**Final Project Report**

****

**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

**Version: Final Report** **Date: 15/11/2019**

**Table of Contents**

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

*[1.1 Purpose](#Purpose) 03*

*1.2 Aims and Objectives 04*

*[1.3 Scope 04](#Scope)*

*[1.4 Definitions, Acronyms, and Abbreviations](#Definitions) 05*

*[1.4 Overview](#Overview) 06*

**[2.](#Overall_Description)** **[The Overall Description 06](#Overall_Description)**

*[2.1 System Interfaces 07](#Prodcut_perspective)*

*[2.2 User Characteristics 07](#User_Characteristics)*

*[2.3 Apportioning of Requirements 08](#Assumptions)*

**[3. Specific Requirements 09](#Specific_Requirments)**

*[3.1 Product Functional Requirments 09](#Functional_Requirments)*

*[3.2 Non-Functional Requirments 10](#NonFunctional_Requirments)*

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

*4.1 Data Objects (Use case Diagram) 13*

*[4.2 Data Objects (Class Diagram) 14](#DataObjects_sequence)*

*[4.3 Data Objects (Sequence Diagrams) 15](#DataObjects_database)*

**[5. Screens Design](#Screen_Design)****[19](#Screen_Design)**

**[6. Screens Flow Diagram](#Dataflow_Diagram)****[20](#Dataflow_Diagram)**

**7. Database 21**

**8. Database Schema 22**

**9. Web Service calls 23**

**10. Conclusion 31**

**11. Bibliography 33**

**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.

**1.2 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

**1.3 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.

**1.4 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Users are the one who accesses the application. (Customers, Staff and Admin) |
| Clerk/Staff(s) | Clerk is the one who manages the car information in this application. |
| Admin | The admin is playing the key role to manage the database and can monitor all the functions in this application |
| Database | The database is working as an internal system here that is storing all the information of the customers, staff and Admin |
| Third-party | The application uses the third-party for financial transactions. (like Internet banking, banking institutions, and so on) |
| Review | The rating, comments that may help the users to improve their experience with this application and cars. Also gives the better idea to the developer team for further improvements. |
| Software Requirements Specification | A report that totally describes all the capacities  of a proposed framework and the requirements under which it must work. |

**1.5 Overview**

The overview of the report gives a brief explanation of this application in the part of the introduction. It reflects the purpose to create this application and help to understand the major characteristics. It also explains the main and quality features of Rent A Car application. In this part, we briefly discussed the scope and advantages of using this application. Also, in the Definitions, Acronyms, and Abbreviations part depicts the active characters along with the definitions. The reference part of this document labels the links of all sources from where this document is inspired and took help in writing.

The Overall Description gives an outline of all the functionalities in the application. The Use Case diagram is used to represent the structure of the application. This report explains the interaction between the user and the application. The hardware and software requirements section of this report lights up all the software requirements and platforms to run this application. The functional and non-functional requirements section of the application helps in explaining all the active characters and their functions.

All areas of the report represent a similar application completely, in any case, are expected for various groups of onlookers and along these lines utilize diverse language.

**2. The Overall Description**

The main important requirements to use Rent A Car application is the internet. It is an online application that can't work without accessing the internet. To utilize this application the customer needs to register by providing all the necessary details. The customers can search the car, based on their car type like SUV, Sedan and Truck.

There are three types of users for this application. The users include registered customers, clerk and admin. The customer can book a car by paying online using Internet Banking and some other payment options available in the application. Customer can access all their previous bookings and can also give their feedback. The clerk can handle all the operations like add, modify and delete the car information. Clerk can also access all the booking information.

The access to the application is only allowed to registered users.

**2.1 System Interfaces**

The applications use the Graphical User Interface (GUI) that allows the users to interact with Icons, Images and hyperlinks and other graphics, unlike text-based interfaces.

