



POLSKO-JAPOŃSKA AKADEMIA
TECHNIK KOMPUTEROWYCH

Wydział Informatyki

**Katedra Inżynierii Oprogramowania
Inżynieria Oprogramowania i Baz Danych**

Łukasz Szeremeta
Nr albumu s8428

Grywalizacja w systemach baz Wiki działających w chmurze

Praca magisterska napisana
pod kierunkiem:

dr inż. Mariusz Trzaska

Warszawa, wrzesień 2016

Streszczenie

Niniejsza praca przedstawia zastosowanie grywalizacji w systemach baz Wiki. Autor przeanalizował istniejące rozwiązania i zaproponował koncepcję rozwiązania, która pozwoli zwiększyć motywację do tworzenia i przeglądania artykułów w omawianych systemach.

Ostatecznie praca została poparta działającym prototypem przedstawiającym działanie grywalizacji w rozwiązaniach typu Wiki. Został on stworzony w oparciu o platformę Goole App Engine działającą w chmurze.

Spis treści

1	Wstęp.....	5
1.1	Cel pracy.....	5
1.2	Rozwiązania przyjęte w pracy.....	5
1.3	Rezultat pracy.....	5
1.4	Organizacja pracy.....	5
2	Analiza wybranych rozwiązań obecnych na rynku.....	6
2.1	Wstęp.....	6
2.2	Problemy obecnych rozwiązań.....	6
2.3	MediaWiki.....	7
2.4	WikiDot.....	8
2.5	Wikia.....	10
2.6	Wikispaces.....	12
2.7	Podsumowanie.....	14
3	Koncepcja nowego systemu typu Wiki.....	15
3.1	Wiki.....	15
3.2	Kategorie.....	15
3.3	Artykuły.....	15
3.4	Wersjonowanie.....	16
3.5	Wiadomości.....	16
3.6	Wyszukiwarka.....	16
3.7	Powiadomienia.....	16
3.8	System dodatków.....	16
3.8.1	Ulubione.....	17
3.8.2	Osiągnięcia.....	17
3.8.3	System punktowy.....	17
3.8.4	Rangi.....	18
3.8.5	Rankingi.....	18
3.8.6	Wyzwania.....	19
3.9	API.....	19
3.10	Aplikacja mobilna.....	20
3.11	Chmura.....	20
4	Użyte technologie oraz oprogramowanie.....	21
4.1	Część Webowa.....	21
4.1.1	Google App Engine.....	21
4.1.2	Python.....	21
4.1.3	HTML.....	22

4.1.4	JavaScript	22
4.1.5	JQuery.....	22
4.1.6	CSS 3.....	22
4.1.7	Bootstrap 3.....	22
4.1.8	Font Awesome.....	23
4.1.9	CKEditor.....	23
4.1.10	Google App Engine Boilerplate	23
4.1.11	PyCharm.....	23
4.2	Część Mobilna.....	23
4.2.1	Java.....	23
4.2.2	Android SDK.....	24
4.2.3	Picasso.....	24
4.2.4	Android Studio	24
4.3	Część wspólna	24
4.3.1	Git / Bitbucket	24
4.3.2	GIMP 2.....	25
4.3.3	Inkscape.....	25
4.3.4	FreeMind	25
5	Aplikacje demonstracyjne	26
5.1	Aplikacja Web.....	26
5.2	System dodatków	37
5.2.1	Osiągnięcia.....	40
5.2.2	Ulubione	41
5.3	Aplikacja na Androida.....	41
6	Podsumowanie.....	46
6.1	Wady i zalety.....	46
6.2	Możliwości rozwoju aplikacji	46
6.3	Zakończenie.....	47
7	Bibliografia.....	48
8	Spis rysunków	49
9	Kody źródłowe aplikacji.....	50

1 Wstęp

Istnieje bardzo dużo zarówno komercyjnych jak i bezpłatnych rozwiązań typu Wiki. Wszystkie systemy pozwalają na wprowadzanie danych w ustalonej formie i strukturze oraz łatwy dostęp do tych informacji. Coraz więcej jest aplikacji mobilnych oraz responsywnych szablonów dla stron www.

Oprogramowanie tego typu stosowane jest jako:

- encyklopedie,
- repozytoria firmowej wiedzy,
- projekty tematyczne.

Nowoczesne programy tego typu wykorzystują edytory WYSIWYG.

1.1 Cel pracy

Celem pracy jest stworzenie propozycji systemu, w którym tworzenie materiałów i zdobywanie wiedzy jest nagradzane, takie rozwiązanie wprowadzi dodatkową motywację użytkowników oraz uatrakcyjni korzystanie z aplikacji. Zgodnie z założeniami koncepcyjnymi oprogramowanie powinno działać na ogólnodostępnej chmurze.

1.2 Rozwiązania przyjęte w pracy

W niniejszej pracy do rozwiązywania problemów istniejących w obecnych systemach Wiki użyto teorii grywalizacji. Do implementacji zaproponowanej koncepcji wykorzystano technologię Python uruchamianą na platformie Google App Engine. Do stworzenia mobilnej części prototypu zastosowano język Java oraz Android SDK.

1.3 Rezultat pracy

W rezultacie powstała propozycja systemu, który umożliwi tworzenie w prosty sposób repozytoriów wiedzy oraz wprowadza elementy gry i rywalizacji. Koncepcja oprogramowania została przedstawiona w prototypie responsywnego systemu webowego oraz aplikacji mobilnej na Androida.

1.4 Organizacja pracy

Praca rozpoczyna się ogólnymi informacjami na temat systemów typu Wiki. Następnie opisane jest najpopularniejsze rozwiązanie znajdujące się na rynku.

W kolejnym rozdziale znajdują się dane na temat narzędzi oraz technologii użytych podczas projektowania oraz implementowania tego oprogramowania.

Później zaproponowana zostaje koncepcja nowej Wiki z elementami grywalizacji, pozwalająca na efektywne korzystanie i tworzenie materiałów.

Kolejny fragment charakteryzuje stworzony prototyp zarówno aplikacji mobilnej jak i strony www.

Na końcu przedstawiono całokształt pracy, podsumowano wszystkie wady i zalety stworzonego rozwiązania oraz ustalono możliwy scenariusz rozwoju aplikacji.

2 Analiza wybranych rozwiązań obecnych na rynku

W niniejszym rozdziale będą omówione istniejące rozwiązania oraz przedstawione zostaną definicje terminów używanych w pracy i pochodzących z dziedziny grywalizacji.

2.1 Wstęp

Niektóre statystki mówią o kilkudziesięciu tysiącach serwisów używających rozwiązań z kategorii Wiki [1]. Na rynku istnieje ponad 140 różnych systemów tego typu, wiele z nich jest naprawdę rozbudowanymi narzędziami [2].

Grywalizacja zwana również gamifikacją jest to stosowanie mechanizmów wykorzystywanych w grach komputerowych do zmieniania zachowań ludzi w celu przywiązania osób do gry, zwiększenia ich motywacji i zaangażowania w sytuacjach niebędących grą. [3]

Gracz – osoba biorąca udział w rozgrywce.

Wybrane elementy grywalizacji:

- Cele – dobrze jest, żeby gracz miał jasno określone cele, które musi realizować.
- Odznaczenia – sposób wynagrodzenia czasu i wysiłku poświęconego na osiągnięcie jakiegoś celu. Trofea powinny być ładne graficznie, ciekawie nazwane oraz zwracające uwagę. Ważne jest, aby znaleźć równowagę pomiędzy przewidywalnością odznaczeń, a ich niezwykłością.
- Wyzwania – jednorazowe bądź cykliczne wydarzenia, które nagradzają gracza za wykonanie jakiejś akcji w określonym czasie.
- Systemy punktowe – nagradzanie punktami po wykonaniu pewnych czynności przez gracza.
- Rankingi – tabele porównujące użytkowników pod jakimś względem, często spotykane w rozwiązaniach zawierających systemy punktowe.
- Poziomy – status, którego osiągnięcie może dawać nowe możliwości, dostęp do wcześniej niedostępnych elementów oraz lepszą renomę w grze.

„Poziom Bossa” jest to odznaczenie, które może zdobyć tylko jeden gracz. Jeśli któryś z graczy zdobędzie takie odznaczenie należy wyżej podnieść poprzeczkę i stworzyć nowe, ciężkie do osiągnięcia zadanie.

W grywalizacji istotne jest, aby nie zasypywać użytkownika zbyt dużą ilością informacji. Stopniowe wprowadzanie użytkownika w świat gry to bardzo dobre rozwiązanie. Nie można zakładać, że gracz z poziomu początkującego przejdzie na etap ekspercki w kilka chwil. Trzeba z czasem wprowadzać gracza, na początku przyznać mu odznaczenie, tak żeby zainteresował się grą.

W projektowaniu tego typu rozwiązań bardzo dobrze sprawdza się programowanie zwinne. Dzięki niemu istnieje możliwość przetestowania nowo zaprojektowanego fragmentu aplikacji w praktyce, sprawdzenia czy wszystko działa jak należy w danym systemie. Zazwyczaj dołożenie grywalizacji do już istniejącego już serwisu jest ciężkie. Stworzenie mechanizmów gry, a dopiero później wpisanie jej w kontekst aplikacji, jest zazwyczaj dużo prostsze.

Zapobieganie próbą oszukiwania systemu stanowi istotny element rozwiązania. Nie da się i nie należy próbować przewidzieć wszystkich nadużyć jakich mogą dopuścić się gracze. Na starcie, podczas tworzenia koncepcji należy skupić się na początkujących graczach, dopiero później należy analizować i poprawiać rozwiązanie.

2.2 Problemy obecnych rozwiązań

W istniejących rozwiązaniach poważnym problemem jest znalezienie osób, które będą chciały tworzyć i utrzymywać treści na wysokim poziomie. Nie należy tylko przyciągnąć autorów, ale także utrzymać ich przy naszym systemie. Wydaje się, że grywalizacja sprawdziłaby się idealnie w bazach

typu Wiki. Mogłaby ona służyć do motywowania zarówno autorów wszelkich treści jak i osób przeglądających artykuły oraz tych uczestniczących w dyskusjach, poprawiających jakość Wiki.

Niestety większość istniejących rozwiązań z daleka omija temat grywalizacji. W niektórych systemach można spotkać elementy takie jak rankingi oraz budowanie społeczności, która ma konkretny cel. Żadna z istniejących Wiki nie została od początku przemyślana i zaprojektowana tak, aby łatwo można było wykorzystać w niej mechanizmy grywalizacji.

System taki jak na przykład Wikipedia co roku zbiera pieniądze na utrzymanie serwerów oraz dalszą działalność platformy. Stworzenie aplikacji, która pozwalałaby na udostępnianie darmowych publicznych Wiki oraz posiadałaby płatne pakiety. Sprzedawać można by choćby prywatne Wiki. Umożliwiłoby to działanie wszystkiego bez konieczności zbierania pieniędzy na dalszą pracę.

Obecne rozwiązania są najczęściej zbiorem wiedzy ogólnej, albo bardziej szczegółowych informacji na konkretny temat. Oznacza to, że użytkownicy zazwyczaj wchodzi na daną stronę w celu uzupełnienia zasobu wiadomości na jakiś temat i na tym kończy się ich wizyta. Bardzo dobrze byłoby, gdyby odwiedzającego udało się zatrzymać i zachęcić go do przeglądania innych materiałów.

2.3 MediaWiki

MediaWiki charakteryzuje się bardzo prostym interfejsem. Jest oprogramowaniem dostępnym na licencji GPL. Aby stworzyć prywatną Wiki należy zainstalować aplikację na serwerze obsługującym PHP i jedną z poniższych baz danych:

- MySQL,
- MariaDB,
- PostgreSQL,
- SQLite 3 [4].

Konfiguracja i utrzymanie tego rozwiązania wymaga fachowej wiedzy. W tym systemie zawarty jest edytor wizualny. Osoba z wiedzą techniczną może dowolnie dostosować wygląd i funkcjonalność strony. Za pomocą tego oprogramowania istnieje możliwość stworzenia wewnętrznego firmowego repozytorium wiedzy, dostępnego tylko w sieci lokalnej.

Wokół MediaWiki skupiona jest bardzo duża społeczność tłumacząca ją na wiele języków, pisząca liczne dodatki oraz boty automatyzujące pewne działania.

Kluczowymi funkcjami tego rozwiązania są:

- edytor WYSIWYG,
- eksport do PDF, HTML,
- klawisze szybkiego dostępu,
- repozytoria plików,
- statystyki,
- śledzenie zmian,
- dyskusje,
- rozszerzenia.

Na Rysunek 1 przedstawiono stronę główną projektu Media Wiki działającą w oparciu o opisane rozwiązanie. Po lewej stronie znajduje się charakterystyczne dla tego systemu menu wraz z logiem na samej górze.

MediaWiki.org [Discussion](#) [Read](#) [View source](#) [View history](#)

Welcome to MediaWiki.org

MediaWiki is a [free software](#) [open source](#) [wiki](#) package written in [PHP](#), originally for use on [Wikipedia](#). It is now also used by several other projects of the non-profit [Wikimedia Foundation](#) and by many other wikis, including this website, the home of MediaWiki.

Use the links below to explore the site contents. You'll find some content translated into other languages, but the primary documentation language is English.

For general questions about MediaWiki see the [communication page](#) or ask at the [support desk](#).

[About this site](#) | [About MediaWiki](#) | [Download](#) | [Help and support](#) | [Contribute](#)

Using MediaWiki



What is a wiki?

- Learn [how to navigate](#)
- Learn [how to edit a](#)

System Administration



Install & configure MediaWiki

- [Upgrade](#) an existing MediaWiki installation

Developing & Extending



Become a MediaWiki hacker

- Learn to use our

Rysunek 1 Strona główna projektu MediaWiki

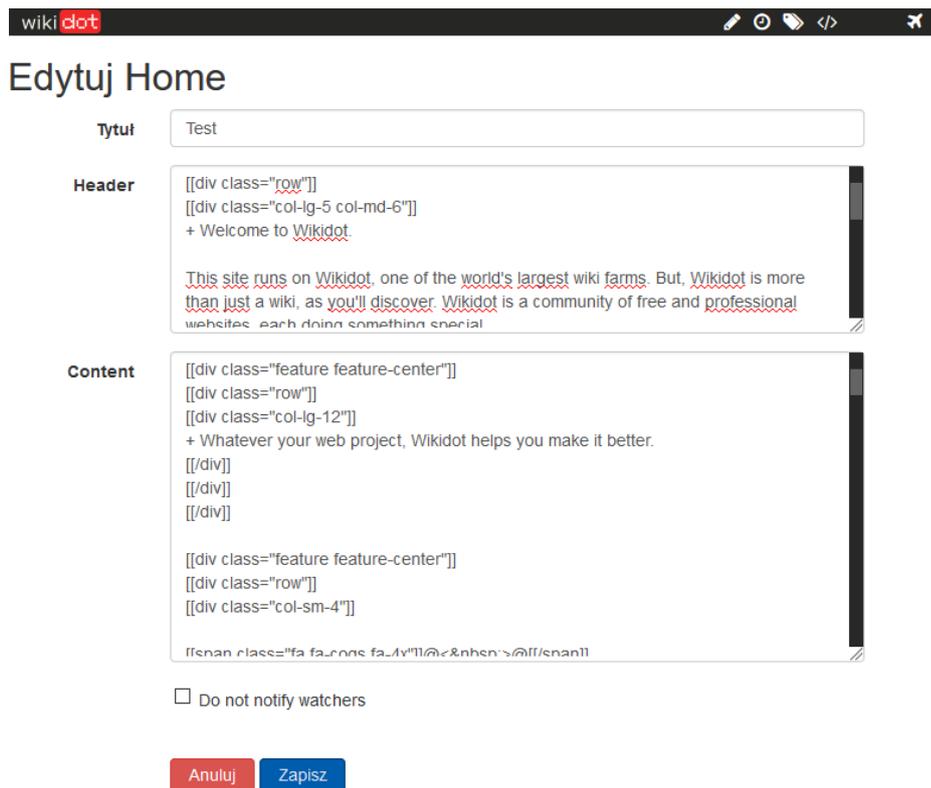
Źródło: Opracowanie własne

2.4 WikiDot

Jest to oprogramowanie polskiej firmy dostępne na licencji AGPLv3. Darmowy pakiet pozwala na stworzenie pięciu Wiki, każda z nich może zajmować co najwyżej 300 MB. Założone repozytorium może mieć przypisany jeden z trzech statusów:

- prywatna – tylko członkowie określonej grupy mogą ją wyświetlić
- ograniczona – tylko uczestnicy mogą ją edytować
- publiczna - każdy ma możliwość edytowania i oglądania.

