Projekt

***Parking Samochodowy***

Autor: Marcin Hinc

Spis treści

[Wprowadzenie 2](#_Toc509563197)

[Cel projektu 2](#_Toc509563198)

[Użytkownicy 2](#_Toc509563199)

[Schemat Projektu 2](#_Toc509563200)

[Implementacja i Realizacja Projektu 3](#_Toc509563201)

[Diagramy UML 3](#_Toc509563202)

[Diagram Klas 3](#_Toc509563203)

[Diagram Obiektów 6](#_Toc509563204)

[Diagram Przypadków Użycia 6](#_Toc509563205)

[Diagram Czynności 8](#_Toc509563206)

[Diagram Stanów - Aplikacji 9](#_Toc509563207)

[Diagram Stanów –Model Parkowania 10](#_Toc509563208)

[Diagram Przebiegu 11](#_Toc509563209)

[Diagram Wdrożenia 12](#_Toc509563210)

[GUI Aplikacji 13](#_Toc509563211)

[Wstępny widok dla formularza zakładania konta 13](#_Toc509563212)

[Wstępny widok dla formularza logowania do aplikacji 14](#_Toc509563213)

[Wstępny widok panelu głównego aplikacji wraz z dodaniem lokalizacji docelowej 15](#_Toc509563214)

[Źródła 16](#_Toc509563215)

# 

# Wprowadzenie

Każdy z Nas na pewno ma lub zna kogoś kto porusza się z dużą częstotliwością samochodem. Miał niejednokrotnie problem ze znalezieniem miejsca parkingowego dla swojego pojazdu. Wychodzę do Naszych kochanych kierowców naprzeciw. Mianowicie opracowałem aplikacje mobilną „Parking Samochodowy”, dzięki której życie kierowców stanie się zdecydowanie przyjemniejsze oraz mniej stresujące. Już nigdy nie będziesz miał problemów z umiejscowieniem swojego pojazdu. Przejdźmy zatem do celu projektu.

## Cel projektu

Projekt „Parking Samochodowy” jest to aplikacja wspierająca kierowców. Jest to łatwa w obsłudze aplikacja mobilna, która umożliwi łatwy dostęp do znalezienia potrzebnego miejsca dla Twojego samochodu. Pozwala ona również na przechowywanie ulubionych miejsc tudzież sprawdzenia historii parkowania.

## Użytkownicy

Użytkownikiem systemu będzie mogła stać się każda osoba, posiadająca urządzenie mobilne które będzie miał stabilny i swobodny dostęp do Internetu oraz GPS.

## Schemat Projektu

Projekt będzie aplikacją mobilną zawierająca:

- formularz rejestracyjny oraz logowania,

- panel obsługi użytkownika,

- ekran dodania lokalizacji

## Implementacja i Realizacja Projektu

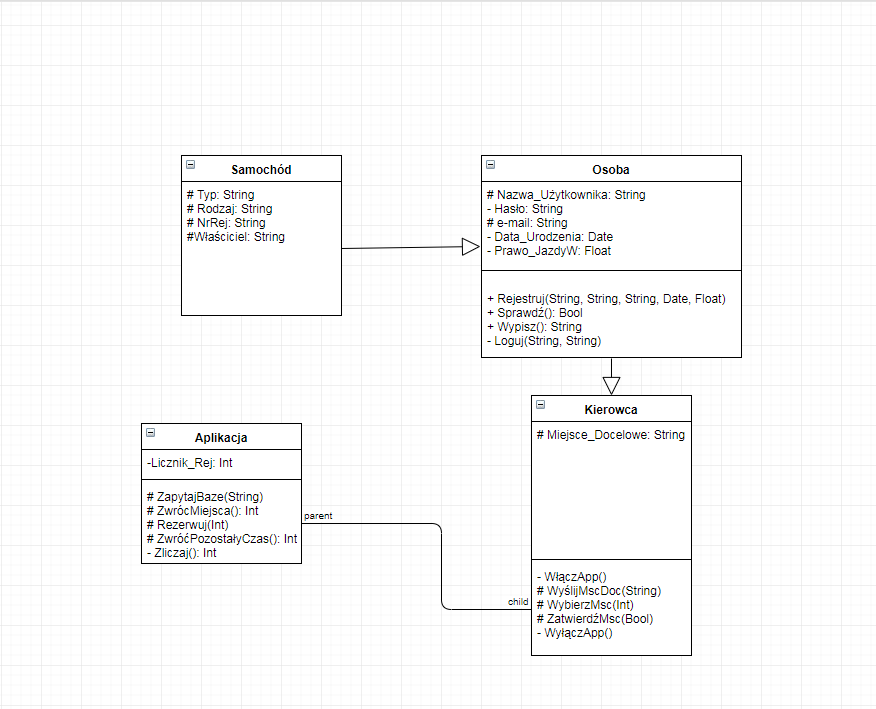
Aplikacja będzie składała się z co najmniej dwóch części. Mianowicie:

- część kliencka – GUI aplikacji(Java), które dodane jest na końcu dokumentu,

- część serwerowa – dane są gromadzone w bazie danych(Oracle) w porozumieniu z aplikacją

# Diagramy UML

## Diagram Klas



***Samochód***

Klasą abstrakcyjną, która opisuję podstawowe parametry samochodu.

* Typ – przechowuje informacje o typie samochodu.
* Rodzaj – przechowuje informacje o rodzaju samochodu.
* NrRej – przechowuje informacje o numerze rejestracyjnym samochodu.
* Właściciel – przechowuje informacje kto jest właścicielem samochodu.

***Osoba***

Klasa dziedzicząca po klasie Samochód. Posiada 3 metody publiczne oraz jedną prywatną.

Pola:

* Nazwa\_Użytkownika– przechowuje informacje o nazwie użytkownika.
* Hasło– przechowuje informacje o haśle użytkownika wraz z solą.
* E-mail– przechowuje informacje o adresie email użytkownika.
* Data\_Urodzenia– przechowuje informacje o dacie urodzenia użytkownika.
* Prawo\_JazdyW – przechowuje informacje o stażu posiadania prawa jazdy przez użytkownika.

Metody:

* Rejestruj() – metoda pozwalająca na rejestracje nowego użytkownika.
* Sprawdź() – metoda sprawdzająca czy dany użytkownik istnieje w bazie danych.
* Wypisz() – metoda wyświetlająca komunikat o poprawności założenia konta.
* Loguj() – metoda pozwalająca na zalogowanie się do aplikacji.

***Kierowca***

Klasa dziedzicząca po klasie Osoba oraz wymienia dane z aplikacją. Posiada 3 metody chronione oraz 2 metody prywatne.

Pola:

* Miejsce\_Docelowe – przechowuje informacje o miejscu docelowym podanym przez użytkownika.

Metody:

* WyślijMscDc() – metoda wysyłająca miejsce docelowe do bazy danych.
* WybierzMsc() – metoda wysyłająca wybrane miejsce parkingowe.
* ZatwierdźMsc() –metoda zatwierdzająca wybrane miejsce parkingowe.
* WłączApp() – metoda włączająca aplikacje.
* WyłączApp() – metoda wyłączająca aplikacje.

***Aplikacja***

Klasa komunikująca się z bazą danych oraz wymianą danych z użytkownikiem. Posiada 4 metody chronione oraz jedną prywatną.

Pola:

* Licznik\_Rej – przechowuje ilość zarejestrowanych użytkowników.

Metody:

* ZapytajBaze() – metoda pytająca bazę danych o miejsca parkingowe w obrębie podanej lokalizacji.
* ZwróćMiejsca() – metoda zwracająca ewentualne wolne miejsca parkingowe.
* Rezerwuj() – metoda rezerwująca miejsce parkingowe.
* ZwróćPozostałyCzas() – metoda zwracająca pozostały czas do zatwierdznia/anulowania miejsca parkingowego.
* Zliczaj() – metoda zliczająca ilość zarejestrowanych użytkowników.

## Diagram Obiektów

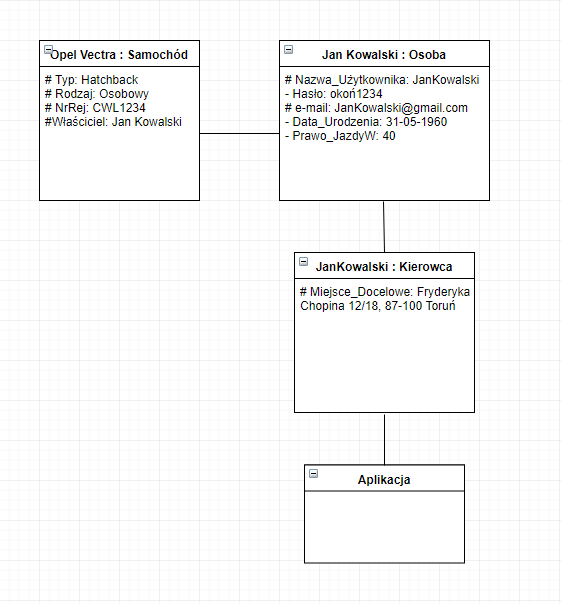
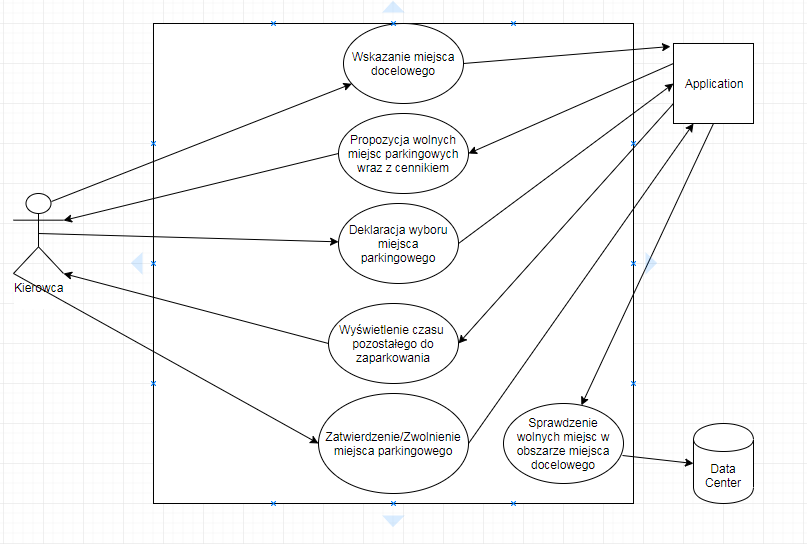


Diagram opisuje obiekty utworzone z wyżej opisanego diagramu klas.

## Diagram Przypadków Użycia



*Kierowca:*

* Wskazanie miejsca docelowego – Kierowca wprowadza poprzez przycisk miejsce w którym planuje się zatrzymać.
* Deklaracja wyboru miejsca – Kierowca wybiera miejsce parkingowe w którym chciałby pozostawić swój pojazd.
* Zatwierdzenie/Zwolnienie miejsca – Kierowca potwierdza lub anuluje miejsce parkingowe.

Aplikacja:

* Propozycja wolnych miejsc – aplikacja wyświetla dostępne miejsca parkingowe.
* Wyświetlenie pozostałego czasu – aplikacja wyświetla pozostały czas do potwierdzenia/anulowania miejsca.
* Sprawdzenie miejsc – aplikacja w porozumieniu z bazą danych dowiaduje się czy są, ile i jakie są wolne miejsca w podanej lokalizacji.

Baza Danych:

* Gromadzi dane dotyczące miejsc parkingowych.
* Porozumiewa się z aplikacją podając jej wolne miejsca parkingowe.

## Diagram Czynności

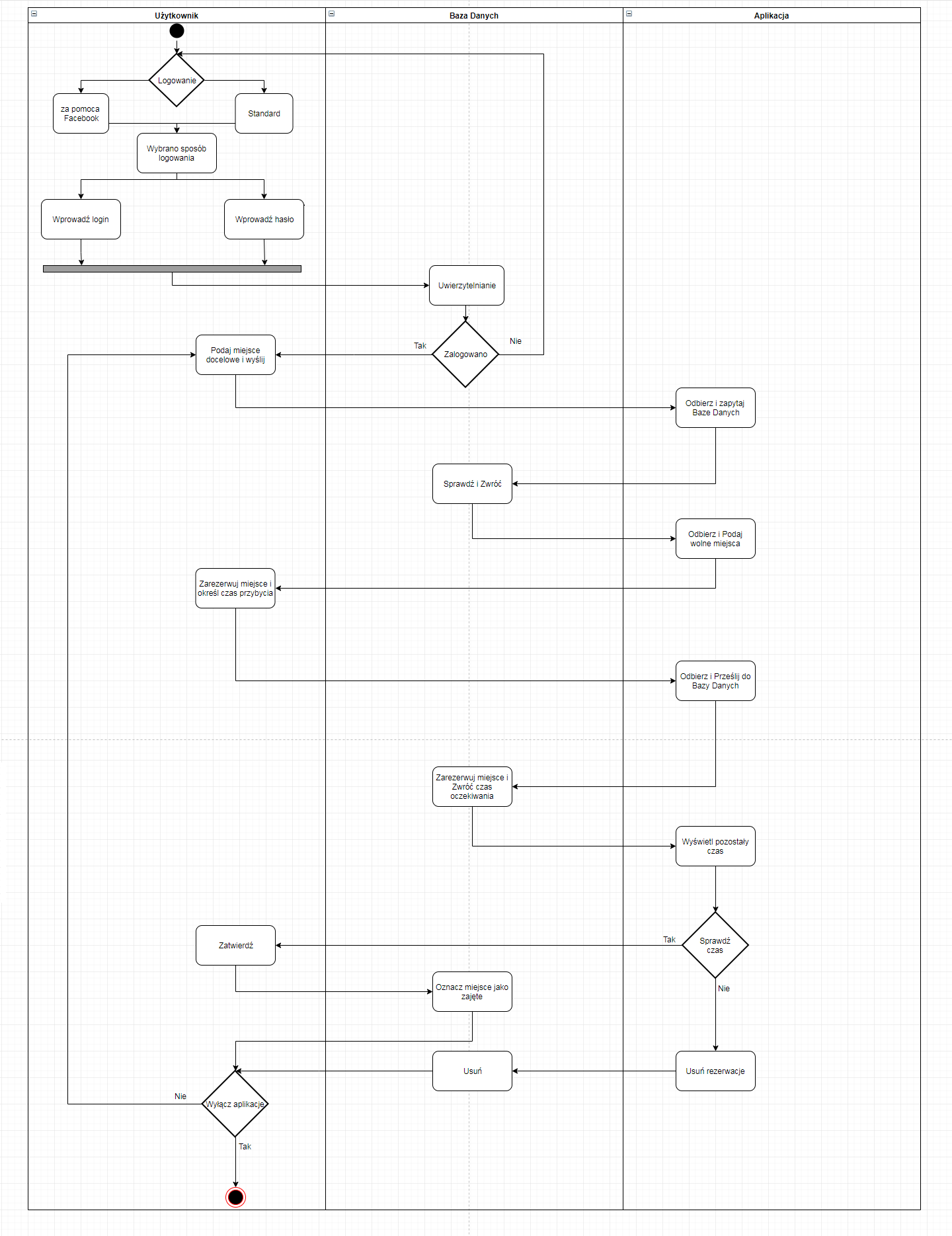


Diagram ten pokazuje jakie działania użytkownik wykonuje podczas korzystania z aplikacji, jak aplikacja współgra z użytkownikiem oraz komunikuje się z bazą danych. Na początku użytkownik wybiera sposób logowania. Są dwie opcje, logowanie za pomocą konta Facebook lub przy użyciu standardowego konta. Następnie użytkownik podaje login oraz hasło. Następuje uwierzytelnianie   
w bazie danych i gdy podano prawidłowe dane, to następuje załadowanie okna, gdzie można wybrać miejsce docelowe, a jeżeli nieprawidłowe dane, to użytkownik jest proszony o ponowną próbę logowania. Po podaniu miejsca docelowego aplikacja wysyła zapytanie do bazy danych o wolne miejsca parkingowe. Aplikacja dostaje odpowiedź od bazy danych i wyświetla ją użytkownikowi, który następnie rezerwuje miejsce i określa w jakim czasie dotrze na miejsce. Aplikacja odbiera dane od użytkownika i wysyła zapytanie do bazy danych odnośnie rezerwacji miejsca i pozostały czas oczekiwania. Aplikacja wyświetla pozostały czas i następnie sprawdza czy podany czas upłynął. Jeżeli użytkownik zdążył dojechać przed upłynięciem czasu i zatwierdził dojazd na miejsce, to baza danych dokonuje wpis odnośnie nowo zajętego miejsca. Na koniec użytkownik jest pytany czy wyłączyć aplikację. Jeżeli się nie zgadza na zamknięcie aplikacji, to zostaje przeniesiony do okna, gdzie można wybrać miejsce docelowe.

## Diagram Stanów - Aplikacji

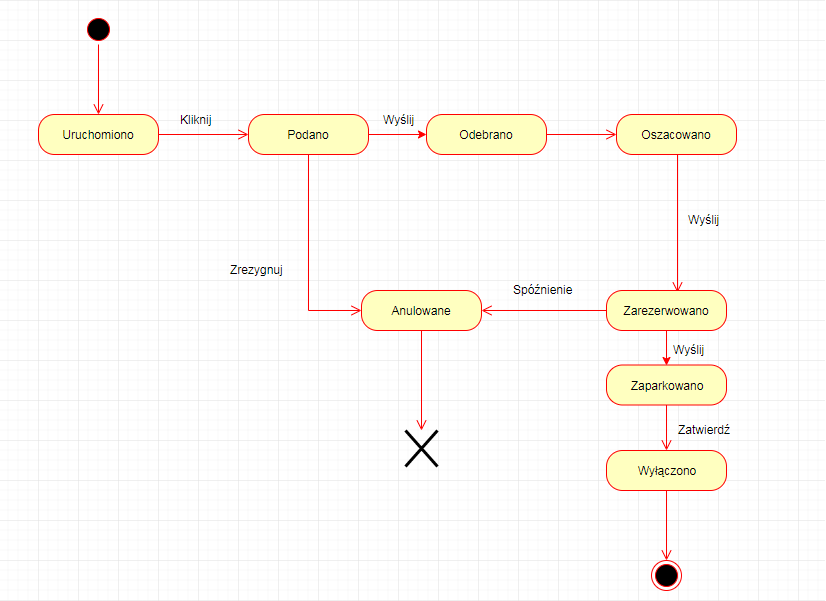
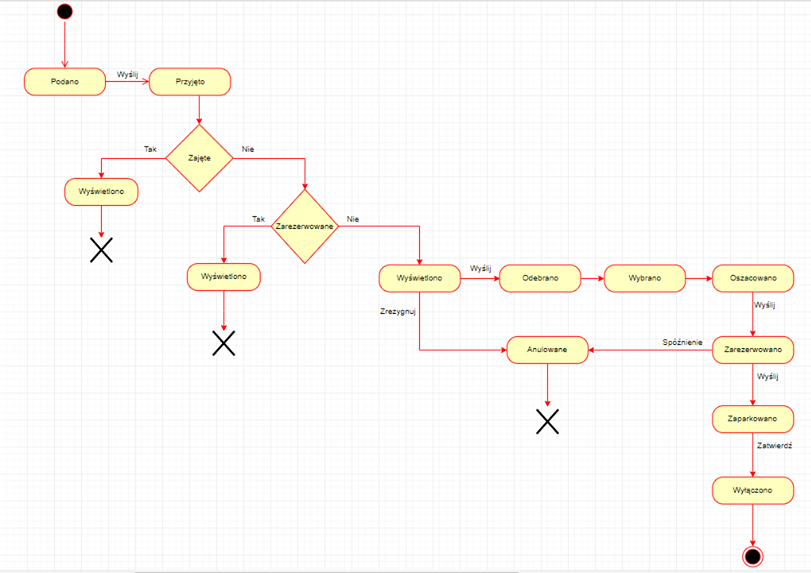


Diagram ten przedstawia stany w jakich znajduje się aplikacja. Po uruchomieniu aplikacji użytkownik podaje miejsce docelowe, jeżeli z tego miejsca zrezygnował, to następuje anulowaniei przejście   
w stan zniszczenia. Zaś gdy zaakceptuje wybór to baza danych otrzymuje prośbę o rezerwacje miejsca. Następnie oszacowuje czas na przybycie i rezerwuje miejsce. Gdy użytkownik się spóźni, to następuje anulowanie i przejście w stan zniszczenia. Natomiast, gdy dotrze na miejsce przed zakończeniem czasu wtedy zatwierdza zaparkowanie i aplikacja jest wyłączana. Następuje przejście   
w stan końcowy

