Final Project Report
Medical Management System
CSC 3002

May 19, 2019

Partners:	Ruixing LIANG	116010126
	Ziheng XIAO	116010244
	Pizeng ZHOU	116010325
	Mingen LI	116010110
	Jiaming LIU	116010143
	Xinyu WANG	116010226
Instructor:	Professor Huang	

1 Introduction

Since 1995, though falling behind the growth rate of government revenue, the annually rate of increase in the budget for national health fund still reached to about 14.2%. However, this seemingly ostentatious and glorious figure cannot conceal a fundamental foible —the scarce medical resources per capita. According to the data released by NHFPC , every 1000 people have two doctors which doubles in many developed countries and every hospitalized patient has 2 medical staffs which is more than 40 in other OECD countries. It presumably causes many grim social problems like deadly conflicts between doctors and patients. And what makes it even worse is that the unquenchable marketization in the medical industry has been disequilibrating the system, specifically speaking, making it more like a steep Lorenz curve, which in turn alienated most people from high quality medical resources.

Apart from the continuous efforts to rationalize the system that can be made on macro level by the government, there are growing interests from both intellectuals and entrepreneurs on AI-integrated smart precise medical platform. With more than 18 billion RMB’ s capital invested in this field, many creative algorithms have been developed in order to achieve the final platform.

2 Project Objective

This project has 2 main objectives:

- As we all know, a well-trained and ideal AI model is not only a product of algorithm, but also a crystal of big data. Especially for temporary AI —data driven science, more weight can be placed on unstained ideal data collected by various hospitals. However, nowadays there is no universal structured system to get data in identical form which can be troublesome to process this huge amount of data before actually using them, let alone a substantial number of them are not labeled right because of inadvertent miss or they are transferred from one hospital to another which use completely different databases to store the information. Therefore, this medical management system is designed to collect data in a structured form, facilitate practical medical usage and improve AI study.
- Presenting registration process in hospital is actually too traditional even though added with fancy WeChat online registration function. Many patients are afflicted by a huge waste of time and money induced by those wrong decisions made by departments or doctors in the hospital. Patients desperately need a platform to make decisions smartly based on their description of symptoms. And a universal system which can track their medical history as well as their health status also make things easier for doctors. In this context, this medical management system is implemented to meet this requirement.

3 Usage

In this section, the flow of usage of this system will be shown by the following steps.

3.1 The Login Interface

Users can login in through the login dialog. In this dialog, the original title is hidden and a more beautiful title is designed to replace it. The dialog can identify whether the user is a doctor or a patient by judging the user' s ID, which leads to different program dialog. In the login dialog, the skin can also be changed if the user wants something different.

Figure 1. Login Interface

3.2 The Register Interface for Patients/Doctors

For any new user, he can register his account by clicking the "病人注册" button or "医生注册" button according to his identity. All users' IDs are assigned by the server.

For patients, the information they have to input is their name, age, gender and so on. They can choose their birthdates by using a calendar, and the initial value of the birthdate box is the current date.

Figure 2. Register Interface for Patients

As for doctors, they need to register their name, gender and so on. They can choose medical department in the combo box.

Figure 3. Register Interface for Doctors

3.3 The Main Interface for Patients/Doctors

For the main interface, patients and doctors have different interfaces. patients can reserve doctors by directly clicking the doctor's ID, and choose the time in the calendar. After the reservation, the patient can send his request to doctor.

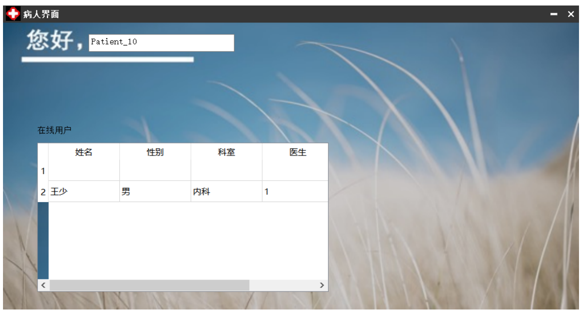

Figure 4. Main Interface for Patients

For the doctors, he can chat with other doctors in the first block, and check the patient's information in the second block.

Figure 5. Main Interface for Doctors

Further: the users can only login in with ID instead of telephone number, and the doctor cannot check when he is reserved.

4 Implementation

This section illustrates the implementation process of this system. The Mind Map below describes the overview of the whole project.

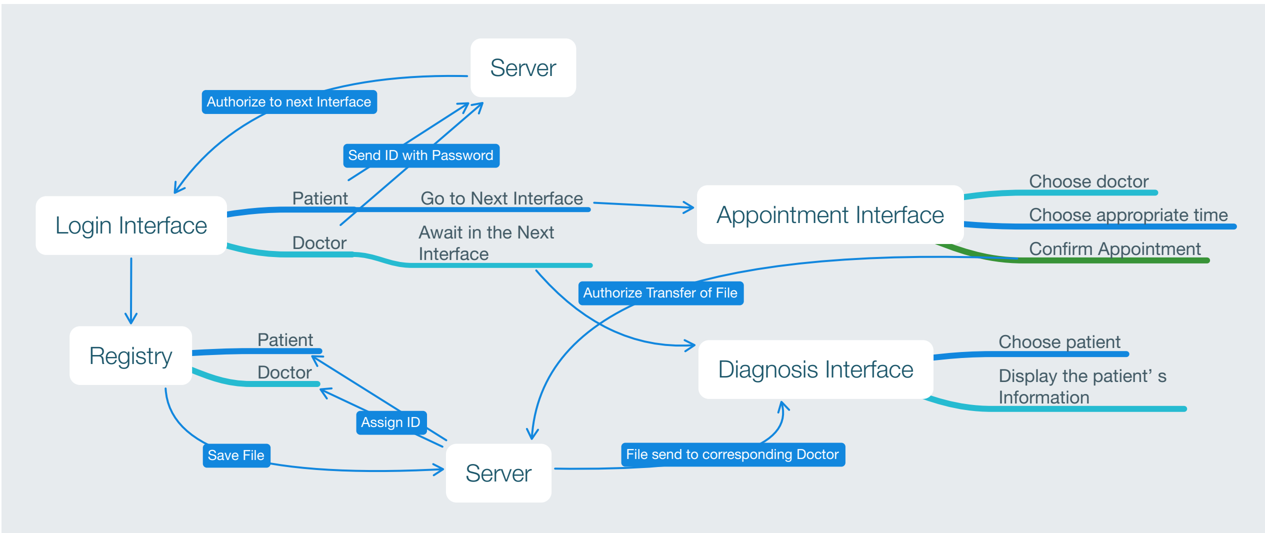

Figure 6. Mind mapping for the system

4.1 Inner Communication: signal and slot

The communication and interaction among server, doctor client and patient client is important. They are the essential function to support the whole system.

In the project, for these three components, there are three different classes are designed as a subclass of QWidget, which is a Qt built-in class. For subclass server, the method in it is to deal with signals from doctor client and patient client, help to transfer data and files between them. For subclass doctor and patient, the method in them is to initiate a connection request to the server and transfer the information files of doctors and patients to the server. Signals in the classes are declared in the Signals field, just like Private, Public fields. To detect, or receive information from the Signals, we need to declare Slots in the classes. And moreover, in order to match different Signals and Slots with each other, the matched signals and slots are connected together by using the connect() expression. The connect expressions are declared in these three classes, the signals are shared among them through transfer()/receive() function.

Here gives an example of realizing the interconnection among doctor client, patient client and server. When a patient client initials a connection require to the server, a signal (a key word) is sent to the server through UDP, after server receive this signal, it will in listening state and returns another signal to the patient client to make patient client to transfer information files of patient and the doctor the patient want to choose to the server. For server, it will match the corresponding doctor and patient after receive the data, and then transfer these data to the doctor client. Then a process is finished. The situation for doctor client is similar except that doctor cannot choose patient they want.

4.2 External Communication: TCP/UDP

With help of Qt built-in classes QUdpSocket, QTcpServer, QTcpSocket, we build the class Client to transfer files between doctor/patient elients and server. With UDP, clients and server communicate with each other by some key words to make sure that they are all ready to transfer or receive files. After that, many defined methods can be used to set up a TCP connection. The transfer()/receive() function in clients and server are almost the same except that the different ports are used in each part. Also, UDP and TCP also use different ports to make sure there is no conflict between protocols.

4.3 Data Structure

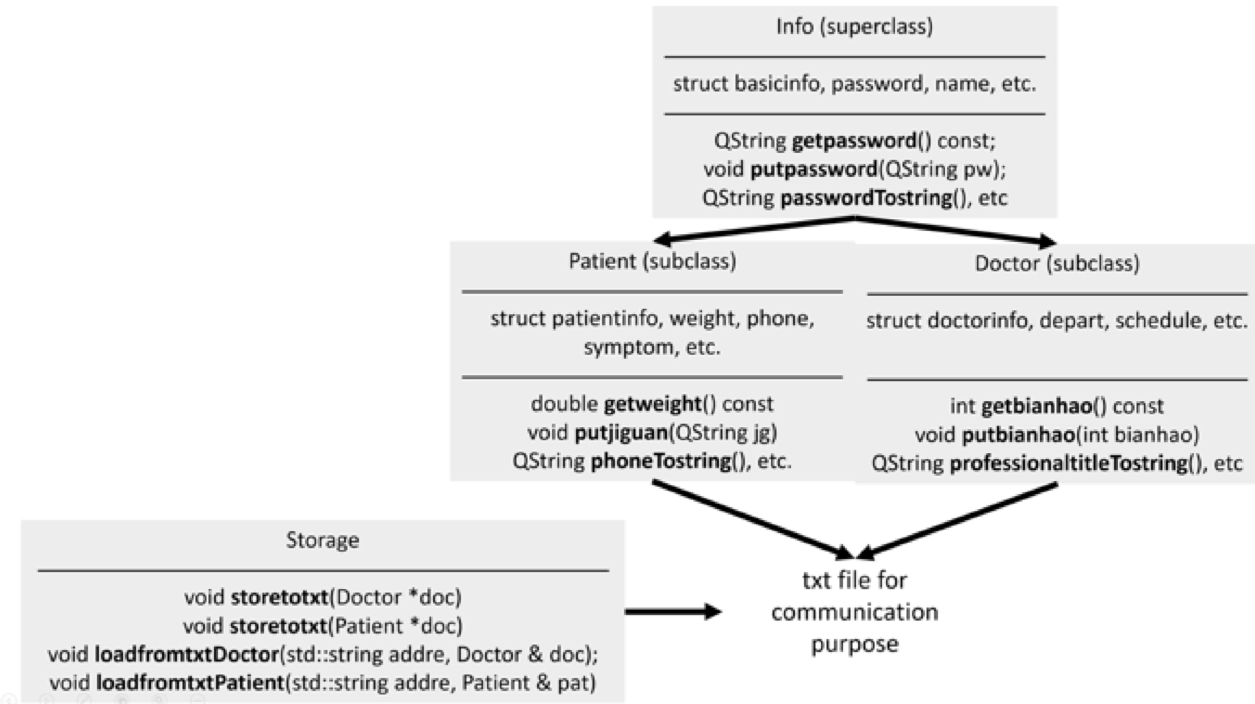

Figure 7. Data Structure Inherence

From figure 7, the implementation of data structure Inherence was shown. Info class served as a super class for Patient and Doctor. If output was needed, the Storage class was used. Inheritance structure was implemented in this data structure design. Since there were many information in common for Doctor and Patient class such as name, login account information like ID and password. These parts of data were set to be stored in the Info class. Other variables were stored in struct in Doctor and Patient classes. Info class was an abstract class with virtual functions dealing with login account which was ID. Thus, it should not be constructed as an object. Due to different output assignment rules for ID for patients and doctors, these functions in Info were overwritten by subclasses. Three main types of functions were included to modified preserved data in object. GET functions like QString getpassword() const. These functions were used to retrieved data in objects while keeping the variables unchanged. PUT functions like void putbirthday(QString bir) were used to modify the variables within the objects. These functions should be the only legal way for users to modify the values.

Also, TOSTRING functions like QString phoneTostring()were implemented in classes for Local Area Network communication. In communication, the objects were stored to a txt-type file and sent to clients or server. Figure 8 was a demonstration of an output file of a patient object and a doctor object. The txt file was named with the assigned binglihao (病历号) and was rewritten in an ordered sequence. In doctor txt file, the schedule was originally a map from QString to integer which represented the patients' ID. The key was the abbreviation of date and time. Using internal iterator, all pairs were output to the file.

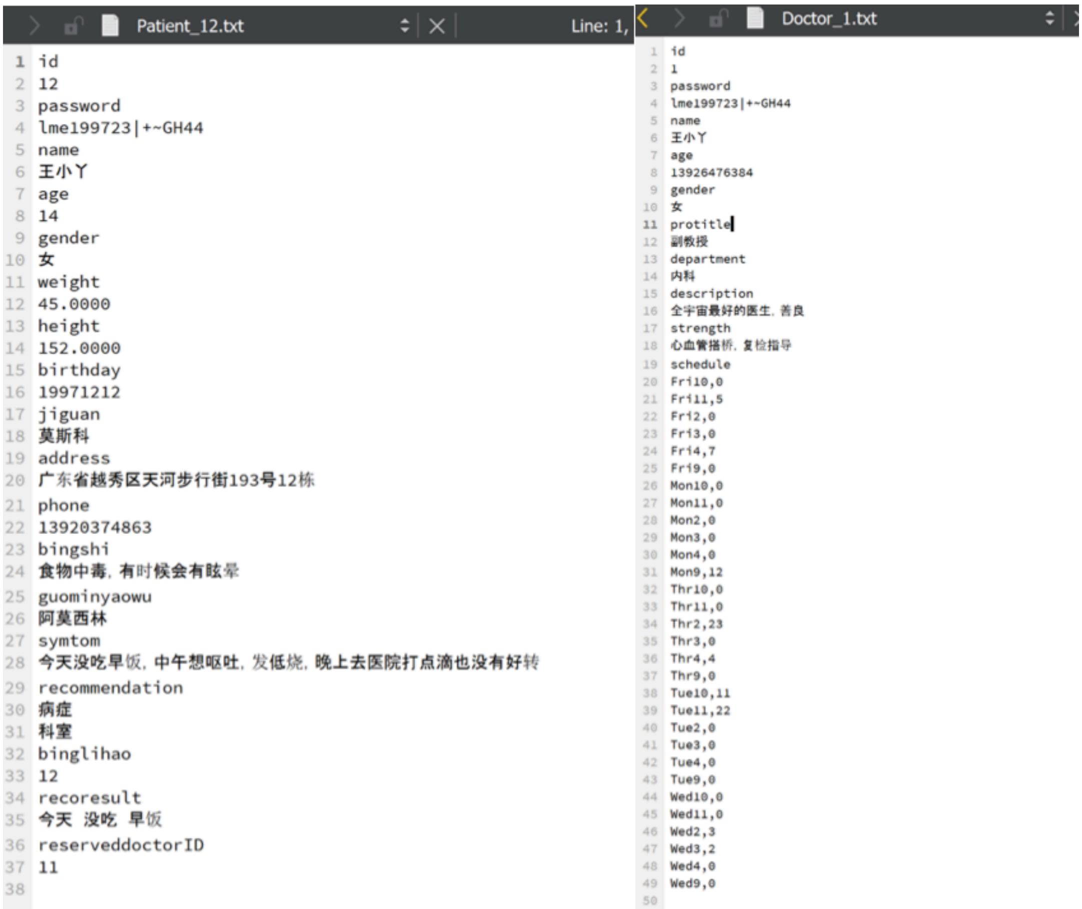

Figure 8

4.4 Algorithm for Symptom Analysis

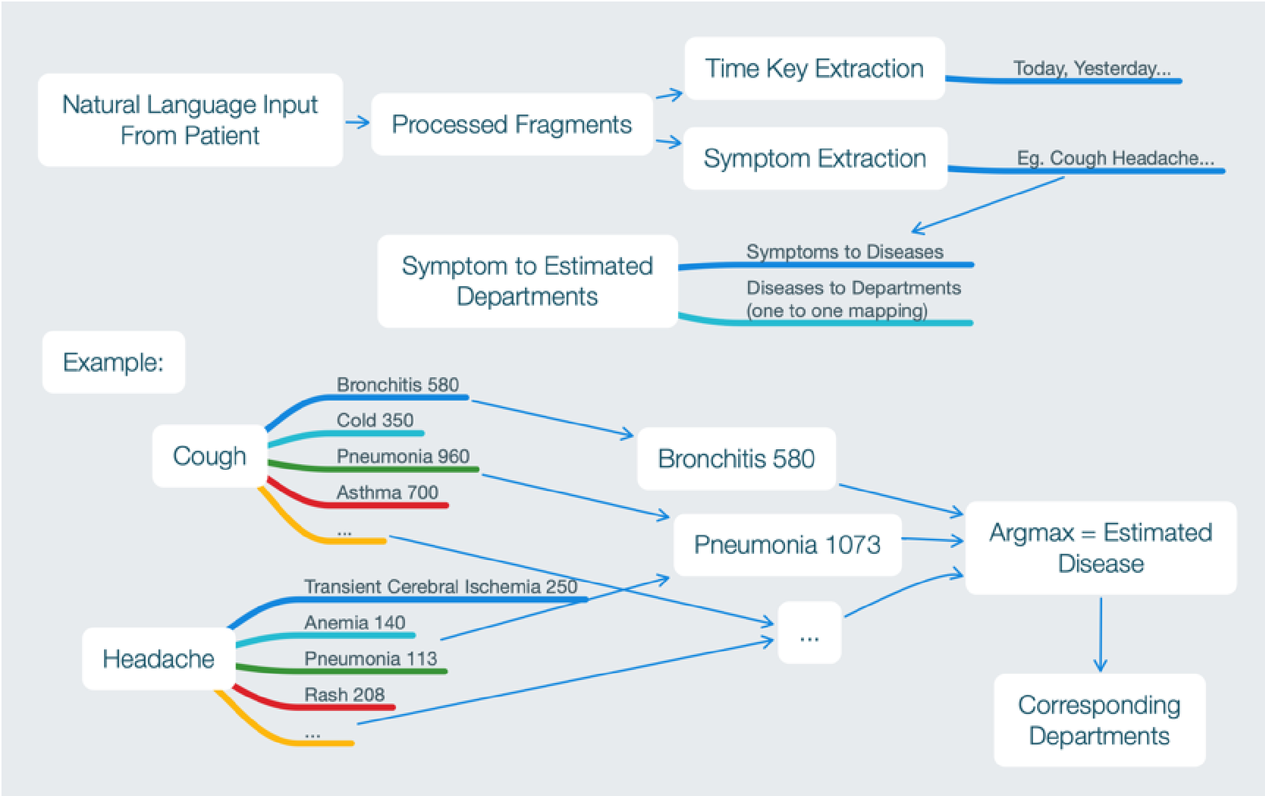

Figure 9. Mind mapping for symptom analysis

Natural language processing (NLP) is the fundamental for effective communication using natural language (particularly in our project, Chinese) between human and machine, which has been generally considered as the most crucial and difficult part of artificial intelligence. Even after studying all the basic conception in the NLP algorithm, it is still difficult to build the NLP engine out from nothing, so it is no shame to admit we mimicked the design of cppjieba project on GitHub and reconstruct its structure, which is prodigal of its gifts. Generally, we utilize two fundamental components HMM (Hidden Markov Model) model and MP (Maximum Possibilities) to generate Mixed Cut which coherently utilizing the corpus we collected from Sougou and Tsinghua medical terms integrated with the common words to carry out the cutting and tagging.

- HMM: This model along with the shortest path algorithm which briefly mentioned in class actually make the base for this model. We used Viterbi, four different states are defined: "B" , "M" , "E" , "S" , traverse the set to find the best states for each word.
- MP: This is just an improved adaptation from maximum matching algorithm configuring DAG graph to recall the best matching.
- Mixed: Do MP then HMM.

难 以 置 信	$1.37*10^{-14}$
难以 置 信	$3.65*10^{-14}$
难 以 置信	$7.41*10^{-13}$
难以 置信	$1.01*10^{-10}$
难以置信	$2.81*10^{-6}$

Table 1

Through the ordeal of dealing with the NLP, we then begin to construct our model to map captured symptoms which is formerly extracted from the patients' natural descriptions on their recent lives to the corresponding departments. Due to the model' s required data is private both in China and aboard, it is somehow difficult to retrieve data to train model, therefore we utilized a set of organized data on GitHub to train the model, we assign different weights on each symptoms' contributions to specific types of diseases. Due to the limitation of the original set, we can only have up to 151 diseases with 713 indicators (symptoms). Then we performed preprocessing the data, and carried classic machine learning to build up the model to get a csv table containing every links with its ends and weights. So sometimes odd diseases may come out from a series of rather common symptoms due to the limited training set which document cases that induced overly high weights assigned on these links.

As for the estimation process, just like it is vividly illustrated on the charts, basically we perform a similar Maximum Likelihood Criterion which sum the weights directed to certain disease and sort the sequence, utilizing their total contributions as the indicator of the possibilities. Explicitly, we take the largest one as the estimated disease output (see the figure below). Then easily with the sophisticated hospital system online, we can get a table to map each disease with the corresponding departments.

```
Possible diseases: 脱水  Weight: 297
Possible diseases: 食管裂孔疝疝气  Weight: 61
Possible diseases: 骨质疏松  Weight: 169
Possible diseases: 慢性酒精中毒  Weight: 70
Possible diseases: 哮喘  Weight: 2505
Possible diseases: 肠梗阻  Weight: 112
Possible diseases: 肿瘤 Weight: 297
Possible diseases: 病态肥胖  Weight: 76
Possible diseases: 前列腺恶性肿瘤  Weight: 163
Possible diseases: 肺动脉栓塞  Weight: 294
```

Figure 10. The disease with maximal weight is: asthma

```
The disease is: 哮喘
The department is: 呼吸科
```

Figure 11. Analyzed disease and department

Besides, we also designed a function that lists every time word and all behaviors belong to it. This is called a medical digest for doctors' reference in analyzing the illness. In the following example, the first one is "zz", which represents significant words that are helpful in disease analysis. If there is no time word, it simply adds an “unknown” tag before actions, as the second line shows. In addition, since this digest is stored in a map structure, these keys are automatically sorted and we didn’t make them in a meaningful order here. This indicates that doctors still need to take a full consideration of the whole symptom description instead of only viewing a few segments. In conclusion, when a patient decides to choose a certain doctor, this digest will be sent to the doctor, and he/she has enough time to prepare for the consultation.

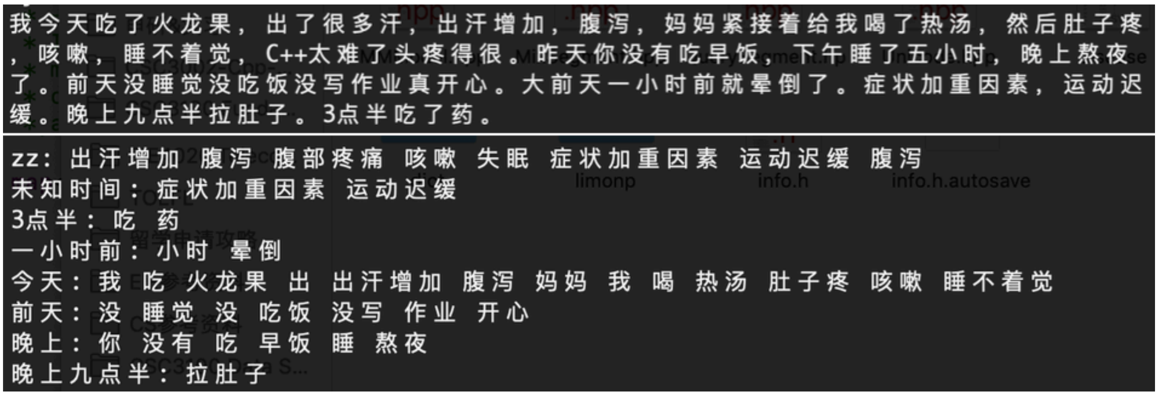

Figure 12. Original symptom description and medical digest for doctors

5 Key Problems and Solutions

In addition to the medical management system we have achieved, there still exists several drawbacks we were not able to tackle before the deadline.

- Data Structure: Although this data structure protected data in class from illegal access from users and forced users to modified data in preset way and the storage files were intelligible and understandable. However, this file could be considered to be lack of data security. The txt file could be accessed by every user and could be easily modified. For instance, password of an account should not be provided to users and it could be dangerous if other modified the password. We may use some kinds of encryption techniques to ensure the safety of data.

- GUI: Despite the effort of the GUI designer to add pictures and make decorations to our interface, our GUI design can still be improved. For example, especially the buttons.

- Algorithm: Due to the limited data set, the match procedure is deficient so that the symptoms and the corresponding estimated diseases are not enough and the weights are somehow not reasonable. So, we can expand the original data set and improve the algorithm to make it better.

- Network: We cannot register the same account type simultaneously. Besides, the server is likely to time out. We can define multiple ports to enable multiple users to access the server at the same time, which makes the server more robust.