy wykorzystują serwis

by wypożyczać sobie wzajemnie przedmioty zamiast kupować. Używają przy tym funkcjonalności przeznaczonej do sprzedaży.

We wszystkich aplikacjach brakuje możliwości dodatkowej ochrony pożyczanego przedmiotu w formie ubezpieczenia wypożyczenia, które oferuje nasz produkt. Sama kaucja często może być niewystarczającą ochroną, a warto zauważyć, że sprzedaż ubezpieczeń jest dochodowym biznesem prowizyjnym, co właśnie zakłada nasz projekt. Dlatego aplikacja stworzona w toku prac naszego zespołu ma być silnie konkurencyjnym produktem, odpowiadającym na potrzeby szerokiej grupy użytkowników zapewniającym szereg usług dodatkowych, czego brakuje w pozostałych serwisach.

3. Propozycja aplikacji obsługującej wypożyczenia

Serwis Szerownia.pl został stworzony podczas poszczególnych etapów pracy nad projektem m.in. analiza, projektowanie, implementacja oraz testowanie. Poniższy rozdział jest opisem pracy na etapie analizy oraz projektowania, znajdziemy w nim opis modułów, które zostały zaimplementowane w systemie, następnie zostały opisane wymagania funkcjonalne jak i niefunkcjonalne. W kolejnych podrozdziałach pokazane są diagramy przypadków użycia, diagramy aktywności, które opisały zakres czynności i odpowiedzialności modułów bądź użytkowników systemu oraz scenariusz wybranego przypadku użycia.

3.1. Moduły systemu

W aplikacji możemy wyróżnić kilka modułów, przeznaczonych dla różnych typów użytkowników oraz służących różnym celom.

- Moduł 1: Dla użytkownika. Przeglądanie ogłoszeń przedmiotów do wypożyczenia.
- Moduł 2: Dla użytkownika zarejestrowanego. Wystawianie ogłoszeń.
- Moduł 3: Dla użytkownika zarejestrowanego. Wypożyczanie.
- Moduł 4: Dla użytkownika zarejestrowanego. Ogłaszanie przedmiotu poszukiwanego.
- Moduł 5: Dla użytkownika zarejestrowanego. Zgłaszanie do moderacji.
- Moduł 6: Dla użytkownika zarejestrowanego. Reklamacje.
- Moduł 7: Dla użytkownika premium. Ubezpieczanie przedmiotu.
- Moduł 8: Dla moderatora. Sprawdzanie zgłoszonych treści, ukrywanie, odblokowywanie do poprawy.
- Moduł 9: Dla likwidatora. Analizowanie szkody i podejmowanie decyzji o odszkodowaniu.
- Moduł 10: Dla rzeczoznawcy. Kosztorysowanie szkody.
- Moduł 11: Dla towarzystwa ubezpieczeniowego. Zarządzanie polisami.
- Moduł 12: Dla administratora. Statystyki systemu.
- Moduł 11: Dla administratora. Zarządzanie kontami użytkowników.

3.2. Wymagania funkcjonalne

Pierwszy etap analizy biznesowej pozwolił na zidentyfikowanie aktorów systemu a także ich potrzeb jako przyszłych użytkowników systemu. Wymagania zostały przyporządkowane do opisanych w poprzednim rozdziale modułów. W kolejnym etapie prac poszczególne funkcjonalności zostały uszczegółowione. Ich opis został przedstawiony w kolejnych podrozdziałach.

3.2.1 Moduł klienta

Jak opisano powyżej, głównymi modułami są te przeznaczone dla użytkowników, klientów:

- przeglądanie ogłoszeń przedmiotów do wypożyczenia,
- wystawianie ogłoszeń, wypożyczanie,
- reklamacje czy też ubezpieczanie przedmiotu.

Poza pierwszym modułem pozostałe są dostępne wyłącznie dla użytkownika zalogowanego oraz dla posiadacza konta premium, jeśli chodzi o moduł ubezpieczeń.

W tabeli 5 przedstawiono szczegółowo wymagania funkcjonalne dotyczące użytkownika niezarejestrowanego oraz użytkowników systemu, którzy dziedziczą (użytkownik zarejestrowany, użytkownik premium). Dostęp do funkcji prezentuje się następująco:

- Zarejestrowanie się z uprzednim przeczytaniem regulaminu,
- Zalogowanie do serwisu oraz ewentualne przypomnienie hasła,
- Wyszukiwanie i przeglądanie ogłoszeń zamieszczonych,
- Zapoznanie z opiniami o użytkownikach.

Tabela 5. Wymagania funkcjonalne dla użytkownika niezarejestrowanego. Źródło: Opracowanie własne.

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Zarejestruj się w serwisie	Jako użytkownik niezarejestrowany mogę zarejestrować się	Użytkownik niezarejestrowany	Wymagane Przeczytanie i zaakceptowanie

		w serwisie by móc się zalogować.		regulaminu
2	Przeczytaj regulamin serwisu	Jako użytkownik niezarejestrowany mogę zapoznać się z regulaminem by móc go zaakceptować celem rejestracji.	Użytkownik niezarejestrowany Użytkownik zarejestrowany Użytkownik premium	Użytkownik, który się rejestruje zapoznaje się z regulaminem i akceptuje go. Użytkownik zarejestrowany może wrócić do regulaminu w każdej chwili.
3	Zaloguj się	Jako użytkownik niezalogowany mogę się zalogować do serwisu by móc korzystać z funkcjonalności dla użytkownika zarejestrowanego.	Użytkownik niezalogowany	Wymagane jest wcześniejsze zarejestrowanie się w serwisie.
4	Przypomnij hasło	Jako użytkownik niezalogowany mogę skorzystać z opcji przypomnienia hasła, którego mogłem zapomnieć.	Użytkownik niezalogowany	
5	Przeglądaj opublikowane ogłoszenia	Jako użytkownik niezarejestrowany mogę przeglądać ogłoszenia	Użytkownik niezarejestrowany Użytkownik zarejestrowany	

		opublikowane w serwisie.	Użytkownik premium	
6	Wyszukaj ogłoszenia zgodnie z zadanymi parametrami	Jako użytkownik niezarejestrowany mogę skorzystać z wyszukiwarki ogłoszeń oraz skorzystać z dostępnych filtrów by ograniczyć wyniki wyszukiwania.	Użytkownik niezarejestrowany Użytkownik zarejestrowany Użytkownik premium	
7	Zapoznaj się z opiniami o użytkownikach zarejestrowanych	Jako użytkownik niezarejestrowany mogę przeczytać opinie wystawione użytkownikowi zarejestrowanemu.	Użytkownik niezarejestrowany Użytkownik zarejestrowany Użytkownik premium	

W tabeli 6 przedstawiono szczegółowo wymagania funkcjonalne dotyczące użytkownika zarejestrowanego. Dostęp do funkcji prezentuje się następująco:

- Dokonanie wypożyczenia a następnie zwrotu przedmiotu,
- Dodanie ogłoszenia poszukiwania, zgłoszenia chęci użyczenia konkretnego przedmiotu,
- Dodanie zdjęcia do ogłoszenia,
- Pisanie wiadomości do użytkowników oraz dodawanie opinii o innym użytkowniku,
- Zgłaszanie reklamacji i naruszeń regulaminu,

- Edycja i usuwanie swoich ogłoszeń, gdzie usuwanie tak naprawdę jest ukryciem ogłoszenia w serwisie dla zwykłych użytkowników,
- Możliwość edycji swoich danych w profilu,
- Opłacanie prowizji za ogłoszenia i możliwość zakupu konta premium,
- Wystawienie ogłoszenia wypożyczenia na podstawie ogłoszenia poszukiwania.

Tabela 6. Wymagania funkcjonalne dla użytkownika zarejestrowanego. Źródło: Opracowanie własne.

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Dodaj wypożyczenia przedmiotu	Jako użytkownik zarejestrowany mogę dodać ogłoszenie wypożyczenia przedmiotu by inny użytkownik zarejestrowany mógł z niego skorzystać	Użytkownik zarejestrowany Użytkownik premium	
2	Dokonaj zwrotu przedmiotu	Jako użytkownik zarejestrowany mogę zwrócić wypożyczony przedmiot	Użytkownik zarejestrowany Użytkownik premium	By dokonać zwrotu muszę mieć trwające wypożyczenie.
3	Dodaj ogłoszenie o chęci użyczenia przedmiotu innemu użytkownikowi	Jako użytkownik zarejestrowany mogę dodać ogłoszenie użyczenia przedmiotu by zamieścić je w systemie	Użytkownik zarejestrowany Użytkownik premium	

4	Dodaj ogłoszenie o poszukiwaniu przedmiotu	Jako użytkownik zarejestrowany mogę dodać ogłoszenie poszukiwania by poinformować innych użytkowników, że jestem zainteresowany wypożyczeniem takiego przedmiotu.	Użytkownik zarejestrowany Użytkownik premium	
5	Dodaj zdjęcie do ogłoszenia	Jako użytkownik zarejestrowany mogę dodać zdjęcie do swojego ogłoszenia użyczenia i/lub ogłoszenia poszukiwania.	Użytkownik zarejestrowany Użytkownik premium	
6	Napisz wiadomość do użytkownika	Jako użytkownik zarejestrowany mogę pisać wiadomości do innych zarejestrowanych użytkowników serwisu	Użytkownik zarejestrowany Użytkownik premium	
7	Dodaj opinie o innym użytkowniku po zakończeniu wypożyczenia	Jako użytkownik zarejestrowany mogę napisać opinie o innym zarejestrowanym użytkowniku po zakończeniu wypożyczenia	Użytkownik zarejestrowany Użytkownik premium	Funkcjonalność dostępna tylko do zakończonych wypożyczeń.
8	Zgłoś reklamacje do wypożyczenia	Jako użytkownik zarejestrowany mogę zgłosić reklamacje do wypożyczenia, z którego skorzystałem	Użytkownik zarejestrowany Użytkownik premium	Funkcjonalność dostępna tylko dla trwających i zakończonych wypożyczeń.

9	Zgłoś naruszenie regulaminu	Jako użytkownik zarejestrowany mogę zgłosić naruszenie regulaminu by moderator zajął się niestosowną treścią lub ogłoszeniem.	Użytkownik zarejestrowany Użytkownik premium	
10	Usuń swoje ogłoszenie	Jako użytkownik zarejestrowany mogę usunąć swoje ogłoszenie by nie było widoczne w serwisie dla innych użytkowników	Użytkownik zarejestrowany Użytkownik premium	Usuwana jest jedynie widoczność w serwisie, samo ogłoszenie jest przechowywane w bazach danych serwisu.
11	Edytuj swoje ogłoszenie	Jako użytkownik zarejestrowany mogę edytować swoje ogłoszenie by móc nanieść poprawki i/lub zastosować się do wytycznych moderatora.	Użytkownik zarejestrowany Użytkownik premium	
12	Zmień dane na swoim profilu	Jako użytkownik zarejestrowany mogę edytować, dodawać, usuwać dane umieszczone w moim profilu użytkownika.	Użytkownik zarejestrowany Użytkownik premium	
13	Opłać naliczoną prowizję	Jako użytkownik zarejestrowany mogę dokonać opłacenia naliczonej prowizji za wypożyczenia, które opublikowałem w serwisie.	Użytkownik zarejestrowany Użytkownik premium	Prowizja jest naliczana do wypożyczeń, które się zakończyły.

14	Dokonaj zakupu konta premium	Jako użytkownik zarejestrowany mogę kupić dostęp do konta premium by skorzystać z funkcjonalności dla niego przeznaczonych	Użytkownik zarejestrowany Użytkownik premium	Będąc użytkownikiem premium mogę dokonać ponownego zakupu jako przedłużenie subskrypcji.
15	Klonuj ogłoszenie poszukiwania przedmiotu	Jako użytkownik zarejestrowany mogę stworzyć ogłoszenie wypożyczenia na podanie ogłoszenia poszukiwania by umożliwić wypożyczenie przedmiotu	Użytkownik zarejestrowany Użytkownik premium	

W tabeli 7 przedstawiono szczegółowo wymagania funkcjonalne dotyczące użytkownika zarejestrowanego, który wykupił dostęp do konta premium. Dostęp do funkcji prezentuje się następująco:

Tabela 7. Wymagania funkcjonalne dla użytkownika premium. Źródło: Opracowanie własne

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Wykup dodatkowe ubezpieczenie wypożyczenia	Jako użytkownik premium mogę wybrać opcje wypożyczenia z dodatkowym ubezpieczeniem.	Użytkownik premium	
2	Zgłoś szkodę do ubezpieczenia	Jako użytkownik premium mogę zgłosić szkodę do wypożyczenia, na które wcześniej kupiłem ubezpieczenie.	Użytkownik premium	Wypożyczenie musi mieć wcześniej wykupione ubezpieczenie by móc zgłosić szkodę.

3	Sprawdź status likwidowanej szkody	Jako użytkownik premium mogę wejść w formularz zgłoszonej szkody, by sprawdzić na jakim etapie likwidacji jest zgłoszona szkoda.	Użytkownik premium	
4	Zadaj pytanie likwidatorowi	Jako użytkownik premium mogę napisać wiadomość spod formularza zgłoszonej szkody do likwidatora celem zadania pytania.	Użytkownik premium	
5	Dodaj ogłoszenie promowane	Jako użytkownik premium mogę wystawić ogłoszenie, które będzie wyświetlane zawsze na samej górze wyszukiwań pasujących wyników. Ogłoszenie znajduje się w specjalnej sekcji na stronie głównej.	Użytkownik premium	

3.2.2 Moduł pracownika serwisu

Moduł pracownika serwisu stanowi punkt wyjścia do komunikacji z klientem oraz rozszerzenia atrakcyjności usług portalu. Funkcjonalności w dużej mierze dotyczą zakresu ubezpieczenia wypożyczeń i likwidacji szkód. Jednakże zawierają również akcje niezbędne do czuwania nad jakością treści zamieszczanych w serwisie i bezpieczeństwa użytkowników.

W Tabeli 8 przedstawiono szczegółowo wymagania funkcjonalne dotyczące procesu obsługi zgłoszeń szkód i ich likwidacji.

Tabela 8. Wymagania funkcjonalne dla użytkownika likwidatora szkód i rzeczoznawcy.

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Przekaż zgłoszenie szkody do TU	Jako likwidator szkód mogę przekazać zgłoszenie szkody do towarzystwa ubezpieczeniowego, w którym jest zawarta polisa celem nadania numeru szkody.	Likwidator	
2	Zleć oględziny do rzeczoznawcy	Jako likwidator szkód mogę wysłać zlecenie wykonania oględzin do rzeczoznawcy w celu wykonania oględzin i kosztorysu naprawy.	Likwidator	Szkoda musi mieć nadany numer szkody z TU
3	Uzupełnij kwotę wyceny szkody/dodaj kosztorys/ dodaj komentarz odnośnie do danych użytkownika	Jako likwidator szkód mogę uzupełnić formularz z danymi po wykonaniu oględzin, by móc kontynuować proces likwidacji szkody.	Likwidator	
4	Wyślij wiadomość do użytkownika	Jako likwidator szkód mogę napisać i wysłać wiadomość do użytkownika będącego stroną w procesie likwidacji szkody.	Likwidator	

5	Zmień status szkody	Jako likwidator szkód mogę zmieniać statusy likwidowanej szkody, by w łatwy sposób zrozumieć, na jakim etapie prac jest likwidowana szkoda.	Likwidator	
6	Zleć wypłacenie odszkodowania do TU	Jako likwidator szkód mogę wysłać zlecenie wypłaty odszkodowania dla klienta w celu rozliczenia i zamknięcia szkody.	Likwidator	
7	Przyjmij zlecenie na oględziny	Jako rzeczoznawca mogę zaakceptować otrzymane zlecenie na wykonanie oględzin.	Rzeczoznawca	
8	Odrzuć zlecenie na oględziny	Jako rzeczoznawca mogę odrzucić otrzymane zlecenie na wykonanie oględzin z obowiązkowym podaniem powodu.	Rzeczoznawca	
9	Uzupełnij formularz z oględzin	Jako rzeczoznawca mogę uzupełnić formularz z oględzin by przesłać go do likwidatora.	Rzeczoznawca	
10	Wyślij wiadomość do użytkownika	Jako rzeczoznawca mogę wysłać wiadomość do użytkownika w celu umówienia się na oględziny.	Rzeczoznawca	

11	Prześlij kosztorys do szkody	Jako rzeczoznawca mogę przesłać sporządzony kosztorys naprawy dla przedmiotu, na którym były wykonane oględziny.	Rzeczoznawca	
12	Prześlij zdjęcia do szkody	Jako rzeczoznawca mogę wysłać do likwidatora zdjęcia uszkodzonego przedmiotu, którego wykonywałem oględziny.	Rzeczoznawca	
13	Zrobienie zdjęcia	Jako rzeczoznawca mogę zrobić zdjęcia przedmiotowi szkody by wysłać je do likwidatora.	Rzeczoznawca Mobile	Akcja dostępna jedynie na urządzeniu mobilnym z aparatem.

W tabeli 9 przedstawiono szczegółowo wymagania funkcjonalne dotyczące moderowania treści w serwisie, rozpatrywania reklamacji i bezpieczeństwa aplikacji.

Tabela 9. Wymagania funkcjonalne dla moderatora i administratora systemu.

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Blokowanie konta użytkownika	Jako moderator mogę zablokować konto użytkownikowi w ramach kary za złamanie regulaminu.	Moderator	
2	Odblokowanie konta użytkownika	Jako moderator mogę odblokować konto użytkownika, które wcześniej zostało zablokowane.	Moderator	Funkcjonalność dostępna tylko dla konta, które wcześniej zostało zablokowane.

3	Weryfikowanie treści ogłoszenia	Jako moderator mogę przeglądać, czytać ogłoszenia zamieszczane w serwisie w celu weryfikacji treści, by stwierdzić czy ogłoszenie powinno być opublikowane czy usunięte z serwisu.	Moderator	
4	Usuwanie ogłoszenia niezgodnego z regulaminem	Jako moderator mogę usuwać ogłoszenia z serwisu, które naruszają regulamin.	Moderator	
5	Zablokuj adres IP użytkownika	Jako admin mogę zablokować adres IP użytkownika, który może być zagrożeniem dla bezpieczeństwa serwisu i innych użytkowników.	Admin	
6	Usuń konto użytkownika	Jako admin mogę usunąć konto użytkownika, które jest nieaktywne w serwisie lub narusza regulamin serwisu.	Admin	
7	Edytuj dane użytkownika	Jako admin mogę edytować dane w koncie użytkownika, które naruszają regulamin serwisu.	Admin	
8	Dodaj nowego użytkownika wewnętrznego	Jako admin mogę dodać konto użytkownika wewnętrznego dla zatrudnionego pracownika serwisu.	Admin	

9	Zarządzaj uprawnieniami użytkownika	Jako admin mogę nadawać, edytować lub odbierać uprawnienia użytkownikom wewnętrznym serwisu, by mieli dostęp do funkcjonalności zgodny z zajmowanym stanowiskiem.	Admin	
10	Zarządzanie kategorią przedmiotów razem z parametrami	Jako admin mogę dodawać, edytować lub usuwać kategorie dla przedmiotów razem z parametrami, by dbać o porządek w ogłoszeniach zamieszczanych w serwisie.	Admin	

3.2.3 Moduł automatycznych funkcjonalności systemu

Ostatni, trzeci moduł dotyczy funkcjonalności, które system wykonuje automatycznie. W fazie analizy biznesowej udało się wyodrębnić powtarzalne czynności. Zamiast przypisywać nowe zadania użytkownikom wewnętrznym zdecydowaliśmy się na automatyzację tych procesów.

Tabela 10 przedstawia zadania wykonywane w określonym czasie.

Tabela 10. Wymagania funkcjonalne dla podsystemu czasu. Źródło: Opracowanie własne.

Lp.	Nazwa	Opis	Aktorzy	Ograniczenia
1	Zrób zestawienie statystyk ogłoszeń dla użytkownika za poprzedni miesiąc	System w sposób automatyczny na początku nowego miesiąca robi zestawienie statystyk ogłoszeń dla każdego użytkownika za poprzedni miesiąc.	Podsystem czasu	
2	Nalicz prowizje za wypożyczenia	System w sposób automatyczny po rozpoczęciu każdego wypożyczenia nalicza prowizje należną serwisowi za wystawienie ogłoszenia.	Podsystem czasu	Prowizja naliczana jest tylko za ogłoszenia, z których skorzystał inny użytkownik
3	Zweryfikuj opłacenie konta premium	System w sposób automatyczny sprawdza, czy użytkownik opłacił konto premium przed wygaśnięciem aktywnej subskrypcji by wiedzieć czy zablokować użytkownikowi dostęp do funkcjonalności.	Podsystem czasu	
4	Odbierz użytkownikowi konto premium	System w sposób automatyczny blokuje dostęp do funkcjonalności użytkownika premium, gdy minie czas opłaconej subskrypcji, a weryfikacja	Podsystem czasu	

		opłacenia kolejnego okresu zakończyła się negatywnie.		
5	Wyślij wiadomość, powiadomienie do użytkownika	System w sposób automatyczny będzie wysyłać wiadomości e-mail, powiadomienia do użytkownika, które mają stałą treść i moment wysłania.	Podsystem czasu	

3.3. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne definiują w jaki sposób powinna działać nasza aplikacja za pomocą dokładnego opisu lub konkretnej jednostki miary. Dzięki temu jesteśmy w stanie dostosować aplikację do pożądanego działania. Wymagania niefunkcjonalne są równie istotne, co wymagania funkcjonalne.

3.3.1 Wymagania dostępności

- Aplikacja powinna być dostępna dla użytkowników przez całą dobę.
- Czas bez awarii systemu nie powinien być krótszy niż 90 dni.
- Konserwacja oraz aktualizacje systemu powinny być przeprowadzane w godzinach nocnych.
- Czas niedostępności aplikacji w przypadku konserwacji lub aktualizacji nie powinien przekroczyć 25 min.
- Aplikacja webowa powinna wspierać wszystkie wersje przeglądarek internetowych takich jak:
 - Internet Explorer 10 i wyższa
 - Firefox wersja 94.0.2 i wyższa
 - Google Chrome wersja 96.0 i wyższa
 - Opera wersja 81.0 i wyższa
 - Safari wersja 13.1.1 i wyższa

3.3.2 Wymagania wydajności

- Aplikacja powinna być dostępna na terenie Polski.
- Maksymalny czas odpowiedzi powinien być mniejszy niż 3 sekundy.
- Aplikacja powinna być przygotowana na obsługiwanie 5000 użytkowników jednocześnie.

3.3.3 Wymagania bezpieczeństwa

- System zapewnia dostęp do kontaktu telefonicznego tylko zalogowanym osobom.
- System zapewnia dostęp administratorom do niejawnych danych po autoryzacji.
- Dane użytkowników są przechowywane w bazie danych w zaszyfrowanej formie jako hasz danych.

3.3.4 Wymagania użyteczności

- Wielkość aplikacji webowej powinna dostosowywać się do ekranu użytkownika.
- Kolorystyka aplikacji powinna być czytelna dla użytkownika.
- Rozmiar czcionki w aplikacji powinien dostosowywać się do ekranu użytkownika.
- Wielkość aplikacji mobilnej powinna być dostosowana do rozmiarów telefonów z systemem Android.

3.3.5 Wymagania etyczne

- Aplikacja nie wykorzystuje danych osobowych użytkowników w celach marketingowych.

3.3.6 Wymagania tworzenia kopii zapasowych

- Kopie zapasowe danych systemu powinny być odpowiednio zabezpieczone przed nieautoryzowanym dostępem.
- Kopie zapasowe danych powinny być przechowywane w sposób niezagrażający im w przypadku utraty możliwości korzystania z domyślnego serwera.

3.4. Aktorzy systemu

Aktorów systemu możemy podzielić na użytkowników zewnętrznych oraz administracyjnych. Do pierwszej z wymienionych grup zaliczamy:

- użytkownik niezarejestrowany,
- użytkownik zarejestrowany,
- użytkownik premium.

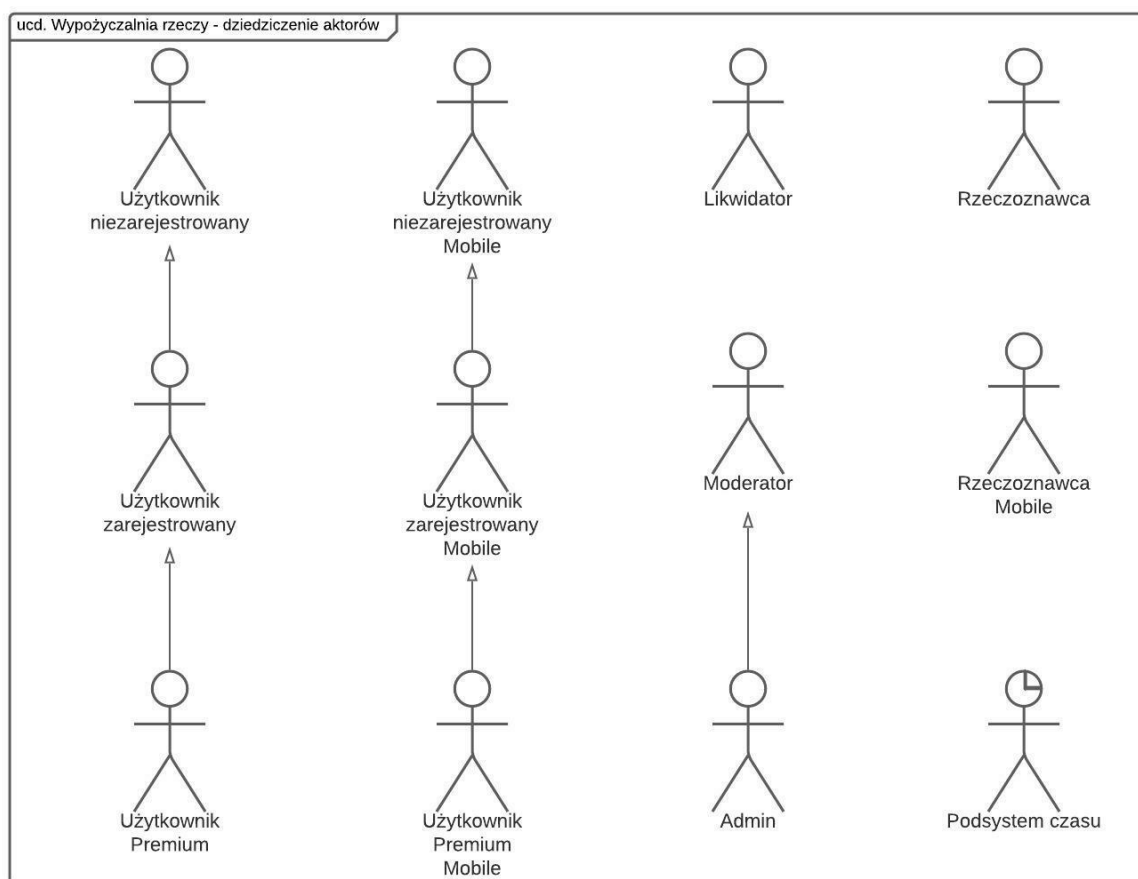
Należy zaznaczyć, że dla aplikacji mobilnej, która powstawała równolegle, aktorzy systemu zostali wyodrębnieni z uwagi na kilka dodatkowych funkcjonalności. Ich nazwy są dokładnie takie same, tylko na końcu dodano słowo “Mobile”.

Pracownicy obsługujący użytkowników i administratorzy serwisu zostali zidentyfikowani jako:

- moderator,
- administrator,
- likwidator,
- rzeczoznawca.

Dla aktorów zaliczających się do grupy pracowników serwisu tylko rzeczoznawca otrzymał swój odpowiednik na aplikację mobilną.

Na rysunku 7 mamy przedstawionych wszystkich aktorów systemu wraz z uwzględnieniem dziedziczenia. Został także uwzględniony podsystem czasu, który jest odpowiedzialny za sterowanie cyklicznie wykonującymi się funkcjonalnościami.

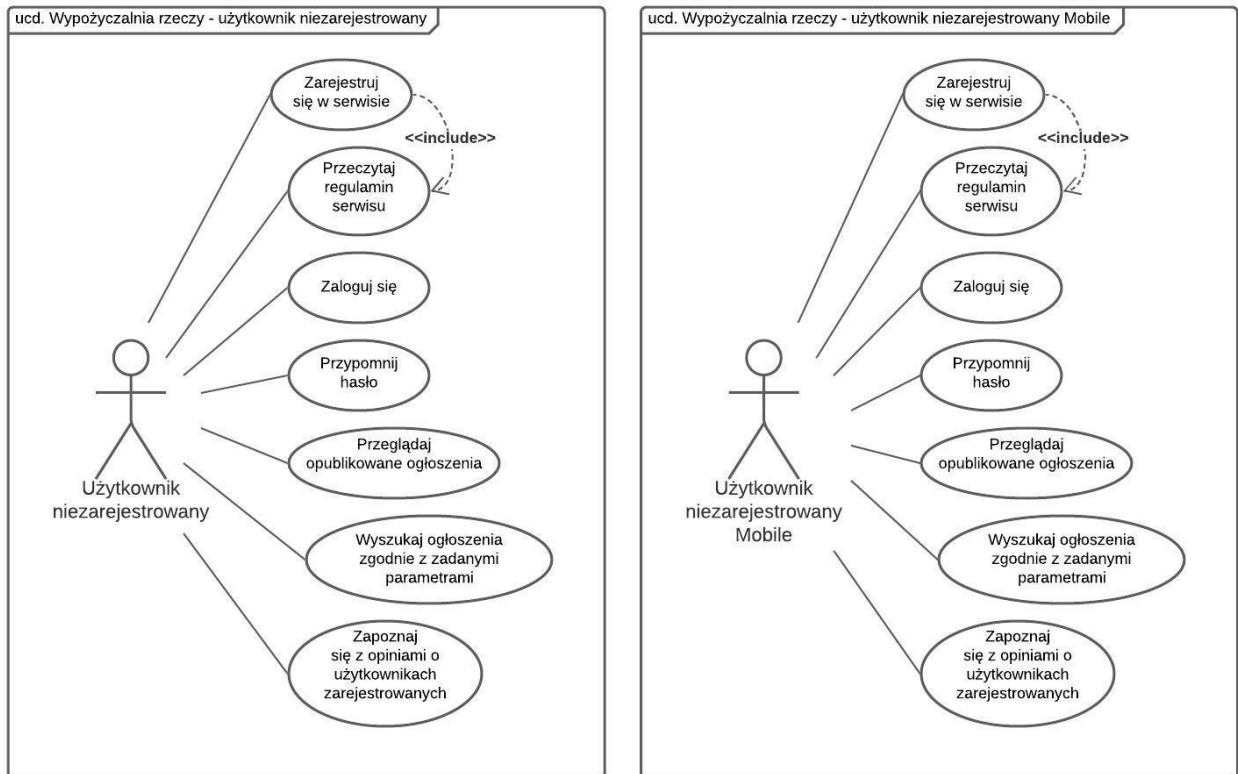


Rysunek 7. Diagram przypadków użycia dla systemu Szerownia.pl. Źródło: Opracowanie własne.

