MOBILE APPLICATION DESIGN AND DEVELOPMENT

Subject code – 2021S_MAD 3004_1

Project - Banking System

Submitted to - Emad Nasrallah

Submitted By - Manpreet Singh Sarao(C0813625)

Miral Lakhani(C0821532)

Najmeh Akbari(C0822530)

Dishant Chaudhary()

Contents

Acknowledgment

As per the requirements of the syllabus of **Mobile Application Design and Development** (MADT) course in semester one, we are required to prepare a project report. Hence regarding to that we have prepared a project on a **Banking System**, that helps in managing all the processes done in bank.

The satisfaction that accompanies of the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to our project guide **Mr. Emad Nasrallah** for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project. Without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

We attempt to make the application of a Banking System to the **Lambton College**.

Student ID	**Student Name**
C0813625	Manpreet Singh Sarao
C0821532	Miral Lakhani
C0822530	Najmeh Akbari
C0xxxxxx	Dishant Chaudhary

Preface

The objective of the project is to design a Banking System which allows user to perform all the operations of bank. The project has been designed in swift language using online swift compiler.

This project report contains the description of system and its attributes, methods, working flow and the difficulties that we have faced and the way we solved them.

A Banking System is a console-based application to create bank account. This project allows the user to create account, login with registered account. This project contains such modules as follows:

- Login
- Register
- Display Current Balance
- Deposit Money
- Withdraw Money
- Transfer Money
- Pay Utility Bills
- Logout

The project tries to accomplish all the following features:

1) Smooth flow of data without any errors.

2) Adequate validation checks for data entry.

3) Registration process is working smoothly.

5) Prompt and specific retrieval of data.

6) Flexibility in the system according to the changing environment.

7) Controlling redundancy in storing the same data multiple times.

8) Accuracy, timeliness, and comprehensiveness of the system output.

9) Enhancement in the completion of work within the constraints of time.

The scope of the system is quite wide. It can be used efficiently and without any difficulty on any system with specific configurations.

System Requirements

Hardware / Software Environment

- **Hardware**
  - Intel i5-7200u processor
  - 500 GB Hard Disk
  - 8GB RAM
- **Software**
  - Operating System
  - Microsoft Windows
- **Development Tools**
  - Online Swift Compiler
- **Documentation & Presentation Tools**
  - Microsoft Word

System design

This system will be used in below Modules which are explained briefly. As all of these have different requirements and the modules are designed to meet their needs and avoid any type of confusion. The Uses of all these Modules have been described below.

System can do the following functions:

- Let user register with details like name, password, mobile no., account type and gender.
- Let user login with their userid and password.
- Let user check their current balance.
- Let user deposit money if they want to deposit.
- Let user withdraw money.
- Let user transfer money one account to another account.
- Let user pay their utility bills using this system.

Classes and Methods:

Classes: This system has one class which name is BankUser which contains attributes that are given below:

- userID
- accountNo
- username
- password
- gender
- mobileNo
- accountType
- balance

Methods: This class has following methods:

- Init: used to initialize all the attributes that are used in the class.
- printDetails: used to display the display the details of logged in user(user ID, Account Number, User Name, Gender, Mobile Number, Account type and balance.)
- displayBalance: used to display the account balance of logged in user.
- checkBalance: it checks if the user have enough balance in his/her account to make the payment.

- deposit: it takes the amount from the user and credit into account.
- withraw: it helps to deduct amount from user’s account whenever there is transfer of funds or during the bill payment.

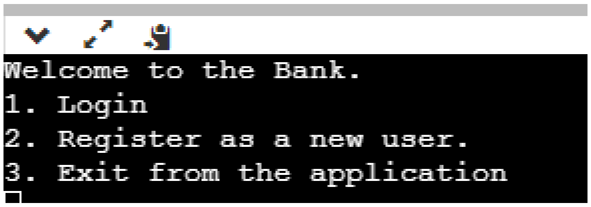
System Tasks:

This system has two menus: main menu and sub menu. Main menu displays the functions login, register and exit from application and sub menu displays the functions display current balance, deposit money, withdraw money, transfer money, pay utility bills and logout. These all methods described below with its functionality:

- **Login**: This method takes userID and password as input to login into user’s account. It assigns values to token as userID and userObj as user details.
- **Register User**: User can register their account by providing details user name, password, gender, mobile no. and account type. registerUser method automatically assign the userID and Account Number as per the last entry.
- **DisplayBalance**: User can check their available balance.
- **Deposit**: User can deposit money by providing the amount of money and it displays the updated balance after the deposit.
- **Withdraw**: User can withraw money by providing the amount of money and it displays the updated balance after the deposit.
- **Transfer Funds**: User can transfer money from his/her account to another user’s account, if the account is available and also display balance after the fund transfer.
- **Pay Bills**: With this method user can pay bills like electricity bill, gas bill etc.
- **Logout**: With this user can logout from their account and go back to main menu.

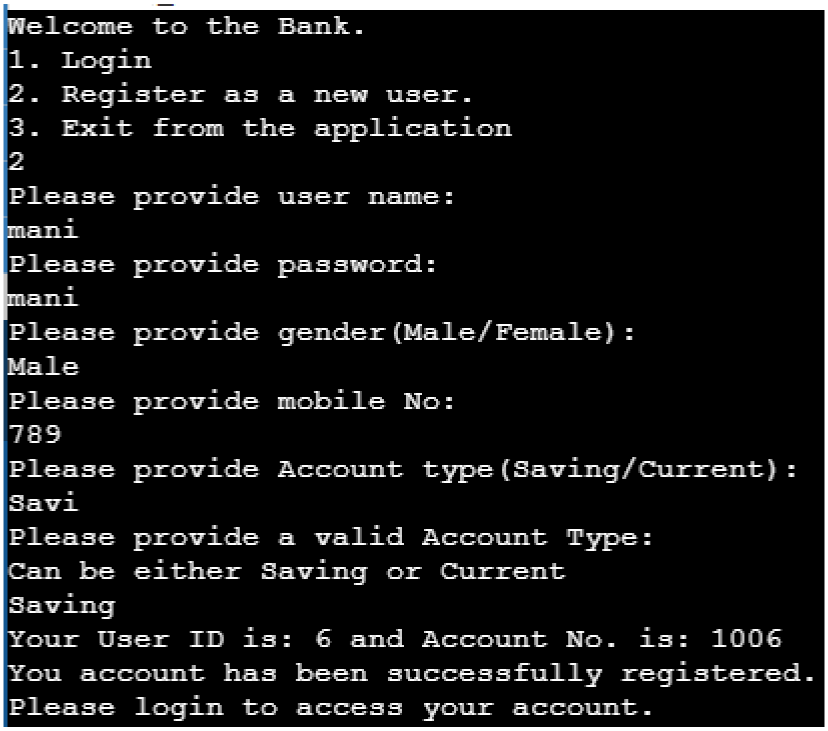
User Manual

Once user start the application. Landing menu will be displayed as below:



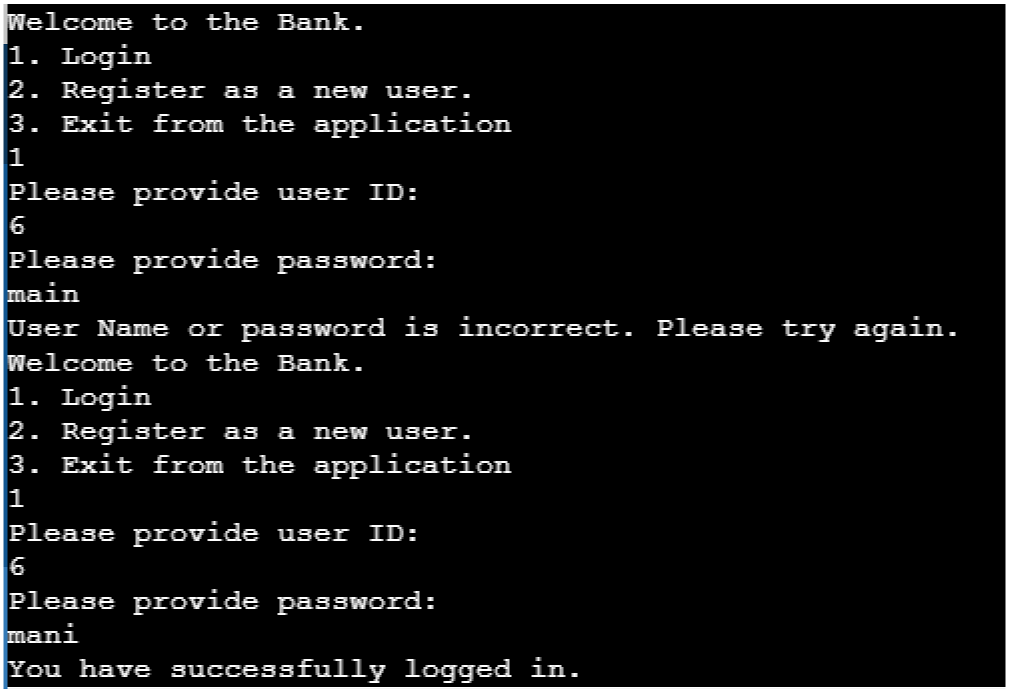
```
Welcome to the Bank.
1. Login
2. Register as a new user.
3. Exit from the application
```

