Wanderlust Wheels
Bike Rentals

by

Jarin J V (2348031)
Mohammed Yasir Khattib (2348043)

Department of Statistics & Data Science

CHRIST (Deemed to be University)

April - 2024

ABSTRACT

Bangalore is an urban city which has people from different parts of the country. People staying in these places would require bikes to travel, be it within the city or on trips outside the city. While there are several bike rental providers in the city in different locations, there are certain issues and details which exist with some of these providers. The aim of this project is to build a bike rental interface (mobile application) which allows users to book their bikes from the different available garages of the rental service in the city. From these options, the user will get to choose and book. In addition, it addresses some of the issues which are predominant in the existing ones like the option to pick up in one location and drop in another, providing details to the user about the bike such as the age of the bike and mileage details which allow users to take better decisions and provide support to the users. The aim is also to provide a mechanism which helps on the company side and makes sure that a fair decision is done on the user's side. This will be useful during the case of accidents which will ensure that there is a good mechanism to ensure that a right decision can be made Additionally, the bike rental interface will feature intuitive pictorial and graphical representations to enhance user experience and facilitate informed decisions. Users will have access to visual displays showcasing the most rented bikes, aiding them in making popular choices based on community preferences. Moreover, the app will offer comprehensive comparisons of bikes based on mileage, allowing users to visually assess fuel efficiency and make eco-friendly choices. This graphical representation will include easy-to-understand charts and graphs illustrating the comparative mileage of various bike models available for rental. Furthermore, the interface will incorporate visual indicators for bike age, enabling users to quickly identify newer or well-maintained options. This visual representation will assist users in selecting bikes that align with their preferences for reliability and performance. By integrating these pictorial and graphical elements into the application, we aim to provide users with a visually engaging and informative platform. This approach not only enhances user interaction but also supports the company's efforts in ensuring transparency and facilitating well-informed decisions for a seamless rental experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter contains an overview of the overall project and how the project will be beneficial for the users and society as a whole.

The introduction comprises of several sections as mentioned below:

1.1 Project Description

The project aims to create a bike rental interface which allows the user to book bikes from the various garages of the company present in the city of Bangalore. Since Bangalore has people from different parts of the country, especially as students or working professionals, they will not have their own bikes. Thus, the need for them to be able to rent bikes to commute from one place to another within the city or to rent bikes for the purpose of going on trips is very high. The bikes they would require will vary from one individual to another based on the distance they wish to travel, the skill level of driving bikes, as well as the interest and passion of driving bikes. For instance, a person who wishes to just commute within the city and who is relatively old and uses bikes just for purposes of moving around would be happy to use a bike with a lower engine capacity of around 120-160CC and sometimes would even be satisfied with a scooter. However, if a person has a passion towards bikes and wishes to drive really good bikes with higher capacities and would like to hit the road over longer distances might want to rent a really good sports bike with probably even a fancy look. Thus, it is necessary to be able to provide to the customers various options to choose from. In addition, during the process of booking, the user would love to be given different important information and details which would help him make a choice about which bike he/she wishes to choose to satisfy their needs in the best possible way. The process of renting a bike involves booking through the website/app, submitting necessary details, getting the bike from the garage after completion of formalities and submission of documents, and returning the bike back to garages. The users would also prefer the entire process to be made simpler. It is important to provide to the users, a simple, yet effective interface for completing all these procedures by making the overall process of

renting as minimal and user-friendly as possible. This project aims to provide exactly that to the user.

This software would provide the user the chance to book their bikes through the mobile app. The application would both be user-friendly and would provide necessary details to users in the best possible way to improve their overall experience of using the software. At present, there are many companies which have their own interfaces for renting out bikes to users in Bangalore. These include companies like Bykemania, Twist Throttle, Royal Brothers, Onn bikes and so on. Some of these run their operations only in Bangalore while others have their garages in various capacities in different parts of the country. However, there are some limitations with the existing ones. Some of the interfaces are not very user friendly and takes time to understand and completely process on how to use it to satisfy the needs.

Most of the companies do not provide the user the option to pick up their bikes at one garage and drop it at another garage in the city. While this is done from the company side to avoid complications and to prevent the situation where there are more of the same bike in one location as compared to another, it is a particular option which the user would prefer. Especially in a city like Bangalore, where traffic is as prevalent as oxygen, it is very much needed for users to be able to change their drop location and drop their bikes at their nearest garage before their drop time in order to ensure they are not charged any extra amount as fine for dropping the vehicle late.

Sometimes, there could arise issues such as accidents in the vehicles and during this time it is necessary that there is sufficient proof about the condition of the bike at the time of renting so that the right amount can be charged from the customer. Both the customer as well as the company requires this proof and this proof on the company side has to be made available at all garages so that it is possible to check the condition in case the customer returns the bike at a different location from the pickup. While a lot of the companies now take minimal photos before and after the renting and check the conditions, this is very specific to the garage and thus the condition cannot be checked at other garages. Also, not complete photos are taken and hence sometimes the exact amount of fine to be charged cannot be decided.

As mentioned earlier, customers would love to have more details about the bike before finalizing which bike they want. The companies at present provide very minimal details, covering limited aspects such as the capacity and concentrate more on the business-related details of it such as the prices. The users do not get to know more details such as the age of the bike, when it was last serviced and the exact model number which could enable them to search up more details on the Internet if needed. They do not get access to the reviews of past users about the bike and the comfort, and their experience of driving the bike. Past reviews would also help the customer know more about the actual mileage of the bike and the expected mileage which the bike would provide. This would enable the users to plan their costs and know how much they might need to spend on fuel.

Once the customer rents the bike, he would want his bike to be confirmed along with the number plate and other details because that gives the customer a proper confirmation and doesn't lead to the bike getting changed in the last-minute barring unexpected circumstances. This is needed because if a particular user books a bike which is around 2 years old, he would expect to get the exact bike rather than getting a different bike which is around 5-6 years old as it changes so many aspects of the bike. At present, the existing providers just provide confirmation and sometimes that bike might not even be available for pickup and that leads to last minute difficulties for the customers. Thus, being provided confirmed details would give the user a sense of faith. Also, details such as the RC and other related aspects which could be asked if stopped by the Police for checking need to be provided to the user. All this, if made available in the application to the user, will help the user to tackle situations such as being stopped by traffic police, or accidents or other related issues.

Sometimes, users may have a particular garage of the company near their place and might want to rent a particular bike from there on some date and time. However, due to circumstances and availability, the bike might not be available for booking from that particular garage. It might be available only at some other garage. Even if the user is willing to pay extra to get the bike at this location, it cannot be made possible with the existing providers. He will thus have to either go to the other garage or look for alternative providers to satisfy his requirements.

It is thus necessary that the customer is provided the best possible experience right from the time he/she enters the application to check out the bikes till the time the vehicles are returned. The enhancing of experience is possible by ensuring a simple, easily navigable interface which provides as many options as possible to provide ease to the user. At the end of the day, every business wants more customers attracted to their business and the more options they provide, the simpler it is for the users, greater the number of customers that will automatically be attracted to them and easier it is for them to retain the existing customers.

That is where the rental interface to be created as part of this project will help. It tries to address the existing problems and provide extra features. Users will now have an option to pick up their bikes in one location and drop the bikes in another location within the city, which as mentioned was not provided by the existing bike rental providers. In order to have a check on accidents and the level of damage, extensive photos taken of the bike at the time of renting will be uploaded onto a common database which is accessible in all the garages of the company across the city. This helps even when the user returns the bike at a different garage from that where he took the bike and thus he can be charged accordingly with sufficient proof of the condition of the bike before and after. Bikes will be made available to the next customer after return only after a certain gap. This is to ensure that sufficient checking of the condition of the bike is done and necessary minimal services are done to ensure that the next customer receives the bike in a good condition and not in the exact condition in which it was dropped. Customers will now have an option of getting bikes transferred from another garage to the garage near their place if they are very particular on renting from that particular garage. This will be done at extra charges to the user. Users will now have more details about the bike available to them at the time of booking such as the age of the bike, the last service and the actual and expected mileage from past user reviews of the bike thus helping them to make better decisions. After booking the bike, the RC and number plate details will be made available and confirmed which can be viewed by the user in the app. The app will also have details about how many kms he has traveled and how much he has left to travel before he is fined extra for driving beyond the distance limit.

1.2 Existing system

There are plenty of bike rental providers in Bangalore. While each of these websites has their own advantages, there are some drawbacks or setbacks that make the interface and process less user-friendly and uncomfortable for the user to book the bike and proceed with their extensive and cumbersome process. Some of the rentals and an explanation about their system are given below.

Site 1: Royal brothers

Royal Brothers is the first start-up to obtain a license to operate bike rentals in Bangalore. It doesn't matter if the user is looking for bike rentals in Bangalore for a short term or a long-term lease, users can rent vehicles to satisfy their diverse needs. They offer a wide variety of vehicles starting from scooters such as Activa, Dio, Aprilia SR, TVS Ntorq, Ather, Aviator to powerful bikes such as Royal Enfield Classic 350, KTM Duke 200, BMW, Yamaha Fazer etc.

Being a start-up does not mean that they do not have enough bikes. Royal Brothers have raised a fund of 1.3 Million in the third round and they have nearly 16 garages across the city from where users can rent their bikes. Those places include Indiranagar (Near metro station), Yeshwanthpur (BMTC Bus Station), Koramangala (Near Oneplus Service Centre), Bangalore REX hub and they give employment to over 250 people both part time and full-time. Royal brothers also provide Intercity rentals.

The main advantages of this company that make people rent their bikes from Royal brothers are that they don't compromise on the quality of their bikes. They don't use bikes that are more than 3 years old, and they do ensure that they maintain the bike condition by performing frequent services. They do provide the option for the users to purchase safety gears such as helmets, riding jackets, shoes and gloves on rental bases. They also have a wallet option in their interface, so that the user can use their money for future bookings.

In spite of having the above-mentioned advantages, they also have some disadvantages which make it cumbersome. Some of those are, the sign-up page takes a lot of details which are not very necessary and it takes a lot of time to verify the required documents

such as Aadhar, driving license etc. The users who take rented bikes from a particular location/garage need to return the bike to the same location/garage.

Site 2: Bykemania:

Bykemania is another famous bike rental company that will allow you to rent a bike for as less as 10 Rupees per hour and their vehicles are available at relatively nominal amounts. Bykemania makes it more convenient for people to rent motorcycles of all types and ranges starting from a simple scooter to a super bike like Dominar 400. They also offer exciting offers and other premium privileges.

This wide range of choice from a bike rental can perfectly satisfy the customer's needs be it for a small trip or a long ride. Bike on rent Bangalore is a boon for ardent bikers as they can choose from a variety of different bikes starting from economical models to medium powered bikes to cruisers to seriously powerful super bikes like the Himalayan. Like different varieties of bikes, they have different locations where they are active, the active areas in Bangalore are Koramangala, BTM Layout, Majestic, Jayanagar, etc.

With minimal processing steps, easy verification of documents, convenient pick-ups and drops, Bykemania is a customer friendly, passionate and honest bike rental in Bangalore. They give some consideration for the late return of the bike for a period of 30 minutes.

They really have very good staff but the bikes' conditions are not always in the best condition. Also, they don't provide helmets or any rider equipment for the riders who are renting the bike which makes it uncomfortable for the users to get the helmets to the renting place and their location. It also leads to a situation where the user has to rent the helmets and other riding equipment from other places which create excess work for the customer. It is also difficult to identify the rental area as they function mostly in the underground or basement of buildings. Again here, they do not allow users to pick-up their bikes in one location and drop it in another.

Site 3: Twist Throttle:

Twist Throttle is a bike rental legacy in Bangalore that started as an idea to assist bikers on their trips/stays by serving that extra mile on their motorcycling journey. Based out of Bangalore, Twist Throttle is renowned as a premium bike rental company with an array of varied rental options in two-wheelers. Whether the user wants to enjoy the thrills of a sports bike, dream of riding a Royal Enfield and various other premium segment bikes or prefer the ease of driving a scooter, Twist Throttle offers the customers options in every choice they make. With Twist Throttle the user can rent a bike in Bangalore with a hassle-free experience.

Twist Throttle is new start-up which serves only in two locations in Bangalore, namely BTM layout and Shanthi Nagar but they have all kinds of sports bikes and superbikes upto 1000cc which is a dream for every bike enthusiast. The procedure for renting a bike in Twist throttle is first you book the bike on their website. Then the Driving License will be verified in original. Original ID proof (Passport, Voter ID, Driving License, College ID, Office ID) needs to be deposited. Passport needs to be deposited for bikes above 500cc (Mandatory). A lot of this makes it cumbersome for the users. One Helmet will be provided complimentary.

The user experience while booking is much more interactive while booking through their website is much more interactive and the procedure is way simple and the renting is much faster than other rentals in Bangalore. They also have very good customer service but those are only during the working hours through call. There is no app extension or bot to contact the customer service. Twist Throttle also doesn't have any mobile app, and users need to sign in every time they wish to rent the bike and also the signup also takes up different details that are not very necessary for the bike rental company. Moreover, they also track the bikes and the riders which does affect the riders' privacy.

There are many more bike rentals in Bangalore that are not that very popular in Bangalore. Most of these do not have any website or app and the only way to book a bike from that rentals are through call or in person to the rentals which is not possible for all the users across the city to go and book and then go and collect the bike. The users might not have any idea about the availability of the bikes in the garage.

Some of the common issues that can be seen with most of these service providers is that they do not provide the option to pick up in one location and drop in another. Also, for the company side, there is no common interface where, if the photos are uploaded by staff of one of the garages, it is not visible at other garages. Also, there are potential issues of unavailability of bikes after booking and this is due to full confirmation not being given to the users after booking. The users also do not have many details made available to them at the time of booking and this leads to them sometimes not being able to make the best possible choices to suit their interests.

1.3 Proposed System

The proposed system strives to provide features to the user which they are lacking and unavailable to receive with the existing bike rental providers. This would include:

1. A simple, yet efficient process of booking and renting the vehicle
2. Allowing the user to choose from the different bikes after extensive comparison and checking
3. Option to pick up bikes at one location and drop them at another
4. Option to obtain rental gears along with the bike
5. Providing Statistical visual representation of the vehicle based on mileage, performance , bike popularity and user feedback on bike condition.
6. Collection of minimal, yet necessary documents for verification
7. Giving to the users, the necessary details about bikes after booking

1.4 Objectives

This bike rental service differs from the others because it offers additional features that make it more feasible for users. These features include the option to pick up and drop off the bike at different garage locations, getting the details of the actual bike before renting it, and detailed insights that help users make the best choices.

Features of this interface includes:

- The ease of pick up and drop off garage locations
- Providing of actual details of the bike to the customers after booking confirmation
- User friendly

Providing these features to the users via the interfaces created as part of this project is the main objective.

There are many feasibility issues when looking for a pick-up and drop-off garage location. One of the most important factors is the distance between the two locations. Another consideration is the traffic patterns in the area.

This bike rental service provides the feasibility of pick-up and drop-off at garage locations. This would be an excellent service for tourists and city residents who want to explore the city on a bike, but don't want to have to worry about having to travel far to reach their drop location on time.

The bike rental service provides potential customers with a detailed description of the bike they are renting. This allows customers to be well-informed about the bike they are renting and to be aware of any potential problems that could occur. The user can also know how much kilometer or miles is driven by the provided interface. Providing the RC digital information and other details can be helped which are necessary details for the police. By being knowledgeable about the bike such as the age of the bike, the last service, the last usage, the weight of the bike, the size of the frame, and the type of brakes and gears, so that customers can feel confident in their choice and be more likely to have a positive experience.

The bike rental service provides a gap in time availability for the bike to undergo service checking and receive necessary services. This will ensure that the bike is always in top condition and can provide a great experience for the customer.

Our bike rental service provides a user-friendly interface that makes it easy to find the perfect bike for your needs. With a variety of bikes to choose from, we have the perfect rental for everyone. Whether you're looking for a leisurely ride around town or a more challenging mountain biking experience, we have the perfect bike for you.

rental providers. This will create a more relevant experience for users when interacting with the interface, and ultimately lead to a more successful business.

1.5 Purpose, Scope and Applicability

1.5.1 Purpose

First and foremost, the main aim of this project is to provide an upgrade on the existing bike rental providers available in the city. As mentioned earlier, there are several issues with the existing systems and addressing these to provide a better interface for the users would improve the overall experience of renting the bike for the users.

