Advance Programming – Assignment 2 (MiniNet)

3613252 – Jyh Woei Yang

3666814 – Yujue Zou

1. Explain the changes if you use a different design compared to your assignment 1

This time, we used GUI scene to display the program instead of displaying in the console, and all the data will be stored inside of the text file - there will be People.txt and Relationship.txt.

2. Explain how the new classes are organized

The start up Class MiniNet creates the GUI of the primary stage and the GUI of the details of each item. Class driver control the entire program. Class Adults, Children and YoungChild extend class Person. All the details of the people and relationships are saved in the text file respectively. By going through PeopleDao.class and RelationshipDao.class, they will have the read from file and write into file functions. Apart from that, they are both connected with the Database and used it as a backup data source when the system cannot find People.txt.

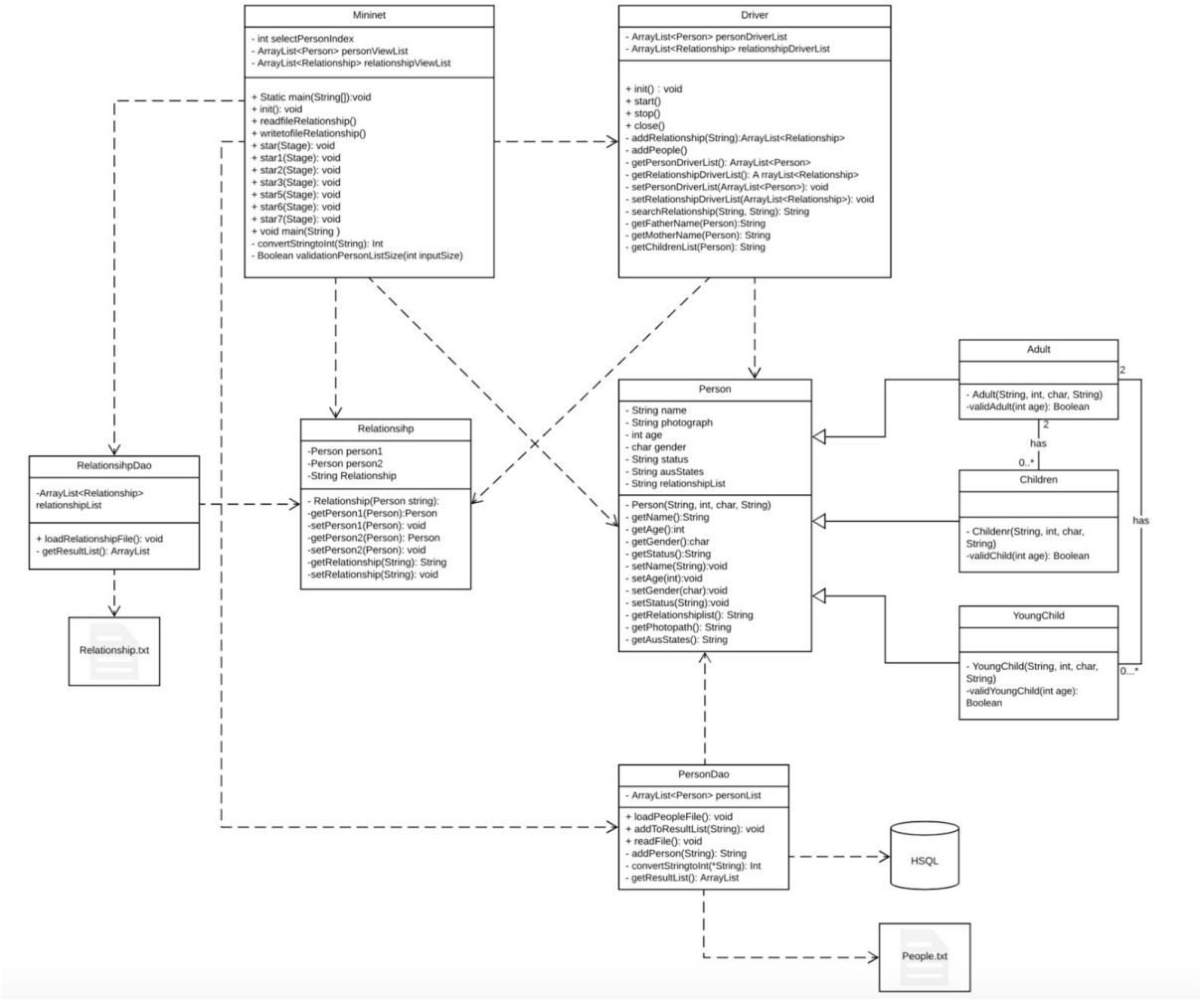
There are eight different kinds of exception classes, when the inputs do not match the requirements, it will throw exceptions including:

- **TooYoungException**
- **NotToBeFriendsException**
- **NoParentException**
- **NoAvailableException**
- **NotToBeCoupledException**
- **NoSuchAgeException**
- **NotToBeColleaguesException**
- **NotToBeClassmatesException**

The main class, MiniNet.class, use try and catch blocks to maintain error handling.

You can see those interactions as following class diagram:

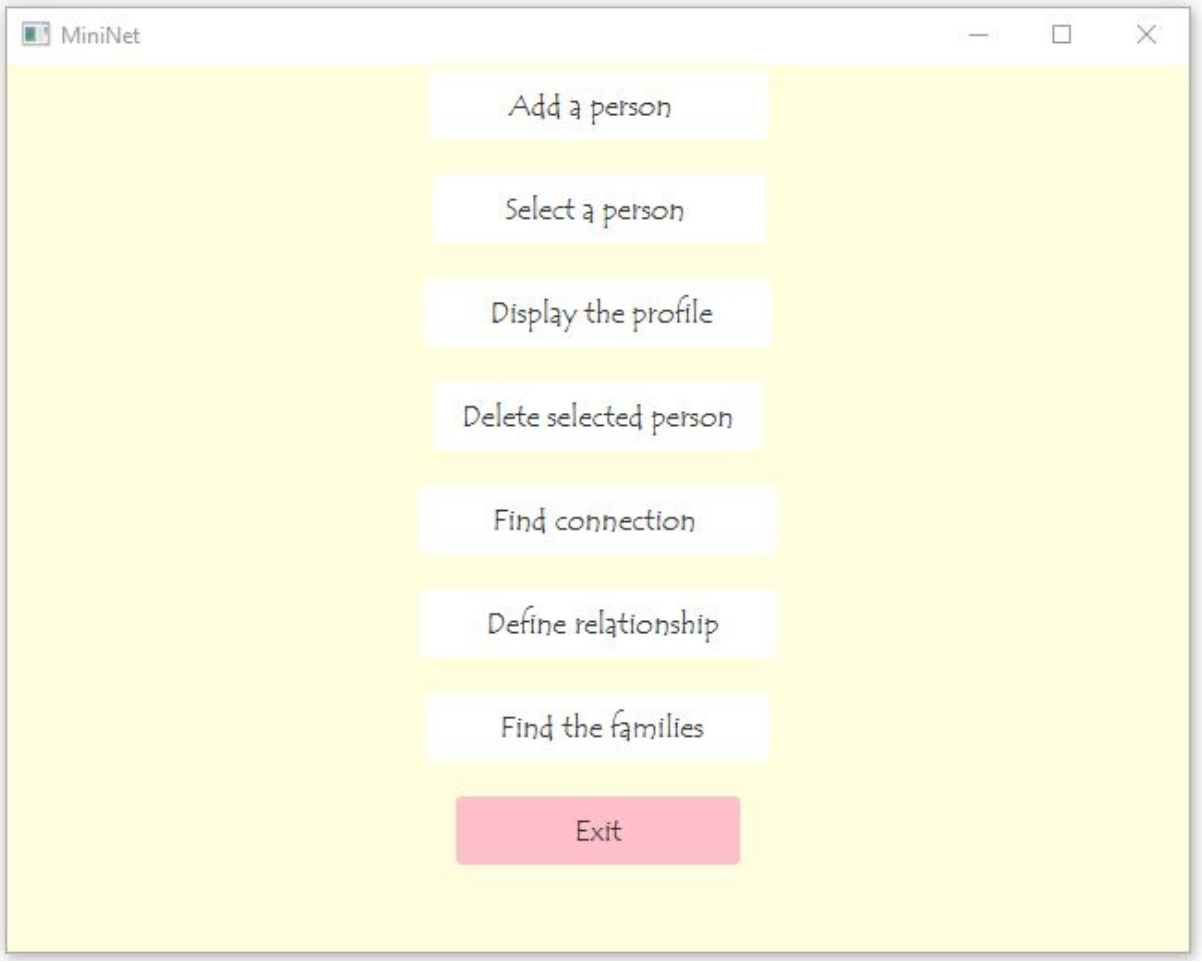
Class Diagram



3. Explain the process by which your program will interact with user and external data source to run a game.

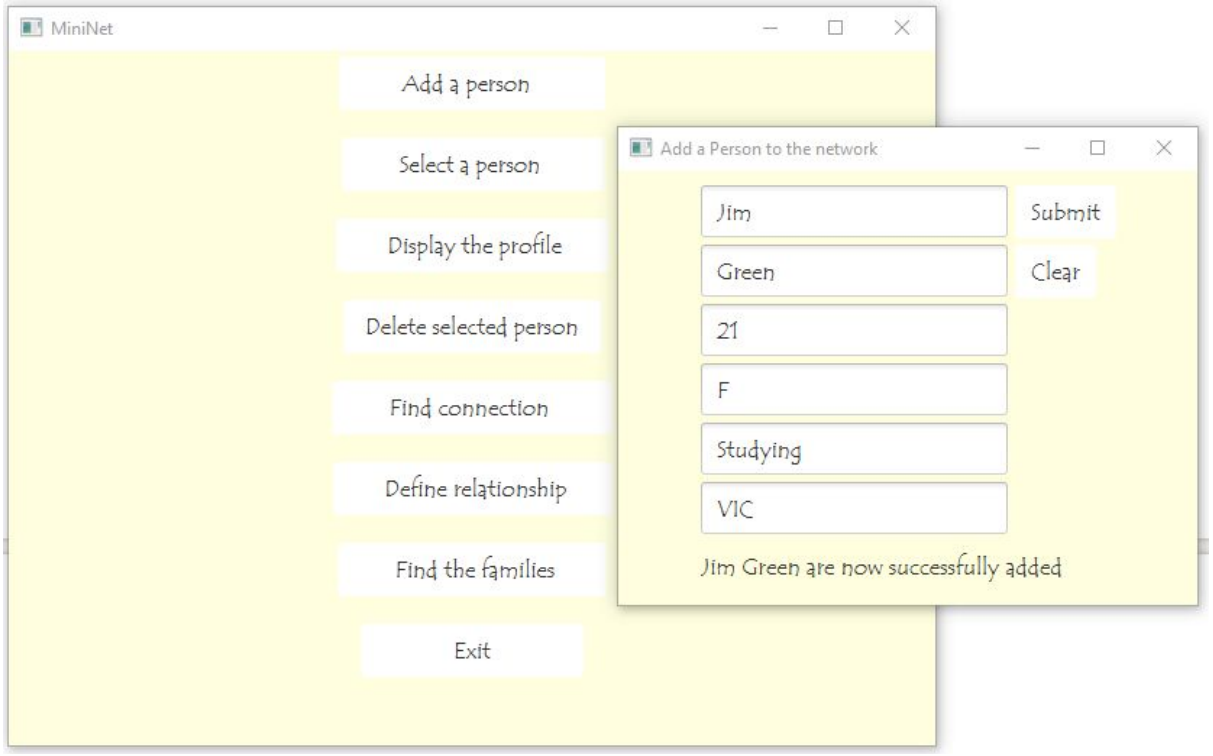
1) MiniNet

The scenes will be created in MiniNet, all the choices of the interface will be displayed in the primary stage, it includes add a person, select a person, display the profile, delete selected person, find connection, define relationship, and find the families. If the user click any of them, it will go into the second stage.



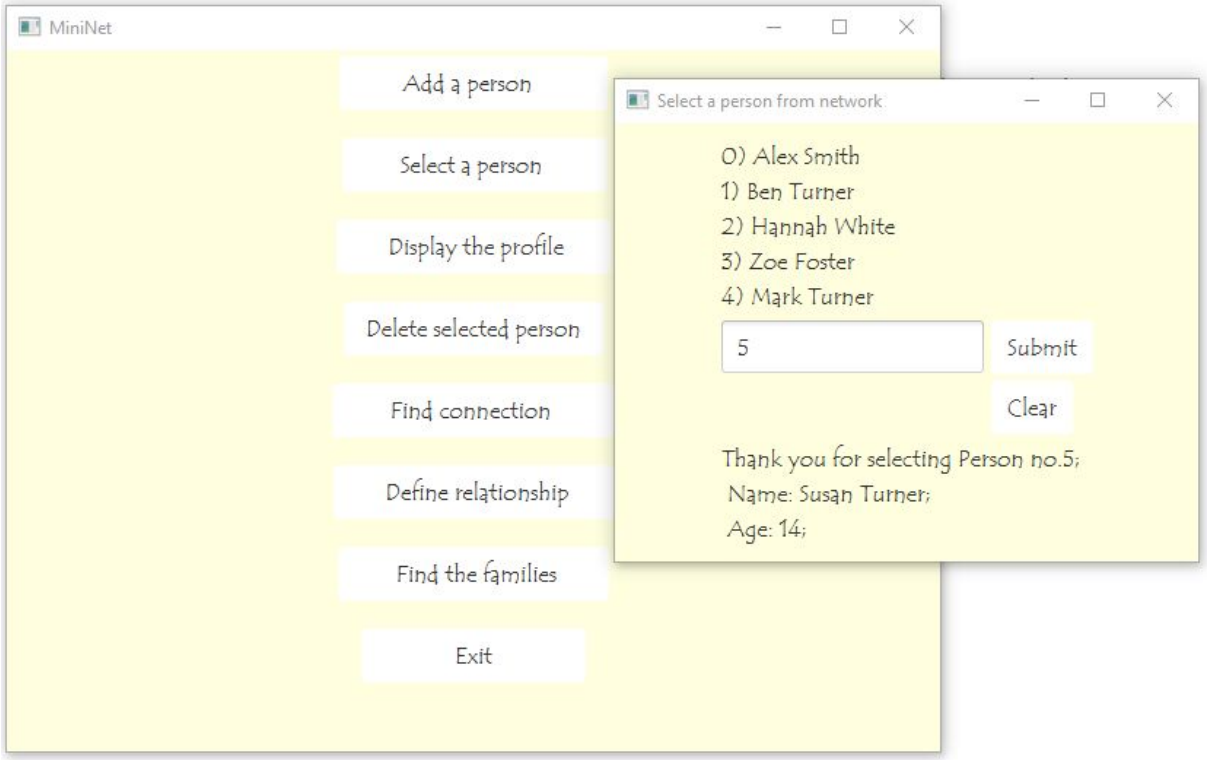
2) Add a person

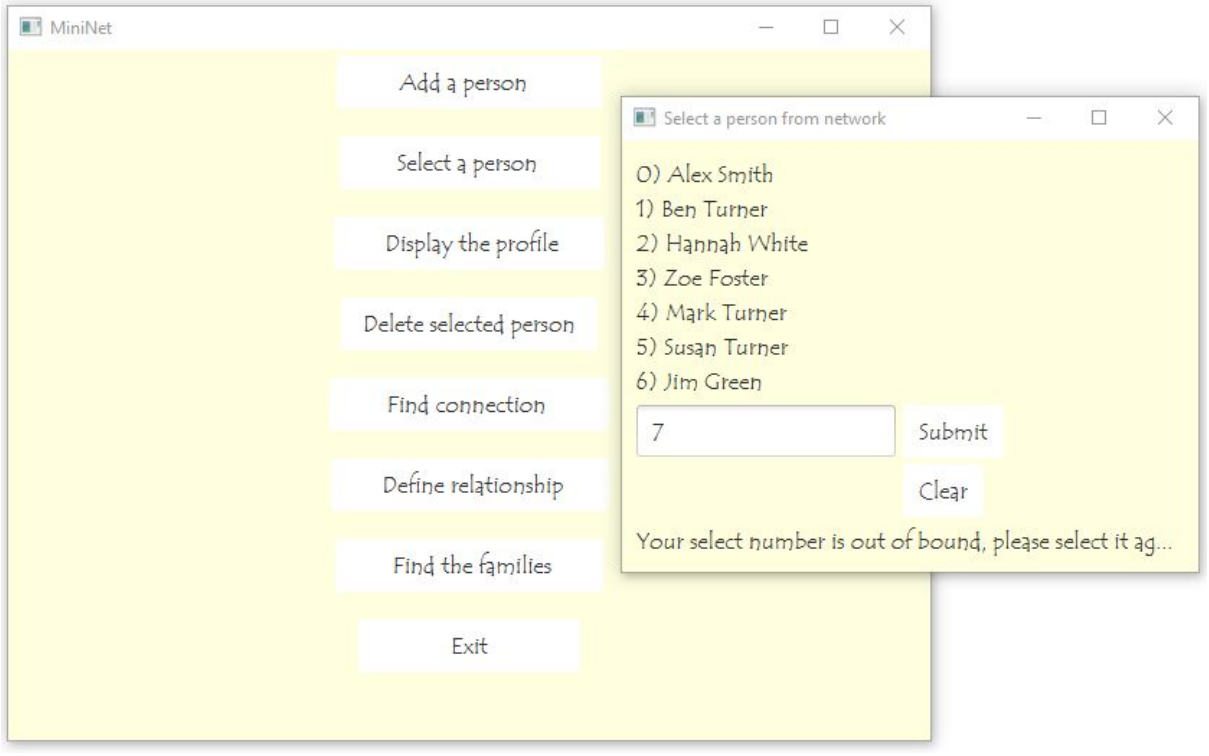
When the user click ‘Add a person’, the user will go into the second scene of ‘Add a person into the network’, the user can input the details of a person, including first name, last name, age, gender, status, state. By doing this, the information of the person will be stored inside PeopleDao.class as an ArrayList. The relationships between persons will be stored inside RelationshipDao.class as an ArrayList, and the result will display on the second GUI stage. Once the user exit the system, the data will be stored back into the text files.



3) Select a person

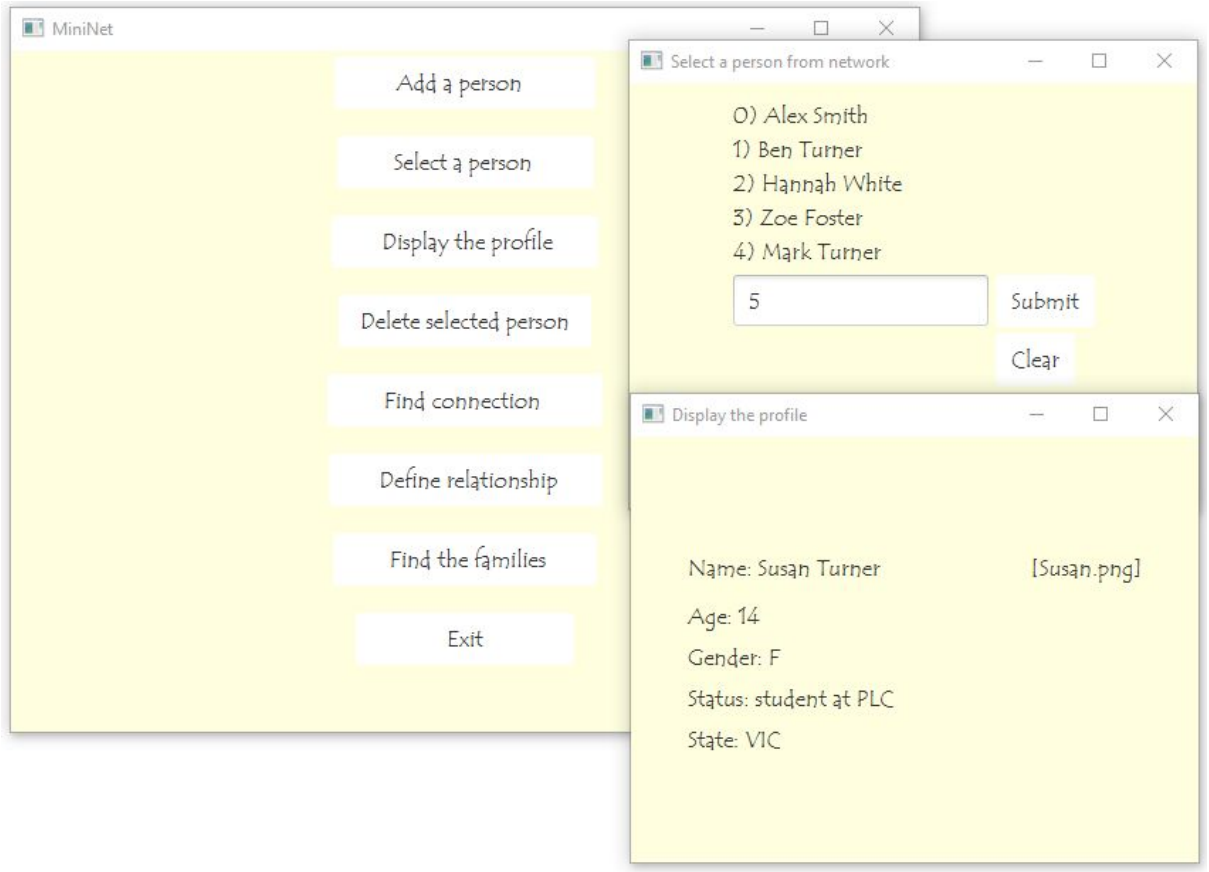
When the user click ‘Select a person’, the user can go into a second stage, and choose submit the number of the person. If the user want to re-enter the number, they can click ‘Clear’. If the select number is out of bound, the user should select again.





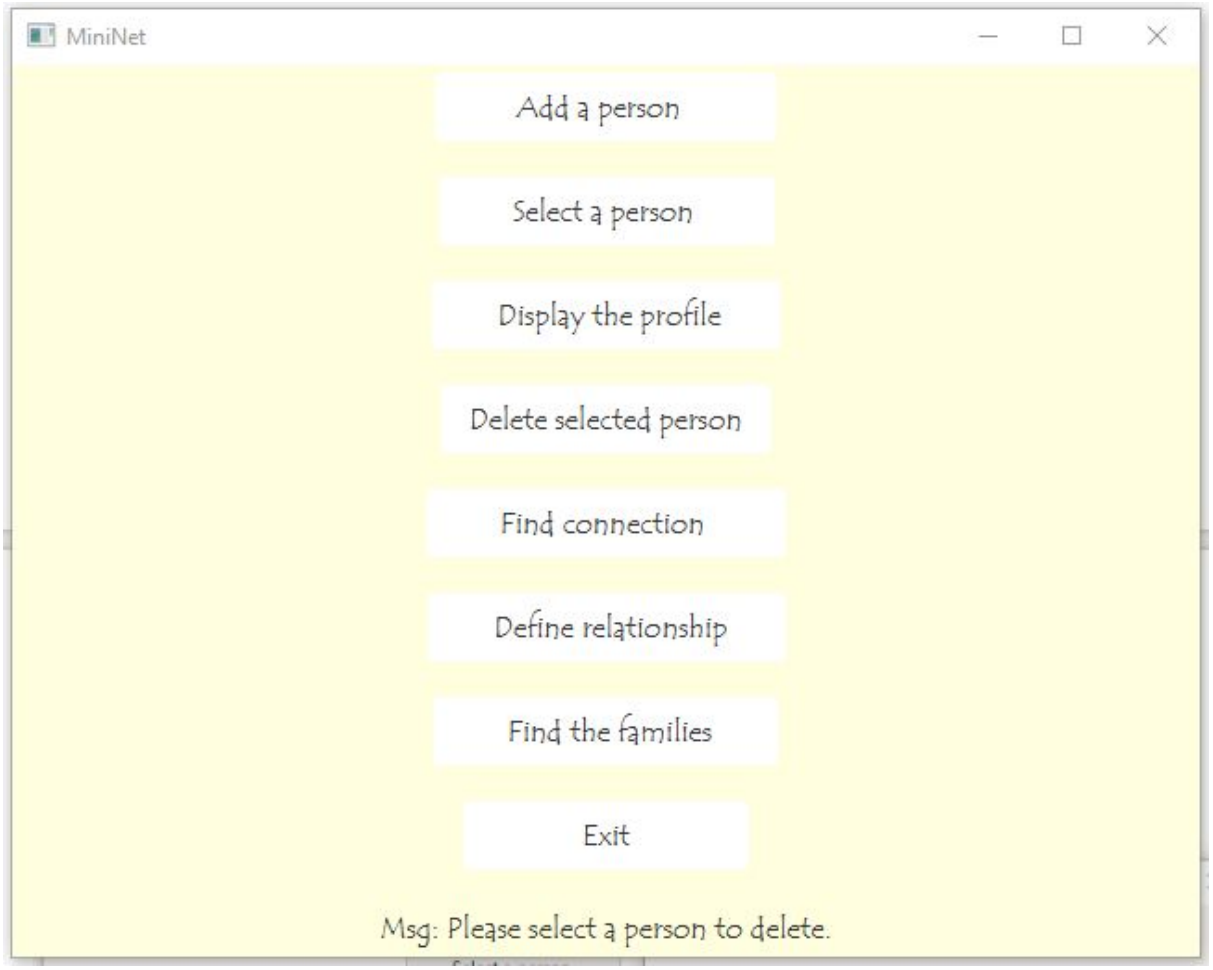
4) Display the profile

When the user select a person and then click ’Display the profile’, the details of the selected persons will be displayed. If the user didn’t select a person, it will display a message ‘Please select a person to display’.



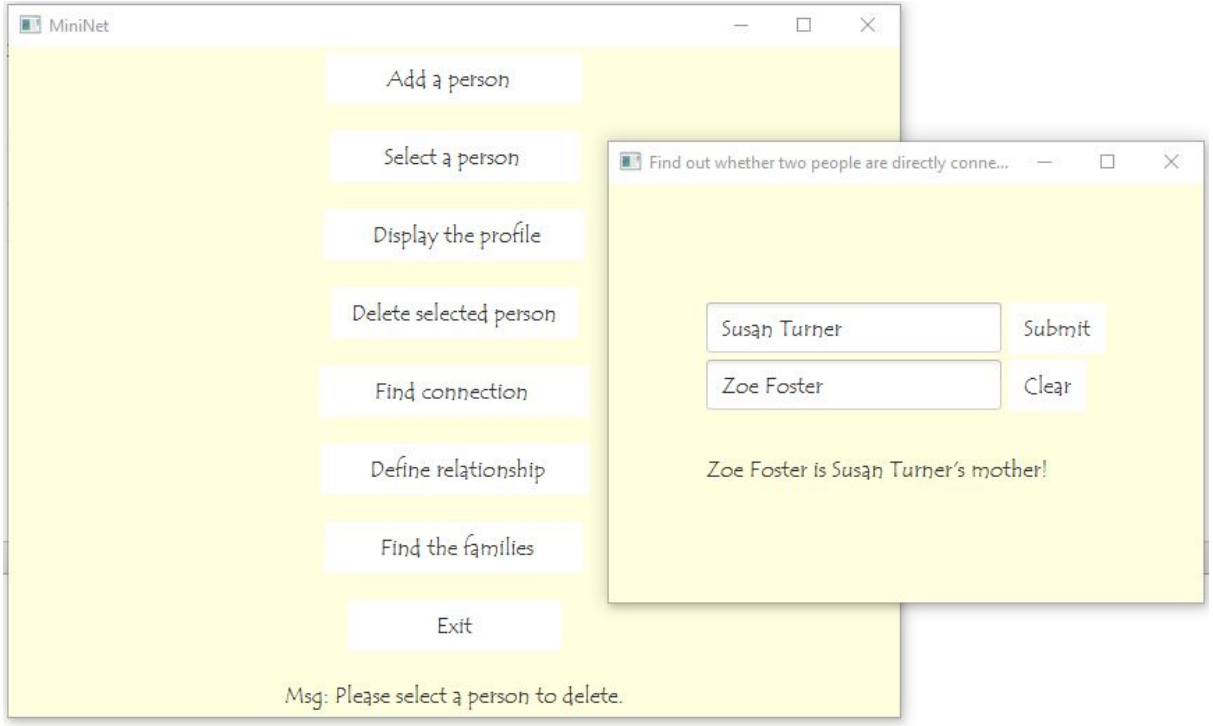
5）Delete selected person

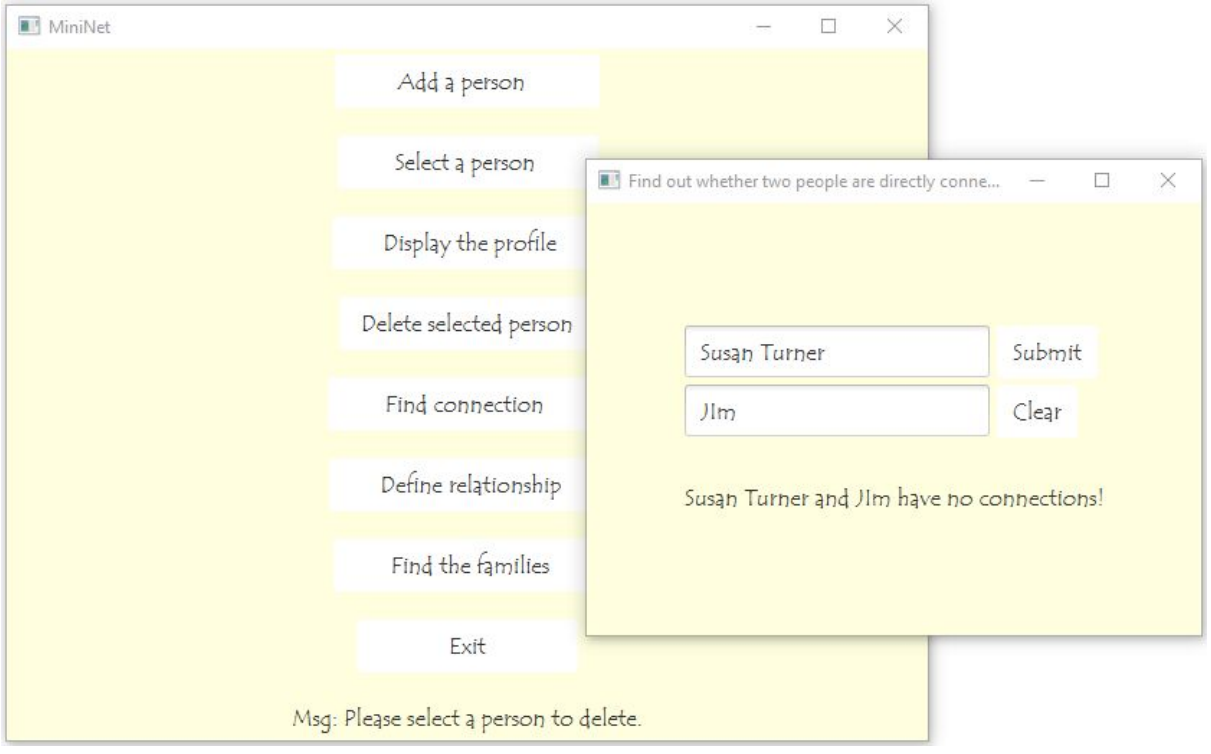
When the user selected a person first, and then click ‘Delete selected person’, the person will be deleted. If the user didn’t select a person, it will display a message ‘Please select a person to delete’.



6) Find connection

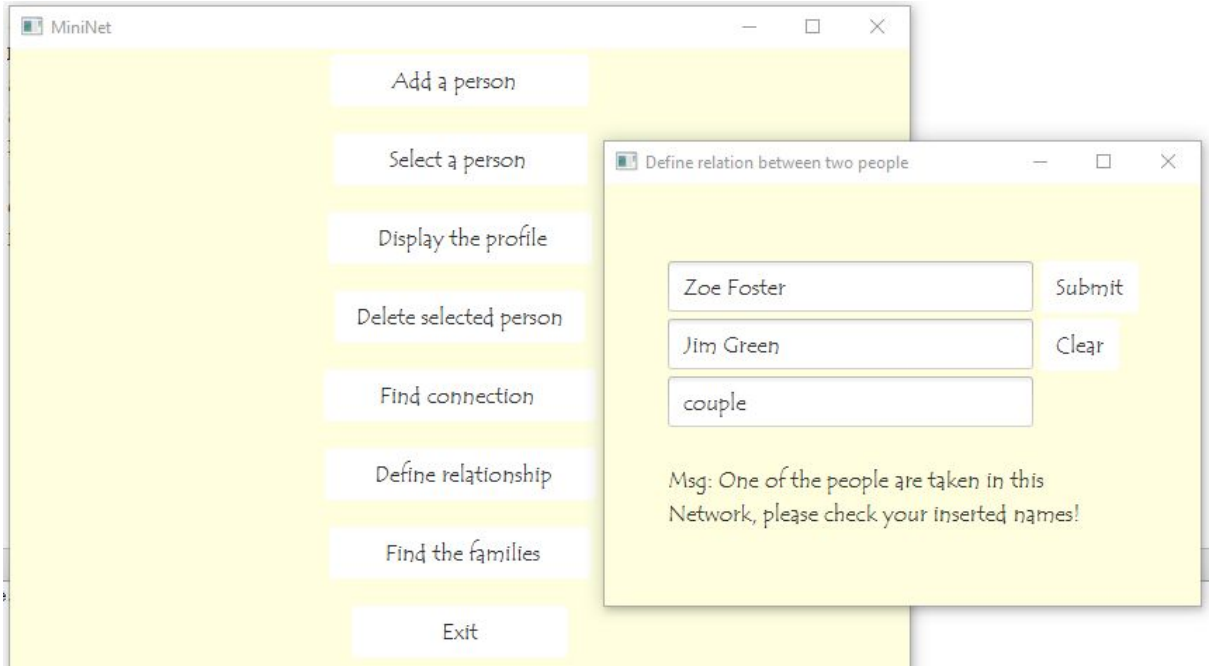
When the user click ‘Find connect’, the user will go into a second stage and input two person’s name, then they can find out person’s connection. The names entered should in the database. If the person’s name are not in the database, it will show that the two persons have no connection.

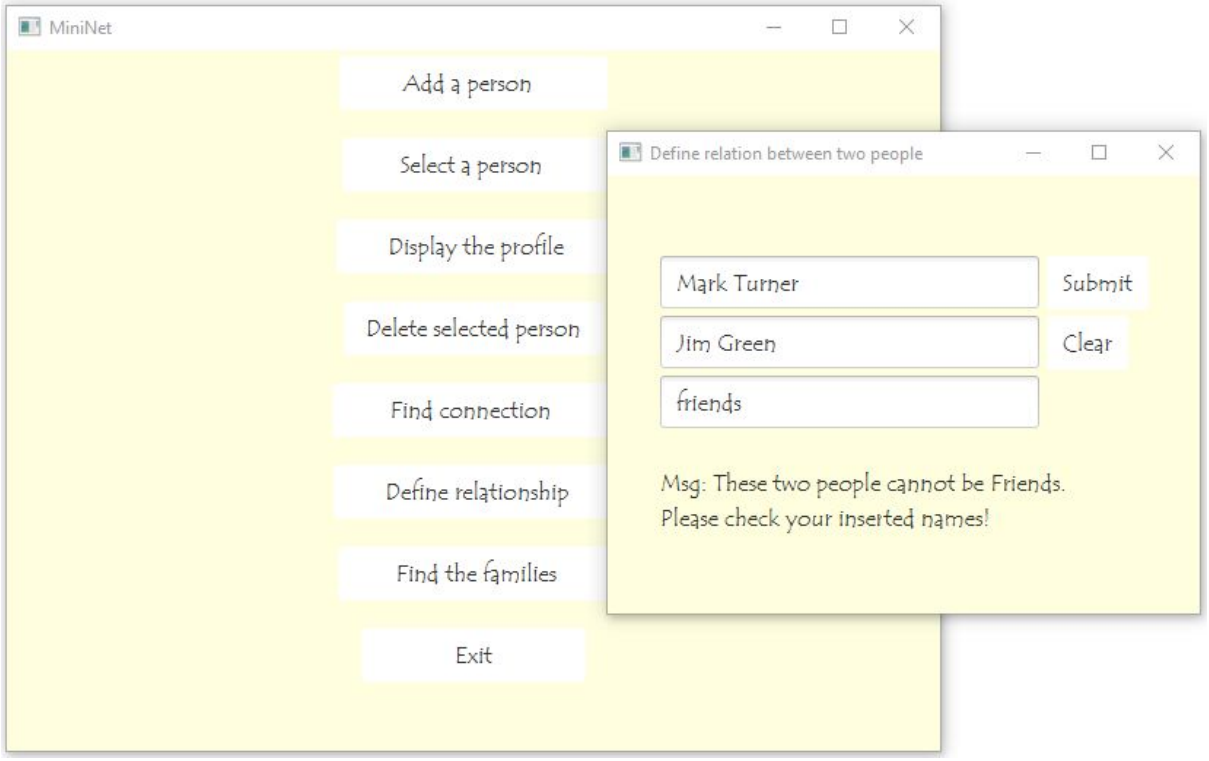


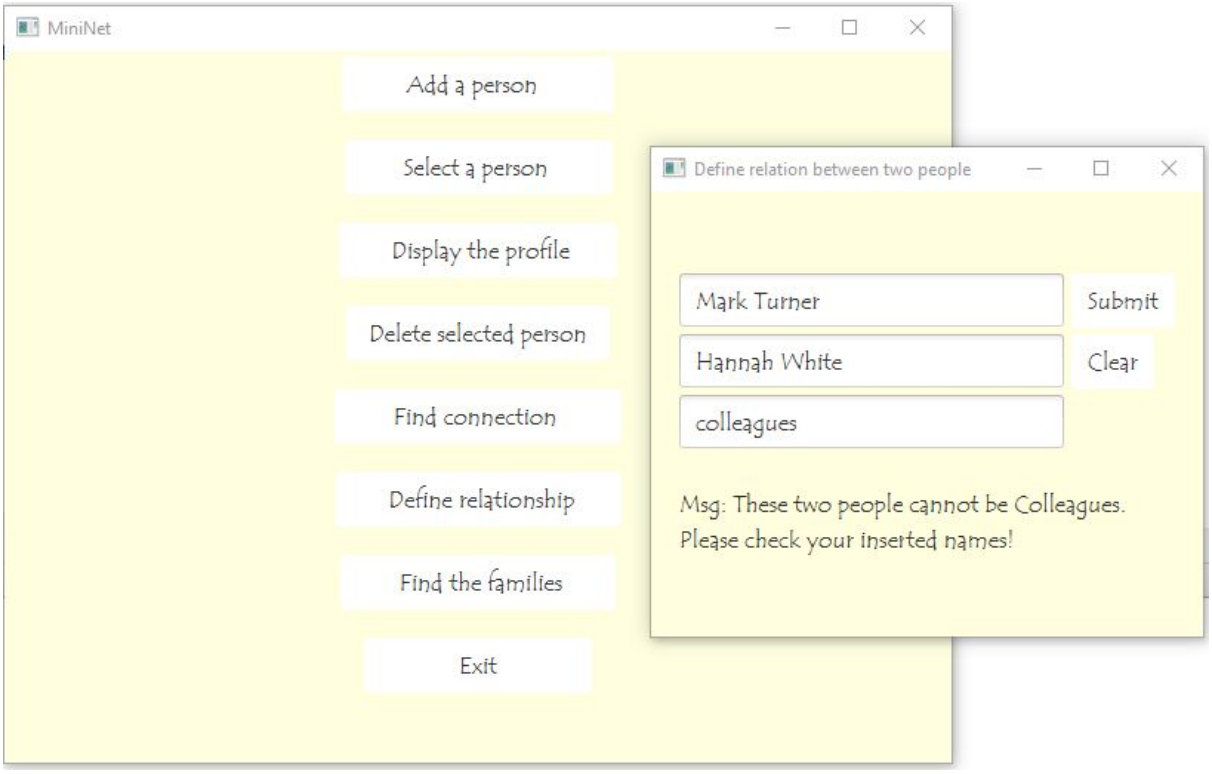


7) Define relationship

When the user click ‘Define relationship’, the user can input two person’s name, and set their relationship. There are some input validation being implemented. At the background, we use Java exceptions to handle these errors. (e.g. If the two children's’ age difference are over than 3 years, they can not be friends.)

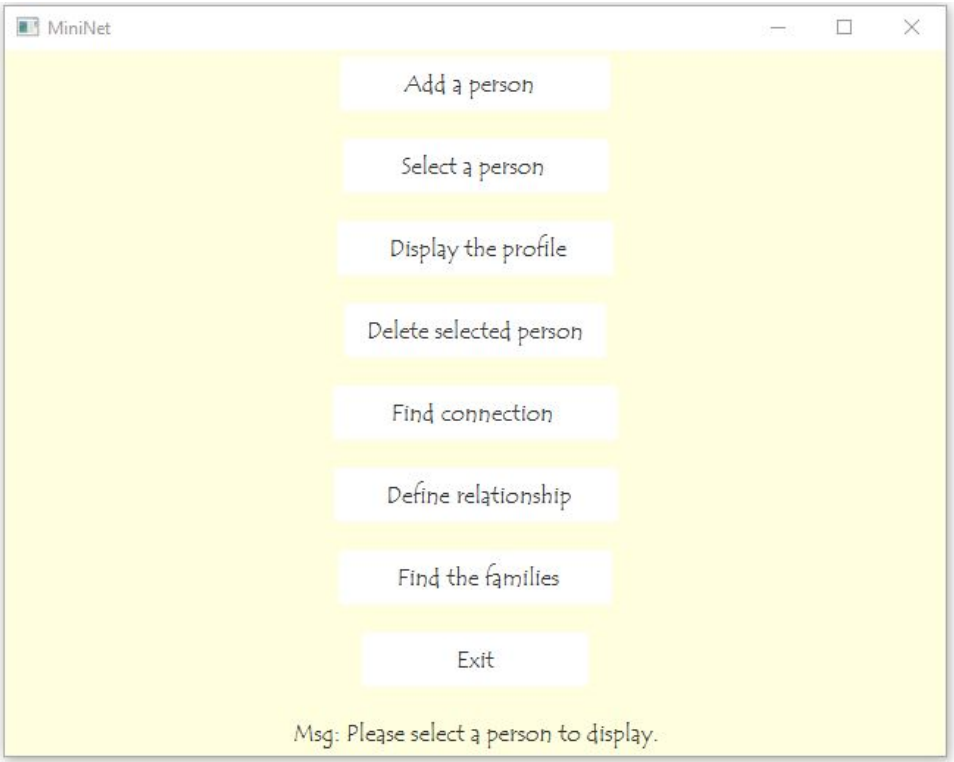


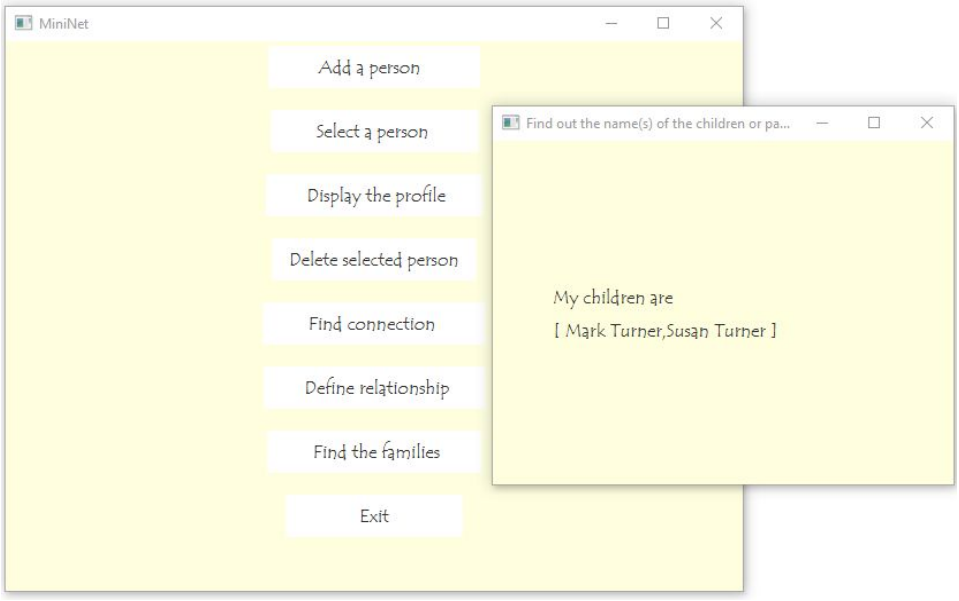

MiniNet
Add a person
Select a person
Display the profile
Delete selected person
Find connection
Define relationship
Find the families
Exit
Define relation between two people
Mark Turner
Jim Green
friends
Submit
Clear
Msg: These two people cannot be Friends.
Please check your inserted names!


MiniNet
Add a person
Select a person
Display the profile
Delete selected person
Find connection
Define relationship
Find the families
Exit
Define relation between two people
Mark Turner
Hannah White
colleagues
Submit
Clear
Msg: These two people cannot be Colleagues.
Please check your inserted names!

8) Find the families

When the user select a person firstly, and then click ‘Find the families’, the second stage will display the name(s) of the children or parents. They have to be the selected person. If the user didn’t select a person first, it will display a message ‘Please select a person to display’.





9) Exit

If the user click ’Exit’, the GUI will close and the whole program will stop.

External Data

When an embedded HSQLDB database connection is found and there is no people.txt in the system, then your program should read the personal data from the database. Any updates on the personal data should be stored in the database as well.

```java
/**
 * A method to read user from file
 *
 * @param
 * @return ArrayList<Person>
 * @throws FileNotFoundException if file is not found
 * @throws IOException while exception during I/O actions
 */
public void loadPeopleFile(){

    String fileName = "People.txt";
    try{
        FileReader inputFile = new FileReader(fileName);
        Scanner console = new Scanner(inputFile);
        while(console.hasNextLine())
        {
            String userString = console.nextLine();
            String[] details = userString.split(",");
            Person person = new Person();
            person.setName(details[0]);
            person.setPhotoPath(details[1]);
            person.setAge(convertStringtoInt(details[4]));
            person.setGender(details[3].charAt(0));
            person.setStatus(details[2]);
            person.setAusStates(details[5]);
            personList.add(person);
            //person.displayPerson();
        }
        inputFile.close();
    }
    catch(FileNotFoundException exception)
    {
        System.out.println(fileName + " not found");
        readFromDB();
    }
    catch(IOException e){
        System.out.println("Error: Invalid file");
        readFromDB();
    }
}
```

If People.txt file not found, it will call readFromDB() to load data from database.

```java
/**
 * A method to read user from database
 *
 * @param
 * @return
 * @throws SQLException if there is a SQL error
 * @throws ClassNotFoundException while exception there is no class found
 */
public void readFromDB()
{
    Server hsqlServer = null;
    Connection connection = null;
    ResultSet rs = null;

    hsqlServer = new Server();
    hsqlServer.setLogWriter(null);
    hsqlServer.setSilent(true);
    hsqlServer.setDatabaseName(0, "MiniNetDB");
    hsqlServer.setDatabasePath(0, "file:MYDB");

    hsqlServer.start();

    //making a connection
    try{
        Class.forName("org.hsqldb.jdbcDriver");
        connection = DriverManager.getConnection("jdbc:hsqldb:MiniNetDB","sa","123");

        connection.prepareStatement("drop table people if exists;").execute();
        connection.prepareStatement("create table people (name varchar(20) not null,photopath varchar(20) not null,age varchar(20) not null,gender varchar(20)

        connection.prepareStatement(addPeople("Andy","N/A","30","M","Working at KFC","VIC")).execute();
        connection.prepareStatement(addPeople("Alex Smith","N/A","21","M","Student at RMIT","WA")).execute();
        connection.prepareStatement(addPeople("Ben Turner","BenPhoto.jpg","35","M","Manager at Coles","VIC")).execute();
        connection.prepareStatement(addPeople("Hannah White","Hannah.png","14","F","Student at PLC","VIC")).execute();
        connection.prepareStatement(addPeople("Zoe Foster","N/A","28","F","Founder of ZFX","VIC")).execute();
        connection.prepareStatement(addPeople("Mark Turner","Mark.jpeg","2","M","N/A","VIC")).execute();
        connection.prepareStatement(addPeople("Susan Turner","Susan.png","14","F","Student at PLC","VIC")).execute();

        // set a flag to count how many people in this network.
        int dataNumber = 7;

        //query from the db
        rs = connection.prepareStatement("select name,photopath,age,gender,status,ausstates from people;").executeQuery();
        for (int i = 0 ; i < dataNumber ; i++)
        {
            rs.next();
            System.out.println(String.format("Name: %1s, Photopath: %1s, Age: %1s, Gender: %1s, Status: %1s, AusStates: %1s", rs.getString(1), rs.getString(2)
```

How to launch this application:

1) Download the COSC1295_AP_Assignment2_3613252_3666814.zip.
2) Unzip the file.
3) Double click the MiniNet.jar in /Release folder to launch

or

Open the terminal, go to the launch folder and execute:

java -jar MiniNet.jar

```
[tomz123s-MacBook-Pro:FirstSubmission tomz123$ cd COSC1295_AP_Assignment2_3613252
 _3666814
[tomz123s-MacBook-Pro:COSC1295_AP_Assignment2_3613252_3666814 tomz123$ ls
 Design Report – COSC1295_AP_Assignment2_3613252_3666814.pdf
 Release
 tomz123s-MacBook-Pro:COSC1295_AP_Assignment2_3613252_3666814 tomz123$ cd Release
[tomz123s-MacBook-Pro:Release tomz123$ ls
 MiniNet.jar             People.txt
 MiniNet_lib             Relationship.txt
[tomz123s-MacBook-Pro:Release tomz123$ java -jar MiniNet.jar
```

Github Links:

https://github.com/rmit-s3666814-yujue-zou/Assignment2

Java Doc: