Gene Information Management System

Project Documentation

Group 9

Chang Li

Yuting Wang

Shixuan Zhang

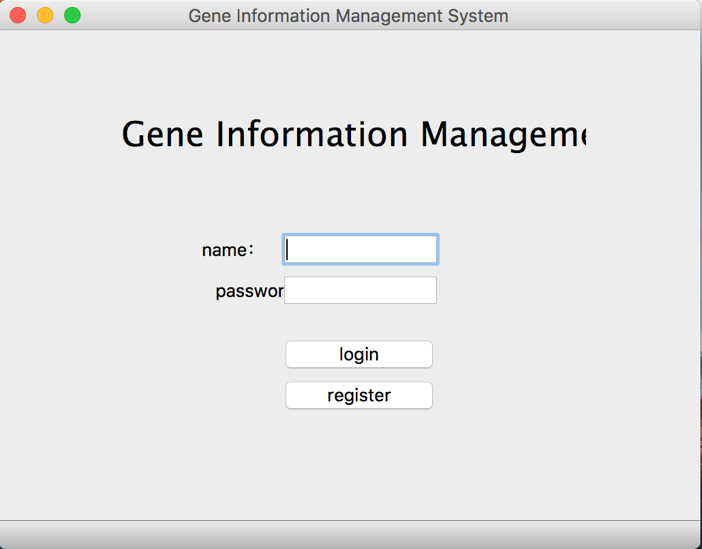
This project documentation mainly describes the purpose of the project and the functions it implements. By analyzing the code segment of the project, explain the main Python functions that implement the program.

1. **Overview**

The purpose of this project is convenient for biology students to query, extract, and modify the genetic sequences needed in biological learning and research. The functions of this system are mainly divided into 3 modules: gene sequence management module, gene sequence query module, user management module. Apart from this, this system also includes register and login module for users and administrators.

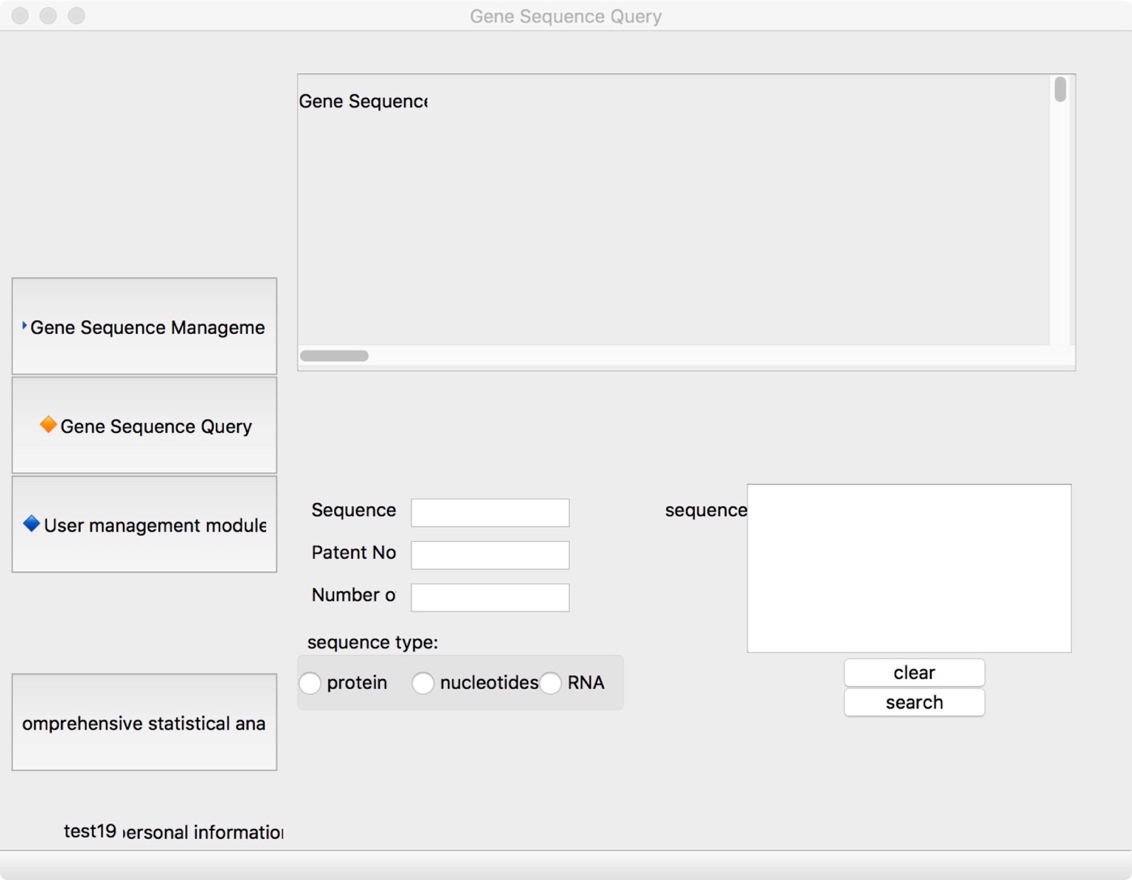
* 1. **Register and login module**

When users and administrators open this Gene Information management software, the register and login page will appear first. They need to enter their own username and password to log in to the main interface. If it is a new user, it needs to choose the registration function. Once the new user has completed registration, they can log in to the system interface.



