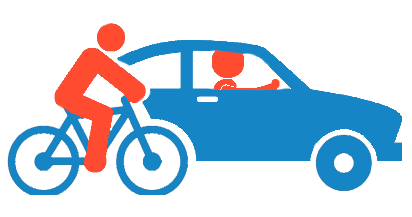
––––

**Cairo University**  
**Faculty of Computers and Information**  
**Computer Science Department**

––

**Graduation project 2017/2018**



**Transportation Rental Services**

**Dr. Abeer El-Korani**

|  |  |  |
| --- | --- | --- |
| ID | Name | Dept. |
| 20140146 | Sayed Ashraf Sayed | CS |
| 20140108 | Khaled Mohamed Ibrahim | CS |
| 20140185 | Omar Adel Mohamed Ramzy | CS |
| 20140099 | Hossam Khaled El-Hwary | CS |

Contents

[**Introduction** 3](#_Toc506738145)

[**Motivation** 3](#_Toc506738146)

[**Problem Definition** 4](#_Toc506738147)

[**Project Objective** 4](#_Toc506738148)

[**Stakeholders** 4](#_Toc506738149)

[**Related work** 5](#_Toc506738150)

[**System functions** 6](#_Toc506738151)

[**General Function** 7](#_Toc506738152)

[**Service Owner** 8](#_Toc506738153)

[**Service User** 9](#_Toc506738154)

[**Service Admin (CRUD)** 10](#_Toc506738155)

[**Non-Functional Requirements** 11](#_Toc506738156)

[**Usability** 11](#_Toc506738157)

[**Reliability** 11](#_Toc506738158)

[**Supportability** 11](#_Toc506738159)

[**Performance** 11](#_Toc506738160)

[**Class Diagram** 12](#_Toc506738161)

[**ERD** 12](#_Toc506738162)

[**Sequence Diagram** 13](#_Toc506738163)

[**Request Car** 13](#_Toc506738164)

[**Add Car** 13](#_Toc506738165)

[**Recharge Quota** 14](#_Toc506738166)

[**System Architecture** 15](#_Toc506738167)

[**Use Case Tables** 16](#_Toc506738168)

[**Future Work** 18](#_Toc506738169)

# **Introduction**

Nowadays when any family or organization want to move to a new apartment or move their old location they will face a major problem which is having a secure and a reliable way of transportation to transport their furniture.

Regards the accidents in Egypt, our application provides a winch for these cases. Also, when a foreigner comes to Egypt and he doesn’t want a taxi but needs to rent a car for a certain amount of time, he won’t find many reliable companies that could offer him a good car for him to rent.

This problem isn’t for car and transportation only, it’s about renting a bicycle.

We can notice also that many young people go to riding a bicycle to their university or school instead of riding a car and anyone who cannot afford buying a bicycle will have the same problem.

So our application will provide all of these solutions for the mentioned above problems to facilitate people’s life in that matter.

# **Motivation**

Our application aims to make people life more comfortable and easy in two main parts:

* Being able to rent a car or bicycle at any moment of time
* Solving the transportation problem for home or building’s furniture and any other related equipment

# **Problem Definition**

The main aim for any customer who wants either to rent a car, bicycle or even a truck for transporting what he needs is being able to rent it at any time and from any place.

Our application will solve that problem as for renting a car or bicycle the customer will have to go to the company’s place to take the car or the bicycle, but for the truck the application will know the location of the customer and send the truck to that location.

# **Project Objective**

This application will help people who are in need of renting a bicycle for their daily routine or even for a certain moment of time.

Some will need this application for reserving a truck which will save time for them to search for one.

In addition the application will locate their place immediately in a more efficient and reliable way without facing any issues in having the wrong address.

# **Stakeholders**

* Citizen/Users, Android application users who use service
* Store & Car Owner, Android application users who provide service
* Administration, manage our app

# **Related work**

****

**Uber**

**Uber is a ridesharing app for fast, reliable rides in minutes – day or night. There’s no need to park or wait for a taxi or bus.**

**Careem**

**Careem is car booking app for a safe, reliable and affordable ride. This app will get user where you need to be. Commuting to work? Airport transfer? Visiting family? Forget about parking, traffic, car rental, or waiting for a taxi**.