Poważną wadą jest brak edytora WYSIWYG. W tym systemie wbudowany jest edytor szablonów, umożliwiający dostosowanie wyglądu strony do potrzeb użytkownika. Domyślnie używany jest popularny responsywny framework - Bootstrap. Edycja stron jest dość skomplikowana, daje za to duże możliwości spersonalizowania strony. Przykład pokazujący język znaczników, w którym formatuje się treści znajduje się na Rysunek 2.



Rysunek 2 Przykładowa strona edycji artykułu w WikiDot

Źródło: Opracowanie własne

Zakładanie nowej strony w WikiDot jest stosunkowo proste, wszystkie pola, które należy wypełnić mieszczą się na jednej stronie (Rysunek 3). Pierwszymi parametrami, które należy uzupełnić są nazwa oraz lista tagów. W pełni obsługiwane jest 12 języków, pozostałe znajdują się w fazie eksperymentalnej. Możemy skorzystać z jednego z 20 dostępnych szablonów. Na formularzu jest możliwość wskazania subdomeny adresu wikidot.com, pod którą będzie zainstalowana stworzona Wiki. Godną uwagi funkcją jest możliwość włączenia reklam na Wiki oraz zarabiania dzięki nim. Każdy użytkownik może zostać członkiem Wiki. Istnieją trzy kategorie, do których można należeć:

- Administratorzy,
- Moderatorzy,
- Zwykli użytkownicy.

Osoby z pierwszej kategorii mogą wysyłać newsletter do użytkowników zapisanych do ich Wiki. Nowych uczestników można w danym projekcie dopisać poprzez podanie nazwy użytkownika lub adres e-mail.

W tym rozwiązaniu nawet dostęp do panelu administratora nie jest uzyskiwany przez protokół https. Jest to poważna luka w bezpieczeństwie i zagrożenie dla prywatności użytkownika. Szyfrowane połączenie włączane jest dopiero w wersji premium.

Właściciel ma możliwość włączenia dostępu do swojej Wiki poprzez API, robienia backupów oraz blokowania użytkowników bądź adresów IP.

The image shows the Wikidot website creation interface. At the top, there is a navigation bar with 'wiki dot' logo and links for 'Features', 'Opinions', 'Pricing', and 'Advertising'. Below this is a 'Sites: 1 / 5' indicator and a '+ Create site' button. The main heading is 'Get your new Wikidot site' with a green checkmark icon. The form consists of several fields: 'Title' (with a description: 'Appears on the top-left corner of your Wikidot site.'), 'Tagline' (with a description: 'Appears beneath the name.'), 'Web address' (with a dropdown for '.wikidot.com'), 'Language' (set to 'English'), and 'Template' (set to 'Standard Template'). A preview of the 'Standard Template' is shown, featuring a blue header and a 'Welcome to Wikidot' message.

Rysunek 3 Zakładanie nowej strony w WikiDot

Źródło: Opracowanie własne

2.5 Wikia

Wikia jest zamkniętym rozwiązaniem nastawionym na repozytoria wiedzy związane głównie z rozrywką. Na Rysunek 5 widać stronę główną tego systemu. To rozwiązanie nie posiada ograniczenia liczby założonych Wiki. Proces tworzenia nowej strony składa się z czterech prostych kroków, podczas których należy uzupełnić następujące pola:

- nazwa,
- adres pod którym znajdować będzie się nasza Wiki,
- język,
- krótki opis,
- kategoria,
- wybór jednego z jedenastu dostępnych motywów graficznych.

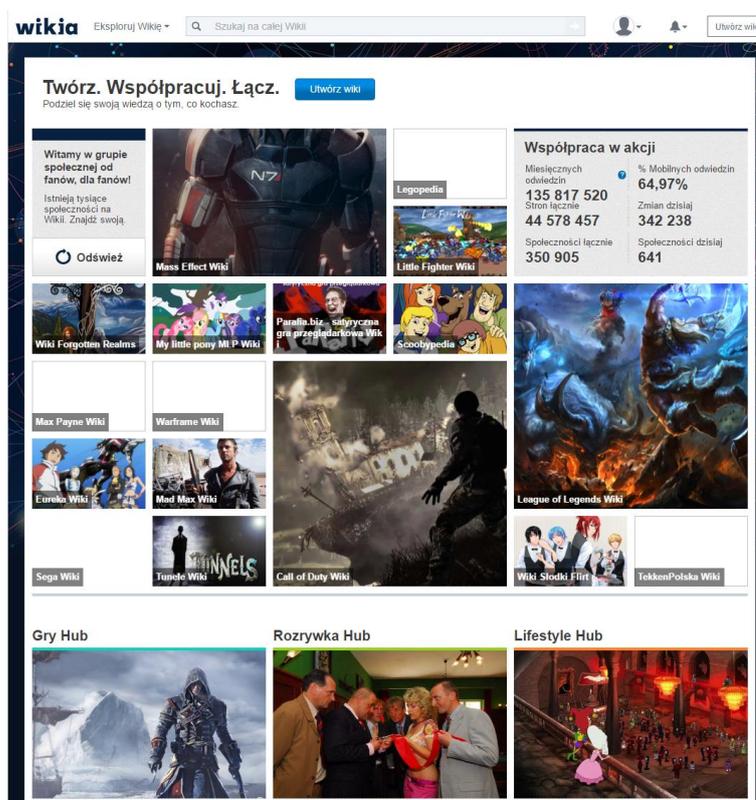
Wszystkie stworzone zasoby są publicznie dostępne. Strona zawiera reklamy, za wyświetlanie których administrator nic nie dostaje i nie ma wpływu na ich wyświetlanie. Do edycji treści służy prosty edytor wizualny umożliwiający między innymi:

- dzielenie na nagłówki i tekst,
- pogrubianie
- stosowanie kursywy,
- linkowanie,
- wstawianie filmów i zdjęć,
- tworzenie list,
- dodawanie przypisów.

Zarządzanie stronami oraz nawigacją jest stosunkowo proste. Mamy możliwość eksplorowania zasobów udostępnionych przez innych użytkowników za pomocą strony najpopularniejszych Wiki, wyszukiwarki oraz drzewa kategorii. Podczas tworzenia i przeglądania zawartości nie mamy możliwości użycia szyfrowania.

Każda strona znajduje się w jednej z siedmiu podstawowych kategorii:

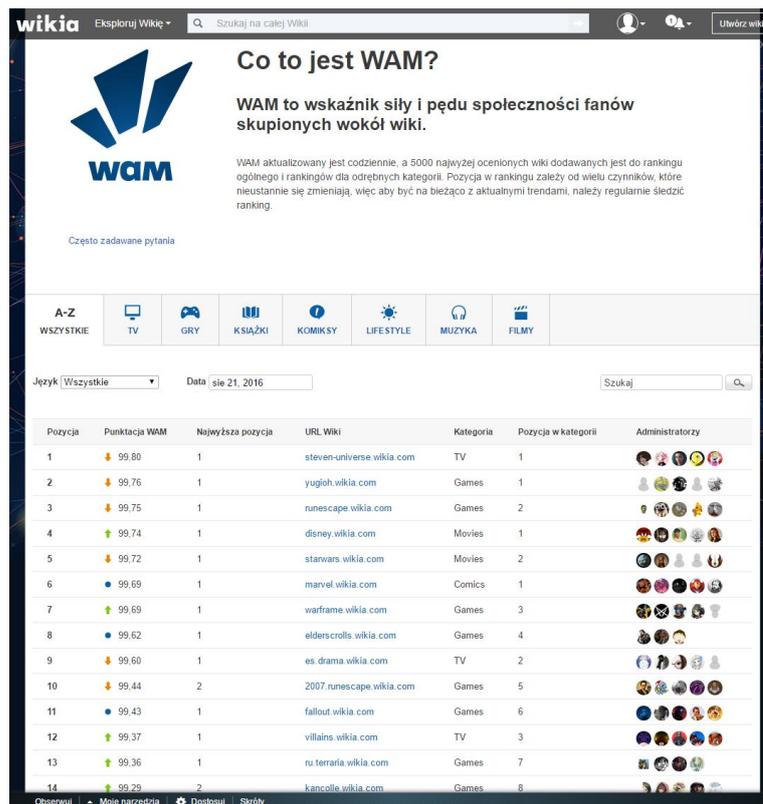
- Gry,
- Filmy,
- Komiksy,
- Książki,
- TV,
- Muzyka,
- Lifestyle.



Rysunek 4 Strona główna Wikia

Źródło: Opracowanie własne

Na stronie Wikia znajduje się ranking najpopularniejszych Wiki zwany WAM (Rysunek 5), jest on aktualizowany raz dziennie. Osoby zaangażowane w tę społeczność nazywane są „Wikianinami” [5]. Jest to jeden z elementów grywalizacji, który możemy znaleźć w tym systemie.



Rysunek 5 Ranking WAM na stronie Wikia

Źródło: Opracowanie własne

2.6 Wikispaces

Po dokonaniu rejestracji użytkownik dostaje darmowy 30 dniowy okres próbny. Najtańszy z dostępnych planów kosztuje 10\$ miesięcznie. W celu utworzenia nowej Wiki należy wybrać w jakim celu zostaje ona utworzona. Do wyboru są następujące opcje:

- Edukacja K-12
- Wyższa edukacja
- Biznes
- Osobiste
- Non-profit
- Rząd
- Inne
- Nie chcę odpowiadać

Podczas procesu zakładania nowej bazy należy podać następujące dane:

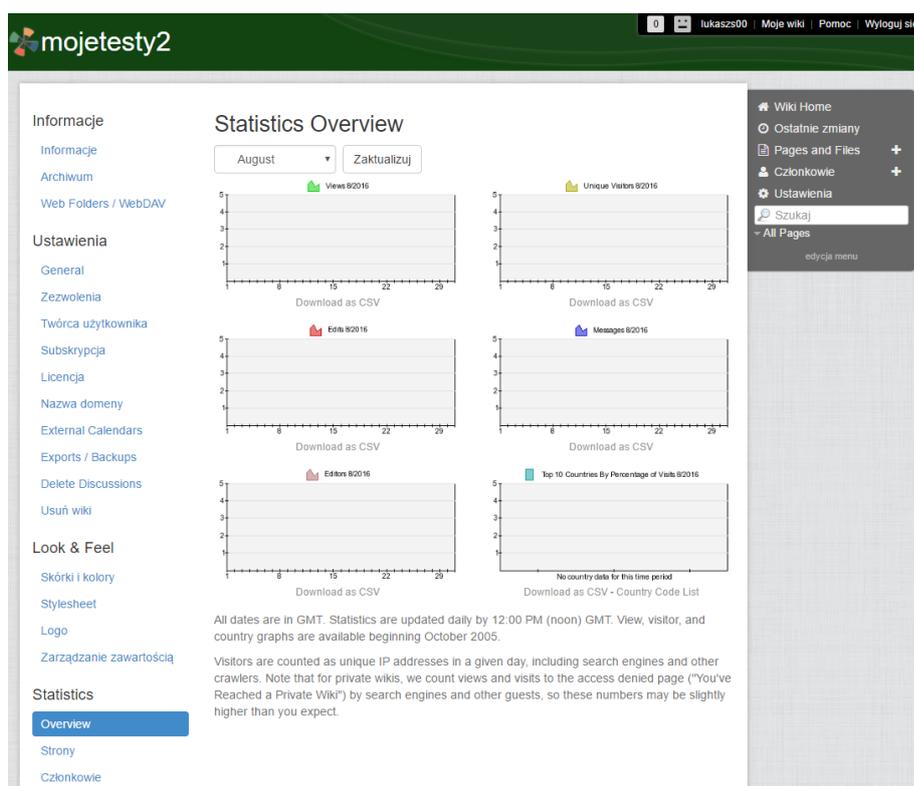
- nazwę, która jest tożsama z subdomeną,
- typ,
- plan opłat.

W Wiki możemy tworzyć nowe artykuły, dodawać pliki oraz zarządzać listą osób uprawnionych do czytania i tworzenia zasobów. Strony można modyfikować za pomocą edytora wizualnego, który pozwala między innymi:

- zastosować pogrubienie, przekreślenie oraz podkreślenie,

- edytować kolor i styl czcionki,
- tworzyć listy,
- zarządzać nagłówkami,
- dodawać pliki,
- wprowadzać linki,
- wstawiać widgety.

Znajduje się tu kilka szablonów, w których za pomocą wizualnego edytora można dostosować kolory przewodnie oraz odcień akcentów. Dużą swobodę w tworzeniu własnego wyglądu strony daje możliwość korzystania z CSS. W opisywanym rozwiązaniu istnieje możliwość dodania własnego loga. Istotną zaletą może być wsparcie dla technologii WebDAV, która pozwala na przeglądania Wiki, tak jakby pliki znajdowały się na lokalnym komputerze.



Rysunek 6 Panel administratora w Wikispaces

Źródło: Opracowanie własne

W panelu administratora znajduje się pięć wykresów pokazujących w czasie liczbę:

- wyświetleń,
- unikalnych odwiedzających,
- edycji,
- wiadomości,
- edytujących.

Ostatni wykres przedstawia procentowy udział odwiedzających z dziesięciu najpopularniejszych krajów. Wyżej wymienione statystyki widoczne w panelu administratora znajdują się na Rysunek 6. Domyślnie szablony korzystają z responsywnego frameworka jakim jest Bootstrap. W planie za 40\$ miesięcznie do dyspozycji użytkownika jest rozszerzone zarządzanie uprawnieniami, pełne API oraz możliwość dodania własnej domeny.

Opisywany system jest rozwiązaniem całkowicie zamkniętym i nie posiada w swojej ofercie żadnych darmowych pakietów. W tej aplikacji jak i wszystkich pozostałych użytkownicy mają możliwość przeglądania historii zmian.

2.7 Podsumowanie

Każde rozwiązanie jest inne oraz posiada swoje wady i zalety, jednak żaden z istniejących systemów nie został zaprojektowany pod kątem grywalizacji i motywowania użytkowników. Wikia jako jedyna w standardzie zawiera proste elementy grywalizacji jakimi są na przykład rankingi najpopularniejszych Wiki. Żadne istniejące oprogramowanie nie wspiera wszystkich elementów grywalizacji. Aby stworzyć skuteczne rozwiązania należy zbierać bardzo duże ilości danych statystycznych, niemalże o wszystkich akacjach użytkowników. Aby system był dobrze zgrywalizowany od początku należy projektować do właśnie pod tym kątem. Ciężko jest dołożyć elementy gry do już istniejącego systemu, jeśli nie została w nim przewidziana. Bardzo istotne jest, aby ciągle analizować posunięcia użytkowników i się do nich dostosowywać.