3.5. Przypadki użycia

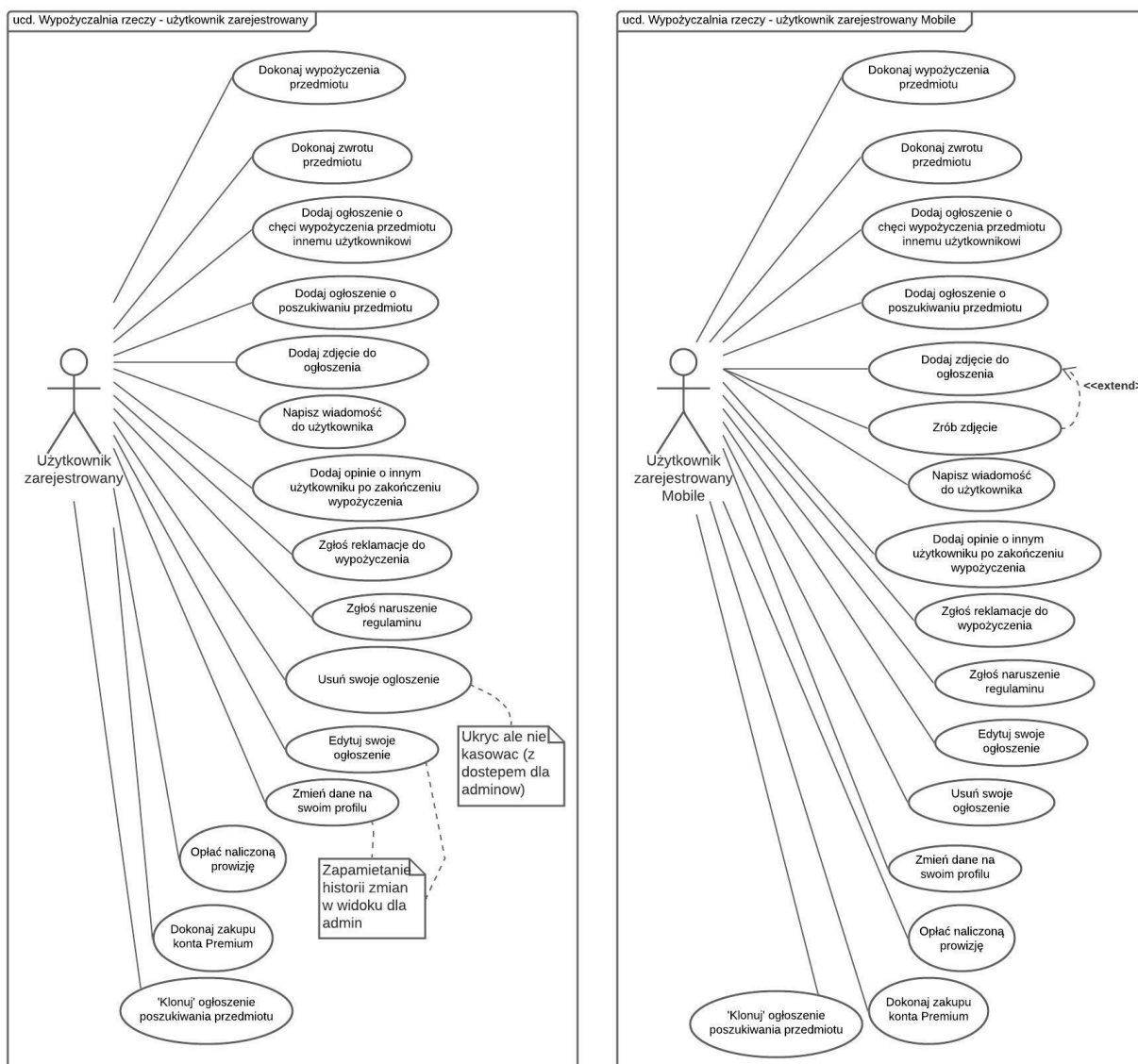
W tym rozdziale zostaną omówione, przygotowane przez zespół analityczny, diagramy przypadków użycia. Dla każdego aktora został narysowany oddzielny diagram z uwagi na bardzo dużą ilość funkcjonalności oraz dla lepszej czytelności.

Rysunek 8 przedstawia funkcjonalności dostępne dla użytkownika niezarejestrowanego.



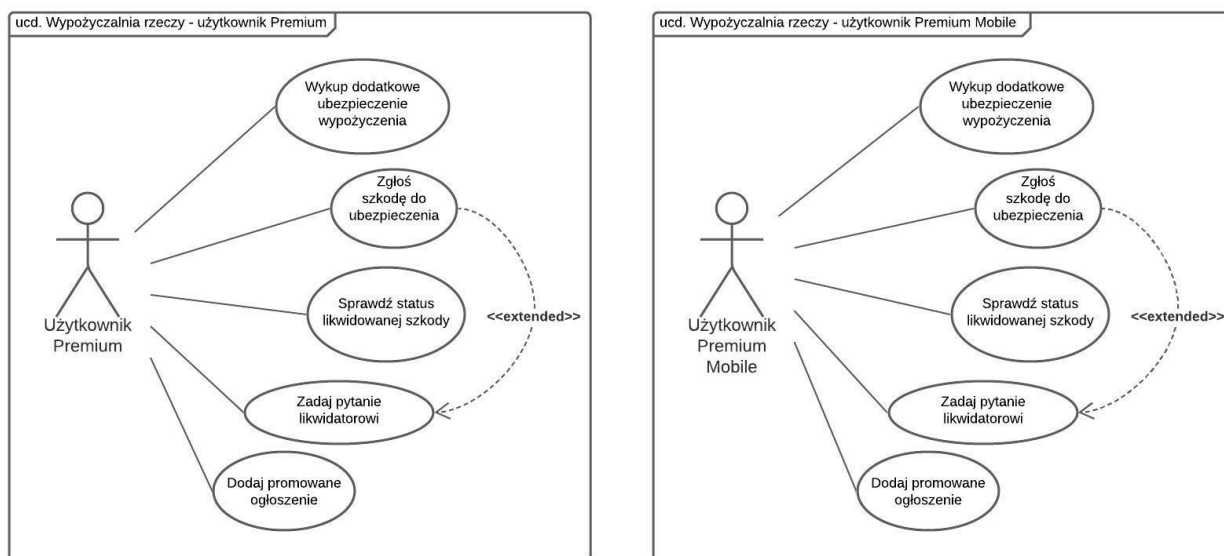
Rysunek 8. Diagram przypadków użycia dla użytkownika niezarejestrowanego. Źródło: Opracowanie własne.

Funkcjonalności dla użytkownika, który zarejestrował się w serwisie przedstawia Rysunek 9.



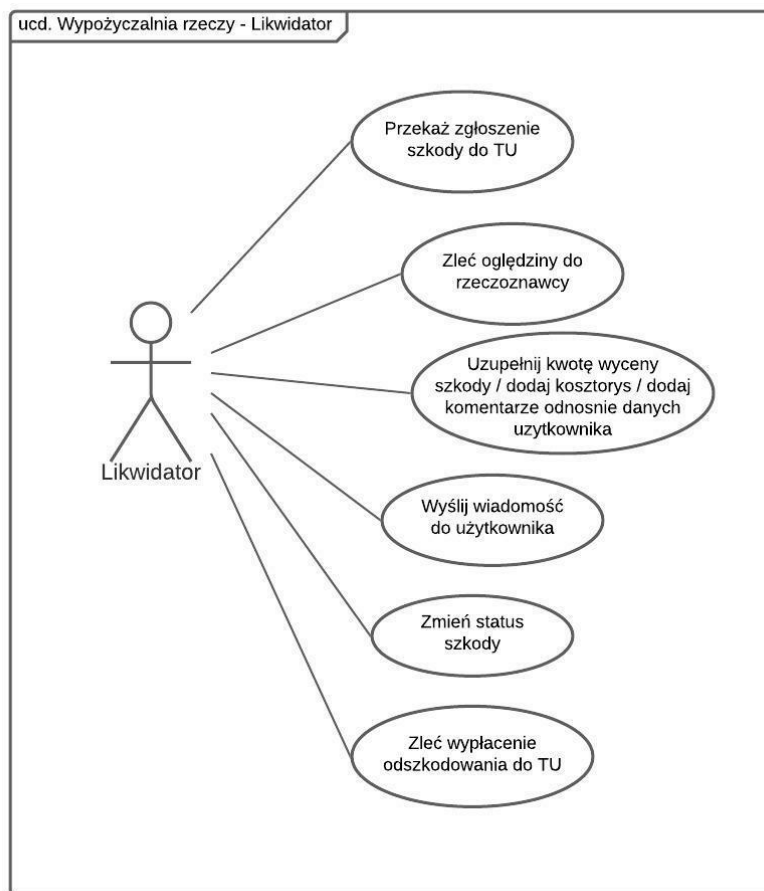
Rysunek 9. Diagram przypadków użycia dla użytkownika zarejestrowanego. Źródło: Opracowanie własne.

Rysunek 10 przedstawia, do jakich funkcjonalności zyska dostęp użytkownik, który zdecyduje się na zakup konta premium.



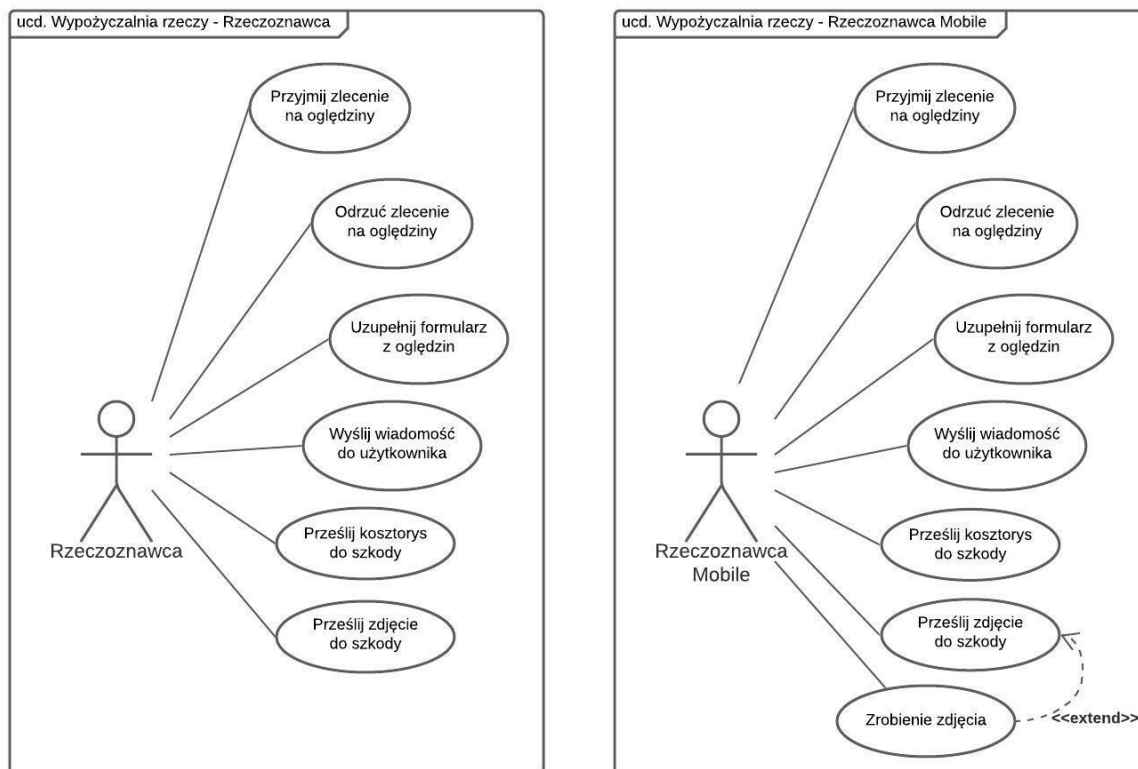
Rysunek 10. Diagram przypadków użycia dla użytkownika premium. Źródło: Opracowanie własne.

Rysunek 11 przedstawia możliwości likwidatora.



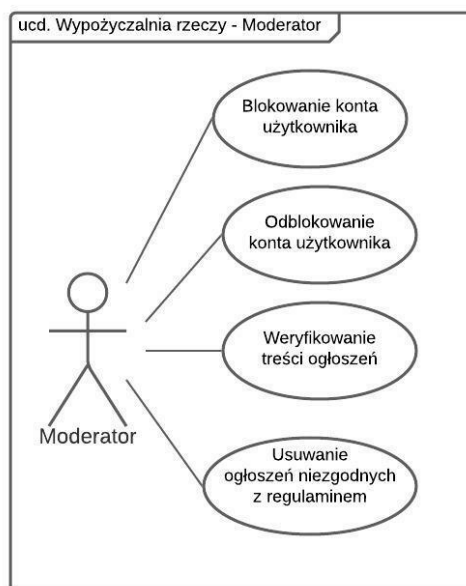
Rysunek 11. Diagram przypadków użycia dla likwidatora. Źródło: Opracowanie własne.

Funkcjonalności dla rzeczoznawcy oraz rzeczoznawcy mobilnego przedstawia Rysunek 12.



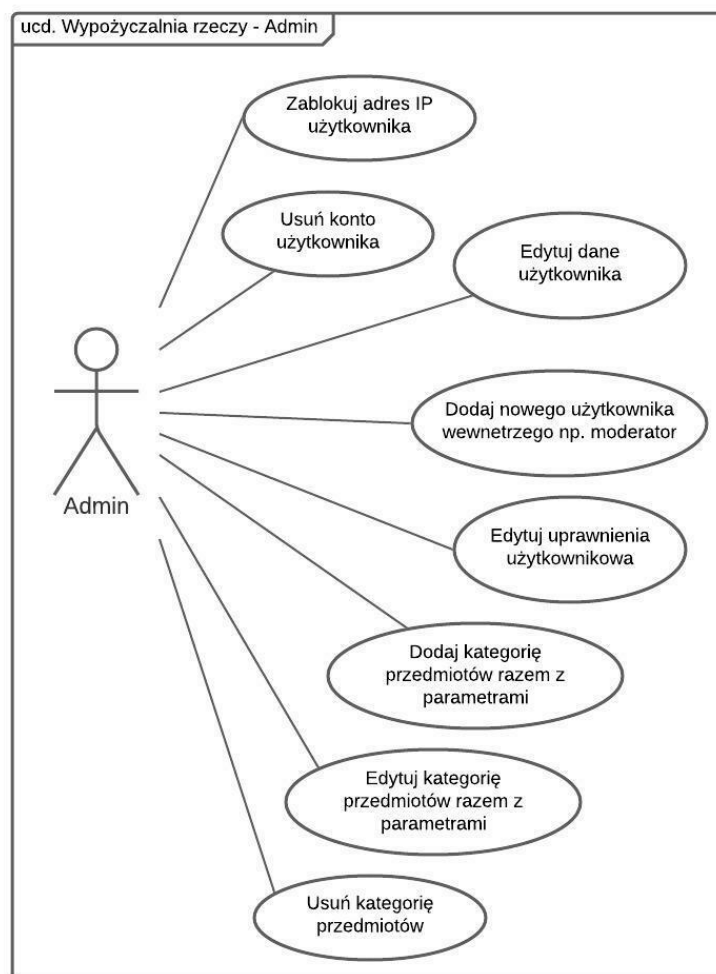
Rysunek 12. Diagram przypadków użycia dla rzeczoznawcy. Źródło: Opracowanie własne.

Rysunek 13 przedstawia przypadki użycia w kontekście aktora jakim jest moderator.



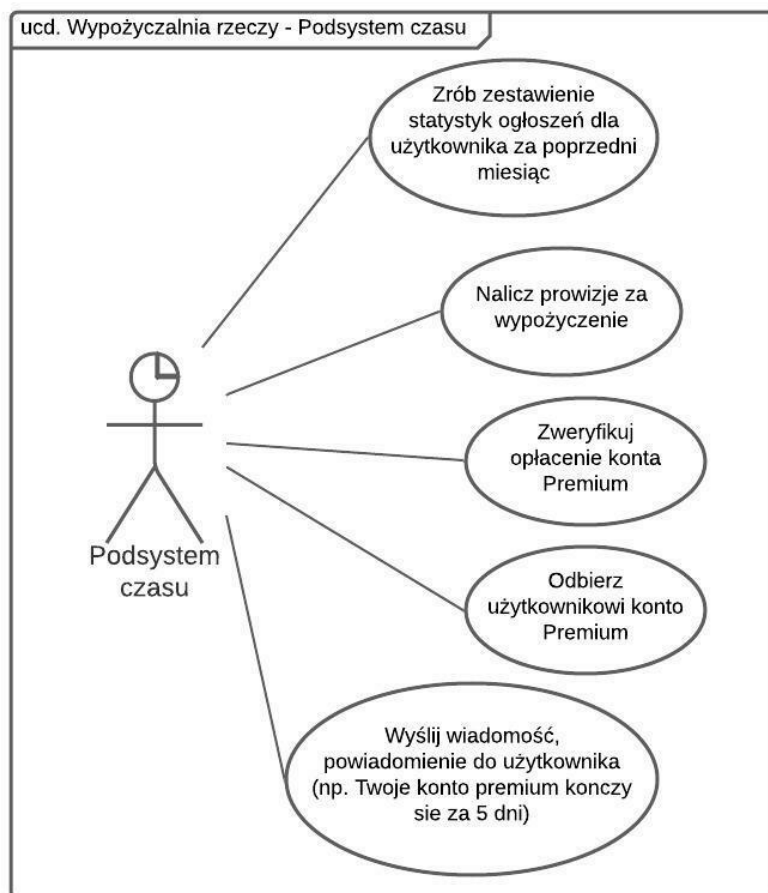
Rysunek 13. Diagram przypadków użycia dla moderatora. Źródło: Opracowanie własne.

Rysunek 14 przedstawia możliwości jakie posiada Admin.



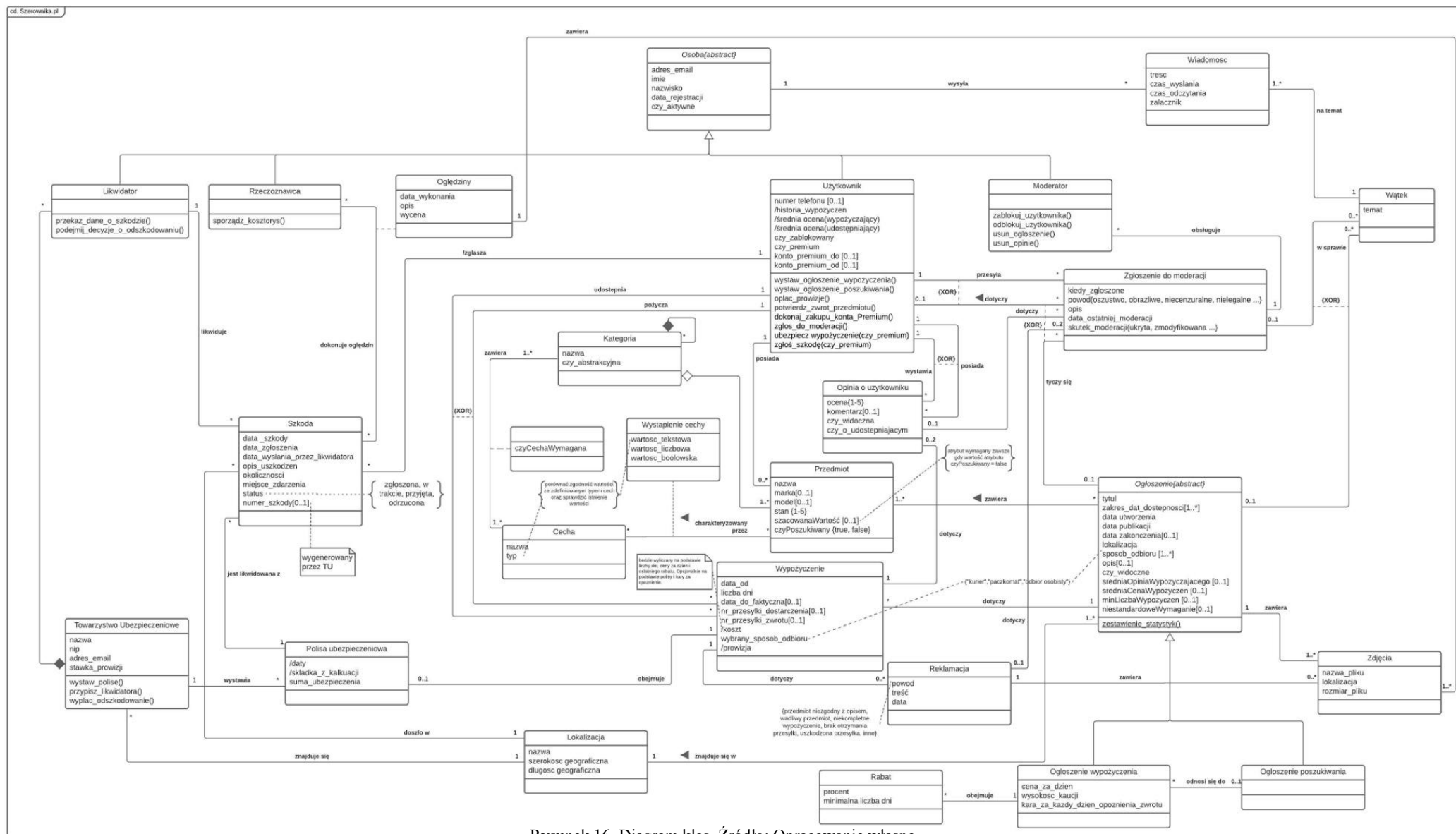
Rysunek 14. Diagram przypadków użycia dla admina. Źródło: Opracowanie własne.

Na rysunek numer 15 przedstawiony został podsystem czasu.



Rysunek 15. Diagram przypadków użycia dla podsystemu czasu. Źródło: Opracowanie własne.

3.6. Diagram klas



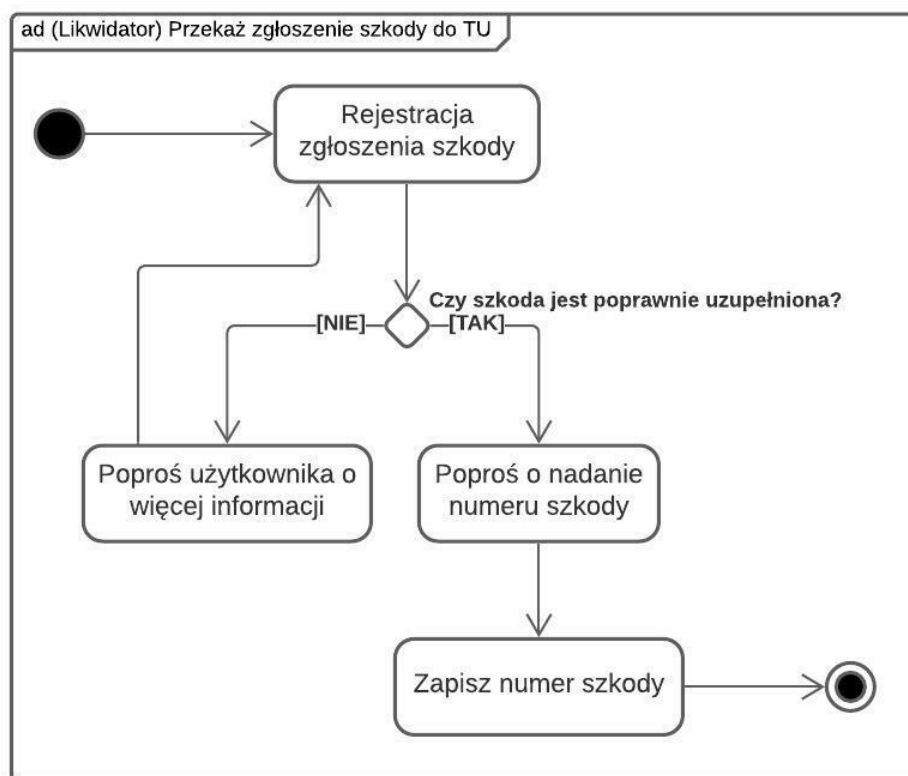
Rysunek 16. Diagram klas. Źródło: Opracowanie własne.

3.7. Diagramy aktywności

Dla wybranych, najistotniejszych przypadków użycia przygotowano diagramy aktywności dotyczące poszczególnych aktorów systemu.

3.7.1 Likwidator – przekaz zgłoszenie szkody do TU

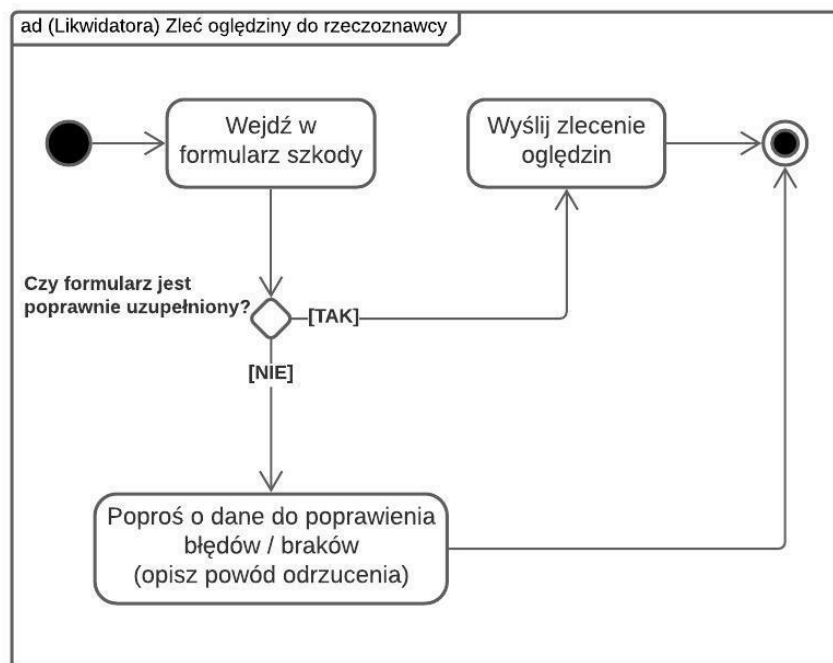
Na rysunku numer 17 jest przedstawiony diagram aktywności dla aktora Likwidator z przypadkiem użycia przekaz zgłoszenie szkody do TU.



Rysunek 17. Diagram aktywności, przekaz zgłoszenie szkody do TU. Źródło: Opracowanie własne.

3.7.2 Likwidator – zleć oględziny do rzeczoznawcy

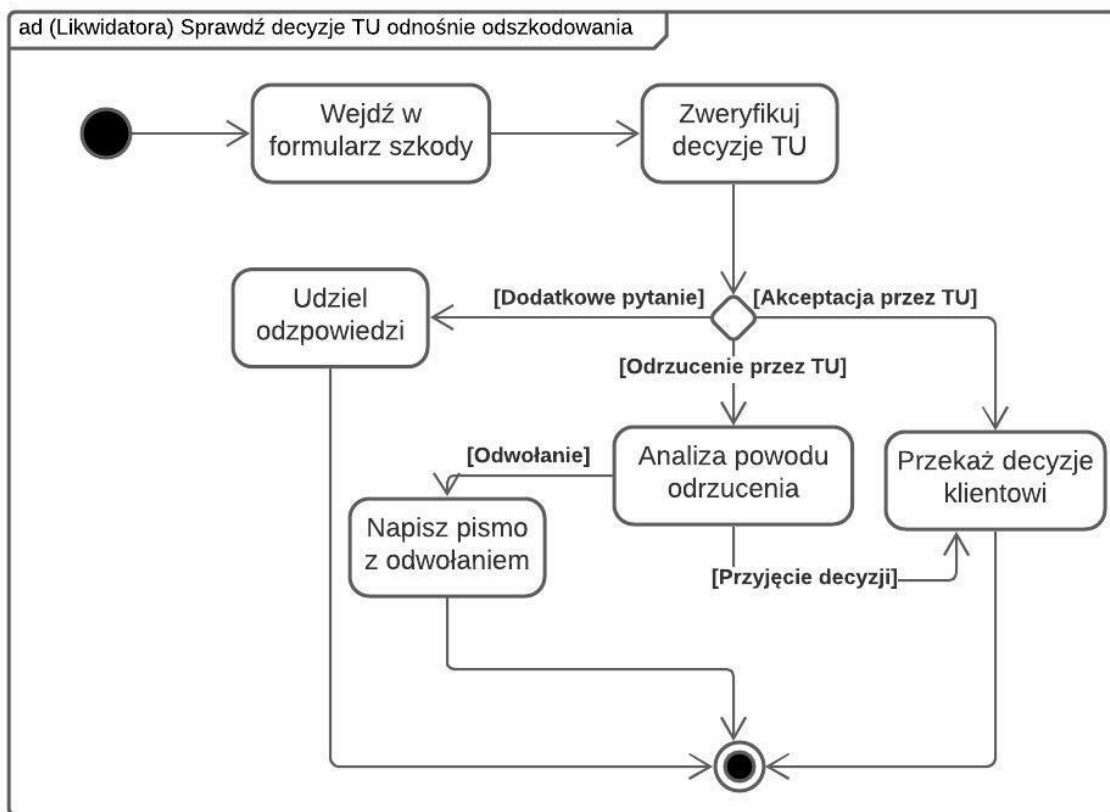
Na rysunku numer 18 jest przedstawiony diagram aktywności również dla aktora Likwidator z przypadkiem użycia zleć oględziny do rzeczoznawcy.



Rysunek 18. Diagram aktywności, zleć oględziny do rzeczoznawcy. Źródło: Opracowanie własne.

3.7.3 Likwidator – sprawdzenie decyzji TU.

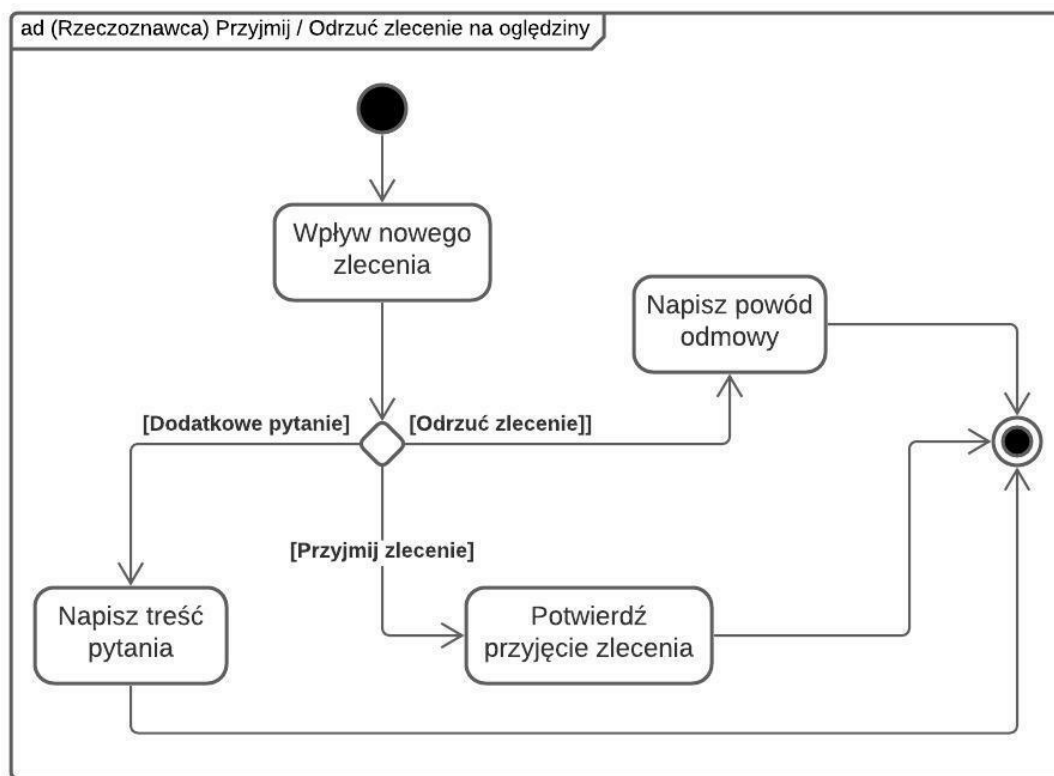
Na rysunku 19 jest przedstawiony diagram aktywności dla Likwidatora z przypadkiem użycia sprawdź decyzje TU odnośnie do odszkodowania.



