



School of
Computing Science

Bicycle Sharing System with programming

Cunxu He

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

Monday 1th of March 2021

Abstract

The inconvenience of travel and the cost of transportation have increasingly become a problem. For example, public transportation usually only provides services on specified road sections, so people can find it inconvenient because they have to walk from stations to their home, although this kind of method is cheap. By contrast to taxi, it is quite expensive. In order to solve this problem, our team designs a bicycle sharing system which can successfully and perfectly meet the requirements of customers. This system will not only solve the consumers problems of travelling, but also help the operators and managers to have a good grasp of operation.

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

I would like to appreciate my supervisor Mireilla Bikanga Ada a lot for all the support and encouragement from her. I also want to say thank you to all the participants in my group. Without everyone's cooperation and hard work, this project will not be accomplished successfully.

Contents

Chapter 1	Introduction.....	1
Chapter 2	Background.....	2
2.1	Bike sharing.....	2
2.2	Related work.....	3
Chapter 3	Requirements Analysis.....	4
3.1	functional requirements.....	4
3.2	Non-functional requirements.....	5
3.3	Moscow analysis.....	5
3.3.1	must have.....	5
3.3.2	should have.....	6
3.3.3	could have.....	6
3.3.4	won't have.....	6
Chapter 4	Design and Implementation.....	7
4.1	System Architecture.....	7
4.2	Implementation.....	7
4.2.1	login and register.....	7
4.2.2	rent and return a bike.....	8
4.2.3	payment.....	9
4.2.4	report the defective bike.....	9
4.2.5	operator.....	9
4.2.6	manager.....	10
Chapter 5	Testing.....	11
Chapter 6	Conclusion and future work.....	13
6.1	Conclusion.....	13
6.2	Future work.....	13
Chapter 7	Contributions.....	15
Appendix A	<database of system>.....	1

***Note:** The chapter headings are by no means prescriptive. Instead use an appropriate report structure for your project as you see fit.*

Chapter 1 Introduction

With the development of economy, people's needs have become more diverse and are more urgent to be satisfied. In the past, when we mentioned the modes of transportation, we hope to choose the fastest mode. However, nowadays people think more highly of the convenient and healthy modes of traffic. Underground and bus have become the main traffic modes in daily life in recent years, and these ways are truly cheap and useful. But the disadvantages of these ways are that they are not flexible enough, which means that they normally have specified and unchangeable routines, and this cannot meet people's requirements to reach home directly. Taxis and uber are convenient enough but they might be a bit expensive. Bikes seem to can perfectly satisfy 'the-last-one-mile' (from station to home) problem [1], but this mode still have its disadvantage. The main issue is that people do not want to purchase a bike and they want to enjoy the convenient brought by bikes. In order to satisfy the demands of customers, we design a bicycle sharing system which can offer convenience for both customers and managers.

According to this bike sharing system, it is not only provided for the customers, but also for operators and managers. Customers can know what place they can find an available bike through this system. After unlocking the bike by scanning the unique code, they can acquire the usage of this bike. When they want to return the bike, what they only need to do is stopping the bike in the Designated parking area if they pay the charges on their account. As for the operators and managers, it is very easy for them to manage these bikes. operators can have a good understanding of the exact location of every bike. This function is very important, because some customers may not obey the rules and park the bike in the wrong places. By the way, they can receive the report of defective bike from customers which can help operators to find these defective bikes and repair them easily. Managers could see the analyzing charts which are created by this system, and this is very useful to make relative decisions.

This report will record all the lifecycle of the project above. The rest of report mainly has five sections. First, chapter 2 will introduce the background survey of this system, including some issues related to bike sharing and related work. Chapter 3 will analyze the functions of this system, and which parts of functions are crucial. Chapter 4 will indicate that the design and implementation of this system, which can see how we implement this system step by step. Chapter 5 will demonstrate the test of this system to prove this system can meet the requirements. Finally, chapter 6 will provide the conclusion and what we will do about this project in the future.

Chapter 2 Background

2.1 Bike sharing

Since the 21st century, shared bicycles have entered a stage of vigorous development. Just as we have mentioned, shared bikes perfectly meet the requirements of customers like university students and short-distance commuters. Apart from this advantage, bike sharing can enrich the composition of urban public transport system, which means when citizens solve their travel demands they can have more choices.