If user is new to the application, he can choose 2 to register. User will be asked to input User name, Password, Male, Mobile Number, and Account type. All the details are validated during the input. If all the details are valid user will be registered and user ID and Account Number will be displayed as shown below:



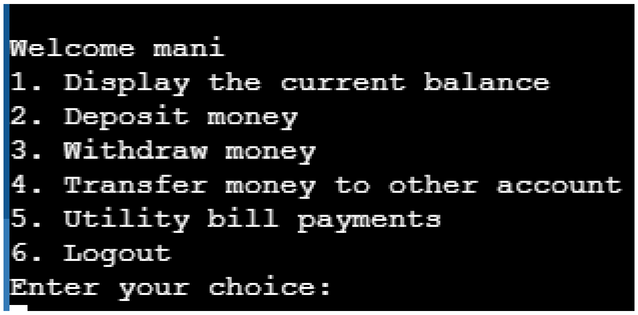
```
Welcome to the Bank.
1. Login
2. Register as a new user.
3. Exit from the application
2
Please provide user name:
mani
Please provide password:
mani
Please provide gender(Male/Female):
Male
Please provide mobile No:
789
Please provide Account type(Saving/Current):
Savi
Please provide a valid Account Type:
Can be either Saving or Current
Saving
Your User ID is: 6 and Account No. is: 1006
You account has been successfully registered.
Please login to access your account.
```

Once registered, user can login to the account by giving 1 as input in the Landing Menu. User will be prompted to provide user ID and password as shown below:



```
Welcome to the Bank.
1. Login
2. Register as a new user.
3. Exit from the application
1
Please provide user ID:
6
Please provide password:
main
User Name or password is incorrect. Please try again.
Welcome to the Bank.
1. Login
2. Register as a new user.
3. Exit from the application
1
Please provide user ID:
6
Please provide password:
mani
You have successfully logged in.
```

If both the ID and password are correct, user will be redirected to Sub Menu, where below operations can be performed:



```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
```

1: with this user can display the current balance of the account.

2: User can deposit money into their account.

3: User can withdraw money from their account.

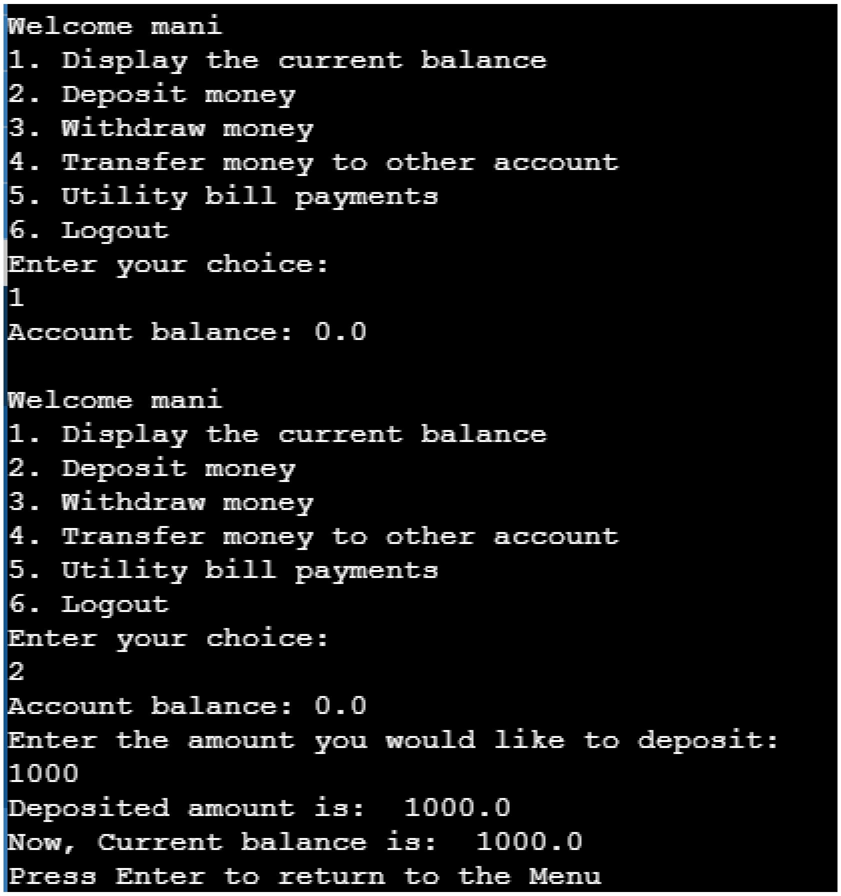
4: Can be used to transfer funds to other user’s accounts. It will display the receiver’s name and take confirmation from user If this the correct receiver.

5: Can be used to make the bill payments.

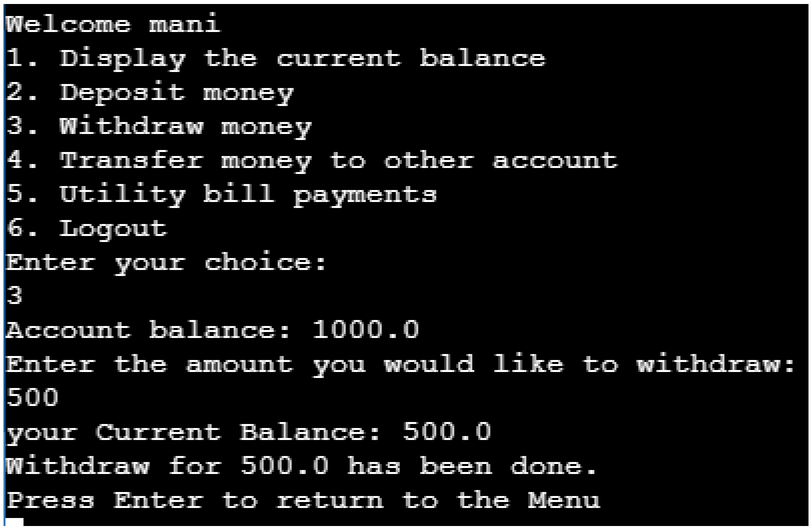
6: Logout from the account and user will be redirected back to Landing Menu.

3,4,5 functions will do the validations if the user have enough money to make the transaction.

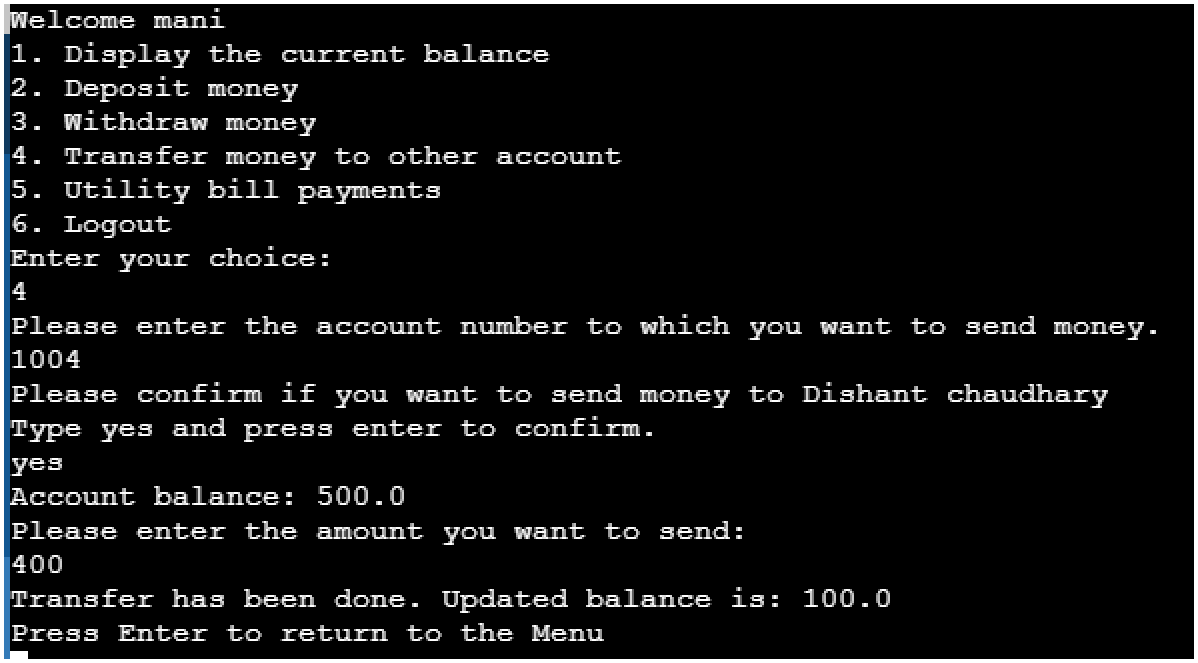
All the above operations screenshots are displayed below.



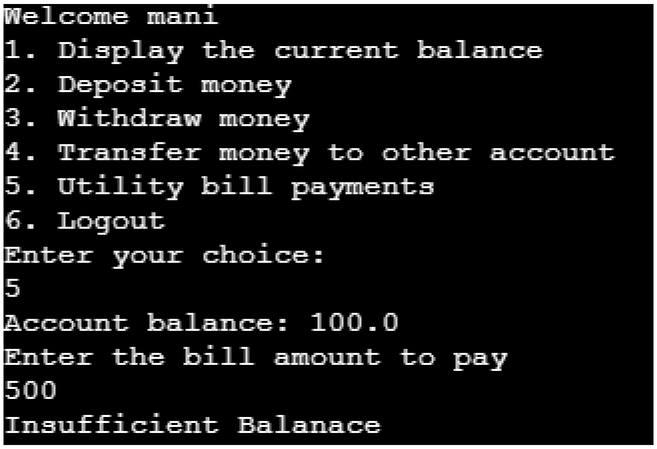
```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
1
Account balance: 0.0

Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
2
Account balance: 0.0
Enter the amount you would like to deposit:
1000
Deposited amount is:  1000.0
Now, Current balance is:  1000.0
Press Enter to return to the Menu
```



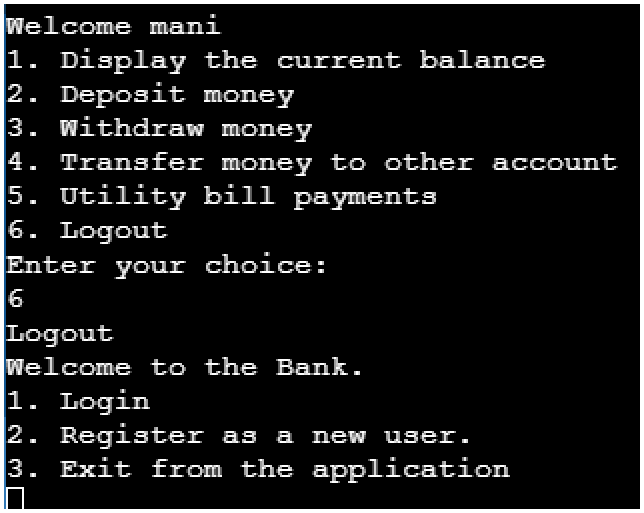
```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
3
Account balance: 1000.0
Enter the amount you would like to withdraw:
500
your Current Balance: 500.0
Withdraw for 500.0 has been done.
Press Enter to return to the Menu
```



```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
4
Please enter the account number to which you want to send money.
1004
Please confirm if you want to send money to Dishant chaudhary
Type yes and press enter to confirm.
yes
Account balance: 500.0
Please enter the amount you want to send:
400
Transfer has been done. Updated balance is: 100.0
Press Enter to return to the Menu
```

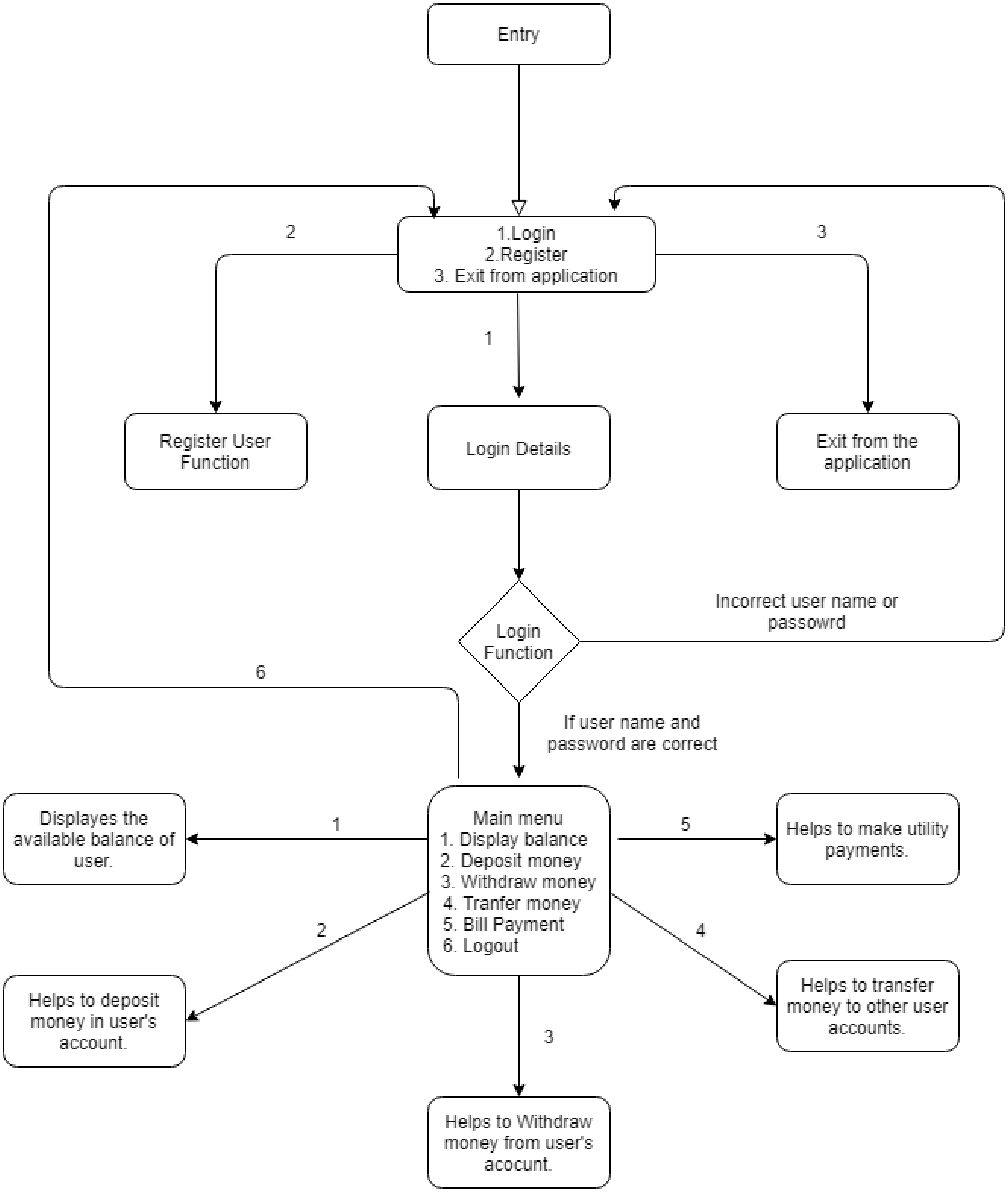


```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
5
Account balance: 100.0
Enter the bill amount to pay
500
Insufficient Balanace
```



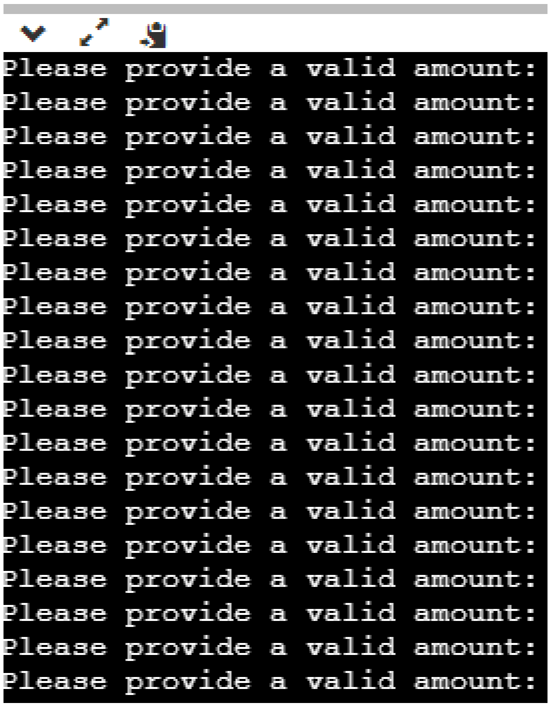
```
Welcome mani
1. Display the current balance
2. Deposit money
3. Withdraw money
4. Transfer money to other account
5. Utility bill payments
6. Logout
Enter your choice:
6
Logout
Welcome to the Bank.
1. Login
2. Register as a new user.
3. Exit from the application
```

Workflow of the Application


Entry
1.Login
2.Register
3. Exit from application
2
1
3
Register User Function
Login Details
Exit from the application
Login Function
Incorrect user name or passowrd
6
If user name and password are correct
Main menu
1. Display balance
2. Deposit money
3. Withdraw money
4. Tranfer money
5. Bill Payment
6. Logout
1
Displayes the available balance of user.
5
Helps to make utility payments.
2
Helps to deposit money in user's account.
4
Helps to transfer money to other user accounts.
3
Helps to Withdraw money from user's acocunt.

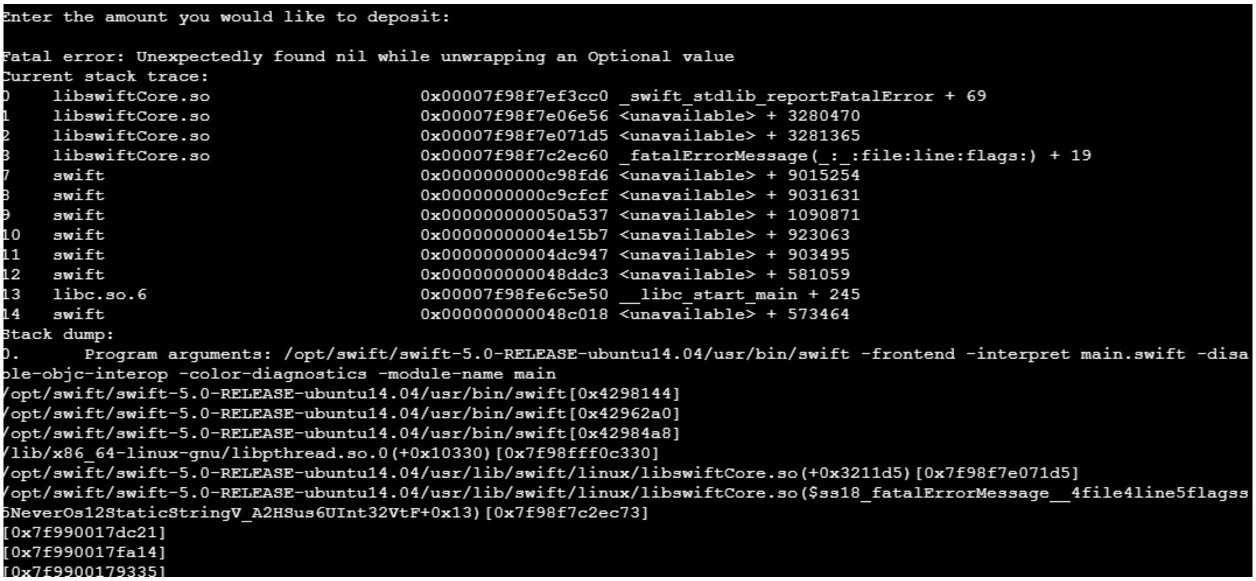
Errors during development:

Code was going into loop as missed to provide input to a variable. After assigning the value to that variable it was running file.



```
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
Please provide a valid amount:
```

We have got multiple errors like below due to type conversion. We have validated them before assigning it to a variable it helped to solve the problem.



```
Enter the amount you would like to deposit:

Fatal error: Unexpectedly found nil while unwrapping an Optional value
Current stack trace:
0    libswiftCore.so                    0x00007f98f7ef3cc0 _swift_stdlib_reportFatalError + 69
1    libswiftCore.so                    0x00007f98f7e06e56 <unavailable> + 3280470
2    libswiftCore.so                    0x00007f98f7e071d5 <unavailable> + 3281365
3    libswiftCore.so                    0x00007f98f7c2ec60 _fatalErrorMessage(_:_:file:line:flags:) + 19
7    swift                              0x0000000000c98fd6 <unavailable> + 9015254
8    swift                              0x0000000000c9cfcf <unavailable> + 9031631
9    swift                              0x000000000050a537 <unavailable> + 1090871
10   swift                              0x00000000004e15b7 <unavailable> + 923063
11   swift                              0x00000000004dc947 <unavailable> + 903495
12   swift                              0x000000000048ddc3 <unavailable> + 581059
13   libc.so.6                          0x00007f98fe6c5e50 __libc_start_main + 245
14   swift                              0x000000000048c018 <unavailable> + 573464
Stack dump:
0.      Program arguments: /opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/bin/swift -frontend -interpret main.swift -disa
ble-objc-interop -color-diagnostics -module-name main
/opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/bin/swift[0x4298144]
/opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/bin/swift[0x42962a0]
/opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/bin/swift[0x42984a8]
/lib/x86_64-linux-gnu/libpthread.so.0(+0x10330)[0x7f98fff0c330]
/opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/lib/swift/linux/libswiftCore.so(+0x3211d5)[0x7f98f7e071d5]
/opt/swift/swift-5.0-RELEASE-ubuntu14.04/usr/lib/swift/linux/libswiftCore.so($ss18_fatalErrorMessage__4file4line5flagss
5NeverOs12StaticStringV_A2HSus6UInt32VtF+0x13)[0x7f98f7c2ec73]
[0x7f990017dc21]
[0x7f990017fa14]
[0x7f9900179335]
```

Conclusion

Project Planning and Information Gathering:

Swift is the computer language used in the financial system of the project.

When we were developing this project, aimed to construct a simulation of a rudimentary banking system that allows the user to open bank accounts for customers and provide them access to some financial transactions.

Because we did not have access to an iOS system, we used the swift online compiler. Our objective is to develop a system in such a way that the user is not perplexed when using it, the menus clearly explain all the project's operations.

We performed research to choose various types of bank accounts, and we enlarged the project to some extent.

By establishing a password for each user, security vulnerabilities were discovered when a client joined the banking system.

Challenges Faced:

• Because we are all learning from different locations with different time zones, it was difficult for us to coordinate project meetings.

• Aside from that, because we are not from a banking system, we had to research the system before we began to adopt the appropriate activities.

• There were a few technical issues, such as software integration difficulties, that we all worked together to fix.

How we divided the work among the team:

To make the project as good as possible, we organized a team using the WhatsApp app to facilitate planning. Then, at the online Google conference, we went through the information fully. we discovered areas of strength for each student and split the project activities properly. The next day, the progress of the job was checked. Finally, all the details were combined, and the final version was ready for submission.

References

Class lectures

https://swift.org/documentation/

https://stackoverflow.com/

Github Link: