**BANK MANAGEMENT SYSTEM**

MIDHUN M MENON

SF ID:105160

# **TABLE OF CONTENTS**

[**INTRODUCTION**](#_ogqqqtkfq5v) **2**

[1.1 Problem Statement](#_ye8pjr7x0piw) 2

[1.2 Project Description](#_qyv84uvr0932) 3

[**REQUIREMENTS**](#_4to2fu5oplqs) **3**

[2.1 Software Used](#_qwnx1h44v2bw) 3

[2.2 Header Files Required](#_ogj6w3o6dra1) 3

[2.3 Functions Required](#_fu1z1u6ucz9m) 4

[2.3.1 Functions for Bank Management System](#_9ytc3hhm1k07) 4

[2.3.2 Functions for ATM facility](#_9vi0awp5af80) 6

[**DESIGN**](#_hwb038716bf3) **7**

[3.1 Activity Diagram](#_nbffkma7u99a) 7

[3.2 Use Case Diagram](#_pb7zeqyob3bb) 8

[**TEST PLAN**](#_hwb038716bf3) **8**

[4.1 Introduction](#_i99m3tcq7322) 8

[4.2 Test Strategy](#_doi2i9vi4v8k) 9

[4.2.1 Scope Of Testing](#_2lorwzavpyzj) 9

[4.2.2 Test Type](#_eq3zy2y9hp85) 9

[4.2 Test Objective](#_kt2tz7phr8q2) 10

[4.3 Test Criteria](#_ul8au8lpso9h) 10

[4.3.1 Suspension Criteria](#_hjfft0d9ndx) 10

[4.3.2 Exit Criteria](#_hb8e5x9003a9) 10

[4.4 Test Environment](#_2mvzm6ofovv7) 10

[**TEST CASES**](#_ntxc93l78j1o) **10**

[**EXPECTED RESULTS**](#_kbqofgv4de36) **15**

[**CONCLUSION**](#_pv0w9nozrlmr) **17**

# **INTRODUCTION**

### **1.1 Problem Statement**

To create a user-friendly bank software management system using C Programming.

### **1.2 Project Description**

The purpose of this Bank management system is to help the user or customer with the daily transactions from his account. A varied number of features are included as a part of this system to help the user. The features available are listed below.

1. Login system.

2. Add, remove, and update account records.

3. View detailed individual account information.

4. Banking transactions.

5. ATM Feature.

To enter the system, the user will have to type in a mandatory password. After that he/she will be directed to the main menu section. A number of features are provided in the main menu. To create a new account, the user has to provide the current date, account number, name, date of birth, age, address, citizenship number, phone number, the amount deposited while creating an account and select account type. Five different account types are provided: Saving, Current, Fixed (1 year), Fixed (2 years), Fixed (3 years). In the second option which is updating the account details, the user is allowed to change the phone number and address. To check the account details of an already existing user, either of account number or account name is mandatory.

Whenever a user wants to withdraw or deposit some amount of money he/she has to provide account number/account name along with the amount to be deposited or withdrawn. Then the system automatically maintains the transaction record with total bank balance according to the transaction amount. And another thing is that while checking a customer’s account in detail the system will display a bank interest information. This system calculates interest per month and displays to the user.

In addition to all these features an ATM option is also added with the system. To access the ATM facility the user has to enter the ATM pin code. The ATM feature allows the user to check for Balance inquiry, deposit money and withdraw money. After accessing ATM features, the user just has to enter the amount number while depositing or withdrawing the amount.

# **REQUIREMENTS**

### **2.1 Software Used**

* IDE : Codeblocks version
* Compiler : MinGW compiler

### **2.2 Header Files Required**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Header File** | **Description** |
| RQ\_H\_01 | <stdio.h>  (Standard input-output header) | Used to perform input and output operations in C. |
| RQ\_H\_02 | <stdlib.h>  (Standard library header) | Perform standard utility functions. |
| RQ\_H\_03 | <windows.h>  (Windows specific header file) | Contains declarations for all of the functions in the Windows API. |
| RQ\_H\_04 | <stdbool.h>  (Standard library header) | Contains four macros for a Boolean data type. |

### **2.3 Functions Required**

#### **2.3.1 Functions for Bank Management System**

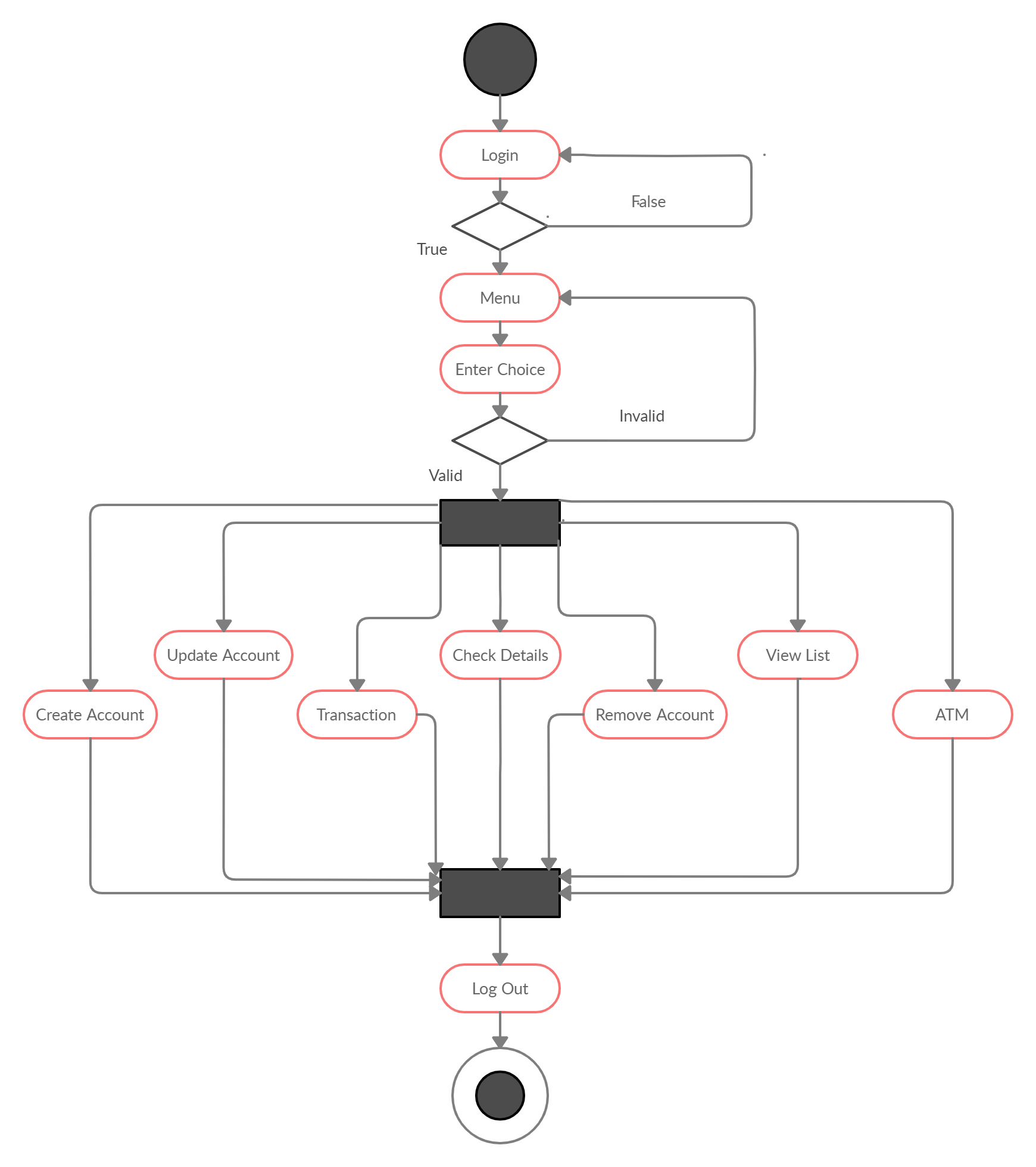
|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| RQ\_F\_01 | void menu() | This function is used to access the menu of the Bank management system. It contains various function choices that the user can choose from. |
| RQ\_F\_02 | void fordelay(int) | This function is used to add required time delay to the process. |
| RQ\_F\_03 | float interest(float, float, int) | This function is used to calculate the interest. |
| RQ\_F\_04 | void new\_acc() | This function is used to gather information to create a new transaction account. |
| RQ\_F\_05 | void view\_list() | This function is used to view the details of an existing account. |
| RQ\_F\_06 | void edit(void) | This function is used to change the phone number or address of an existing account holder. |
| RQ\_F\_07 | void transact(void) | This function is used to handle both deposit and withdraw transaction operations. |
| RQ\_F\_08 | void erase(void) | This function is used to erase data of an existing account. |
| RQ\_F\_09 | void see(void) | This function is used to see how much interest the customer will get from the amount that he has deposited. |
| RQ\_F\_10 | void close(void) | This function is used to close the program. |
| RQ\_F\_11 | int atm() | This function is used to access the ATM facility. |

#### **2.3.2 Functions for ATM facility**

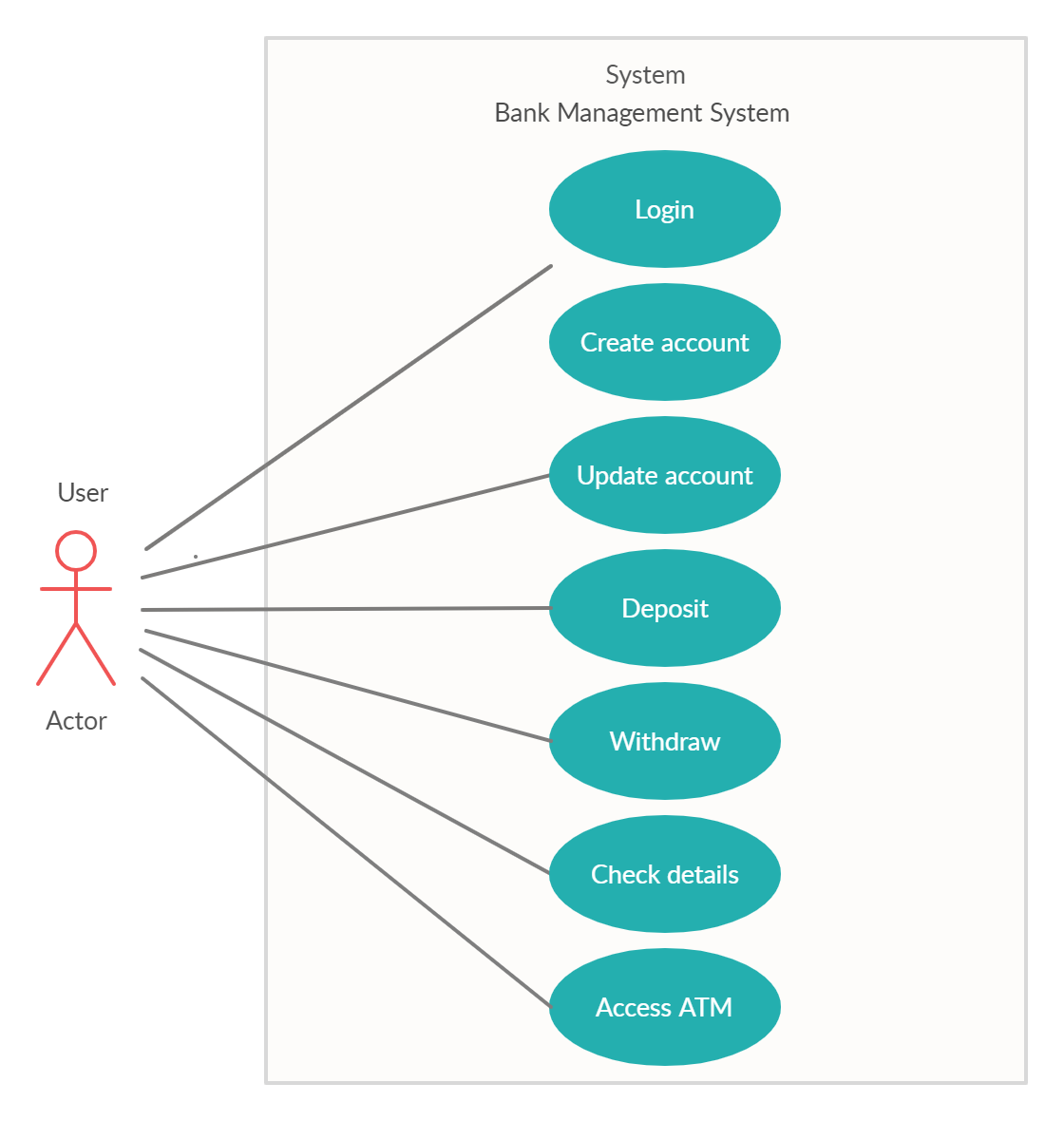
|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| RQ\_F\_12 | void mainMenu() | This function is used to access the main menu of the ATM option. |
| RQ\_F\_13 | void checkBalance(float) | This function is used to check the balance of the existing account. |
| RQ\_F\_14 | float moneyDeposit(float) | This function is used to deposit money into the existing account. |
| RQ\_F\_15 | float moneyWithdraw(float) | This function is used to withdraw money from the existing account. |
| RQ\_F\_16 | void menuExit() | This function is used to exit from the ATM feature. |
| RQ\_F\_17 | void errorMessage() | This function is used to show error messages if the entered pin number is wrong. |
| RQ\_F\_18 | int login(void) | This function is used for the login purpose of the ATM. |

# **DESIGN**

### **3.1 Activity Diagram**



### **3.2 Use Case Diagram**



# **TEST PLAN**

### **4.1 Introduction**

Test plan is designed to advise the scope, approach and strategy of testing an application, to identify risks and issues, to define test criteria and test environment, and to identify the type of testing to be performed on the Application Under Test(AUT).

### **4.2 Test Strategy**

#### **4.2.1 Scope Of Testing**

**Features to be tested**

|  |  |
| --- | --- |
| **Module** | **Description** |
| Login | User and admin can log into the system |
| Create Account | User can create a new account by entering the required details |
| Update Account | Existing user can update his account details |
| Transactions | Users can deposit and withdraw money from their account |
| Check details | User can check the account details of their account |
| Remove Account | User can remove their existing account |
| View customer list | Bank admin can access the overall customer list |
| ATM | User can access the ATM facility |

**Features not to be tested**

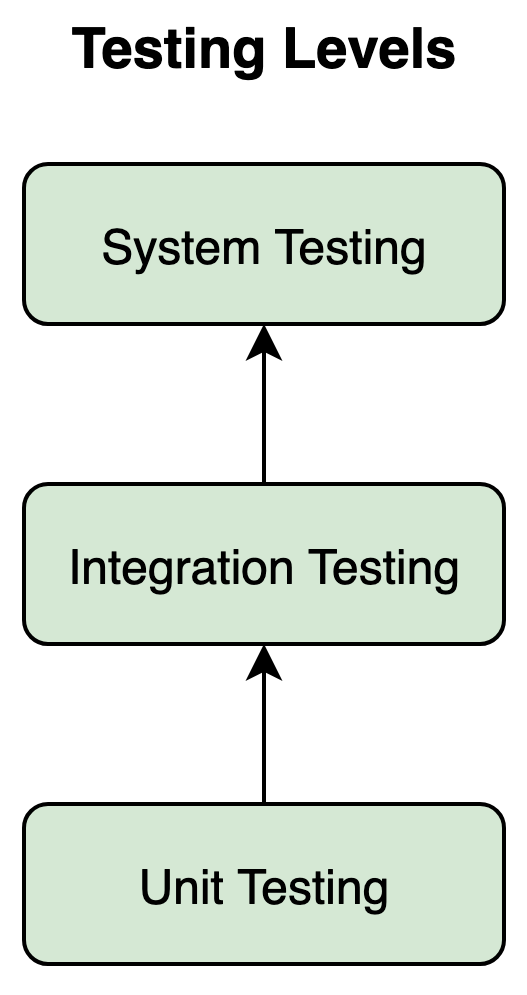
These features are not to be tested because they are not included in the software requirement specifications.

* User Interfaces
* Hardware Interfaces

#### **4.2.2 Test Type**

In this project there are three types of testing to be conducted:

* **Unit Testing :** It is a type of software testing where individual units or components of a software are tested.
* **Integration Testing** **:** It is a type of software technique where individual software modules are integrated and tested as a group.
* **System Testing :** It is a level of testing that validates the complete and fully integrated software product. It is done to check whether the software meets specified requirements or not.



Unit tests help to fix bugs early in the development cycle. Unit testing is of two types :

* Manual
* Automated

Under automated approach the developer writes a section of code in the application just to test the function. It is later commented out and finally removes the test code when the application is deployed.

### **4.2 Test Objective**

The test objectives are to verify the different functionalities of the Application Under Test (AUT). The project should focus on testing operations such as entering, editing, deleting, storing and searching data to make it work without errors on an actual environment.

### **4.3 Test Criteria**

#### **4.3.1 Suspension Criteria**

If more than 40% of test cases are failed then the testing is suspended until all failed cases are fixed.

#### **4.3.2 Exit Criteria**

This denotes a successful completion of a test phase.

* It is mandatory that the run rate is 100%.
* Pass rate is 95%.

### **4.4 Test Environment**

All test cases are carried out in Code Blocks IDE version 20.03.

# **TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** |
| TU01 | Check Login with a valid password. | 1. Enter password  2. Click Enter. | password = ‘pass’ |
| TU02 | Check Login with an invalid password | 1. Enter password  2. Click Enter. | password = ‘user’ |
| TU03 | Check whether the create a new account option in main menu is working | 1. Enter choice  2. Click Enter | choice = 1 |
| TU03\_01 | Check if new account details can be successfully added. | 1. Enter the details  2. Click ‘1’ to go to the main menu or ‘0’ to exit. | account\_number= 123456789  name= “userone”  date\_of\_birth= 25/03/1998  age= 22  address= “houseone”  citizenship\_number= 1234  phone\_number=9449999999  amount\_to\_deposit= 1000  type\_of\_account=”saving” |
| TU04 | Check whether the ‘check the details of existing account’ option in main menu is working | 1. Enter choice  2. Click Enter | choice=4 |
| TU04\_01 | Check if the existing account details can be viewed by entering account number. | 1. Enter choice 2. Click Enter 3. Enter account number 4. Click Enter 5. Click ‘1’ to go to the main menu or ‘0’ to exit. | choice=1  account\_number=  123456789 |
| TU04\_02 | Check if the existing account details can be viewed by entering an invalid account number. | 1. Enter choice 2. Click Enter 3. Enter invalid account number 4. Click Enter 5. Click ‘1’ to go to the main menu or ‘0’ to exit | choice=1  account\_number=  999999999 |
| TU04\_03 | Check if the existing account details can be viewed by entering account name. | 1. Enter choice 2. Click Enter 3. Enter account name 4. Click Enter 5. Click ‘1’ to go to the main menu or ‘0’ to exit | choice=2  account\_name=  “userone” |
| TU04\_04 | Check if the existing account details can be viewed by entering an invalid account name. | 1. Enter choice 2. Click Enter 3. Enter invalid account name 4. Click Enter 5. Click ‘1’ to go to the main menu or ‘0’ to exit | choice=2  account\_name=  “abcdef” |
| TU05 | Check whether the ‘View customer’s list’ option in main menu is working | 1. Enter choice 2. Click Enter 3. Click ‘1’ to go to the main menu or ‘0’ to exit. | choice=6 |
| TU06 | Check whether the ‘Transactions’ option in main menu is working | 1. Enter choice 2. Click Enter 3. Enter account number | choice=3  account\_number=  123456789 |
| TU06\_01 | Check whether the ‘deposit’ option in Transactions section is working | 1. Enter choice 2. Click Enter 3. Enter amount to deposit 4. Click ‘1’ to go to main menu or ‘0’ to exit | choice=1  amount\_to\_deposit=1000 |
| TU06\_02 | Check whether the ‘withdraw’ option in Transactions section is working | 1. Enter choice 2. Click Enter 3. Enter amount to withdraw 4. Click ‘1’ to go to main menu or ‘0’ to exit | choice=2  amount\_to\_withdraw= 500 |
| TU06\_03 | Check whether the ‘Transactions’ option in the main menu is working with an invalid account number. | 1. Enter choice 2. Click Enter 3. Enter invalid account number | choice=3  account\_number=  999999999 |
| TU07 | Check whether there is change in account details after transaction by accessing ‘check the details of existing account’ | 1. Enter choice 2. Click Enter 3. Enter account number 4. Click ‘1’ to go to the main menu or ‘0’ to exit. | choice= 4  account\_number=  123456789 |
| TU08 | Check whether the ‘Update information of an existing account’ option in the main menu is working | 1. Enter Choice 2. Click Enter | choice=2 |
| TU08\_01 | Check whether address of an existing account can be changed | 1. Enter choice 2. Click Enter 3. Enter account number 4. Click Enter 5. Enter new address 6. Click ‘1’ to go to the main menu or ‘0’ to exit. | choice=1  account\_number=  123456789  new\_address= “house” |
| TU08\_02 | Check whether phone number of an existing account can be changed | 1. Enter choice 2. Click Enter 3. Enter account number 4. Click Enter 5. Enter new phone number 6. Click ‘1’ to go to the main menu or ‘0’ to exit. | choice=2  account\_number=  123456789  new\_phone\_number= 9999999999 |
| TU08\_03 | Check whether the ‘Update information of an existing account’ option in the main menu is working with an invalid input | 1. Enter Choice 2. Click Enter | choice=9 |
| TU09 | Check whether the details have been updated by checking the ‘View customer list’ option | 1. Enter Choice 2. Click Enter | choice=6 |
| TU10 | Check whether the ATM feature is working | 1. Enter Choice 2. Click Enter 3. Enter PIN number 4. Click Enter | choice=7  PIN number=1234 |
| TU10\_01 | Check whether the ‘Check the balance’ option of‘ ATM is working | 1. Enter Choice 2. Click Enter 3. Click Y/N for ‘Would you like another transaction’ | choice=1  choice=1 |
| TU10\_02 | Check whether the ‘Deposit’ option of‘ ATM is working | 1. Enter Choice 2. Click Enter 3. Enter amount to deposit 4. Click Y/N for ‘Would you like another transaction’ | choice=2  amount\_to\_deposit=  100  choice=1 |
| TU10\_03 | Check whether the ‘Withdraw’ option of‘ ATM is working | 1. Enter Choice 2. Click Enter 3. Enter amount to withdraw 4. Click Y/N for ‘Would you like another transaction’ | choice=3  amount\_to\_withdraw= 100  choice= 2 |
| TU10\_04 | Check whether the ‘Exit’ option of‘ ATM is working | 1. Enter Choice 2. Click Enter | choice=4 |

# **EXPECTED RESULTS**

|  |  |
| --- | --- |
| **Test Case ID** | **Expected Results** |
| TU01 | Login should be successful and output window should show main menu |
| TU02 | Login should be unsuccessful and password should be asked again |
| TU03 | Entry should be accepted and a screen showing different variables needed to open a new bank account should appear. |
| TU03\_01 | Entry should be accepted, all details should be collected and finally a message which says ‘Account created successfully’ must appear on the screen. |
| TU04 | Entry should be accepted and a screen should appear showing a query to enter either of account number or account name. |
| TU04\_01 | Entry should be accepted and account details should be shown. |
| TU04\_02 | Entry should not be accepted and should ask for re-entry |
| TU04\_03 | Entry should be accepted and account details should be shown. |
| TU04\_04 | Entry should not be accepted and should ask for re-entry |
| TU05 | Entry should be accepted and a screen showing existing customer list and details should appear. |
| TU06 | Entry should be accepted and a screen should appear with a query asking the user to choose from either deposit or withdraw option |
| TU06\_01 | Entry should be accepted, deposit amount should be accepted and finally a message which says ‘Deposited Successfully!’ should appear. |
| TU06\_02 | Entry should be accepted, deposit amount should be accepted and finally a message which says ‘Withdrawn Successfully!’ should appear. |
| TU06\_03 | Entry should not be accepted and should ask for re-entry of account number. |
| TU07 | Entry should be accepted and a screen should appear showing the details of the existing account. |
| TU08 | Entry should be accepted and a screen should appear with a query asking the user to select either of address or phone number to be updated |
| TU08\_01 | Entry should be accepted,a new address should be collected and a message which says ‘Changes saved!’ should appear on the screen. |
| TU08\_02 | Entry should be accepted,a new phone number should be collected and a message which says ‘Changes saved!’ should appear on the screen. |
| TU08\_03 | Entry should not be accepted and should ask again for a valid input. |
| TU09 | Entry should be accepted and a screen should appear showing the existing customer list. |
| TU10 | Entry should be accepted and a screen should appear showing different features available in the ATM option. |
| TU10\_01 | Entry should be accepted and a screen should appear showing the available balance of the corresponding account, and an option to do another transaction if necessary. |
| TU10\_02 | Entry should be accepted and a screen should appear showing the amount of money deposited, and an option to do another transaction if necessary. |
| TU10\_03 | Entry should be accepted and a screen should appear showing the amount of money withdrawn, and an option to do another transaction if necessary. |
| TU10\_04 | Entry should be accepted and the Menu screen should appear. |

# **CONCLUSION**

This bank software management system helps in the effective management of the user’s bank account. Through this software any new user can create their new bank account and the already existing users can update, remove and check details of their bank account. It also provides the user an effective way to deposit and withdraw money. An added ATM feature makes it a complete banking software. It provides a simple user interface, which can be easily understood by anyone.