In fact, bike sharing is not a new thing anymore. The world's first public bicycle initiative is believed to be Witte Fietsen, or White Bikes, launched in Amsterdam in 1964 [2]. At first, this bike sharing program was created to solve the problem which is bicycle theft. This is because city planners believe that if all the bicycles are free, nobody will steal bicycles. But unfortunately, this plan failed without much implementation. When entering the early 1990s, bike sharing started to have some new functions. When customers wanted to use the bike sharing, they needed to pay the charges rather than pay nothing [3]. And, bikes were not scattered in every corner of the city. These bicycles were docked at some specific places, and this gave a great convenience for managers to manage those bikes.

Nowadays, with the development of technology and economic progress, people use bike sharing more convenient, and bike sharing also has more advantages to adapt to the times. For one thing, the increase in the use of bicycles is of great benefit to environmental protection [4]. Cycling can not only reduce carbon emissions and greenhouse effect, but also reduce people's dependence on fossil energy [5]. For another thing, riding bicycle can be to exercise. It is healthier than driving and riding transit.

Apart from the advantages, the use of bicycles usually leads to some bad phenomena. First, Safety issues caused by bicycle travel may become the focus of social attention. Bike safety varies widely from place to place. Some cities are concerned about the maintenance of bicycle lanes, while others seem to ignore this point, which may become hidden danger. As local people, people may know the condition of the road very well, but for tourists, when riding on the road with unknown road information, it is very easy to have safety problems. Secondly, the management of bicycle dock may become a problem [6]. For some popular scenic spots, there are usually more bicycles than the ruled number of bicycles, while in some other places, the number of bicycles can be very few. Although we can set up special operators to move bicycles, it will increase the operating costs, which is not conducive to the effective operation of the company. Finally, the excessive wear and tear of bicycles will cause the waste of social resources. In the daily operation process, it is very normal for bicycles to be damaged during riding, so operators are usually sent to repair them. However, if the vehicle damage rate is higher than the normal limit, such as improper use of consumers in the process of riding, malicious damage of pedestrians when parking, and unreasonable setting of bicycle docks, the vehicle damage rate will be too high. Moreover, when

the bicycle sharing system stops operation, there will be a large number of unclaimed used bicycles in the society, the disposal of these wastes will also become a problem.

2.2 Related work

In this chapter, review the current tools and websites that do similar things to what your tool is intended to accomplish. Maybe also look at some academic references? Put all the tools, websites, and references into your Bibliography.

Several applications have been put on the market in order to solve the users' need of using bicycles. So, in this section, we will review these applications and discover how they meet the requirements of users.

MeiTuan bike, hello bike and orange bike are three main bike share applications in China, and the first two have mastered the main market. All of these applications have satisfied the demands of using bikes and also their payment is very close, but the first two have mastered the main market which is mainly because they have better functions, such as reporting defectives and user experience.

MeiTuan bike and hello bike both are web-based too applications which are mainly paying attention to sharing and using bikes. About the location of using bikes, both set up the use area and punish the users who park the bicycles outside the parking area. And, on the interfaces of these two applications, we can find the button easily which is used to report the defective bicycles and other impolite behaviors, such as locking the bike personally and scraping off the unique bike code. About the payment, the wallet is connected many different payment methods, like WeChat pay and credit card, which is very convenient.

In contrast to orange bike, this application doesn't have a place to submit the defective bikes report and when users park the bicycles outside the area, it will punish more severely the other applications. However, the main advantage of it is that it provides more options to the users, which is users can choose to use electric bikes. This is very important for orange bike to survive from other competitors.

One significant limitation is that these applications fail to solve many problems absolutely, such as parking bikes in chaos and the low efficiency of bike using. They only can tackle these problems by punishment or hiring employees to relocate the bikes. These actions are useless and in some special positions, people still cannot find enough bikes to use.

Chapter 3 Requirements Analysis

3.1 functional requirements

- track bike:

Operators can know the location and status of each bike.

- Repair bike:

Operators can repair the defective bike and change the status of bike.

- Move bike:

Operators can move bikes from a location to another location.

- Generate reports:

Manager can get a report. The report shows all bike location change and status over a time period.

- Rent bike:

This function allows the user to rent a bike from a selected location if there is at least one available working bike.

- Return a bike:

Allows the user to return the used bike to any selected location if there is at least one available parking.

- Report a bike:

The user can send a report about a defective or damaged bike.

- Payment:

This function calculates the total charge depend on the journey length, and allows the user to pay the full amount.

