Goods Transport System

Thesis submitted to the Institute of Information Technology, Quaid-i-Azam University, Islamabad, for the partial fulfillment of the degree of Master of Science in Information Technology.

Submitted By:

Aimen Khan

Supervised by:

Dr. Khurram Gulzar Rana

Institute of Information Technology Quaid-I-Azam University Islamabad

Session 2019-2021

Statement of Submission

This is certifying that ―Aimen Khan” Registration No. ―01161911039” has successfully completed the final project as **“Goods Transport System”** Quaid-I-Azam University, Islamabad to fulfill the partial requirement of the degree **“Master of Science in Information Technology”.**

External Examiner

Dr. Khurram Gulzar Rana

Acknowledgement:

All the praises, thanks and acknowledgments are for the creator Allah Almighty, the most Beneficent, the most Merciful, who gave me strength and enabled me to undertake and execute this task.

First and foremost I would like to express my deepest gratitude to all those who provided me the possibility to complete this project. A special gratitude to my respected teacher and supervisor Dr. Khurram Gulzar Rana, whose guidance and kindness helped me to coordinate in my project. Furthermore, I would like to give thanks to all faculty members of Information Technology department for their help, giving proper attention and time.

In addition I would also like to express my gratitude to my parents and siblings for their encouragement, which helped me throughout my life. My thanks and appreciation also goes to my class fellows and friends, who have willingly helped me out with their abilities.

Aimen Khan

Contents

List of Tables

List of Figures

Abstract

The project designs and develops an android-based application that provides user to request for booking (along with luggage information) and driver (along with car details) to accept the booking after they have registered themselves on to the application. They both provide information about themselves like name, mobile number, location, CNIC while registering. Then after logged in to the system, user can request for booking. Driver can accept booking request if luggage capacity specified by user is less than or equal to driver's transport luggage capacity.

This application is a booking app and works like Careem and Uber that books a truck to shift household appliances and luggage. It is useful for all those interested in shifting home or office from one place to another. As Shifting is laborious work. As they spend most of the time by visiting offices and gather information. First one has to come out of his/her comfort zone to ride a heavy vehicle to carry home appliances and other home luggage. Such verbal booking is not reliable and systematic in the sense that such shifting can cost a lot to some people especially when they are not aware of transportation cost.

This application will reduce the struggle required to shift home appliances from one place to another place within a city and even outside the city in few clicks. By this application user can directly book vehicle for their house hold transportation. It is designed for the user to save time and money.

Project Brief

Project Title	Goods Transport System
Undertaken By	Aimen Khan-01161911039
Supervised By	Dr KhurramGulzar Rana
Date Started	20-10-2020
Date Completion	
Language and Technology Used	Java HTML5 CSS Firebase Database GPS
Tools Used	Android Studio Drwa.io MS word
Operating System	Window10
System Used	Intel(R)Core (TM) i3-8130UCPU@ 2.20GHz 2.21GHz

CHAPTER 1

INTRODUCTION

1.1 Introduction

Goods Transport app is an Android Based Mobile Application that books a vehicle to shift household appliances and luggage. This application is a booking app and works like Careem and Uber. It is useful for all those interested in shifting home or office from one place to another. Shifting is laborious work. First one has to come out of his/her comfort zone to ride a heavy vehicle to carry home appliances and other home luggage. Such verbal booking is not reliable and systematic in the sense that such shifting can cost a lot to some people especially when they are not aware of transportation cost. This application will reduce the struggle required to shift home appliances from one place to another place within a city and even outside the city.

1.2 Problem Definition

There are no online services (in the form of application) for luggage transportation in Pakistan. When citizens shifting home from one city to another and within city it's difficult to search for truck in short any transportation/automobile. It's a daily life problem. Many people don't know where to find automobiles for this reason. In many places, automobiles are very far from that location. In situations like this it's a headache for people to transport that luggage because household items are too many. Also the luggage is very heavy. So, it's problematic for people to take them to destination place on their own.

1.3 Objective

This application is structured to give citizens administration to make their life simple by overcoming the difficulty of luggage transportation from one place to another place i.e. within the city or from one city to another city. Using this application, they will register themselves and add their current location, desired location where they want to send their luggage for booking. Basic Objective of this application is to make transportation process convenient for everyone.

1.4 Scope

This application is made to provide a service to citizens so that they can request for booking and transportation of their luggage from one place to other. Its scope is for home shifting. When citizens are about to shift whether within a city or outside the city, they first register themselves on app and then make a request. Get the transport, loads the luggage and when reached to desired location give the payment to driver. Add the feedback along with rating and complaints (if any).

1.5 Motivation

The motivation for designing this application is to provide citizens a platform which solve issues like bargaining. People normally do bargaining in daily life in order to save their money by least spending them. While doing so it creates a chaos for both, user who wants to book transport and the driver. It also saves time and energy. Because all user has to click one button and ride will be booked. Normally people go physically outside and search the transport. This takes a lot of time and energy. User who have not done shifting before, don't know about the procedure that where to go and how to search for transport, it's price and based on weight, which type of transport will be used they are entertained through this application. By providing some information about luggage they get the transport. Language barrier, it plays an important role. As we know, the literacy rate is very low in Pakistan. Mostly drivers are not educated that’s why they feel uncomfortable using applications. As mostly applications are in English. They can’t read or understand the language, it’s procedure so they are hesitant and avoid to use it. This application is made by considering people understanding and comfort so that they can use it properly. As there is no surety or confirmation while booking transport verbally after talking to driver. That’s why it’s made sure in this application. The booking details are saved continuously in order to avoid such chaos. Existing system supports unreliable booking procedure. As cancellation of booking verbally creates an uncomfortable environment between user and driver. They normally get into fight or use abusive language in the end. Using this application user and driver get rid of all this. This application provides them comfortable and reliable environment. Even if they want to cancel a ride it doesn’t create a big chaos. Because their identities are hidden as they are not talking face to face. They are communicating through application, so both users and drivers can cancel the ride on time. Otherwise the pay the penalty.

That’s why it’s not comfortable for user to do booking with existing system. They have to go out of their comfort zone in order to do booking and face a lot of circumstances but using this application it’s just about few clicks and booking is done.

1.6 Modules

A module is a separate unit of any software or hardware who can manage, control and use the system. Modules of the proposed project are:

- Admin
- Driver
- User

1.7 Existing System

Here are some of the existing systems that works in Pakistan.

- **Truck ADA**

  Truck ADA is a platform in Pakistan for goods transport where fleet operators and shippers can connect

  Website: https://truckada.pk/

- **Truck Load**

  It is a cargo service in Karachi and Lahore that besides household also transports cars.

  Website: https://www.truckload.pk/

1.8 Advantages of Online System over Physical System

By using this application driver and user are free from bargaining issue. They don’t have to fight or argue for reducing the money. In this system, there is fixed price generated by system (calculated by considering distance) and user have to pay it.

User don’t have to go physically in order to search for transport. Transport which are available at the time when users are requesting for booking will be shown. Users who don’t know how to do shifting (unaware of the procedure that where to go for searching the transport, price and based on weight how many trucks will be needed).

Driver’s willingness to use application is done by making interface in a form that they can understand.

Google map tracks the transport location where it is at current time. This helps the user to know about where transport is right now. And how much time it will take to arrive.

The main advantage of using this application is that user as well as driver’s time is saved (as time is very important for each and every one). Nobody wants to waste their time in these activities.

It’s a reliable system because this system is application based which handles many things, most importantly saving of time. It provides such an environment to user to get services without going out of their comfort zone.

1.9 Limitations of Proposed System

It is easier way to Book luggage transport online but there is a chance that the users’ data may be misused by someone. This application doesn’t work without internet connection.

1.10 Synopsis of Thesis

In the first Chapter, I have discussed about my Project, existing system and advantages of proposed system. In the second chapter, I have described Requirements and system analysis of my project that includes Use Case Model, Use Case Brief and Description Detail, Class and sequence Diagram. In the third chapter, I have discussed about system analysis and Design. In Chapter 4, I have talked about system implementation, tools and technologies. In the fifth Chapter, I have talked about system testing and in sixth chapter I have discussed about future works and references.

CHAPTER 2

Requirements and System Analysis

2.1 Introduction

A requirement is a specification or want that must be met or satisfied within a certain time frame. The requirements for a system are the descriptions of the services that a system should provide; requirements reflect the needs of customers for a system that serves a certain purpose.

Requirements analysis is also called requirements engineering. It is the process of determining user expectations for a new or modified product. The purpose of requirements analysis is to setup common understanding among project stakeholders, the output of analysis is a requirement document, and the document can be business requirement specification, technical requirement specification, user stories, some screen shots drawings, or any other documents. Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized can satisfy customer requirements. Effective requirements analysis encompasses four types of activities.

2.1.1 Requirements Elicitation

Before requirements can be analyzed, modeled, or specified they must be gathered through an elicitation process. Requirements elicitation is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders. The practice is also sometimes referred to as "requirement gathering". Requirements elicitation involves meeting with stakeholders of different kinds to discover information about the proposed system.

2.1.2 Requirements Analysis

The requirements analysis we determine the needs or conditions to meet the new or altered product or project. The requirements should be documented, actionable, measurable, testable, related to identified needs or opportunities.

2.1.3 Requirements Specifications

Requirements specification is the process of writing down the user and system requirements in a requirements document. The user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent. The user requirements for a system should describe the functional and nonfunctional requirements so that they are understandable by system users who do not have detailed technical knowledge.

2.1.4 Requirements Validations

Requirements validation is the process of checking that requirements define the system that the customer really wants. It overlaps with elicitation and analysis, as it is concerned with finding problems with the requirements. During the requirements validation process, different types of checks should be carried out on the requirements in the requirements document. These checks include

- Validity checks
- Consistency checks
- Completeness checks
- Realism check
- Verifiability
- Realism check

2.2 Functional Requirements

Functional requirements are features or functions that developer must implement to enable the users to accomplish their tasks. Functional requirements describe what a system should do. Functional requirement describes the main functionality of the system. These requirements are requested by the customer to the developer. The presence of functional requirements is very important for the system because if any one of the functional requirements is missing in the system, our system will be incomplete. Functional requirements deal with what the system should do or provide for users.

They include description of the requirement functions, outlines of associated reports, online queries or detail of data to be held in the system. Some main functional requirements of our system:

- **Register:** Both driver and user will register
- **Enter Information:** This function will allow the user and driver to enter data.
- **Validation of Information**: This function will validate data.
- **Login:** User and driver from will be logged in using this function.
- **Booking:** User will request for booking.
- **Accept Booking:** Driver will accept booking according to his transport luggage capacity.
- **Confirmation of Booking:** Driver will confirm the booking if he is comfortable with timings and location.
- **Cancellation of Booking (by both user and driver):** Both driver and user can cancel the booking any time if they have any issue.
- **System Response (about time, price):** At the end of ride system will response in terms of price that will be according to the time, weight of luggage and distance.

