

Bike-Sharing System

Yosita Chotchueng, Weiran Tian,

Tony Simon Akkara, Chengcheng Zheng

Mani Shankar Chowdhury

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A project presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

4th November 2021

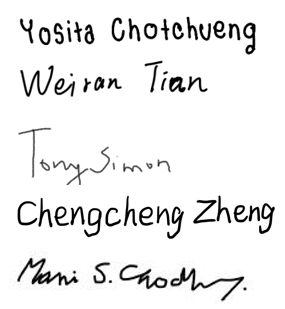
**Abstract**

Advanced technology has played a crucial role in people's lives since technology is a part of urbanization and modernization in society. The following problems come after every development; one is the environmental issue, including pollution, lack of resources, and resource misuse. Global climate change is an issue that every country worldwide pays attention to solve this problem since the industry and household sectors are part of gas emissions. There are various campaigns has launched to reduce gas emissions, the sharing system is one of the beneficial campaigns. The sharing system allows people to give or use something with others; therefore, the bike-sharing campaign can reduce carbon dioxide, a primary reason for global climate change.

The bike sharing system proposed here lets users book bikes at any station available and return to any of the stations. Users can also use a wallet to store money and pay for the transactions. Reporting of defective is made possible in the system. It also consists of an operator module which allows them to track location of bikes, move bikes and repair bikes. It also has a manger module which displays a set of visualisations and provide insights over a defined period of time.

# Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Yosita Chotchueng, Signature:

Weiran Tian,

Tony Simon Akkara,

Chengcheng Zheng

Mani Shankar Chowdhury

# Acknowledgements

We would like to express our immense gratitude to Dr Mireilla Bikanga Ada for her supervision and support throughout the whole project. Dr Ada’s genuine enthusiasm, insightful advice, and devoted guidance were paramount to the enhancement and completion of this work.

Many thanks to our teammates, for their support and patience. They are very helpful and dedicated to the project. Everyone has put great effort into this project.

# Contents

Contents i

1. Introduction 1

1.1. Project Motivation 1

1.2. Project Aims and Objectives 1

2. Background Survey 2

2.1. Literature Review 2

2.1.1. Sharing economy 2

2.1.2. Bike sharing scheme 2

2.2. Existing products 3

2.2.1. Santander Cycles 3

2.2.2. Nextbike 3

2.2.3. Lime 4

2.2.4. Comparison of the core features 4

3. Requirements Analysis 5

3.1. Functional Requirements 5

3.1.1 Must have 5

3.1.2 Should have 6

3.1.3 Could have 7

3.1.4 Would not have 7

3.2 Non-Functional Requirements 7

4. Design 8

4.1. Development methodology 8

4.1.1 Gantt Chart 8

4.2. System Architecture 8

4.3. ER Diagram 8

4.4. Prototype 9

4.4.1 Paper prototypes 9

4.4.2 Wireframes 9

5. Implementation 10

5.1. Frontend 10

5.1.1. Web app interface 10

5.1.2. Map 10

5.1.3. Data Visualization 10

5.2. Database 11

5.3. Backend 11

5.3.1. separate interface for different users 11

5.3.2. Rent and return bikes 11

5.3.3. Report defective bikes 12

5.3.4. Wallet and Billing 12

5.3.5. Track locations and move bikes 13

5.3.6. Repair bikes 13

6. Testing 14

7. Future work & Conclusion 15

7.1 Future work 15

7.2 Conclusion 15

References 16

Appendix A 18

Appendix B 23

# Introduction

This chapter introduces the Bike Share System Project, describing the motivation and goals for developing this application. The chapter then concludes with an overview of the remaining chapters and contents of this paper.

## Project Motivation

As innovative technology has developed, society has moved towards urbanization and modernization. Becoming an urbanized and modernized society leads to elevated air pollution, one of the crucial contributors to global climate change. Road congestion and heavy traffic are related to air pollution; studies show that more than 80% of nitrogen dioxide emissions come from road vehicles[5]. There are various solutions to address this problem; the sharing system is the most popular and realistic solution, which is environmentally friendly and environmentally sustainable[13].

The consequences derived from the pollution problem contribute to the bike system project, which involves reducing lethal gas emissions, traffic congestion, and fuel usage. Furthermore, public bicycles are more special since individuals can utilize them on a needed basis, providing timesaving and convenience [14]. However, the bike-sharing system project plays a crucial role in urbanization development as the project enables the promotion of economic growth. The advancement of technology allows involved organizations to cope with real-time data that will be the most beneficial to the economy and society.

Thereby, this bike-sharing application is a platform to satisfy the prior limitations and gaps from different applications by allowing customers to rent, return and pay a bill from anywhere. Furthermore, there are services from the operators who assist and support customers when defective bikes exist in various locations and managers who manage the business overview by applying the report to obtain the business flow.

## Project Aims and Objectives

The aim of the project is to design and generate software to support a bike-sharing system, including an end-to-end function prototype and illustrating the appropriate data. The objectives of this project are to develop software that:

1. To provide an interface for customers to reserve and return bikes
2. To provide an interface for customers to report defective bikes
3. To provide an interface for customers to pay their bills.
4. To allow operators to assess the system's state and make changes if necessary, including tracking the location of all bikes in the city, repairing a defective bike, and moving bikes to different locations around the city as needed.
5. To provide the usage reports to managers.

# Background Survey

## Literature Review

### Sharing economy

The concept of “sharing economy” dates to 1980s. It is the predecessor of “collaborative consumption”, which was proposed by Felson and Spaeth. They suggested that consumers need a product’s use value rather than the product itself. Accordingly, it contends that it is more beneficial to lease than to buy. The sharing economy is defined by Eckhardt et al as a scalable socio-economic system that employs technology-enabled platforms that provide users with temporary access to tangible and intangible resources that may be crowdsourced. The authors define five attributes of the sharing economy: crowdsourced supply, temporary access, transfer of economic value, platform mediation, expanded consumer role[6]. Munoz and Cohen define the sharing economy as an “a socio-economic system enabling an intermediated set of exchanges of goods and services between individuals and organizations which aim to increase efficiency and optimization of sub-utilized resources in society.”[1] The concept of sharing economy is defined in many ways and there is no common understanding of sharing economy. But the core concept is clear which is transferring property value to someone who needs it.

Hossain, Mokter find that sharing business is mainly in accommodation and transportation business [17]. The first time sharing economy entered common public disclosure is the huge successes of Airbnb and Uber in 2011 [10]. After that, a lot of sharing economy appeared including sharing bikes.

### Bike sharing scheme

Bike sharing scheme has changed three generations in the past 45 years. The first generation was the Witte Fietsenplan: The White Bicycle Plan. It is conceived by the Provo anarchist movement to be against pollution in Amsterdam. This scheme failed soon because the bicycles were quickly confiscated by the police. The second version of bisk sharing scheme was marked by Bycykler København programmer in the Danish capital in 1995, but these bikes suffered high levels of vandalism and theft by the anonymous users. In order to address the problems, the latest generation of bike sharing scheme adopts plenty of advanced improvements including electronically locking racks or bike locks, telecommunication systems, smartcards and fobs, and mobile phone access. These technologies make users easier to identify and bikes easier to keep track of [10].

Free-floating bike sharing appeared as an innovation bike sharing scheme model. It is considered to be the fourth generation. The original version of the free-floating bike sharing system consisted of for-rent-bicycles locked with a combination code number which a registered user would access by calling the bike management company. In 2000, Deutsche Bahn launched Call a Bike System enabling users to unlock the bikes by variety of methods including SMS, phone call and smartphone app. It saved a lot of money to construct docking stations compared to station-based bake sharing scheme[2].

Nowadays many cities all over the world have observed a resurgence of cycling as a way of travel. Currently, there are over 9 million shared bicycles in more than 2000 cities all over the world [11]. Bike sharing scheme provide people with a convenient travel method by freeing people from the need to secure their bikes. What is more, decisions to make a trip by bike can be made very easily and quickly because it is affordable and healthy. Bike sharing scheme enhance access to public transport system by improving first and last mile connectivity as well [17].

## Existing products

This section presents the analysis of three existing bike-sharing apps in UK. The first one is Santander Cycle which provide self-service for short journeys in London. The second is Nextbike which is popular in Glasgow. The last one Lime which is different from the other two apps and is a dockless bicycle sharing app.

### Santander Cycles

Santander Cycles is the most famous bike-sharing app in the UK. The Santander Cycles are run by Transport for London and was launched in 2010. In 2020, There are more than 11,500 bicycles and 780 docking station in the system. One reason for the success of Santander Cycles is that all bikes are docked and therefore, the bikes cannot be dumped on the streets.

Santander Cycles main color is red. The interface of this App is clean and intuitive. The main interface of this program is a map showing the nearby docking stations and the current number of bikes in each station to the user. Its main functions are at the top and bottom of the main interface. The upper left corner of the main interface is the main menu button. After clicking it, it shows some options such as sign in/out, plan a journey, report an abandoned bike and so on. The upper right corner is the button to refresh current location and the button to find bikes at the specified location. The bottom of the main interface is mainly to show users the buttons of their nearby, recently used and favorite docking station. After selecting a docking station, you also can see the status of the station which including the number of bikes available and the station spaces and so on. Figure A. **1** Santander Cycles Interface shows the interface of Santander Cycles.

It costs £2 to be able to use the bikes for the next 24 hours (access fee). Extra ride charges are weighted to promote the constant circulation of bicycles. The first 30 minutes of each journey are free, after that it costs £2 for each additional period of up to 30 minutes. Alternatively, yearly membership costs £90, where the first 30 minutes of each ride is free and longer journeys cost £2 for each extra 30 minutes or less.

### Nextbike

Nextbike is a German company and operates in more than 300 cities all over the world. It was introduced to Glasgow in July of 2014. There were almost 200,000 rentals in the first two years. There was steady growth afterwards. The bikes can be rented and returned via app, hotline, terminal. What should be noticed is that it not only provides normal bikes but also electric bikes in Glasgow.

Nextbike’s main color is green. Its main interface is also a map showing nearby docking stations and bikes, but fewer functions are placed on the main interface. In the upper left corner of the main interface, there is also a button for the main menu, which will display rental activity, wallet, account settings and other functions. In the upper right corner, there is a button to refresh positioning and a button to filter bike types. There is only one button to scan the code at the bottom of the main interface compared to Santander Cycles. After selecting a docking station, you also can see available racks and available bikes with their bike number. Figure A. 2 Nextbike Interface shows the interface of Nextbike.

Its pricing strategy is different from Santander Cycles. Upon registration a £5 deposit will be taken as a security deposit and it usually charges at the standard Pay As You Ride rate of £1 for every 30 minutes, or a full-day price of £10 for 24 hours. If you don’t return the bike to a official station, you will get fined. E-bikes charge you £2 for every 20 minutes. You can also subscribe monthly or annual membership.

### Lime

Lime is based in San Francisco. As part of trials approved by UK government, Lime began to operate e-scooters in London in June 2021. What sets Lime apart from the other apps is that Lime provides electronic vehicles to the public including e-bikes and e-scooters in London and it is dockless.

Lime;s main color is also green. The interface of Lime is simple and is a map showing nearby vehicles. The upper left corner of the main interface is a main menu button, after clicking it, you will see the entrance of your payment, history and other functions and the bottom of main interface is a scan button. By selecting a vehicle on the map, you can easily get some basic information about the vehicle such as the vehicle code and quantity of electricity. Figure A. 3 Lime Interface shows the interface of Lime.

A Lime electric-assist bike costs £1 to unlock and 15p per minute to ride. The minute rate is 16p per minute for e-scooters.

### Comparison of the core features

As bike-sharing apps, the three apps share the same core function. The main differences between them are the products they provide and the pricing strategy. Santander Cycles provides normal docked bikes to users in London, Lime provides dockless e-bikes and e-scooters while Nextbike provides both normal docked bikes and e-bikes to users. Regarding the pricing strategy, Santander Cycles and Lime charge you both access fees and additional ride fee (Santander Cycles charges ride fees by 30 minutes but Lime charges ride fees by minute) while Nextbike only charges ride fees by 30 minutes.

# Requirements Analysis

This chapter will outline bike-sharing system requirements related to accomplishing the goals and objectives of this project by applying the MoSCoW analysis method. The provisions of the bike share program are based on discussions with the customers (supervisor) and the evaluation of the available literature in this area by reviewing and analyzing existing products.

MoSCoW analysis is a method for ranking the relevance of ideas and product requirements. This method will apply in business and management in gathering requirements. MoSCoW is an abbreviation from Mo – Must have, S – Should have, Co – Could have, and W – will not have.

In this chapter, MoSCoW analysis is implemented with two types of requirements: functional and non-functional requirements. Functional requirements are the basic system functioning describing what the system does or must not do, whereas non-functional requirements define how the system does. They are not affected by the basic functionality of the system.

## Functional Requirements

### 3.1.1 Must have

* **User login to the system**
  + **system** 
    - asking the user to sign-in by keying in user-id and passcode
    - verified user-id and passcode to identify user's unique identity
  + **user**
    - input user-id and passcode
* **Sign up for new user**
  + **system**
    - username, password, gender and age are stored in the database
* **Track location of all bikes** 
  + **system** 
    - bike's locations able to be tracked instantly
  + **operator**
    - able to track each bike's location
* **Repair defective bikes**
  + **system**
    - demonstrate bike information including bike id, bike's status, issue, and location to operators
  + **operator**
    - select locations to identify defective bikes
    - change bike's status from defective to working when the bikes have repaired
* **Move bikes from different locations** 
  + **operator**
    - move bikes from one location to another location when the number of bikes trend to be unbalanced
* **Billing** 
  + **system**
    - calculate charges to the customer by duration multiplied with a fixed range of time rate
  + **customer**
    - pay calculated charges if the balance is not negative
* **Rent bikes**
  + **system**
    - allow customer rent if the balance is not negative
  + **customer**
    - if the balance is negative, customers must pay the bill before renting the bikes
* **Return bikes**
  + **system**
    - check rent activity to confirm there is exists transaction ready to return
* **Report on a defective bike**
  + **operator** 
    - access defective bikes detail and location to fix the issues
* **customer** 
  + - able to report defect bike number and broken parts
* **View balance**
  + **system**
    - not allow customers to proceed with renting process, the customers must pay the bill to get zero or positive balance
  + **customer**
    - user should be able to add an amount to his / her own wallet

### 3.1.2 Should have

* **Smart interface for different users**
  + **system**
    - identify user's role by username login
    - separate interface website for different users (customers, operators, and managers)
* **Generate viewdata reports.** 
  + **manager**
    - generate reports from different aspects
      * bike activities report
        + Transaction per day
        + Transaction in each location
      * revenue report
        + Revenue per day (in pound)
      * user type report
        + Transaction by gender
        + Transaction by age group

### 3.1.3 Could have

* **Payment security**.
  + **system**
    - check the calculation accuracy of charges and balance
* **Promotion**
  + **system** 
    - offer promotion for outstanding activities or special age group including top spender or people who are under 20

### 3.1.4 Would not have

* **Sensitive data security** 
  + **system** 
    - convert sensitive data, including username and password, with hashing before storing in the database

## 3.2 Non-Functional Requirements

* **System stability:** the system must remain stable even there is a high demand of requests to access the website
* **Interactive:** the website should be flexible, andno specific time to access the website
* **User-friendly system:** inexperienced users can visit and process activities easily

# Design

This chapter will discuss the software architecture processes and practices used throughout the project, including development methodology, system architecture, ER diagram, and prototypes. Finally, this chapter will explain the technologies decided for building a bike-sharing system.

## Development methodology

The project focused and followed Agile and Scrum methodologies which are the standard approach for project management. The agile method is a sprint-based method that splits the development process into smaller steps, including discovery, design, development, and test. Scrum method is a framework applied from the Agile method; this method provides more adaptive and flexible through the project life cycle [16].

### 4.1.1 Gantt Chart

The bike-sharing project applied Gantt Chart to track the project's progress and assist in tracking time spent on each task. Tasks were split into sets of dependent tasks and mapped their dependencies to the Gantt Chart. If a job were in danger of not being complete, teams could eliminate that task and all dependency tasks from the sprint [15]. The Gantt chart of the bike-sharing system project is shown in Table B. 1 Gantt Chart.

## System Architecture

The bike-sharing system project was decided to be a web application. The primary framework for this project is Streamlit.

Once an existing user logs in there are three significant components in the application, including Customer, Operator, and Manager roles. For a new user there is a sign-up page to provide basic details and register into the application. The system architecture is shown in Figure A. **4** High Level System Diagram

Action performed by any user in the system is logged in to the ActivityLogs table for verifying the transactions and allowing them.

## ER Diagram

An Entity-Relationship (ER) diagram is one of the flowcharts that demonstrate how entities are related to each other within a system. It is most often used to design a database or debug relational databases applied in a database design to structure data [103]. The database's cardinality can divide into three types: one-one, one-many (a record in one entity can refer to multiple records in other entities), and many-many[3].

The significant entities of the biking-sharing system are User, ActivityLog, Location, and Bike shown in Figure A. 5 ER Diagram. Each table contains primary keys, which are unique values that identify each row of a table, and some tables have foreign keys, which are a set of attributes that refer to the primary key in other tables.

## Prototype

### 4.4.1 Paper prototypes

Paper prototyping's primary goal is to communicate ideas and user flow by using hand-sketched screens that represent a product between developers, designers, and users. This prototype is the first step of developing ideas and designing the user flow for a high-level user experience to understand the product's appearance [12]. Several paper prototypes are generated for a different part of the web application.

The main idea behind Figure A. 6 is to show that customers can book, return and report defective bikes. The customer can select tasks they wish to do, and the map will appear to provide more understanding of the location of the bikes and how many availabilities. The other customer, operator, and manager aspects are shown in Figure A. 7-10.

### 4.4.2 Wireframes

A wireframe is a web page's layout that demonstrates what interface elements will appear on the website [7]. The wireframes categorized by user's roles were implemented.

An example of the navigation page is shown in Figure A. 11, the Menu includes overview sign-in, and sign-up presented to users to choose the task. The navigation page illustrated a reference manual for booking a bike, controller, and manager. If the user clicks on the different functions on the Menu, a website will be assigned to a separate web page.

# Implementation

This chapter will describe the implementation of key features in the bike-sharing program. Additionally, this chapter will include various tools and libraries we used to implement the system and discussing any difficulties we encountered during the implementation. Here is the video demo of our system: <https://www.youtube.com/watch?v=KyhrYWzbkic>.

## Frontend

### Web app interface

To build our bike-sharing system, we choose to use streamlit. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps. It’s convenient to get started and useful to build and deploy powerful data apps. Because Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app. Also streamlit provide a convenient development flow, if the source code of the streamlit’s python script changes the app shows whether to rerun the application or not in the top-right corner. This means every time we make some changes, it will reflect immediately in the web app. This loop between coding and viewing results live makes the code seamlessly with the app.

We use streamlit to create an overview page to simply show the structure of the website. We also have sign-in and sign-up page. After users signing in, different role will access in different page tailored for them. This is how we organize our web. All the information and requests from users will be sent to the backend to process and then result will be sent again to users by showing on the interface.

### Map

Folium makes it easy to visualize data that’s been manipulated in Python on an interactive leaflet map. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map. The library has a number of built-in tilesets from OpenStreetMap, Mapbox, and Stamen, and supports custom tilesets with Mapbox or Cloudmade API keys.

we use streamlit-folium to determine what functionality is desirable for a Folium and Streamlit integration. Technically, we just use one method folium\_static(), which takes a folium.Map or folium. Figure object and displays it in a Streamlit app.

### Data Visualization

Plotly is a very famous and powerful open-source data visualization framework. It displays information by constructing interactive charts based on the web form displayed by the browser. It can create up to dozens of exquisite charts and maps.

Plotly Express is a high-level package of Plotly.py, built-in practical, modern drawing template, users only need to call simple API functions, you can quickly generate beautiful interactive charts.

We use Plotly Express to draw chars. Technically, we use px.line(), px.pie(), px.bar() to draw our charts. These charts are helpful when the managers need to analyze the operation situation of this products, also it is our extra function.

Pydeck is another tool we used in our data visualisation. Pydeck allow users to create deck.gl maps without having to know a lot of Javascript. Pydeck has the availability of the full deck.gl layer catalog in Python and it supports for large-scale updates, like color changes or data modification, to hundreds of thousands of visualized data points.

We create histogram of rent activities in the map by using pdk.Deck, pdk.Layer functions. Mangers could directly know the bike activity from the histogram from the map.

## Database

SQLite is a widely used database management system. We choose SQLite to build our database. We established 4 tables to store our data. Acticitylogs is used to store all transaction information, including bike ID, user ID, location, time, amount to be paid. Users is used to store all users basic information, user ID, password, the role of users. Bike table stores all bike information, including ID, location, status.

The last table location stores all the dock station information. In the backend part, we will describe how each main function use these data and update these data. Below is a picture to show the relationship of these four tables.

## Backend

We will organize this part by describing how these main functions were achieved. We develop backend programs mainly use Datetime to process data and time, SQLite to connect to database, pandas to manipulate date, NumPy to calculate.

### separate interface for different users

When users login, their roles need to be recognized. Different roles should have different interfaces. Sharing one interface will cause lots of confusion for users as different types of users use the system for different purposes. So, we add role information to each user in the database. When users log in, the system will read the role information of the user from the database and then different interfaces will be created for different roles by using if statement.

### Rent and return bikes

We investigated existing products in UK and found that Mobike exited the UK market because they suffered huge losses from their dockless bike products. So, we decided to have a docked bike product. We store the dock station data in the location database.

**Rent bikes**

When users choose to book, firstly, the system will check if the user’s balance is negative. If the balance is negative (if the balance is 0, we decided to let users continue to rent), the system will display a new button ask the user to pay the previous charge. After that the user can continue to rent a bike. Next, when the user presses the book button, the system will check if the selected location has at least one bike to rent and whether the user has already rented a bike by reading data from bike and users database respectively. After that, the system will rent a bike to the user and change the bike status in the bike database and create a new bike activity record in ActivityLog database.

**Return bikes**

The logic of “Return” is similar but slightly different. After the user choose to return a bike, the system, firstly, check whether the user has rented a bike from ActivityLog database. After that, it will calculate the amount to be charged and update the bike status and location information in bike database and the bike activity record which is created when the user rents a bike.

### Report defective bikes

This function only can be used by those who has rented bikes. When the user chooses to report defective bikes, the system will firstly check if he has rented a bike. After that a listbox is created, the user can choose the defective type here but when the user chooses “another defect”, they will be asked to specify where the problem is. At last defective information will be updated in the bike database and “Return” function will be called to return the bike and charge the user.

### Wallet and Billing

This part contains three main functions: view balance, topup wallet and calculate bills.

**View balance**

It is a simple function, the system will search the balance information from Users database and show it to users.

**TopUp wallet**

As recharging accounts need us to access a third-party payment systems or banks, we don’t make it like a real function. We just let users to input the amount they want to recharge, when the system receives the number, it will add the user database directly.

**Calculate bills**

This function is in the ‘Return Bike’ function now because we are currently only providing one billing model (we are billing 1 penny per 10 seconds, but the first 5 seconds are not charged and subsequently every 10 seconds is charged.). After the system calculates the amount that needs to be charged, it will update the ActivityLog to keep history of charges and users’ wallet as well.

The system is provisioned to use a different rate of billing as required in future.

### Track locations and move bikes

**Track locations**

This is a simple function. The system will read data from BIKE database, and sort them by locations and show the result to operators directly.

**Move bikes**

If operators want to move bike, they can choose the move task. We create two boxes for operators to choose the location they want to move bikes and the destination they want these bikes are moved to. Also another box is created for them to enter the number of bike they want to move. The system will firstly check if there is enough bikes to move, then update the database.

### Repair bikes

These functions are slightly different from move but similar. Firstly, the system will read data from Bikes database showing the operators the detail of each defective bikes. Then two listboxes are created letting the operators to choose the location and ID of defective bike they want to repair. After clicking “repair the bike” button, they system will update the bike database changing the bike status. We don’t make it complex, because in the actual situation, we need someone to repair the bike on site and then the status of bike can be changed from “defective” to normal.

# Testing

Test is a crucial stage for every program development. We take tests to ensure every function is achieved. Below is our test result:

|  |  |  |
| --- | --- | --- |
| **Bike Sharing Test Table** | | |
| **Test case** | **Description** | **Result** |
| **General function** | | |
| Sign in | Enter user name and password. Managers would be directed to the managing page, operators would be directed to the operating page, new customers would be directed to the customer page. | Pass |
| Sign up | Enter sign-up information, new account is created and this account could sign in successfully. | Pass |
| Sign out | able to sign out | Pass |
| **Customer function** | | |
| Rent a bike | Allow customer rent a bike if the balance is not negative. if the balance is negative, customers must pay the bill before renting the bikes. | Pass |
| Return a bike | If a customer has a rented bike, the system allows the customer to return the bike. If a customer has not a rented bike, the system rejects the request.  Riding fee is deducted from user's account | Pass |
| View balance | Show to customer the balance of the account to two decimal places | Pass |
| TopUp Wallet | Allow customer to recharge their account and the balance increases. | Pass |
| Report a defective bike | Allow customer to report defect bike number and broken parts. | Pass |
| **Operator function** | | |
| Track bike location | Show all dock and number of bikes on the map. Show a list that contains all docks and the number of bikes in each duck. Show a list that contains all bike status. | Pass |
| Repair a bike | able to repair a bike by selecting the bike ID | Pass |
| Move bikes | able to move bikes by choosing the original location and destination. | Pass |
| **Manager function** | | |
| Data visualisation | Show bike activity on the map. Show revenue per day, transaction per day on line charts. Show transaction in each location on a pie chart. Show transactions by age, by gender on bar charts. | Pass |

# Future work & Conclusion

## 7.1 Future work

In future the following enhancements can be made:

* As the basic framework supports Web based applications, it can be used by multiple users at a time for a real-time booking application. However, the scope needs to be tested.
* We can enhance the application to book and return a bike at any location as per user's choice, in future.
* We may offer promotions to the highest spenders per month and student promotions in future.
* System can suggest a location where the number of bikes is going low, and the operator can move the bikes to that location.
* System can track the number of defective bikes in a location and automatically notify the operator or manager.

## 7.2 Conclusion

The purpose of this project is to design software to support bike-sharing systems. This article gives research regarding existing products and then makes decision to make our product as docked bike product. Then we gather requirements from customers, operators and managers who are the key users of this product, and we also classified these requirements by MoSCoW model. In the design chapter, we concretized our ideas. System architecture, database, and interface were made. Next chapter we describe in detail how our system was achieved and the technologies we used. Additionally, we also have a test of main functions.

In conclusion, although there are still a few incomplete features, the project has successfully achieved all the requirements in Chapter2.

# References

1. BoydCohen, Pablo Muñozc, Sharing cities and sustainable consumption and production: towards an integrated framework, Journal of Cleaner Production, 134, 87-89
2. D’Almeida, L., Rye, T., & Pomponi, F. (2021). Emissions assessment of bike sharing schemes: The case of Just Eat Cycles in Edinburgh, UK. Sustainable Cities and Society, 71, 103012. doi:10.1016/j.scs.2021.103012
3. Dan-Chuku, C. (2018) 'How To Implement One to One, One to Many and Many to Many Relationships When Designing A Database', medium, 10th October. Available at: https://medium.com/@emekadc/how-to-implement-one-to-one-one-to-many-and-many-to-many-relationships-when-designing-a-database-9da2de684710 (Accessed:29th October)
4. DeMaio, Paul. (2009). Bike-sharing: History, Impacts, Models of Provision, and Future. Journal of Public Transportation, 12 (4): 41-56.
5. Department for Environment, Food and Rural Affairs (2017) UK plan for tackling roadside nitrogen dioxide concentrations, An overview, July 2017
6. Eckhardt, Giana M., Houston, Mark B., Jiang, Baojun, Lamberton, Cait, Rindfleisch, Aric, Zervas, Georgios (2019), “Marketing in the Sharing Economy,” Journal of Marketing, 83 (5), 5–27.
7. Experience UX (2019) ‘What is wireframing’, Available at :https://www.experienceux.co.uk/faqs/ what-is-wireframing/#:~:text=A%20wireframe%20is%20a%20layout. (Accessed: 29th October 2021)
8. Felson, M. and Spaeth, J.L. (1978), “Community structure and collaborative consumption: a routine activity approach”, American Behavioral Scientist, Vol. 21 No. 4, pp. 614-624.
9. Hossain, Mokter (2020). Sharing economy: A comprehensive literature review. International Journal of Hospitality Management, 87(), 102470

https://creately.com/blog/diagrams/er-diagrams-tutorial/ (Accessed: 29th October 2021)

1. Martin, Chris J. (2016). The sharing economy: A pathway to sustainability or a nightmarish form of neoliberal capitalism?. Ecological Economics, 121(), 149–159.
2. Meddin, R., DeMaio, P., O’Brien, O., Rabello, R., Yu, C., & Seamon, J. (2020). The meddin bike-sharing world map. Retrieved December 1, 2020 https://bikesharingworldmap. com/#/all/2.7/0/51.5/.
3. Mifsud, J. (2020) 'Paper Prototyping As A Usability Testing Technique', usabilitygreek, Available at: https://usabilitygeek.com/paper-prototyping-as-a-usability-testing-technique/ (Accessed:29th October 2021)
4. Nishadha (2021) 'Ultimate Entity Relationship Diagram Tutorial (ER Diagrams)', creately, 27th September. Available at: Paton, G. (2015) ‘Cycling at its most popular in 24 years’, the times, 23rd May. Available at: <https://www.thetimes.co.uk/article/cycling-at-its-most-popular-in-24-years> tzzjdtwgvj8 (Accessed: 25th October 2021)
5. Qiu, L and He, L Bike Sharing and the Economy, the Environment and Health-Related Externalities, Sustainability, 2018, 10, 1145
6. Ramos, D. (2021) 'Using Gantt Charts in an Agile Environment', smartsheet, 27th July. Available at: https://www.smartsheet.com/content/agile-gantt (Accessed: 29th October 2021)
7. Stobierski, T. (2021) 'Agile vs Scrum: What's the Difference?', northeastern graduate program, 25th March. Available at: https://www.northeastern.edu/graduate/blog/agile-vs-scrum/ (Accessed: 29th October 2021)
8. Sun, Guibo; Zacharias, John (2017). Can bicycle relieve overcrowded metro? Managing short-distance travel in Beijing. Sustainable Cities and Society, 35(), 323–330. doi:10.1016/j.scs.2017.08.010

# Appendix A

**Additional Figures**

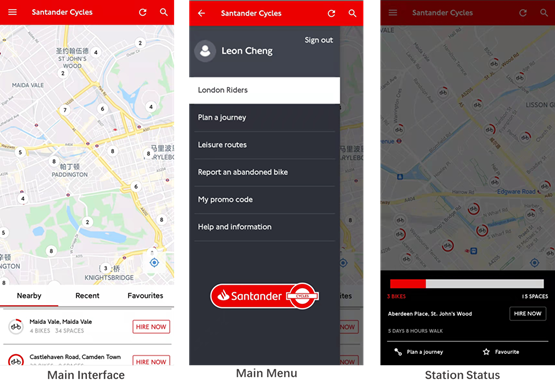


Figure A. 1 Santander Cycles Interface

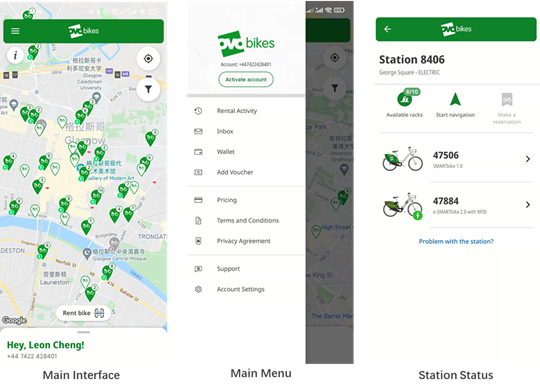


Figure A. 2 Nextbike Interface

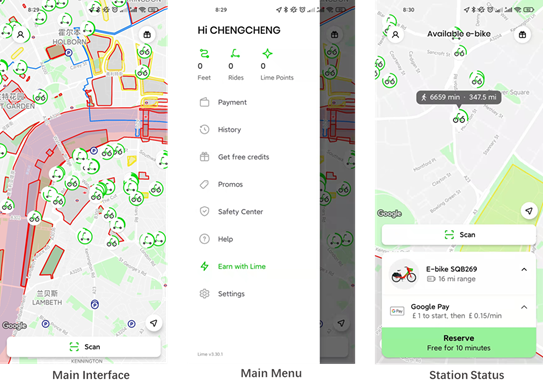


Figure A. 3 Lime Interface

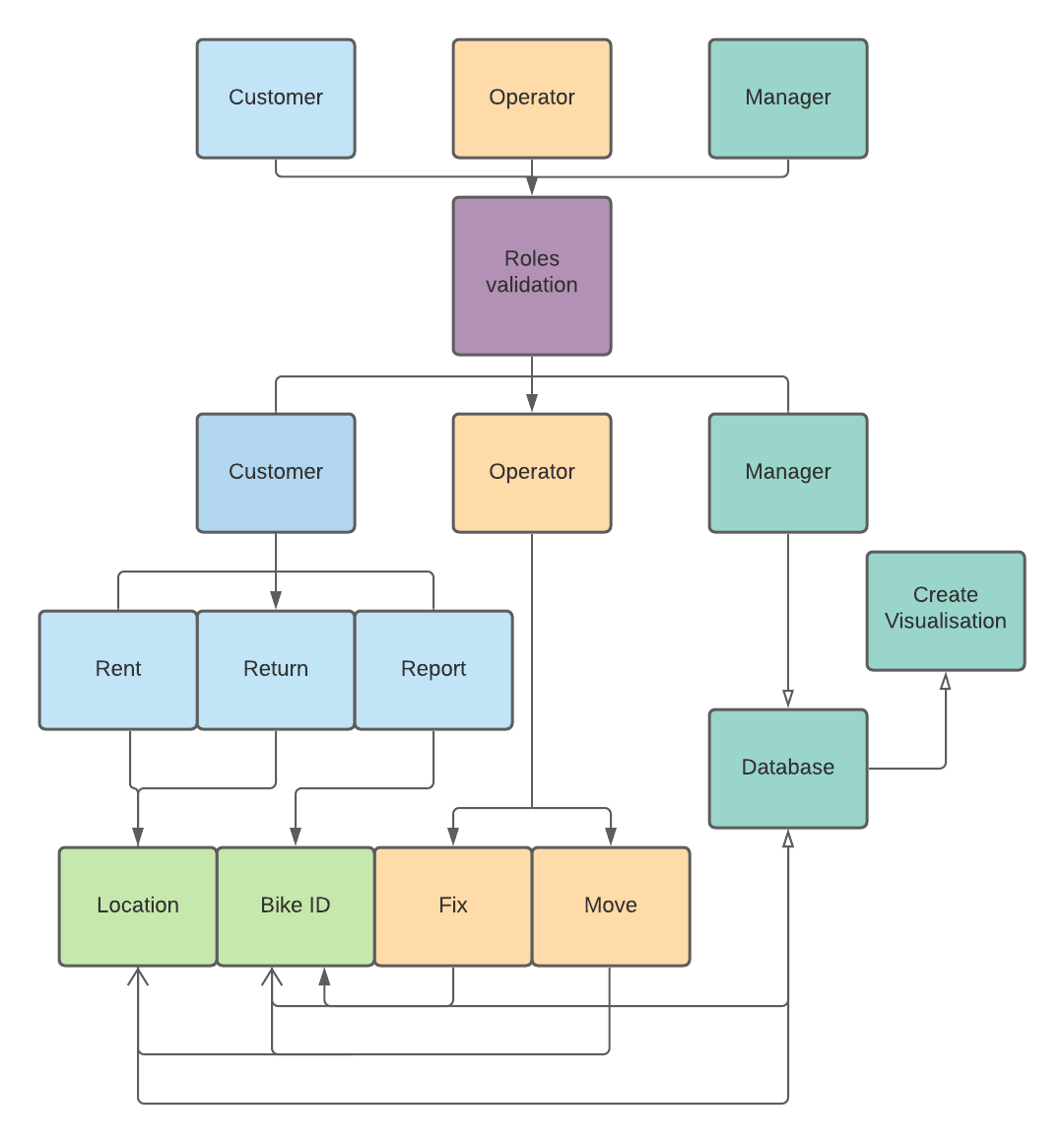


Figure A. 4 High Level System Diagram

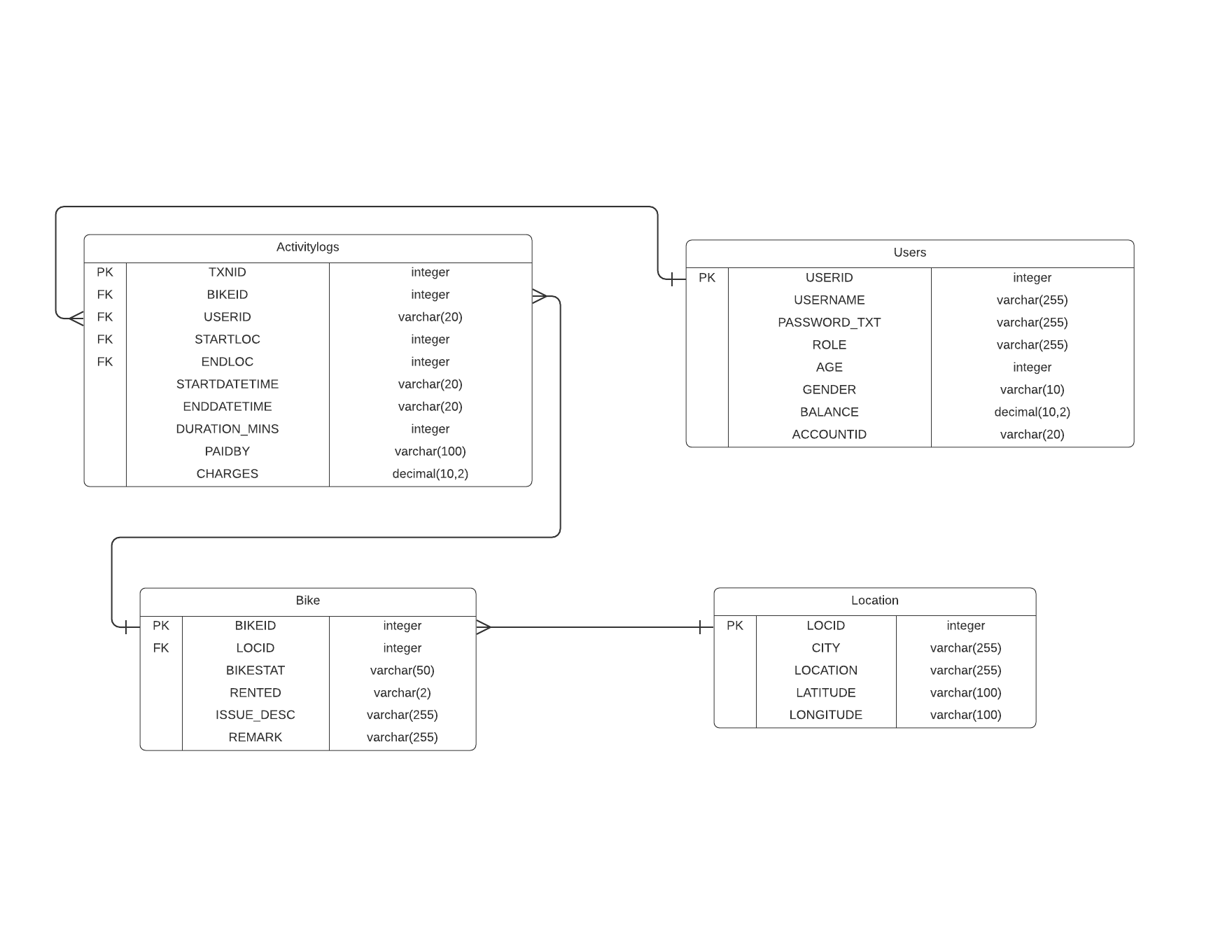


Figure A. 5 ER Diagram

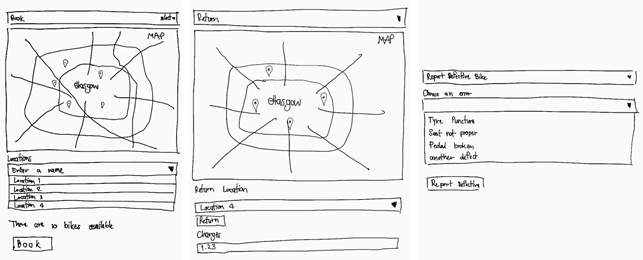


Figure A. 6 Booking, returning, and reporting defective bikes of customer prototypes.



Figure A. 7 Navigation Overview

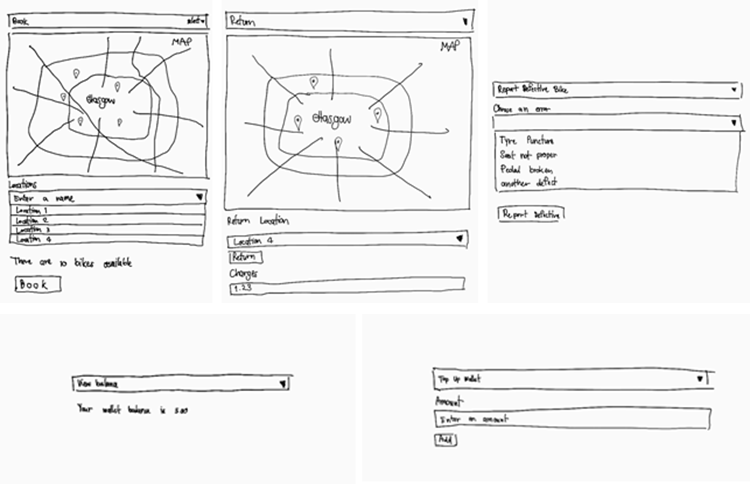


Figure A. 8 Customer Interface

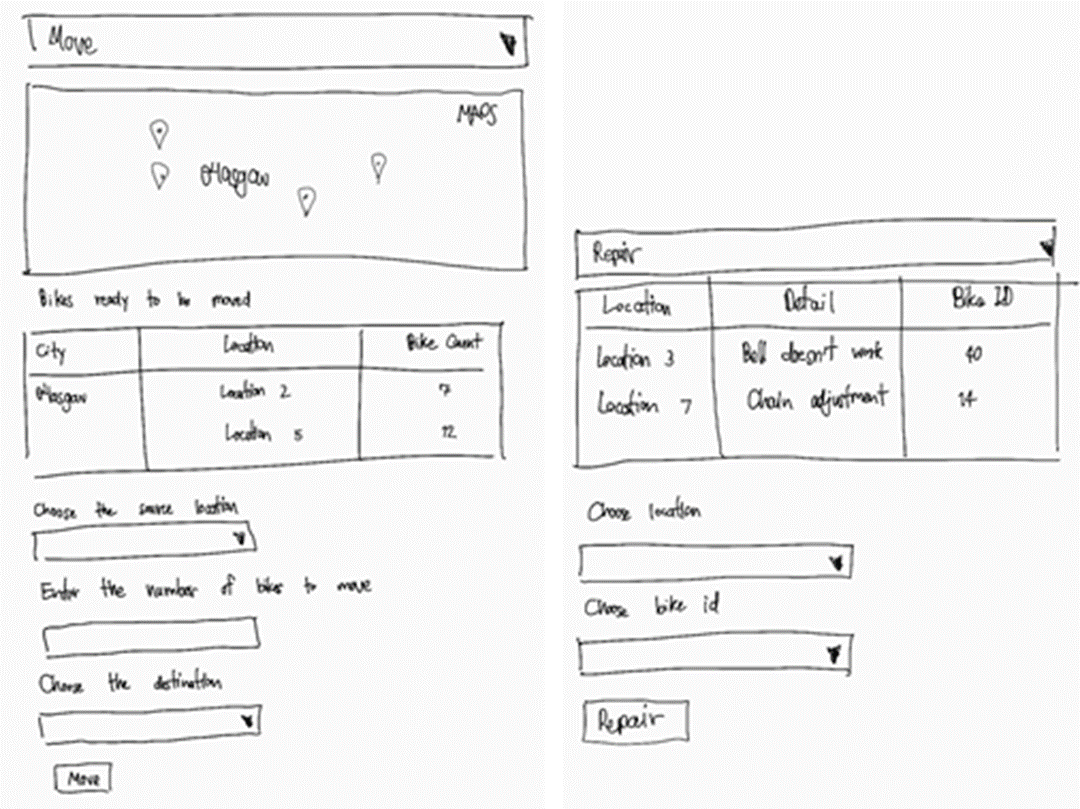


Figure A. 9 Operator Interface

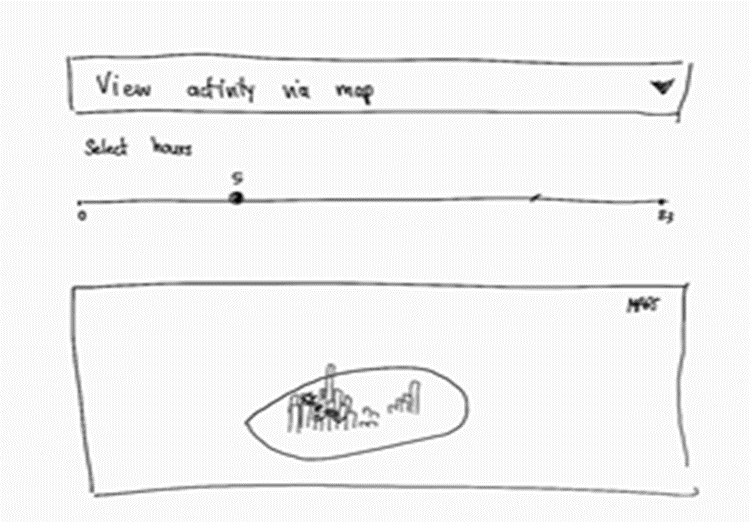


Figure A. 10 Manager Interface

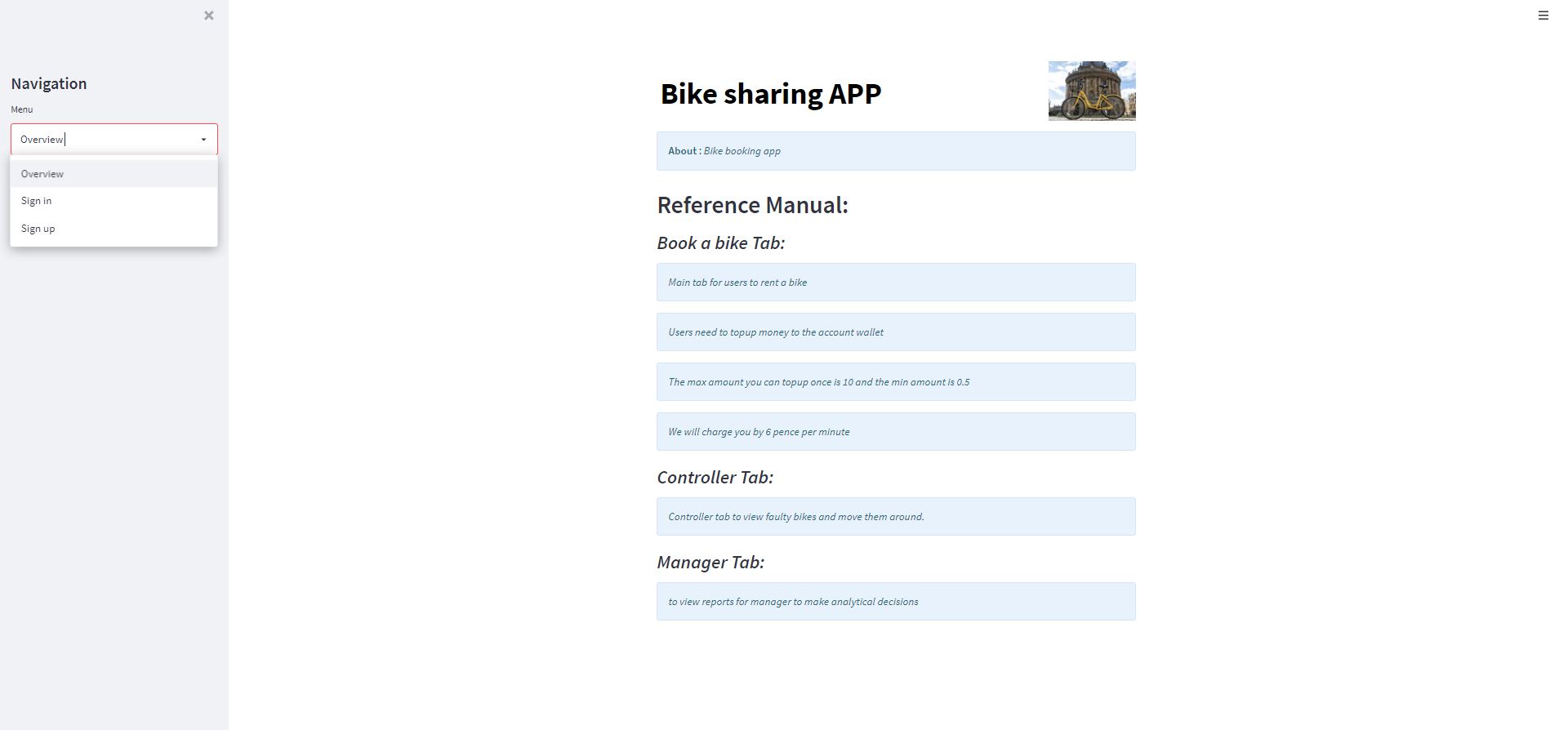


Figure A. 11 Booking, returning, and reporting defective bikes of customer prototypes.

# Appendix B

**Additional Tables**

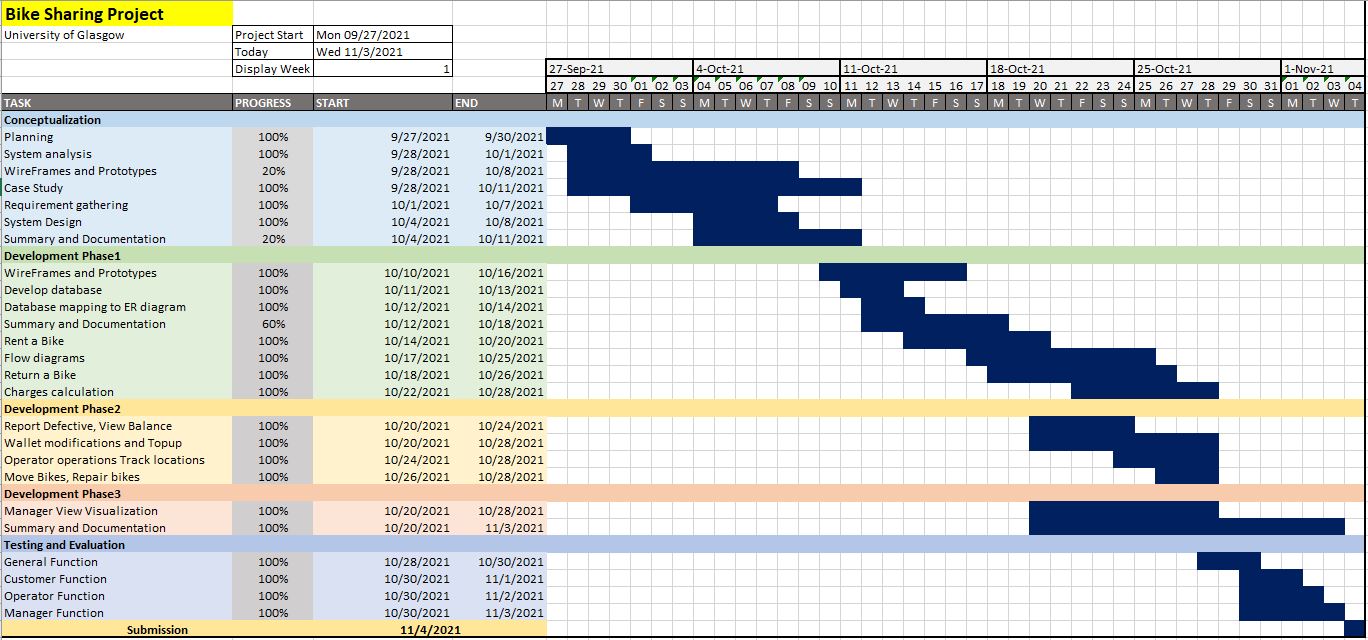


Table B. 1 Gantt Chart