3 Koncepcja nowego systemu typu Wiki

W poniższym rozdziale przedstawiono koncepcję nowego rozwiązania działającego w chmurze i zawierającego system dodatków wprowadzających grywalizację.

3.1 Wiki

Jest bytem najwyższego poziomu, są z nią powiązane kategorie i artykuły. Każdy użytkownik może utworzyć swoją Wiki albo zacząć edytować już istniejącą, o ile będzie miał do tego uprawnienia. Aby nadać dostęp konkretnej jednostce można utworzyć link pozwalający na wyświetlanie bądź edycję prywatnej lub publicznej Wiki. Osoba, która otworzy Wiki przez specjalne łącze może użyć przycisku subskrypcji, aby zapamiętać tę stronę i w przyszłości móc łatwo dotrzeć do tych zasobów oraz otrzymywać powiadomienia o wszelkich zmianach. Każda Wiki w systemie może mieć przypisane następujące właściwości:

- Nazwa
- Opis – krótki tekst, który wyświetlany jest na wszystkich listach
- Grafika
- Status
 - Publiczna – każdy może ją oglądać, tylko właściciel i osoby posiadające link mogą edytować.
 - Prywatna – dostęp do oglądania i edycji ma administrator i użytkownicy, którzy otrzymali link do edycji lub wyświetlania.
- Strona główna – jest to strona, która będzie wyświetlana po wejściu Wiki. Powinny znajdować się na niej kluczowe informacje dotyczące na przykład przeznaczenia repozytorium wiedzy.

3.2 Kategorie

W Wiki mogą znajdować się kategorie, które przyjmują strukturę drzewa, dzięki czemu użytkownik może w łatwy sposób nawigować oraz łatwo odnaleźć interesujące go informacje. Do każdego węzła można przypisać dowolną liczbę artykułów oraz innych węzłów. We wszystkich Wiki powinna istnieć systemowa kategoria, do której należałyby artykuły bez kategorii.

3.3 Artykuły

Jest to najważniejsza składowa całego systemu. Artykuły zawierają wszelkie informacje, tworzenie ich powinno być możliwie proste. Wprowadzanie treści powinno odbywać się za pomocą edytora wizualnego, który będzie umożliwiał między innymi:

- pogrubienie,
- kolorowanie czcionki,
- ustawiania nagłówków,
- linkowanie,
- określanie sposobu ułożenia tekstu.

Wszystkie artykuły powinny być indeksowane, aby później można było odnaleźć wyszukiwane hasła. Przed skatalogowaniem powinny zostać wykonane dwa kroki. Najpierw z tekstu należy usunąć wszelkie znaczniki formatujące, ponieważ nie zawierają one treści i zmniejszają czytelność dla komputera. Następnie należy podzielić dany tekst na słowa.

Nazwa artykułu powinna zostać zaindeksowana w taki sposób, aby możliwe było wyszukiwanie tekstu dokładnie tak jak wpisano. Jeśli zostanie wpisany fragment słowa to artykuł powinien zostać odnaleziony.

3.4 Wersjonowanie

Ważnym elementem jest zapisywanie informacji o tym co dokładnie i kiedy zostało zmienione. Osoba z prawem do edycji ma możliwość przeglądania wszystkich wersji artykułu. Dzięki tym danym w prosty sposób można ustalić, dlaczego finalna postać artykułu wygląda właśnie w ten sposób. Podczas śledzenia zmian nie jest kluczowe zapisywanie zmodyfikowanej nazwy i parametrów artykułu, istotne są wszelkie edycje treści.

3.5 Wiadomości

Użytkownicy powinni mieć możliwość wymieniać między sobą prywatnych wiadomości. Dzięki temu w prosty sposób można skontaktować się z autorem Wiki, w której edycji chce ktoś pomóc. Na stronie dialogu powinno być wyraźnie kto z kim rozmawia. Wskazane jest aby komunikaty nieprzeczytane były specjalnie oznaczone na przykład gwiazdką, ponadto użytkownik powinien dostawać powiadomienie o otrzymaniu nowego przekazu. Dobrze byłoby gdyby użytkownik miał możliwość wyszukiwania informacji w historii.

3.6 Wyszukiwarka

W stworzonym rozwiązaniu powinna znaleźć się wyszukiwarka, która będzie efektywnie przeszukiwać zawartość strony pod względem fraz podanych przez użytkownika. Ważne jest aby szukała ona zarówno w nazwach jak i w treści pomijając nieistotne znaczniki formatowania.

Treść artykułu powinna być przeszukiwana pod względem słów kluczowych. Podczas wyszukiwania nie jest istotna kolejność słów ważne jest, aby to słowo znajdowało się w zawartości strony. Istotne jest także, aby nie pokazywać nieistotnych wyników. Jeśli użytkownik wpisze słowo „gry” nie powinien znaleźć artykułów zawierających „grywalizacja” w treści. Za to w nazwach artykułów taki przypadek jest zalecany, jeśli ktoś poda „Lorem ips” powinien odnaleźć także „Lorem ipsum”.

3.7 Powiadomienia

W stworzonej aplikacji powinno być miejsce widoczne na każdej stronie, w którym będą pojawiać się powiadomienia. Powinny one mieć trzy istotne parametry:

- małą grafikę,
- tekst,
- link do elementu, którego dotyczą.

3.8 System dodatków

System dodatków realizuje koncepcję grywalizacji. Osoba pisząca wtyczkę powinna mieć do dyspozycji wszystkie funkcje wybranego języka. Aby możliwe było właściwe działanie systemu programista powinien móc:

- tworzyć modele odpowiedzialne za przechowywanie danych,
- dodać dowolną stronę,
- wstawiać elementów menu,
- dołączyć swoją akcją po wbudowanych zdarzeniach,
- załączać własne
 - pliki graficzne
 - kod JavaScript
 - CSS
 - HTML.

Ze względu na fakt, że grywalizacja wymaga zbierania wielu danych statystycznych bardzo istotne jest, aby programista miał do dyspozycji prosty mechanizm zapisywania dowolnych struktur do

bazy. Żeby dodatki mogły sprawnie funkcjonować istotne jest, aby programista miał możliwość tworzenia własnych stron, które będą mogły zawierać HTML, JavaScript, CSS oraz pliki graficzne. Dobrze byłoby gdyby twórca mógł edytować pewne elementy takie jak menu, ponieważ ciężko sobie wyobrazić dodatek związany z grywalizacją, do którego nie byłoby dostępu z poziomu menu. Kluczowe jest napisanie ważnych funkcji, takich jak tworzenie Wiki, z możliwością uruchomienia procedury zaimplementowanej w dodatkach. Może to zostać zrealizowane na przykład za pomocą rozgłaszania zdarzeń systemowych.

3.8.1 Ulubione

Użytkownik powinien mieć możliwość dodania Wiki do ulubionych, aby łatwo odnaleźć ją w przyszłości. W przeciwieństwie do subskrypcji w przypadku wstawienia do ulubionych i zaistnienia jakiegoś zdarzenia na Wiki osoba nie dostanie powiadomienia o danej akcji. Ten dodatek będzie wymagał zapisywania dodatkowych informacji w bazie danych.

3.8.2 Osiągnięcia

Jest to bardzo istotny dodatek umożliwiający graczowi otrzymywanie odznaczeń za wykonanie przewidzianych akcji. W tej wtyczce powinny się znaleźć zarówno zwykłe jak i specjalne odznaki. Pierwsze z wcześniej wymienionych może zdobyć każdy i nie ma na nich żadnych limitów ilościowych. Trofea specjalne przewidziane jako realizacja „poziomu bossa”, może je dostać tylko ten użytkownik, który jako pierwszy wykona określone akcje. Niezwykle istotne jest, aby odznaczenia były estetyczne oraz ciekawe graficznie.

W grywalizacji kluczowe jest ciągłe badanie i rozwijanie wszystkich elementów, dlatego nie da się określić wszelkich odznaczeń, które powstaną. Na początku w systemie powinny się znaleźć odznaczenia, które zostaną przyznane za:

- podglądanie profili innych użytkowników,
- kliknięcie w link osoby polecającej system, przez kogoś innego,
- czytanie artykułów,
- pisanie artykułów,
- zakładanie nowych Wiki,
- za kompletne uzupełnienie profilu.

Oczywiście dla każdego z powyżej opisanych przypadków może powstać odznaczenie realizujące element „poziomu bossa”. Na przykład dla osoby, która jako pierwsza napisze sto artykułów.

3.8.3 System punktowy

Bardzo istotne jest dobranie dobrego algorytmu liczącego punkty, powinien on brać pod uwagę to, że należy wciągać w grę zarówno użytkowników, którzy chcą korzystać z systemu jako źródła wiedzy, jak i tych którzy chcą udzielać się jako twórcy zasobów. Nie da się przewidzieć jaki będzie stosunek liczby osób tworzących Wiki do osób czytających, dlatego ułożenie dobrego algorytmu przydzielającego punkty jest bardzo trudne. Są dwa możliwe podejścia. Jedno to próbować dopasować współczynniki algorytmu tak, żeby można było porównywać twórców i przeglądających, a drugie to tworzenie wszystkich porównań i rankingów osobno dla osób autorów i wyświetlających.

W pierwszym podejściu twórcy zasobów powinny dostawać dużo punktów za czynny udział w poszerzaniu ilości dostępnych materiałów, sporo więcej niż osoby przeglądające materiały. Jednocześnie należy pamiętać, że osoba która tylko przegląda treści powinna mieć możliwość osiągnięcia podobnej ilości punktów co pisarze.

Drugie podejście jest dużo prostsze ponieważ zawsze będzie można porównywać między sobą osoby, które tworzą zasoby oraz na innym zestawieniu te które przeglądają. Problemem mogą być

osoby, które będą występować w obu rankingach. Kłopotliwe będzie też mnożenie wszystkich raportów. W takim podejściu nie będzie można jednoznacznie wskazać najlepszego gracza, można będzie wyłonić tylko osobę najlepszą w danej dziedzinie.

W obu rozwiązaniach trzeba wziąć pod uwagę bardzo dużo czynników i dobrze dostosować wszystkie wagi. Poniżej znajduje się Tabela 1 przedstawiająca proponowaną liczbę punktów za wykonanie poszczególnych akcji przez użytkownika. Dane z tej tabeli muszą zostać zweryfikowane w działającym rozwiązaniu.

Opis	Rodzaj akcji	Liczba punktów
Utworzenie Wiki	Twórca	5000
Stworzenie kategorii	Twórca	50
Dodanie nowego artykułu	Twórca	200
Modyfikacja Wiki	Twórca	100
Edycja artykułu	Twórca	100
Wysłanie wiadomości	Przeglądający	1
Przeczytanie wiadomości	Przeglądający	1
Otwarcie Wiki	Przeglądający	20
Przeglądanie artykułu	Przeglądający	25
Wyświetlanie profilu użytkownika	Przeglądający	10
Uzupełnienie profilu użytkownika	Przeglądający	200
Utworzenie subskrypcji Wiki	Przeglądający	10
Dodanie użytkownika do obserwowanych	Przeglądający	10
Polecenie systemu	Przeglądający	500
Założenie konta	Przeglądający	100

Tabela 1 Liczba punktów za wykonanie poszczególnych akcji

3.8.4 Rangi

Nadawanie rang użytkownikom jest bardzo starym, stosowanym od dawna elementem grywalizacji, pozwalającym bardziej zmotywować graczy. Status użytkownika zależałoby od ilości przyznanego doświadczenia, zgodnie ze stworzonym systemem punktowym. W tym rozwiązaniu powinny być przygotowane przynajmniej trzy rangi:

- Młoda sówka od zera do pięćdziesięciu tysięcy punktów
- Dojrzała sówka od pięćdziesięciu tysięcy punktów do pięciuset tysięcy punktów
- Sówka weteran powyżej pięciuset tysięcy punktów

Po pierwszych testach może się okazać, że będzie istniała konieczność korekcy ustanowionych progów punktowych.

3.8.5 Rankingi

Opisywana koncepcja systemu zakłada zaprojektowanie przynajmniej kilku rankingów graczy. Ranking powinien być odświeżany tylko raz dziennie ze względu na wydajność. Tworzenie takiego raportu przy każdym żądaniu byłoby zadaniem bardzo obciążającym serwer, zwłaszcza przy większej liczbie użytkowników. Zestawienia powinny brać pod uwagę następujące dane:

- ilość punktów,
- liczbę odznak,
- liczbę stworzonych Wiki,
- liczbę napisanych artykułów.

Istotne jest dobre zaprojektowanie rankingu, musi on pokazywać użytkownikowi, że jest dobry, jednocześnie musi mu uświadamiać, że nie może przestać się starać. W tym celu zestawienie należy podzielić na trzy części:

- trzech pierwszych graczy,
- otoczenie gracza,
- atrapy trzech ostatnich graczy.

W pierwszej części powinni znajdować się trzej najlepsi gracze w danej kategorii, jest to dla nich nagroda, ponieważ wszyscy inni użytkownicy mogą zobaczyć ich nazwy. W następnej części powinno znaleźć się najbliższe otoczenie gracza, czyli pięciu lepszych od niego graczy i pięciu gorszych. Koniec rankingu powinien być atrapą, należy tam pokazać trzy ostatnie fikcyjne konta. Grywalizacja polega na ciągłym nagradzaniu gracza. Nawet dopiero po zarejestrowaniu się w systemie, dana osoba nie może znaleźć się na końcu zestawienia, ponieważ byłaby to dla niej kara i mógłby stracić chęć do dalszego uczestnictwa w rozgrywce. Ostatnie sztuczne konta powinny mieć tyle punktów ile gracz dostaje na starcie, wtedy patrząc na ranking widać by było jak wiele osiągnął od początku swojej gry.

Zestawienie graczy powinien zawierać kilka istotnych z punktu widzenia grywalizacji kolumn:

- pozycje gracza w rankingu,
- zdjęcie profilowe gracza,
- widoczna nazwa użytkownika z linkiem do profilu,
- liczba rankingowa.

3.8.6 Wyzwania

Koncepcja przewiduje możliwość zaprojektowania specjalnych wyzwań pojawiających się okazjonalnie dla wszystkich użytkowników oraz takich, które będą widoczne w zależności od pewnych parametrów jak na przykład staż gracza albo ilość utworzonych artykułów. Zdarzenie te pozwalałyby zdobywać dodatkowe punkty oraz specjalne odznaczenia. Wyzwania przyciągają gracza oraz uzależniają go od częstego przeglądania platformy, ponieważ będąc nieobecny może przegapić jakieś wyzwanie, które trwa tylko określony czas i już nie będzie mógł zdobyć specjalnego odznaczenia. Wyzwania mogą się pojawiać w okolicy różnych świąt, takich jak na przykład Wielkanoc albo Boże Narodzenie.

3.9 API

Planowane jest przygotowanie interfejsu API umożliwiającego tworzenie aplikacji mobilnych, aplikacji desktopowych oraz dodatków na strony internetowe. W nowoczesnych serwisach publiczne interfejsy do komunikacji są bardzo popularne, ponieważ dają programistą możliwość skorzystania z serwisu i stworzenie programu używającej danego systemu. W tym przypadku API powinno przede wszystkim służyć do odczytywania wszelkich danych z systemu. Z punktu biznesowego dobrze byłoby zastanowić się nad ograniczeniami dotyczącymi wykorzystywania systemu przez obce narzędzia. Niewykluczone, że przydałyby się pewne ograniczenia na ilości wywołań. Po przekroczeniu tych limitów dobrze by było wprowadzić płatne plany.

