Report on Database Project of SUSTech Railway 12305 System

Chi Xu 11912224 , Yunhao Luo 11813221, Haowei Kuang 11910710

December 2020

1 Introduction

Here we will show a fantastic system of railway, just like 12306, we call it SUSTech Railway 12305 System. In the following sections, we will introduce our system in these aspects: data crawling, table establishment, and our system functions with many extra points.

2 Design

Here is our design steps. First, we need to get authentic data of the trains, stations, time table, interval distance, order of stations, their relationships, etc. Second, we need to find their close and simple connection, then build several tables to connect them. Meanwhile, we must follow the normalization very strictly, since all the project is based on these diagram, and we almost spent one week to discuss the design of table. Third, distribute the workload and then everyone does his own part. However, since the distribution cannot be total fair and the competence is also different, almost everyone dose one third.

3 Process and bright-spot

3.1 Data

At the very beginning of the project, in the process of obtaining the information of real train numbers, we considered to crawl the network data by learning Python. After studying BeautifulSoup on our own, we started to try to get data by crawling the 12306 ticket platform. However, the complex login verification code of 12306 makes our simulated login operation unable to complete, resulting in the failure of data crawling.Without access to the most authoritative websites, we had to find a new way to get the information from other websites. After a search, we finally accepted the reference List of train numbers using the train ticket network (http://search.huochepiao.com), through Python to perform request URL, get response, call BS4 module to crawl the data. In the process of crawling, we adopted the crawling method according to the train number, starting from the train number in the reference table, and obtained the content through label comparison, and crawled the detailed data layer by layer. Through several methods, the operation of request URL, obtaining station name, obtaining train ticket information and so on was respectively performed, and the crawled data form was sorted and summarized according to the requirements of database construction, so as to obtain the final CSV master table, which was applied in the process of the final import into the database.

3.2 Table and relationship

3.3 Functions

When it comes to functions, I will list them one by one: the query of ticket, query of train, query of station, administrator station management, administrator train management, user management, privilege management, exception handling, hint of operations, and etc. Our project is divided into two general page: user and administrator. Here is the entrance of the system.
Then if we want to login the user page, we need provide a user name and password, just like this:
This web is somewhat useless, since in the requirement user is a general concept, however, this is not the point. Next I will show the system main function.

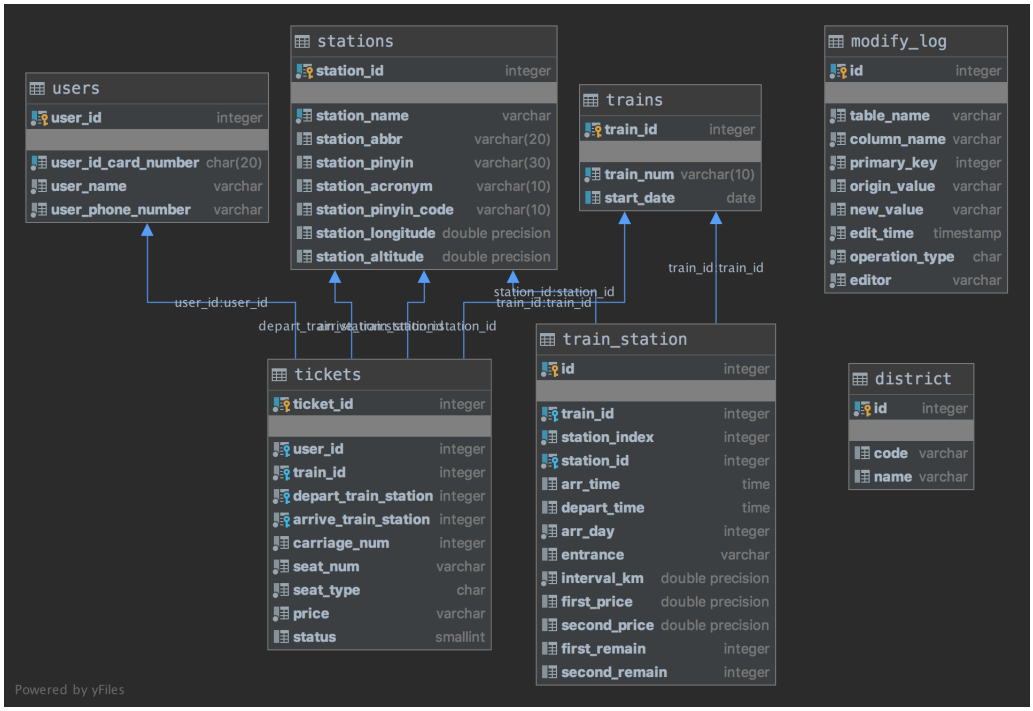

Figure 1: TABLE

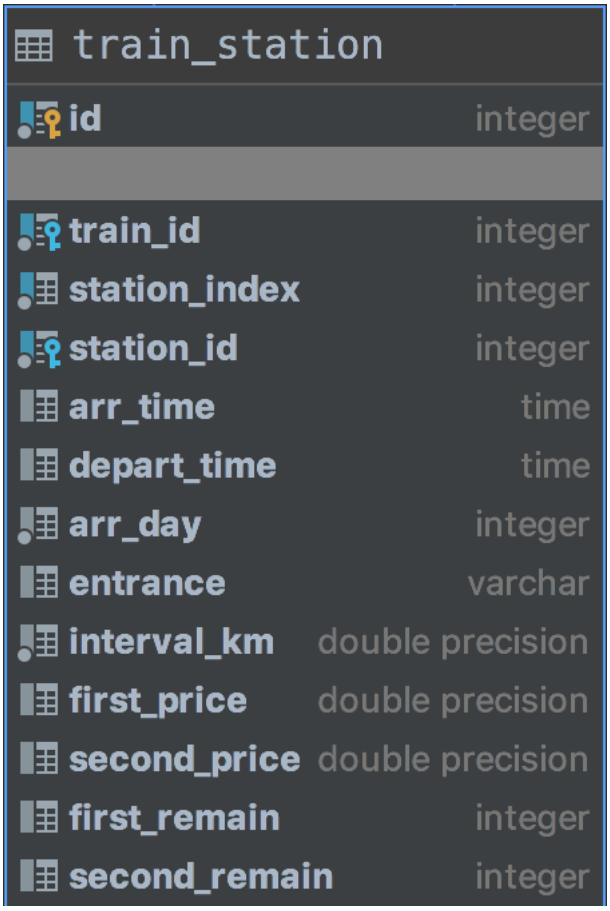

Figure 2: TABLE2

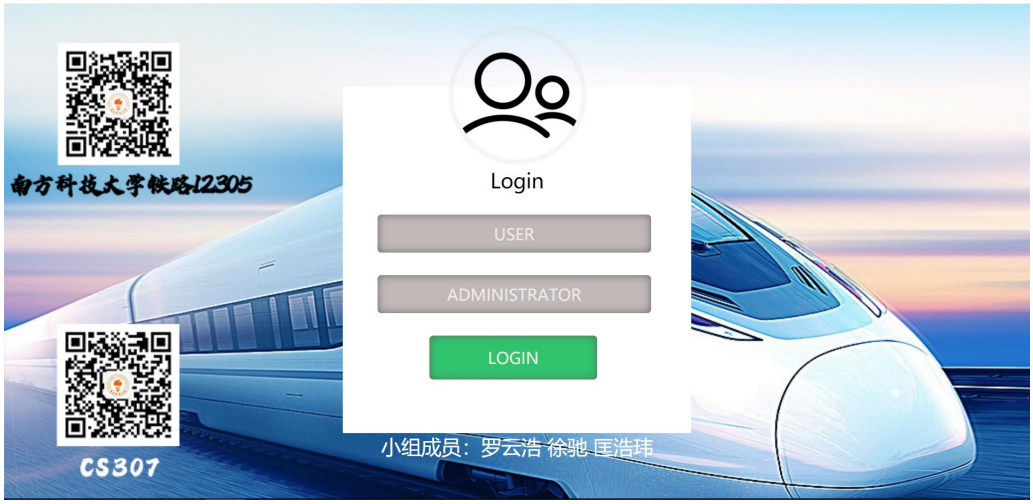

Figure 3: login

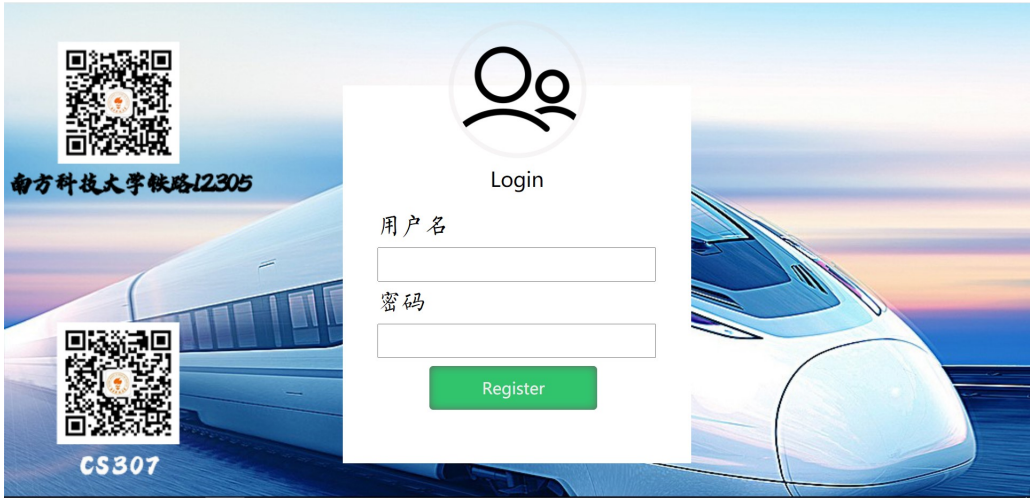

Figure 4: register

3.3.1 Query of ticket

When we enter the departing station, the arriving station and the date, the result will like the page below. NB: to make reader understand our system more easily, we choose BeijingXi to GuangzhouNan in 2020-12-30 as our example. Meanwhile, if we want know more information about a certain train, the button beside it could help us to get the hyper-reference.

Figure 5: userquery

3.3.2 Query of train and station

Now that we got the the trainNum, we want to get all station this train will pass. So let us set G69 as an example. We enter G69 and 2020-12-30 in the text bar, and the result is as follows.

Figure 6: usertrainquery

Here we can see, all stations has been presented, including the interval time and ticket remain. If you want to but one ticket, just try to click the navigation bar option.

3.3.3 Buy ticket

Then it comes to the buying page, which is a little complex compared to the previous webs. However, its structure is really clear, which will help everyone buy a ticket without much effort. Here is the picture. In this

section, I will not show the process in the report, since that behavior may effect the structure and outline of this page.

Figure 7: userbuy

3.3.4 Query of order

When an user buy a ticket, he or she can query it by his or her own idcardnumber, which then can help he or she refund it. Let show it below.

Figure 8: refund

3.3.5 Addition of station

New station can be added. For the sake of adding train, this is a must.

3.3.6 Subtraction of station

Station which is redundant can be deleted.

3.3.7 Addition of train

New train can be added, with a lot of new information added.

3.3.8 Subtraction of train

Any train can be deleted.