2.3 Non-Functional Requirements

Non-functional requirements elaborate performance characteristics of a system. They are also known as quality attributes

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes of a system. Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. Non-functional requirements essentially specify how the system should behave and that it is a constraint upon the system's behavior. They specify benchmarks that judge the operation of a system, rather than specific behaviors. Every prerequisite must be objective; there must be some quantifiable approach to evaluate whether the necessity has been met. Following are some non-functional requirements

2.3.1 Project Cost

Project can be divided into two ways and calculated as follows;

- Software Cost
- Hardware Cost

2.3.1.1 Software Cost

Software cost includes the cost of the required software's for our project. We did not have to spend money in getting the necessary software for our project. The software we used for our system is the free edition version and thus no money was put in it.

2.3.1.2 Hardware Cost

Hardware cost for our project can be considered as a moderate amount of money spent. It does not fall under a cheap project neither it is a relatively smaller one. However, having said that, the cost of the hardware components implemented does amount to significant figures. We needed smart phone and a personal computer in order to control the whole system. Other hardware expenses are not as significant when compared to it, but they do accumulate to considerable amount. But taking into consideration that this is a one-time investment, the cost cannot be said to be too expensive.

2.3.2 Portability

The application is android based so it will run on an android phone which is itself a portable device so user can use this application on the android platform which is Lollipop (21API) or higher.

2.3.3 Reliability

The application should be reliable and does not counter any failure. The application should never crash or hang, other than as the result of an operating system error.

2.3.4 Availability

The application will be available to every user until he is logged in to the application and has internet connection. Without internet connection user can't use the application.

2.3.5 Security

As described earlier, this is the android based mobile application. Therefore, it is a sensitive application from the security's point of view. So, in order to make the user experience better we will store the data of this application on the Android device in private mode.

2.3.6 Maintain ability

During the development period, all the things will be properly documented so that we can easily make changes and upgrade our application.

2.4 Interface Requirements

It specifies hardware, software, or database elements with which a system or component must interface.

2.4.1 User Interfaces

The background will be light in color with dark colored font to enhance the contrast and visibility. Controls which allow the user to interact with the application will be clear and imply their functionality within the application. Icons and fonts will be kept simple to make users understand the function of feature easily.

2.4.2 Hardware Interfaces

1. Touch keyboard is used for input and screen is to display output.
2. Since it is application based system, so internet connection is must.
3. An android phone having 512MB RAM and 1.3 GHZ processor will be compatible with this application.

Note: These are the minimum requirements for this application.

2.4.3 Software Interfaces

Android platform will be used for the development of this application. Android phone users from Jellybean (4.1) to Nougat (7.0) can easily install and use this application.

2.4.4 Communication Interfaces

As the system is internet based, therefore it will require some standard networking protocols for communication. These protocols are usually installed automatically by the operating system running on server.

- HTTP: It is a protocol used by the WWW service to make communication
- TCP/IP: it is a protocol used to communicate data all around the internet

2.4.5 Database Requirements

The database has 10 entities. Therefore, for each entity table is created to store data as shown in figure below:

- Admin Table is used to store admin name and password
- User-Information table is used to store Name, CNIC and Mobile Number
- User table is used to store User-Id, Name and Password
- Driver-Information table is used to store Name, CNIC and Mobile Number.
- Driver table is used to store Driver-Id, Name and Password.
- Car table is used to store Car-Name, Car-Plate-Number and Car-Model.

- Car-Information table is used to store Car-Plate-Number, Owner and Certificate of Registration.
- Booking-request table is used to store User-Id, Driver-Id and Luggage-Items.
- Luggage-Items table is used to store Item-Id, Item-Name and Item-Quantity.
- Location table is used to store User-Current-Location, User-Destination-Location and Driver Location.

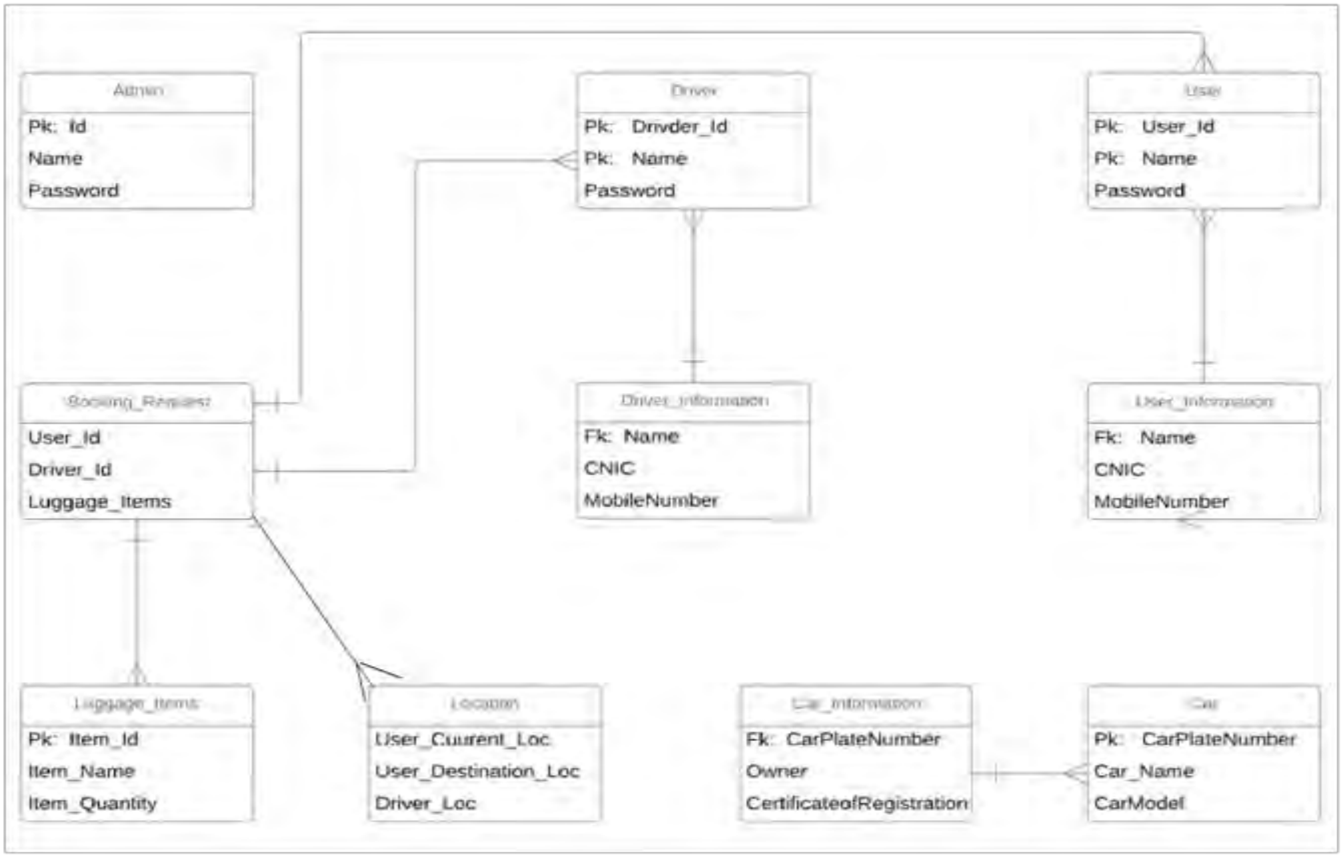


Fig 2.1: Entity Relationship Diagram

2.5 Summary

In this chapter we analyzed and specified requirements of our project. Our requirements are of three types i.e. functional requirements, non-functional requirements and interface requirements. Functional requirements describe what a system's functionality will be. We have specified 6 basic functionalities (functional requirements) which our project must provide. Non-functional requirements are the quality requirements of system i.e. system should be efficient, accurate and maintained. And last but not the least are interface requirements i.e. what hardware and software resources our system needs.

CHAPTER 3

System Analysis and Design

3.1 Introduction

System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysts solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design technique. Systems development is systematic process deal with understanding and specifying in detail what system should do and how the components of the system work together. It includes planning, analysis, design, deployment, and maintenance. Here, I will primarily focus on

- Analysis
- Design

3.1.1 Analysis

Analysis is the process of breaking a complex topic or substance into smaller parts to gain a better understanding of it. Analysis specifies what the system should do.

It is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique which is used to improve the system and ensures that all the components of the system work efficiently to accomplish their purpose.

The key activities included in analysis are:

- Conduct the risk assessment.
- Analyze security requirements
- Perform functional and security testing
- Analyze system's workflow
- Prepare initial documents for system implementation

Although this section presents the system components in a sequential top-down manner, the order of completion is not necessarily fixed. Analysis of complex systems need to be iterated until consistency and completeness is achieved.

3.1.2 Design

A design is a plan or specification for the construction of an object or system. Or specification in the form of a prototype, product, or process. System Design focuses on how to accomplish the objective of the system. It is used to create a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. During this phase the system architecture is designed. It specifies that how the functionality of system will be implemented and how different parts of system will communicate and coordinate with each other.

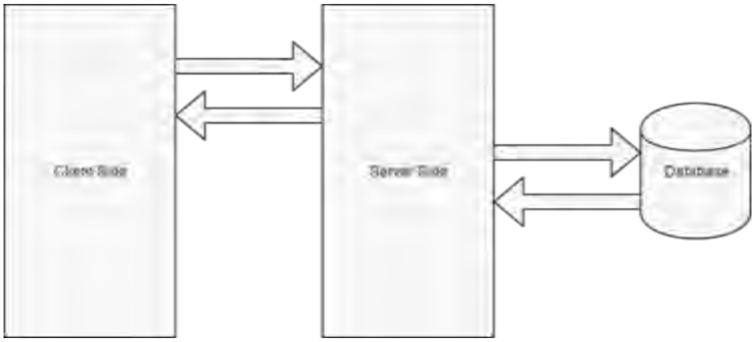


Fig 3.1: System Design

3.2 Component Overview

It is an online site for people where they can easily get access to book ride for luggage transportation to save their time and money. The main functionality of the system is described in following UML diagrams.

3.3 UML Diagram

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts, or classes, to better Understand, alter, maintain, or document information about the system. There are two types of UML diagrams

- Behavioral Diagrams
- Structural Diagrams

3.3.1 Behavioral Diagrams

It shows how the system behaves and interacts with itself and other entities (users, other systems). They show how data moves through the system and how objects communicate with each other.

3.3.1.1 Use Case Diagram

Use cases are a set of actions, services, and functions that the system needs to perform. Use case diagrams model the functionality of a system using actors and use cases. A "system" is something being developed or operated, and –actors" are people or entities operating under defined roles within the system. Use case diagrams are usually referred to as behavior diagrams used to describe a set of use cases that some system or systems should or can perform in collaboration with one or more external users of the system. Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Use case diagrams are used to specify:

- External needs on a subject, mandatory usages of a system.
- The functionality offered by a subject – what the system can do.
- The requirements, which the specified subject poses on its environment.

3.3.1.2 Use case Diagram Notations

System: It is a rectangular shape system boundary contains use cases in it and actors are place outside the system.

Use Case: It is an oval shape which represents the functions of the system.

Actor: Actors are the users of a system. They are represented with the actor stereotype.

Relationship: A relationship between two use cases is basically a dependency between the two use cases. The types of relationships are: Include, Extend, Generalization etc.

3.3.1.3 Use case diagram

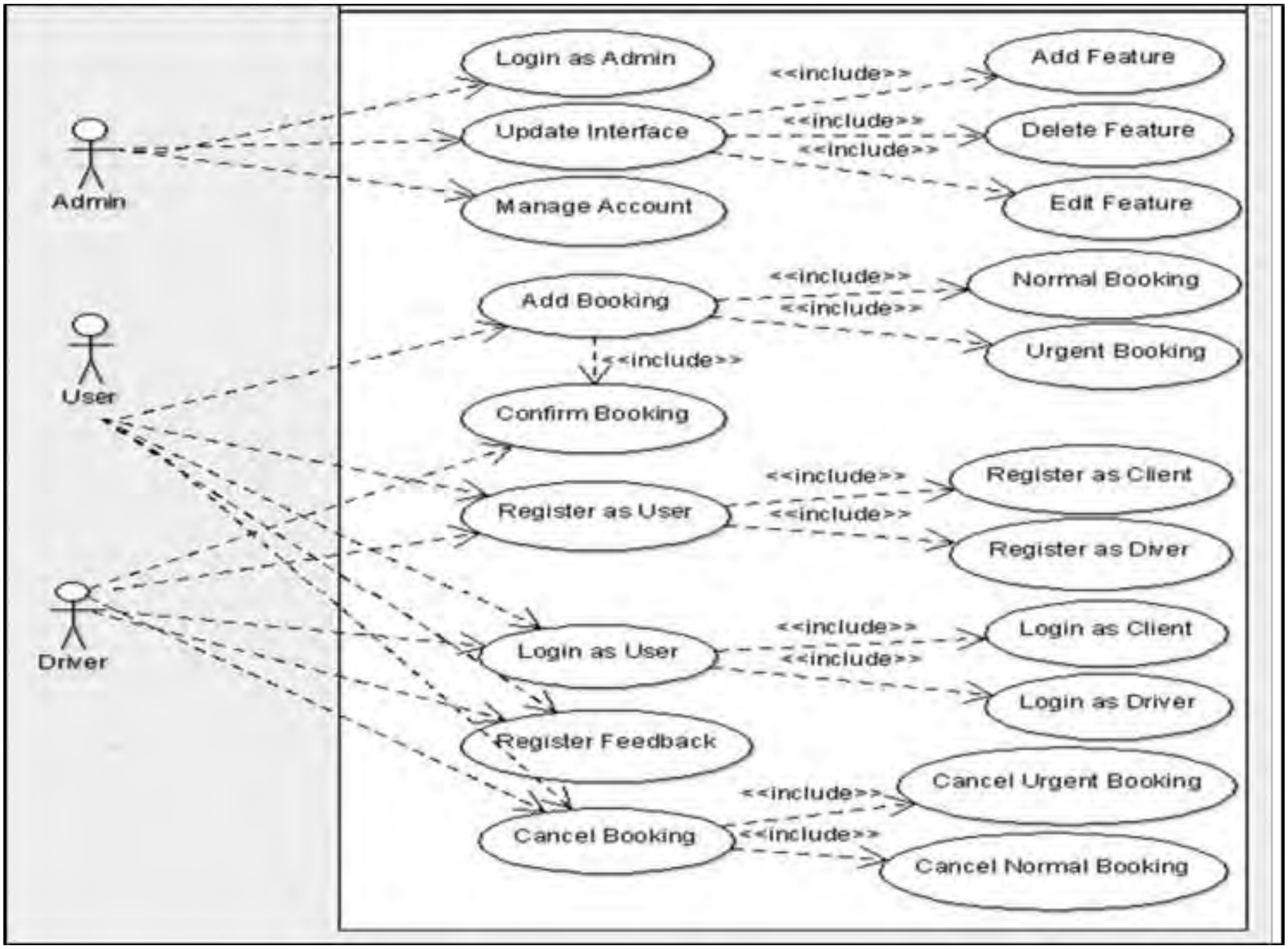


Fig 3.2: Use case diagram

3.3.2 Use Case Description

3.3.2.1 Login as Admin

Admin will perform functionality as shown in table 3.1 in order to Login into system as admin.

Table 3.1: Login as Admin

UC-1: Login as Admin	
Primary actor	Admin
Pre-condition	Admin opened the application and login screen appeared.
Post-condition	Admin has logged in successfully.
Main Success Scenario	1.System ask for Enter Admin's username and password. 2.Admin enters username and its password. 3.System directed admin to the admin account.
Alternate Scenario	4 a) Admin enter incorrect information. or b) Admin submit clicked submit button without entering anything in username and password fields. System shows error message ―Your username and password is invalid. Enter again."
Frequency	Each time admin wants to open the application.

3.3.2.2 Update Interface

Admin will perform functionality as shown in table 3.2 in order to update interface.

Table 3.2: Update Interface

UC-2: Update Interface	
Primary actor	Admin
Pre-condition	Admin opened the application.
Post-condition	Interface updated successfully.
Main Success Scenario	1.Admin select one of the options: Include (Add Interface), Include (Edit Interface), Include (Delete Interface). 2.System go to the option selected.
Alternate Scenario	1 a) No option selected by the admin. System shows error message ―Select any option."
Frequency	Each time admin wants to update the application's interface.

3.3.2.3 Manage Account

Admin will perform functionality as shown in table 3.3 in order to manage accounts.

Table 3.3: Manage accounts

UC-3: Manage Accounts	
Primary actor	Admin
Pre-condition	Admin have access to accounts of users and drivers.
Post-condition	Account/accounts are deleted.
Main Success Scenario	1. Admin will see the list of drivers and users account. and check the complaints and feedback. 2. According to responses admin will delete user and driver account respectively. 3. Admin will delete user account which has not used this application in the time period of 5 years and driver account if he has not accepted any booking for 1 year. 4. Updated information is saved in database.
Alternate Scenario	6a) Error:" System is not connected with database." System asks the user for reconnecting with database.
Frequency	Once in 6 months, admin will perform this.

3.3.2.4 Login as User

User will perform functionality as shown in table 3.4 in order to login himself.

Table 3.4: Login as User

UC-4: Login as User	
Primary actor	User
Pre-condition	User must have registered himself/herself on this application.
Post-condition	User will be logged into the application.
Main Success Scenario	1.User select one of the options (Include (Login as Client), Include (Login as Driver)). 2. System go to the option selected.
Alternative Scenario	1a) User didn't select any option. System give error message —You haven't selected any option. Try again".
Frequency	Each time user wants to login.

3.3.2.5 Add Booking

User will perform functionality as shown in table 3.5 in order to do booking.

Table 3.5: Add Booking

UC-5: Add Booking	
Primary actor	User
Pre-condition	User must be logged in and screen appeared to add booking.
Post-condition	Request for booking will be send to system.
Main Success Scenario	User select one of the options (Include (Normal Booking), Include (Urgent Booking)). Selection is according to time. System go to the option selected.
Alternate Scenario	1a) User didn't select any option. System give error message ―You haven't selected any option. Try again".
Frequency	Each time user wants to request for booking.

3.3.2.6 Confirm Booking

Driver will perform functionality as shown in table 3.6 in order to do confirm booking.

Table 3.6: Confirm Booking

UC-6: Confirm Booking	
Primary actor	User
Pre-condition	User must be logged in and screen appeared to add booking.
Post-condition	Driver must receive booking request of user.
Main Success Scenario	1. Driver will check the requests for booking made by users. 2. According to his transport luggage capacity driver will accept the request and select the confirm booking button (if he agrees to conditions according suitability, availability and transport size). If user's luggage is more than what transport can carry driver will not accept the booking. 3. The confirmation message is send to system. Updated information is saved in database.
Alternate Scenario	1a) User didn't select any option. System give error message ―You haven't selected any option. Try again".
Frequency	Each time user wants to request for booking.

3.3.2.7 Cancel Booking

User will perform functionality as in table 3.7 to cancel booking.

Table 3.7: Cancel Booking

UC-7: Cancel Booking	
Primary actor	User
Pre-condition	User must request for a booking and screen appeared to cancel the booking.
Post-condition	Request will be cancelled and a message will be send to client and driver.
Main Success Scenario	User select one of the options (Include (Cancel Normal Booking), Include (Cancel Urgent Booking)). System go to the option selected.
Alternative Scenario	1a) Error: ―User didn't select any option." System ask user to select option.
Frequency	Each time user wants to cancel the request for booking.

3.3.2.8 Register Feedback

User will perform functionality as shown in table 3.8 to register feedback.

Table 3.8: Register Feedback

UC-8: Register Feedback	
Primary actor	User
Pre-condition	User must be logged in.
Post-condition	Feedback will be send to system.
Main Success Scenario	Screen appears so that user can give feedback. User(Client/Driver) enter the feedback or give rating and press the SUBMIT button. Information is stored in database.
Alternative Scenario	3a) Error:" System is not connected with database." System asks the user for reconnecting with database.
Frequency	Each time user wants to give response.

3.3.3 Sequence Diagram

A sequence diagram is a kind of interaction diagram that shows how processes operate with one other and in what order. It is a construct of a message Sequence Chart. A sequence diagram shows object interaction arrange in time sequence.

Sequence Diagram Notation:

- **Actor:** It is a type of role played by an entity that interacts with the subject.
- **Lifeline:** A lifeline represents an individual participant in the Interaction.
- **Activations**: A thin rectangle on a lifeline) represents the period during which an element is performing an operation.
- **Call Message:** A message defines a particular communication between Lifelines of an Interaction.

3.3.3.1. Sequence Diagram between Admin and System

As shown in figure 3.3 below, admin will open the system. He manages the accounts of both user and driver based on their usage, complaints and feedback. If they are not using application in time period of 5 years' admin will delete their accounts. He will verify via information entered by user and driver while registering is valid or not. He will make interfaces. He can update interfaces according to user and driver's complaints and feedbacks. Similarly, he can delete interfaces if he finds them irrelevant.

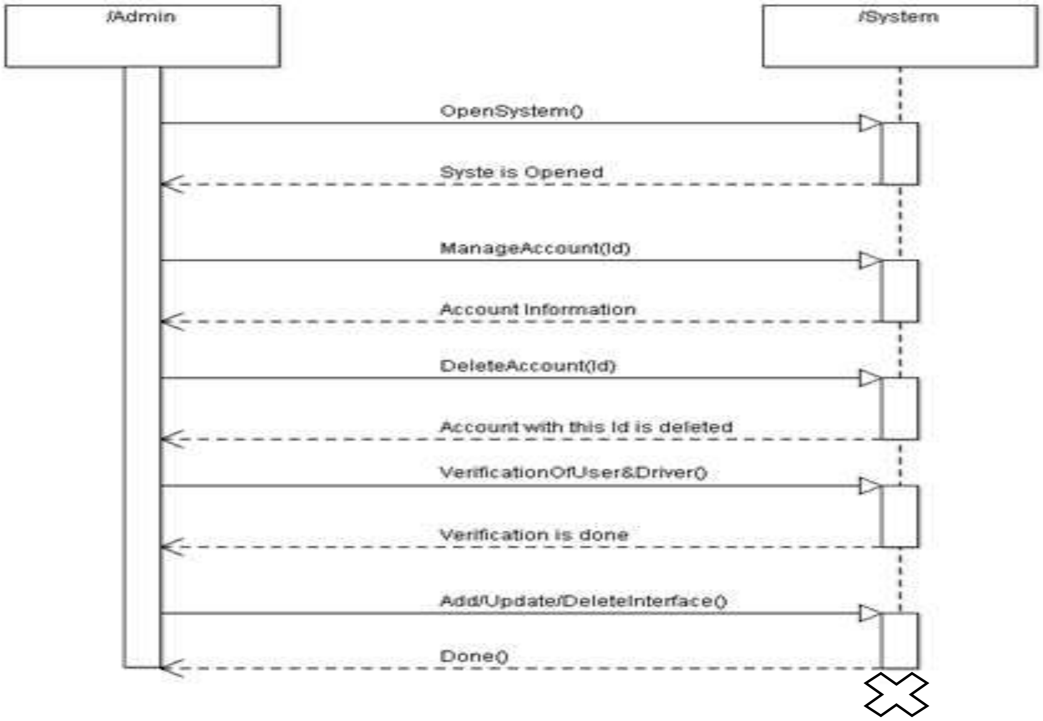


Fig 3.3: Sequence diagram between Admin and System

3.3.3.2. Sequence Diagram between Driver and System

As shown below in figure 3.4, driver will first register himself on the application in order to use the service. He will login with the name password given by system. After this he will be available and get the requests from the system (if there are request made by user). He will accept the request according to its transport luggage capacity and time availability. He can cancel the booking within 10 minutes. Otherwise he has to pay penalty. He can add complaints and feedback.

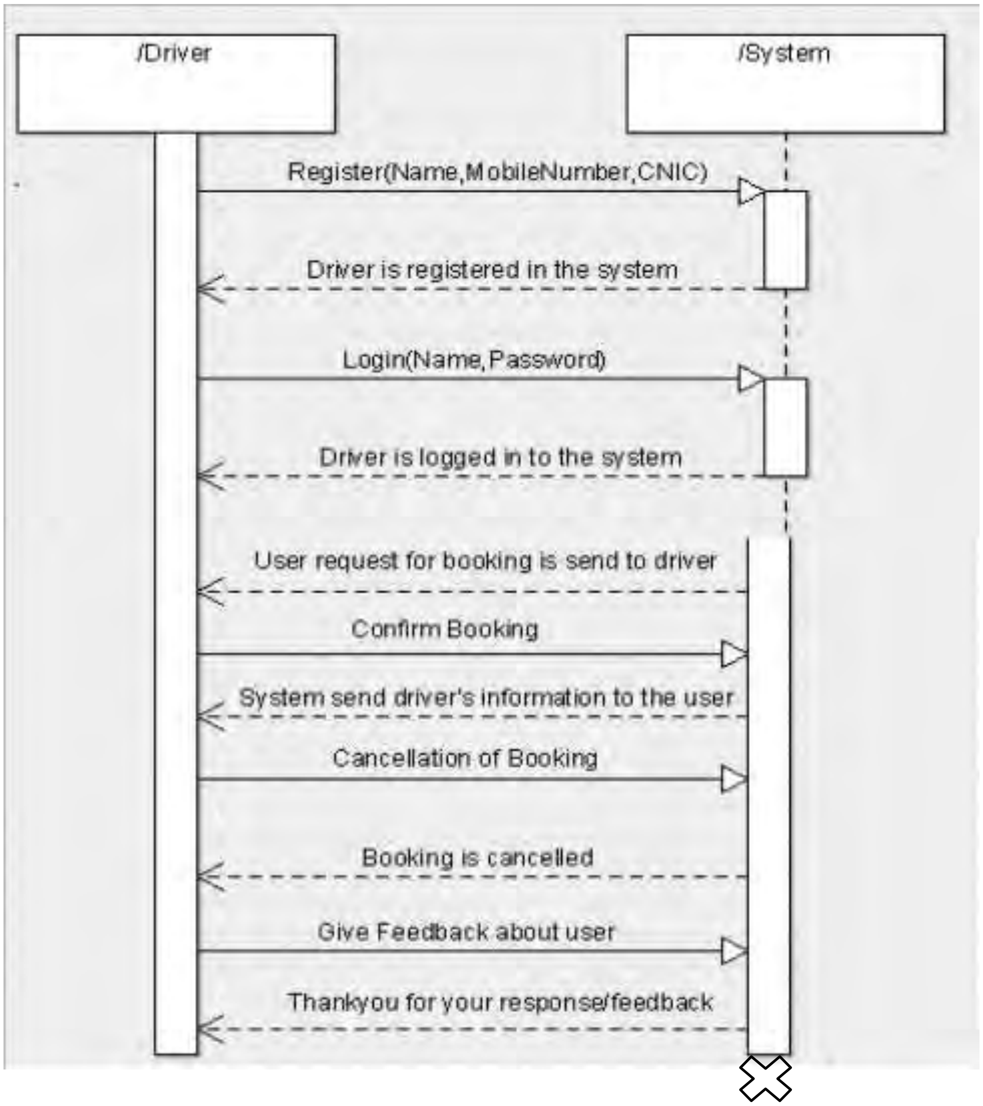

/Driver
/System
Register(Name,MobileNumber,CNIC)
Driver is registered in the system
Login(Name,Password)
Driver is logged in to the system
User request for booking is send to driver
Confirm Booking
System send driver's information to the user
Cancellation of Booking
Booking is cancelled
Give Feedback about user
Thankyou for your response/feedback

Fig 3.4: Sequence diagram between Driver and System

3.3.3.3 Sequence Diagram between User and System

As shown below in figure 3.5, user will register himself/herself on this application. Then user will be able to make booking request. For this, he needs to provide his current location and desired location along with luggage information.

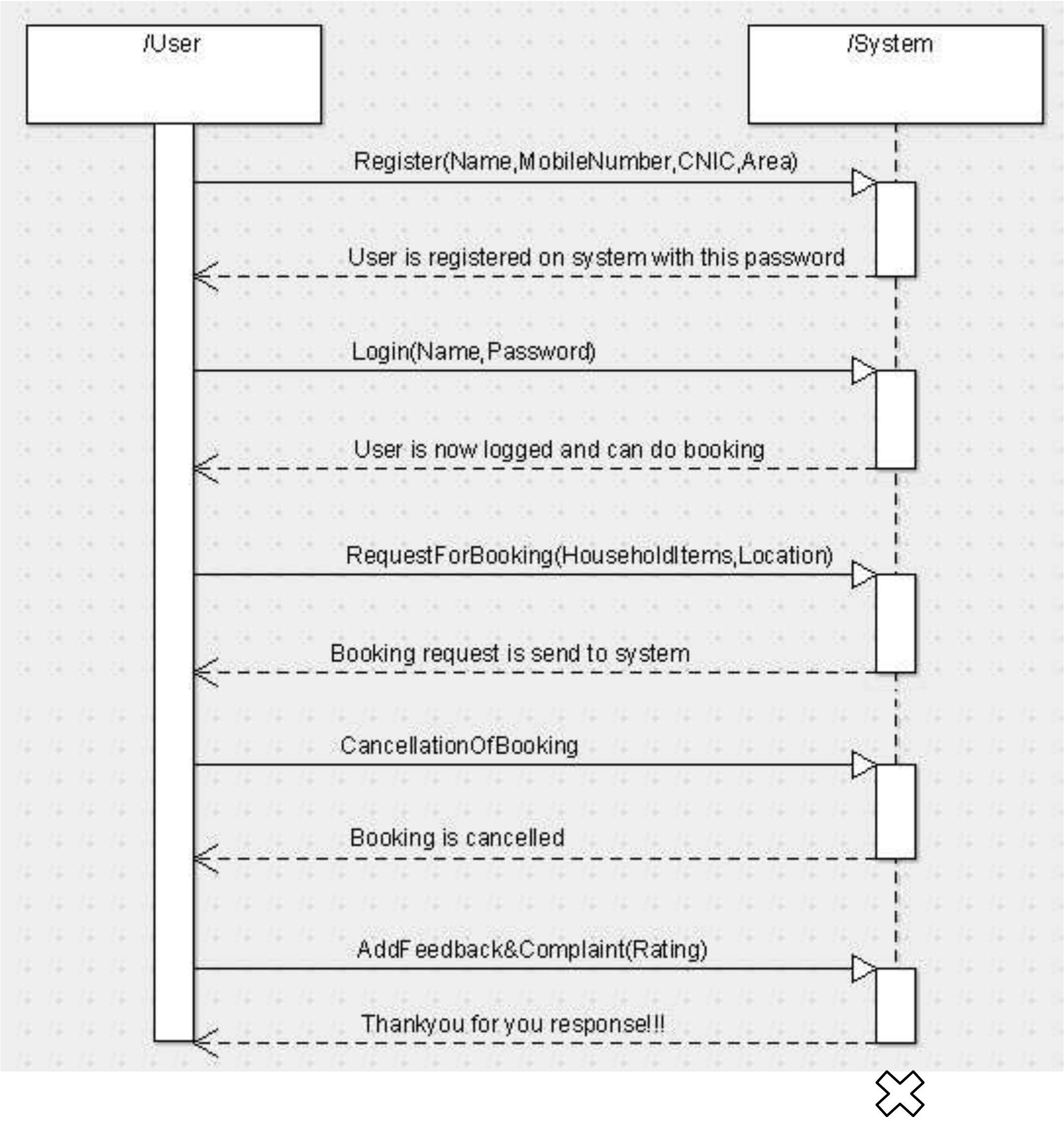

/User
/System
Register(Name,MobileNumber,CNIC,Area)
User is registered on system with this password
Login(Name,Password)
User is now logged and can do booking
RequestForBooking(HouseholdItems,Location)
Booking request is send to system
CancellationOfBooking
Booking is cancelled
AddFeedback&Complaint(Rating)
Thankyou for you response!!!

Fig 3.5: Sequence diagram between User and System

3.3.4 System Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order a group of objects works together. This diagram is used by software developers and business professionals to understand requirements for a new system or to document an existing process. It is a good way to visualize and validate various runtime scenarios.

These can help to predict how a system will behave. In our application there are two panels. One is driver panel and other is user panel. It is made for providing service of luggage transportation that’s why user and driver panels are different. They both provide their personal information while registering that is kept private. User request for booking. Driver accept the request if he is available. They can add complaints and feedback according to the service provided to them as shown below in figure 3.6:

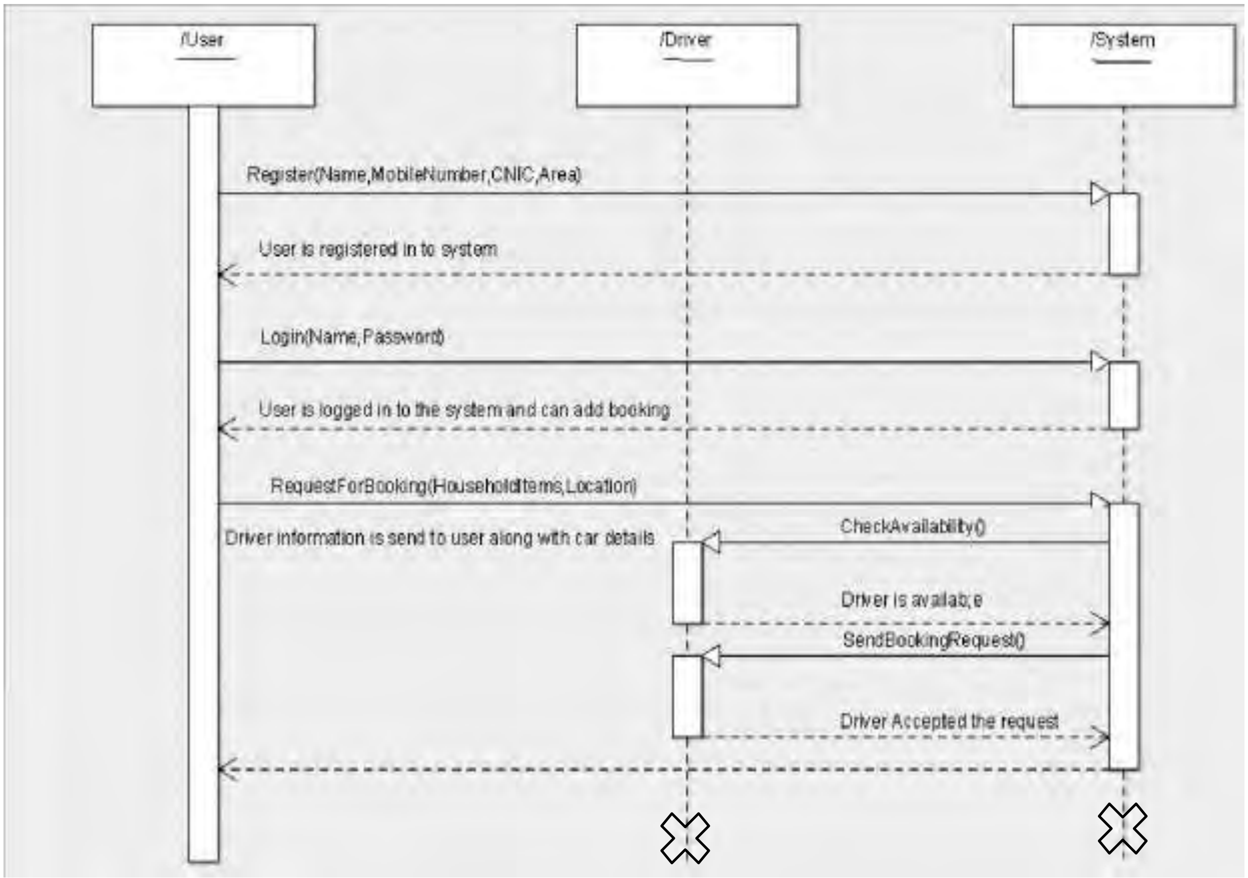

/User
/Driver
/System
Register(Name,MobileNumber,CNIC,Area)
User is registered in to system
Login(Name,Password)
User is logged in to the system and can add booking
RequestForBooking(HouseholdItems,Location)
Driver information is send to user along with car details
CheckAvailability()
Driver is availab;e
SendBookingRequest()
Driver Accepted the request

Fig 3.6: System Sequence Diagram

3.3.5 Class Diagram

A class diagram in the Unified Modeling Language is a type of static structure diagram. Class diagram models the static structure of a system. It clearly maps out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects.

Admin have the access to whole application. He can add, delete and update information. User add the booking request after registering. He provides information like location (both current and desired), luggage details (household items) and mobile number in order to be in contact with driver. Driver similarly accept the request according to his transport luggage capacity and availability. He picks up the luggage and go to the desired location given by user. User pays and provide the feedback (and complaints if any) if he is satisfied from service. Driver can also provide feedback and complaint as shown below in figure 3.7:

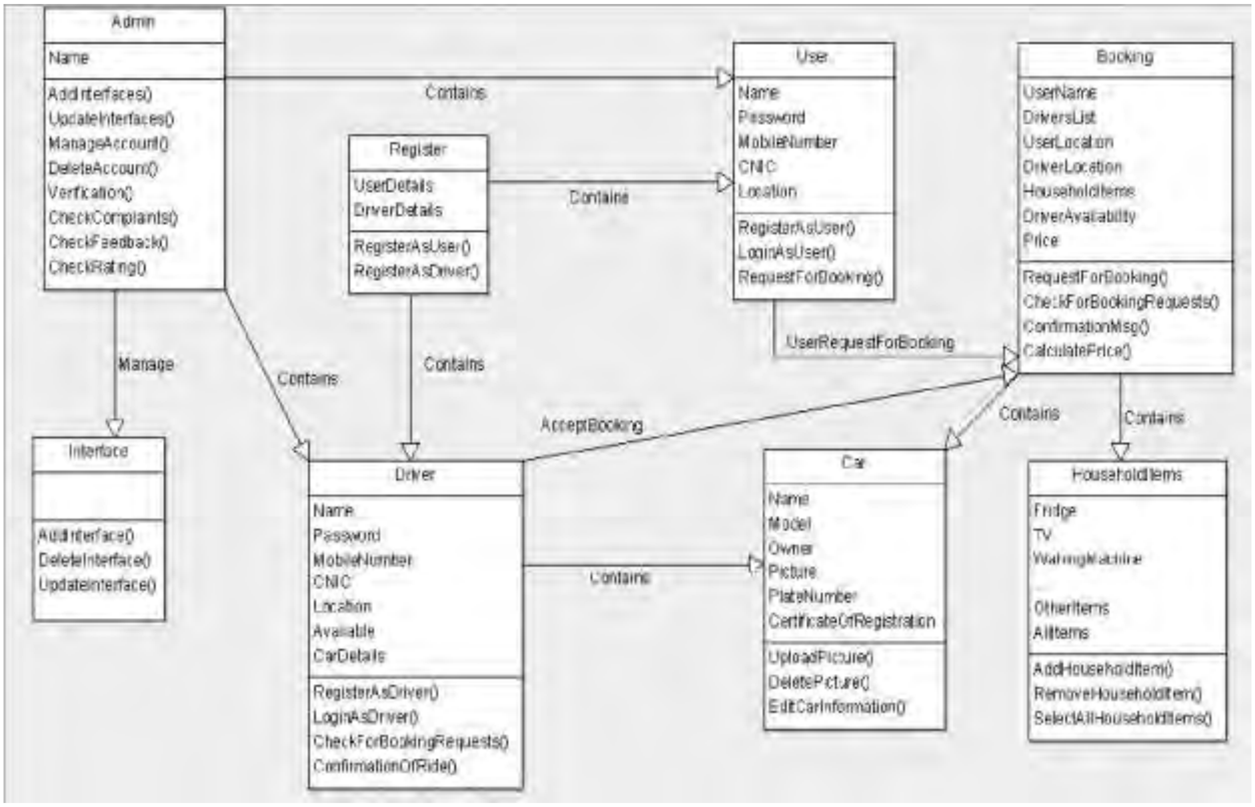


Fig 3.7: Class Diagram

3.4 Summary

System design describes that how different components of system are created and how they will interact with each other. In this chapter we have analyzed and designed our system. Different UML diagrams have been used to show relationship between different entities, and how data flows between them. Similarly, we have written different use cases to describe that how users will interact with system to perform different tasks.

CHAPTER 4

Software Description and Implementation

4.1 Introduction

System implementation is the process of defining how the information system should be built ensuring that the information system is operational and used and meets quality standard. Implementation is the process of realizing the design as a program.

4.2 Details of Implementation Tools and Technologies

The details of the languages, tools and technologies we used are following:

4.2.1. Language Selection

The project is implemented in the following language:

4.2.1.1 Java

Java is a multi-platform, object-oriented, and network-centric language. It is among the most used programming language. Java is also used as a computing platform. It is considered as one of the fast, secure, and reliable programming languages preferred by most organizations to build their projects. I have used it for coding in android studio as Java is −

- Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- High Performance − With the use of Just-In-Time compilers, Java enables high performance.

4.2.1.2 HTML 5

HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviors, and a larger set of technologies that allows the building of more diverse and powerful Web sites and applications.

Latest HTML5 technologies are:

- **Semantics:** allowing you to describe more precisely what your content is.
- **Connectivity:** allowing you to communicate with the server in new and innovative ways.

- **Offline and storage:** allowing webpages to store data on the client-side locally and operate offline more efficiently.
- **Multimedia:** making video and audio first-class citizens in the Open Web.
- **2D/3D graphics and effects:** allowing a much more diverse range of presentation options.
- **Performance and integration:** providing greater speed optimization and better usage of computer hardware.
- **Device access:** allowing for the usage of various input and output devices.
- **Styling:** letting authors write more sophisticated themes.

4.2.1.3 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

4.2.2 Tools Selection

The project is implemented in the following tool:

1) Android Studio
2) Emulator

4.2.3 Resources

The API needed is as follows:

- Google Map API used for location
- Firebase database for storing data online

4.3 User Interface Design

User interface (UI) design is the design of user interfaces for software or machines, such as the look of a mobile app, with a focus on ease of use and learn ability for the user. It is everything designed into an information device with which a human being may interact—including display screen, keyboard, mouse, light pen the appearance of desktop, illuminated characters. The user interface can include total user experience which may include the aesthetic appearance of the device, response time, and content that is presented to the user within the context of the user interface.

4.4 Interfaces for Application

As described earlier it is a mobile application so will show responsiveness on every android phone having variable screen sizes. Following are the interfaces for our application.

4.4.1 UI:1 Register Options

It has two options as described earlier it has two panels for user and driver as shown in figure 4.1 below:

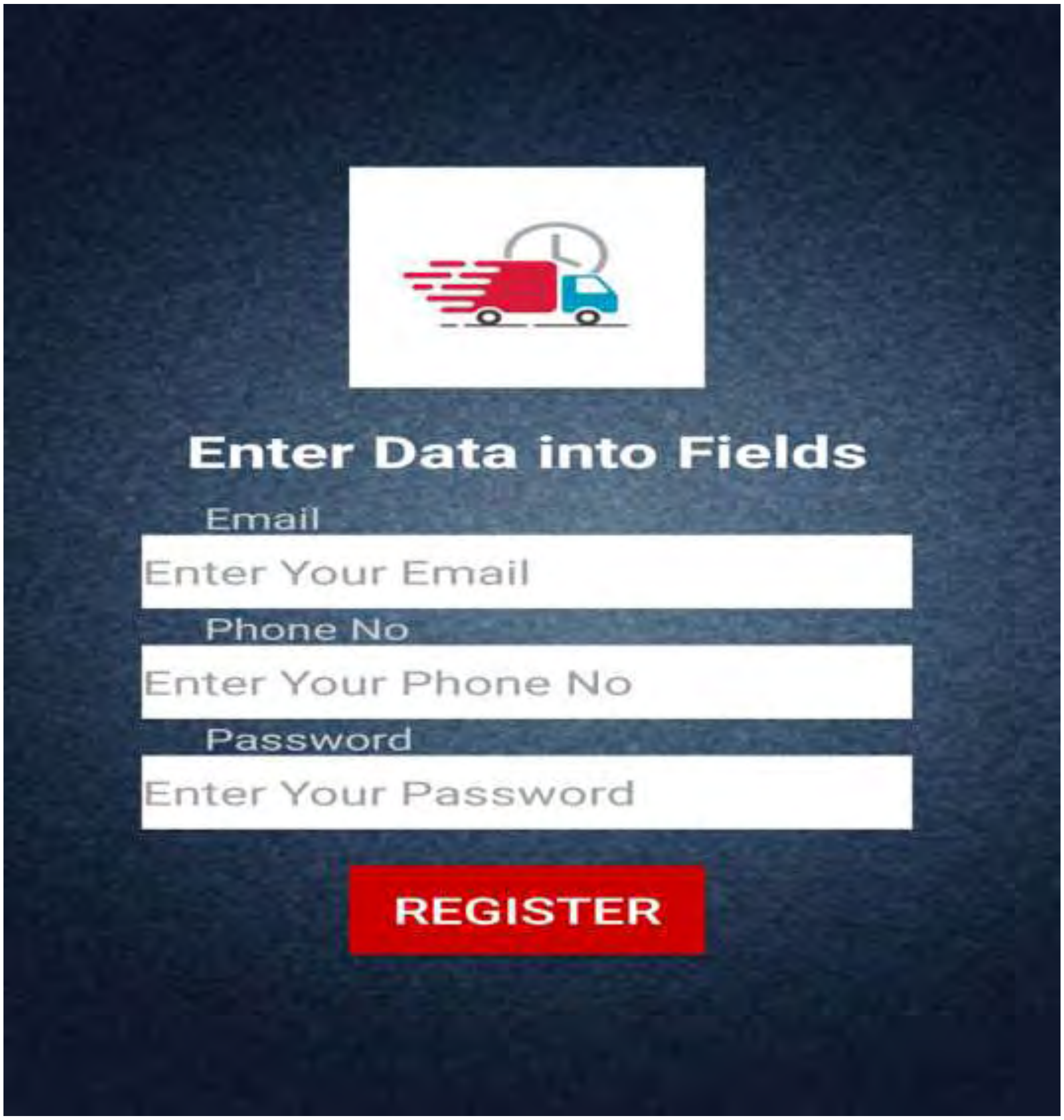


Fig 4.1: Register Options

4.4.2 UI:2 Login as User

User will add his information in these fields. They all are required (not optional).

Without entering all these input fields, he can’t register on this application as shown in figure 4.2 below.

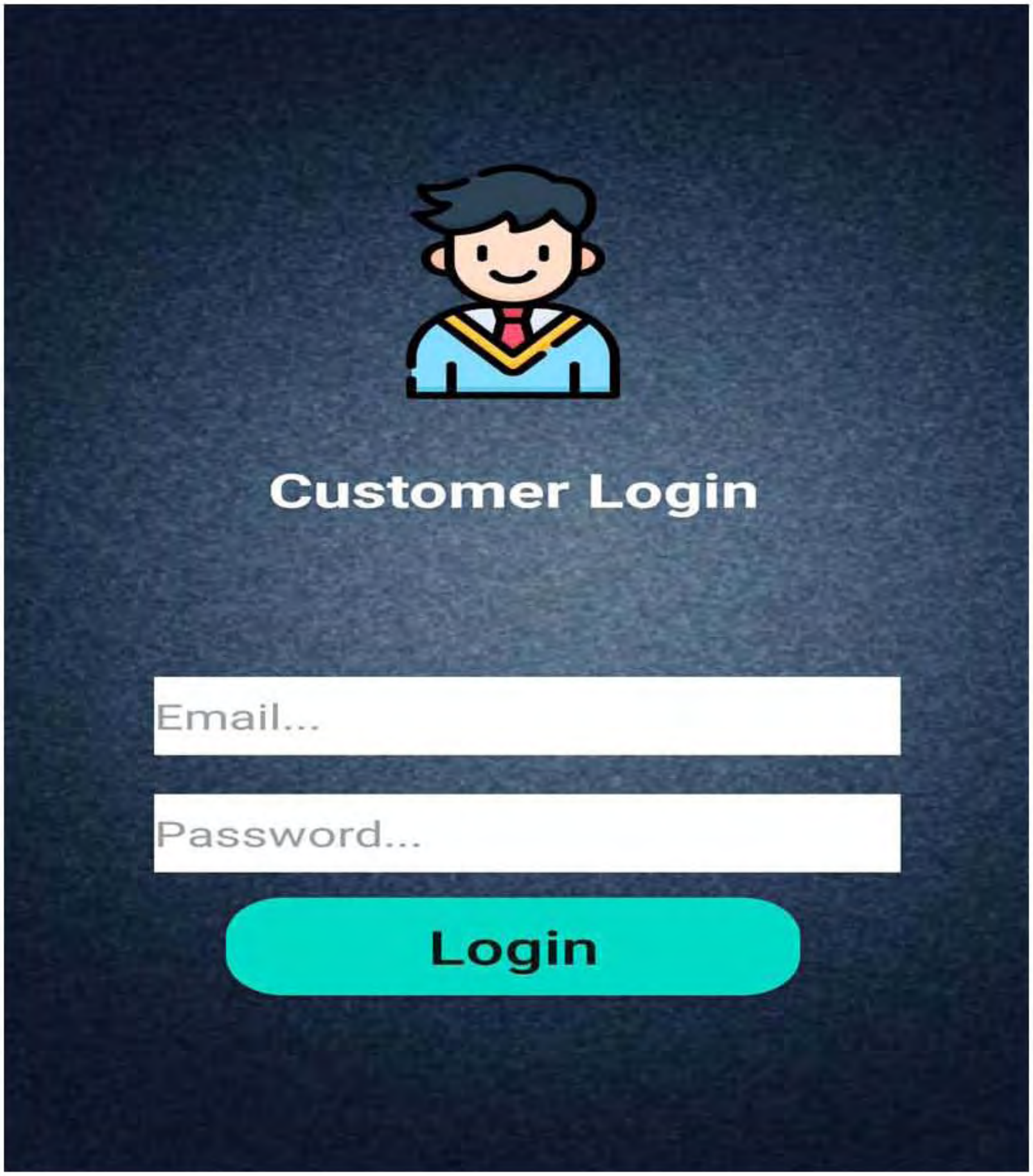


Fig 4.2: Login as User

4.4.3 UI:3 Login as Driver

Driver will add his information in these fields. They all are required (not optional).

Without entering all these input fields, he can't register on this application as shown in figure 4.3 below.

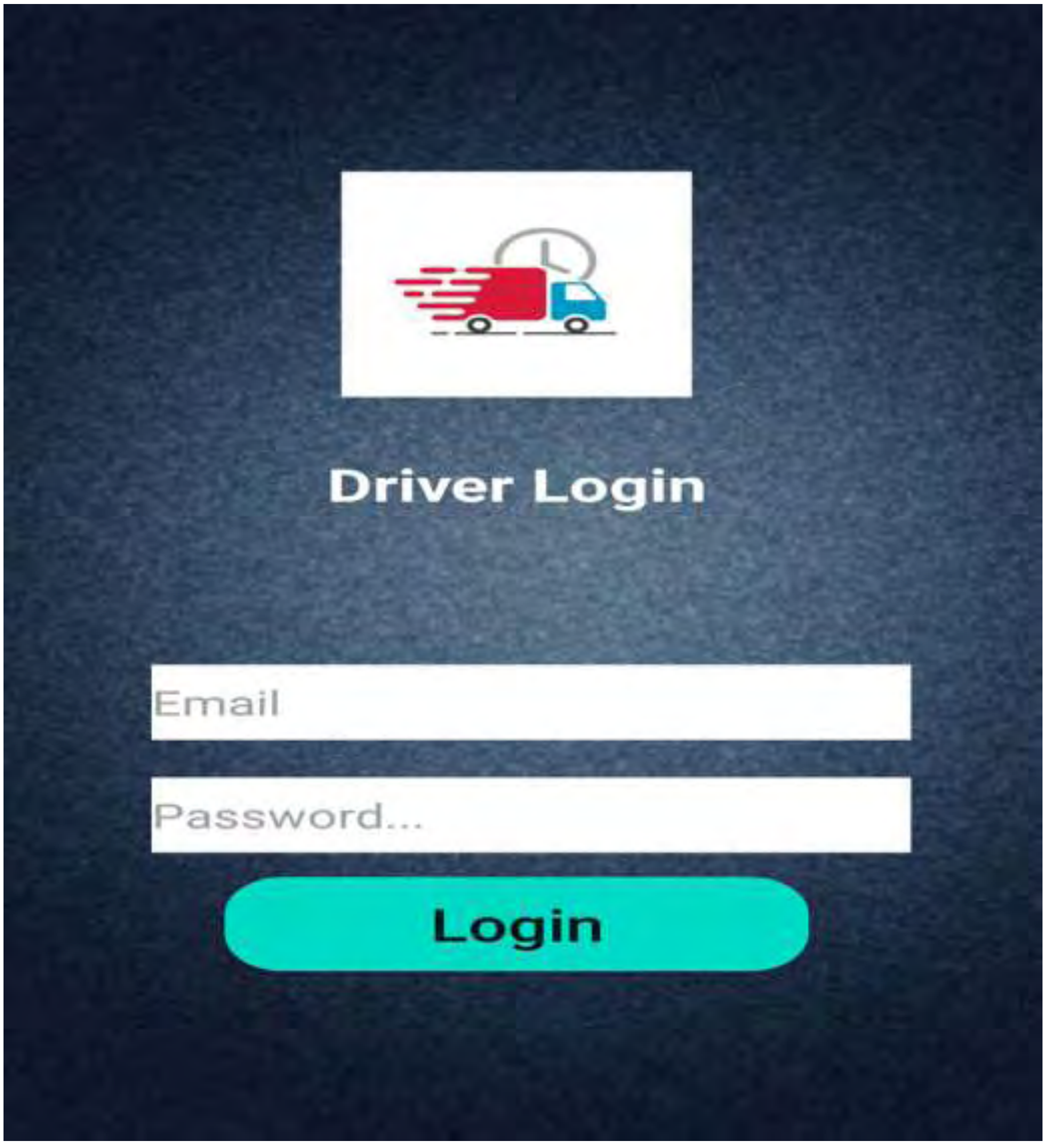


Fig 4.3: Login as Driver

4.4.4 UI:4 Add Booking

User will enter current and desired location. Add the luggage details and then click on this booking button as shown in figure 4.4 below:

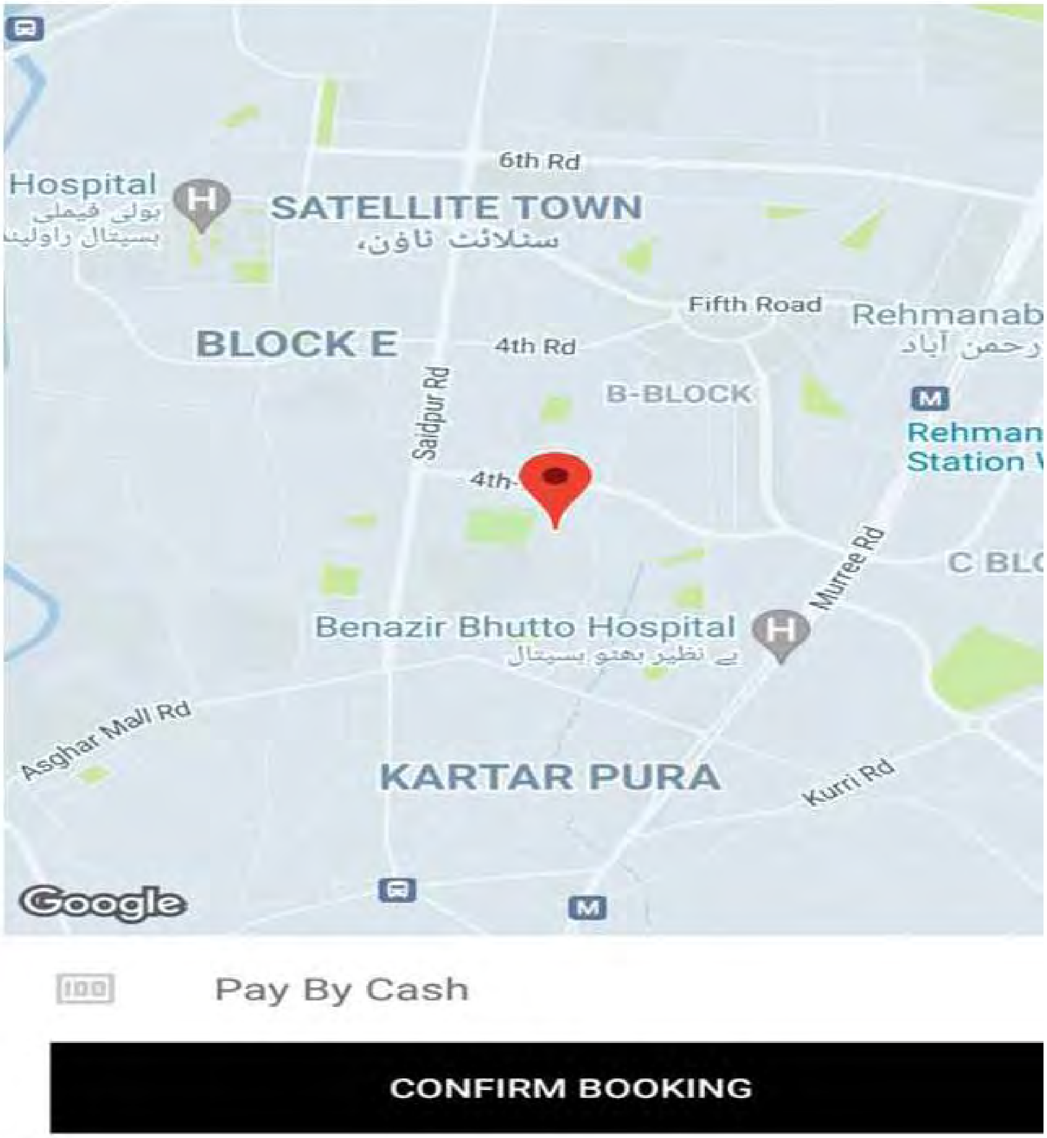
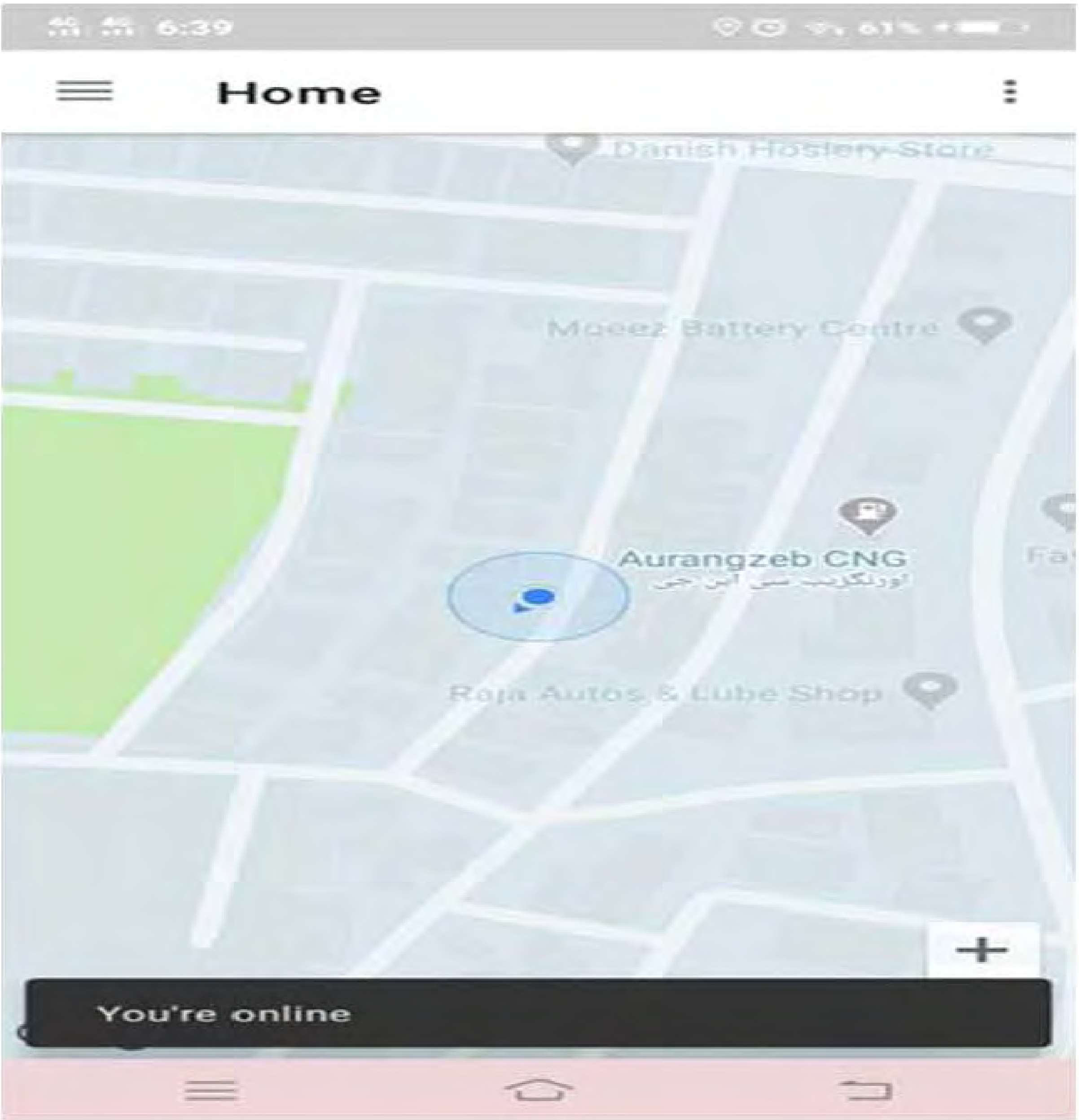


Fig 4.4: Add Booking

4.4.5 UI:5 Google Map

User will select and set the location using GPS (Global Positioning System) as shown in figure 4.5 below:



You're online

Fig 4.5: GPS

CHAPTER 5

System Testing

5.1 Introduction

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The system testing is an inescapable part of the verification and validation process for software. Testing is aimed at finding errors in the test object and giving confidence in its correct behavior by executing the test object with selected input values.

Testing is more than just debugging. The main purposes of testing are quality assurance, verification, or reliability estimation in order to find problems and purpose of finding those problems is get them fixed. It is the process of analyzing a software item to detect the difference between existing and required conditions (such that defects/error etc.) and to evaluate the features of the software item. It attempts to make things go wrong to determine if things happen when they should. It mainly includes unit and integration testing. Main goals of testing are fault identification, correction, removal and quality improvement. Testing does not lead to quality improvement unless errors are not detected and removed. Since mobile applications are used by different groups of people therefore it needs special testing techniques. It is hard to find the future number of users and there way of interacting interior design application.

5.2 Objective

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding defects which may be created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

5.3 Goals

Software testing is the mechanism of knowing that what's the expected result and what the actual result a software project or product has given. You simply say that software testing is nothing but validation and verification. Main goal of software testing is to ensure that software should always be defect free and easily maintained. Important goals of software testing:

- Always identifying the bugs as early as possible.
- Preventing the bugs in a project and product.
- Check whether the customer requirements criterion is met or not.
- And finally, main goal of testing to measure the quality of the product and project.

5.4 System Testing

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of testing where external working of the software is evaluated with the help of requirement documents an and as such, should require no knowledge of the inner design of the code or logic.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a Requirement Specification. System testing tests not only the design, but also the behavior and even the believed expectations of the customer and user. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

As a rule, system testing takes, as its input, all of the integrated software components that have passed testing. The process of testing an integrated system is to verify that it meets specified requirements. The objective of System Testing is to verify that the integrated information system as a whole is functionally complete and satisfies both functional and non-functional design requirements. Functional testing is concerned with what the system does whereas Non-Functional testing is concerned with how the system does what it does. Primary goal is the defect/error detection based on software application behavior, software design and expectation of end user.

Two important type of testing are white box testing and black box testing.

5.5 White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

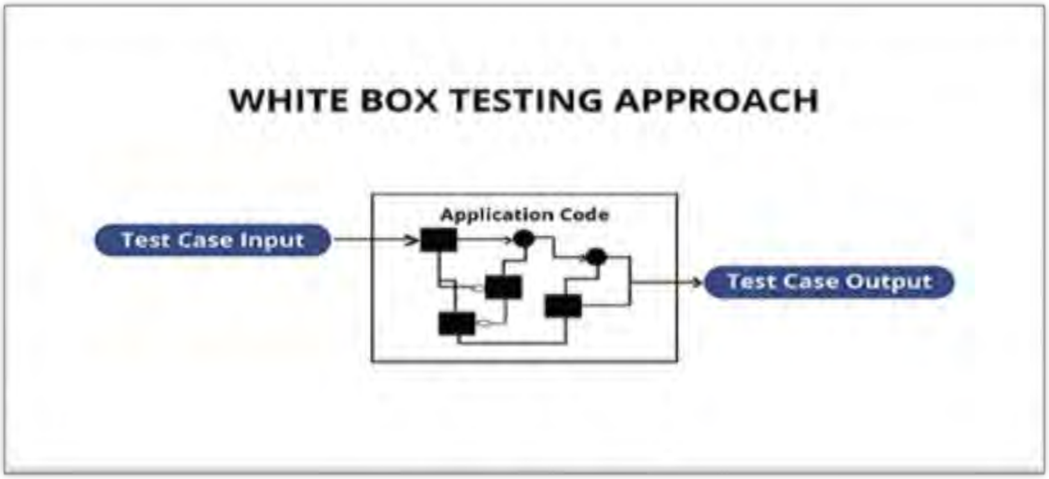


Figure 5.1: White box testing

5.6 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. The types of testing under this testing are totally depending on the testing for requirements and functionality of software application. Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

This method of test design is applicable to all levels of software testing: unit, integration, system and acceptance. The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

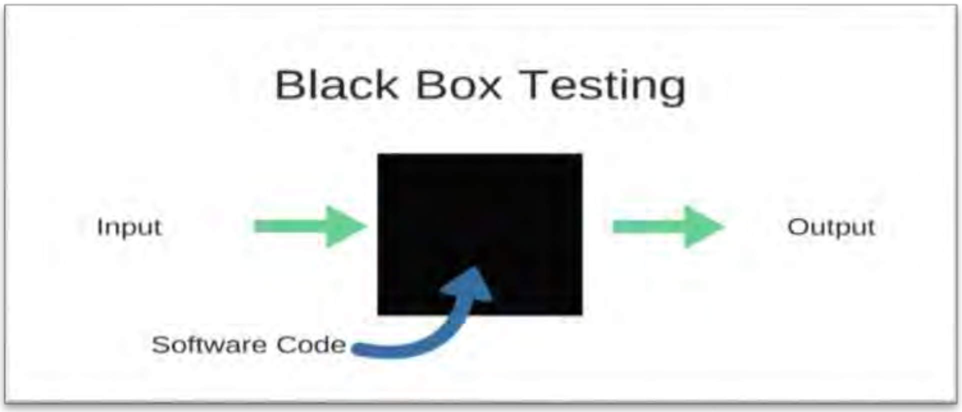


Fig 5.2: Black box testing

5.7 Test Strategy

A Test Strategy document is a high-level document and normally developed by a project manager. This document defines ―Software Testing Approach” to achieve testing objectives. Some companies include the ―Test Approach” or ―Strategy” inside the Test Plan, which is fine and it is usually the case for small projects.

5.7.1 Test Plan

A test plan is a document detailing the objectives, target market, internal beta team, and processes for a specific beta test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow.

5.7.2 Features to be tested

The following are the features to be tested:

1) Register as User
2) Register as Driver
3) Booking Request
4) Confirmation of Request
5) Cancel Booking Request

5.7.3 Features not to be tested

Features not to be tested are from the developer's point of view.

1) How much memory is consumed by the android studio?
2) Software risk factor
3) Maintainability of the application

5.8 Testing Tools and Environment

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured. Test bed or test environment is configured as per the need of the Application under Test. Here black box testing environment is used. Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

5.9. Test Cases

In software engineering, a test case is a set of conditions or variables under which a tester will determine if a requirement upon an application is partially or fully satisfied. It may take many test cases to determine that a requirement is fully satisfied. In order to fully test that all the requirements of an application are met, there must be at least one test case for each requirement. A test case has components that describe an input, action or event and an expected response, to determine if a feature of an application is working correctly

Test cases for the application are:

5.9.1 Register as User

User will test whether he is registered or not using instructions in table 5.1 as below:

Table 5.1: Register as User

ID	TC 01
Description	User registers on the application
Tester	User
Test Type	Black box testing
Setup	User has opened application
Instructions	User clicks on the register button. User enter the name, location, mobile number and CNIC All information is in correct format and verified from system
Expected Result	User is registered on the application.
Required test scripts	No

5.9.2. Register as User (Alternative Scenario)

User will test whether he is registered or not using instructions in table 5.2 as below:

Table 5.2: Register as User (Alternative Scenario)

ID	TC 02
Description	User registers on the application
Tester	User
Test Type	Black box testing
Setup	User has opened application
Instructions	User clicks on the register button. User enter the name, location, mobile number and CNIC Option appear for user —RETRY” because the information entered is either not in correct format or is invalid.
Expected Result	User is not registered on the application.
Required test scripts	No

5.9.3 Register as Driver

Driver will test whether he is registered or not using instructions in table 5.3 as below:

Table 5.3: Register as Driver

ID	TC 03
Description	Driver registers on the application
Tester	Driver
Test Type	Black box testing
Setup	Driver has opened application
Instructions	Driver clicks on the register button. Driver enter the name, location, mobile number, CNIC, driving license and transport details All information is in correct format and verified from system
Expected Result	Driver is registered on the application.
Required test scripts	No

5.9.4 Register as Driver (Alternative Scenario)

Driver will test whether he is registered or not using instructions in table 5.4 as below:

Table 5.4: Register as Driver (Alternative Scenario)

ID	TC 04
Description	Driver registers on the application
Tester	Driver
Test Type	Black box testing
Setup	Driver has opened application
Instructions	Driver clicks on the register button. Driver enter the name, location, mobile number, CNIC, driving license and transport details (name, model, owner, picture and Certificate of Registration) Option appear for driver ―RETRY” because the information entered is either not in correct format or is invalid.
Expected Result	Driver is not registered on the application.
Required test scripts	No

5.9.5 Add Booking

User will test whether he has done booking or not using instructions in table 5.5 as below:

Table 5.5: Add Booking

ID	TC 05
Description	User request for booking
Tester	User
Test Type	Black box testing
Setup	User has logged in to the application
Instructions	User clicks on the ADDBOOKING button. User provide the luggage information and it's current + desired location. All information is in correct format and verified from system. Request is send to system.
Expected Result	Booking request is made by the user.
Required test scripts	No

5.9.6 Add Booking (Alternative Scenario)

User will test whether he has done booking or not using instructions in table 5.6 as below:

Table 5.6: Add Booking (Alternative Scenario)

ID	TC 06
Description	User request for booking
Tester	User
Test Type	Black box testing
Setup	User has logged in to the application
Instructions	User clicks on the ADDBOOKING button. User provide the luggage information and its current + desired location. All information is in correct format and verified from system. Request is send to system.
Expected Result	Booking request is not made by the user.
Required test scripts	No

5.9.7 Confirmation of Booking

Driver will test whether he has confirmed booking or not using instructions in table 5.7 as below:

Table 5.7: Confirmation of Booking

ID	TC 07
Description	Driver confirms the booking
Tester	Driver
Test Type	Black box testing
Setup	Driver is logged in
Instructions	Driver gets the booking request from system. Driver selects the request according to luggage capacity of its transport.
Expected Result	Confirmation message is send to system and then from system to user along with driver and car information.
Required test scripts	No

4.9.8. Confirmation of Ride (Alternative Scenario)

Driver will test whether he has confirmed booking or not using instructions in table 5.8 as below:

Table 5.8: Confirmation of Booking (Alternative Scenario)

ID	TC 08
Description	Driver confirms the booking
Tester	Driver
Test Type	Black box testing
Setup	Driver is logged in
Instructions	Driver gets the booking request from system. Driver select the request according to luggage capacity of its transport. If the capacity is small from the user specified luggage then driver will not accept the request.
Expected Result	Request is not accepted by the driver. No confirmation message found.
Required test scripts	No

CHAPTER 6

CONCLUSION AND FEATURE ENHANCEMENT

6.1 Introduction

This chapter describes the project conclusions (what we have done in our project) and future enhancements (what type of functionality can be added with passage of time).

6.2 Summary

The application allows user (driver and client) to book a transport in order to do shifting through internet. User don't have to go physically in order to search for transport. Transport which are available at the time when users are requesting for booking will be shown. This will help the Users who don't know how to do shifting (unaware of the procedure that where to go for searching the transport, price and based on weight how many trucks will be needed) and Google map tracks the transport location where it is at current time. This helps the user to know about where transport is right now. And how much time it will take to arrive.

By using this application driver and user are free from bargaining issue. They don't have to fight or argue for reducing the money. In this system, there is fixed price generated by system (calculated by considering distance) and user have to pay it. This application provides the mechanism where both users and drivers can cancel the ride within specified time interval. This saves the time. After cancellation time period is over, system generates a message of confirmation of ride so that driver start driving and user see the tracking of transport on Google.

It's a reliable system because this system is application based which handles many things, most importantly saving of time. It provides such an environment to user to get services without going out of their comfort zone.

6.3 Conclusions

The purpose of the project was to develop a android application that help people in moving households. Project comprises of an application that has two panels User and admin. User view can see suitable ride, registers/login to system, add booking, cancel booking and give his feedback at the end of his trip.

Following are the conclusions:

- User both client and driver will register themselves on this application.
- After logging in to this application, both client and driver will use it.
- Client can do booking (normal or urgent according to his choice).
- Driver will accept the request and booking will be confirmed.

6.4 Future Enhancements

The future implications of the project are very great considering the amount of time and resources it saves. The project we have undertaken can be used as a reference or as a base for realizing a scheme to be implemented in other projects of greater level.

In future software product can be enhanced as:

- Client and driver's information will be verified by taking permission from NADRA and respective transport institute.
- Office can be made in which people are hired for different purposes.
- Due to limited time I only make android application for the proposed project. An web based application should also be made to cater customer requirements like viewing products, placing orders, viewing their profile and redeem their existing points. This will be more convenient and easily accessible for customers.

References

- [[1] IEEE Software Engineering Standards Committee, ―IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", June 25, 1998.1]
- Tools and technologies
- https://firebase.google.com/docs/database
- https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5
- UML diagrams
- https://draw.io.com
- https://www.fluidui.com
- Guidelines for Extend Relationship
- http://sce.uhcl.edu/helm/rationalunifiedprocess/process/modguide/md_exrel.htm
- The Include and Extend Relationships in Use Case Model
- http://www.karonaconsulting.com/downloads/usecases_includesandextends.pdf