Design Document

Advanced Programming (004316)

s3622026 BOCHUAN ZHU   60%

s3643477 XU ZHE   40%

Contents

Introduction:

There are two different patterns when using this app, GUI and command line. When click the "Run" button in Eclipse, the GUI will display, after close the GUI window, the command line pattern will be operated. (You need close the open window rather than click "Exit" button on GUI window).

In general, this app consists of 6 main classes, includes "GUI", "MiniNet", "Driver", "InformationDB", "Account", "Run". Besides, there are 1 interface which is "Method", and 8 Exception class.

This app designed by using Eclipse as IDE, and run on the Windows environment. We also use JDK 8, JavaFx, HSQLDB.

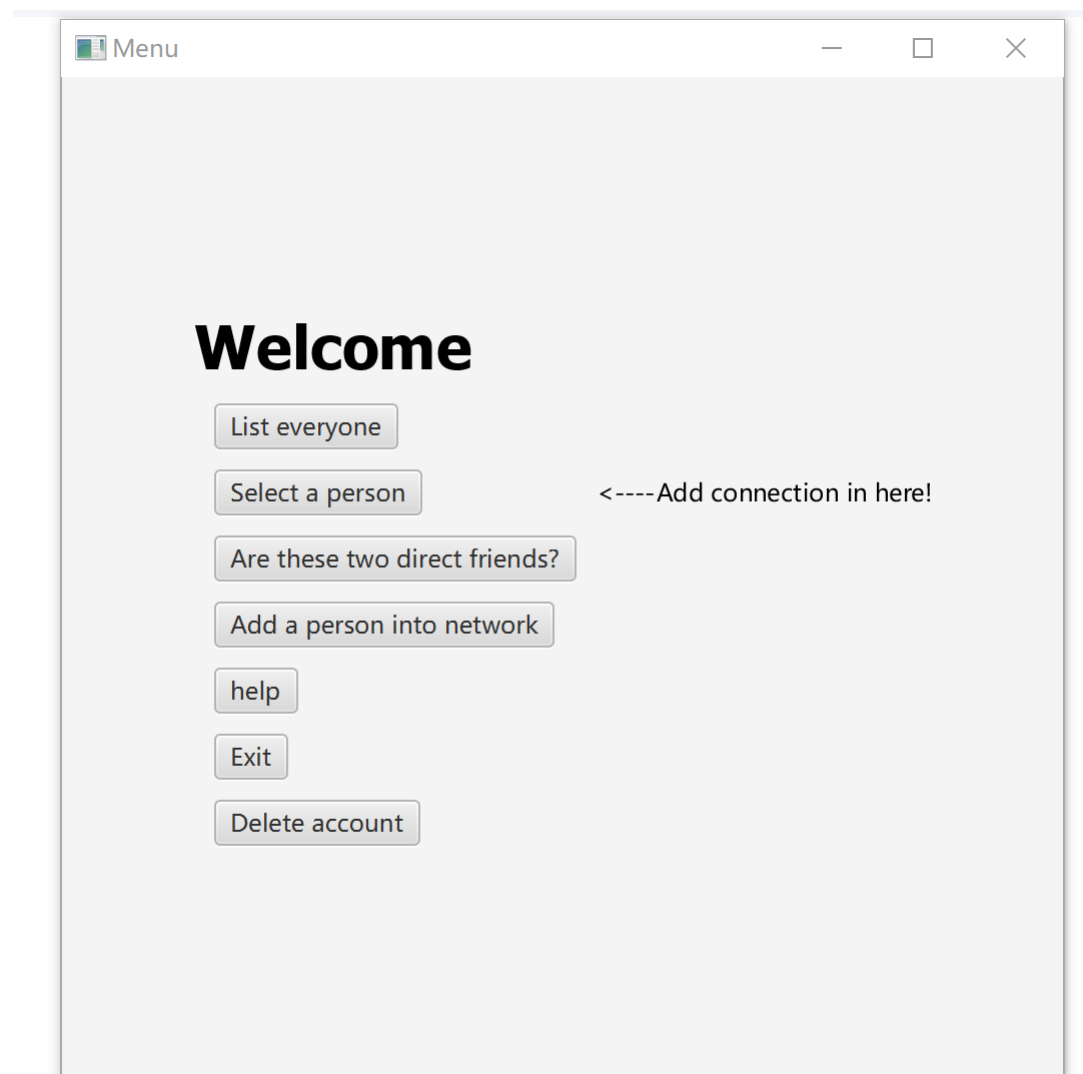(when you run this app, if there are HSQL exception as "HSQL eclipse java.net.BindException: Address already in use", which means this process is busy, please reboot your IED, then run it again)
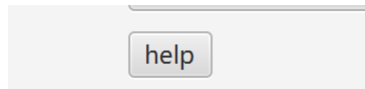
How to use?