## Diagram Stanów –Model Parkowania



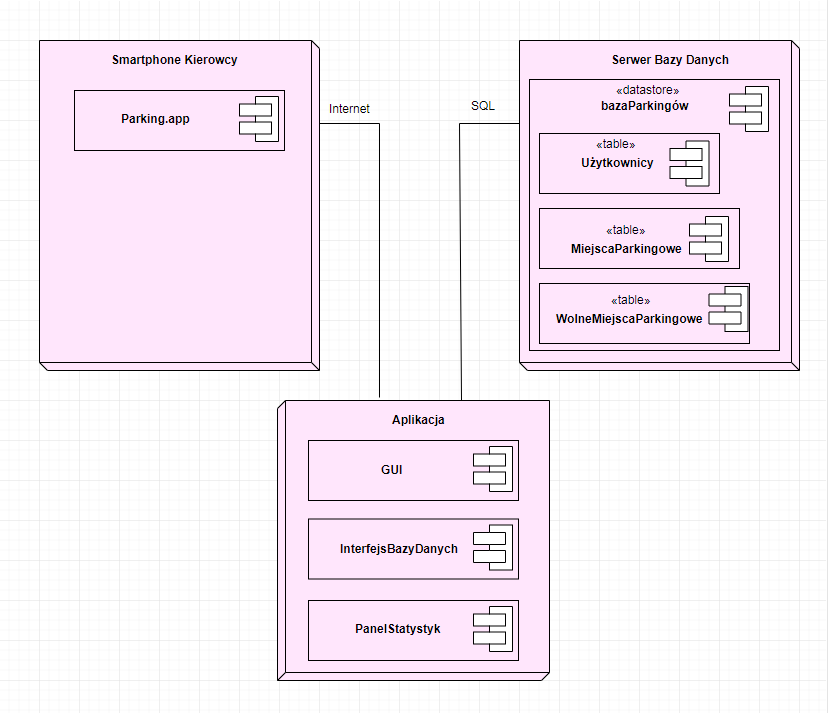
Na diagramie ukazany jest poszczególny stan miejsc parkingowych. Użytkownik podaje miejsce docelowe, następnie aplikacja przyjmuje je i wysyła zapytanie do bazy danych. Gdy miejsce jest zajęte, to wyświetla stosowną informacje i przechodzi w stan zniszczenia, czyli zakończenia maszyny stanowej wskutek zniszczenia adekwatnego obiektu. Natomiast gdy miejsce nie jest zajęte to baza sprawdza czy jest zarezerwowane, jeżeli tak to wyświetla stosowny komunikat i przechodzi w stan zniszczenia, a jeżeli nie jest zarezerwowane, to wyświetla stosowny komunikat i się pyta czy zarezerwować. Gdy użytkownik zrezygnuje, to zarezerwowanie miejsca jest anulowane i następuje przejście w stan zniszczenia. Gdy użytkownik jednak zdecyduje się zarezerwować miejsce, to zatwierdza wybór, baza danych otrzymuje zapytanie. Następnie dokonuje rezerwacji miejsca, wysyła oszacowany czas. Gdy użytkownik się spóźni to następuje anulowanie i przejście w stan zniszczenia. Lecz gdy dotrze na czas, to zatwierdza zaparkowanie, następuje wpis do bazy danych i aplikacja przechodzi w stan końcowy.

# **Diagram Przebiegu**C:\Users\Lukasz\Desktop\Project IO\sekwencji.png

***Kierowca***

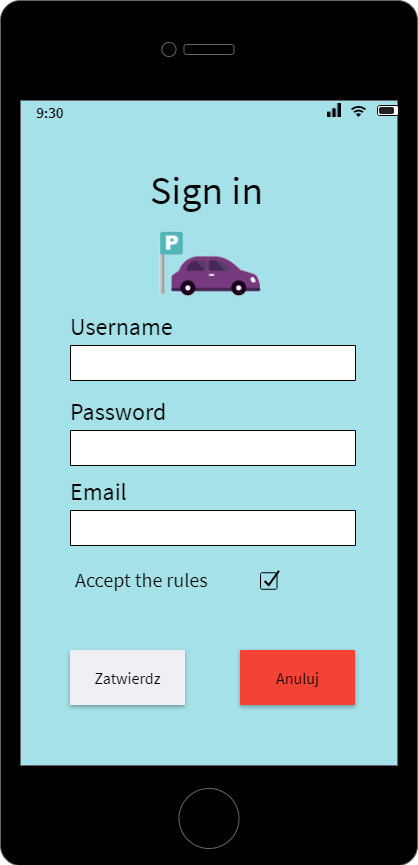
Zalogowany użytkownik(Kierowca) uruchamia aplikacje. Następnie dodaje lokalizacje, w której planuje znaleźć miejsce na pozostawienie samochodu. Dalej aplikacja odpytuje bazę danych o wolne miejsca w obszarze podanym przez użytkownika. Aplikacja otrzymuje informacje zwrotną   
i natychmiast przekazuję ją użytkownikowi. W dalszej części Kierowca wybiera miejsce, w którym chciałby zaparkować i w jakim czasie tam dotrze podając tę informację do aplikacji. Aplikacja po otrzymaniu informacji wysyła żądanie do bazy danych o rezerwacje miejsca. Otrzymuję odpowiedź zwrotna, przekazuje użytkownikowi pozostały czas jaki mu pozostał do zatwierdzenia wybranego miejsca. Po dojechaniu na miejsce użytkownik ma obowiązek zatwierdzić miejsce lub je porzucić. Porzucenie może nastąpić również samoistnie przez aplikację wówczas gdy upłynie czas wyświetlany przez aplikację. Na końcu użytkownik może lecz się musi wyłączyć aplikacje.

## Diagram Wdrożenia

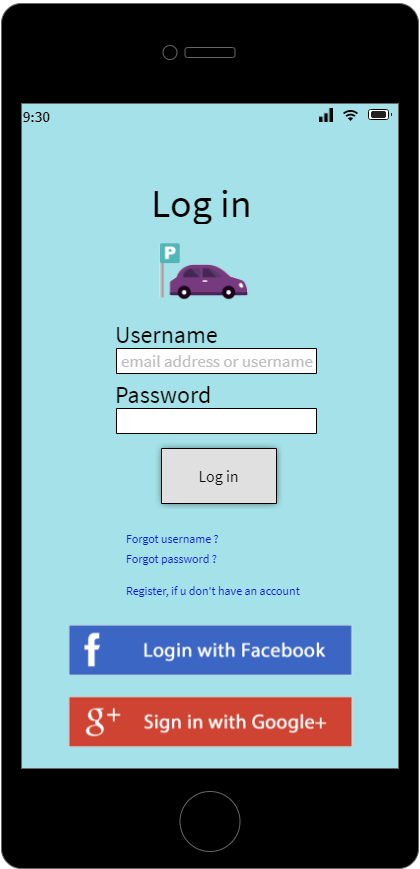


* Smartphone Kierowcy – instalacja aplikacji Parking.app( aplikacja mobilna, dostępna przy korzystaniu z internetu).
* Serwer Bazy Danych – serwer bazodanowy napisany w Oracle, posiadający tabele służące do przechowywania danych zarówno podanych od użytkownika podczas rejestracji jak i do przechowywania stanów miejsc parkingowych.
* Aplikacja – posiada interfejs graficzny napisany w Javie, InterfejsBazyDanych(formularze rejestracyjne, logowanie) oraz PanelStatysyk przechowywujący odpowiednie dane zarówno wyświetlane dla użytkownika jak i dla wiadomości dla administratora.

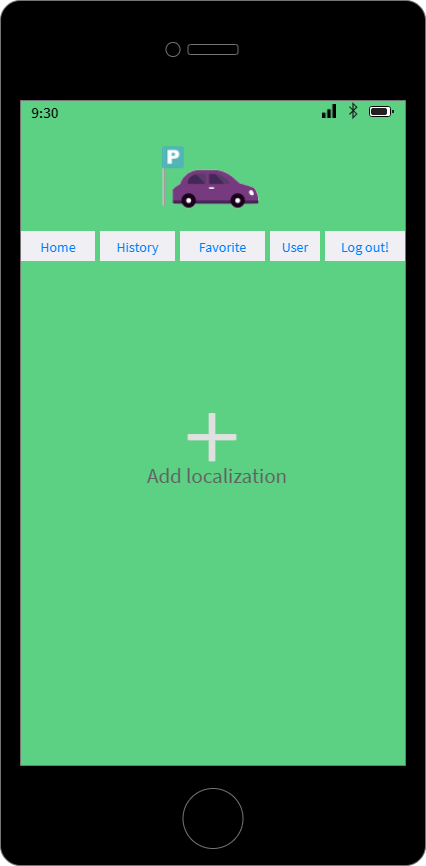
# GUI Aplikacji

Wstępny widok dla formularza zakładania konta

## Wstępny widok dla formularza logowania do aplikacji

****

## Wstępny widok panelu głównego aplikacji wraz z dodaniem lokalizacji docelowej

****

# Źródła

* **Diagramy:** [**http://zasoby.open.agh.edu.pl/~09sbfraczek/**](http://zasoby.open.agh.edu.pl/~09sbfraczek/)
* **GUI:** [**https://www.mockflow.com/**](https://www.mockflow.com/)
* **Wykonanie diagramów:** [**https://www.draw.io/**](https://www.draw.io/)