TestReport

Software testing final project

Hossain Al Imran

201853082041

Content

INTRODUCTION

Introduction to the ATM system:

Automated Teller Machine enables the clients of a bank to have access to their account without going to the bank. This is achieved only by development the application using online concepts.

When the product is implemented, the user who uses this product will be able to see all the information and services provided by the ATM, when he enters the necessary option and arguments. The product also provides services like request for cheques, deposit cash and other advanced requirement of the user. The data is stored in the database and is retrieved whenever necessary. The implementation needs ATM machine hardware to operate or similar simulated conditions can also be used to successfully use the developed product.

To develop this ATM system the entire operation has been divided into the following step:

- verification process
- language, service and account selection
- Banking services
- Transactions
- Special services

The program is designed in such a way that the user has to card and pin number. Once verified, he is provided a menu and he/she had to enter the option provided in the menu. For example, when the user wants to view the list of payment history than he/she had to enter the option for payment history provided in the main menu. When the option is entered alone with the respective argument, then the payment history is displayed on the screen.

The user also must be given option to browse through the pages like previous page, next page, etc. The user may experience a delay in retrieving or viewing the data, when there are many users logged on to the same bank branch system.

Project Planning:

Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project. It prevents obstacles that arise in the project such as changes in projects or organization's objectives, non-availability of resources, and so on.

Project Planning Process

Project planning process consists of the following activities.

- Identification of project requirements: Before starting a project, it is essential to identify the project requirements as identification of project requirements helps in performing the activities in a systematic manner. These requirements comprise information such as project scope, data and functionality required in the software, and roles of the project management team members.
- Identification of cost estimates: Along with the estimation of effort and time, it is necessary to estimate the cost that is to be incurred on a project. The cost estimation includes the cost of hardware, network connections, and the cost required for the maintenance of hardware components.
- Identification of risks: Risks are unexpected events that have an adverse effect on the project. Software project involves several risks (like technical risks and business risks) that affect the project schedule and increase the cost of the project.
- Identification of critical success factors: For making a project successful, critical success factors are followed. These factors refer to the conditions that ensure greater chances of success of a project.
- Preparation of project charter: A project charter provides a brief description of the project scope, quality, time, cost, and resource constraints as described during project planning. It is prepared by the management for approval from the sponsor of the project.
- Preparation of project plan: A project plan provides information about the resources that are available for the project, individuals involved in the project, and the schedule according to which the project is to be carried out.

Test Overview:

Test Object:

The test object for the unit testing is a simple banking application. Made and designed using NetBeans IDE 12.2 software in Java language and it uses Java DB as the databases.

Test Time:

The tests are performed from May 15 through June 18 of 2021.

Test Methods:

The method used in this test is called the white box testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing). This method is named so because in the eyes of the tester, the software program is like a white/transparent box, where one can see clearly what's inside.

> White box testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester . The tester, usually a developer as well, chooses inputs to traverse the paths of the code and determines the appropriate outputs. Knowledge of programming and implementation is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.
>
> One of the advantage of using white box testing is the test is more comprehensive and can cover most paths. But because the testing can be very complex, it requires highly skilled resources and sufficient programming and implementation knowledge.

Definition by ISTQB (International Software Testing Qualifications Board):

- White-box testing: Testing based on an analysis of the internal structure of the component or system.
- White-box test design technique: Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

Test Constraint

Testers

The overall test constraints for this test are listed below:

- The test report is based on the tested software version.
- All tests are based on the same test environment as the development environment (including the operating system, database, and etc.);

The tester is also the author of this document and the developer of this project.

Need for the ATM system:

Millions of times per day around the globe people are instantly withdrawing money at automatic teller machines (ATMs). Given the fast-pace of the world today, it is not surprising that the demand for access to quick cash is so immense. The power of ATMs would not be possible without secure connections. The final act of ATM dispending cash is the result of an amazingly fast burst of the customer never sees, but a trust is being done in a confidential manner.

SYSTEM SPECIFICATION

Hardware Requirements:

- Processor – Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
- RAM – 4 GB
- Hard Disk – 50GB
- Mouse – Standard Mouse
- Keyboard – Logitech Keyboard
- Processor Speed – 2.5GHZ

PROJECT DESCRIPTION

Need For The Software:

Now a days every one very busy in their work. So they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into a record or file and it may be missed somewhere so, this system help to keep the record of the customer it also keeps the details of he customer. It is also easy to access.

Problem description :

The system mainly used by the bank clients. When a client comes to ATM centre to update and delete their account. It reduces the time consumption and lot of paperwork. For any single operation it involves numerous references and updating also takes subsequent changes in other places.

SYSTEM STUDY AND ANALYSIS

Existing System:

The existing system is manual

system. The manual system is prone

to error.

This system involves a lot of manual entries with the application to perform a desired task.

Usage of papers and records in the process leads to less efficiently less productivity.

Increase lots of mistakes while writing in paper.

Time delay between the user and customer is

reduced. For this reason the new system in

invented.

Proposed System:

The system customer transactions, satisfies the requirements of the existing system in full-fledged manner. Through this system, customer can make fast transactions and view the last transactions easily.

System Analysis:

Understand the problem before the system to create analysis model there is a tendency to rush to a solution, even before the problem is understood.

Develop prototypes that enables user to understand how human/machine interaction will occur. Since the perception of the quality of software is often
based on the perception of the "friendliness" of the interface prototyping is highly recommended.

Record the origin of and the reason for every requirement. This is the first step-in establishing traceability back to the customer.

Use multiple views of requirements building data, functional and behavioral models provide the software engineer with three different views. This reduces the likelihood that something will be missed and increases the likelihood that inconsistency will be recognized.

Rank requirements. Tight deadlines may preclude the implementation of every software requirements to be delivered in the first increment must be identified.

Work to eliminate ambiguity because most requirements are described in a natural language, the opportunity for ambiguity abounds.

Testing of ATM System

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

White box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing. Black box testing is often used for validation and white box testing is often used for verification. One advantage of the black box technique is that no programming knowledge is required. Whatever

biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight.

Test Situation

Test Content

The test is mainly carried out to check the functional feature using the application interface. The test mostly analyses the three interfaces which is the login window (login function), clerk’s window (create account, search account, 4 | P a g e and accounts summary
function) and customer’s window (balance inquiry, deposit, withdraw, transfer, and transaction history function). See table 3 for the login test cases. See table 4, table 5, table 6, and table 7 for the tests executed on the clerk’s window. See table 8, table 9, table 10, table 11, table 12, and

Test case ID	Test content	Execution times	Test Results
Log-0	Login with an em	3	Achieve the expected result.

	pty account number and PIN f		

	ield		
Log-1	Login with an empty account number field	3	Achieve the expected result.
Log-2	Login with an empty	3	Achieve the expected result.

	PIN field.		
Log-3	Login with a non-existent account number in the MySQL database.	3	Achieve the expected result.
Log-4	Minimize log	3	Achieve the expected result.

	in window		
Log-5	Close login window	3	Achieve the expected result.

table 13 for the tests performed on the customer's window. The table below lists the test cases on the login window:

Log-6	Click on the account number field	3	Achieve the expected result.
Log-7	Click on the PIN field	3	Achieve the expected result.
Log-8	Click on customer's radio button	3	Achieve the expected result.
Log-9	Click on clerk's radio button	3	Achieved the expected result.

Table 3 Login test cases

List of fields in the clerk's window (create account tab):

- PIN
- Confirm PIN
- Name
- Account Type (it has 3 type of account like saving account, credit account and checking account).
- Interest rate
- Deposit

The table below lists the test cases for making a new customer account on the clerk's window (create account tab):

Test case ID	Test content	Execution times	Test Results
CrtAct-0	Attempt to create a new account without inputting anything on all the fields (all the fields listed above are empty)	3	Achieve the expected result.
CrtAct-1	Attempt to create a new account with an invalid PIN (not a 6-digit number), and let all the other fields empty	3	Achieve the expected result.
CrtAct-2	Attempt to create a new account with a valid PIN, confirm PIN doesn't match the PIN, and let all the other fields empty	3	Achieve the expected result.
CrtAct-3	Attempt to create a new account with a valid PIN, confirm PIN matches the	3	Achieve the expected result.

	PIN, valid name, and let all the other fields empty		
CrtAct-4	with a valid PIN, confirm PIN matches the PIN, valid name and saving account as the account type and let all the other fields empty	3	Achieve the expected result.
CrtAct-5	Attempt to create a new account with a valid PIN (a 6-digit number), confirm PIN matches the PIN, valid name and saving account as the account type, valid interest rate, and empty deposit field	3	Achieve the expected result.
CrtAct-6	Create a new account with a valid PIN, confirm PIN matches the PIN, valid name, one of the three types of account types, valid field number 5 listed above the table, and valid deposit field	3	Achieve the expected result.
CrtAct-7	Click on each text field in the create account tab	8	Achieve the expected result.
CrtAct-8	Click on the reset button	3	Achieve the expected result.

Table 4 create account test cases

The table below lists the test cases for searching a customer account on the clerk's window (search account tab):

Test case ID	Test content	Execution times	Test Results
SrchAcc-0	Search account by pressing query button	3	Achieve the expected result.
SrchAcc-1	Show the data from MySQL DB on the table	3	Achieve the expected result.

Table 5 Search account test cases

The table below lists the additional test cases on the clerk's window:

Test case ID	Test content	Execution times	Test Results
Clerk-0	Click the minimize button	3	Achieve the expected result.
Clerk-1	Click the close button	3	Achieve the

			expected result.
Clerk-2	Click log out button	3	Achieve the expected result.

Table 6 additional test cases on clerk's window

The table below lists the test cases on the customer's window (deposit tab):

Test case ID	Test content	Execution times	Test Results
Dep-0	Click the deposit tab after login	4	Achieve the expected result.
Dep-1	Click the deposit tab after withdrawing	3	Achieve the expected result.
Dep-2	Click the deposit tab after transferring	3	Achieve the expected result.
Dep-3	Click enter after inputting the amount to deposit	3	Achieve the expected result.
Dep-4	Click deposit button with an empty amount	3	Achieve the expected result.

Table 7 Deposit test cases

The table below lists the test cases on the customer's window (withdraw tab):

Test case ID	Test content	Execution times	Test Results
Withdraw-0	Click the withdraw tab after login	3	Achieve the expected result.
Withdraw-1	Click the withdraw tab after depositing	3	Achieve the expected result.
Withdraw-2	Withdraw some amount of money from the account	3	Achieve the expected result.
Withdraw-3	Click withdraw button with an empty amount	3	Achieve the expected result.

Table 8 Withdraw test cases

The table below lists the test cases on the customer's window (transfer tab):

Test case ID	Test content	Execution times	Test Results
Transfer-0	Click the transfer tab after login	3	Achieve the expected result.
Transfer-1	Click the transfer button after inputting a valid amount to withdraw (without inputting the target account)	3	Achieve the expected result.

Transfer-2	Click the transfer button without inputting anything in all the three fields (empty amount field, empty target account field, empty description field)	3	Achieve the expected result.

Table 9 Transfer test cases

The table below lists the additional test cases on the customer's window:

Test case ID	Test content	Execution times	Test Results
Custmr-0	Click the minimize button	3	Achieve the expected result.
Custmr-1	Click the close button	3	Achieve the expected result.
Custmr-2	Click log out button	3	Achieve the expected result.
Custmr-3	Click on each test field in the customer's window	3	Achieve the expected result.

Table 10 Additional test cases on customer's window

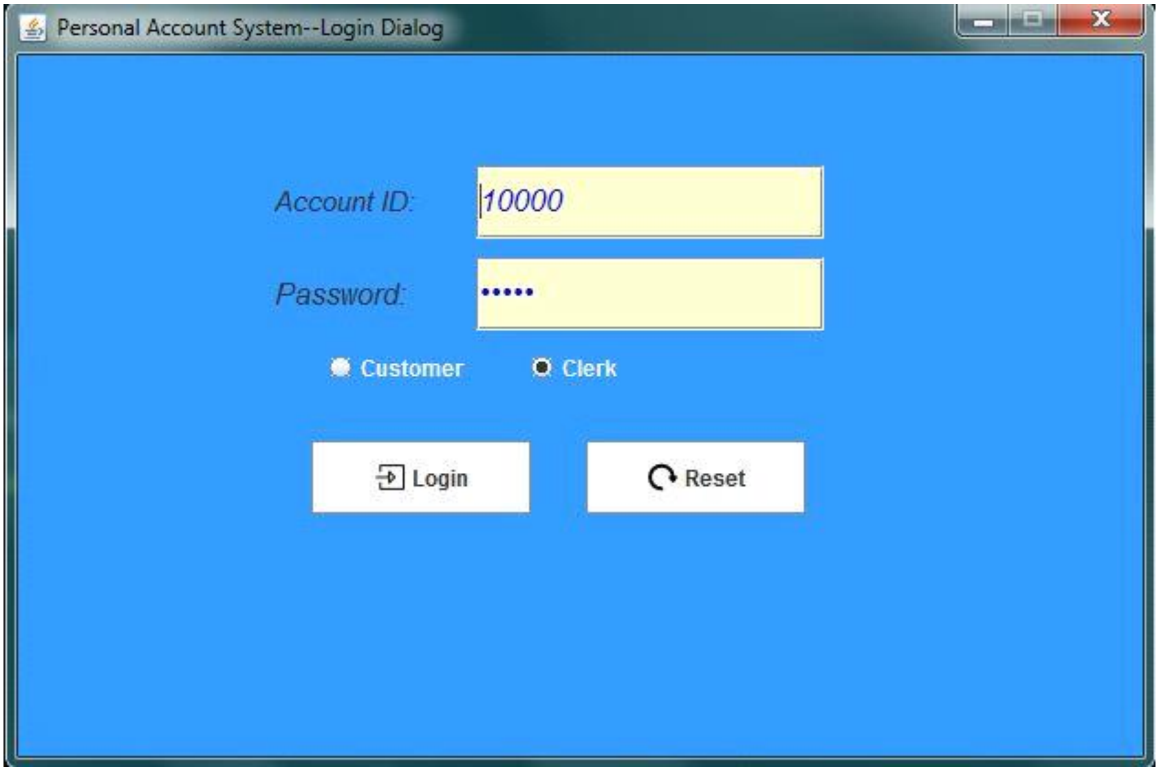
1.1 Test Completion Situation

Test task Name	Content	Progress (% Complete)
Test plans, writing test case	Find resources for the different formats used to display	100%
Prepare test data and environment	See test cases	95%
Sort out test data, writing test report	Sorting out test data to write test report	100%

Table 11 Test completion situation

SCREENSHOTS:

Login Form:



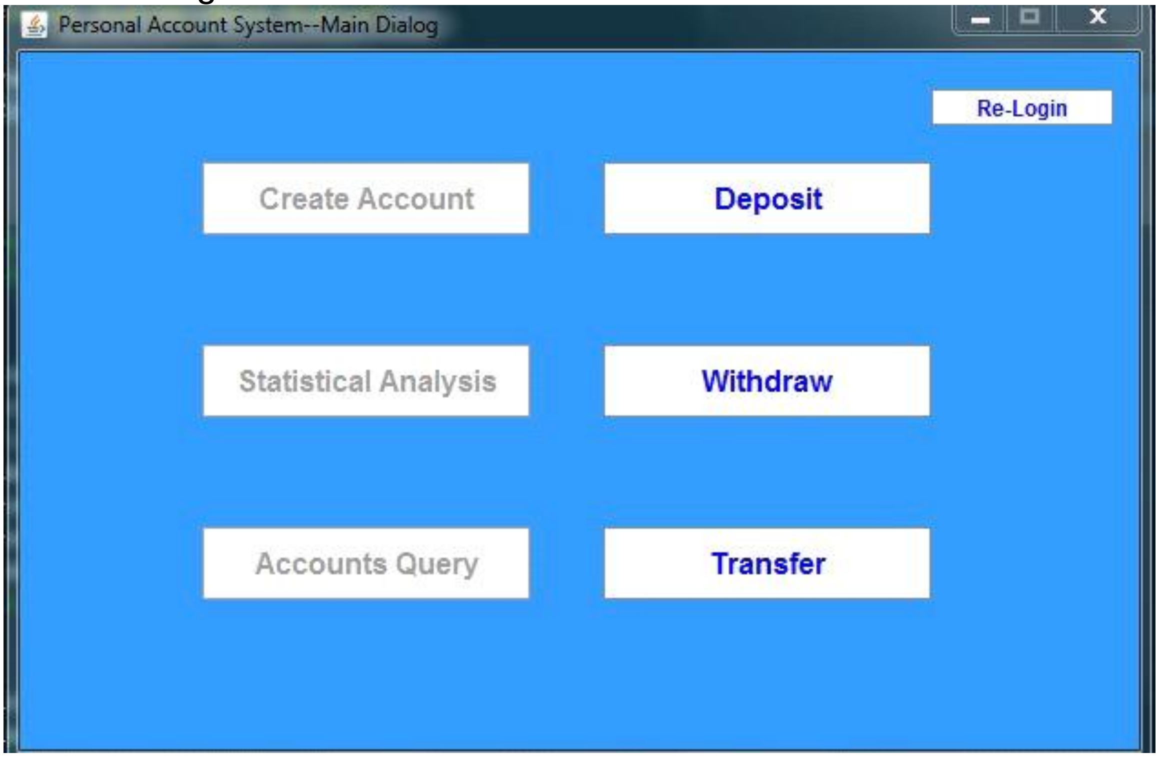
Personal Account System--Login Dialog

Account ID: 10000

Password: •••••

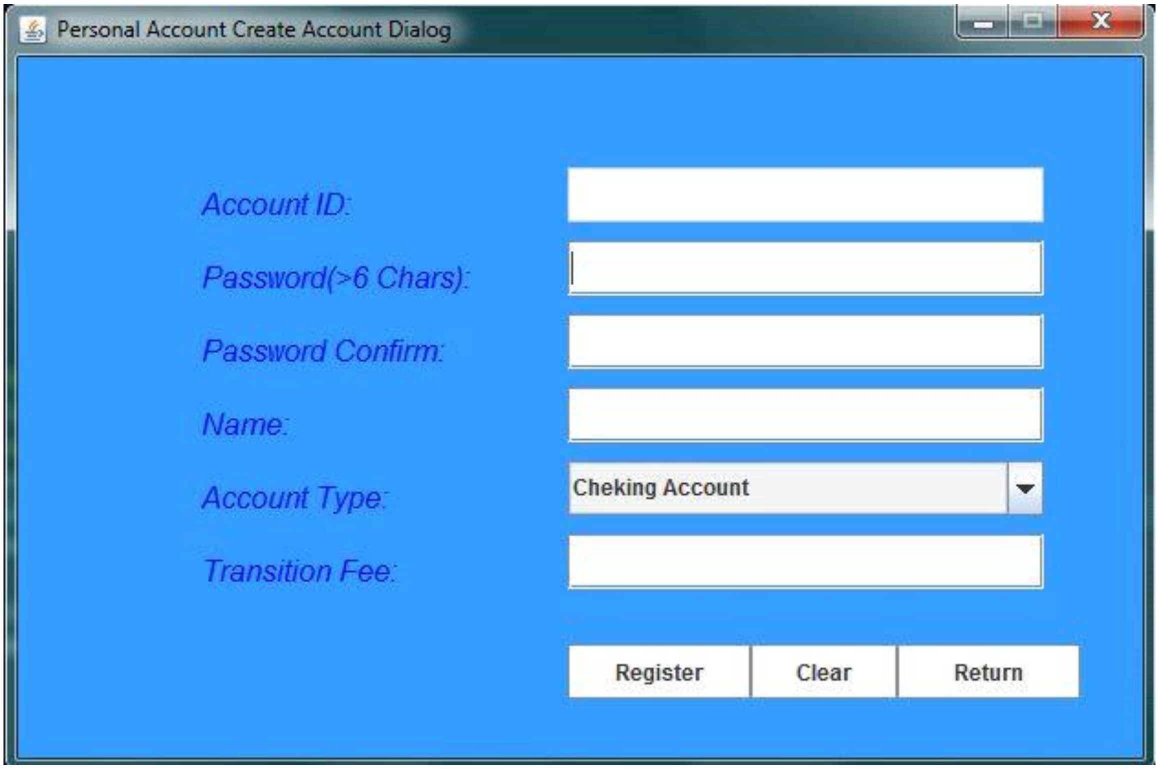
○ Customer ● Clerk

Main Dialog Form:



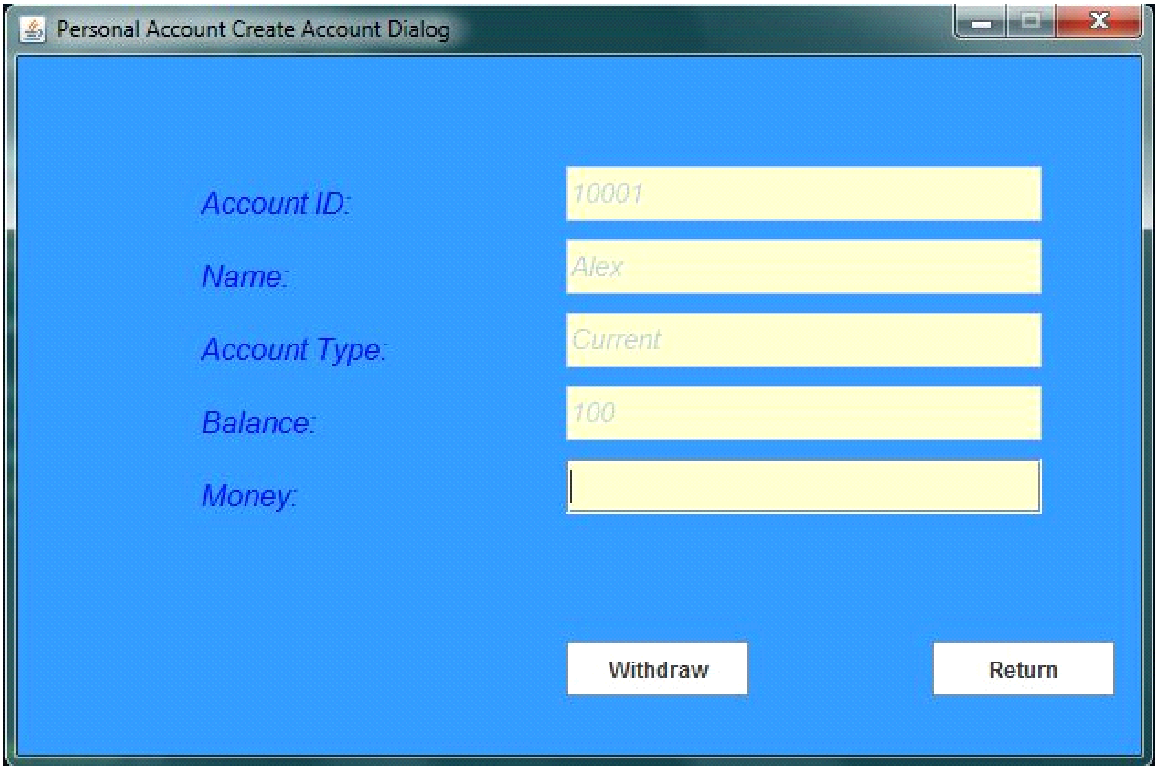
Personal Account System--Main Dialog

Create Account Form:



Account Type: Cheking Account

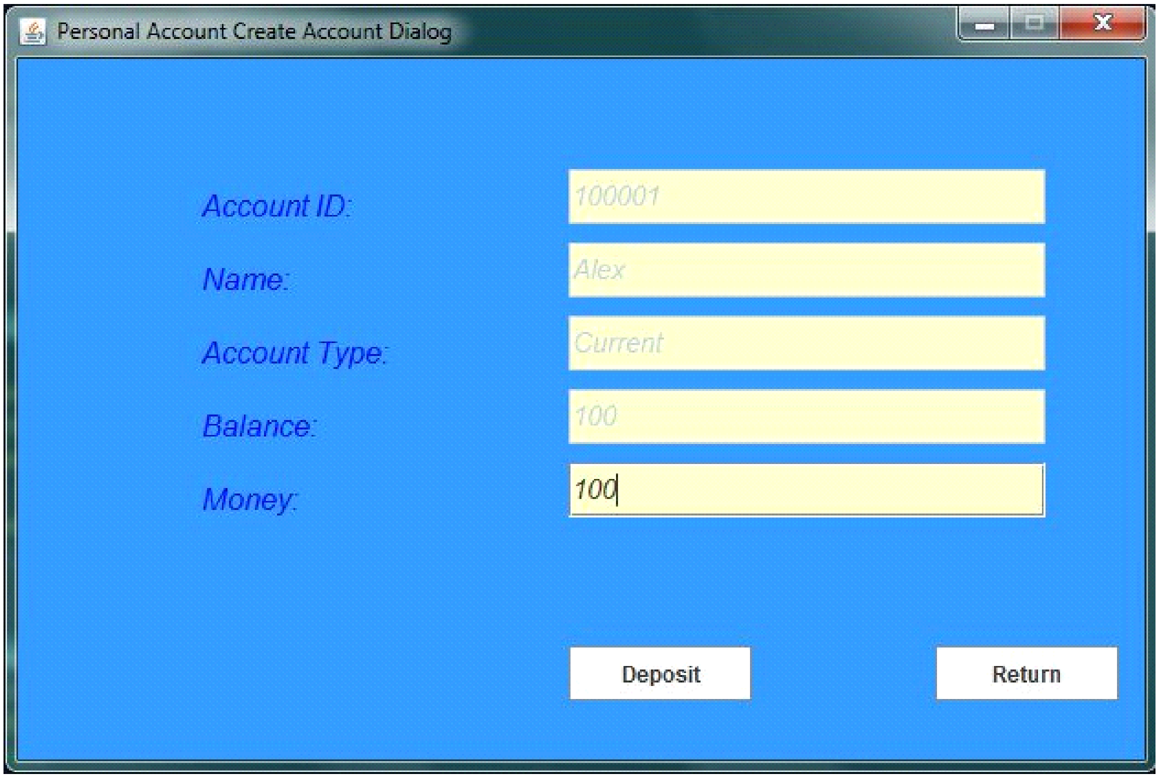
Withdraw Money Form:



Personal Account Create Account Dialog

Account ID:	10001
Name:	Alex
Account Type:	Current
Balance:	100
Money:	

Deposit Money Form:



Personal Account Create Account Dialog

Account ID: 100001

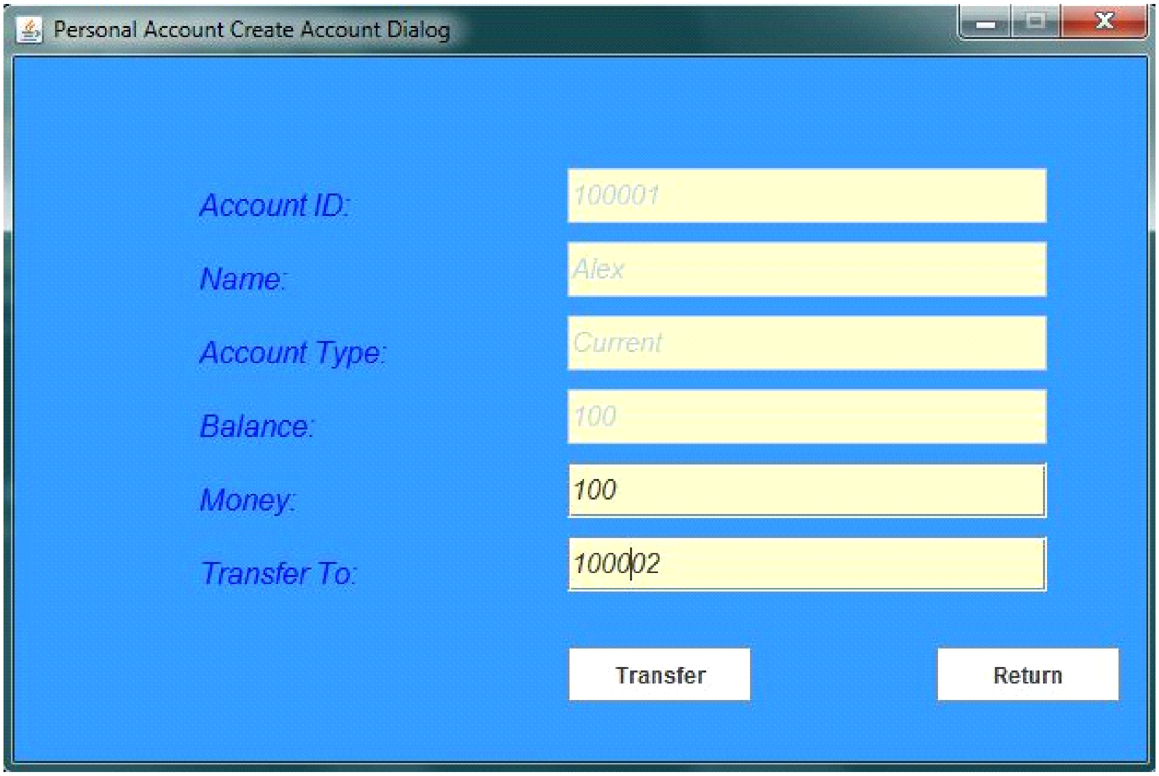
Name: Alex

Account Type: Current

Balance: 100

Money: 100

Transfer Money Form:



Personal Account Create Account Dialog

Account ID: 100001

Name: Alex

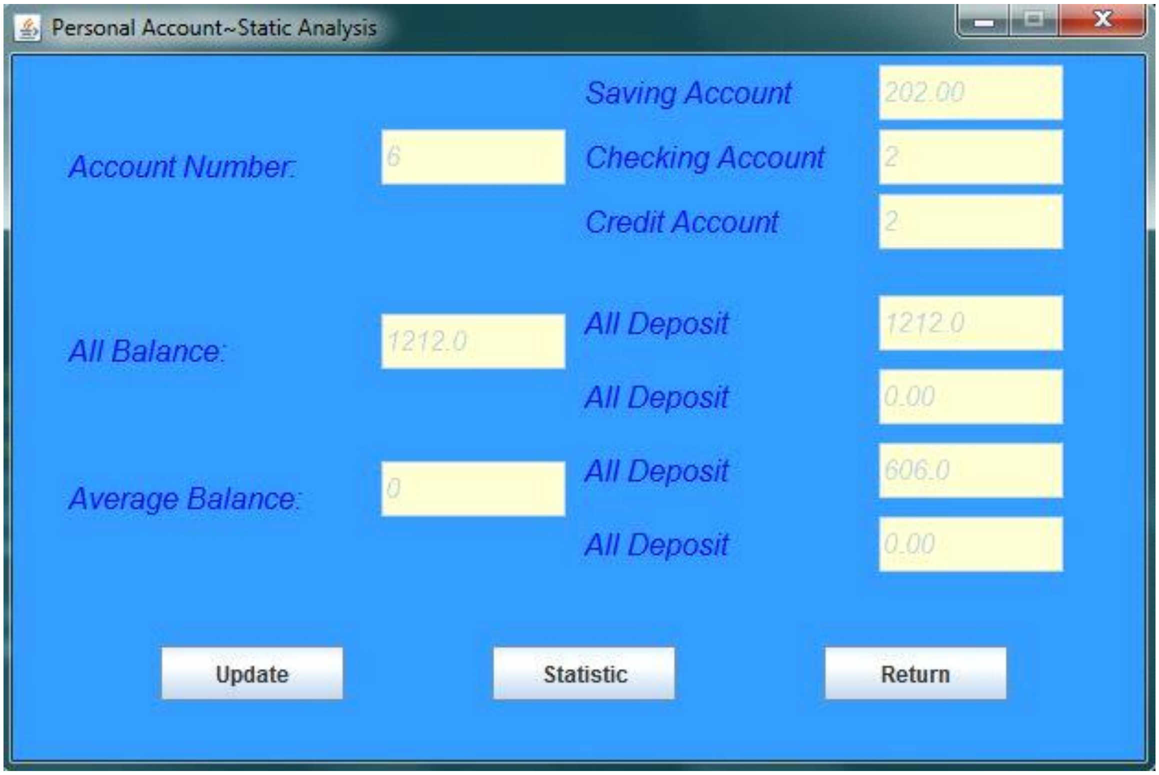
Account Type: Current

Balance: 100

Money: 100

Transfer To: 100002

Account Statistical Form:



Saving Account 202.00

Account Number: 6

Checking Account 2

Credit Account 2

All Balance: 1212.0

All Deposit 1212.0

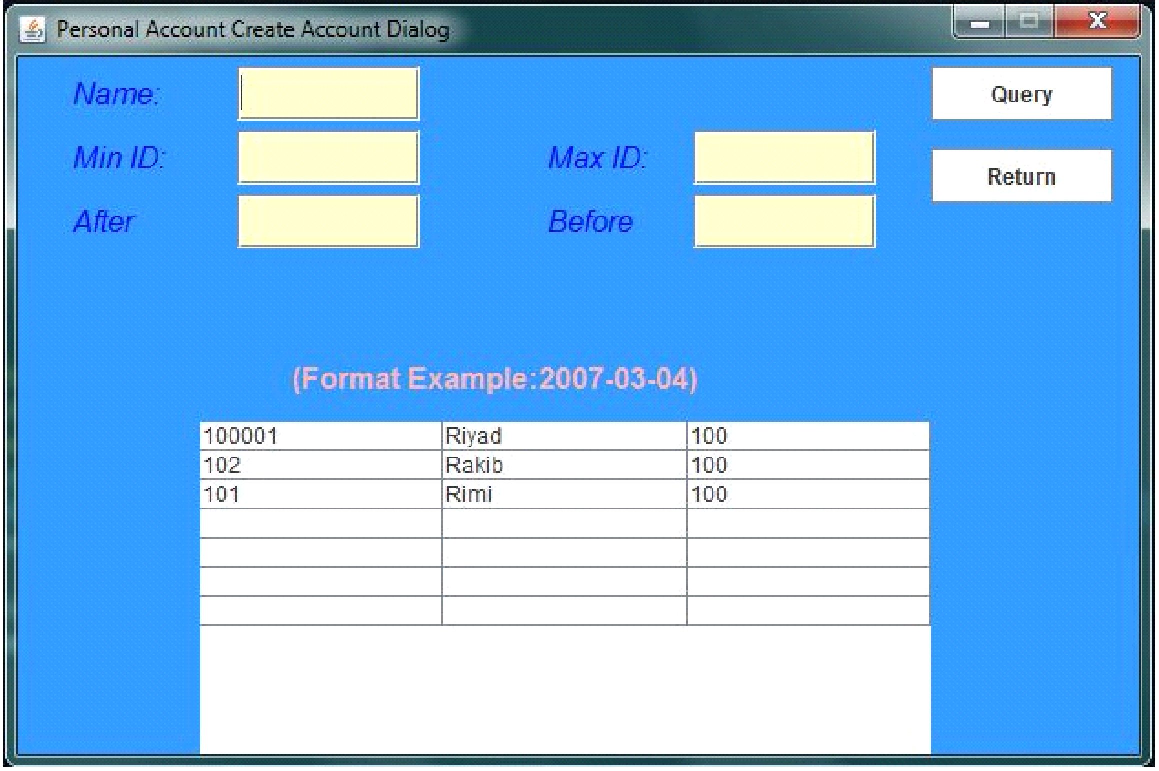
All Deposit 0.00

Average Balance: 0

All Deposit 606.0

All Deposit 0.00

Account Query Form:



Name:

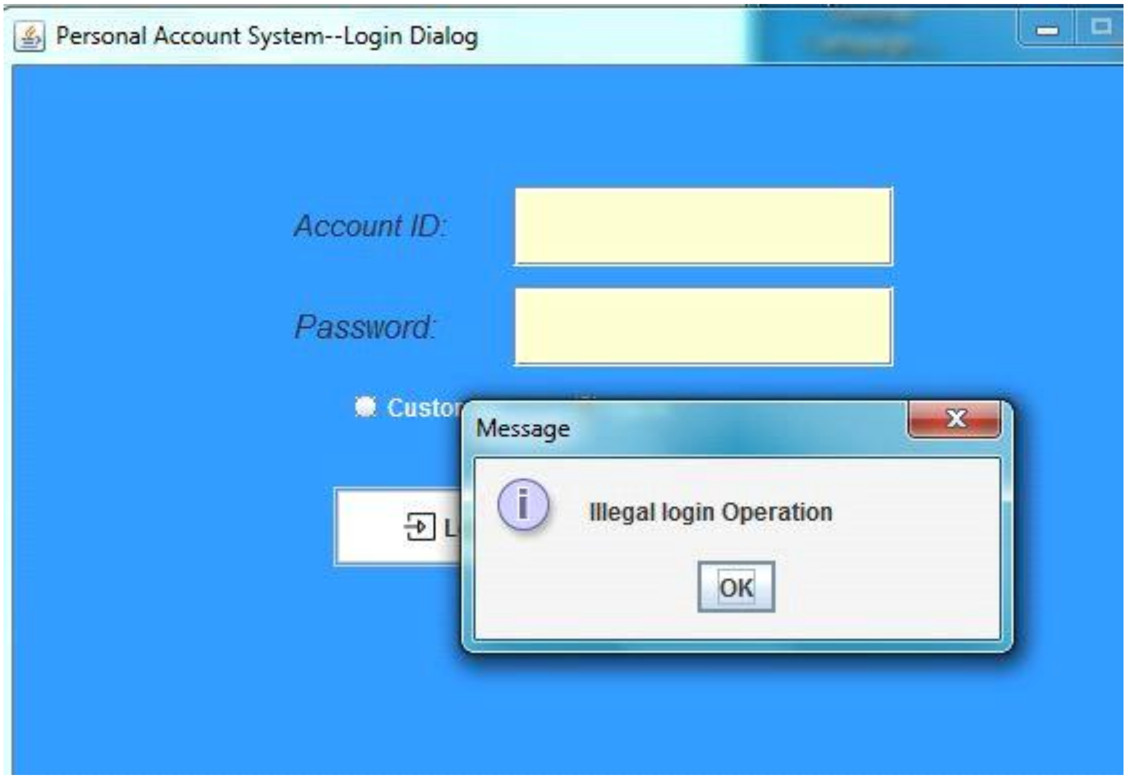
Min ID: Max ID:

After Before

(Format Example:2007-03-04)

100001	Riyad	100
102	Rakib	100
101	Rimi	100

Error Window:



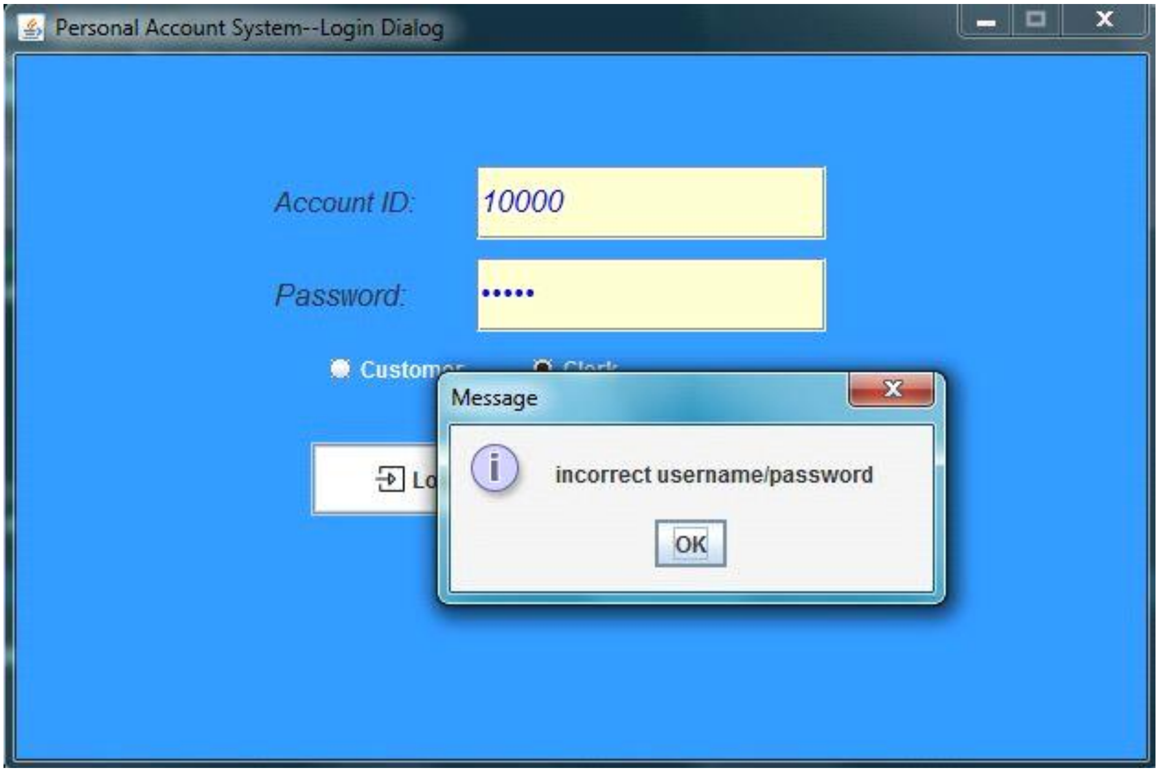
Personal Account System--Login Dialog

Account ID:

Password:

Illegal login Operation

Figure 1: Illegal Operation Warning Message



Personal Account System--Login Dialog

Account ID: 10000

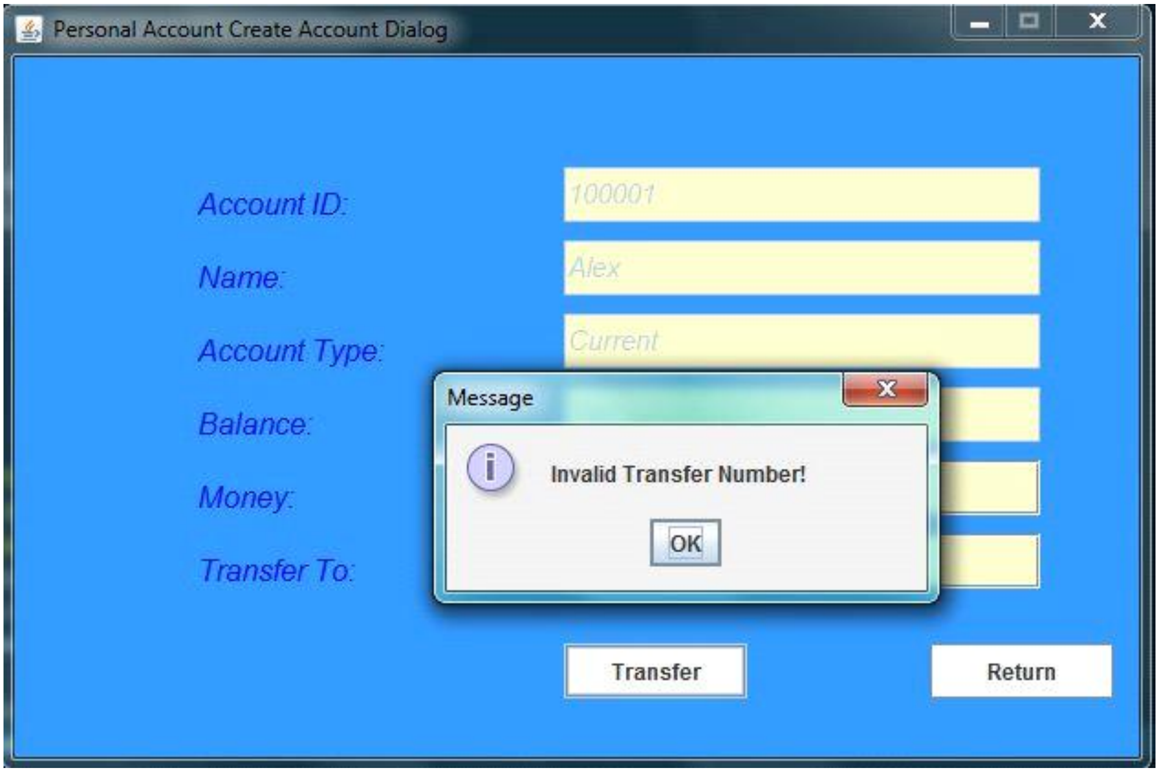
Password: •••••

○ Customer ● Clerk

Message

incorrect username/password

Figure 2: Incorrect Username and Password Warning Message



Account ID: 100001

Name: Alex

Account Type: Current

Balance:

Money:

Transfer To:

Invalid Transfer Number!

Figure 2: Invalid Transfer ID Number Warning Message

Software testing is an important part of the software development process. It is not a single activity that takes place after code implementation, but is part of each stage of the lifecycle. A successful test strategy will begin with consideration during requirements specification. Testing details will be fleshed through high and low level system designs, and testing will be carried out by developers and separate test groups after code implementation.

As with the other activities in the software lifecycle, testing has its own unique challenges. As software systems become more and more complex, the importance of effective, well planned testing efforts will only increase.

Evaluation

The tests mentions in this document are performed from 13th May through 15 June of 2021 by the author of this document who is also both the tester and the developer of this project. Below are some of the summary of the project:

- The software has complete functions. It provides functions such as login, create account, search accounts, deposit, withdraw, transfer, transaction history and other functions. It is comprehensive, reliable, and an easy-to-use software.

- It is easy to change the functions because they are relatively independent.

- The system is reliable. It has clear permission restrictions for different users, accurate error, warning, or information prompts.

- The operation is convenient and easy to comprehend. The interface of each function of the system is simple, the style is consistent and the layout is convenient for users to use.

- Based on the tests that are performed, all the output of the tests achieves the expected result.