# **System functions**

* Calculate Minimum Distance
* Bike: Select nearest store and view estimated time to reach it.
* Heavy Car: Calculate minimum distance to destination.
* Winch: Calculate minimum distance to destination.
* Private Car: Calculate minimum distance to destination or nearest store.
* Calculate estimated amount (Cash, Trip time, quota):
* Bike: Fixed amount of money per hour.
* Heavy car: Fixed amount of money per kilometer.
* Private car: Fixed amount of money per hour (minimum 12).
* Winch: amount of Furniture moved
* Payment

Allow user to choose the method he wants to pay.

* Cash.
* Credit card.
* Quota (Bike)

* Show Categories

 Show categories in our system for user to choose from them which type of Rental services he want (Bike, Heavy car, Private car, and winch).

* Show Map with Information of selected category.

Show map containing stores and vehicle near from him.

* Show Notifications

Show notification of new incoming (promo code, requested canceled)

* Updating Locations

In Case of owner change his location, system must update his location on the map that shown to user.

# **General Function**

* CreateAccount
* Allows to make an account on the Application to be able to use its features by entering your Information and it must Sending you a confirmation mail to admit that you make an account successively.
* Login option
* Access your data and entering in your App.
* Forget Password
* Sending new password to his e-mail.
* Email Verification
* Confirming the information account.
* Hamburger Menu
* Profile
* Show user information, picture.
* Balance (Wallet)
* Which Show the user balance?
* Setting
  + Edit profile information.
* History
  + Contain all trips for user.
* Payment method
  + Credit Card
  + Cash
* Review, Rate (User to owner)
  + Allows you to Rate the trip and the Owner
* Logout
  + End user session.
* Search box
  + Shows the searching bar in the App which allow you to drop off  includes the history of recent locations

**Service Owner**

Store page

* Types of vehicles
  + - Car, bike ,Winch and el lorry information
* Add new car
  + - Allows owner to add new car in his store and update information
* Status ( active or not )
  + - If status is online, place will appear on map and if he close or has no vehicle he will be offline and not display on map
* Requests ( accepted , rejected )
  + - Accepted request will send notification to user and display information about the driver , reject request will return the user to select another one
* Notifications ( if user cancel request )
  + - Notification will send to driver and if the car are already moved the –ve value will added to user balance
* Upcoming trips
  + - show to owner our upcoming trips to avoid the conflict
* Terminate rental ( end trip ) (review owner to user )
  + - owner can terminate the trip in case something happened in vehicle

# **Service User**

●    Choose category

* Allow user to select one category out of four

●     Choose store

* After show map to user he will select store or vehicle and display this activity to him

●     Request (2 steps) (choose, confirm)

* Private car
* Type of car
* Delivery (location user)
* Duration

●     Bike

* Once
* Month

Calculate quote with time of trip

●     Heavy or winch

* Type
* Trip road

●     Drop estimation cost

* Estimate the trip fare before booking it.

●     Hold for specific duration

* After specific duration if user didn't come request will be denied

●     Cancel request

* Allow you to cancel request after booking it

●     End trip

* Ending trip after reached the destination and calculates the cost.

●     Feedback, rate

* Rating trip and vehicle after ending trip or before start another one.

●     Notifications (car arrive, owner cancel, cost)

* Notified that is the driver reached to location, In case of owner cancel the trip the driver notified that user cancel the ride and notification for costing after ending trip

●     Mothy user (quota)

* Add free hours to user balance

# **Service Admin (CRUD)**

* Create account
  + Admin is allowed to add new users to system and other admins)
    - User
    - Owners
    - Admins
* Soft delete users
  + Admins are able to soft delete owners and users
* Retrieve all users and filter them
  + Filter users by (latest, soft deleted, oldest)
* Activate users after soft delete
  + Admins are able to activate user again after deleting

# **Non-Functional Requirements**

## **Usability**

* Our system will be easy to use by choosing simple mock-up, and we will enhance interface by taking feedback from Stakeholders.
* Reach any function in minimum number of clicks (3 for example).
* If system will delay for any reason, we will produce message to user telling him to wait.
* Button in interface will have Symbols that show the function of that button.
* System should be secure from point of view users, as it provide secure way of payment process.

## **Reliability**

* System will handle wrong inputs from stakeholders by 100% every input from user will be checked.
* Percentage of failure will be less than 20%.