**2.1.1 Hardware Interfaces**

Android devices have different types of hardware that are built in and accessible to developers. The mobile phone is operated on Mobile data or Wi-Fi association. Sensors, such as a camera, accelerometer, magnetometer, pressure sensor, temperature sensor, and proximity sensor, are available on most devices. Telephony, Bluetooth, and other wireless connections are also accessible to the developer in some form.

**2.1.2** **Software Interfaces**

Software interface is user interface is everything that the user can see and interact with. The mobile application communicates with the database to get the information about the availability of cars. The communication between the database and the application consist of operations concern both reading and modifying operations. The applicant can access this application by using Wi-Fi or a cellular data network.

For the online transactions the application supported by the third party like internet banking for the security purpose.

**2.2 User Characteristics**

There are a several characteristics of this application. The application is specially designed for the people of Canada and also to the people who live in Canada. It is user-friendly like not only a youngster and the senior customer can use and teenager can use it easily. One thing is main the people who are below thirteen are not authorized to access this application. It is available for Android.

**2.3 Apportioning of Requirements**

The clerk can add, update and delete the car by itself and close the ride option will be adding over there. There are many other options like rating a car, booking a car, searching for a car, taking help.

There is additionally an extra feature that helps customers seeking assistance by email, message or a phone call to a customer representative who are 24\*7 available online.

**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.

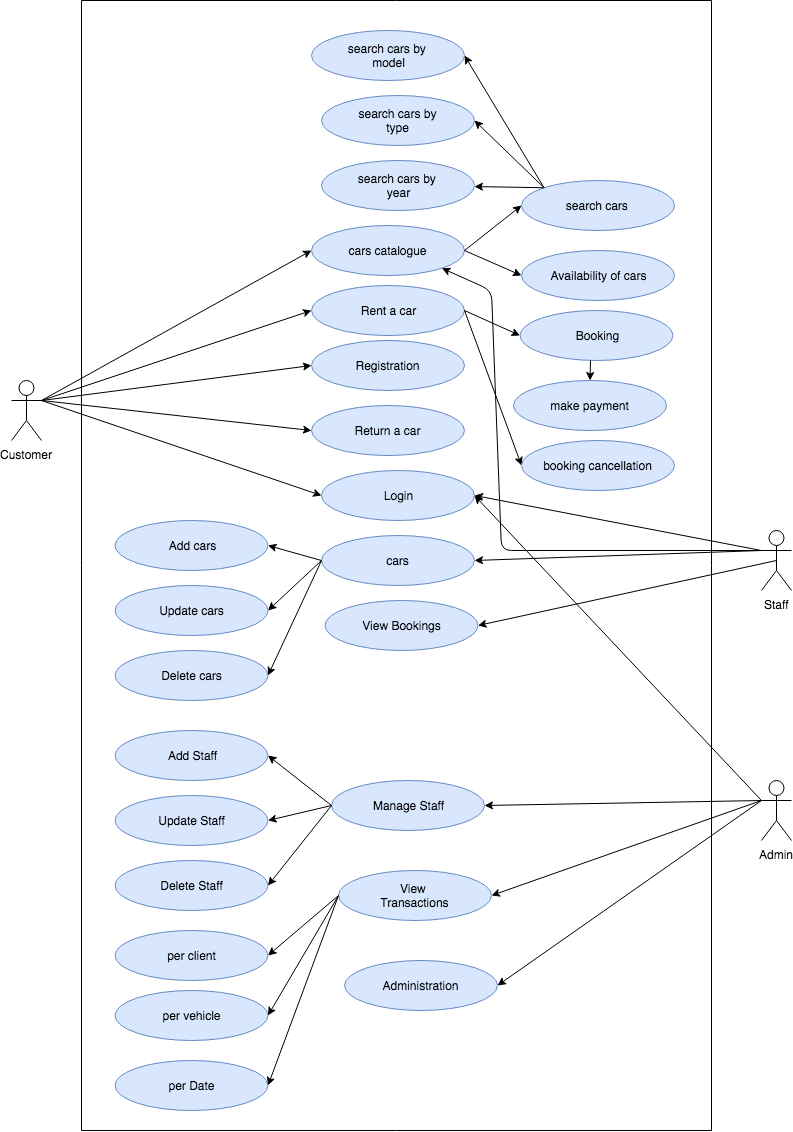
**Software Requirements**

* 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**

The Figure 4.1.1 below represents 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.

