AIN SHAMS UNIVERSITYFACULTY OF ENGINEERING



**Student Name: Hussien Ahmad Abdelgelil Mohammed**

**Student ID (ASU): 2000459**



**Computer Engineering and Software Systems Program - CreditHours Engineering Programs (iCHEP)**

**STUDENT PORTFOLIO - Academic**

**Year2022/2023**

|  |  |  |
| --- | --- | --- |
| *ASU Course Code*  **CSE338** | *ASU Course Name*  **Software Testing, Validation, and Verification** | |
|  | **Semester**  spring 2023 | **Date of Submission**  19/7/2023 |

AIN SHAMS UNIVERSITY

1. Credit Hours Engineering Programs (i.CHEP)

Marking criteria

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Code: | | CSE338 | | | Course Name: | | | Software Testing,validation,  and Verification | | | | | | Major Task | | | | Date: | | |  | | | | |
| Student Name: | | **Hussien ahmad abdelgelil mohammed khalifa** | | | | | | | | | | | | | | Student ID: | | | | 2000459 | | | | | |
|  | Mastery ≥90% | | | | | | Accomplished ≥75 &  <90% | | | | | | Adequate ≥60 & <75% | | | | | | | | Inadequate <60% | | | | |
| 100 | | 96 | 93 | | 90 | 89 | 84 | 79 | 75 | | | 74 | | 69 | 64 | | | 60 | | 59 | 40 | 20 | 0 | |
|  | * Write down correctly all used equations and formula * Substitute all given date correctly * Solve to get the required values correctly * Using related SI units * Draw related – sketches, wave forms, items arrangements, * Comment in results | | | | | | * Write down correctly all used equations and formula * Substitute all given date correctly * Solve to get the required values correctly * Using related SI units * Partially Draw related – sketches, wave forms, items arrangements, * Comment in results | | | | | | * Write down correctly all used equations and formula * Substitute all given date correctly * Solve to get the required values * Did not use related SI units * Partially Draw related – sketches, wave forms, items arrangements, * Did not comment in results | | | | | | | | * Write down partially all used equations and formula * Substitute all given date partially correctly * Solve to get the required values * Using related SI units * Draw related – sketches, wave forms, items arrangements, * Did not comment in results | | | | |
| **x** | |  |  | |  |  |  |  |  | | |  | |  |  | | |  | |  |  |  |  | |
| **1st marker Total** | | | | | |  | | 1st marker Signature | | | | | | Dr. Islam Elmaddah | | | | | | | **ASU** Agreed  Mark | | 100% | | |
| **2nd Marker Total** | | | | | |  | | 2nd marker Signature | | | | | | TA. Mahmoud | | | | | | | **UEL** Agreed  Mark | | 100% | | |
| **General Comments:** | | | | | | | | | | |  | **UEL Grading System** | | | | | **Agreed Mark Range** | | | | **ASU Grading Scale** | | | |  |
| **% equivalent at UEL** | | | | | **% at ASU** | | | **Grade** |  |
| 95% and higher | | | | | x | | | | 97% and higher | | | A+ |  |
| 82% to less than 95% | | | | |  | | | | 93% to less than 97% | | | A |  |
| 70% to less than 82% | | | | |  | | | | 89% to less than 93% | | | A- |  |
| 66% to less than 70% | | | | |  | | | | 84% to less than 89% | | | B+ |  |
| 63% to less than 66% | | | | |  | | | | 80% to less than 84% | | | B |  |
| 60% to less than 63% | | | | |  | | | | 76% to less than 80% | | | B- |  |
| 56% to less than 60% | | | | |  | | | | 73% to less than 76% | | | C+ |  |
| 53% to less than 56% | | | | |  | | | | 70% to less than 73% | | | C |  |
| 50% to less than 53% | | | | |  | | | | 67% to less than 70% | | | C- |  |
| 45% to less than 50% | | | | |  | | | | 64% to less than 67% | | | D+ |  |
| 40% to less than 45% | | | | |  | | | | 60% to less than 64% | | | D |  |
| Less than 40% | | | | |  | | | | Less than 60% | | | F |  |
|  | | | | |  | | | |  | | |  |  |

1



**Ain Shams University Faculty of Engineering**

**Computer Engineering and Software Systems**

**CSE338: Software Testing, Validation, and Verification**

**Testing project**

Online banking documentation

**Submitted To**

Prof. Islam El Maddah

**Submitted by Team 5**

* + Nada Wagdy Mohamed 20P3253
  + Gana Allah Mohamed Abdel kriam 20P9538
  + Abdelrahman salah ahmed 18P9174
  + Hussien ahmed abdelgelil 2000459
  + Mohammed elhag mohammed 19p1472
  + Omar ayman ayoub abdelshafi 1900804

Contents

[*Online banking Desktop application* 3](#_bookmark0)

[*Description* 3](#_bookmark1)

[*Project components:* 3](#_bookmark2)

[*Development* 4](#_bookmark3)

[*test cases and suit* 4](#_bookmark4)

[*gui screens* 5](#_bookmark5)

[*Unit Testing:* 9](#_bookmark6)

[*Account class testing* 9](#_bookmark7)

[*Item class testing* 10](#_bookmark8)

[*User class testing* 11](#_bookmark9)

[*Output of junit testing using test suit* 12](#_bookmark10)

[*GUI testing:* 13](#_bookmark11)

[*sequence of creation a valid account* 13](#_bookmark12)

[*sign in a valid account* 14](#_bookmark13)

[*create a bank account with valid balance* 14](#_bookmark14)

[*buy items &bills, do transaction and show statements sequence* 14](#_bookmark15)

[*check defaults* 15](#_bookmark16)

[*Edge & state coverage diagram* 17](#_bookmark17)

[*state coverage and transition coverage* 18](#_bookmark18)

[*performance testing:* 19](#_bookmark19)

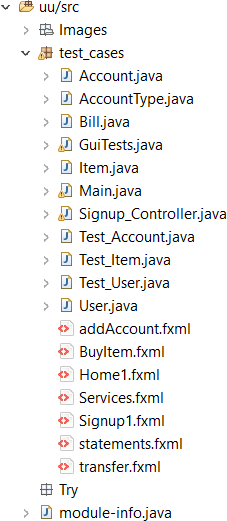
# *Online banking Desktop application*

## *Description:*

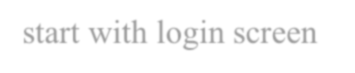
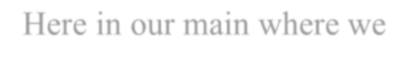
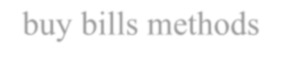
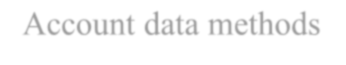
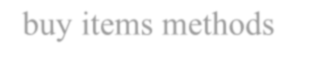
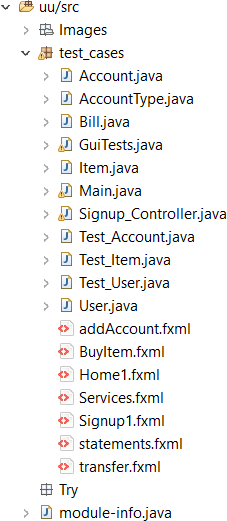
Our system is online banking. Through our system, users can buy items, pay bills, and transfer money to other accounts. They must be able to see the statements of their bank accounts and transactions.

## *Project components:*

In our project we got classes for functionality, GUI controller and test suits



***Development:***



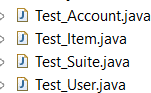
Account data methods buy bills methods

buy items methods

Here in our main where we start with login screen

***test cases and suit:***

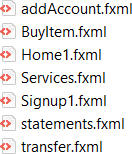
Here where all sets of functionality test cases are organized



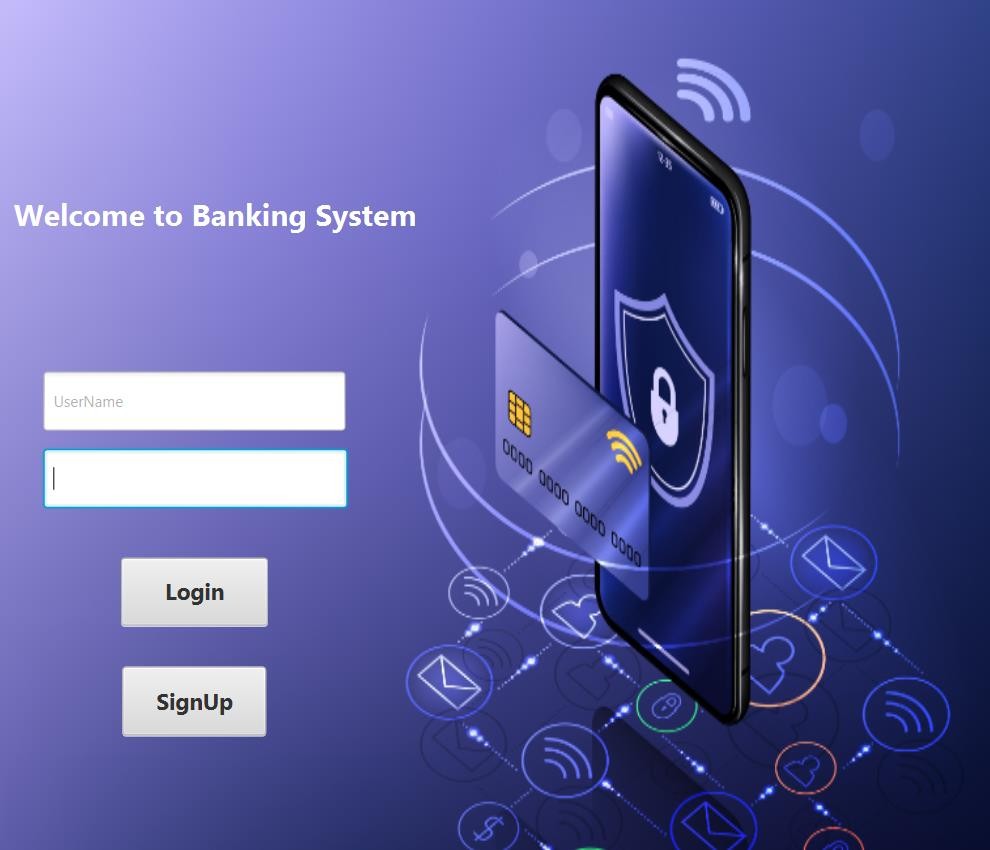
*testsuite*

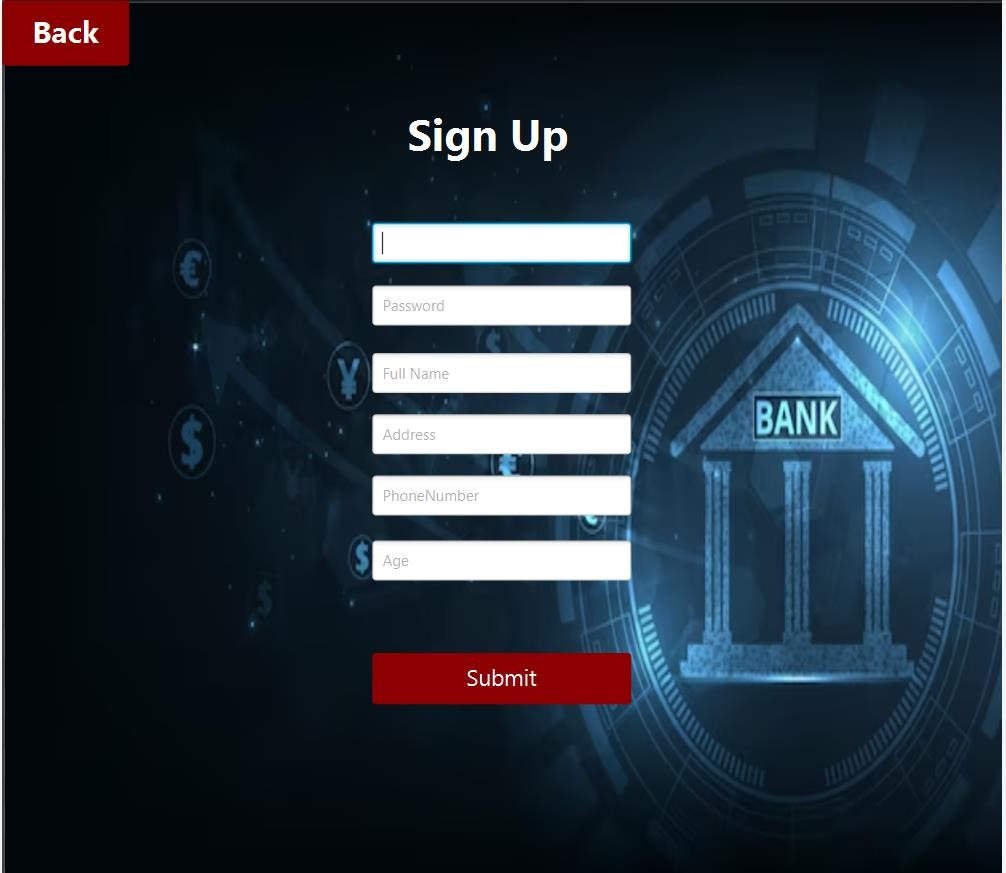
***gui screens***

We got 7 connected screens and the Home is login first screen :

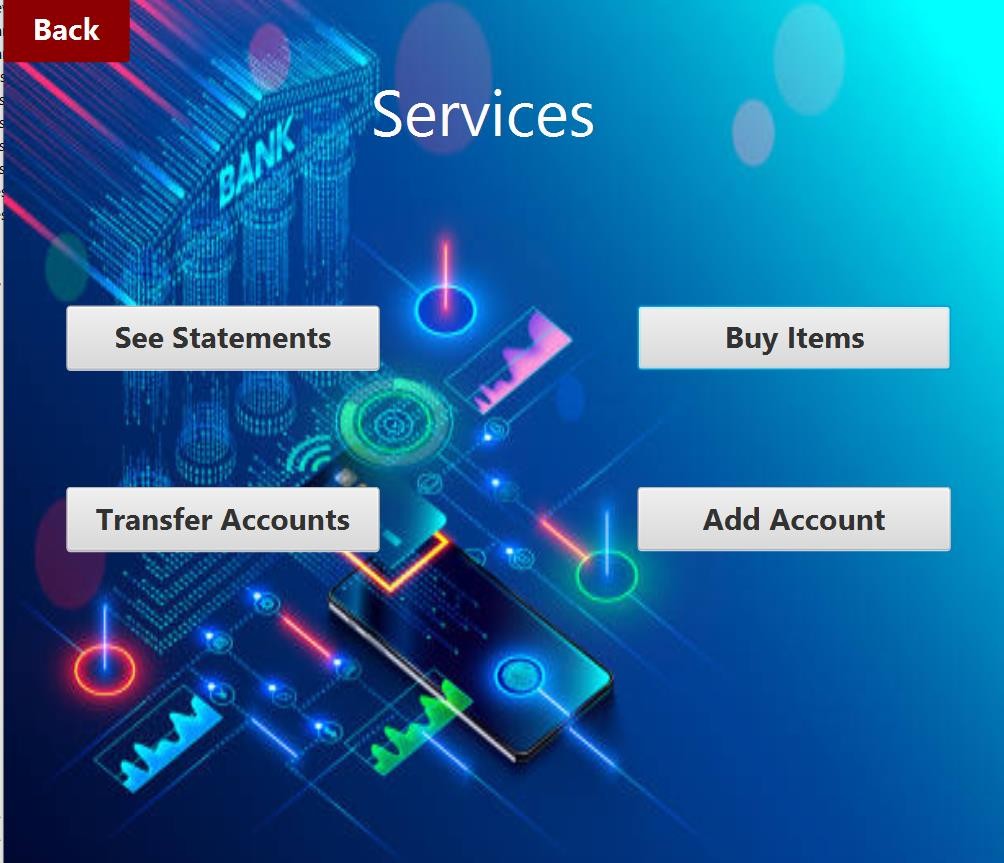


Login screen:

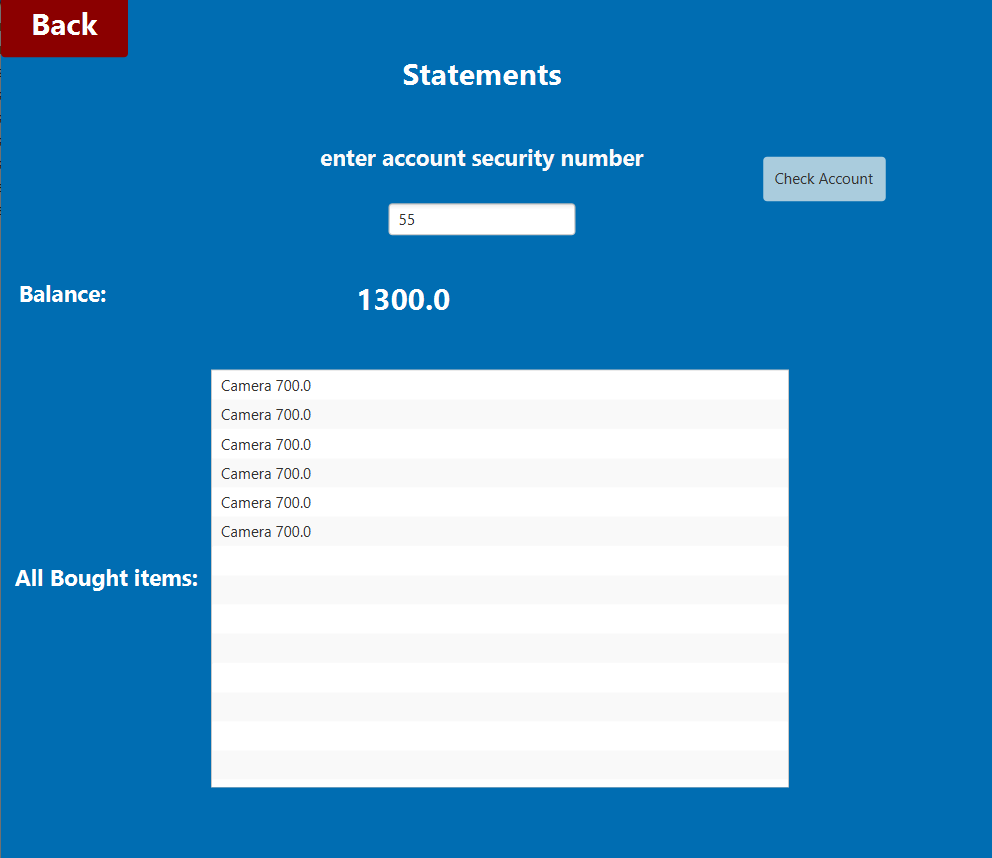


Signup screen:

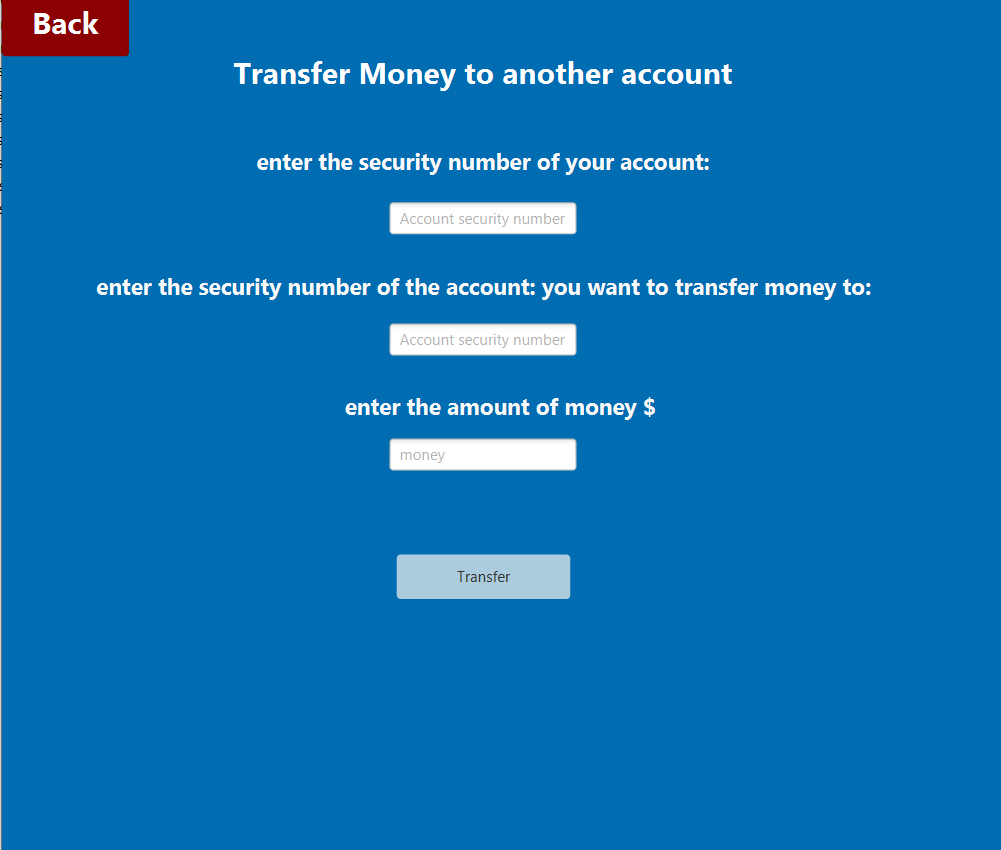
Services screen

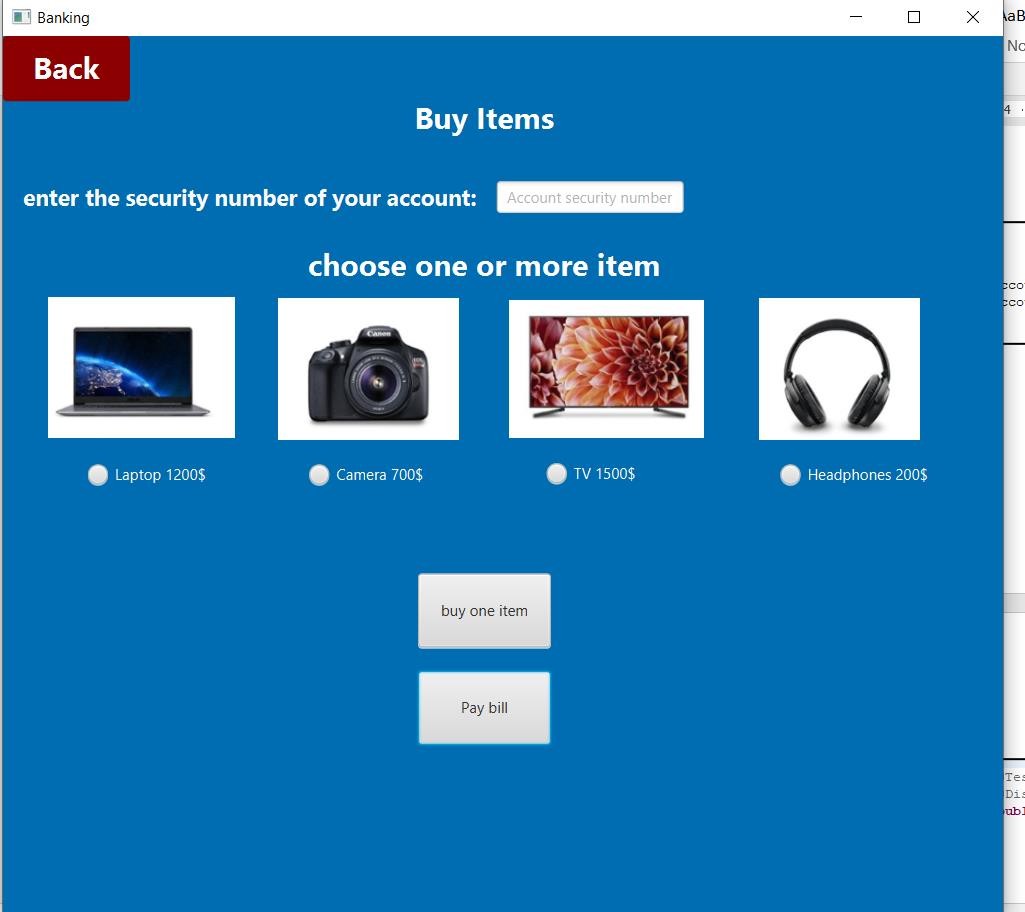


Showing statements screen ( for purchases history )

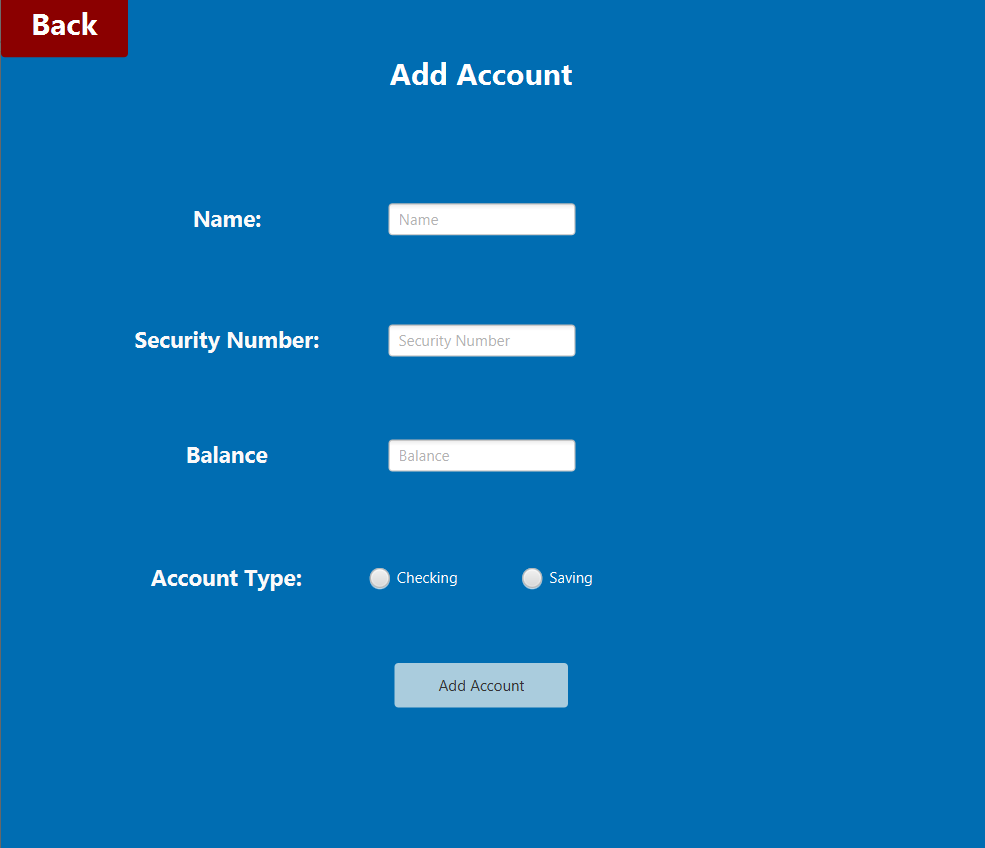


transfer money screen



Buy items & bills screen :

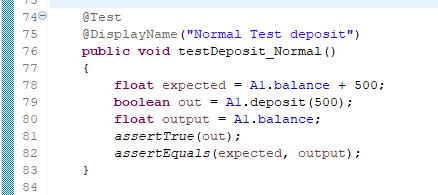
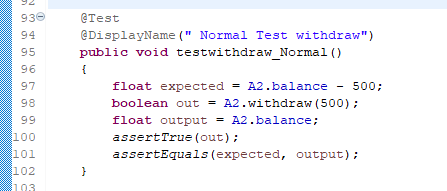
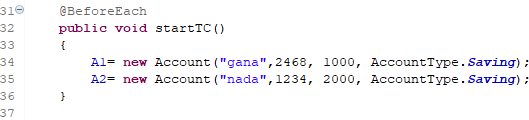
New bank account screen



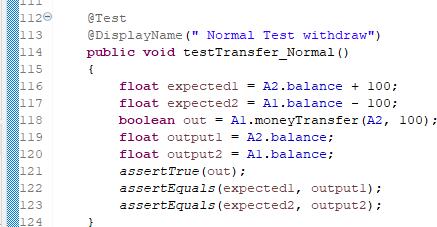
## *Unit Testing:*

***Account class testing***

First, we generated the basic test cases

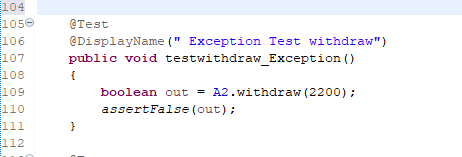
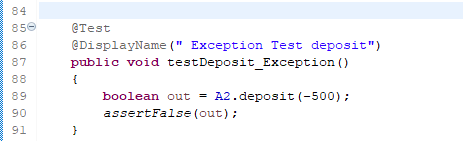


*Figure 1 withdraw test case Figure 2 deposit test case*

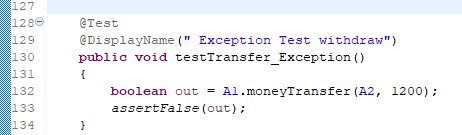


*Figure 3 transfer money test case*

But these test cases didn’t cover the exceptions, so we added more test cases to cover these exceptions these test cases got failed, so we developed our code to handle these exceptions and add more test cases to try the exceptions and they got passed:



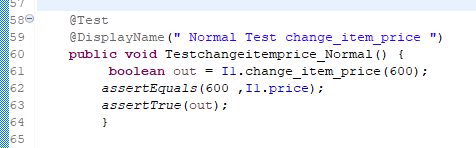
*Figure 4 withdraw test case Figure 5 Deposit test case*



***Item class testing***

*Figure 6 transfer money*

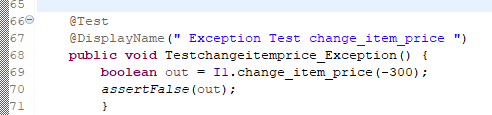
We did the same thing we did in Account class starting by generating a basic test case



*Figure 7 changeitemprice test case*

This test case did not cover the exception, so we added an exception test case which got failed so we developed our code to handle this exception then we

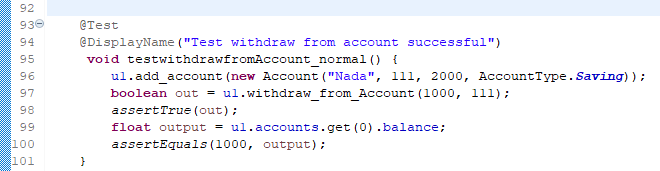
tried an exception test case which got passed



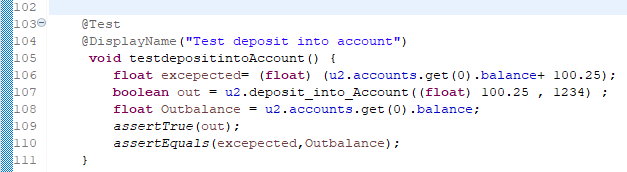
***User class testing***

*Figure 8 changeitemprice test case*

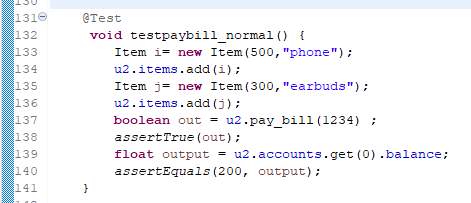
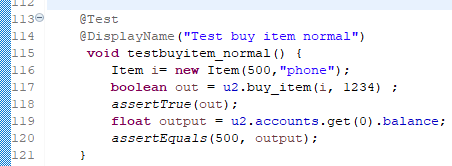
We did the same thing we did in two other classes starting by generating a basic test case



*Figure 9 withdraw fromAccount test case*



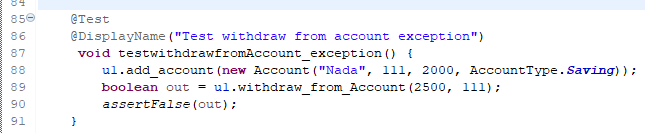
*Figure 10 deposit into Account test case*



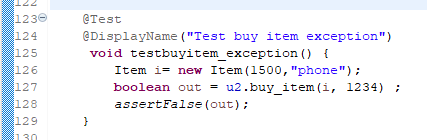
*Figure 11 buyitem test case Figure 12 paybill test case*

These test cases did not cover the exceptions, so we added exception test cases which got failed so we developed our code to handle this exception then we

tried an exception test case which got passed

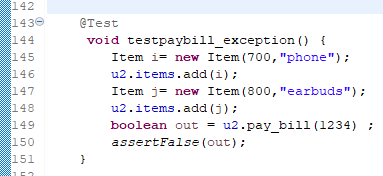


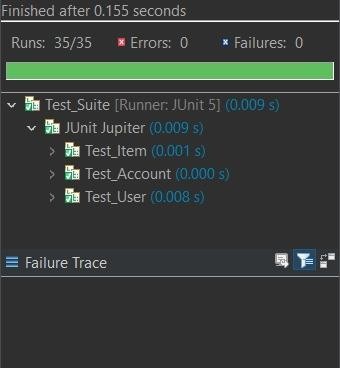
*Figure 13 withdrawfromAccount test case*

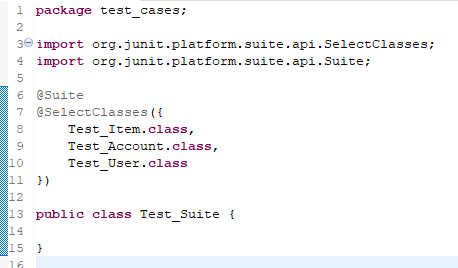


*Figure 14 buyitem test case*

*Figure 15 paybill test case*



***Output of junit testing using test suit***

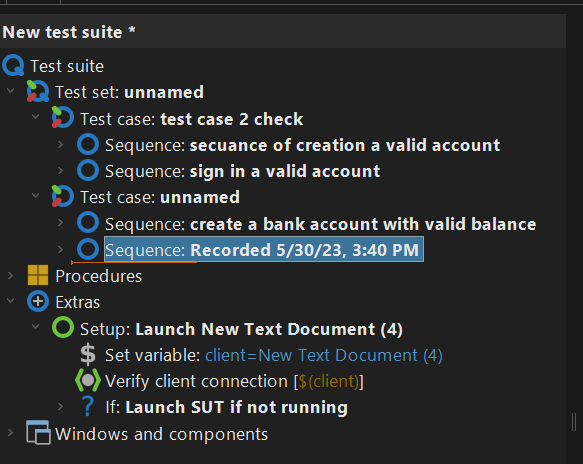


*Figure 16 Test\_Suite*

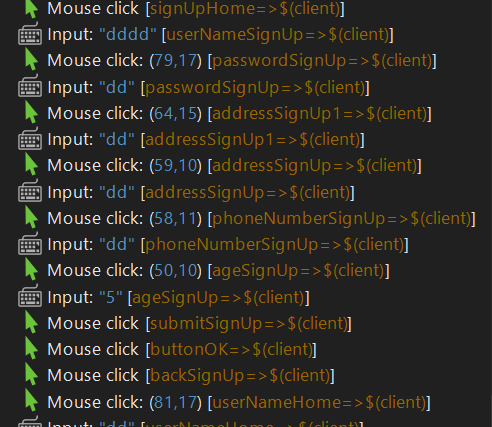
*Figure 16 Test\_Suite output*

## *GUI testing:*

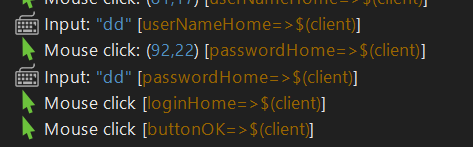
**We are using QF-test for gui testing :**

Here we extracted a stand-alone executable javaFx file so that the QF-test can deal with it

In this tool it can do recorded sequence and check the expected behaviours and outputs:

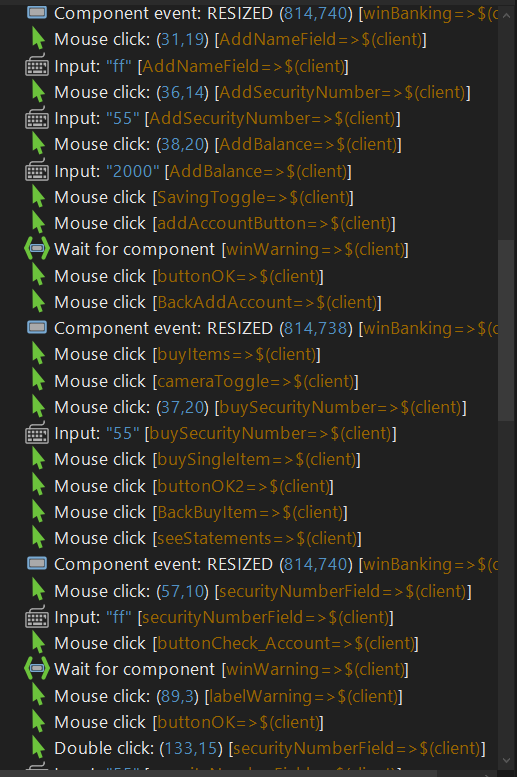
***sequence of creation a valid account***

***sign in a valid account***

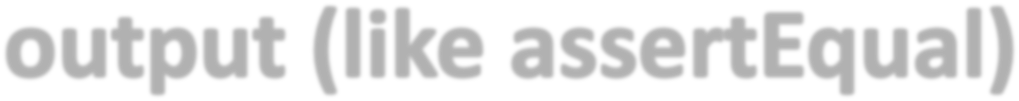
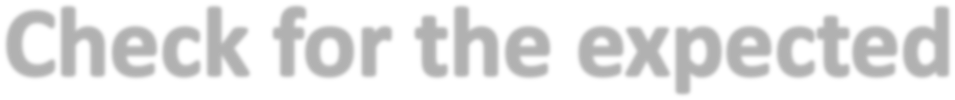
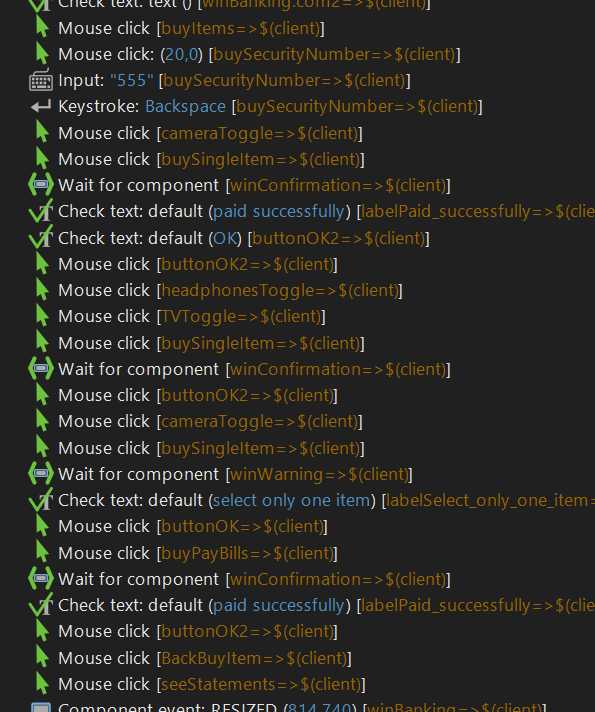
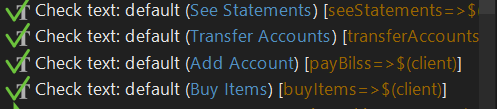
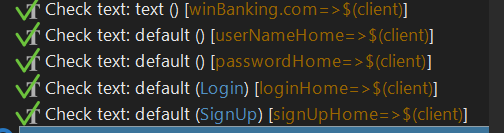


***create a bank account with valid balance***

***buy items &bills, do transaction and show statements sequence***



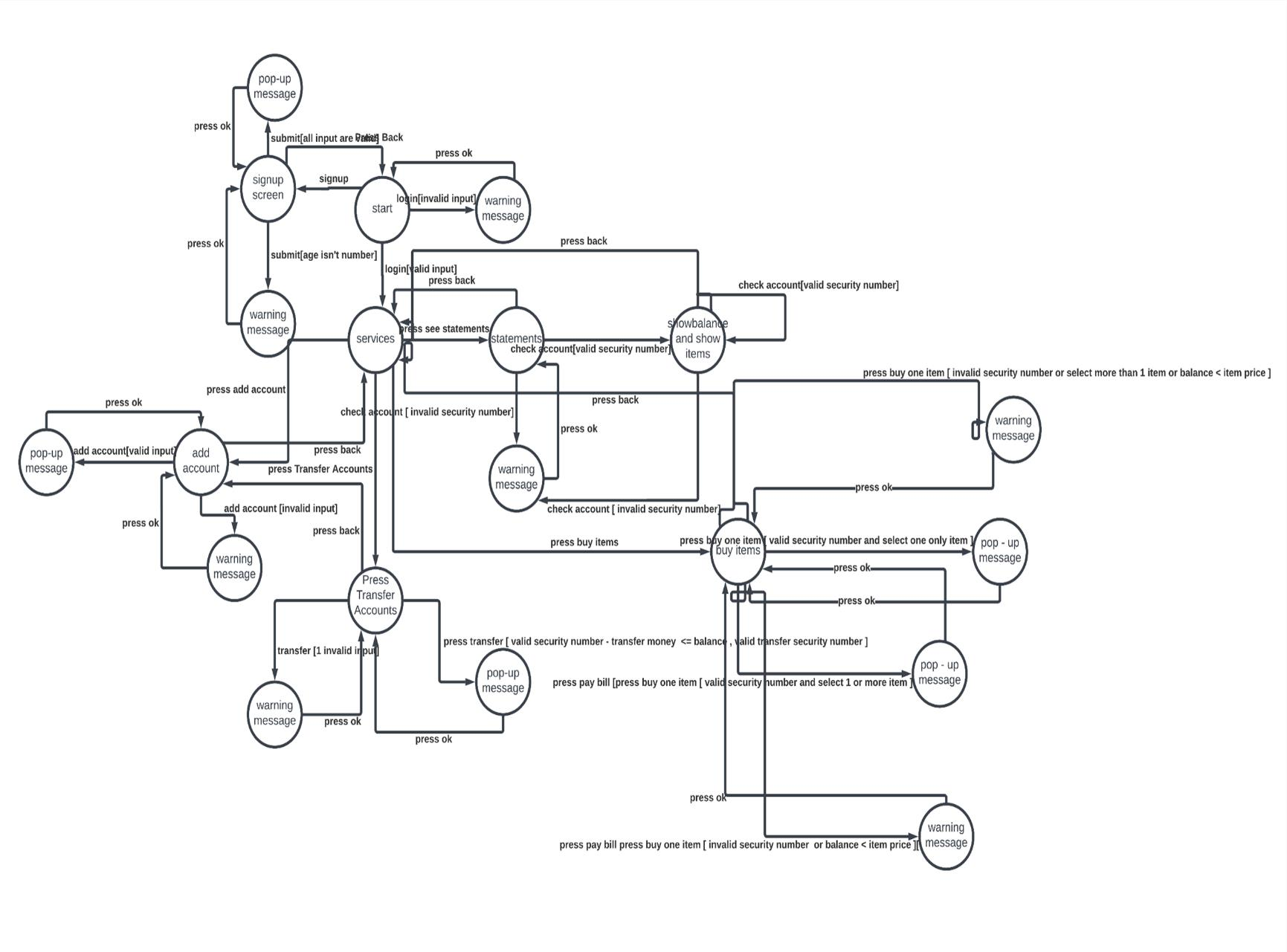
***check defaults***



***After test run :***



***Edge & state coverage diagram***



## *state coverage and transition coverage*

TC : ( < sign up, submit [ age isn't number], press ok, submit [all input are valid ], press ok, press back , login [invalid input], press ok , login [valid input] , press add acount, add account [ invalid input], press ok , add account [ valid input ] , press ok , press back

, press transfer accounts, transfer [ invalid input] , press ok , press transfer [ valid security number - transfer money <= balance , valid transfer security number ] ,press ok , press back , press buy items , press pay bill press buy one item [ invalid security number or balance < item price ] , press ok , press pay bill [press buy one item [ valid security number and select 1 or more item ], press ok, press buy one item [ invalid security number or select more than 1 item or balance < item price ], press ok, press buy one item [ valid security number and select one only item ], press ok, press back , press see statements, check account [ invalid security number], press ok ,check account[valid security number],check account[valid security number], check account [ invalid security number], press ok, press back >)

expected output < show up sign up menu , warning message , close message, successful message , close message , show login screen , show services screen , show add account screen, warning message , close message, successful message , close message , show services , show transfer screen, warning message , close message, successful message , close message, show services , show items screen, warning message , close message, successful message , close message , warning message , close message, successful message , close message , show services , show statements screen, warning message , close message, show balance and show list of items , show balance and show list of items , warning message , close message , show services >

## *performance testing:*

stability, speed, scalability and stress testing:

