Specyfikacja wymagań projektu z Inżynierii Oprogramowania

Lizaveta Hurskaya, Bartłomiej Kręgielewski, Szymon Lepianka, Małgorzata Pinior

1. Wprowadzenie

1.1 Cel

Zadaniem aplikacji jest wyznaczenie tras do zwiedzania po wcześniejszym uwzględnieniu wybranych przez użytkownika punktach zwiedzania. Użytkownik ma do wyboru listę obiektów wartych odwiedzenia wraz z krótkim opisem, dzięki czemu będzie mógł zadecydować, co chce umieścić na swojej liście. Aplikacja umożliwia również udostępnianie tras dla innych użytkowników, ich ocenianie i komentowanie oraz dodawanie obiektów do zwiedzania do mapy.

1.2 Wybrane grono odbiorców

Przewidywanymi odbiorcami systemu są osoby posiadające smartfona z systemem operacyjnym Android, aby móc pobrać aplikację na telefon i móc korzystać z niej w trakcie zwiedzania czy też planowania podróży.

Użytkownicy:

1) Osoby odwiedzające dane miasto
2) Właściciele obiektów do odwiedzenia
3) Promotorzy miast

1.3 Definicje

1) Logowanie przez aplikację zewnętrzną - logowanie do aplikacji poprzez zewnętrzną aplikację z wykorzystaniem protokołu OAuth 2.0
2) Obiekt - miejsce do odwiedzenia, zaznaczone na mapie, dodane do mapy.
3) Trasa - uporządkowana lista obiektów do odwiedzania.
4) Mapa - mapa z Google Maps

2. Opis

2.1 Potrzeby użytkowników

Z myślą o osobach, które nie mają czasu na szukanie obiektów do zwiedzania, powstał pomysł na tą aplikację. Dzięki możliwości udostępniania tras, każdy może zaplanować sobie zwiedzanie i w trakcie mieć możliwość czytania informacji o danym obiekcie. Poprzez komentowanie i ocenianie tras można pomóc innym wybrać najlepszą drogę dla siebie. Dodawanie nowych obiektów jest dobrą opcją dla turystyki i rozwoju danego miasta, gdyż nie wszystkie ciekawe miejsca można znaleźć na Google Maps. Z tego powodu aplikacja spełnia potrzeby nie tylko docelowych turystów, ale również osób, które chcą rozpromować swój obiekt.

2.2 Założenia i zależności

Głównymi odbiorcami aplikacji są osoby posiadające telefony z systemem Android, co jest głównym założeniem. Dodatkowo należy posiadać połączenie z internetem oraz włączoną lokalizację GPS, aby usprawnić funkcjonowanie aplikacji.

3. Funkcje i wymagania systemu

3.1 Wymagania funkcjonalne

1) Ekran logowania

   Ekran logowania zawiera okno do wprowadzenia emaila i hasła, przycisk do logowania się, przycisk do utworzenia konta i przycisk do logowania się przez aplikację zewnętrzną. Użytkownik może zalogować się, wykorzystując swoje konto w aplikacji zewnętrznej lub podając e-mail i hasło i przejść do własnego panelu w aplikacji. W przypadku, gdy użytkownik nie posiada konta w systemie, może stworzyć nowe.

2) Tworzenie nowego konta

   Dla utworzenia nowego konta nie używając aplikacji zewnętrznej użytkownik podaje imię, nazwisko, e-mail i hasło. Po poprawnym utworzeniu konta przechodzi do panelu użytkownika.

3) Panel użytkownika

   W panelu użytkownika znajdują się dane profilowe użytkownika, przyciski do wylogowania się i do usunięcia konta.

4) Moje trasy

   Ekran, zawierający listę dodanych lub utworzonych przez użytkownika tras do zwiedzania. Po kliknięciu na trasę wyświetla się informacja o niej i przycisk “Zobacz na mapie”. Jest możliwość utworzenia nowej trasy.