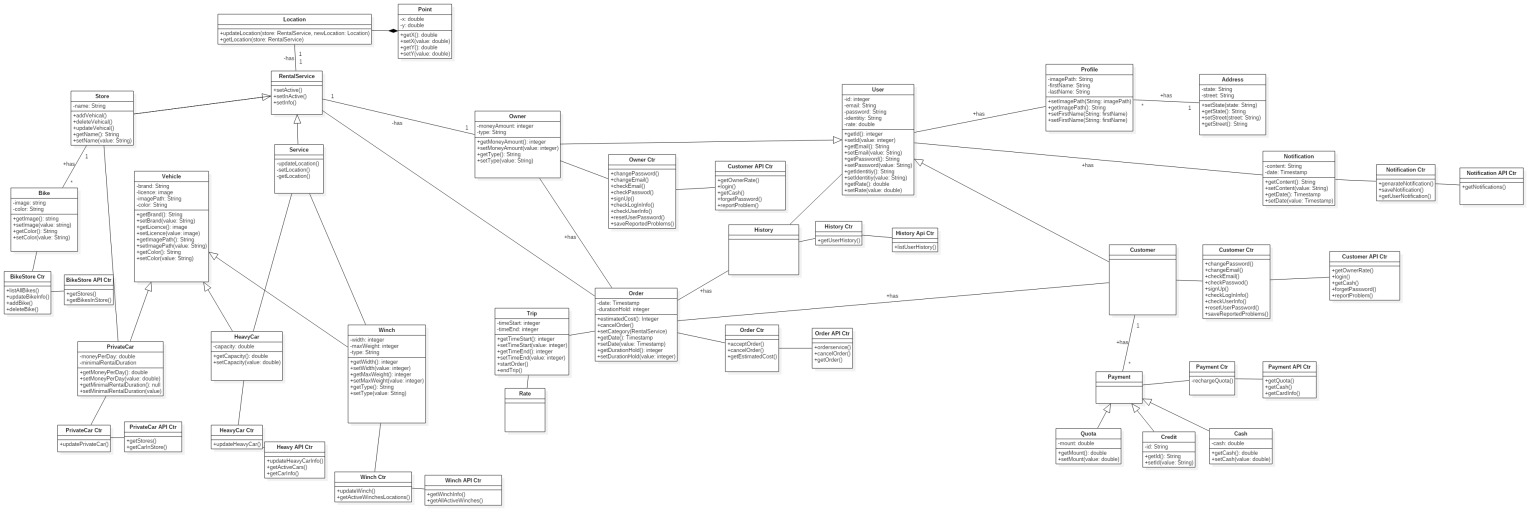
## **Supportability**

* System can be adapted to different environments and configurations.
* Programmers can add new features and test it no more than one week.
* Re-testing the whole system in one week
* System will be down for maintenance less than 24 hour.

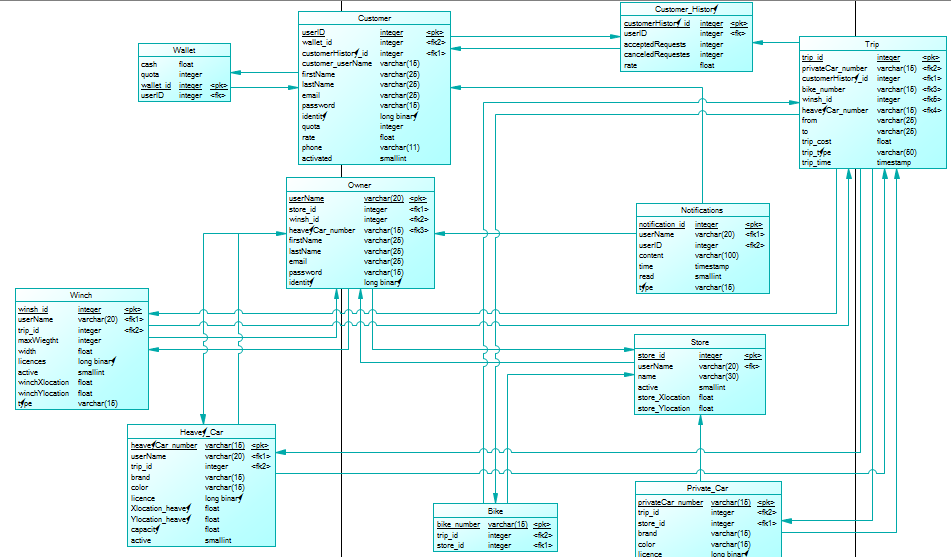
## **Performance**

* Response time: maximum operation time will be 1: 30 seconds.
* Scalability: The system will be accept new featured that we would add in future easily and we will cover large scale of places.
* Availability: The system will be available 24 hours a day.