1. click "run" button on Eclipse
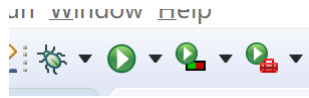
2. This is the Menu page

There are several functions, make your choice. And In the second

line there is a hint as "add connection in here!", if you want to add

friends, couple, child, parents and so on, you need click "select a person" button first, and a new window will be opened.

3. help

help

Click "help" in the menu page, you will find useful information.

you need choose an option,you can select or add person
you need create account in different name

4. Use command line rather than GUI

If you don't want to use GUI, you need close GUI window first.

— □ ×
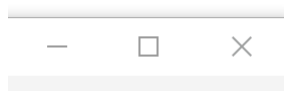
Click "X" on window rather than "Exit" button.!!

If you click "Exit", the whole app will be shut down.!!

5. Now, you can use it as command line pattern

```
===============================
MiniNet Menu

===============================

1.List everyone
2.Select a person
3.Are these two direct friends?
4.Add a person into network
5.help
6.Exit
0.Delete account


Enter an option:__
```

Scenarios:

When you run this app, the app will initial information, which means there are some people and relations in the system.

All account (person) information will be stored in the "people.txt" file, "relations.txt" file as well as database (HsqlDB), you can check it by click "search" button.

How we design this app?

The app will root from "MiniNet" class, because of main method. "GUI" class is responsible for GUI design and display, "Driver" class are mainly used for display menu as command line pattern; "Run" class can patch many methods from other class, mainly from "Account" class; "Account" class store account information, and it implements the interface of "Method". "GUI" class and "Driver" class need interact with "InformationDB" class, "MiniNet" class will link with "GUI" and "Driver" class. (more details displayed in the last page as class diagram)

Choice and Result:

1. List people

List everyone
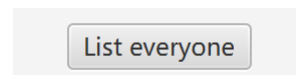
Click this button, and all of people's information will be displayed.

As:

Result	— □ ✕
All People	

Alice   Bob   Cindy   Don   [read from Memory]

```
Name: Alice        [Read from people.txt file]
Age: 40       [Read from people.txt file]
Gender: female          [Read from people.txt file]
Status: "I like dancing"           [Read from people.txt file]
Name: Bob          [Read from people.txt file]
Age: 40        [Read from people.txt file]
Gender: male        [Read from people.txt file]
Status: "I am a chief"           [Read from people txt file]
```

```
AliceBob Don [friends]      [Read from relations.txt file]
AliceBob [couple]       [Read from relations.txt file]
AliceBob [parent]       [Read from relations.txt file]
[classmates]      [Read from relations.txt file]
[colleagues]      [Read from relations.txt file]
```

```
Name: Alice Age:40      [read from database]
Name: Bob Age:40     [read from database]
Name: Cindy Age:10     [read from database]
Name: Don Age:20     [read from database]
```

There are 4 areas, the first area shows info stored in the memory;

The second and third area shows info stored in the file.
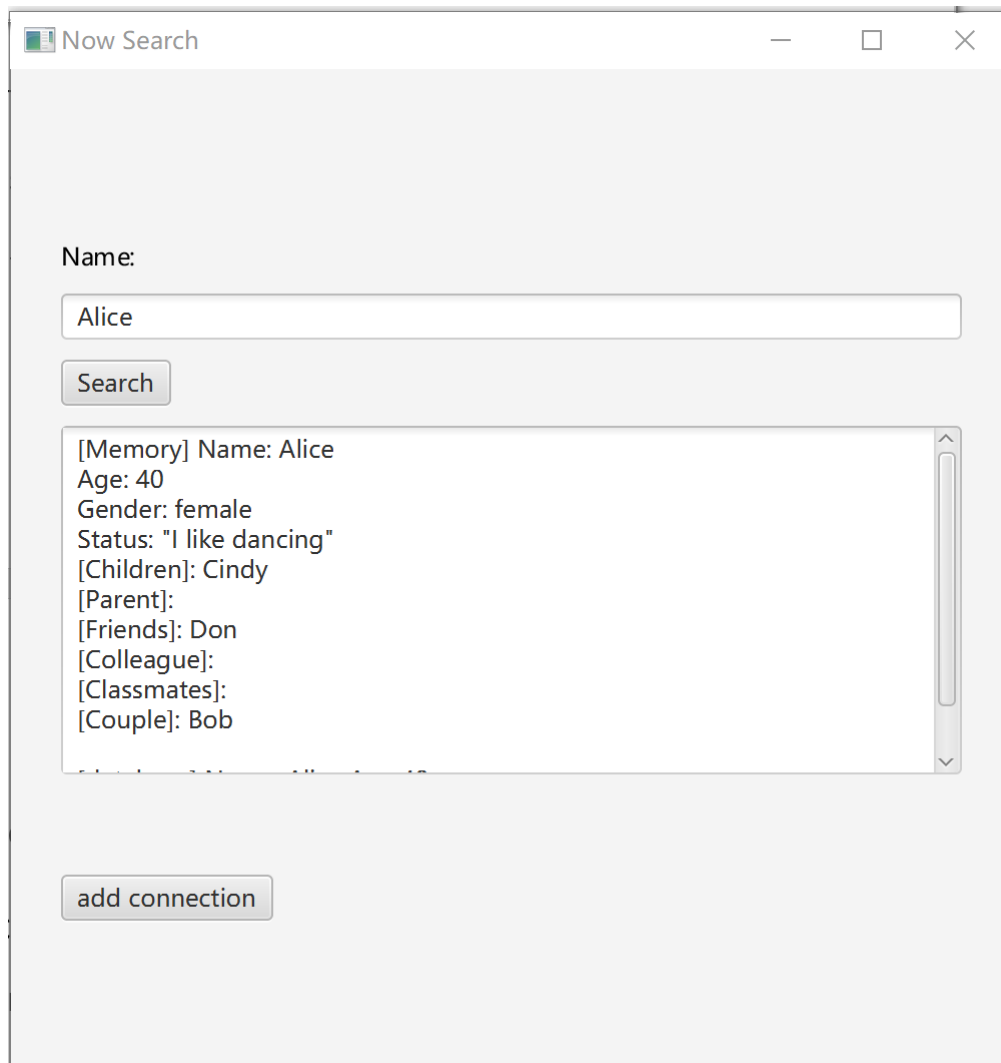
The forth area shows info searched from database.

2. Select a person

Click this button   Select a person

Enter the name you want to search, this is the result;

▣ Now Search   —   □   ✕

Name:

Alice

Search

[Memory] Name: Alice
Age: 40
Gender: female
Status: "I like dancing"
[Children]: Cindy
[Parent]:
[Friends]: Don
[Colleague]:
[Classmates]:
[Couple]: Bob

add connection

3. Add connection

Click "connection" button, a new window will be opened.

add friends

add child

add parents

add colleague

add classmates

add couple

Make your choice.

Here is an example as "add friends".

Click "add friends".

your name

friend's name

add

A new window opened. Fill the name and click "add".

your name

Alice

friend's name

Bob

add

your name

Alice

friend's name

Bob

Finished!!

Look, the Button change to "Finished".

You have finished.

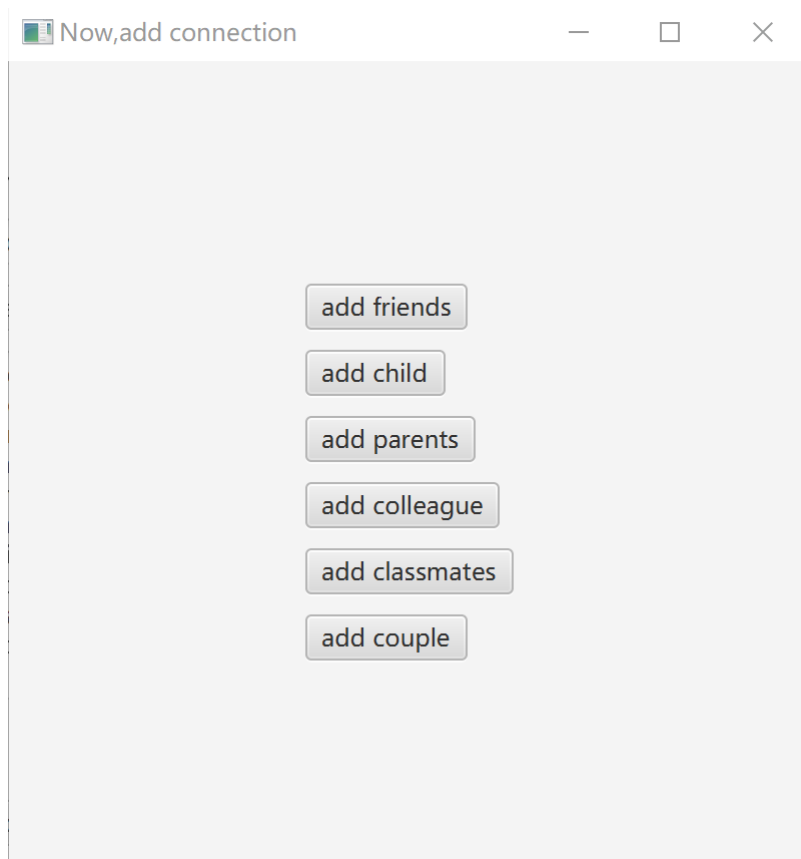Let's click "search" button to check Alice. Well done.

Name:
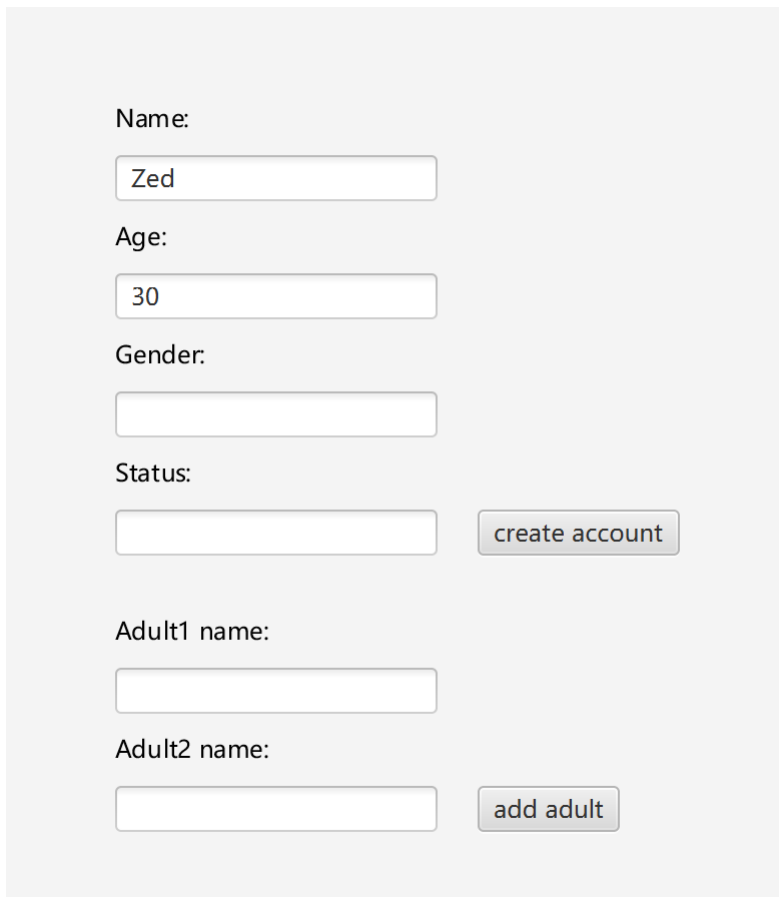
Alice

Search

[Memory] Name: Alice
Age: 40
Gender: female
Status: "I like dancing"
[Children]: Cindy
[Parent]:
[Friends]: Don Bob
[Colleague]:
[Classmates]:
[Couple]: Bob

add connection

4. Add a person (account)

Click "add a person into network" on the main page.

After that, a new window opened.

Name:

Zed

Age:

30

Gender:

Status:

create account

Adult1 name:

Adult2 name:

add adult

Fill the text bar, gender, status can be empty. And click create

account. The system will judge your age, if you are too young,

there is a notification, in this case, you have to add two adults.

Name:

Zed

Age:

3

Gende

Carefully   ✕

Carefully   ⓘ

Status:

your age is under the 16 years old
your must linked with two adults

确定

Adult1 name:

Adult2 name:

add adult

And if you enter the name which is invalid, the Error window will

be opened.

Name:

error   ✕
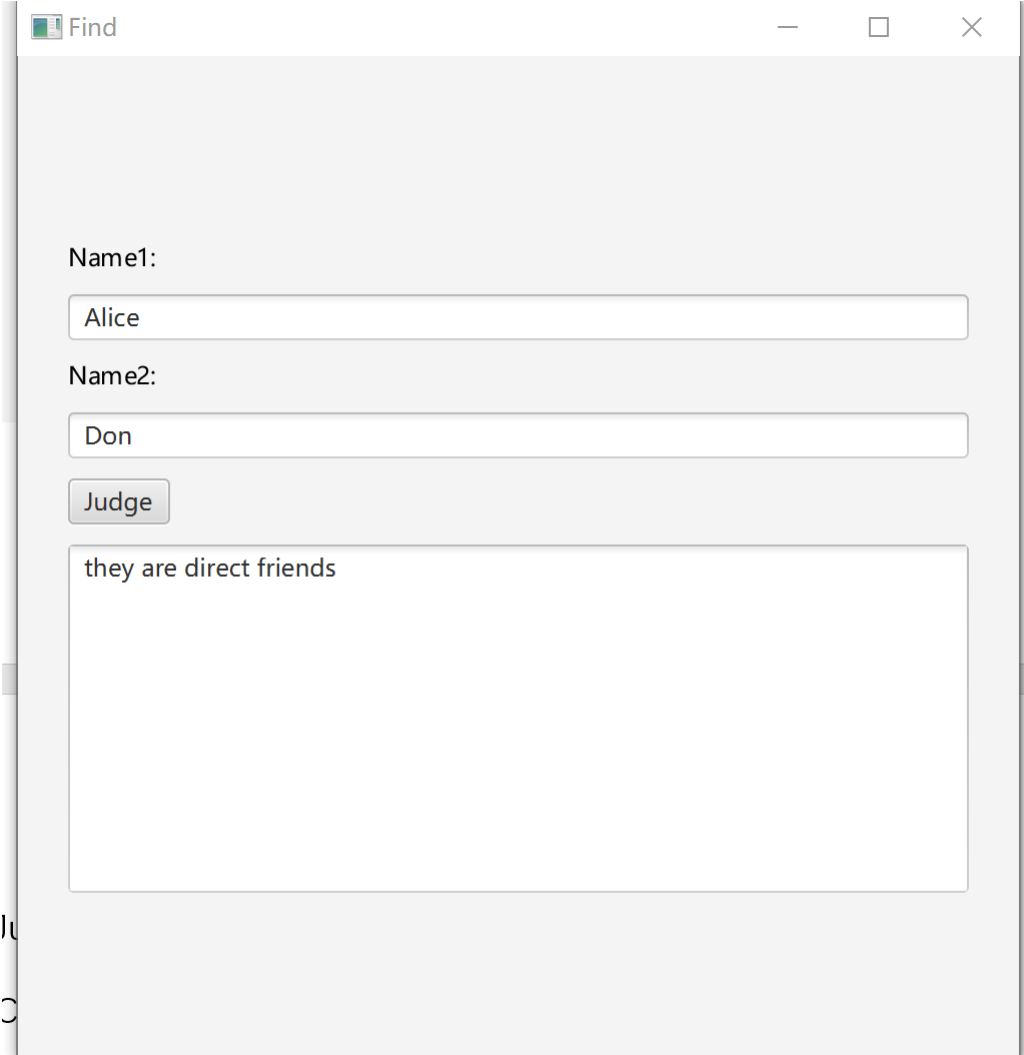
NoSuchAgeException   ✕

age need bigger than 0 and smaller than 150!

确定

Status:

create account

Adult1 name:

Adult2 name:

add adult

5. Judge friends

Click "Are these two direct friends" from main page.

One more thing, if your operation is invalid or don't meet requirement, the exception will display, sometimes you need reboot this app when the error is serious.

Exception:

```
NoAvaliableException  ───▷  Exception  ◁───  NoParentException
```

```
         △
         │
  ┌──────┼──────┬──────────┬──────────┬──────────┐
```

| NosuchAgeException | NotToBeClassmatesException | NotToBeColleaguesException | NotToBeCoupledException | NotToBeFriendsException | TooYoungException |

```
Application
    △
    │
```

| GUI | | MiniNet | | Driver |

```
Driver
─────────────────
+void Menu()
+void choice()
+void choice2()
+void addFriends(String, String)
+void addCouple(String, String)
```

+void choice()
+void choice2()
+void addFriends(String, String)
+void addCouple(String,String)
+void addParents(String,String,String)
+void addAge(String)
+void addChildren()
+void judgeFriends(String,String)
+void addColleague(String,String)
+void addClassmates(String,String)

+void start(Stage)
+void handle(ActionEvent)

+void main(String[] args)

Account

-String name
-int age
-String gender
-String status=null
-String[] couple=new String[1]
-Stirng []friends=new Stirng[100]
-Stirng []parents=new String[100]
-String[]children=new String[100]
-String[]colleague=new String[100]
-String[]classmates=new String[100]

+String [] getFriends(): return String
+void setFriends(String)
+void setParents(String)
+void setChildren(String)
+void setColleague(String)
+void setClassmats(String)
+String listParents(): return String
+String listChildren(): return String
+String listColleagues(): reutrn String
+String listClassmates(): return String
+String listFriends(): return String
+void setName(String)
+String getName(): return String
+void setAge(int)
+void setCouple(String)
+String listCouple():return String
+int getAge(): reutn int
+String getinfo(): return String
+void setStatus(String)
+String getStatus():return String
+void setGender(String)
+String getGender():return String

InformationDB

+void createDatabase()
+void insertDatabase()
+String listDatabase()
+String searchDatabase(String)

Run

+void file()
+void output(Sring,String)
+void changeLine(String)
+void input(String)
+String input2(Stirng): return String
+void outputRelations(String)
+void initialAccount()
+Account getAccByName(Stirng): return Accunt
+String listArray(): return String
+boolean createAccount(String): return boolean
+Account search(String): return null
+void addAge(String)
+void addFriends(String,String)
+void addColleague(String,String)
+void addClassmates(String,String)
+void addCouple(String,String)

Method

+void setName(String)
+Stirng getName()
+void setAge(int)
+int getAge()
+void setStatus(String)
+String getStatus()
+void setGender(String)
+String getGender()