Shared Bike System

Yang Haowen(2446324), Zhou yantong(2442809), Huang Can(2443320), You Xie(2433580), Li Yize(2447939)

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

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

06/11/2019

Abstract

The classic computer science education is losing its attraction in most university curriculums. Educators thus tried to find efficient and appealing teaching pattern in class to motivated their interests and cultivate more computer science graduates. One of the methods is objects-first strategy. It forces students to dive into the program immediately and therefore, they can understand the object-oriented language very soon. To be specific, the overall goal of this project is to build a bike share system to simulate a sharing bike system like nextbike in Glasgow and Mobike in China. This project can completely run to imitate the rough first generation sharing-bike, and is designed to reinforce the collaborative study as a teamwork, give them concepts of basic programming language learnt in class and enhance their programming skill while having fun.

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: Signature:

Acknowledgements

We would like to thank my supervisor Dr Mireilla Bikanga Ada and assistant tutors for guiding us to complete the whole project and all the advice they provided.

Contents

Chapter 1 Introduction

Computer Science has contributed a lot to the continuous progress being achieved in the field of Information Technology(IT). Accompany with increasing need for IT graduates and works, educators gradually realize that traditional methods of teaching IT skills doesn't work efficiently anymore since most undergraduates felt it is boring to practice tedious programming codes and frustrated when they failed in the exams or just struggled in compiling the projects in this era.

Recent researches concerning on the up-to-date way to import computer knowledge to higher-level students is mainly about Objects-first strategy(Cooper,2003). Objects-first strategy aims to attract students' attention and motivated their interests to dive into the projects using an object-oriented language like Java and C++ (Especially, this text uses python and MySQL) and eventually export a user-friendly GUI. The challenge is that when the beginner faces an objects-first task, he is required to dive into the theory and basement of the programming system immediately. Apart from all of this, basic learning path includes mastering the usual concepts of classes, variables, functions, and methods, as well as details of complex syntax(abid). In fact, there is general agreement that object-oriented programming is not only the core of modern software development but also an essential topic in computer science curriculum (Goldwasser & Letscher,2008).

1.1 Aim

The overall goal of this project is to build a bike share system to simulate a real scene where users can lend, return and accomplish a payment in their account; operators can obtain and modify the information of all bikes; as well as administer can receive the report of the whole system as their wish. It can also promote collaboration learning among postgraduates studying programming. The task is designed to teach students cope with challenging object-oriented programming problems as a way of reinforcing their concepts of basic programming language learnt in class and stimulate their interest in studying IT methods, which not only equips them with necessary IT skills, but also assists them have fun with programming.

1.2 Objectives

The objectives of this project are to develop a bike share system that:

a. Is user-friendly, which means users, operators and administer can keep in touch with the system GUI easily

b. Reinforces programming skills learnt in class and extends database knowledge

c. Improve the ability to solve object-oriented problem

d. Is engaging and needs individual's contribution

Chapter 2 Background Survey

2.1 Nextbike in Glasgow

Nextbike GmbH is the world's most extensive sharing bike provider and its branch released bikes in Glasgow. The service provided in Glasgow is a classic bike-sharing system. Its process are as follows:

New User must create an account by registering. When it comes to rent, customer input the bike number or scan the QR code to release the lock. Subsequently, after the bike is used, the user is ought to push the bike into a dock at an official e-bike station and lock it. Finally, a confirm of successful return will be sent to your mobile app to charge a payment (nextbike GmbH,2019). The overall progression is very the same to our project in the user's terminal.

Like nextbike, various sharing-bike providers have launched mutual bikes in almost each city throughout the world. Indeed, "Bike-sharing has had profound influence on creating a larger cycling population, increasing transit use, decreasing greenhouse gases, and improving public health", criticized by DeMaio (2009).

2.2 Fourth generation sharing-bike

Known as fourth generation bike, the dockless bikes equipped with a GPS are free-floating, which means they can be parked anywhere and doesn't require an official station. In this project, we connected the Google Map API to the database to track each bike. For instance, in China, the world's largest bike share operators Mobike released millions of free-floating bikes spread over 100 cities (DeMaio,2009). As the price of fuel rises, traffic congestion worsens, populations grow, and a greater world-wide consciousness arises around climate change, it will be necessary for leaders around the world to find new modes of transport and better adapt existing modes to move people in more environmentally sound, efficient, and economically feasible ways. Bike-sharing is evolving rapidly to fit the needs of the 21st century. The provision of sharing bike is clear: as the price of fossil fuels arise rapidly, and a greater consciousness aroused of climate change, it will be vital to find eco-friendly way for transportation and human's brighter future.

