|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  [Project Name]  Version: [1.0]   |  |  | | --- | --- | | Project Code |  | | Supervisor |  | | Co Supervisor |  | |  |  | | Project Team |  | | Submission Date |  | |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
|  | |  |
|  | |  |
|  | |  |

Document Sign-Off

[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1. Introduction 6](#_Toc372710946)

[1.1. Purpose of Document 6](#_Toc372710947)

[1.2. Intended Audience 6](#_Toc372710948)

[1.4 Document Convention 6](#_Toc372710949)

[2. Overall System Description 6](#_Toc372710950)

[2.1. Project Background 6](#_Toc372710951)

[2.2. Project Scope 6](#_Toc372710952)

[2.3. Not In Scope 6](#_Toc372710953)

[N/A **Error! Bookmark not defined.**](#_Toc372710954)

[2.4. Project Objectives 6](#_Toc372710955)

[2.5. Stakeholders 6](#_Toc372710956)

[2.6. Operating Environment 6](#_Toc372710957)

[2.7. System Constraints 6](#_Toc372710958)

[2.8. Assumptions & Dependencies 6](#_Toc372710959)

[3. External Interface Requirements 6](#_Toc372710960)

[3.1. Hardware Interfaces 6](#_Toc372710961)

[3.2. Software Interfaces 7](#_Toc372710962)

[3.3. Communications Interfaces 7](#_Toc372710963)

[4. Functional Requirements 7](#_Toc372710964)

[4.1. Functional hierarchy 7](#_Toc372710965)

[4.1.1. Signup **Error! Bookmark not defined.**](#_Toc372710966)

[4.1.2. Login **Error! Bookmark not defined.**](#_Toc372710966)

[4.1.3. Set electronic appliances **Error! Bookmark not defined.**](#_Toc372710966)

[4.1.4. Keep staus of the appliances **Error! Bookmark not defined.**](#_Toc372710966)

[4.1.5. Add gesture **Error! Bookmark not defined.**](#_Toc372710966)

[4.1.6. Recognize gesture **Error! Bookmark not defined.**](#_Toc372710966)4

[4.1.7. Trigger raspberry pi **Error! Bookmark not defined.**](#_Toc372710966)4

[4.1.8. Perform action on the appliances **Error! Bookmark not defined.**](#_Toc372710966)4

[4.2. Use Cases 7](#_Toc372710967)

[4.2.1. Login/Signup use case 7](#_Toc372710968)

[4.2.2. System Use case **Error! Bookmark not defined.**](#_Toc372710969)

[5. Non-functional Requirements 8](#_Toc372710970)4

[5.1. Performance Requirements 8](#_Toc372710971)4

[5.2. Safety Requirements **Error! Bookmark not defined.**](#_Toc372710972)4

[5.3. Security Requirements 8](#_Toc372710973)4

[5.4.User Documentation 8](#_Toc372710975)4

[6. References 8](#_Toc372710976)5

[7. Appendices 8](#_Toc372710978)6

1. Introduction

* 1. Purpose of Document

A Software Requirements Specification (SRS) document defines what the software will do and how it will be expected to perform. It plays an important role in the development process as it defines the scope of the project. It also defines the functional requirements of the product. It helps to make software with right requirements that fulfill the needs of the user.

It helps the designer to design the system and provide all the functional and non-functional requirements of the software. Programmers use the srs as a reference point to understand business needs and requirements.

* 1. Intended Audience

The intended audience of the project includes developers, testers and customer. developer uses this as a reference to start and develop a project and follow its mention requirements .tester test the requirements of srs document to make sure it perform the way as it is excepted.

**1.3 Abbreviations**

**APK** – Android Application Package.  A file format used to deliver mobile apps to Android devices for testing or distribution.

**GUI**– Graphic User Interface.  An interface that allows users to interact through graphical icons and visual indicators.

**IDE** – Integrated Development Environment.  A software application that allows computer programmers to code software for mobile development.

**SDK** – Software Development Kit.  A set of tools that can be used to develop software applications targeting a specific platform.

**UI** – User Interface. The look and feel of your app. This includes design, menus and other elements of the app that the user sees.

* 1. Document Convention

The font used is Arial. Headings’ font size is 16 and for subheadings it is 12(both in bold and italics). Calibri is used for the body with size 11.

1. Overall System Description
   1. Project Background

Rental Car System the mobile app in which user can rent a car in more efficient way. This app provide the features to the user to book the rental car online and also pay online and select the desired car through this app. The app also combines the all the verified rental car companies come on one platform. The user can also do the online payment because it is safe and much quicker and handy in an online environment.

This app provides a platform for the rental companies to register in the app and give their services to the user of rental car. This platform unite several verified companies together and the company which are approved by an administrator. Each registered company can place its cars in the app and the user looking for the particular car and receives the results from all the companies which get user requirements and can select the car and book it for rent.

* 1. Project Scope

The car rental application has the scope in the market place. The scope points of the project are as follow:

* This application will be useful for the people who want to book a desire selected car ride online at any time in the easy way by just few clicks in the application.
* This project will be used as an android based application. It will be viewed or downloaded using play store.
* This application is a very good opportunity for the car dealer’s owners to expand their business.
* This application will be developed in English language will be easily understood by people.
* This application will keep the detail records of both cars and the customer, the duration they rent car as well as the type of the car they rent.
* This application will have the ability to generate and print invoice for each successful transaction.
  1. Not In Scope

There is no requirement or point that is not in scope because every requirement is well documented and analysis.

* 1. Project Objectives
* To provide where user can book a car online easily.
* To provide a secure app that user can book and make booking payment safely.
* Provide car dealers platform to expand their business.
  1. Stakeholders

Stakeholder is a member of groups without whose support the application would not exist. The major stakeholder of the app is the customer, the person who books a car and use the app.

* 1. Operating Environment

Operating environment is the environment in which users run his application. The environment consists of a user interface provided by the application manager. Android studio is the official integrated Development (IDE) for android app development. Generally to build android application we should have java development kit (jdk) and Android SDK (Software development kit).

* 1. System Constraints

Constraints are anything in the application that slows down the system or prevents it from achieving its goal. Whereas car Rental system is well organized and designed to handle any constraints in the application all the constraints are handle in the background that it did not slow down the app or prevent it from his task.

* 1. Assumptions & Dependencies

The Assumption of the project is as follow:

* Customer database consist all the information about a customer with name, age, email, password, password and billing information.
* Car database consist all the general information of the available cars such as name, mode, 3D image, seats and available data etc.
* Booking database contains all the booking information like date of booking, date to pick and drop, name, number etc.

**Resources:** we have all the resources to start the project. The laptop, 64 bit Microsoft window 8/10, 8 GB Ram and 32 processor and android studio. The android studio software is used to develop an android mobile application and it is online free software.

**Project team members:** the project team member has all the coding and designing skills to and software’s to start the project. We have the android studio software for the application and the **member name** has the skill of design application in xml file and **member name2** has the skill of java coding for the backend coding.

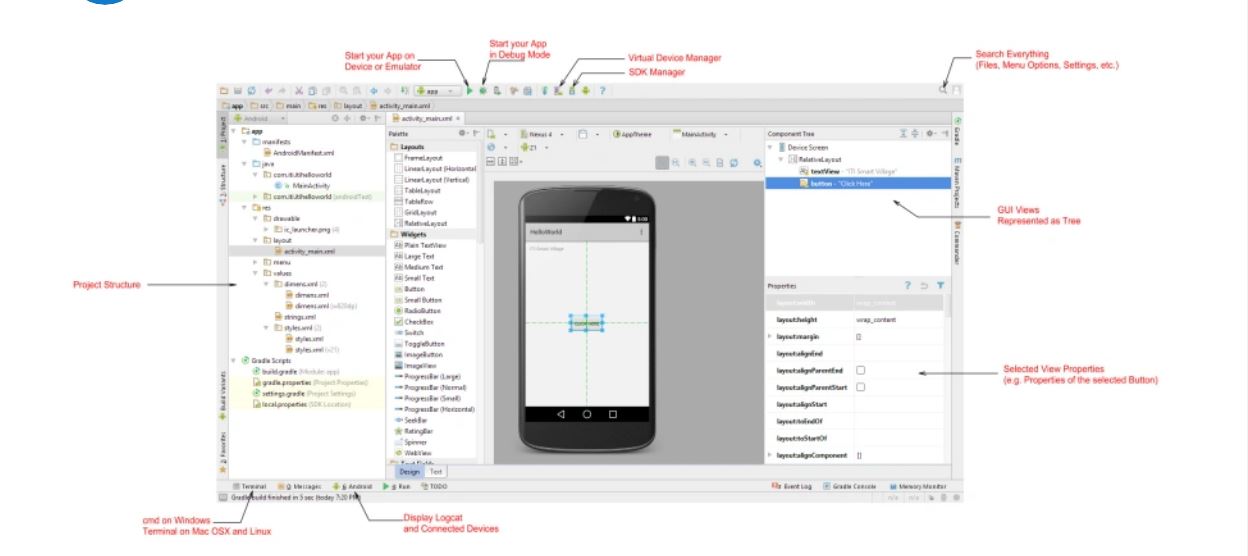
**Time Schedule:**  The project team members have all the resources to complete their individual tasks on time.

1. External Interface Requirements
   1. Hardware Interfaces

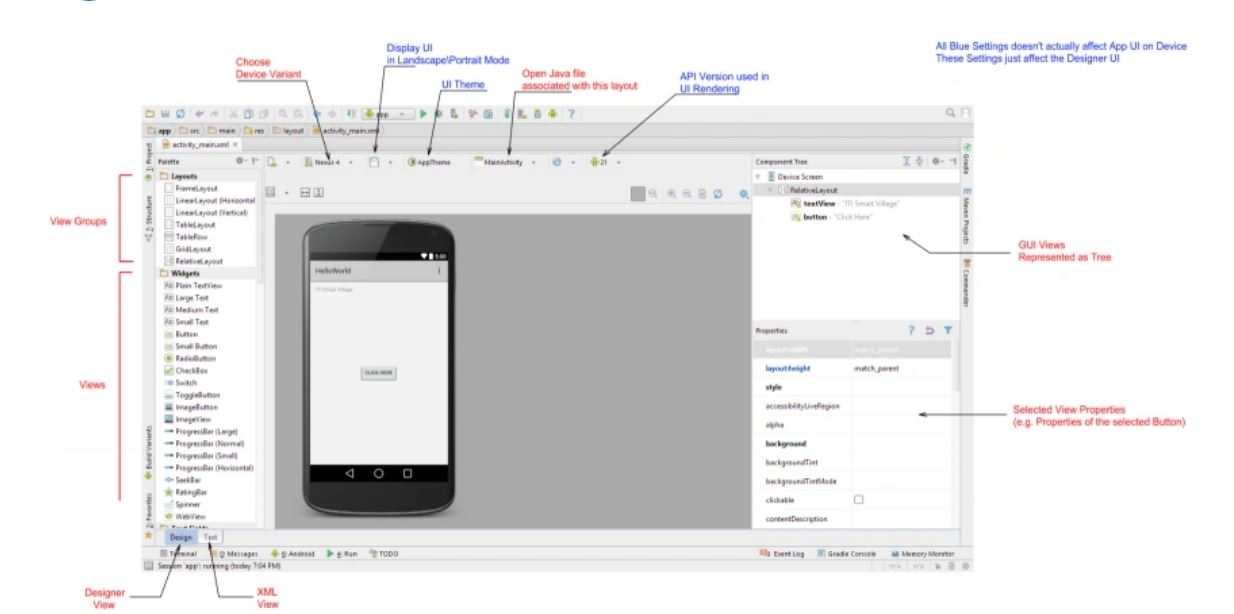
* 64-bit Microsoft® Windows® 8/10/11.
* x86\_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
* 8 GB RAM or more.
* 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
* 1280 x 800 minimum screen resolution.
  1. Software Interfaces

Software interfaces enables users to control and customize their display of an app. It allows customizing the design of the application. The main software interfaces are mention below:

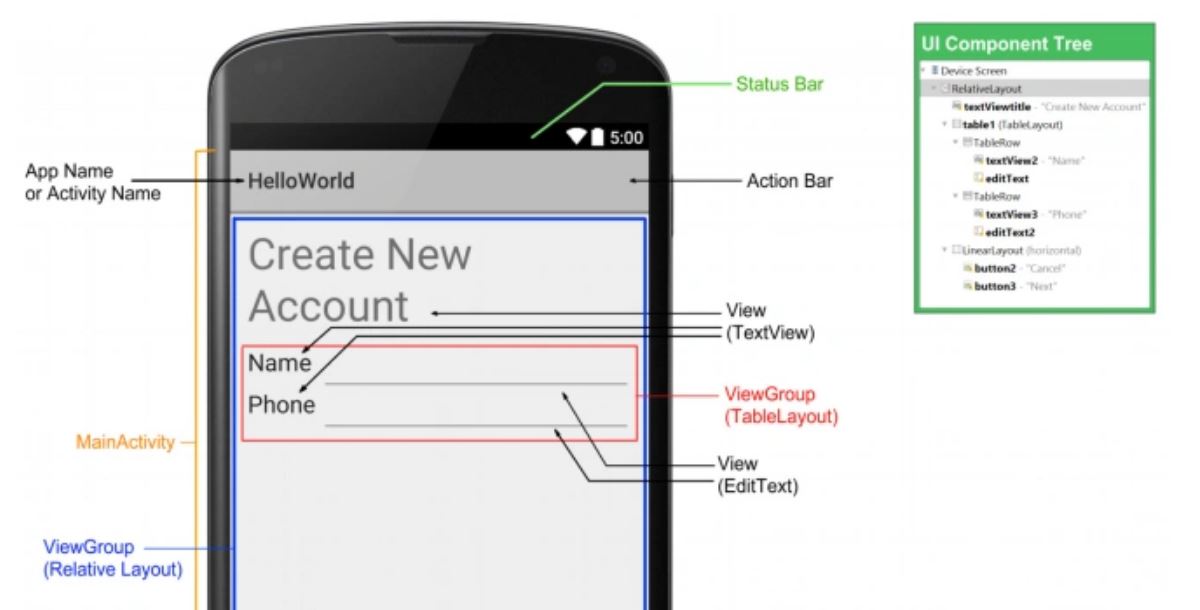
* Main UI



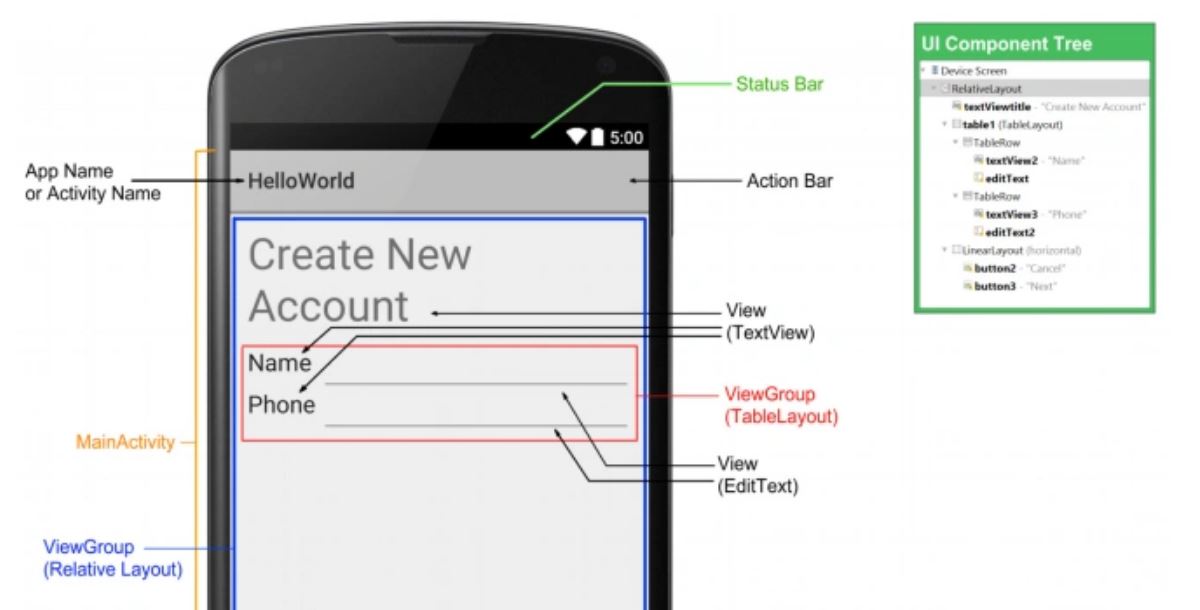
* Designer UI



* Project Structure



* Application Interface



* 1. Communications Interfaces

Application Communication components are the essential block of android application. These components are coupled in the manifest file that describes each component of the application and how they interact. There are four main communication components of android studio explain below:

1. Activities: Activities dictate the UI and handle the user interaction to the smart phone screen.
2. Services : it handles background processing associated with an application
3. Broadcast Receivers: it handle communication between Android OS and applications.
4. Content Provider: It handle data and database management issues.
5. Functional Requirements
   1. Functional Hierarchy

## User

# User can register his account.

# User can login in his account.

# User can view available cars.

# User can book a car and pay the bill.

# User can view his car booking history.

# User can sign-out

## 

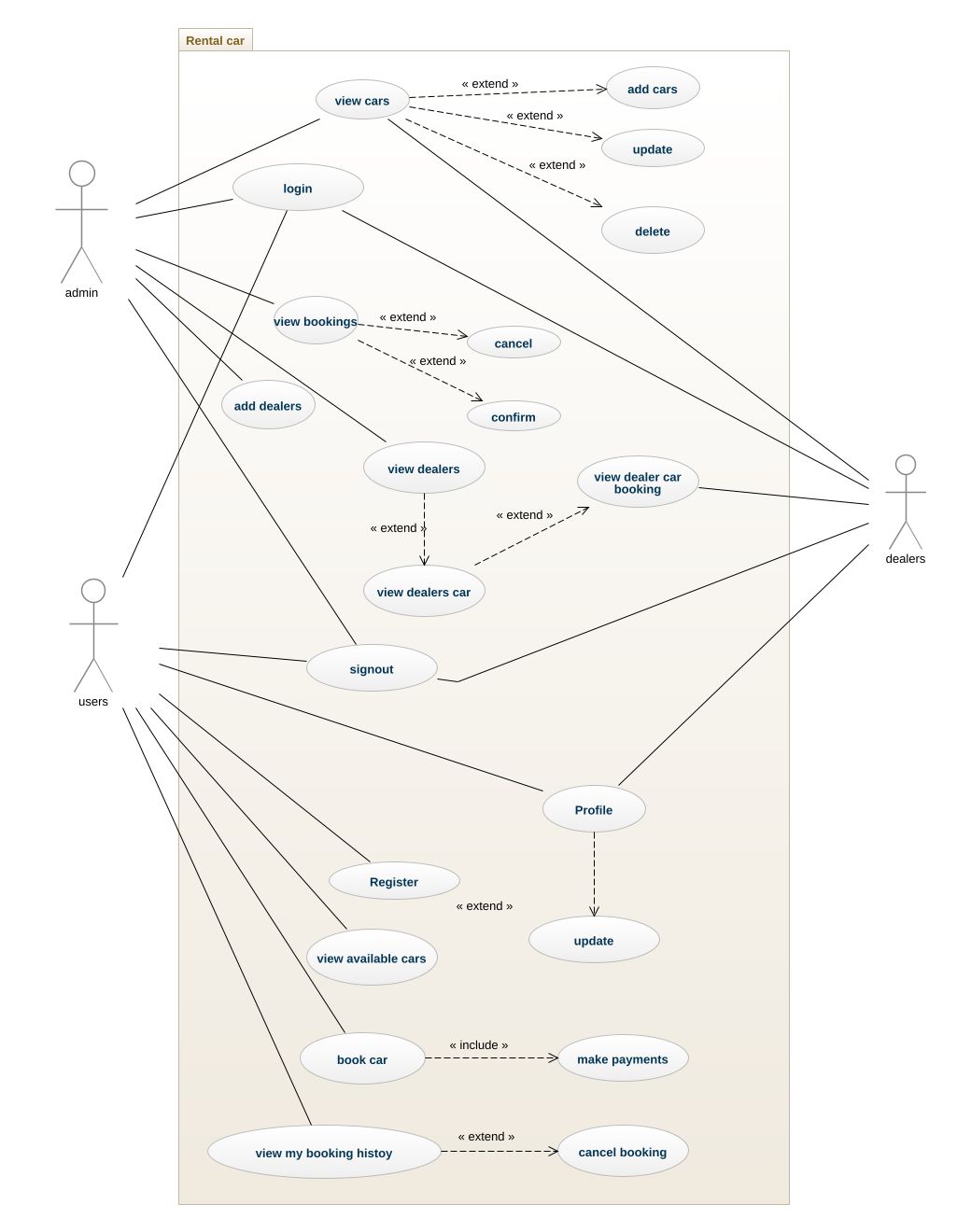
## Admin

1. Admin can login
2. Admin can add dealers/ delete/view /edit dealers.
3. Admin can add /view / delete /edit cars.
4. Admin can handle bookings.
5. Admin can handle dealers.

Dealer

1. Dealer can login.
2. Dealer can add/view/delete and edit car.
3. Dealer can view bookings.

* 1. Use Cases



* + 1. Register use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-01: Register** | | | | |
| **Use case Id:** | | UC-01 | | |
| **Actors:**  User | | | | |
| **Feature:** Actor wants to create account in the app | | | | |
| **Pre-condition:** | | Android Phone  Internal Connection  Application installed properly | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User open the application and select register option | | | **2.** System open the registration screen. |
| **3.** | User fill the registration form details | | | **4.** system check the form data and show error if any form data is invalid |
| **5.** | User submit the form | | | 6. System save the user detail in the database and create user account. |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **5a:** User entered the wrong data.  **6a:** The error message will be displayed.  **7a:** Application will ask user to re-enter data on error fields. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The user successfully register his account in the application | | | |
| **Use Case Cross referenced** | | | Login | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-02: Login** | | | | |
| **Use case Id:** | | UC-02 | | |
| **Actors:**  User / dealer / admin | | | | |
| **Feature:** Actor wants to sign in his account to use the application | | | | |
| **Pre-condition:** | | The actor should register his account in the application. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User open the application | | | **2.** System provides options to select between sign in and sign up |
| **3.** | User tap on Login button | | | **4.** System open the login form |
| **5.** | User fill the login form by entering User email and Password | | | **6.** System will verify and validate the given user email and password. |
| **7.** | User gets access to the application. | | |  |
| **Alternate Scenarios:** | | | | |
| **5a:** User entered the wrong ID or password.  **6a:** The error message will be displayed.  **7a:** Application will ask user to re-enter ID and password. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully login in his account | | | |
| **Use Case Cross referenced** | | | Add car , book a car , view booking, view cars | |
|  | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-03: view available cars** | | | | |
| **Use case Id:** | | UC-03 | | |
| **Actors:**  User | | | | |
| **Feature:** actor want to view the available cars of the app | | | | |
| **Pre-condition:** | | The actor should login in his account. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User open the user dashboard screen | | | **2.** System fetch all the available car data from the database and display them in the screen. |
| **3.** | User view the list of cars and select any car | | | **4.** System open the selected car full detail in the screen |
| **5.** | User view the single car complete detail | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully view the available car | | | |
| **Use Case Cross referenced** | | | Book car | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-04: book a car** | | | | |
| **Use case Id:** | | UC-04 | | |
| **Actors:**  User | | | | |
| **Feature:** actor want to book his selected car | | | | |
| **Pre-condition:** | | The actor should login in his account and select the car . | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User open the user dashboard screen | | | **2.** System fetch all the available car data from the database and display them in the screen. |
| **3.** | User view the list of cars and select any car | | | **4.** System open the selected car full detail in the screen |
| **5.** | User view the single car complete detail | | |  |
| **6.** | User select the book a car button | | | 7. System opens a book a car form screen. |
| **8.** | User fill the booking car form details and click submit button | | | 9.system check user enter data if any fill data is invalid it display message , else it will save booking detail in the database. |
| **Alternate Scenarios:** | | | | |
| **8a. user enter wrong booking detail**  **8a. system show notification message of invalid message** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully login and select one of any available car | | | |
| **Use Case Cross referenced** | | | Booking payment | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-05: booking payment** | | | | |
| **Use case Id:** | | UC-05 | | |
| **Actors:**  User | | | | |
| **Feature:** actor want to book his selected car | | | | |
| **Pre-condition:** | | The actor should book the selected car | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User open the user dashboard screen | | | **2.** System fetch all the available car data from the database and display them in the screen. |
| **3.** | User view the list of cars and select any car | | | **4.** System open the selected car full detail in the screen |
| **5.** | User view the single car complete detail | | |  |
| **6.** | User select the book a car button | | | 7. System opens a book a car form screen. |
| **8.** | User fill the booking car form details and click submit button | | | 9. System check user enter data if any fill data is invalid it display message , else it will save booking detail in the database. |
|  |  | | | 10. system shows the online payment form |
| **11.** | User fill the payment form and click submit | | | 12. system save the payment transaction detail in the database. |
| **Alternate Scenarios:** | | | | |
| **8a. enter wrong booking detail**  **8a. system show notification message of invalid message**  **11a . user enter any invalid payment detail** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully pays the online bill of the car booking. | | | |
| **Use Case Cross referenced** | | | View my booking history | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-06: view my booking history** | | | | |
| **Use case Id:** | | UC-06 | | |
| **Actors:**  User | | | | |
| **Feature:** actor want to view his car booking history | | | | |
| **Pre-condition:** | | The actor should have place booking of the car. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User select view my booking history option | | | **2.** System fetches all the login user booking data from the database and display them in the screen. |
| **3.** | User view his car booking history | | |  |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. internet not connected.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully view his car booking history | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-07: profile** | | | | |
| **Use case Id:** | | UC-07 | | |
| **Actors:**  User | | | | |
| **Feature:** actor want to view or update his profile information | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User select view my profile option | | | **2.** System fetches login user profile data from the database and displays them in the screen. |
| **3.** | User view the profile data and edited any information and click updated button | | | 4. system save the new edited information of the user profile in the database, |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. internet not connected.**  **3a. user enter invalid detail** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully view and edited his profile | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-08: sign out** | | | | |
| **Use case Id:** | | UC-08 | | |
| **Actors:**  User, dealer, admin | | | | |
| **Feature:** actor wants to sign-out from his account | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User select sign-out option | | | **2.** System sign out the current login user from the application. |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. internet not connected.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully signs out from his profile. | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-09: Add dealers** | | | | |
| **Use case Id:** | | UC-09 | | |
| **Actors:**  admin | | | | |
| **Feature:** actor wants to add dealer in the application | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User select add dealers option | | | **2.** System open the add dealer form |
| **3.** | Actor fill the add dealer form and click add | | | 4. System save the dealer detail in the database and notify it. |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. actor enter invalid data** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully add the dealer | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-10: views** | | | | |
| **Use case Id:** | | UC-09 | | |
| **Actors:**  admin , dealer | | | | |
| **Feature:** actor wants view his added car or delete or edit them | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User select add car option | | | **2.** System open the add car form |
| **3.** | Actor fill the add car form and click add | | | 4. System save the car detail in the database and notify it and open a view car screen. |
| **5.** | Actor view the car detail list screen and if select any car delete button | | | 6. The system deletes the selected car detail from the database and opens a view car screen. |
| **7.** | Actor view the car detail list screen and if select any car edit button | | | 8. System opens a edit form and fetch data of selected car data from database and show in the form. |
| **9.** | Actor enter new data in the form and click update | | | 10. system update the selected car data in the database. |
| **Alternate Scenarios:** | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully add, delete, view or edit the car | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-11: view booking** | | | | |
| **Use case Id:** | | UC-11 | | |
| **Actors:**  admin | | | | |
| **Feature:** actor want to view all the booking of the cars | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | admin select view booking option | | | **2.** System fetches booking data from the database and displays them in the screen. |
| **3.** | Admin view the booking details in the form of list | | |  |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. internet not connected.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully view the bookings | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UC-12: view dealer booking** | | | | |
| **Use case Id:** | | UC-12 | | |
| **Actors:**  dealer | | | | |
| **Feature:** actor want to view all the booking of his cars | | | | |
| **Pre-condition:** | | The actor should login in his account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | dealer select view booking option | | | **2.** System fetches booking data of his car from the database and displays them in the screen. |
| **3.** | Dealer view the booking details in the form of list | | |  |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a. internet not connected.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | The actor successfully view his car bookings | | | |
| **Use Case Cross referenced** | | |  | |

1. Non-functional Requirements
   1. Performance Requirements

Rental car system designed and developed in such a way that it did not utilize too many resources. It performs the given task in minimum time possible and it uses minimum storage capacity.

* 1. Security Requirements

Rental car system designed in such a way that it provides security of data and only authorized user can use the application. It handles all the security and data loss threats. Registration and login system is safe and secure.

* 1. User Documentation

We are going to distribute the project documentation to the supervisor in the form of soft copy.

1. References

1. System, Online. 'Online Car Rental System'. Academia.edu. N.p., 2015. Web. 9 June 2015.

2. Scribd.com, Online. '49930505 Car Rental System Project Report'. N.p., 2015. Web. 9 June 2015.

3. Scribd.com, Online. 'Car Rental System Documentation'. N.p., 2015. Web. 9 June 2015.

4. Freelancer, Online. 'Project Documentation Car Rental Company Software Development Freelancers and Jobs - Freelancer'. N.p., 2015. Web. 9 June 2015.

5. Slideshare.net, Online. 'Zook Car Rental System Project'. N.p., 2015. Web. 9 June 2015.

Appendices

N/A