3.1.1 User stories

Customers

- The bike-share application aims to serve three types of users. They are the service consumers, operators and managers.
- Service consumer:
As a service consumer I want rent a bike from any location.
As a service consumer I want to be able to return the bike to any location.
As s service consumer I want to able to pay the full amount.

Operators

- As an Operator I want to track the location and status of bike.
- As an Operator I want to repair a bike when it is defective.
- As an Operator I want to move bikes from a location to another location because of the user needs.

Manager

- As a manager I want to get a report which shows all bike actives.

3.2 Non-functional requirements

- Effectiveness :
Can be used in any time.
- Easily useable :
Operators and managers may not be proficient in software.d
- Security:
Others cannot steal information easily.

3.3 Moscow analysis

3.3.1 must have

- Customers can rent and return a bike.
- Customers can report the defective bike
- Customers can pay for the charges
- Operators can track and move the bike
- Operators can repair the defective bike reported by customers
- Managers can acquire the data analyzing charts

3.3.2 should have

- Administration page:

This page is designed for the three kinds of users to log in this bike sharing system. For customers, they can choose to register or log in the system and then gain the access to it, so every customer will have their own id and password. For operators and managers, they will only have password, because there won't be too many operators and managers.

- Adding bikes:

Operators can add a new bike to increase the scale of the bike resources.

- Interactive:

The interface of the system should be easily understood.

3.3.3 could have

- Membership card:

Customers can purchase the membership cards in order to get the discount at the time of payment.

- Feedback:

Managers can gather the information of the customers' idea through this function, which can help us know how to improve the system.

3.3.4 won't have

- Distance detection:

According to the survey of existing related applications, the charges normally are related to the time which can be more easily to calculate.

Chapter 4 Design and Implementation

This chapter will discuss the design and implementation. Our bike sharing system is mainly designed for university students and commuters with short-distance needs. Customers can temporarily acquire the usage of bicycle by renting a bike. Since this system will provide a large number of bikes, we need operator and manager to manage them.

Describe what you created, and how you built it. Talk about libraries you used, design patterns, programming techniques etc. Maybe include some UML diagrams to show overall structure.

4.1 System Architecture

In the design stage of software system, the choice of programming language is very important. After discussion, our team chose to use Python. In the aspect of database, we use SQLite to provide data support. For the internal architecture of the whole system, we use case diagram to explain in detail. The reason why we choose case diagram to design is that case diagram is a model diagram of how the system functions as observed by actors, and it is also a blueprint for the system.

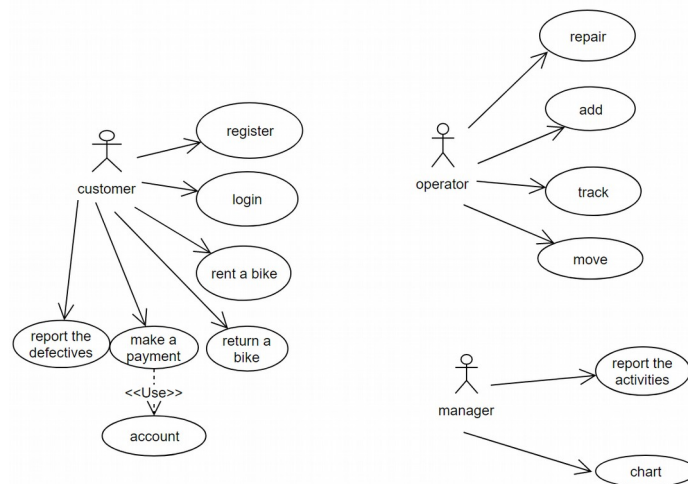


Figure 1

4.2 Implementation

4.2.1 login and register

The login page is the first page when users enter the system, and different kinds of users will see different pages. As for customers, they can see figure 2. This page will have two functions, login or register. If the customer is the first time using our system, he/she needs to enter their id and password to write data into our database. Otherwise, they can enter their id and password and click login button directly. The username and the password will be checked if this

combination in the database or not, if not the system will ask the user to create a new account. As for manager and operator, they can see figure 3. Because we only have one operator and one manager, we do not need allocate id for them. They only need to enter related password and click confirm button. The password of operator is 0 and the password of manager is 1.

Figure 2

Figure 3

4.2.2 rent and return a bike

In this function, customers will rent a bike and return this bike in the specific location. When they decide to rent a bike, they need to select a location at first, and then system will allocate an available bike to customer. After clicking the rent button, customer will have the usage of bike and the timer will begin to count. When customer wants to end the trip, he needs to select the location which is near his destination.