5) Ulubione miejsca

   Ekran przedstawiający listę wybranych przez użytkownika miejsc z możliwością wyświetlenia ich jako punktów na mapie.

6) Lista tras

   Ogólnodostępna lista wszystkich tras, stworzonych przez użytkowników aplikacji. Zawiera nazwy, opisy, oceny i komentarze do tras. Możliwe jest wybieranie tras pasujących do kryteriów wyszukiwania.

7) Informacje o trasie

   Ekran, który zawiera informacje o trasie, oceny, komentarze i reprezentuje trasę na mapie. Posiada przycisk który pozwala dodać daną trasę do listy swoich tras. Użytkownik ma możliwość komentowania danej trasy lub jej oceniania.

8) Moja trasa

   Ekran, który zawiera informacje o trasie, tworzonej przez użytkownika i reprezentuje trasę na mapie. Posiada przycisk, który pozwala wybierać tryb udostępnienia (prywatny/publiczny).

9) Informacje o obiekcie

   Ekran, reprezentujący obiekt, jego opis, ocenę i komentarze oraz położenie na mapie. Umożliwia dodanie obiektu do listy ulubionych lub do którejś z tras tworzonych przez danego użytkownika. Użytkownik również może skomentować lub ocenić dany obiekt.

10) Dodaj obiekt

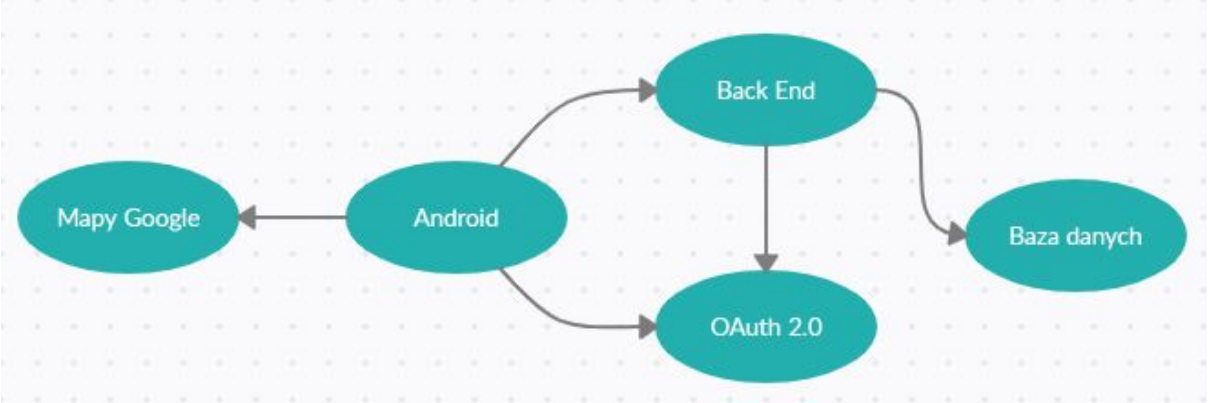
Ekran, służący do dodawania nowych obiektów do bazy. Zawiera mapę, na której wybiera się punkt oraz pole dla opisu tego obiektu.

11) Pomoc

Ekran zawiera informacje o tym, jak należy korzystać z aplikacji wraz ze zrzutami ekranu dla każdej możliwej do wykonania akcji.

3.2 Zewnętrzne wymagania interfejsu

Aplikacja korzysta z Google Maps, swoje dane pobiera z API (back channel). Dane przechowywane są na serwerze, gdzie zapisujemy wszystkich użytkowników i ich dane, a także informacje o obiektach i trasach - wszystkie zapisane informacje. Dodatkowym elementem jest możliwość logowania przez zewnętrzną aplikację.