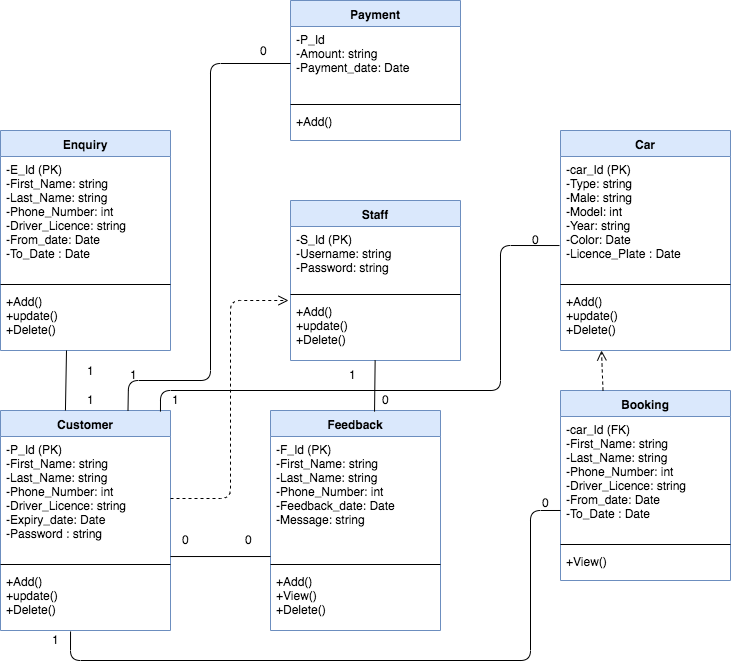


**Fig 4.1.1 Use case Diagram**

**4.2 Class Diagrams**

The Figure 4.2.1 below represents the class diagram which is the most fundamental model for a system to be done. 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

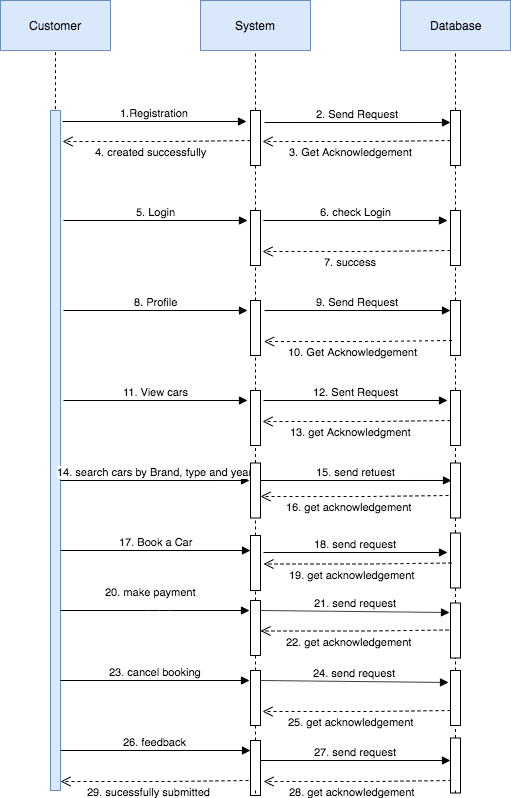


**Fig: 4.2.1 Class Diagram**

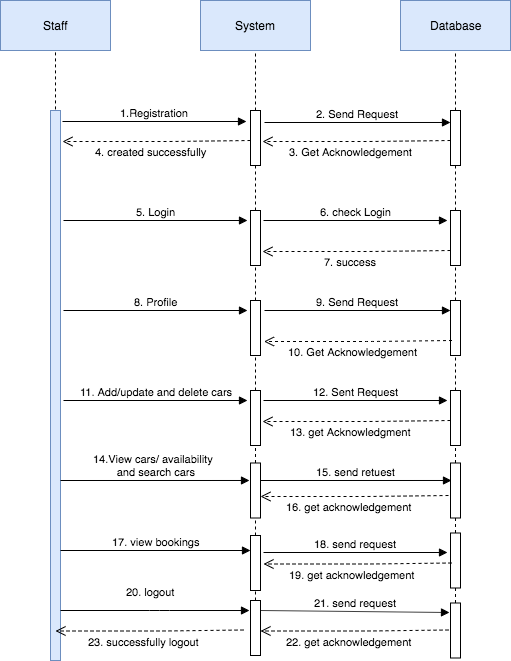
**4.3 Sequence Diagrams**

Figure 4.3.1 and 4.3.2 below represents sequence diagram shows object interactions arranged in time sequence. These figures represent the interaction between the customer and system, Staff and system respectively.

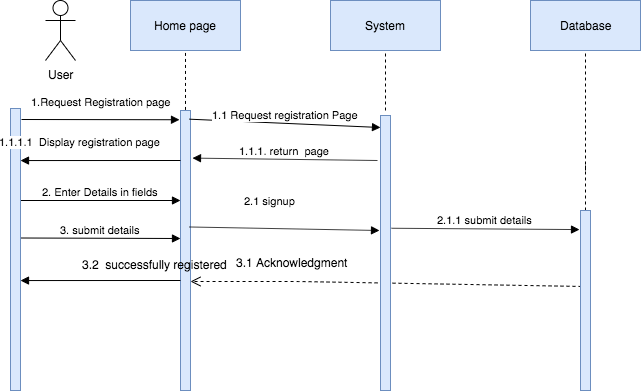
**Fig: 4.3.1 Sequence Diagram for Customer**

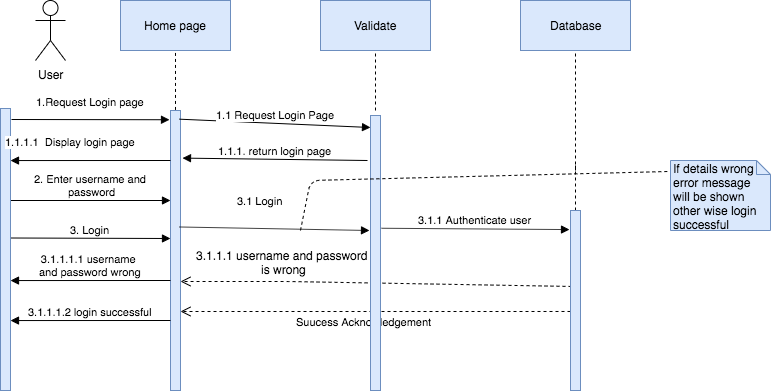


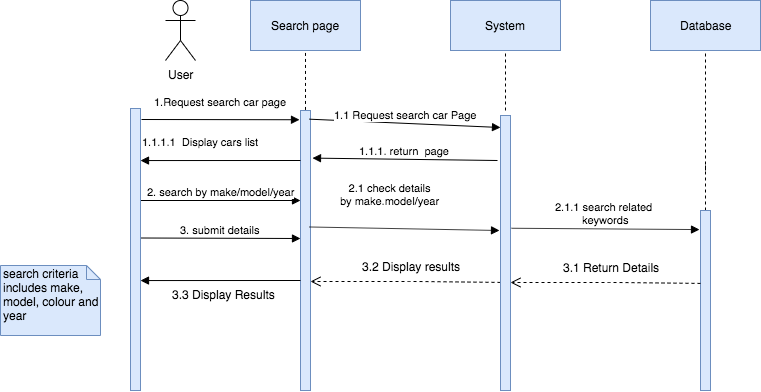
**Fig: 4.3.2 Sequence Diagram for Staff**



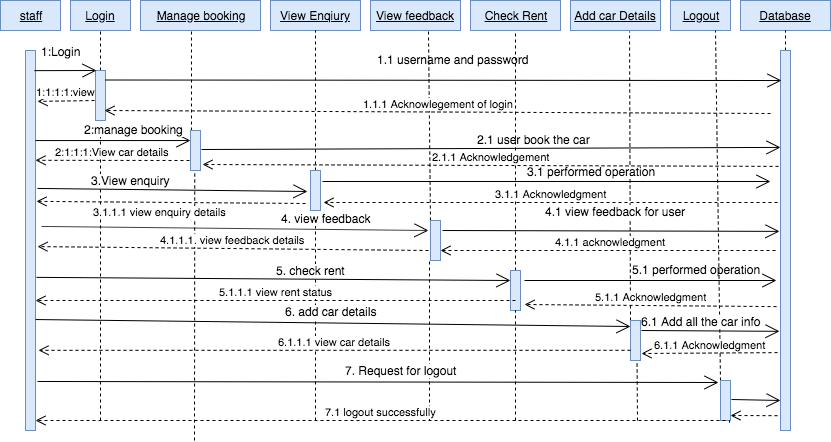
**Fig: 4.3.3 Sequence Diagram for Registration**



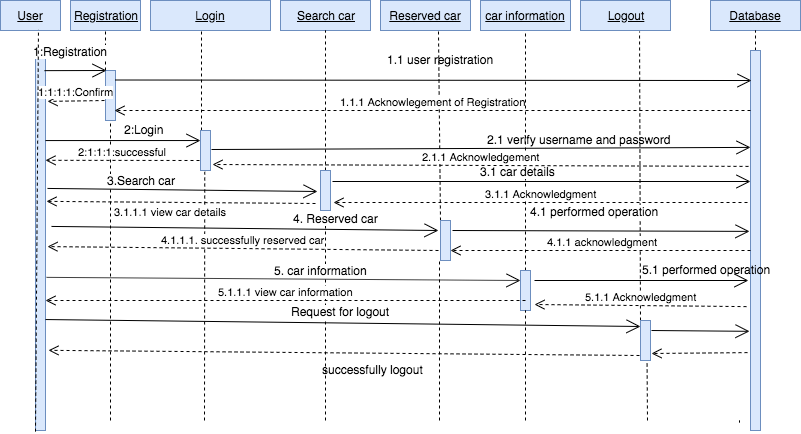
**Fig: 4.3.4 Sequence Diagram for LoginFig: 4.3.5 Sequence Diagram for Search car**



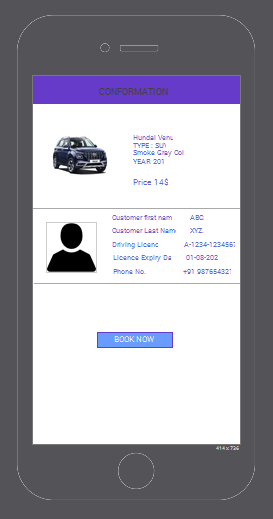
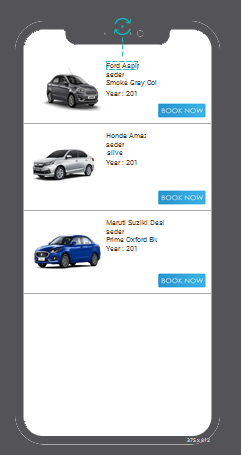
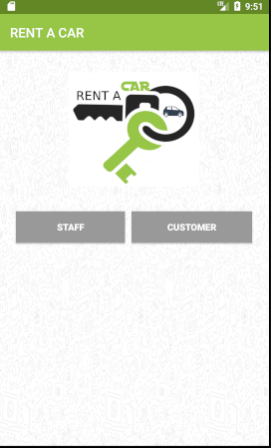
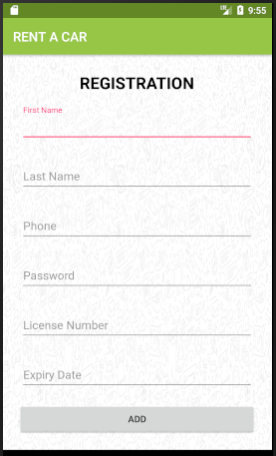
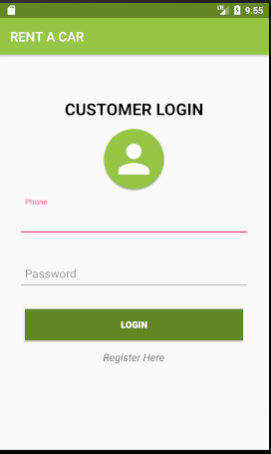
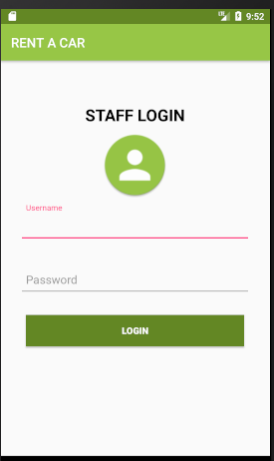
**Fig: 4.3.6 Sequence Diagram for all actions performed by Staff**



