

Semester Project Documentation

Semester Project Title: Bank Management System

GitHub Repository Link: <https://github.com/16arsal/qr-dms-ma-wh>

Student Details

(Fill according to number of team/group members)

| | Student Name | Student Reg # | Student Degree |
|-----------|-----------------|---------------|-------------------------|
| Student-1 | Muhammad Aarsal | 2022350 | Computer Engineering |
| Student-2 | Wardah Haya | 2022622 | Artificial Intelligence |

Main Features

1. Creating the bank account for the customer
 2. Deposit amount for the customer
 3. Withdraw amount for the customer.
 4. Balance Enquiry by the customer
5. Closing or terminating a bank account
 6. Updating the bank account
 7. Admin access

Types of Users

User

Admin

Requirements Breakdown

1.1 User Class: Create a "User" class with attributes such as username, password, name, address, and contact information.

1.2 Authentication: Implement a username/password authentication mechanism for user login.

1.3 User Registration: Create a registration feature that collects user information and stores it securely, using a database or file storage.

2.1 Account Class: Implement an "Account" class that represents a user's bank account, including attributes like account number and balance.

2.2 Deposit Function: Provide a function in the Account class that allows users to deposit money into their account, updating the account balance accordingly.

3.1 Withdrawal Function: Provide a function in the Account class that allows users to withdraw money from their account, updating the account balance accordingly.

4.1 Balance Inquiry Function: Provide a function in the Account class that allows users to check their account balance, returning the current balance value.

4.2 User Interactions: Define functions for features like account balance inquiries, fund transfers,

5.1 User Class: Create a "User" class that represents a bank user, including attributes like username, password, and account information.

5.2 Admin Privileges: Implement a mechanism to grant the admin the necessary privileges to delete user accounts. This can be done by providing a specific function or method in the admin's class that allows them to delete user accounts, taking the user's account information (such as username or account number) as input.

6.1 Access Control: Define different roles (admin+user) with varying permissions using inheritance or composition.

7.1 Admin Class: Create an "Admin" class to represent the administrator of the system. Include attributes like username and password.

7.2 Authentication: Implement a login mechanism specifically for the admin using username and password.

7.3 Admin Privileges: Define admin-exclusive functionalities, such as managing user accounts and generating reports.

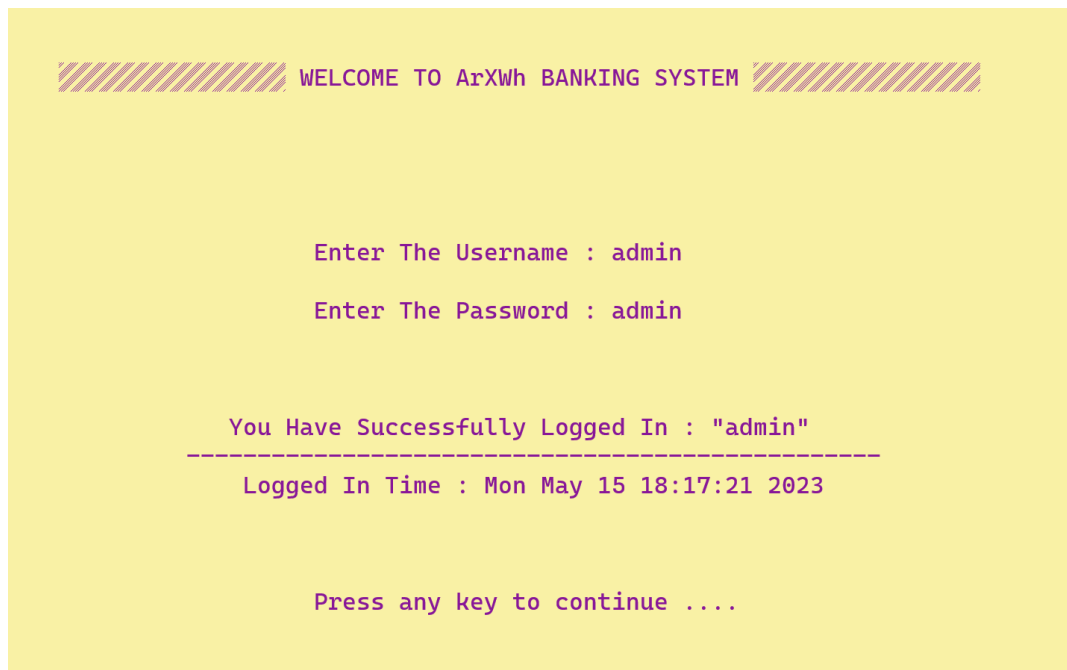
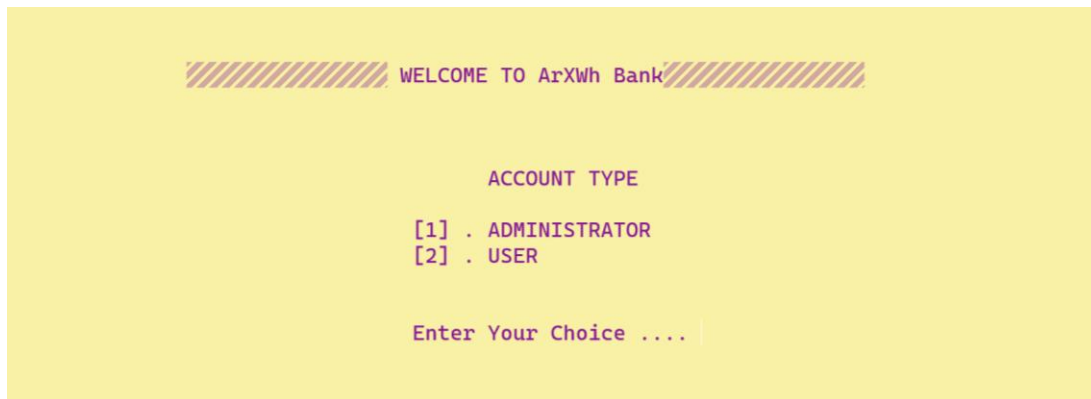
7.4 User Management: Provide functions for the admin to create, update, and delete user accounts.

7.5 Security Measures: Implement basic security measures, like password encryption and validation, to ensure admin access is secure.

7.6 Access Control: Allow access to admin functionalities only after successful authentication.

GUI Screens / Design / Wireframes

We didn't make GUI. However, we designed our terminal in a way to make it look more appealing and user friendly.



WELCOME TO MAIN MENU

- [1] . View Customer Accounts
- [2] . Customer Account Registration
- [3] . Edit Customer Account
- [4] . Delete Customer Account
- [5] . Search Customer Account
- [6] . Transaction
- [7] . Log Out !!!
- [8] . About US

Please Enter Your Choice [1-8] :

CUSTOMER ACCOUNT REGISTRATION

- [1] . Enter Your Name : Hasan
- [2] . Enter Your Account Number : 99
- [3] . Enter Your Phone Number : 111000114893
- [4] . Enter Your Address : 555wc
- [5] . Enter Your E-mail : hassan@live.com
- [6] . Enter Your Citizenship No.: 531
- [7] . Enter Amount To Deposit : \$898

CUSTOMER ACCOUNT REGISTRATION SUCCESSFULL

Do You Want To Add Another Record ? (Y/N) :

CUSTOMERS LIST

| A/C No. | Account Name | UserID | Email Address | Phone No. | Balance |
|---------|--------------|------------|-------------------------|--------------|---------|
| 5678 | Ahmed | 8644864302 | ahmed@gmail.com | 98989898 | 1\$ |
| 8888 | affan | 2416809743 | af@gmail.com | 9999999999 | 99\$ |
| 1988 | Rose | 8910997606 | rose@gmail.com | 03001234097 | 4576\$ |
| 4367 | Taha | 6440179498 | taha@outlook.com | 039013901240 | 777\$ |
| 5555 | Arsal | 6824597074 | muhammadarsal@gmail.com | 111244622 | 87000\$ |
| 99 | Hasan | 6952705125 | hassan@live.com | 111000114893 | 898\$ |
| 19167 | Wardah | 6042656562 | wardah@yahoo.com | 090036434917 | 50000\$ |

Press [Enter] to return back to main menu.

/// ABOUT US ///

It is a Simple Bank Management System Project for our 2nd Semester based on C++ programming language OOPs.

Members of our Team:

[1] . Muhammad Aرسال-2022350
[2] . Wardah Haya-2022622

Press any key to return back to main menu.

Features to Coddling Matrix

(In the following table you will mention the following items for each feature, mention the items in each column for each feature of your application)

| Sr no. | Feature Name | OOP Type Used | Functions Created | Variables / Obj Created | Line of Code Written |
|--------|---|-------------------------|---|-------------------------|----------------------------------|
| 1 | Data Encapsulation | Bundling of Data | | | Line 29 onwards |
| 2 | Authenticate | Authentication OOP | Void authenticate() Void login() Int verify () | | Line 58,64,80,89 |
| 3 | Menu | | Void menu () | | Line 127 onwards |
| 4 | Options | | Void Options () | | Line 154 |
| 5 | Searching Account | | Void srch() Void search_acc() Void search_name() | | Line 193 Line 215 Line 216 |
| 6 | User Features | | void chkblnc() void deposit() void withdrawl() void transfer() | | Line 271 onwards |
| 7 | Exitting Menu | | Void menuexit() | | Line 276 |
| 8 | About Us for relevant information | Returns our information | | | Line 277 onwards |
| 9 | Record | Inheritance | Class record{} | | Line 303 |
| 10 | Module for adding account | | Bms::add() | | Line 375 |
| 11 | Module for checking account in database | | Int chkacc() | | Line 437 |
| 12 | Module for deleting account | | bms::del() | | Line 531 |
| 13 | Module for logging out of the program. | | bms::menuexit() | | Line 1049 |