Rysunek 19. Diagram aktywności, sprawdź decyzje TU odnośnie do odszkodowania. Źródło: Opracowanie własne.

3.7.4 Rzeczoznawca – Przyjmij / odrzuć zlecenie oględzin.

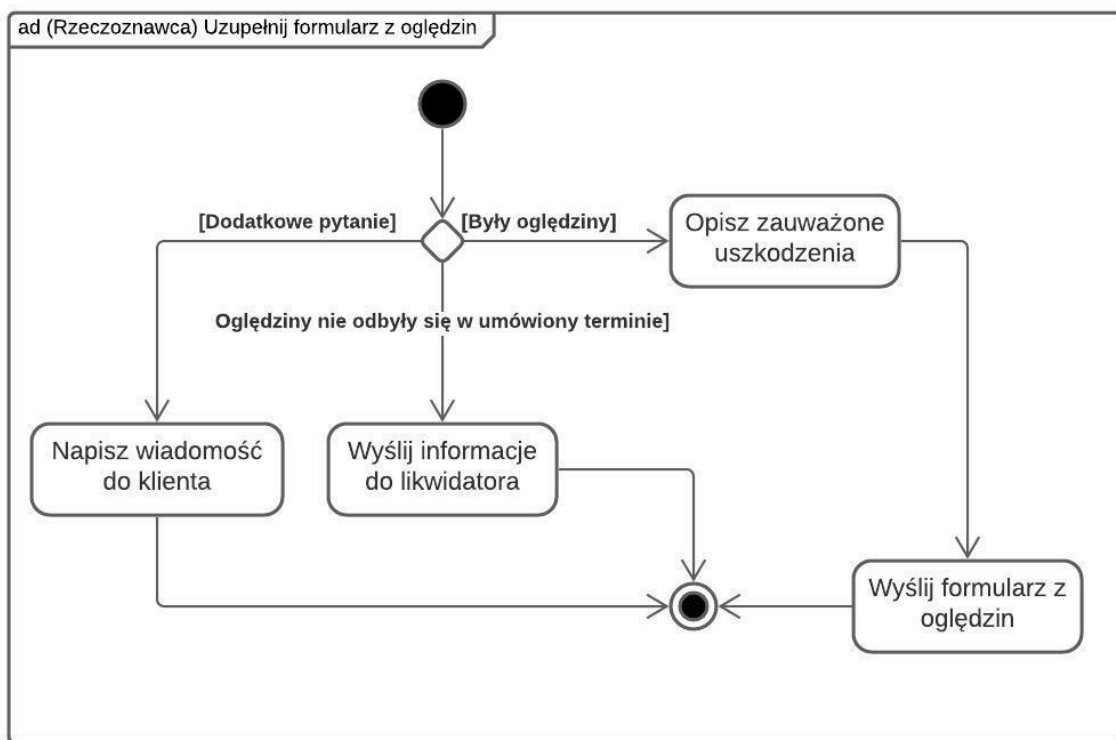
Na rysunku numer 20 przedstawiony jest diagram aktywności dla aktora rzeczoznawca z przypadkiem użycia przyjmij / odrzuć zlecenie na oględziny.



Rysunek 20. Diagram aktywności, przyjmij / odrzuć zlecenie na oględziny. Źródło: Opracowanie własne.

3.7.5 Rzeczoznawca – uzupełnij formularz z oględzin

Na rysunku 21 przedstawiony jest kolejny diagram aktywności dla rzeczoznawcy z przypadkiem użycia uzupełnij formularz z oględzin.



Rysunek 21. Diagram aktywności, uzupełnij formularz z oględzin. Źródło: Opracowanie własne.

3.8. Omówienie wybranego przypadku użycia

Aplikacja jest zbudowana z wielu modułów, które stanowią integralną całość aplikacji realizujące wszystkie funkcjonalności. Głównym jest wypożyczanie, ponieważ to na nim właśnie opiera się cała idea aplikacji. Następnym bardzo ważnym modułem jest Zakup konta premium, który jest istotną biznesowo funkcjonalnością. Użytkownik takiego konta dostaje dostęp do ubezpieczeń przedmiotów. W tabeli numer 11 został przedstawiony przepływ zdarzeń i scenariusz dla przypadku użycia - zakup konta premium.

Tabela 11. Przypadek użycia zakupu konta premium wraz ze scenariuszem. Źródło: Opracowanie własne.

Nazwa przypadku użycia	Zakup konta premium	Id.
		01
Opis	Zakupienie konta premium przez użytkownika zarejestrowanego	
Aktor główny	Użytkownik zarejestrowany	

Aktor pomocniczy	Brak
Warunki początkowe	Użytkownik posiada aktywne konto w systemie
Warunki końcowe	Użytkownik posiada konto premium
Scenariusz	
Krok	Akcja
1	Aktor uruchamia przypadek użycia Zakup konta premium
2	Aktor wybiera pakiet premium, który chce kupić
3	Aktor dokonuje płatności za wybrany pakiet konta premium
4	System informuje użytkownika o poprawnym przeprocesowaniu zakupu
5	System kończy przypadek użycia
Alternatywny przepływ zdarzeń	
4a	System informuje użytkownika o błędzie podczas procesowania płatności
4b	System pyta użytkownika czy chce podjąć kolejną próbę opłacenia pakietu, jeśli tak to zacznij od kroku numer 2.

4. Technologie i narzędzia wykorzystane w pracy

W trakcie tworzenia projektu zostały wykorzystane różne technologie i narzędzia. Ich opis znajduje się w tym rozdziale.

4.1. Java

Na początku technologia użyta do implementacji części backendowej projektu dostała nazwę Oak jednak ostatecznie nazwaną ją Java w 1995 r. na konferencji SunWorld w San Francisco. Sun Microsystems przedstawił wtedy pierwszą publiczną wersję tego języka programowania [6]. Wydawanie nowych wersji można było zaobserwować na poziomie raz na pół roku. W 2019 roku została wydana już 16 wersja Java. W naszym projekcie korzystamy z Java SE 11, z września 2018 r.

Gdy została wydana wersja 9, język Java stał się bardzo dojrzałym konkurentem innych popularnych języków programowania. C++, Python czy Ruby powoli ustępowały miejsc na podium.

Klasy i biblioteki do jakich oferuje dostęp są w stanie zaspokoić potrzeby technologiczne wielu programistów i projektów. Przykładami są:

- Interfejsy i klasy wewnętrzne,
- Obsługa wyjątków,
- Wielowątkowość,
- Obiekty rozproszone,
- Klasy zbiorcze,
- Internacjonalizacja

Niewątpliwie największą zaletą jest niezależne od platformy środowisko programistyczne, w którym uruchamiane są aplikacje. Dzięki wirtualnej maszynie Javy ten sam kod można spokojnie uruchomić pod Windows, Linux, Macintosh czy Solaris. Kolejną jest wyeliminowanie przez Javę konieczności ręcznej alokacji i dealokacji pamięci, można powiedzieć, że pamięć w Javie jest ‘sprzątana’ automatycznie. Nie trzeba się zamartwiać jej uwalnianiem.

4.2. Spring

Spring jest narzędziem do pisania aplikacji w języku Java. Rozwojem tego popularnego frameworku zajmuje się firma Pivotal. Nie jest to nowe rozwiązanie. Już w 2003 r. pojawiła się jego pierwsza wersja, a w 2005 r. pobrano go około 400 000 razy, co świadczy o błyskawicznie rosnącej popularności. Budowa aplikacji w J2EE okazała się być złożona pod względem standardów, co spowodowało zapotrzebowanie na stworzenie narzędzia takiego jak Spring [7].

Spring zawiera szablony, z których z których programista dobiera właściwe dla potrzeb konkretnego przedsięwzięcia. Najczęściej używane rozwiązania z tego frameworka to:

- Kontener IoC, który zapewnia zarządzanie konfiguracją w oparciu o odwrócenie sterowania,
- Szablon programowania aspektowego – dotyczy między innymi zarządzania transakcjami,
- Szablon dostępu do danych – ulepszona obsługa JDBC (łącze do baz danych w języku Java), z możliwością integracji z Hibernate,
- Szablon zarządzania transakcjami, dzięki któremu nie trzeba uruchamiać środowiska Java EE,
- Szablon MVC – oddzielenie warstwy biznesowej od interfejsu i zarządzania akcjami podejmowanymi przez użytkownika.

Spring składa się z elastycznych bibliotek, chętnie wykorzystywanych przez programistów. Rozwiązania oparte o ten framework obejmują zakupy online oraz tak zaawansowane przedsięwzięcia jak transmisje telewizyjne czy inne innowacje. Spring wykorzystywany jest przez wiele znanych na rynku IT marek, takich jak Alibaba, Amazon, Google, czy Microsoft.

Punktem wyjścia dla tworzenia oprogramowania w tej technologii jest Spring Boot. Łączy on w sobie elementy niezbędne, takie jak kontekst aplikacji i automatycznie konfigurowany, wbudowany serwer sieciowy, co pomaga w tworzeniu mikro usług. Wykorzystując scaffolding, otrzymujemy gotowy szkielet naszego programu z automatyczną konfiguracją.

Spring został zaprojektowany z myślą o zwiększeniu wydajności aplikacji. Oznacza to domyślnie szybkie uruchamianie, szybkie zamykanie i zoptymalizowane wykonywanie poleceń użytkownika. Wbudowane serwery internetowe, automatyczna konfiguracja i „fat jars”, czyli pliki zawierające wszystkie zewnętrzne zależności bibliotek, powodują, że pracę z aplikacją można rozpocząć szybko. Producent deklaruje, że nowy projekt Spring może być uruchomiony nawet w kilka sekund, za pomocą Spring Initializr na start.spring.io.

Inicjalizator jest przedstawiony na Rysunku 22. W niniejszej pracy został użyty SpringBoot 2.4.3.

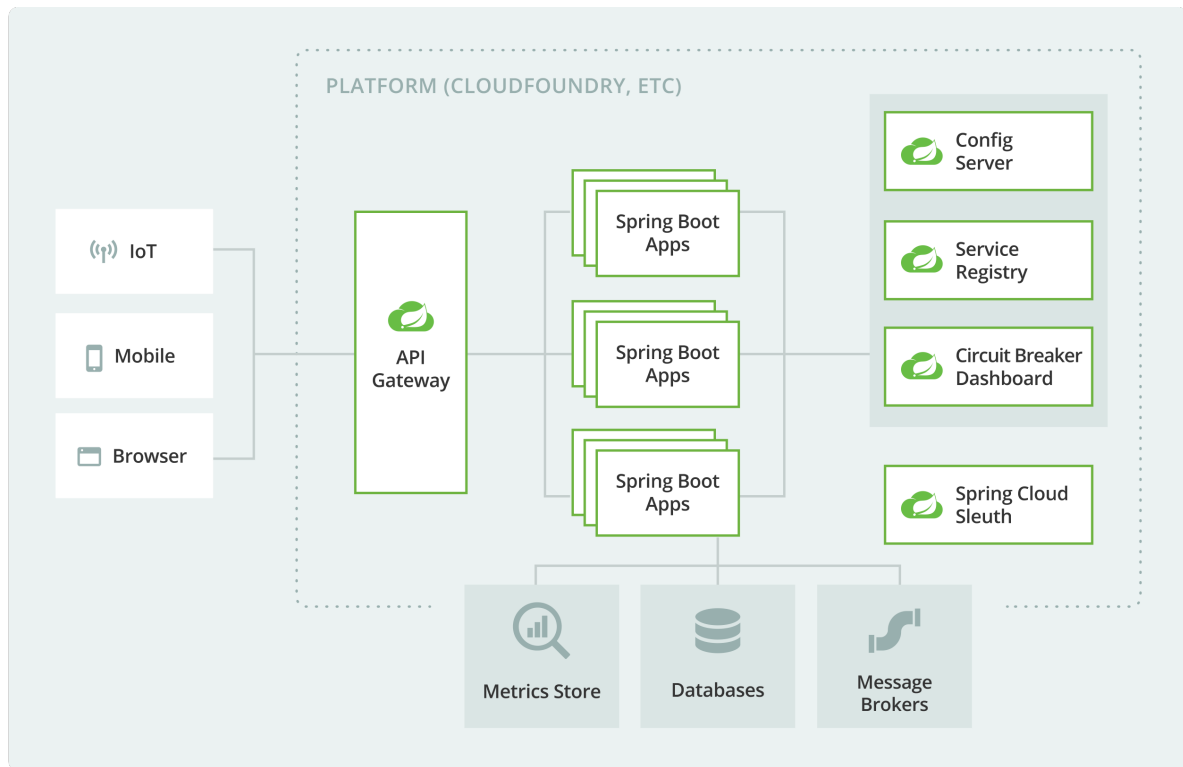
Rysunek numer 22 przedstawia okno instalacji i generowania nowego projektu wraz z dostępnymi opcjami.

The screenshot shows the Spring Initializr web application interface. On the left is a sidebar with the Spring Initializr logo and the text "Bootstrap your application". The sidebar contains sections for "Project", "Language", "Spring Boot", "Project Metadata", and "Dependencies". The main content area on the right allows for configuring a new project. It has tabs for "Maven Project" (selected) and "Gradle Project". Under "Language", "Java" is selected, with "Kotlin" and "Groovy" as options. The "Spring Boot" section shows version options: "2.2.0 (SNAPSHOT)", "2.1.4 (SNAPSHOT)", "2.1.3" (selected), and "1.5.19". The "Project Metadata" section includes input fields for "Group" (com.example) and "Artifact" (demo). Below this is a "More options" button. The "Dependencies" section has a search bar with the text "Search dependencies to add" and a list of suggestions: "Web, Security, JPA, Actuator, Devtools...". At the bottom left, there is copyright information: "© 2013-2019 Pivotal Software", "start.spring.io is powered by", and links to "Spring Initializr" and "Pivotal Web Services". At the bottom right is a large green button labeled "Generate Project - alt + ⌘".

Rysunek 22. Okno instalacji Spring. Źródło: Opracowanie własne.

Istotną częścią usługi Spring jest platforma Spring Cloud. Dzięki niej możemy tworzyć aplikacje dostępne w chmurze. Spring Cloud ma za zadanie uproszczenie budowy systemów rozproszonych. Celem jest zmniejszenie złożoności i mniejsze narażenie na błędy. Spring Cloud pomaga skoordynować działanie mikroservisów. Dostarcza narzędzi do zarządzania serwisami i ich instancjami, obsługuje awarie, śledzi żądania i upraszcza odpytywanie zasobów w komunikacji opartej o http.

Rysunek numer 23 przedstawia przykładową architekturę mikroservisów z wykorzystaniem Spring Cloud.



Rysunek 23. Przykładowa architektura mikroservisów z wykorzystaniem Spring Cloud.

Źródło: <https://spring.io/microservices>

Spring Cloud zawiera również rozwiązanie API Gateway. Zapewnia ono bibliotekę do budowania bramy API na bazie Spring WebFlux. Spring Cloud Gateway ma na celu zapewnienie prostego sposobu kierowania do interfejsów API, przy zapewnieniu monitorowania i bezpieczeństwa transakcji. Rysunek 23 przedstawia schemat architektury Spring Cloud.

4.3. **Hibernate**

Co do zasady, aplikacje komputerowe operują na danych, które są wprowadzane, zapisywane i odczytywane. Należy zatem zadbać o trwałość danych. Chodzi w skrócie o to, aby nasze dane były bardziej żywotne niż czas uruchomienia aplikacji, by były właściwie zorganizowane, przechowywane oraz by zapewnić do nich optymalny dostęp.

Projektując aplikację komputerową w obiektowym języku programowania zazwyczaj napotykamy problem polegający na niezgodności impedancji. Pojawia się on wówczas, gdy w relacyjnej bazie danych (a takie są wciąż najbardziej powszechne) chcemy przechowywać informacje, które obejmują obiekty i zależności między nimi. „Ręczne” modelowanie danych wykorzystywanych w programie do formatu, który może być przechowywany w relacyjnej bazie danych, może być skomplikowane i stwarza ryzyko potencjalnych błędów zapisu i odczytu danych.

Aby zapewnić prawidłową trwałość danych w relacyjnej bazie danych, dobrą praktyką jest zastosowanie mapowania obiektowo-relacyjnego (ang. Object-Relational Mapping – ORM), jako odwzorowania obiektowej architektury systemu na relacyjną bazę danych.

Można oczywiście stworzyć aplikację działającą wyłącznie po stronie bazy danych. Wówczas problem niezgodności impedancji nie istnieje. Bezpośrednia praca z relacyjną bazą danych nie jest jednak wygodna. Jednym z rozwiązań zapewniających łatwiejsze uzyskanie trwałości danych jest biblioteka Hibernate [8] (projekt open source rozwijany od 2001 r.). Opiera się on na wykorzystaniu opisu struktury danych za pomocą języka XML

Nie jest to rozwiązanie idealne. Nadal „po drugiej stronie” istnieje relacyjna baza danych, a biblioteka Hibernate jedynie zmniejsza problem niezgodności impedancji, między innymi przez automatyzację (z punktu widzenia programisty) operacji związanych z zapisem i odczytem danych. Optymalizacje, poza „rozpoznawaniem” struktur danych, np. relacji między tabelami w asocjacjach, dotyczą takich zagadnień, jak: cache’owanie obiektów (przy niezmienionym obiekcie nie musimy sięgać do bazy danych), cache’owanie wyników zapytań (jak wyżej, tylko przy zapytaniach).

Do innych usprawnień możemy zaliczyć:

- wykonywanie poleceń SQL dopiero wówczas, gdy są potrzebne,
- jeśli obiekt nie został zmodyfikowany, nie aktualizujemy go w bazie,
- efektywne zarządzanie kolekcjami,
- wiele zmian w programie połączone jest w jedno UPDATE.

Pozycja na rynku mapowania obiektowo-relacyjnego w języku Java – JPA. Specyficzny dla Hibernate interfejs programistyczny oferuje dużo bogatszy zakres opcji i metod niż standard JPA.

4.4. Maven

Maven jest narzędziem automatyzującym budowę oprogramowania w projektach informatycznych [9].

Narzędzie w znacznym stopniu upraszcza nam zarządzanie strukturą projektu i jego zależnościami, dzięki czemu Maven sam sprawdza, które biblioteki są wymagane w projekcie oraz pilnuje ich najnowszych wersji.

Praca Mavena opiera się na tak zwanym cyklu życia. Obecnie są to trzy cykle: default, clean i site.

Każdy z nich odpowiada za inne funkcje:

- default – obsługuje wdrażanie projektu,
- clean – pomoc w czyszczeniu projektu,
- site – tworzeniem witryny internetowej.

Cykl default składa się z siedmiu faz:

- validate – sprawdza czy kod jest poprawny,
- compile – odpowiada za kompilację kodu źródłowego,
- test – testuje skompilowany kod za pomocą frameworka do testów jednostkowych,
- package – pakuje już skompilowany kod do plików w formacie np. JAR,
- verify – przeprowadzając testy integracyjne, sprawdza czy spełnia wszystkie kryteria jakościowe,
- install – instaluje paczkę w repozytorium w celu jej dostępności dla innych projektów,
- deploy – końcową paczkę zapisuje w zdalnym repozytorium w celu jej udostępnienia.

Maven to niezwykle pomocne narzędzie, dzięki niemu możemy uniknąć wielu problemów oraz przyspieszyć pracę nad projektem. Zastosowanie go w projekcie w znacznym stopniu ułatwi nad nim prace.

4.5. Serwer bazy danych MySQL

MySQL to system zarządzania relacyjnymi bazami danych stworzonym przez Sun Microsystems i MySQL AB [10]. Obecnie jest rozwijany przez firmę Oracle. Jest to najpopularniejszy

system zarządzania bazami danych na świecie. Jest dostępny na wszystkie najpopularniejsze platformy (Windows, MacOS, Linux). MySQL został napisany z uwzględnieniem kompatybilności ze standardem SQL.

4.5.1 Język SQL

Język SQL to język deklaratywny stworzony w celu tworzenia i modyfikowania baz danych oraz zarządzania nimi. Zapytania w języku SQL należą do czterech zbiorów.

DML – Data Manipulation Language – język manipulacjami danymi	
INSERT	Wprowadzenie danych
UPDATE	Zmiana danych
DELETE	Usunięcie danych

DDL – Data Definition Language – język definicji danych	
CREATE	Stworzenie struktury
DROP	Usunięcie struktury
ALTER	Zmiana struktury

DCL – Data Control Language – język kontroli nad danymi	
GRANT	Nadanie uprawnień
REVOKE	Odebranie uprawnień
DENY	Zabronienie wykonania operacji

DQL – Data Query Language – język definiowania zapytań	
Jest traktowany jako część języka DML, lecz służy manipulowaniu danymi z użyciem opcjonalnych klauzuli.	
FROM	Wskazanie konkretnych danych źródłowych
WHERE	Filtrowanie wierszy
GROUP BY	Grupowanie danych
HAVING	Filtrowanie wierszy
ORDER BY	Sortowanie na podstawie kolumn

4.6. JavaScript i React

Rozdział poświęcony przybliżeniu technologii jakie zostały wykorzystane na frontendzie aplikacji.

4.6.1 JavaScript

JavaScript to wieloplatformowy, skryptowy język programowania, którego twórcą jest Brendan Eich. Pierwsze wydanie pojawiło się w grudniu 1995 roku [11].

W 1997 roku, za sprawą europejskiej organizacji ECMA, JavaScript otrzymał swoją własną specyfikację nazwaną ECMAScript. Od tego momentu uznaje się go za w pełni niezależny język programowania.

JavaScript wykorzystywany jest zarówno po stronie serwera jak i po stronie klienckiej. Pozwala on na tworzenie interakcyjnych stron internetowych. Podczas, gdy języki takie jak HTML czy CSS wprowadzają odpowiednio strukturę i wygląd tworzonej stronie internetowej, tak JavaScript dostarcza elementów, które angażują użytkownika, co wzbogaca jego doznania. JavaScript może posłużyć między innymi do walidacji formularzy znajdujących się na stronie, aktualizacji wyświetlanych danych, zastosowania ciekawych animacji czy też podpowiedzi. Reasumując, JavaScript „ożywia” statyczne strony internetowe.

JavaScript posiada mnóstwo dodatkowych bibliotek, które pomagają zaoszczędzić sporo czasu jak również ułatwić pracę deweloperom. Dzieje się tak dlatego, że owe biblioteki, dostarczają gotowych rozwiązań w postaci różnych funkcji, metod czy obiektów, które możemy zastosować w naszym projekcie bez konieczności tworzenia ich od podstaw.

Jednym z przykładów JavaScript-owej biblioteki jest React.js.

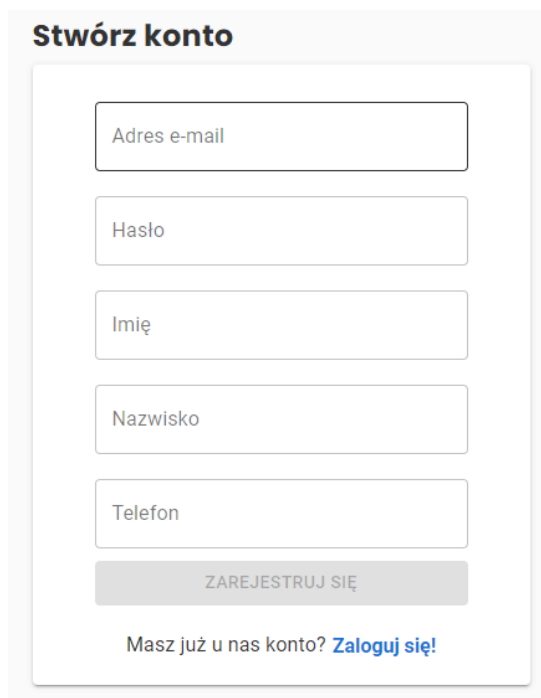
4.6.2 React

React.js został opracowany w 2011 roku przez Facebook-a przez Jordana Walke [12]. W tamtym czasie platforma rozrosła się do tego stopnia, że postanowiono stworzyć własną front-end-ową bibliotekę, która sprostałaby wymaganiom portalu społecznościowego. W 2013 roku React.js stał się open source-owy tj. nie jest wymagana żadna płatność, aby z niego korzystać. Odkąd React.js stał się dostępny dla wszystkich jego popularność stale rosła, a to za sprawą jego ciągłego rozwijania poprzez dodawanie nowych funkcji oraz dużego grona użytkowników.

React.js bazuje na komponentach, są to takie „cegiełki”, z których budujemy naszą aplikację. Takim elementem może być np.: wyskakujące okienko z wiadomością, ikona, przycisk czy kontener służący do ich przechowywania i pozwalający na przekazywanie danych między nimi. Całość sprowadza się to tego, że „mówimy” Reactowi jak nasze komponenty powinny wyglądać w zależności od stanu aplikacji, a on się tym zajmuje wprowadzając konieczne zmiany w wyglądzie, które będą odpowiednie do zmian w stanie. Takie podejście nazywane jest deklaratywnym programowaniem. Jest ono bardzo intuicyjne i ułatwia tworzenie bardzo skomplikowanych aplikacji front-end-owych.

4.7. Material-UI

Material-UI to biblioteka, która pozwala importować i używać różnych komponentów do tworzenia interfejsów w aplikacjach React [13]. Widżety Material-UI są inspirowane zasadami Google dotyczącymi tworzenia interfejsów użytkownika. Dzięki temu możemy stworzyć atrakcyjnie wizualne oraz dopasowane wyglądem do systemu aplikacje. Rysunek numer 24 pokazuje propozycję widoku umożliwiającego rejestrację użytkownika w systemie.



Stwórz konto

Adres e-mail

Hasło

Imię

Nazwisko

Telefon

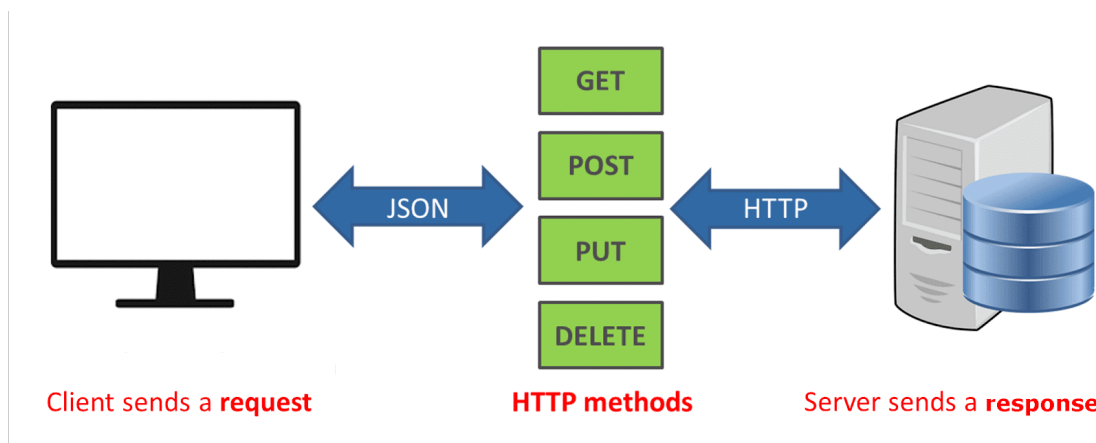
ZAREJESTRUJ SIĘ

Masz już u nas konto? [Zaloguj się!](#)

Rysunek 24. Przykładowa makieta widoku do rejestracji użytkownika. Źródło: Opracowanie własne.

4.8. REST API

REST API jest to styl oprogramowania, które działa na podstawie określonych reguł. Opisuje on, w jaki sposób definiowane są zasoby oraz umożliwia dostęp do nich. REST (REpresentational State Transfer) to rozwiązanie zapewniające gotowe standardy komunikacji między systemami komputerowymi w Internecie, ułatwiające przekazywanie danych między systemami, dzięki gotowym protokołom. [14]. Jednym z wykorzystywanych przez interfejsy internetowe REST API jest protokół HTTP, wykorzystujący wzorzec żądanie-odpowiedź, w którym dzięki metodom w nim zawartym (np. GET, POST, PUT, DELETE) można uzyskać dostęp lub wykonywać akcje na zasobach za pomocą parametrów zakodowanych w adresie URL lub ciele żądania, w którym możliwe jest przesłanie danych w formacie JSON lub XML, a w odpowiedzi możemy otrzymać potwierdzenie wykonanej operacji.

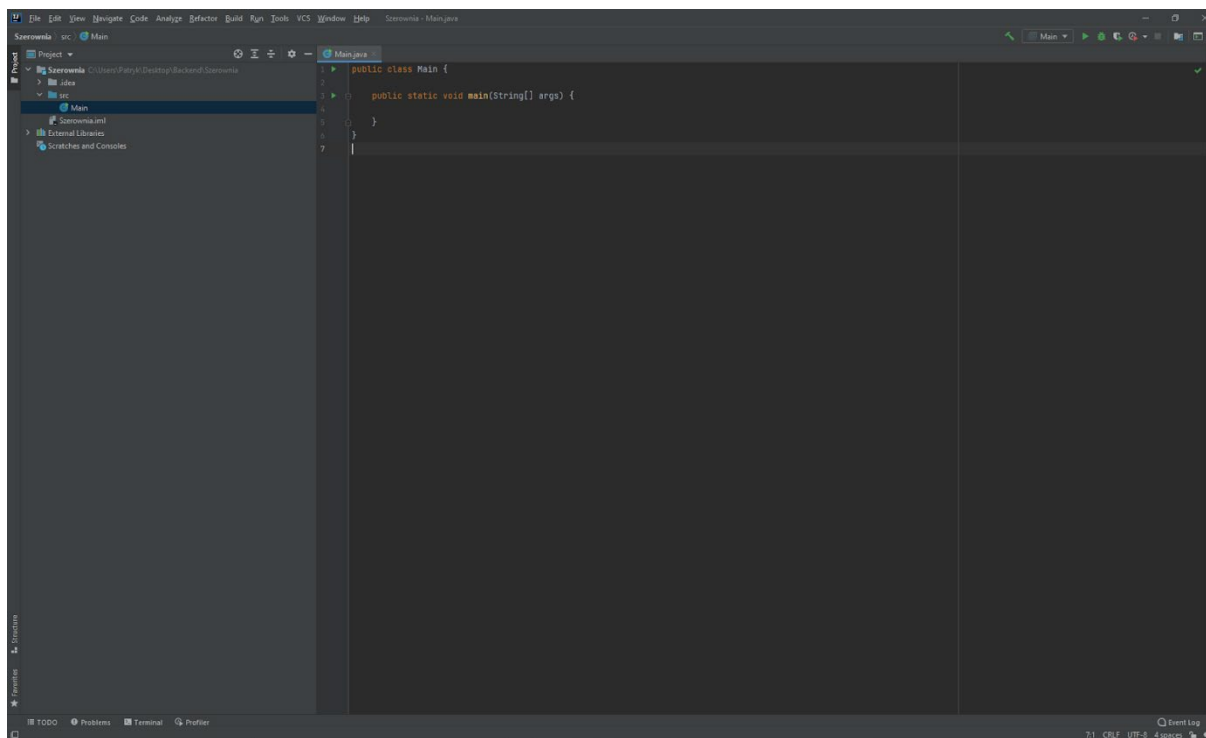


Rysunek 25. Schemat komunikacji w REST API. Źródło: <https://phpenthusiast.com/blog/what-is-rest-api>

Wykorzystanie REST API pozwala na niezależną implementację funkcjonalności oraz tworzenie architektury warstwowej. Wykorzystywany protokół bezstanowy pomaga w zachowaniu wysokiej wydajności i niezawodności, a przesyłane zadania są wykonywane niezależnie, po czym zwracana jest odpowiedź. REST API dla bezpieczeństwa powinno być podzielone na poszczególne warstwy, dzięki temu, jeśli atakowana jest jedna z warstw nie oznacza to, że atakujący będzie miał dostęp do kolejnej warstwy. Element Code on demand został oznaczony przez twórcę jako dodatkowy, zakłada, że można modyfikować kod aplikacji dzięki pobraniu kolejnych pakietów z serwera.

4.9. IntelliJ IDEA

Podczas implementowania części serwerowej, skorzystano z aktualnie popularnego IDE firmy JetBrains - IntelliJ [15]. Producent oferuje użytkownikom dwie wersje programu: Community - bezpłatna oraz Ultimate - płatna, która posiada dodatkowe funkcjonalności poprawiające wygodę tworzenia oprogramowania. Jako że wszystkim z zespołu przysługuje licencja studencka, to każdy z implementacji korzystał z wersji płatnej.



Rysunek 26. IntelliJ IDEA wersja Ultimate. Źródło: Opracowanie własne.

IntelliJ IDEA posiada całą gamę zaimplementowanych funkcji, które mają za zadanie ułatwienie nam pracy oraz zwiększenie możliwości w elastycznym wolnym od błędów budowaniu aplikacji:

- Zintegrowane narzędzia programistyczne, np. Debugger, Dekompilator
- Możliwość pracy z Docker lub Tomcat
- Możliwość tworzenia testów aplikacji z pomocą np. ScalaTest
- IntelliJ posiada interfejs służący do kontrolowania wersji za pomocą narzędzi np. Git, Subversion
- Narzędzia do budowy aplikacji. Wspiera m.in Gradle, Maven, NPM
- Wsparcie frameworków, np. Android studio, Swing, Hibernate

4.10 Postman i Swagger

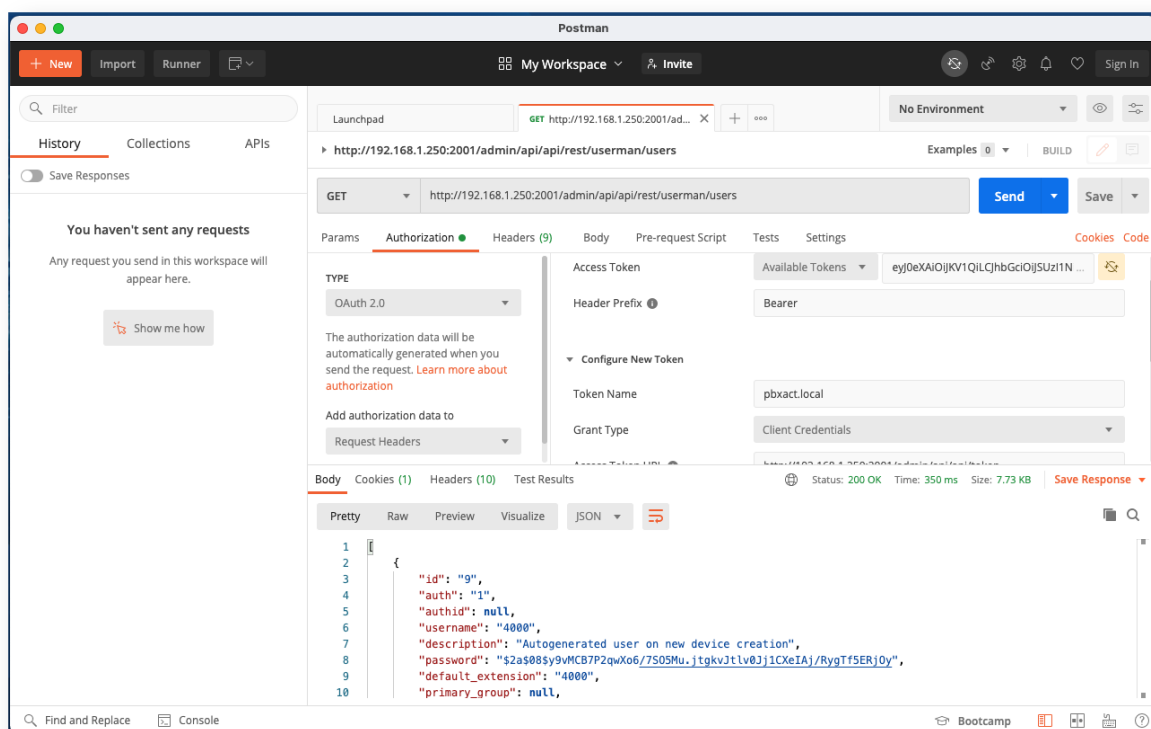
Rozdział opisuje pomocnicze narzędzia jakie zostały użyte w cyklu wytwarzania oprogramowania dla projektu.

4.10.1 Postman

Postman jest darmowym narzędziem rozwijanym od przeszło 10 lat. Jest uważany za najpopularniejsze oprogramowanie do testów API na rynku [16]. Dzięki wieloletniemu rozwojowi,

twórcy w odpowiedzi na potrzeby użytkowników nieustannie rozwijają aplikację o nowe funkcjonalności. Postman to bardzo rozbudowane narzędzie, niezbędne podczas tworzenia oprogramowania. Służy głównie do wysyłania żądań http, ale nie tylko. Pozwala na tworzenie własnych paczek zapytań, które w prosty sposób mogą być udostępniane między użytkownikami. Umożliwia to usprawnienie procesu wytwarzania oprogramowania. Jedną z kluczowych zalet Postman-a, jest możliwość generowania dokumentacji niezbędnej do dalszej pracy przy rozwoju aplikacji.

Rysunek numer 27 prezentuje okno aplikacji Postman z wywołaniem metody GET oraz fragment jej ciała.



Rysunek 27. Przykładowe okno aplikacji Postman. Źródło: Opracowanie własne.

4.10.2 Swagger

Swagger to framework, który pozwala wizualizować i korzystać z API aplikacji, przy okazji tworząc dokumentację [17]. Podstawowa konfiguracja Swaggera nie jest zbyt trudna, trwa zwykle nie więcej niż dwadzieścia minut. Swagger sprawia, że projektowanie interfejsu API jest znacznie prostsze. Największym plusem wykorzystania tego narzędzia jest jednak uzyskanie samoaktualizującej się dokumentacji. Dzięki temu programiści nie muszą wraz z każdą zmianą dokonywać modyfikacji zapisanej dokumentacji. Jej utrzymanie jest trudne, często zawiera błędy wynikające z drobnych

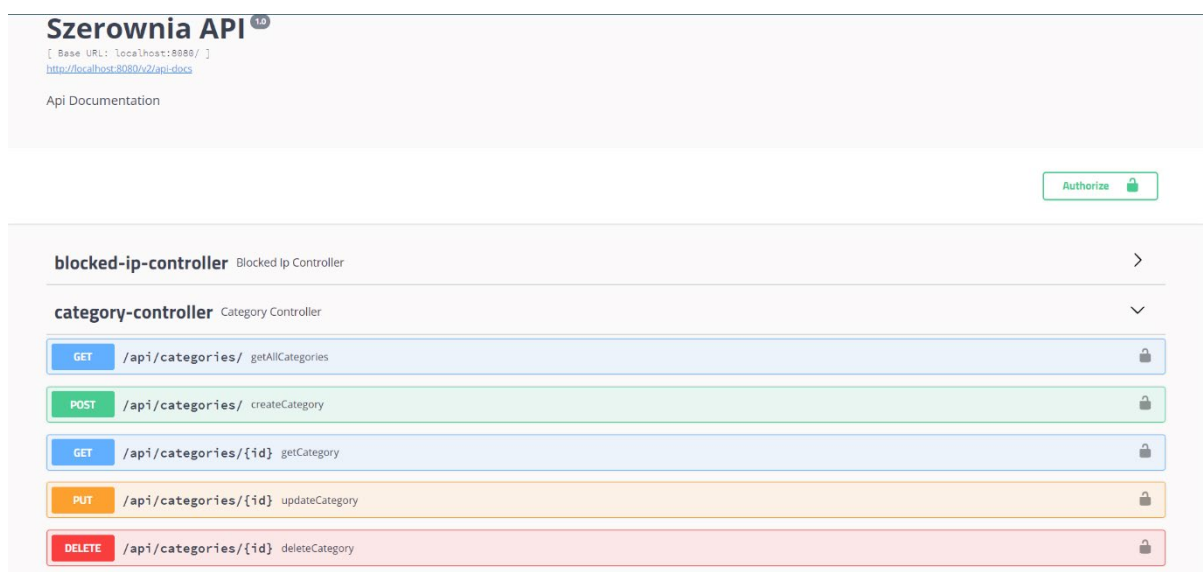
pomyłek programisty w jej tworzeniu lub rozbieżności pomiędzy wersją API a wersją dokumentacji. Na szczęście powstał Swagger, który ściąga z barków programistów ten obowiązek.

Kiedy chcemy wygenerować dokumentację referencyjną, mamy do wyboru naprawdę szeroką gamę programów, które mogą to zrobić na podstawie definicji API. Najlepiej sprawdzają się narzędzia z pakietu Swaggera, za którego sprawą standard ten w ogóle powstał. W pakiecie Swaggera jest kilka programów. Najczęściej wykorzystywane to:

- **Swagger Editor** – program ten umożliwia pisanie specyfikacji OpenAPI bezpośrednio w przeglądarce. Wtedy w czasie rzeczywistym następuje weryfikacja, czy pisana definicja jest zgodna z odpowiednim standardem. Od razu generuje się też podgląd dokumentacji. Ułatwia to dostrzeżenie błędów merytorycznych, jeżeli takowe wystąpią.
- **Swagger UI** – możemy z niego skorzystać, kiedy mamy już gotową definicję OpenAPI. Swagger UI wygeneruje na jej podstawie dokumentację referencyjną. Powstałą tym sposobem stronę internetową można umieścić na dowolnym serwerze, gdzie użytkownicy będą mogli z niej korzystać.

Ważnym elementem wiedzy o OpenAPI jest specyfikacja. Specyfikacja ta polega na opisie API RESTful za pomocą pliku JSON/YAML – można jej użyć do tworzenia próbnego serwera, generowania dokumentacji przez użytkownika, generowania kodu i wielu innych rzeczy. Specyfikacja OAS (OpenAPI Specification) jest niezależna od języka, w którym piszemy – jest rozumiana przez ludzi oraz komputery.

Rysunek numer 28 przedstawia widok metod w Swagger dla API Szerownia.pl



Rysunek 28. Widok API dla Szerownia.pl w swagger. Źródło: Opracowanie własne.

Standard API jest oparty na standardzie REST, wykorzystując jednocześnie format reprezentacji danych definiowany standard JSON-a. Zalecenia związane ze standardami dokumentacji API:

- opis zasobów,
- opis obsługiwanych metod i punktów dostępowych,
- parametry obsługiwane przez punkt dostępowy,
- przykłady kodu żądań, obrazujących użycie złożonych, ale nieskomplikowanych zapytań.

Programiści chętnie korzystają z tego narzędzia, ponieważ ich praca dzięki temu jest sprawniejsza, szybsza, a ich działania bardziej efektywne i obciążone mniejszym ryzykiem błędu. Dzięki prostym plikom takim jak JSON czy YAML mogą zwiększyć jakość swojej pracy.

4.11. Android SDK

Zanim przejdziemy do Android SDK (zestawu narzędzi dla programistów), należy w skrócie opisać, czym jest sam Android oraz dlaczego aplikacje mobilne warto projektować przede wszystkim z myślą o użytkownikach Androida [18].

Android jest systemem operacyjnym przeznaczonym na urządzenia mobilne (smartfony, tablety itp.), a także telewizory (Android TV), zegarki czy samochody. W ostatnich latach żaden inny system nie dorównuje mu popularnością. W 2021 korzystało z niego niemal 73% urządzeń mobilnych. Jego najpoważniejszy konkurent – iOS – może pochwalić się wynikiem na poziomie zaledwie 26%. Wg danych z lipca 2021 r., pozostałe systemy mobilne zajmują jedynie 0,82% rynku.

Jednym z czynników, który spowodował tak duży udział Androida w rynku jest to, że jądro systemu oraz część jego składowych zostały opublikowane na licencji GNU GPL, czyli w postaci wolnego i otwartego oprogramowania. Android, kiedyś rozwijany przez Android Inc., następnie firma ta została kupiona przez Google, a obecnie jest częścią Open Handset Alliance (konsorcjum, którego częścią są również Google, Motorola, czy HTC).

Z myślą o systemie Android powstają aplikacje, których ogólna liczba w Google Play (oficjalne źródło aplikacji na tę platformę, poprzednio był to Android Market) osiągnęła w 2021 r. liczbę około 3,5 miliona.

Jedną z tajemnic sukcesu Androida jest jego dostępność. Deweloperzy mogą docierać do klientów z całego świata dostarczając im unikalne produkty, a producenci urządzeń instalują Androida za darmo, bez kupowania licencji czy tworzenia własnego systemu operacyjnego. Każdy producent może używać systemu do oprogramowania, bez żadnych opłat i umów z Google. Takie rozwiązanie obniża koszty produkcji, a o za tym idzie również cenę produktu końcowego. Ma to

znaczenie szczególnie w krajach, których mieszkańcy nie dysponują finansami pozwalającymi na zakup kosztownego sprzętu. Na większości smartfonów w Indiach i Afryce zainstalowany jest właśnie Android.

Przejdźmy teraz do Android SDK (Software Development Kit). Służy on do tworzenia aplikacji i składa się z dwóch komponentów: SDK Tools oraz Platform Tools.

Środowisko programistyczne Android obejmuje następujące najważniejsze składniki: dokumentację, tutoriale, przykłady aplikacji, emulator urządzeń mobilnych, biblioteki debugger. Programista może tworzyć oprogramowanie na androida w Linuksie, OS X oraz Microsoft Windows (najnowsze wersje).

Zestaw narzędzi do tworzenia aplikacji na Androida składa się z modułów. Dowolny moduł można zainstalować i odinstalować. Przykładem może być emulator, do którego dodajemy wydania systemu Android. Innymi modułami są sterowniki oraz źródła Software Development Kit.

SDK Tools abstrahuje od konkretnej wersji Androida i składa się z następujących najważniejszych narzędzi: android, debugger, emulator, analizator layoutów, mksdcard (tworzenie obrazu dysku do symulowania obecności zewnętrznej pamięci), ProGuard (optymalizacja i zaciemnianie kodu).

Drugi komponent Android SDK – Platform tools – obejmuje zestaw bibliotek niezbędnych do skompilowania aplikacji dla określonej, niekoniecznie najnowszej, wersji Androida. Najpopularniejszym narzędziem z tego pakietu jest Android Debug Bridge (adb). Służy ono do kontroli urządzenia z Androidem lub zainstalowanego emulatora.

Innym programem jest fastboot, dzięki któremu możemy wczytać obraz systemu na urządzenie, zarządzać partycjami, czy odblokowywać bootloader.

Jeśli chodzi o środowisko programistyczne, to do tworzenia aplikacji może posłużyć Android Studio. Jest to oficjalne narzędzie na platformę Android, które korzysta z IntelliJ IDEA firmy JetBrains. Poza wygodnym pisanem kodu, dostosowanym do programowania aplikacji na urządzenia mobilne, Android Studio umożliwia skorzystanie z wielu emulatorów m.in. telefonów i tabletów w celu przetestowania aplikacji.

4.12. Selenium - testy automatyczne

Selenium WebDriver to najpopularniejsze narzędzie typu open source do automatyzowania operacji wykonywanych przez przeglądarkę, które pojawiło się w 2009 r. Selenium 2.0 ma za zadanie zautomatyzowanie testów frontentu aplikacji webowych [19]. Selenium WebDriver przyjmuje zadania

dane mu przez użytkownika, a następnie wysyła je do przeglądarki. W przeciwieństwie do Selenium 1, gdzie serwer Selenium był niezbędny do uruchamiania testów, Selenium WebDriver nie potrzebuje specjalnego serwera do wykonywania testów. Zamiast tego WebDriver bezpośrednio uruchamia instancję przeglądarki i kontroluje ją. Obsługuje większość popularnych przeglądarek internetowych takich jak: Google Chrome, Safari, Opera czy Firefox, oraz działa w systemach operacyjnych: Windows, macOS, Linux.

4.13. Testy jednostkowe

Testy jednostkowe polegają na przetestowaniu indywidualnych fragmentów programu. Ich celem jest sprawdzenie poszczególnych części bądź komponentów aplikacji, pod kątem wykrycia ewentualnych błędów oraz sprawdzenia, czy działają one zgodnie z postawionymi im wymaganiami. Testy jednostkowe są przeprowadzane w trakcie rozwoju aplikacji przez deweloperów. Najbardziej popularną formą testów jednostkowych są testy zautomatyzowane. Wyglądają one tak, że twórca najpierw pisze sekcję kodu w aplikacji, która następnie sprawdza poprawność działania np. jakiejś funkcjonalności. Zaletami testów jednostkowych są:

- Jakość kodu,
- Szybsze i łatwiejsze wykrycie potencjalnych błędów,
- Lepsze zrozumienie funkcjonalności aplikacji,
- Redukcja kosztów.

Do tworzenia zautomatyzowanych testów jednostkowych, weryfikujących poprawność aplikacji tworzonych w języku Java, służy framework JUnit. Dostarcza on następujące funkcjonalności:

- Środowisko do tworzenia i przeprowadzania testów,
- Adnotacje, oznaczające testujące metody,
- Asercje do weryfikacji oczekiwanych wyników,
- Automatyczne wykonywanie testów, z ich natychmiastowym wynikiem.

Testy jednostkowe w JUnit-cie polegają na tworzeniu tzw. „test case-ów”, czyli części kodu, które sprawdzają np. czy testowana metoda działa tak jak powinna. Każdy test case charakteryzuje informacja o tym, co ma być na wejściu, a co na wyjściu testującej metody. Przykład takiego testu został zaprezentowany na listingu numer 1.

Listing 1. Przykładowy kod testu jednostkowego. Źródło: Opracowanie własne.

```
@Test
void getProductsList() {

    Category category = new Category("testCategory", false, (Category)null);
    Product product1 = new Product("testName1", "testMake1", "testModel1", 5,
222);
    Product product2 = new Product("testName2", "testMake2", "testModel2", 5,
222);
    Product product3 = new Product("testName3", "testMake3", "testModel3", 5,
222);

    product1.setCategory(category);
    product2.setCategory(category);
    product3.setCategory(category);

    List <Product> productList = List.of(product1, product2, product3);
    Mockito.when(this.productRepo.findAll().thenReturn(productList));

    List <ProductListResponseDTO> productListResponseDTO =
    this.productService.getProductList();

    Assertions.assertNotNull(productListResponseDTO);
    Assertions.assertEquals(3, productListResponseDTO.size());

    for(int i = 0; i < 3; ++i){

        Assertions.assertEquals(((Production)productionList.get(i)).getName(),
((ProductionListResponseDTO)productListResponseDTO.get(i)).getName());

        Assertions.assertEquals(((Production)productionList.get(i)).getMake(),
((ProductionListResponseDTO)productListResponseDTO.get(i)).getMake());

        Assertions.assertEquals(((Production)productionList.get(i)).getModel(),
((ProductionListResponseDTO)productListResponseDTO.get(i)).getModel());

        Assertions.assertEquals(((Production)productionList.get(i)).getCondition(),
((ProductionListResponseDTO)productListResponseDTO.get(i)).getCondition());

        Assertions.assertEquals(((Production)productionList.get(i)).getEstimatedValue
(), ((ProductionListResponseDTO)productListResponseDTO.get(i)).getEstimatedValue
());

    }

}
```

5. Projekt i implementacja systemu

Aplikacja przygotowana została na bazie Spring Framework. Jest to szkielet tworzenia aplikacji w języku Java. Uznawany jako lekki, który nie wymusza specyficznego modelu programowania. Jego stosowanie jest bardzo popularne wśród programistów Javy. Szkielet oferuje dużą swobodę w tworzeniu rozwiązań, jednocześnie jest dobrze udokumentowany i zawiera rozwiązania wielu zagadnień, często występujących w programowaniu. Dodatkowo, w Spring Framework istnieje wiele rozszerzeń, które pozwalają budować aplikacje webowe [20].

Spring pozwala programistom zbudować niemal każdą aplikację, jaką można sobie wyobrazić. Zestaw narzędzi udostępnianych przez framework pozwala na wykorzystanie takich funkcji jak Inversion of Control (IoC) i Dependency Injection (DI). Uniwersalność tego szkieletu powoduje, że może on zostać użyty do tworzenia aplikacji bezpiecznych, reaktywnych, opartych na chmurze, służących do udostępniania usług internetowych o niewielkiej skali, jak i do obsługi złożonych przepływów danych dla dużych przedsiębiorstw [21].

Framework Spring zarządza Java Beans, czyli komponentami wielokrotnego użytku dla platformy Java. Komponenty te opakowują wiele obiektów w jeden obiekt zwany ziarnem (ang. bean), przestrzegając ściśle określonych konwencji, który może być przenoszony i używany w różnych miejscach aplikacji. Ziarno jest serializowalnym obiektem z bezargumentowym konstruktorem, do którego właściwości można dostać się przy pomocy metod dostępowych.

Do stworzenia interfejsu użytkownika aplikacji Szerownia.pl wykorzystano React, który jest biblioteką języka programowania JavaScript. To popularne rozwiązanie często wykorzystywane jest do tworzenia aplikacji typu *Single Page Application*. Biblioteka stosuje Obiektowy Model Dokumentu (ang. Document Object Model). Model jest przechowywany w pamięci, a po zmianie stanu wyszukiwane są różnice między wirtualnym i prawdziwym DOM, następnie aktualizowane są zmiany. Z takiego rozwiązania korzystają na swoich stronach internetowych takie firmy jak: Netflix, Imgur, PayPal, Archive.org, Gamepedia, SeatGeek, HelloSign czy Walmart [20].

Bibliotekę zastosowano zgodnie z wzorcem architektonicznym MVC - Model-View-Controller. Jest to rozwiązanie służące do tworzenia aplikacji, w których obok logiki biznesowej i operacji na danych, dostarczane są również GUI (graficzne interfejsy użytkownika) [22]. Taki wzorzec dzieli aplikację na poniższe, wzajemnie połączone części:

- Model, który obejmuje problem biznesowy i logikę działania aplikacji.
- Widok, zgodnie z którym elementy modelu wyświetlane są użytkownikowi (odświeżany po aktualizacji modelu).
- Kontroler, któremu przekazywane są żądania użytkownika, odpowiedzialny za dokonywanie zmian w modelu, a także odświeża interfejs w odpowiedzi na żądania lub zmiany spowodowane żądaniami [23].

Repozytorium aplikacji jest wówczas niezależne od widoku i do każdego z serwisów (w przypadku serwisu Szerownia.pl będzie to aplikacja przeglądarkowa i aplikacja mobilna) wstrzyknięte zostaje to samo repozytorium danych.

5.1. Wzorzec projektowy

Wzorzec projektowy jest rozwiązaniem często pojawiających się problemów w tworzeniu oprogramowania, które może być stosowane uniwersalnie, w różnych projektach opartych o obiektowe języki programowania. Taki wzorzec opisuje rozwiązanie problemu, pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz utrzymanie kodu. Implementacja kodu w oparciu o wzorzec pozostawiona jest programiście [24].

Wzorce projektowe możemy podzielić na:

- Kreacyjne - opisujące elastyczne sposoby tworzenia obiektów, uniezależniające system od sposobu tworzenia obiektów,
- Strukturalne - opisujące sposób konstrukcji struktur obiektowych, korzystające z dziedziczenia i delegacji,
- Behawioralne - opisujące algorytmy i przydział odpowiedzialności, charakteryzujące sposób interakcji między obiektami [25].

Poza wspomnianym wcześniej wzorcem architektonicznym MVC, aplikacja została zbudowana w oparciu o wzorzec Singleton. Jest to kreacyjny wzorzec projektowy, którego celem jest ograniczenie możliwości tworzenia obiektów danej klasy do jednej instancji oraz zapewnienie globalnego dostępu do stworzonego obiektu. Może on również służyć do ustalenia maksymalnej liczby obiektów, jakie mogą istnieć w systemie [26].

Implementację wzorca Singleton możemy podzielić na dwa etapy:

- W pierwszym należy ograniczyć dostęp do domyślnego konstruktora przez uczynienie go prywatnym, aby zapobiec stosowaniu operatora `new` w stosunku do klasy implementującej wzorec Singleton.
- W drugim należy utworzyć statyczną metodę kreacyjną, która będzie pełniła rolę konstruktora. Metoda ta wywoła prywatny konstruktor, aby utworzyć instancję obiektu i umieści go w polu statycznym klasy. Kolejne wywołania tej metody zwrócą już istniejący obiekt.

Powyższe rozwiązanie spowoduje, że aplikacja może wywołać statyczną metodę klasy i tym samym za każdym razem otrzyma ten sam obiekt [27].

5.1.1 Warstwa widoku

Warstwa widoku, czyli część kliencka aplikacji, została zrealizowana przy użyciu języka JavaScript. Wykorzystuje ona technologię SPA (*Single Page Application*). Oznacza to, że strona wczytywana jest z jednego pliku HTML, którego zawartość jest dynamicznie dostosowywana w trakcie działania aplikacji. Do jej implementacji wykorzystano bibliotekę React.js, która umożliwiła pisanie swoich własnych, uniwersalnych komponentów, które następnie posłużyły do budowania widoków interfejsu użytkownika. Reactowe komponenty przyczyniły się do ich wielokrotnego użycia. Jednym z przykładów takiego komponentu jest *Advertisement*, przedstawia go Listing 2.

Listing 2. Przykładowy komponent Advertisement. Źródło: Opracowanie własne.

```
const Ad = ({ ad, isSearchAd }) => {
  const classes = useStyles({ isSearchAd });
  const [failedToLoad, setFailedToLoad] = useState(false);
  return (<Grid item xs={12} sm={6} md={4} lg={3} className={classes.grid}>

    <Card
      component="a"
      href={` /ad/${isSearchAd ? "search" : "rental"}/${ad.id}`}
      target="_blank"
      elevation={0}
      className={classes.root}
      id={`rental-ad-${ad.id}`}
    >

    {!isSearchAd ? (
      ad.imagesIdList && !failedToLoad ? (
        <img src={` ${images.get}/${ad.imagesIdList[0]} `}
          alt={ad.title}
          className={classes.image}
          onError={() => setFailedToLoad(true)}
        />
      ) : (
        <Box className={classes.image}>
          <Typography className={classes.searchIcon}>?</Typography>
        </Box>
      )
    )}
```

```

    )
  ) : (
    <Box className={classes.image}>
      <SearchIcon className={classes.searchIcon} />
    </Box>
  )}
  <Box className={classes.bottomRoot}>
    <Typography className={classes.title}>{ad.title}</Typography>
    <Box>
      <Box className={classes.iconBox}>
        <RoomIcon className={classes.icon} />
        <Typography className={classes.iconText}>
          {ad.location && ad.location.split(",").join(", ")}
        </Typography>
      </Box>
      <Box className={classes.priceBox}>
        {ad.pricePerDay && (
          <Typography className={classes.price}>
            {ad.pricePerDay.toFixed(0)} zł
          </Typography>
        )}
      </Box>

      {ad.promoted && (
        <Box display="flex" alignItems="center">
          <StarIcon className={classes.star} />
          <Typography className={classes.promotedText}>
            Ogłoszenie promowane
          </Typography>
        </Box>
      )}
    </Box>
  </Box>
</Card>
</Grid>

```

Jako właściwości przyjmuje obiekt z danymi ogłoszenia, które ma zostać wyświetlone oraz opcjonalną wartość logiczną `bool` służącą do identyfikacji czy dane ogłoszenie jest ogłoszeniem poszukiwania. Listing 3 przedstawia ten komponent, który wykorzystywany jest do wyświetlania:

- ogłoszenia na stronie głównej
- wyników wyszukiwania

Listing 3. Zastosowanie komponentu Ad w wynikach wyszukiwania. Źródło: Opracowanie własne.

```

{!isLoading && results.length > 0 && (
  <Grid container spacing={3} className={classes.grid}>
    {results.map((ad) => (
      <Ad key={`ad-${ad.id}`} ad={ad} isSearchAd={isSearchAd} />
    ))}
  </Grid>
)}

```

Oprócz własnych komponentów, tworzonych na potrzeby aplikacji, wykorzystywane są również gotowe rozwiązania oferowane przez bibliotekę MUI (Material-UI). Listing 4 i 5 przedstawiają

autorski komponent *SearchResults*, służący do prezentacji wyników wyszukiwania ogłoszeń. Jest on zbudowany m.in z `<Grid>` `<Box>` czy `<Typography>`, które pochodzą właśnie z MUI.

Listing 4. Komponent SearchResults - definicja. Źródło: Opracowanie własne.

```
const SearchResults = ({
  results,
  page,
  pages,
  changePage,
  isSearchAd,
  isLoading,
}) => {
  const classes = useStyles();
  return (
    <React.Fragment>
      <Box className={classes.paginationRoot}>
        <Pagination
          page={page}
          count={pages}
          showFirstButton
          showLastButton
          color="primary"
          onChange={(e, v) => changePage(v)}
        />
      </Box>
      {isLoading && (
        <Box className={classes.progressRoot}>
          <CircularProgress size={60} />
        </Box>
      )}
      {!isLoading && results.length === 0 && (
        <Box className={classes.progressRoot}>
          <Typography>Nie znaleziono wyników ;(</Typography>
        </Box>
      )}
      {!isLoading && results.length > 0 && (
        <Grid container spacing={3} className={classes.grid}>
          {results.map((ad) => (
            <Ad key={`ad-${ad.id}`} ad={ad} isSearchAd={isSearchAd} />
          ))}
        </Grid>
      )}
    </React.Fragment>
  );
};
```

Listing 5. Zastosowanie SearchResults na stronie głównej aplikacji. Źródło: Opracowanie własne.

```
return (
  <Box width="100%" maxWidth="1500px">
    <Search />
    <Tabs
      className={classes.tabs}
      classes={{ flexContainer: classes.flex }}
      value={isSearchAd}
      onChange={(e, v) => setIsSearchAd(v)}
    >
```



```

        <Tab value={0} label="Ogłoszenia wypożyczenia" />
        <Tab value={1} label="Ogłoszenia poszukiwania" />
    </Tabs>
    <SearchResults
      results={ads}
      isSearchAd={isSearchAd}
      page={page}
      pages={pages}
      changePage={setPage}
      isLoading={isLoading}
    />
  </Box>
);

```

Komunikacja z serwerem odbywa się za pomocą biblioteki Axios. Listing 6 przedstawia jej definicję w programie.

Listing 6. Definicja stałej request. Źródło: Opracowanie własne.

```

const request = axios.create({ baseURL:
  `${process.env.REACT_APP_BASE_URL}/api` });

request.interceptors.request.use((config) => {
  const accessToken = window.localStorage.getItem("apiKey");
  config.headers = {
    Authorization: accessToken ? "Bearer " + accessToken : undefined,
  };
  return config;
});

```

Do stałej request przypisana jest nowa instancja axiosa, która konfiguruje bazowy adres URL serwera oraz nagłówki, zawierające tokeny uwierzytelniające użytkowników.

Listing 7 przedstawia funkcję, pobierającą z serwera listę ogłoszeń wypożyczenia do wyświetlenia po stronie frontendu.

Listing 7. Pobieranie listy ogłoszeń wypożyczenia. Źródło: Opracowanie własne.

```

const getRentalAdsList = () =>
  request.get(rentalAds.getAll).then((response) => response.data);

```

Korzystamy ze wcześniej zdefiniowanej stałej request i wykonujemy żądanie HTTP typu GET. Jako parametr przekazywany jest dokładny adres endpointu zapisany jako stała *rentalAds*, której definicja jest wydzielana do oddzielnego pliku. Następnie w ciele funkcji *then()* mamy dostęp do obiektu *response*, którego składowa *data* zawiera dane ogłoszeń wypożyczenia.

5.1.2 Warstwa kontrolera

Realizacja warstwy kontrolera została podzielona na dwie części:

- część kontrolerów tj. wszystkie klasy z adnotacją `@RestController`
- część serwisową tj. wszystkie klasy oznaczone stereotypem `@Service`

Klasy kontrolera są komponentami, które umożliwiają aplikacji webowej tudzież mobilnej na komunikację z serwerem. Inaczej mówiąc tworzą punkt dostępu do poszczególnych zasobów czy funkcjonalności aplikacji. Prezentuje to listing 8.

Listing 8. Przykład klasy kontrolera wypożyczeń HireController. Źródło: Opracowanie własne.

```
@RestController
@RequestMapping("/api/hires")
public class HireController {
    private HireService hireService;

    @Autowired
    public HireController(HireService hireService) {
        this.hireService = hireService;
    }

    @GetMapping
    public List<HireResponseDTO> getAllHires() {
        return hireService.getAllHires();
    }

    @GetMapping("/{id}")
    public HireResponseDTO getHire(@PathVariable Long id) {
        return hireService.getHire(id);
    }

    @PostMapping("/estimate")
    public CalculateHireCostDTO calculateHireCost(@RequestBody
CalculateHireCostRequestDTO request) {
        return hireService.calculateHireCost(request);
    }

    @PostMapping()
    public void createHire(@RequestBody HireRequestDTO hire, @RequestHeader
String authorization) {
        hireService.createHire(hire, authorization);
    }

    @PutMapping("/{id}")
    public void updateHire(@PathVariable Long id, @RequestBody LocalDate
dateToActual) {
        hireService.updateHire(id, dateToActual);
    }

    @DeleteMapping("/{id}")
    public void deleteHire(@PathVariable Long id) {
```

```

hireService.deleteHire(id);
    }

    @GetMapping("myBorrows")
    public List<HireMyBorrowsResponseDTO> getMyBorrows(@RequestHeader
String authorization, @RequestParam(required = false) HireStatus
hireStatus) {
        return hireService.getMyBorrows(authorization, hireStatus);
    }

    @GetMapping("myShares")
    public List<HireMyBorrowsResponseDTO> getMyShares(@RequestHeader String
authorization, @RequestParam(required = false) HireStatus hireStatus) {
        return hireService.getMyShares(authorization, hireStatus);
    }

    @PostMapping("{id}/package/outgoing")
    public void addOutgoingShipmentNumber(@PathVariable Long id,
@RequestHeader String authorization, @RequestBody String
outgoingShipmentNumber) {
        hireService.addOutgoingShipmentNumber(id, authorization,
outgoingShipmentNumber);
    }

    @PostMapping("{id}/package/incoming")
    public void addIncomingShipmentNumber(@PathVariable Long id,
@RequestHeader String authorization, @RequestBody String
incomingShipmentNumber) {
        hireService.addIncomingShipmentNumber(id, authorization,
incomingShipmentNumber);
    }

    @PutMapping("{id}/package/confirm")
    public void confirmReceivingShipment(@PathVariable Long id,
@RequestHeader String authorization) {
        hireService.confirmReceivingShipment(id, authorization);
    }
}

```

Zdefiniowane są w nim endpointy, które będą obsługiwać przychodzące żądania dotyczące właśnie wypożyczeń tj. pozwalające na wykonywanie podstawowych operacji typu CRUD oraz:

- obliczanie kosztu wypożyczenia
- dodanie numeru przesyłki wychodzącej
- dodanie numeru przesyłki zwrotnej
- potwierdzanie odbioru przesyłki

Springowa adnotacja, znajdująca się nad nazwą klasy tj. *@RequestMapping('/api/hires')* określa bazową ścieżkę URL dostępu do funkcjonalności znajdujących się w *HireController*. Ponadto każda z metod jest oznaczona dodatkowym stereotypem, określającym typ żądania HTTP z opcjonalną ścieżką dostępu, w której mogą znajdować się parametry żądania tzw. *PathVariable*.

Klasy kontrolera służą tak naprawdę wyłącznie do odpowiedniego kierowania żądań wraz z ewentualnymi danymi, które przychodzą na serwer w formacie JSON. Natomiast cała logika biznesowa została wydzielona do odpowiednich metod znajdujących się w klasach serwisowych.

Klasy ze stereotypem `@Service` są miejscem, gdzie zachodzi cała logika biznesowa aplikacji jak m.in. walidacja uprawnień, dostęp do danych przez repozytoria (DAO), tworzenie nowych obiektów i stosownych relacji między nimi, implementacja algorytmów typowo domenowych jak obliczanie kosztu wypożyczenia co przedstawia Listing 9.

Listing 9. Algorytm obliczający koszt wypożyczenia. Źródło: Opracowanie własne.

```
public CalculateHireCostDTO calculateHireCost(CalculateHireCostRequestDTO
request) {
    RentalAd ad = AService.Get(RentalAd.class, rentalAdRepo,
request.getRentalAdId());

    if (request.getStartDate() == null || request.getEndDate() == null)
        throw new Hire.MissingDates();

    if (request.getEndDate().isBefore(request.getStartDate()))
        throw new Hire.InvalidDate("endDate must be after startDate");

    Integer numberDaysElapsed = (int)
ChronoUnit.DAYS.between(request.getStartDate(), request.getEndDate()) + 1;
    Optional<Integer> discountValue =
discountRepo.getBestAvailableDiscount(ad.getId(), numberDaysElapsed);
    CalculateHireCostDTO response = new CalculateHireCostDTO();

    if (discountValue.isPresent()) {
        Integer discountVal = discountValue.get();
        Discount discount =
discountRepo.getFirstByValueAndRentalAdId(discountVal, ad.getId());
        DiscountDTO discountDTO = mapper.map(discount, DiscountDTO.class);
        response.setUsedDiscount(discountDTO);
    }

    Double deposit = ad.getDepositAmount();
    Double totalCost = calculate(ad, discountValue, numberDaysElapsed);
    response.setPeriod(numberDaysElapsed);
    response.setCost(totalCost);
    response.setDepositAmount(deposit);
    response.setPricePerDay(ad.getPricePerDay());
    response.setCommission(commission(totalCost));

    return response;
}
```

Na początku pobierana jest encja ogłoszenia z repozytorium. Następnie walidowana jest poprawność podanej przez użytkownika daty wypożyczenia i obliczana jest jego długość. Po sprawdzeniu czy przysługuje zniżka na dane ogłoszenie prywatna funkcja *calculate()*, która jako parametry przyjmuje ogłoszenie, ewentualny rabat oraz długość wypożyczenia wylicza jego koszt.

W ostatnim etapie tworzony jest obiekt DTO (*Data Transfer Object*) zawierający niezbędne dane, które posłużą do prezentacji w warstwie widoku jako odpowiedź na żądanie.

Wydzielenie implementacji logiki biznesowej do oddzielnych klas serwisowych pozwoliło na ustrukturyzowanie kodu aplikacji. Każdy komponent jest odpowiedzialny za coś innego. Można powiedzieć, że klasa kontrolera pełni funkcję swego rodzaju „kierownika”, który przyjmuje zlecenie i wskazuje na odpowiedniego „wykonawcę”, czyli właśnie serwis. Dzięki temu kod jest dużo bardziej czytelny, łatwiej jest go refaktoryzować oraz wzrasta możliwość ponownego użycia np. konkretnej metody.

5.2. Pozostałe wzorce projektowe

Rozdział opisuje kilka innych przykładowych wzorców projektowych.

5.2.1 Fasada

Fasada wzorzec strukturalny ma za zadanie stworzyć uproszczony interfejs, który jest już wyposażony w frameworki, biblioteki czy inne złożone klasy [28]. Listing 10 prezentuje definicję tego wzorca. Zazwyczaj wykorzystujemy ten wzorzec, kiedy klient nie musi znać zasady działania danych funkcjonalności lub nie chcemy dawać dostępu do wewnętrznych metod podsystemu [29].

Listing 10. Wdrożenie wzorca fasada na przykładzie klasy Authorization. Źródło: Opracowanie własne.

```
public interface Authorization{
    public boolean isModerator(String token);
    public boolean isAdmin(String token);
    public boolean isCurrentUser(User user, String token);
    public Long getIdUserFromToken(String token);
    public Role getRoleFromToken(String token);
    public String getEmailFromToken(String token);
    public String generateToken(String email, Long id, Role role);
    public String generateRefreshToken();
}
```

5.2.2 Data Access Object

Data Access Object (DAO) [30] jest wzorcem strukturalnym, prezentuje go listing 11. Pozwala na odizolowanie warstwy aplikacji od warstwy trwałości. Polega na ukryciu przed warstwą aplikacji złożoności związanych z operacjami CRUD. Dzięki temu obie warstwy mogą ewoluować oddzielnie, nie wiedząc o sobie nic nawzajem.

Listing 11. Przykład zastosowania wzorca DAO. Źródło: Opracowanie własne.

```
public interface DiscountRepo extends JpaRepository<Discount, Long>{

String GET_BEST_AVAILABLE_DISCOUNT = "SELECT MAX(VALUE) FROM DISCOUNT WHERE
RENTAL_AD_ID = :rentalAdId AND MIN_NUM_DAYS <= :elapsedDays";

@Query(value=GET_BEST_AVAILABLE_DISCOUNT, nativeQuery=true)
Optional<Integer> getBestAvailableDiscount(Long rentalAdId, Integer
elapsedDays);

Discount getFirstByValueAndRentalAdId(Integer value, Long rentalAdId);
}
```

5.2.3 Data transfer Object

Data Transfer Object (DTO) [31] reprezentuje wzorzec dostępu do danych, który jest przedstawiony na listing 12. Ma za zadanie zoptymalizowanie kosztu przesłania informacji. Powstaje po to, aby nie wysyłać całego obiektu.

Listing 12. Przykład zastosowania wzorca DTO. Źródło: Opracowanie własne.

```
public class DiscountDTO{
private Integer value;
private Integer minNumDays;
public Integer getValue(){
return value;
}

public void setValue(Integer value){
this.value = value;
}

public Integer getMinNumDays(){
return minNumDays;
}

public void setMinNumDays(Integer minNumDays){
this.minNumDays = minNumDays;
}
}
```

5.2.4 Wstrzykiwanie zależności

W projekcie zostało zaimplementowane wstrzykiwanie przez konstruktor z wykorzystaniem adnotacji Spring `@Autowired`. Listing 13 prezentuje owe wstrzykiwanie.

Listing 13. Przykład zastosowania wstrzykiwania zależności. Źródło: Opracowanie własne.

```
@Service
@Transactional
public class DiscountService{

    private final ModelMapper mapper
    private DiscountRepo discountRepo;
    @Autowired

    public DiscountService(DiscountRepo discountRepo){
        this.discountRepo = discountRepo;
        mapper = new ModelMapper();
    }
}
```

5.3. Komunikacja z bazą danych

Komunikacja z bazą danych odbywa się przy pomocy framework-a Hibernate. Jest to projekt open source. Daje możliwość odwzorowania klas na relacyjną bazę danych za pomocą odpowiednich adnotacji [32]. Dzięki temu diagram klas sporządzony w notacji UML ma swoje odwzorowanie w bazie danych. Specjalne adnotacje zapewniają też możliwość implementowania asocjacji dwukierunkowych, co prezentuje listing 14.

Listing 14. Przykładowe zastosowania adnotacji w projekcie. Źródło: Opracowanie własne.

```
@Entity
public class Claim{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(unique = true, nullable = false)
    private Long id;
    @Column(nullable = false)
    private LocalDate claimDate;
    @Column(nullable = true)
    private String damageDescription;
    @Enumerated(EnumType.STRING)
    @Column(name = "status")
    private ClaimStatus status = ClaimStatus.SUBMITTED;
    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;
}
```

Łączenie z bazą danych odbywa się dzięki *JpaRepository*, który dostarcza także niejawną implementację podstawowych zapytań do bazy [32]. JPA pozwala na tworzenie zapytań na podstawie nazw metod za pomocą refleksji [33]. Przykładowe zastosowanie JPA prezentuje listing 15.

Listing 15. Przykładowe zastosowanie JpaRepository. Źródło: Opracowanie własne.

```
@Repository

public interface BlockedIpRepo extends JpaRepository<BlockedIp, Long>{
    @Query(value = "SELECT IP FROM BLOCKED_IP ", nativeQuery = true);
    List<String> findAllBlockedIp ();
    @Query(value = "SELECT * FROM BLOCKED_IP WHERE upper(IP) like
        upper(concat('%', :word, '%')) ", nativeQuery = true)

    List<BlockedIp> findAllBlockedIp(
        Pageable page, @Param("word") String word);
    Optional<BlockedIp> findByIp(String ip);
    Optional<BlockedIp> deleteByIp(String ip);
    Long countByIpContaining(String ip);
}
```

5.4. Mechanizmy zapewniające bezpieczeństwo

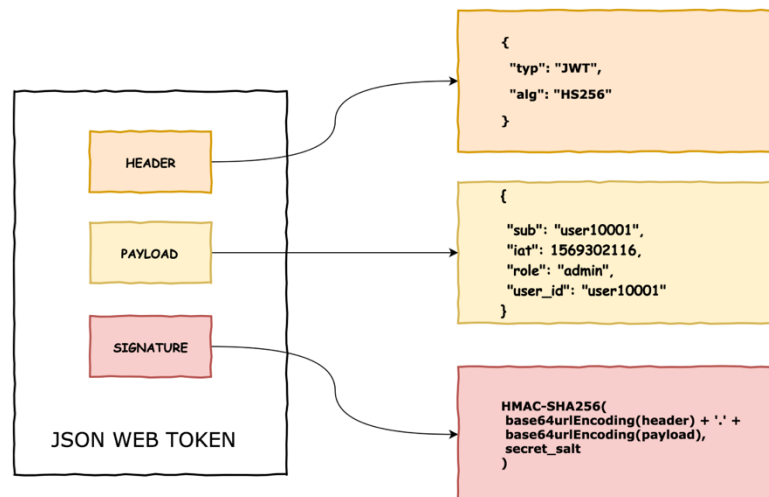
W celu prawidłowego funkcjonowania aplikacji musi ona przetwarzać wrażliwe dane użytkowników. Aby mogły być bezpiecznie przechowywane zostały zaimplementowane mechanizmy zapewniające ich bezpieczeństwo. Dostęp do nich mają tylko uprawnione osoby.

5.4.1 Logowanie do systemu

Logowanie do naszego systemu odbywa się poprzez wprowadzenie adresu e-mail użytkownika oraz hasła. Podczas tworzenia nowego hasła wymagane jest, aby posiadało minimum 8 znaków. W celu zabezpieczenia aplikacji przed niepożądanym dostępem do konta użytkownika podczas odzyskiwania hasła system nie podaje czy podany adres e-mail jest prawidłowy. Po wprowadzeniu adresu e-mail do odzyskania hasła do użytkownika na jego skrzynkę wysyłana jest wiadomość z linkiem do ustawienia nowego hasła. Dzięki temu rozwiązaniu zabezpieczamy hasło użytkownika przed ewentualnym wyciekiem w przypadku przejęcia jego skrzynki e-mail. Dodatkowym zabezpieczeniem przed przejęciem konta użytkownika jest automatyczne wylogowanie go w przypadku niepodjęcia przez niego żadnej akcji przez określony czas.

5.4.2 JWT Token

JSON Web Token to standard RFC 7519, który definiuje sposób wymiany danych między stronami w bezpieczny sposób poprzez obiekt JSON. Informacje, które są wymieniane mogą być zweryfikowane dzięki cyfrowemu podpisowi, który jest częścią tokenu. W aplikacji wykorzystujemy go do autoryzacji podczas logowania. Rysunek 29 prezentuje strukturę JWT Token.



Rysunek 29. Struktura JWT Token. Źródło: <https://medium.com/@sureshsk>.

5.4.3 Dostęp do funkcjonalności wymagających logowania

Po stronie Front-end'u został zastosowany *ProtectedRoute*, który jest oparty na rolach. Jest to komponent z paczki React'a „react-router-dom”. *ProtectedRoute* ma za zadanie kontrolować dostęp dla ról oraz dla konkretnych statusów użytkowników (Zalogowany, Niezalogowany). Implementacja tej klasy jest przedstawiona na listing 16.

Właściwość *allowedRoles* pozwala nam zdefiniować rolę, dla której dostępna jest zdefiniowana ścieżka. *NoLogin* oznacza, że ścieżka dostępna jest wyłącznie dla użytkowników niezalogowanych, path to adres URL pod którym dostępna jest zdefiniowana ścieżka.

Listing 16. Klasa *ProtectedRoute* oraz najważniejsze elementy z nią związane. Źródło: Opracowanie własne.

```
import React from "react";
import { Redirect, Route } from "react-router-dom";
import { useSelector } from "react-redux";
const ProtectedRoute = ({ noLogin, allowedRoles, children, ...props }) => {
  //Informacja o użytkowniku jest przechowywana w Redux

  const isLoggedIn = useSelector((state) => state.userReducer.isLoggedIn);
  const role = useSelector((state) => state.userReducer.role);
  const validateAccess = () => {
```

```

    //Ścieżka jest dostępna tylko dla niezalogowanych użytkowników a
    użytkownik jest zalogowany

    //Brak dostępu

    if (isLoggedIn && noLogin) {
        return false;
    }

    //Ścieżka jest dostępna tylko dla zalogowanych użytkowników a
    użytkownik jest zalogowany

    //Brak dostępu

    if (!isLoggedIn && !noLogin) {
        return false;
    }

    //Sprawdzenie czy zostały podane role które mogą odwiedzić daną ścieżkę

    if (allowedRoles) {
        //Udzielenie dostępu w zależności czy rola użytkownika jest na liście
        return allowedRoles.includes(role);
    }
    //Brak dodatkowych wymagań, użytkownik może odwiedzić ścieżkę
    return true;
};

return (
    <Route
        {...props}
        render={() => (validateAccess() ? children : <Redirect to="/" />)}
    />
);
};
export default ProtectedRoute;

```

W aplikacji zostały stworzone role do których przypisywani są konkretni typy użytkowników, w systemie stworzono 5 ról:

- Admin,
- User,
- Moderator,
- Appraiser,
- Insurance_adjuster

Do każdej roli nadawane są konkretne uprawnienia, którymi może się posługiwać.

Po stronie back-end'u został zastosowany Spring Security, jego konfiguracje możemy znaleźć w klasie *SecurityConfig*, która dziedziczy klasę *WebSecurityConfigurerAdapter*. W klasie zastosowana została metoda *configure()*, w która odpowiada za przypisanie odpowiednich dostępu dla konkretnych

adresów URL wymienionym wcześniej rołom. Część klasy *SecurityConfig*, w której umieszczona została metoda została przedstawiona w listing 17.

Listing 17. Fragment klasy SecurityConfig z metodą configure(). Źródło: Opracowanie własne.

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.cors().and()
        .csrf().disable()
        .headers().frameOptions().disable()
        .and()
        .sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
        .and()
        .addFilter(new JwtCreatorFilter(authenticationManager()))
        .addFilterAfter(new JwtInspectorFilter(secret),
JwtCreatorFilter.class)
        .addFilterAfter(new SecurityFilter(),
JwtInspectorFilter.class)
        .authorizeRequests()
        .antMatchers("/swagger-ui.html").permitAll()
        .antMatchers("/api/products/").permitAll()
        .antMatchers("/api/products/myProducts").authenticated()
        .antMatchers(POST, "/api/users").permitAll()
        .antMatchers(PUT, "/api/users/**").authenticated()
        .antMatchers("/api/users/details").authenticated()
        .antMatchers("/api/users/premium").authenticated()
        .antMatchers("/console").permitAll()
        .antMatchers("/console/**").permitAll()
        .antMatchers(GET, "/api/rentalAd").permitAll()
        .antMatchers(GET,
"/api/rentalAd/getStatus/{\\d+}").authenticated()
        //
        .antMatchers(GET,
"/api/searchingAd/{\\d+}").authenticated()
        .antMatchers("/api/rentalAd/getStatus").authenticated()
        .antMatchers("/api/vulgarWords/**").hasAnyAuthority(ADMIN,
OPERATOR)
        .antMatchers("/api/blockedIp/**").hasAnyAuthority(ADMIN,
MODERATOR)
        .antMatchers("/api/users/**").hasAnyAuthority(ADMIN,
INSURANCE_ADJUSTER)
        .antMatchers("/api/insuranceadjuster/**").hasAuthority(INSURANCE_ADJUSTER)
        .antMatchers("/api/inspection/**").hasAuthority(APPRAISER);
}
```

5.5. Testowanie aplikacji

Historia informatyki oraz wszelkiego rodzaju systemów informatycznych pokazuje, że odpowiednie przetestowanie stworzonego oprogramowania jeszcze na etapie produkcji jest bardzo ważne. Użycie stosownych technik testowania oprogramowania odpowiednio zminimalizuje liczbę możliwych błędów znalezionych przez potencjalnych użytkowników serwisu. Testy rozpoczynane są podczas implementacji pierwszej funkcjonalności serwisu. Zminimalizuje to możliwości wytwarzania wadliwego oprogramowania [34].

5.5.1 Manualne testy integracyjne i systemowe.

Kod aplikacji przechowywany jest w zasobach hostingowego serwisu internetowego przeznaczonego do projektów programistycznych - Bitbucket. Kod podzielony jest na trzy zasoby:

- frontend,
- backend
- wersja mobilna.

Odnosić do aktualnych repozytoriów kodu udostępnione są, wraz z innymi istotnymi informacjami oraz dokumentacją aplikacji, na dysku wspólnym, a dostęp do nich mają wszyscy członkowie zespołu projektowego.

Testowanie działania aplikacji rozpoczyna się od pobrania najbardziej aktualnego wydania za pomocą rozproszonego systemu kontroli wersji Git. Następnie, tester uruchamia lokalnie silnik aplikacji w środowisku programistycznym, np. IntelliJ firmy JetBrains.

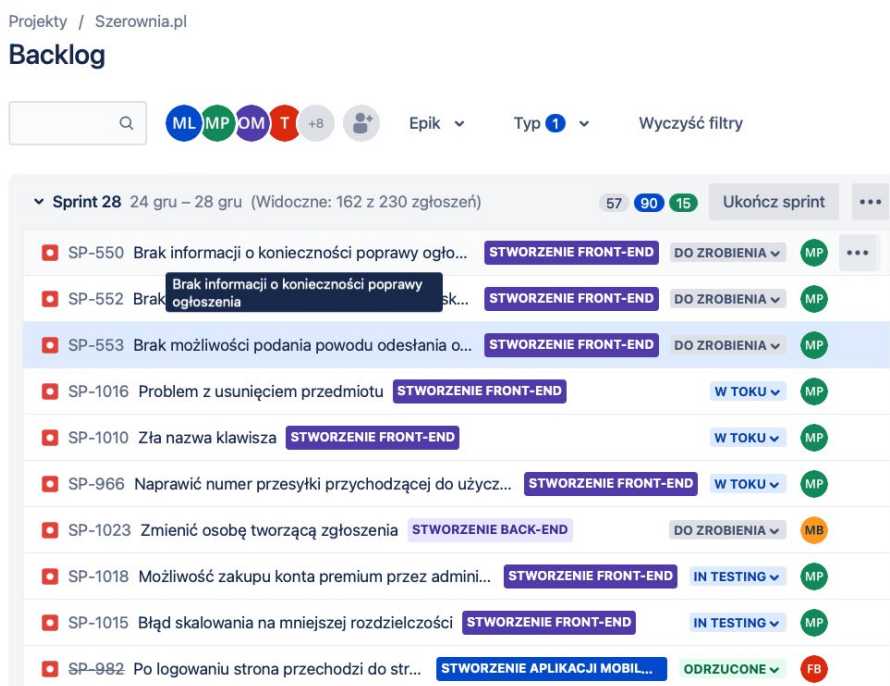
Aby przetestować aplikację w wersji przeglądarkowej, kod odpowiedzialny za frontend instalowany jest i uruchamiany w środowisku programistycznym. Szerownia.pl jest wówczas dostępna z poziomu przeglądarki internetowej na jednym z portów localhost.

Dla wersji mobilnej, zespół programistów przygotowuje wydanie w wersji skompilowanej. Tester uruchamia gotową aplikację za pomocą emulatora wbudowanego w oprogramowanie Android Studio (testowanie aplikacji mobilnej również wymaga działającego backendu). Narzędzie jest oficjalnym zintegrowanym środowiskiem programistycznym dla systemu operacyjnego Android firmy Google i umożliwia sprawdzenie działania aplikacji z poziomu użytkownika urządzenia mobilnego, w tym zachowanie ekranu dotykowego oraz natywnych przycisków systemu Android.

Scenariusze testów zostały stworzone na podstawie przypadków użycia. Są przechowywane we współdzielonym repozytorium i dostępne dla każdego z członków zespołu testującego. Scenariusze obejmują wszystkie najważniejsze funkcjonalności dla poszczególnych typów użytkowników (użytkownik niezarejestrowany, użytkownik zarejestrowany, moderator, administrator itp.). Uwzględnione zostało pożądane działanie aplikacji dla przewidywalnych, prawidłowych działań użytkownika, jak również dla sytuacji, w których np. wprowadzone są niekompletne lub nieprawidłowe dane.

Błędy oraz niezgodności z założeniami, w tym z diagramem klas i diagramem przypadków użycia są zamieszczane w systemie zgłoszeniowym. Zespół programistów analizuje przekazane błędy

i uwzględnia poprawki w kolejnych wydaniach aplikacji, oznacza również problemy, które zostały już rozwiązane. Rysunek numer 30 przedstawia backlog błędów zgłoszonych podczas procesu testowania.



Rysunek 30. Widok backlogu błędów z JIRA. Źródło: Opracowanie własne.

Aplikacja jest narzędziem spójnym i nie składa się z odrębnych podsystemów. Testy w ujęciu systemowym są prowadzone pod kątem prawidłowego współdziałania komponentów. Przykładem może być użycie aplikacji przez użytkownika, który korzysta z wyszukiwarki, aby odnaleźć interesujący go produkt, następnie zgłasza ogłoszenie tego przedmiotu do moderacji, a moderator weryfikuje zasadność zgłoszenia i odsyła je właścicielowi do poprawy. W ten sposób testowana jest jedna z typowych ścieżek, przy jednoczesnym sprawdzeniu kilku przypadków użycia.

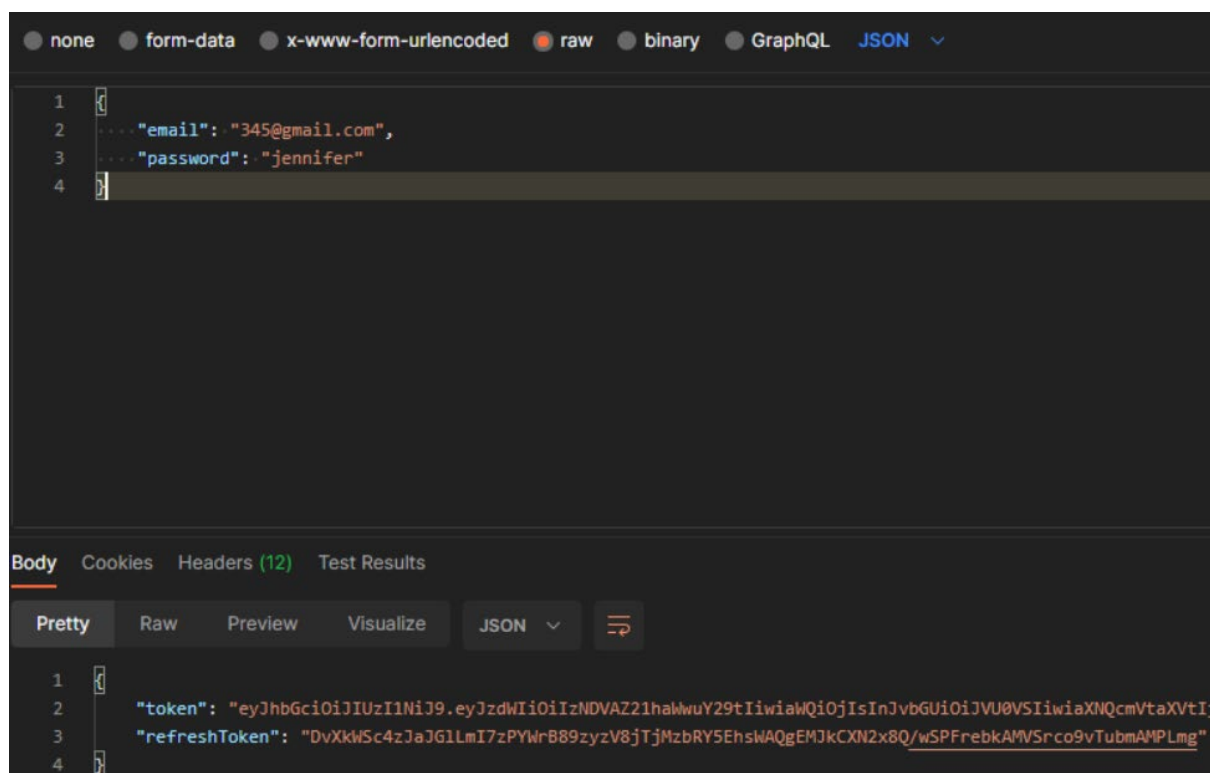
Takie podejście pozostaje w zgodzie z prowadzeniem testów integracyjnych typu mieszanego, które łączy w sobie testowanie bottom-up z podejściem top-down. Oznacza to, że jednocześnie testowane są komponenty najniższego i najwyższego poziomu, na przykład dodawanie zdjęcia dla ogłaszanego przedmiotu oraz cały proces ogłaszania wypożyczanego produktu. Ten typ testów skoncentrowany jest na potwierdzeniu działania określonych funkcjonalności oprogramowania. Powyższe podejście wymaga kompletności scenariuszy testowych w odniesieniu do przypadków użycia aplikacji.

5.5.2 Manualne testy integracyjne i systemowe.

Do przeprowadzania manualnych testów API służył program Postman. Za jego pomocą, zespół deweloperów był w stanie budować odpowiednie żądania HTTP, a następnie wysyłać je na serwer. Rozwiązanie to umożliwiało sprawdzenie czy zaimplementowana funkcjonalność działa poprawnie tj. czy nie występują żadne defekty natury technicznej oraz czy spełnione są wszelkie wymogi funkcjonalne.

Metoda *getMyRentalAds()* zwraca listę ogłoszeń wypożyczenia danego użytkownika. Jako parametry wejściowe przyjmuje token służący do uwierzytelnienia oraz opcjonalne parametry paginacji, czyli numer strony oraz liczbę jej elementów. Domyślnie wartości te wynoszą odpowiednio 0 oraz 25. W celu przetestowania danej metody należy najpierw wygenerować token czyli zasymulować zalogowanie się osoby na swoje konto w aplikacji. Dlatego budowane jest żądanie HTTP typu POST, które przyjmuje strukturę typu JSON. Znajdują się w niej e-mail oraz hasło użytkownika. Po prawidłowym uwierzytelnieniu otrzymujemy odpowiedź z serwera z wygenerowanym tokenem.

Rysunek numer 31 przedstawia użycie metody do logowania w Postman.



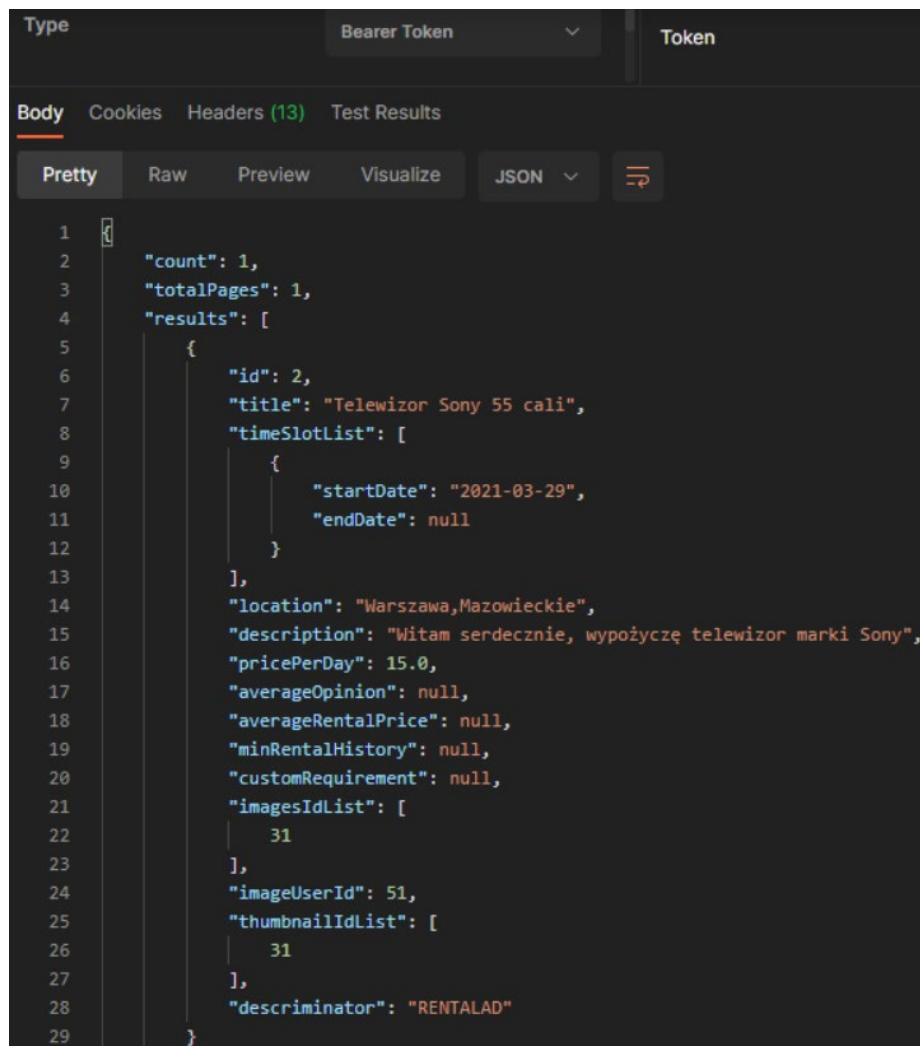
Rysunek 31. Ekran aplikacji Postman prezentujący działanie funkcjonalności logowania. Źródło: Opracowanie własne.

Kolejnym krokiem będzie skonstruowanie żądania sprawdzającego działanie docelowej metody. W tym celu:

- Wpisany jest adres testowanego endpointu
- Wybrany zostaje typ żądania jako HTTP GET
- Z zakładki Authorization -> Type wybrany został Bearer Token

Do sekcji token wklejono wygenerowany wcześniej token.

Rysunek numer 32 prezentuje wywołanie metody `getMyRentalDsa()` w Postman.



Rysunek 32. Wywołanie metody `getMyRentalDsa()`. Źródło: Opracowanie własne.

Po wysłaniu żądania serwer wysłał jako odpowiedź listę ogłoszeń danego użytkownika, której zawartość jest sprawdzana ze stanem bazy danych. Jeśli zwrócone dane są poprawne to można przyjąć, że zaimplementowana funkcjonalność działa prawidłowo.

5.5.3 Testy automatyczne

Oprócz testów manualnych również zostały przeprowadzone testy zautomatyzowane. Ich celem jest sprawdzenie funkcjonalności aplikacji wymagających wprowadzania dużych ilości informacji oraz zautomatyzowanie czynności wykonywanych wielokrotnie. Jest to bardzo duże ułatwienie dla testera, który może zautomatyzować test i wykonać go dowolną ilość razy jednym kliknięciem. Testy zautomatyzowane zostały przeprowadzone w programie IntelliJ IDEA przy użyciu frameworka Selenium.

Selenium jest frameworkiem dostarczającym narzędzie do odtwarzania zadań w celu utworzenia powtarzalnych testów funkcjonalnych. Do przeprowadzenia testu platforma wykorzystuje przeglądarkę internetową Google Chrome. Narzędzie uruchamia aplikację a następnie wykonuje zadania określone w kodzie. Przykład takiego testu prezentuje kod z listing 18 i 19.

Listing 18. Test automatyczny na weryfikację funkcjonalności rejestracji. Źródło: Opracowanie własne.

```
@Test
public void userShouldSuccessfullyRegister() throws Exception{
    new Home()
        .openSignInPage()
        .submitCreateAccountForm()
        .submitFormWithValidData()
        .userShouldBeSuccessfullyRegistered();
}
```

Listing 19. Metoda wybierająca sposób wysyłki przedmiotu. Źródło: Opracowanie własne.

```
private void shippingMethod() throws Exception{
    type.click();
    chooseFromTheList2(personalPickup, "Odbiór osobisty");
    chooseFromTheList2(personalPickup, „Paczkomat”);
}
```


6. Interfejs aplikacji

Przemyślany i przyjemny dla oka, zarazem wygodny i intuicyjny interfejs implementowanej aplikacji ma kluczowe znaczenie by odnieść sukces na rynku. Często się zdarza, że te właśnie cechy wpływają na wybór potencjalnego klienta. Ergonomia korzystania z aplikacji, a czasem nawet liczba kliknięć by przejść przez proces biznesowy mają kluczowe znaczenie dla użytkowników, których aplikacja ma być codziennym narzędziem pracy. Zdarzą się, nierzadko, niechęć pracowników do nowych rozwiązań informatycznych z powodu konieczności przyswojenia obsługi aplikacji i czasu na wdrożenie się. Dlatego tak ważne jest by system na pierwszy rzut oka sprawiał wrażenie przyjaznego i łatwego w użytkowaniu.

Tworząc interfejs aplikacji kierowano się dobrymi praktykami z dziedziny UX/UI oraz podjęto starania by zachować nowoczesność i estetykę [34]. Spójność, czytelność i powtarzalność to cechy, na które zwracano szczególną uwagę przy projektowaniu interfejsów. Responsywność oraz skalowalność nie zostały pominięte by każdy użytkownik, niezależnie od posiadanego sprzętu miał możliwość wygodnie pracować. Dla urządzeń mobilnych powstała dedykowana aplikacja na system operacyjny Android.

6.1 Nawigacja po aplikacji

W niniejszym podrozdziale przedstawiono sposób nawigowania po aplikacji przeglądarkowej oraz rozwiązania wprowadzone w aplikacji mobilnej.

Nawigacja w aplikacji przeglądarkowej. Strona podzielona jest na trzy główne sekcje:

- górna belka (nagłówek) - zablokowana w stałej pozycji podczas przewijania ekranu,
- sekcja zawierająca wybraną treść
- stopka.

Nagłówek i stopka są widoczne, niezależnie od przeglądanych treści. W nagłówku znajdują się takie dane jak:

- nazwa portalu zawierająca hiperłącze do strony głównej,
- przejście do okna logowania.

Dla użytkowników zalogowanych na górnej belce pojawiają się dodatkowe ikony:

- skrzynka wiadomości,
- szczegóły konta.

Stopka zawiera:

- regulamin,
- odnośnik do strony logowania oraz do założenia konta.

Na stronie głównej prezentowana jest lista ogłoszeń z przedmiotami do wypożyczenia. Nawigacja do podstron następuje przez kliknięcie w kafelek przedstawiający poszczególne ogłoszenie. Pojedyncze kliknięcie otwiera nową zakładkę w przeglądarce. Aby wrócić do strony głównej należy zamknąć lub przełączyć się między otwartymi zakładkami. Powyżej kafelków z przedmiotami znajdują się przyciski przełączające widok między ogłoszeniami przedmiotów do użyczenia oraz przedmiotów poszukiwanych. Gdy ilość prezentowanych ogłoszeń przekracza liczbę szesnaście wtedy użytkownik ma do dyspozycji specjalne strzałki umożliwiające, przełączanie się między prezentowanymi ogłoszeniami.

Wyszukiwanie działa na dwa sposoby:

- można wybrać kategorię przedmiotów z listy rozwijalnej,
- można wpisać poszukiwane słowo kluczowe i wyszukać za pomocą ikony strzałki.

Strona ogłoszenia zawiera zdjęcia przedmiotów wypożyczenia. Ikony strzałek pozwalają przesuwac się między fotografiami.

W aplikacji znajdują się następujące narzędzia nawigacyjne:

- przyciski graficzne (piktogramy),
- przyciski tekstowe (także podświetlane po najechniu kursorem),
- kalendarz do wyboru dat dostępności przedmiotu lub pożądanym dat wypożyczenia,
- suwaki (do wyboru stanu przedmiotu lub minimalnej wymaganej średniej opinii użytkownika),
- listy rozwijalne,
- checkboxy.

Dodatkowe elementy służące do nawigacji to:

- natywne rozwiązania przeglądarki, służące do pokazania kolejnej i poprzedniej strony,
- wgrania plików za zdjęciami,
- narzędzia plugin'ów, np. mapy OpenStreetMap [35].

Aplikacja mobilna opiera się na rozwiązaniach natywnych systemu Android. Są to:

- gotowe przyciski nawigacyjne,
- biblioteka gotowych ikon, w tym dostępny wygląd suwaka w postaci gwiazdek do wystawiania oceny.

Dodatkowo można skorzystać z przycisków udostępnianych przez system operacyjny:

- wstecz,
- ekran główny telefonu,
- wybór aktywnej aplikacji.

Użytkownik niezalogowany, po uruchomieniu aplikacji widzi krótką informację na temat narzędzia do wypożyczeń oraz aktualności (na przykład o aktualnie dostępnych nowych funkcjonalnościach). Użytkownik zalogowany rozpoczyna pracę z aplikacją od strony z kafelkami - miniaturami ogłoszeń, a po wyborze interesującego go przedmiotu, może wyświetlić szczegóły. Dysponuje także wyszukiwarką kategorii lub konkretnych przedmiotów. Na dole ekranu znajduje się panel przycisków:

- start - informacje o aplikacji Szerownia.pl
- przeglądaj - lista ogłoszeń
- dodaj - możliwość dodania własnego ogłoszenia
- wiadomości - czat z innymi użytkownikami aplikacji
- konto - szczegóły konta i możliwość wykupienia wersji premium.

6.2. Użytkownik niezalogowany

W aplikacji użytkownicy niezalogowani mają dostęp do podstawowych funkcji systemu. Najważniejszymi z nich są możliwość zalogowania się na swoje konto oraz możliwość zarejestrowania w serwisie.

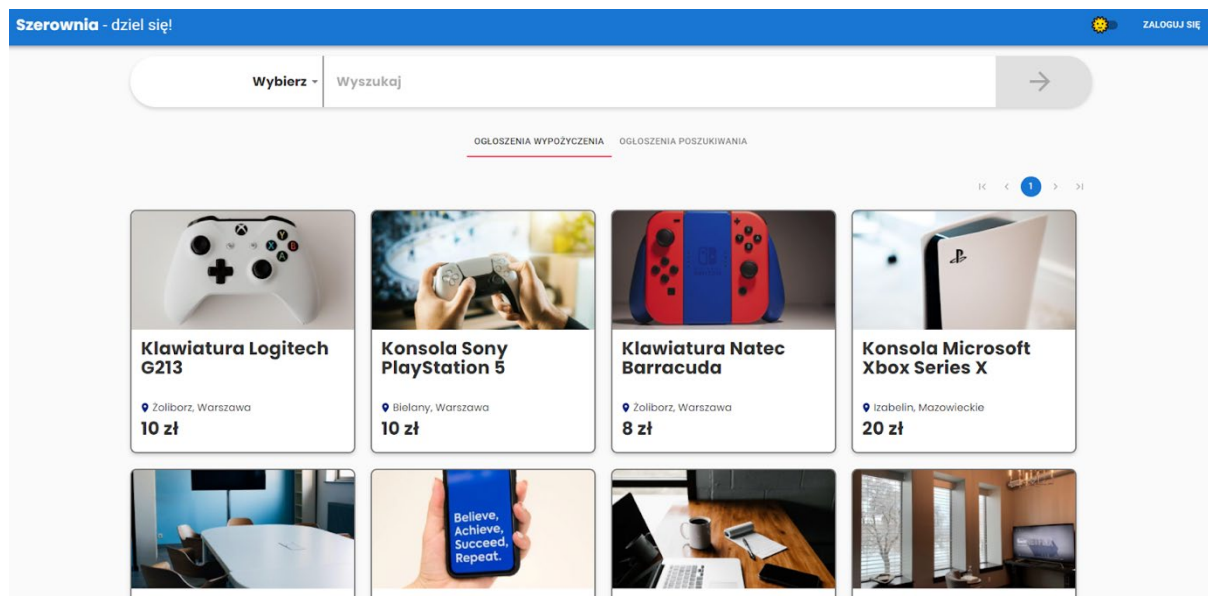
Rysunek 33 przedstawia ekran służący do zalogowania w systemie, na który jest dostępna także funkcjonalność założenia nowego konta oraz przypomnienia hasła.

Rysunek 33. Ekran panelu logowania do aplikacji. Źródło: Opracowanie własne.

Natomiast rysunek 34 pokazuje interfejs służący do zakładania nowego konta.

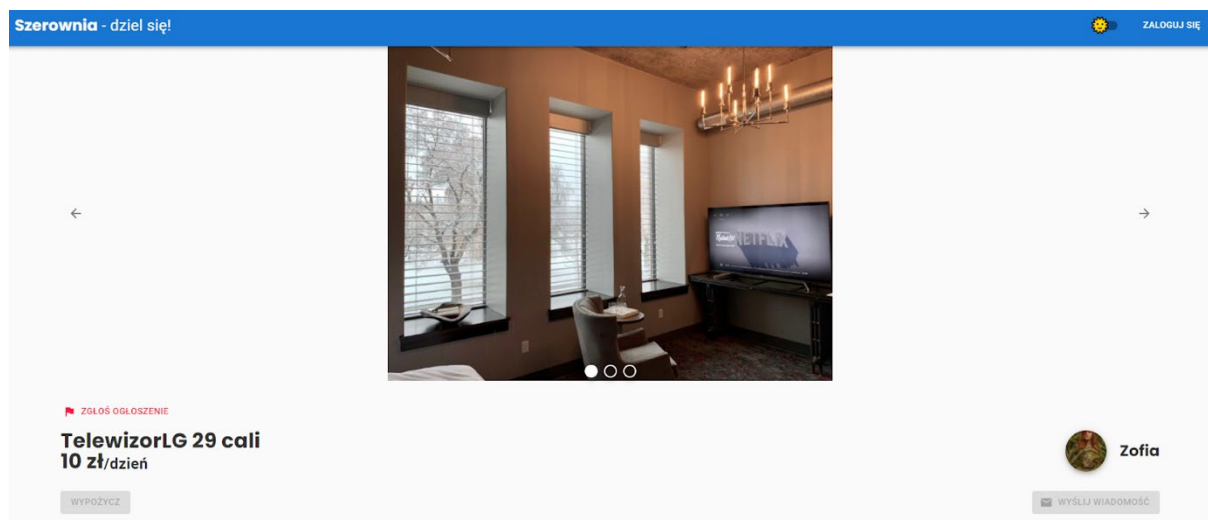
Rysunek 34. Ekran zakładania nowego konta użytkownika. Źródło: Opracowanie własne.

Dodatkowo strona główna daje użytkownikom niezalogowanym dostęp do prezentowanych ogłoszeń z okrojoną możliwością interakcji. Bez zalogowania się do serwisu nie można wypożyczać przedmiotów oraz wysłać wiadomości do właścicieli ogłoszeń. Rysunek 35 przedstawia widok całego okna aplikacji bez zalogowania się.



Rysunek 35. Ekran główny serwisu Szerownia.pl dla użytkownika niezalogowanego. Źródło: Opracowanie własne.

Kolejna ilustracja numer 36 pokazuje widok konkretnego ogłoszenia z perspektywy użytkownika niezalogowanego.



Rysunek 36. Widok szczegółów ogłoszenia dla użytkownika niezalogowanego. Źródło: Opracowanie własne.

6.3. Użytkownik zalogowany i premium

Główne funkcje naszego systemu są dostępne dopiero po zalogowaniu się przez użytkownika do systemu i w tym podrozdziale zostaną omówione.

6.3.1 Użytkownik zalogowany

Przed wszystkim w naszym systemie możemy wyróżnić 5 głównych interfejsów:

- wystawianie ogłoszeń wypożyczenia,
- wypożyczenie przedmiotu,
- wystawienie ogłoszeń poszukiwania,
- zgłaszanie ogłoszeń do moderacji oraz reklamacje.

Interfejs modułu wystawiania ogłoszeń wypożyczenia jest bardzo intuicyjny, każde z pól zostało odpowiednio podpisane, tak żeby użytkownik nie miał najmniejszych wątpliwości przy dodawaniu ogłoszenia wypożyczenia. Mamy możliwość dodania zdjęć naszego przedmiotu oraz wybrania lokalizacji skąd chcemy wypożyczyć nasz przedmiot i okres na jaki nasz przedmiot jest dostępny do wypożyczeń.

Rysunek numer 37 i 38 przedstawia widok dodawania ogłoszenia.

Dodaj nowe ogłoszenie

OGŁOSZENIE UŻYCZENIA

OGŁOSZENIE POSZUKIWANIA

Tytuł

Opis

Cena za dzień zł

Kaucja zł

Cena za dzień opóźnienia zł

☐ Promuj ogłoszenie

Zdjęcia

Pierwsze zdjęcie będzie zdjęciem głównym

Zdjęcia przyjmujemy w formacie .jpg, .albo .png. Maksymalny rozmiar zdjęcia to 5MB.

Rysunek 37. Ekran dodawania nowego ogłoszenia. Źródło: Opracowanie własne.

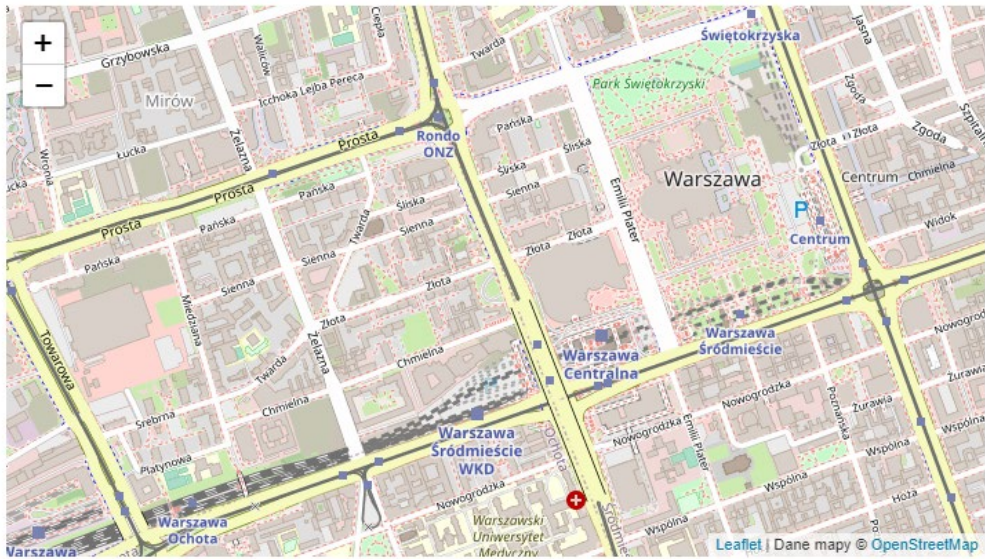
Pod przyciskiem “Dodaj nowy przedmiot” możemy dodać przedmiot, który ma zostać wypożyczony wraz z jego wszelkimi cechami czy funkcjonalnościami. Sposób wysyłki oferuje nam możliwość zaznaczenia jak przedmiot zostanie wypożyczony, a wymagania pozwalają nam zaznaczyć jakie kryteria musi spełniać potencjalny wypożyczający.

Twoja lokalizacja

Wyszukaj

Q WYSZUKAJ

Wyszukaj adres lub zaznacz punkt na mapie



Okresy dostępności

Przedmioty Wybrano 0 przedmiotów

Sposób wysyłki Nie wybrano żadnego

Wymagania

DODAJ NOWE OGŁOSZENIE

ANULUJ

Rysunek 38. c.d. dodawania nowego ogłoszenia. Źródło: Opracowanie własne.

Interfejs wystawiania ogłoszeń poszukiwania został znacznie uproszczony względem wyżej wspomnianego. Znajdziemy na nim miejsce na wpisanie tytułu naszego ogłoszenia oraz dodanie

krótkiego opisu. Tak jak poprzednio możemy dodać lokalizację, w której szukamy przedmiotu, okres wypożyczenia czy też dodać dokładny przedmiot ze wszelkimi jego cechami. Rysunek numer 39 przedstawia formularz dodawania ogłoszenia poszukiwania.

Dodaj nowe ogłoszenie

OGŁOSZENIE UŻYCZENIA

OGŁOSZENIE POSZUKIWANIA

Tytuł

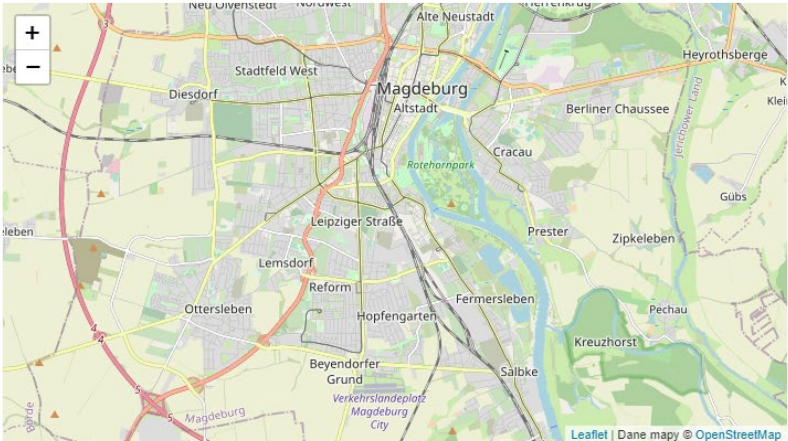
Opis

Twoja lokalizacja

Wyszukaj

WYSZUKAJ

Wyszukaj adres lub zaznacz punkt na mapie



Okresy dostępności

PrzedmiotyWybrano 0 przedmiotów

Sposób wysyłkiNie wybrano żadnego

DODAJ NOWE OGŁOSZENIE

ANULUJ

Rysunek 39. Ekran dodawania ogłoszenia poszukiwania. Źródło: Opracowanie własne.

W module wypożyczenia mamy możliwość wybrania okresu w jakim chcemy dokonać wypożyczenia przedmiotu oraz wybrania sposobu jego dostawy. W ogłoszeniu na przedstawionym rysunku 40 wypożyczający zgadza się na wszystkie opcje dostawy. W przypadku wyboru opcji „Paczkomat” system umożliwi nam wybór paczkomatu, do którego chcielibyśmy, żeby został wysłany

przedmiot. Opcja “Kurier” pozwoli nam na wypełnienie swoich danych adresowych do wysyłki. Natomiast opcja „Odbiór osobisty” da możliwość indywidualnego umówienia się na odbiór przedmiotu.

Dokonaj wypożyczenia

Wybierz kiedy chcesz dokonać wypożyczenia

<

styczeń 2022

>

pon	wt	śr	czw	pt	sob	ndz
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Wybierz sposób dostawy

☒

 Odbiór osobisty

☐

 Kurier

☐

 Paczkomat

WYPOŻYCZ

ANULUJ

Wypożyczasz



iPhone X
Witam, wypożyczę iPhone X. Stan jak na foto

Koszt wypożyczenia

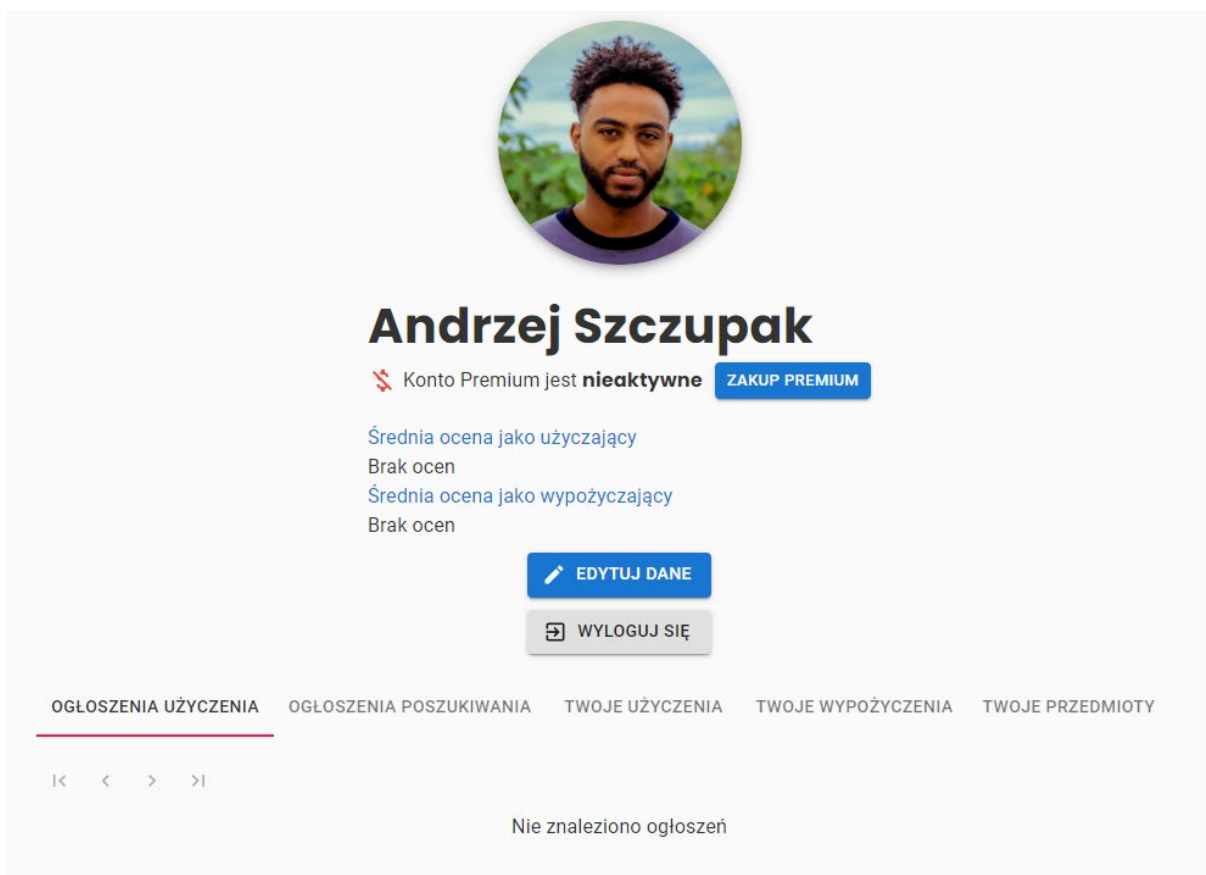
Cena za dzień	35.00 zł
Ilość dni	x 16
Rabat	brak
Kaucja	600 zł
Prowizja	16.80 zł
Razem	1176.80 zł

Rysunek 40. c.d. wypożyczenia przedmiotu, wybór daty i sposobu dostawy. Źródło: Opracowanie własne.

Użytkownik zalogowany może również zgłaszać ogłoszenia, które naruszają regulamin. Funkcjonalność ta jest dostępna pod czerwonym przyciskiem „Zgłoś ogłoszenie”. Wyświetli się okno, w którym będziemy mogli krótko opisać naruszenie regulaminu oraz z listy rozwijanej wybrać powód zgłoszenia.

Mamy również możliwość spojrzenia na własne konto używając przycisku „Twoje konto”, który jest widoczny na każdym ekranie. Znajdują się tam oceny, które wystawili inni użytkownicy, możliwość edycji swoich danych osobowych oraz informacje na temat wszystkich naszych wypożyczeń oraz użyczeń.

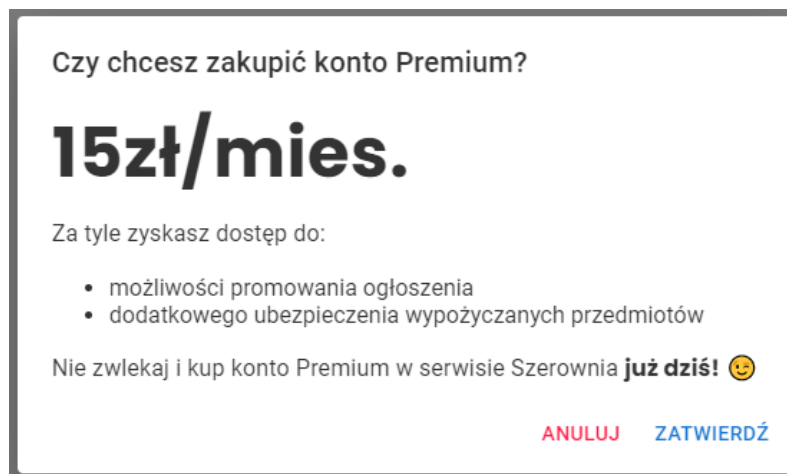
Rysunek numer 41 prezentuje widok po wejściu w „Twoje konto”.



Rysunek 41. Widok po kliknięciu w "Twoje konto". Źródło: Opracowanie własne.

System daje również możliwość złożenia reklamacji pod przyciskiem “Złóż reklamację” na przedmiot, który wypożyczyliśmy. Jest to bardzo ważna funkcjonalność, dzięki której użytkownik wypożyczający może od razu zgłosić wszelkie uwagi do wypożyczonego przedmiotu.

Zgodnie z biznesowymi założeniami, użytkownik zalogowany ma możliwość uzyskania dostępu do dodatkowych funkcjonalności jakie płyną z posiadania konta premium. W tym celu należy wykupić dostęp, co prezentuje rysunek numer 42.

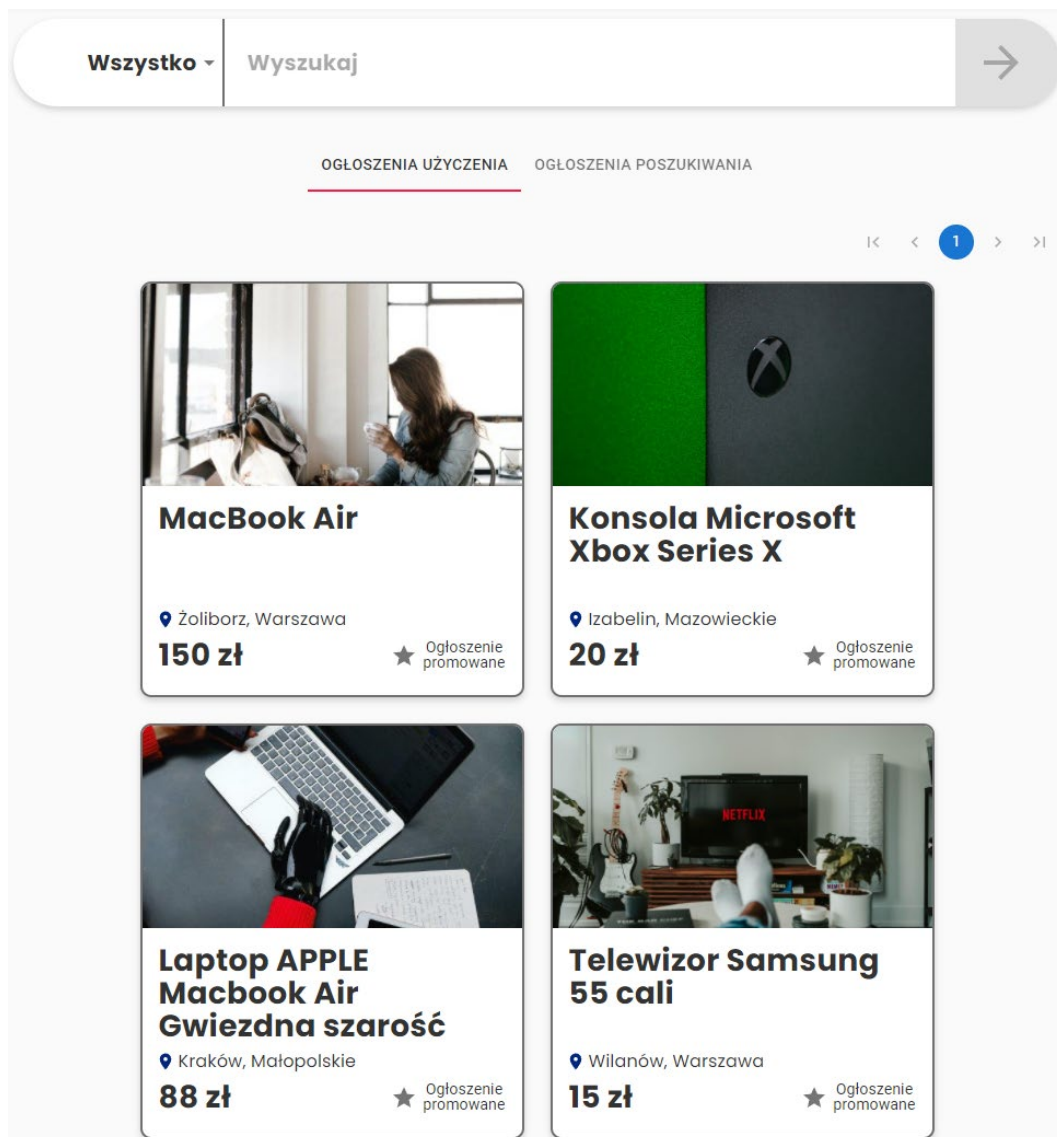


Rysunek 42. Ekran kupowania dostępu do konta premium. Źródło: Opracowanie własne.

6.3.2 Użytkownik premium

Użytkownicy, którzy posiadają już konto w serwisie mogą dodać do niego dodatkowe funkcje dzięki zakupie konta premium na miesiąc. Oczywiście jest możliwość przedłużenia subskrypcji na kolejne miesiące. Dostęp do konta premium został wyceniony na 15 zł miesięcznie. Niesie za sobą możliwość promowania ogłoszeń, dzięki czemu, będą oznaczone jako wyróżnione na samej górze tablicy ogłoszeń.

Rysunek numer 43 przedstawia widok listy ogłoszeń, na której znajdują się ogłoszenia promowane. Można je rozpoznać po znaku gwiazdki i napisie w dolnym prawym rogu ogłoszenia.



Rysunek 43. Widok listy ogłoszeń z promowanymi ogłoszeniami użyczenia. Źródło: Opracowanie własne.

By ogłoszenie pojawiło się na stronie głównej jako promowane użytkownik posiadający konto premium ma dostępny specjalny checkbox podczas dodawania nowego ogłoszenia. Rysunek numer 44 przedstawia formularz dodawania ogłoszenia z zaznaczoną opcją promowania.

Dodaj nowe ogłoszenie

OGŁOSZENIE UŻYCZENIA OGŁOSZENIE POSZUKIWANIA

Tytuł
iPhone X

Opis
Witam, wypożyczę iPhone X, Stan jak na foto

Cena za dzień
35 zł

Kaucja
500 zł

Cena za dzień opóźnienia
80 zł

☒ Promuj ogłoszenie

Rysunek 44. Formularz dodawania ogłoszenia z opcją promowania go. Źródło: Opracowanie własne.

W ramach jednej subskrypcji konta premium użytkownik może jednocześnie promować pięć ogłoszeń. O ilości pozostałych promowań informuje komunikat pod checkboxem.

Kolejną dodatkową funkcjonalnością dla posiadacza konta premium jest możliwość ubezpieczenia użyczanego przedmiotu oraz wsparcie w procesie likwidacji ewentualnej szkody.

Możliwość ubezpieczenia użyczanego przedmiotu jest podczas kroku wysyłania przesyłki do wypożyczającego. Zaznaczając dodatkowy checkbox użytkownik może wybrać w jakim towarzystwie ubezpieczeń ma być wystawiona polisa i zatwierdzić wysyłkę. Rysunek 45 przedstawia krok wysyłki przedmiotu wraz z ubezpieczeniem go.

Nadaj przesyłkę do wypożyczającego

Potwierdź nadanie wypożyczenia wpisując numer przesyłki w pole poniżej.

Numer przesyłki

☒ Chcę wykupić dodatkowe ubezpieczenie. Koszt: 1440 zł.

Firma ubezpieczeniowa

☐ Link4 Towarzystwo Ubezpieczeń S.A.

☐ PZU Ubezpieczenia

☒ Towarzystwo Ubezpieczeń i Reasekuracji Warta S.A.

☐ Proama

☐ InterRisk Ubezpieczenia

ANULUJ

POTWIERDŹ

Rysunek 45. Ekran zatwierdzenia wysłania przesyłki z dodatkowym ubezpieczeniem. Źródło: Opracowanie własne.
Rysunek numer 46 prezentuje formularz zgłoszenia szkody.

Zgłoś szkodę

Zgłoś szkodę do wypożyczenia. Poprzez wypełnienie formularza potwierdzasz że szkoda nastąpiła podczas tego wypożyczenia.

Opis powstałej szkody

Okoliczności wystąpienia szkody

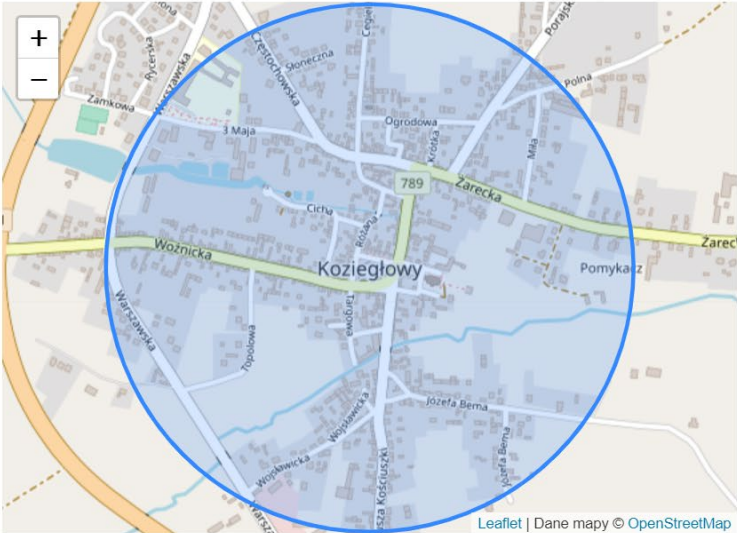
Data wystąpienia szkody

Miejsce wystąpienia szkody

Wyszukaj

WYSZUKAJ

Koziegłowy, śląskie



ANULUJ POTWIERDŹ

Rysunek 46. Formularz zgłoszenia szkody. Źródło: Opracowanie własne.

6.4. Likwidator

Likwidator to kolejny aktor jaki został zidentyfikowany na etapie analizy. Jego rolą jest obsługiwanie roszczeń klientów, którym wydarzyła się szkoda w trakcie użyczenia przedmiotu. Sam proces logowania odbywa się tak samo jak w przypadku zwykłych użytkowników. Wygląd i dostęp do panelu likwidatora uzyska użytkownik, któremu zostało nadane specjalne uprawnienie. Po udanym uwierzytelnieniu ukazuje się dedykowany interfejs co przedstawia rysunek numer 47.

111

Panel likwidatora

	Status zgłoszenia	Nazwa ogłoszenia	Uzyczący	Data zgłoszenia ↓	Data szkody	Data przekazania rzeczoznawcy	Rzecznawca	Suma ubezpieczenia	
▼	Oczekiwanie na oględziny	Telewizor Samsung 65 cali	Stromil Bogdan	16.01.2022 14:39	04.03.2019	27.03.2021	Pajak Robert	7000 zł	>

Rysunek 47. Widok panelu likwidatora. Źródło: Opracowanie własne.

Widok panelu likwidatora składa się z tabelki zawierającej podstawowe informacje dotyczące szkód przypisanych do danego likwidatora. Każdy z wierszy można rozwinąć, korzystając ze strzałek znajdujących się na końcu każdego rekordu. Wynikiem ich działania będzie wyświetlenie dodatkowych informacji o danej szkodzie, takich jak:

- Opis szkody
- Opis okoliczności powstania szkody
- Mapa z lokalizacją miejsca zdarzenia
- Informacje o użytkowniku zgłaszającym szkodę z możliwością przejścia do jego profilu

Rysunek numer 48 przedstawia widok rozwiniętego rekordu szkody w panelu likwidatora.

Panel likwidatora

	Status zgłoszenia	Nazwa ogłoszenia	Uzyczący	Data zgłoszenia ↓	Data szkody	Data przekazania rzeczoznawcy	Rzecznawca	Suma ubezpieczenia	
▲	Oczekiwanie na oględziny	Telewizor Samsung 65 cali	Stromil Bogdan	16.01.2022 14:39	04.03.2019	27.03.2021	Pajak Robert	7000 zł	>

Opis szkody

Obraz z telewizora jest tylko czarno-biały

Opis okoliczności

Wczoraj jeszcze działał normalnie

Użytkownik

Bogdan Stromil

PRZEJDŹ DO PROFILU →

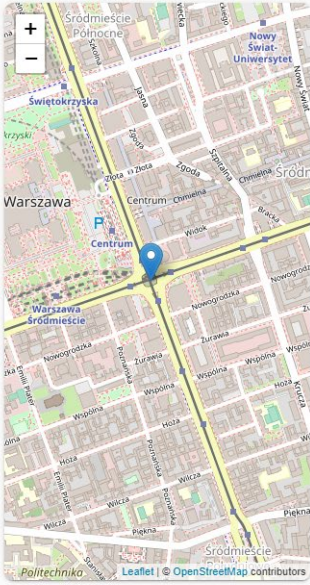


Rysunek 48. Przedstawia rozwinięty rekord zgłoszonej szkody. Źródło: Opracowanie własne.

Za przejście do szczegółowego widoku szkody, widocznego na rysunku numer 49, odpowiada niebieska strzałka znajdująca się na końcu każdego wiersza.

