



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

Praca dyplomowa - inżynierska

Aplikacja webowa do udostępniania niezwykłych miejsc w Polsce

Michał Bryła

słowa kluczowe:

Web

Firestore

Reactive programming

krótkie streszczenie:

Celem pracy jest zaprojektowanie i zaimplementowanie aplikacji webowej umożliwiającej tworzenie bazy niezwykłych miejsc przez użytkowników. Przedstawiono po krótku konkurencyjne rozwiązania, przypadki użycia, wymagania funkcjonalne i нефункционалне aplikacji, oraz najbardziej interesujące aspekty szczegółów implementacyjnych

opiekun pracy dyplomowej
	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego
	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do:*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław 2017

Dla Karoliny, za cierpliwość.

Streszczenie

W obecnych czasach kiedy Internet jest powszechnie dostępny, stał się nieodłącznym elementem życia każdego człowieka. Zostaje wykorzystywany przez ludzi do coraz to nowszych zastosowań, co wymaga opracowywania coraz to wydajniejszych technologii. W ramach zafascynowania tak szybkim rozwojem tworzenia aplikacji webowych i postanowiono napisać tytułową pracę w celu poznania głębiej tajników jednej z technologii.

Zaprezentowany zostanie projekt i implementacja aplikacji webowej umożliwiającej użytkownikom stworzenie bazy najpiękniejszych miejsc w Polsce, która ułatwi planowanie podróży i skupi w jednym miejscu wszystkie niezbędne informacje takie jak: lokalizacja, opis, zdjęcia, oceny czy komentarze. Pokazane zostaną konkurencyjne rozwiązania, wraz z ich zaletami i wadami, wymagania funkcjonalne i нефункционалне aplikacji, przypadki użycia, interfejs użytkownika oraz ciekawsze aspekty szczegółów implementacyjnych.

Efektem pracy jest gotowa, w pełni działająca, responsywna aplikacja, zbudowana w oparciu o architekturę serverless, pozwalająca, na tworzenie przez jej użytkowników bazy niezwykłych miejsc.

Abstract

Nowadays, when Internet is widely available, it has become an inseparable part of everyone's life. It's used by people for more ever newer applications which need ever more efficient technologies. As part of fascination with such a rapid development of web applications, decided to make title app in order to learn the dense secrets of one of the technologies.

There will be presented the project and implementation of web application, which allow users to create a database of the most beautiful places in Poland, which would help with planning the trip and bring all the information user need in one place such as: location, description, photos, ranks or comments. It will be showed current solutions on the market, along with their both pros and cons, the functional and non-functional requirements of application, use cases, user interface and more interesting aspects of implementation.

The result is a ready-made, fully-functional, responsive application, built on a serverless architecture, which allows users to create a database of places.

Spis treści

1. WSTĘP	5
1.1. WPROWADZENIE W TEMATYKĘ.....	5
1.2. PRZEGLĄD ISTNIEJĄCYCH ROZWIĄZAŃ.....	5
1.3. CEL PRACY	8
2. PROJEKT	9
2.1. WYMAGANIA FUNKCJONALNE I NIEFUNKCJONALNE	9
2.2. UŻYTKOWNICY.....	9
2.3. DIAGRAM PRZYPADKÓW UŻYCIA.....	10
2.4. SCENARIUSZE PRZYPADKÓW UŻYCIA	11
2.5. ARCHITEKTURA APLIKACJI	18
2.6. SCHEMAT BAZY DANYCH	19
3. INTERFEJS UŻYTKOWNIKA.....	20
3.1. STRONA STARTOWA	21
3.2. WIDOK MAPY	21
3.3. WIDOK SZCZEGÓŁÓW MIEJSCA	23
3.4. WIDOK DODAWANIA/EDYTOWANIA MIEJSCA	25
3.5. WIDOK RANKINGÓW	26
3.6. WIDOK PROFILU UŻYTKOWNIKA.....	27
3.7. WIDOK REJESTRACJI	28
3.8. WIDOK LOGOWANIA	29
4. IMPLEMENTACJA.....	30
4.1. ŚRODOWISKO I NARZĘDZIA PROGRAMISTYCZNE	30
4.2. UŻYTE BIBLIOTEKI I KOMPONENTY.....	30
4.3. WYKORZYSTANE WZORCE PROJEKTOWE.....	30
4.4. WYMIANA DANYCH.....	31
4.5. PROBLEMY IMPLEMENTACYJNE.....	32
4.6. PLANY ROZWOJU APLIKACJI.....	35
5. PODSUMOWANIE.....	36
BIBLIOGRAFIA	37
SPIS ILUSTRACJI	39
SPIS TABEL	40

1. Wstęp

1.1. Wprowadzenie w tematykę

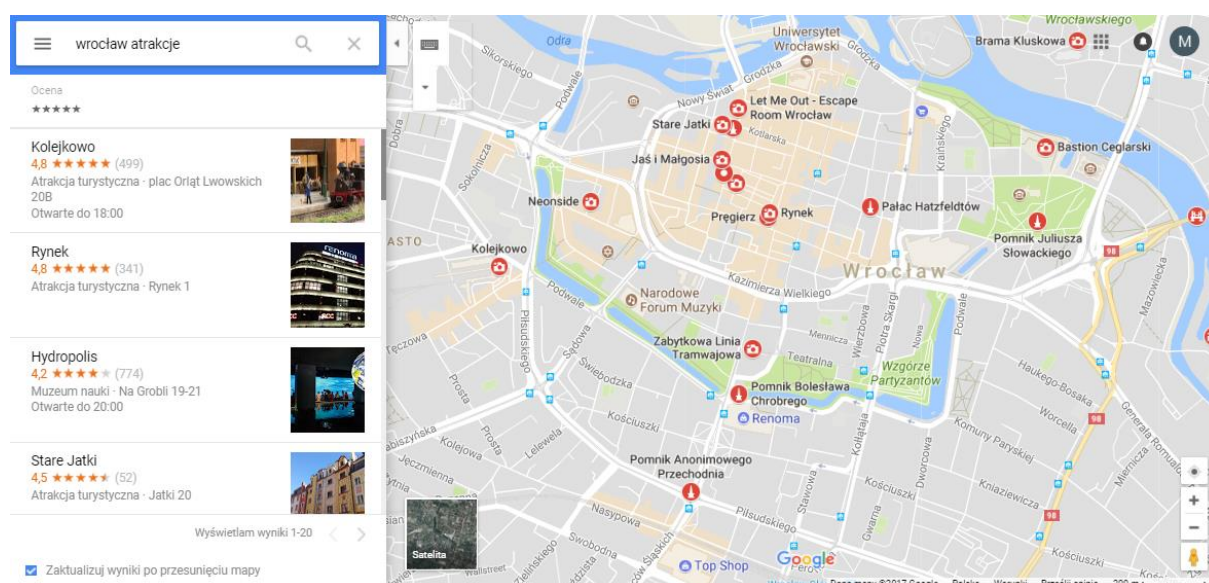
Podróże, nie muszą składać się jedynie z wyprawy w najbardziej znane i oddalone zakątki świata. Ludzie, rzadko zdają sobie sprawę z atrakcji jakie znajdują się w ich niedalekiej okolicy. Najczęściej jedynymi miejscami, które mogą polecić to ratusze, czy rynki z pobliskich miast. Jest to głównie spowodowane brakiem źródła wiedzy, ponieważ podczas szukania w Internecie lokalizacji godnych zobaczenia ciężko jest trafić na takie które jest zarówno obiektywne, posiada wiele informacji oraz liczną bazę takich miejsc.

Geneza pracy opiera się na osobistych doświadczeniach autora i jego najbliższych znajomych, którzy wielokrotnie borykali się z podobnym problemem podczas próby zaplanowania niedalekiej wycieczki po okolicznych atrakcjach turystycznych. Postanowiono rozwiązać dany problem, czego wynikiem jest tytułowa aplikacja webowa.

1.2. Przegląd istniejących rozwiązań

Istnieją już gotowe rozwiązania, aczkolwiek nie spełniają wszystkich wymagań. Paru z nich blisko do ideału, jednak to nadal nie jest produkt, który w pełni zadowoliłby wymagającego użytkownika. Podczas poszukiwań okolicznych atrakcji, większość ludzi podejmuje te same kroki tj. wpisanie w wyszukiwarce internetowej nazwy okolicy oraz frazy „atrakcje”, bądź „ciekawe miejsca”, ewentualnie jest to bardziej skonkretyzowana treść. Następnie użytkownik ma do wyboru parę opcji. Spośród wielu postanowiono wybrać te, cieszące się obecnie na rynku największym zainteresowaniem.

Pierwszą są niewątpliwie „Google Places” (Rys.1.1) dostępne na „Google Maps” [15]. Większość internautów to tutaj ląduje na początku podczas poszukiwania okolicznych atrakcji, co nie znaczy, że na tym kończą się ich poszukiwania. Jest to chyba najbardziej popularne rozwiązanie, ze względu na olbrzymią bazę danych, oraz integrację z wyszukiwarką, ponieważ są to produkty tego samego giganta branży informatycznej.



Rysunek 1.1 Aplikacja „Google Maps”

Zalety:

- Ogromna baza danych
- Tworzona przez użytkowników
- Zachęcanie użytkowników do wyrażania opinii poprzez program „Lokalny przewodnik”
- Wyznaczanie trasy do danej lokalizacji

Wady:

- Brak filtrów

Kolejną opcją, są liczne fora i wpisy na blogach, gdzie użytkownik może najczęściej znaleźć zestawienia najciekawszych miejsc. Niestety takie rozwiązanie jest bardzo niewygodne w użytkowaniu, ponieważ internauta jest zmuszony do przeszukiwania wielu takich stron z wspomnianymi zestawieniami, które dodatkowo bardzo często są subiektywną opinią autora takiej listy. Dodatkowo różne wpisy dotyczące tej samej okolicy, zawierają w sobie zazwyczaj te same, najbardziej znane i oklepane miejsca. Przykładem może być zaprezentowany na poniższym zrzucie ekranu wpis (Rys.1.2) dostępny na stronie fly4free [7].

FLY FREE TANIE LATANIE PORÓWNAJ LOTY, HOTELE, LOT+HOTEL • FORUM • FLY4FREE.COM

Wpisz szukaną frazę

Znalazłeś okazję? Napisz do nas »

Cudze chwalicie, swego nie znacie. Oto najbardziej niezwykle i tajemnicze miejsca w Polsce

6 maja 2015, 19:26 | Jakub Witoński | komentarzy 13

Share 15

FLY FREE TANIE LATANIE

Jeśli jeszcze nie znacie tych miejsc, musicie koniecznie je odwiedzić. Naprawdę warto.

Gdy pogoda dopisuje i jest już naprawdę ciepło, warto wyruszyć na wycieczkę po naszym pięknym kraju. Tym bardziej że kryje on przed nami wiele niesamowitych, tajemniczych i dziwnych miejsc.

Kaplica Czaszek w Czeremnej

W województwie dolnośląskim możemy podziwiać niezwykłą kaplicę. Koło kościoła parafialnego św. Bartłomieja w Czeremnej mieści się kaplica, której ściany i sufit pokryte są... trzema tysiącami ludzkich czaszek i kości! Pod kaplicą znajdują się szczątki kolejnych 21 tysięcy ofiar zaraz i wojen, które przetoczyły się przez ziemię kłodzką. Powstała ona pod koniec XVIII wieku za sprawą pochodzącego z Czech księdza Wacława Tomaszka. Będąc najprawdopodobniej pod silnym

Lastminuter

Przeloty czarterowe na Sri Lankę za 1299 zł

Wylot 25 listopada z Warszawy do Kolombo na 11 dni.
20 godz. 16 min. temu

Wyspy Zielonego Przylądka, 4* hotel na dwa tygodnie za 2014 zł

Wylot 26 listopada z Katowic.
21 godz. 47 min. temu

Kenia, 3 i 4* hotele z HB od 2154 zł!

Wylot 25 listopada z Warszawy na tydzień.
21 godz. 12 min. temu

Starsze »

KALENDARZE LOTÓW

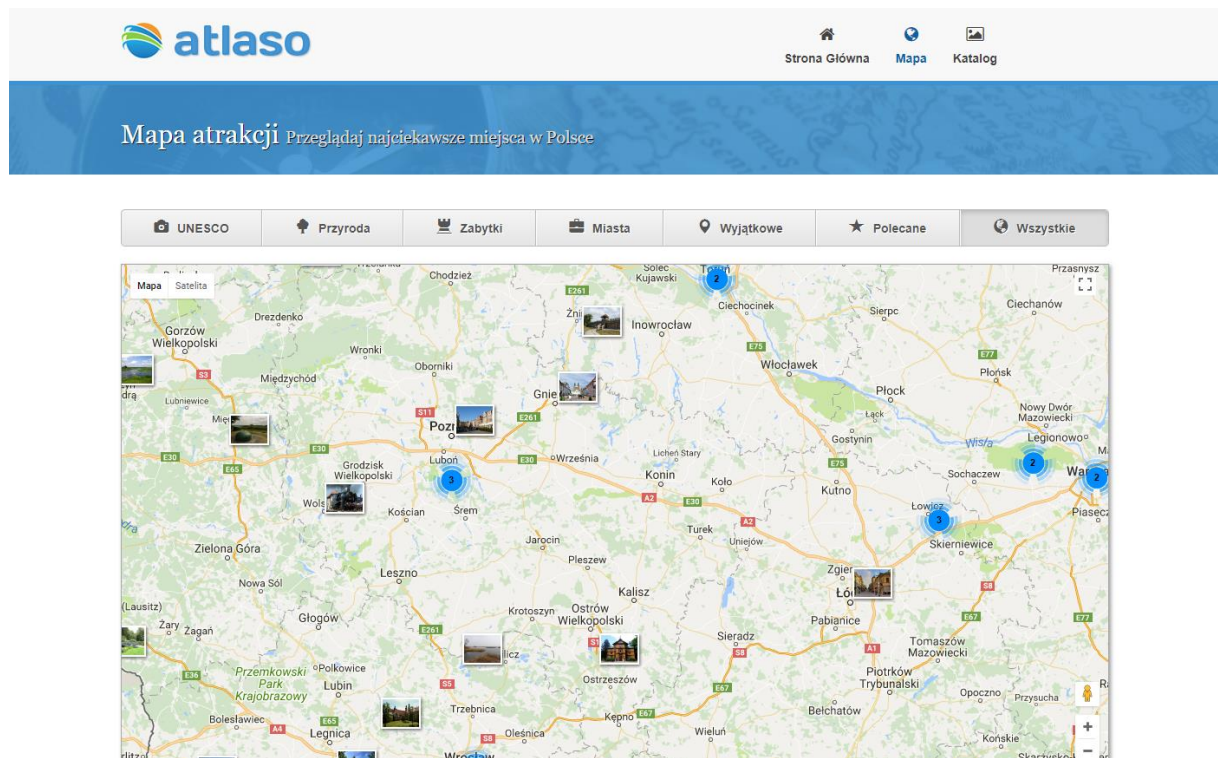
Warszawa -> Londyn -> Warszawa	73 zł
Warszawa -> Bolonia -> Warszawa	95 zł
Warszawa -> Eindhoven -> Warszawa	95 zł
Kraków -> Praga -> Kraków	78 zł
Warszawa -> Liverpool -> Warszawa	95 zł
Warszawa -> Lizbona -> Warszawa	188 zł
Warszawa -> Eilat -> Warszawa	68 zł
Katowice -> Chania -> Katowice	438 zł
Kraków -> Neapol -> Kraków	137 zł

Rysunek 1.2 Wpis na stronie fly4free.pl

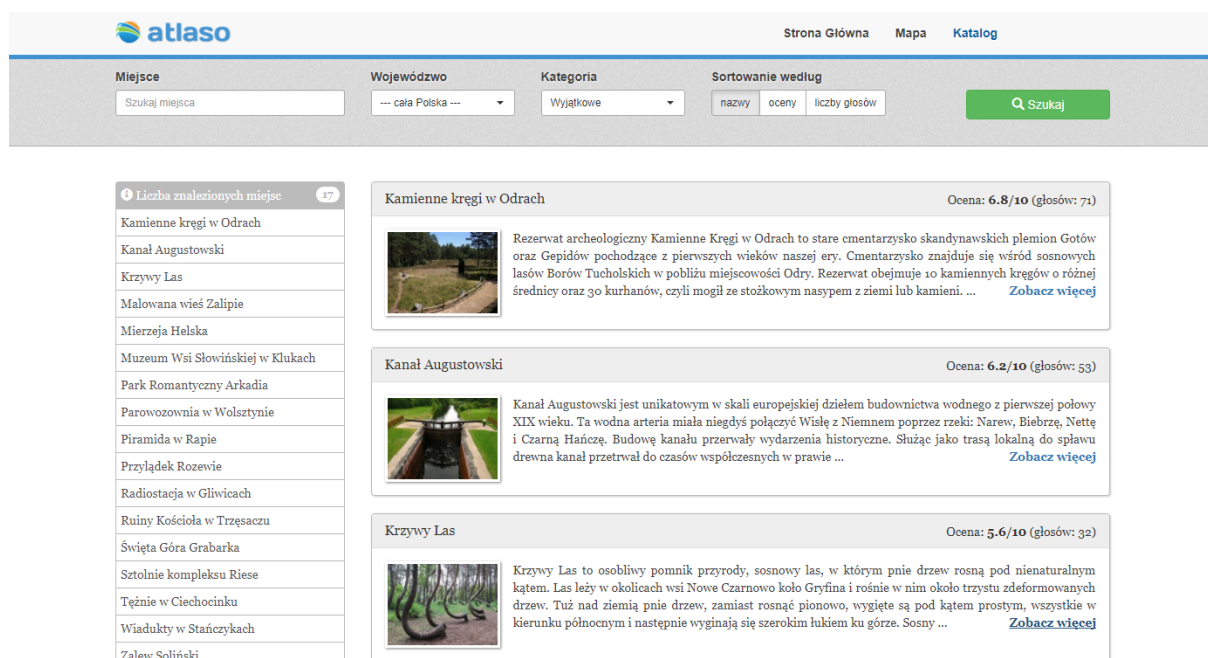
Wady:

- Subiektywne opinie
- Powtarzające się miejsca
- Wpisy najczęściej dotyczą znanych każdemu miejsc
- Brak możliwości wyszukiwania na mapie

Wyjątkową w tym krótkim zestawianiu, jest witryna znajdująca się pod adresem www.atlaso.pl [16] (Rys.1.3)(Rys.1.4). Na tle innych rozwiązań wyróżnia się tym że oprócz znanych lokalizacji posiada też faktycznie niezwykle i mało popularne miejsca. Jest to prawie idealne rozwiązanie ponieważ posiada jedną, aczkolwiek bardzo dużą wadę, mianowicie baza miejsc jest tworzona przez ograniczoną liczbę osób, co skutkuje tym, że jest ona bardzo mała.



Rysunek 1.3 Aplikacja „atlaso”, widok mapy



Rysunek 1.4 Aplikacja „atlaso”, widok listy

Zalety:

- Baza zawierająca faktycznie wyjątkowe miejsca
- Możliwość szukania zarówno po mapie jak i w klasyczny sposób

Wady:

- Baza jest tworzona przez ograniczoną liczbę osób

Wszystkie wyżej wymienione rozwiązania posiadają szereg niezbędnych funkcjonalności, jednak każda z nich posiada również mniejsze, bądź większe wady, które sprawiają że nie są to aplikacje idealne, spełniające wszystkie wymagania.

1.3. Cel pracy

Celem pracy jest zaprojektowanie i zaimplementowanie aplikacji webowej pozwalającej użytkownikom na stworzenie bazy najpiękniejszych miejsc w Polsce, która ułatwi planowanie podróży i skupi w jednym miejscu wszystkie informacje takie jak: lokalizacja, opis, zdjęcia, oceny czy komentarze.

Zakres pracy obejmuje zbudowanie w pełni działającej i responsywnej aplikacji webowej opartej na architekturze serverless., która ma umożliwiać przeglądanie, dodawanie i wyszukiwanie miejsc na podstawie wpisanych kryteriów w polu edycyjnym, a także przez lokalizację na mapie. Każdy użytkownik będzie posiadać swój profil, na którym będą udostępniane informacje na temat dodanych przez niego miejsc do bazy i oznaczonych jako ulubione. Dodatkowo użytkownicy będą mogli komentować i oceniać najlepsze według nich lokalizacje, na podstawie czego tworzone będą rankingi najciekawszych miejsc.

2. Projekt

2.1. Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne:

- Tworzenie własnego konta w aplikacji
- Możliwość logowania się do aplikacji kontem Google
- Dodawanie miejsca, wraz z wszelkimi niezbędnymi informacjami
- Możliwość aktualizowania wybranego miejsca
- Zatwierdzanie miejsca, czy jest autentyczne
- Usuwanie miejsca z bazy
- Dodanie miejsca do ulubionych przez użytkownika
- Usunięcie miejsca z ulubionych przez użytkownika
- Możliwość oceny miejsca
- Wyszukiwanie miejsc na mapie
- Wyszukiwanie miejsc za pomocą pola tekstowego
- Filtrowanie wyników ze względu na kategorię i typ
- Komentowanie wybranych miejsc
- Edycja i usuwanie komentarzy
- Przeglądanie własnego profilu
- Edycja własnego profilu
- Przeglądanie profili innych użytkowników
- Przeglądanie rankingów najlepiej ocenianych miejsc
- Możliwość sprawdzenia ostatnio dodanych miejsc

Wymagania нефункционалне:

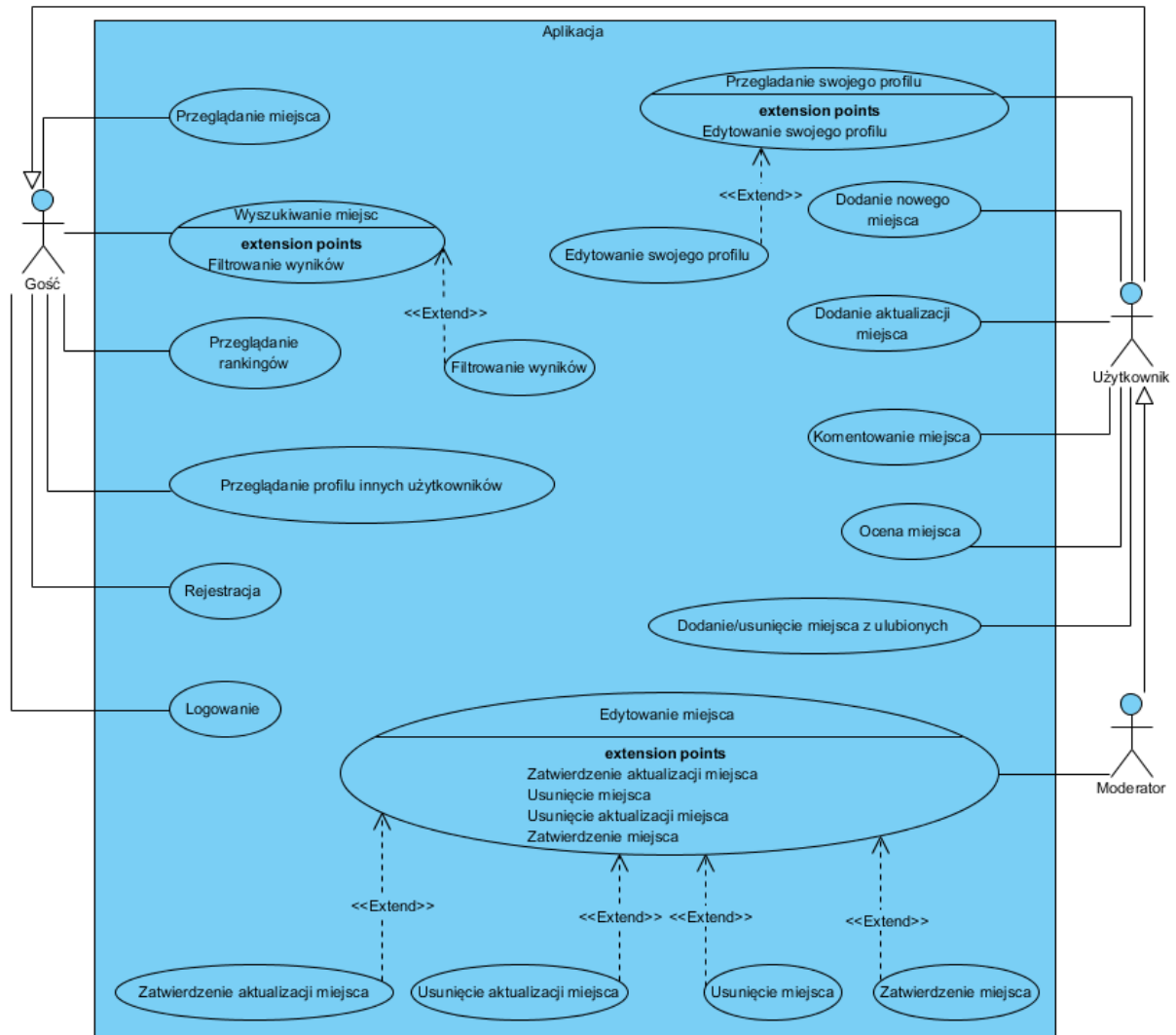
- Poprawne działanie na najpopularniejszych przeglądarkach: Chrome, Firefox, Safari, Edge
- Skalowalność aplikacji na różnych rozdzielczościach ekranu
- Obsługa 100 połączeń na raz przez bazę danych
- Obsługa do 20tyś. zapisów w bazie danych na dzień
- Obsługa do 50tyś. odczytów z bazy danych na dzień
- Obsługa do 20tyś. usunięć w bazie danych na dzień

2.2. Użytkownicy

W aplikacji zdefiniowano trzy typy aktorów:

- **Gość** – aktor, który może korzystać jedynie z podstawowych funkcji, czyli przeglądania danych, oraz ma możliwość zarówno założenia konta jak i zalogowania.
- **Użytkownik** – aktor, którym staje się Gość po zalogowaniu się do aplikacji. Posiada wszystkie jego możliwości, a także może zarówno dodawać miejsca do bazy jak i tworzyć aktualizacje miejsc. Dodatkowo uzyskuje dostęp do komentowania, oceniania czy dodawania do ulubionych.
- **Moderator** – aktor, który jest szczególnym przypadkiem Użytkownika, ponieważ rozszerza jego możliwości o całkowitą edycję miejsc tj. zatwierdzanie stworzonych przez Użytkowników aktualizacji miejsc, czy też usuwanie.

2.3. Diagram przypadków użycia



Rysunek 2.1 Diagram przypadków użycia aplikacji

2.4. Scenariusze przypadków użycia

Lp.	PU-1
Nazwa:	Wyszukiwanie miejsc
Aktorzy:	Gość, Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie startowej
Opis:	Aktor chce wyszukać miejsca w konkretnej okolicy.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor wypełnia pole edycyjne. 2. Aktor zatwierdza wpisane dane poprzez wciśnięcie przycisku „Szukaj”. 3. Aktor zostaje przekierowany na stronę z widokiem mapy i wynikami wyszukiwania.
Ścieżka alternatywna:	
Warunki końcowe:	Aktor znajduje się na stronie mapy z wynikami wyszukiwania

Tabela 2.1 PU-1 Wyszukiwanie miejsc

Lp.	PU-2
Nazwa:	Filtrowanie wyników
Aktorzy:	Gość, Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie z mapą
Opis:	Aktor chce przefiltrować wyniki.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor zaznacza filtry do zaaplikowania. 2. Aktor zatwierdza wybrane filtry poprzez wciśnięcie przycisku „Aplikuj filtry”.
Ścieżka alternatywna:	2a. Aktor anuluje wybrane filtry poprzez wciśnięcie przycisku „Wyczyść filtry”.
Warunki końcowe:	Wyniki wyszukiwania są przefiltrowane według wskazanych przez Aktora kryteriów

Tabela 2.2 PU-2 – Filtrowanie wyników

Lp.	PU-3
Nazwa:	Przeglądanie miejsca
Aktorzy:	Gość, Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie z mapą
Opis:	Aktor chce wyświetlić szczegóły wybranego miejsca.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika wybrany marker na mapie. 2. Wyświetlona zostaje mała karta wybranego miejsca, zawierająca zdjęcie nazwę oraz ocenę. 3. Aktor klika na kartę. 4. Aktor zostaje przekierowany na stronę ze szczegółami dotyczącymi wybranego miejsca.
Ścieżka alternatywna:	3a. Aktor klika na inny marker.
Warunki końcowe:	Aktor znajduje się na stronie z szczegółami wybranego miejsca

Tabela 2.3 PU-3 – Przeglądanie miejsca

Lp.	PU-4
Nazwa:	Przeglądanie rankingów
Aktorzy:	Gość, Użytkownik, Moderator
Warunki początkowe:	
Opis:	Aktor chce wyświetlić rankingi miejsc
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na opcję „Rankingi” znajdującą się w menu nawigacyjnym. 2. Aktor zostaje przekierowany na stronę z domyślnym rankingiem tj. „Najlepiej oceniane”
Ścieżka alternatywna:	<ol style="list-style-type: none"> 3. Aktor wybiera inny interesujący go ranking.
Warunki końcowe:	Aktor znajduje się na stronie z rankingami miejsc

Tabela 2.4 PU-4 – Przeglądanie rankingów

Lp.	PU-5
Nazwa:	Przeglądanie profili innych użytkowników
Aktorzy:	Gość, Użytkownik, Moderator
Warunki początkowe:	
Opis:	Aktor chce wyświetlić profil dowolnego użytkownika
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na odnośnik do profilu dowolnego użytkownika. 2. Aktor zostaje przekierowany na stronę z widokiem profilu wybranego użytkownika.
Ścieżka alternatywna:	2a. Konto wybranego użytkownika zostało usunięte, zostaje wyświetlony odpowiadający tej informacji komunikat.
Warunki końcowe:	Aktor znajduje się na stronie profilu wybranego użytkownika

Tabela 2.5 PU-5 – Przeglądanie profili innych użytkowników

Lp.	PU-6
Nazwa:	Logowanie
Aktorzy:	Gość
Warunki początkowe:	
Opis:	Aktor chce zalogować się do swojego konta w aplikacji
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na opcję „Zaloguj” znajdującą się w menu nawigacyjnym. 2. Zostaje wyświetlone okno logowania. 3. Aktor wypełnia pole dotyczące jego adresu email i hasła. 4. Aktor zatwierdza wpisane dane poprzez kliknięcie przycisku „Zaloguj”
Ścieżka alternatywna:	<p>3a. Wpisane dane nie spełniają wymagań, zostaje wyświetlony odpowiedni komunikat.</p> <p>3b. Aktor chce zalogować się do aplikacji zewnętrznym kontem „Google” i klika na przycisk „Zaloguj kontem Google”</p> <p>3b1. Zostaje wyświetlone okno zewnętrznego uwierzytelniania.</p> <p>4a. Wpisane dane nie zostają odnalezione w bazie i zostaje wyświetlony odpowiedni komunikat błędu.</p>
Warunki końcowe:	Aktor zalogował się do swojego konta

Tabela 2.6 PU-6 – Logowanie

Lp.	PU-7
Nazwa:	Rejestracja
Aktorzy:	Gość
Warunki początkowe:	Aktor ma wyświetlone okno logowania
Opis:	Aktor chce założyć konto w aplikacji
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na przycisk „Zarejestruj się” 2. Aktor zostaje przekierowany na stronę rejestracji. 3. Aktor wypełnia wszystkie pola. 4. Aktor zatwierdza wpisane dane poprzez kliknięcie przycisku „Zarejestruj”
Ścieżka alternatywna:	3a. Wpisane dane nie spełniają wymagań, zostaje wyświetlony odpowiedni komunikat.
Warunki końcowe:	Aktor zarejestrował się i posiada własne konto

Tabela 2.7 PU-7 – Rejestracja

Lp.	PU-8
Nazwa:	Dodanie nowego miejsca
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	
Opis:	Aktor chce dodać nowe miejsce do bazy
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na opcję „Dodaj miejsce” znajdującą się w menu nawigacyjnym. 2. Aktor zostaje przekierowany na stronę dodawanie nowego miejsca. 3. Aktor wypełnia pola dotyczące miejsca tj. nazwa i opis. 4. Aktor wybiera kategorię miejsca z listy. 5. Aktor dodaje zdjęcia. 6. Aktor dodaje znacznik na mapie. 7. Aktor zatwierdza wpisane dane poprzez kliknięcie przycisku „Dodaj nowe miejsce” 8. Zostaje wyświetlony pasek ładowania 9. Aktor zostaje przekierowany na stronę ze szczegółami dotyczącymi wybranego miejsca.
Ścieżka alternatywna:	<p>3a. Wpisane dane nie spełniają wymagań, zostaje wyświetlony odpowiedni komunikat.</p> <p>5a. Wybrane zdjęcia przekraczają dozwolony limit rozmiarowy, zostaje wyświetlony odpowiedni komunikat.</p>
Warunki końcowe:	Nowe miejsce zostało dodane do bazy

Tabela 2.8 PU-8 – Dodanie nowego miejsca

Lp.	PU-9
Nazwa:	Dodanie aktualizacji miejsca
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce zaktualizować informacje dotyczące wybranego miejsca
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na przycisk „Edytuj miejsce”. 2. Aktor zostaje przekierowany na stronę edytowania miejsca. 3. Aktor zmienia interesujące go dane. 4. Aktor zatwierdza wpisane dane poprzez kliknięcie przycisku „Edytuj miejsce” 5. Zostaje wyświetlony pasek ładowania 6. Aktor zostaje przekierowany na stronę ze szczegółami dotyczącymi wybranego miejsca.
Ścieżka alternatywna:	<p>3a. Wpisane dane nie spełniają wymagań, zostaje wyświetlony odpowiedni komunikat.</p> <p>3b. Wybrane zdjęcia przekraczają dozwolony limit rozmiarowy, zostaje wyświetlony odpowiedni komunikat.</p>
Warunki końcowe:	Nowa wersja miejsca została dodana do bazy i czeka na zatwierdzenie przez moderatora

Tabela 2.9 PU-9 – Dodanie aktualizacji miejsca

Lp.	PU-10
Nazwa:	Komentowanie miejsca
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce dodać komentarz dotyczący wybranego miejsca
Ścieżka główna:	1. Aktor wypełnia pole dotyczące treści komentarza. 2. Aktor zatwierdza wpisany komentarz poprzez kliknięcie przycisku „Dodaj komentarz”
Ścieżka alternatywna:	
Warunki końcowe:	Nowy komentarz zostaje dodany i widnieje na liście komentarzy wybranego miejsca

Tabela 2.10 PU-10 – Komentowanie miejsca

Lp.	PU-11
Nazwa:	Ocena miejsca
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce dodać lub zmienić swoją ocenę wybranego miejsca
Ścieżka główna:	1. Aktor wybiera liczbę gwiazdek ze skali. 2. Średnia ocena miejsca aktualizuje się
Ścieżka alternatywna:	
Warunki końcowe:	Średnia ocena miejsca zostaje zaktualizowana o wartość podaną przez Aktora

Tabela 2.11 PU-11 – Ocena miejsca

Lp.	PU-12
Nazwa:	Dodanie/usunięcie miejsca z ulubionych
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce dodać lub usunąć miejsce z ulubionych
Ścieżka główna:	1. Aktor klika na przycisk „Dodaj do ulubionych”. 2. Przycisk zmienia się na „Usuń z ulubionych”.
Ścieżka alternatywna:	1a. Aktor klika na przycisk „Usuń z ulubionych”. 2a. Przycisk zmienia się na „Dodaj do ulubionych”.
Warunki końcowe:	Lista ulubionych miejsc Aktora zostaje zaktualizowana

Tabela 2.12 PU-12 – Dodanie/usunięcie miejsca z ulubionych

Lp.	PU-13
Nazwa:	Przeglądanie swojego profilu
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	
Opis:	Aktor chce przeglądać swój profil
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na opcję „Profil” znajdującą się w menu nawigacyjnym. 2. Aktor zostaje przekierowany na stronę z widokiem swojego profilu.
Ścieżka alternatywna:	1a. Aktor klika na odnośnik do swojego profilu.
Warunki końcowe:	Aktor znajduje się na stronie swojego profilu

Tabela 2.13 PU-13 – Przeglądanie swojego profilu

Lp.	PU-14
Nazwa:	Edytowanie swojego profilu
Aktorzy:	Użytkownik, Moderator
Warunki początkowe:	Aktor znajduje się na stronie swojego profilu, i nie jest zalogowany poprzez konto z zewnętrznego systemu „Google”
Opis:	Aktor chce zmienić dane swojego konta
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor wypełnia pola, które chce zmienić. 2. Aktor zatwierdza zmiany poprzez kliknięcie przycisku „Zatwierdź zmiany”. 3. Wyświetla się komunikat o pomyślnym przeprowadzeniu zmian.
Ścieżka alternatywna:	1a. Wpisane dane nie spełniają wymagań, zostaje wyświetlony odpowiedni komunikat.
Warunki końcowe:	Dane Aktora zostają zaktualizowane

Tabela 2.14 PU-14 – Edytowanie swojego profilu

Lp.	PU-15
Nazwa:	Edytowanie miejsca
Aktorzy:	Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce edytować miejsce
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor wybiera spośród listy dostępnych wersji, tę którą chce zatwierdzić bądź usunąć. 2. Wyświetlają się szczegóły miejsca wybranej przez Aktora wersji.
Ścieżka alternatywna:	
Warunki końcowe:	Wybrana przez Aktora wersja miejsca zostaje wyświetlona

Tabela 2.15 PU-15 – Edytowanie miejsca

Lp.	PU-16
Nazwa:	Zatwierdzenie aktualizacji miejsca
Aktorzy:	Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami wybranej wersji miejsca
Opis:	Aktor chce zatwierdzić wybraną wersję miejsca
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na przycisk „Zatwierdź wersję” 2. Wybrana przez Aktora wersja staje się aktualną wersją miejsca.
Ścieżka alternatywna:	
Warunki końcowe:	Miejsce zostaje zaktualizowane o wybraną przez Aktora wersję

Tabela 2.16 PU-16 – Zatwierdzenie aktualizacji miejsca

Lp.	PU-17
Nazwa:	Usunięcie aktualizacji miejsca
Aktorzy:	Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami wybranej wersji miejsca
Opis:	Aktor chce usunąć wybraną wersję miejsca
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na przycisk „Usuń wersję” 2. Wyświetlają się szczegóły aktualnej wersji miejsca.
Ścieżka alternatywna:	
Warunki końcowe:	Wybrana wersja zostaje usunięta z listy wersji danego miejsca

Tabela 2.17 PU-17 – Usunięcie aktualizacji miejsca

Lp.	PU-18
Nazwa:	Usunięcie miejsca
Aktorzy:	Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce usunąć miejsce
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aktor klika na przycisk „Usuń miejsce” 2. Aktor zostaje przekierowany na stronę startową
Ścieżka alternatywna:	
Warunki końcowe:	Wybrane miejsce zostaje usunięte z bazy

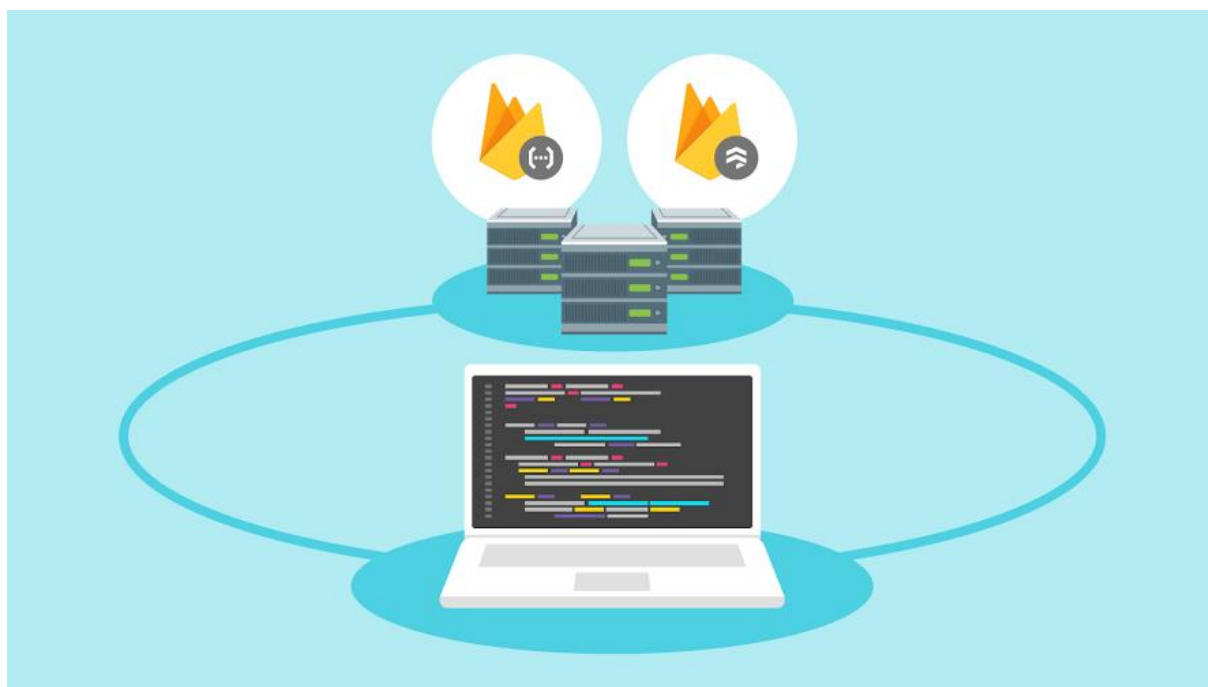
Tabela 2.18 PU-18 – Usunięcie miejsca

Lp.	PU-19
Nazwa:	Zatwierdzenie miejsca
Aktorzy:	Moderator
Warunki początkowe:	Aktor znajduje się na stronie ze szczegółami miejsca
Opis:	Aktor chce zatwierdzić miejsce
Ścieżka główna:	1. Aktor klika na przycisk „Zatwierdź miejsce” 2. Przycisk „Zatwierdź miejsce” staje się nieaktywny
Ścieżka alternatywna:	
Warunki końcowe:	Wybrane miejsce zostaje zatwierdzone i zaktualizowane

Tabela 2.19 PU-19 – Zatwierdzenie miejsca

2.5. Architektura aplikacji

Projekt zrealizowano w architekturze serverless. Aplikacja webowa w przeglądarce komunikuje się z usługami Back-end as a Service dostarczonymi przez Google Firebase [12] za pomocą JavaScript SDK (Software Development Kit), które pozwala na prawie natychmiastową synchronizację danych w aplikacji. Użytkownik korzystając z interfejsu graficznego wywołuje API usług chmurowych takich jak: wymiana danych z bazą, czy uwierzytelnianie użytkowników. Wykorzystano bazę danych Cloud Firestore, która jest skalowalną bazą NoSQL przechowującą dane w dokumentach, znajdujących się w zorganizowanych kolekcjach. Część przetwarzania danych wykonuje się bezpośrednio w aplikacji, natomiast bardziej skomplikowane operacje za które normalnie byłby odpowiedzialny serwer, są obsługiwane przez Google Cloud Functions, które są JavaScriptowymi funkcjami wdrożonymi przez programistę w chmurze. Uproszczony schemat architektury przedstawiono na poniższej ilustracji (Rys 2.2), w której widać że pomiędzy aplikacją kliencką a chmurą na której znajdują się wszystkie usługi i baza danych nic nie ma, ponieważ za wszystko odpowiedzialny jest Firebase.



Rysunek 2.2 Uproszczona ilustracja architektury serverless z wykorzystaniem „Google Firebase” <https://firebase.google.com/products/firestore/>

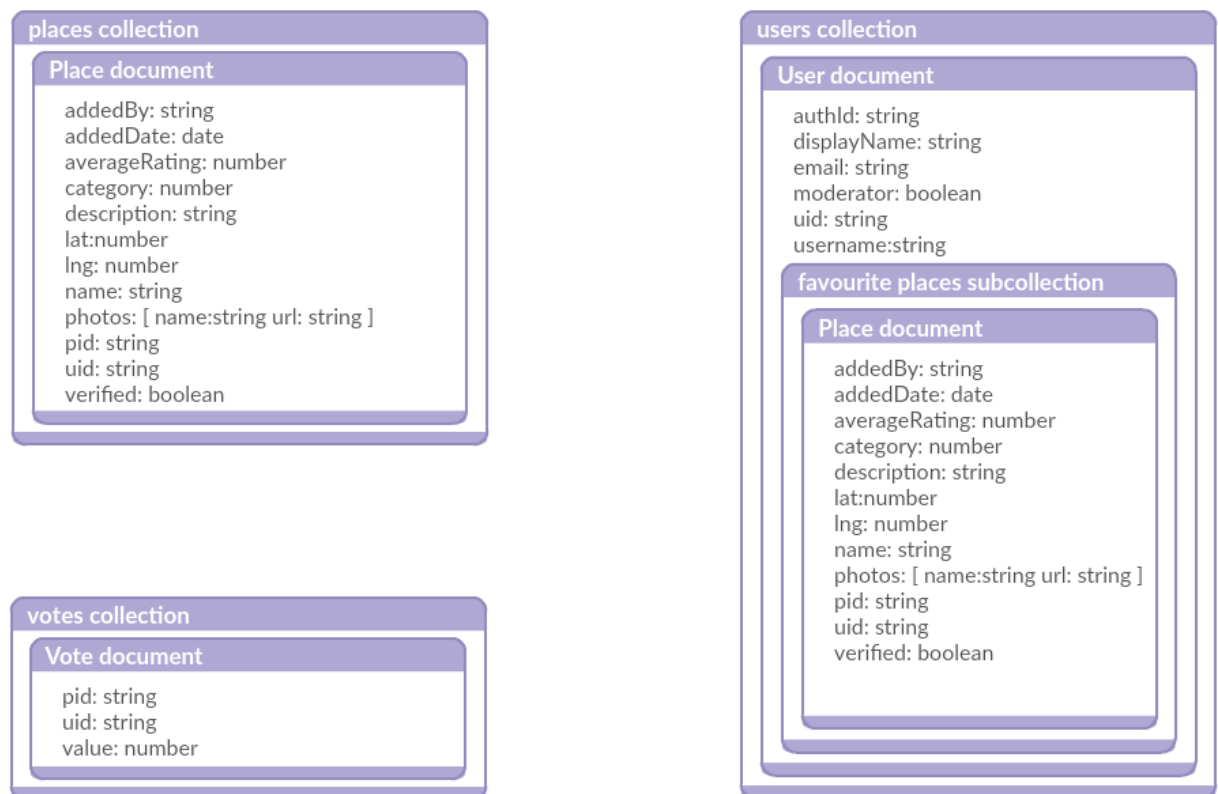
2.6. Schemat bazy danych

Do przechowywania danych jak już wyżej wspomniano wykorzystano NoSQL'ową bazę danych Cloud Firestore, przechowującą dane w dokumentach, znajdujących się w zorganizowanych kolekcjach. Poniżej przedstawiono schemat organizacji wspomnianych kolekcji oraz dokumentów(Rys.2.3).

Baza zawiera trzy główne kolekcje:

- Kolekcja miejsc – places collection
- Kolekcja użytkowników – users collection
- Kolekcja głosów – votes collection

Każda z nich zawiera w sobie dokumenty o określonych polach. Dodatkowo dokument użytkownika zawiera w sobie kolekcję ulubionych miejsc.



Rysunek 2.3 Schemat bazy danych

3. Interfejs użytkownika

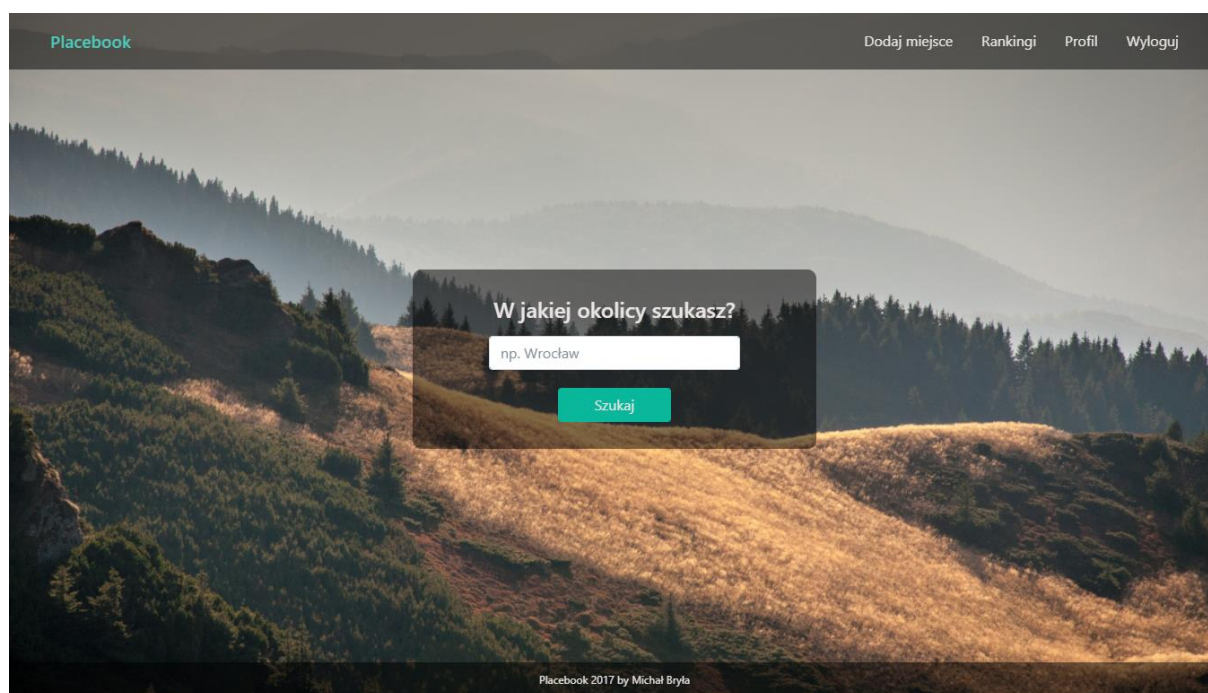
Interfejs aplikacji prezentuje się w formie ośmiu widoków. Zarówno „User Interface” jak i „User Experience”, są bardzo ważnymi aspektami podczas projektowania wygodnej w użytkowaniu aplikacji, dlatego podczas wykonywania widoków kierowano się wskazówkami zaczerpniętymi z książki autorstwa Krug, S. pt. „*Don't make me think, revisited: A common sense approach to web usability*” [1]. Wszystkie ikony zostały zaczerpnięte ze strony www.fontawesome.io [18], natomiast grafiki tła na stronie głównej z www.unsplash.com [19], reszta elementów graficznych takich jak markery na mapie wykonano osobiście.

Widoki aplikacji:

1. Strona startowa
 - pozwala na wpisanie użytkownikowi okolicy, w której chce szukać
2. Widok mapy
 - pozwala na przeglądanie mapy z umiejscowionymi markerami miejsc, a także na filtrowanie wyników
3. Widok szczegółów miejsca
 - wyświetla wszystkie informacje na temat wybranego miejsca
4. Widok dodawania/edytowania miejsca
 - pozwala na dodanie nowego miejsca do bazy lub edycje istniejącego
5. Widok profilu użytkownika
 - wyświetla informacje na temat użytkownika
6. Widok rankingów
 - wyświetla rankingi miejsc
7. Widok rejestracji
 - Pozwala na stworzenie konta w aplikacji
8. Widok logowania
 - Pozwala na zalogowanie się do aplikacji

3.1. Strona startowa

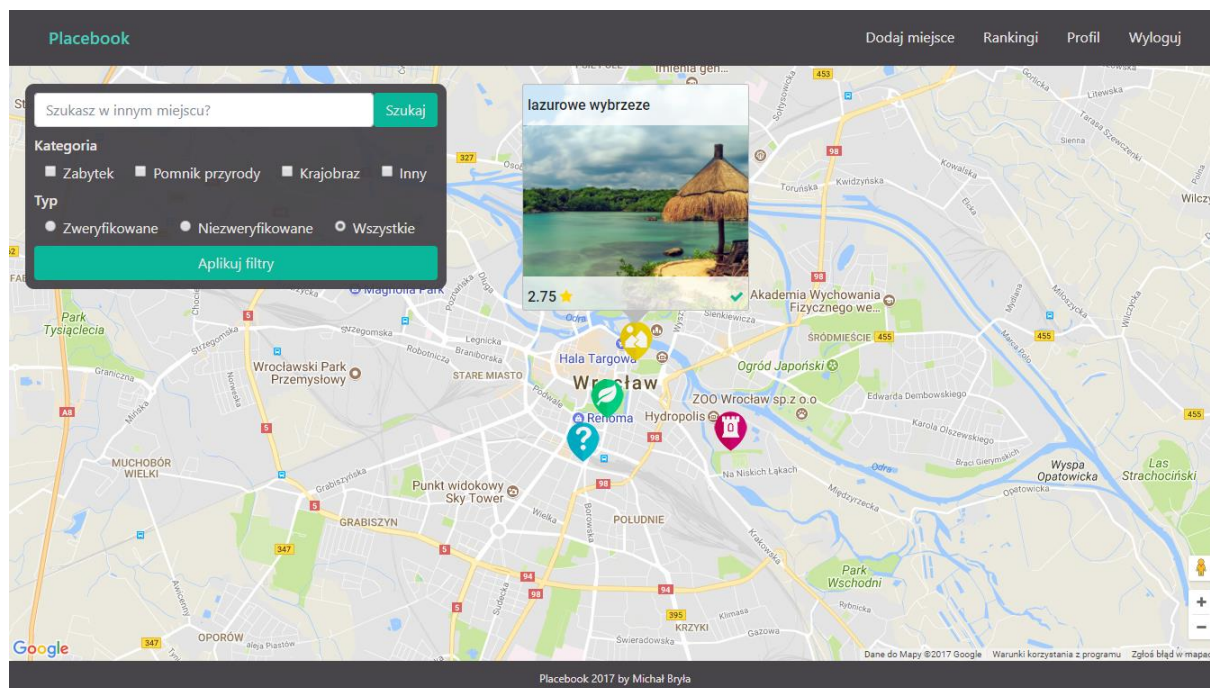
Pierwszy widok aplikacji (Rys.3.1), jest prosty i intuicyjny. Posiada na środku pole edycyjne w które użytkownik wpisuje okolicę, w której chce szukać miejsc. Dodatkowo by poprawić komfort zaimplementowano autouzupełnianie od „Google”, tak by po wpisaniu pierwszych liter podpowiedziano i dano możliwość wyboru z dostępnych lokalizacji. Po wpisaniu okolicy i zatwierdzenie wyboru, poprzez wciśnięcie przycisku „Szukaj”, użytkownik zostaje przekierowany do widoku mapy.



Rysunek 3.1 Strona startowa aplikacji

3.2. Widok mapy

W tym widoku prezentowana jest mapa wraz z naniesionymi markerami miejsc (Rys.3.2). Wygląd markera odpowiada kategorii miejsca i na chwilę obecną są to: Pomnik przyrody (Rys.3.3), Zabytek (Rys.3.4), Krajobraz (Rys.3.5) oraz Inne (Rys.3.6), która reprezentuje typy miejsc niekwalifikujące się pod żadną z poprzednich kategorii. Po zaznaczeniu markera przez użytkownika, ukazują się kafelki z podglądem zdjęcia i informacjami o nazwie miejsca, ocenie oraz czy zostało zweryfikowane (świadczy o tym zielony ptaszek w prawym dolny rogu kafelka). Po kliknięciu w kafelki widok zmienia się na dotyczący szczegółów miejsca. Dodatkowo w lewym górnym rogu widoku znajduje się sekcja z polem edycyjnym by wyszukać w innej okolicy oraz filtrami. Użytkownik może wybrać jakie kategorie miejsc chce widzieć oraz czy mają być to tylko zweryfikowane, niezweryfikowane, bądź domyślnie wszystkie typy miejsc. Mapa jest w pełni interaktywna, zatem można ją przybliżać i oddalać, a także przesuwając, dzięki czemu użytkownik ma możliwość wyszukiwania po mapie.



Rysunek 3.2 Widok mapy



Rysunek 3.3 Kategoria: Pomnik przyrody



Rysunek 3.4 Kategoria: Zabytek



Rysunek 3.5 Kategoria: Krajobraz



Rysunek 3.6 Kategoria: Inne

3.3. Widok szczegółów miejsca

Widok ten wyświetla wszystkie informacje na temat miejsca (Rys.3.7). Znajdują się tu takie szczegóły jak:

- nazwa
- kategoria
- autor
- data dodania
- średnia ocena
- galeria zdjęć
- opis
- lokalizacja na mapie
- komentarze

Na rysunku (Rys.3.7), przedstawiono najbardziej rozbudowany widok, który jest dostępny dla moderatora. Od widoku zwykłego zalogowanego użytkownika wyróżnia go górne menu, na którym znajdują się przyciski „Usuń miejsce” oraz „Zatwierdź miejsce”, a także lista wersji do zweryfikowania. Dzięki tym opcjom moderator ma możliwość kontrolowania aktualnej wersji miejsca. Gościowi, czyli użytkownikowi, który się nie zalogował dodatkowo nie pokazują się przyciski „Edytuj miejsce” i „Dodaj do ulubionych” oraz skala gwiazdek do oceny miejsca, ponieważ są to funkcjonalności dostępne wyłącznie dla uwierzytelnionych użytkowników aplikacji. Gość nie widzi również znajdującego się na samym dole pola edycyjnego dotyczącego dodania komentarza wraz z przyciskiem „Dodaj komentarz”.

Usuń miejsce

Zatwierdź miejsce

Wersje do zweryfikowania: Oryginał eY9hJcPIBUsq2XWDEIA

Pomnik przyrody !

Kategoria: Zabytek

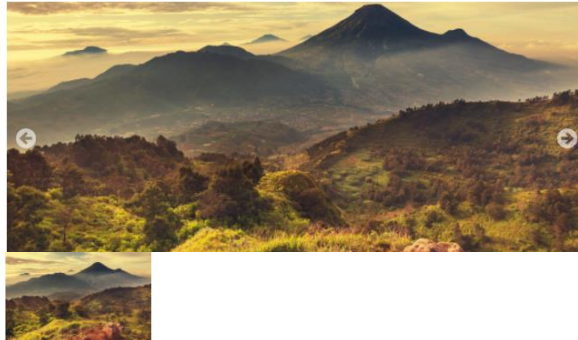
Dodane przez: brylka333, 11/16/2017, 11:49 PM

Ocena: 3 ★

Edytuj miejsce

Dodaj do ulubionych

Twoja ocena: ★ ★ ★ ★ ★



Opis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Komentarze

mkbryla, 11/28/2017, 7:45 PM
super miejsce!



Wpisz swój komentarz

Dodaj komentarz

3.4. Widok dodawania/edytowania miejsca

Widok dodawania/edytowania miejsca (Rys.3.8) umożliwia użytkownikowi dodanie nowego miejsca lub edycję istniejącego. Znajdują się w nim pola edycyjne dotyczące nazwy i opisu, oraz pole wyboru kategorii z listy. Dodatkowo pozwala na dodanie zdjęć zarówno poprzez kliknięcie przycisku „Wybierz zdjęcia”, a następnie wybraniu z ukazanego okna eksploratora plików obrazów, jak i zwykłym przeciągnięciu obrazów w szary obszar strefy przeciągania. Niżej widnieje interaktywna mapa, na której użytkownik może zaznaczyć lokalizację miejsca.

Placebook

[Dodaj miejsce](#)[Rankingi](#)[Profil](#)[Wyloguj](#)

Nazwa

Testowy zabytek

Kategoria

Zabytek


Opis


Testowy opis zabytku

WYBIERZ ZDJĘCIA


WYCZYŚĆ

lub upuść pliki tutaj





*Maksymalny rozmiar pliku wynosi 5mb



Dane do Mapy ©2017 Google

Warunki korzystania z programu

Zgłoś błąd w mapach

Dodaj nowe miejsce!

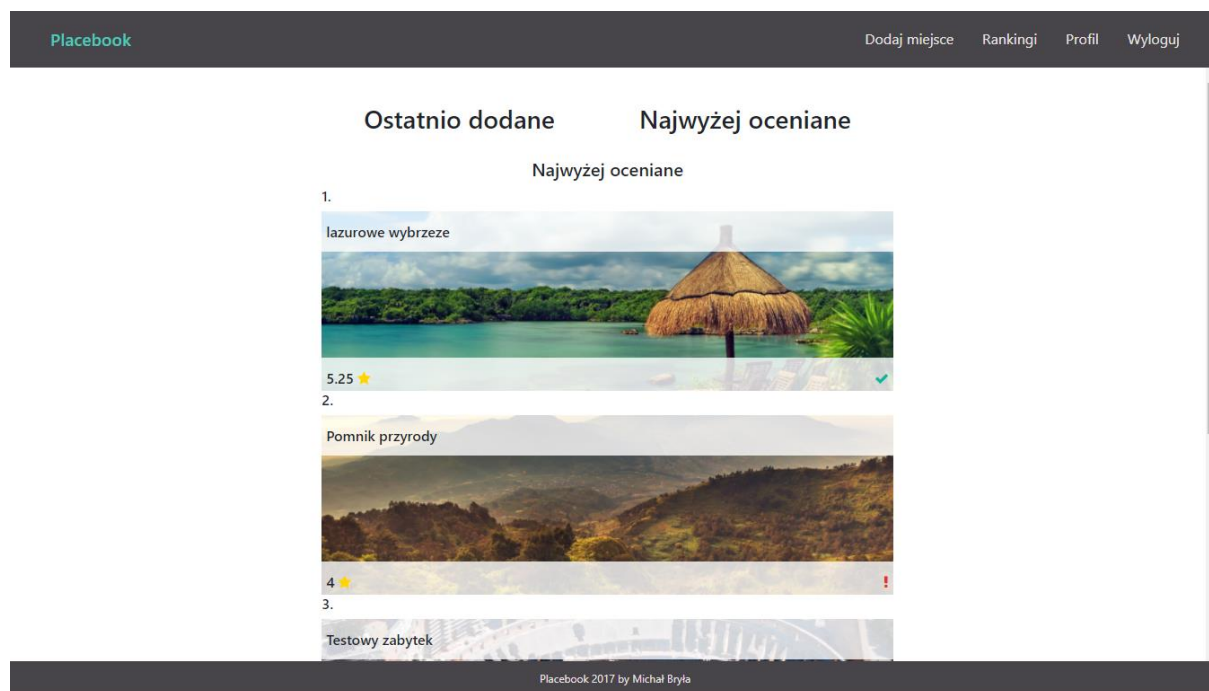
Placebook 2017 by Michał Bryła

Rysunek 3.8 Widok dodawania/edytowania miejsca

25

3.5. Widok rankingów

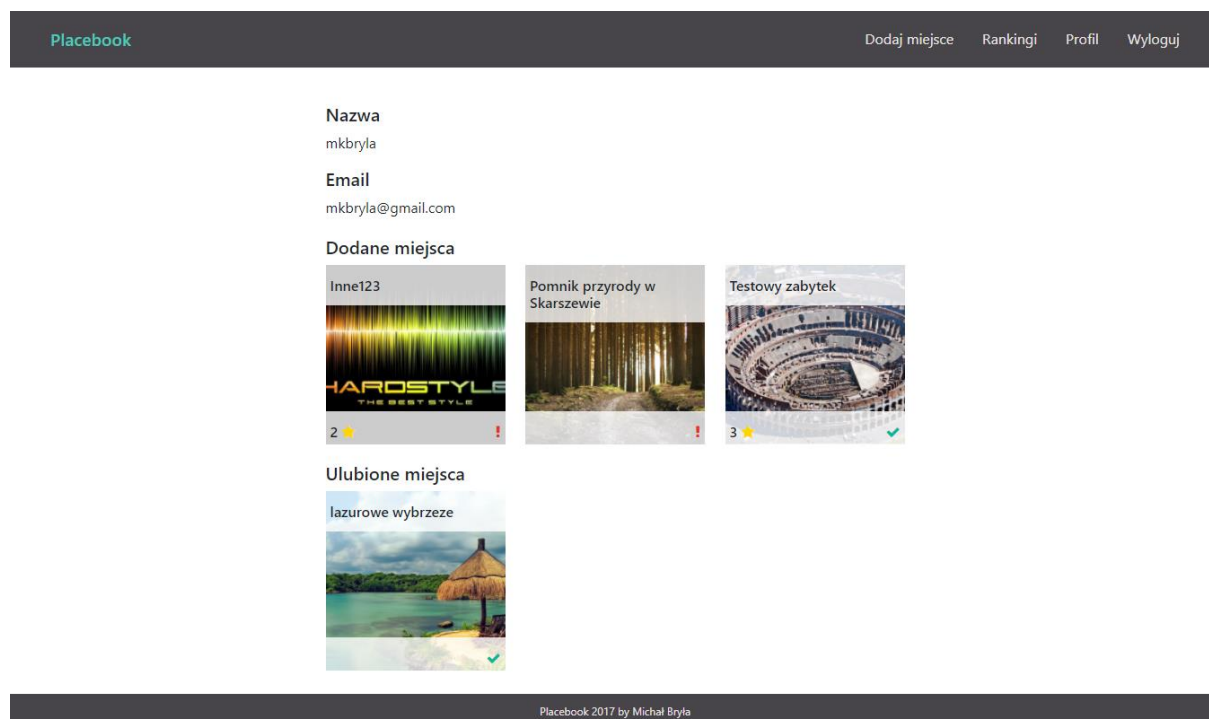
W tym widoku wyświetlana jest lista rankingowa miejsc (Rys.3.9) składająca się z tych samych kafelków co w widoku mapy i tak samo jak tam po kliknięciu w niego użytkownik zostaje przekierowany do strony ze szczegółami. Domyślnie miejsca są sortowane według najlepszej oceny, aczkolwiek istnieje możliwość przełączenia widoku na listę ostatnio dodanych.



Rysunek 3.9 Widok rankingów

3.6. Widok profilu użytkownika

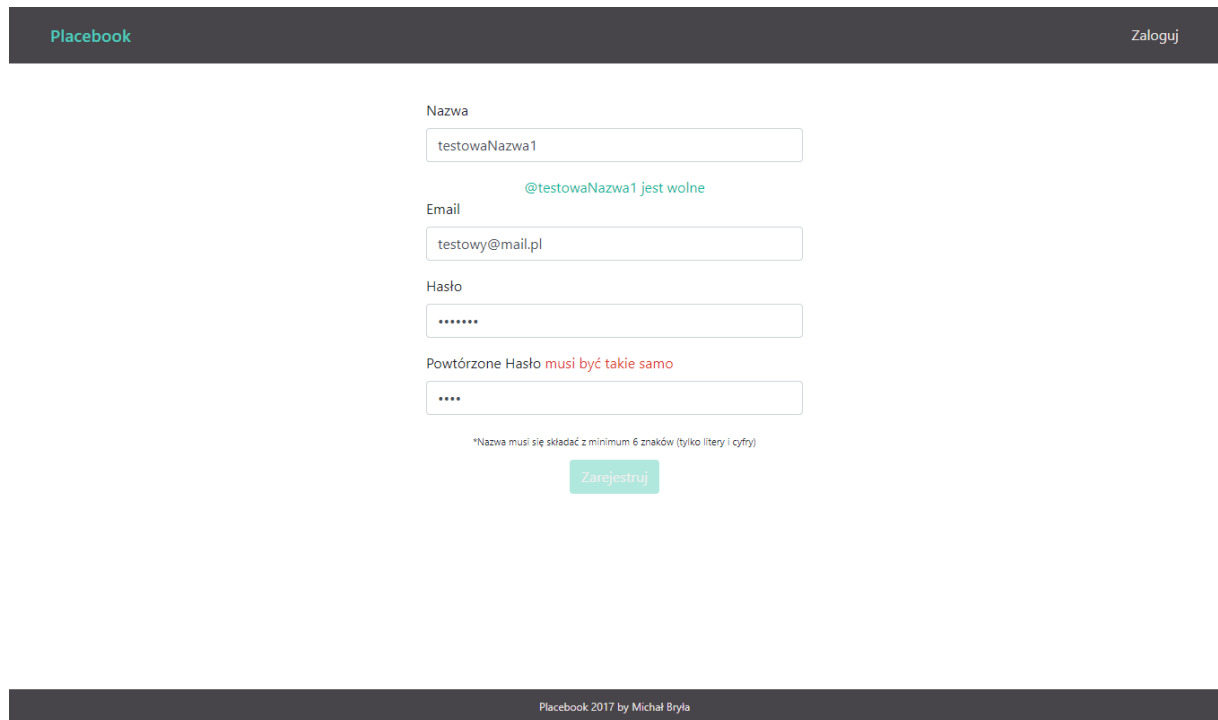
Przedstawia on podstawowe informacje o użytkowniku (Rys.3.10). Znajdują się tam jego nazwa i adres email, a także dodane i ulubione miejsca prezentowane w postaci kafelków.



Rysunek 3.10 Widok profilu użytkownika

3.7. Widok rejestracji

Udostępnia możliwość dodania nowego konta do aplikacji, poprzez wypełnienie wszystkich zaprezentowanych w poniższym widoku (Rys.3.11) pól edycyjnych. Dodatkowo pod nazwą znajduje się etykieta, która mówi użytkownikowi czy wpisana przez niego nazwa jest dostępna, która jest dynamicznie aktualizowana podczas wpisywania.



The registration form is displayed within a dark header bar. The header contains the 'Placebook' logo on the left and a 'Zaloguj' (Login) link on the right. The form itself is centered and consists of several input fields and a submit button. The 'Nazwa' (Name) field contains 'testowaNazwa1', and a green message '@testowaNazwa1 jest wolne' (available) is shown below it. The 'Email' field contains 'testowy@mail.pl'. The 'Hasło' (Password) field contains six asterisks, and the 'Powtórzone Hasło' (Repeat Password) field contains four asterisks, with a red message 'musi być takie samo' (must be the same) between them. A small asterisk note at the bottom states: '*Nazwa musi się składać z minimum 6 znaków (tylko litery i cyfry)'. A green 'Zarejestruj' (Register) button is located at the bottom of the form. The footer of the page shows 'Placebook 2017 by Michał Boryla'.

Placebook

Zaloguj

Nazwa

testowaNazwa1

@testowaNazwa1 jest wolne

Email

testowy@mail.pl

Hasło

Powtórzone Hasło musi być takie samo

*Nazwa musi się składać z minimum 6 znaków (tylko litery i cyfry)

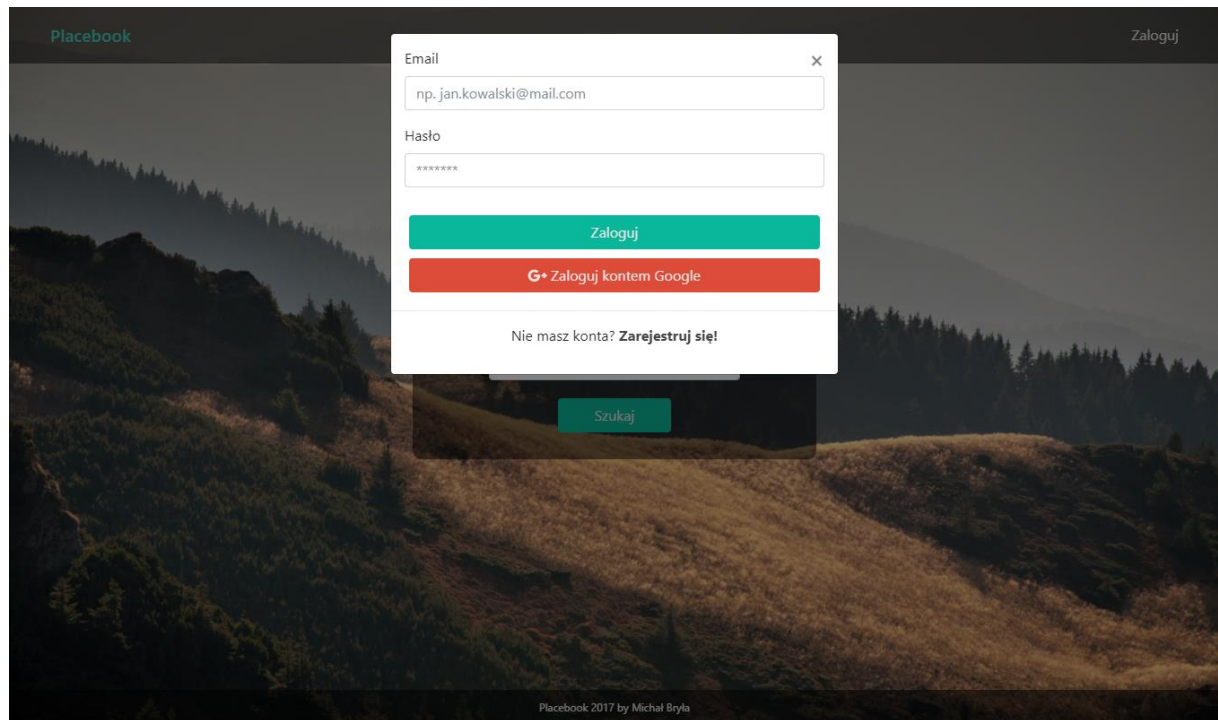
Zarejestruj

Placebook 2017 by Michał Boryla

Rysunek 3.11 Widok rejestracji

3.8. Widok logowania

Umożliwia zalogowanie się do aplikacji poprzez wypełnienie pól edycyjnych dotyczących adresu email oraz hasła, a także poprzez zewnętrzne konto Google. W przypadku braku konta użytkownik ma możliwość założenia nowego konta poprzez kliknięcie etykiety „Zarejestruj się”.



Rysunek 3.12 Widok logowania

4. Implementacja

4.1. Środowisko i narzędzia programistyczne

Środowiskiem programistycznym wykorzystanym do napisania aplikacji był **WebStorm** firmy JetBrains. Do zautomatyzowania zadań związanych z rozwijaniem kodu aplikacji wykorzystano narzędzie **Angular CLI**, natomiast do zarządzania pakietami niezbędnymi do tworzenia aplikacji posłużono się narzędziem **yarn**. Do napisania logiki aplikacji użyto języka **TypeScript**, który jest nadzbiorem JavaScriptu. Wykorzystano go głównie ze względu na jego rozszerzenie o statyczne typowanie. Kod starano się napisać zgodnie z wskazówkami zawartymi w książce autorstwa Roberta C. Martina „Clean Code” [2]. Do napisania widoków interfejsu wykorzystano języki **HTML5** oraz **SCSS**, który rozszerza możliwości CSS3. Podczas projektowania i tworzenia diagramów wykorzystano narzędzie **Visual Paradigm** oraz aplikację webową **Creately** [17]. Markery wykonano przy użyciu **Adobe Photoshop CS6**.

4.2. Użyte biblioteki i komponenty

Następujące biblioteki programistyczne i komponenty zostały wykorzystane w czasie rozwoju aplikacji:

- **Angular 4** - Platforma ułatwiająca pisanie aplikacji webowych w języku JavaScript
- **RxJs** – Biblioteka do programowania reaktywnego, opierająca się na wykorzystaniu Observables
- **AGM – Angular Google Maps** – Zbiór komponentów służących do obsługi Google Maps
- **ng2-image-upload** – Mała biblioteka udostępniająca komponenty do wgrywania zdjęć
- **ngx-gallery** – Zestaw komponentów do tworzenia galerii zdjęć
- **Bootstrap 4** – Biblioteka komponentów służących do tworzenia responsywnych aplikacji webowych

4.3. Wykorzystane wzorce projektowe

Podczas implementacji projektu wykorzystano następujące wzorce projektowe:

- **Obserwator** – wzorec projektowy wykorzystany do wymiany danych pomiędzy klientem, a bazą danych, polegający na powiadamianiu zainteresowanych obiektów o zmianach stanu innego obiektu.
- **Wstrzykiwanie zależności** – wzorec projektowy i architektury oprogramowania, polegający na przekazywaniu (wstrzykiwaniu) gotowych instancji obiektów, udostępniających swoje metody i właściwości.

4.4. Wymiana danych

Połączenie pomiędzy klientem aplikacji, a serwerem w chmurze odbywa się przy pomocy zewnętrznej biblioteki AngularFire [10]. Jest to biblioteka ułatwiająca tworzenie aplikacji korzystających z usług Firebase, pisanych we frameworku Angular [9]. Pozwala na pracę z Cloud Firestore, która jest najnowszą (faza beta testów) bazą danych czasu rzeczywistego, której ogromną zaletą jest efektywność, gdyż czas wykonywania zapytania nie jest zależny od zbioru danych na którym jest ono wywoływane, a zbiorze wynikowym. Cała wymiana danych odbywa się asynchronicznie, strumieniowo, z wykorzystaniem RxJs [8], biblioteki służącej do reaktywnego programowania, gdzie kluczową rolę odgrywają „Observables”, na które się nasłuchuje.

W aplikacji tworzone są referencje do kolekcji oraz dokumentów z bazy, na których wykonuje się operacje **CRUD** (od ang. create, read, update and delete). Poniżej omówiono każdą z nich i przedstawiono ich implementacje na przykładzie komentarzy miejsc.

Dodawanie nowych dokumentów do bazy polega na wywołaniu metody `add()` na kolekcji, która zwraca JavaScriptowy Promise, zawierający referencje do nowo utworzonego obiektu.

```
public addComment(pid: string, comment: Comment): void {
    this.afs.collection<Comment>(`places/${pid}/comments`).add(comment)
        .then((doc) => {
            let commentToSetId: Comment = comment;
            commentToSetId.cid = doc.id;
            this.afs.doc<Comment>(`places/${pid}/comments/${doc.id}`)
                .set(commentToSetId)
        })
        .catch((error) => console.log(error))
}
```

Edytowanie polega na wywołaniu metody `set()` lub `update()`, na dokumencie. Obie nadpisują dokument o nowy przekazany w parametrze i zwracają Promise, natomiast różnią się tym, że `update` w przypadku wykonaniu metody na nieistniejącym dokumencie wyrzuci błąd, podczas gdy `set()` utworzy nowy dokument.

```
public editComment(pid: string, commentId: string, comment: Comment): void
{
    this.afs.doc<Comment>(`places/${pid}/comments/${commentId}`)
        .update(comment)
        .catch((error) => console.log(error))
}
```

Usuwanie wymaga użycia metody `delete()` na dokumencie i również zwraca Promise.

```
public deleteComment(pid: string, commentId: string): void {
    this.afs.doc<Comment>(`places/${pid}/comments/${commentId}`)
        .delete()
        .catch((error) => console.log(error))
}
```

Odczytywanie polega na wywołaniu metody `valueChanges()`, na kolekcji bądź dokumencie, która zwraca nie Promise a Observable. Dzięki temu można nasłuchiwać na zmiany w danej kolekcji lub dokumencie.

```
public getPlaceComments(pid: string): Observable<Comment[]> {
    return this.afs.collection(`places/${pid}/comments`, ref =>
    ref.orderBy('date', 'desc')).valueChanges()
}
```

W miejscu gdzie pobieramy dane, tworzona jest Observable oraz Subscription. Nasłuchiwanie na zmiany polega na wywołaniu metody `subscribe()`, na wspomnianej wcześniej Observable, dzięki czemu jest się informowanym na bieżąco o każdej zmianie. Wewnątrz można dowolnie modyfikować przychodzące dane. Trzeba pamiętać o odsubskrybowaniu, w momencie kiedy nie chce się już otrzymywać informacji, poprzez wywołanie `unsubscribe()` na Subscription.

```
public comments$: Observable<Comment[]>;
public commentsSubscription: Subscription;

this.comments$ = this.placeService.getPlaceComments(this.pid);
this.commentsSubscription = this.comments$.subscribe((comments) => {
    this.comments = comments;
    comments.map((comment) => {
        this.comments.editing = false;
    })
});

this.commentsSubscription.unsubscribe();
```

4.5. Problemy implementacyjne

Podczas rozwijania aplikacji napotkano na następujące problemy:

Lp.	PI-1
Nazwa	Edytowanie miejsc
Opis problemu	Początkowo aplikacja miała posiadać tylko jeden typ użytkownika, niestety z racji faktu iż to właśnie użytkownicy wypełniają bazę miejscami koniecznym było dodanie weryfikacji dodawanych danych. O ile to można było rozwiązać prostym algorytmem, który na podstawie liczby ocen i średniej decydowałby czy miejsce zasługuje na zatwierdzenie, tak przy edytowaniu miejsc nie jest to takie proste, ponieważ miejsce może mieć wiele wersji na raz czekających na zatwierdzenie.
Rozwiązanie	Postanowiono dodać do aplikacji drugi typ użytkownika, mianowicie Moderadora, który posiada poszerzony zakres możliwości między innymi o zarządzanie aktualizacjami miejsc

Tabela 4.1 PI-1 – Edytowanie miejsc

Lp.	PI-2
Nazwa	Filtrowanie wyników
Opis problemu	Z racji wykorzystania najnowszej bazy danych dostarczonej przez Firebase, czyli Cloud Firestore znajdującą się jeszcze w beta testach, nie obsługuje ona na tę chwilę zapytań do bazy z parametrem OR.
Rozwiązanie	Wymyślono dwa rozwiązania. Jedno polegało na wysyłaniu wielu zapytań i łączeniu wyników w jedną listę, jednak po szybkiej analizie wywnioskowano iż jest to bardzo kosztowne rozwiązanie, ze względu na pomnożenie ilości zapytań w zależności od parametrów. Drugim rozwiązaniem było filtrowanie wyników po stronie klienta. Wadą jest obciążenie klienta i przeglądarki. Reasumując zalety metod wybrano opcję z filtrowaniem po stronie klienta, gdyż generuje to znacznie mniejsze koszty utrzymania aplikacji.

Tabela 4.2 PI-2 – Filtrowanie wyników

Lp.	PI-3
Nazwa	Dodawanie miejsc do ulubionych.
Opis problemu	Kiedy użytkownik dodaje miejsce do ulubionych trzeba zapisać gdzieś tą informację. Są dwie możliwości, albo w obiekcie miejsca co nie wydaje się najlepszym rozwiązaniem w przypadku kiedy atrakcja jest lubiana przez dużą liczbę użytkowników, albo w obiekcie użytkownika.
Rozwiązanie	W bazie dodano do dokumentu użytkownika, kolekcje ulubionych miejsc.

Tabela 4.3 PI-3 – Dodawanie miejsc do ulubionych

Lp.	PI-4
Nazwa	Sprawdzanie unikalności nazwy użytkownika.
Opis problemu	Sprawdzanie odbywa się przy każdej interakcji użytkownika z polem edycyjnym do którego wpisuje nazwę, zatem przy każdym naciśnięciu klawisza. Dane przychodzą asynchronicznie przy każdym takim naciśnięciu dlatego jeżeli użytkownik szybko wpisze nazwę istnieje możliwość, że podczas wpisywania dostanie nieaktualną informację o dostępności nazwy.
Rozwiązanie	Rozwiązaniem problemu było stworzenie „Subject”, obiektu z biblioteki do reaktywnych rozszerzeń RxJs, który jest strumieniem do którego wrzucano wartości wpisane przez użytkownika, a następnie nałożono na niego „debounceTime”, które działa jak opóźnienie pomiędzy emisjami kolejnych wartości z tym, że przekazuje jedynie najświeższą wartość z każdego bloku czasowego, co dodatkowo poprawia efektywność i zmniejsza ilość zapytań. <pre> this.inputSubject .debounceTime(100) .switchMap((input) => this.authService.checkUsername(input)) </pre>

Tabela 4.4 PI-4 – Sprawdzanie unikalności nazwy użytkownika

Lp.	PI-5
Nazwa	Zarządzanie komentarzami.
Opis problemu	Dodając dokument do bazy Cloud Firestore, są dostępne dwie opcje: wywołanie metody add na kolekcji z podanym w parametrze obiektem oraz wywołanie metody set na dokumencie z podanym w parametrze obiektem, która nadpisuje wybrany dokument lub tworzy nowy w przypadku braku jego istnienia. Do zarządzania komentarzami, czyli ich edytowaniu i usuwaniu potrzebny jest unikalny identyfikator by móc odwołać się do konkretnego elementu. W pierwszym przypadku wspomniane ID jest generowane automatycznie, natomiast w drugim musimy sami je zadeklarować. Niestety dostając z bazy listę komentarzy nie posiadamy ich identyfikatorów, jedynie listę obiektów jakimi są komentarze, co uniemożliwiało zarządzanie nimi.
Rozwiązanie	<p>By korzystać z automatycznie generowanych identyfikatorów, do dodawania nowych komentarzy skorzystano z metody add, która zwraca JavaScriptowy Promise, posiadający referencje do utworzonego w bazie dokumentu, co za tym idzie również jego ID, które wykorzystano do nadpisania nowo utworzonego komentarza o jego identyfikator, dzięki czemu przy pobieraniu listy komentarzy, zawierają one pola z identyfikatorem.</p> <pre> public addComment(pid: string, comment: Comment): void { this.afs.collection<Comment>(`places/\${pid}/comments`) .add(comment) .then((doc) => { let commentToSetId: Comment = comment; commentToSetId.cid = doc.id; this.afs.doc<Comment>(`places/\${pid}/comments/\${doc.id}`) .set(commentToSetId) }) .catch((error) => console.log(error)) } </pre>

Tabela 4.5 PI-5 – Zarządzanie komentarzami

Lp.	PI-6
Nazwa	Unikalne nazwy użytkowników
Opis problemu	Każdy użytkownik musi posiadać swoją unikatową nazwę. Przy rejestracji nowego konta podaje on wymyśloną przez siebie nazwę, natomiast przy korzystaniu z konta Google, dostajemy obiekt użytkownika posiadający już pole „Display Name”. Niestety wartości tego pola nie są unikalne dla każdego użytkownika przez co nie można go wykorzystać w tym celu.
Rozwiązanie	Po wybraniu uwierzytelniania kontem Google, przy pierwszym takim logowaniu wyświetlane jest okno, gdzie należy wpisać unikalną nazwę użytkownika.

Tabela 4.6 PI-6 – Unikalne nazwy użytkowników

Lp.	PI-7
Nazwa	Zarządzanie pakietami.
Opis problemu	Początkowo, przy rozwijaniu aplikacji do zarządzania pakietami wykorzystywano najpopularniejsze narzędzi do tego typu pracy jakim jest npm. Niestety po pewnym czasie przestał on działać co uniemożliwiało dalszą pracę, gdyż nie można było instalować kolejnych pakietów.
Rozwiązanie	Po głębszej analizie przyczyny nie znaleziono i postanowiono zmienić narzędzie na yarn, które służy do tego samego, a ponadto okazało się szybsze od npm'a.

Tabela 4.7 PI-7 – Zarządzanie pakietami

4.6. Plany rozwoju aplikacji

Produkt końcowy służący do tworzenia i przeglądania bazy niezwykłych miejsc w Polsce zapewnia wszystkie niezbędne funkcjonalności potrzebne do komfortowego korzystania z niego. Można go rozbudować o :

- Wprowadzenie nagradzania aktywnych użytkowników np. poprzez odznaki które widniałyby na ich profilach.
- Poprawienie interfejsu moderatora w widoku szczegółów miejsca
- Wprowadzenie możliwości logowania się za pomocą konta Facebook
- Zwiększenie ilości rankingów
- Dodanie możliwości komentowania profili innych użytkowników

5. Podsumowanie

Celem pracy inżynierskiej, było nie tylko sformułowanie wymagań, zaprojektowanie i zaimplementowanie aplikacji, służącej do tworzenia bazy niezwykłych miejsc w Polsce, ale również chęć podjęcia pracy w najnowszych technologiach dostępnych na rynku.

W trakcie pracy nad aplikacją przeanalizowano istniejące już na rynku rozwiązania. Zwrócono szczególnie uwagę na ich funkcjonalności oraz interfejs użytkownika, który w wielu przypadkach potrafi skreślić dobrze zaprojektowaną aplikację. Wypisano ich największe zalety oraz wady i podstawie tych informacji postanowiono stworzyć nowy produkt, posiadający najlepsze rozwiązania.

Wszystkie założenia implementacyjne zostały spełnione, natomiast przypadki użycia oraz przypisane im scenariusze, mają odzwierciedlenie w końcowej aplikacji. Zdecydowana większość napotkanych problemów, była spowodowana brakiem doświadczenia oraz faktem iż korzystano z najnowszych technologii i narzędzi, które nie zawsze posiadały pełną dokumentację i nie były w stu procentach stabilne. Aplikacja została przetestowana na różnych rozdzielczościach pod kątem responsywności i różnych środowiskach przeglądarkowych. Zawarto możliwe plany rozwoju aplikacji, które zaprezentowano w punkcie 4.6.

Realizacja projektu inżynierskiego związana jest z rozstrzygnięciem problemu braku obecnie na rynku rozwiązania zadowalającego wymagającego użytkownika. Praca opierała się na wykorzystaniu nowoczesnych technologii i skupieniu się na jak najlepszym komforcie korzystania dla użytkownika co musi zagwarantować sukces.

Bibliografia

- [1] S. Krug, Don't make me think, revisited: A common sense approach to web usability., 2014.
- [2] R. Martin, Clean Code (1st ed.), Prentice Hall, 2008.
- [3] B. Rady, Serverless single page apps: Fast, scalable, and available, 2016.
- [4] R. Miles, K. Hamilton, R. Szpoton i G. W. Helion, UML 2.0 : Wprowadzenie, Gliwice: Helion, 2007.
- [5] Y. Moiseev, Angular 2 Development with TypeScript (1st ed.), Manning Publications, 2016.
- [6] P. Krill, „Google's Firebase taps serverless Cloud Functions,” 22 5 2017. [Online].
<https://www.infoworld.com/article/3197745/application-development/googles-firebase-taps-serverless-cloud-functions.html>.
[Data uzyskania dostępu: 20 11 2017].
- [7] J. Wroński, „Cudze chwalicie, swego nie znacie. Oto najbardziej niezwykle i tajemnicze miejsca w Polsce,” 6 5 2015. [Online].
<https://www.fly4free.pl/oto-najbardziej-niezwykly-i-tajemnicze-miejsca-w-polsce/>.
[Data uzyskania dostępu: 22 10 2017].
- [8] „RxJs, Dokumentacja techniczna,” [Online].
<http://reactivex.io/rxjs/>.
[Data uzyskania dostępu: 20 11 2017].
- [9] „Angular 4, Dokumentacja techniczna,” [Online].
<https://angular.io/docs>.
[Data uzyskania dostępu: 10 11 2017].
- [10] „AngularFire, Dokumentacja techniczna,” [Online].
<https://github.com/angular/angularfire2>.
[Data uzyskania dostępu: 14 11 2017].
- [11] „Angular Google Maps, Dokumentacja techniczna,” [Online].
<https://angular-maps.com/api-docs/>.
[Data uzyskania dostępu: 10 10 2017].
- [12] „Google Firebase, Dokumentacja techniczna,” [Online].
<https://firebase.google.com/docs/>.
[Data uzyskania dostępu: 18 11 2017].

- [13] „Google Maps, Dokumentacja techniczna,” [Online].
<https://developers.google.com/maps/documentation/javascript/>.
[Data uzyskania dostępu: 15 10 2017].
- [14] „Bootstrap 4, Dokumentacja techniczna,” [Online].
<https://getbootstrap.com/docs/4.0/getting-started/introduction/>.
[Data uzyskania dostępu: 6 11 2017].
- [15] „Google Maps,” [Online].
<https://www.google.pl/maps/>.
[Data uzyskania dostępu: 25 11 2017].
- [16] „Atlaso,” [Online].
<http://www.atlaso.pl>.
[Data uzyskania dostępu: 2 10 2017].
- [17] „Creately,” [Online].
<https://creately.com/>.
[Data uzyskania dostępu: 23 11 2017].
- [18] „Font Awesome,” [Online].
<http://fontawesome.io/icons/>.
[Data uzyskania dostępu: 16 11 2017].
- [19] „Unsplash,” [Online].
<https://unsplash.com/>.
[Data uzyskania dostępu: 2 10 2017].

Spis ilustracji

Rysunek 1.1 Aplikacja „Google Maps”	5
Rysunek 1.2 Wpis na stronie fly4free.pl	6
Rysunek 1.3 Aplikacja „atlaso”, widok mapy.....	7
Rysunek 1.4 Aplikacja „atlaso”, widok listy.....	8
Rysunek 2.1 Diagram przypadków użycia aplikacji	10
Rysunek 2.2 Uproszczona ilustracja architektury serverless z wykorzystaniem „Google Firebase” https://firebase.google.com/products/firestore/	18
Rysunek 2.3 Schemat bazy danych	19
Rysunek 3.1 Strona startowa aplikacji	21
Rysunek 3.2 Widok mapy	22
Rysunek 3.3 Kategoria: Pomnik przyrody	22
Rysunek 3.4 Kategoria: Zabytek	22
Rysunek 3.5 Kategoria: Krajobraz	22
Rysunek 3.6 Kategoria: Inne	22
Rysunek 3.7 Widok szczegółów miejsca	24
Rysunek 3.8 Widok dodawania/edytowania miejsca	25
Rysunek 3.9 Widok rankingów	26
Rysunek 3.10 Widok profilu użytkownika	27
Rysunek 3.11 Widok rejestracji	28
Rysunek 3.12 Widok logowania	29

Spis tabel

Tabela 2.1 PU-1 Wyszukiwanie miejsc	11
Tabela 2.2 PU-2 – Filtrowanie wyników	11
Tabela 2.3 PU-3 – Przeglądanie miejsca.....	11
Tabela 2.4 PU-4 – Przeglądanie rankingów	12
Tabela 2.5 PU-5 – Przeglądanie profili innych użytkowników	12
Tabela 2.6 PU-6 – Logowanie.....	13
Tabela 2.7 PU-7 – Rejestracja.....	13
Tabela 2.8 PU-8 – Dodanie nowego miejsca	14
Tabela 2.9 PU-9 – Dodanie aktualizacji miejsca	14
Tabela 2.10 PU-10 – Komentowanie miejsca	15
Tabela 2.11 PU-11 – Ocena miejsca	15
Tabela 2.12 PU-12 – Dodanie/usunięcie miejsca z ulubionych	15
Tabela 2.13 PU-13 – Przeglądanie swojego profilu.....	16
Tabela 2.14 PU-14 – Edytowanie swojego profilu	16
Tabela 2.15 PU-15 – Edytowanie miejsca	16
Tabela 2.16 PU-16 – Zatwierdzenie aktualizacji miejsca	17
Tabela 2.17 PU-17 – Usunięcie aktualizacji miejsca	17
Tabela 2.18 PU-18 – Usunięcie miejsca	17
Tabela 2.19 PU-19 – Zatwierdzenie miejsca.....	18
Tabela 4.1 PI-1 – Edytowanie miejsc.....	32
Tabela 4.2 PI-2 – Filtrowanie wyników	33
Tabela 4.3 PI-3 – Dodawanie miejsc do ulubionych	33
Tabela 4.4 PI-4 – Sprawdzanie unikalności nazwy użytkownika	33
Tabela 4.5 PI-5 – Zarządzanie komentarzami.....	34
Tabela 4.6 PI-5 – Unikalne nazwy użytkowników	34
Tabela 4.7 PI-7 – Zarządzanie pakietami	35