3.10 Aplikacja mobilna

Aplikacja mobilna powinna korzystać z wystawionego API. W założeniu jest ona tylko dla użytkowników wykorzystujących system jako źródło informacji. W części mobilnej nie będzie możliwości edycji Wiki czy artykułów. Zaprojektowany program powinien umożliwiać:

- przeglądanie Wiki,
- czytanie artykułów,
- wyszukiwanie artykułów,
- przeglądanie i wysyłanie wiadomości.

3.11 Chmura

Cały system powinien działać w chmurze, aby proces utrzymania aplikacji był możliwie uproszczony.

Dzisiejsze chmury mają wbudowane mechanizmy, które umożliwiają automatyczne skalowanie aplikacji. Zewnętrzne serwerownie zapewniają większe bezpieczeństwo danych niż własne rozwiązania. Dostawcy usług dbają, aby kopie bezpieczeństwa były regularnie wykonywane. Centra danych objęte są ścisłymi regułami. Dane replikowane są w wielu lokalizacjach, zapewnia to szybszy dostęp dla użytkowników znajdujących się w różnych miejscach na całym świecie, a ponadto zapewnia to dodatkową ochronę danych. Chmury posiadają bardzo duże moce obliczeniowe i przepustowości dzięki temu przeprowadzenie ataku typu DDoS będzie trudniejsze do realizacji, dodatkowo dostawcy usługi często mają zabezpieczenia oraz potrafią szybko wykrywać i reagować w tego typu przypadkach.

Utrzymanie własnych serwerów jest zazwyczaj dużo droższe i narażone na więcej błędów oraz niebezpieczeństw, niż rozwiązania chmurowe.

4 Użyte technologie oraz oprogramowanie

W poniższym rozdziale znajduje się lista narzędzi i technologii wykorzystanych do stworzenia pracy magisterskiej.

4.1 Część Webowa

Niniejszy rozdział opisuje technologie i narzędzia użyte podczas implementacji webowej części aplikacji.

4.1.1 Google App Engine

Potężna platforma do budowania aplikacji i automatycznego skalowania w chmurze. Możliwość pisania w jednym z popularnych języków:

- Python
- Java
- PHP
- GO

Google App Engine ma wbudowane skanery bezpieczeństwa, które sprawdzają aplikacje pod kątem popularnych luk w zabezpieczeniach. W tym rozwiązaniu do dyspozycji programisty jest zarówno baza NoSQL jak i klasyczna SQL.

Baza NDB jak większość baz typu NoSQL nie posiada schematu, wystarczy stworzyć nowy model i już możemy z niego korzystać. Aby utrwalić obiekt wystarczy wywołać jedną metodę i od razu zostanie zapisany w bazie.

Ten system umożliwia szybkie pisanie i wdrażanie aplikacji. Opisywana platforma deweloperska zapewnia bardzo prostą konfigurację serwera, sprowadza się ona do kilku nie zbyt trudnych kroków oraz przesłania kodu do chmury [6].

Google App Engine jest rozwiązaniem zamkniętym, więc cały projekt jest zależny od firmy Google. Dla właściciela projektu bardzo dużym problemem byłoby:

- wycofanie się tej korporacji ze wsparcia projektu,
- zablokowanie projektu,
- znaczący wzrost opłat za hosting.

Kłopotliwe na tej platformie jest instalowanie dodatków firm trzecich. W Pythonie istnieje bardzo dużo gotowych rozwiązań, które ciężko jest użyć w Google App Engine.

4.1.2 Python

Interpretowany język programowania wysokiego poziomu, posiadający rozbudowaną bibliotekę standardową. Został on zaprojektowany w taki sposób, aby maksymalnie zwiększyć czytelność kodu. Można używać trzech paradygmatów programowania:

- obiektowego,
- proceduralnego,
- funkcyjnego.

Charakteryzuje się on dynamicznym typowaniem zmiennych, automatycznym zarządzaniem pamięci oraz późnym wiązaniem nazw zmiennych oraz funkcji [7].

Opisywany język programowania ma wiele zalet, poniżej znajduje się lista cech, dzięki którym został on wybrany spośród tych dostępnych w Google App Engine.

- Interpretowalność ta cecha umożliwia w stosunkowo prosty sposób stworzenie potężnego systemu dodatków.
- Kod języka jest przejrzysty oraz bardzo czytelny.
- Prosty do nauki.

Oczywiście język ten ma też swoje wady. Za jego słabą stronę można uznać interpretowalność, która jest również główną zaletą. Przez brak optymalizacji podczas kompilacji programista musi poświęcać więcej uwagi wydajności kodu. Ponadto aplikacja musi być bardzo dokładnie przetestowana. Jeśli autor programu popełni literówkę w wywołaniu funkcji, która znajduje się w instrukcji warunkowej, to ten błąd zostanie ujawniony dopiero, gdy przepływ sterowania wskaże dokładnie tę linię kodu.

4.1.3 HTML

Niezwykle popularny hipertekstowy język znaczników stosowany do opisywania stron WWW [8].

4.1.4 JavaScript

Skryptowy język programowania szeroko wykorzystywany do wzbogacania stron internetowych. Zapewnia on bardziej dynamiczną interakcję z użytkownikiem. [9]

4.1.5 JQuery

Jest to wieloplatformowa, niezwykle funkcjonalna biblioteka do języka JavaScript. Pozwala ona w łatwy sposób manipulować drzewem dokumentu HTML, sprawia że zarządzanie zdarzeniami, animacje i wywołania AJAX'a są dużo prostsze [10].

4.1.6 CSS 3

Kaskadowe arkusze stylów w prosty sposób opisują jak ma wyglądać strona internetowa. CSS umożliwia dostosowywanie wyglądu do różnych urządzeń. W najnowszej wersji tego standardu znajdują się moduły takie jak:

- animacje,
- selektory,
- efekty tekstowe,
- obramowania,
- tła,
- możliwość zaokrąglenia rogów ramki [11].

4.1.7 Bootstrap 3

Popularny, responsywny framework korzystający z HTML, CSS i JavaScript wydawany na licencji MIT. Zawiera elementy za pomocą których w łatwy sposób można budować stronę WWW. Zawiera komponenty takie jak:

- wektorowe glify,
- listy,
- panele,
- przyciski,
- pola formularzy,
- listwa nawigacyjna [12].

4.1.8 Font Awesome

Jest to biblioteka dostarczająca skalowalne, wektorowe ikony stworzone jako czcionka, dzięki czemu można je dowolnie dostosowywać. Programista w prosty sposób może zmieniać ich:

- kolor,
- rozmiar
- dodać cień,
- oraz wykorzystać dowolną funkcje z CSS 3 [13].

4.1.9 CKEditor

CKEditor jest największym i najpopularniejszym edytorem WYSIWYG. Zawiera on bardzo duży zestaw rozszerzeń, można do niego wklejać tekst z dokumentów Word. [14]

4.1.10 Google App Engine Boilerplate

Jest zbiorem funkcji, procedur i klas, które są używane niemal w każdej aplikacji Webowej przy użyciu platformy Google App Engine. Jest to typowy szablon aplikacji [15].

4.1.11 PyCharm

Niezwykle potężny i rozbudowane IDE dla języka Python ze wsparciem dla Google App Engine. Jego główne zalety to:

- autouzupełnianie,
- szybka i bezpieczna refaktoryzacja,
- możliwość wgrzywania plików bezpośrednio do chmury Google,
- obsługa HTML, JavaScript i CSS
- wbudowana aplikacja do testowania RESTa,
- debugger,
- wbudowana konsola,
- proste i efektywne wyszukiwanie w całym projekcie [16].

4.2 Część Mobilna

Poniższy rozdział zawiera informacje o rozwiązaniach, z których autor pracy korzystał podczas tworzenia prototypu na Androida.

4.2.1 Java

Kompilowany, obiektowy język programowania, którego kod bajtowy uruchamiany jest przez maszynę wirtualną, zawiera bogatą bibliotekę standardową. Charakteryzuje go:

- automatyczne zarządzanie pamięcią,
- silne typowanie,
- możliwość uruchomienia programu na wielu platformach [17].

Jest to język stosunkowo prosty język, niezależny od żadnej platformy. Jego ważną zaletą jest wypisywanie całego stosu wywołania z informacją o błędzie, dzięki temu bardzo łatwo można zlokalizować wyjątek, który wystąpił w programie.

4.2.2 Android SDK

Android SDK jest potężnym, funkcjonalnym frameworkiem pozwalającym tworzyć innowacyjne mobilne aplikacje. Dzięki przemyślanemu mechanizmowi dostosowywanie interfejsu użytkownika, można w łatwy sposób dopasować program do każdego urządzenia. Istnieje wiele dopracowanych elementów API, które można łatwo użyć do budowy systemu [18].

Android to bardzo popularny otwarty system operacyjny na urządzenia mobilne. Ważnymi cechami opisywanego rozwiązania są między innymi:

- szybki, przyjazny emulator,
- szablony,
- lokalizacja aplikacji,
- fragmenty,
- zadania asynchroniczne,
- intencje,
- powiadomienia,
- szczegółowe uprawnienia.

4.2.3 Picasso

Jest to bardzo przyjazna biblioteka służąca do pobierania i buforowania obrazków na Androida. Pozwala wczytywać zarówno lokalne pliki jak i te znajdujące się w lokalizacji zdalnej. Można dokonywać dzięki niej prostych przekształceń takich jak na przykład skalowanie [19].

4.2.4 Android Studio

Android Studio jest to bardzo szybkie i bogate w przydatne funkcje IDE. Tworzenie aplikacji jest niezwykle wydajne dzięki błyskawicznemu uruchamianiu zmienionego fragmentu kodu na urządzeniu lub w emulatorze. Podpowiadanie składni oraz sugerowanie akcji możliwych do wykonania na danym fragmencie programu bardzo przyspiesza proces tworzenia oprogramowania. Opisywane IDE charakteryzuje się:

- inteligentnym edytorem kodu,
- bezpieczną refaktoryzacją,
- szybkim podglądem szablonów,
- pełnym wsparciem dla Android SDK [18].

4.3 Część wspólna

Ten rozdział zawiera informację o programach, które zostały użyte podczas tworzenia pisemnej części pracy oraz były wykorzystywane zarówno w aplikacji webowej jak i mobilnej.

4.3.1 Git / Bitbucket

Git jest oprogramowaniem do zapisywania i śledzenia zmian przeważnie w kodzie źródłowym. Wydawany jest na licencji GPL2. Dzięki temu, że jest on rozproszonym systemem kontroli wersji może działać offline. Na kopi znajdujących się w lokalnym systemie plików można dokonać dowolnych modyfikacji, przywracać wcześniejsze wersje oraz porównywać zmiany.

Głównymi cechami rozwiązania Bitbucket dostarczanego przez firmę Atlassian są między innymi:

- darmowe prywatne repozytoria w Git,
- system do śledzenia zgłoszeń,

- możliwość przeglądania i porównywania kodu w przeglądarce,
- obsługa plików readme.md.

4.3.2 GIMP 2

Otwarto źródłowy program do edycji grafiki rastrowej. Zawiera wiele dodatków i efektów, które można zastosować do tworzenia ilustracji. Jego wyjątkowo ważną zaletą jest to, że jest darmowy i jednocześnie bardzo rozbudowany [20].

4.3.3 Inkscape

Oprogramowanie na licencji GPL wykorzystywane do tworzenia wektorowych obrazów. Opisywana aplikacja zawiera elastyczne narzędzia służące do rysowania, potężne funkcje tekstowe oraz dużą ilość obsługiwanych formatów plików [21].

4.3.4 FreeMind

Darmowe narzędzie do tworzenia i zarządzania mapami myśli, napisane w Javie i dostępne na licencji GPL. Dzięki niniejszemu narzędziu można tworzyć hierarchiczne mapy myśli. To oprogramowanie zawiera kilka charakterystycznych cech:

- możliwość dodawania ikon do elementów,
- wyszukiwanie w mapie myśli,
- eksport do wielu formatów,
- graficzne połączenia między węzłami [22].

5 Aplikacje demonstracyjne

W niniejszym rozdziale zostanie zaprezentowany prototyp przedstawiający wybrane i zaimplementowane elementy z koncepcji opisanej w rozdziale 3.

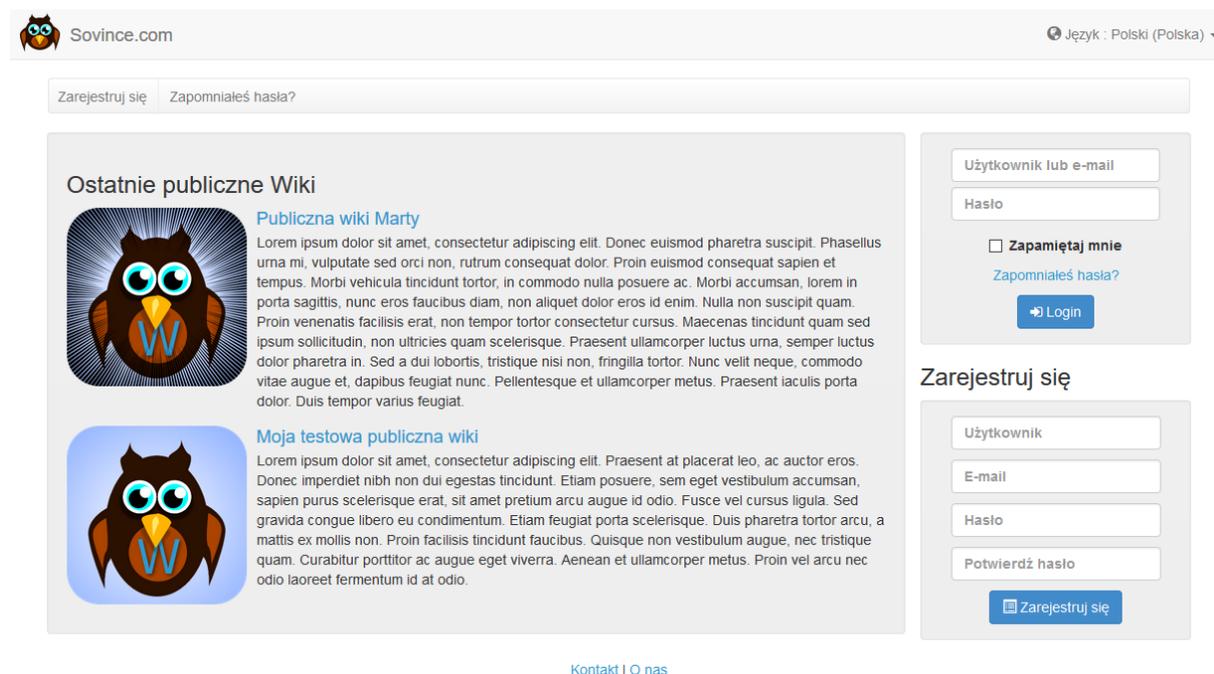
5.1 Aplikacja Web

Część Webowa została zaimplementowana w języku Python przy użyciu platformy Google App Engine. Jako magazyn danych w prototypie została użyta biblioteka NDB.

