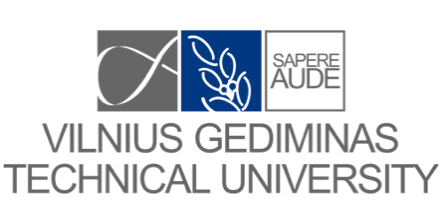
|  |
| --- |
| VILNIUS GEDIMINAS TECHNICAL UNIVERSITY |
| TOPIC |
| Banking System implementation in C++ |
| Author: Yash Puthran, ITFUC-19 |
| VILNIUS 2020 |



# Description:

My project is an attempt to recreate almost all or most basic functionalities a bank has to offer to their customers. Customers can open accounts, deposit money, withdraw money, send money, edit their details, take out loans, check or even improve their credit ratings, etc.

Git repository link: <https://github.com/nottoobright/MyBank>

# Main program:

The main program consists of various functions which involve an auth function, a menu function, a main function, various functions to implement CRUD functionalities. The code was written from the perspective of a bank employee and not a customer, so the employee can add accounts, edit them, deposit or withdraw or send money, approve loans, check credit ratings, etc.

# Libraries

The libraries I used were

1. Cctype – For character handling and transforming
2. Iomanip – For all input/output operations
3. Fstream – Same reason as above
4. STL – For some pre-built functions and data types like strings and vectors.

# Classes

1. Account – The account class is the main and the longest class. It is also the parent to to the loan class as loans are associated with accounts. This class consists every detail of the account holder, from their name to balance, loaned amount, their credit rating, etc. It follows object-oriented principles and all data is encapsulated.
2. Loan – The loan class is the child of the account class. It contains limited variables or amount. I had thought of implementing a rate of interest as well but did not. The loan class approves or disapproves depending upon whether the customer has at least 15% of the loan amount he’s asking in his account and his credit rating is more than 5(C).
3. Transaction – This class records each transaction between people. It contains a transactions vector which contains the sender, receiver, and the amount that was transferred between those two people.
4. Auth – The auth class is a very basic implementation of authentication. It saves the username and password in an unencrypted “.txt” file named after the user. It writes and reads from their to reconfirm details for the login.

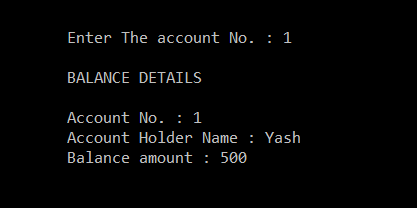
# LIMITATIONS

1. When implementing the loan functionality, the credit rating is reduced by 3 points when the loan is issued. This prevents the customer from accumulating additional debt, and he can improve his rating by doing timely repayments (+1)\_or repaying the entire loan (+3). The limitation of this model is that if a loan of even $1 was to be approved, his rating would still go down 3 points.
2. The login credentials aren’t being encrypted or saved in a binary file, mostly for the sake of convenience in case I forgot the password instead of just going through the registration again.

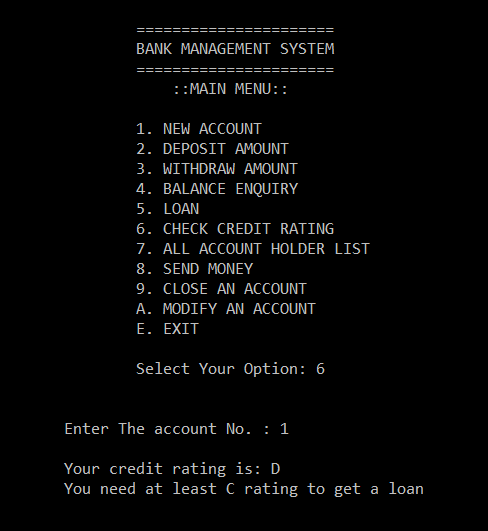
# Initial data inputs

1. Login/Register – Basic auth where you have to set username and password.
2. Create account
3. Account number – Stores the account number
4. Holder name – Saves name of the account holder
5. Type of account – Whether it’s a savings account or a current account.
6. Initial amount – Initial deposit to the account

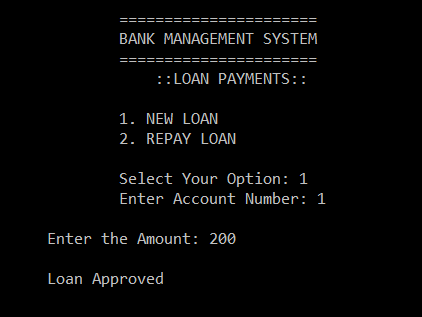
# Program results



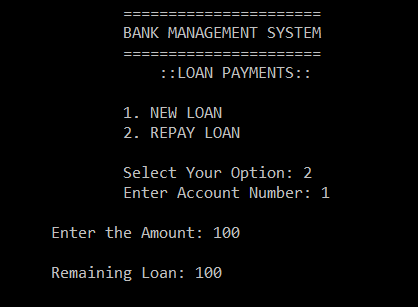
This output shows the balance of each account.



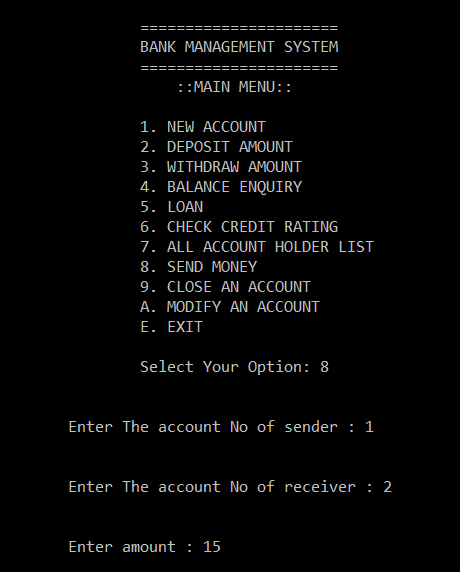
This output shows the credit rating of any account



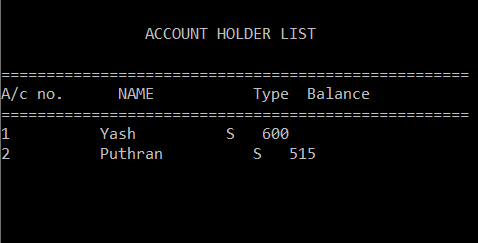
This output shows the approved loan menu.

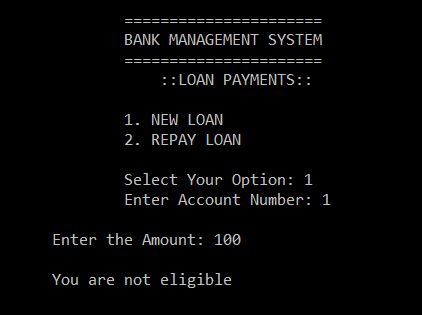


This output shows the repayment loan screen.



This output shows the send payment options between two accounts.



This output shows the updated balance after the send transaction was recorded.

This output shows loan being rejected because of low credit rating or low account balance.