****

**Renting A Car Mobile Application**

Keerthana Nimmagadda 1893679

Durga priya Kalam 1892807

Sai Manasa Goka 1892793

Padma Prabhasa Inturi 1893686

Navya Damuluri 1892832

Kowkshitha idamakanti ranga 1893145   
Rajwinder singh 1795544

Navdeep kaur 1795515

**Software Requirements Specification**

**Document**

**Version: 3** **Date: 26/10/2019**

**Table of Contents**

**1.** [**Introduction**](#Introduction) **02**

*[1.1 Purpose](#Purpose) 03*

*[1.2 Scope 04](#Scope)*

*[1.3 Definitions, Acronyms, and Abbreviations](#Definitions) 04*

*[1.4 Overview](#Overview) 04*

**[2.](#Overall_Description)** **[The Overall Description 04](#Overall_Description)**

**[3. Specific Requirements 06](#Specific_Requirments)**

*[3.1 Product Functional Requirments 06](#Functional_Requirments)*

*[3.2 Non-Functional Requirments 07](#NonFunctional_Requirments)*

**[4. Data Model 10](#DataModel)**

[*4.1 Data Objects (Use case Diagram) 10*](#DataObjects)

*[4.2 Data Objects (Class Diagram) 11](#DataObjects_sequence)*

*[4.3 Data Objects (Sequence Diagrams) 12](#DataObjects_database)*

**[5. Screens Design](#Screen_Design)****[17](#Screen_Design)**

**[6. Screens Flow Diagram](#Dataflow_Diagram)****[18](#Dataflow_Diagram)**

**7. Database 19**

**8. Database Schema 21**

**9. Web Service calls 22**

**1.Introduction**

Rent A Car Mobile Application was designed and implemented to Rent cars better than the traditional methods. Renting A Car is an Android based mobile application will be helpful to the people who are looking for cars for rent in temporary bases. The project Renting A Car System keeping all these things in mind provides a total solution to the field of auto rental industry. The idea behind the proposed mobile application is to develop consistent, robust and user-friendly App that allows customers to rent any vehicle as per his/her needs. This Application will maintain the information about the customer details, vehicle details, booking details and transaction details of the customer.

This App feature enables customers to quickly rent a car without any hassle. By using this app, customers can easily get a rental car. This makes the app more useful who don't have an own car. Renting a car is an app built for android smart phones using the ADT bundle (64 bit) package. Renting a car uses technologies such as Android SDK and XML as front-end technology and PHP as middleware, JAVA and MYSQL database as backend technology.

Smartphones are designed to enhance flexibility, usability, and functionality of the communication system. There are a variety of operating systems available for smartphones and android platform was the choice of preference for the app development in this project. Android architecture consists of Linux kernels, libraries and APIs written in C and inbuilt java compatible libraries for developers and an inbuilt server, a free type of software and open source license, aspect that makes it very attractive among developers. Hence Android architecture was used to develop an application for this project.

**1.1 PURPOSE**

Renting A Car Mobile Application is a car rental service application. This application is managed by Rental service Team. A car rental is a vehicle that can be used temporarily for a fee during a specified period. Getting a car for rent helps people get around despite the fact they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who needs a car must contact a rent a car company and contract out for a vehicle with valid documents like license. This system increases customer retention and simplify vehicle and staff management.

**AIMS AND OBJECTIVES**

* To produce a mobile based application that allow customer to register and reserve cars through this application and for the company to effectively manage their car rental business
* To ease customers task whenever they need to rent a car

**1.1 SCOPE**

This project traverses a lot of areas ranging from business concept to computing field and required to perform several researches to be able to achieve the project objects.

The area covers include:

* Car rental mobile application: This includes study on how the car rental business is being done, process involved and opportunity that exists for improvement.
* Android used for the development of the application.
* Company’s staff will be able to use the system effectively.
* Mobile platform means that the system ease to use and will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal.
* It increases the efficiency of the management at offering quality services to the customers.
* It provides custom features development and support with the software

This section describes the features which are in the scope of the developed application.

* View the main screen with options
  + clerk login - For existing Clerk's login.
  + Admin login - For Admin.
* Clerk Login
* Upon login Clerk can view a Dashboard with options
  + - Home screen
    - View cars
    - Search cars
    - Add new cars
    - Rent a car
    - Reserve a car
    - Return service
    - Reports (Daily/ Weekly and monthly)
* Admin Login
* Upon login Admin can view a Dashboard with options
  + - Home screen
    - Add or register, modify and delete Staff (Clerk role)
    - View transactions
    - Search transactions
    - View cars
    - Search cars
    - Add new cars
    - Reports or Transactions (Daily/ Weekly and monthly)
  + Upon registration, the Admin can register clerks by using their name, username, password, mobile number and email address. The details are stored in the database.
  + Upon giving the valid credentials, the staff (clerk) will be successfully registered with the application. Now the staff can log in and use all the features of the application.

