DROPTRIP – A Tour Management Application

A PROJECT REPORT

Submitted in Partial Fulfillment of the requirements for the Award of the Degree of BACHELOR OF Computer Science

Session: BSCS Fall 2019-23

Project Advisor: Sir Bilal Ahmad

Submitted By

Shaikh Usama Bin Naeem	**B-23511**
Rana Faraz Aslam Khan	**B-24112**
Inbisat Rana	**B-23566**

University of South Asia

Department of Computer Science

STATEMENT OF SUBMISSION

It is to certify that the final year project of BS(CS) "DropTrip – A Tour Management Application" was developed by Shaikh Usama Bin Naeem (B-23511), Rana Faraz Aslam Khan (B-23595) and Inbisat Rana (B-23566) under the supervision of "Sir Bilal Ahmad" that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

_____

Project Office Supervisor
Department of Computer Science

_____

Project Primary Advisor
Name: Sir Bilal Ahmad
Designation: Lecturer of CS Department, University of South

_____

Project Examiner
Name: Sir Khurram Jillani
Designation: Head of Department, University of South Asia.

Proofreading Certificate

It is to certify that I have read the document meticulously and circumspectly. I am convinced that the resultant project does not contain any spelling, punctuation or grammatical mistakes as such. All in all, I find this document well organized and I am in no doubt that its objectives have been successfully met.

_____

Sir Bilal Ahmad
Lecturer, University of South Asia.

Abstract

Tour Management is a multipurpose field that is related to tourism, hospitality, and valuable customers. Tour Management companies grab the customers for tours through their social media handles by running ads and that is how they organize tours. Tour Management companies share their packages to the targeted audience through ads and later they have to chat and convince them so they can book the trips with them. This is a lot of hassle and requires manpower to handle things and convince the customers. Tour organizers have been facing issues to gather customers for the tours and for that they started using students so they can get customers from universities. Recently tourism has grown a lot in Pakistan according to the international tourism index, Pakistan moved six spaces up from 89th to 83rd position this shows that our tour industry is growing rapidly. So, we decided to create an application that bridges the gap between customers who are looking for domestic tours within Pakistan especially.

We will be making an application the main purpose of this app will be to gather all the tour organizers in one place and offer discounted and flexible tours to our customers. Our customers will then be deciding on the tours according to their budgets and services the tour organizers are including. Hence, we will be saving them time and money for our customers and giving more opportunities for tour organizers and a hassle-free and optimized management portal where they can list and manage their upcoming and ongoing tours without any issue, they will be able to get the maximum out of our application. Tour organizers will be the ones organizing everything from accommodation to food and transport as well, we are just going to provide a place where they can register and list their tours to get something out of it. In this app we will be implementing some security features as well that will help them to connect the audience with Police and safety departments efficiently.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task. We are greatly indebted to our project supervisor “Sir Bilal Ahmad”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Shaikh Usama Bin Naeem

Rana Faraz Aslam Khan

Inbisat Rana

Abbreviations

SRS	Software Require Specification
PC	Personal Computer

Table of Contents

Table of Figures

CHAPTER 1

INTRODUCTION

CHAPTER 1

1.1 Introduction

Tourism is the largest and fastest-growing industry across the world. It is a source of revenue and employment. It also gives the opportunity for people to understand the culture, civilization, and religious aspects of a country. Tourism is a unique type of highly labor-intensive industry. It provides different services that are needed as well as expected by the incoming tourists. Tourism is one of the largest industries in terms of money spent by tourists in the countries they visit. The amount received from the Tourism industries sometimes exceeds the Gross National Product of many countries.

As the tourism industry of Pakistan is growing rapidly and we are getting International Tourists that are mostly visiting the Northern Areas of Pakistan and we are aware of the technology that is being used in the developed countries. So, we have just tried to make an application that will help the tourists to get everything from one application from Offered Tours to Custom Tours to the north.

So, our application will be a great opportunity for tourists to search for tours on their favorite locations. We will attract tourism companies through our marketing strategies and will eventually make them post their tours on our application to get the most out of the industry and tourists. This will help both the companies and tourists and will also increase the competition between different companies, it will help the tourists by giving them multiple tour options for one route and will give them an opportunity to select from the best deals. Initially we won't be charging anyone and as soon as our market grows, we will start using methods to increase the revenue for ourselves and the tourism companies.

1.2 Relevance to Course Modules

Programming Fundamentals (CMP-221)

We have studied the basics of programming in this course. Using the concepts of C++, we implemented this project.

Object Oriented Programming (CS-3244)

Using the concepts of OOP approach in programming pattern and coding design we implemented our project.

Software Engineering & Object-Oriented Software Engineering (CMP-3310 & CS-3044)

This course helped us to evaluate our development methodologies which are used in our project.

Database Systems (CSC271)

In this project, we develop relational database system for efficient working of the project by using the concepts of database systems.

1.3 Project Background

Tourism focuses on the tourists and tour companies their relationship and transparency plays a vital role in a successful tourism company. It has been thought that tourists prefer transparency in operations and nothing is left behind in the fulfilling of their needs from cultural and luxurious needs.

Currently tourists are searching for tours on social media platform and the tour companies spend a huge amount of their revenue on ads to get the customers. As the technology is increasing our application will bridge the gap between tourists and tours that they have to spend money and time to get their best packages from one place.

DropTrip will help get the best packages for tours in one place and will give the tourists a long list of available options for each destination they will get filters based on their mindset. We have different mindsets in the society some tourists prefer reviews over money and some prefer money over reviews. So, you don't have to worry about that we will be taking care of that for you that will help you select your tour operator based on your mindset.

1.4 Literature Review

All applications based on the domain of tour management have distinct features and they also work on different sections. There are very few applications that deals with tours specifically and we have narrowed our market as well that it is only working in Pakistan for now but will be expanded in the future. Tourists have to download multiple applications to book their tours and now after application gets live, they will get everything in one place. The application that is specifically designed for tourism should have an attractive user interface that straight away impact on your mental health and you start loving the complete process we will make sure that our application gives a positive and healthy feel to our users through the animations used in the application on different steps and will help the users to go through the process easily without any hassle.

1.5 Analysis from Literature

As far as tourism is concerned technology is of great interest for the tourists as it will help them to get better results without any communication barrier will obviously help them get into visiting multiple places in Pakistan. Foreign Tourists are the main concern for us as most of them have to face language barrier and through our app they will be getting all the necessary information required for their tours.

So, keeping in view the interest of foreign tourists we have proposed Tour Management Application which is an iOS and android based Tour Management Application. It will be a great opportunity for tour operating companies and tourists to get on one platform and fulfill all their needs from one place. It will be beneficial for both tourists and tour operators because there will be no need of spending thousands of rupees on ads to get customers and moreover as it is free so tourists will not have to pay heavy amounts to download the app to use our services it will surely increase the revenue of the companies and will help the tourists to get maximum options after spending minimum resources on searching for a better trip.

1.6 Methodology and Software Lifecycle for This Project

Whenever a small or large project starts, first thing all of project managers required is to select a software development model. Software Development Models is a way of developing a project, in which all of the programmers gather the user's requirements, design the project, develop it, and after all this testing and deployment of the project and in the end the maintenance of the project. There are several existing software development models that can be used to develop this application using software development models like Waterfall Model, Agile Models and Spiral Model etc. We will be using Agile model in our project

1.6.1 Adopted Methodology

The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only. At a time one iteration is planned, developed, and deployed to the customers. Long-term plans are not made.

Agile model is the combination of iterative and incremental process models. The steps involve in agile SDLC models are:

- Requirement gathering
- Requirement Analysis
- Design
- Coding

- Testing
- Deployment

The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. Though the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.

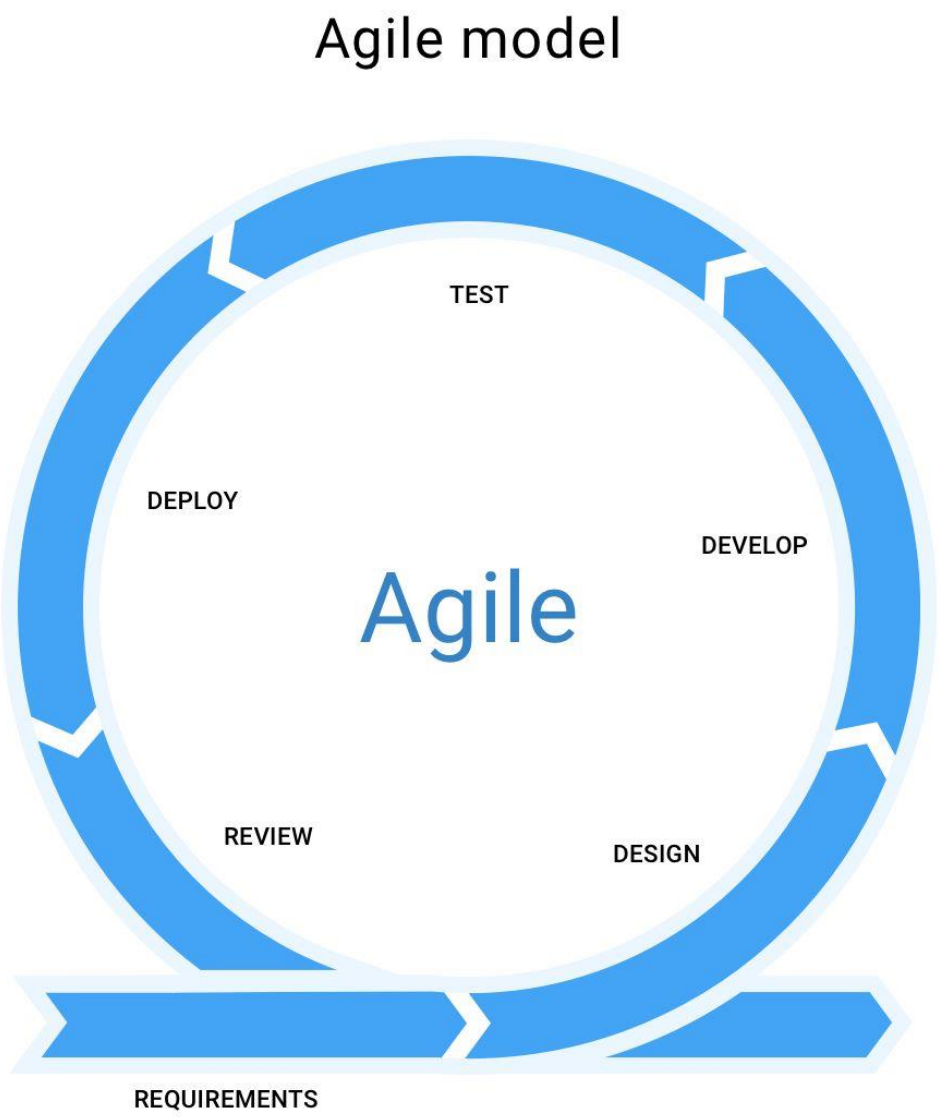

Figure 1: Incremental Model

1.6.2 Rationale behind Selected Methodology

The Agile methodology is a popular approach to software development that emphasizes flexibility, collaboration, and rapid iteration. The rationale behind selecting Agile methodology is that it allows teams to quickly respond to changes and feedback, and to deliver working software to customers in a timely manner.

One of the key benefits of Agile is that it allows for a more flexible approach to project management. Instead of following a rigid, linear process, Agile teams work in sprints, delivering small, incremental pieces of functionality in short timeframes. This allows teams to adapt to changing requirements and to respond quickly to feedback from customers.

Another major benefit of Agile is that it promotes collaboration and communication among team members. Agile teams work in a cross-functional, self-organizing way, with all team members involved in planning, development, and testing. This helps to ensure that everyone is on the same page, and that all team members are aware of progress and any issues that may arise.

Agile also focuses on delivering a working software incrementally, with each sprint delivering a potentially shippable product, this allows the stakeholders to see the progress and make changes if needed. This approach helps in avoiding the big-bang approach which may lead to delays, cost overruns and not meeting the stakeholders' requirements.

Overall, the Agile methodology is a popular choice for software development projects because it promotes flexibility, collaboration, and rapid iteration. It allows teams to respond quickly to changes and feedback, and to deliver working software in a timely manner.

CHAPTER 2

PROBLEM DEFINITION

Chapter 2: Problem Statement

2.1 Problem Statement

It is difficult to find best tours in good rates on internet users have to search a lot to find their desired tours but they are still unable to find it due to a lot of reasons like often the contact numbers provided by the companies are closed, they don't reply on social media, their companies are closed and their listing has incorrect details. That is why we have planned to develop this application to provide ease to our customers by providing all the things in one place users can chat directly with the tour guide, they can see all the minor details on the tour information page.

2.2 Deliverables and Development Requirements

Currently, we should deal with these drawbacks in an innovative way by providing information to user about Locations, Agents, Featured, Your Trips and Custom menu on the home page of the application. Some mobile applications have functionalities that are not executable and have bugs these things disturbs and irritates the tourists and the user avoids using such type of applications. So, we have decided to develop a mobile application that will help our users to get all information of all the current tours scheduled in one place without any buggy UI and functions that don't respond.

This application targets all audience, users can select tours according to their need and we have implemented filters as well that will further help the users to narrow down their search. User will select the tour from the listing and according to their budget then they will select the Number of people going on the tour after that the user will add the details of all the members and after submitting their tour will be booked.

There are three panels of this application:

- User Panel
- Agent Panel
- Admin Panel

Through the user panel user can sign up and after login the user can view the home page of the application which displays 5 different menus which are Locations, My Trips, Custom, Agents and Featured. Agent panel can be accessed by the tour companies and agents can add and edit details about their tours and can manage their bookings from the application they can generate an excel sheet from the application which will be downloaded in the application on one click.

Hardware that is required in development phase is:

- iOS and Android Devices
- Laptop core i5
- High Speed Internet Connection

Software and language that are required to develop this project are:

- Android Studio
- VS Code
- Fire base Database
- Firefoo
- Figma

2.3 Current System

The project relatable to our project is TripAdvisor. On TripAdvisor there only focus is on the western market and no localization is done on their side. In this application there are no international tours at the moment we have made sure to localize the application as much as we can. In our application we can give the information of top 5 best tour organizers of the year to help the users select from the best of the best in the market through their reviews. We are allowing the users to stay on one app and process the whole tour by staying on our app from searching for the desired tour, booking the tour, discussing the tour with the agent and paying for the tour through the application.

CHAPTER 3

REQUIREMENT ANALYSIS

Chapter 3

3.1 Requirement Analysis

In this chapter requirements analysis, feasibility study, planning, forecasting, modeling, scheduling and design of the project is discussed. For developing any project, the major problem is requirement gathering. Asking questions from clients is straightforward than collecting requirements. We will also focus on functional and non-functional requirements. The procedure for gathering requirements has its own defined procedure according to the complexity of the application. To define project schedule and processing, different models and techniques also focused on this chapter.

3.2 Requirements Gathering Techniques

A requirement can be defined as a condition or capability that must be processed by a product or an application. Techniques that can be used for collecting requirements are as follows:

- By survey and interviews
- By observations
- Using software tools
- Using techniques for decision making
- Use of prototype

Figure 2: Requirement Gathering

The techniques we have used to collect requirements are observations and conduct meetings with the gasoline station managers.

3.3 Use Cases Diagram(s)

An important part of the analysis phase is to drawing the diagrams of Use cases. They are used through the phase of analysis of a project to find and divide functionality of the application. Application is separated into actors and use cases. Actors play the role that are played by the application users. Use cases define the application behavior when one of the actors sends any particular motivation. This type of behavior can be described by text. It describes the

motivation nature that activates use case, the inputs and outputs to some other actors and the behavior of conversion of inputs to the outputs. Usually, the use case describes everything that can go wrong during the detailed behavior and what will be helpful action taken by the application. Some of the use cases are as follows:

3.3.1 System Level Use Case

Actors: User, Admin and Unregister User

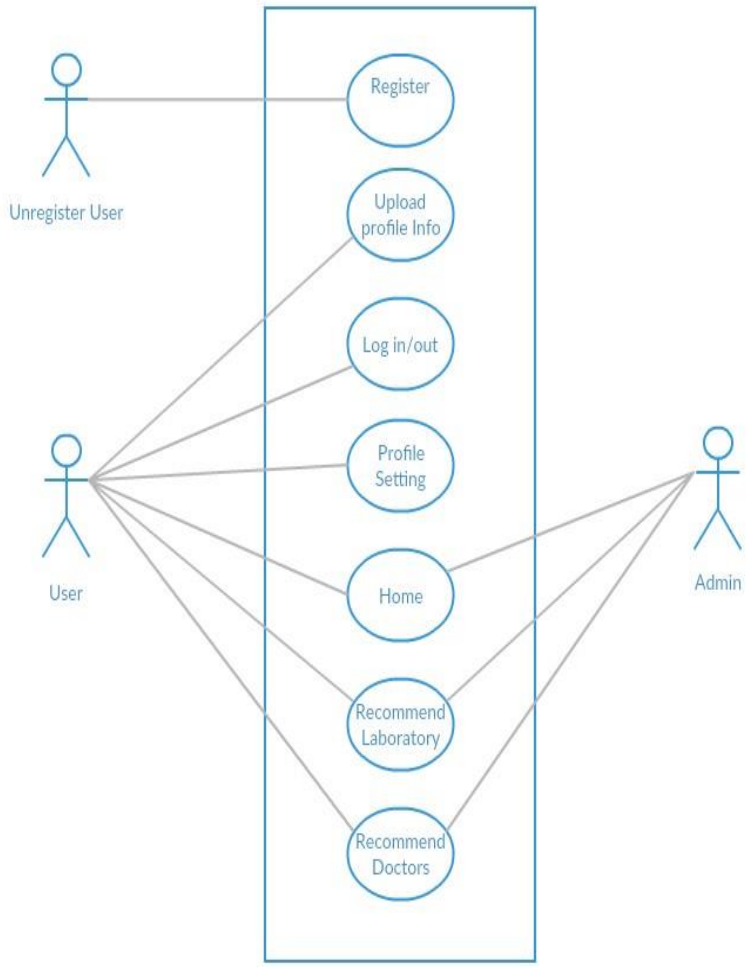

Figure 3: System Level Use Case

3.4 Detailed Use Case

3.4.1 Sign Up

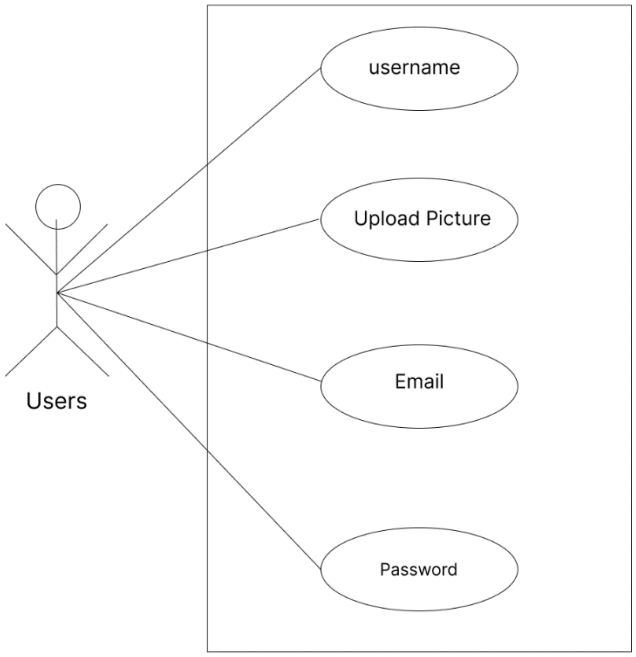

Figure 4: Sign Up Use Case Diagram

3.4.2 Sign in

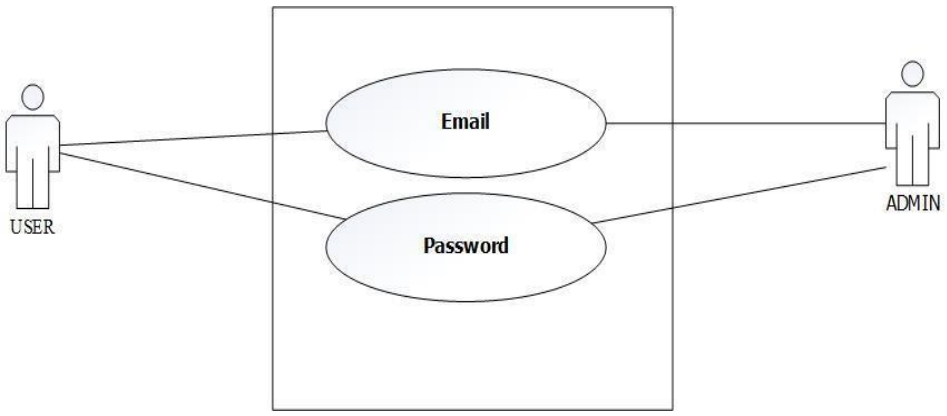

Figure 5: Sign in

3.4.3 Add Trip Use Case

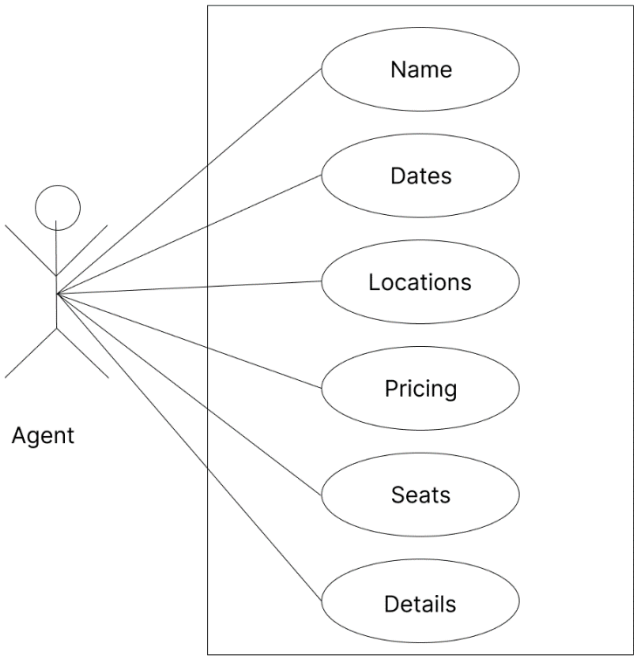

Figure 6: Add Trip

3.4.4 Book Trip

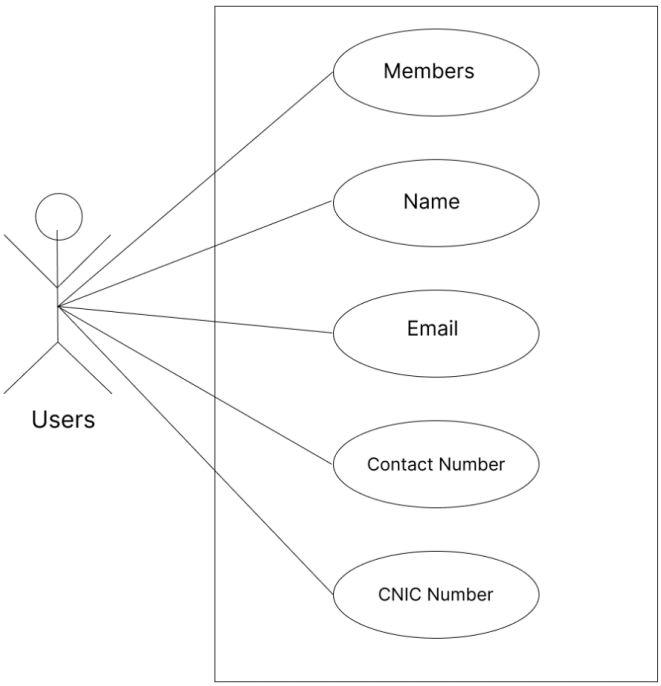

Figure 7: Book Trip

Requirements analysis is the process of planning, forecasting and studying the overall former needs of the application requirements. Requirements analysis is further divided into two parts:

1. Functional Requirements
2. Non-Functional Requirements

3.5 Functional Requirements

Functional requirements are the requirements that the system must have to provide after the successful installation of the project. Functional requirements are basically the need of the users. Functional requirements tell the behavior of the applications in different situations and in different environments with different types of input values.

Functional requirements of the DROPTRIP – Tour Management Application are as follows:

- User Registration & Login
- Tour Search & Browsing
- Tour Booking
- Push notifications
- Integration with third-party services such as firebase authentication, storage and Firestore.
- User profile management

3.5.1 User Registration Portion

Following are the functional requirements for the User Registration portion:

- FR-1: User should be able to create a profile
- FR-2: User should be able to Log in to his profile

3.5.2 Tower Search & Browsing

Following are the functional requirements for the Tower Search & Browsing portion:

- FR-3: User should be able to search and browse available tours

3.5.3 Tour Booking

Following are the functional requirements for the Tour Booking portion:

- FR-4: User should be able to book the tours

3.5.4 Push Notifications

Following are the functional requirements for the Push Notifications portion:

- FR-5: User should be able to see the push notifications of the application

3.5.5 Integration of Third-Party Services

Following are the functional requirements of Integration of Third-Party Services portion:

- FR-6: User should be able to upload pictures in while posting trips
- FR-7: User should be able to upload his own profile picture

3.5.6 User Profile Management

Following are the functional requirements for the User Profile Management portion:

- FR-8: User should be able to manage his own profile

3.6 Non-Functional Requirements

Following are the non-functional requirements for DropTrip:

3.6.1. Usability

The usability requirements for DROPTRIP are following:

• USE-1: Ensuring that the app is easy to use and understand, with a clear and consistent user interface.

3.6.2 Performance

The performance requirements for the application are:

- PER-1: Ensuring that the app is responsive and can handle a high number of users.

3.6.3 Compatibility

The compatibility requirements for the application are:

- COM-1: Ensuring that the app is compatible with a variety of device types and operating systems, including iOS and Android.

3.6.4 Testability

The testability requirements for the application are:

- TEST-1: Ensuring that the app can be easily tested and quality-checked before release.

3.6.5 Reliability

The reliability requirements for the application are:

- REL-1: Ensuring that the app is available and functional at all times, with minimal downtime.

CHAPTER 4

DESIGN AND ARCHITECTURE

Chapter 4

4.1 User Interface (UI):

The app features a simple and clean layout that makes it easy for users to navigate and find the information they need. The main screen of the app displays a list of upcoming tours, which users can scroll through and tap on to view more details. Users can also easily search for specific tours, create and manage their own tours, and access their account settings from the main menu. The app also features an interactive map view, which allows users to see the locations of upcoming tours and get directions to them. Overall, the user interface for DropTrip is designed to provide a seamless and efficient experience for users on both iOS and Android devices.

4.2 Database:

DropTrip, a tour management mobile application developed using React Native, utilizes Firestore as its database. Firestore is a flexible and scalable NoSQL cloud database that allows for real-time data synchronization across multiple devices. This means that any changes made to the tours in the database by one user will be immediately reflected on all other connected devices. It also provides powerful querying capabilities, allowing the app to easily retrieve specific tours based on different criteria. Additionally, Firestore's security rules provide a way to control access to data in the database and ensure that only authorized users can make changes. This makes it an ideal choice for DropTrip as it allows for real-time updates, easy data retrieval and easy management of data access rights. Firestore's seamless integration with React Native also makes it easy to implement and use within the application.

4.3 Cloud Services:

DropTrip, a tour management mobile application developed using React Native, makes use of Firebase Authentication for its user authentication needs. Firebase Authentication is a powerful and flexible cloud-based service that allows for easy and secure user authentication. It provides multiple options for users to sign-in such as Email/Password, social media login providers like Google, Facebook, Twitter and many more. This allows users to quickly and easily create an account and log in to the app, without the need to manage their own password and user information.

4.4 Push Notifications:

FCM is a cloud-based messaging service that allows for the delivery of real-time notifications to users. The notifications can be sent to all users or targeted to specific users based on certain criteria such as their location or past behavior in the app. With FCM, DropTrip can send notifications to users about upcoming tours, changes to existing tours, promotions, and other important updates. This keeps the users informed and engaged with the app. FCM also allows for the creation of custom notifications with rich media and interactive actions, which can improve user engagement and retention. FCM also provides a way to track the delivery and engagement of push notifications, which can be used to optimize the performance of the notifications and improve the overall user experience.

4.5 Authentication & Authorization:

We used Firebase Authentication for its user authentication needs and Firestore for its data storage. Firebase Authentication provides multiple options for users to sign-in such as Email/Password. This allows users to quickly and easily create an account and log in to the app, without the need to manage their own password and user information. Additionally, Firebase Authentication also provides built-in security features such as email verification, password reset, and phone number verification that ensures the authenticity of the users and provides an extra layer of security to the app. On the other hand, Firestore provides an efficient way to store and retrieve the data used in the app. Firestore also provides robust security rules that can be used to control access to the data in the database, ensuring that only authorized users can make changes. This ensures that the data is safe and can only be accessed by the right people. In addition, Firebase Authentication and Firestore also allows to define roles and permissions for different types of users and restrict access to specific parts of the app and data accordingly. This allows DropTrip to provide different levels of access to different users based on their roles, ensuring that only authorized users can access sensitive information and perform certain actions.

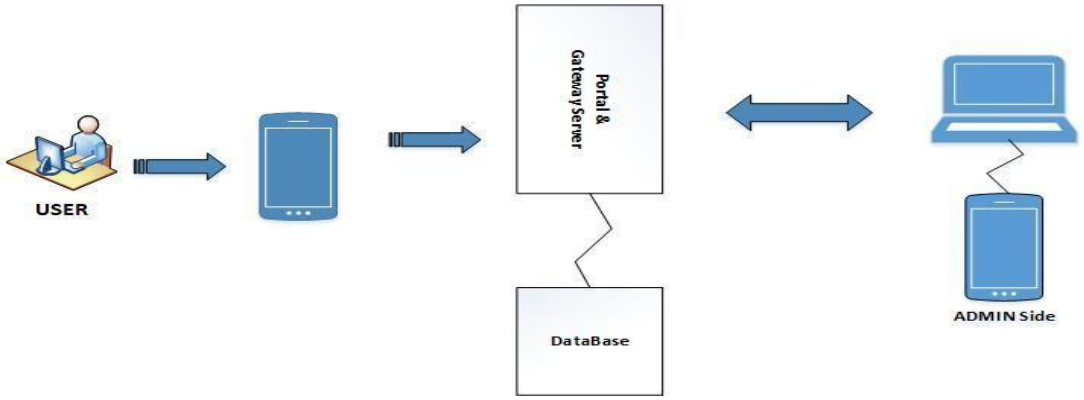

Figure 8: System Architecture

4.6 Data Representation

Data refers to the symbols that represent people, events, things, and ideas. Data can be a name, a number, the colors in a photograph, or the notes in a musical composition.

- Data Representation refers to the form in which data is stored, processed, and transmitted.
- Devices such as smartphones, iPods, and computers store data in digital formats that can be handled by electronic circuitry.

4.6.1 Process Flow/Representation

A Process Flow Diagram (PFD) is a type of flowchart that illustrates the relationships between major components at an industrial plant. It's most often used in chemical engineering and process engineering, though its concepts are sometimes applied to other processes as well. It's used to document a process, improve a process or model a new one. Depending on its use and content, it may also be called a Process Flow Chart, Flowsheet, Block Flow Diagram, Schematic Flow Diagram, Macro Flowchart, Top-down Flowchart, Piping and Instrument Diagram, System Flow Diagram or System Diagram. They use a series of symbols and notations to depict a process. The symbols vary in different places, and the diagrams may range from simple, hand drawn scrawls or sticky notes to professional-looking diagrams with expandable detail, produced with software.

4.7 Design Models

A design model in Software Engineering is an object-based picture or pictures that represent the use cases for a system. Or to put it another way, it is the means to describe a system's implementation and source code in a diagrammatic fashion. This type of representation has a

couple of advantages. First, it is a simpler representation than words alone. Second, a group of people can look at these simple diagrams and quickly get the general idea behind a system. In the end, it boils down to the old adage, 'a picture is worth a thousand words.

The applicable models may include:

- Class Diagram
- Sequence Diagram
- Activity Diagram

4.7.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

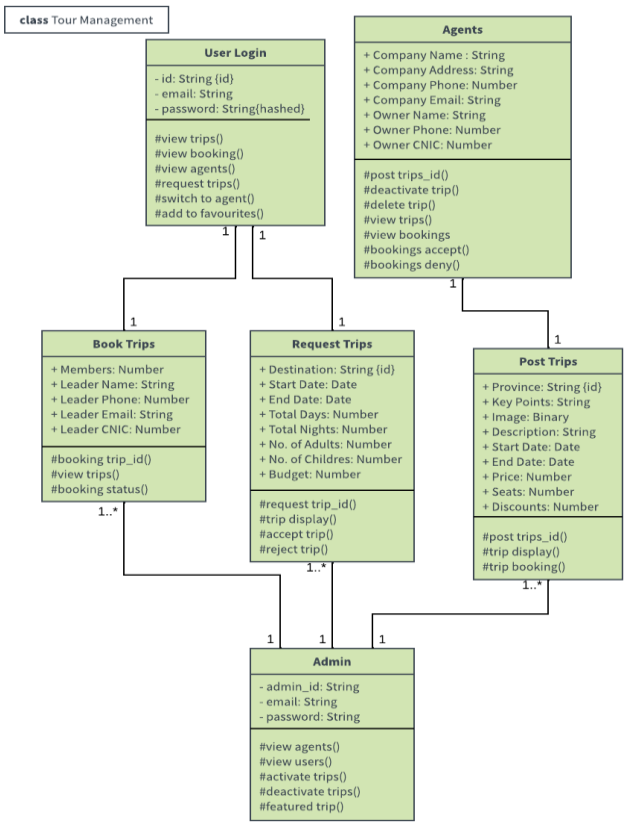

Figure 10: Class Diagram

4.7.1.1 Actors:

User

User can create and then login into the account and can view all the trips posted by agents and can book trips, check the status of their bookings, can add multiple trips in favorites.

Admin

Admin can view all the registered users, can view all the agents, can activate trips and also the admin can add trips in featured section.

Agent

Agents can post trips, delete trips they can view the trips posted by other agents, they can accept and deny the bookings done by the users and they can also delete and deactivate their own posted trips.

4.7.2 Sequence Diagram

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

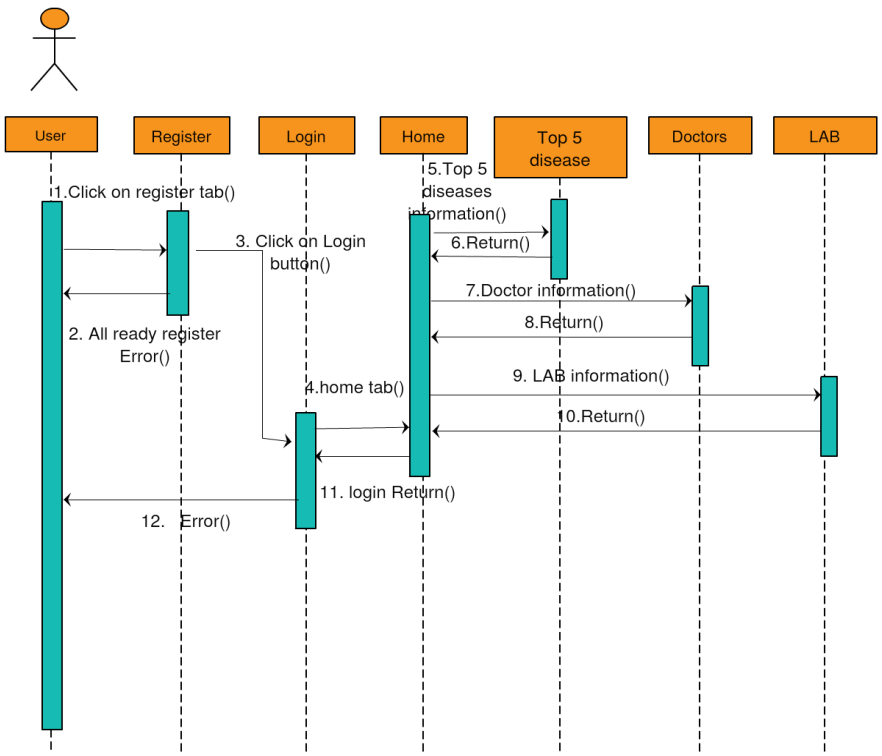

Figure 11: Sequence Diagram

4.7.3 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

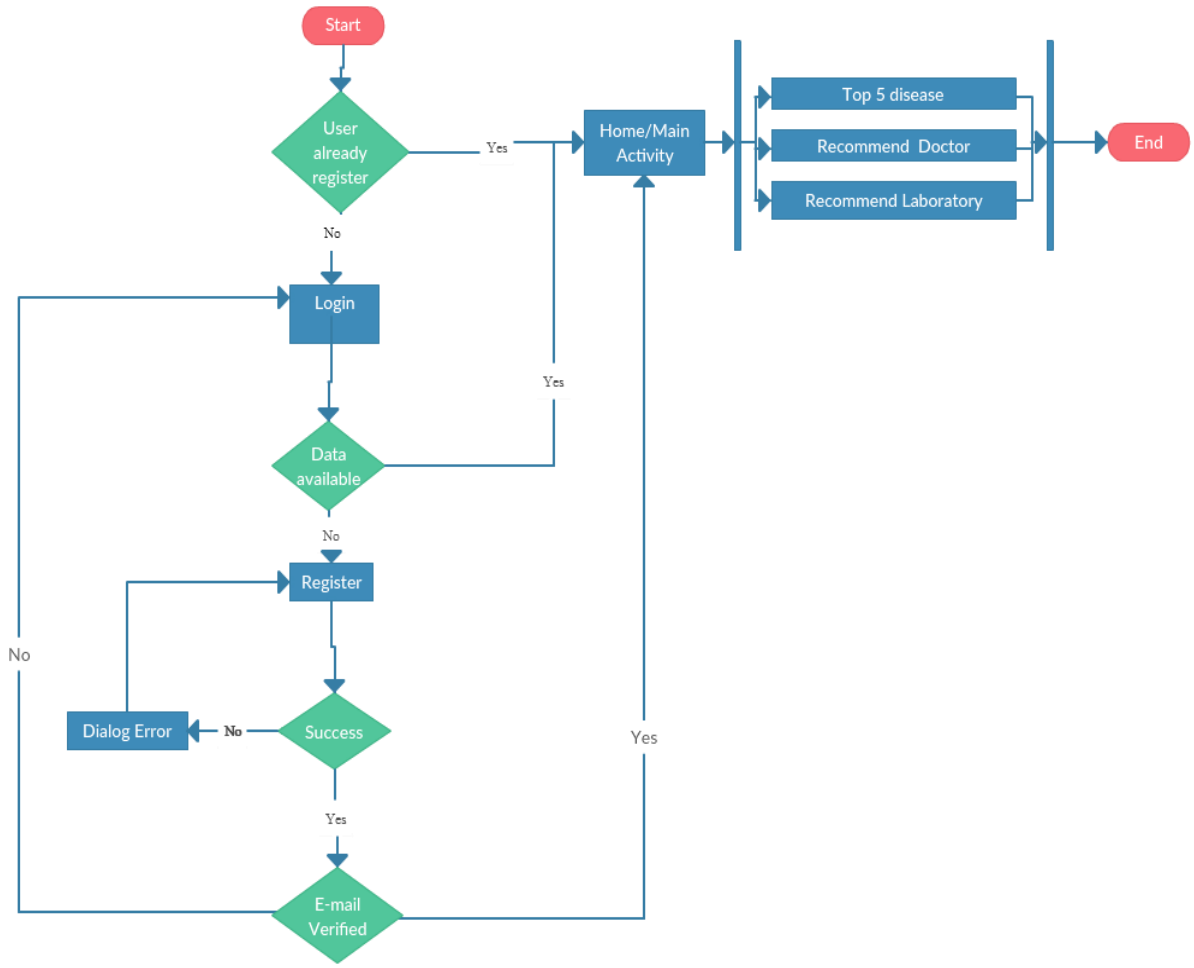

Figure 12: Activity diagram

CHAPTER 5

IMPLEMENTATION

Chapter 5

5.1 Implementation

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

5.2 Algorithm

We have a login form for both admin and user, after submitting the basic information for registration, admin will enter the list of doctors and the diseases which will save in database. The application process to access the data of doctors and disease from database and show the list of doctors and diseases to user which provide information. Simply, Admin upload data which will be seen by the user.

5.3 External APIs

In the mobile application DropTrip built using React Native, we utilized various APIs from Firebase to handle authentication, storage, and database functionality. For authentication, we used Firebase's authentication API to securely sign up and log in users, allowing them to access the application's features. Firebase's storage API was utilized for storing and retrieving user data, such as images and other files, efficiently and securely. Additionally, we used Firebase's database API to manage and store the application's data in a real-time, scalable manner. These APIs from Firebase allowed us to offer a seamless and secure experience to the users of DropTrip and provided robust infrastructure to support the application's growth.

5.3.1 Firebase Authentication API:

Firebase Authentication API provides a variety of services to help developers authenticate and manage users in their applications. Some of the services we used by Firebase Authentication API include:

1. **Email and password authentication:** Allows users to sign in with an email address and password, and also to create new accounts with the same.
2. **User management:** Provides tools to manage users, including creating, updating, and deleting user accounts.
3. **Security:** Firebase Authentication API implements industry-standard security measures such as encrypted data storage and security tokens to protect user data.

4. **Custom Claims:** Allows administrators to manage custom claims and assign additional privileges to users.

These are some of the services provided by Firebase Authentication API, which can be used to implement user authentication and authorization in your application.

5.3.2 Firebase Storage API:

Firebase Storage API provides a scalable and secure cloud-based storage solution for developers to store and serve user-generated content, such as images, audio, and video files. Some of the services we used by Firebase Storage API include:

1. **File upload and download:** Provides the ability to upload and download files to and from Firebase Storage, with the option to use resumable uploads for large files.
2. **Secure file access:** Firebase Storage implements security measures such as access controls and signed URLs to ensure that only authorized users have access to the stored files.
3. **Real-time updates:** Firebase Storage provides real-time updates when files are added or updated, making it easy to build dynamic and responsive user experiences.
4. **File metadata:** Allows developers to store and retrieve metadata about stored files, such as creation date, file size, and content type.
5. **Cost-effective:** Firebase Storage provides cost-effective storage with automatic scaling to handle high traffic and large amounts of data.

These are some of the services we used by Firebase Storage API, which are used to make the app more powerful and scalable storage capabilities to your application.

5.3.3 Firebase Firestore Database API:

Firestore Database API is a NoSQL document-oriented database provided by Firebase. Some of the services we used by Firestore Database API include:

1. **Real-time updates:** Provides real-time updates of the database, making it easy to build dynamic and responsive user experiences.
2. **Document-oriented:** Firestore stores data as collections of documents, allowing for flexible and scalable data modeling.

3. **Querying:** Supports complex queries to retrieve data based on multiple conditions and sorting.
4. **Indexing:** Automatically indexes data for fast and efficient querying.
5. **Offline support:** Firestore supports offline access, allowing users to interact with the database even when they are not connected to the internet.
6. **Serverless:** Firestore is a serverless database, allowing developers to focus on building their applications without having to manage infrastructure.
7. **Secure:** Firestore implements security measures such as access controls and encryption to ensure the privacy and security of data.

These are some of the services we used by Firestore Database API, which can be used to add powerful and scalable database capabilities to your application.

5.4 User Interface

All of these Screenshot Represent the Project.

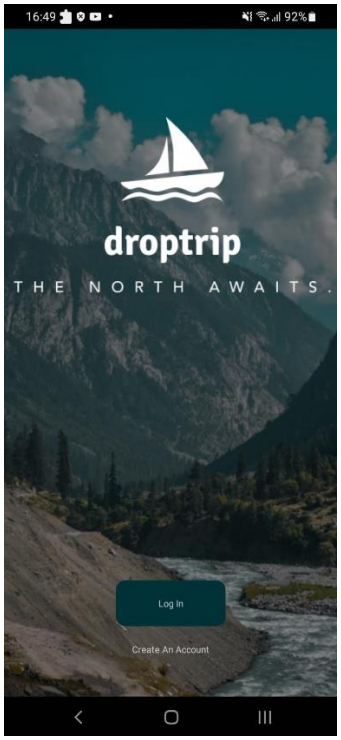

Figure 13: Landing Screen

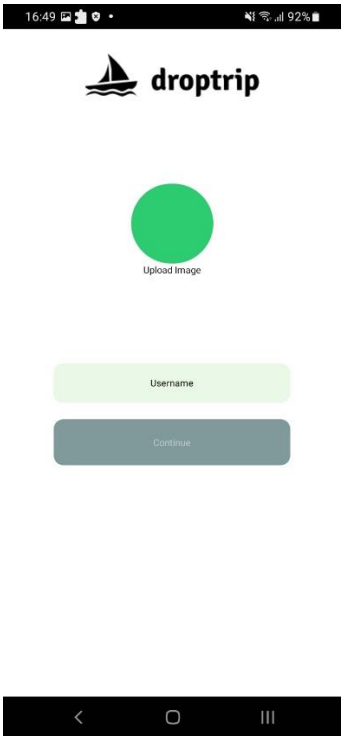

Figure 14: Sign Up Screen

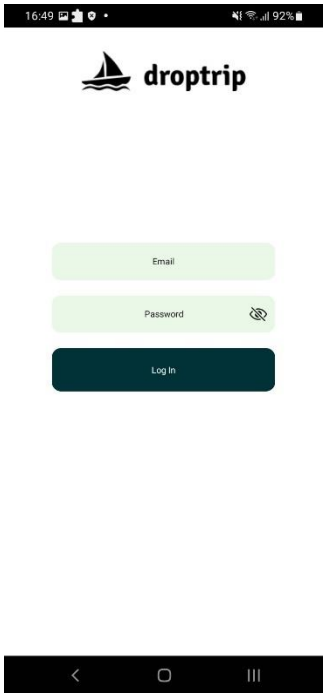

Figure 15: Log In Screen

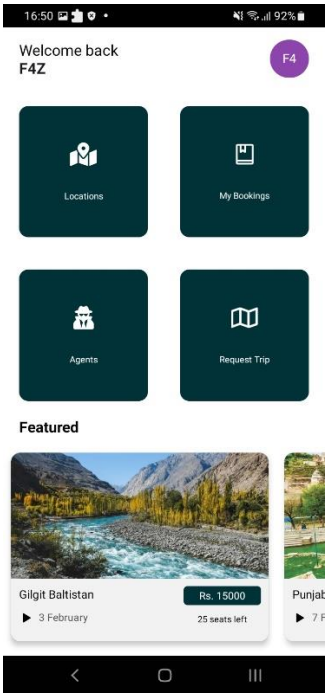

Figure 16: Home Screen

Figure 17: Province Screens

Figure 18: Trips Screen

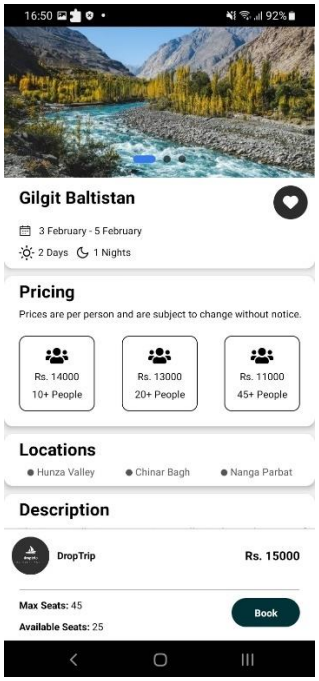

Figure 19: Trip Screen

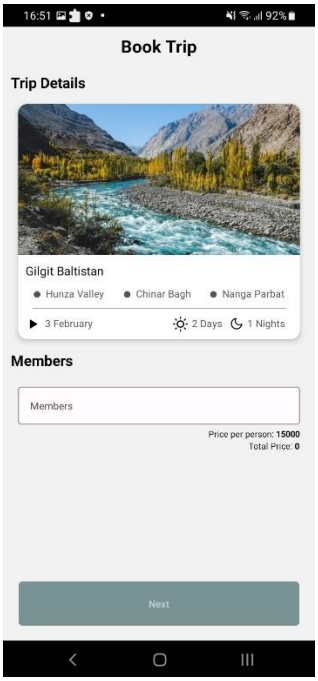

Figure 20: Booking Screen

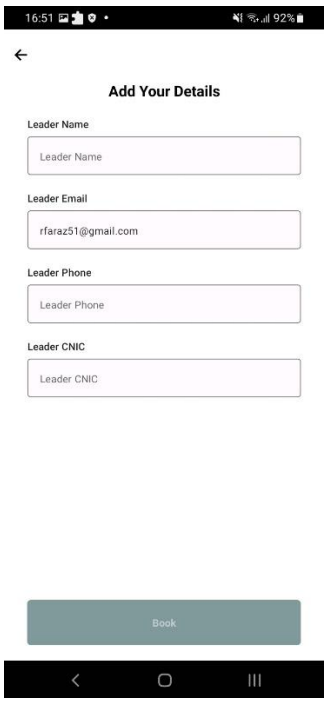

Figure 21: Booking Screen

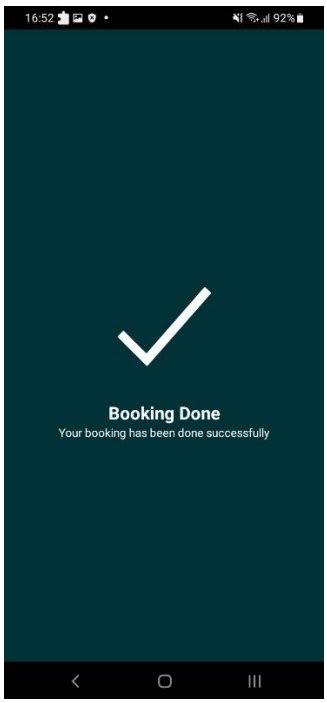

Figure 22: Booking Done Screen

Agent View:

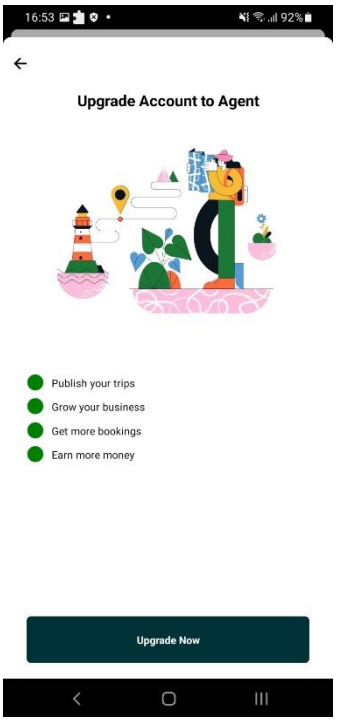

Figure 23: Agent Screen

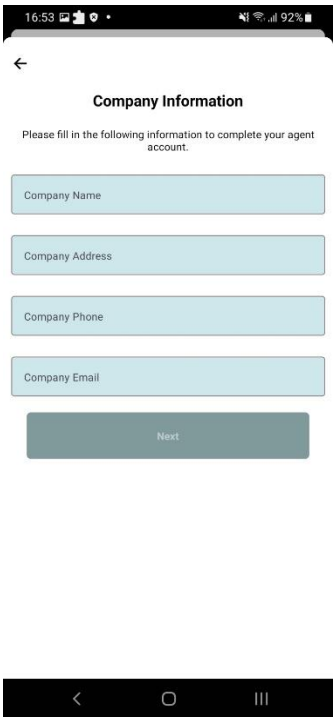

Figure 24: Agent Information Screen

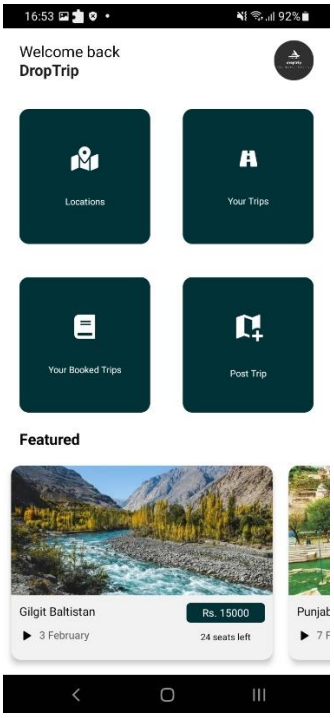

Figure 25: Agent Home Screen

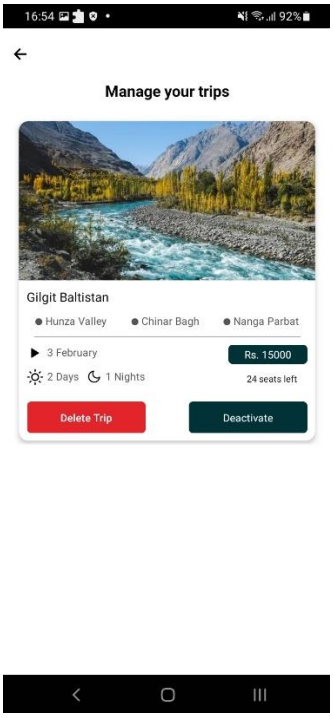

Figure 26: Manage Trips Screen

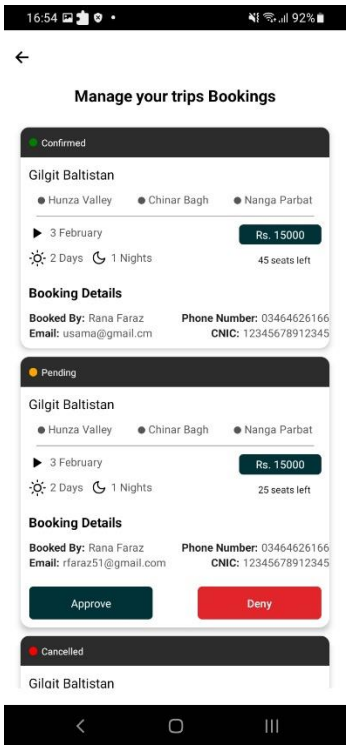

Figure 27: Manage Trips Screen

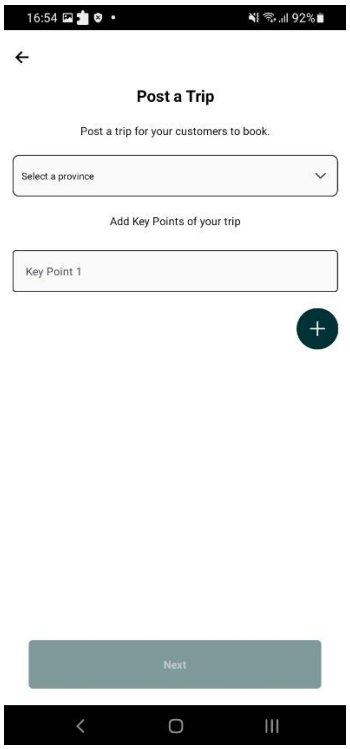

Figure 28: Post Trip Screen

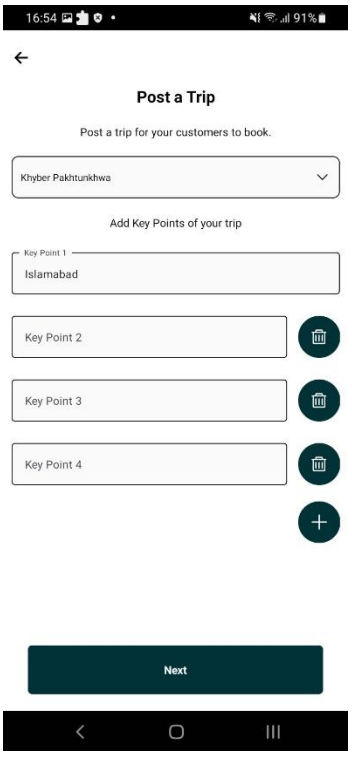

Figure 29: Post Trip Screen 2

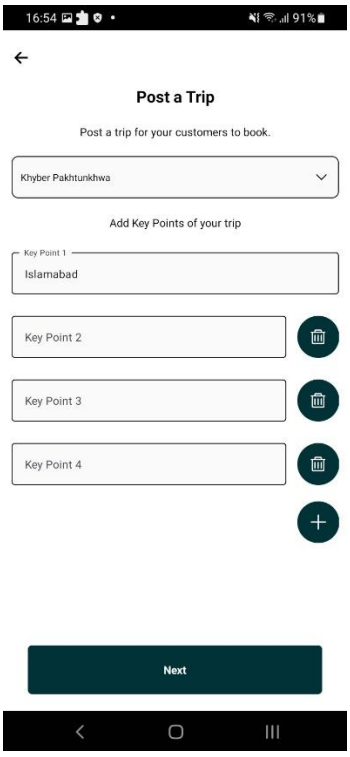

Figure 30: Post Trip Screen 3

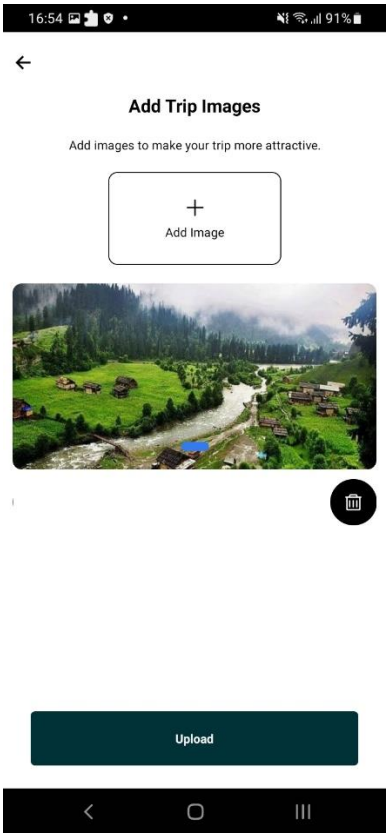

Figure 31: Post Trip Screen 4

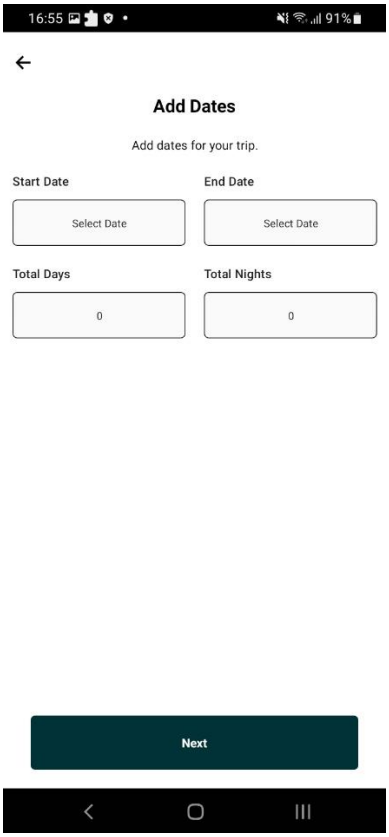

Figure 32: Post Trip Screen 5

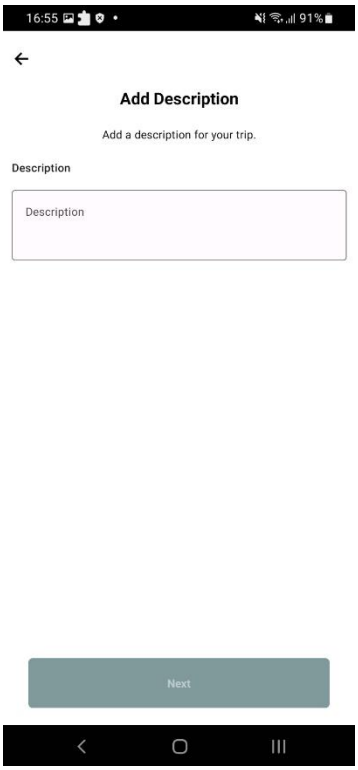

Figure 33: Post Trip Screen 6

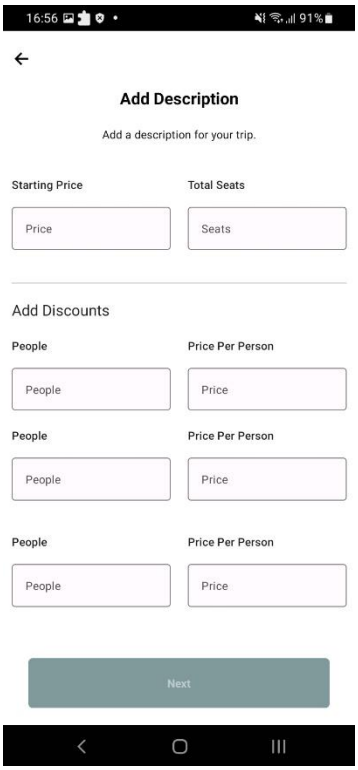

Figure 34: Post Trip Screen 7

Admin View:

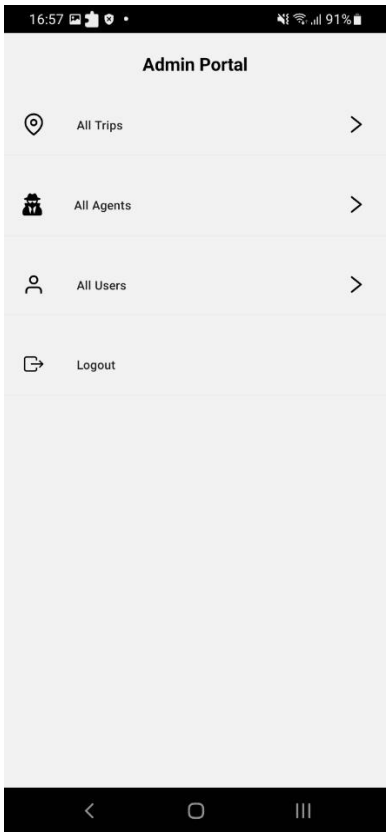

Figure 35: Admin View

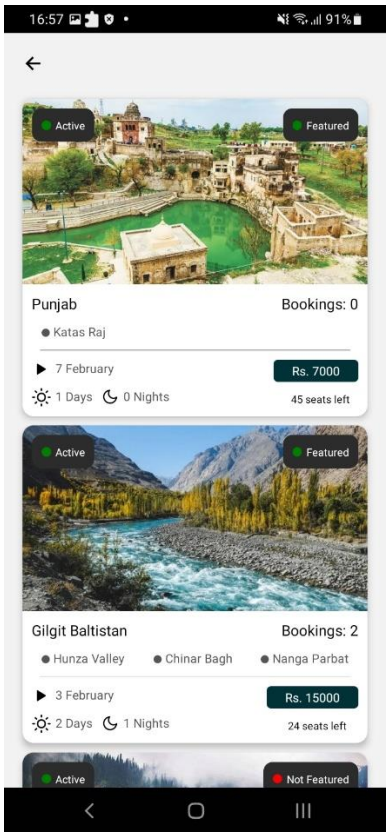

Figure 36: Trips Screen

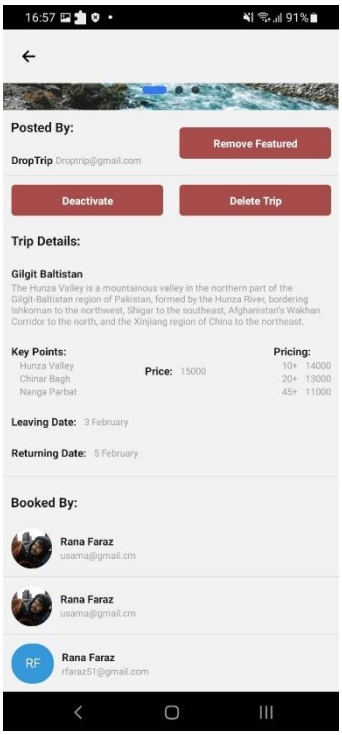

Figure 37: Booked Trips Screen

CHAPTER 6

TESTING AND EVALUATION

Chapter 6

6.1 Testing and Evaluation

Testing and evaluation play an important role in ensuring the quality and functionality of DropTrip, a hybrid mobile application built in React Native which can be installed on both iOS and android devices. Several types of testing can be performed System, Unit, Integration, Non-Functional, Smoke, Sanity, Ad-hoc, UI, Regression, and UAT Testing.

Evaluation is the process of measuring the success of the project, and it can be done through various methods, which mainly include User Feedback, Analytics, Comparative Analysis. Overall, testing and evaluation play a crucial role in ensuring that DropTrip meets the high standards expected for a hybrid mobile application and provides an exceptional user experience.

6.2 Manual Testing

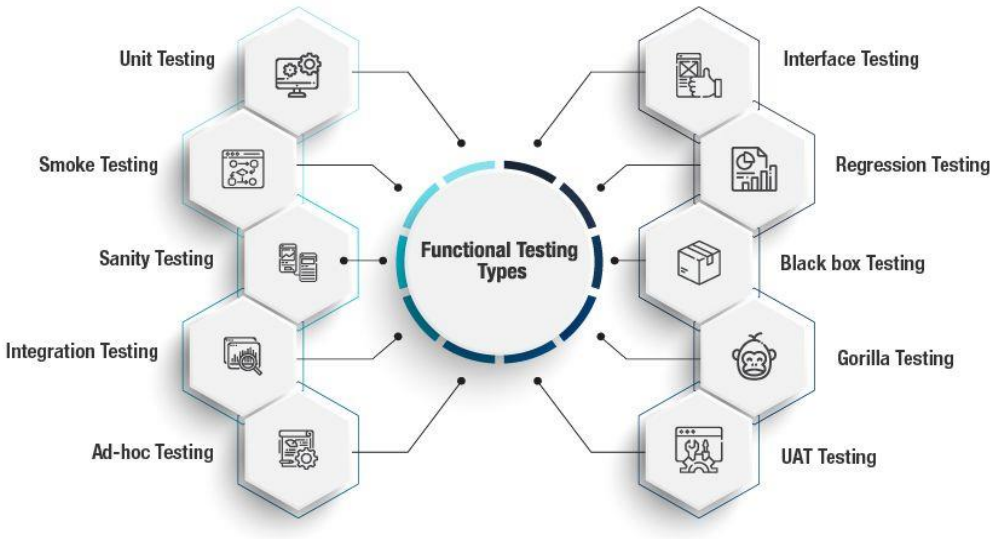

Figure 24: Functional Testing

6.2.1 System Testing

System testing was performed on mobile application. The purpose of system testing was to validate the entire application as a system and to ensure that it was working as expected in a real-world environment. During system testing, the application was tested in a simulated real-world scenario,

and various tests were performed to validate its functionality, performance, and stability. The tests were designed to verify the end-to-end functionality of the application and to uncover any issues that may not have been detected during earlier phases of testing. The results of the system testing helped us to identify any remaining issues and to ensure that the application was ready for release. System testing was a critical part of the quality assurance process for DropTrip, and helped to ensure that the application would provide a stable, reliable, and satisfying experience to its users.

6.2.2 Unit Testing

Unit testing was performed on DropTrip. The purpose of unit testing was to validate the individual components of the application and to ensure that each unit was working as expected. During unit testing, small, isolated portions of the code were tested in isolation from the rest of the application. The tests were designed to verify the behavior of each unit and to catch any errors early in the development process. The results of the unit testing helped the team to identify any issues with the code and to ensure that it was working as intended. Unit testing was a crucial part of the development process and helped to increase the overall quality of the application by catching any bugs early on and reducing the risk of more serious issues arising later. The thorough unit testing performed on DropTrip helped to ensure that the application would provide a stable and reliable experience to its users.

6.2.3 Integration Testing

Integration testing was performed on DropTrip. The purpose of integration testing was to verify the interaction between the various components of the application and to ensure that they were working together as expected. During integration testing, different modules of the application were combined and tested, and the data flows between these modules were verified. The team also tested the integration of the application with external systems and APIs that it depended on, such as authentication, storage, Firestore api's, and other third-party services. Integration testing helped to uncover any issues that may have arisen due to the combination of different components and to ensure that the application was able to function correctly in a real-world environment. Now the application was ready for release, and it would provide a seamless user experience to its users.

6.2.4 Non-Functional Testing

Non-functional testing was performed on our application. The purpose of non-functional testing was to evaluate the performance, scalability, security, and other non-functional requirements of the application. During non-functional testing, various tests were performed to measure the performance of the application under different loads and conditions, to verify its ability to scale as needed, and to assess its security against potential threats. We also tested the application's compatibility with different hardware and software configurations, and its ability to function in different network environments. The results of the non-functional testing helped the team to identify any potential issues and to ensure that the application met the high standards for performance, scalability, security, and other critical non-functional requirements. This helped to ensure that the application was able to deliver a positive user experience, even in challenging conditions.

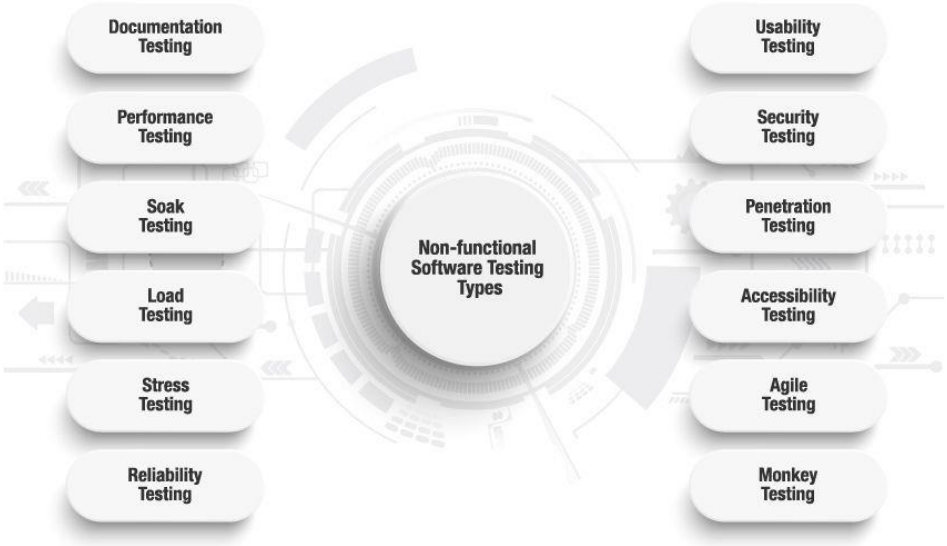

Figure 25: Non-Functional Testing

6.2.5 Smoke Testing

Smoke testing was performed on our mobile application. The purpose of smoke testing was to quickly assess the basic functionality of the application and to determine if it was ready for further testing. During smoke testing, a set of critical and high-level tests were executed to verify the most important features of the application, including its ability to launch, navigate, and perform basic tasks. The tests

were performed on Android platforms to ensure that the application was functioning correctly on both devices. The results of the smoke testing provided us with an early indication of the overall stability of the application, and any issues found were immediately addressed. Smoke testing helped to ensure that the application was ready for more in-depth testing and that it met the minimum requirements for functionality.

6.2.6 Sanity Testing

Sanity testing was performed on DropTrip. The purpose of sanity testing was to quickly verify that the most critical functionality of the application was working as expected after recent changes or updates. A subset of test cases, covering the most essential features of the application, was executed during sanity testing. The tests were designed to be quick and efficient, allowing us to confirm that the application was stable and ready for further testing. The results of the sanity testing provided us with confidence that the application was functioning correctly, and any issues found were immediately addressed to maintain the quality of the application. Sanity testing helped to catch any major problems early on in the testing process, reducing the risk of more serious issues arising later.

6.2.7 Ad-Hoc Testing

Ad-hoc testing was performed on the mobile application. Ad-hoc testing involves exploring the application in an unstructured manner to find any potential issues that may not have been identified in other types of testing. We carried out ad-hoc testing by using the application in a variety of different scenarios, and by trying out different combinations of inputs and actions. This approach allowed them to discover any unexpected behavior or bugs that may not have been found through other testing methods. The results of the ad-hoc testing were used to further improve the quality and stability of the application, and to ensure that it provided a seamless user experience on both iOS and Android devices. Ad-hoc testing is an important complement to other types of testing and helps to ensure that the application is thoroughly tested and free of defects.

6.2.8 UI Testing

UI (User Interface) testing was conducted on DropTrip. The purpose of UI testing was to verify the overall look, feel, and functionality of the application from the user's perspective. During UI testing, we focused on testing the visual elements, such as buttons, images, and text, to ensure that they were

displayed correctly and that the application was easy to navigate. They also tested the responsiveness of the application on different screen sizes and resolutions to ensure that it provided an optimal user experience on both iOS and Android devices. The results of the UI testing showed that the application was visually appealing and easy to use, which helped to increase user engagement and satisfaction. The findings from the UI testing were used to make further improvements to the application, further enhancing the user experience.

6.2.9 Regression Testing

Regression testing was conducted on the application. The purpose of regression testing was to verify that the recent changes and updates made to the application did not negatively impact its existing functionality. A comprehensive suite of test cases was executed to cover all major and minor functionalities of the application. The tests were conducted on both iOS and Android platforms to ensure that the application worked as expected on both devices. The results of the regression testing showed that the changes made to the application had not resulted in any unintended consequences, and the application continued to function as expected. This helped to maintain the high level of quality and reliability of the application and ensured a positive user experience for the end-users.

6.2.10 UAT Testing

UAT (User Acceptance Testing) was conducted on DropTrip, a tour management application that is available on both iOS and Android devices. During UAT, a group of end-users was tasked with testing the application to verify that it met the required business and functional requirements. They carried out a range of tests, including navigating through the application, booking tours, and managing itineraries, among others. The results of the UAT tests were used to identify any bugs or issues, and the development team was able to resolve them before the final release of the application. This helped to ensure that the end-users received a high-quality, bug-free product that met their needs and expectations.

6.3 Test Scenario

Test Scenario 1: Login as a User and admin

Testing Objective: To ensure the login and signup functionality is working correctly

No	Test Case	Data	Expected Result	Result
01	User Login by entering the username and password	Username: usamanjc@gmail.com Password: test1234	User Successfully logged in to the system	Pass
02	Admin Login by entering the username and password	Username: admin@gmail.com Password: 12345678	Admin Successfully logged in to the system	Pass
03	If the User has not registered himself, he needs to be registered first by signup	Enter Name Enter Password Confirm Password Enter Email Address	Successfully Registered as a new user	Pass

Test Scenario 2: Show Trips inside the locations screen.

Testing Objective: To ensure the trips are display correctly.

No	Test Case	Data	Expected Result	Result
01	User comes on the home page and clicks on the locations and then select the province of his own choice to view the trips.	N/A	User Successfully get information	Pass

Test Scenario 3: Show Information about the Agents

Testing Objective: To ensure that the agents are displayed correctly.

No	Test Case	Attribute and Value	Expected Result	Result
01	User comes on the home page and clicks on the agents and can view all the registered and approved agents.	N/A	User Successfully gets information	Pass

Test Scenario 4: Admin approves the uploaded trips

Testing Objective: To ensure that the admin approves the trips.

No	Test Case	Attribute and Value	Expected Result	Result
01	Admin view a posted trip by an agent and can approve/deny the trip	N/A	Admin Successfully approved/denied a trip	Pass

Test Scenario 5: Admin Update disease list

Testing Objective: To ensure that the list is updated.

No	Test Case	Attribute and Value	Expected Result	Result
01	Admin select the category in which he want to update the list	Disease Information Precaution	Admin Successfully updated a list	Pass

Test Scenario 6: Logout as a User and admin

Testing Objective: To ensure that the user and admin successfully logout from the system

No	Test Case	Attribute and Value	Expected Result	Result
01	Click the exit button and logout	Logout	User /Admin Successfully exit from the system	Pass

6.4 Automated Testing:

Tools used:

Tool Name	Tool Description	Applied on [list of related tests cases / FR / NFR]	Results

Appium	Appium is an open-source tool for automating native, mobile web, and hybrid applications on iOS and Android platforms. Testing the DropTrip mobile application with Appium involves creating and executing automated test scripts that interact with the application UI elements and verify its functionality.	Sign Up User Login Admin Login Post Trips View Trips Booking Switch To Agent Approve/Deny Trips Approve Agent Requests Log Out	Pass

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion:

In conclusion, DropTrip is an innovative and user-friendly tour management mobile application built using React Native, offering a seamless experience for both iOS and Android users. With its advanced features and functionalities, DropTrip is the perfect solution for tour management, making it easier for travelers to plan, book, and manage their trips with just a few taps on their mobile devices. Whether you're an individual traveler or part of a group, DropTrip is the ultimate travel companion, providing you with everything you need to make your trip as enjoyable and stress-free as possible.

Advantages:

- Cross-platform compatibility: DropTrip can run on both iOS and Android devices, which means it can reach a larger audience.
- Faster Development: React Native enables faster development cycles compared to traditional mobile app development, reducing the time to market.
- Better User Experience: The use of native components in React Native ensures a smoother and more responsive user experience.
- Cost-effective: As a single codebase can run on both platforms, it reduces the development cost compared to developing separate apps for iOS and Android.
- Ease of Maintenance: The single codebase makes it easier to maintain and update the app, reducing long-term costs.
- Community Support: React Native has a large and active community, providing continuous support and updates.
- Seamless Integration: React Native allows seamless integration with native functionalities like camera, GPS, and more, providing a richer user experience.
- Performance: React Native provides good performance, delivering smooth and fast app experiences for the users.

7.2 Future Work:

To improve the overall user experience, DropTrip could consider incorporating new technologies such as integrating with popular travel and payment apps could streamline the booking process and make payments and expense tracking easier. Another feature that could be added is the ability to access real-time travel information, such as weather updates and local events, to help users plan and make the most

of their trips. Finally, adding support for virtual and augmented reality-based destination previews can enhance the app's value for both leisure and business travelers. By leveraging the latest technologies, DropTrip can continue to set itself apart as a top-of-the-line tour management application for iOS and Android devices. Here are some potential updates that can be made to a React Native app like DropTrip:

1. **Advanced Booking System:** Implementing an advanced booking system with features like booking history, booking status, and payment management to improve the user experience.
2. **Real-time tracking:** Adding real-time tracking of tour groups and guides to provide live updates and improved safety measures.
3. **Personalized Recommendations:** Adding personalized recommendations based on the user's past tours and preferences to enhance the user experience.
4. **Multilingual Support:** Adding support for multiple languages to expand the app's reach to a global audience.
5. **Chat and Support:** Implementing in-app chat and support functionality to provide quick and efficient customer support to users.
6. **Integrations with Other Services:** Integrating the app with other travel and booking services to provide a one-stop-shop experience for users.
7. **User Feedback:** Adding features for users to provide feedback and ratings to help improve the app and tour offerings.

These are just a few examples, and the updates can be customized based on the specific requirements and goals of the app.

CHAPTER 8

REFERENCES

References

[1] Kopparapu, S. K. (2008, November). Natural language mobile interface to register citizen complaints. In *TENCON 2008-2008 IEEE Region 10 Conference* (pp. 1-6). IEEE.

[2] Gupta, D. N. (2006). Citizen-centric Approach for e-Governance. *available at: www. csisigegov. org/1/5_392. pdf (accessed 30 December 2013)*.

[3] Maheen, F. F., & Sumithra, M. D. (2018). Development of Smart Complaint Portal based on Geotagging and Proximity Search.

[4] binti Aziz, A., Baharudin, A. S., & Karkonasasi, K. (2016). ADUN e-Community Portal: eComplaints for Community Service. *International Journal of Applied Engineering Research*, *11*(6),

4214-4218.

[5] scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1058&context=cseng_senior