Chapter 3 Requirements

Through the research and analysis of the existing shared bicycle systems in market, the main functions of similar systems are determined. Based on similar systems' main functions and the project specification, the requirements of this project are determined. This section will discuss these requirements in detail, which will help to refine the project and clarify the main objectives of the project.

3.1 Functional Requirements

• Login

A login page which can grant user, manager, and operator access according to different account types.

• Main page for customers

An operation page can deal with customers' commands such as renting bikes, return bikes, reporting defective bikes. It also shows the location information of all bike stations.

• Main page for operators

Operators can track bikes location information, repair bikes, and move bikes' position through this page

• Main page for managers

Get a visualized report which can be file

• My account

A page shows users' recent bills, history bills, and account balance. Users can pay their recent bills. Balance can be updated automatically.

• Register

3.1.1 User Stories

The main users of shared bikes system are customers, operators, and managers.

Customers

As a customer, I want to view all the bike stations' position information

As a customer, I want to rent a bike through this system

As a customer, I want to return a bike

As a customer, I want to login in to this system through registered account

As a customer, I want to report a defective bike

As a customer, I want to pay for my recent bill

As a customer, I want to check my account balance

Operators

As an operator, I want to view all the bikes' position information

As an operator, I want to change a bike's position

As an operator, I want to change defective bike's status information once repairing is finished

As an operator, I want to login to the system to operate the bikes in this system

Managers

As a manager, I want to see visualized report of bikes situation.

3.2 Non-functional requirement

Easily usable:

The system is expected to user friendly

Extensible:

The system can add some new functions in the future

3.3 MoSCow Analysis

3.3.1 Must-have

Main page for customers: customers can control this system (rent, return, and report) through clicking buttons in this page. In addition, it can forward to my account page.

Main page for operators: operators can control this system (move, view, and repair) through clicking buttons in this page.

Main page for managers: managers can view report through clicking a button in this page.

My account: customers can check balance and pay for their recent bill

3.3.2 Should-have

Register: make sure this system can add new account.

Login: shared bikes system should include login system to correctly assign user privileges. Customers can have right to rent a bike, return a bike, report a defective bike. Operator can have right to check bikes' position information, repair bikes, and move bikes. Managers can have right to see visualized report.

3.2.3 Could-have

Account management: customers can add their extra account information such as address to receive paper bills. In addition, customers can change their password or find their lost password.

3.2.4 Would not have

Real-time GPS of bikes: it will track all bikes position information through GPS, this system would not consider real-time tracking.

Customers feedback: it can be used to get users feedback, there would no feedback system.

Chapter 4 Design and Implementation

Shared Bike System is one system but provides two different services to both user and stuff. It is built to provide bike rent service and it includes each necessary function to finish the whole rent-return-pay process. Aside from fulfilling required function, we also add Google Map API into our system for user to have a much better experience to use this program. Also, it is built to provide some management services for both operators and managers. Including daily work like viewing all the bikes, repairing bikes and moving bikes. It also provide a very useful function for managers, which is generating chart to illustrate recent bike activities by one click.

4.1 Design pattern

In general, we try to implement the MVC frame to build our software to make us manage our software easily in the future and separate the UI and the business logic clearly.

The Model is the part of the application that handles the logic of the application data. The model object is responsible for accessing the data in the database.

The View is the part of the application that processes the display of data and provide interactive interface for users.

The Controller is responsible for reading data from the view, controlling user input, and sending data to the model.

In this system, when the user clicks the button or enters data in the view layer, the controller obtains the user's behavior and data, calls the appropriate model, the model passes the processed data to the controller, and the controller refreshes the page to display the new data.

Advantages of our design:

Separating all business logic into the Controller makes it easier to manage subsequent updates (when business logic changes, you only need to change the Controller, not the View and Model).

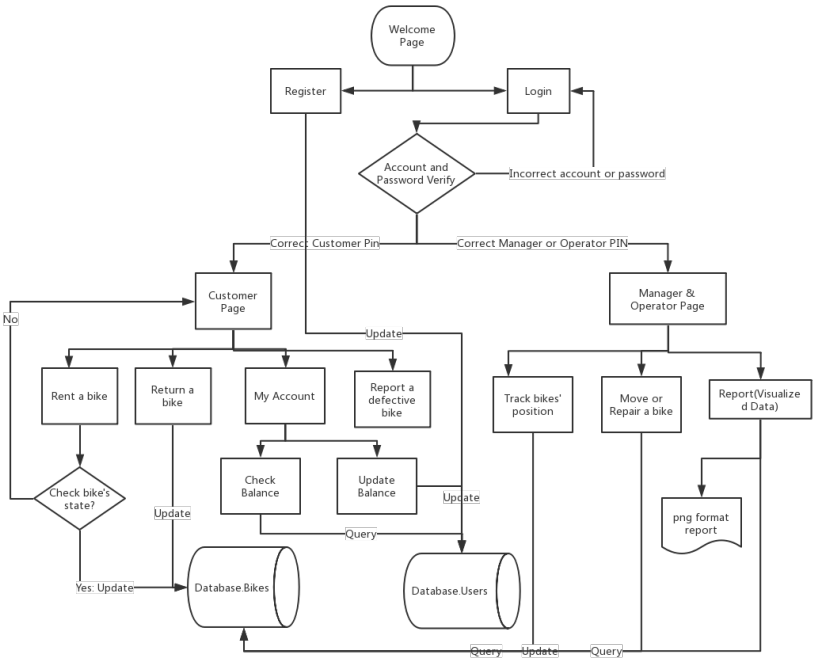

Figure 1: Logic Diagram

4.2 ER Diagram of Database

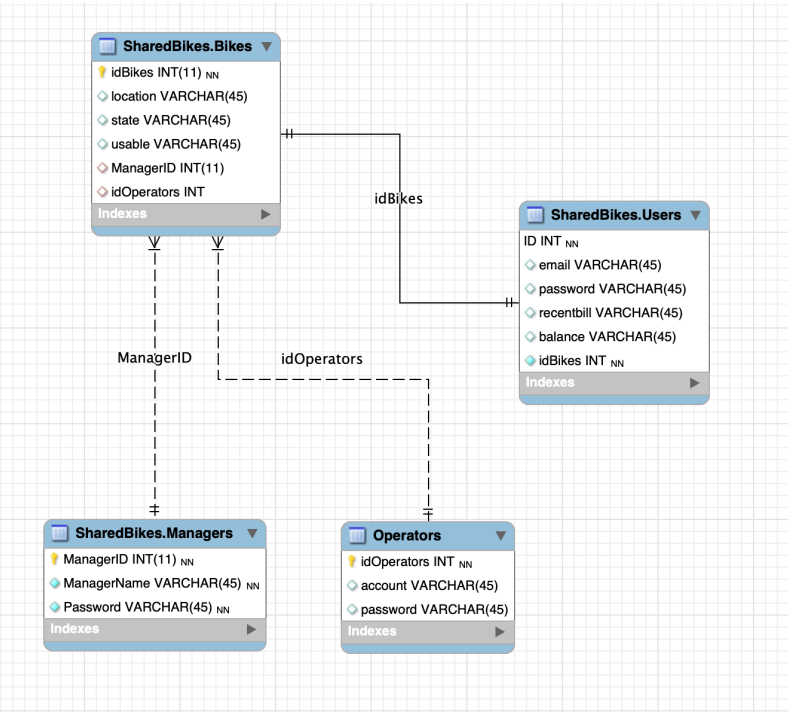

Figure 2: ER Diagram

4.3 Implementation

4.3.1 Login & Register System

The user or operator or manager will be firstly shown this page when the system is launched. This page provides two main function. First one is to validate whether the man is a registered user or stuff. By inputting correct username and password, the man can login to corresponding system. Otherwise he will be warned for a wrong username or password. No further action needed because it will distinguish user and stuff automatically. If he is not registered yet(only for user), he can register into our system by click Register button and input his information.

See Figures below for login design.

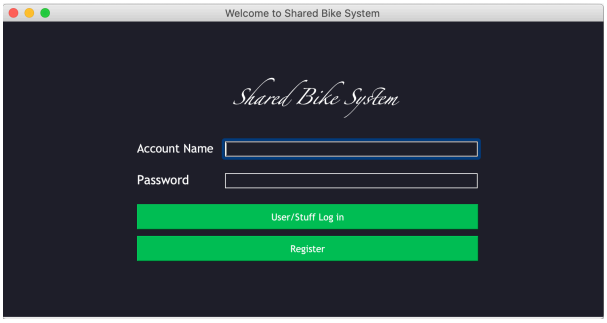

Figure 3: Login Page

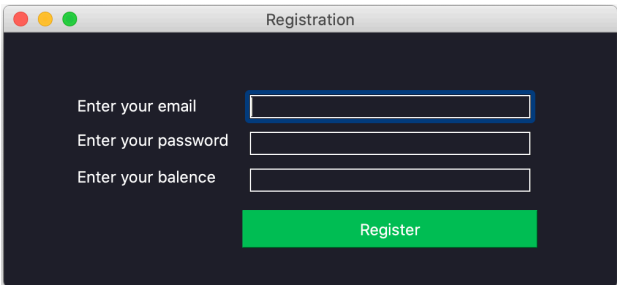

Figure 4: Register Page

4.3.2 User Interface

Once user entered our shared bike system, he will be shown a map with many functional buttons around. The map is moveable and scalable by clicking the + or – button in lower right corner. There are markers on the map showing all bike stations and their station number. User can input the station number only to check for all the bikes in that station. Then choose one of them by inputting that bike's id into the box lies above the Rent button. If this bike user chose is in good condition, he will receive a message box saying Success. Otherwise he will be warned this bike is broken or being used. Besides, user can report a defective bike by inputting bike id and clicking Report button. After riding bike, user can input the station number to tell the system which station that bike will be returned. And system will record that then calculate the cost. When the whole process is done. User can click My Account Button to check his balance and bill information. Sometimes it can be delayed so Refresh button can help you get the true information. Pay button can do as it says, pay the bill.

See Figures below.

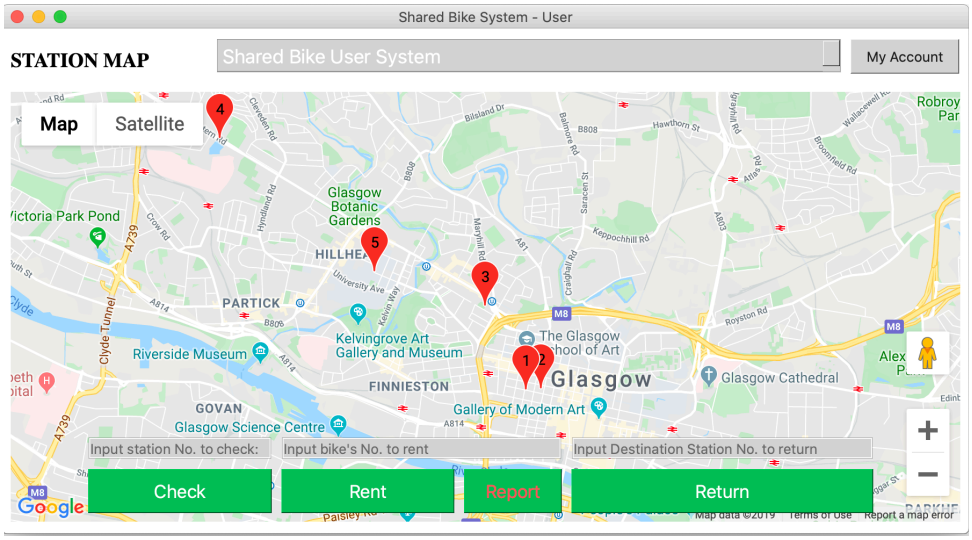

Figure 5: User main window

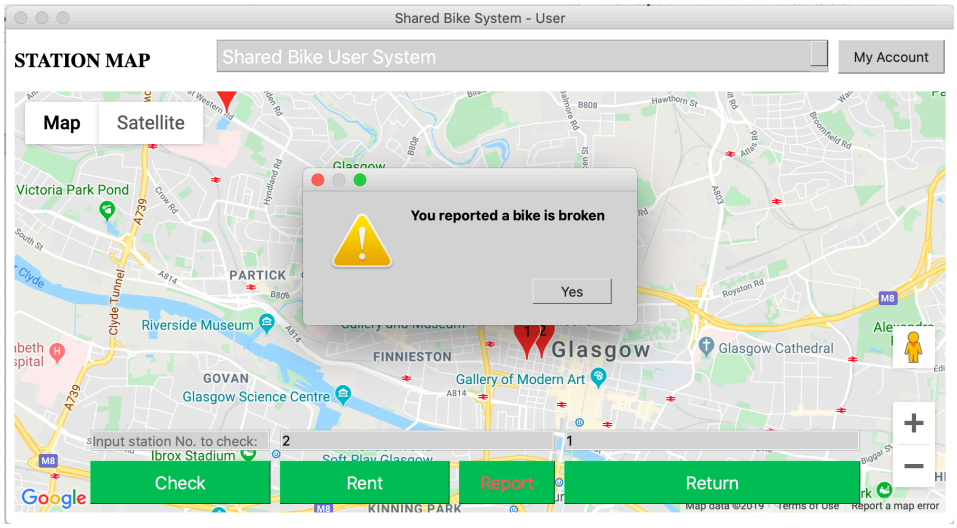

Figure 6: Report a broken bike

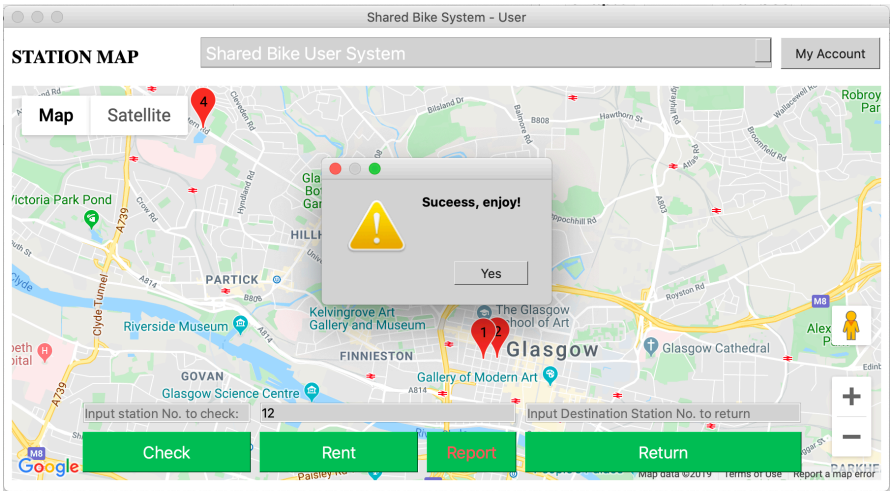

Figure 7: User successfully rent a bike

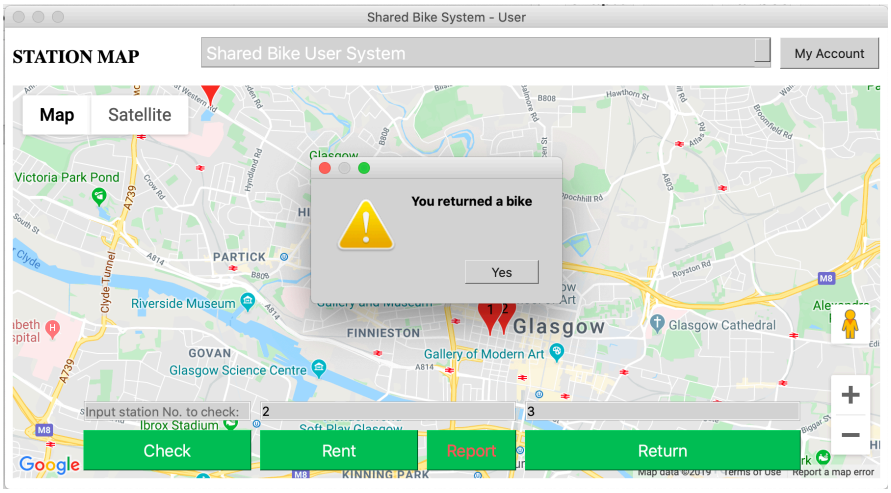

Figure 8: User successfully return a bike

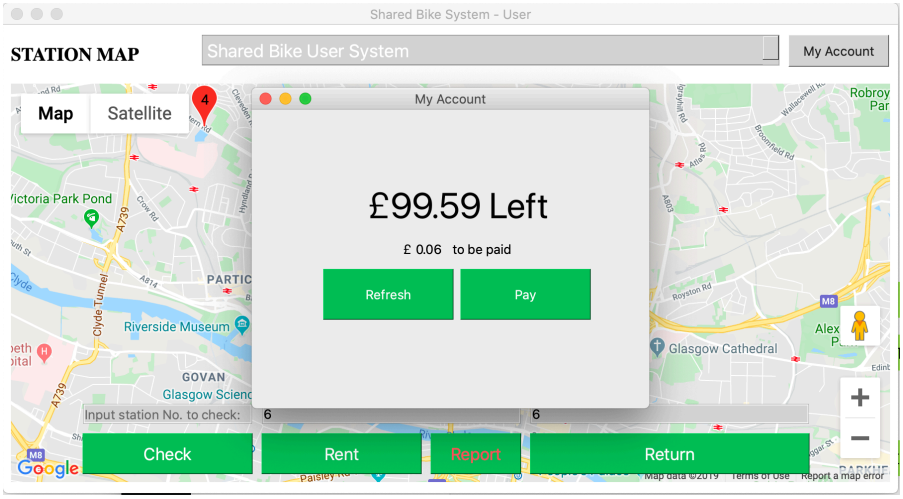

Figure 9: My Account interface

4.3.3 Operator & Manager Interface

When Operator or Manager enters the system, a all bikes current status list will be shown. The order is by station number and bikes' id. Stuff can check all bikes' status and decide what to do next. Once a broken bike has been found, an operator can go to that station to repair it. Then input that bike's id and click Repair button. It can be put into use again. The status of that bike will be updated in the database as well. If operator found there are too many bikes in one station, he can choose to move bikes. Also, by inputting bike's id and destination station number, the data in database will be updated. Finally, if a manager want to see a report. He can simply click Generate Report button to check it.

See figures below.

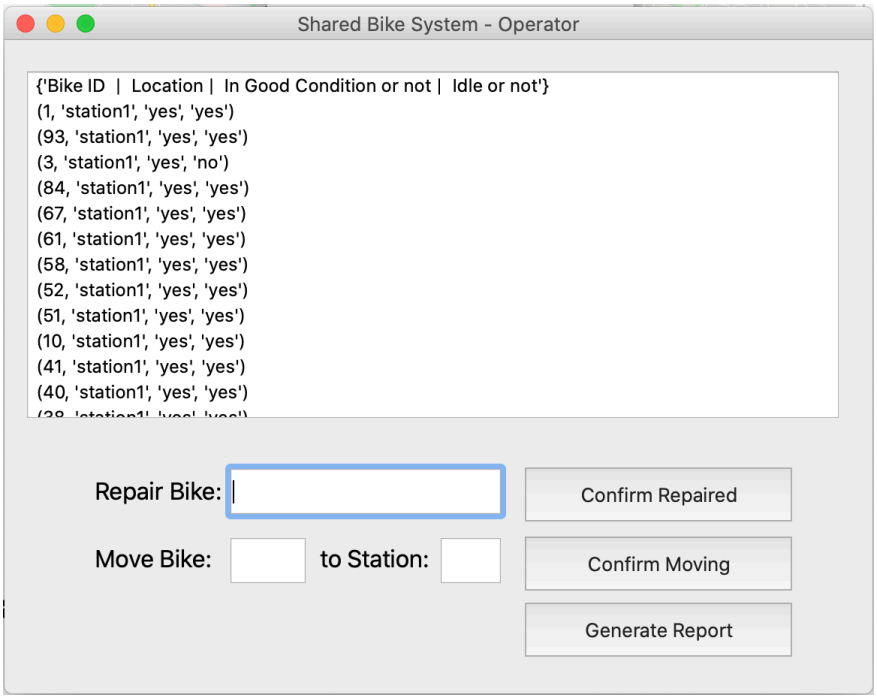

Figure 10: Stuff main window

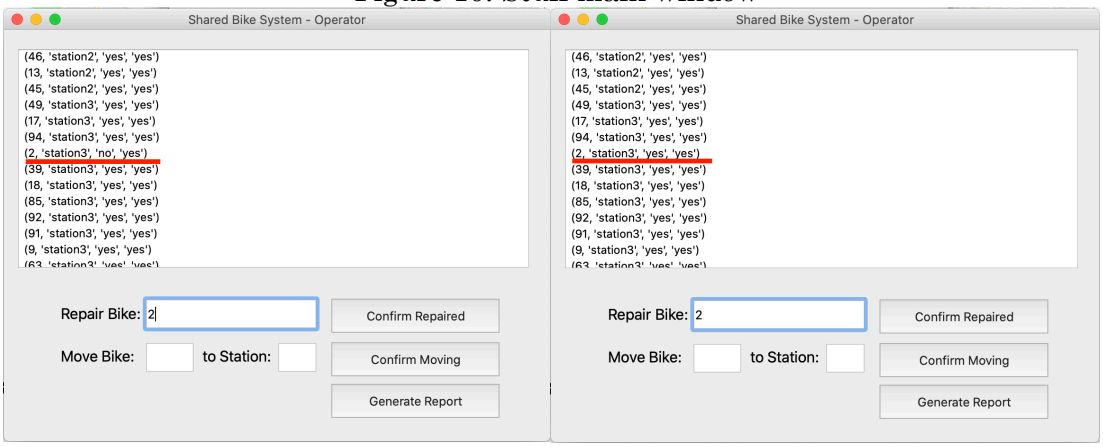

Figure 11: Confirm repairing a bike

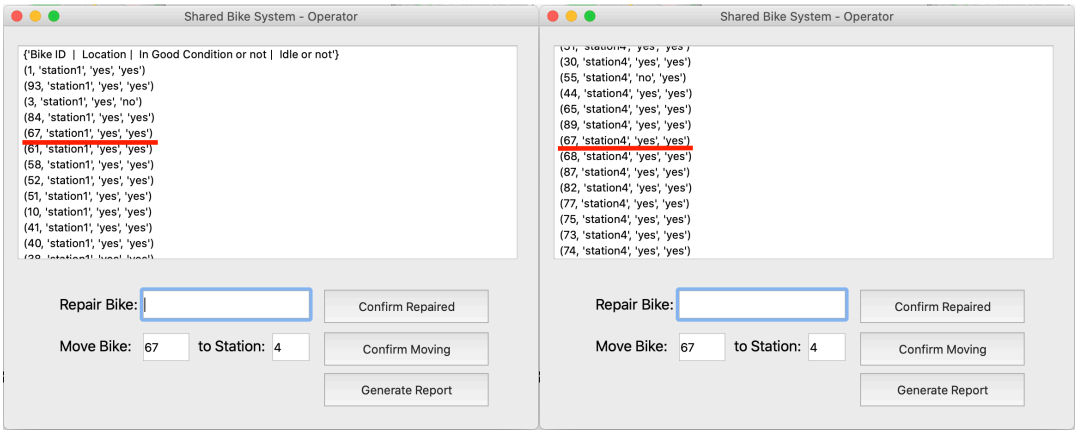

Figure 12: Confirm moving a bike

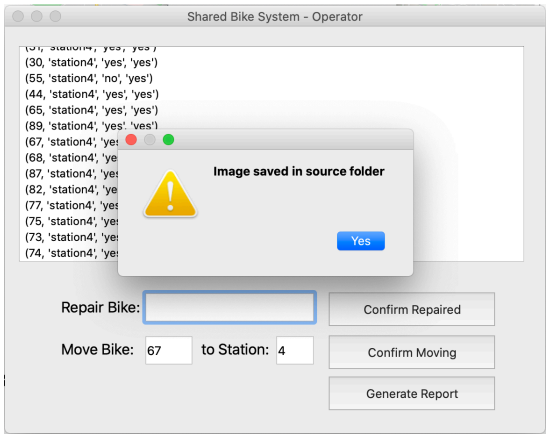

Figure 13: Generate report

4.4.4 **Data Visualization**

First of all, we need to produce a figure to show bike distribution so that the managers could see where the bikes are. After that, we produce a pie chart to make it clearer to see the proportion of the number of bikes in different stations so that the manager could determine if they should move bikes in one station which has many bikes to another station which does not have enough bikes. The two figures are as follows:

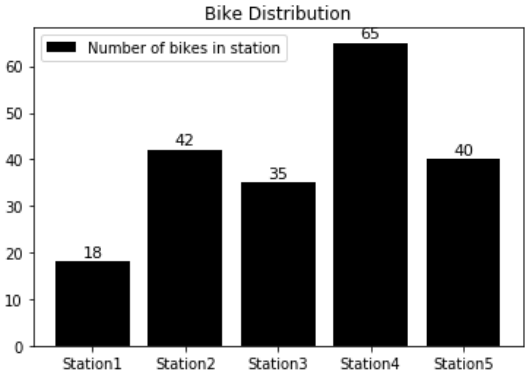

Figure 14: Bike Distribution

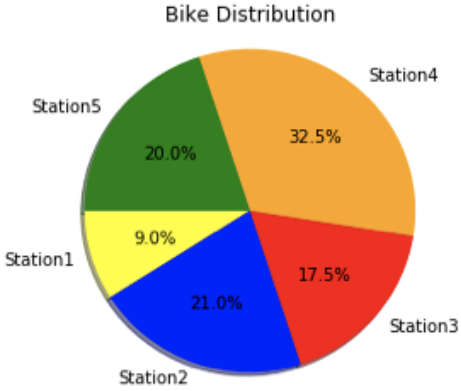

Figure 15: Bike Distribution (pie chart)

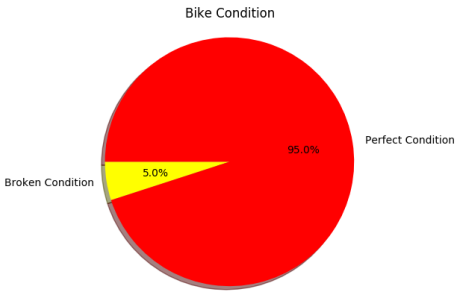

Figure 16: Perfect Bike Rate

4.4 Technologies

4.4.1 MySQL

MySQL is is an open-source relational database management system which, in our case, is used as our database system for storing all users, managers and bikes' information. MySQL is very simple to use and have great compatibility with python. By using it, our system can easily access lots of data and run properly. It also can be manipulated with standard SQL commands.

4.4.2 Workbench

Workbench is a unified visual tool for database management. During the development and test process, workbench makes it a lot easier to check how our program is running and whether it do as we programmed it to manipulate the database.

4.4.3 PyQt

PyQt is a set of python v2 and v3 bindings for Qt application framework. It provides a visual interface to design UI other than just coding. It is a tool of Qt Creator and works well with python. Besides, after designing your interface, we also use Ui2Py tools to convert.ui file to executable.py file. By importing them to main file and bound some functions to some buttons, it can be used directly.

4.5.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

4.5 Methodology

We follow the Seven Phases of the System-Development Life Cycle(Architects, 2018) to design our program. It clearly shows a pattern to develop a complete useable system, develop and test step is crucial cause some function has already been given and we do not need to think about them. It also fits well with our Shared

Bike system, especially in Test part. Because the algorithm is not difficult but requires handling many deviant branches to maintain the stability of this system.

Chapter 5 Testing

5.1 Testing

In order to test the reliability of the software, we divide the testing process into three main parts, which are unit testing (component level testing), system testing and acceptance testing. The purposes of testing are related to 'Defect Testing' and 'Validation Testing' since not only the flaws should be discovered, but also the requirements for the module should be fulfilled. As a 'Customer Software', different techniques during different testing stages were used. This chapter will discuss the detailed testing process.

5.1.1 Unit testing

```
import unittest
from Bikes import *


class Test_Fun(unittest.TestCase):


#Test functions in Bikes class
    def test_CheckState(self):
        bike = Bikes(1)
        self.assertEqual(1, bike.CheckState())
        bike = Bikes(3)
        self.assertEqual(2, bike.CheckState())
#Test functions in Customers class

    def test_Register(self):
        self.assertFalse(Register(111, 111, 100))


    def test_Login(self):
        self.assertEqual("user", Login(111,'111'))
        self.assertEqual("wrong", Login(111,'123'))

if __name__ == '__main__':
    unittest.main()
```

Figure 14: Unit testing code example

Figure 14 shows the partial code of the test. This test tested all methods in bicycles, customers and operators class. assertEqual( ) can compare whether the value is identical, and assertTrue / assertFalse is the boolean type judgment。

```
Ran 4 tests in 0.036s

OK
```

Figure 15: Partial testing result

In unit testing, we used white-box techniques. In each selected component test, it tries two conditions: correct situation and incorrect situation in order to ensure its validity.

5.1.2 Black-box Testing and System Testing

In the black box testing phase, the project tried to stand in the user's perspective, trying various usage scenarios, ignoring the internal logic of the code. For example, on the login page, the test will try to enter the correct password, enter the wrong password, enter a username that does not exist, do not enter a username, do not enter a password, and so on. In this test phase, the GUI interface is connected. System testing is performed concurrently with black box testing. Whenever a class is completed, the method of that class will be connected to the previous network segment, and at the same time, the user's usage will be tested. When the new class is completed, the new and old methods will be tested again. The entire system testing process uses regression testing techniques. After the final project is completed, check that all user needs are met.

Chapter 6 Conclusion & Future Work

6.1 Current Status

The purpose of the project is to develop a bike shared system that is able to rent a bike at any location in the city as long as there is a working bike available at the location; return a bike to any location, when a customer returns a bike, their account is charged an amount depending on how long the bike rental was; report a bike as defective; track the location of all bikes in the city; repair a defective bike; move bikes to different locations around the city as needed and generate reports showing all bike activities over a defined time period. Firstly, we make a login and register system which allows users, operators and managers to register and login to corresponding system; after that we make the user interface, when the user enters our system, he will be shown a map with many functional buttons which allow him to rent, report defective bikes, check bikes in stations and return bikes. We also make the operator and manager interface, when operator or manager enters the system, they can see a list of current status of all the bikes. When it comes to broken bikes, the operator will go to the station to repair it. The status of bikes will be updated in the database so if the operators find there are too many bikes in one station, they can move bikes to the other stations. Finally, if the manager wants to see a report, he can just click Generated Report button to see the plots about bike distribution.

6.2 Future Work

This system still need to be improved by adding some security function to keep bikes from being stolen or being reported broken by accident. Besides, this system need to be transplanted to mobile device for a better user experience because under most circumstances, all actions should happen outdoor near the station instead of in front of a computer indoor. And more functions for operators and managers should be developed to increase efficiency.

References

[1] Cooper, S., Dann, W., Pausch, R., & Pausch, R. (2003). Teaching objects-first in introductory computer science. ACM SIGCSE Bulletin, 35(1), 191-195.

[2] Goldwasser, M. H., & Letscher, D. (2008, June). Teaching an object-oriented CS1-: with Python. In Acm sigcse bulletin (Vol. 40, No. 3, pp. 42-46). ACM.

[3] nextbike GmbH, 2019. Retrieved from

https://www.nextbike.co.uk/en/glasgow/information/

[4] DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. Journal of public transportation, 12(4), 3.

[5] John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; The Matplotlib development team,2019. Retrieved from https://matplotlib.org/

[6] Architects, I. (2018). The seven phases of the system-development life cycle. Retrieved March, 12, 2018.