Rentsly Car Rental Mobile Application

Final Year Project

Session 2017-2021

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior College, Lahore

Spring 2021

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of Specialization	Android Development
FYP ID	FYP-BCSM-S21-002

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F17-026	Bazan Tahir	bcsm-f17-026@superior.edu.pk	
(ii)	BCSM-F17-056	M Bassam Hashmi	bcsm-f17-056@superior.edu.pk	
(iii)	BCSM-F17-269	Muhammad Asim	bcsm-f17-269@superior.edu.pk	

The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Bazan Tahir S/D of Ashfaq Ahmad Tahir, group leader of FYP under registration no BSCM-F17-026 at Computer Science Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 2-17-2022 Name of Group Leader: Bazan Tahir Signature: _____

Name of Supervisor: Ms. Sabah Arif Designation: Senior Lecturer

Signature: _____

HOD: Dr. Irfan-u-Din Signature: _____

Rentsly Car Rental

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback from Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

This work is dedicated to all those people in my life who supported me and prayed for me in my tough times. To all those people who wanted me to succeed in my life. Especially my parents who made me this much capable and helped me in my hardships. All teachers who Helped us along the way epically Ms. Sabah Arif our supervisor, his dedication towards us is remarkable.

Acknowledgements

Throughout the writing of this documentation, we have received a great deal of support and assistance. We would first like to thank our supervisor, Ms. Sabah Arif, whose expertise was invaluable in completing the documentation and development of our Final Year Project. His insightful feedback pushed us to sharpen our thinking and brought our work to a higher level. We would also like to thank our teachers for their valuable guidance throughout our studies. They've provided us with the tools that we needed to choose the right direction and successfully complete our Final Year Project. In addition, we would like to thank our parents for their wise counsel and sympathetic ear. Finally, we could not have completed this Final Year Project without the support of our friends, who provided stimulating discussions as well as happy distractions to rest our mind outside of our work.

Executive Summary

Rentsly Mobile Application will automate the car rental process and provide rental of economy, standard and luxury automobiles in its targeted market. The system will provide comfort to end users to book a car or just to get estimated cost by using mobile application, and will provide opportunity to car owner to earn by renting out their car. Each type of car should have a different rental fee per day. Rental fee depends on number of days, brand and where the car is to be taken. The system should have the following functionalities:

Rent: The system equipped to answer Customer's inquiries about the availability and rental fee of various types of cars for certain dates in the future. When the customer makes a decision about the Type of car and the Dates, the system should be able to start chat with Agent and Reserve the requested type of car for requested dates. The customer should be given a Confirmation Number.

Pick Up: The system processes a Car Pick Up and Agent will be notified. Customer walks to the Agent and supplies either the confirmation number, or name. The system should pull up all the reservation information about this customer.

Return: The system processes a return. The system should record the date, time and processed by Depending on these parameters, the system calculates the final rental amount.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

Rentsly is a company which rent cars for short period of time for a fee to their customer. Rentsly Mobile Application will provide a platform where people can find cars to rent directly from car owners at affordable prices and car owners can make an earning by just giving away their keys. The car owner can be an individual or an agency offering car. The system being developed is a system to handle the business needs of renting out vehicles to customers, maintaining records and data on vehicle fleet, operating the customer portal website, and reporting the state of the system to the company.

1.1. Background

Transport facility is a matter of headache for those people who do not have any personal Transport. On occasions like Wedding, Vacation, and tour outside the city and on many other situations they feel the necessity of a personal vehicle (Unlike Taxi; Uber or Careem) to sort out the problems. So, if it is possible to design or develop an easy-to-use mobile application for availing transport whenever and wherever possible, then it will be beneficial for both renter and transport provider. Now a day, by some clicks only, we can get whatever you want at home. We already know about the online shopping, e-banking etc. Similarly, Rentsly Application is the facility to book cars online from your mobile device. Some people cannot afford to have a car, for those people this system becomes very helpful. This system includes various cars, as per the customer order and comfort, it place the order and pick up the car as per the location within the area.

1.2. Motivations and Challenges

With this software house in need of a better system for general public idea by a company, we are assigned. To develop such a system that would not only ease the burden on the company's
customers, but for the general public. Our team has an immense amount of knowledge when it
comes to problem solving, programming, and communication. Not only would we strive to give
the Rentsly service everything they desired, but we will continue to make sure the application is at its very best and beyond. Each one of us will always and will continue to give 100% and more to making the transition a breeze for the car rental service.

1.3. Goals and Objectives

Our goal is to create an application that will:

- Simplify the extraneous process of Car Renting.
- To produce a mobile application system that allow Client to register and reserve car online and for the Agent to effectively manage their car rental business.
- To ease customer's task whenever they need to rent a car.
- This application will offer a bridge between Client (Car renter) and Agent (Car owner).

1.4. Literature Review/Existing Solutions

In Pakistan, taxi service is automated to mobile applications like Uber and Careem that allows consumers to travel with a driver. Meanwhile, Car rental companies are facing downtime as people don't prefer to go to their offices by themselves to even ask for the estimation cost. However, some car rental agencies still use a manual system to manage rental car operations by spreading of their available car to local resident. This method wasting money and time for both rental person and car rental owner. Therefore, it is proposed to have a system that can be used to provide booking and management to make easier for both of them. Also, there are some people who have a car, they don't drive it more often but don't have enough time or some other reason to enroll, drive as a taxi and earn.

1.5. Gap Analysis

Renting a car in Pakistan right now a tiresome and expensive process. There are bunch of rental cars that owned by different owner. Some of the car that the owner provided are different from other and each of the car have their own advantages and disadvantages. There is various type of car will give burden for user to choose which car is the best for them. Besides that, manual system does not allow user to booking online and hard to keep track on the record of rental cars. Instead, the car owner only spread the word about their available car to a local resident only. User must contact a car rental owner and contract out for a vehicle and this will delay the process of renting car. This method consume time for both car rental and car rental owner.

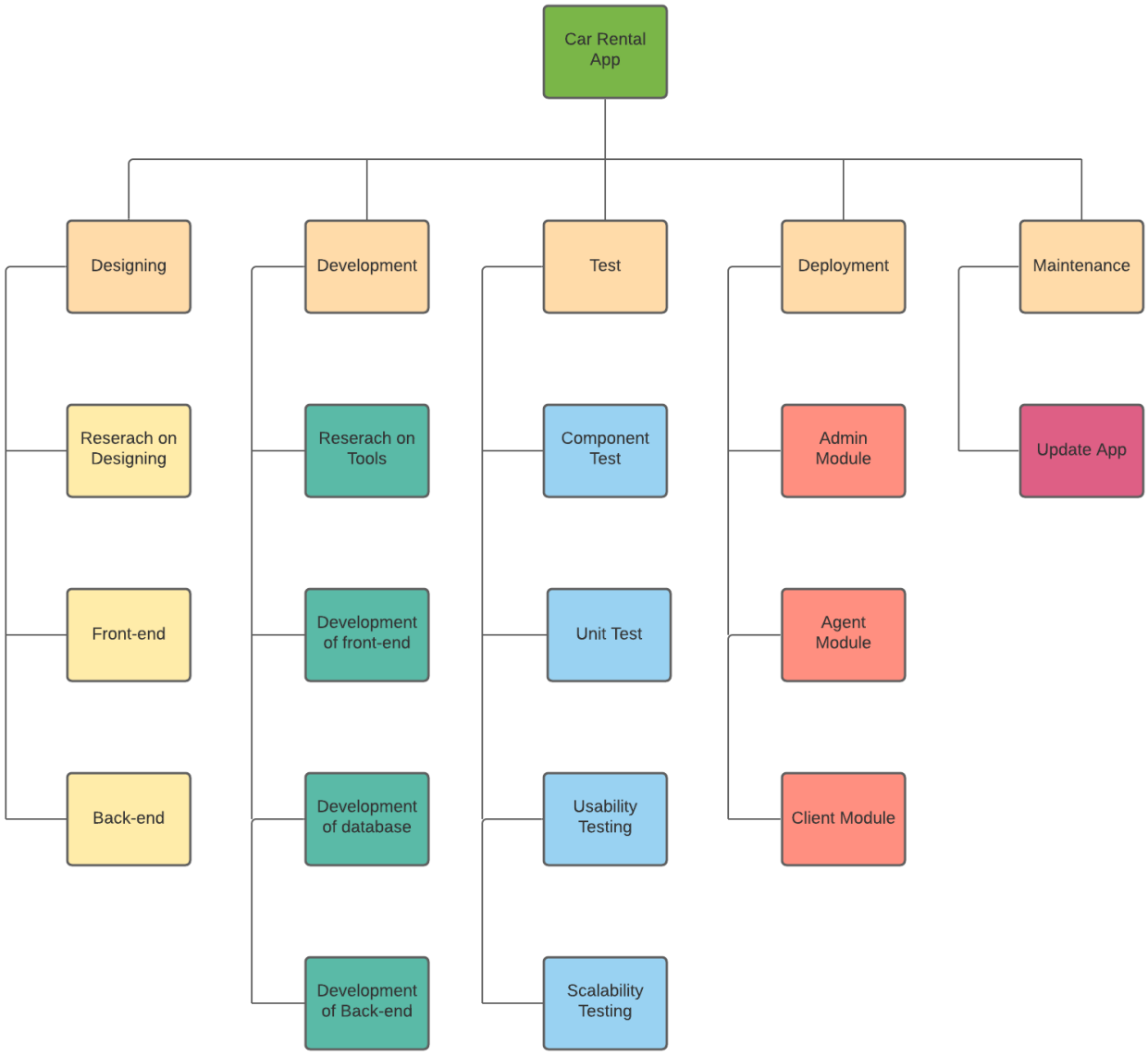
1.6. Proposed Solution

An interaction application having have a user interface that will allow their customers to view the models, descriptions and prices of different cars available. The user has the ability to register and log in and track their rental plan. App will be responsive, allowing for the customer to view it on android mobile device, from tablets to mobile phones. The administrator will also be able to login through the same form but have the ability to add/remove user, edit car details and approve/disapprove KYC (Know Your Customer) of user. The system works on P2P (Peer-to-Peer) business model as there is no third-party involved. The Client can directly connect with the Agent that is lending their car on rent. The user can also check estimated cost per day by selecting the type of car, days and geographical area (with-in or out-of-city) where the car is to be taken. The payment gateway will include Cash and Credit/Debit Card payment.

1.7. Project Plan

The following section contains information regarding Work Break down Structure and Gantt chart for the activities we'll be performing through our Final Year Project. We've managed to break down tasks for each individual member in our team. All activities are assigned by the team leader with respect to each team member's abilities.

1.7.1. Work Breakdown Structure

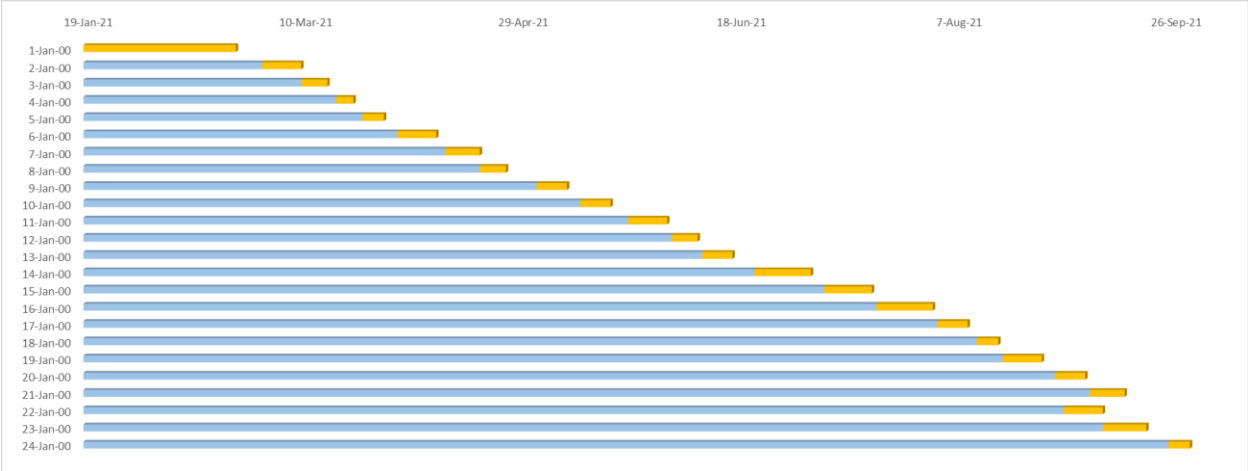


1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
01	Idea Presentation	1.1	-		All Team Members
02	Idea Selection	1.2	1.1		All Team Members
03	Requirement Gathering	1.3	1.2		All Team Members
04	Requirement Analysis	1.4	1.3		All Team Members
05	Requirement Elicitation	1.5	1.4		All Team Members
06	Documentation	1.6	1.5		All Team Members
07	Software Selection	1.7	1.6		M Bassam Hashmi
08	Functional Requirements	1.8	1.7		M Asim
09	Non-Functional Requirements	1.9	1.8		Bazan Tahir
10	User Interface Development	1.10	1.9		All Team Members
11	Backend Coding	1.11	1.10		All Team Members
12	Prototype	1.12	1.11		M Asim
13	Server Testing	1.13	1.12		Bazan Tahir
14	Prototype Testing	1.14	1.13		M Bassam Hashmi
15	Deployment	1.15	1.14		All Team Members

1.7.3. Gantt chart



1.8. Report Outline

In this chapter we've discussed about various aspects of our project like its Introduction, Background, Motivation, Goals and Objectives. From comparing our system with existing solutions to providing gap analysis to have a better understanding of our system. What is our proposed solution and our project plan as well as roles and responsibility matrix, everything is well explained? The upcoming chapters will contain information like its Software requirements from Functional and non-Functional aspects. Who is our intended audience and what the scope of our project is along with diagrams like use-cases and fully dressed use-cases?

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

The purpose of this software requirement specification document is to provide a detailed description of the functional and working of the software application. This document will cover each and every feature of the application including connecting agents and clients, client advertisements and other features. This document will cover all intended feature and technical dependencies.

2.1.2. Document Conventions

Standard rule of documentation has been followed in order of standardize work. This software requirement document use following conventions.

Description		**Appearance**
Application		Rentsly Mobile Application
Database		Firebase
Agents		Car owners
Clients		Car Renter/ Consumer

Figure 2.1.2. Document Conventions

2.1.3. Intended Audience and Reading Suggestions

This document is intended for developer, project manager and users of this application who wish to read about what the project can do. Rest of SRS contains work flow of the system how system works, flow of data in firebase and other technical functionalities of the software.

- For developer there is need to start from introduction and understand a work flow diagrams.
- For project manager sequence of reading is from introduction to class diagram and use cases.
- For users of applications there is need to read from introduction and read all description including scope and functionalities

2.1.4. Product Scope

The scopes for this project are identified to make the system development process easier.

I. Admin
   a) Manage and monitor the application
   b) Admin can view report

II. Owner
    a) Register and login profile
    b) Register, update, and delete car details
    c) Approve booking status
III. User
    a) Register and login profile
    b) Search for car base on criteria chosen
    c) Book car

This business model works on P2P (Peer-to-Peer) model as there is no third-party involved. The consumer directly connects with the car owner. Both registered and non-registered users will be able to search car rentals by price, model, seating and any other potential searches. They will also be able to select and pay for the service. Only KYC approved consumer can book the car.

2.1.5. References

Car Rental Application: Development Cost and Timeline

Android Studio Developers Guide

Android payment app developers guide

Firebase

Lottie Files - Free animations files

2.2. Overall Description

2.2.1. Product Perspective

This system is designed and developed to facilitate people who have to go through a hectic process for booking a car on rent or lending a car for rent. Therefore, it is proposed to have a system that can be used to provide booking and management to make easier for both of them. Also, there are some people that have a car, they don't drive it more often but don't have enough time or some other reason to enroll drive as a taxi and earn. This will allow the agents to earn money by lending their car on rent and client can search, view the models, descriptions and prices of different cars available from the application and book anytime.

Client:

- Sign up
- Log in
- Create Profile
- Update Profile
- Account Verification / KYC (Know Your Customer)
- Enter car name and share location and find the nearest available car
- Check Car Details
- Contact agents
- Select Renting packages
- Check the Estimated Cost

Agent:

- Sign up
- Log in
- Add/Update Car Details
- Update Profile
- Contact Clients
- Define Rent Package

Administration:

- Sign up
- Log in
- Manage Users
- Verify Users
- Approve Rent Package Defined by Agents

2.2.2. User Classes and Characteristics

User Classes	Characteristics
Client	Client class will contain information regarding the customer looking to book a car on rent. Client will have features like finding available Cars for rent.
Agent	Agent class will contain information regarding the person that are owner of the car and are lending their cars for rent.
Administration	Administration class will contain information regarding both Agent and Clients. Administration will manage both Agent and Clients.

Figure 2.2.2. User Class and Characteristics

2.2.3. Operating Environment

The Operating Environment will be as following:

- **Operating System:** Android
- **Database:** Firebase
- **Integrated Development Environment:** Android Studio / Visual Studio Code

2.2.4. Design and Implementation Constraints

The limitation and issues for developers is as following:

- Client and Agent have to decide the place where they will take over the car before and after rent.
- Firebase Firestone only offers a certain amount of data to be stored in their document after that the data will automatically get replaced.
- Administrator have to deal with all the customer's KYC.

2.2.5. Assumptions and Dependencies

There is multiple type of users so:

- At the time of sign up each user will be stored as a document in their respective category they've chosen to sign up in.
- Android studio will be used to initially develop this application so cross-platform support will have to wait a little longer.
- Agents will be restricted from contacting Clients first. Only Client has the authority to decide which Agent to contact to.

2.3. External Interface Requirements

2.3.1. User Interfaces

Mobile User Interface will be developed graphical and touch-sensitive display are available worldwide on any modern mobile device. The UI will allow the user to interact with the application's features, content and functionalities. At the first screen after logging in the users will be able to see the dashboard and they will be guided with the current information of application. Users can also navigate through various pages using bottom navigation view or menu bar on the left side. All the User Interface is design to ensure usability, readability and consistency for every type of user.

2.3.2. Hardware Interfaces

The supported devices for our application will be as following:

- Smart Phones and Tablets running on Android Operating Systems
- Android Operating System must be higher than Android 7.0 Nougat

- Smart Phone should have a minimum space of about 500 MB free in order to install this application.
- Smart Phone should have a minimum RAM of about 2 GB to ensure a smooth and proper working.
- Smart Phone should must have a higher Adreno level then 450.

2.3.3. Software Interfaces

Android Studio and Visual Code are platforms used to design and develop professional native applications for Android Operating System. This application uses Java as its primary backend language, XML for its User Interface and Firebase for its database. This application uses Android Software Development kit (SDK). So, any smart phone running Android operating system can run this application.

2.3.4. Communications Interfaces

Rentsly Mobile Application is based on Software Development Kit framework which runs on mobile devices or tablets and uses Firebase as its primary database. This application will use Clients and Agents to connect to Server. The exchanging of data between the server and the users will take place through Firebase connection. For encrypted transaction security will be used for encrypting all data due to increasing respect for privacy on mobile devices and used authorized APIs only. All passwords will be stored in an encrypted form.

2.4. System Features

This application will help Clients and Agents to directly connect with one another and rent / lend the car without any hurdles. In this application both Agents and Clients can sign up, after that Agents can create their profile and add Car details that they want to lend for rent. Clients can do Verification/KYC (Know Your Customer) by submitting the required documents and then search the car depending upon their need and they'll be listed with nearest available cars to their shared location. A communication channel will be created in order for them to communicate. After selecting the renting package and finalizing the deal, the pickup location of car will be shared to Client by Agent. Client can select payment method and make payment in advance. At point of returning the car back to the Agent, Agent will be asked of any update/extension of renting package that was selected. Client can confirm, make payment of that and provide feedback.

2.4.1. Login

2.4.1.1. Description and Priority

In this application the stakeholders such as the Agents and Clients will their own login.

This feature will verify that a stakeholder exists in our database or not. The identity of each stakeholder will be verified before logging in. This feature has a high priority and feature risk scale is high. This feature is not concerned with the level of stakeholder, it must verity if stakeholder has a valid email and password.

2.4.1.2. Stimulus/Response Sequences

First the stakeholder will enter the email and password which was used at the time of registering. The system will validate and verify the email and password from Firebase Login Authentication. If the validation and verification is complete, the stakeholder will move to next activity. If validation and verification fail the system will give an error message and access to the system will be denied.

2.4.1.3. Functional Requirements

REQ-SF1-1: This feature will have two input types: Email and Password.

REQ-SF1-2: This feature will have one button for login action.

REQ-SF1-3: This feature will firstly check if the user has entered Email and Password or has left it black.

REQ-SF1-4: Then it will check if the validation of Email and Password.

REQ-SF1-5: If the Email and Password are not validated it will send an error toast to the user.

REQ-SF1-6: If the Email and Password is verified the user will be displayed a success toast and will move to its Dashboard.

2.4.2. Agent Registration

2.4.2.1. Description and Priority

In this feature the Agent will create an account. This feature will firstly check if the user has created a previous login or not. Afterwards it will verify if the Email and Password given are acceptable or not. This feature has high priority. The feature risk in terms of scale is high. This feature should not mix the stakeholder level and must verify if stakeholder email and password is valid.

2.4.2.2. Stimulus/Response Sequences

First the stakeholder will enter Email and Password into the input text boxes. The system will validate and verify the Email and Password from Firebase Authentication that email already exists or not. If the validation and

verification is complete, the stakeholder will move to next activity. If not, the system will give him an alert message.

2.4.2.3. Functional Requirements

REQ-SF2-1: This feature will have three input type Email, Password and confirm
Password.

REQ-SF2-2: This feature will have one button for registration action.

REQ-SF2-3: This feature will firstly check if the Password and Confirm Password field matches or not.

REQ-SF2-4: If the Password and Confirm Password fields do not match, it will give an error toast to the stakeholder.

REQ-SF2-5: After this another check is put in-place if the stakeholder has entered minimum 6 characters in his/her Password field.

REQ-SF2-6: Then the feature will check the validation of Email and Password.

REQ-SF2-7: If the Email and Password is not validated will give an error.

REQ-SF2-8: This feature will check if the Email already exists in database or not.

REQ-SF2-9: If the Email already exists, it will give an error.

2.4.3. Client Registration

2.4.3.1. Description and Priority

In this feature the Client's will create an account. This feature will firstly check if the user has created a previous login or not. Afterwards it will verify if the Email and Password given are acceptable or not. This feature has high priority. The feature risk in terms of scale is high. This feature should not mix the stakeholder level and must verify if stakeholder email and password is valid.

2.4.3.2. Stimulus/Response Sequences

First the stakeholder will enter Email and Password into the input text boxes. The system will validate and verify the Email and Password from Firebase Authentication that email already exists or not. If the validation and

verification is complete, the stakeholder will move to next activity. If not, the system will give him an alert message.

2.4.3.3. Functional Requirements

REQ-SF3-1: This feature will have three input type Email, Password and confirm Password.

REQ-SF3-2: This feature will have one button for registration action.

REQ-SF3-3: This feature will firstly check if the Password and Confirm Password field matches or not.

REQ-SF3-4: If the Password and Confirm Password fields do not match, it will give an error toast to the stakeholder.

REQ-SF3-5: After this another check is put in-place if the stakeholder has entered minimum 6 characters in his/her Password field.

REQ-SF3-6: Then the feature will check the validation of Email and Password.

REQ-SF3-7: If the Email and Password is not validated will give an error.

REQ-SF3-8: This feature will check if the Email already exists in database or not.

REQ-SF3-9: If the Email already exists, it will give an error.

2.4.4. Create Agent Profile

2.4.4.1. Description and Priority

In this feature the Car Owner aka Agent will create their profiles. In this feature will fill various input types such as Agent's name, Picture, address, phone number, location. This feature has high priority. In terms of feature risk scale, it is medium. This feature will help the Client to view Agent's profile, car details and picture and book the car.

2.4.4.2. Stimulus/Response Sequences

After completing Agent registration, the Agent will move to Dashboard activity. Update Profile option will be in the Dashboard settings. In this feature stakeholder will fill various input types such as name, address, phone number, location and profile picture. This will check all inputs are valid. If all inputs are valid, the system will save all data in database.

2.4.4.3. Functional Requirements

REQ-SF4-1: This feature will check if any of the input is empty.

REQ-SF4-2: If any of the input is empty an error toast will be displayed.

REQ-SF4-3: This feature will have one button as an action for updating profile.

REQ-SF4-4: This feature will check all information is valid or not.

REQ-SF4-5: If the information is not valid it gives an error toast.

2.4.5. Create Client Profile

2.4.5.1. Description and Priority

In this feature the Car Renter aka Client will create their profiles. In this feature will fill various input types such as Name, Picture, address, phone number, location. This feature has high priority. In terms of feature risk scale, it is medium. This feature will help the Agent to view Client's profile and authenticate before lending the car.

2.4.5.2. Stimulus/Response Sequences

After completing Client registration, the Client will move to Dashboard activity. Update Profile option will be in the Dashboard settings. In this feature stakeholder will fill various input types such as name, address, phone number, location and profile picture. This will check all inputs are valid. If all inputs are valid, the system will save all data in database.

2.4.5.3. Functional Requirements

REQ-SF5-1: This feature will check if any of the input is empty.

REQ-SF5-2: If any of the input is empty an error toast will be displayed.

REQ-SF5-3: This feature will have one button as an action for updating profile.

REQ-SF5-4: This feature will check all information is valid or not.

REQ-SF5-5: If the information is not valid it gives an error toast.

2.4.6. Client Verification KYC (Know Your Customer)

2.4.6.1. Description and Priority

In this feature Clients are asked to provide required documents for account verification. The Administrator will validate and allocate Verified badge to Client's profile. This feature has high priority. In terms of feature risk scale, it is high.

2.4.6.2. Stimulus/Response Sequences

Clients will provide National ID Card / Passport and Driving License for account verification. The Administrator will validate and allocate Verified badge to Client's profile.

2.4.6.3. Functional Requirements

REQ-SF6-1: This feature will get required documents from Client for verification.

REQ-SF6-3: Check if Client has attached National ID Card and Driving License.

REQ-SF6-4: If any of the input is empty an error toast will be displayed.

REQ-SF6-5: This feature will have one button as an action for Submit Documents.

REQ-SF6-5: This feature will let Administrator validate all information provided and whether it is valid or not.

REQ-SF6-7: If the information is not valid it gives an error toast.

REQ-SF6-8: In case of success, verified badge will be given to Client.

2.4.7. Add Car Details and Rent Package

2.4.7.1. Description and Priority

In this feature Agents will enter the Car details, Rent Package and submit to Admin for approving the Car for renting. This feature has high priority. In terms of feature risk scale, it is high.

2.4.7.2. Stimulus/Response Sequences

Agents will enter Car details; Brand name, Model name, Engine Power, Registration number, Pictures, Rent per day. At time of submission the Agent will be asked to provide any document that proves the ownership. After acceptance from the Administrator the Car will be listed for renting.

2.4.7.3. Functional Requirements

REQ-SF7-1: This feature will get Car details like Model name, Engine Power, Registration number and Pictures.

REQ-SF7-2: If any of the input is empty an error toast will be displayed.

REQ-SF7-3: On the next screen the Agent will enter Rent rate of a day.

REQ-SF7-4: If any of the input is empty an error toast will be displayed.

REQ-SF7-5: Next the Agent will attach any document any document that proves Agent has the ownership right of the car.

REQ-SF7-6: This feature will have one button as an action for Submit for Listing.

REQ-SF7-7: This feature will let Administrator validate all information provided and whether it is valid or not.

REQ-SF7-8: If the information is not valid it gives an error toast.

REQ-SF7-9: After validation by Administrator the car will be available for Rent.

2.4.8. View All available Cars on Rent

2.4.8.1. Description and Priority

In this feature the application will display all available cars that are available for renting. The order of the list is prioritized by nearest location to the shared physical location of client. This feature has high priority. In terms of feature risk scale, it is high.

2.4.8.2. Stimulus/Response Sequences

Clients can directly go to available cars menu where all of the available car's list will be retrieved from the database.

2.4.8.3. Functional Requirements

REQ-SF8-1: This feature will get all available cars data from the database.

REQ-SF8-2: If there are not any available cars then it will give a toast.

REQ-SF8-3: The feature will sort the available car as nearest to the Client's location

REQ-SF8-4: When Client goes to available cars menu then list of all cars will be displayed.

REQ-SF8-5: If there are not any available cars then it will give a toast.

2.4.9. Search for Desired Car by Model

2.4.9.1. Description and Priority

In this feature Clients will search the Car by entering the name of car in the search bar at the top. The feature will display nearest filtered available car to the shared physical location Client. This feature has high priority. In terms of feature risk scale, it is high.

2.4.9.2. Stimulus/Response Sequences

Clients will enter the name of the car in the search box at the top and tap search. Then the nearest available cars of that model will be displayed in order.

2.4.9.3. Functional Requirements

REQ-SF9-1: Client enters the name/model of car in search.

REQ-SF9-2: This feature will look into the database for the search query.

REQ-SF9-3: If there are not any available cars of that name/model then it will give a error toast.

REQ-SF9-4: If available then they will be listed in nearest first order.

2.4.10. Select the Desired Rental Car

2.4.10.1. Description and Priority

In this feature after searching for the available cars as per required, Clients will be able to select any car and view car details, rent package and contact Agent. After selecting the package and contacting Agent, Client can book the car on rent and pickup location will be shared. This feature has high priority. In terms of feature risk scale, it is high.

2.4.10.2. Stimulus/Response Sequences

In the available cars list, Client will check the quick preview of cars; Car model and rate/day, and then click on desired car to view car specification and rent per day in detail.

2.4.10.3. Functional Requirements

REQ-SF10-1: This feature allows Client to tap on any of the available cars in the list.

REQ-SF10-2: It displays a separate section where Car Name, Model, Pictures, rent per day and other specifications will be listed.

REQ-SF10-3: Client will enter the number of days he/she wants to book the car for and click Book Now.

REQ-SF10-4: Client will then select payment option.

REQ-SF10-5: This feature will send offer to the Agent.

REQ-SF10-6: If the Agent accepts the offer, communication channel will be developed between them.

REQ-SF10-7: They can finalize the deal and Client can make payment.

REQ-SF10-8: Agent will share pickup location after the payment.

2.4.11. Start Chat

2.4.11.1. Description and Priority

In this feature Client will select the available Car and Rent Package and start a communication channel with Agent. This feature has high priority. In terms of feature risk scale, it is high.

2.4.11.2. Stimulus/Response Sequences

After the Client has selected the Car, a communication channel will be created between them. Although the Agent will have the option to accept or decline the renting. If accepted the Client can then pay for the renting.

2.4.11.3. Functional Requirements

REQ-SF11-1: This feature will check if the Agent has accepted Client's offer.

REQ-SF11-2: If the Agent has accepted then a communication channel will be created.

REQ-SF11-3: The system will give a toast to Client that communication channel has been created.

REQ-SF11-4: The Client will be redirected towards the chat activity interface.

2.4.12. Give Feedback

2.4.12.1. Description and Priority

In this feature both Agents and Clients can provide feedback to each other about the renting service/experience. This feature has high priority. In terms of feature risk scale, it is high.

2.4.12.2. Stimulus/Response Sequences

Agent and Client can rate each other after the closing. A text box will be available to enter comments.

2.4.12.3. Functional Requirements

REQ-SF12-1: This feature will check if the Agent or Client has rated out of 5 stars

REQ-SF12-2: Text box will appear and where comments can be entered.

REQ-SF12-3: The system will give a toast to both that feedback successfully published.

2.4.13. Edit and Delete Car Details

2.4.13.1. Description and Priority

In this feature Agents can edit and delete car that is active/available on renting. This feature has medium priority. In terms of feature risk scale, it is low.

2.4.13.2. Stimulus/Response Sequences

The Agent can edit or delete car by going into their My Rides option.

2.4.13.3. Functional Requirements

REQ-SF14-1: This feature will check if the Agent want to edit or delete the Car.

REQ-SF14-2: The Delete button will remove all of the Car data from Firebase database.

REQ-SF14-3: The system will give a toast to Agent that Car successfully deleted.

2.4.14. Edit and Delete Agent Profile

2.4.14.1. Description and Priority

In this feature Agents can edit and delete their profiles. This feature has medium priority. In terms of feature risk scale, it is low.

2.4.14.2. Stimulus/Response Sequences

The Agent can edit or delete their account by going into their settings option.

2.4.14.3. Functional Requirements

REQ-SF3-1: This feature will check if the Agent want to edit or delete their account.

REQ-SF3-2: The Delete button will remove all of the Agents data from Firebase database.

REQ-SF3-3: The system will give a toast to Agent that Account successfully deleted.

2.4.15. Edit and Delete Client Profile

2.4.15.1. Description and Priority

In this feature Agents can edit and delete their profiles. This feature has medium priority. In terms of feature risk scale, it is low.

2.4.15.2. Stimulus/Response Sequences

The Agent can edit or delete their account by going into their settings option.

2.4.15.3. Functional Requirements

REQ-SF3-1: This feature will check if the Agent want to edit or delete their account.

REQ-SF3-2: The Delete button will remove all of the Agents data from Firebase database.

REQ-SF3-3: The system will give a toast to Agent that Account successfully deleted.

2.4.16. Payment

2.4.16.1. Description and Priority

In this feature client pay the agent. This feature has high priority.

2.4.16.2. Stimulus/Response Sequences

The client will pay the amount by going in payment option.

2.4.16.3. Functional Requirements

REQ-SF3-1: This feature allow client to pay their pending amount.

REQ-SF3-2: The payment option allows to pay amount by selecting the payment method.

REQ-SF3-3: The system will give a toast to client that Payment done.

2.5. Nonfunctional Requirements

2.5.1. Performance Requirements

- The application should save the instance of the logged in stakeholder; they don't need to login again once they have done it already.
- The application should not allow to move back on login UI without login out from the application.
- The application should have better response time on slow internet.
- The application should save the state if the application in on pause.
- The chat box should be able to do automatic refresh to get new messages
- Agents should be notified once a client starts a conversation with them.
- In find Agent, the nearby Agents profile should be displayed to Client.

2.5.2. Safety Requirements

- The application should be able to recover account if any user lost or forget its password.
- All user data will be saved in real time database, if somehow the application crashes the application should be able to get the data from database when it's working.

2.5.3. Security Requirements

- The Password should be saved in an encrypted form and not accessible by anyone.
- Email and password will be required from every user to login with.

- The application should use firebase Email authentication for verification of the user login.
- The application should check if the input is valid or not.
- Each user will have unique User-ID, so the data should be an instance to that ID.

2.5.4. Usability Requirements

- The application UI should be user friendly.

2.5.5. Reliability Requirements

- The application should be available on any internet connection.
- The application should maintain its data if the application crashes.

2.5.6. Maintainability/Supportability Requirements

- The application should be testable.
- The application should not crash on data-load.
- Input checks must be fulfilled.

2.5.7. Portability Requirements

- The application should be able to run on every android device.

2.5.8. Efficiency Requirements

- The application should be able to get the data from the database on any android device.

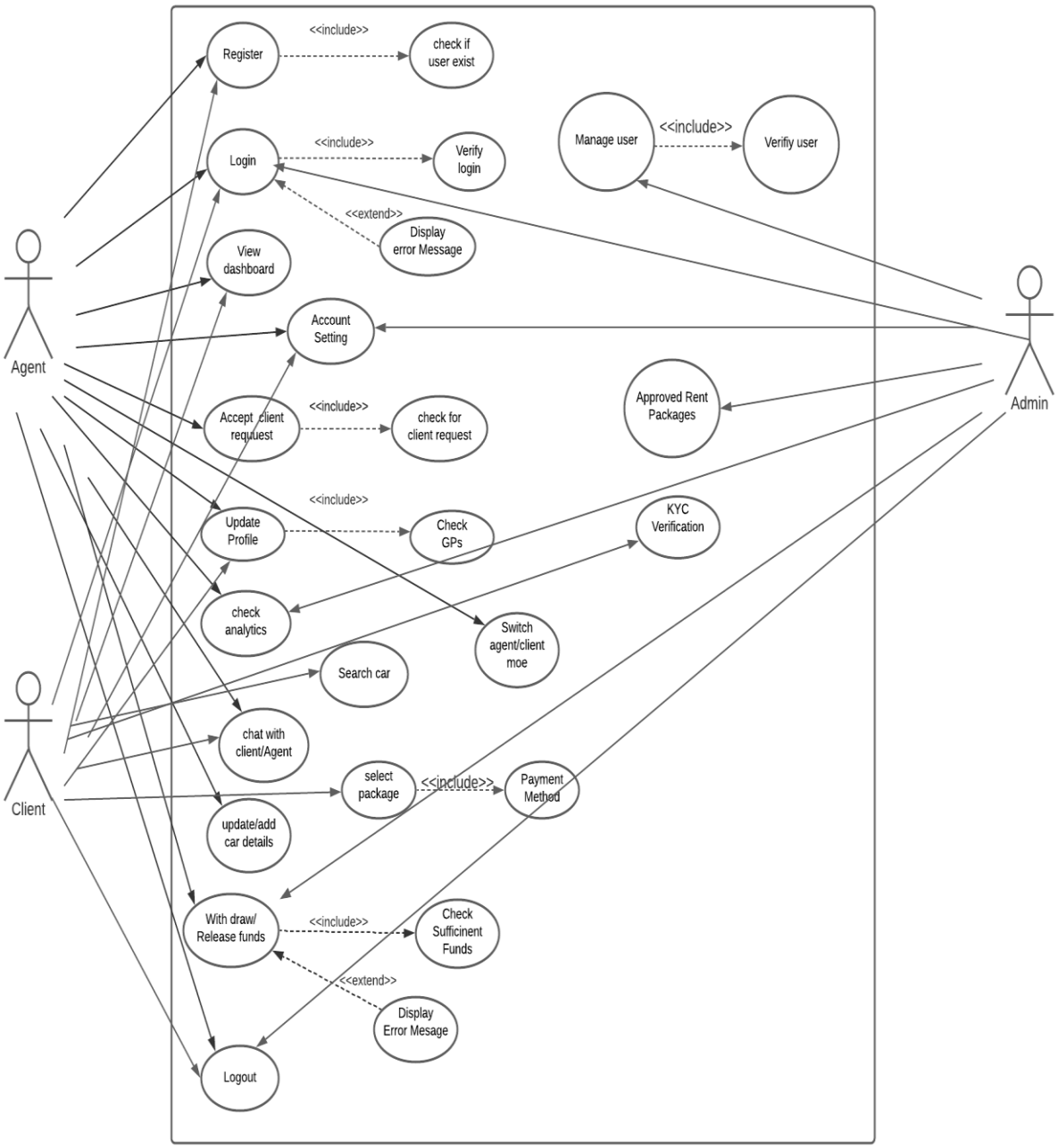
Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

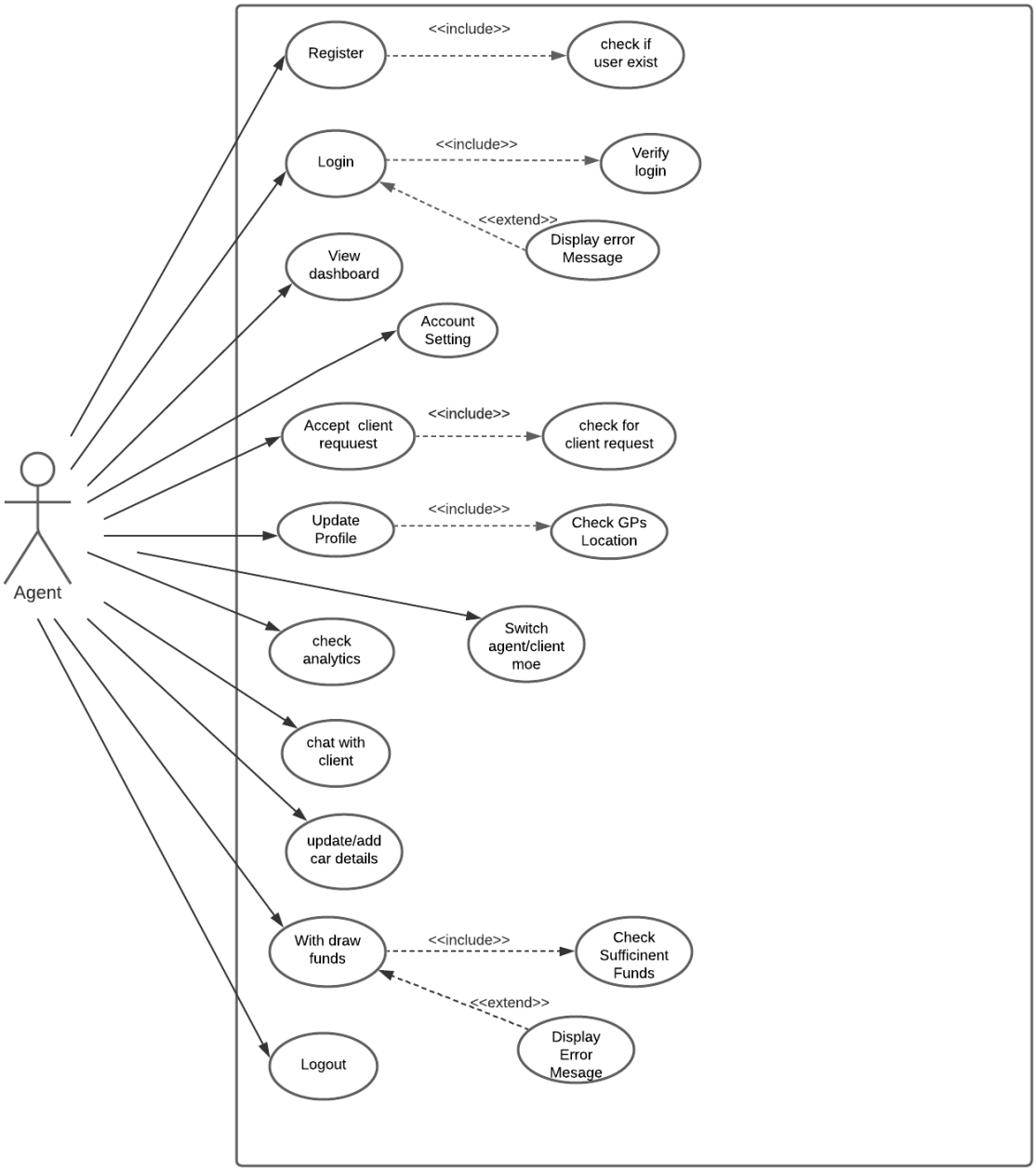
This chapter is about use case diagrams in which we properly define each and every functional and non-functional requirement according to the respected users.

3.1. Use Case Model



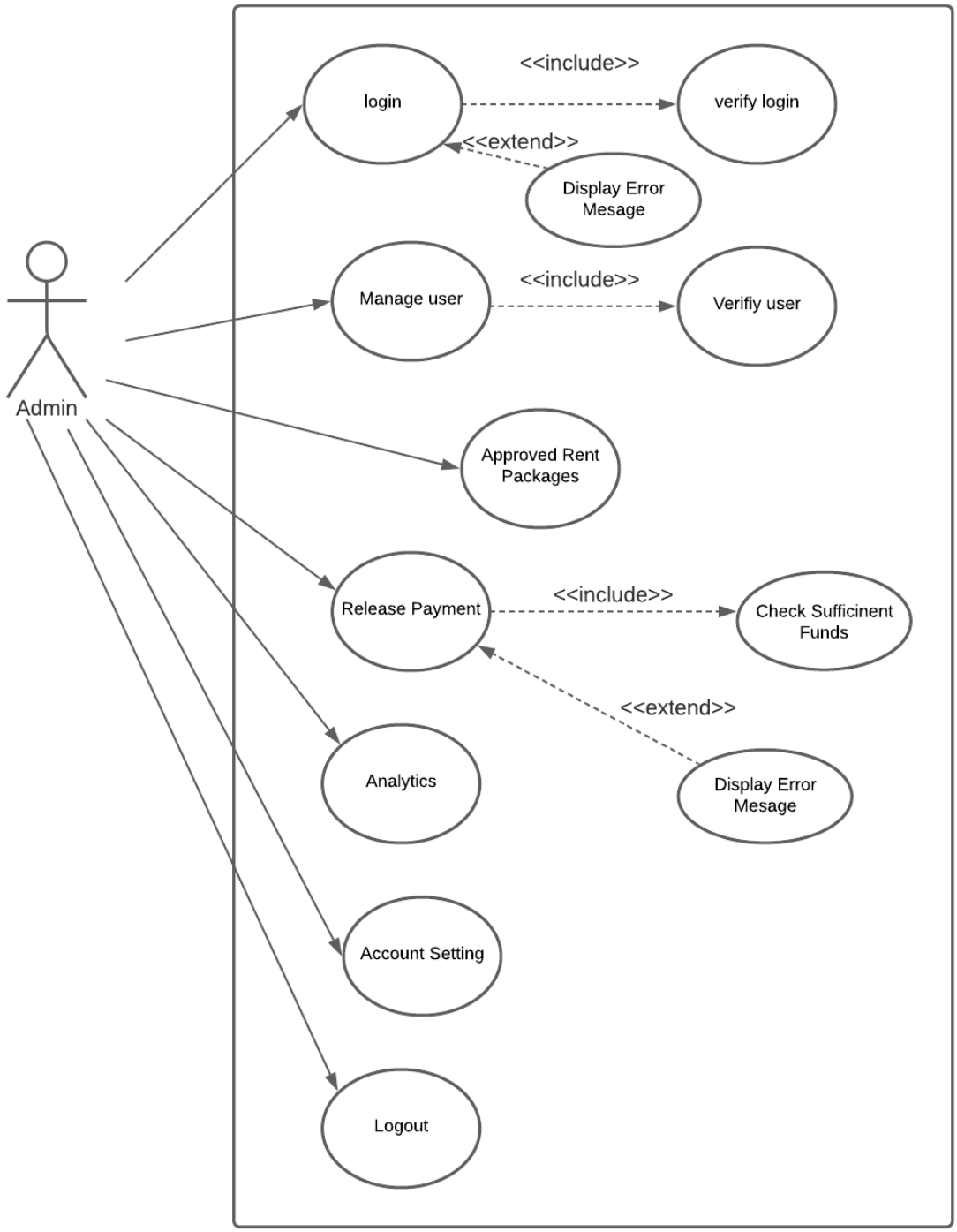
Following are the use cases for each individual actor in the system.

Use Case for Agent

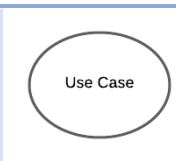
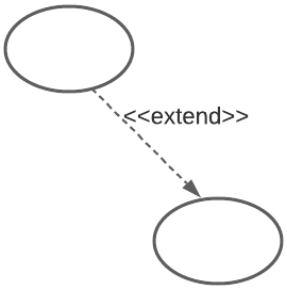


Use Case for Client

Use Case for Admin



3.2. Use Cases Description

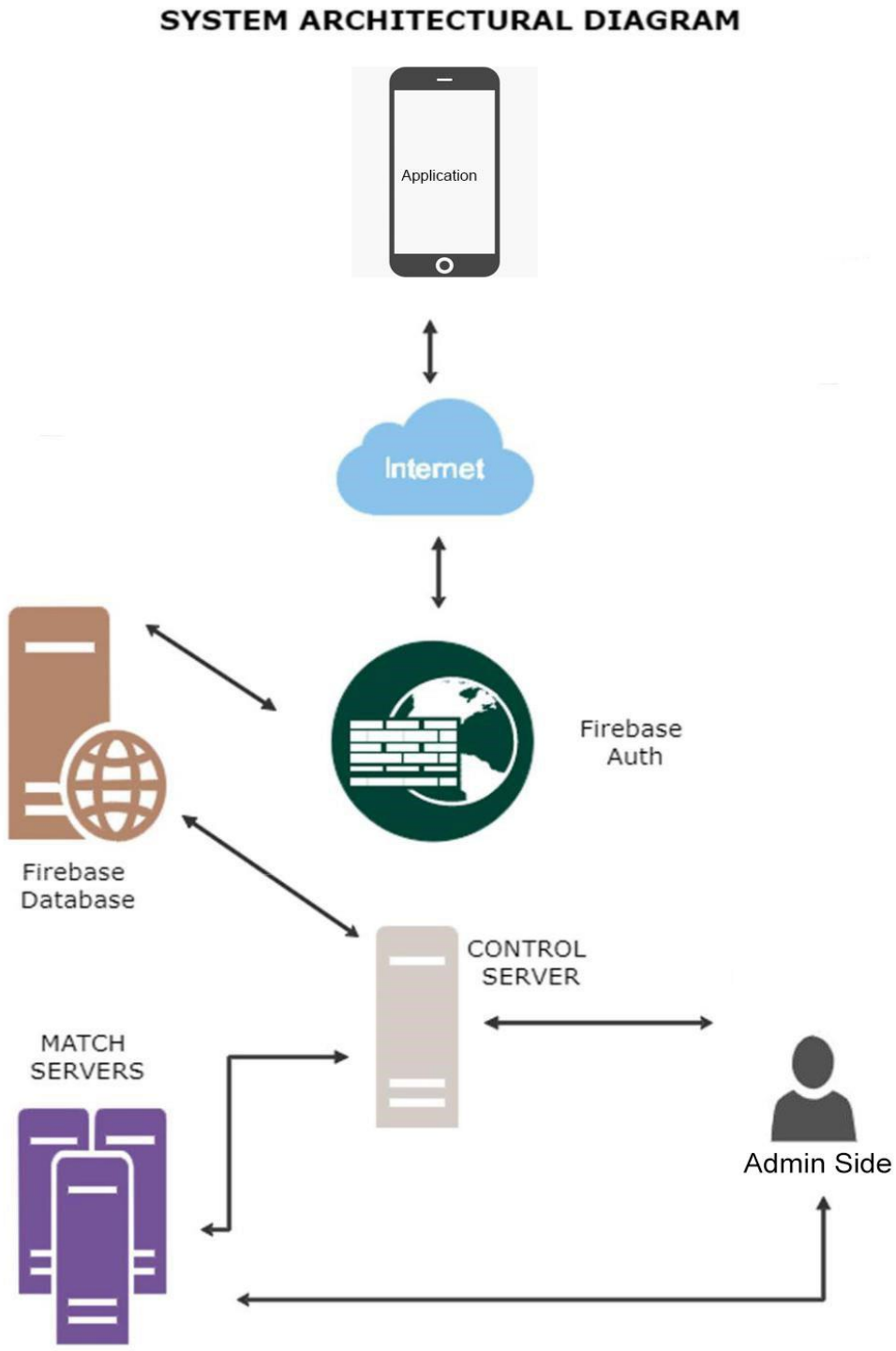
No.	Name	Description	Component
1.	Use Case	Use case are a list of action a user performs in the system to achieve a goal.	
2.	Actor	Actors in a system are people that interact with the system according to their roles.	
3.	Unidirectional Association	The relationships between and among the actors and the use cases	
4.	Include	Include is a directed relationship between two use cases which is used to show that behavior of the included use case (the addition) is inserted into the behavior of the including (the base) use case.	
5.	Extend	Extend is used when a use case conditionally adds steps to another first class use case.	

Chapter 4

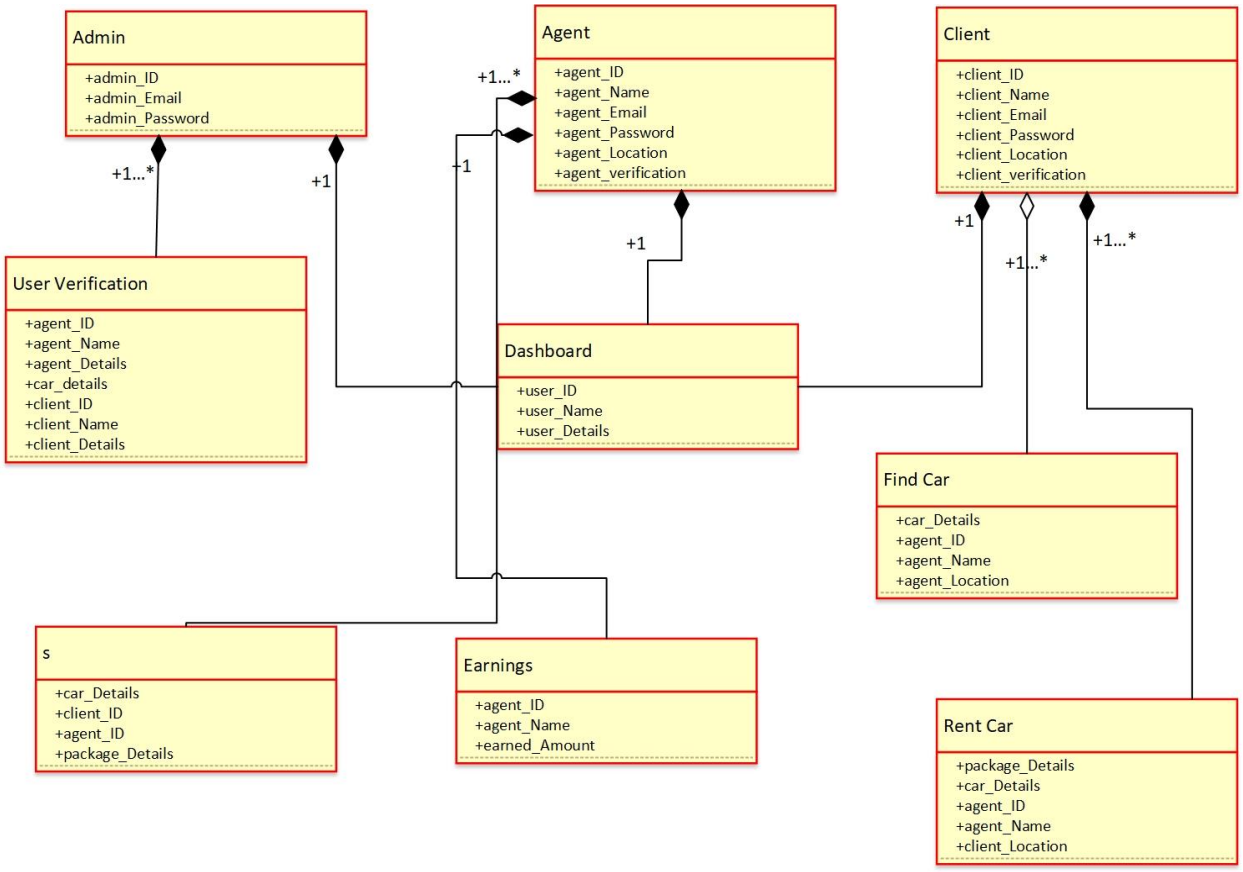
System Design

Chapter 4: System Design

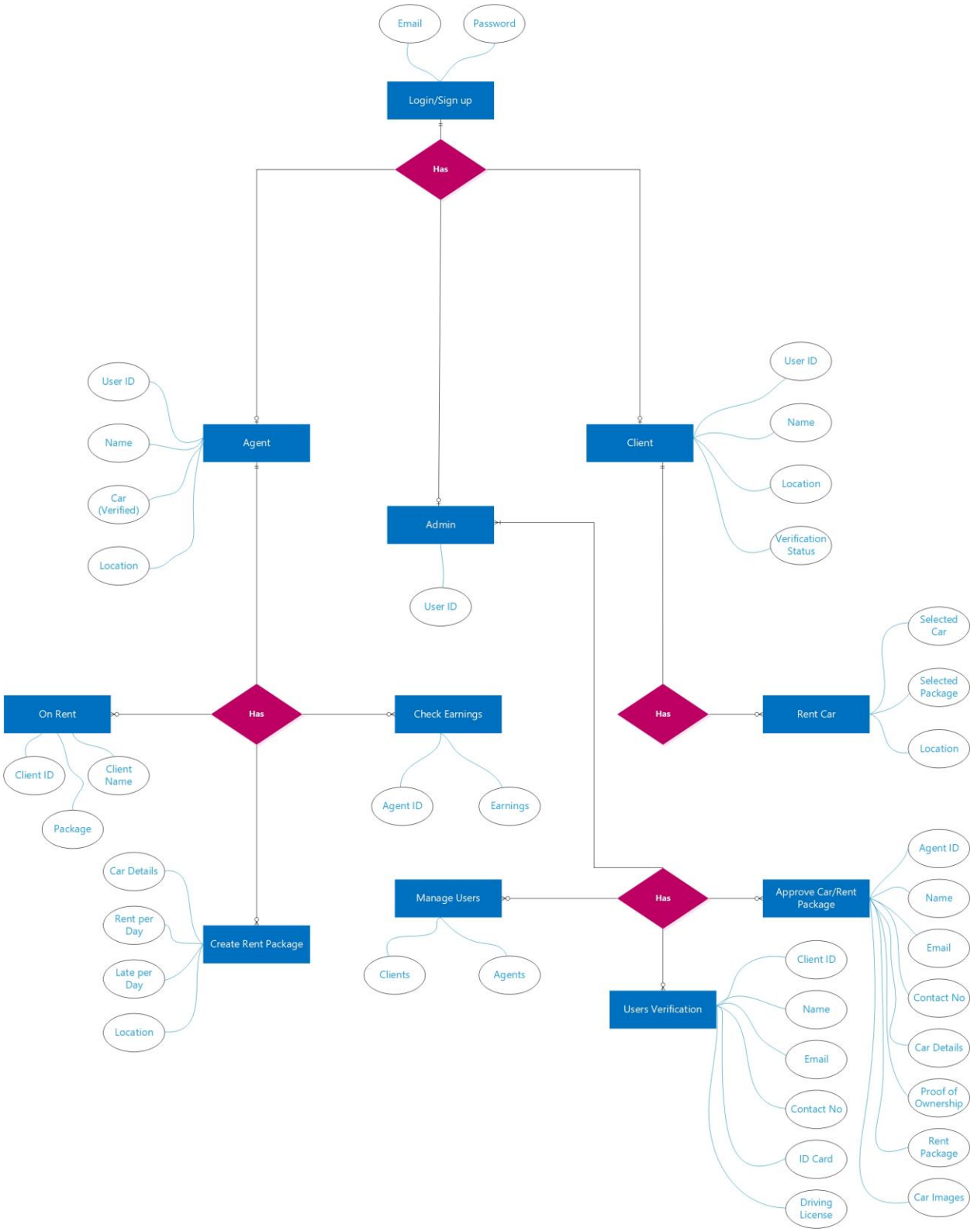
4.1. Architecture Diagram



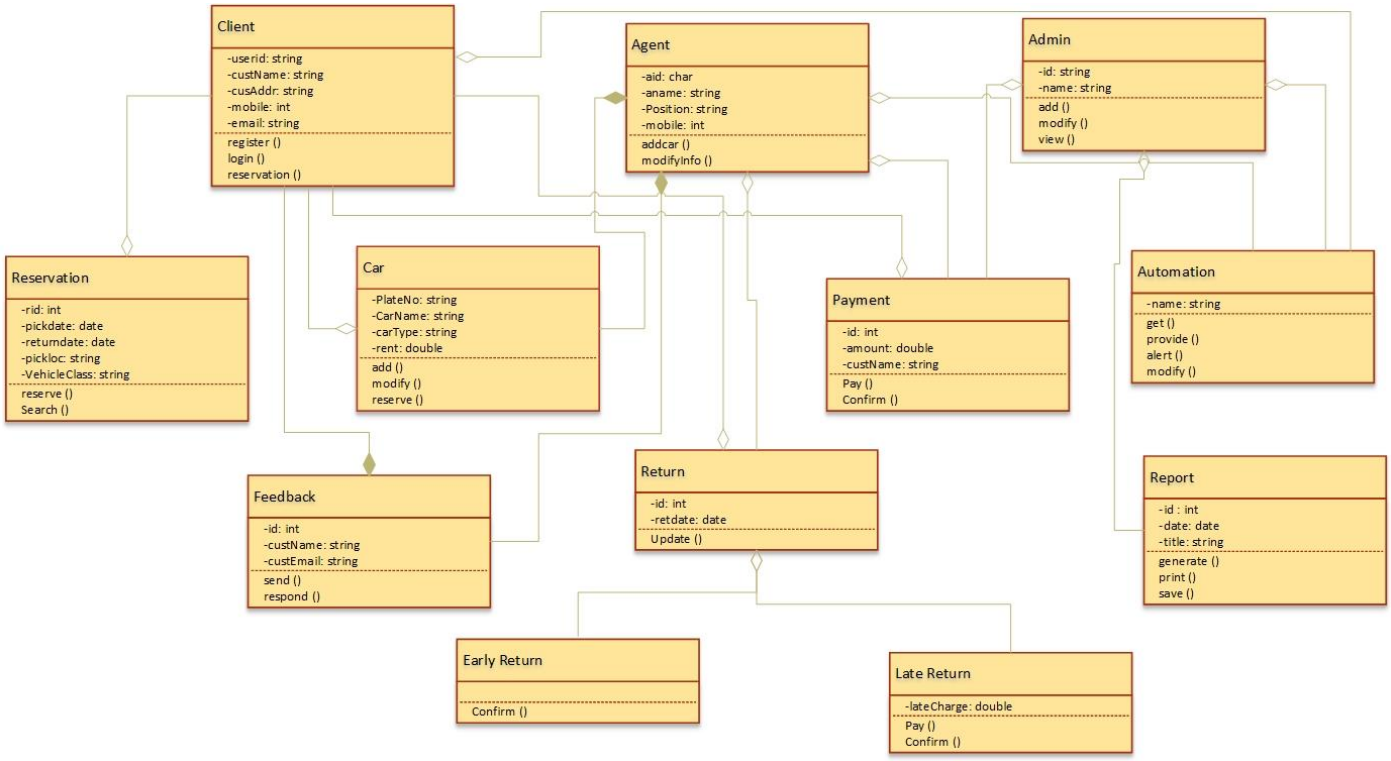
4.2. Domain Model



4.3. Entity Relationship Diagram with data dictionary

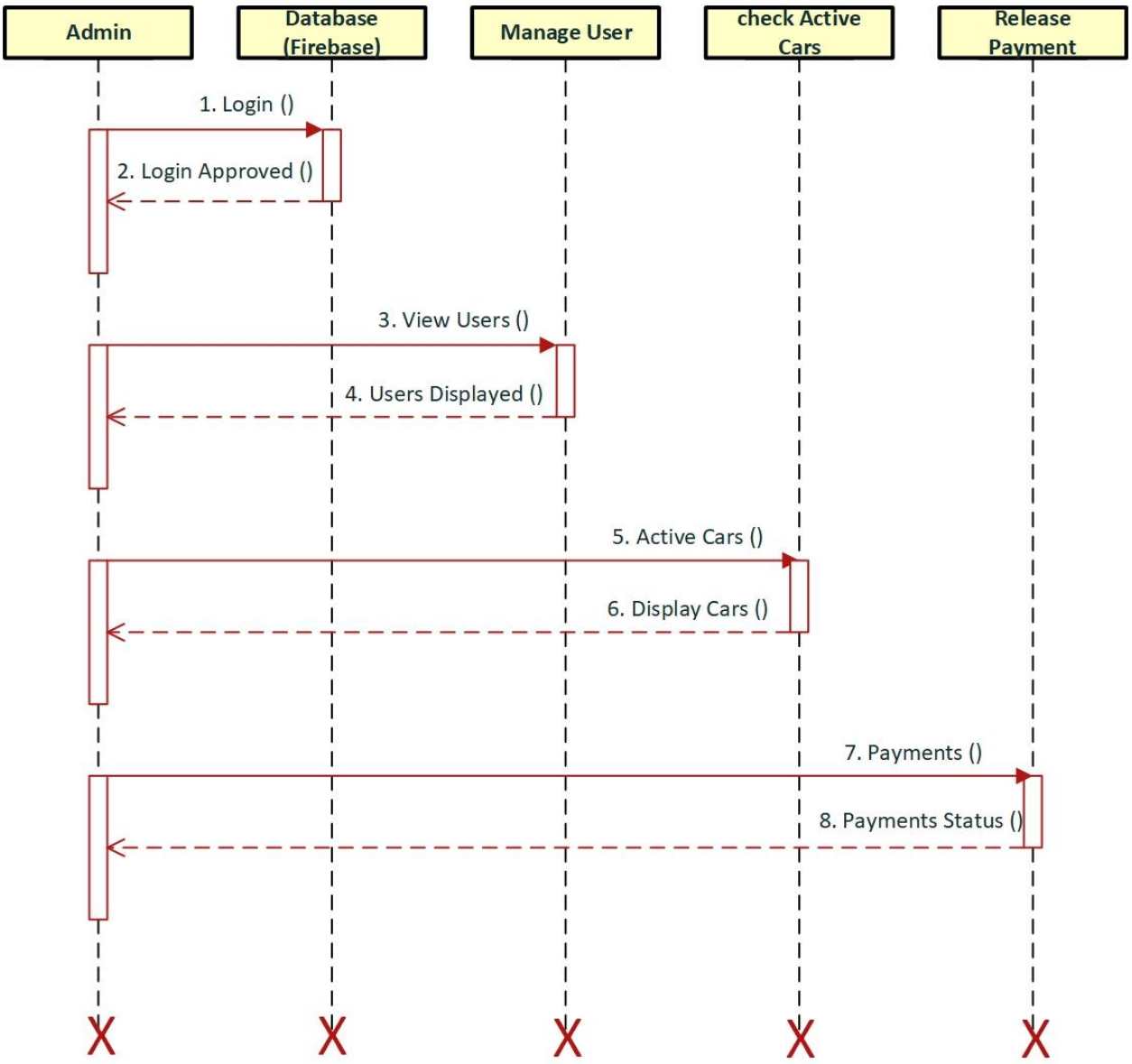


4.4. Class Diagram

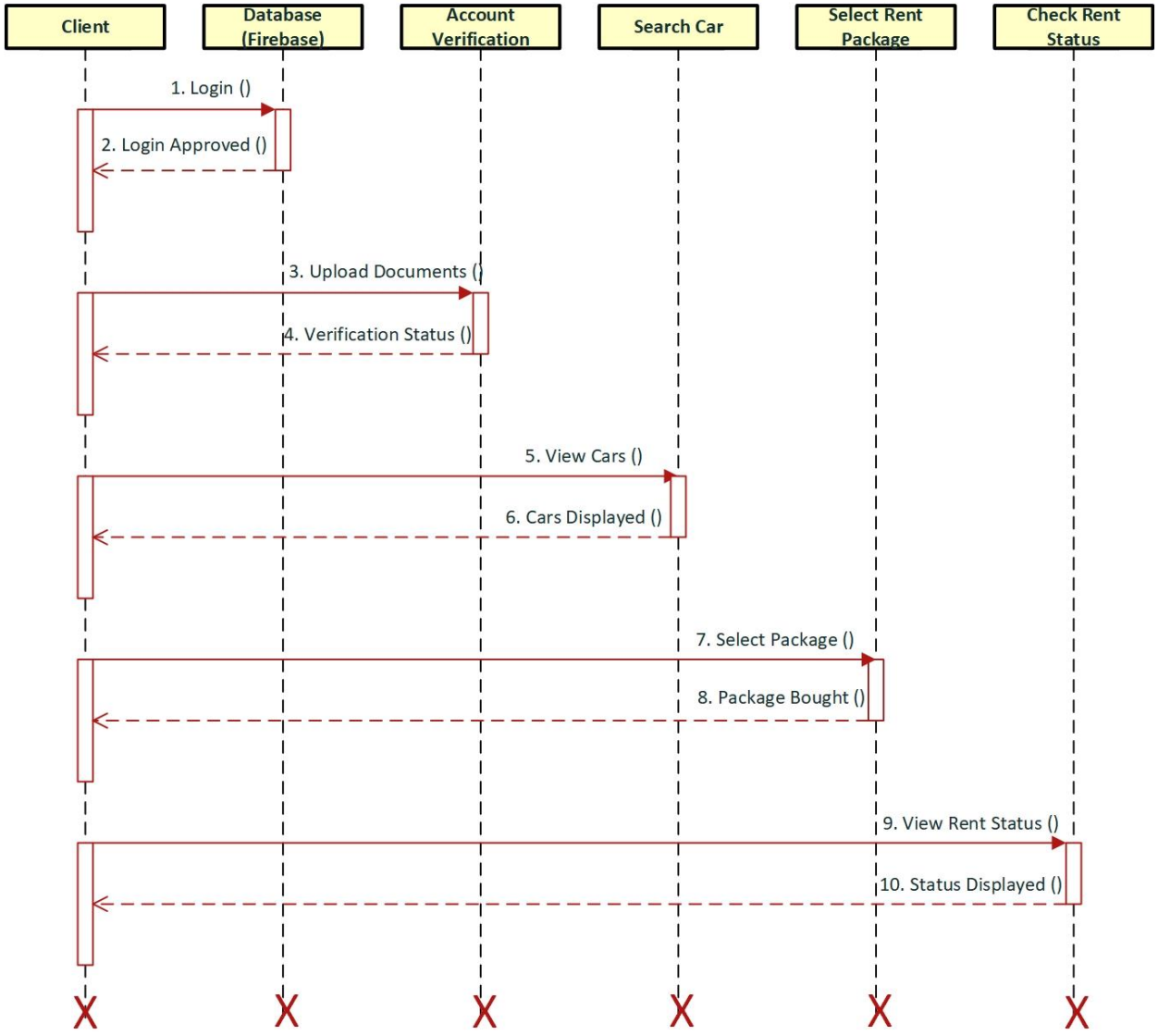


4.5. Sequence / Collaboration Diagram

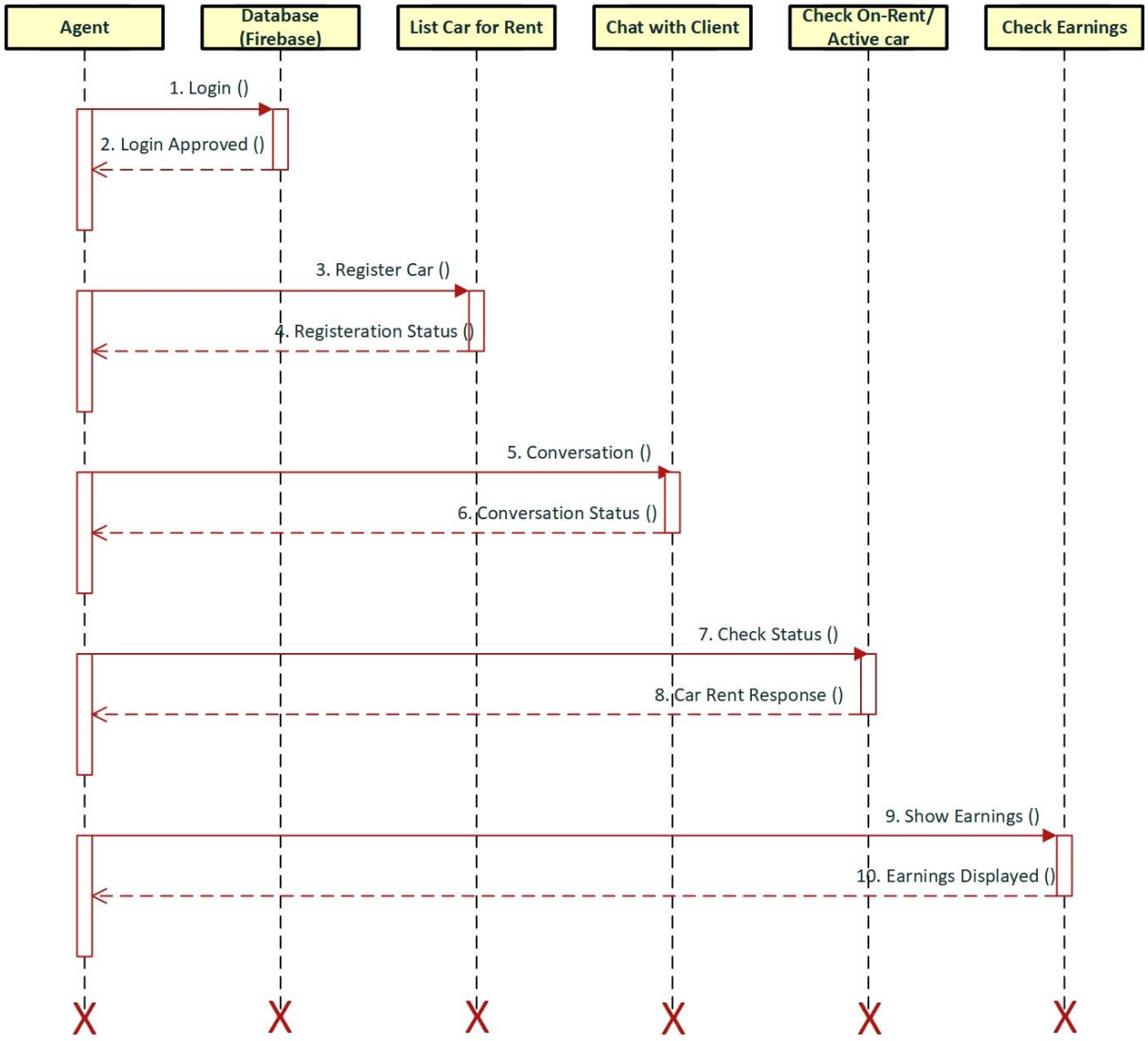
Admin Sequence / Collaboration Diagram



Client Sequence / Collaboration Diagram



Agent Sequence / Collaboration Diagram



4.6. Operation contracts

4.6.1. Login:

Operation Name:

- Login (Email, Password)

Cross Reference:

- Use-Case: Use-Case Model

Pre-Condition:

- User needs to have a login interface

Post Condition:

- Handle operations according to back-end functionalities (Email Authentication)

4.6.2. Registration:

Operation Name:

- Register (Email, Password, Repeat Password)

Cross Reference:

- Use-Case: Use-Case Model

Pre-Condition:

- User needs to interact with field of Email, Password and Repeat Password

Post Condition:

- Handle operations according to back-end functionalities (User is created in Database)

4.6.3. Rent a Car:

Operation Name:

- Rent a Car (Clickable Button)

Cross Reference:

- Use-Case: Use-Case for Client

Pre-Condition:

- User needs to have an interface of the application for an informed decision

Post Condition:

- Move towards desired activity (Enter Required Conditions)

4.6.4. Chat with Agents:

Operation Name:

- Chat with Agents (Clickable Recycler View)

Cross Reference:

- Use-Case: Use-Case for Client

Pre-Condition:

- User is displayed with a list of available Agents.

Post Condition:

- Handle operations according to back-end functionalities (Start chat interface with specified Agent.

4.6.5. Car Rent Request

Operation Name:

- Accept Car Rent Request (Clickable Button)

Cross Reference:

- Use-Case: Use-Case for Agent

Pre-Condition:

- User needs to have an interface for accepting or declining car rent request by a client.

Post Condition:

- Handle operations according to back-end functionalities (Client's Selected Package and Payment Method will be displayed to Agent).

4.6.6. Create Packages

Operation Name:

- Create Packages (Car Details, Price/day, Location)

Cross Reference:

- Use-Case: Use-Case for Agent

Pre-Condition:

- User (Agent) needs to have an interface for inputting required information

Post Condition:

- Handle operations according to backend functionalities (Store created packages in database)

4.6.7. Overview Users

Operation Name:

- Overview Users (Clickable Button)

Cross Reference:

- Use-Case: Use-Case for Admin

Pre-Condition:

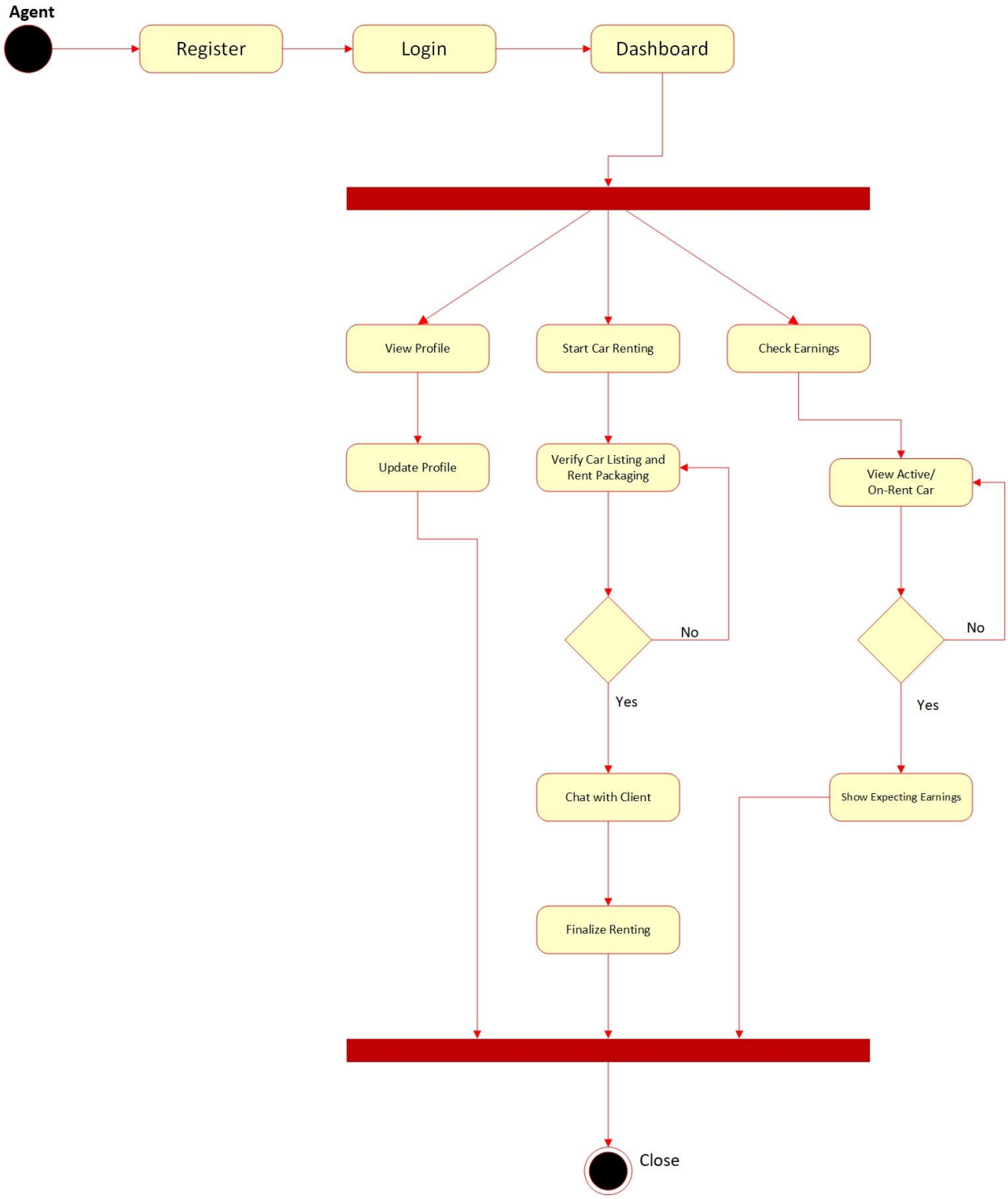
- User needs to have an interface for performing required functions

Post Condition:

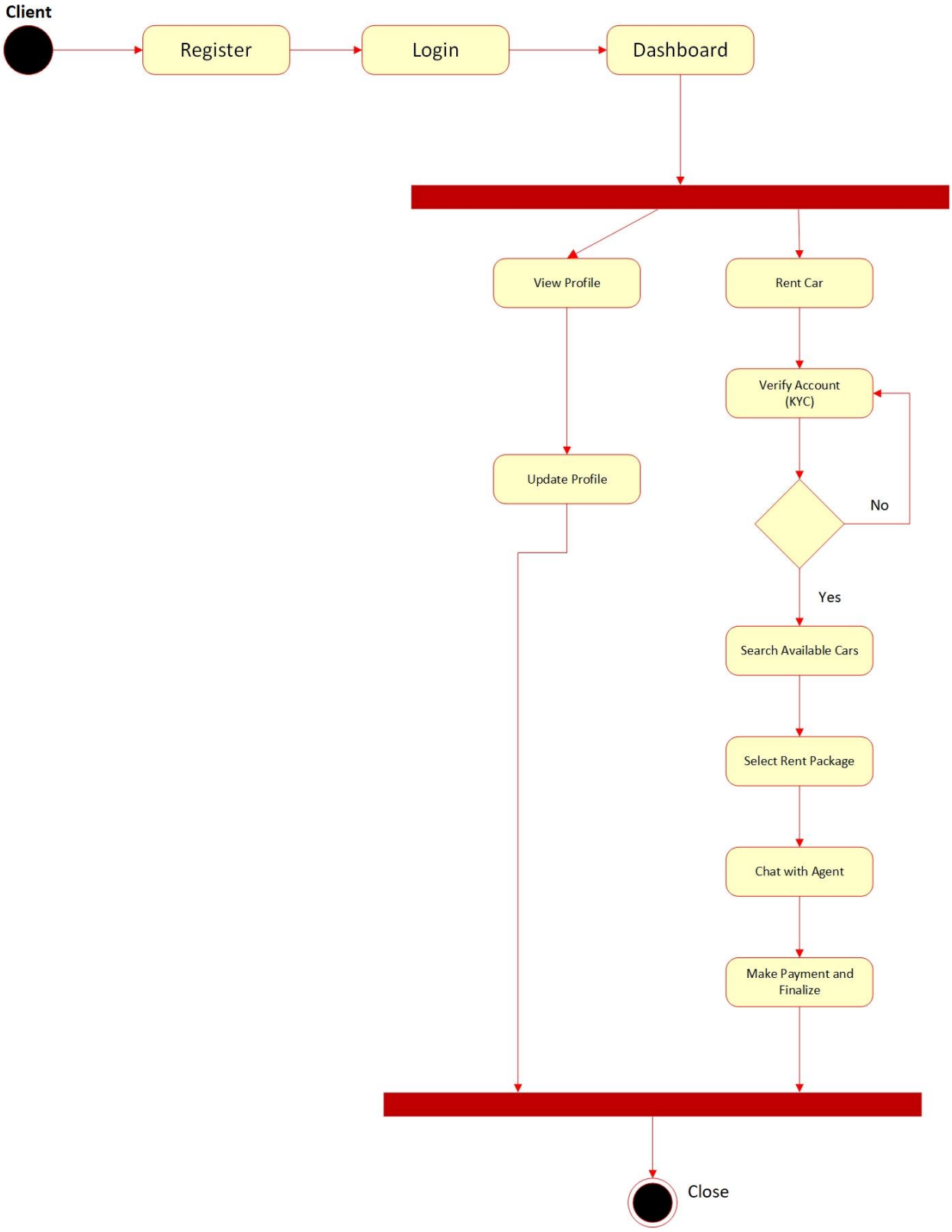
- Handle operations according to backend functionalities (Display all available users in database)

4.7. Activity Diagram

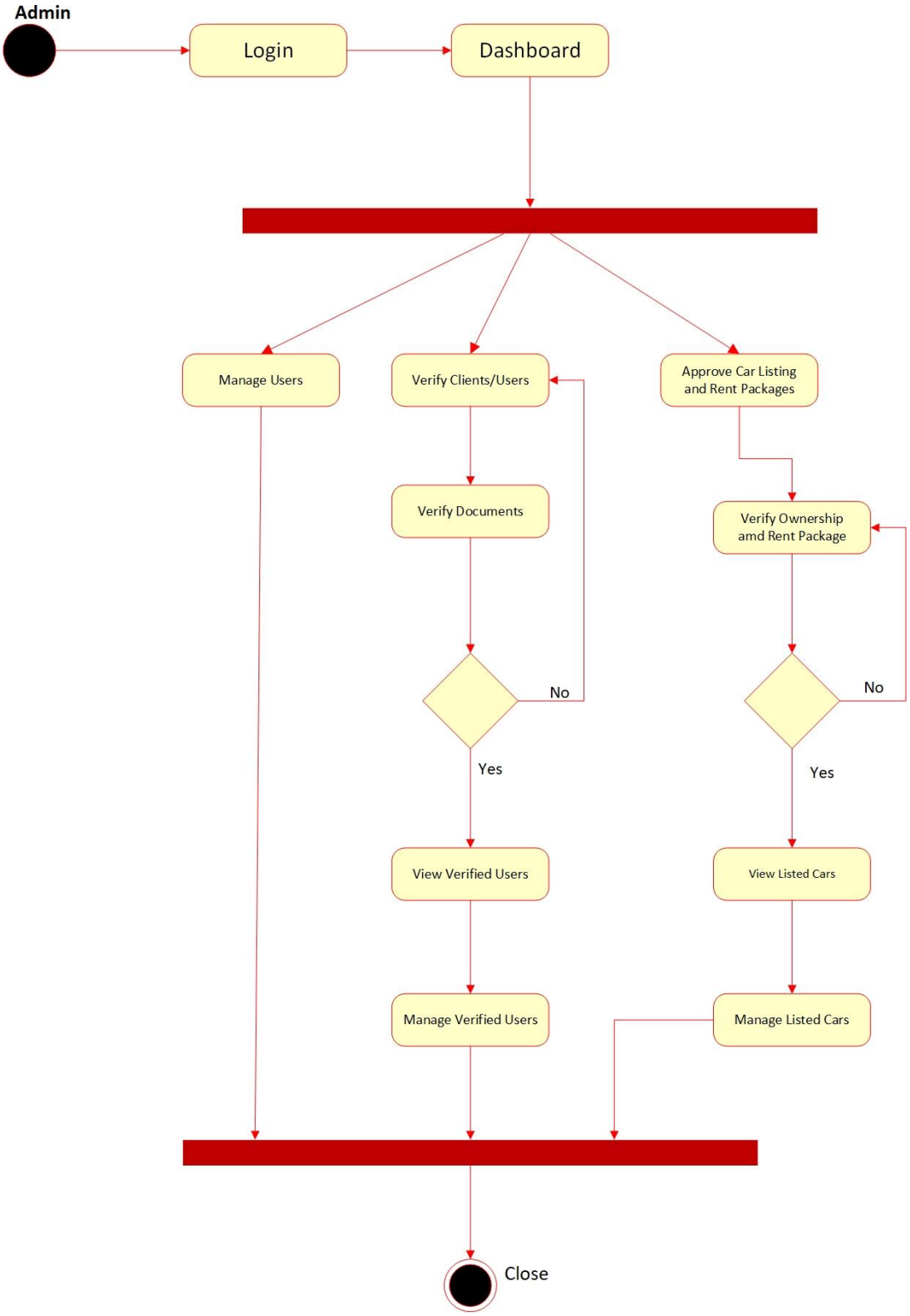
Agent Activity



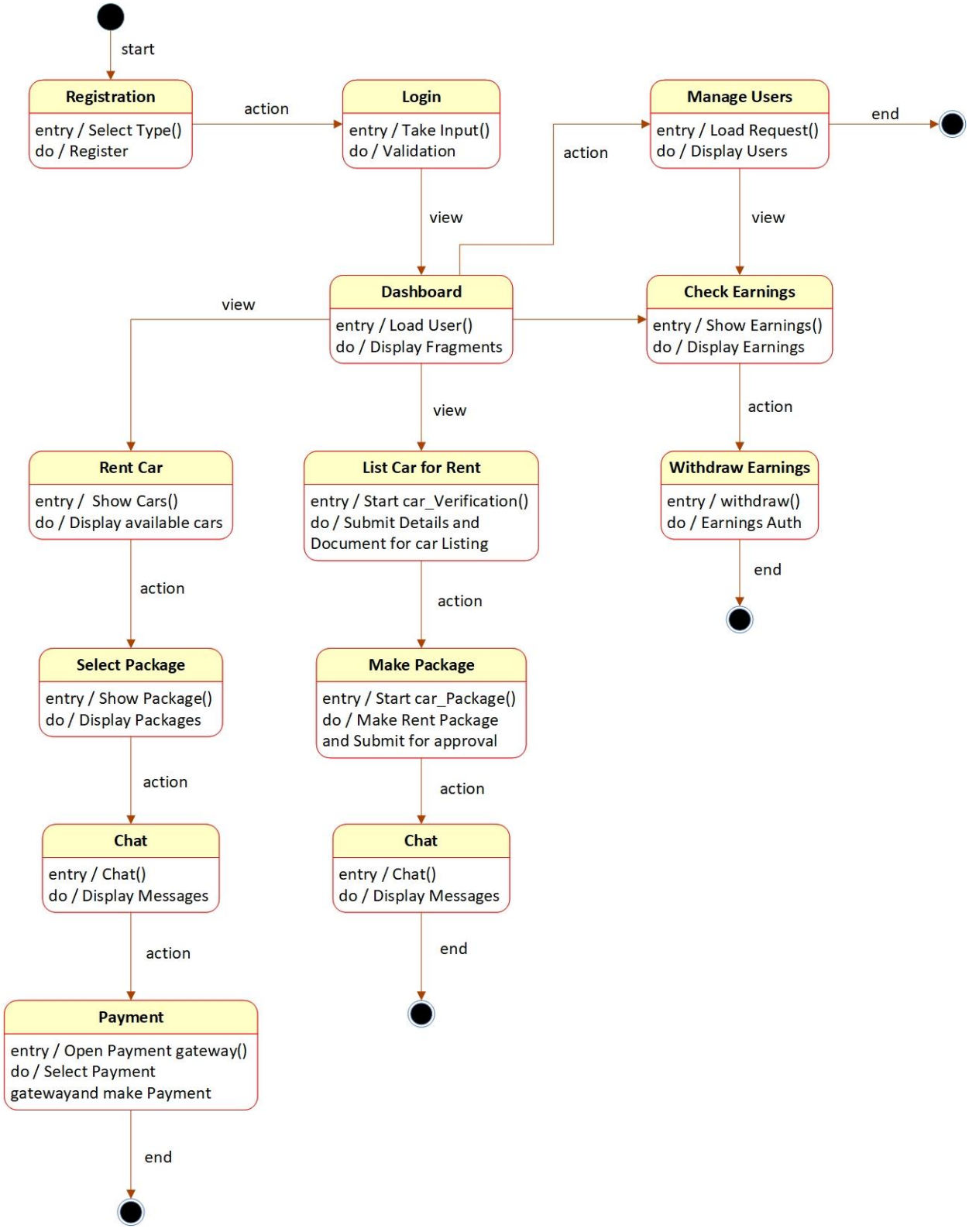
Client Activity



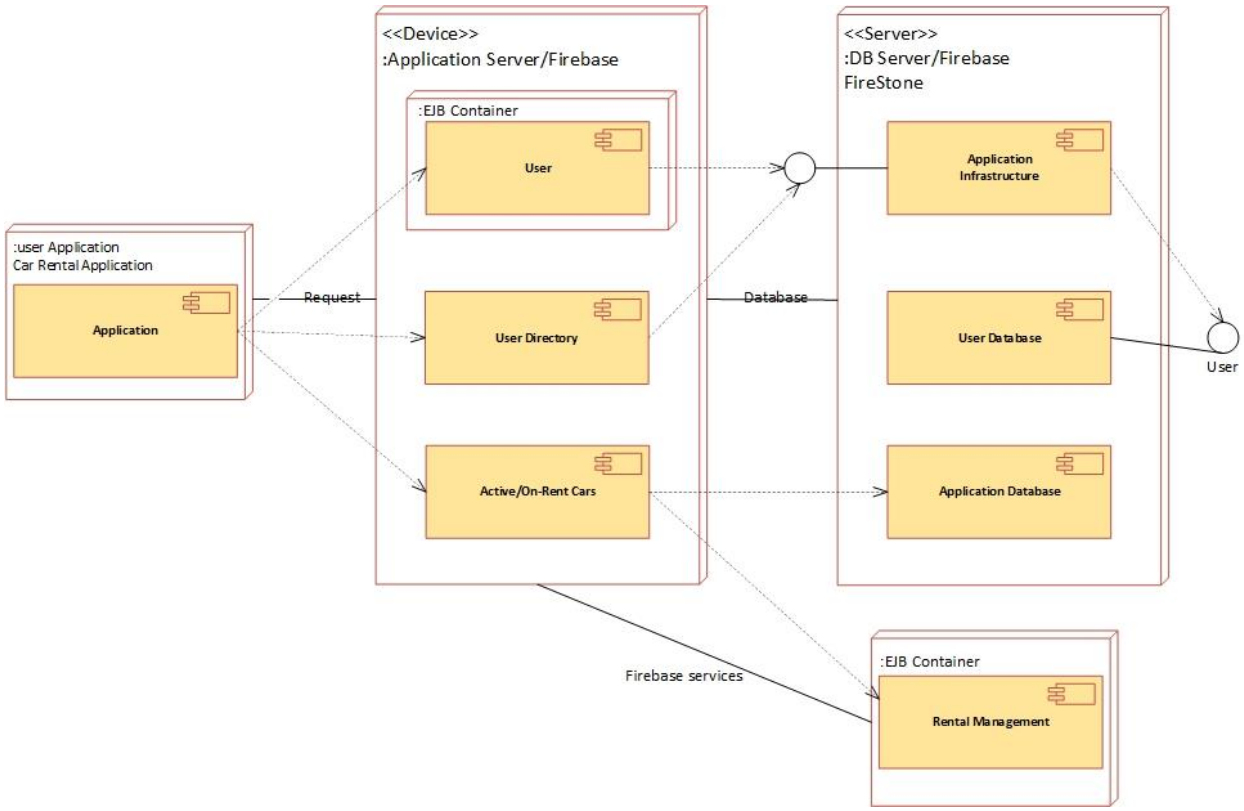
Admin Activity



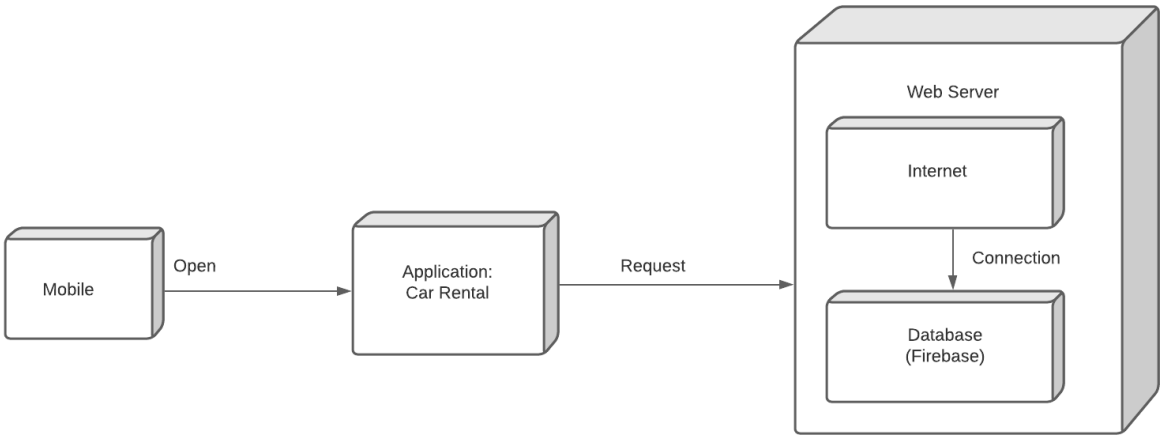
4.8. State Transition Diagram



4.9. Component Diagram

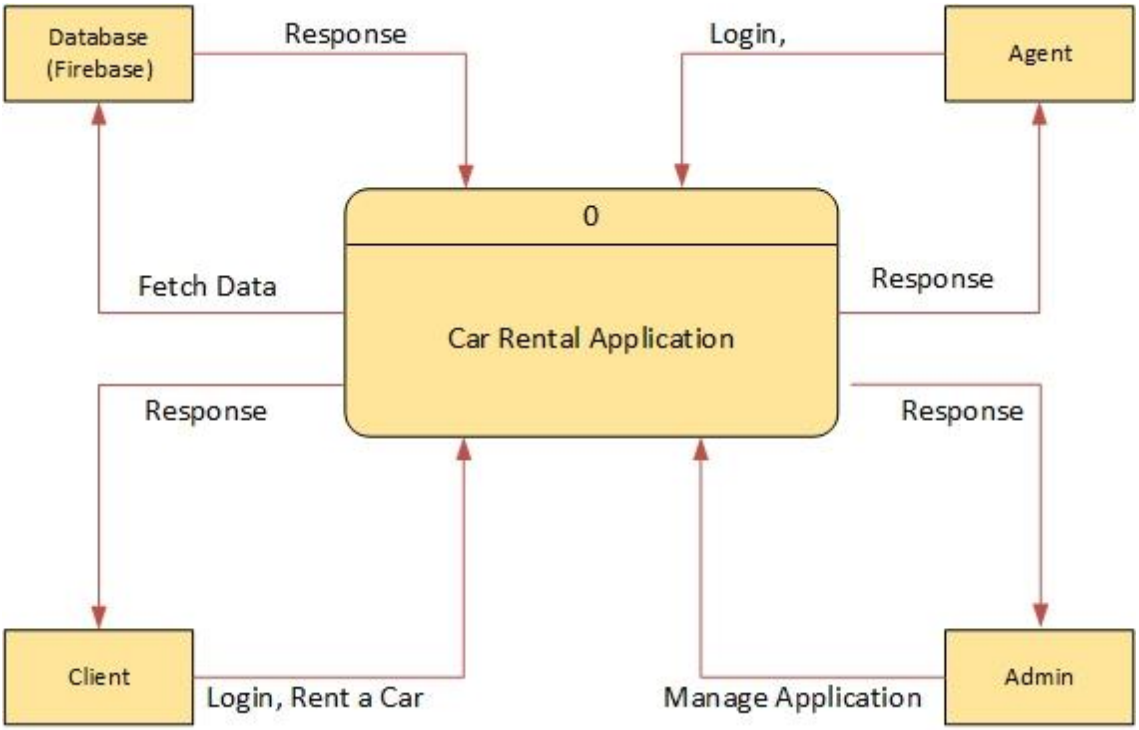


4.10. Deployment Diagram

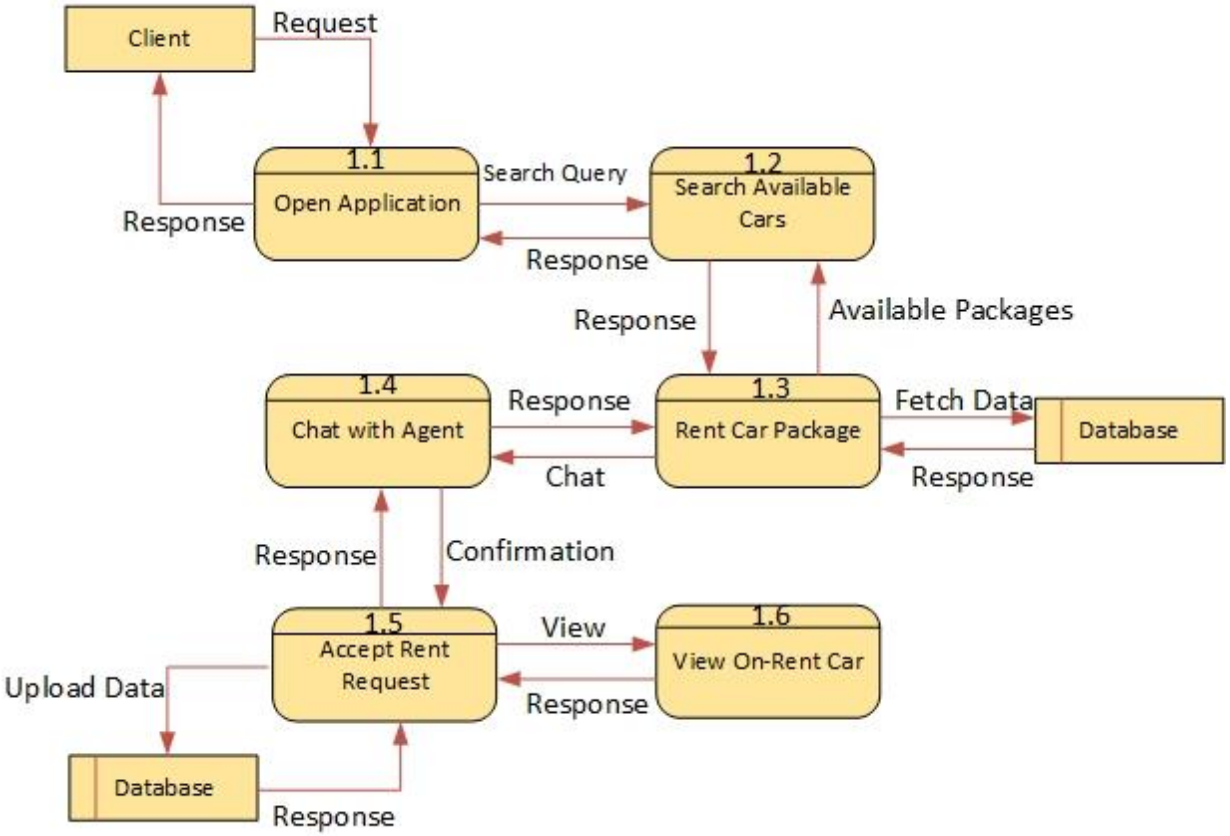


4.11. Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

Level 0



Level 1


Client
Request
1.1
Open Application
Response
Search Query
Response
1.2
Search Available Cars
Response
Available Packages
1.4
Chat with Agent
Response
Chat
1.3
Rent Car Package
Fetch Data
Response
Database
Response
Confirmation
1.5
Accept Rent Request
Upload Data
View
Response
1.6
View On-Rent Car
Database
Response

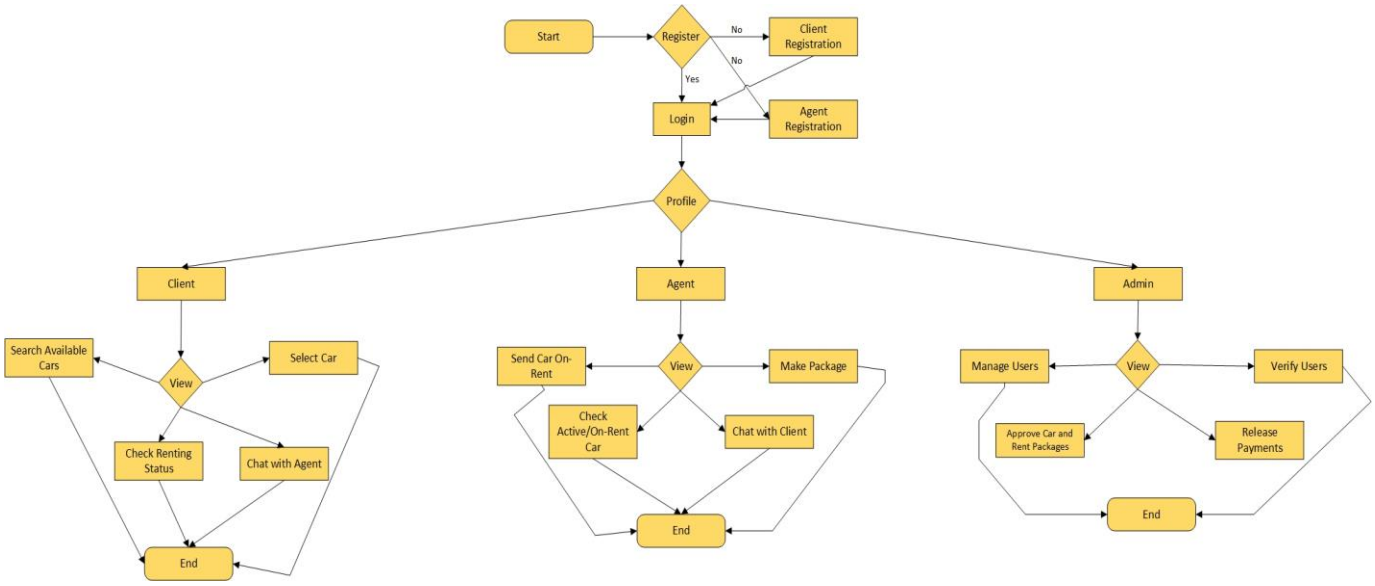
Chapter 5

Implementation

Chapter 5: Implementation

There is a wide range of modules used in this application such as Signing-up of users, how Renting a car work and dashboards of Client as well as Agents. In this chapter will walk through about every fundamental module and try to visualize the flow of advancing in this mobile based application.

5.1. Important Flow Control/Pseudo codes



5.2. Components, Libraries, Web Services and stubs

Following are the list of components used in this application:

- Firebase Firestone
- Firebase Real-time Database
- Lottie Animations
- Google Authentication
- Facebook and Instagram APIs
- Chat kit

5.3. Deployment Environment

Following are the deployment environments for Rentsly Application.

- Android Smartphones
- Firebase Database Server

5.4. Tools and Techniques

Following are the tools and techniques used for the deployment of Rentsly Application:

- Android Studio
- Microsoft Visual Code
- JDK

5.5. Best Practices / Coding Standards

Following are the practices and coding standards followed for the development of Rentsly Application:

- Improvement of application on the basis of user feedback
- User friendly design
- Simple coding conventions followed
- Everything is sorted in either files or folders

5.6. Version Control

Version Control for this application is minimum android version to properly run and download this application is 7.0 and minimum available space should be 1 GB.

Chapter 6

Testing

&

Evaluation

Chapter 6: Testing and Evaluation

In this chapter we will do all kinds of testing including unit testing and checking each individual components and module's integration with the system. By doing these tests we will find out that our system is ready for public use or not.

6.1. Use Case Testing

Login Test Case Number 1

Test Suit ID	**001**
Test Case ID	Login 01
Actor	Agent
Test Case Summary	If the Agent is already registered in the system, the he or she should be able to log in into the system
Related Requirements	Agent should be registered into the system
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Agent Category Enter Email and Password Click on Login button
Test Data	Email: Bazantahir@gmail.com Password: 11223344
Expected Result	The user must be able to go to the dashboard after successful login
Status	Pass

Login Test Case Number 2

Test Suit ID	**001**
Test Case ID	Login 02
Actor	Agent
Test Case Summary	Agent is trying to log in into the system with incorrect email and password
Related Requirements	Agent should be registered into the system
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Agent Category Enter Email and Password Click on Login button
Test Data	Email: Bazantahir@gmail.com Password: 112233
Expected Result	Error dialogue box will appear with the message wrong username or password
Status	Pass

Login Test Case Number 3

Test Suit ID	**002**
Test Case ID	Login 03
Actor	Client
Test Case Summary	If the Client is already registered in the system, the he or she should be able to log in into the system
Related Requirements	Client should be registered into the system
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Client Category Enter Email and Password Click on Login button
Test Data	Email: asim001@gmail.com Password: 11223344
Expected Result	The user must be able to go to the dashboard after successful login
Status	Pass

Login Test Case Number 4

Test Suit ID	**002**
Test Case ID	Login 04
Actor	Client
Test Case Summary	Client is trying to log in into the system with incorrect email and password
Related Requirements	Client should be registered into the system
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Client Category Enter Email and Password Click on Login button
Test Data	Email: asim001@gmail.com Password: 112233
Expected Result	Error dialogue box will appear with the message wrong username or password
Status	Pass

Register Test Case Number 1

Test Suit ID	**001**
Test Case ID	Register 01
Actor	Agent
Test Case Summary	Agent is able to register by providing valid email and password
Related Requirements	Agent will be able to log in into the system with the registered email
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must not be registered before
Test Procedures	Launch the application Select Agent Category Click on Sign-up Button Enter Email and Password Click on Create Account Button
Test Data	Email: asim001@gmail.com Password: 11223344
Expected Result	The user must be able to go to the login page after successful registration
Status	Pass

Register Test Case Number 2

Test Suit ID	**001**
Test Case ID	Register 02
Actor	Client
Test Case Summary	Client is able to register by providing valid email and password
Related Requirements	Client will be able to log in into the system with the registered email
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must not be registered before
Test Procedures	Launch the application Select Client Category Click on Sign-up Button Enter Email and Password Click on Create Account Button
Test Data	Email: asim001@gmail.com Password: 11223344
Expected Result	The user must be able to go to the login page after successful registration
Status	Pass

Update Profile Test Case Number 1

Test Suit ID	**001**
Test Case ID	Update Profile 01
Actor	Agent
Test Case Summary	Agent is able to update his or her profile after signing up
Related Requirements	Agent will be able to update his or her profile after logging in into the system with the registered email
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Agent Category Enter Email and Password Click on Login Button Head over to Profile Update Profile
Test Data	Name: Bazan Tahir Address: 251 A2 Johar Town Phone Number: 03063179378 Profile Picture: Any valid image will be converted to JPG format
Expected Result	The user must be able to update his or her profile after logging in into the application
Status	Pass

Update Profile Test Case Number 2

Test Suit ID	**001**
Test Case ID	Update Profile 02
Actor	Client
Test Case Summary	Client is able to update his or her profile after signing up
Related Requirements	Client will be able to update his or her profile after logging in into the system with the registered email
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be registered before
Test Procedures	Launch the application Select Client Category Enter Email and Password Click on Login Button Head over to Profile Update Profile
Test Data	Name: Muhammad Asim Address: 251 A2 Johar Town Phone Number: 03154226733 Profile Picture: Any valid image will be converted to JPG format
Expected Result	The user must be able to update his or her profile after logging in into the application
Status	Pass

Find Agents Test Case Number 2

Test Suit ID	**001**
Test Case ID	Find Agents 01
Actor	Client
Test Case Summary	Client is able to find Agents on the basis of his location
Related Requirements	Client will share the destination where he/she want to go and Agents will be shortlisted on the basis of the location
Pre-Requisites	Application must be installed on the device. Device must have an active internet connection User must be logged in into the application
Test Procedures	Launch the application Select Client Category Enter Email and Password Click on Login Button Dashboard will appear Click Find Agents Button
Test Data	Location share
Expected Result	The user must be able to find agents on the basis of his or her Business / Brand's location.
Status	Pass

Delete Profile Test Case Number 1

Test Suit ID	**001**
Test Case ID	Delete Profile 01
Actor	Agent
Test Case Summary	Agent is able to delete his or her profile
Related Requirements	Agent should be registered in the application
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must be logged in into the application
Test Procedures	Launch the application Select Agent Category Enter Email and Password Click on Login Button Open Profile Open Settings Click on Delete Profile
Test Data	Enter Password in order to confirm Profile Deletion Password: 11223344
Expected Result	The user should be able to delete his or her profile
Status	Pass

Delete Profile Test Case Number 2

Test Suit ID	**001**
Test Case ID	Delete Profile 02
Actor	Client
Test Case Summary	Client is able to delete his or her profile
Related Requirements	Client should be registered in the application
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must be logged in into the application
Test Procedures	Launch the application Select Client Category Enter Email and Password Click on Login Button Open Profile Open Settings Click on Delete Profile
Test Data	Enter Password in order to confirm Profile Deletion Password: 11223344
Expected Result	The user should be able to delete his or her profile
Status	Pass

6.2. Add a new Car

Test Suit ID	**001**
Test Case ID	Add Car
Actor	Agent
Test Case Summary	Agent will add car details
Related Requirements	Agent should be registered in the application
Pre-Requisites	Application must be installed on the device Device must have an active internet connection User must be logged in into the application
Test Procedures	Launch the application Select Agent Category Enter Email and Password Click on Login Button Open Profile Open add a car Click to add
Test Data	New car will add with document.
Expected Result	Agent will now able to a new car on rent.
Status	Pass

6.3. Equivalence partitioning

Equivalence class partitioning (ECP) is a software or application Testing technique in which input data of a software unit divides into partitions of equivalent data. These test cases are designed to cover each and every portion at least once.

Derived Equivalence Classes

- Email with @ and .com. (Valid)
- Email without @ or .com. (Invalid)
- Password should be more than 5 characters. (Valid)

Serial Number	Test Data	Expected Outcome
1	asim001@gmail.com	True
2	Asim666#	False
3	11223344	True

6.4. Boundary value analysis

Boundary Value Analysis is another Black Box Test Design Technique, which is used to find the errors at boundaries of input domain rather than finding those errors in the center of input.

BVA Test Number 1: Wrong Password Input

Serial Number	Test Data	Outcome
1	112233	System will not accept this password as this is less than the minimum number characters.
2	11223344	System will accept this password as this is greater than 5 characters.

6.5. Data flow testing

Data flow testing is a family of test strategies based on selecting paths through the program's control flow in order to explore sequences of events related to the status of variables or data objects. Dataflow Testing focuses on the points at which variables receive values and the points at which these values are used.

In Data Flow Testing we will keep track of the following things:

- Behavior of Application on the basis of user input
- Reaction of Application in case it crashes
- Application status under rigorous testing and stress
- Application should be prepared for all exceptions and must not crash on the basis of User input.

6.6. Unit testing

Unit testing is the level of testing which involves individually testing unit of code to confirm that it works on its own. The main purpose this test is to check that each single unit of the software performs as perfectly designed.

Module Name	Test Data	Expected Data	Actual Result
Login	asim001@gmail.com 11223344	User should be logged in the Application	Pass
Sign up	asim001@gmail.com 11223344 11223344	User should be able to register in the Application	Pass
Agent Profile	Valid Login	View logged in Agent's Profile	Pass
Client Profile	Valid Login	View logged in Client's Profile	Pass

6.7. Integration testing

Test ID	Test Case Description	Procedure	Expected Result	Actual Result
1	When user click login button he or she should be able to login to the application	Enter Username and Password Click Login Button	If Username and Password is correct, user should be able to login	User logged in successfully.
2	When user clicks rent a car he or she should be displayed available agents	Click find Agents	If the entered location is correct display available agents in the area	Agents displayed successfully.
3	View Profile	Click Update Button	Profile Preview while editing	Profile Displayed Successfully

6.8. Performance testing

Performance testing is kind of testing performed to check how a system performs in case of responsiveness and stability of software under a particular workload.

Test Case Number 1: System Response Time Testing

Test Type	Performance Testing	Test Result
Required Performance	The response time of the system should be under 6 seconds	Test Pending
System Performance	The actual response time is approximately 6 seconds	Test Passed

Test Case Number 2: Page Load Time Testing

Test Type	Performance Testing	Test Result
Required Performance	The response time of the system should be under 5 seconds	Test Pending
System Performance	The actual response time is 5 seconds	Test Passed

Test Case Number 3: Huge Numbers of Users

Test Type	Performance Testing	Test Result
Required Performance	System should behave normally and efficiently with huge number of active users	Test Pending
System Performance	The system performed well under stress	Test Passed

Test Case Number 4: Waiting Time Test

Test Type	Performance Testing	Test Result
Required Performance	The average waiting time of the system should be under 3 seconds	Test Pending
System Performance	The actual performance time of system is 2 seconds on an average	Test Passed

Test Case Number 5: System Crashing Test

Test Type	Performance Testing	Test Result
Required Performance	The system should be able to save data in case the application crashes	Test Pending
System Performance	The system is able to save users data before crashing	Test Passed

6.9. Stress Testing

In stress testing our system performed well even with large numbers of active users. Through this testing, we monitored our system's vitals and made sure our system worked properly:

- Application must save all data before crashing
- Making sure that Application maintains connection to and from server
- Application should update and delete records correctly

Chapter 7

Summary, Conclusion

&

Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Rentsly Mobile Application will automate the car rental process and provide rental of economy, standard and luxury automobiles in its targeted market. The system will provide comfort to end users to book a car or just to get estimated cost by using mobile application, and will provide opportunity to car owner to earn by renting out their car. Each type of car should have a different rental fee per day. Rental fee depends on number of days, brand and where the car is to be taken.

7.2. Achievements and Improvements

From this project we learnt a lot of things including how to work efficiently as a team and achieve goals. We learnt Project management tools and skills including time management, UML modeling language to specify our system, Software Quality assurance, testing and much more. During this process we have improved our coding skills and learnt new as well. We learnt Java, XML and Firebase Cloud Storage and how they correspond with each other while creating this application.

7.3. Critical Review

In order to provide a complete a critical review we need to understand a few concepts about this application. The first concept is that peoples to come on this application. This application is simply a car rental for clients. The payment process will be through either Jazz Cash or Easy Paisa.

7.4. Lessons Learnt

Lessons learnt from this final year project are as following:

- Learnt how to work together as a team
- Learnt new tools for Project Management
- Learnt about Universal Modeling Language and Importance of documentation

- Learnt about new tools and technologies
- Learnt about how to create efficient logics
- Learnt about different techniques to solve a problem
- Learnt about software quality assurance and testing
- Learnt about debugging and error handling
- Learnt how to develop better documentation

7.5. Future Enhancements/Recommendations

Future enhancements or recommendations for Rentsly Application will be:

- Deploy Web Application
- Work more on making mobile Application smooth and optimized
- Take feedback from active users and improve system using their suggestions

Appendices

Appendix A: Information / Promotional Material

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this appendix all about*]

A.1. Broacher

A.2. Flyer

A.3. Standee

A.4. Banner

A.5. First Level heading [16 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

A.5.1. Second level heading [14 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

A.1.1.1. Third level heading [12 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

Appendix [no.]: Appendix Title

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

A.1. First Level heading [16 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

A.1.1. Second level heading [14 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

A.1.1.2. Third level heading [12 pt, Calibri, Bold, Left aligned]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

Reference and Bibliography

Reference and Bibliography

[1] M. Sher, M. Rehman, "*Title of the Paper*" Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.

[2] ……

Index

Index