**3. Specific Requirements**

**3.1. Functional Requirements**

Functional requirements define specific functionality that define what a system is supposed to accomplish. A function is described as a set of inputs, the behavior, and outputs. Requirement analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product, taking into consideration the possible conflicting requirements of the various users.

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data should the system holds and the interfaces with the user

Following are some functional requirements:

* The system allows Admin, clerk, and customers to login using their username and password.
* The system allows the Admin to create/modify/delete clerk’s account
* Customer can view detail description of vehicle based on his need.
* The system provides option for advanced search for different categories of vehicles. E.g. By Brand, Type and Model.
* The system must allow the customers to view list of available vehicles during reservation and select specific vehicle using different search category while reservation.
* Clerks can also cancel the reservations by using booking reference number provided during the reservation process.
* Admins, Clerks have access to check the status of the vehicle.
* Clerk can update information of vehicle using his own unique id.
* Clerk can view reservations made by customers.
* Clerk can register and update customers details into rental list.
* Clerk has access to search rent record of customers using specific categories.
* Clerk can view all customers car rental record.
* Clerk / Admin can add/delete vehicles.
* Users can search vehicles by specific type, model or year.
* Clerk can update information of the vehicle.
* Clerks can update all lists of vehicles by which customers can see in the catalogue.
* Admin / Clerk can search the history of transactions made by customers booking reference.

**3.2. Non-functional requirements**

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system implementation such as the capabilities of I/O devices or the data representations used in interfaces with other systems. Non-functional requirements, such as performance, security, or availability, usually specify or constrain characteristics of the system.

**Usability**

* The system provides help and support menu in all interfaces for the user to interact with the system.

**Security**

* The system provides security to the customer by encrypting all their personal data.

**Performance**

* The system response time for every instruction conducted by the user must not exceed more than a minimum of 10 seconds.
* Ability to maintain mass number of customers on the server at once without crashing

● Speedy performance / transmission of data

● Send any emails immediately

● Have a quick recovery time if anything were to go wrong

● Display accurately and efficiently on all devices (responsive view)

**Error handling**

* In case of system crash, application recovers the user data by maintaining backup of customer records in multiple databases.

**Availability**

* The system is available for 24 hours, 7 days a week.

**Quality Attributes:**

● Maintain a user-friendly environment that is visually appealing

● Easy to see and use navigation

● Maintain readable content

● Searching cars should be accessible to people who are and are not logged in

● Selecting and making a payment should be available to customers who are and are not logged in.

**Security Requirements:**

● Secure any transmissions of private information between the customer and the

company

● Prevent any potential threats such as SQL injections through the forms or search boxes.

● Prevent third party users at administration level

● Verify website security certificates (that lock in the address bar)

● Prevent false information from being used as payment

● Prevent false email inputs from being used when registering

**3.3. OTHER REQUIREMENTS**

**Hardware Requirements**

* Device: Android Smartphone
* Memory Space: 32 GB of disk space, 4 GB of RAM for development machine
* Connector: USB cable to export the app from development machine to Android device.
* 1 GB for Android SDK, emulator system images and cache.

