

Project Report

Drive Link:

<https://drive.google.com/file/d/112MvKCtdWsfFCtsQgeWs9yVzLxCpbvmR/view?usp=sharing>

Introduction:

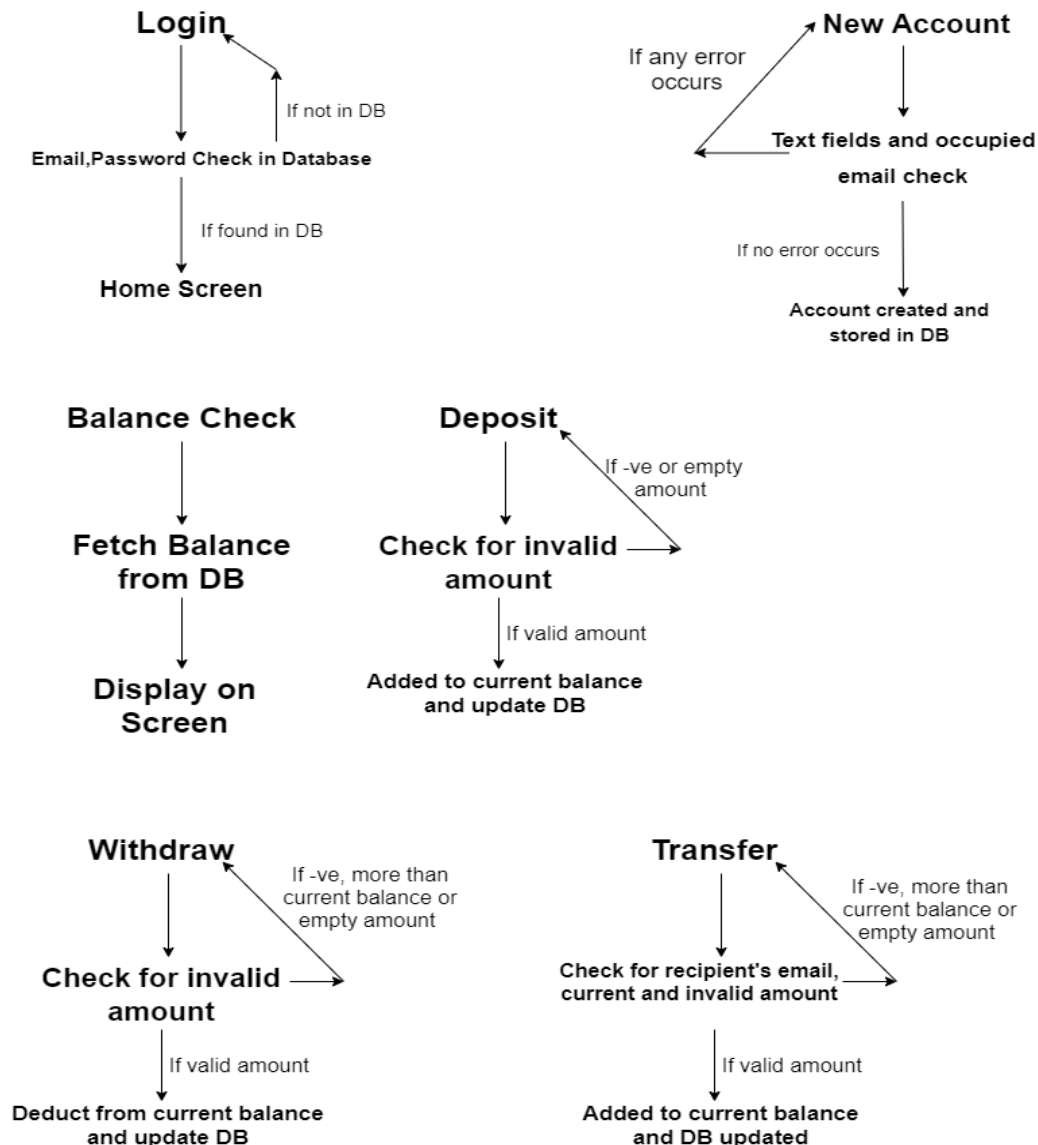
This application is a simple representation of a bank account management system with following features:

- Creation of a new account.
- Deposition, withdrawal and transfer of money.
- Password reset in case user forgets his password.
- Account statement

The application is built on Microsoft's .NET Framework (version 4.7.2) and written in C#. It makes use of Microsoft's SQL LocalDB service for storing data related to accounts.

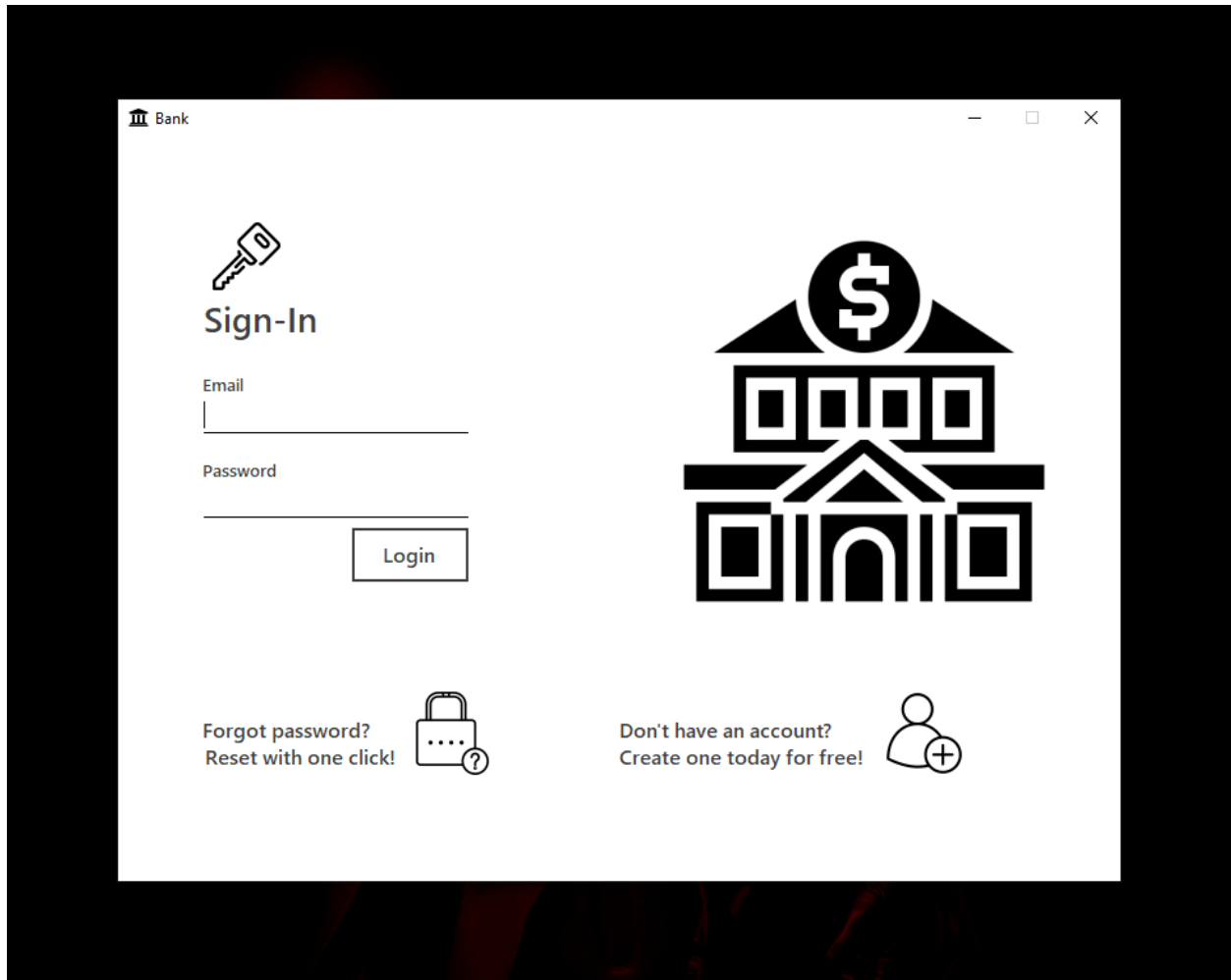
Flow of The Program:

Below is the flow of the program based on the feature chosen by the user.



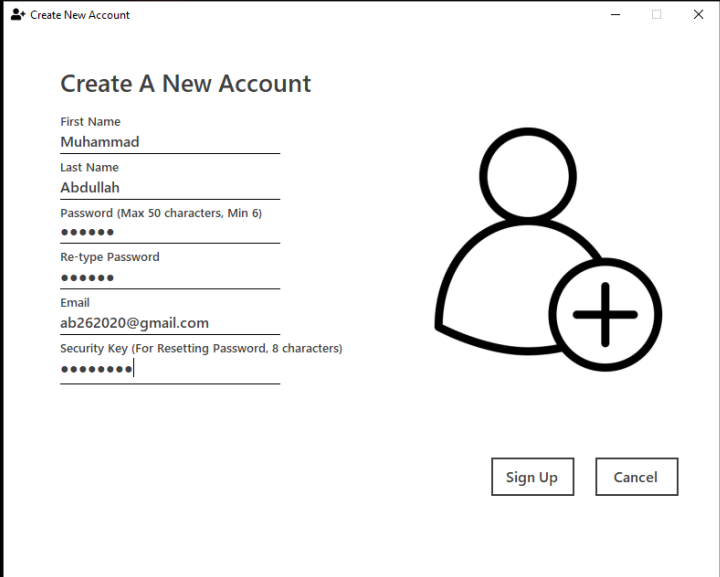
Start-up:

Upon launching, the first screen is the login form. Here, you'll find the features such as logging in to your account, creating a new account or resetting the password of your account if you've forgotten it.



New Account:

Clicking on the new account button will take the user to account creation form. In the code, it is named 'Newusers'. The user can type his name, password, email and security key in their respective text boxes. Clicking on the Sign Up button will cause the program to call the 'addacc_m' method from the 'Account' class. The information given by the user will be passed as arguments to the 'addacc_m' method. This method will perform a validity check, through 'checkinfo' method in 'DataChk' class, on the given data if the name is typed in correct format, length of password, if password matches with the one typed in 'Re-type password' text box or if any if the text boxes is empty. If no error occurs, the 'addacc_db' method in the 'DB_access' class is executed which performs a query against the local database and stores the information in it and a message for success is displayed after which the user is taken back to the login form.



Create New Account

Create A New Account

First Name
Muhammad

Last Name
Abdullah

Password (Max 50 characters, Min 6)
.....

Re-type Password
.....

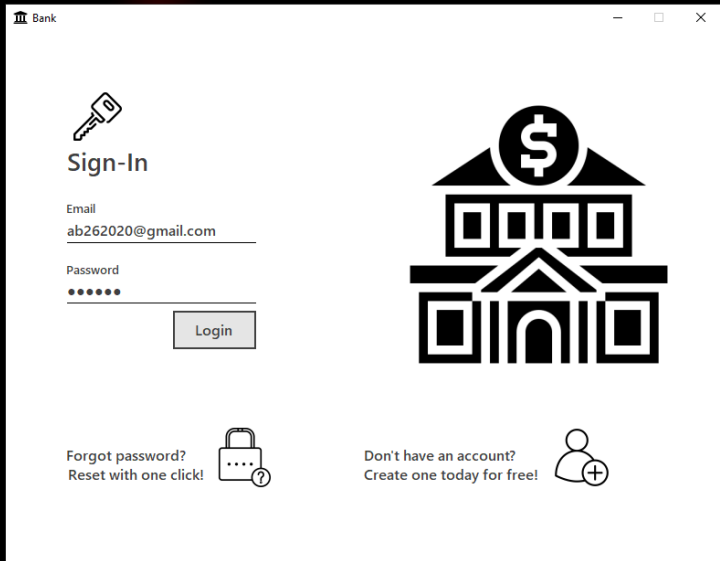
Email
ab262020@gmail.com

Security Key (For Resetting Password, 8 characters)
.....

Sign Up Cancel

Login:

To log in to his account, the user has to type his email and password and click on the login button. This will activate the 'login_m' method in the 'Account' class which will first perform a validity check on the typed email, password and check for empty text boxes. If all is clear, the control will be transferred to the 'loginfetch' method defined in the 'DB_Access' class. The arguments for this function are the email and password. First, the method will check for the email and password in the database. If either of them is wrong, the program will give an error. If everything is good, the user will be logged in to his account and taken to home screen.




Bank


Sign-In

Email
ab262020@gmail.com

Password
.....

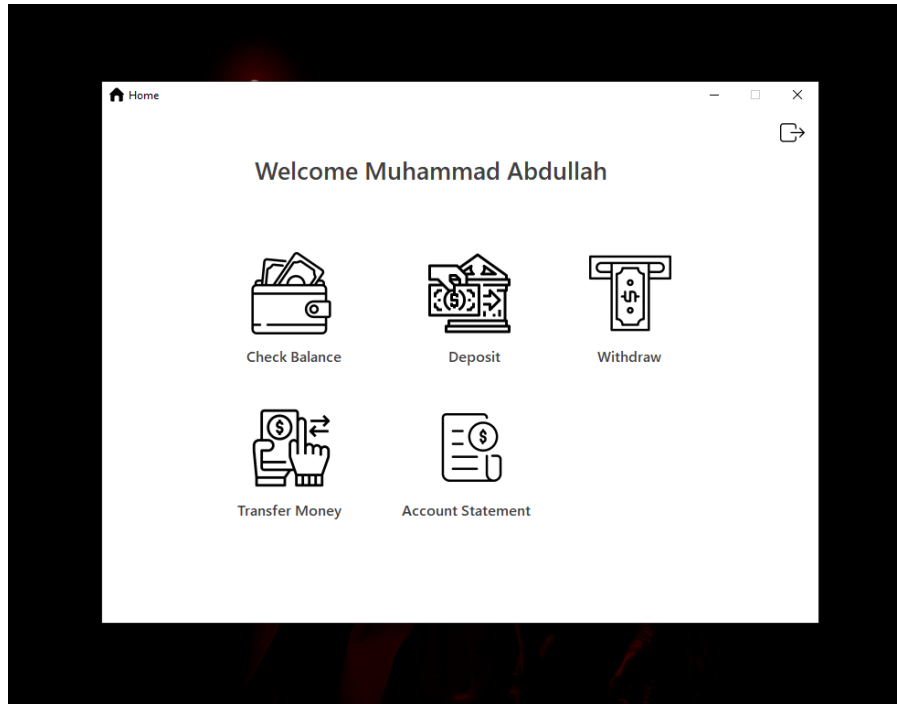
Login

Forgot password? Reset with one click! 

Don't have an account? Create one today for free! 

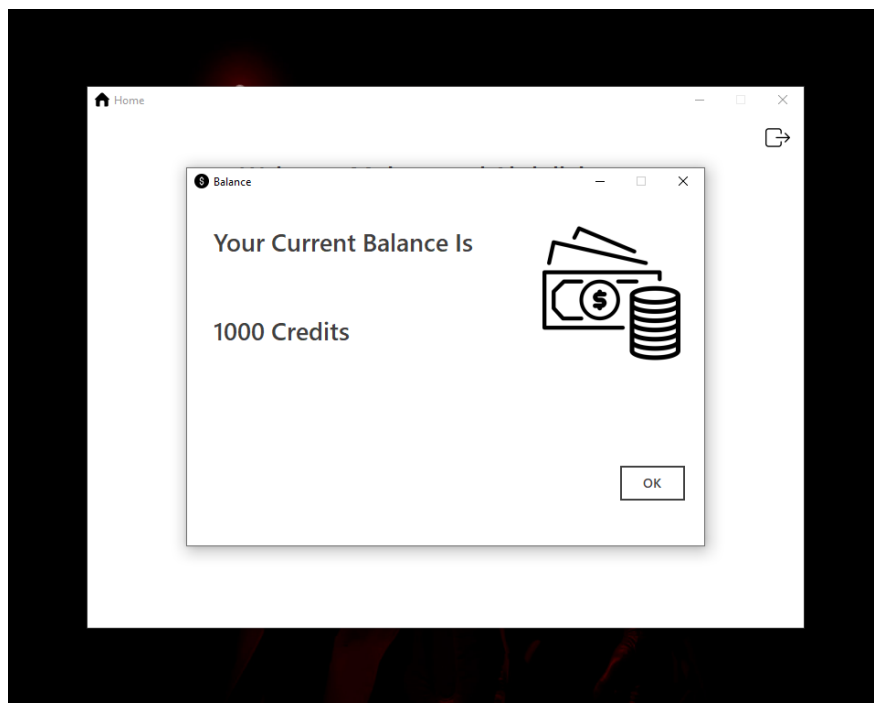
Home:

On this form, the user will find the options for checking his balance, depositing, withdrawing and transferring credits. The user can also check the activity for his account.



Balance:

The user can check his current balance by clicking on the 'Check Balance' button. The program call the 'fetchbal' method from the 'DB_Access' class. The balance is fetched based upon the account id of the logged in user which is stored in the Account.tmpid field defined in the 'Account' class.

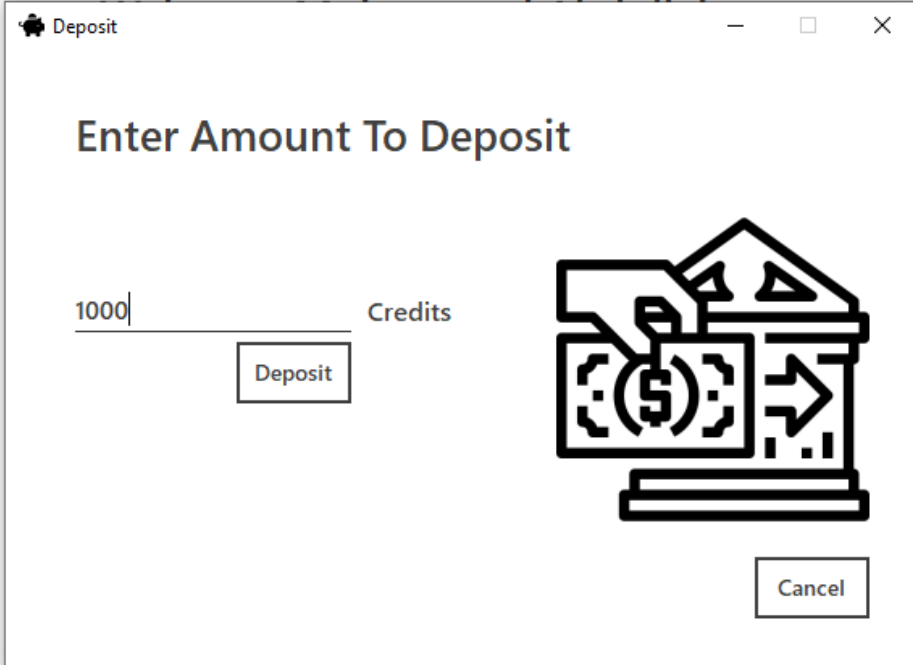


Deposit:

Clicking on the deposit button will create an instance of the 'Deposit' form and show it on the screen. Here the user can type the amount he wants to deposit. The amount should be a positive number or else the user will come across an error. The Deposit button will launch the 'deposit_m' function in the 'Account' class, perform a validation using the 'chkdepo_m' method in 'DataChk' class and in case no error pop ups, the method 'deposit_db' in 'DB_Access' class will add the deposited amount in the current balance and update it. It will also call the 'deposit_stat' method from the 'Accstat' class and update the user's account statement.

 Home

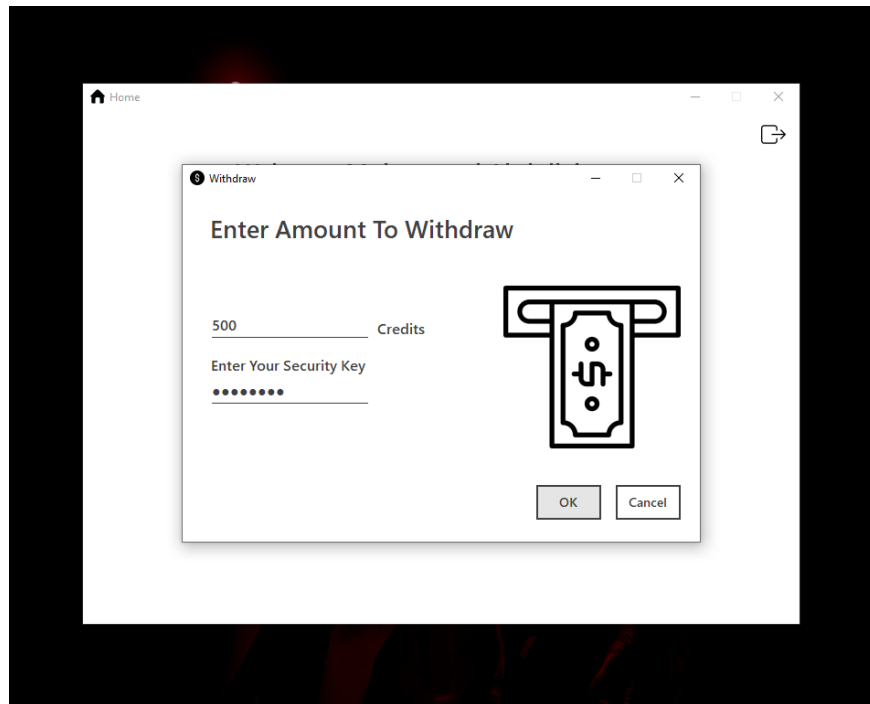
— □ ×



The image shows a window titled "Deposit" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area has the heading "Enter Amount To Deposit". Below this, there is a text input field containing the number "1000". To the right of the input field is the word "Credits". Below the input field is a button labeled "Deposit". To the right of the input field and button is a large, stylized icon of a bank building with a dollar sign and an arrow pointing into it. At the bottom right of the window is a button labeled "Cancel".

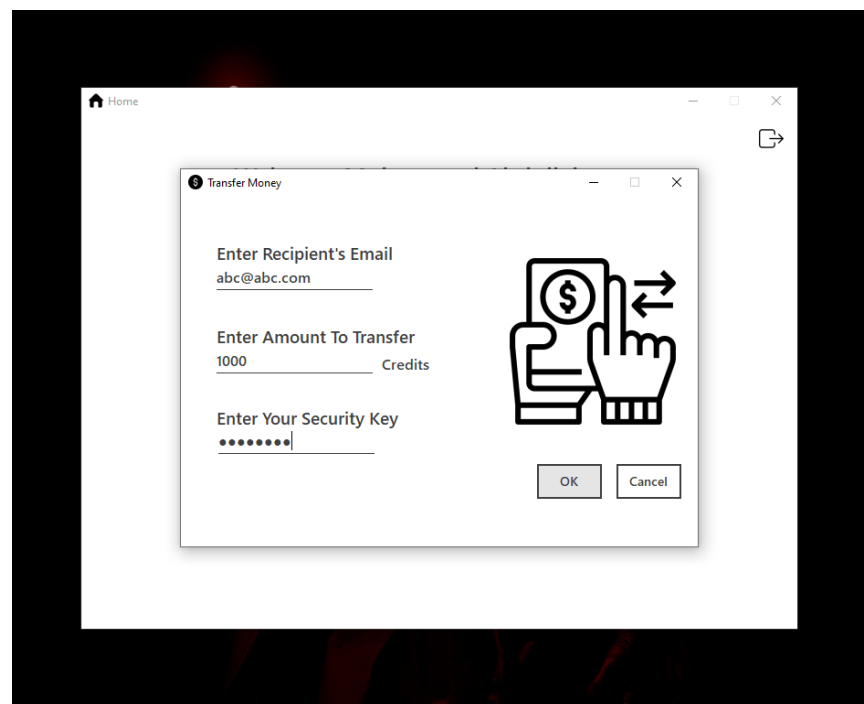
Withdraw:

Just like the deposit process, the Withdraw button on the 'Withdraw' form will take the typed amount from the text box, pass it as an argument to the 'withdraw_m' method in 'Account' class, check if the amount is negative, and upon no errors, the 'withdraw_db' function in 'DB_Access' class will be executed. It also checks if the security key matches the one stored temporarily in RAM when the user is logged in for preventing theft. First, it will check if the amount being withdrawn is not more than the current balance. If it is, the function will return an error, Else, it will deduct the amount from the balance and update it in database. Then, it will call the 'withdraw_stat' function in the 'Accstat' class and write the record of transaction in the account statement file of the user.



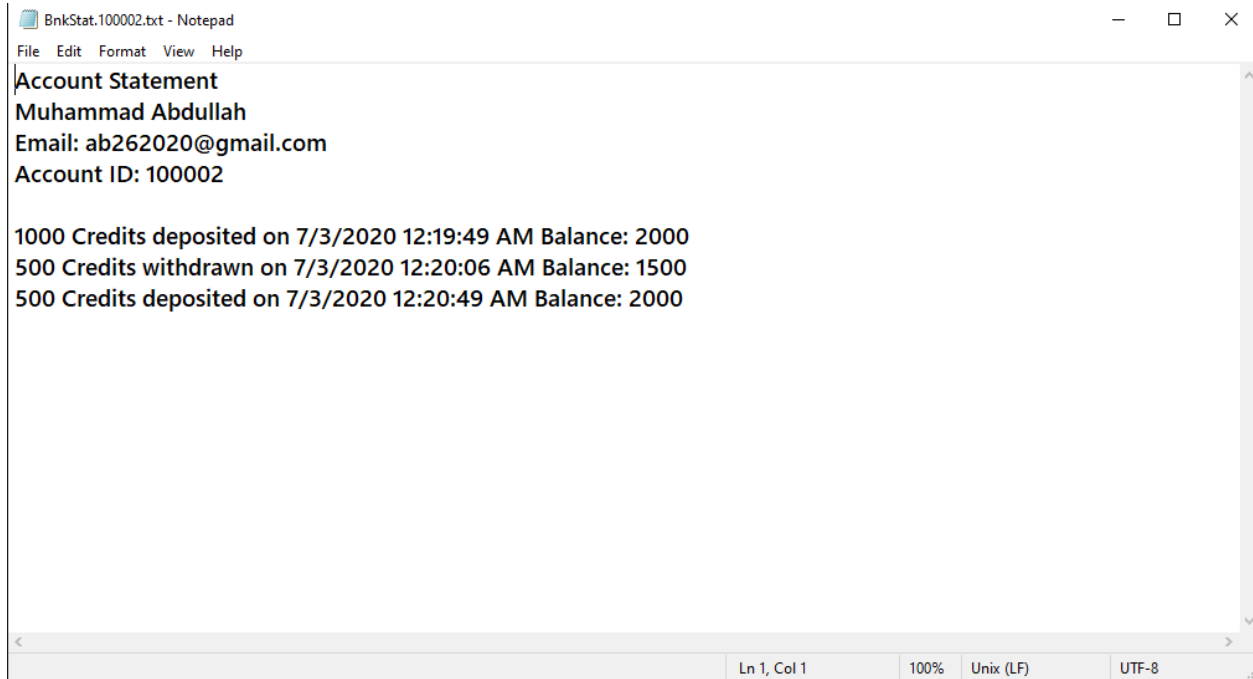
Transfer:

The process of transferring amount to another account is similar just that it takes two additional parameters. First, the email of the recipient and second, the user's security key. Just like above mentioned features, the program will check for valid amount and security key. All this information is passed as arguments to the 'transfer_m' function in the 'Account' class. The 'transfer_db' method in 'DB_Access' class updates both the sender's and recipient's balances in the database.



Account Statement:

The account statement file is created when the account is created. It is of the txt format and is accessed through FileStream class. The text is UTF-8 encoded. All the account statement files are stored in the directory 'C:\Users\Public\Public Documents\Bank Statements'. They are named after the account ID of their respective users. They can be opened without any hassle by clicking on the 'Account Statement' button. This will open the text file in the application designated to open text files on your computer.



The screenshot shows a Notepad window titled 'BnkStat.100002.txt - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The text content is as follows:

```
Account Statement
Muhammad Abdullah
Email: ab262020@gmail.com
Account ID: 100002

1000 Credits deposited on 7/3/2020 12:19:49 AM Balance: 2000
500 Credits withdrawn on 7/3/2020 12:20:06 AM Balance: 1500
500 Credits deposited on 7/3/2020 12:20:49 AM Balance: 2000
```

The status bar at the bottom indicates 'Ln 1, Col 1', '100%', 'Unix (LF)', and 'UTF-8'.

Error and Success Messages:

All the error messages are defined in the 'Msgs' class and they are all methods of void type. They are called by the 'DataChk' class when any anomaly is found by the program from user's side. It also has messages to let user know if the desired operation has been performed successfully. For example, the user will be notified when the account has been created or amount has been withdrawn successfully.

The Data Check Class:

This class is responsible for handling the validation of arguments passed to functions through text boxes. The methods in this class are first called by the functions in 'Account' class before the functions in 'DB_Access' class are called to perform operations on the database.

The Database:

The database used for storing the accounts of users is a SQL local database. It is accessed through SQL connection, commands and queries. A SQL data adapter is used to read from and write data to the database. It also loads the data related to the logged in user to the RAM namely the user's email, password, account ID and security key.

Resources:

The following links lead to the helpful material that was utilized in this project:

<https://www.guru99.com/c-sharp-access-database.html> (Used to understand the method for accessing the database only for this project.)

<https://www.youtube.com/watch?v=ijZNJr3jEZE> (For solving the issue of multiple instances of a single form.)

<https://www.youtube.com/playlist?list=PLxefhmF0pcPIDKe7smJMoHPNS1tJQ7w7q> (First three or four videos from this playlist to understand the basics of Windows Forms.)

Microsoft Docs for solving the issue of database.

Issues:

Email feature is missing due to security reasons thrown by mail service providers.

PDF format of account statement couldn't be implemented due to an issue with iTextSharp. I tried to follow possible solutions available but still it didn't work.