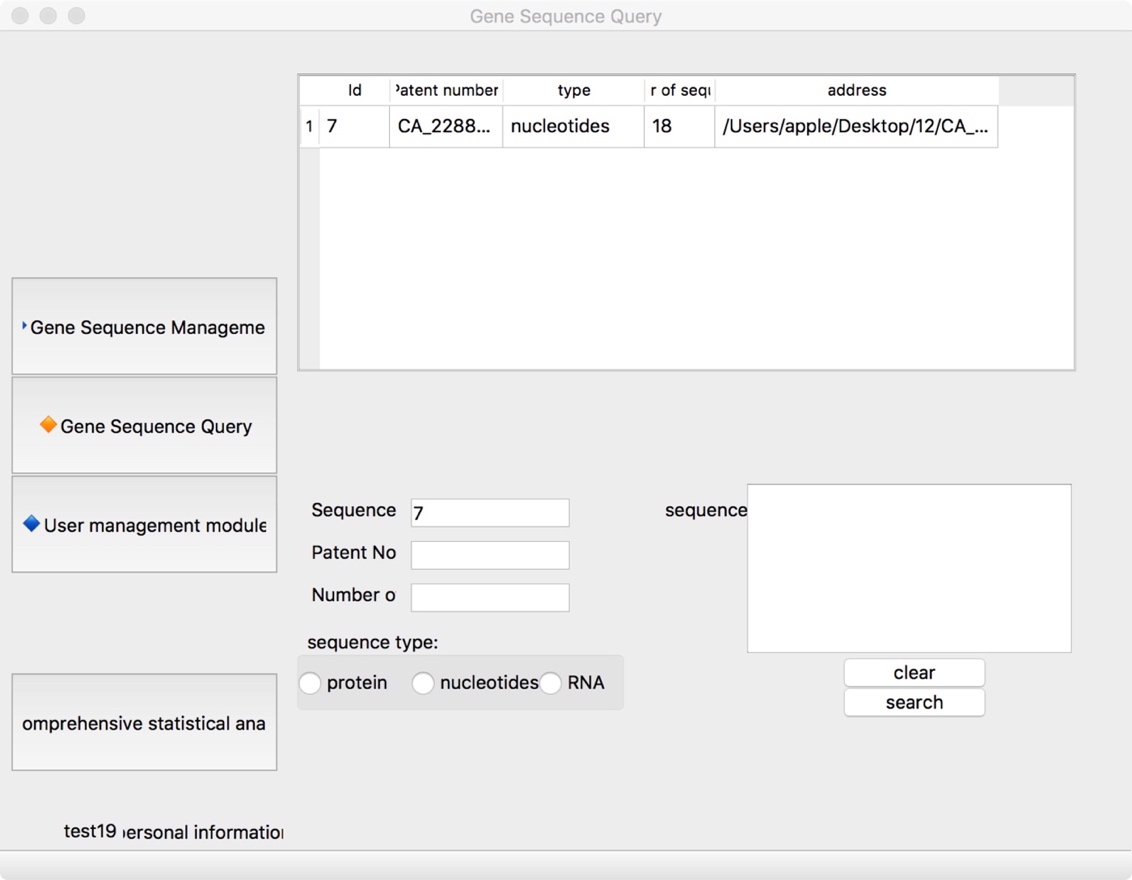
* 1. **Gene sequence management module**

This module is only available to administrators. In this module, the gene sequence information in the database will be displayed in the administrators’ interface. Administrators can add, modify, and delete gene sequence information. When adding genetic sequence information, administrators can add them one by one or batch import.



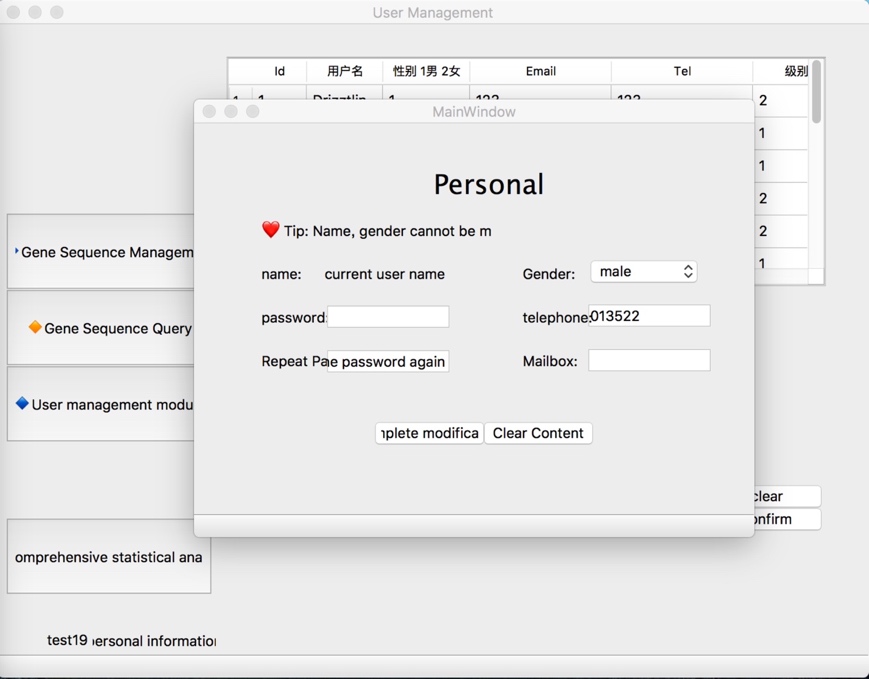
* 1. **Gene sequence query module**

This module in available to users and administrators. In this module, users and administrators can query the gene sequence information they want. They can query in precise or fuzzy terms. For example, if they enter a character, the system will display all sequences containing that character.