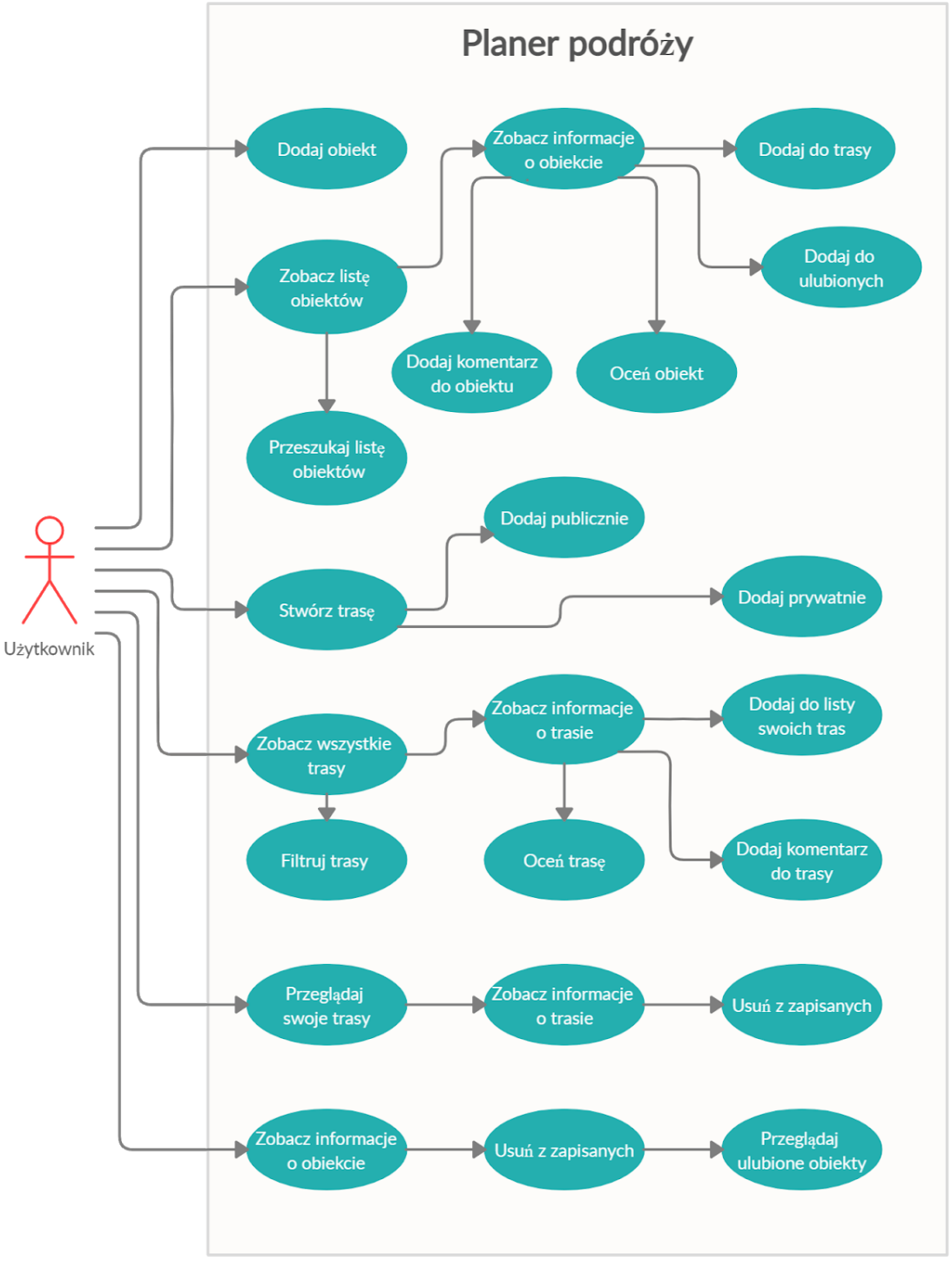
3.3 Funkcjonalności systemu

3.3.1 Dostępne funkcje:

- Utwórz konto
- Usuń konto
- Zaloguj się
- Zaloguj się przez konto w aplikacji zewnętrznej
- Wyloguj się
- Dodaj nowy obiekt do zwiedzania
- Oceń obiekt
- Zostaw komentarz do obiektu
- Dodaj obiekt do ulubionych
- Usuń obiekt z ulubionych
- Pokaż mapę
- Wybierz obiekty pasujące do kryteriów wyszukiwania
- Stwórz nową trasę