# **Class Diagram**

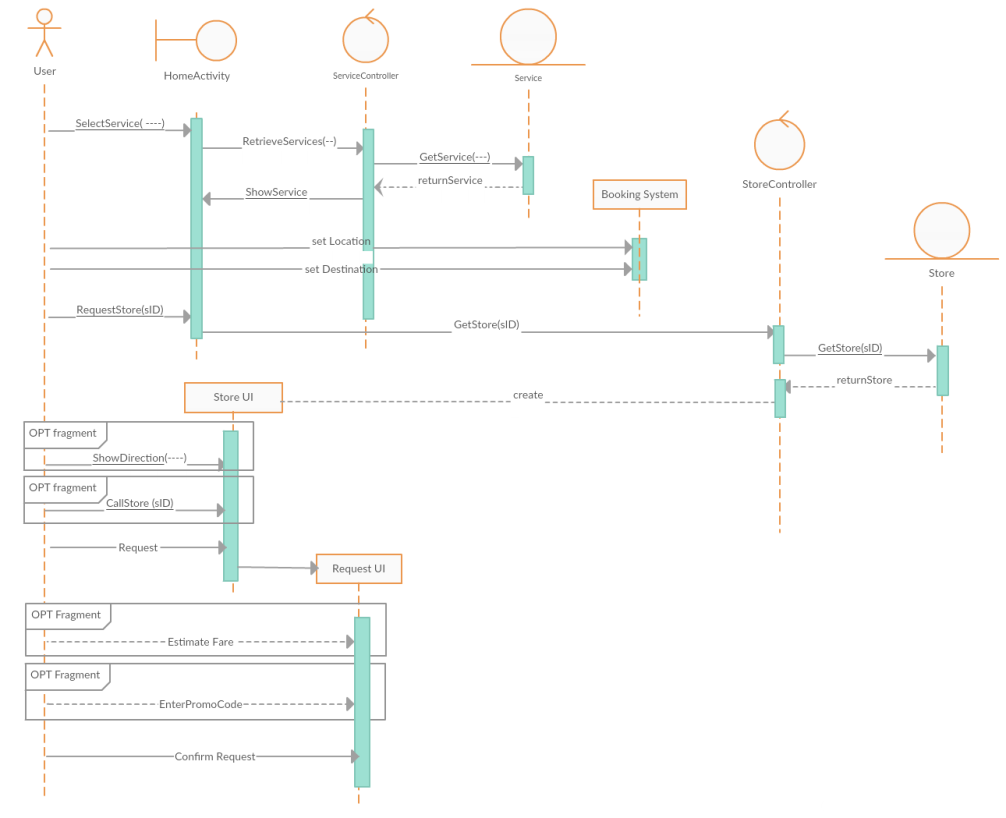


# **ERD**

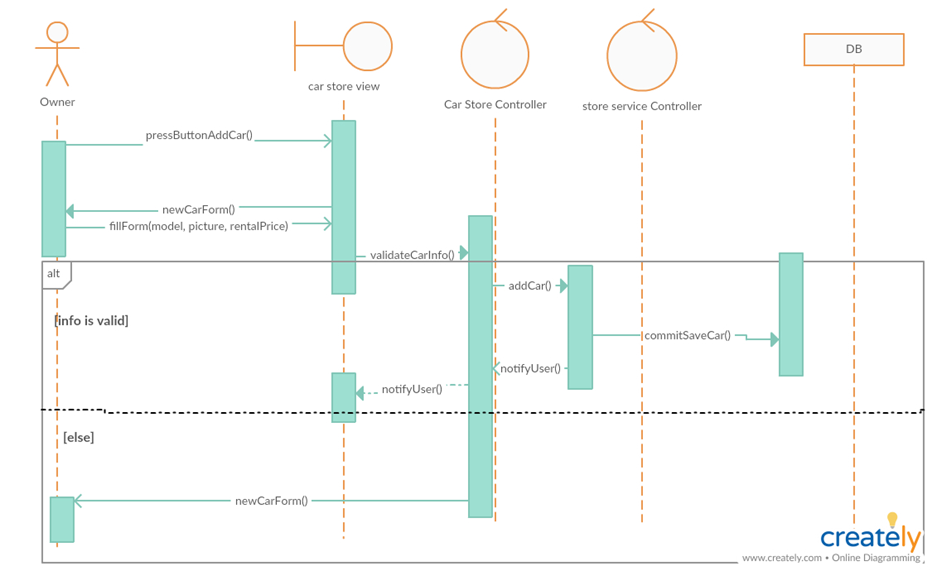


# **Sequence Diagram**

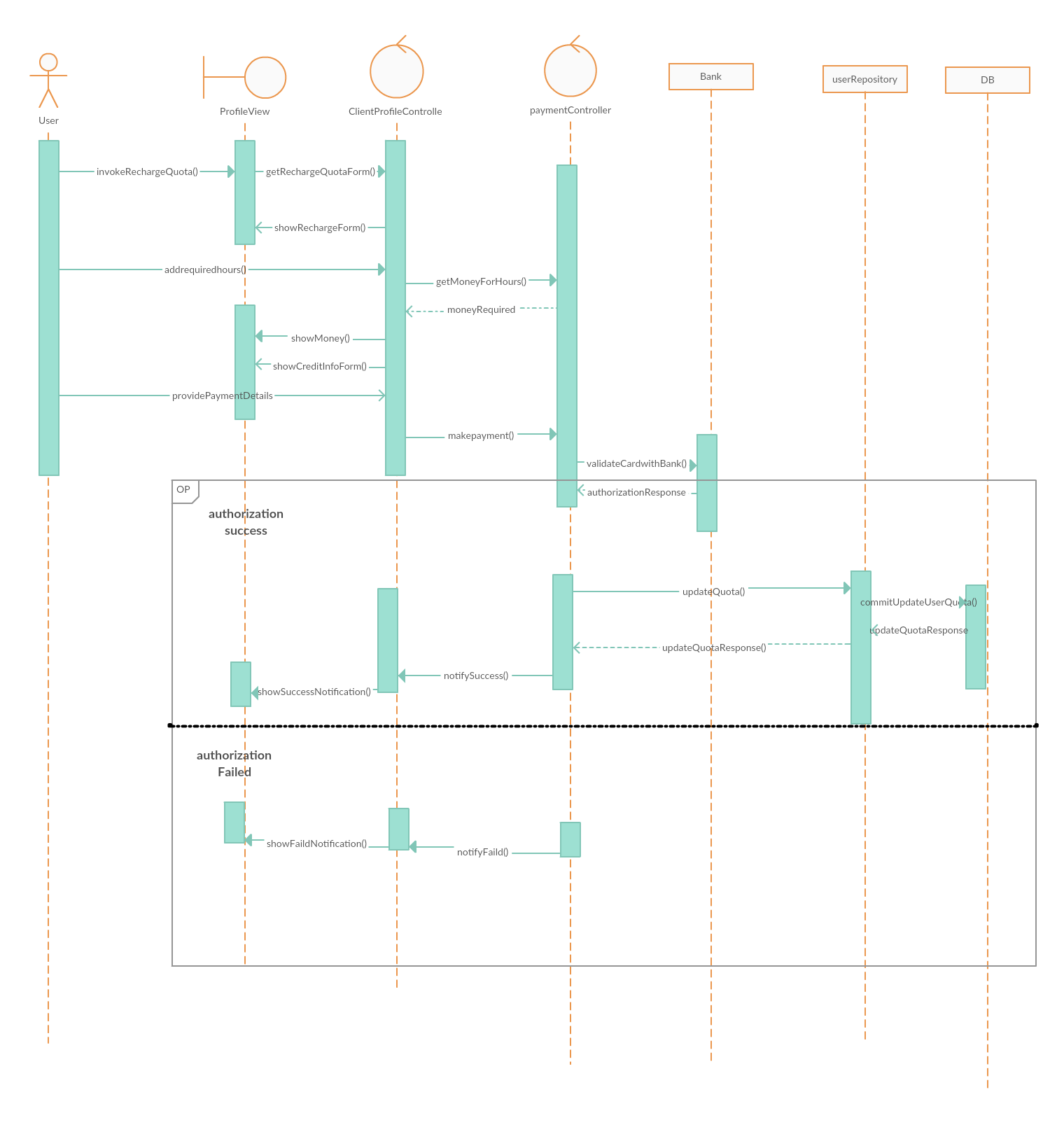
## **Request Car**



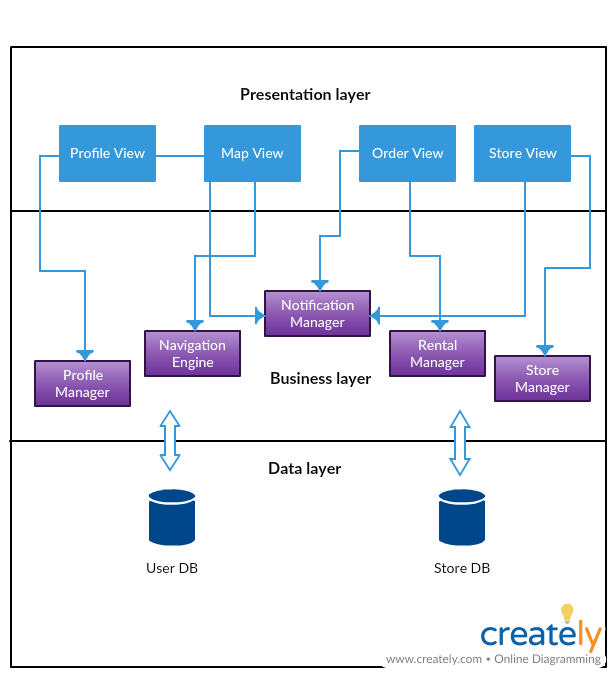
## **Add Car**



## **Recharge Quota**



# **System Architecture**



# **Use Case Tables**

|  |  |  |
| --- | --- | --- |
| Use Case Name: | activate Payment Methods | |
| Actors: | Service User | |
| Entry condition: | Sign in | |
| Exit-conditions: | System redirect user to profile | |
| Flow of events: | **User Action** | **System Action** |
| 1. User clicks on add payment method |  |
|  | 1. System response by presenting card info form. |
| 1. User fills out the form and submit. |  |
|  | 1. System Approved. |
| Exceptions: | **User Action** | **System Action** |
| 1. User enters invalid details. |  |
|  | 1. System rejects payment method – invalid information. |
| Quality requirements | User has valid credit card. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Forget password | |
| Actors: | Service User/Owner | |
| Entre-conditions: | Sign up | |
| Exit-conditions: | System reset password | |