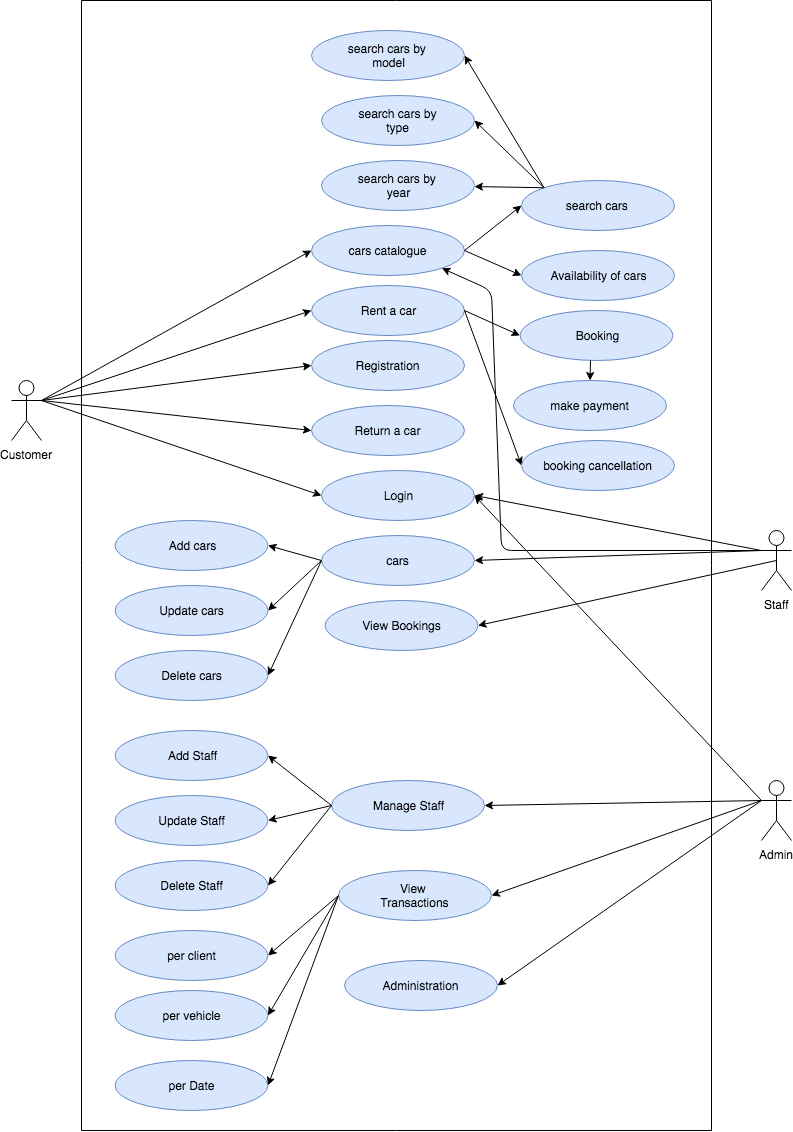
**Software Requirements**

* Eclipse JAVA IDE EE for web developers.
* Android SDK and AVD installed on development machine.
* Java (JDK) 1.6
* Android Plug-in
* MYSQL for database

**4. ANALYSIS MODELS**

**4.1 Use Case Diagrams**

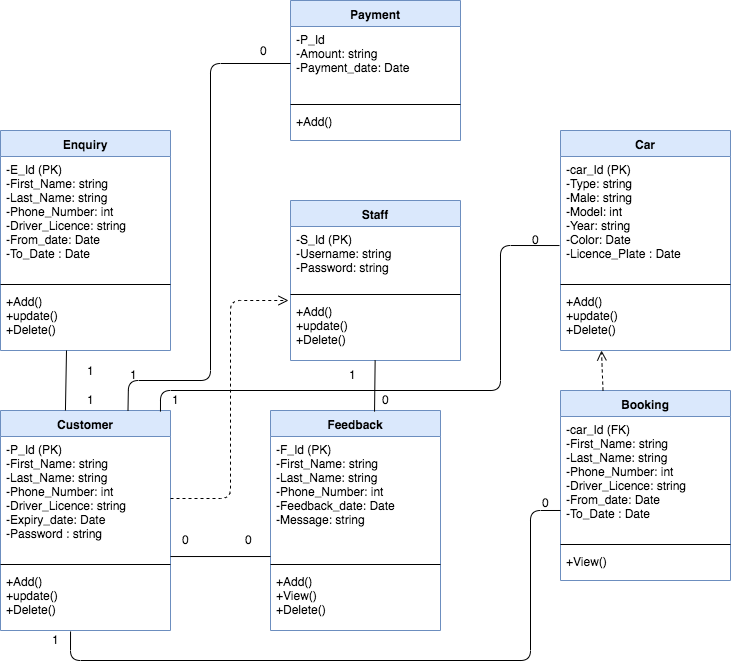
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



**4.3 Class Diagrams**

A class model is the most fundamental model for a system to be done. This is because it describes what is changing within a system. This can be simply viewed with a Design Class Diagram as the class model mainly specifies data structures and operations that the behavioral and functional models operate on. This leads me to the key feature (subject) of a class model – Classes Elements of class model include:

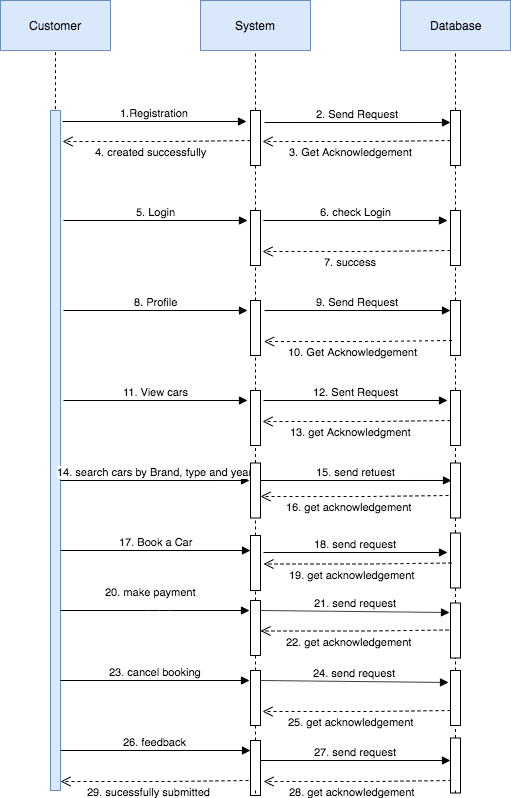
* Identity
* Relationships
* Attributes
* Methods



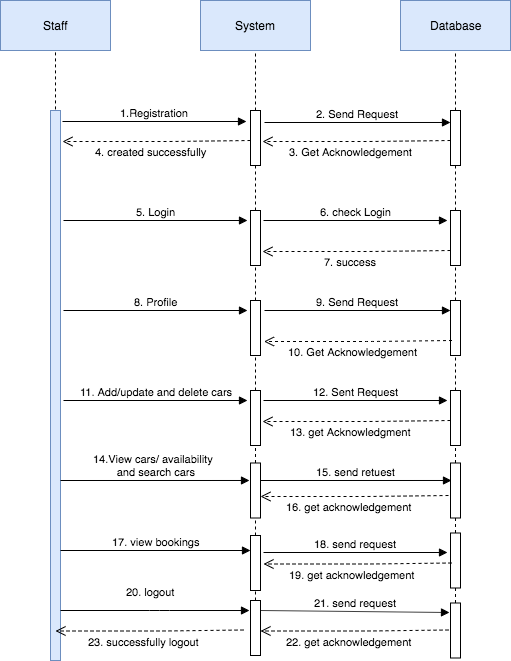
**4.2 Sequence Diagrams**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