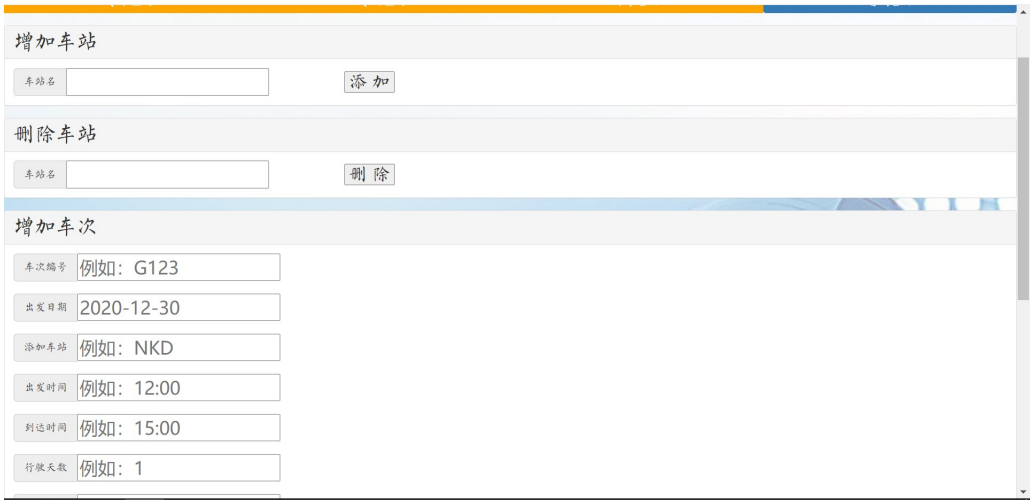

Figure 9: Admin

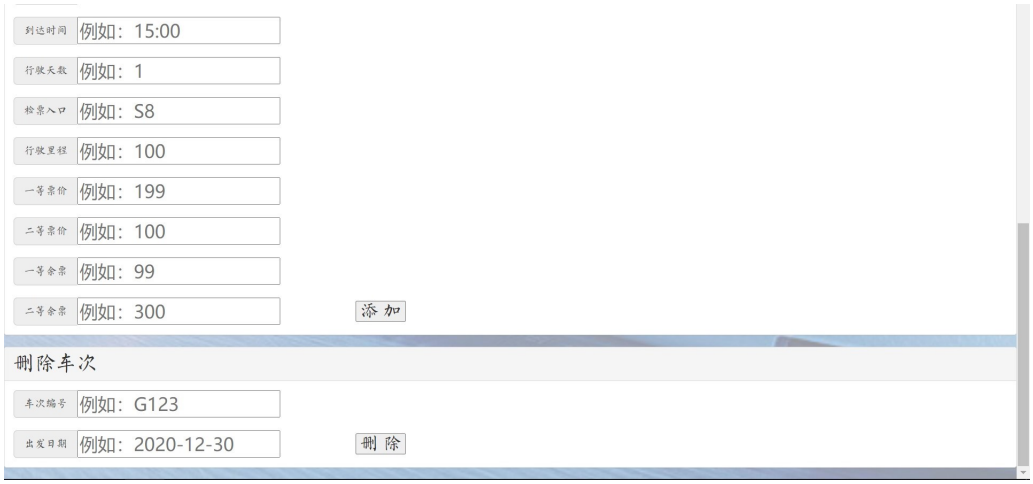

Figure 10: Admin2

3.3.9 High concurrency manipulation

We test 10,1000,10000,100000 users buying a ticket, here is the result. We submit 10,000 request to buy a ticket to our database. The output below shows that 1,000 tickets are sold correctly and we finish this test in 27.3 seconds, which is relatively fast, since our database is stored in a portable laptop rather than some high-performance servers.

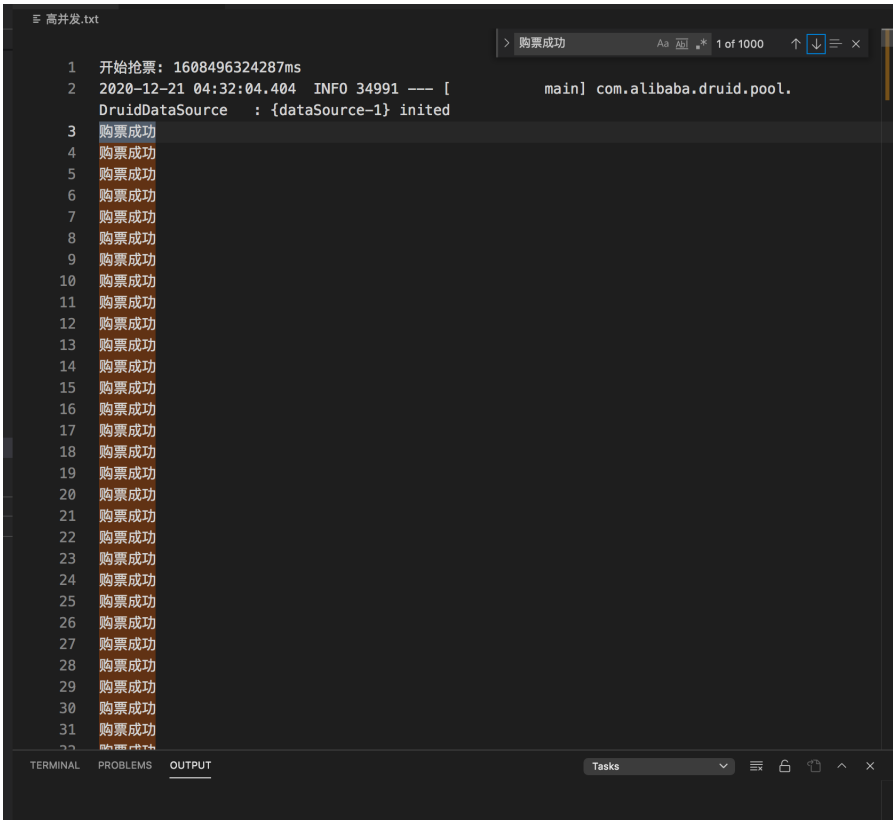

Figure 11: high

amount	time/s
10	0.592
1000	3.369
10000	27.327
100000	9997.971

Figure 12: result

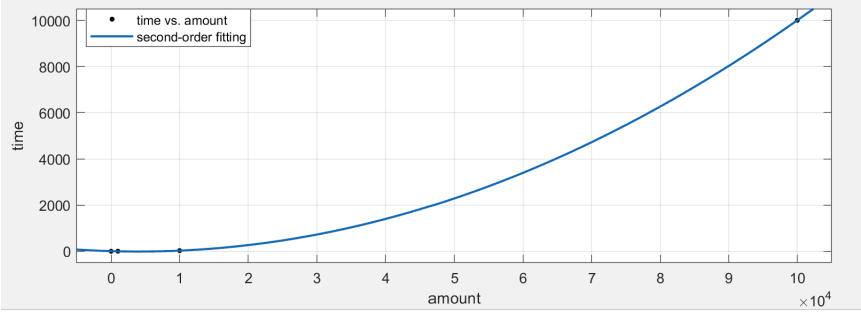

Figure 13: result1

4 Conclusion

In this project, we do and practice a lot. We learn a basic model for running a website. The whole procedure, from the client side, backend server side all the way to the lowest level database operation side is implemented. Although The SUSTech Railway 12305 System is to some degree great, though with some shortcoming. We will go!