W zaprojektowanym na potrzeby aplikacji demonstracyjnej szablonie można wyróżnić następujące części:

- górny pasek narzędzi i powiadomień,
- główne menu,
- obszar, na którym wyświetlane są wszystkie treści,
- stopka.

Po otwarciu strony Sovince.com, widoczny jest podgląd ostatnich publicznych Wiki. Mamy możliwość zarejestrowania się za pomocą uproszczonego formularza albo zalogowania (Rysunek 7).



Rysunek 7 Logowanie do aplikacji

Źródło: Opracowanie własne

Dostęp do wszystkich zasobów wymaga posiadania konta, jeśli wybierzemy, którąś z widocznych Wiki pojawi się prośba o zalogowanie do systemu, za pomocą loginu lub adresu e-mail oraz naszego hasła. W tym miejscu, przedstawionym na Rysunek 8 mamy możliwość:

- odzyskania zapomnianego hasła,
- zarejestrowania się,
- użycia trwałego logowania.

Login

Zarejestruj się Zapomniałeś hasła?

Użytkownik lub e-mail* :

Hasło* :

Zapamiętać?

[Login](#)

[Zarejestruj się](#) [Zapomniałeś hasła?](#)

[Kontakt](#) | [O nas](#)

Rysunek 8 Prośba o zalogowanie do systemu

Źródło: Opracowanie własne

Na Rysunek 9 znajdują się informacje dostępne dla zalogowanego użytkownika. Na pasku narzędzi są do dyspozycji następujące elementy:

- powiadomienia,
- wiadomości,
- menu użytkownika,
- wybór języka,
- menu główne.

[Utwórz Wiki](#) [Moje Wiki](#) [Subskrypcje](#) [Ulubione](#) [Moje osiągnięcia](#)

Ostatnie publiczne Wiki

Publiczna wiki Marty

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec euismod pharetra suscipit. Phasellus urna mi, vulputate sed orci non, rutrum consequat dolor. Proin euismod consequat sapien et tempus. Morbi vehicula tincidunt tortor, in commodo nulla posuere ac. Morbi accumsan, lorem in porta sagittis, nunc eros faucibus diam, non aliquet dolor eros id enim. Nulla non suscipit quam. Proin venenatis facilisis erat, non tempor tortor consectetur cursus. Maecenas tincidunt quam sed ipsum sollicitudin, non ultricies quam scelerisque. Praesent ullamcorper luctus urna, semper luctus dolor pharetra in. Sed a dui lobortis, tristique nisi non, fringilla tortor. Nunc velit neque, commodo vitae augue et, dapibus feugiat nunc. Pellentesque et ullamcorper metus. Praesent iaculis porta dolor. Duis tempor varius feugiat.

Moja testowa publiczna Wiki

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent at placerat leo, ac auctor eros. Donec imperdiet nibh non dui egestas tincidunt. Etiam posuere, sem eget vestibulum accumsan, sapien purus scelerisque erat, sit amet pretium arcu augue id odio. Fusce vel cursus ligula. Sed gravida congue libero eu condimentum. Etiam feugiat porta scelerisque. Duis pharetra tortor arcu, a mattis ex mollis non. Proin facilisis tincidunt faucibus. Quisque non vestibulum augue, nec tristique quam. Curabitur porttitor ac augue eget viverra. Aenean et ullamcorper metus. Proin vel arcu nec odio laoreet fermentum id at odio.

[Kontakt](#) | [O nas](#)

Rysunek 9 Główny ekran widoczny po zalogowaniu

Źródło: Opracowanie własne

Powiadomienia zostały zaprojektowane tak, że każdy dodatek może tworzyć je w bardzo prosty sposób. Wystarczy stworzyć specjalny obiekt. W każdym powiadomieniu możemy wstawić:

- dowolną ikonkę,
- tekst powiadomienia,
- link, na który zostanie przekierowany użytkownik po kliknięciu.

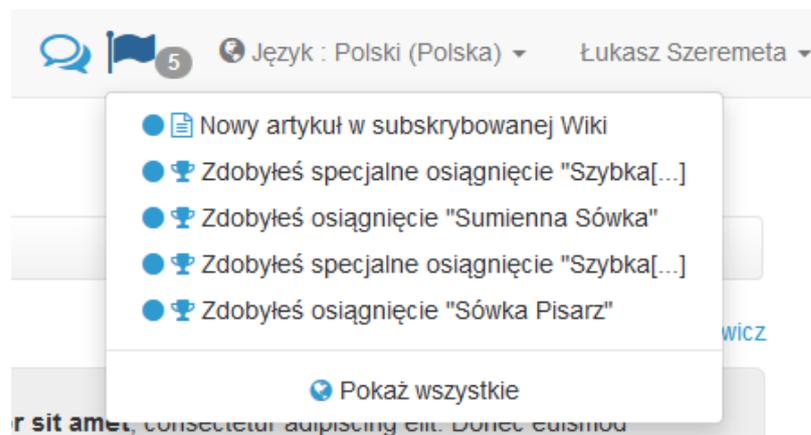
W działającym prototypie zaimplementowano powiadomienia pokazujące następujące informacje:

- zmiany w subskrybowanych Wiki,
- nowe osiągnięcia z dodatków (Rysunek 10).

W pasku powiadomień ikonki wiadomości i powiadomień zmieniają swój wygląd w zależności od tego czy zalogowana osoba ma nowe, nieodczytane informacje. Obok tej ikonki znajduje się małe kółko, w którym jest liczba pokazująca jak dużo jest nowych komunikatów lub powiadomień. W przypadku gdy będzie ich ponad pięćdziesiąt wyświetli się „50+”.

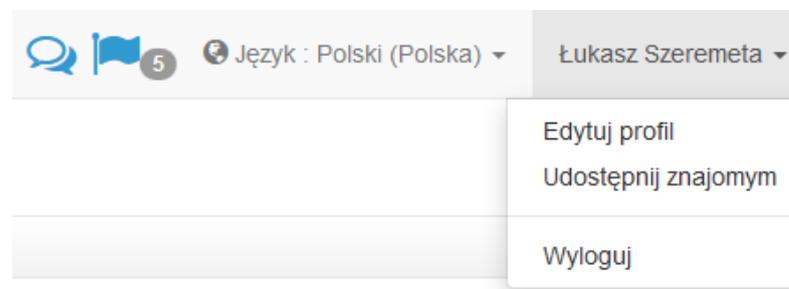
Wchodząc w menu użytkownika widoczne na Rysunek 11 zalogowana osoba ma do dyspozycji następujące pozycje:

- edycja profilu,
- wylogowanie się,
- udostępnianie informacji o Sovince.com. W tym miejscu osoba może pobrać specjalny link polecający, pozwalający podzielić się stroną Sovince.com ze znajomymi i przy okazji zdobyć odznaczenie.



Rysunek 10 Ostatnie powiadomienia

Źródło: Opracowanie własne



Rysunek 11 Menu użytkownika

Źródło: Opracowanie własne

Menu użytkownika jest w pełni dostosowywalne. Wszystkie dodatki mogą bez problemu dokładać do niego swoje własne pozycje. Pozycja tego elementu składa się z:

- ikonki,
- tekstu przycisku,
- strony na którą ma przekierowywać.

Tworząc Wiki istnieje możliwość wybrania czy będzie ona dostępna dla wszystkich zalogowanych użytkowników, czy tylko dla tych, którzy posiadają specjalny link. Polami obowiązkowymi do wypełnienia są nazwa i krótki opis, który będzie wyświetlany na wszystkich zbiorowych listach. Powinien on być ciekawie napisany, aby zachęcić użytkownika do zapoznania się z zawartością opisywanego elementu. Jak widać na Rysunek 12 znajduje się tu pole strony głównej z edytorem WYSIWYG. Pole to wyświetlane jest po wejściu w daną Wiki. W zależności od urządzenia i rozdzielczości może być wyświetlane w różnych miejscach. Jednak zawsze widoczne jest na tym samym ekranie co drzewo kategorii.

The screenshot shows the 'Utwórz Wiki' (Create Wiki) form on the Sovince.com website. The form is organized into several sections:

- Navigation:** A top bar with 'Utwórz Wiki', 'Moje Wiki', 'Subskrypcje', 'Ulubione', and 'Moje osiągnięcia' buttons.
- Form Fields:**
 - Nazwa*:** A text input field with the placeholder 'Wprowadź nazwę Wiki'.
 - Alias:** A text input field with the placeholder 'Wprowadź alias'.
 - Status:** A dropdown menu currently set to 'Publiczna'.
 - Opis*:** A large text area with the placeholder 'Wprowadź opis'.
- Strona główna (Main Page):** A section containing a rich text editor toolbar with various icons for text formatting, alignment, and insertion. Below the toolbar is a large, empty text area for the main page content.
- Buttons:** At the bottom of the form are two buttons: 'Anuluj' (Cancel) and 'Utwórz' (Create).

[Kontakt](#) | [O nas](#)

Rysunek 12 Zakładanie nowej Wiki

Źródło: Opracowanie własne

Po utworzeniu Wiki użytkownikowi pokaże się stosowny komunikat i zostanie przekierowany do nowo utworzonego projektu. Jeśli została oznaczona jako publiczna wszystkie inne osoby będą mogły znaleźć ją w wyszukiwarce, albo na listach takich jak ostatnie publiczne Wiki.

Gdyby autor chciał dopracować swój nowo stworzony projekt przed upublicznieniem go powinien utworzyć Wiki ze statusem „Prywatna”. Ewentualnie może dać dostęp do edycji i odczytu osobą, które mają pracować nad treścią oraz tym, których zadaniem jest weryfikacja wprowadzonej zawartości. W momencie, w którym autor stwierdzi, że Wiki jest już gotowa do publikacji wystarczy, że zmieni jej status na publiczną i wtedy każdy użytkownik systemu będzie mógł ją wyświetlić.

W przypadku gdy w systemie włączony jest dodatek osiągnięć po utworzeniu swojej pierwszej Wiki twórca zostanie powiadomiony o zdobyciu nowego trofeum. Po kliknięciu w powiadomienie zostanie przekierowany do zakładki „Moje osiągnięcia”.

Za pomocą edytora wizualnego w prosty sposób możemy wykonać zarówno proste jak i bardziej złożone formatowania. Między innymi jest to:

- pogrubienie,
- kursywa,
- podkreślenie,
- przekreślenie,
- zmiana koloru czcionki,
- wybranie wielkości czcionki,
- stworzenie indeksów górnych i dolnych,
- lista numerowana i nienumerowana,
- określenie położenia tekstu.

Na Rysunek 13 znajduje się lista wszystkich Wiki, których autorem jest zalogowany użytkownik. Widoczny jest tu obrazek, który możemy wstawić podczas edycji Wiki. Zostało to zaprojektowane w ten sposób, aby maksymalnie uprościć ilość informacji koniecznych do wprowadzenia podczas jej tworzenia. Przedstawiony na tym rysunku obrazek jest domyślną grafiką.



[Kontakt](#) | [O nas](#)

Rysunek 13 Moje Wiki

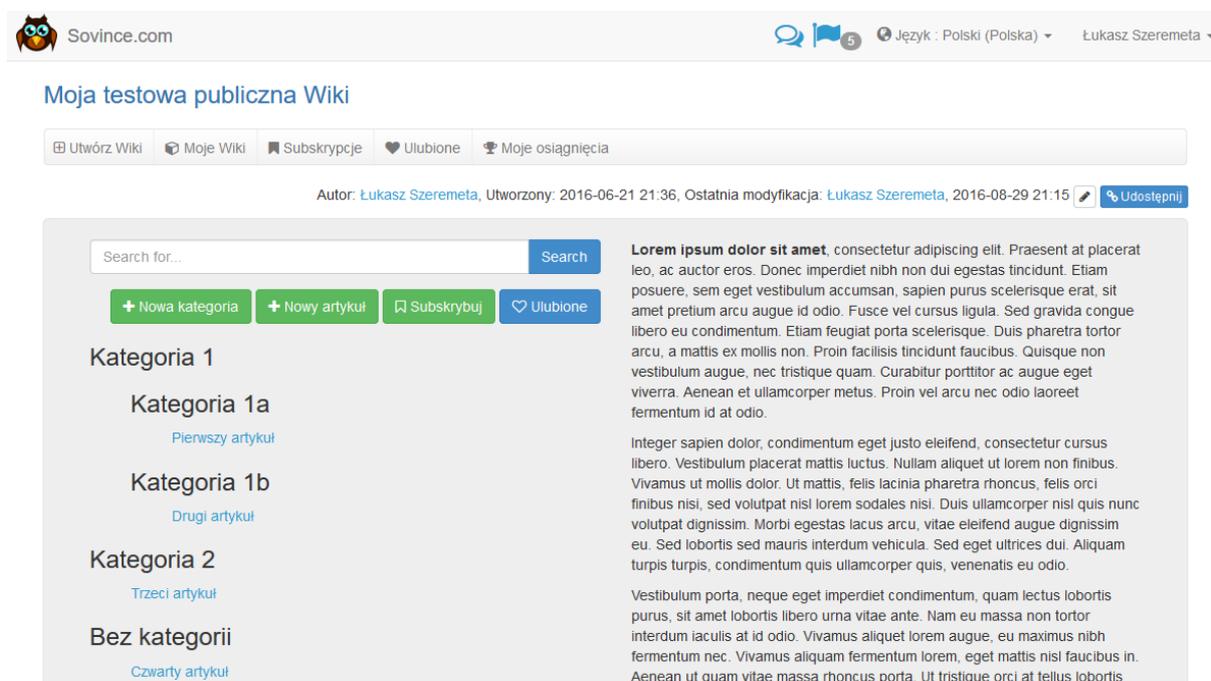
Źródło: Opracowanie własne

Jak widać na Rysunek 14 przedstawiona jest strona Wiki w sposób jaki widzi ją jej właściciel. Pozostali użytkownicy mogą nie mieć dostępnych wszystkich elementów. Niektóre z nich mogą być modyfikowane przez dodatki. Autor Wiki dysponuje następującymi elementami:

- wyszukiwarką,
- menu Wiki,
- drzewem kategorii zawierającym wszystkie artykuły,
- stroną główną,
- informacją o modyfikacjach,
- przyciskiem umożliwiającym udostępnienie wybranej Wiki,
- guzikiem pozwalającym na przejście w tryb edycji.

Poniższa lista przedstawia operacje jakie widzi wyłącznie twórca Wiki lub osoba, której udostępnił ją w trybie do zapisu:

- tworzenie nowych kategorii za pomocą przycisku „Nowa kategoria”,
- dodawanie artykułów przy użyciu guzika „Nowy artykuł”,
- edytowanie informacji o Wiki,
- udostępnienie jej do edycji lub wyświetlania (Rysunek 15).