The present system sometimes becomes very cumbersome to the user with the extensive details they ask with a lot of them being unnecessary. The overall process of renting is also very long starting from the time they have to book bikes on the website till they go to rent the bike and return it back. Having features like picking up and dropping in different locations will simplify things for the user and provide more convenience for them. With the induction of this interface into the society, the users will be able to experience a much simpler and more efficient system as compared to the other existing providers.

1.5.2 Scope

With the existing system insufficient to satisfy the needs of the customers in the best possible way, the proposed system would help in being able to fulfill the needs of customers across Bangalore. The issues with the existing systems had been mentioned in detail earlier in the report and the need to deal with them was stated. With all that in mind, with the help of this project, the objectives can be achieved.

For the accident check, it was mentioned that extensive photos will be taken of the bike at the time of renting from the garage and this will be uploaded on the app. Since this application is linked to a common database, the same will be available at the other bike rental providers as well and thus it enables the user to be able to return the bikes at a different location from where it was rented.

Also, the database with the details of all bikes will be available and from this, the user will be made visible the details which are necessary. Some of the details will be made

available before booking the bike such as the features of the bike and after the booking, the more specific details such as the number plate and so on. Further, from the same database, the user will be given RC details of the bike at the time of coming to pick up the bike. All these will be made available to the user on the app on his mobile. So, the uploading of details will be from the database onto the user's side of the application. The overall database will have a column for the garage at which the bike is located so based on the location required by the user, the particular rows will be filtered out and taken and uploaded onto the website for the user to view.

The users will also get a view of the peak hours of different bikes. This will be done with the help of data from the past bookings and by getting comparative graphs of the demand for bikes at different timelines in a day. The bookings, once confirmed, will be uploaded onto a separate database for the bookings and from this, the details about particular bikes can be analyzed and peak hours will be made known to the user allowing him to take a better decision. Knowing whether the time at which he is looking to book the bike for will be a peak hour or not will enable him to take a better decision as to whether to book the bike immediately or hold for sometimes before finalizing his booking.

1.5.3 Applicability

The bike rental service provides an easier and effective way to rent a bike so that users can make good decisions in choosing the bike. There are many applications that make the interface interactive and give relevant experience to the user and also help in contributing to society.

Provides a sustainable alternative for short-distance trips:

One of the common applications is to provide a Sustainable Alternative for Short-Distance Trip. With a growing population and an increasing demand for transportation, it has become increasingly important to find sustainable transportation solutions. Our bike rental interface is one way to help meet this demand.

Self-explanatory user interface:

A bike rental app should have a user-friendly interface that is easy to understand and use. Users should be able to understand and use it easily without help or support. This will enhance the experience and make the app more likable.

Compatible price:

This interface is very reasonable as it does not charge any extra money from the user. Everything it earns is through commissions and advertising and does not charge any additional fees to the users.

Environmental and sustainability implications:

Bikes in general lead to a lot of carbon emission. Here, the bike rental interface provides gaps in the availability of bikes to check and maintain them in good working order. This keeps the bike in good condition as it is serviced at each time interval, thus reducing the emission and thus helping in reducing pollution.

Provides details and features available in current rental services

Bike rentals provide details such as availability, changing peak hours, and suggestions based on distance. One user might want to rent a bike to go to work, and another user might want to rent a bike for travelling. The choices are different. Some users feel comfortable on any bike, while others want exactly the same bike they reserved for Superbike. So, providing details and suggestions will help us meet demands and requirements. A bike can be rented at a low cost. Since they have the option of looking up and comparing the prices, they get the benefit of choosing the bike according to their budget. As a result, users can choose from a variety of bikes at low prices, expanding the range of responses to business needs.

Friendly interface

Who wants to rent a bike where the user can't cancel the ride in case of an emergency? Well, it's annoying for everyone. Therefore, one of the features that contributes to the success of this apps is the ability for users to cancel a ride at any time by notifying the

bike owner in advance of the cancellation, and the refund should be processed immediately.

1.6 Tools used

The following tools will be used for the development of the interfaces. An explanation of these tools as well as why it is feasible and the advantages of the same have also been mentioned.

Android Studio:

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and Github integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

Android Studio uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis. Applications built in Android Studio are then compiled into the APK format for submission to the Google Play Store.

Android Studio is well equipped for fast-paced development while ensuring high-quality output of apps across all Android devices. Being powered by the IntelliJ IDEA, this IDE provides fast code completion time and instant evaluation of the workflow. The Android Studio comes with an Emulator that helps start the entire app faster than the actual device. The emulator, by allowing you to test the app across multiple devices, including phones, tablets, Android Wear, and Android TV, can simulate several different hardware features like GPS, multiple touch inputs, motion and acceleration sensors, etc.

Android Studio offers many testing tools and frameworks to help test Android apps by using functional UI testing tools. The advantages of Android Studio are also shown with all kinds of advanced testing tools and frameworks for every purpose. The tests carried out by these tools can be on actual devices, emulators or robust integration environments and by using the Firebase Test Lab. Android Studio offers a robust repository of handy tools and established best coding practices allow better collaboration and teamwork in app development projects.

Programming languages

Android App are mostly developed in JAVA language using Android SDK (Software Development Kit). Other languages like C, C++, Scala etc. can also be used for developing Android App, but JAVA is most preferred and mostly used programming language for Android App Development.

Java**:**

JAVA is a programming language which is used in Android App Development. It is class based and object-oriented programming whose syntax is influenced by C++. The primary goals of JAVA is to be simple, object-oriented, robust, secure and high level.

JAVA application runs on JVM (JAVA Virtual Machine) but Android has it's own virtual machine called **Dalvik Virtual Machine (DVM)** optimized for mobile devices.

Advantages of java in android studio:

The main objective behind Android development was to create a platform-independent application environment that can run on every device. As we know java already have this quality so java was chosen for android development. Android applications run on a special virtual machine called the Dalvik VM that is a direct inspiration from java virtual machine called JVM. Android application can run on any device where special Dalvik VM is implemented. These way android applications are compiled and run in optimum performance environment with the feature of platform independence.

The good approach towards software development is the object oriented approach. Java is based on the oops concept. Android relies heavily on Java fundamentals like classes and objects and its other useful features of oops.

Java has an extensive set of libraries. It is easy to take advantage of these libraries. Android SDK has many standard Java libraries included. These provide functionalities for data structure, math functions, graphics implantation, and networking functions and much more. These java libraries help us to do everything else we could want. This way java helps develop Android applications fast and inefficient manner.

Android is made to run on different platforms i.e. hardware platforms. Thus architectural neutrality is desired and necessary. Android code is written once and to execute need to compile and optimise native code for better performance on various devices. Java has platform independent feature so it is used for android development.

Java is very popular language due to its awesome features and performance. The community of Developers those have proficiency is really big. Thus android developers to choose java as there is already a good base of java programmers are available that can help in creating, improving android applications plus with many libraries and tools of java make developers life easier. Large java developer base enables to develop a lot of android apps fast so it is based on java.

Developers those do not use java have to deal with serious problems like memory leads and bad pointer usage. Sometimes these problems harm at the highest level like the crash of application or crash of OS. Android easily implement and fix common problems with other programming languages with the help of Java. These are some problems those never occur when you program with java. Java is machine independent and runs only in JVM space so it protects you from these problems.

For the UI: XML:

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. It is derived from Standard Generalized Markup Language(SGML). Basically, the XML tags are not predefined in XML. We need to implement and define the tags in XML. XML tags define the data and used to store and organize data. It’s easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it’s a lightweight markup language that doesn’t make layout heavy. XML only contains tags, while implementing they need to be just invoked.

1.7 Overview of the report

The overall contents of the report will involve the entire process of building and testing the software with several stages involved in it. The process will be divided into chapters based on the stage of the software development

Chapter 1 will involve a detailed introduction about the project. This part will explain the limitations and issues existing in the present system and how this project will be helpful to overcome the existing issues and give the users a better experience. It will also briefly explain about the various rental interfaces existing at present in the city and will talk about how this will be an upgrade over the other systems and its applicability in society.

Chapter 2 will involve conceptual models such as the entity relationship diagram, the data flow diagram and other building blocks of the overall software.

Chapter 3 will involve in-depth details about the database and its structure and explanations about the various attributes as part of the database.

Chapter 4 will involve the coding aspect of the model and the actual implementation of the website and the application.

Chapter 5 will involve the various test cases and in-detail testing about various aspects of the model.

Chapter 6 will finally conclude about the overall model and its implication to the society and how it'll be advantageous to the society. It will also involve certain aspects about the implementation and the limitations faced while building and implementing.

CHAPTER 2

SYSTEM ANALYSIS AND REQUIREMENTS

2.1 Problem Definition

More than a problem as such, this project aims to provide an upgrade on the existing bike rental systems in the city. This will be achieved by addressing the issues which were mentioned before. Any website has to be very user-friendly and the overall process of providing service must be smooth and must give the user the best possible experience. This is what will ensure that the users come back after the first time. Especially when we consider a service like providing bikes for rent, it is necessary that the vehicles are in the best possible conditions and the only responsibility that the user should have should be to drive the vehicle and bring it back safely.

The entire process which includes several steps like the booking of the bikes online and this will involve the user to sign up first and then search for his preferred bike. While booking, the user will want to know many important details about the bike he is going to rent and all these have to be provided to allow the users to take the best possible decision with respect to booking their bikes. After completion of booking, he will have to go to the booked garage to collect his bike and will need the bike to be in the best possible condition and properly serviced. They would also require the bike details to be made available to them and accessible on the application so that they can use it in case of any issues.

Thus, the overall problem can be divided into smaller ones so that one problem at a time can be concentrated upon.

<u>Core requirements</u>

1. The basic requirements have to be satisfied first which is that the users should be able to reserve their bikes for a particular date and time.
2. They should have complete details of the bike before booking including the age of the bike so that they can take a decision on the best possible bike they wish to rent.
3. User-friendly Interface: The interface should be easy to navigate and use, with clear instructions and intuitive design. It should have a simple and

straightforward booking process that allows users to browse available motorbikes, select their preferred model, and book their rental period.

4. Pricing and Payment: The interface should have transparent pricing options andallow users to pay securely and conveniently. It should have a clear pricing structure that includes the rental fee, insurance, and any other additional chargesor fees.
5. Admin should get clear instructions on what time and how much before the bikehas to be kept ready at which garage so that there is no time waste or any otherissues for the users when they rent.

Additional Requirements

1. Multiple Pickup and Drop-off Locations: The interface can provide multiple pickup and drop-off locations, such as in a different garage as compared to the pickup, to make it more convenient for users to rent and return the motorbikes on time.
2. Data-Driven Insights for Optimal Choices: We are taking user experience a step further by implementing a unique data visualization system powered by statistical analysis. This system will present graphical representations of our bike fleet, showcasing key metrics like:

    - Bike Popularity: See which bikes are most in demand, helping you choose a reliable option.
    - Mileage: Get a clear picture of a bike's wear and tear to make an informed decision.
    - User Feedback on Bike Condition: Gain valuable insights from other riders regarding the bike's performance.

The project will first aim at solving the satisfying the core requirements which are the basic requirements before moving on to the additional requirements which would be bonus add-ons to the base project.

2.2 Requirements Specifications

Requirements Specification

In this section we will discuss different functions we will be performing to run our app successfully. Mentioning all the functions is out of this document's scope so we will talk about a few important functions:

Payment Function

Due to limited sources, adding a payment gateway is out of our scope. To tackle this situation, we will develop a dummy gateway page so that the user can book a bike and get the confirmation and get the full details of the bike in the application.

Time Interval function

This function is to ensure that the bike that has returned from a user needs enough time to get the bike serviced and cleaned and maintained for the next user. So that the bike also doesn't get damaged and the next customer can get a clean bike and doesn't have any issues while riding.

Hardware Requirements

For mobile app

This project will need a mobile phone to download and run the application. Students can use all the services of this application via phone itself. The android version of the phone must be greater than lollipop (version 5). The mobile application requires camera permissions to capture and upload the bike photos. Since the data is transferred to the database it requires Internet connection throughout the time. It is a small application that does not require huge amount of RAM but a of or above 4 GB RAM is advisable.

2.3 Block Diagram

Admin interface

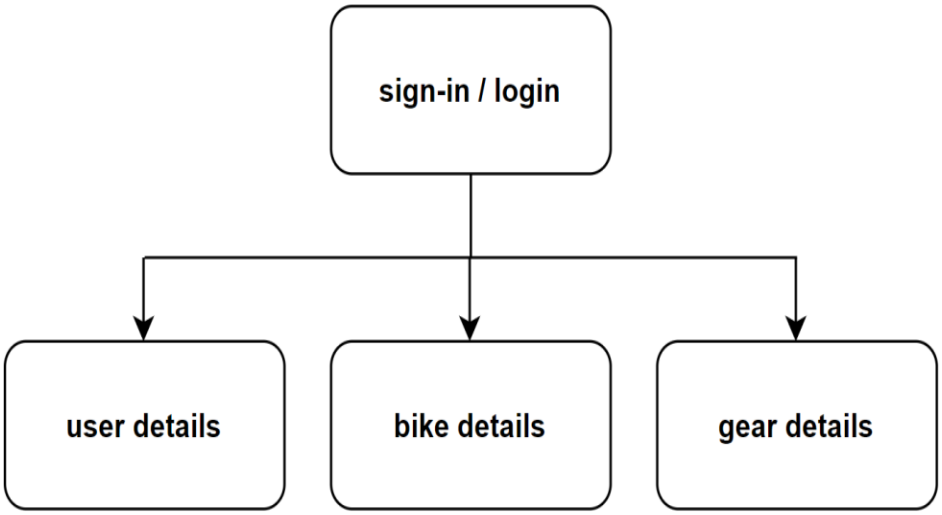

Image 2.1: Admin side block diagram

User interface

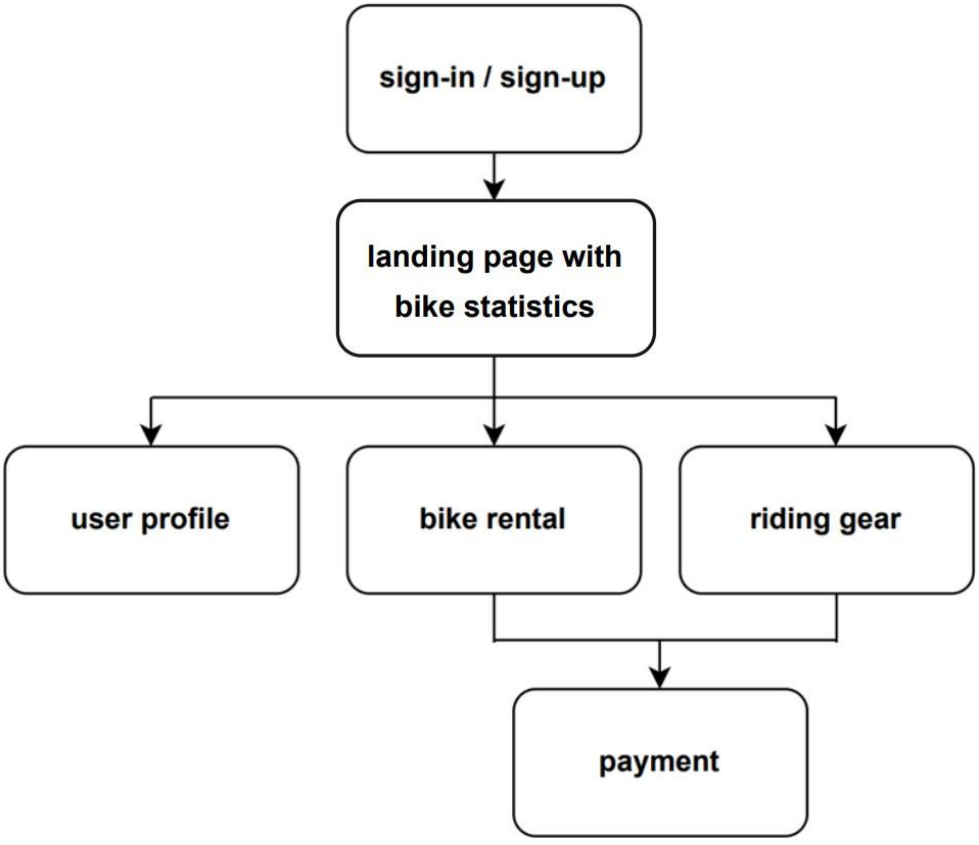

Image 2.2 User side block diagram

2.4 System Requirements

2.4.1 User Characteristics

The users of this interface could be of various kinds and the bikes could be rented for several different needs. Based on the needs and requirements, the different kinds of users can be classified as:

Tourists and Travelers: This user group may be interested in renting motorbikes as a convenient and affordable way to explore a new area. They may be looking for options that allow them to cover more ground and reach destinations that are not easily accessible by other modes of transportation. This user group may value features such as availability of different types of motorbikes, ease of use, and affordable pricing options.

Adventurers: Adventurers are individuals who enjoy outdoor activities and seek out new experiences. They may be interested in renting motorbikes for off-road adventures or scenic drives. This user group may value features such as availability of powerful and rugged motorbikes, access to remote locations, and the ability to rent gear such as helmets and protective clothing.

Commuters: Commuters are people who need a quick and efficient way to get around their city or town. They may be interested in renting motorbikes as a cost-effective alternative to owning a car or taking public transportation. This user group may value features such as availability of motorbikes near transit stations or popular commuter routes, ease of use, and affordable pricing options.

Motorbike Enthusiasts: This user group consists of individuals who have a passion for motorbikes and may be interested in renting different types of bikes for different purposes. They may be looking for options that allow them to test out new models or customize their rental experience. This user group may value features such as availability of high-performance motorbikes, access to accessories and customization options, and the ability to rent bikes for longer periods of time.

Business Travelers: Business travelers are people who travel for work and may need a motorbike rental service to get around during their visit. They may be looking for

reliable and high-quality motorbikes that can be rented on a short-term or long-term basis. This user group may also value features such as availability of motorbikes near their hotel or conference center, ease of use, and the ability to reserve bikes in advance.

2.4.2 Software and Hardware Requirements

1. **Software Requirements**
    - **a. Software Tools Specification**
        - **i.** Visual Studio 2022
        - **ii.** Android Studio Electric Eel | 2022.2
        - **iii.** Firebase (Database, Storage, etc
    - **b. Languages**
        - **i. Android**
            1. Java
            2. XML
            3. JSON
            4. Python
    - **c. Database**

      User Table: This table contains a list of all the users who have signed up for the bike rental service. It includes user details such as name, email address, phone number, and password.

      Admin Table: This table contains a list of all the administrators who have access to update and edit the bike rental service's details, such as bike information and user data. It includes admin details such as name, email address, phone number, and password.

      Bike Table: This table contains a list of all the bikes available for rent, including their details such as bike model, availability, location, and milage rate.

Bike Status Table: This table contains a list of all the bikes' status, including their availability and location. It helps the users to know which bikes are available for rent and where they can find them.

Booking Table: This table contains a list of all the bookings made by the users, including booking details such as the user's name, bike rented, rental period, and the total kilometres driven.

CHAPTER 3

SYSTEM DESIGN

This chapter will cover about the overall architecture and design of the system. It covers about the various components of the project as a whole including the database and the different modules.

3.1 System Architecture

The bike rental interface will follow a 2-tier architecture. This is suitable over a 3-tier architecture due to the simplicity of the interactions between the client and the server.

Client Tier

The client tier is the user-facing part of the bike rental application. It consists of the mobile app that users interact with to browse available bikes, check their availability, make reservations, and manage rentals. The client tier's primary responsibility is to provide a user-friendly interface and to ensure that users can easily access and interact with the application's features.

The client tier typically includes the following components:

1. User Interface: This component is responsible for presenting the app's features and functionality to the user. It's typically designed to be user-friendly, visually appealing, and easy to navigate. The user interface allows users to browse available bikes, select bikes, and make reservations or rentals.

2. Input Processing: This component is responsible for processing user input, validating it, and sending it to the server tier for further processing. It ensures that the data entered by users is accurate and complete.

Server Tier

The server tier is the backend part of the bike rental application. It consists of the server-side application logic and the bike rental database. The server tier's primary responsibility is to process user requests, retrieve and store data, and ensure that the application runs smoothly.

The server tier typically includes the following components:

1. Application Logic: This component is responsible for handling user requests, processing them, and sending responses back to the client tier. It's also responsible for coordinating the interactions between the client and server tiers, ensuring that data is properly stored and retrieved from the database.

2. Database: This component is responsible for storing all the relevant data related to the bike rental application, such as bike information, rental history, user data, and other details. The database should be designed to handle high volumes of data and to ensure data integrity.

3. Security: This component is responsible for ensuring the security of the bike rental application. It's responsible for authenticating users, authorizing access to data, and protecting sensitive user data from unauthorized access.

Overall, the client and server tiers work together to provide a seamless bike rental experience for users. The client tier is responsible for providing a user-friendly interface, while the server tier handles the processing of user requests and ensures that data is properly stored and retrieved from the database.

3.2 Module design

There are several individual modules which have been put together to create the entire application. These have been explained below:

1. User Authentication Module: This module is responsible for authenticating users, managing user accounts, and enforcing access control. It is a critical module that ensures the security of user data and prevents unauthorized access to the application. The module includes features such as user registration, login, and account management.

2. Bike Inventory Module: This module is responsible for managing the bike inventory. It includes features such as adding and removing bikes, updating bike information, and checking bike availability. The module should ensure the accuracy of bike data and prevent data duplication or inconsistency. It should provide features such as bike search, filter, and sorting to help users find the right bike. The module should also provide features for managing bike attributes, such as bike type, model, and location,

to help users find the right bike for their needs. This is on the admin side of the application.

3. Booking Module: This module is responsible for managing bike reservations. It includes features such as allowing users to book their bikes in advance, managing reservation requests, and notifying users of successful reservations. The module should ensure the accuracy of reservation data and prevent double booking or overbooking. It should provide features such as date and time selection, bike availability checking, and notification of successful reservations.

4. Visualization Module: This module is responsible for managing the visualization of the bike on basic of mileage, performance and customer feedback on Bike condition plotted on the respective plots such as histogram , pie chart and so on,

5.

6. Billing Module: This module is responsible for managing billing and invoicing for bike rentals. The module should is a dummy payment gateway created for demonstration purposes.

These are the major building blocks of the project.
However, there are a few smaller modules as well which are explained below.

7. Document upload module: The user has to upload his documents onto the application. This includes the Digilocker verified copy of his/her License and Aadhar Card. Passport is an additional upload which will be needed in case the user is renting bikes of more than 500cc.

8. Photo upload module: This module is on the admin side which requires the admin to upload extensive photos of the bike before the booking is finalized and the bike is rented. This is necessary in order to check when there is a case of accident. The extent of the accident can be judged in a better way with more photos of the bike uploaded. The same photos will also be available on the user side of the application.

3.3 Database Design

3.3.1 Tables and Relationships

The main entities in the database are:

User Table: This table contains a list of all the users who have signed up for the bike rental service. It includes the following attributes:

1. Name
2. Phone number
3. Email
4. Password
5. User ID (Primary Key)
6. Date of birth
7. License
8. Aadhar
9. Passport (Optional)
10. Photo (Optional)

Admin Table: This table contains a list of all the administrators who have access to update and edit the bike rental service's details, such as bike information and user data. The main attributes are:

1. Admin ID (Primary key)
2. Password

Bike Table: This table contains a list of all the bikes available for rent.

The attributes are:

1. Bike ID (Primary key)
2. Garage ID
3. Model
4. Number plate
5. Rent per day
6. Brand
7. Fuel
8. Last service date
9. Mileage

Garage table: This table contains the list of the garages

The attributes are:

1. Garage ID (Primary key)
2. Bike ID

Relationships:

The user books bike is the relationship between the User and the Bike table. The Booking relationship has its own attributes which include:

1. Booking ID (Primary key)
2. Bike photos
3. User ID
4. Bike ID
5. Garage ID
6. Pickup time
7. Drop time
8. Kms driven
9. Total fare
10. Status

3.3.2 Data Integrity and constraints

1. Primary Key Constraints:

A primary key constraint is a type of database constraint that ensures that a unique identifier is assigned to each row in a table. It ensures that no two rows have the same identifier and that the identifier cannot be null. A primary key constraint is a way of identifying and distinguishing one record from another in a table. It is a fundamental component of relational databases and is used to establish relationships between tables. Here, the primary keys of the respective tables are:

1. User – User ID
2. Admin – Admin ID
3. Bike – Bike ID
4. Bike Status – Bike ID
5. Garage – Garage ID
6. Booking – Booking ID

2. Foreign Key constraints:

Foreign key constraints are a type of database constraint that ensures that a relationship exists between two tables. A foreign key is a column or combination of columns in one table that refers to the primary key of another table. A foreign key constraint ensures that the value in a column in one table matches the value in the primary key of another table. It establishes a relationship between the two tables, which is essential for querying and manipulating data in a relational database. Foreign key constraints are important because they help to maintain data integrity and consistency in the database. By establishing relationships between tables, foreign key constraints prevent data inconsistencies and help ensure the accuracy of the data stored in the database. They also help to enforce business rules and prevent invalid data from being inserted into the table.

Admin table: The garage ID is a foreign key here referencing the primary key of the garage table. Thus, there cannot be an admin with a garage ID not in the Garage table.

Bike table: Again here, the garage ID of the bike references to the primary key in the garage table and acts as a foreign key.

Booking table: The booking table has details about the Bike ID, User ID, and Garage ID. All of these are foreign keys referencing the primary keys in the Bike, User and Garage tables respectively.

3. Non-null constraints: Non-null constraints are a type of database constraint that ensures that a particular field in a table cannot have a null or empty value. In other words, a non-null constraint ensures that a value must be provided for a field when a new record is inserted or updated in the table.

Non-null constraints are important because they help maintain data integrity and consistency in the database. Without non-null constraints, it is possible for a field to have a null value, which can cause issues when querying or manipulating the data.

Here, for our database, most of the fields have non-null constraints except the optional fields like the Passport of the user and Photo of the User.

3.4 Interface Design

3.4.1 User Interface Design

The task was to develop a simple User Interface which is simple, yet efficient and self-explanatory. The navigation from one page to another in the application should be well-guided and it should not be a situation where the users have to search too much in order to find buttons for navigation.

In order to ensure that the overall Interface is simple and interactive, the following features were used as part of the project.

1. Card view – The card view improves the overall experience for the user by providing a very responsive page for the user while viewing. It helps in ensuring that the content is organized in a proper manner and also is visually pleasing for the user viewing it.

2. Scroll view – This is a feature which is very useful in mobile applications in particular which allows the users to scroll through the different items by scrolling and viewing them.

3. Font adjustments – Different fonts and font sizes have been used in the design in order to highlight more important information in the application.

4. Dialog boxes - Dialog boxes have been used in order to give messages to the user regarding the availabilities, successful sign-up and sign in and others.

5. Navigation – Buttons have been placed at the bottom of every page in the application whose symbols are self-explanatory and easy for the user to navigate. These will make it very easy for the user to guide themselves through to different parts of the application

6. Fragments – It is a feature which allows a single page to be divided into sub-pages with just simple features of scrolling between them. It is similar to the feature that is present on Whatsapp.

7. Animation – Some simple animation has been included in the landing page of the application which could provide a better interface for the user.

8. Search box – The search box allows for users to search not just based on the name of the bike but also based on any of the bike details including the model, fuel type and so on. This could compensate for the lack of a proper filter based on any other category other than the category of the vehicle.

Some screenshots of the User Interface from Android Studio

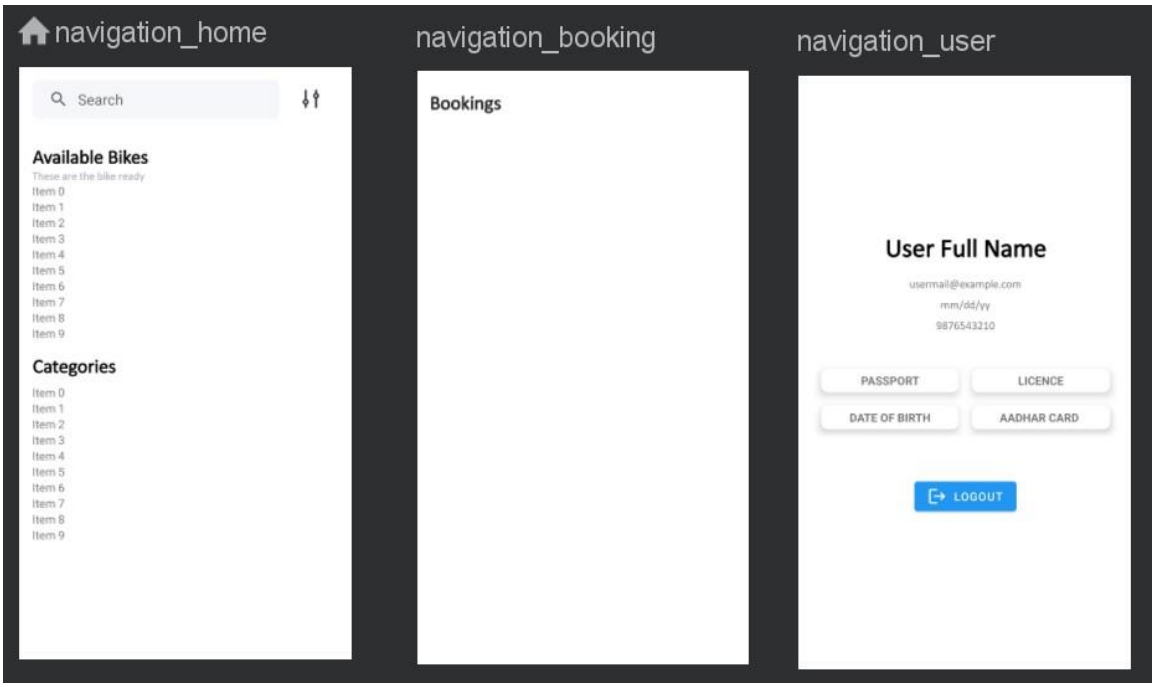

Image 3.1 Navigation to different pages on user side

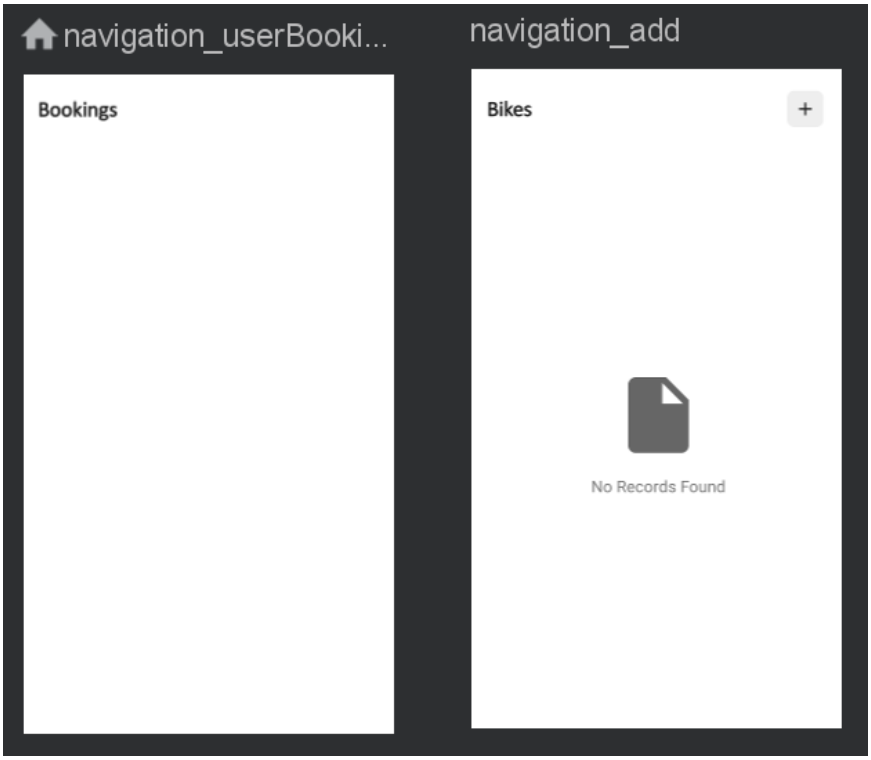

Image 3.2 Navigation on admin side

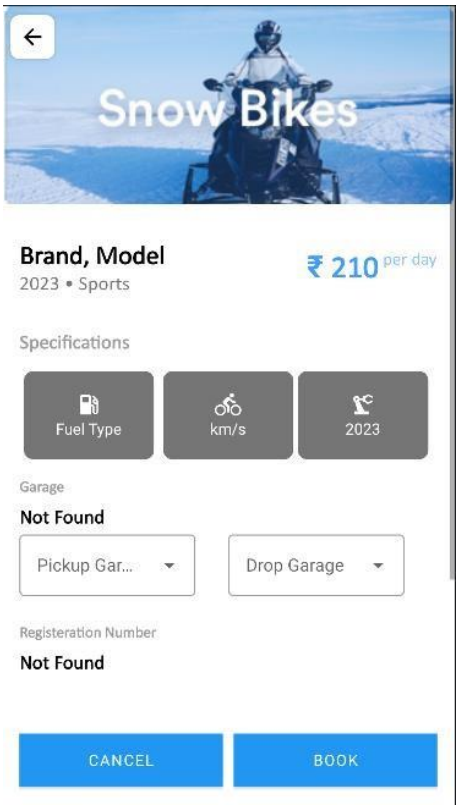

Image 3.3 Booking page design

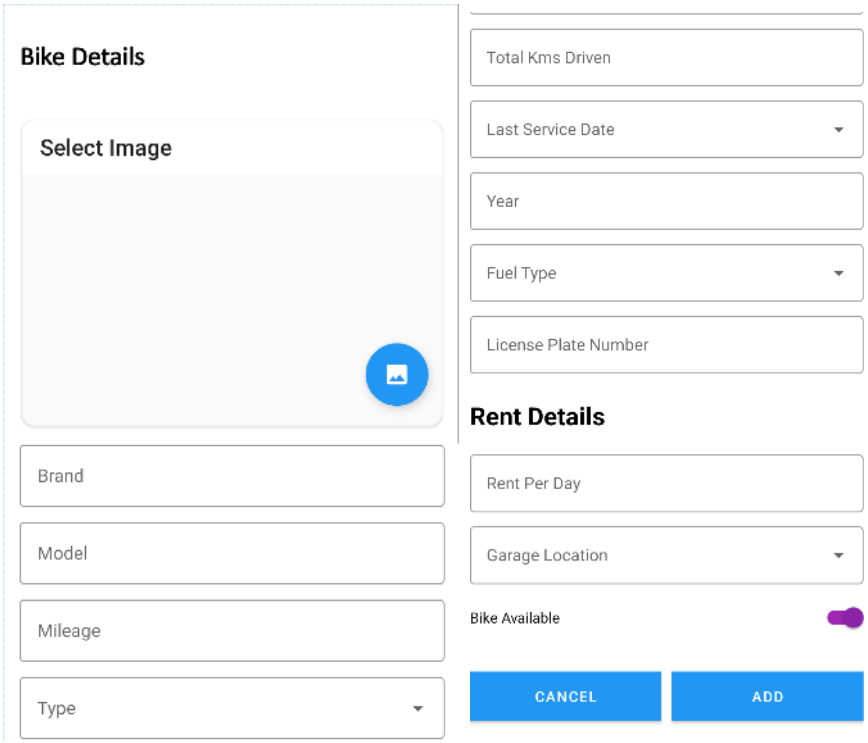

Image 3.4 Adding Bike UI in Scroll View

CHAPTER 4

IMPLEMENTATION

4.1 Implementation Approach

For the implementation we used the water fall model.

The model takes a linear approach towards project management with the project being broken down into sequences with the kickoff of a phase dependent on the completion of the preceding one.

This method primarily consists of 4 stages:

Idea Engineering – System Design – Implementation – Testing & Validation all these four are discussed below in detail.

Ideal Engineering:

The first step in the implementation approach for the Bike Rental app is to conduct a detailed analysis of the requirements. Through this we gathered information on the expected features, functionality, and user experience for the app. From his information we used to create a detailed project plan that outlines the scope, timelines, and resource requirements for the implementation process.

Technology Selection:

The appropriate technology for an android app was android Studio because it works on all devices ,we choose java as the programming language as it will suit for the implementation, frameworks, and libraries are chosen in a way that they are best suited to the requirements of the project.

Development:

The development phase we wrote was writing code for the app according to the design specifications. This will include front-end development, back-end development, and integration with third-party systems (firebase database). Here we tried as the best to keep the best practices for code quality, testing, and version control.

Testing:

The testing phase involves verifying the functionality of the app and ensuring that it meets the requirements specified in the project plan. This will include unit testing, integration testing, system testing, and user acceptance testing. Any issues or defects found during testing was be addressed and resolved.30

4.2 Coding standards

Coding standards are a set of guidelines and best practices that ensure that the code is written in a consistent and readable manner. This helps to improve the maintainability, scalability, and quality of the codebase. The following are the coding standards that we used for writing the code of the Bike Rental app:

Code Formatting:

The codes for the app are formatted consistently and follow the standard indentation and spacing conventions java being an object oriented programming language all the codes are given in each class. This makes it easier to read and understand the code.

Naming Conventions:

The naming of variables, functions, and classes, all follow a consistent naming convention. such as bike booking activity is named as booking , the variables have a clear meaning so that it makes easier to understand the purpose of the code and reduces confusion.

Code Comments:

The code should be well-documented with clear and concise comments. So that it makes easier for us to find the main purpose of that codes to understand the functionality of the code and helps with maintenance and debugging id we encounter.

Error Handling:

Each and all individual module we tested it again and again to see if there is any error or bugs, so we can assure that error handling is implemented consistently throughout the codebase. This ensures that the app handles errors gracefully and provides meaningful error messages to users.

Code Reusability:

Here in Bike rental app is built in java base so all the codes are written in format of function so that it is modular and reusable manner. This allows the code to be used in multiple parts of the app, reducing the amount of code that needs to be written and maintained.

Performance:

The code that is used are optimized for performance and scalability. This includes minimizing the use of loops and other performance-intensive operations and using caching and other optimization techniques.

Testing:

From the testing we found the Bike Rental app to be functional and easy to use. The app passed all of the test cases, and no major defects were identified. Some minor issues were found, such as occasional slow loading times and minor display issues, but these did not impact the overall functionality of the app and resolved.

4.3 Coding Details

Our project contains many activities in android studio which contains approximately 50 sets of codes (UI in android studio and Java combined). Showing all the screenshots of all codes is nearly impossible so we will be displaying the source code of some of the important pages or activities.

The following code is to show the different garage in available

```
public void onClickBook(View view) {

        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("Garage");
        String p = pickup_atv.getText().toString().trim();
        String d = drop_atv.getText().toString().trim();
        Location loc = new Location(p,d,bikeId);
```

```
        ref.child(bikeId).setValue(loc);


        Intent intent = new Intent(getApplicationContext(), BikeBookActivity.class);
        Bundle extras = new Bundle();
        extras.putString(BikeBookActivity.BIKE_ID,bikeId);
        extras.putString(BikeBookActivity.BOOK_OPERATION,
BikeBookActivity.BOOK_ADD);
        intent.putExtras(extras);
        startActivity(intent);
 }
```

The following code is to provide the details of the vehicles and the details about it

```
public Bike(String id, String brand, String model, int mileage,
String type, String fuel, String LastServiceDate, int totalKmsDriven,
int year, String regNumber, int rentPerDay, boolean bikeAvailable,
String garage) {
    this.id = id;
    this.brand = brand;
    this.model = model;
    this.mileage = mileage;
    this.type = type;
    this.fuel = fuel;
    this.year = year;
    this.regNumber = regNumber;
    this.LastServiceDate = LastServiceDate;
    this.totalKmsDriven = totalKmsDriven;
    this.rentPerDay = rentPerDay;
    this.bikeAvailable = bikeAvailable;
    this.garage = garage;
}
```

Fare calculation part:

```
public void calculateTotalFare() {
//        totalFare = until - from;
```

```
        long milliseconds = until - from;
        float days = milliseconds/(1000f*60*60*24);
        totalFare = Math.round(days * rentPerDay * 100) / 100f;
//        totalFare = days * rentPerDay;
    }
```

To check if the bike is available or not

```
public static void isAvailable(String bikeId, Long from, Long until,
BookingDao bookingDao) {
    Booking booking = new Booking();
    booking.setBikeId(bikeId);
    booking.setFrom(from);
    booking.setUntil(until);
    booking.isAvailable(bookingDao);
}

public void isAvailable(BookingDao bookingDao) {
    if (from >= until) {
        bookingDao.getBoolean(false, invalidErrorMessage);
    } else {
        Booking.getBookingByBike(bikeId, new BookingDao() {
            @Override
            public void getBookingList(List<Booking> bookingList) {
                boolean available = true;
                if (bookingList != null && bookingList.size() > 0) {
                    for (Booking booking : bookingList) {
                        if ((booking.getStatus().equals(UPCOMING) ||
booking.getStatus().equals(ON_PROGRESS)) &&
!booking.isAvailable(from, until)) {
                            available = false;
                            break;
                        }
                    }
                }
                bookingDao.getBoolean(available, available ? null :
bookedErrorMessage);
            }
        });
    }
}
```

```
private boolean isAvailable(Long from, Long until) {
    return from > this.until || until < this.from;  /* Non
Overlapping Boundaries – Ensures that the booking time is after the
present time*/
}
```

Confirmation from the admin side and uploading the picture for the purpose of accident check.

```
private void confirmation( String i) {
    myDialog = new Dialog(getActivity());
    myDialog.setContentView(R.layout.activity_view_bike_details);
    confirm = myDialog.findViewById(R.id.confirm_button);
    pic1 = myDialog.findViewById(R.id.bike1);
    pic2 = myDialog.findViewById(R.id.bike2);
    pic3 = myDialog.findViewById(R.id.bike3);
    pic4 = myDialog.findViewById(R.id.bike4);

    id = i;

    pic1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 1;
            pickImage(pic1);
        }
    });
    pic2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 2;
            pickImage(pic2);
        }
    });
    pic3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 3;
```

```
            pickImage(pic3);
        }
    });
    pic4.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 4;
            pickImage(pic4);
        }
    });

    myDialog.show();
    confirm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            bol = true;
            myDialog.dismiss();
        }
    });
}
```

The code is on the admin side available as options to choose related to the rent

```
// Main Functions
public void startRent(BookingDao bookingDao) {
    if (status.equals(UPCOMING)) {
        status = ON_PROGRESS;
        from = new Date().getTime();
        updateBooking(bookingDao);
    } else {
        bookingDao.getBoolean(false, "Only " + UPCOMING + " Bookings
can be started");
    }
}

public void stopRent(BookingDao bookingDao) {
    if (status.equals(ON_PROGRESS)) {
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
```

```
            public void run() {
                status = COMPLETED;
                until = new Date().getTime();
                calculateTotalFare();
                updateBooking(bookingDao);
            }
        }, 3600000);
    } else {
        bookingDao.getBoolean(false, "Only " + ON_PROGRESS + "
Bookings can be stopped");
    }
}

public void cancel(BookingDao bookingDao) {
    if (status.equals(UPCOMING)) {
        status = CANCELLED;
        updateBooking(bookingDao);
    } else {
        bookingDao.getBoolean(false, "Only " + UPCOMING + " Bookings
can be cancelled");
    }
}
```

Confirmation on the admin side to start the rent and upload the necessary photos

```
switch (item.getItemId()) {
    case R.id.booking_start:
        confirmation(booking.getId());
        if(bol){
            booking.startRent(new BookingDao() {
                @Override
                public void getBoolean(Boolean result, String error)
{
                    if (result) {

Toast.makeText(getActivity().getApplicationContext(), "Rent Started",
Toast.LENGTH_SHORT).show();
                        retrieveBookingInfo();
                    } else {

Toast.makeText(getActivity().getApplicationContext(), error,
```

```
Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
        break;
```

```
private void confirmation( String i) {
    myDialog = new Dialog(getActivity());
    myDialog.setContentView(R.layout.activity_view_bike_details);
    confirm = myDialog.findViewById(R.id.confirm_button);
    pic1 = myDialog.findViewById(R.id.bike1);
    pic2 = myDialog.findViewById(R.id.bike2);
    pic3 = myDialog.findViewById(R.id.bike3);
    pic4 = myDialog.findViewById(R.id.bike4);

    id = i;
    pic1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 1;
            pickImage(pic1);
        }
    });
    pic2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 2;
            pickImage(pic2);
        }
    });
    pic3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            u = 3;
            pickImage(pic3);
        }
    });
    pic4.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```
            u = 4;
            pickImage(pic4);
        }
    });

    myDialog.show();
    confirm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            bol = true;
            myDialog.dismiss();
        }
    });
}
```

The below code is to display the options on the user's side

```
switch (item.getItemId()) {
        case R.id.booking_edit:
            Intent editIntent = new
Intent(getActivity().getApplicationContext(),
BikeBookActivity.class);
            Bundle editExtras = new Bundle();
            editExtras.putString(BikeBookActivity.BOOK_OPERATION,
BikeBookActivity.BOOK_MODIFY);
            editExtras.putString(BikeBookActivity.CAR_ID,
booking.getBikeId());
            editExtras.putString(BikeBookActivity.BOOK_ID,
booking.getId());
            editIntent.putExtras(editExtras);
            startActivity(editIntent);
            break;
        case R.id.booking_view_bike:
            Intent intent = new
Intent(getActivity().getApplicationContext(),
BikeDetailActivity.class);
            Bundle extras = new Bundle();
            extras.putString(BikeDetailActivity.BIKE_ID,
booking.getBikeId());
            intent.putExtras(extras);
            startActivity(intent);
```

```
            break;
        case R.id.booking_details:
            Intent intent1 = new
Intent(getActivity().getApplicationContext(), BikePhotos.class);
            Bundle ex = new Bundle();
            ex.putString(BikePhotos.BOOKING_ID,booking.getId());
            intent1.putExtras(ex);
            startActivity(intent1);
            break;

        case R.id.booking_cancel:
            booking.cancel(new BookingDao() {
                @Override
                public void getBoolean(Boolean result, String error)
{
                    if (result) {

Toast.makeText(getActivity().getApplicationContext(), "Booking
Cancelled", Toast.LENGTH_SHORT).show();
                        retrieveBookingInfo();
                    } else {

Toast.makeText(getActivity().getApplicationContext(), error,
Toast.LENGTH_SHORT).show();
                    }
                }
            });
            break;
        default:
            return super.onContextItemSelected(item);
    }
    return true;
}
```

REFERENCES

[1] Abhishek Chandrashekar, CEO and Co-Founder, Royal Brothers. *6 Challenges a Bike Rental Company Faces While Starting Up* - JAN 3, 2020 <https://www.entrepreneur.com/en-in/growth-strategies/6-challenges-a-bike-rental-company-faces-while-starting-up/344479>

[2] Anirban Chowdhury. *Designing Interface for an Online Bike Rental Service for Personified Tourism* - February 2021 <https://www.researchgate.net/publication/349433225_Designing_Interface_for_an_Online_Bike_Rental_Service_for_Personified_Tourism>

[3] Hristo Andreev - Sep 16, 2019. *UX/UI case study — Bike renting app. A step-by-step guide to UX design process*. <https://medium.com/@AndreevHristo/bike-renting-app-ux-ui-case-study-64aedaca2bbe>

[4] "Designerrs" *Bike Rental UX Case Study* <https://designerrs.com/bike-rental-ux-case-study/>

[5] "Q3TECH" *The Complete Guide On bike Rental App Development* <https://www.q3tech.com/blogs/the-complete-guide-on-bike-rental-app-development/>

[6] "Royal Brothers Rentals" <https://www.royalbrothers.com/>

[7] "Onn Bikes Rentals" <https://www.onnbikes.com/bike-rental-bangalore.html>

[8] "Bykemania Bike Rentals" <https://www.bykemania.com/>

[9] "Twist Throttle Bike Rentals" <https://twistthrottle.in/>

[10] "ANDROID STUDIO" <https://developer.android.com/studio>

[11] "FireBase Documentations" <https://firebase.google.com/docs>

[12] "Github" <https://github.com/>

[13] "Stack Over-flow" <https://stackoverflow.com/>

[14] "Geeks For Geeks" <https://www.geeksforgeeks.org/>

[15] "Tutorials Point" <https://www.tutorialspoint.com/>

[16] "Sage Bot – AI"<https://poe.com/Sage>