* 1. **User management module**

In this module, the administrators have the right to view all user’s information and modify and delete this information. Ordinary users can modify their personal information.



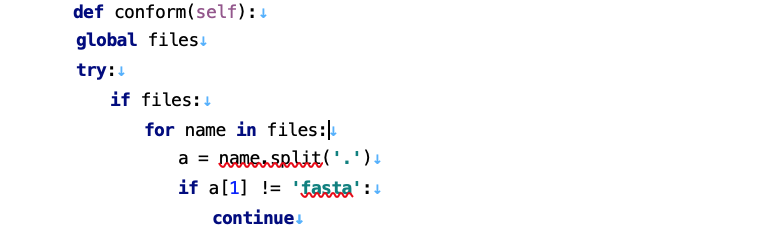
* 1. **Requirements completed and not completed**

The work in the project plan is too much to be completed in a month. We cut down from the first 7 modules to 4 modules to simplify the functions of the system: removed the implement functional modules and legal information for gene-patented products. The completed system can implement the user and administrator's login registration function, query, modify and delete gene information, and modify and delete user information.

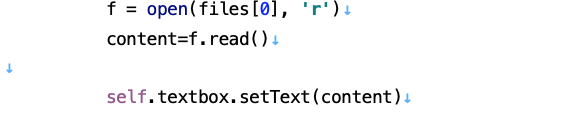
1. **Main Functions**
   1. **Bulk import**

One function in the gene sequence management interface is that administrators can import information into the database storage in bulk. Next, we will describe how to implement this function.

First, we use the function **read** to reads the ending with fasta (data file) and their contents under the current address and returns **def read (file)**. In the process, we used a function that can split the string from the middle, **str.split ()**. This function divides the string and places each segment as an element in an array. For example, the result returned by 'abcdas'.split('a') is ['', 'bcd', 's']. With this function, we can separate the file name from the suffix name so that we can classify the file. Then, we call the **read** function and use **for** loop structure stores file contents to database.



We also implemented the preview function of the file.

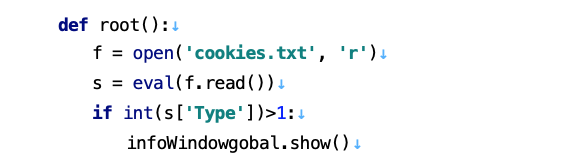


This is the implementation of the bulk import feature.

* 1. **Authority management function**

In this program, there are two kinds of permissions. One is a normal user and the other is an administrator user. Ordinary users can perform operations such as querying gene sequence information and modifying their own personal information but have no authority to modify or delete gene sequence information and other user information. The administrator user has the authority to modify and delete the gene sequence information and other users' information.

We use **root** to manage the users’ permission.



After the user logs in, the information will be saved in a cookies.txt file. Each time an administrator operation is involved, the root function will be called to query the cookies to determine whether they have permissions.

MySql data query function

SQL fuzzy query, use **like** comparison keyword, plus wildcard in SQL, for example:

1. LIKE 'Mc%' will search for all strings beginning with the letter Mc (eg McBadden).

2. LIKE '%inger' will search for all strings ending with the letter inger (eg Ringer, Stringer).

3. LIKE '%en%' will search for all strings containing the letter en at any location (eg Bennet, Green, McBadden).

4. LIKE'\_heryl' will search for all six-letter names ending in the letter heryl (eg Cheryl, Sheryl).

5. LIKE'[CK]ars[eo]n' will search for the following strings: Carsen, Karsen, Carson, and Karson (such as Carson).

6. LIKE'[M-Z]inger' searches for all names (such as Ringer) that begin with the string inger and begin with any single letter from M to Z.

7. LIKE'M[^c]%' will search for all names beginning with the letter M and the second letter is not c (eg MacFeather)

**2.3 Register and login**

The user registration interface can input the user name, email, password and other information input by the user into the database. In addition, We use **if\_else** structure to determine whether the user's two password inputs are consistent. If the two passwords match, the registration is successful; if they are inconsistent, it will show wrong password.

We use the function **m.update(pwd.encode)** to encrypt passwords. Then, we use **psw = cus1.fetchall(**) to query password according the username. These functions help us to match the user's password and complete the user's login operation.

**2.4 Query function**

We use the function sql = 'select \* from law where ID=%s or patentNum=%s' to query the information through the ID number. And **result** function. **Result** likes dictionary, result[“key”] could acquire the corresponding content.

1. **Build Introduction**

Use the link: <https://github.com/CSCI6221-Spring-2019-Renner/Group9-project1.git>. Download the compressed file *6221\_project 1.zip*.

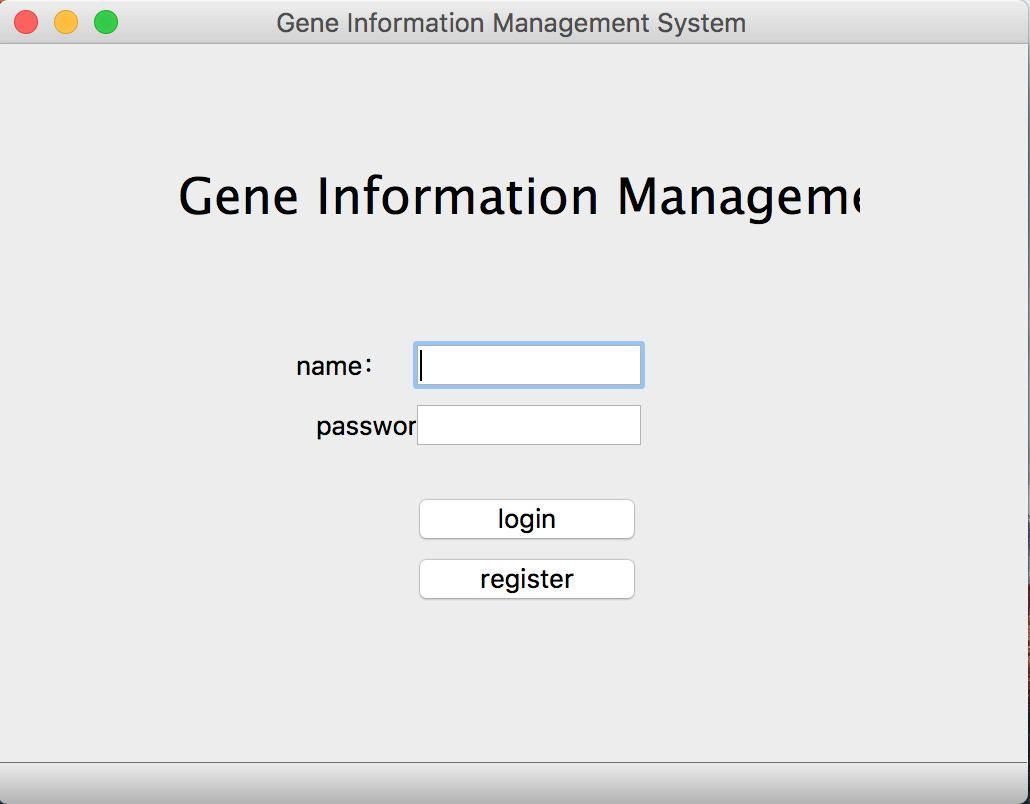
Configuration MySQL database.

Open the Python project with Python compiler.

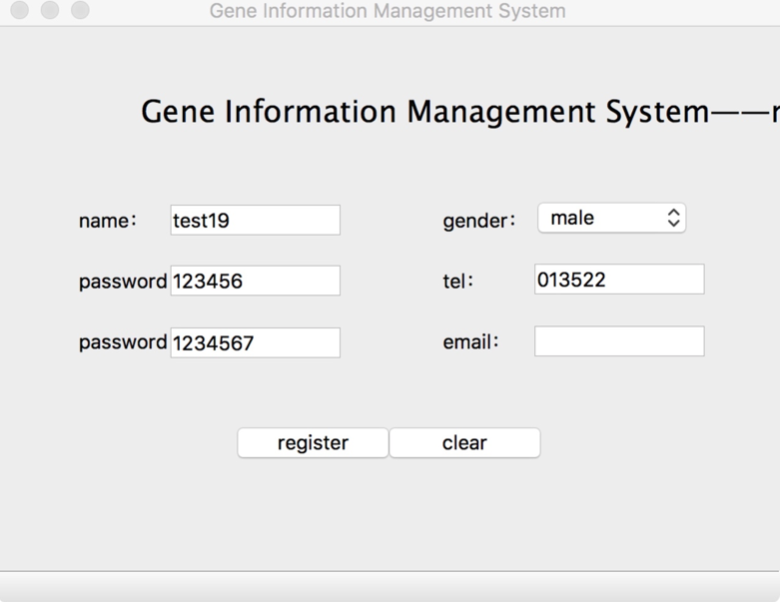
1. **Program running results**
   1. **Register and login**

**4.1.1 Register**

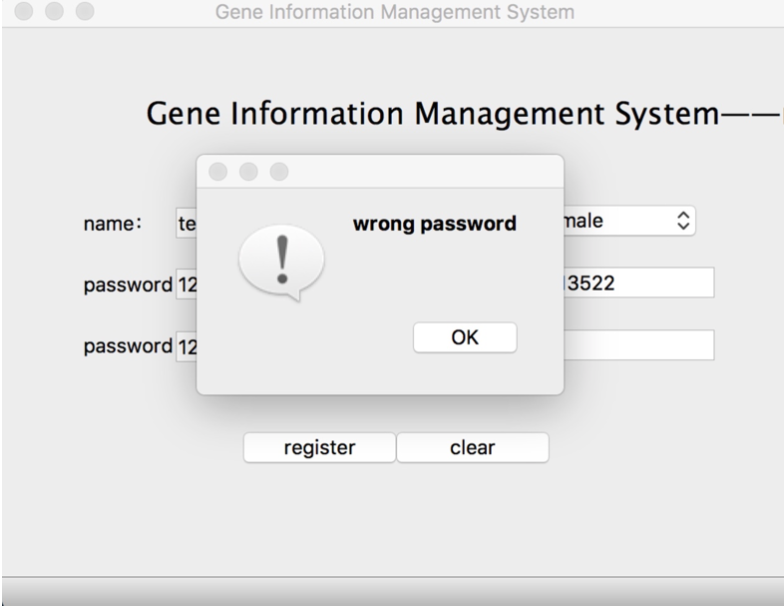
New user could click the register button to register.



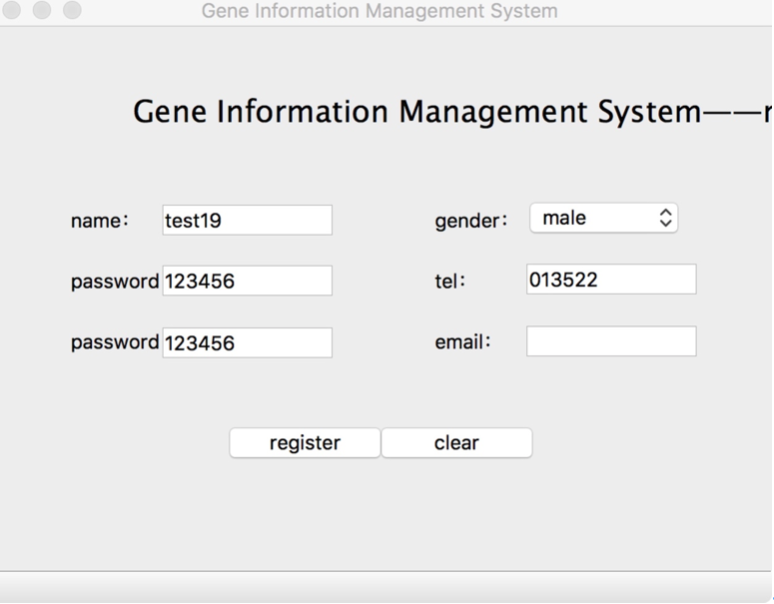
This is the register interface. New user should input their name, gender, password and email to register successfully.

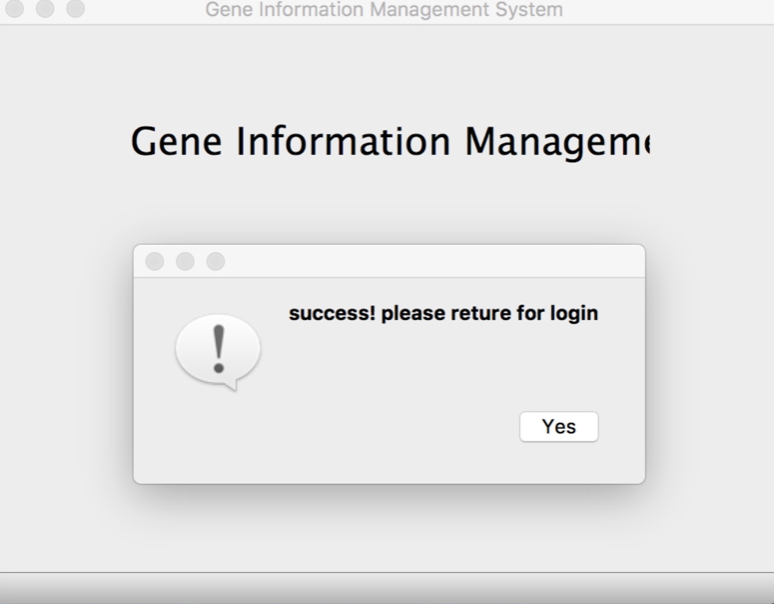


If the two passwords entered by the user are mismatch, the actual registration fails.



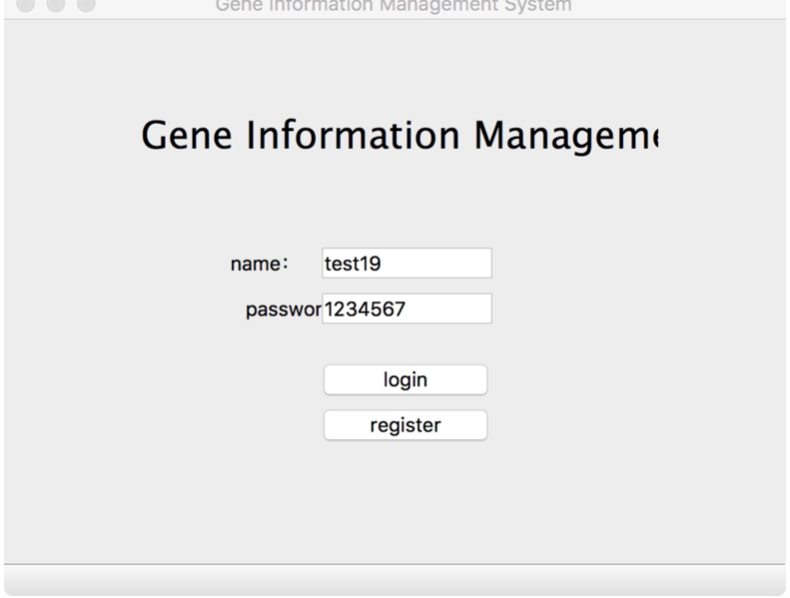
If the two passwords entered by the user are match, the actual registration success.

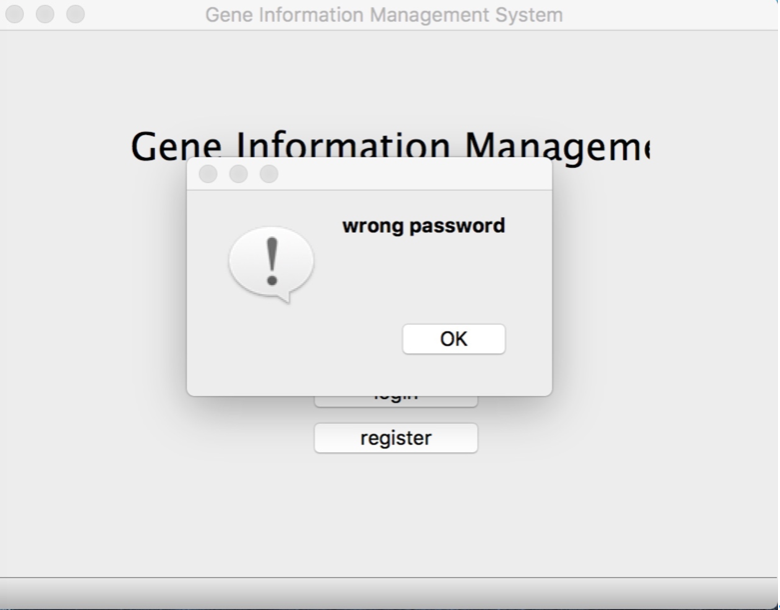




**4.1.2 Login**

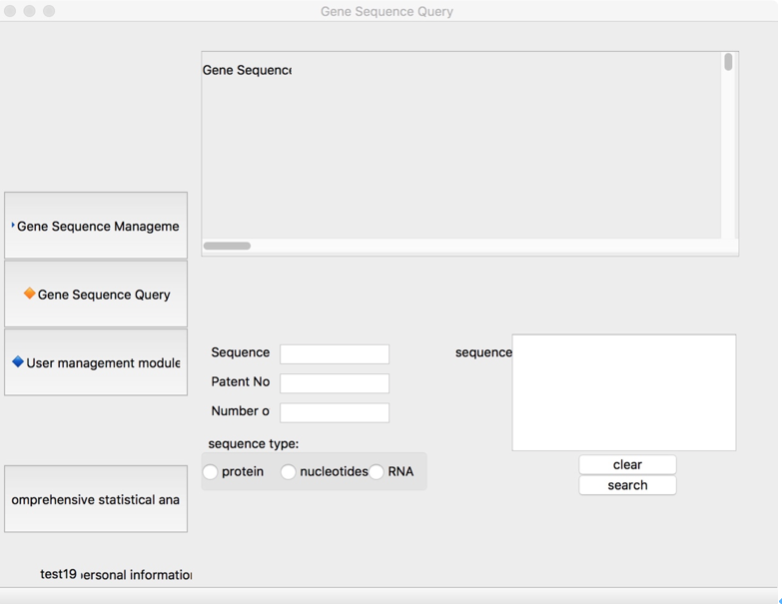
After the successful registration, the user can log in to the system through the login interface. However, when the user enters the wrong password, the user cannot log in to the system.





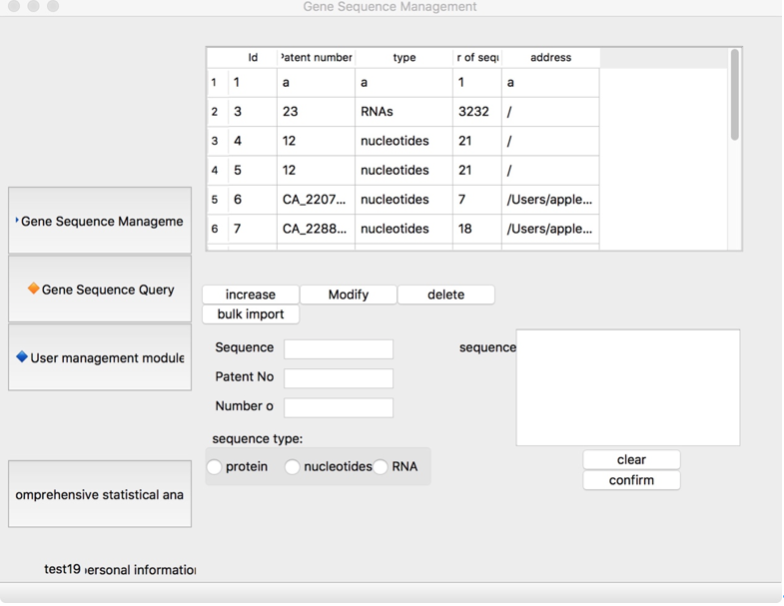
**4.2 Main interface**

This is the main system interface when the user successfully logs in. There are three modules in the left: gene sequence management, gene sequence query and user management module.

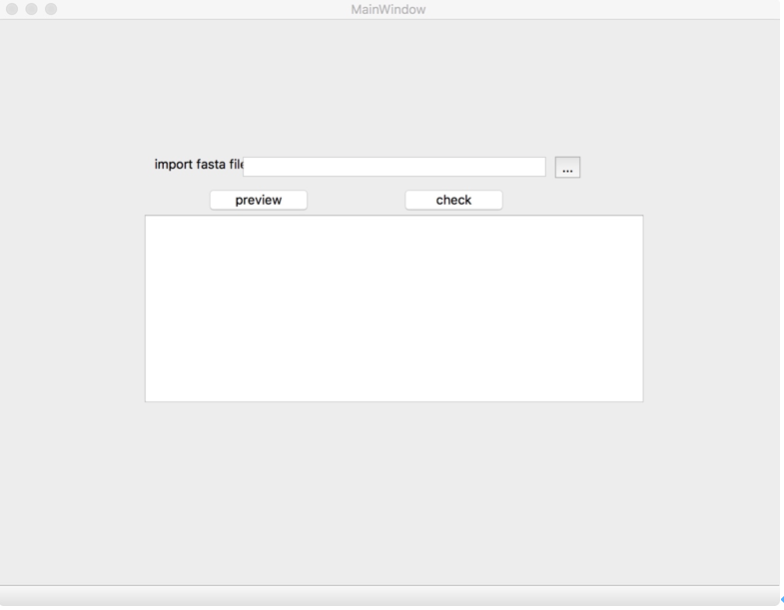


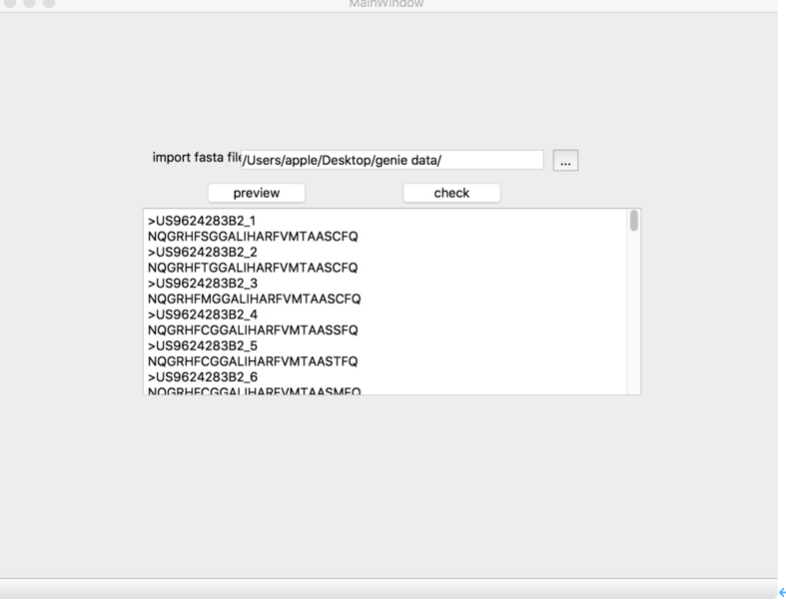
**4.2.1 Gene sequence management module**

This is the gene sequence management module. Users can view the gene list information in the database through the module above. There are also three buttons for adding, modifying, and deleting. Users can add, modify, and delete information through them. There also has a bulk import button, user could use it to add bulk data.



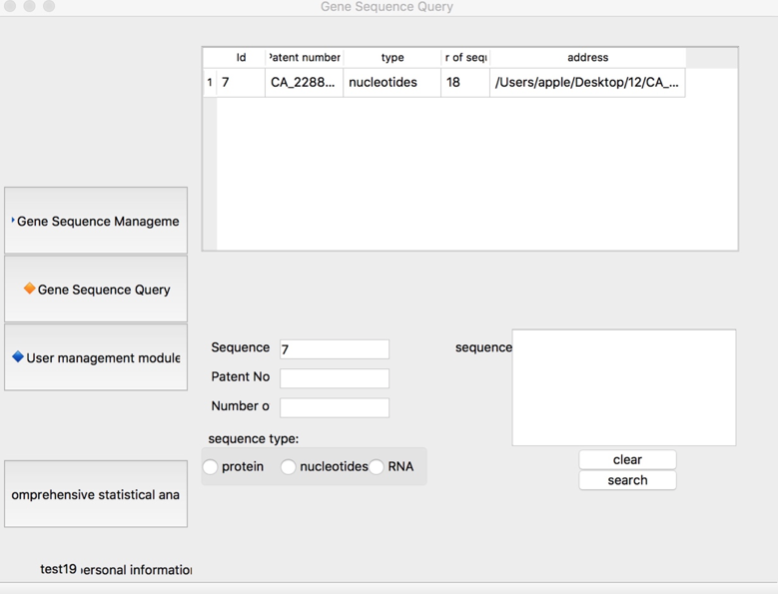
This is the bulk import interface.

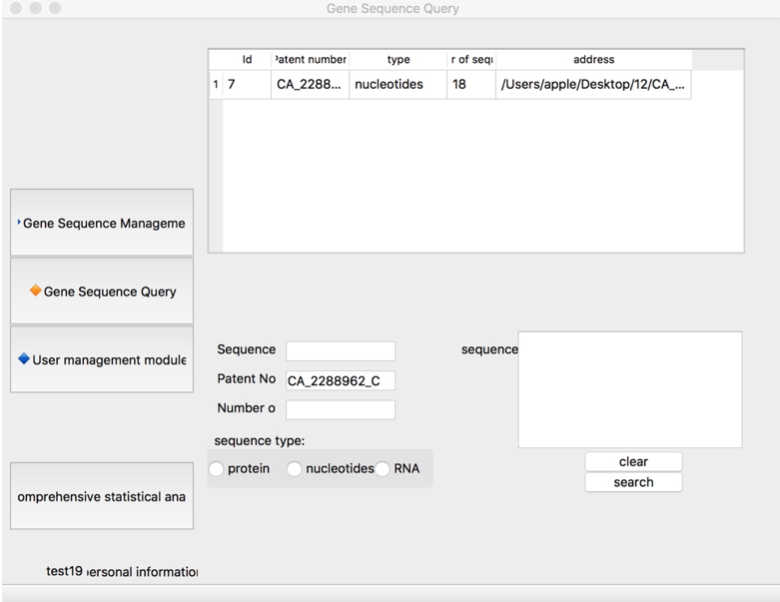


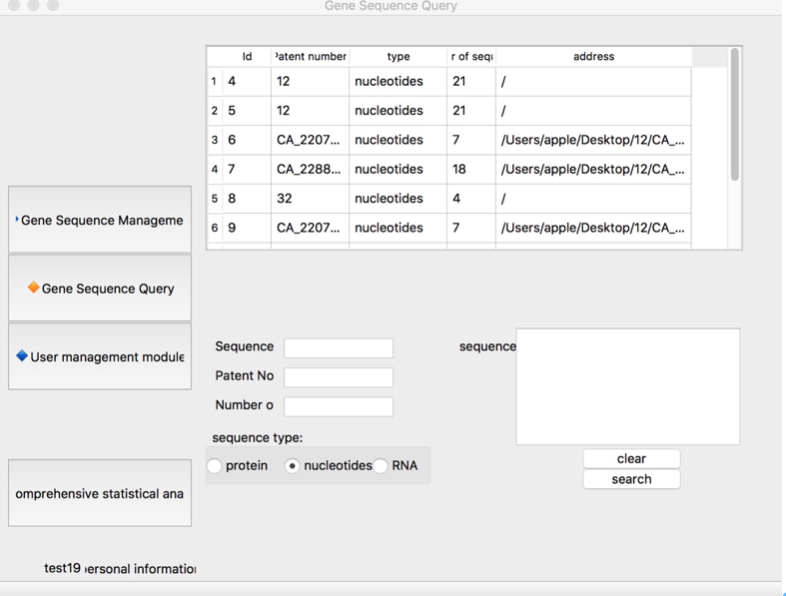


**4.2.2 Gene query module**

This is gene query module. Users can query gene information by entering keywords.

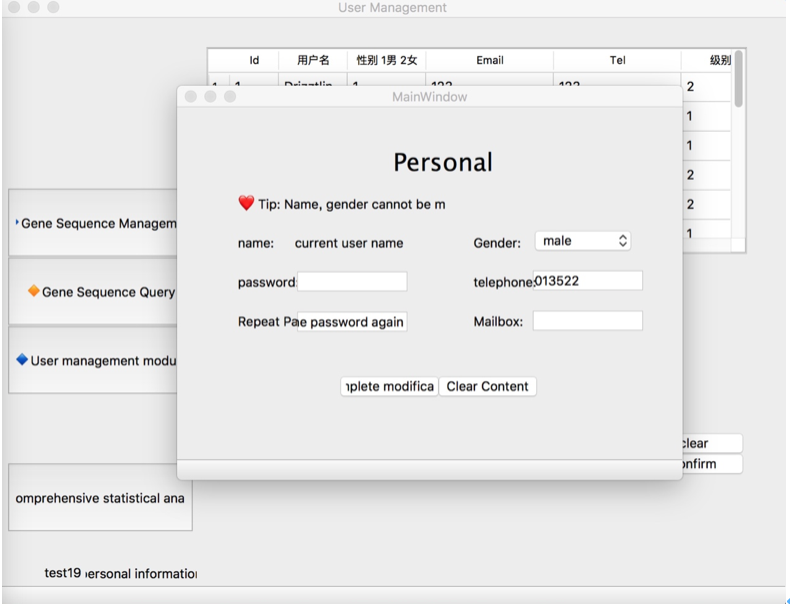






**4.2.3 User management interface**

This is the user management interface. User could modify their personal information through this module.



1. **Conclusions**

Python is an interpreted language, and although it runs relatively slowly, it is very efficient to design and develop a project. Through the design and development of this Python project, we have reached several conclusions. First of all, the design and development of the project are to complete the construction of the main body of the project and then complete the development of the main elements (main components, main functions), and finally the functional improvement and design to enhance the user experience. Second, in the process of writing code, we must remember to refactor the code. While developing, we must think about whether a function is too redundant. If it is, then we should consider the refactoring function. Refactoring really needs It takes a certain amount of time, but it can bring greater convenience to future development. Third, the documentation of the development process is also very important. It is important to note that each function is written with an online functional description, which brings great convenience to the subsequent code modification and maintenance. In the demand phase, time and effort should be allocated reasonably, and a very difficult plan cannot be developed in a short period of time. In addition, the main components of a project are the main program, configuration files, function file, various objects required by the project, and how to determine the object needs to be determined according to the specific needs of the project.

References

Frese, R. (2003). Project success and failure: what is success, what is failure, and how can you improve your odds for success?

Retrieved from <http://www.umsl.edu/~sauterv/analysis/6840_f03_papers/frese/>

Gulla, J. (2012). Seven Reasons IT Projects Fail.

Retrieved from [http://ibmsystemsmag.com/power/systems-management/workload- management/project\_pitfalls/](http://ibmsystemsmag.com/power/systems-management/workload-%20management/project_pitfalls/)

What is management information systems? (n.d.).

Retrieved from <http://mays.tamu.edu/info/what-is-mis/>

Delivering Business Analytics and Technology Solutions. (n.d.).

Retrieved from [https://saunders.rit.edu/undergraduate/majors-minors/management- information-systems-mis-degree-overview#](https://saunders.rit.edu/undergraduate/majors-minors/management-%20information-systems-mis-degree-overview)

Boykin, G. (2017). The History of Management Information Systems.

Retrieved from [https://bizfluent.com/about-5444925-history-management-information- systems.html](https://bizfluent.com/about-5444925-history-management-information-%20systems.html)