| Flow of events | **User Action** | **System Action** |
| 1. User invoke forget password. | 1. System responds by presenting form to enter Email. |
| 1. User fills out form by entering his email. | 1. System sends form link that user should entering new password in. |
|  | 1. User fills new password. | 1. System send confirmation email that password was changed. |
| Exceptions: | **User Action** | **System Action** |
| 1. User enters invalid password. |  |
|  | 1. Systems ask user for new password |
| Quality requirements | User should change password within 1h. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Add new Car | |
| Actors: | Service Owner | |
| Entre-conditions: | Sign in | |
| Exit-conditions: | System Notify that a new car is added | |
| Flow of events | **User Action** | **System Action** |
| 1. Select Add new car. |  |
|  | 1. System return form contain car information needed. |
| 1. Services owner insert car information and submit. |  |
|  | 1. System approves a new car and add it to database. |
| Exceptions: | **User Action** | **System Action** |
| 1. Users enter invalid information |  |
|  | 1. System ask for new car info. |
| Quality requirements |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Sign up | |
| Actors: | Service User/Owner | |
| Entry condition: | User start the application | |
| Exit condition: | The system redirect user to home activity | |
| Flow of events | **User Action** | **System Action** |
| 1. The user activates Sign-Up function | 1. The system responds by presenting a form to the user |
| 1. The User fills out the form and submit it. | 1. System approves information and ask user for the confirmation code that sent via email. |
| 1. The User fills confirmation code. | 1. The system save user and redirect to home activity |
| Exceptions: | **User Action** | **System Action** |
| 1. The User fills invalid data. | 1. The System rejects user as required information not completed – invalid information. |
| 1. The User fills invalid registration code. | 1. The system asks to re-enter code. |
| Quality requirements | * User activates confirmation code within 1h. | |

# **Future Work**

* Application will work on other platforms.
* Application will cover more places.
* Add more rental devices