Zgłoszenie szkody dotyczącej ogłoszenia Telewizor Samsung 65 cali

Status zgłoszenia Oczekiwanie na oględzinę Data zgłoszenia szkody 16.01.2022 14:39 Suma ubezpieczenia 7000 zł Data zlecenia oględzin 27.03.2021 11:00	Data wystąpienia szkody 04.03.2019 Rzecznik odpowiedzialny za oględzinę Robert Pajak
Opis szkody Obraz z telewizora jest tylko czarno-biały Opis okoliczności Wczoraj jeszcze działał normalnie	
Użyczający  Bogdan Stromil Średnia ocena jako użyczający ★★★★☆ 3.0/5 Średnia ocena jako wypożyczający Brak ocen PRZEJDŹ DO PROFILU	



Rysunek 49. Formularz prezentujący szczegóły konkretnej szkody. Źródło: Opracowanie własne.

Widok dostarcza likwidatorowi wszystkich, niezbędnych informacji o szkodzie. Dodatkowo użytkownik ma podgląd na dane użyczającego jak również wypożyczającego z możliwością przejścia do ich profili. Ponadto interfejs umożliwia:

- Zgłoszenie szkody do TU (Towarzystwo Ubezpieczeniowe)
- Zlecenie oględzin wybierając rzeczoznawcę z rozwijanej drop listy
- Zaakceptowanie szkody z podaniem uzasadnienia
- Odrzucenie szkody z podaniem uzasadnienia

Każda z powyższych funkcjonalności jest dostępna poprzez dynamicznie zarządzane przyciski, których pojawienie się i aktywność zależne są od stanu biznesowego szkody.

Na rysunku numer 50 widok szkody, która ma dostępna opcje zlecenia oględzin.

The screenshot displays a web interface for a damage report. At the top left, there is a link 'SCHOWAJ INSPEKCJE'. The main section contains the following information:

- Inspekcja z dnia 16.01.2022**
- Oszacowanie szkody**
5000 zł
- Opis szkody**
Telewizor nie działa, zepsuł się. Nie wyświetla obrazu.

On the right side, there is a map showing the location of the damage. Below the map are three buttons: 'ODRZUĆ SZKODĘ' (red), 'PRZEKAŻ SZKODĘ DO TU' (grey), and 'ZAAKCEPTUJ SZKODĘ' (grey).

Below the inspection details, there are two user profiles:

- Użyczający**
Bogdan Stromil
Średnia ocena jako użyczający: 3.0/5 (4 stars)
Średnia ocena jako wypożyczający: Brak ocen
Button: PRZEJDŹ DO PROFILU
- Wypożyczający**
Krzysztof Karaś
Średnia ocena jako użyczający: Brak ocen
Średnia ocena jako wypożyczający: 1.0/5 (1 star)
Button: PRZEJDŹ DO PROFILU

Rysunek 50. Widok szczegółów szkody, gdzie jest dostępne zlecenie oględzin. Źródło: Opracowanie własne.

Po użyciu przycisku „Zleć oględziny” pojawi się okno pop-up – rysunek numer 51 - które umożliwi wybranie rzeczoznawcy, który otrzyma zlecenie.

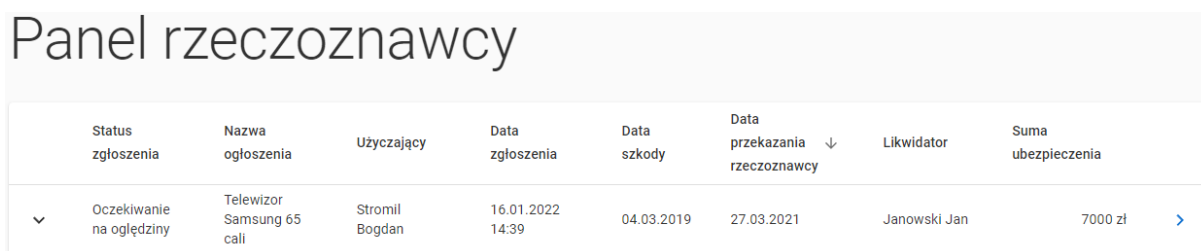
The screenshot shows a pop-up window titled 'Przypisz rzeczoznawcę'. Inside the window, there is a section labeled 'Wybierz' with a list of four experts, each with a profile picture and name:

- Robert Pajak
- Geralt Z Rivii
- Julian Barakowski
- Ricki Zielonka

Rysunek 51. Wybór rzeczoznawcy z listy dostępnych. Źródło: Opracowanie własne.

6.5. Rzeczoznawca

Rzeczoznawca to kolejny aktor zidentyfikowany podczas analizy, który uczestniczy w procesie likwidacji szkody. Użytkownik posiadający takie uprawnienie, tak samo jak likwidator, loguje się do aplikacji i posiada dedykowany panel do pracy. Widok panelu rzeczoznawcy prezentuje rysunek numer 52.



The image shows a web interface titled "Panel rzeczoznawcy". It contains a table with the following data:

Status zgłoszenia	Nazwa ogłoszenia	Uzyczający	Data zgłoszenia	Data szkody	Data przekazania rzeczoznawcy	Likwidator	Suma ubezpieczenia
▼ Oczekiwanie na oględziny	Telewizor Samsung 65 cali	Stromil Bogdan	16.01.2022 14:39	04.03.2019	27.03.2021	Janowski Jan	7000 zł

Rysunek 52. Widok panelu rzeczoznawcy po zalogowaniu. Źródło: Opracowanie własne.

W panelu rzeczoznawcy znajdziemy najważniejsze informacje na temat szkody skierowanej do oględzin takie jak:

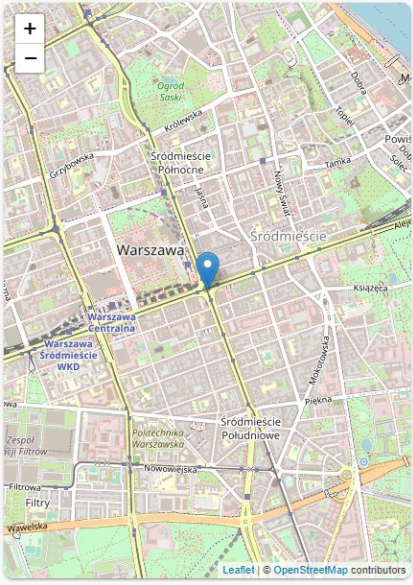
- status zgłoszenia, który mówi nam jaki obecnie status posiada szkoda,
- właściciela zgłoszonego przedmiotu,
- dokładny opis szkody wraz z jej okolicznościami,
- mapę z zaznaczoną lokalizacją miejsca, w którym nastąpiła szkoda.

Klikając w niebieską strzałkę w prawym rogu zgłoszenia otworzy nam się strona, na której rzeczoznawca może przeczytać komu przedmiot został wypożyczony oraz wycenić i opisać szkodę klikając w przycisk "Dokonaj oględzin".

Formularz ten jest widoczny na rysunku numer 53.

Zgłoszenie szkody dotyczącej ogłoszenia Telewizor Samsung 65 cali

Status zgłoszenia Oczekiwanie na oględzinę Data zgłoszenia szkody 16.01.2022 14:39 Suma ubezpieczenia 7000 zł Data zlecenia oględzin 27.03.2021 11:00	Data wystąpienia szkody 16.01.2022 Rzecznawca odpowiedzialny za oględzinę Robert Pajak
Opis szkody Obraz z telewizora jest tylko czarno-biały Opis okoliczności Wczoraj jeszcze działał normalnie	
Użyczający  Bogdan Stromil Średnia ocena jako użyczający ★★★★☆ 3.0/5 Średnia ocena jako wypożyczający Brak ocen PRZEJDŹ DO PROFILU	



DOKONAJ OGŁĘDZIN

Rysunek 53. Formularz z szczegółowymi danymi szkody do oględzin. Źródło: Opracowanie własne.

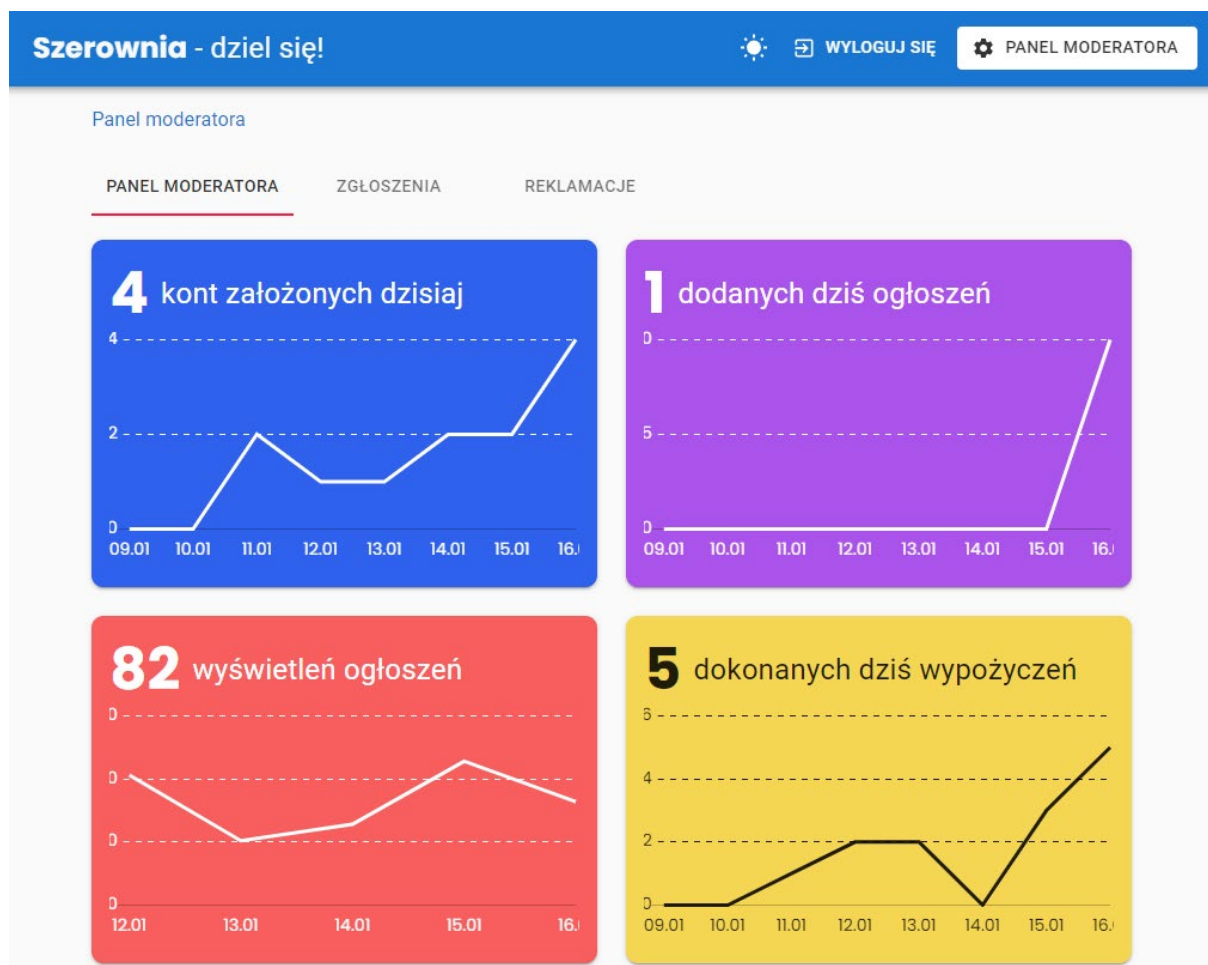
Panel rzeczoznawcy jest przejrzysty i bardzo prosty w obsłudze, dzięki czemu praca z nim staje się bardzo przyjemna.

6.6. Moderator

Po zalogowaniu się jako moderator użytkownik ma dostęp do narzędzi umożliwiających częściowe zarządzanie serwisem. Na stronie głównej wyświetla się panel, w którym widnieją statystyki:

- Liczba założonych kont,
- Liczba dodanych ogłoszeń,
- Liczba dokonanych wypożyczeń,
- Liczba wyświetleń wszystkich ogłoszeń.

Rysunek numer 54 prezentuje widok panelu moderatora po zalogowaniu do systemu.



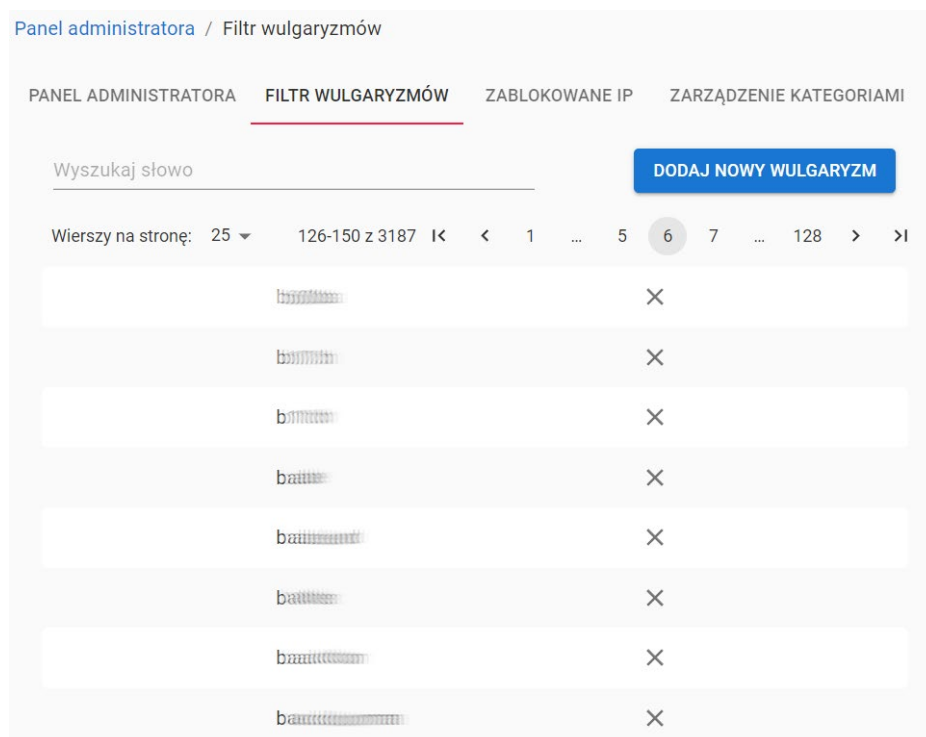
Rysunek 54. Widok panelu moderatora po zalogowaniu. Źródło: Opracowanie własne.

W drugiej zakładce – zgłoszenia - moderator ma dostęp do zgłoszeń, które napływają od użytkowników serwisu. Dotyczyć mogą wystawionych opinii, opublikowanych ogłoszeń jak i samych użytkowników. Moderator rozpatruje zgłoszenie i może usunąć ogłoszenie lub opinie oraz ma możliwość zablokowanie użytkownika.

6.7. Administrator

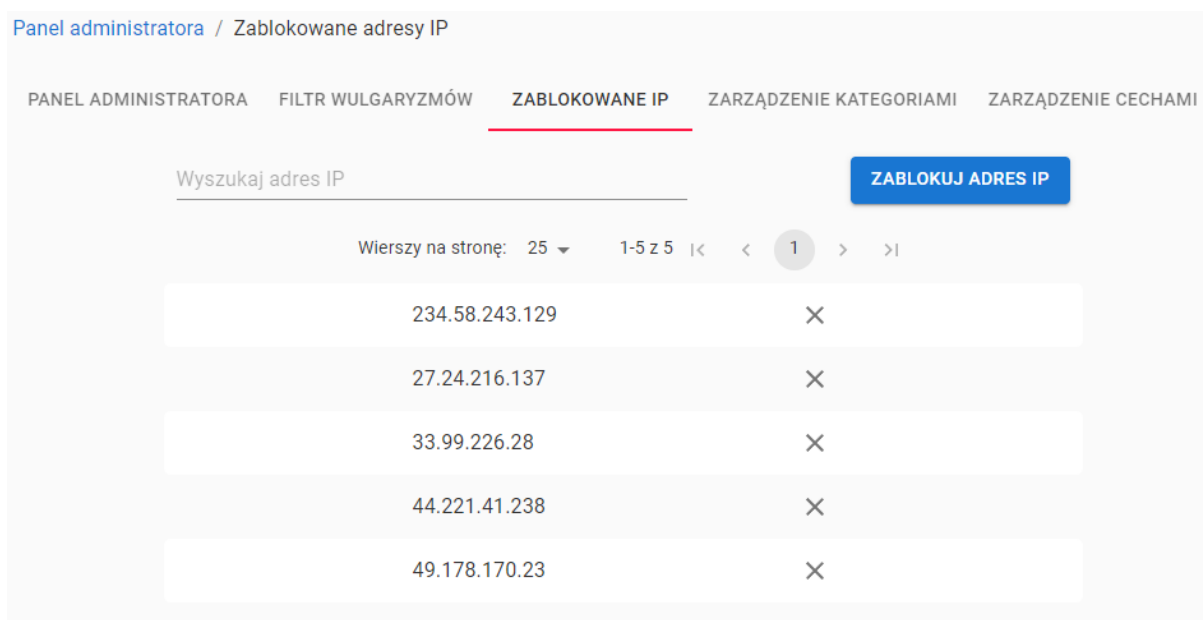
Po zalogowaniu się jako administrator zostaje wyświetlona strona ze statystykami portalu, która jest bardzo podobna do widoku panelu moderatora. Administrator ma możliwość dodawania nowych, przeglądania oraz usuwania słów wulgarnych lub powszechnie uważanych za wulgarne. W następstwie słowa dodane do listy będą blokowane, czy to w wiadomościach między użytkownikami, czy przy dodawaniu ogłoszenia.

Rysunek numer 55 przedstawia widok zakładki do zarządzania wulgarnymi słowami w panelu administratora.



Rysunek 55. Ekran zarządzania bazą słów wulgarnych. Źródło: Opracowanie własne.

Aplikacja pozwala również na blokowanie dostępu do serwisu przez dodanie adresu IP do listy adresów nieobsługiwanych. Rysunek numer 56 pokazuje widok zakładki do tej funkcjonalności.



Rysunek 56. Ekran do blokowania adresów IP. Źródło: Opracowanie własne.

6.8. Aplikacja mobilna

Interfejs aplikacji Szerownia.pl dla urządzeń mobilnych został opracowany na system operacyjny Android. Udostępniono go z myślą o użytkownikach telefonów komórkowych, którzy za pomocą tych urządzeń chcieliby korzystać z takich funkcjonalności aplikacji jak ogłaszanie przedmiotów do użyczenia, przeglądanie ogłoszeń, wypożyczanie przedmiotów, kontakt z innymi użytkownikami platformy.

Aby w pełni korzystać z aplikacji, należy się zalogować. Użytkownik niezalogowany może jedynie przeglądać wystawione do użyczenia lub poszukiwane przedmioty. Nowy użytkownik musi założyć w aplikacji konto. Po zalogowaniu i ponownym uruchomieniu aplikacji, użytkownik pozostaje zalogowany.

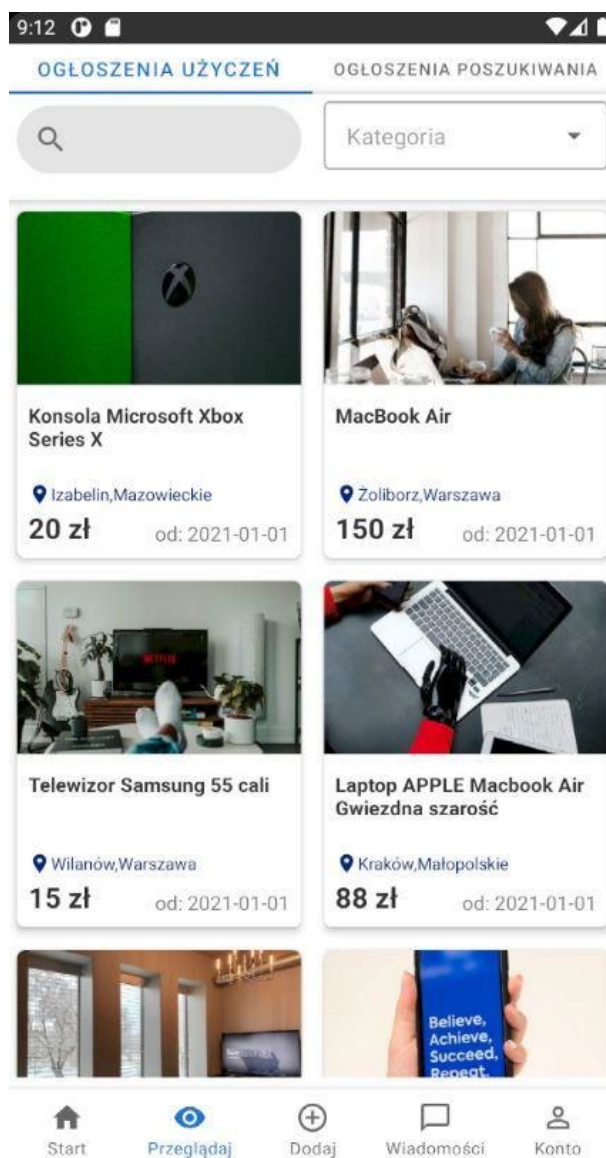
Strona główna zawiera najistotniejsze informacje na temat aplikacji, dla użytkownika zalogowanego, wyświetlane są również promowane ogłoszenia. Prezentuje to rysunek numer 57.



Rysunek 57. Strona główna serwisu dla użytkownika niezalogowanego. Źródło: Opracowanie własne.

Jak wcześniej wspomniano, aplikacja korzysta z udostępnianych przez system operacyjny Android gotowych komponentów, takich jak przyciski, ikony, pola formularza, suwaki.

Rysunek numer 58 prezentuje listę ogłoszeń w widoku aplikacji mobilnej.



Rysunek 58. Lista ogłoszeń z przedmiotami do wypożyczenia. Źródło: Opracowanie własne.

Po kliknięciu w miniaturę, użytkownik przenoszony jest do widoku szczegółów przedmiotu, co prezentuje rysunek numer 59.



Rysunek 59. Widok szczegółów przedmiotu do wypożyczenia. Źródło: Opracowanie własne.

W szczegółach przedmiotu znajdują się wszystkie istotne informacje: galeria zdjęć, opis przedmiotu, parametry, warunki wypożyczenia. Istnieje również możliwość skontaktowania się z użyczającym przedmiot za pomocą wiadomości.

Ogłoszenia, które naruszają regulamin – stanowią oszustwo, próbę wyłudzenia lub zawierają nielegalny przedmiot – mogą zostać zgłoszone do moderacji.

Na rysunku 60 widać napis uruchamiający proces zgłoszenia.



Rysunek 60. Finalizowanie wypożyczenia. Źródło: Opracowanie własne.

Użytkownik posiadający konto ma również dostęp do numeru telefonu osoby, która opublikowała ogłoszenie, a aplikacja ułatwia nawiązanie połączenia za pomocą przycisku „Zadzwoń”. Interesujący nas przedmiot możemy wypożyczyć, wybierając odpowiednie daty wypożyczenia oraz sposób dostawy. Serwis Szperownik.pl posiada wbudowany komponent czatu, za pomocą którego możliwe jest przysyłanie wiadomości między użytkownikami, na przykład w celu ustalenia szczegółów odbioru lub otrzymania informacji o parametrach przedmiotu, które nie zostały ujęte w ogłoszeniu.

Użytkownik zalogowany dodaje nowe ogłoszenia przedmiotu poszukiwanego lub przedmiotu, który posiada do użyczenia za pomocą formularza. Może również edytować swoje dane, a także przeglądać swoje ogłoszenia, obecne użyczenia i wypożyczenia, jak również historię. Dodatkowo,

istnieje możliwość wykupienia wersji premium, dzięki której użytkownik będzie mógł dodać swoje ogłoszenie jako promowane na stronie głównej.

7. Podsumowanie prac nad aplikacją

Śmiało można stwierdzić, że te trzy semestry specjalizacji dały zespołowi kompleksowy obraz tego, jak wygląda praca w zespole projektowym. Proces wytwarzania aplikacji w ramach pracy inżynierskiej pozwolił na dokładne poznanie całego cyklu wytwarzania oprogramowania. Było to bardzo ciekawe doświadczenie, które nauczyło studentów specjalizacji prawdziwej pracy, z jaką będą mieli styczność w życiu zawodowym. Jednocześnie w toku tworzenia kolejnych funkcjonalności zespół odczuwał co raz większą presję i stawiał czoło różnym wyzwaniom jakie się pojawiały. W kolejnych podrozdziałach zostały opisane zebrane spostrzeżenia na temat prowadzonych prac i możliwości rozwojowych aplikacji.

7.1 Wprowadzenie

Szerownia.pl została stworzona przez dwunastoosobową grupę studentów, którzy zdecydowali się na specjalizację Inżynieria oprogramowania i baz danych (A_IO). Temat pracy inżynierskiej oraz przedmiot prac programistycznych został uzgodniony przez studentów i zatwierdzony przez promotora. Inspiracją dla wyboru tematu był dynamiczny rozwój branży *e-commerce* oraz popularność trendów *less waste*, zgodnie z którymi lepszym rozwiązaniem jest częste używanie już zakupionych przedmiotów, niż ciągłe kupowanie nowych. Rzecz, która jest potrzebna wielu osobom, nie będzie „marnowała się” w piwnicy lub garażu.

Część zespołu mogła pochwalić się już pierwszymi doświadczeniami zawodowymi w obszarze wytwarzania oprogramowania, jednak pozostali posiadali jedynie umiejętności zdobyte podczas studiów. Z tego powodu pierwsze zajęcia zostały poświęcone do przekazania wiedzy niezbędnej do realizacji prac oraz ćwiczenia praktyczne. Dzięki wewnętrznym warsztatom członkowie zespołu mogli lepiej poznać swoje mocne i słabe strony w danych dziedzinach inżynierii oprogramowania. Tym samym podjęcie decyzji, do których podzespołów chcą należeć było bardziej świadome.

7.2. Organizacja pracy

Prace nad aplikacją rozpoczęły się wraz z semestrem zimowym 2020/2021 i trwały przez kolejne trzy semestry. W tym czasie powstała w pełni funkcjonalna aplikacja webowa i mobilna oraz praca inżynierska opisująca funkcjonalności i rozwiązania techniczne serwisu Szerownia.pl. Grupa studentów podzielona została na podzespoły, których kompetencje odzwierciedlają rzeczywisty sposób funkcjonowania firm budujących oprogramowanie. Studenci mogli brać udział w pracy więcej niż jednego zespołu, a sposób przyporządkowania zależał od preferencji i zdolności poszczególnych członków grupy. Każdy z podzespołów miał swojego lidera, natomiast nad całością prac czuwał kierownik projektu.

Zespół analizy odpowiedzialny był za przygotowanie założeń funkcjonalnych aplikacji, diagramu klas oraz diagramu przypadków użycia i aktywności. Następnie, w toku prac deweloperskich, analitycy mieli za zadanie monitorować zgodność aplikacji z pierwotnymi założeniami i w razie konieczności, służyć pomocą programistom we właściwej interpretacji diagramów.

Zespół projektowania przygotowywał makiety poszczególnych paneli użytkowników. W kolejnym etapie prac makiety były poddawane weryfikacji względem dobrych praktyk UX. Gdy do implementacji aplikacji wchodziły kolejne ekrany użytkownika zespół weryfikował ich zgodność z wcześniej przygotowanymi makietami.

Zespół programistów przygotowywał kod źródłowy aplikacji, a studenci byli podzieleni na programistów:

- backendu,
- frontendu wersji przeglądarkowej,
- aplikacji mobilnej.

Zespół testów sprawdzał postępy w tworzeniu aplikacji względem zaplanowanych przypadków użycia, monitorował występujące błędy oraz niezgodności z przyjętymi wcześniej założeniami.

Przez cały czas trwania prac nad aplikacją, równolegle zespół dokumentacji opracowywał treść pracy inżynierskiej.

Podczas cotygodniowych spotkań, które z uwagi na sytuację epidemiczną odbywały się w trybie zdalnym za pośrednictwem aplikacji Microsoft Teams, zespół programistów przedstawiał postępy w realizacji prac deweloperskich, testerzy opowiadali o znalezionych błędach i niezgodnościach z wymaganiami, analitycy odpowiadali na wątpliwości programistów. Następnie ustalane były cele na kolejną iterację prac, a do poszczególnych zadań przypisywano wykonawców.

7.3. Uwagi o technologiach

Podczas pracy nad projektem zespół napotkał na wiele trudności i wyzwań. W czasie fazy analizy występowały problemy z wersjami diagramów. Z uwagi na brak doświadczenia u części zespołu i ilości aktorów i klas wymagały one wielu poprawek lub ponownego modelowania.

Kolejnym wyzwaniem było utracenie danych z systemu Confluence, który był wykorzystywany do udostępniania tekstu pracy oraz scenariuszy testowych. Niestety nie udało się odtworzyć wszystkich utraconych danych i część pracy trzeba było wykonać ponownie.

Jedną z mniej oczekiwanych trudności, które wstąpiły było zrozumienie samej branży e-commerce oraz ubezpieczeniowej. Tylko niewielka część zespołu posiadająca doświadczenie zawodowe. Dopiero po wielu rozmowach i dyskusjach w trakcie zajęć udało się przyswoić terminy i mechanizmy specyficzne dla obu gałęzi biznesowych.

Podczas testowania aplikacji jedynym problemem, który czasami występował było testowanie niedokończonych wersji aplikacji. Było to spowodowane tym, że zespół implementacji udostępniał nie gotowe wersje przez co były zgłaszane błędy, które zostały już naprawione.

W fazie implementacji wystąpiły problemy z serwerem udostępnionym przez uczelnię. Brak doświadczenia w jego obsłudze powodowały brak możliwości korzystania z niego przez pewien czas. Skutkowało to koniecznością korzystania z aplikacji zainstalowanej i uruchomionej lokalnie na urządzeniu testera.

7.4. Rozwój aplikacji

Niniejszy podrozdział przedstawia hipotetyczne możliwości rozwoju dla aplikacji Szerownia.pl, w przypadku sukcesu.

W przypadku sukcesu aplikacji w wersji mobilnej na systemie Android oraz zwiększenia na polskim rynku udziału telefonów firmy Apple, naturalnym krokiem będzie opracowanie aplikacji na system iOS. Zmiany rozwojowe mogą zostać wymuszone technologią – dotyczy to na przykład rozwiązań w urządzeniach mobilnych, nowych właściwości przeglądarek internetowych, standardów budowania i wyświetlania aplikacji internetowych. Obecnie trudno jest sprecyzować kierunek rozwoju w tym zakresie, ale takie zmiany w otoczeniu muszą być monitorowane.

W najnowszej historii trudno znaleźć przykład, aby aplikacja użytkowa skierowana do ogółu społeczeństwa została wyprodukowana w Polsce, a następnie odniosła światowy sukces. Niemniej, w przypadku sprzyjających okoliczności, na przykład dużego zapotrzebowania rynkowego i powstałej w ten sposób niszy, należy przygotować wersję anglojęzyczną. Może to być również inna wersja językowa, jeśli aplikacja miałaby szansę zaistnieć w którymś z krajów sąsiadujących.

Aplikacja w swojej obecnej wersji nie posiada rozbudowanego modułu logistycznego do obsługi wysyłki przedmiotów. Biorąc pod uwagę rosnącą popularność paczkomatów, należy rozważyć rozwój w zakresie wsparcia użytkowników w generowaniu etykiet, anonimizacja danych wrażliwych, numerów telefonu, jeśli przesyłka nadawana i odbierana jest przez paczkomat. Należy rozważyć możliwość integracji systemowych z API firmy Furgonetka.pl.