Figure 4

Figure 5

Figure 6

4.2.3 payment

when the user is the first time open the system, the account is empty. He needs to click the check the account button and charge money button, then the account will have 10 pounds. When he wants to end the trip, system will calculate the payment. When the fees are settled, the balance in the account will be automatically deducted.

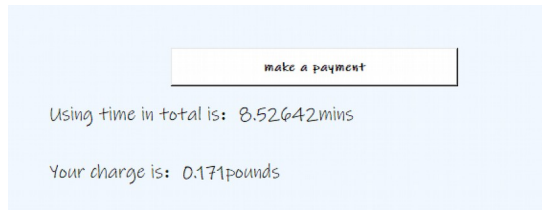


Figure 7



Figure 8

4.2.4 report the defective bike

If customer finds the bike is defective, he can send a report the operator.

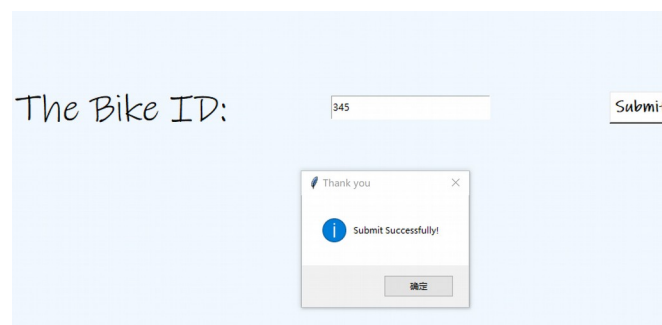


Figure 9

4.2.5 operator

As for operator, he needs to track and move all bikes, and repair the defective bikes. In addition to this, he also can add a number of new bikes.

Operators can enter the id of a bike and click track button to track a specific bike in order to know where this bike is and the status of this bike. Operator can also set up the limitation like the status and location to get data of bikes. If he wants to repair a defective, he can select a defective bike and click repair button to change the status of the bike. The status of bike will be changed from defective to normal. Moving and adding bikes are similar.

Track one specific bike by its ID:

Track bikes in location

To

To

ID	Status	Location
0	working	1
1	working	1
2	working	1
3	working	1
4	working	1
5	working	1
6	working	1
7	working	1
8	working	1
9	working	1
10	working	1

1853 result(s)

Figure 10

4.2.6 manager

When entering the manager page, users need to enter the beginning time and ending time, and then click the report button. After these actions, users can get all the activities of all bikes. Also, when clicking the chart button, the system will show all the bikes whose status are riding.

Generate reports showing all bike activities

Start time: 2021-02-12 09:00:00 End time: 2021-02-28 09:00:00

(input like 2021-02-12 09:00:00)

Bike Id	User Id	Start Time	End Time	Start Location	End Location	Activity Type
8	42	2021-02-12 09:22:42	2021-02-12 09:22:42	None	None	repair
99	43	2021-02-12 09:53:56	2021-02-12 09:53:56	1	2	move
32	44	2021-02-12 10:01:00	2021-02-12 10:01:00	None	None	repair
530	45	2021-02-26 17:22:09	2021-02-26 17:22:09	5	None	add
531	46	2021-02-26 17:22:09	2021-02-26 17:22:09	5	None	add
532	47	2021-02-26 17:22:09	2021-02-26 17:22:09	5	None	add
533	48	2021-02-27 12:23:35	2021-02-27 12:23:35	2	None	add
534	49	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add
535	50	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add
536	51	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add
537	52	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add
538	53	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add
539	54	2021-02-27 19:30:38	2021-02-27 19:30:38	1	None	add

Figure 11

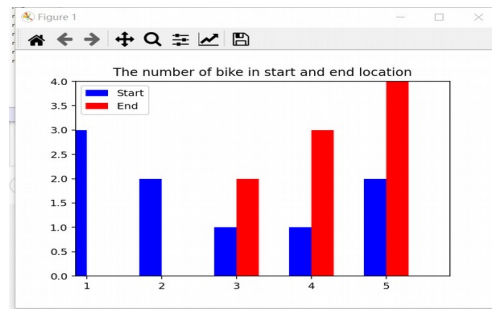


Figure 12

Chapter 5 Testing

Testing and evaluation are key stages of development process in this bike sharing system. Through these actions, we can easily comprehend which parts have been implemented well, and which sections need to be improved.

Test Case	Description	Result
Register	Enter user name and password and click register button to write the data in the database. This function is only related to the customers. After registration, customers can directly enter the interface of renting a bike.	Pass
Login	Customers: enter username and password and click the login button. Then customers can be directly to the 'rent a bike' page. Operator and manager: only need to enter the right password (oprator:0 manager:1) and click confirm button. Then they can be directly to the Corresponding operation interface.	Pass
Rent a bike	Customers can click select button to choose the closest bicycle dork to rent a bike. The system will show the beginning time. If this place does not have an available bike anymore, system will give a message.	Pass
Return a bike	Click the select button to choose the closest dork to return bike. The system will show the ending time.	Pass
Payment	Click the payment button, the system will calculate how much you need to pay. And you can also check the account to know your money.	Pass
Report the defective bike	Enter the id of defective bike and click report button	Pass
Track bikes	Operators can enter the id of a bike and click track button to track a specific bike in order to know where this bike is and the status of this bike. Operator can also set up the limitation like the status and location to get data of bikes.	Pass

Repair bikes	Operator can select a defective bike and click repair button to change the status of the bike. The status of bike will be changed from defective to normal.	Pass
Move bikes	Operator can select a normal bike, a location and click move button to change the location of the bike.	Pass
Add bikes	Operator can add new bikes to the bike resources. They need to enter the number of new bikes, the corresponding location and click the add button.	Pass
Show all activities of all bikes	Enter the beginning time and ending time separately and click report button	Pass
Data visualization	Click the chart button and you will see all the bikes which are in the riding status	Pass

Chapter 6 Conclusion and future work

6.1 Conclusion

The goal of this project is to establish a software system that can solve the problem of bicycle rental for consumers, and help operators and managers manage these bicycles. Based on realizing these basic goals, the function is supplemented and optimized to provide better services. First, we start from the background of bike sharing, analyze the feasibility of the project implementation based on the development and advantages and disadvantages of bike sharing, and investigate the functions of other bike sharing software which is developing rapidly in the market. Through comprehensive analysis, we conclude that the development of bike sharing software is feasible and in demand. Then we list and analyze the functions of the software, and decide to realize the basic requirements first, and add new content on these infrastructures. In the design and implementation phase of the system, we encountered some problems, such as how to connect the various parts of the function, whether it is necessary to introduce GPS positioning and so on. After solving these problems one by one, the rudiment of a software system came into being. In order to enhance the usability of the system, we have tested the functional requirements of the system, and modified the functions that did not pass the test.

In conclusion, this software has been proved to be a bicycle sharing system that can meet the needs of consumers. However, due to the lack of time and knowledge architecture, our system still has some problems in some aspects. We will make further analysis on the future improvement direction in the next part.

6.2 Future work

Although the core functions of the shared bicycle system have been realized, as a bicycle management system that can be operated online, it is not enough to only realize the basic functions. In this part, we will focus on the discussion of future work.

- Return function: punish the consumers who park their bicycles outside the designated area

Although the system can send operators to move bicycles between different areas, if no punishment mechanism is introduced, too many bicycles will be parked outside the specified area, which will cause a waste of resources.

- Operators : operators can add/delete locations

With the expansion of the operation scale, it is obviously not enough to only have five parking spots, so the function of adding / deleting parking spots will be added in the operation interface of operators

- Manager : add/delete operators

- payment : add membership

We can increase the membership mechanism to stimulate consumers to buy membership cards to get car rental discounts

- report of customers : add feedback

It is not enough for consumers to submit information about damaged bicycles. In order to provide better service, we need to hear more from consumers, so we should add feedback function to listen to consumers' valuable opinions.

Chapter 7 Contributions

Yifan Jin :

- Design prototype for Report and Payment
- Write the code of the report function

Zhiyuan Jing:

- Prototype in Group 2 analysis and design
- Write the code of operator.py

Guannan Lu:

- Requirement analysis and user story of operator and manager
- Prototype of operator
- GUI design and bar chart of manager.py

Tengyu Zhang:

- SQL of manager.py
- E-R graph of operator and manager
- Prototype of manager

Cunxu He:

- Functional requirement of customer
- Write the submitted report

Fuyu Jia:

- Designed the prototype of login, logout, rent and return function in 'user' part
- Payment function and bank account function of user.py
- Insert related information in ride table in the database

Dina Khalid A Alsharif:

- Design the Data base
- Implement log in function, registration function, rent bike function, Return bike function

References

- [1] Fishman, E., 2015. Bikeshare: A Review of Recent Literature. *Transp. Rev.* 36, 92–113.
- [2] DeMaio, P. and Gifford, J. (2004). Will Smart Bikes Succeed as Public Transportation in the United States? *Journal of Public Transportation*, 7(2), 1-14.
- [3] DeMaio, P., 2009. Bike-sharing: History, impacts, models of provision, and future. *J. Public Transp.* 12, 41–56.
- [4] Ma trai, T., To th, J., 2016. Comparative Assessment of Public Bike Sharing Systems. *Transp. Res. Procedia* 14, 2344–2351.
- [5] Midgley, P., 2011. Bicycle-Sharing Schemes: Enhancing Sustainable Mobility in Urban Areas. *Comm. Sustain. Dev.* Nine teent, 24.
- [6] Vogel, P., Ehmke, J.F., Mattfeld, D.C., Systems, I., Berlin, F.U., 2014. Service Network Design of Bike Sharing Systems 1–33.

Appendix A <database of system>

Tables (6)

Name	Type	Schema
bike		CREATE TABLE "bike" ("bikeid" INTEGER, "bike_status" TEXT, "location_id" INTEGER, "userid" INTEGER, PRIMARY KEY("bikeid" AUTOINCREMENT), FOREIGN KEY("userid") REFERENCES "user"("userid") ON UPDATE CASCADE, FOREIGN KEY("location_id") REFERENCES "location"("location_id") ON UPDATE CASCADE)
bikeid	INTEGER	"bikeid" INTEGER
bike_status	TEXT	"bike_status" TEXT
location_id	INTEGER	"location_id" INTEGER
userid	INTEGER	"userid" INTEGER
location		CREATE TABLE "location" ("location_id" INTEGER, "location_satus" TEXT, location_number INTEGER, PRIMARY KEY("location_id" AUTOINCREMENT))
location_id	INTEGER	"location_id" INTEGER
location_satus	TEXT	"location_satus" TEXT
location_number	INTEGER	"location_number" INTEGER
operations		CREATE TABLE "operations" ("operationid" INTEGER, "bikeid" INTEGER, "operation_type" TEXT, "start_location" INTEGER, "end_location" INTEGER, "operate_time" datetime, PRIMARY KEY("operationid" AUTOINCREMENT), FOREIGN KEY("bikeid") REFERENCES "bike"("bikeid") ON UPDATE CASCADE)
operationid	INTEGER	"operationid" INTEGER
bikeid	INTEGER	"bikeid" INTEGER
operation_type	TEXT	"operation_type" TEXT
start_location	INTEGER	"start_location" INTEGER
end_location	INTEGER	"end_location" INTEGER
operate_time	datetime	"operate_time" datetime
ride		CREATE TABLE ride(rideid INTEGER, bikeid INTEGER, userid INTEGER, journey_length REAL, start_time datetime, end_time datetime, start_location INTEGER, end_location INTEGER, receipt REAL, PRIMARY KEY("rideid" AUTOINCREMENT), FOREIGN KEY("bikeid") REFERENCES "bike"("bikeid") ON UPDATE CASCADE, FOREIGN KEY("userid") REFERENCES "user"("userid") ON UPDATE CASCADE)
rideid	INTEGER	"rideid" INTEGER
bikeid	INTEGER	"bikeid" INTEGER
userid	INTEGER	"userid" INTEGER
journey_length	REAL	"journey_length" REAL
start_time	datetime	"start_time" datetime
end_time	datetime	"end_time" datetime
start_location	INTEGER	"start_location" INTEGER
end_location	INTEGER	"end_location" INTEGER
receipt	REAL	"receipt" REAL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
user		CREATE TABLE "user" ("userid" INTEGER, "user_name" TEXT NOT NULL, "password" TEXT NOT NULL, "bikeid" INTEGER, "account_balance" REAL, PRIMARY KEY("userid" AUTOINCREMENT), FOREIGN KEY("bikeid") REFERENCES "bike"("bikeid") ON UPDATE CASCADE)
userid	INTEGER	"userid" INTEGER
user_name	TEXT	"user_name" TEXT NOT NULL
password	TEXT	"password" TEXT NOT NULL
bikeid	INTEGER	"bikeid" INTEGER
account_balance	REAL	"account_balance" REAL