Rysunek 14 Widok przykładowej Wiki wyświetlanej przez autora

Źródło: Opracowanie własne

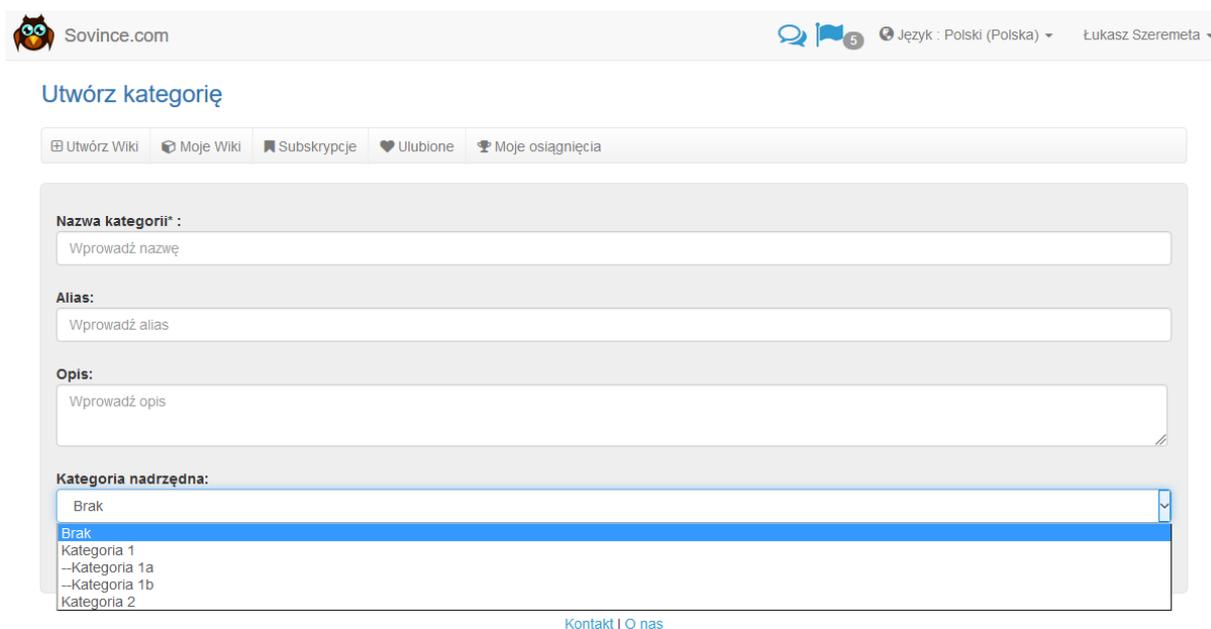
Drzewo kategorii widoczne Rysunek 14, pokazuje tylko niepuste elementy. Jeśli jakaś pozycja nie zawiera żadnych treści, to nie zostanie ona wyświetlona, ponieważ kluczową informacją są artykuły, a puste nagłówki znacznie zmniejszają czytelność. Wyświetlanie drzewa napisane jest rekurencyjnie, dzięki czemu może być ono dowolnej głębokości.



Rysunek 15 Udostępnianie Wiki

Źródło: Opracowanie własne

Jak widać na Rysunek 16 kategorie tworzymy w formie drzewa, dzięki czemu można w przystępny dla czytającego sposób podzielić wszystkie artykuły. Istnieje możliwość podania własnego aliasu, dzięki temu możemy tworzyć przyjazne dla użytkownika linki, jeśli nie wypełnimy tego pola zostanie on wygenerowany automatycznie.



Rysunek 16 Tworzenie nowej kategorii

Źródło: Opracowanie własne

Do tworzenia treści artykułów używany jest edytor wizualny (WYSIWYG). Pole treści zostanie zaindeksowane z pominięciem znaczników formatujących, aby umożliwić sprawne wyszukiwanie. Podobnie jak we wcześniejszym przypadku istnieje możliwość wprowadzenia własnego aliasu w celu zwiększenia czytelności adresu strony. Na Rysunek 17 pokazana jest specjalna, podstawowa kategoria - „Brak”, w takim przypadku artykuł będzie widoczny pod pozycją „Bez kategorii”.

[Kontakt](#) | [O nas](#)

Rysunek 17 Tworzenie artykułu

Źródło: Opracowanie własne

Na Rysunek 144 widać, że pod wyszukiwarką znajduje się przycisk „Subskrybuj” naciśnięcie go spowoduje dodanie danej Wiki do listy subskrybowanych, widocznej na Rysunek 18. Dzięki obserwowaniu danego projektu użytkownik będzie dostawał powiadomienia o wszelkich akcjach na nim wykonanych, na przykład o nowym artykule.

Dodanie Wiki do tej listy jest dobrym sposobem na zapamiętanie repozytorium ze statusem prywatnym innego użytkownika. Wcześniej musi zostać udostępniony link umożliwiający wyświetlanie bądź edycję projektu. Na tym rysunku widzimy pozycję „Prywatna Wiki Maćka” do której dostęp jest ograniczony tylko dla autora oraz posiadaczy specjalnego linku.

[Kontakt](#) | [O nas](#)

Rysunek 18 Subskrybowane Wiki

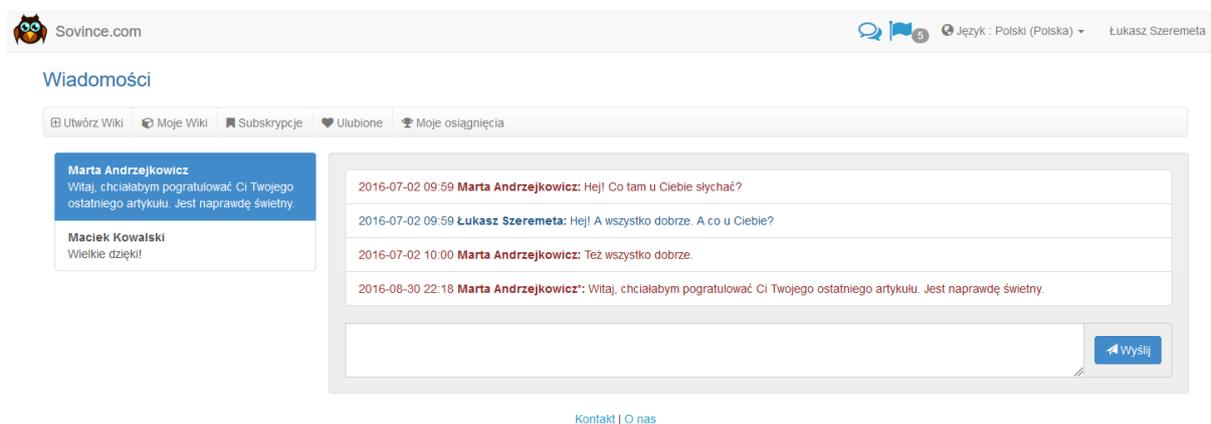
Źródło: Opracowanie własne

W prototypie został zaimplementowany prosty moduł wiadomości pozwalający użytkownikom na komunikację między sobą.

Na Rysunek 19 widzimy dwa główne elementy, zbiór osób, z którymi ostatnio prowadziliśmy konwersację oraz lista rozmów z konkretnym człowiekiem.

Lista osób zawiera informacje z kim prowadzimy dyskusję oraz pokazuje nam ostatnią wiadomość w dyskusji.

Po otwarciu okna z dialogiem z danym użytkownikiem widzimy nieprzeczytane wiadomości, które oznaczane są za pomocą gwiazdki. W rozmowie nadawca ma kolor niebieski, a adresat czerwony, aby w prosty sposób można było rozróżnić kto co napisał. Na tej liście znajduje się również informacja, o której godzinie została wysłana wiadomość. W momencie otwarcia konwersacji z danym użytkownikiem komunikaty nieprzeczytane zostają oznaczone jako przeczytane i ikonka w menu powiadomień się zmienia. W tym oknie mamy możliwość wprowadzania tekstu, który zostanie przekazany do wybranej osoby.



Rysunek 19 Zakładka wiadomości

Źródło: Opracowanie własne

W prototypie istnieje możliwość oglądania profilu innego użytkownika. Znajdują się tam informacje o jego ostatnich publicznych Wiki oraz publicznych artykułach. W profilu danej osoby mogą znajdować się dodatkowe informacje wstawiane przez dodatki. Na Rysunek 20 widać trofea zwykłe i specjalne wstawiane przez wtyczkę osiągnięcia. Na stronie profilu znajdują się przyciski umożliwiające wysłanie wiadomości do danego użytkownika oraz subskrypcje wybranej osoby. Naciśnięcie drugiego z tych przycisków spowoduje, że po ważniejszych akcjach jak na przykład dodanie nowego artykułu człowiek subskrybujący będzie otrzymywał powiadomienie o danym zdarzeniu.

Ponadto na profilu gracza możemy znaleźć podstawowe informacje o nim, między innymi możemy dowiedzieć się:

- od kiedy jest członkiem Sovince.com,
- jaki ma login,
- jaką ustawił widoczną nazwę,
- o kraju jego pochodzenia,
- jaki ma adres e-mailowy.

Informacje profilowe - Marta Andrzejkowicz

Utwórz Wiki Meje Wiki Subskrypcje Ulubione Meje osiągnięcia



Użytkownik: **martaa**
 Widoczna nazwa: **Marta Andrzejkowicz**
 E-mail: **kontakt@lukasz.pl**
 Kraj: **Polska**
 W Sovince.com od: **2016-07-01**

Ostatnie publiczne Wiki

[Publiczna wiki Marty](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec euismod pharetra suscipit. Phasellus urna mi, vulputate sed orci non, rutrum consequat dolor. Proin euismod consequat sapien et tempus. Morbi vehicula tincidunt tortor, in commodo nulla posuere ac. Morbi accumsan, lorem in porta sagittis, nunc eros faucibus diam, non aliquet dolor eros id enim. Nulla non suscipit quam. Proin venenatis facilisis erat, non tempor tortor consectetur cursus. Maecenas tincidunt quam sed ipsum sollicitudin, non ultricies quam scelerisque. Praesent ullamcorper luctus urna, semper luctus dolor pharetra in. Sed a dui lobortis, tristique nisi non, fringilla tortor. Nunc velit neque, commodo vitae augue et, dapibus feugiat nunc. Pellentesque et ullamcorper metus. Praesent iaculis porta dolor. Duis tempor varius feugiat.

Ostatnie publiczne artykuły

[Artykuł w kategorii popularne](#)
[Drugi artykuł Marty](#)
[Pierwszy artykuł Marty](#)

Trophies



Special Trophies



[Kontakt](#) | [O nas](#)

Rysunek 20 Informacje o wybranym użytkowniku

Źródło: Opracowanie własne

Każdy użytkownik może edytować swoje dane takie jak:

- zdjęcie,
- nazwę,
- imię,
- nazwisko,
- hasło,
- kraj,
- strefa czasowa.

Poniżej znajduje się opis implementacji elementów, które mogą zostać użyte podczas tworzenia dodatków. Podczas programowania zostały stworzone dwa podstawowe modele „*AuthorAndTime*” oraz „*StatsLog*”. Pierwszy z nich zawiera informacje takie jak:

- data utworzenia,
- data modyfikacji,
- autor,
- informacja czy jest usunięte.

Model ten używany jest między innymi w:

- Wikich,
- kategoriach,
- artykułach,

- zdjęciach profilowych
- awatarach Wiki,
- wersjonowaniu artykułów.

Drugi z podstawowych modeli służy do zbierania wszelkich statystyk o sposobie korzystania z systemu przez użytkowników, zawiera on datę zdarzenia oraz informację o osobie, która wykonała daną akcję.

Powiadomienia zostały zaimplementowane za pomocą specjalnego modelu „*Notification*”. Zawiera on kilka pól, które może określić twórca powiadomienia:

- treść,
- z powyższego pola wyliczana jest zawartość skróconą do 38 znaków,
- link do elementu, którego dotyczy,
- odbiorca,
- ikona.

W aplikacji do obsługi wszelkich żądań wykorzystywany jest „*DefaultHandler*”, w dodatkach należy wykorzystywać klasę, która dziedziczy z opisywanej, a nazywa się „*PluginHandler*”. Zawiera ona wszelkie przydatne informacje, między innymi są to:

- informacje o obecnie zalogowanym użytkowniku,
- używane menu,
- liczby nieprzeczytanych wiadomości,
- ostatnie powiadomienia,
- obecnie wybrany język,
- dane o tym czy żądanie zostało wysłane z urządzenia mobilnego,
- informacje o widocznej nazwie obecnie zalogowanej osoby,

Ponadto użytkownik może wykonać na opisywanym kontrolerze takie operacje jak na przykład dodanie komunikatu za pomocą funkcji „*add_message*”

Stworzony obiekt „*MenuItem*” można dodać do dowolnego menu, jeżeli ktoś chce korzystać z tej funkcjonalności w dodatkach powinien użyć specjalnej klasy „*PluginMenuItem*”. Obie posiadają następujące pola, które mogą być uzupełnione przez programistę:

- tekst,
- adres, na który wskazuje,
- podmenu,
- ikona.

Aby dodać guzik do listy przycisków, na przykład takiej jak znajduje się na stronie głównej Wiki należy utworzyć obiekt klasy „*Button*”. W nim można wypełnić następujące pola:

- adres, na który wskazuje,
- tekst,
- ikonę,
- klasę CSS.

Gdyby twórca dodatku chciał stworzyć dodatkowe wywołanie API może być konieczne napisanie własnej klasy dziedziczącej z „*MyEncoder*”, pozwala on zamieniać na postać JSONa niestandardowe obiekty, takie jak:

- *ArticleCategoryTree* – drzewo kategorii,
- *CategoryREST* – kategorie na potrzeby RESTa,

- *ArticleREST* – artykuły.

Podczas tworzenia własnych „handlerów” należy pamiętać o tym, że wszystkie obiekty zwrócone przez bazę danych należy przepisać do własnych klas. Obiekty zwracane z bazy danych zawierają bardzo dużo zbędnych typów i niepotrzebnych danych. Wartości takich pól jak na przykład klucz nie zostaną pomyślnie zrzutowane do JSONa, za pomocą dostępnej funkcji „*json.dumps*”.

Aby móc dodać jakiś element na już istniejącej stronie należy skorzystać z klasy „*ElementHTML*”, w której możemy określić kod HTML do wyświetlenia oraz klasę CSS. W przypadku dodatków można ją wykorzystać poprzez dziedziczenie z *IUserProfileHTML* i implementację funkcji *user_profile_html_editor*, która powinna zwracać tablicę elementów opisywanej klasy.

5.2 System dodatków

Wtyczki zostały zaprojektowane w ten sposób, aby skutecznie realizować funkcjonalność związaną z grywalizacją systemu. Oczywiście za pomocą dodatków można dowolnie rozszerzyć działanie rozwiązania.

Koncepcja przewiduje stworzenie lekkiego rdzenia, a wszystkie opcjonalne elementy powinny być realizowane za pomocą pluginów. Dzięki temu powstanie system, który można dowolnie dopasować do potrzeb administratora.

Struktura podstawowego dodatku wygląda następująco:

- *nazwa_dodatku* – główny katalog
 - *user* – folder
 - *images* – katalog ze zdjęciami
 - *templates* – szablony definiujące sposób wyświetlania informacji o dodatku
 - *user_main.py* – główny plik
 - *__init__.py* – Pythonowy plik inicjujący
 - *config.json* – plik konfiguracyjny
 - *info.json* – informacje o pluginie

Każdy dodatek powinien zawierać plik „*config.json*” (Kod aplikacji 1), informujący o tym, czy dany dodatek jest włączony.

```
{
  "enabled": 1
}
```

Kod aplikacji 1 config.json

Programista może do dodatku załączyć następujące informacje. Należy umieścić je w pliku „*info.json*”, przykład takiego pliku można znaleźć w sekcji Kod aplikacji 2.

```

{
  "name": "Favourite",
  "version": "0.1 beta",
  "version_code": 1,
  "author": "Łukasz Szeremeta",
  "author_website": null,
  "create_date": "2016-03-23",
  "modify_date": "2016-03-30",
  "plugin_url": null,
  "description": "Plugin pozwalający dodawać Wiki do ulubionych, a
później przeglądać",
  "prefix": "favourite"
}

```

Kod aplikacji 2 info.json

W Pythonie nie ma pojęcia interfejsów, dlatego implementowanie interfejsów polega na dziedziczeniu z klasy i przeciążeniu danej metody. W poniższym tekście może być używane pojęcie implementowania interfejsów należy je rozumieć jako skrót myślowy.

Głównym plikiem, który jest wykonywany w dodatku jest „user_main.py”. W nim należy zdefiniować klasę „Plugin” dziedziczącą z „IPluginUser” oraz należy przeciążyć procedurę „start” (Kod aplikacji 3).

```

class Plugin(IPluginUser):
    def start(self, *args, **kwargs):
        return None

```

Kod aplikacji 3 Główna klasa dodatku

W systemie dodatków możemy utworzyć dowolny model dziedziczący ze standardowego „ndb.Model”. Modele zaleca się umieszczać w katalogu „user” w pliku „models.py” (Kod aplikacji 4).

```

class FavouriteWiki(ndb.Model):
    create_date = ndb.DateTimeProperty(auto_now_add=True)
    user = ndb.KeyProperty(kind='User', required=True)
    wiki = ndb.KeyProperty(kind='Wiki', required=True)
    wiki_alias = ndb.StringProperty()

    @classmethod
    def get_by_user(cls, user):
        return cls.query(cls.user == user).fetch()

```

Kod aplikacji 4 Model ulubionych Wiki

Klasy służące do obsługi żądań wyświetlenia strony powinny dziedziczyć z „PluginHandler” (Kod aplikacji 5).

```

class FavouriteHomeRequestHandler(PluginHandler):
    def get(self):
        favourites = FavouriteWiki.get_by_user(self.user_key)
        params = {'favourites': favourites}
        return self.render_template(TEMPLATES_PATH + 'home.html',
**params)

```

Kod aplikacji 5 Dziedziczenie z PluginHandler

Aby powiązać adres url z utworzonym wcześniej „handlerem” należy zaimplementować interfejs „IPluginRoutes” (Kod aplikacji 6).

```

def get_routes(self):
    return [
        PluginRoute('/favourite-wiki/<alias>',
FavouriteClickHandler, "favourite_wiki"),
        PluginRoute('/', FavouriteHomeRequestHandler, "home")
    ]

```

Kod aplikacji 6 Definiowanie adresów url

Dodawanie przycisków do głównego menu odbywa się za pomocą klasy „IPluginLoggedTopMenu” oraz metody „get_logged_top_menu_item” (Kod aplikacji 7). W podobny sposób dodaje się elementy do menu Wiki, należy użyć „IPluginWikiButtons” oraz metody „wiki_buttons”

```

def get_logged_top_menu_item(self):
    return [
        PluginMenuItem(url=self.plugin_uri_for('home'),
name_='Favourite'), faicon='fa-heart')
    ]

```

Kod aplikacji 7 Dodawanie przycisków do górnego menu dla zalogowanego użytkownika

W prototypie zaimplementowana jest obsługa następujących akcji:

- Klasa *IPluginAfterOpenWiki* posiada metodę *after_open_wiki* wykonywana jest po otwarciu Wiki.
- Klasa *IPluginAfterOpenArticle* posiada metodę *after_open_article* wykonywana jest po otwarciu artykułu.
- Klasa *IPluginAfterOpenProfile* posiada metodę *after_open_profile* wykonywana jest po otwarciu profilu użytkownika.
- Klasa *IPluginAfterCreateWiki* posiada metodę *after_create_wiki* wykonywana jest po stworzeniu Wiki.
- Klasa *IPluginAfterCreateArticle* posiada metodę *after_create_article* wykonywana jest po utworzeniu artykułu.
- Klasa *IPluginAfterSaveProfile* posiada metodę *after_save_profile* wykonywana jest po zapisaniu profilu.
- Klasa *IPluginAfterSaveWiki* posiada metodę *after_save_wiki* wykonywana jest po zapisaniu Wiki.

- Klasa *IPluginAfterOpenShareSovince* posiada metodę *after_open_share_sovince* wykonywana jest po otwarciu linku służącego do informowania znajomych o systemie Sovince.com.
- Klasa *IPluginAfterRegisterFromShareSovince* posiada metodę *after_register_from_share_sovince* wykonywana jest po zarejestrowaniu użytkownika za pomocą poleconego linka.

5.2.1 Osiągnięcia

Strona moje osiągnięcia prezentuje odznaczenia zdobyte oraz te możliwe do wygrania przez użytkownika. W tym dodatku zostało zastosowanych kilka elementów grywalizacji, jednym z nich są ładne i ciekawe trofea. W prototypie gracz ma możliwość zasłużenia na poniższe medale.

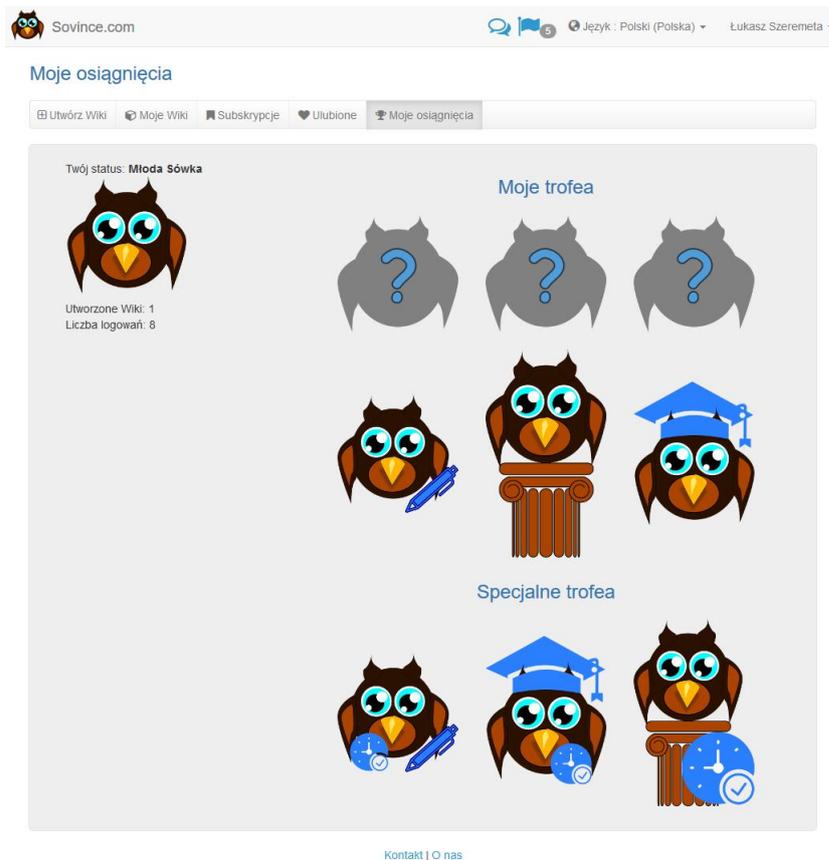
- Sówka Podglądacz – użytkownik może otrzymać za przeglądanie profili innych osób.
- Sówka Plotkara – gracz dostaje je jeśli odpowiednią liczbę razy poleci aplikację Sovince.com swoim znajomym.
- Sówka Czytelnik – osoba może zdobyć to odznaczenie za czytanie artykułów.
- Sówka Pisarz – ten medal przyznawany jest za pisanie artykułów.
- Sówka Twórca – jeśli ktoś tworzy nowe Wiki może liczyć na to osiągnięcie.
- Sumienna Sówka – gracz, który uzupełni wszystkie dane na swoim profilu zostanie nagrodzony tym orderem.

W tej wtyczce stosowany jest również „poziom bossa” dla wybranych z powyższych odznak. Warunkiem otrzymania specjalnego orderu jest bycie pierwszym użytkownikiem systemu, który otrzyma jeden z powyższych medali. W prototypie znajdują się następujące wyjątkowe trofea:

- Szybka Sówka Czytelnik
- Szybka Sówka Pisarza
- Szybka Sumienna Sówka
- Szybka Sówka Twórca
- Szybka Sówka Plotkara

Kolejnym elementem grywalizacji jest system punktowy w ramach którego przydzielane są rangi.

Wyżej opisany dodatek znajduje się na Rysunek 21. Na Rysunek 20 widać trofea zdobyte przez użytkownika, którego profil jest przeglądany.



Rysunek 21 Dodatek moje osiągnięcia

Źródło: Opracowanie własne

Do realizacji tego dodatku zostały użyte funkcje z niemal wszystkich interfejsów. Ponadto konieczne było stworzenie własnych kontrolerów, modeli, dołożenie pozycji do menu oraz stworzenie własnych powiadomień o otrzymaniu odznaki.

5.2.2 Ulubione

Jest to prosty dodatek, który w przeciwieństwie do subskrypcji nie generuje żadnych powiadomień odnośnie oznaczonych Wiki. Może służyć do zapisania prywatnego projektu innego użytkownika.

Podczas jego implementacji został stworzony jeden model i jeden kontroler. Dodatkowo do menu został dodany przycisk „Ulubione”.

5.3 Aplikacja na Androida

Prototypowa aplikacja mobilna stworzona na platformę Android realizuje funkcję przeglądarki treści. Została zaprojektowana dla użytkowników, którzy chcą przeglądać materiały i nie interesuje ich dodawanie nowej zawartości.

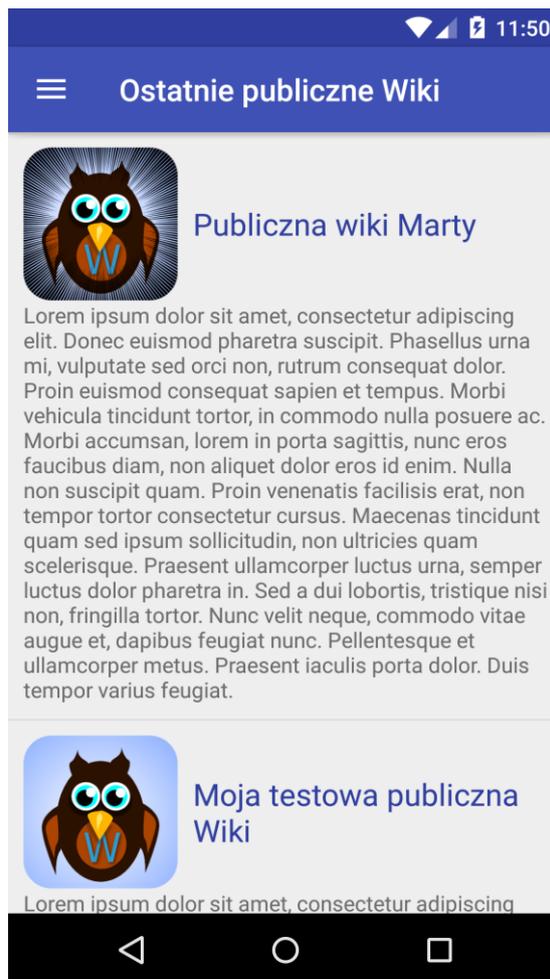
Zaraz po uruchomieniu programu pojawia się ekran logowania z prośbą o podanie nazwy użytkownika lub loginu oraz hasła (Rysunek 22).

Po podaniu prawidłowych danych pokazuje nam się ekran „Ostatnie publiczne Wiki” (Rysunek 23).



Rysunek 22 Ekran logowania

Źródło: Opracowanie własne

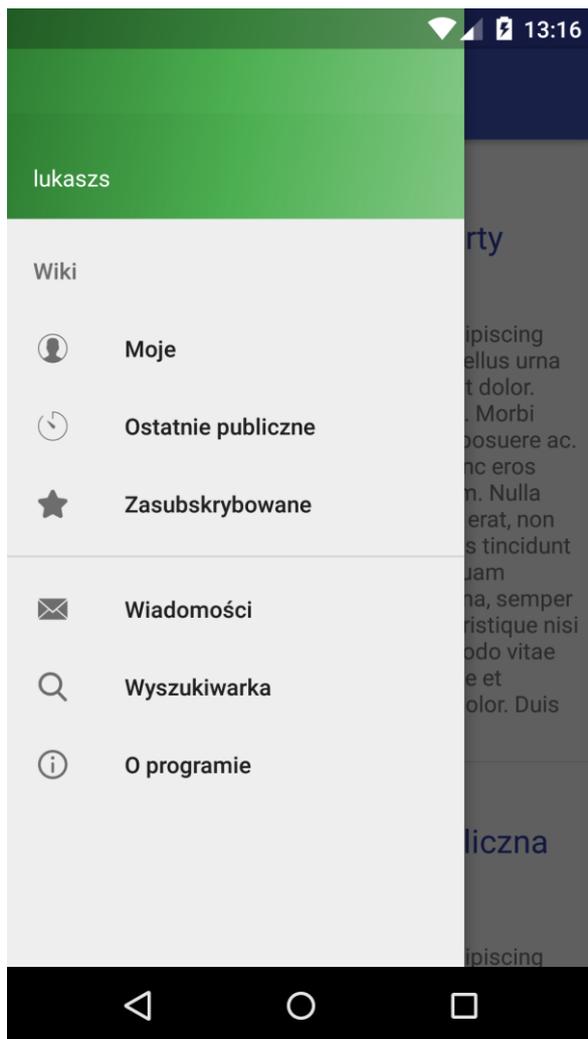


Rysunek 23 Ostatnie publiczne Wiki

Źródło: Opracowanie własne

Użytkownicy mogą nawigować po systemie za pomocą wysuwanego z lewego boku menu. Znajdują się tam następujące pozycje:

- Wiki
 - Moje
 - Ostatnie publiczne
 - Zasubskrybowane
- Wiadomości
- Wyszukiwarka
- O programie (Rysunek 24)



Rysunek 24 Menu

Źródło: Opracowanie własne



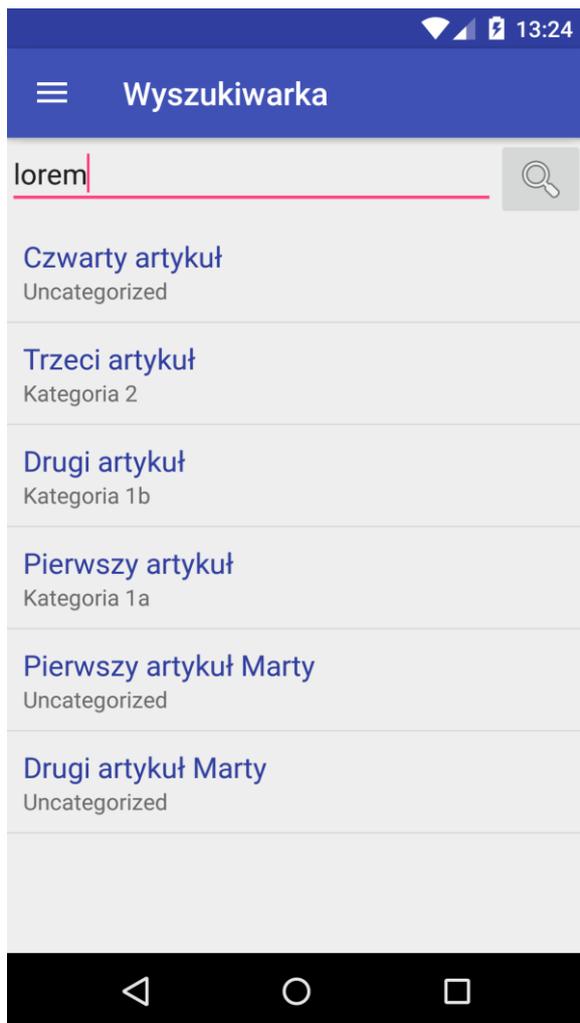
Rysunek 25 Wiadomości

Źródło: Opracowanie własne

Użytkownicy mają możliwość wysyłania i odbierania wiadomości. Podobnie jak w aplikacji Webowej najpierw pojawia się lista wszystkich konwersacji, a po wybraniu konkretnej rozmowy pojawia się okno widoczne na Rysunek 25. Jak widać znajduje się tu informacja o nadawcy, treść oraz data wysłania komunikatu. Mechanizm oznaczania wiadomości jako przeczytane działa dokładnie tak samo jak w przypadku wersji webowej.

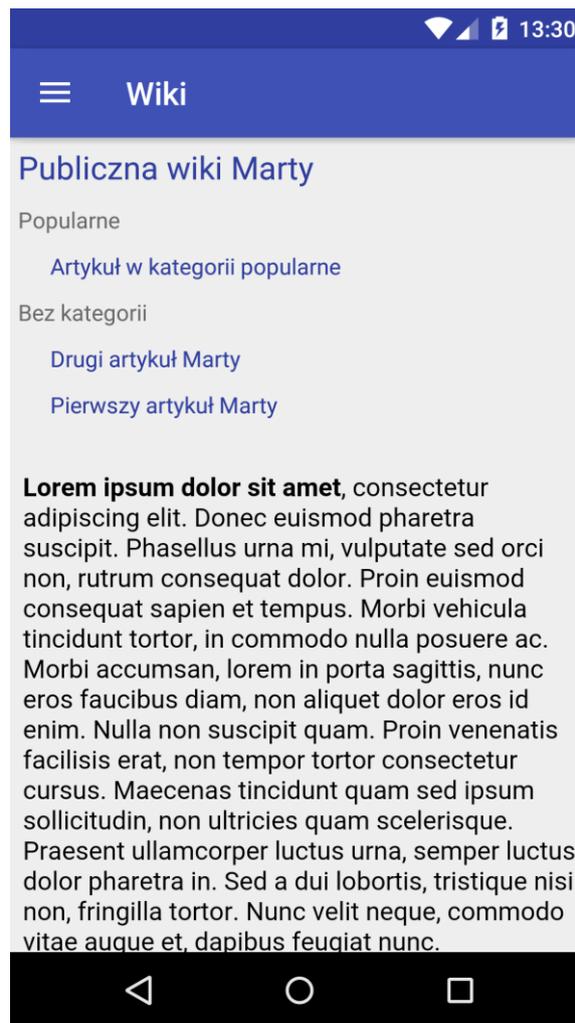
Program na smartfony umożliwia wyszukiwanie informacji w nazwach artykułów oraz ich treściach. Przeszukiwane są publiczne Wiki, te których zalogowana osoba jest autorem oraz subskrybowane Wiki prywatne. W rezultacie wyszukiwania mamy widoczną nazwę oraz kategorię, w której znajduje się dana treść (Rysunek 26).

Po wybraniu Wiki widoczny jest ekran przedstawiony na Rysunek 27. Znajduje się na nim drzewo kategorii podobnie jak w wersji webowej budowane jest rekurencyjnie, więc nie ma limitu głębokości drzewa. Widzimy tu nagłówki, które zawierają przynajmniej jeden artykuł. Strona główna wygląda dokładnie tak samo jak w wersji webowej, ewentualnie może być trochę inaczej sformatowana.



Rysunek 26 Wyszukiwarka

Źródło: Opracowanie własne



Rysunek 27 Widok Wiki

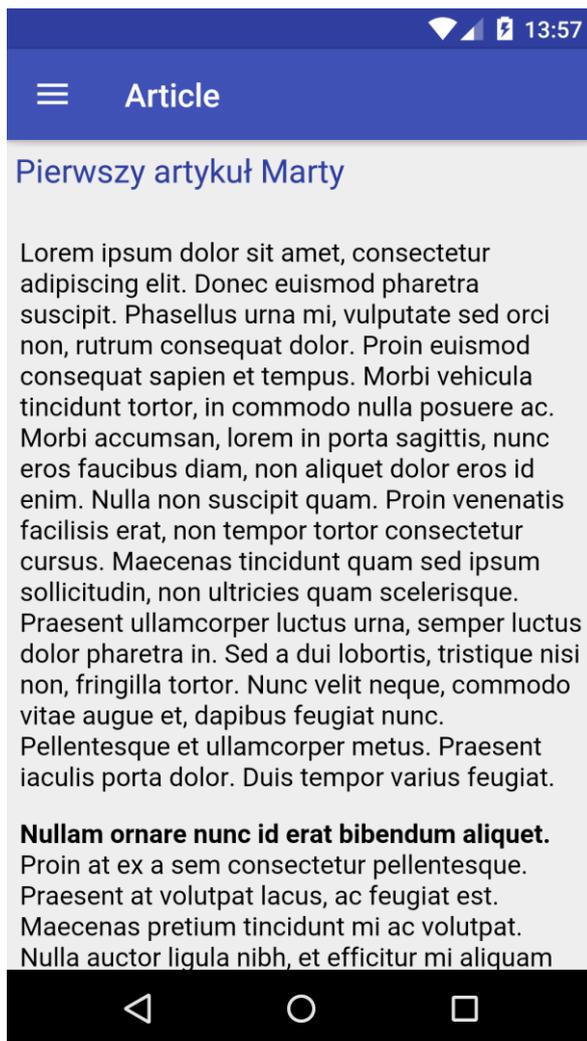
Źródło: Opracowanie własne

Po wybraniu konkretnego artykułu wyświetli się fragment zawierający tytuł oraz treść, która jest wyświetlana tak jak sformatował ją autor. Na Rysunek 28 widać pogrubienie tekstu zastosowane przez twórcę treści.

W aplikacji znajduje się prosta strona informująca, kto jest twórcą programu, jest również napisane, że został on przygotowany jako prototyp części mobilnej pracy magisterskiej.

Cały system został napisany zgodnie z najnowszymi zaleceniami firmy Google. Wszystkie widoki zaprojektowane są na fragmentach i korzystają z wywołań zwrotnych, dzięki czemu dostosowanie aplikacji do innych urządzeń na przykład z większym ekranem będzie stosunkowo proste.

Wszystko działa na dwóch wątkach, jeden służy do wyświetlania i obsługi zdarzeń. Drugi natomiast odpowiada za wysyłanie zapytań do chmury i przetwarzanie odpowiedzi.



Rysunek 28 Artykuł

Źródło: Opracowanie własne

6 Podsumowanie

Niniejszy rozdział zawiera wszystkie wnioski, do których doszedł autor podczas opracowywania koncepcji oraz implementacji prototypu.

6.1 Wady i zalety

Pisząc prototyp aplikacji autor pracy zauważył zalety i wady koncepcji oraz zastosowanych w niej rozwiązań.

Dzięki wykorzystaniu systemu dodatków istnieje bardzo duża elastyczność w dokładaniu nowych elementów grywalizacji jak i rozszerzaniu funkcjonalności systemu.

Dużą zaletą jest wykorzystanie chmury Google App Engine, która jest darmowa o ile aplikacja nie przekroczy określonych limitów. Prototyp zmieścił się w niepłatnych ograniczeniach. Gdyby aplikacja odniosła sukces nie było by technicznych problemów, ponieważ chmura zapewniłaby potrzebne zasoby.

W przypadku aplikacji mobilnej wadą okazał się system dodatków, ponieważ na platformę Android ciężko jest przygotować taki system, w którym twórca mógłby w łatwy sposób stworzyć także mobilną część swojej wtyczki. W Programie można by obsługiwać kilka popularnych pluginów, pobierać informację, czy są obecnie zainstalowane w odpowiedniej wersji oraz włączone i wtedy pokazywać dane dotyczące danego elementu.

Wśród osób, które testowały prototyp została przeprowadzona krótka ankieta, mająca na celu zbadanie, czy zastosowanie grywalizacji w systemach Wiki jest dobrym pomysłem. Badani niemal jednomyślnie powiedzieli, że zastosowanie odznak i systemu punktowego wprowadza dodatkową motywację zarówno w tworzeniu jak i w przeglądaniu zasobów.

Po przetestowaniu systemu przez kilka osób udało się odkryć pewne błędy, które zostały naprawione. Okazało się jednak, że potrzeba jeszcze dużo pracy, aby prototyp stał się produktem gotowym do wypuszczenia na rynek.

6.2 Możliwości rozwoju aplikacji

Po stworzeniu prototypu i przetestowaniu rozwiązania przez autora oraz kilka osób powstały następujące propozycje rozwoju systemu:

- Stworzenie wydarzeń i wyzwań pozwoliłoby jeszcze lepiej wciągnąć graczy w stworzony świat gry.
- Wymyślenie historii na której bazują wszystkie odznaczenia. Obecnie istnieje tylko wspólny motyw sowy, gra powinna opowiadać ciekawą, wciągającą historię.
- Zaprojektowanie kolejnych odznaczeń oraz przeprojektowanie istniejących przez profesjonalnego grafika. Zdecydowanie należałoby wymyśleć więcej „poziomów bossa”, aby gracz nie spotkał sytuacji, że nie ma żadnego celu.
- System punktowy wymaga dopracowania. Niestety nie jest możliwe przewidzenie, rzeczywistej częstotliwości tworzenia Wiki, artykułów oraz przeglądania zasobów. Najlepszym sposobem na zebranie danych do jego dopracowania było by rozpoczęcie testów na większą skalę.
- Po wielu testach w ostatecznej fazie testowania prototypu wyszło, że dużo lepszym rozwiązaniem, w którym Wiki publiczne dostępne są również dla niezalogowanych użytkowników.
- Należałoby stworzyć system płynnego wejścia. Polegałby on na tworzeniu użytkownikowi nienazwanego konta działającego bez logowania. Informacje o jego akcjach byłyby zapisywane, a w momencie zdobycia jakiejś odznaki prośba o podanie koniecznych danych, by otrzymać dany medal.

- Ważnym elementem było by dobre zaimplementowanie mechanizmów grywalizacji w aplikacjach mobilnych.

6.3 Zakończenie

Mogłoby się wydawać, że dzięki długiej historii obecne systemy Wiki są już dopracowane pod każdym względem. Wciąż jednak można je udoskonalać pod wieloma względami. Między innymi można wprowadzać w nich rozwiązania takie jak opisane w niniejszej pracy, czyli zwiększające zainteresowanie oraz wciągające użytkowników w proces tworzenie i przeglądania zasobów. Według autora pracy grywalizacja bardzo dobrze sprawdza się na tym polu.

7 Bibliografia

- [1] [Online]. Available: <http://s23.org/wikistats/>. [Data uzyskania dostępu: 06 08 2016].
- [2] [Online]. Available: <http://www.wikimatrix.org/>. [Data uzyskania dostępu: 08 08 2016].
- [3] P. Tkaczyk, Grywalizacja. Jak zastosować reguły gier w działaniach marketingowych, Helion, 2012.
- [4] [Online]. Available: https://www.mediawiki.org/wiki/Special:MyLanguage/Manual:Installation_requirements. [Data uzyskania dostępu: 06 08 2016].
- [5] [Online]. Available: http://spolecznosc.wikia.com/wiki/Centrum_Spo%C5%82ecznie%C5%9Bci:Wikianie. [Data uzyskania dostępu: 08 08 2016].
- [6] [Online]. Available: <https://cloud.google.com/appengine/>. [Data uzyskania dostępu: 10 08 2016].
- [7] [Online]. Available: <https://www.python.org/>. [Data uzyskania dostępu: 10 08 2016].
- [8] [Online]. Available: <https://www.w3.org/MarkUp/>. [Data uzyskania dostępu: 11 08 2016].
- [9] [Online]. Available: <https://pl.wikipedia.org/wiki/JavaScript>. [Data uzyskania dostępu: 11 08 2016].
- [10] [Online]. Available: <https://jquery.com/>. [Data uzyskania dostępu: 11 08 2016].
- [11] [Online]. Available: <https://www.w3.org/Style/CSS/>. [Data uzyskania dostępu: 11 08 2016].
- [12] [Online]. Available: <http://getbootstrap.com/>. [Data uzyskania dostępu: 11 08 2016].
- [13] fontawesome. [Online]. Available: <http://fontawesome.io/>. [Data uzyskania dostępu: 01 09 2016].
- [14] [Online]. Available: <http://ckeditor.com/>. [Data uzyskania dostępu: 11 08 2016].
- [15] [Online]. Available: <https://github.com/coto/gae-boilerplate>. [Data uzyskania dostępu: 01 09 2016].
- [16] [Online]. Available: <https://www.jetbrains.com/pycharm/>. [Data uzyskania dostępu: 11 08 2016].
- [17] [Online]. Available: <https://www.java.com>. [Data uzyskania dostępu: 11 08 2016].
- [18] [Online]. Available: <https://developer.android.com>. [Data uzyskania dostępu: 20 08 2016].
- [19] [Online]. Available: <http://square.github.io>. [Data uzyskania dostępu: 20 08 2016].
- [20] [Online]. Available: <https://www.gimp.org/>. [Data uzyskania dostępu: 20 08 2016].
- [21] [Online]. Available: <https://inkscape.org>. [Data uzyskania dostępu: 20 08 2016].

[22] [Online]. Available: http://freemind.sourceforge.net/wiki/index.php/Main_Page. [Data uzyskania dostępu: 01 09 2016].

8 Spis rysunków

Rysunek 1 Strona główna projektu MediaWiki.....	8
Rysunek 2 Przykładowa strona edycji artykułu w WikiDot.....	9
Rysunek 3 Zakładanie nowej strony w WikiDot.....	10
Rysunek 4 Strona główna Wikia	11
Rysunek 5 Ranking WAM na stronie Wikia	12
Rysunek 6 Panel administratora w Wikispaces	13
Rysunek 7 Logowanie do aplikacji	26
Rysunek 8 Prośba o zalogowanie do systemu	27
Rysunek 9 Główny ekran widoczny po zalogowaniu.....	27
Rysunek 10 Ostatnie powiadomienia	28
Rysunek 11 Menu użytkownika	28
Rysunek 12 Zakładanie nowej Wiki.....	29
Rysunek 13 Moje Wiki.....	30
Rysunek 14 Widok przykładowej Wiki wyświetlanej przez autora	31
Rysunek 15 Udostępnianie Wiki	32
Rysunek 16 Tworzenie nowej kategorii	32
Rysunek 17 Tworzenie artykułu.....	33
Rysunek 18 Subskrybowane Wiki.....	33
Rysunek 19 Zakładka wiadomości	34
Rysunek 20 Informacje o wybranym użytkowniku.....	35
Rysunek 21 Dodatek moje osiągnięcia.....	41
Rysunek 22 Ekran logowania	42
Rysunek 23 Ostatnie publiczne Wiki	42
Rysunek 24 Menu.....	43
Rysunek 25 Wiadomości	43
Rysunek 26 Wyszukiwarka	44
Rysunek 27 Widok Wiki	44
Rysunek 28 Artykuł.....	45

9 Kody źródłowe aplikacji

Kod aplikacji 1 config.json.....	37
Kod aplikacji 2 info.json	38
Kod aplikacji 3 Główna klasa dodatku.....	38
Kod aplikacji 4 Model ulubionych Wiki	38
Kod aplikacji 5 Dziedziczenie z PluginHandler	39
Kod aplikacji 6 Definiowanie adersów url	39
Kod aplikacji 7 Dodawanie przycisków do górnego menu dla zalogowanego użytkownika...	39