Poza rozbudowywaniem aplikacji w skali makro, czyli dodawanie kolejnych modułów, modyfikację technologii, opracowywaniem wersji językowych, warto zadbać o drobne usprawnienia – niewielkie w skali całej aplikacji i niewpływające na jej główne funkcjonalności. Przykładem może być, obok wstawiania fotografii, możliwość dodawania krótkich materiałów wideo, zależna od postępu technologicznego – wydajności serwerów i dostępności przestrzeni dyskowej.

Innym ciekawym pomysłem jest dodanie możliwości odkupienia przedmiotu, choć byłaby to funkcjonalność poboczna. Aplikacja powinna nadal być zoptymalizowana pod kątem wypożyczeń, ale

jeśli użytkownik, poza chęcią użyczenia, byłby też skłonny sprzedać przedmiot, Szerownia.pl mogłaby mu to ułatwić.

Pozostałe kierunki rozwoju i usprawnienia aplikacji byłyby dodawane na podstawie zapotrzebowania użytkowników oraz obserwacji rozwiązań stosowanych przez konkurencję.

7.5. Podsumowanie

Prace programistyczne dotyczące projektu Szerownia.pl zostały w dużej mierze zrealizowane. Najważniejsze funkcjonalności, niezbędne do kompleksowego obsługiwanie użytkowników zostały zaimplementowane, przetestowane i wdrożone. By zapewnić bezpieczeństwo użytkowników i pracowników obsługujących serwis zadbano o:

- rejestrację i logowanie użytkowników,
- zarządzanie rolami i uprawnieniami użytkowników.

Niezbędnym było obsłużenie szyfrowania dla danych logowania.

Zakładane funkcjonalności w fazie analizy zostały w pełni zrealizowane:

- publikowanie ogłoszeń użyczenia i poszukiwania,
- kupienie konta premium i korzystania z benefitów wynikających z jego posiadania,
- wspieranie pracowników likwidacji szkód i rzeczoznawców w codziennej pracy,
- daje możliwość obsługi użytkowników i weryfikacji ich zgłoszeń przez moderatorów.

Bez wykonania tej czynności dostępna jest jedynie funkcjonalność przeglądania ogłoszeń. W zależności od wykupienia dodatkowego uprawnienia jakim jest status konta premium, użytkownik zyskuje możliwość promowania swoich ogłoszeń. Dodatkową funkcjonalnością przewidzianą w założeniach biznesowych jest dystrybucja ubezpieczeń na użyczane przedmioty, a także wsparcie merytoryczne w przypadku wystąpienia na nim szkody. Wykupienie konta premium nie jest wymagane do korzystania z głównych funkcjonalności aplikacji. Użytkownik zalogowany ma pełną możliwość zarządzania swoimi ogłoszeniami i przedmiotami jakie posiada na użyczenie.

W ramach aplikacji obsługującej ogłoszenia powstała druga grupa użytkowników, tak zwanych obsługujących ruch w systemie. W ich skład wchodzi likwidatorzy, rzeczoznawcy, moderatorzy i administratorzy. Użytkownik posiadający konto premium może ubezpieczyć przedmiot, który użycza. W razie wystąpienia na nim szkody likwidator służy wsparciem merytorycznym i pomocą przy procesowaniu likwidacji szkody. W tym procesie pomaga mu również rzeczoznawca, którego zadaniem jest wycenienie szkód jakie powstały na przedmiocie użyczenia. Natomiast moderator i administrator zajmują się bieżącym ruchem w serwisie: od obsługi zgłoszeń o naruszeniach od użytkowników przez weryfikację ogłoszeń i ich treści względem regulaminu serwisu po dbanie o bezpieczeństwo danych wszystkich użytkowników.

Utrzymanie obszernego serwisu, przez który może przewijać się setki użytkowników oraz tysiące ogłoszeń, wymaga dopasowanego do modelu biznesowego narzędzia. Rozwiązania oparte o moduły i architektura mikroservisów gwarantuje szybkie reagowanie na zmieniające się potrzeby biznesu. Szerownia.pl z całą pewnością jest systemem ułatwiającym wzajemne pożyczanie przedmiotów i obsługę uczestników tych wypożyczeń.

Bibliografia

- [1] Microsoft 2022. [Online] Lipiec 2018. Available: <https://www.microsoft.com/pl-pl/microsoft-teams/log-in>. [Data uzyskania dostępu: 04.01.2022].
- [2] Atlassian, "Jira Software". [Online] Available: <https://www.atlassian.com/software/jira>. [Data uzyskania dostępu: 04.01.2022].
- [3] Atlassian, "Confluence". [Online]. Available: <https://www.atlassian.com/software/confluence>. [Data uzyskania dostępu: 04.01.2022].
- [4] Kodeks cywilny Dz.U.2020.0.1740 t.j. - Ustawa z dnia 23 kwietnia 1964 r. -. ArsLege. [Online] Available: <https://arslege.pl/kodeks-cywilny/k9/s2018/>. [Data uzyskania dostępu: 04.01.2022].
- [5] Kancelaria Sejmu. Dz.U. 2018 poz. 1000. USTAWA z dnia 10 maja 2018 r. o ochronie danych osobowych. [Online] Available: <https://isap.sejm.gov.pl/isap.nsf/download.xsp/WDU20180001000/T/D20181000L.pdf>. [Data uzyskania dostępu: 03.04.2021].
- [6] Horstmann Cay S., "Java. Podstawy. Wydanie XI.", Gliwice, Helion SA, 2019.
- [7] Walls Craig., "Spring w akcji Wydanie V.", Gliwice, Helion SA, 2019.
- [8] Christian Bauer Gavin King., "Hibernate w akcji.", Gliwice, Helion SA, 2019.
- [9] Smart John Ferguson., "Java. Praktyczne narzędzia.", Gliwice, Helion SA, 2019.
- [10] Dubois Paul., "MySQL. Podręcznik administratora.", Gliwice, Helion SA, 2019.
- [11] Flanagan David., "JavaScript Przewodnik. Poznaj język mistrz w programowania.", Gliwice, Helion SA, 2020.
- [12] Kamiński Paweł., "React. Wstęp do programowania.", Gliwice, Helion SA, 2021.

- [13] Boduch Adam., "React Material-UI Cookbook.", Gliwice, Helion SA, 2019.
- [14] Mehta Bhakti., "REST. Najlepsze praktyki i wzorce w języku Java.", Gliwice, Helion SA, 2014.
- [15] JetBrains. IntelliJ IDEA. [Online] Available: <https://www.jetbrains.com/idea/>.
[Data uzyskania dostępu: 18.10.2021].
- [16] Postman Inc. Postman. [Online] Available: <https://www.postman.com>.
[Data uzyskania dostępu: 18.10.2021].
- [17] SmartBear Software.Swagger. [Online] Available: <https://swagger.io/tools/swagger>.
[Data uzyskania dostępu: 18.10.2021].
- [18] Developers Google. Android Studio. [Online] Available: <https://developer.android.com/studio>.
[Data uzyskania dostępu: 18.10.2021].
- [19] Software Freedom Conservancy. Selenium. [Online] Available: <https://www.selenium.dev>.
[Data uzyskania dostępu: 18.10.2021].
- [20] VMware Inc. Spring. [Online] Available: <https://spring.io/projects/spring-framework>.
[Data uzyskania dostępu: 08.11.2021].
- [21] VMware, Inc. Spring | Why Spring? [Online] Available: <https://spring.io/why-spring>.
[Data uzyskania dostępu: 08.11.2021].
- [22] Walke Jordan. Wikipedia. Wolna encyklopedia. [Online] Available:
<https://pl.wikipedia.org/wiki/React.js>. [Data uzyskania dostępu: 08.11.2021].
- [23] Wikipedia. Wolna encyklopedia. " Model-View-Controller" [Online]
Available: <https://pl.wikipedia.org/wiki/Model-View-Controller>.
[Data uzyskania dostępu: 08.11.2021].
- [24] Piechowiak Jerzy. Blog Helion. [Online]
Available: <https://blog.helion.pl/mvc-wprowadzenie-wzorca/>. [Data uzyskania dostępu:
08.11.2021].

- [25] Wikipedia. Wolna encyklopedia. "Wzorzec projektowy" [Online]
Available: [https://pl.wikipedia.org/wiki/Wzorzec_projektowy_\(informatyka\)](https://pl.wikipedia.org/wiki/Wzorzec_projektowy_(informatyka)). [Data uzyskania dostępu: 08.11.2021].
- [26] Projektowanie system w komputerowych. [Online] Available:
<http://zasoby.open.agh.edu.pl/~09sfraczek/wzorze-projektowe-wstep%2C1%2C57.html>.
[Data uzyskania dostępu: 08.11.2021].
- [27] Wikipedia. Wolna encyklopedia. [Online] Available:
[https://pl.wikipedia.org/wiki/Singleton_\(wzorzec_projektowy\)](https://pl.wikipedia.org/wiki/Singleton_(wzorzec_projektowy)). [Data uzyskania dostępu: 08.11.2021].
- [28] Refactoring.Guru. Singleton. [Online] Available: <https://refactoring.guru/pl/designpatterns/singleton>. [Data uzyskania dostępu: 08.11.2021].
- [29] Refactoring Guru. Fasada. [Online] Available: <https://refactoring.guru/pl/designpatterns/facade>. [Data uzyskania dostępu: 08.11.2021].
- [30] Nowak Dawid. Wzorzec projektowy Fasada. [Online] Available:
<http://devfoundry.pl/wzorzec-projektowy-fasada/>. [Data uzyskania dostępu: 08.11.2021].
- [31] Baeldung. Baeldung. [Online] Available: <https://www.baeldung.com/java-dao-pattern>. [Data uzyskania dostępu: 08.11.2021].
- [32] Wikipedia. Wolna encyklopedia. [Online] Available:
https://en.wikipedia.org/wiki/Data_access_object. [Data uzyskania dostępu: 15.11.2021].
- [33] Wikipedia. Wolna encyklopedia. [Online] Available:
https://en.wikipedia.org/wiki/Data_transfer_object. [Data uzyskania dostępu: 15.11.2021].
- [34] Stowarzyszenie jakości system w informatycznych. Certyfikowany tester Sylabus poziomu podstawowego ISTQB. [Online] Available: <https://sjqi.org/ist-qb/do-pobrania/> [Data uzyskania dostępu: 12.12.2021].

- [35] OpenStreetMap. Wiki OpenStreetMap. [Online] Available:
https://wiki.openstreetmap.org/wiki/Pl:Main_Page. [Data uzyskania dostępu: 18.12.2021].

Spis ilustracji

Rysunek 1. Zadania zaplanowane za pomocą narzędzie Jira. Źródło: Opracowanie własne.	7
Rysunek 2. Strona główna serwisu www.lokpo.pl . Źródło: Opracowanie własne.	16
Rysunek 3. Formularz dodawania nowego ogłoszenia. Źródło: Opracowanie własne.	17

Rysunek 4. Strona główna serwisu OLX. Źródło: Opracowanie własne.	19
Rysunek 5. Strona główna serwisu plenti. Źródło: Opracowanie własne.	22
Rysunek 6. Wszystkie kategorie serwisu plenti. Źródło: Opracowanie własne.	23
Rysunek 7. Diagram przypadków użycia dla systemu Szerownika.pl. Źródło: Opracowanie własne.	45
Rysunek 8. Diagram przypadków użycia dla użytkownika niezarejestrowanego. Źródło: Opracowanie własne.	46
Rysunek 9. Diagram przypadków użycia dla użytkownika zarejestrowanego. Źródło: Opracowanie własne.	47
Rysunek 10. Diagram przypadków użycia dla użytkownika premium. Źródło: Opracowanie własne.	48
Rysunek 11. Diagram przypadków użycia dla likwidatora. Źródło: Opracowanie własne.	49
Rysunek 12. Diagram przypadków użycia dla rzeczoznawcy. Źródło: Opracowanie własne.	50
Rysunek 13. Diagram przypadków użycia dla moderatora. Źródło: Opracowanie własne.	50
Rysunek 14. Diagram przypadków użycia dla admin. Źródło: Opracowanie własne.	51
Rysunek 15. Diagram przypadków użycia dla podsystemu czasu. Źródło: Opracowanie własne.	52
Rysunek 16. Diagram klas. Źródło: Opracowanie własne.	53
Rysunek 17. Diagram aktywności, przekaz zgłoszenie szkody do TU. Źródło: Opracowanie własne.	54
Rysunek 18. Diagram aktywności, zleć oględziny do rzeczoznawcy. Źródło: Opracowanie własne.	55
Rysunek 19. Diagram aktywności, sprawdź decyzje TU odnośnie do odszkodowania. Źródło: Opracowanie własne.	56
Rysunek 20. Diagram aktywności, przyjmij / odrzuć zlecenie na oględziny. Źródło: Opracowanie własne.	57
Rysunek 21. Diagram aktywności, uzupełnij formularz z oględzin. Źródło: Opracowanie własne.	58
Rysunek 22. Okno instalacji Spring. Źródło: Opracowanie własne.	63
Rysunek 23. Przykładowa architektura mikroserwisów z wykorzystaniem Spring Cloud. Źródło: https://spring.io/microservices	64

Rysunek 24. Przykładowa makietka widoku do rejestracji użytkownika. Źródło: Opracowanie własne.	69
Rysunek 25. Schemat komunikacji w REST API. Źródło: https://phpenthusiast.com/blog/what-is-rest-api	69
Rysunek 26. IntelliJ IDEA wersja Ultimate. Źródło: Opracowanie własne.	71
Rysunek 27. Przykładowe okno aplikacji Postman. Źródło: Opracowanie własne.	72
Rysunek 28. Widok API dla Szerownia.pl w swagger. Źródło: Opracowanie własne.	74
Rysunek 29. Struktura JWT Token. Źródło: https://medium.com/@sureshdsdsk .	91
Rysunek 30. Widok backlogu błędów z JIRA. Źródło: Opracowanie własne.	96
Rysunek 31. Ekran aplikacji Postman prezentujący działanie funkcjonalności logowania. Źródło: Opracowanie własne.	97
Rysunek 32. Wywołanie metody getMyRentalDsa(). Źródło: Opracowanie własne.	98
Rysunek 33. Ekran panelu logowania do aplikacji. Źródło: Opracowanie własne.	103
Rysunek 34. Ekran zakładania nowego konta użytkownika. Źródło: Opracowanie własne.	103
Rysunek 35. Ekran główny serwisu Szerownia.pl dla użytkownika niezalogowanego. Źródło: Opracowanie własne.	104
Rysunek 36. Widok szczegółów ogłoszenia dla użytkownika niezalogowanego. Źródło: Opracowanie własne.	104
Rysunek 37. Ekran dodawania nowego ogłoszenia. Źródło: Opracowanie własne.	105
Rysunek 38. c.d. dodawania nowego ogłoszenia. Źródło: Opracowanie własne.	106
Rysunek 39. Ekran dodawania ogłoszenia poszukiwania. Źródło: Opracowanie własne.	107
Rysunek 40. c.d. wypożyczenia przedmiotu, wybór daty i sposobu dostawy. Źródło: Opracowanie własne.	108
Rysunek 41. Widok po kliknięciu w "Twoje konto". Źródło: Opracowanie własne.	109
Rysunek 42. Ekran kupowania dostępu do konta premium. Źródło: Opracowanie własne.	109
Rysunek 43. Widok listy ogłoszeń z promowanymi ogłoszeniami użyczenia. Źródło: Opracowanie własne.	110
Rysunek 44. Formularz dodawania ogłoszenia z opcją promowania go. Źródło: Opracowanie własne.	111
Rysunek 45. Ekran zatwierdzenia wysłania przesyłki z dodatkowym ubezpieczeniem. Źródło: Opracowanie własne.	111
Rysunek 46. Formularz zgłoszenia szkody. Źródło: Opracowanie własne.	112

Rysunek 47. Widok panelu likwidatora. Źródło: Opracowanie własne.	113
Rysunek 48. Przedstawia rozwinięty rekord zgłoszonej szkody. Źródło: Opracowanie własne.	113
Rysunek 49. Formularz prezentujący szczegóły konkretnej szkody. Źródło: Opracowanie własne.	114
Rysunek 50. Widok szczegółów szkody, gdzie jest dostępne zlecenie oględzin. Źródło: Opracowanie własne.	115
Rysunek 51. Wybór rzeczoznawcy z listy dostępnych. Źródło: Opracowanie własne.	115
Rysunek 52. Widok panelu rzeczoznawcy po zalogowaniu. Źródło: Opracowanie własne.	116
Rysunek 53. Formularz z szczegółowymi danymi szkody do oględzin. Źródło: Opracowanie własne.	117
Rysunek 54. Widok panelu moderatora po zalogowaniu. Źródło: Opracowanie własne.	118
Rysunek 55. Ekran zarządzania bazą słów wulgarnych. Źródło: Opracowanie własne.	119
Rysunek 56. Ekran do blokowania adresów IP. Źródło: Opracowanie własne.	119
Rysunek 57. Strona główna serwisu dla użytkownika niezalogowanego. Źródło: Opracowanie własne.	120
Rysunek 58. Lista ogłoszeń z przedmiotami do wypożyczenia. Źródło: Opracowanie własne.	121
Rysunek 59. Widok szczegółów przedmiotu do wypożyczenia. Źródło: Opracowanie własne.	122
Rysunek 60. Finalizowanie wypożyczenia. Źródło: Opracowanie własne.	123

Spis tabel

Tabela 1. Podział na podzespoły i przydział zadań w projekcie Szerownia.pl. Źródło: Opracowanie własne.	8
Tabela 2. Obszary wspierane przez serwis LokPo. Źródło: Opracowanie własne.	17
Tabela 3. Obszary wspierane przez serwis OLX. Źródło: Opracowanie własne.	20

Tabela 4. Obszary wspierane przez serwis Plenti. Źródło: Opracowanie własne.	24
Tabela 5. Wymagania funkcjonalne dla użytkownika niezarejestrowanego. Źródło: Opracowanie własne.	28
Tabela 6. Wymagania funkcjonalne dla użytkownika zarejestrowanego. Źródło: Opracowanie własne.	31
Tabela 7. Wymagania funkcjonalne dla użytkownika premium. Źródło: Opracowanie własne	34
Tabela 8. Wymagania funkcjonalne dla użytkownika likwidatora szkód i rzeczoznawcy.	36
Tabela 9. Wymagania funkcjonalne dla moderatora i administratora systemu.	38
Tabela 10. Wymagania funkcjonalne dla podsystemu czasu. Źródło: Opracowanie własne.	40
Tabela 11. Przypadek użycia zakupu konta premium wraz ze scenariuszem. Źródło: Opracowanie własne.	58

Spis listingów

Listing 1. Przykładowy kod testu jednostkowego. Źródło: Opracowanie własne.	77
Listing 2. Przykładowy komponent Advertisement. Źródło: Opracowanie własne.	81
Listing 3. Zastosowanie komponentu Ad w wynikach wyszukiwania. Źródło: Opracowanie własne.	82
Listing 4. Komponent SearchResults - definicja. Źródło: Opracowanie własne.	82
Listing 5. Zastosowanie SearchResults na stronie głównej aplikacji. Źródło: Opracowanie własne.	83

Listing 6. Definicja stałej request. Źródło: Opracowanie własne.	84
Listing 7. Pobieranie listy ogłoszeń wypożyczenia. Źródło: Opracowanie własne.	84
Listing 8. Przykład klasy kontrolera wypożyczeń HireController. Źródło: Opracowanie własne.	85
Listing 9. Algorytm obliczający koszt wypożyczenia. Źródło: Opracowanie własne.	87
Listing 10. Wdrożenie wzorca fasada na przykładzie klasy Authorization. Źródło: Opracowanie własne.	88
Listing 11. Przykład zastosowania wzorca DAO. Źródło: Opracowanie własne.	88
Listing 12. Przykład zastosowania wzorca DTO. Źródło: Opracowanie własne.	89
Listing 13. Przykład zastosowania wstrzykiwania zależności. Źródło: Opracowanie własne.	89
Listing 14. Przykładowe zastosowania adnotacji w projekcie. Źródło: Opracowanie własne.	90
Listing 15. Przykładowe zastosowanie JpaRepository. Źródło: Opracowanie własne.	90
Listing 16. Klasa ProtectedRoute oraz najważniejsze elementy z nią związane. Źródło: Opracowanie własne.	92
Listing 17. Fragment klasy SecurityConfig z metodą configure(). Źródło: Opracowanie własne.	93
Listing 18. Test automatyczny na weryfikację funkcjonalności rejestracji. Źródło: Opracowanie własne.	99
Listing 19. Metoda wybierająca sposób wysyłki przedmiotu. Źródło: Opracowanie własne.	99