# UITMBER (WSIiZ/UITM UBER)

Data Driven Development

### Założenia funkcjonalne

* Logowanie/Rejestracja
* Możliwość zamówienia kierowcy/taxi

//Kierowca

* Możliwość zgłoszenia jako kierowca
* Możliwość przejęcia kursu (na podstawie zgłoszenia/zamówienia)
* Zestawienie kursów
* Wyświetlenie ulubiona piosenka klienta

## Aplikacja mobilna

Android/IOS

Xamarin.Forms

### Spis Funkcjonalności

1. Rejestracja konta
2. Logowanie
   1. SSO (do zrobienia jeśli zostanie nam czasu):
   2. AppleId,
   3. Google
   4. Microsoft?
3. Wyświetlenie mapy z kierowcami w okolicy,
4. Zamawianie kursów:
   1. Wybranie punktu początkowego
   2. Wybranie punktu końcowego
   3. Typ auta (Uber, UberX, 7-osobowe auto)
   4. Zaakceptowanie kosztu/przejazdu
5. Historia przejazdu ze szczegółami (cena, kierwca, czas, trasa, klient)
6. Ocena kursu i pasażera (kierowcy)
7. Ulubiona lokalizacja
8. Pobieranie aktualnej lokalizacji
9. Możliwość spięcia z Spotify
10. Podgląd na żywo gdzie jest mój kierowca
    1. Informacje o kierowcy (zdjecie, auto, rejestracja)

Kierowca

1. Mechanizm zgłaszania chęci współpracy (osoba chce być kierowcą)
   1. Zdjęcie auta
   2. Model/Marka,
   3. Rocznik,
   4. Tablica rejestracyjna
2. Mechanizm zarządzania kursem (rozpoczęcie, kontakt z klientem, informacje o kliencie)
3. Wybór piosenki
4. Wyświetlanie obecnie zamówionych kursów
5. Wybór i zaakceptowanie zamówionego kursu
6. Ocena pasażera (po zakończeniu przejazdu)

## Założenia

-TabbedPage

-colorPrimary #bb171e

-colorPrimaryDark #941016

-colorText #232323

-colorTextDescription #212121

-colorAccent #ffbf00

Services: (12 na zajęciach, 17 grupa)

Base:

RequestService - jjonca

#### Authentication - jjonca

**POST​/Authentication​/Authenticate**

#### **POST​/Authentication​/Register** (w60054)

#### Application (w60071)

**POST​/Application​/SendApplication**

**GET​/Application​/GetMyApplications**

#### Car (w60100, w60078)

**GET​/Car​/GetMyCars**

**POST​/Car​/Add**

**PUT​/Car​/Update**

**DELETE​/Car​/Delete​/{id}**

#### Clients (w60101)

**GET​/Clients​/GetMyProfile**

**PUT​/Clients​/UpdatePhoto**

#### Discounts (w60079)

**GET​/Discounts​/Verify**

**POST​/Discounts​/AddToOrder**

#### Driver (w60104)

**GET​/Driver​/GetNerbyDriveres**

**GET​/Driver​/GetProfile**

#### Location (w60054)

**POST​/Location​/SaveMyLocation**

#### Orders (w60102, w60103, w60061, w61068)

**GET​/Orders​/GetMyOrders**

**GET​/Orders​/GetCarTypes**

**GET​/Orders​/GetClientOrderDetails**

**GET​/Orders​/GetLuggageTypes**

**GET​/Orders​/GetCost**

**POST/Orders/OrderPayment**

**POST​/Orders​/OrderAccept**

**PUT​/Orders​/ClientRate**

#### UFLocations (w61068, w61091)

**POST​/UFLocations​/AddLocation**

**DELETE​/UFLocations​/DeleteLocation​/{id}**

**GET​/UFLocations​/GetMyLocations**

# Przygotowanie stron:

**Mapa z customowymi Pinami : jjonca**

Strona Logowania (Button register) **(w60072)**

Strona Rejestracji  **(w60054)**

Strona z listą moich zamówień: **(w60071, w60100, w60052)**

* Strona z CollectionView
* Strona Details

Strona Profile: **(w60078, w60101, w60079,w60104)**

* Profile
* MyUFLocation
* AddUFLocation

#### Strona Driver: **(w60102, w60103, w60061, w60068, w61068)**

* MyApplication
* MyCars (CRUD)
* SendApplication

## Serwer danych/API

.Net Core Api

1. Metody rejestracji
2. Metody logowania
3. Pobranie listy dostępnych kierowców w naszej okolicy,
4. Mechanizm zamawiania kursów
5. Mechanizm oceny kursu pasażera/kierowcy
6. Mechanizm wyboru piosenki
7. Pobranie obecnie zamówionych kursów (kierowca)
8. Mechanizm potwierdzania kursów
9. Mechanizm zwracania szczegółów użytkownika,
10. Mechanizm zgłaszania chęci współpracy
11. Mechanizm zapisywania ulubionych lokalizacji

Endpoint:

**/Authentication** – jjonca

**/Cars (w60078)**

-GetMyCars() - HttpGet

-Add(model) - Post

-Update(model) -Put

-Delete(id) - Del

**/Clients (w60054)**

-GetMyProfile() (mój profil wraz z oceną)

-UpdatePhoto(photo) (aktualizacja zdjęcia na profilu)

**/Drivers: (w60071)**

-GetNerbyDriveres(lat,long) – (Pobieramy kierowcow z jednego miasta)

-GetProfile(id)

**/Locations (w60071)**

-SaveMyLocation(id, lat,lang) – Post (zapis mojej aktualnej pozycji)

**/Favourites (w60101)**

-GetMyLocations() GET

-AddLocation() POST

-DeleteLocation() DEL

**/Orders**

-GetMyOrders() GET **(w60104)** (szczegóły z informacją o kliencie, kierowcy)

-GetCarTypes() – typy samochodów/przejazdów GET **(w60104)**

-GetClientOrderDetails(id) – info o kursie + info o kierowcy GET **(w60104)**  
  
-GetCost(date, distance) GET **(w60102)**

-OrderAccept(id,date) POST **(w60102)**

-OrderPayment(id) Post **(w60102)**

-ClientRate(id, date, info, rating) PUT **(w60102)** (ocena kierowcę przez klienta)

**-** GetLuggageTypes **(w60061)**

**/DriverOrders**

-GetAvalilableOrders(lat,long) GET **(w60103)** (zamówienia ze statusem nowy)

-ProcessOrder(id) – przejęcie zamówienia PUT **(w60103)**

-GetClientInfo(id) GET **(w60103)**

-DriverRate(id, date, info, rating) PUT **(w60100)** (ocena klienta przez kierowcę)

-GetDriverOrderDetails(id) – info o kursie + info o pasażerze GET **(w60100)**

-ChangeOrderStatus(id, status, data, info) PUT **(w60100)**

**/Applications**

-SendApplication(Model application) POST **(w61068)** (wysłanie zgłoszenia, id auta)

-GetMyApplications() GET **(w61068)**

**/Music**

-GetClientMusic(id) GET

-RegisterAccount() – POST – do przeczytania api Spotify