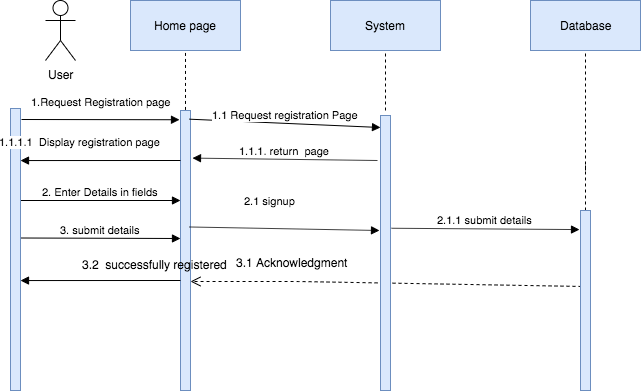
**Sequence Diagram for Customer**

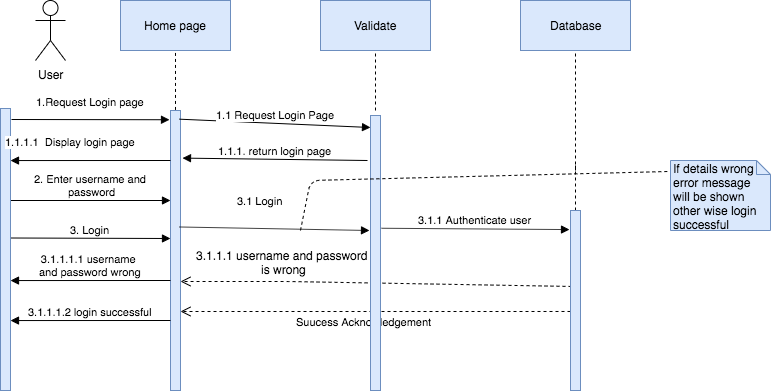


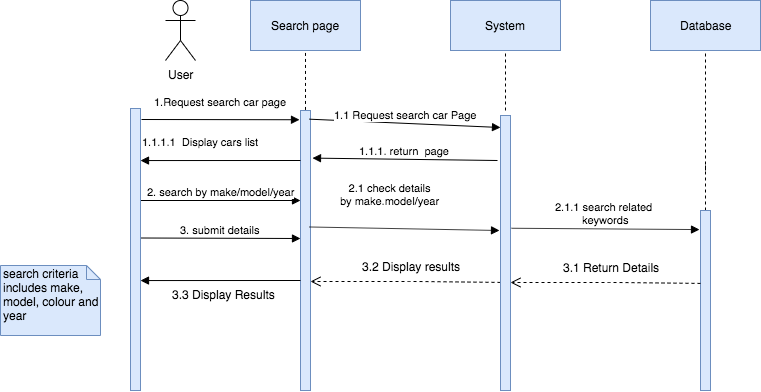
**Sequence Diagram for Staff**



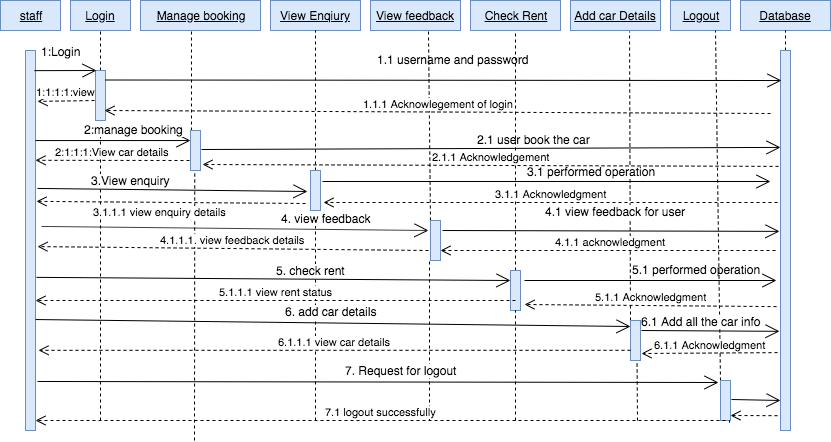
**Sequence Diagram for Registration**



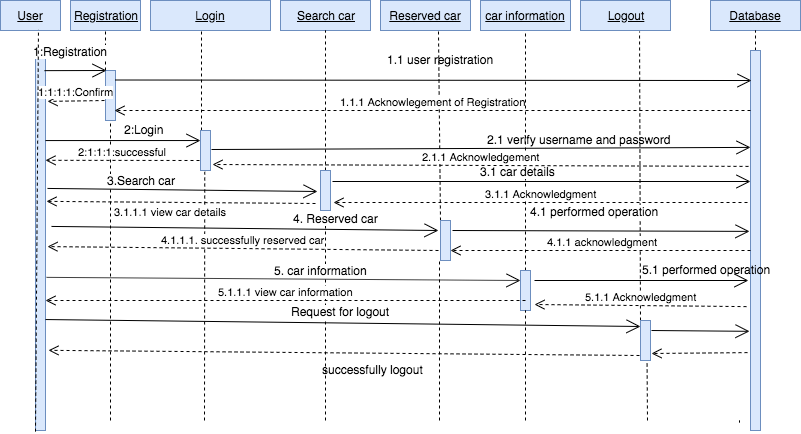
**Sequence Diagram for LoginSequence Diagram for Search car**



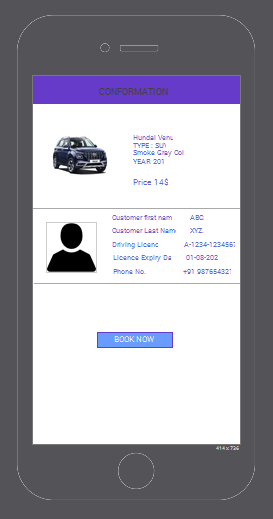
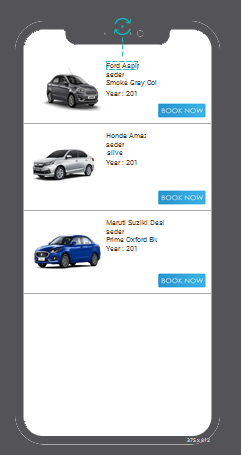
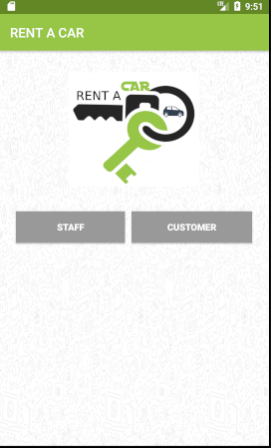
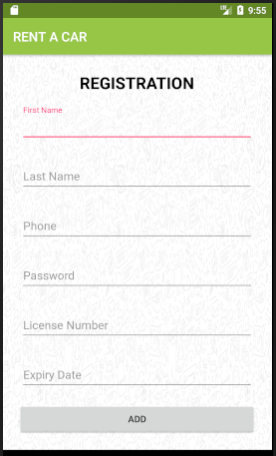
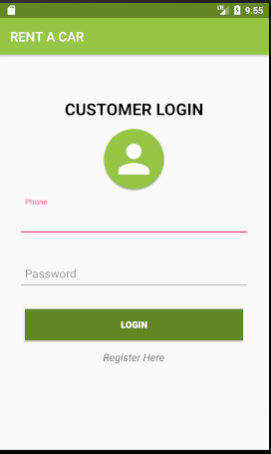
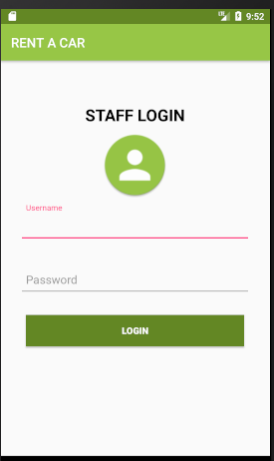
**Sequence Diagram for all actions performed by Staff**