- Dodaj opis trasy
- Dodaj obiekt do swojej trasy
- Usuń trasę
- Podziel się trasą - upublicznij lub zostaw prywatną
- Oceń trasę
- Zostaw komentarz do trasy
- Wybierz trasy pasujące do kryteriów wyszukiwania

3.4 Przypadki użycia



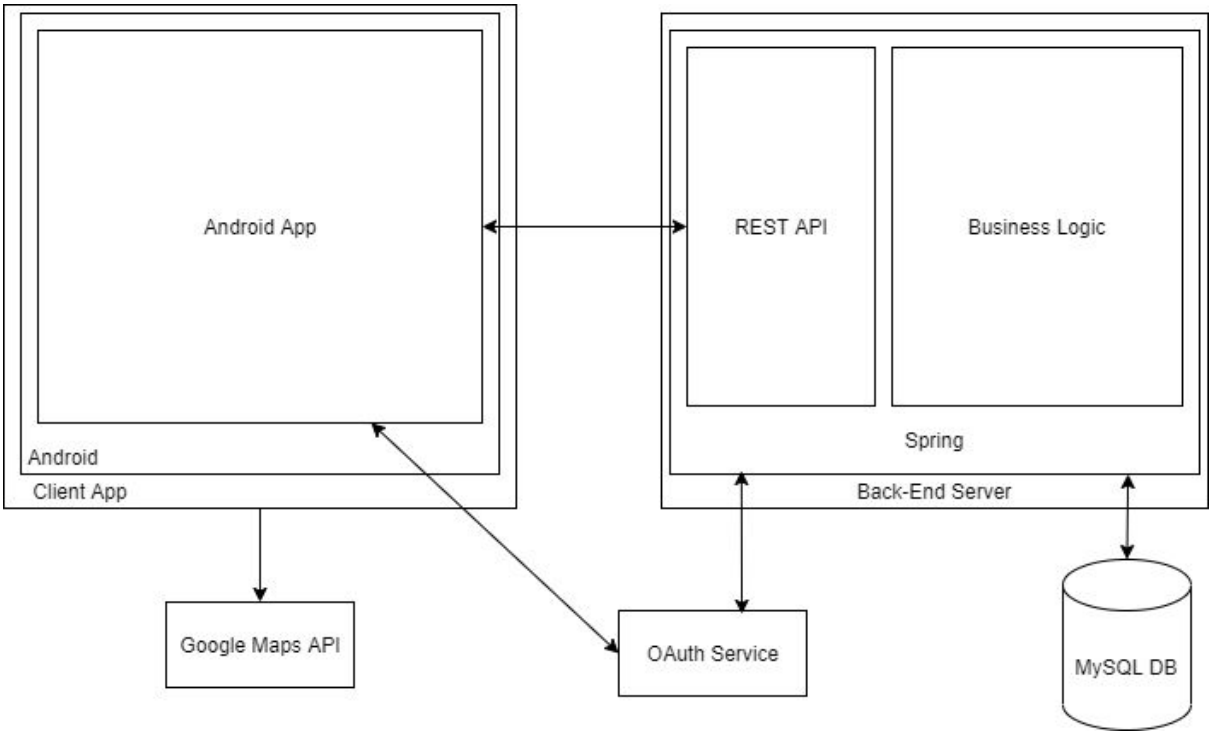
3.5 Wymagania niefunkcjonalne

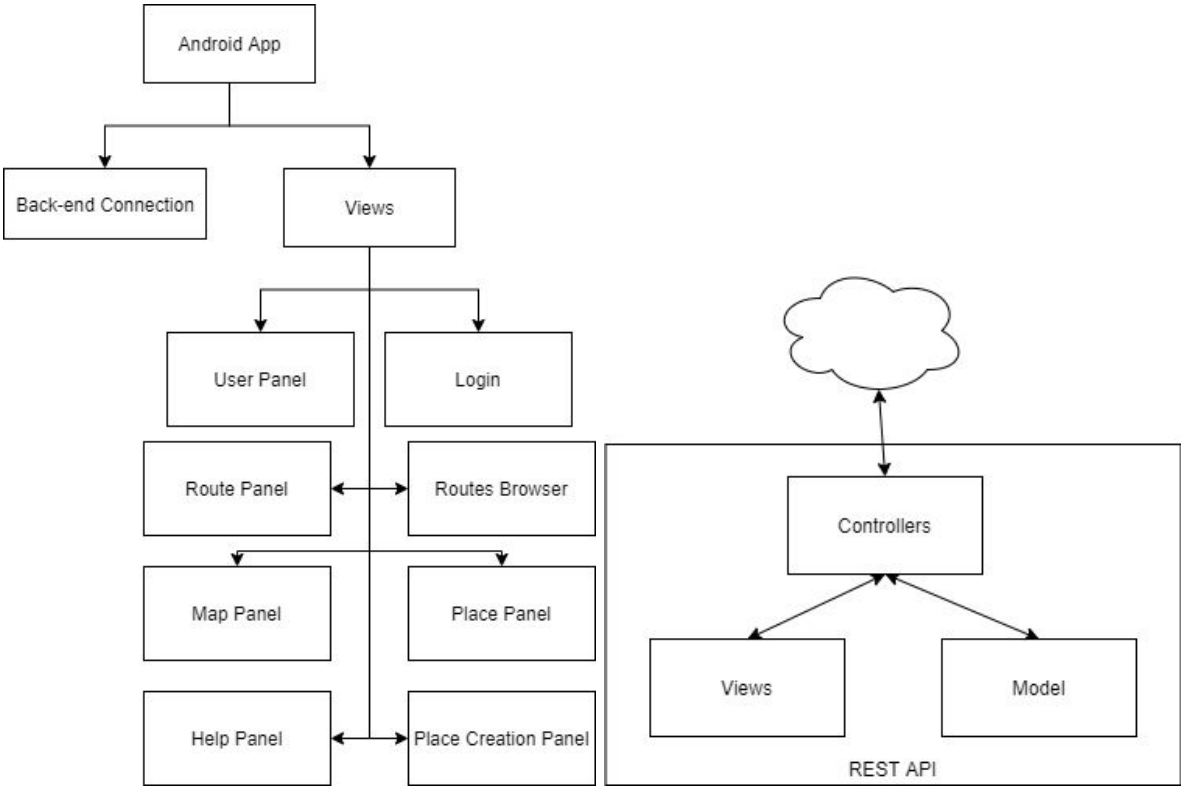
1) Wydajność - Załadowanie zaplanowanej trasy w zależności od warunków powinno nastąpić w czasie poniżej:
    a) 4 sekundy - localhost z dobrym połączeniem do map Google (sieć 300 mbps)
    b) 10 sekund - test na przeciętnym telefonie w warunkach miejskich

   Płynny interfejs
    a) Zadowolenie użytkowników - 8+ w skali 1-10
2) Intuicyjność - aplikacja łatwa i przyjemna w użytkowaniu, intuicyjna. Dodatkowo przygotowany jest poradnik do obsługi zawierający zrzuty ekranu prezentujące podstawowe operacje. Zakłada się użycie barw z ciepłej tonacji w motywie jasnym. Korzystanie z standardowych ikon w systemie Android, które są uniwersalne.
    a) Zadowolenie użytkowników - 8+ w skali 1-10
3) Kompatybilność - aplikacja współpracuje z Google Maps oraz jest skierowana na platformę Android.
    a) Kompatybilność z Android 6.0 (Marshmallow - API Level 23) i nowszymi

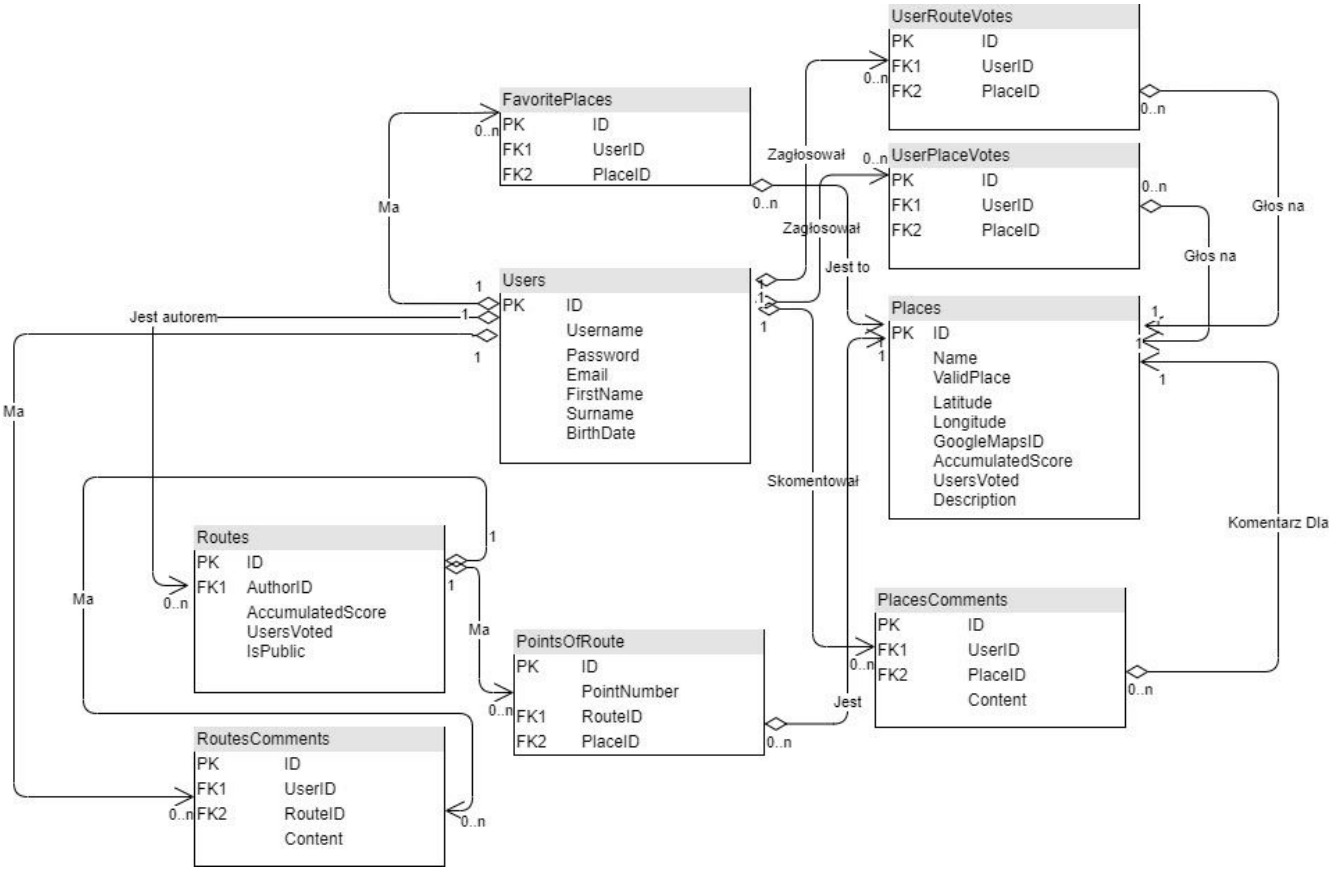
4. Projekt systemu

4.1 Architektura systemu





4.2 Struktura bazy danych



4.3 Opis interfejsów

1. Android App - aplikacja mobilna przeznaczona do uruchamiania na urządzeniu z systemem Android.
2. REST API - styl architektoniczny interfejsu aplikacji (API), który używa żądań HTTP do uzyskiwania dostępu do danych i korzystania z nich. Dane te mogą służyć do GET, PUT, POST i DELETE. Możliwe operacje dotyczące zasobów: odczyt, aktualizacja, tworzenie i usuwanie.
3. Business Logic - element odpowiedzialny za przetwarzanie danych, określający sposób tworzenia, przechowywania i zmieniania danych.
4. Spring - framework zapewniający wszechstronny model programowania i konfiguracji dla nowoczesnych aplikacji opartych na języku Java
5. MySQL DataBase - system zarządzania relacyjnymi bazami danych oparty na języku SQL
6. Google Maps API - interfejs służący do korzystania z Google Maps.
7. OAuth Service - zewnętrzny serwis służący do autoryzacji i uwierzytelniania kont. Pozwala właścicielom zasobów autoryzować dostęp osób trzecich do zasobów serwera bez udostępniania ich poświadczeń. Pakiet wykorzystuje standard OAuth 2.0.

4.4 Stos Technologiczny

- Android 6.0
- Baza Danych MySQL
- Spring 5.0
- JDK 15

4.5 Projekt testów

1. Rodzaj testów: jednostkowe
   Technologia: JUnit
   Testowany podmiot:
   a. metody Client App
   b. metody Back-end Server
   c. REST API