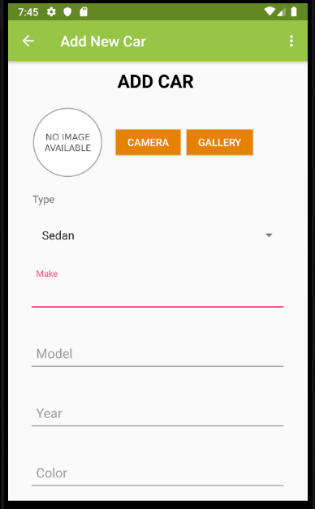
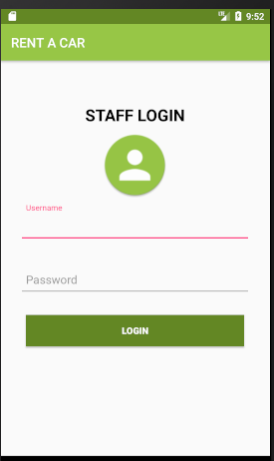
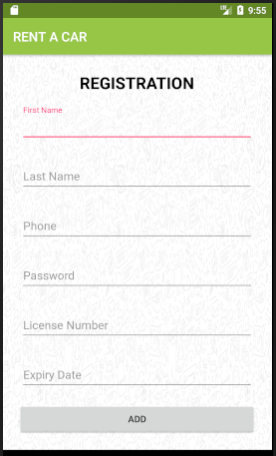
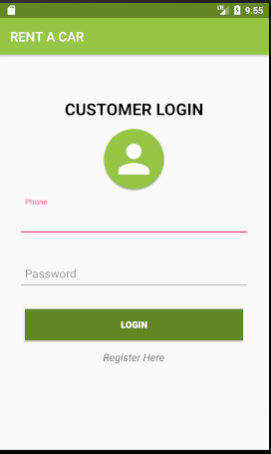
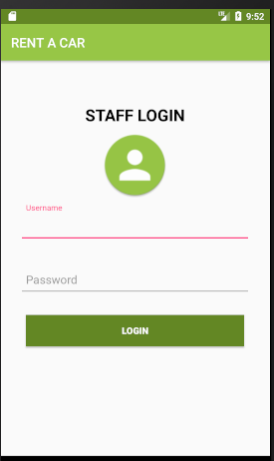
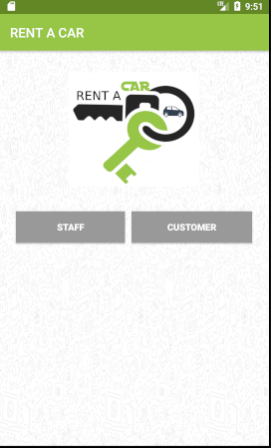
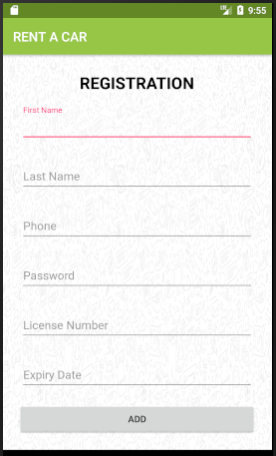
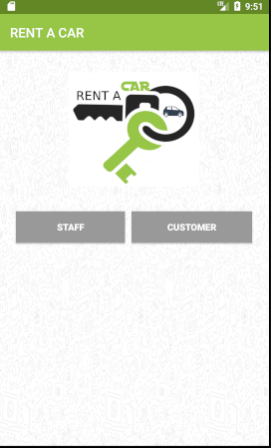
**Fig: 4.3.7 Sequence Diagram for all actions performed by User**



**5. Screens Design**



**6. Screen Flow Diagrams:**



**7. Database**

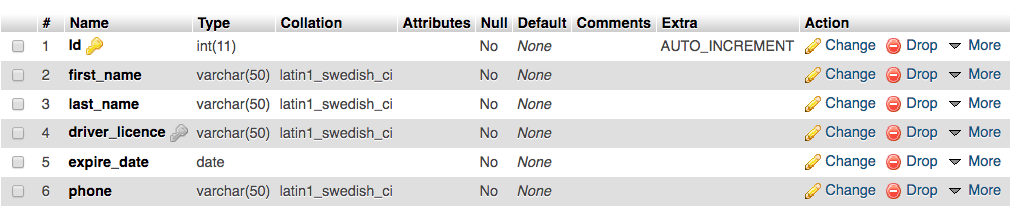
In order to create database in My SQL, we just need to use the methods ‘open or create Database’ with the database name and mode as a parameter. For example, we use “Create database training” to create the database with the name training.

**Creating tables:**

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

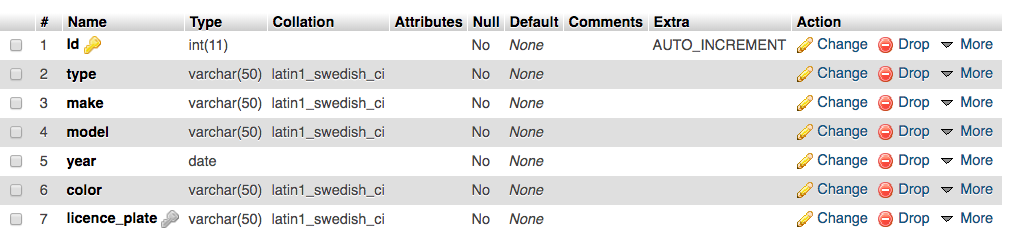
**Client Table Syntax:**

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

****

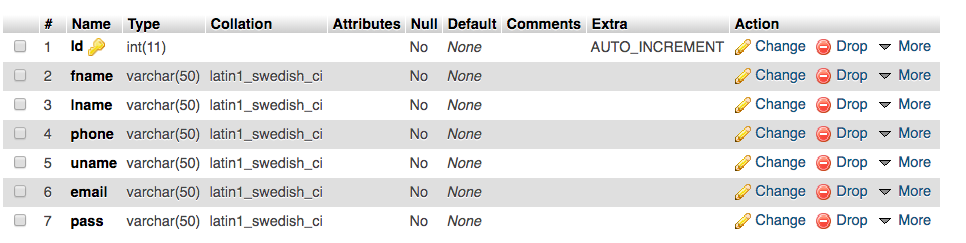
**Vehicle Table Syntax**

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

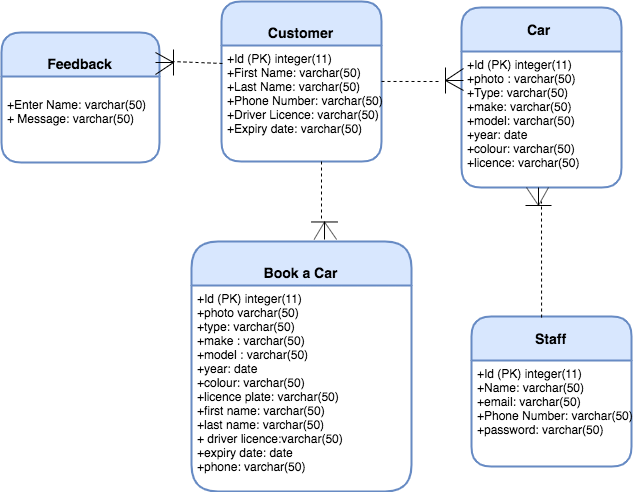


**Staff Registration Table**

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



**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 | |

**10. Conclusion**

The problem that undergo for Car Rental was a lack of real time car renting for their customers. In order to overcome this problem, we develop an android application that would allow for users to purchase their service with ease and admiration. Customers should have been able to rent and view vehicles by logging into the application. Requests should be transferred directly into the database and displayed on the software within the company building to allow for a proper response and review to commence.

The solution was to provide Car Rental with a user-friendly Android application that would allow for customers to access easily. The Application is designed to stay up to date by giving staff the ability to change/add/remove any featured vehicles on the App. The application will verify and store any information the user may input when making a rental purchase request. This will appear in real time onto the application used by the employees. This application provides an easy-to-use interface toallow for simple access to rental requests and customer information.

**11. Bibliography**

**References:**

* Programming Android by O’REILLY
* Beginning Android Tablet Application Development by ROBBIE MATHEWS
* Pressman RS, “Software Engineering”
* R.Fairly, “Software Engineering concepts”
* Uml by GRADY BOOCH
* The Complete Reference JAVA by Herbert Schildt

**Sites Visited:**

* My sql:
* http://www.vogella.com/tutorials/AndroidSQLite/article.html
* Introduction to android 4.4:
* https://developer.android.com/about/versions/android-4.4.html
* Intent filter:
* https://developer.android.com/guide/topics/manifest/intent-filter-element.html
* Different types of intent:
* https://developer.android.com/guide/components/intents-filters.html
* starting activities:
* http://www.vogella.com/tutorials/AndroidIntent/article.html
* what are intents:
* http://www.vogella.com/tutorials/AndroidIntent/article.html
* On destroy:
* https://stackoverflow.com/questions/13927269/use-of-ondestroy-in-android
* On create:
* https://stackoverflow.com/questions/19538976/what-is-a-oncreate-method-in-android
* Activity life cycle:
* https://developer.android.com/guide/components/activities/activity-lifecycle.html
* Android emulator:
* https://stackoverflow.com/tags/android-emulator/info
* Essential command line tools:
* https://developer.android.com/studio/command-line/index.html
* Publishing:
* https://developer.android.com/studio/publish/index.html
* upto developement:
* <https://hheell.files.wordpress.com/2017/06/introductiontoandroidapplicationdevelopment4thedition.pdf>
* android architecture:
* http://www.techotopia.com/index.php/An\_Overview\_of\_the\_Android\_Architecture
* Debugging and testing:
* https://www.d.umn.edu/~gshute/softeng/testing.html
* data transfer to the target component: no change
* ww.developer.android.com
* [www.android.com](http://www.android.com)
* [www.vogella.com](http://www.vogella.com)
* [www.developers.google.com](http://www.developers.google.com)
* [www.sqlite.org](http://www.sqlite.org)
* Android Developer -<https://developer.android.com/index.html>
* MySQL -<https://www.mysql.com/>
* Stack Overflow -<https://stackoverflow.com/>-
* PHP -<http://php.net/manual/en/index.php>
* <https://developer.android.com/design/material/index.html>
* <http://www.java2s.com/Code/Android/CatalogAndroid.html>
* <https://www.javatpoint.com/android-option-menu-example>