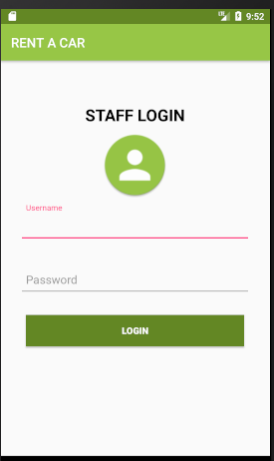
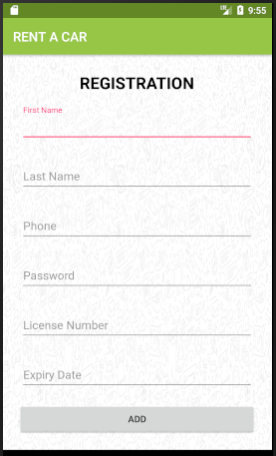
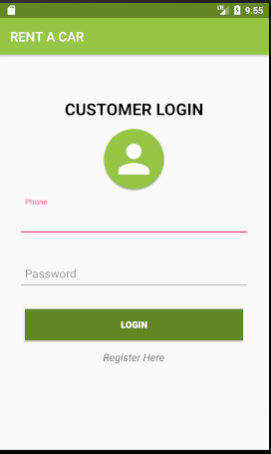
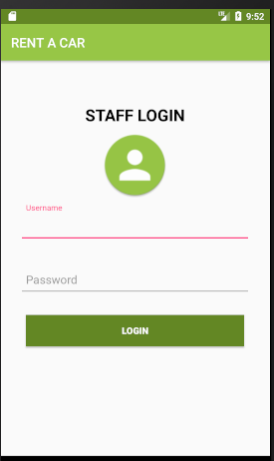
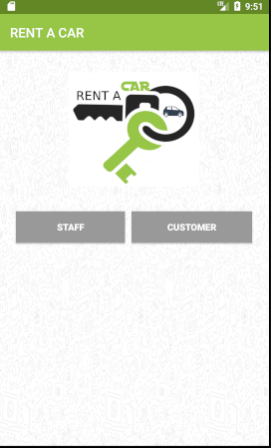
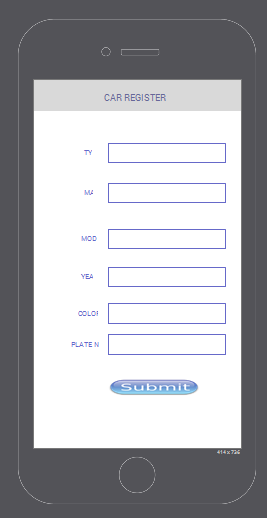
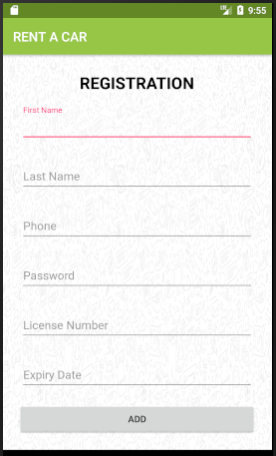
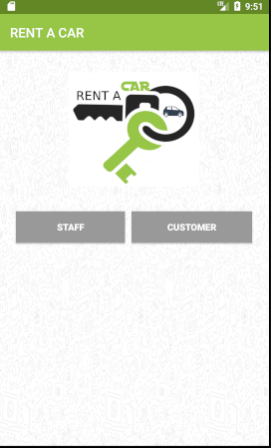
**Sequence Diagram for all actions performed by User**



**5. Screens Design**



**6. Screen Flow Diagrams:**



**7. DATABASE**

In order to create database, we just need to call this method open or create Database with your database name and mode as a parameter.

**CREATING TABLES**

Once database is created, it’s time to create some tables in the database. The CREATE TABLE statement creates the database.

**Client Table Syntax**

A client record consists of an Id, first and last name, driver licence (unique number), expiration date and phone number.

**CREATE TABLE `client\_record` (**

**`Id` int(11) NOT NULL,**

**`first\_name` varchar(50) NOT NULL,**

**`last\_name` varchar(50) NOT NULL,**

**`driver\_licence` varchar(50) NOT NULL,**

**`expire\_date` date NOT NULL,**

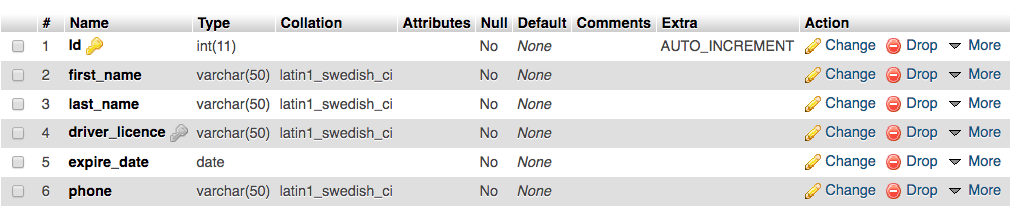
**`phone` varchar(50) NOT NULL**

**) ENGINE=MyISAM DEFAULT CHARSET=latin1;**

**ALTER TABLE `client\_record`**

**ADD PRIMARY KEY (`Id`),**

**ADD UNIQUE KEY `driver\_licence` (`driver\_licence`);**

****

**Vehicle Table Syntax**

A vehicle record consists of an Id, type, make, license plate (unique number), year, color and model.

**CREATE TABLE `vehicles` (**

**`Id` int(11) NOT NULL,**

**`type` varchar(50) NOT NULL,**

**`make` varchar(50) NOT NULL,**

**`model` varchar(50) NOT NULL,**

**`year` date NOT NULL,**

**`color` varchar(50) NOT NULL,**

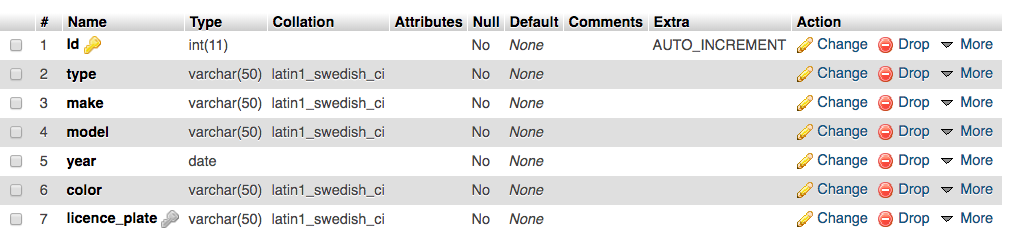
**`licence\_plate` varchar(50) NOT NULL**

**) ENGINE=MyISAM DEFAULT CHARSET=latin1;**

**ALTER TABLE `vehicles`**

**ADD PRIMARY KEY (`Id`),**

**ADD UNIQUE KEY `licence\_pate` (`licence\_plate`);**



**Staff Registration Table**

A registration record consists of an Id, name, email, phone and password. By using email and password clerk will login into applications.

CREATE TABLE `staff` (

`Id` int(11) NOT NULL,

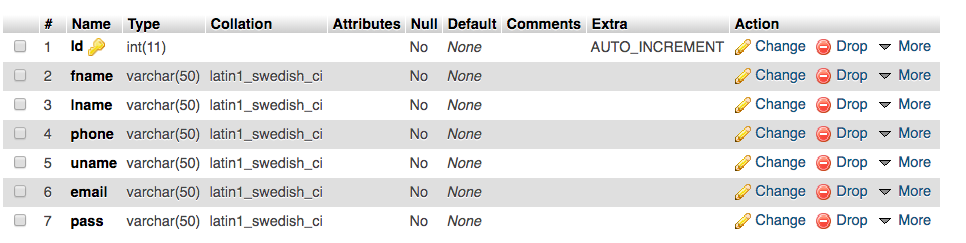
`name` varchar(50) NOT NULL,

`email` varchar(50) NOT NULL,

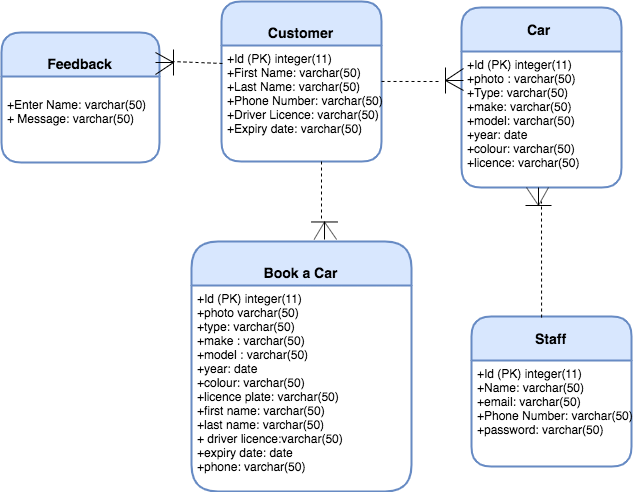
`phone` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;



**8. Database Schema:**

****

**9. Web services Calls:**

|  |  |
| --- | --- |
| URL  Method | <http://possakrishna.com/Rentacar/register.php>  POST |
| Call | Customers’s Sign up |
| Parameters | Fname = Keerthana  Lname = Nimmagadda  Email = keerthana@gmail.com  Password = 123  PhoneNumber =123  Driver’s Licence Number = A-1234-123456  Expiry Date =22/12/2022 |
| Response | Case1  {“Status”:”OK”,  “Message”: “You have been registered.”  }  Case2  {  “Status”: “ERROR”,  “Message”: “This email is already taken, please use a different email.”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | | “Message” | A piece of information for the user | Mandatory | | “Phone” | The phone of the user | Mandatory | | “Password” | The password of the user | Mandatory | |

|  |  |
| --- | --- |
| URL  Method | <http://possakrishna.com/Rentacar/login.php>  POST |
| Call | Customers’s Login In |
| Parameters | PhoneNumber =123  Password = 123 |
| Response | Case1  {“Status”:”OK”,  “Message”: “Login Successfully.”  }  Case2  {  “Status”: “ERROR”,  “Message”: “Invalid Details”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | | “Message” | A piece of information for the user | Mandatory | | “Phone” | The phone of the user | Mandatory | | “Password” | The password of the user | Mandatory | |

|  |  |
| --- | --- |
| URL  Method | <http://possakrishna.com/Rentacar/addstaff.php>  POST |
| Call | Add Staff |
| Parameters | Name = Keerthana  Email = keerthana@gmail.com  Password = 123  PhoneNumber =123 |
| Response | Case1  {“Status”:”OK”,  “Message”: “New Staff Added Successfully”  }  Case2  {  “Status”: “ERROR”,  “Message”: “This email is already taken, please use a different email.”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | | “Message” | A piece of information for the Staff | Mandatory | | “Phone” | The phone of the Staff | Mandatory | | “Password” | The password of the Staff | Mandatory | |

|  |  |
| --- | --- |
| URL  Method | <http://possakrishna.com/Rentacar/stafflogin.php>  POST |
| Call | Staff Login In |
| Parameters | PhoneNumber =321  Password = 321 |
| Response | Case1  {“Status”:”OK”,  “Message”: “Login Successfully.”  }  Case2  {  “Status”: “ERROR”,  “Message”: “Invalid Details”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | | “Message” | A piece of information for the user | Mandatory | | “Phone” | The phone of the staff | Mandatory | | “Password” | The password of the staff | Mandatory | |

|  |  |
| --- | --- |
| URL  Method | <http://possakrishna.com/Rentacar/addcar.php>  POST |
| Call | Add Car |
| Parameters | Photo = Keerthana  type = SUV  make = Jeep  model = Wrangler  year = 2019  Licence Plate = XCB 468  color = Black |
| Response | Case1  {“Status”:”OK”,  “Message”: “Car Added Successfully.”  }  Case2  {  “Status”: “ERROR”,  “Message”: “Server Issue”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | | “Message” | Car added successfully | Mandatory | | “Photo” | The photo of the car | Mandatory | | “type” | The type of the car | Mandatory | |

|  |  |
| --- | --- |
| URL  Method | [http://possakrishna.com/Rentacar/getcars.php](http://possakrishna.com/Rentacar/addcar.php)  GET |
| Call | View Cars |
| Parameters | Photo = Keerthana  type = SUV  make = Jeep  model = Wrangler  year = 2019  Licence Plate = XCB 468  color = Black |
| Response | Case1  {  “Status”: “ERROR”,  “Message”: “Server Issue”  }  Case2  {  “Status”: “ERROR”,  “Message”: “No cars Found”  } |
| Keys | |  |  |  | | --- | --- | --- | | “Status” | Status of the response  Possible values :“OK”, “WRONG”,“ERROR” | Mandatory | |