2. Rodzaj testów: integracyjne
   Technologia: JUnit
   Testowany podmiot:
   a. współgranie z zewnętrznymi usługami
   b. integracja między elementami
   c. poprawne tworzenie konta użytkownika
   d. łączenie z Google Maps API

3. Rodzaj testów: manualne
   Technologia: -
   Testowany podmiot:
   a. aplikacja mobilna
   b. interfejs użytkownika

4.6 Analiza Ryzyka

Oceniany jest wpływ(1 - niski, 4 - bardzo wysoki) i prawdopodobieństwo(Małe, średnie, duże) wystąpienia ryzyk z poszczególnych kategorii. Na podstawie tych metryk oceniane jest ryzyko zgodnie z tabelą:

	1 - niski wpływ	2 - średni wpływ	3 - wysoki wpływ	4 - bardzo wysoki wpływ
Małe prawdopodobieństwo	Brak	Niskie	Niskie	Średnie
Średnie prawdopodobieństwo	Brak	Niskie	Średnie	Średnie
Duże prawdopodobieństwo	Niskie	Średnie	Wysokie	Wysokie

4.6.1 Harmonogram

Ryzyko:	**Konsekwencje:**	**Wpływ (1-4)**	**Prawdopo-dobieństwo wystąpienia:**	**Reakcja:**

Niedotrzymywanie terminów etapów pracy	Opóźnienia	3	Średnie	Uniknięcie (pozostawienie buforu czasowego)
Opóźnienia w trakcie wykonywania poszczególnych zadań	Niedostarczenie początkowo zaplanowanej aplikacji, pogorszenie jakości końcowego projektu	4	Duże	Minimalizacja (uproszczenie zadań)
Niedoskonałe oszacowanie czasu potrzebnego do wykonania zadań/etapów realizacji projektu	Niedostarczenie początkowo zaplanowanej aplikacji, pogorszenie jakości końcowego projektu	4	Duże	Minimalizacja (uproszczenie zadań)

4.6.2 Zespół

Niedoświadczony zespół	Niedoskonałe zaplanowanie pracy, opóźnienia związane z brakiem doświadczenia w poszczególnych technologiach	3	Duże	Minimalizacja (uproszczenie zadań)
Różna dyspozycyjność członków zespołu	Utrudnienie w przeprowadzeniu spotkań, skutkujące pogorszeniem jakości wykonywania zadań i stratami czasowymi	2	Duże	Minimalizacja (wcześniejsze zaplanowanie spotkań, komunikacja na bieżąco)
Niedoskonałe zaplanowanie architektury/testów/bazy danych	Nagłe zmiany klas, tracenie czasu na zaplanowanie modułów na nowo	3	Średnie	Akceptacja
Rezygnacja ze studiów członka zespołu	Konieczność rozdzielenia obowiązków danej osoby wśród pozostałych członków. Możliwe opóźnienia.	4	Niskie	Akceptacja

4.6.3 Sprzęt

Brak możliwości	Możliwe problemy nie zostaną	2	Średnie	Akceptacja

przeprowadzenia testów na niektórych typach urządzeń (np. tablet)	wykryte i rozwiązane przed zakończeniem pracy nad projektem			
Fizyczna awaria sprzętu	Opóźnienia spowodowane przez brak możliwości wykonywania zadań	3	Małe	Minimalizacja (częste wykonywanie backupów)
Awaria zasilania	Zgubienie istotnych danych, niemożliwość wykonania zadań na czas przez potrzebę zrobienia na nowo tego co zostało zgubione	4	Małe	Minimalizacja (częste wykonywanie backupów)
Niewystarczające zasoby dla testowania	Brak możliwości przetestowania wszystkich funkcjonalności aplikacji, nie wykryte problemy skutkujące pogorszeniem jakości aplikacji.	3	Średnie	Akceptacja

4.6.4 Technologia

Utrata kodu w repozytorium przez błąd osób trzecich	Opóźnienie spowodowane usystematyzowaniem aktualnego kodu	4	Małe	Minimalizacja (częste wykonywanie synchronizacji wersji)
Utrata kodu poprzez złe użytkowanie SVN	Opóźnienie przy przywracaniu odpowiednich wersji aktualnego stanu projektu	3	Małe	Akceptacja (nauczenie obsługi SVN)
Zawieszenie serwera przechowującego kod	Opóźnienie spowodowane przez ręczną aktualizację wersji, zmiana serwera	4	Średnie	Akceptacja

4.6 Lista narzędzi planowanych do użycia przy realizacji projektu

- SVN:
    - Git
- IDE:
    - IntelliJ IDEA
    - Android Studio
- Kanban Board
    - Tara.ai
- Github Actions

    - Automatyczne uruchamianie testów
- Narzędzia do wizualizacji
    - Diagrams.net
- Komunikatory
    - Discord

5. Uruchomienie aplikacji w środowisku lokalnym, deployment na produkcję.

5.1 Back-end server

Do uruchomienia serwera będziemy potrzebowali bazy danych MySQL o nazwie “io_db” (lub innej, wtedy wymagane są odpowiednie zmiany w “application.properties”) i odpowiednio skonfigurowanego pliku “application.properties”. Wymagana jest jednocześnie Java 15 (lub nowsza) oraz Gradle 6.6.1 (lub nowszy). Do pełnego wykorzystania wszystkich funkcji potrzebny jest odpowiednio skonfigurowany “appSecret” (w pliku “Authorization.java”) z serwisu OAuth, który generuje token na podstawie którego użytkownik jest autoryzowany oraz adres serwisu OAuth (w pliku “GetUserData.java”).

5.2 Aplikacja mobilna

Aplikację mobilną można zainstalować generując plik “.apk” lub w emulatorze, korzystając np. z Android Studio. Aby zalogować się do aplikacji wymagane jest konto w serwisie OAuth. Uruchamiając lokalnie wszystkie serwisy musimy odpowiednio skonfigurować adresy serwera back-end oraz serwera OAuth (zmiany w plikach “httpClient.java”, “LoginDataSource.java” i “strings.xml”).