**Mini Project**

**On**

**ATM Transaction Program**

**Submitted by: Mehulkumar P Jain**

**Candidate ID: 105387**

**Table of Contents**

1. Introduction ………..………………………………………………………...… 1
2. Problem Statement …………………………………………………………..… 2
3. Description …………………………………………………………………….. 2
4. Requirements ………………………………………………………………….. 2
5. Flow Chart ………………………………………………………...…………… 3
6. Test Plan ………………………………………………………………….……. 4
7. Test Cases ……………………………………………………………………… 4
8. Expected Results ………………………………………………………………. 8
9. Conclusion …………………………………………………………………… 10
10. Reference ……………………………………………………………………. 10

**List of Figure**

1. Flow chart of ATM Transactions ……………………………………………… 3
2. **Introduction**

An ATM (Automated Teller Machine) is an electronic telecommunication device that enables customers of different banks or financial institutions to perform financial transactions.[1] The financial transaction could be balance enquiry, cash withdrawals, cash deposit and etc.., at any time without need for direct interaction with bank or financial institution staff. The ATM has variety of names in different country’s such as in Canada it is known as ABM (Automated Banking Machine), in British English, the terms cashpoint, cash machine are different names used for the ATM.

ATM are very helpful because one can make any bank transaction at any time. Even the ATMs can also be used to withdraw money in a foreign country. If the bank account is dominated with the different currency then the currency being withdrawn from ATM will be converted by exchange rate at bank or financial institutions. The Customers are typically identified whenever they insert the ATM card (or some other acceptable payment card) into the ATM machine. The customer is authenticated by entering a PIN (Personal Identification Number), which should match with the PIN stored in bank’s (financial institution’s) database or PIN stored on the card chip.

The typical hardware device inside the ATM machines are CPU, magnetic or chip card reader, a PIN pad, secure cryptoprocessor, display, functional keys buttons, touch screen, record printer, vault (to store the parts of machinery requiring restricted access), housing (for aesthetics and to attach signage to) and sensors and indicators. The software used in ATMs are standard commercial “off-the-shelf” OS and different programing environments.[1]

The main objective of the proposed system is to enhance and upgrade the existing system by increasing its security. In the proposed system two way security is provided that user need to enter PIN and OTP (which will be sent to registered mobile number) for authentication of the customer while doing the transaction. The OTP facility does not allow the misuse of the card if it stolen or lost. If any other person who get the card details and try to do transaction, will not be able to do as the OTP will be sent to registered mobile number and the customer can inform to the bank or financial institution immediately.

1. **Problem Statement**

Perform ATM transaction such as balance checking, cash withdraw and cash deposition when user enters the correct PIN and OTP.

1. **Description**

In the existing ATM the user insert the ATM card and enter the PIN and then do the transaction but if the some other person get the card and its details then it could be easily misused. To avoid this a OTP facility is added in this project that is whenever user want to do transaction along with PIN, OTP sent to registered mobile number need to be entered, which provide the double security that even if the card is lost or stolen it cannot be misused. As the proposed model is completely software card insert option is not provided but instead the user name is asked. The PIN, OTP and account type are stored as array of strings for the respective users. Initially the five users name, PIN, OTP and account type are stored. For cash withdraw, deposit, balance check the different functions are written accordingly.

1. **Requirements**

Software requirement: CodeBlocks IDE.

Operating System requirements: Windows 7 and higher versions.

Functional and other requirements: The user details such as name, PIN, OTP and account type are stored as array of strings. The various functions required in the project are:

* **Balance( ):** The Balance( ) function is used to display the current balance in the user account.
* **Withdraw( ):** The Withdraw( ) function is used to withdraw amount form the account. The Withdraw( ) function ask user to enter the amount to withdraw. Here the user need to enter the amount in the multiple of 100. If the user enter the amount which is not multiple of 100 then the function displays the message “Please Enter The Amount In Multiples Of 100”. The minimum amount need to be maintained in the account is 1000. If the user enters the amount which is greater than initial amount – 1000, then the function displays Insufficient Balance. If the above two criteria are met then amount is withdraw successfully.
* **Deposit( ):** The Deposit( ) function asks the amount to be deposited. Again the user need to entre deposit amount in multiples of 100 and entered amount by the user is get added to current balance.
* **Quit( ):** The Quit( ) function is used to end the transaction.

1. **Flow Chart**

The below figure shows the flow chart for the ATM transactions.

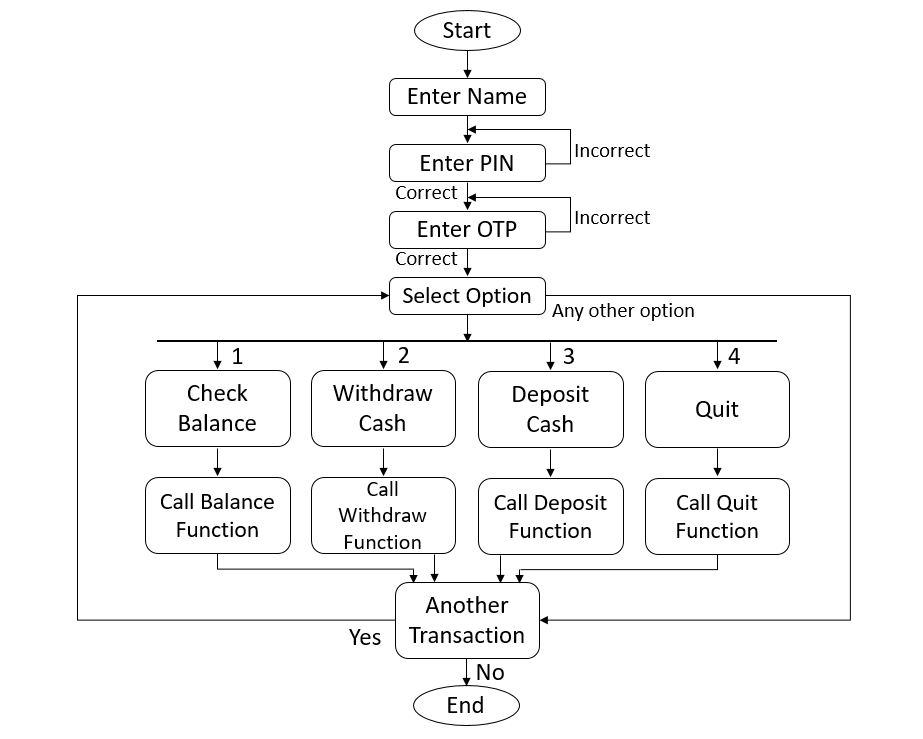


Figure 1. Flow chart of ATM transactions

1. **Test Plan**

|  |  |
| --- | --- |
| **Test Steps** | **Test Scenario** |
| 1 | Check whether user entered correct name or not. |
| 2 | Check whether user entered correct PIN or not. |
| 3 | Check whether user entered correct OTP or not. |
| 4 | Check whether user selected correct option or not. |
| 5 | Check whether user entered correct account type or not. |
| 6 | Check user entered withdrawal amount is multiples of 100 or not and is less than initial balance - 1000 and verify the amount is deducted correctly. |
| 7 | Check user entered deposit amount is multiple of 100 or not and verify the amount is added to initial balance correctly. |
| 8 | Check user wish to have another transaction or not. |

1. **Test Cases**

To understand the test cases in better way considering an example in which user name, PIN, OTP, account type are stored in array of strings as follows:

* Names = {"xyz", "abc", "zyx", "pqr", "efg"};
* PIN = {"1234", "2345", "3456", "4567", "5678"};
* OTP = {"12345", "23456", "34567", "45567", "56789"};
* Account type = {"savings", "savings", "current", "current", "savings"};

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 1 | Check user entered valid name | Enter the name: xyz | User is asked to enter PIN | As expected | Pass |
| Enter the name: roy | User is asked to enter PIN | User asked to enter name again | Fail |

In above test step 1, “xyz” is found in array of strings so next user is asked to enter PIN which is as expected so test case passes and “roy” is not found in array of strings so test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 2 | Check user entered valid PIN | Enter Your PIN: 1234 | User is asked to enter OTP | As expected | Pass |
| Enter Your PIN: 4567 | User is asked to enter OTP | User asked to enter PIN again | Fail |

In above test step 2, user entered PIN “1234” is found correct for the name xyz in the array of string so the test case passes and PIN “4567” is incorrect so the test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 3 | Check user entered valid OTP | Enter Your OTP: 12345 | User is asked to select option | As expected | Pass |
| Enter Your PIN: 98765 | User is asked to select option | User asked to enter OTP again | Fail |

In above test step 3, user entered OTP “12345” is found correct for the name xyz in the array of string so the test case passes and OTP “98765” is incorrect so the test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 4 | Check user entered correct option | Select the option: 1 or 2 or 3 or 4 | User is asked to enter account type | As expected | Pass |
| Select the option: (any option other than 1, 2, 3 and 4) | User is asked to enter account type | Display “Invalid Option” | Fail |

In above test step 4, if the user select the option 1 or 2 or 3 or 4 then user asked to enter account type which is as expected so test case passes and if the user select any other option the expected and actual results does not match and test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 5 | Check user entered correct account type | Enter your account type: savings | Depending on option selected call appropriate function | As expected | Pass |
| Enter your account type: current | Depending on option selected call appropriate function | User asked to enter account type again | Fail |

In above test step 5, user entered account type “savings” is found correct for the name xyz in the array of string so the test case passes and account type “current” is incorrect so the test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 6 | Check user entered withdrawal amount is multiple of 100 and less then initial balance - 1000 | Enter amount to withdraw: 1000 | Entered amount is deducted from balance and display the current balance | As expected | Pass |
| Enter amount to withdraw:  -150 | Entered amount is deducted from balance and display the current balance | Display “Enter the amount in multiples of 100” | Fail |
| Enter amount to withdraw:  6000 | Entered amount is deducted from balance and display the current balance | Display “Insufficient Balance” | Fail |

In above test step 6, user entered withdrawal amount 1000 which is deducted from current balance and display the current balance which is as excepted so test case passes and amount -150 is not multiple of 100 the test case fails. Assuming initial balance as 5000, the withdrawal amount entered 6000 is greater than initial balance - 1000 so the test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 7 | Check user entered deposit amount is multiple of 100 | Enter amount to deposit: 500 | Entered amount is added to initial balance and display the current balance | As expected | Pass |
| Enter amount to deposit: -80 | Entered amount is added to initial balance and display the current balance | Display “Enter the amount in multiples of 100” | Fail |

In the test step 7, user entered amount 500 is added to initial balance which is as expected so test case passes and amount -80 is not multiple of 100 so test case fails.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Scenario** | **Test Data** | **Expected Results** | **Actual Results** | **Pass / Fail** |
| 8 | Check user wish to have another transaction | Enter ‘y’ or ‘Y’ for yes and ‘n’ or ‘N’ for no | If yes, repeat the steps from select option and if no, cancel the transaction | As expected | Pass |

1. **Expected Results**

The expected results from the ATM transaction program are as follows by considering example:

Name: abc

PIN: 2345

OTP: 23456

Type: savings

* First the user is asked to enter name, the entered name is compared in the names that are stored in array of strings and corresponding index value is retuned. For the example considered i.e., for name abc, if the index value is 1, then 1 is returned or stored in another variable.
* Next the user is asked to enter PIN, the entered PIN (2345) is compared in PINs that are stored in array of strings for index position 1. If the PIN entered (2345) match with the PIN in position 1 of array the next user is asked to enter the OTP, otherwise display the message “Please Enter the Valid PIN” and ask to enter PIN again.
* Next the user us asked to enter OTP, the entered OTP (23456) is compared in OTPs that are stored in array of strings for index position 1. If the OTP entered (23456) match with the OTP in position 1 of array then next step is executed, otherwise display the message “Please Enter the Valid OTP” and ask to enter OTP again.
* Next the user is asked to enter the option 1 for Check Balance, 2 for Withdraw Cash, 3 for Deposit Cash and 4 for Quit. Is the user enter the other option then “Invalid Option” message is displayed.
* Next the user is asked to enter the account type i.e., savings or current. Here the entered type (savings) is matched with position 1 of array of account type that are stored in array of strings. If match is correct the next step is executed otherwise user is asked to enter the account type again.
* If the option is 1, the Balance( ) function is called to display the current balance of the user.
* If the option is 2, the Withdraw( ) function is called and ask user to enter amount to withdraw. If the user enter the amount which is not in multiples of 100, then display the message “Please Enter The Amount In Multiples Of 100”. If the entered amount is in multiple of 100 and is lesser than initial balance - 1000, then amount is withdraw successfully and display two messages “Please Collect Your Cash” and “Your Current Balance is: 4500” (if initial balance is 5000 and withdraw amount entered is 500). If the amount is greater than initial balance - 1000 “Insufficient Balance” message is displayed because the minimum balance need to maintained is 1000.
* If the option is 3, the Deposit( ) function is called and ask the user to enter the amount to be deposited in multiples of 100. If the entered amount is multiple of 100 then it get added to initial balance and displayed on the screen “Your Current Balance Is: 6000” (if the deposit amount is 1500), otherwise display the message “Please Enter the amount in multiples of 100”.
* If the option is 4, the Quit( ) function is called which cancel the last transaction and display the message “Your Transaction Is Canceled”.
* Next the user is asked for another transaction. If the user enter char ‘y’ or ‘Y’ the procedure repeats from “enter option” step, if the user enter char ‘n’ or ‘N’ then the transaction is canceled and “Thanks For Using Our ATM Service” message is displayed.

1. **Conclusion**

The proposed model provides two way security i.e., authentication by PIN and OTP for customers doing ATM transaction. The existing system has only PIN facility. If the card is stolen or lost and any other person get the card and its details then it can be easily misused. So avoid this the proposed model is developed with OTP facility, in which the OTP is sent to registered mobile number of customer so any other person will not be able to do transaction. Here in this project the OTP are predefined for users in array of strings. Whenever the users enters the OTP, it is compared in array of string for that corresponding user and check whether it correct or not. If correct then proceeded with next step otherwise user is asked to enter OTP again.

**10. Reference**

[1] <https://en.wikipedia.org/wiki/Automated_teller_machine>

[2] <https://sourceforge.net/projects/codeblocks/>

[3] <https://www.guru99.com/test-case.html>