Assignment 3: Implementing a 3 Tier Object Oriented System

Due: Friday 28nd November

Rory Nee – A00190040

Project concept

I choose to generate the data set around bank customers and account as I felt it lent itself to C.R.U.D.L. operations.

Create – Make a new account
Read – Search, Read transactions, read the statistics
Update – Update account information, deposit money and withdraw money
Delete – Delete an account
List – Show all accounts

I created a bank account system that can be used by bank teller to Create, Read, Update, Delete and List the accounts. As part of the bank accounts I also wanted to keep track of a customer's transactions (withdraw and deposit) just like a real bank would do.

Also I wanted to be able to generate dynamic statistics in relation to that data so I could generate 3d bar charts, line graph and other data in relation the account held in a bank.

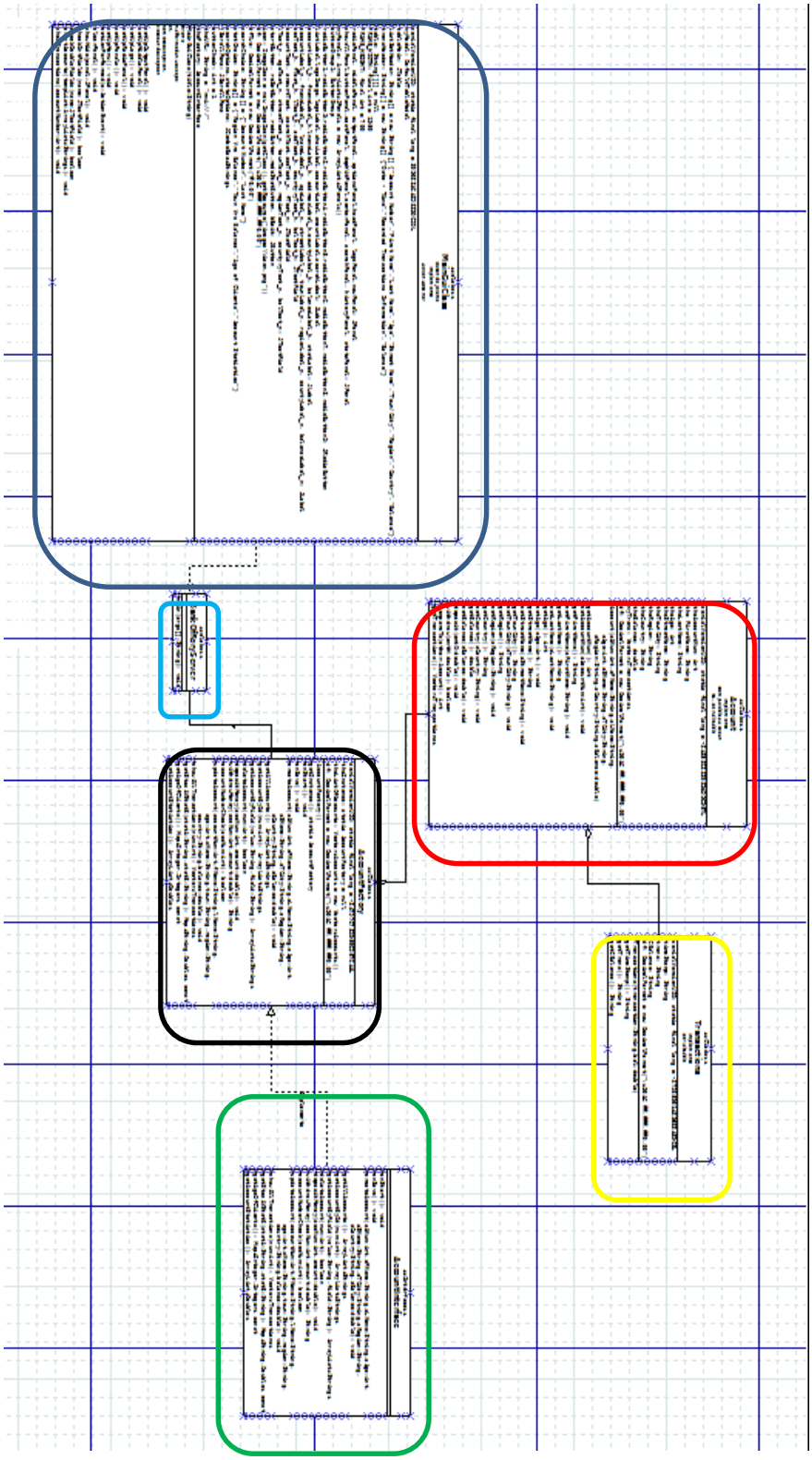
System Architecture

I set up the project as described in the "Implementing a 3 Tier Object Oriented System" outline. My 'AccountFactory' (in black) class implements the 'AccountInterface' (green) class. The 'Account' class is aggregated to the 'AccountFactory' class and the 'transaction' class is aggregated to the 'Account' class. The Account factory through the Account interface uses RMI's mechanism of the RMI studs and the RMI registry to connect with the server. The server in turn can be connected by the 'MainGuiClass' and using an Identification (URL) to find the target machine where the RMI registry and remote objects are located. Then the 'MainGuiClass' requests the RMI registry on the target machine to return an object reference that applies to the well-known name or password. Then using the 'AccountInterface' it can pass the objects back and forth to each other.

Due to the size of the class diagrams I have choose to include the diagram in the upload as some of the screen shots are both hard to see and hard to read.

Main Class Structure

Note: There are more comments on the class available by viewing the Class Diagram through Dia e.g. implements, extend and throws. It would have been too big to include all of them.

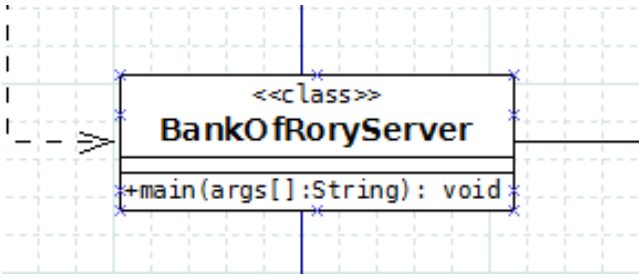


Red – Account.Java **Yellow** – Transactions **Green** – AccountInterface	**Black** – AccountFactory **Light Blue** – BankOfRoryServer **Blue** - MainGuiClass

Individual Screen Shots shown below

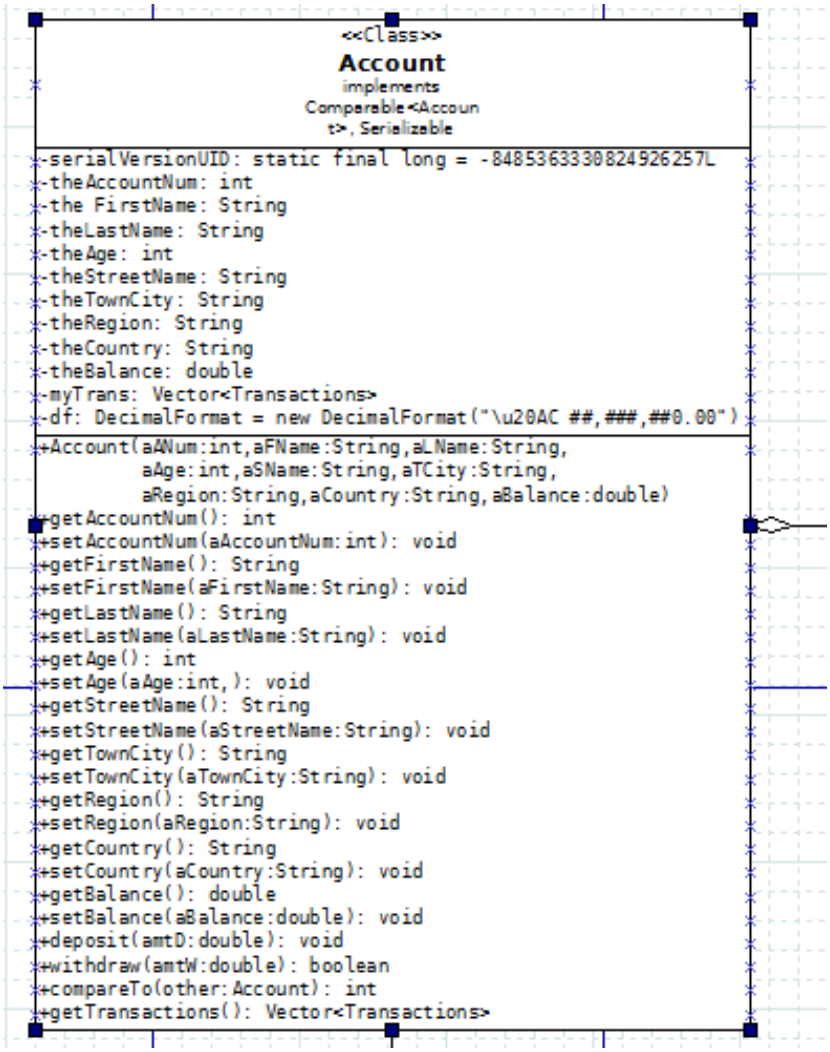
Light Blue – BankOfRoryServer

The Bank Of Rory Server binds the well-known (“BankOfRoryServer”) name to the object of the account factory and stores that in the Rmi registry.



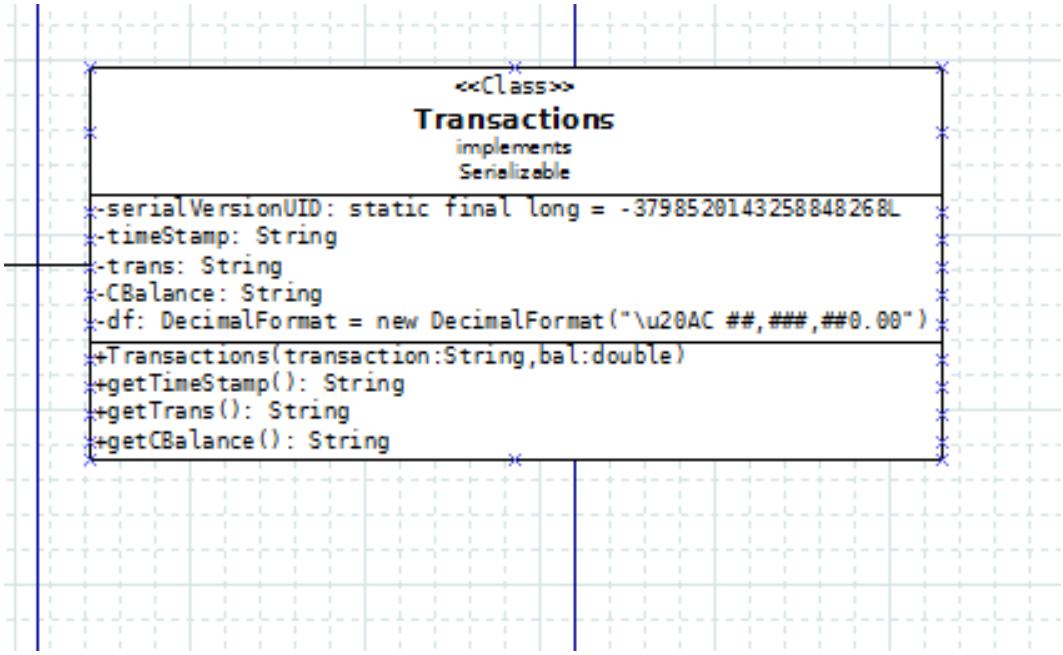
Red – Account.Java

The Account class outlines all the information involded in a bank account. The Account class also uses a vector of type ‘Transactions’ to hold all the transactions that are made by an account. The types of transactions that are take note of is when the account was created and the deposit and withdraw money. I have also implemented the comparable interface so I will be able to compare one account to another account using the account number. This is used when sorting the acount number in the table.



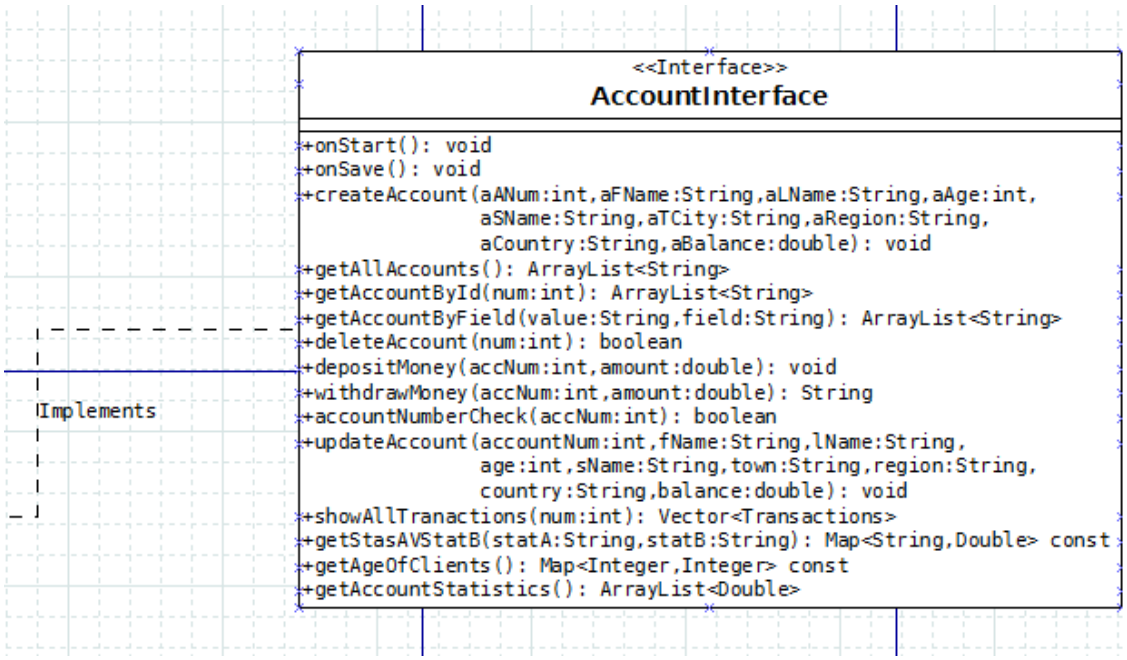
Yellow – Transactions

The Transaction class outlines the information that is used to make up a transaction. Here the time stamp is generated using a simple data format and the balance is also noted at the time of the transaction been made. There is no setter methods here are I did not want somebody to be able to change the transactions once they were made.



Green – AccountInterface

The account interface acts as an interface for the account factory. The RMI stubs are created using the interface.
Note: There are more comments on the class available by viewing the Class Diagram through Dia e.g. implements, extend and throws. It would have been too big to include all of them.



Black – AccountFactory

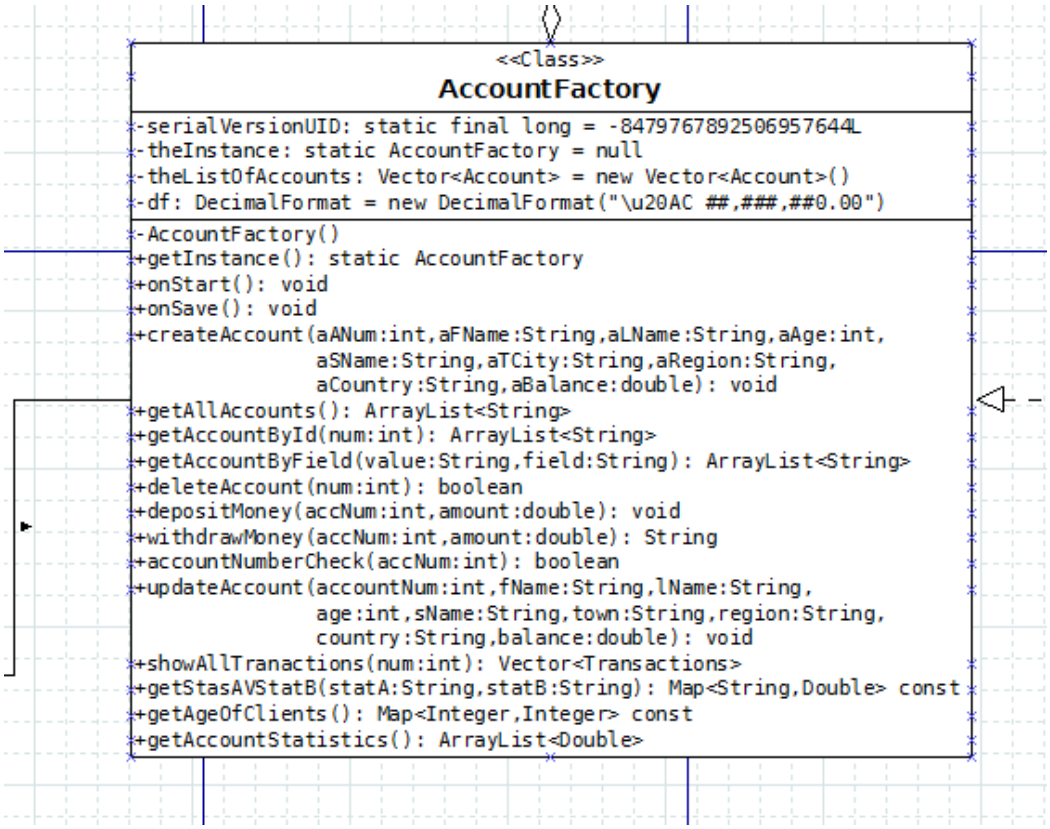
The account factory holds all the methods for the CRUDL actions.

- **C**reate – createAccount()
- **R**ead – getAccountById(), getAccountByField(), showALLTranactions(), getAgeOfClients, getStasAVStatB(), getAccountStatistics()
- **U**pdate – updateAccount(),depositMoney(), withdrawMoney()
- **D**elete – deleteAccount()
- **L**ist – getAllAccounts()

The method called getInstance() makes only one instantiation of the AccountFactory and give back a reference to the object of the account factory if another class requests to make another object of the account factory.

There is also a method to check weather a account number Is in the system. I also have a onStart() and an onSave() method to serialize and desterilize the data to file.

Note: There are more comments on the class available by viewing the Class Diagram through Dia e.g. implements, extend and throws. It would have been too big to include all of them.



Blue – MainGuiClass

Attributes

As you can see below there is a lot of attributes declared in the MinGuiClass. In the code I have reused ‘private JLabel’ a few times as there is a lot of label and did not want to have them untidy.

A few thing to note, I have created an ArrayList of Jpanels to be used by a method to hide all panels and to make visible the panel passed to it (showHide() method).

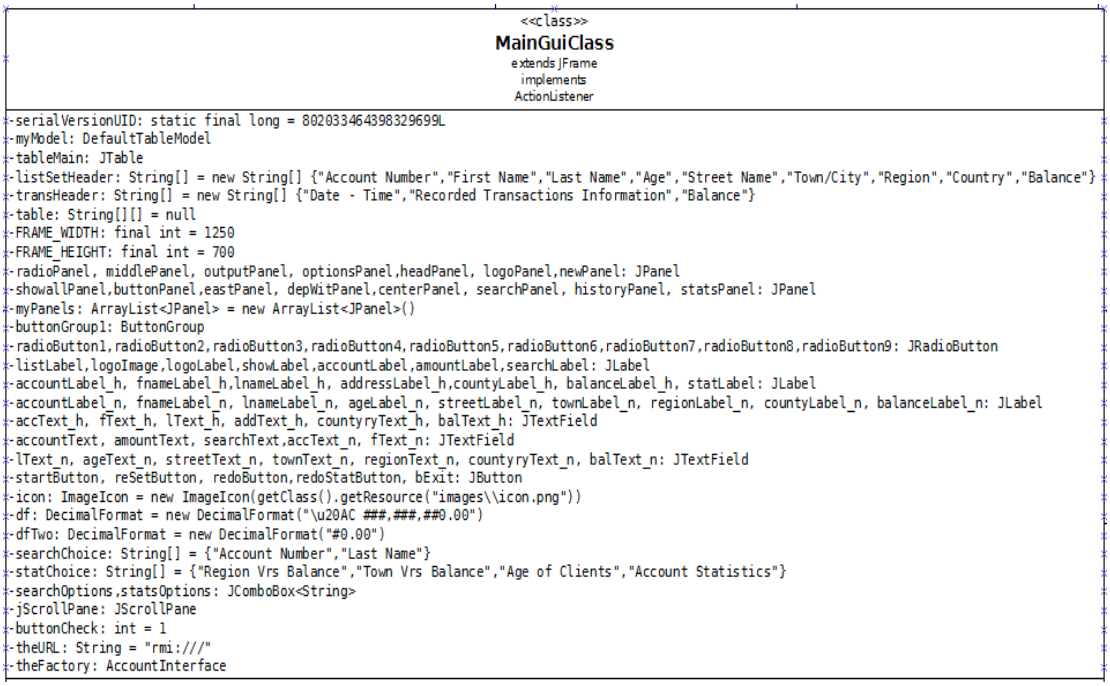
The table was set up using the defaultTableModel with a 2d array called ‘table’ and two arrays for the heading called ‘listSetHeader’ and ‘transHeader’.

There are two separate decimal format mentioned here to accommodate the information in the table and for the input of new values (new account, update account).

The variable called ‘buttonCheck’ is used to see what radiobutton is checked so I can clear it.

Also here you see the inclusion of theURL and the AccountInterface used by RMI. The Accouny Factory will be referred to as ‘theFactory’ when it is been called in the GUI.

Note: There are more comments on the class available by viewing the Class Diagram through Dia e.g. implements, extend and throws. It would have been too big to include all of them.



Methods

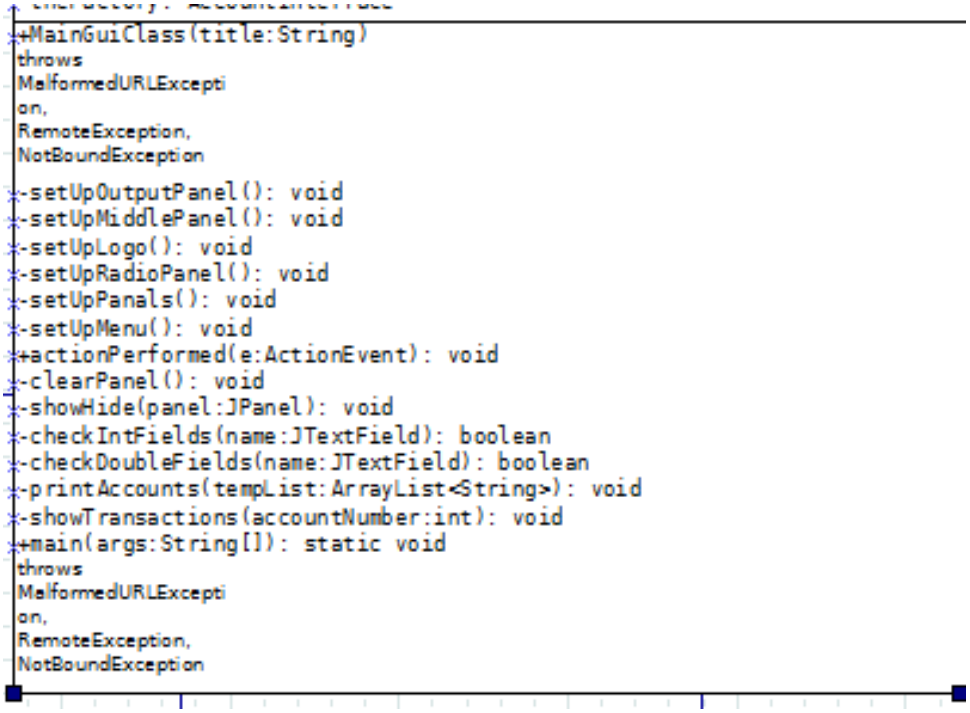
I have employed the use of multiple helper methods here so as not to clutter up the code.

```
setUpMenu();  // set up the drop down menu
setUpPanals();// set up the panels to be used
setUpLogo();   // Set up to Logo and the header label
setUpMiddlePanel();   // set up the middle panel
setUpRadioPanel();     // set up the panel with the radio buttons
setUpOutputPanel();   // set up the main display output area
```

The actionPerformed method handles all the events form the user.

There other methods of note hare are the showHide() (I mentioned earlier) and the checkIntFields() and CheckDoubleFields(). The latter two function test weather a number is a number and not text.

I also have two different function that print the data to the table because I show the data different in the show history page. (printAccounts() and showTransactions()).



Screen Shots

Below are the screen shots from my project running.

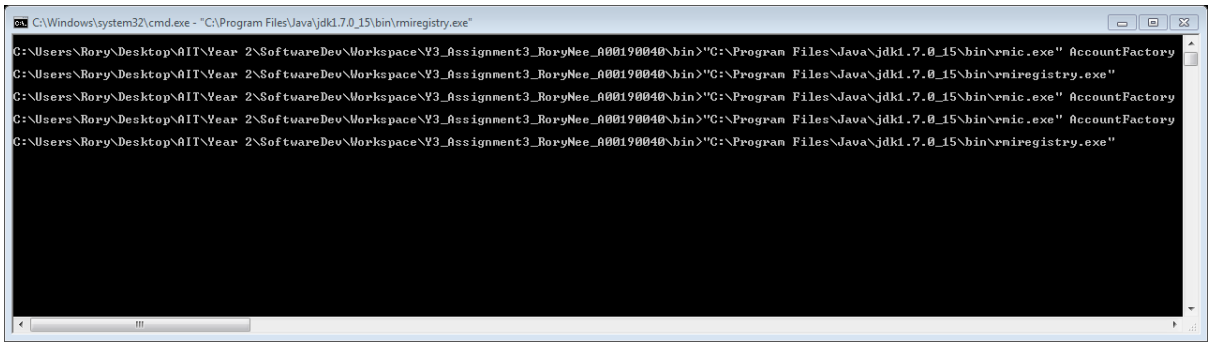


Fig 1 – rmic.exe running and the rmiregistry.exe running

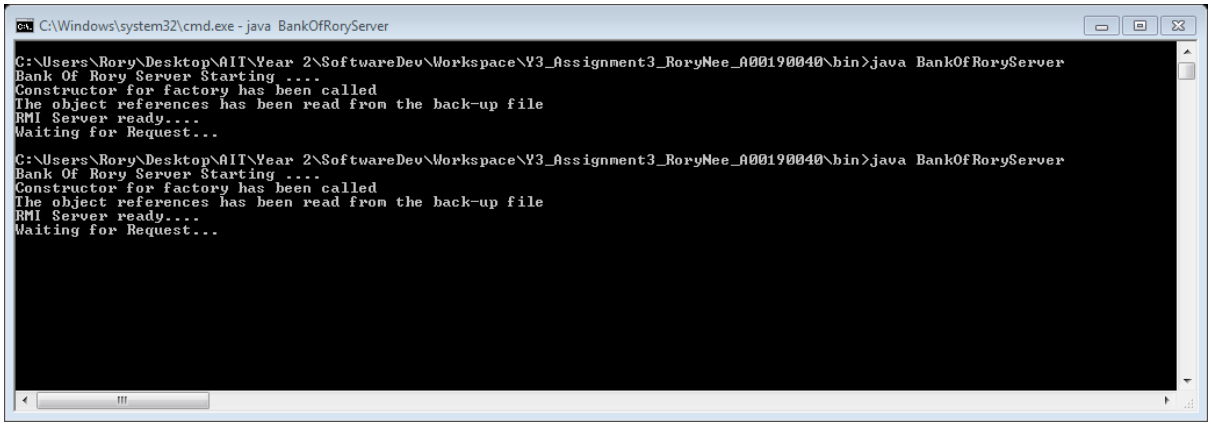


Fig 2 – The BankOfRoryServer running

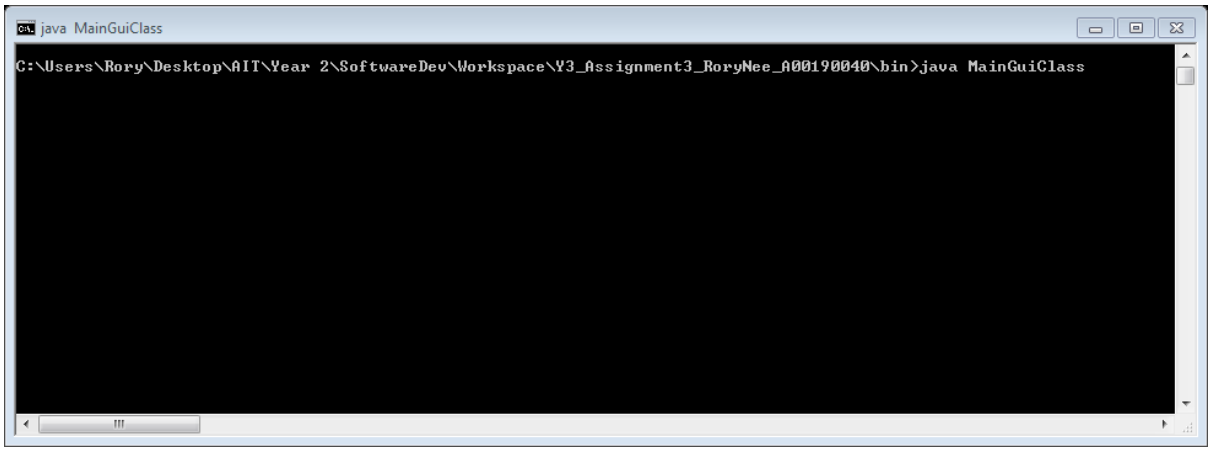
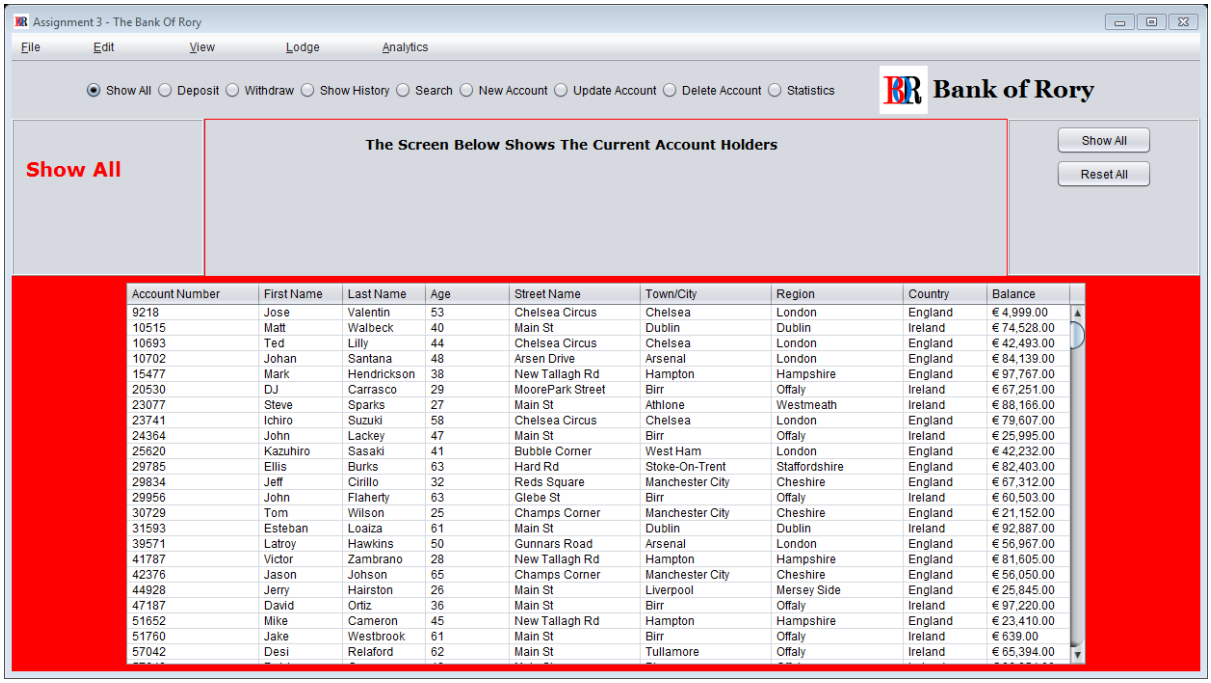
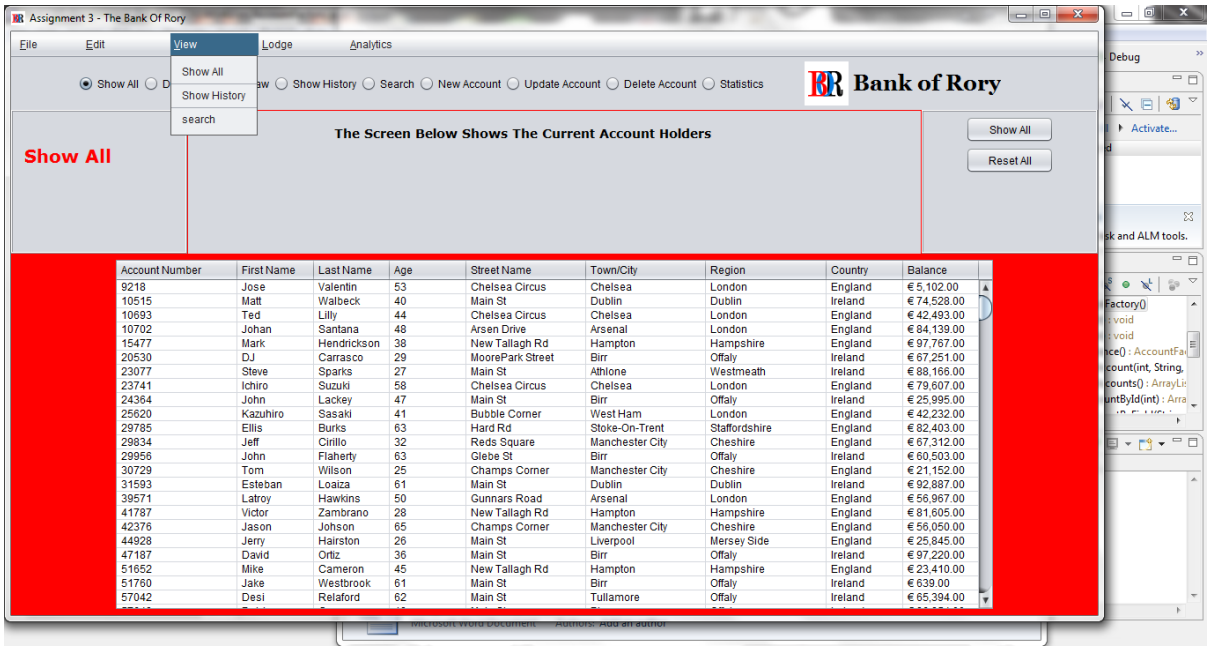


Fig 3 – The MainGuiClass Running



Account Number	First Name	Last Name	Age	Street Name	Town/City	Region	Country	Balance
9218	Jose	Valentin	53	Chelsea Circus	Chelsea	London	England	€ 4,999.00
10515	Matt	Walbeck	40	Main St	Dublin	Dublin	Ireland	€ 74,528.00
10693	Ted	Lilly	44	Chelsea Circus	Chelsea	London	England	€ 42,493.00
10702	Johan	Santana	48	Arsen Drive	Arsenal	London	England	€ 84,139.00
15477	Mark	Hendrickson	38	New Tallagh Rd	Hampton	Hampshire	England	€ 97,767.00
20530	DJ	Carrasco	29	MoorePark Street	Birr	Offaly	Ireland	€ 67,251.00
23077	Steve	Sparks	27	Main St	Athlone	Westmeath	Ireland	€ 88,166.00
23741	Ichiro	Suzuki	58	Chelsea Circus	Chelsea	London	England	€ 79,607.00
24364	John	Lackey	47	Main St	Birr	Offaly	Ireland	€ 25,995.00
25620	Kazuhiro	Sasaki	41	Bubble Corner	West Ham	London	England	€ 42,232.00
29785	Ellis	Burks	63	Hard Rd	Stoke-On-Trent	Staffordshire	England	€ 82,403.00
29834	Jeff	Cirillo	32	Reds Square	Manchester City	Cheshire	England	€ 67,312.00
29956	John	Flaherty	63	Glebe St	Birr	Offaly	Ireland	€ 60,503.00
30729	Tom	Wilson	25	Champs Corner	Manchester City	Cheshire	England	€ 21,152.00
31593	Esteban	Loaiza	61	Main St	Dublin	Dublin	Ireland	€ 92,887.00
39571	Latroy	Hawkins	50	Gunnars Road	Arsenal	London	England	€ 56,967.00
41787	Victor	Zambrano	28	New Tallagh Rd	Hampton	Hampshire	England	€ 81,605.00
42376	Jason	Johson	65	Champs Corner	Manchester City	Cheshire	England	€ 56,050.00
44928	Jerry	Hairston	26	Main St	Liverpool	Mersey Side	England	€ 25,845.00
47187	David	Ortiz	36	Main St	Birr	Offaly	Ireland	€ 97,220.00
51652	Mike	Cameron	45	New Tallagh Rd	Hampton	Hampshire	England	€ 23,410.00
51760	Jake	Westbrook	61	Main St	Birr	Offaly	Ireland	€ 639.00
57042	Desi	Relaford	62	Main St	Tullamore	Offaly	Ireland	€ 65,394.00

Fig 4 – The Show All screen showing all the accounts. Here we can also see the change to the icon and the inclusion of the icon and the label header. Also when a new tab is selected the colouring changes and the name ‘Show All’ & ‘Show All (button text)’ changes along with it.



Account Number	First Name	Last Name	Age	Street Name	Town/City	Region	Country	Balance
9218	Jose	Valentin	53	Chelsea Circus	Chelsea	London	England	€ 5,102.00
10515	Matt	Walbeck	40	Main St	Dublin	Dublin	Ireland	€ 74,528.00
10693	Ted	Lilly	44	Chelsea Circus	Chelsea	London	England	€ 42,493.00
10702	Johan	Santana	48	Arsen Drive	Arsenal	London	England	€ 84,139.00
15477	Mark	Hendrickson	38	New Tallagh Rd	Hampton	Hampshire	England	€ 97,767.00
20530	DJ	Carrasco	29	MoorePark Street	Birr	Offaly	Ireland	€ 67,251.00
23077	Steve	Sparks	27	Main St	Athlone	Westmeath	Ireland	€ 88,166.00
23741	Ichiro	Suzuki	58	Chelsea Circus	Chelsea	London	England	€ 79,607.00
24364	John	Lackey	47	Main St	Birr	Offaly	Ireland	€ 25,995.00
25620	Kazuhiro	Sasaki	41	Bubble Corner	West Ham	London	England	€ 42,232.00
29785	Ellis	Burks	63	Hard Rd	Stoke-On-Trent	Staffordshire	England	€ 82,403.00
29834	Jeff	Cirillo	32	Reds Square	Manchester City	Cheshire	England	€ 67,312.00
29956	John	Flaherty	63	Glebe St	Birr	Offaly	Ireland	€ 60,503.00
30729	Tom	Wilson	25	Champs Corner	Manchester City	Cheshire	England	€ 21,152.00
31593	Esteban	Loaiza	61	Main St	Dublin	Dublin	Ireland	€ 92,887.00
39571	Latroy	Hawkins	50	Gunnars Road	Arsenal	London	England	€ 56,967.00
41787	Victor	Zambrano	28	New Tallagh Rd	Hampton	Hampshire	England	€ 81,605.00
42376	Jason	Johson	65	Champs Corner	Manchester City	Cheshire	England	€ 56,050.00
44928	Jerry	Hairston	26	Main St	Liverpool	Mersey Side	England	€ 25,845.00
47187	David	Ortiz	36	Main St	Birr	Offaly	Ireland	€ 97,220.00
51652	Mike	Cameron	45	New Tallagh Rd	Hampton	Hampshire	England	€ 23,410.00
51760	Jake	Westbrook	61	Main St	Birr	Offaly	Ireland	€ 639.00
57042	Desi	Relaford	62	Main St	Tullamore	Offaly	Ireland	€ 65,394.00

Fig 5 – Here is a demonstration of the drop down menu which can be activated by hitting alt and either F, E, V, L or A.

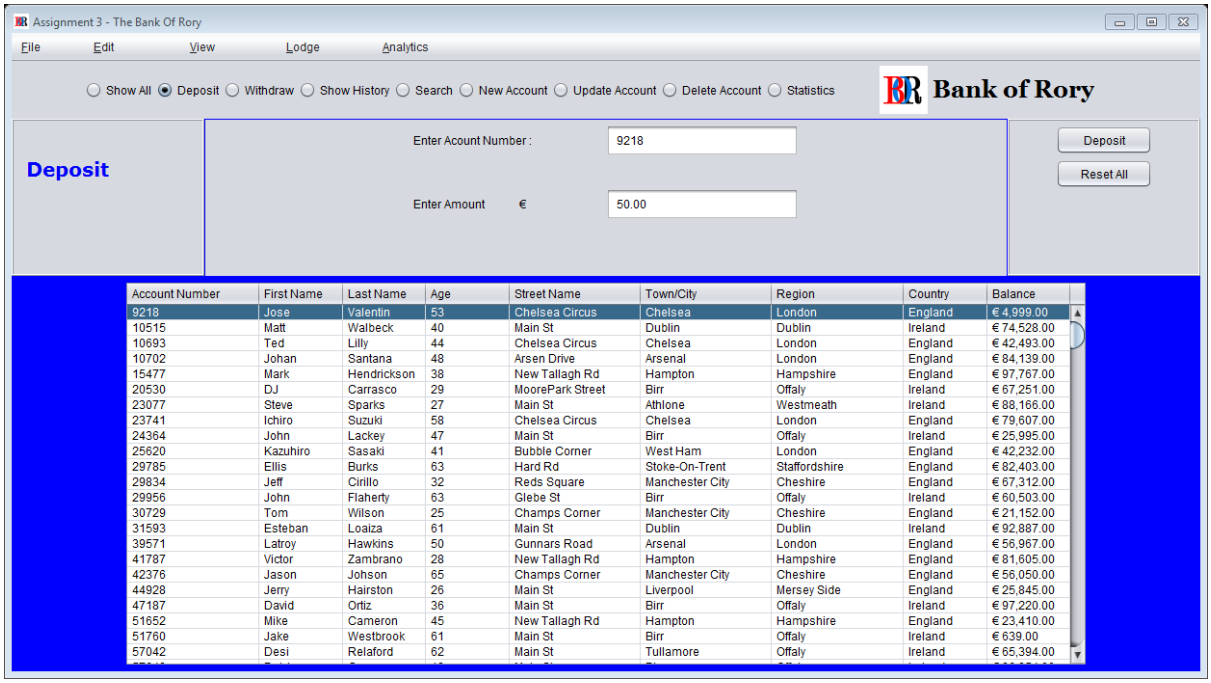


Fig 6 – This shows the deposit screen in action. You can either click on an account or enter in the number yourself that you want to make a deposit in.

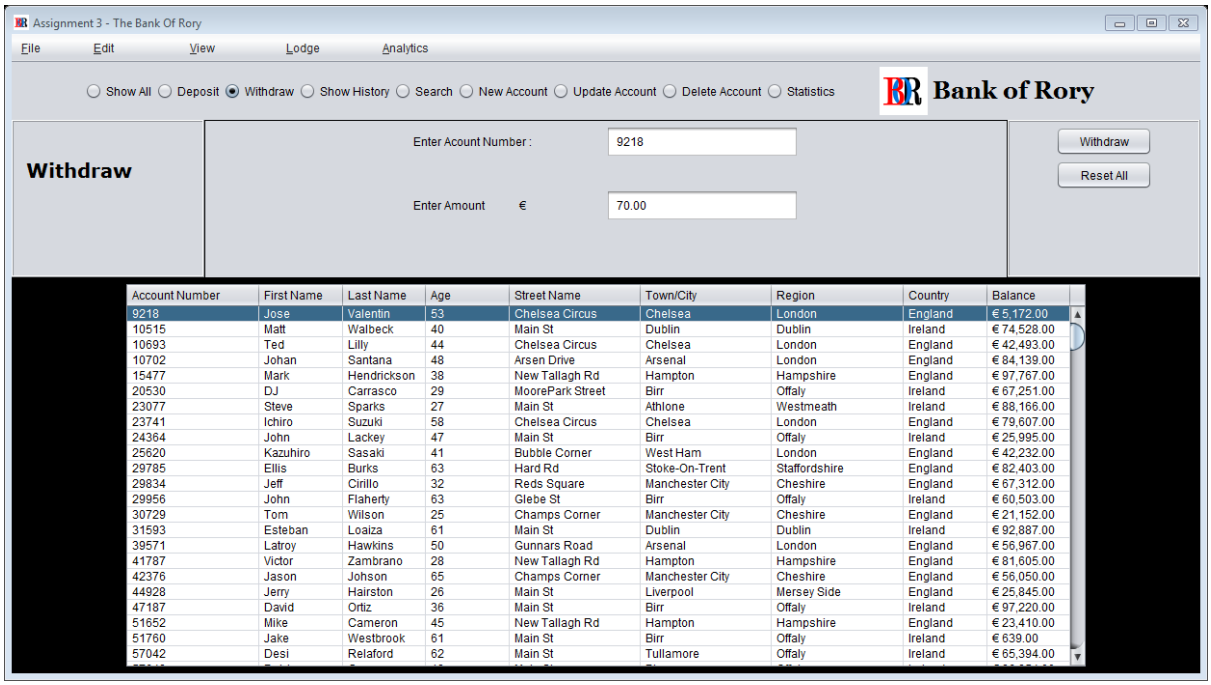


Fig 7 – This shows the withdraw screen in action. You can either click on an account or enter in the number yourself that you want to make a withdrawal out of. When you hit the withdraw button the account information and new balance show on the table below so that the user can see the change.

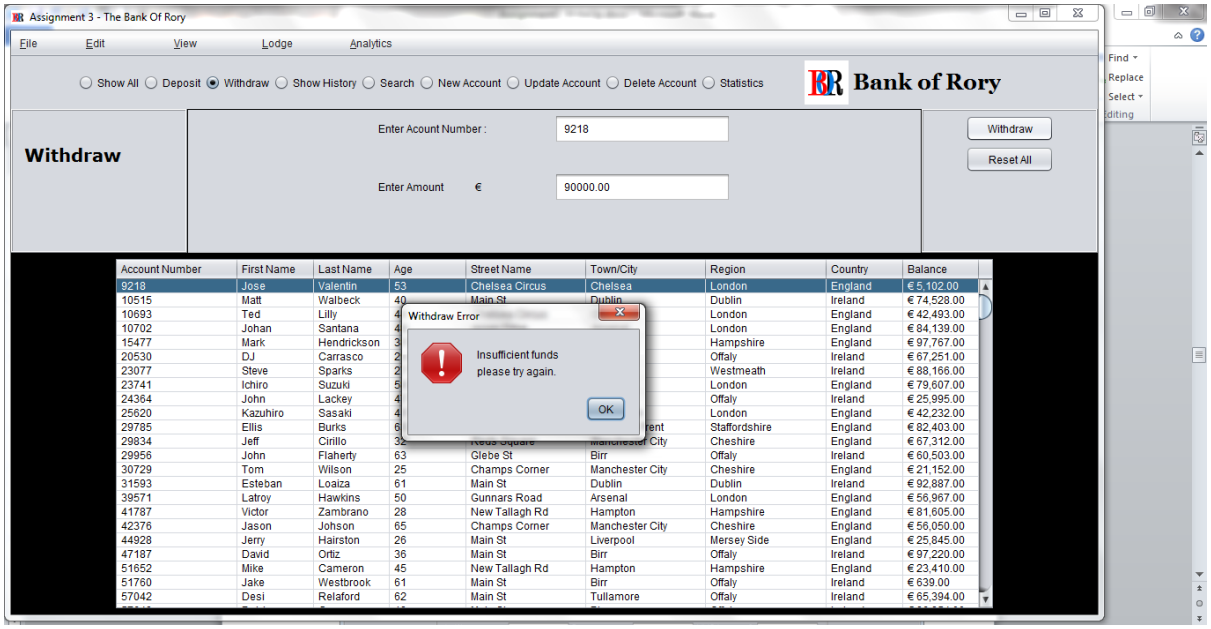


Fig 8 – There is a check to make sure that the customer cannot take out too much money. The JOptionPane box show the 'withdraw error warning'. All throughout the program there is extensive checking of number (to see if they are number) and string (for space or nothing entered).

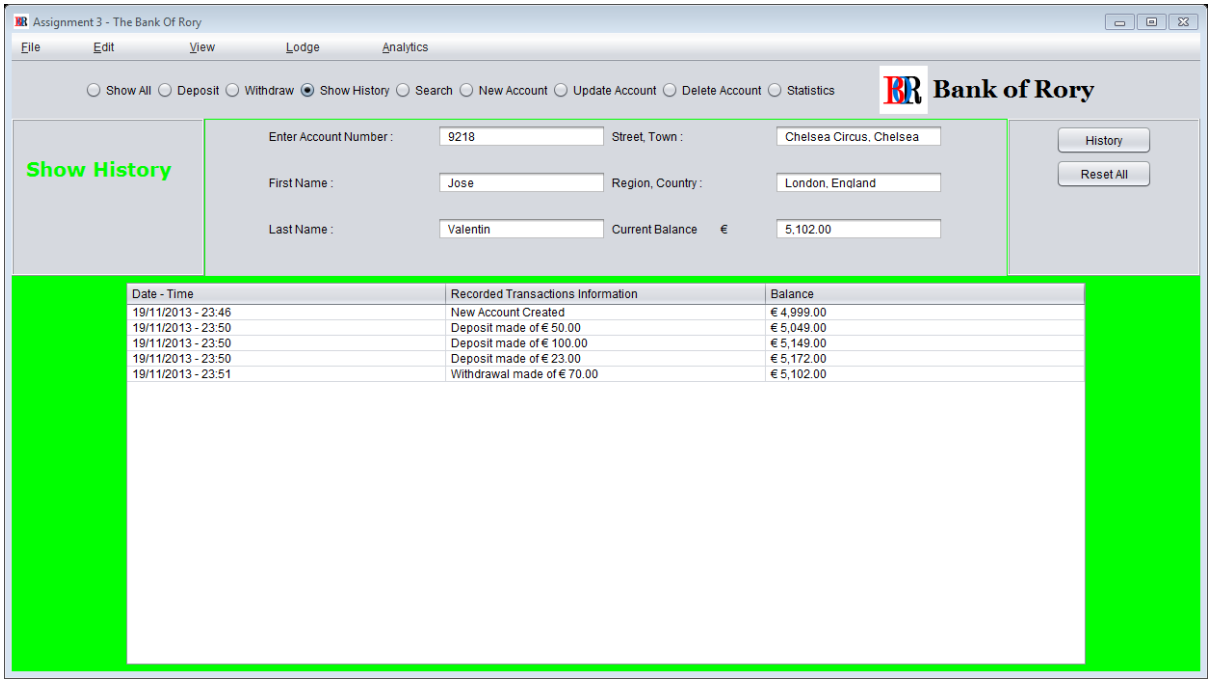


Fig 9 – The user can enter in an account number or click on the account number they would like to view the transactions of. The only text field that is editable here is the account number so the information cannot be changed or compromised.

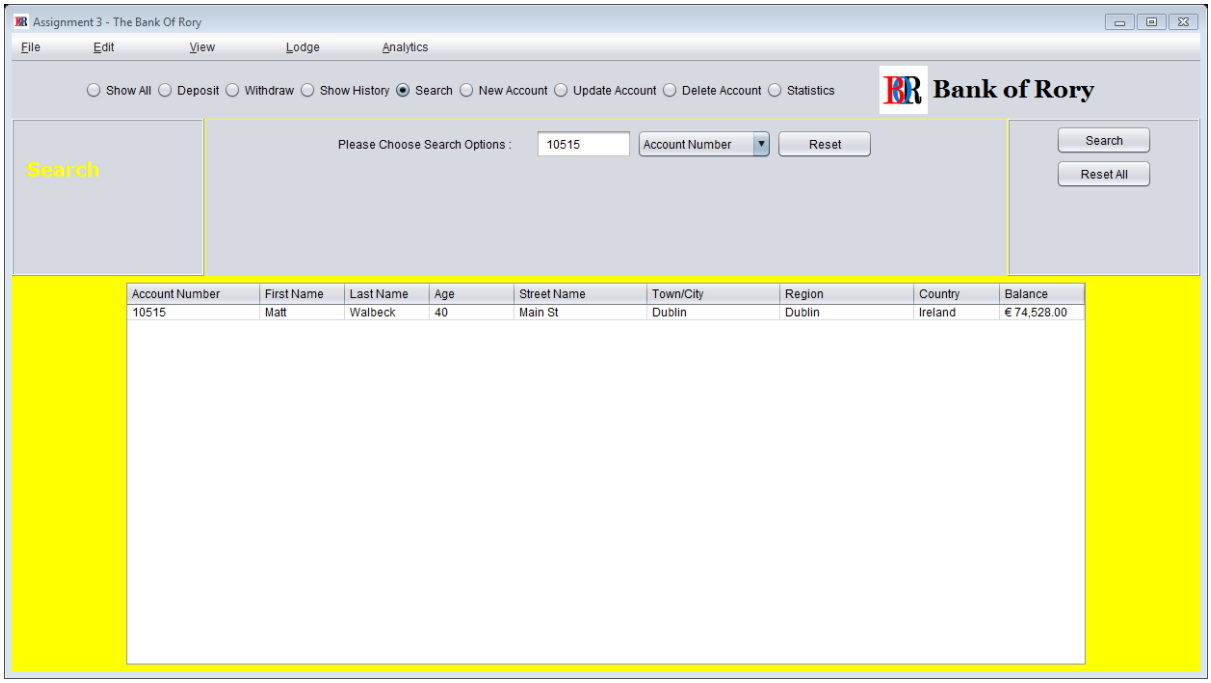


Fig 10 – The search window allows you to search by account number or by the last name as shown in Fig 11.

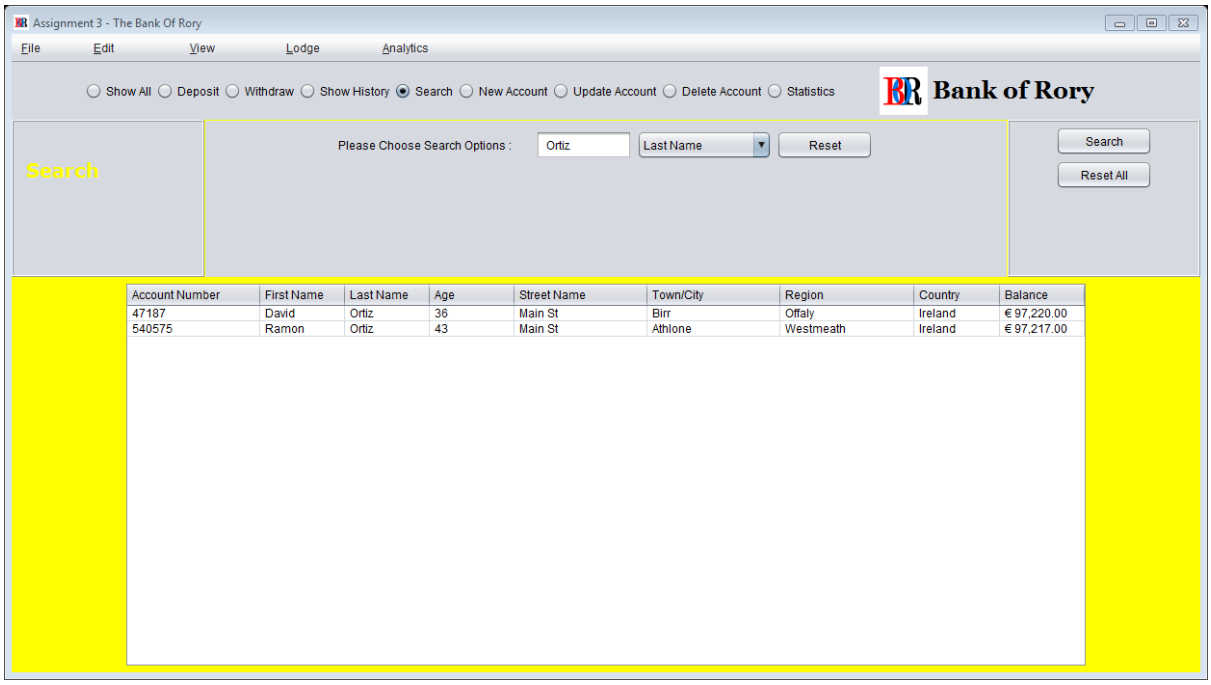


Fig 11

The Error checking for the search screen is shown in Fig 12.

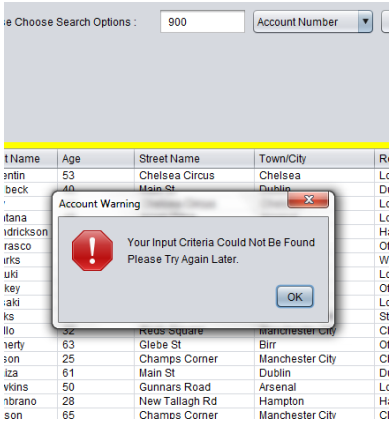


Fig 12

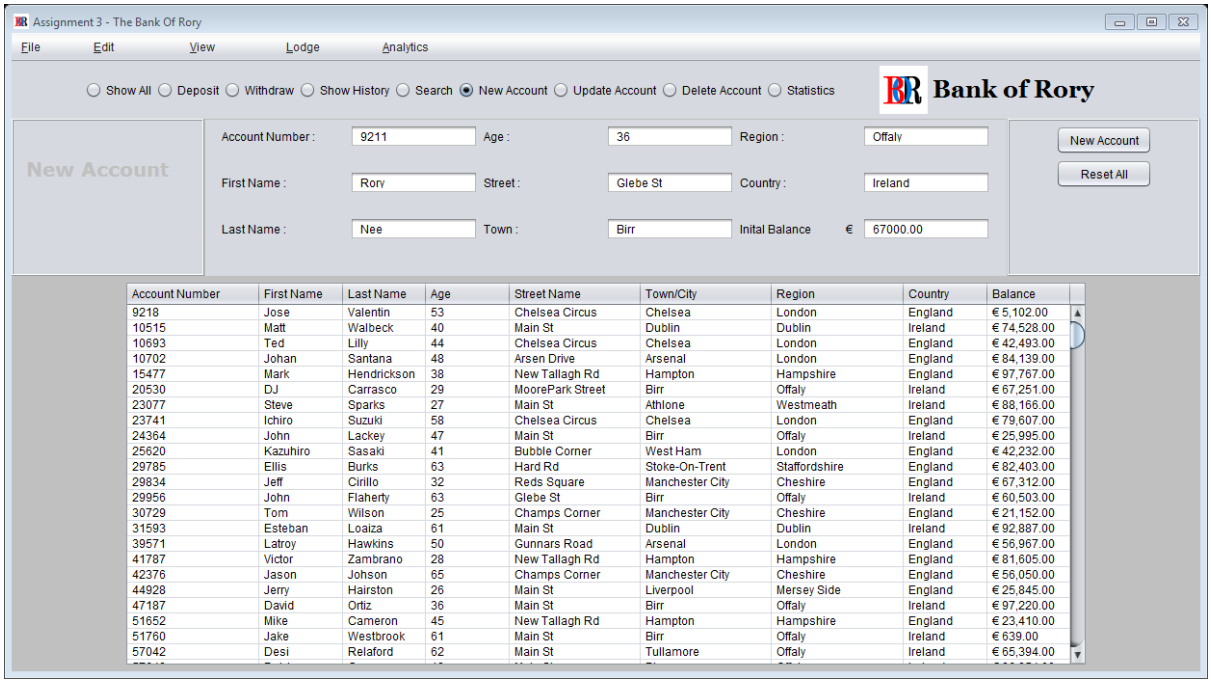


Fig 13 – This is the New Account screen where the user can make an account and when it is finished it will be shown in the table below in Fig 14. The error below in Fig 15 shows that the account number is already been used. Also if a field is incorrectly filled out you would get an error also.

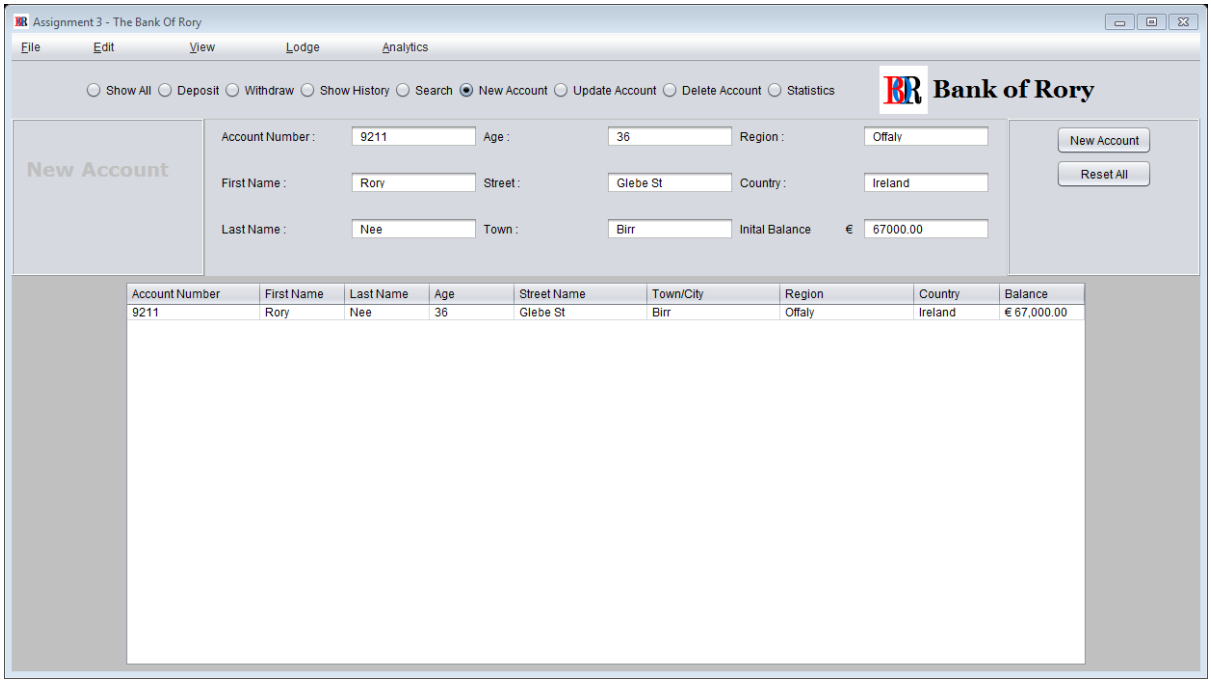


Fig 14

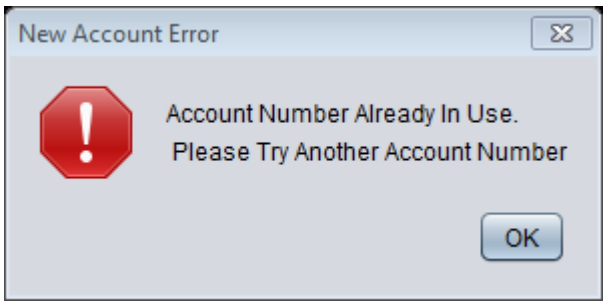


Fig 15

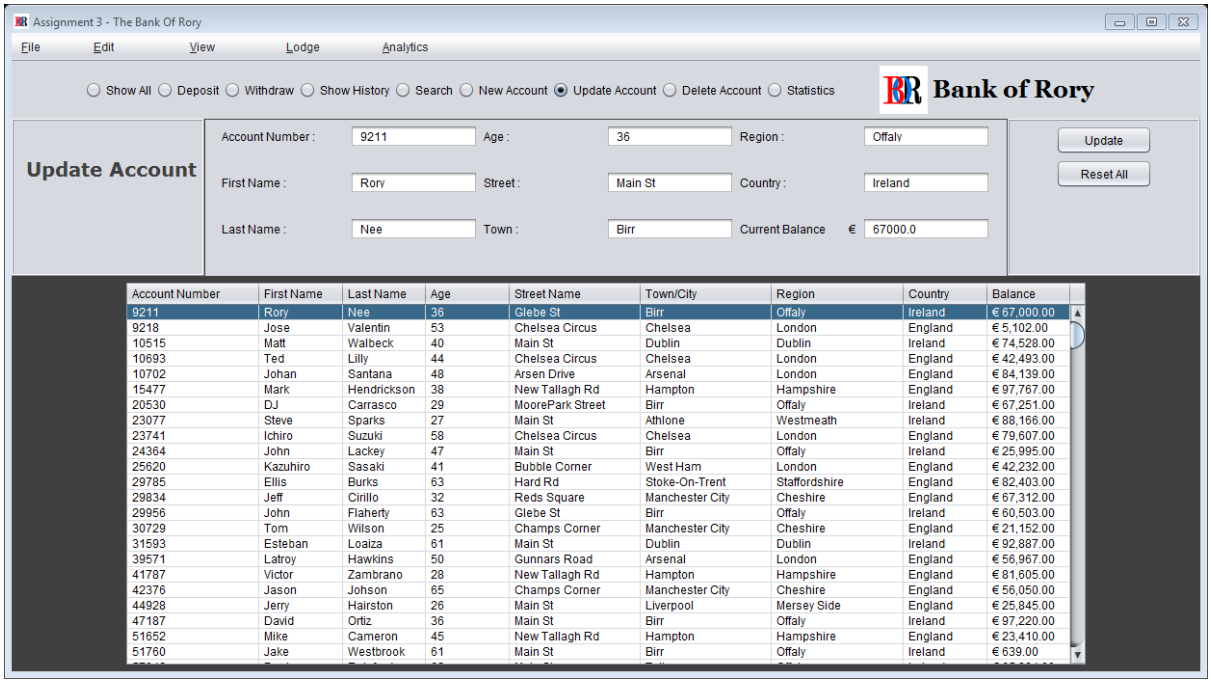


Fig 16 – The screen shot show the update account in action. Here the table is clickable for to make it easy to fill out the fields. Extensive error checking is also used here. In Fig 17 below we can see that the updated account and it information has been show back on screen,

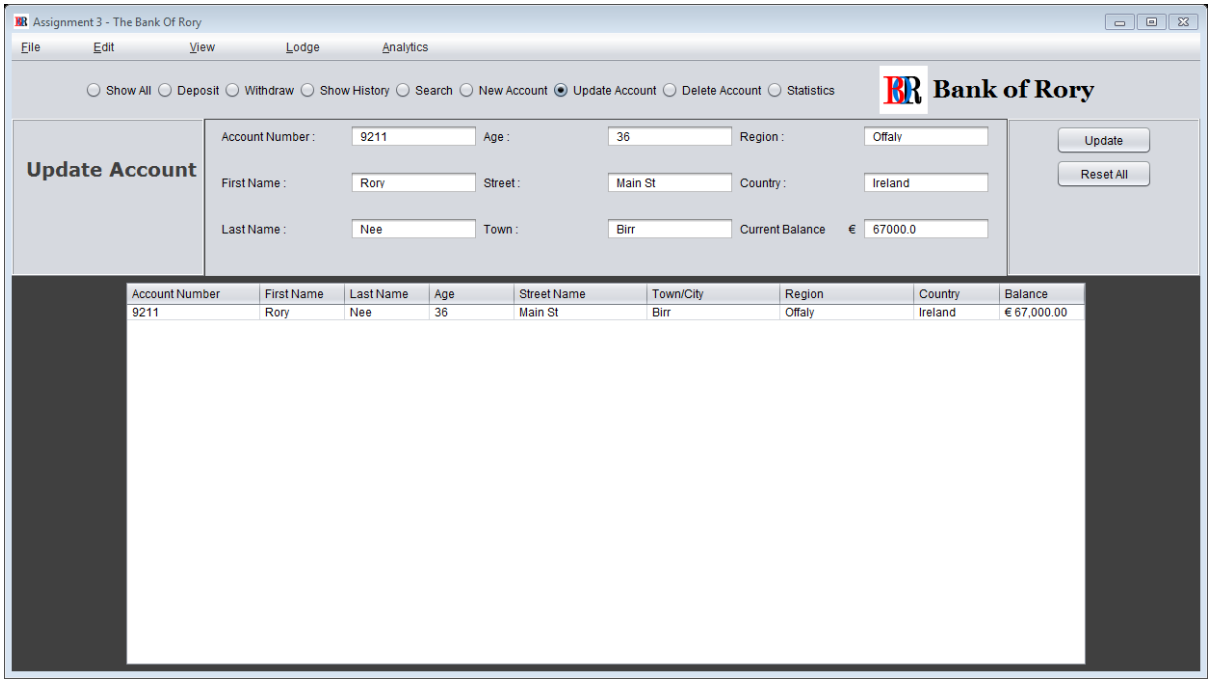


Fig 17

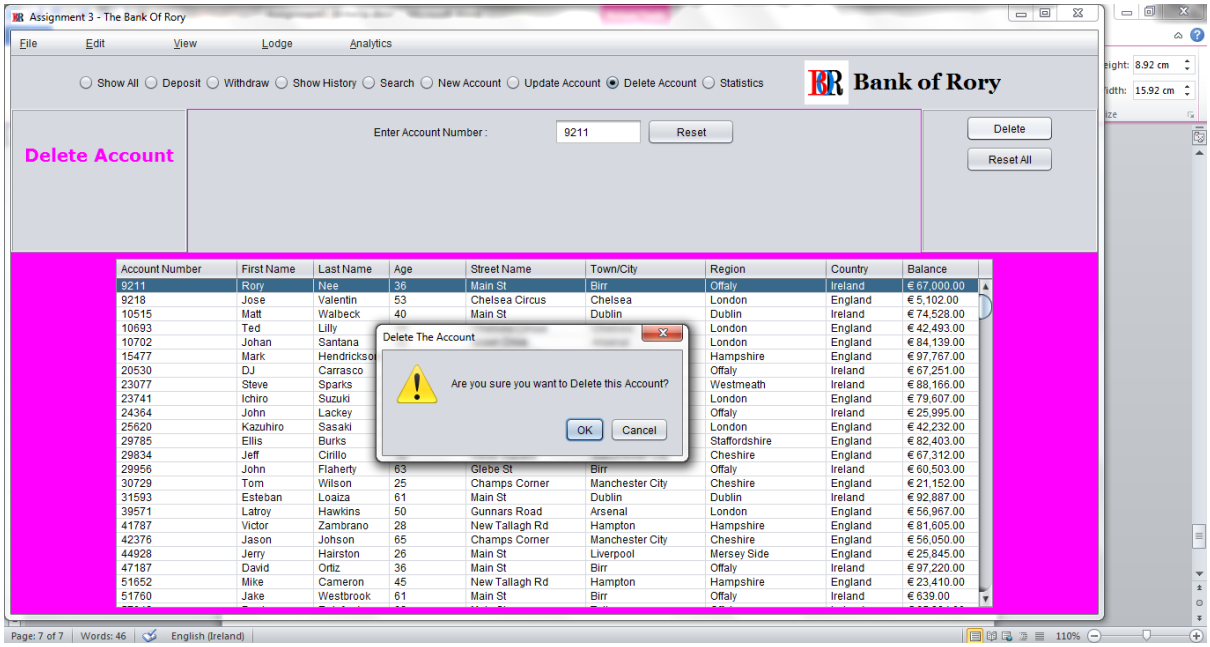


Fig 18 – When deleting an account the user can click an account to delete or type it in. The user will see a warning massage that they are about to delete an account. Then after they press ok in Fig 19 they will see a success message and the account list will be updated.

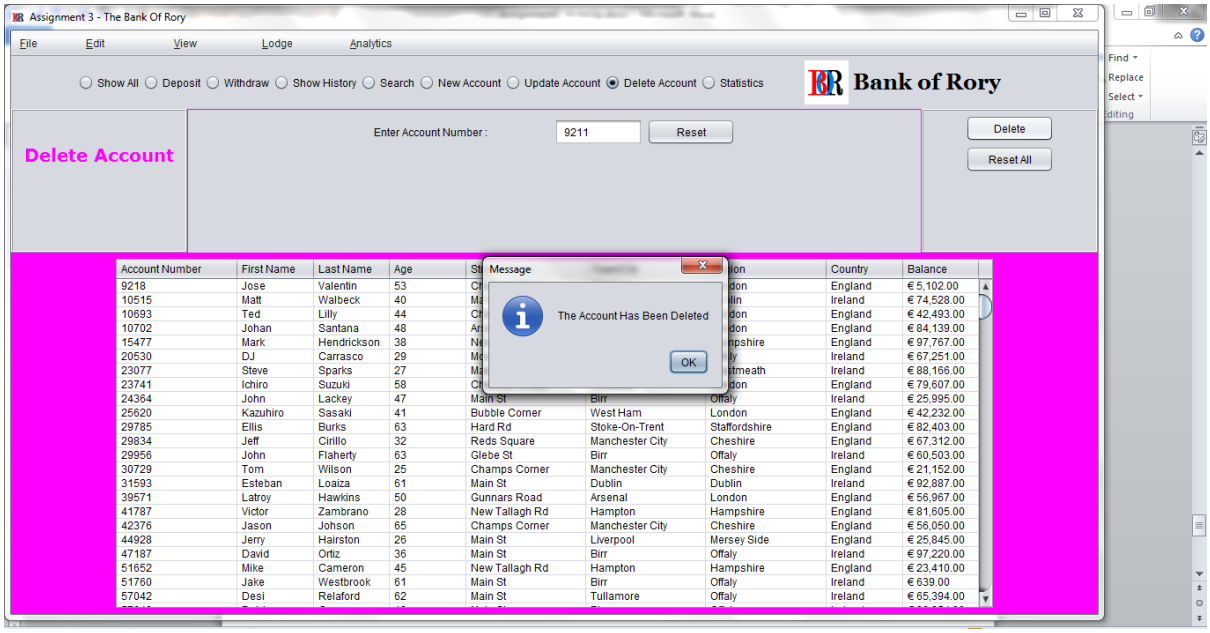


Fig 19

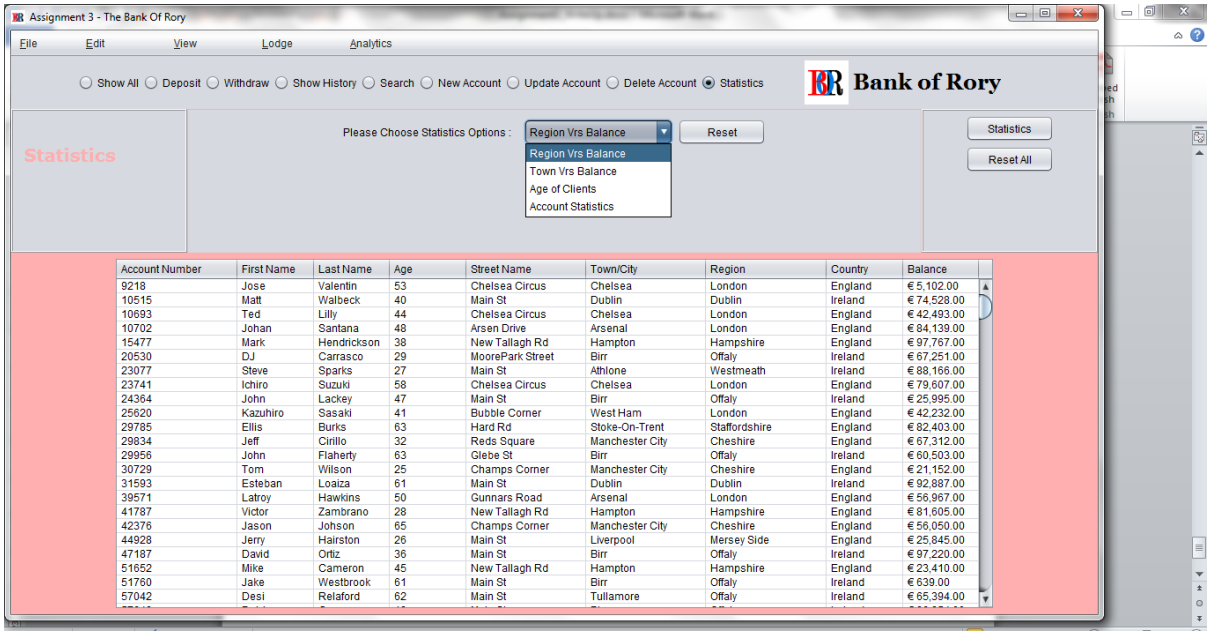


Fig 20 – The statistics option show a list of four different types of statistics to choose from.

Region vs. Balance – Balance's for each region mapped on a bar chart. Fig 21

Towns vs. Balance – Balance's for each town mapped on a bar chart. Fig 22

Age of Clients – Show the age of clients vs. the number of clients on a Line Graph. Fig 23

Overall account statistics – General information about the accounts. Fig 24

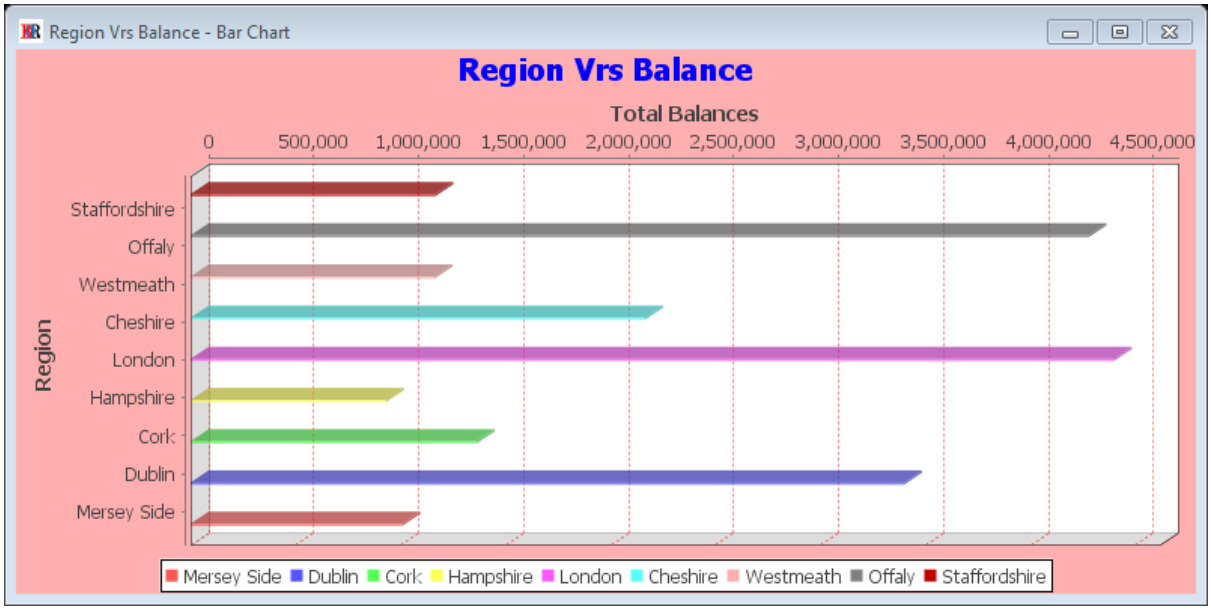


Fig 21

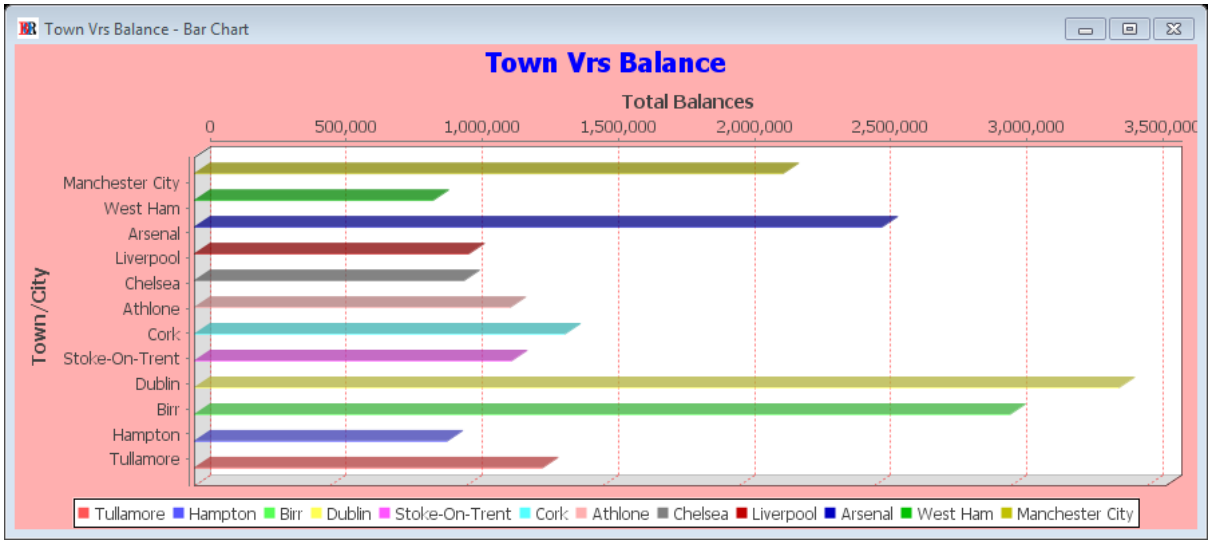

Town Vrs Balance - Bar Chart
Town Vrs Balance
Total Balances
0
500,000
1,000,000
1,500,000
2,000,000
2,500,000
3,000,000
3,500,000
Town/City
Manchester City
West Ham
Arsenal
Liverpool
Chelsea
Athlone
Cork
Stoke-On-Trent
Dublin
Birr
Hampton
Tullamore
Tullamore Hampton Birr Dublin Stoke-On-Trent Cork Athlone Chelsea Liverpool Arsenal West Ham Manchester City

Fig 22

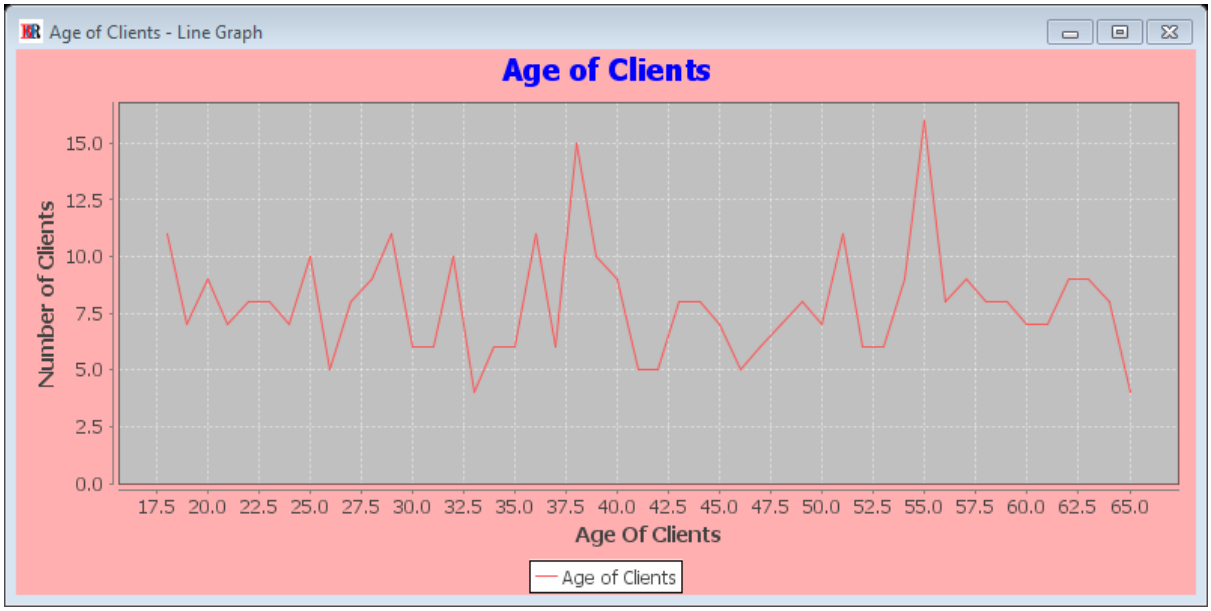

Age of Clients - Line Graph
Age of Clients
Number of Clients
15.0
12.5
10.0
7.5
5.0
2.5
0.0
17.5 20.0 22.5 25.0 27.5 30.0 32.5 35.0 37.5 40.0 42.5 45.0 47.5 50.0 52.5 55.0 57.5 60.0 62.5 65.0
Age Of Clients
Age of Clients

Fig 23

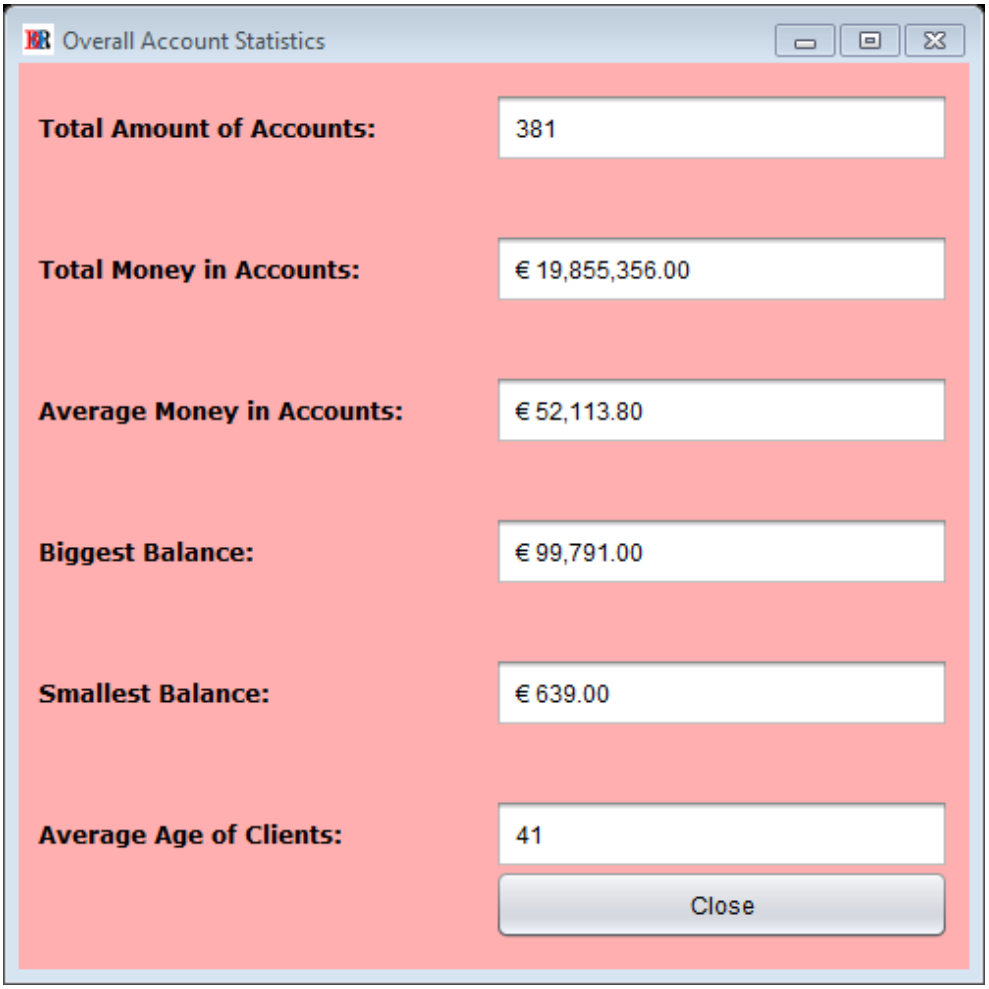


Fig 24

Conclusion

I feel this assignment has widened my knowledge of three Tier Object Oriented Systems.

I did come across some problems getting the serialisation, RMI and calculating the statistics working but I was able to overcome these.

I think my design of the Gui and the overall system was implements to a high standard but I do feel that I could have wrote the MainGuiClass's code differently. It was always my intention to split up the MainGuiClass in to individual classes but I ran out of time towards the end of the project. Because I felt this may happen, I planned for it by splitting up some of the Gui setup into smaller 'helper' methods. This way the code would still be more readable even if I ran out of time.

In conclusion I feel I have fulfilled the brief of the assignment by creating a bank account system that can be used by bank teller to Create, Read, Update, Delete and List the accounts. I feel I have also went further by adding in dynamic statistical information gathered from the list of accounts.

Rory Nee
A00190040