# ZAQ Bank Management System

# Feature Manual

**Course:** Programming Fundamentals  
**Semester:** 2nd  
**Language:** C++  
**Project Type:** Console-Based Application  
**Data Storage:** File Handling  
**Students:** **Zulqarnain Saeed**

**📌 Project Overview**

The ZAQ Bank Bank Management System is a console-based C++ application developed as a second-semester project to simulate real-world banking operations. It uses core programming concepts such as conditional statements, loops, arrays, functions, pointers, and file handling. The system is divided into two major roles: Customers and Employees, each having their own secure login and operational interface.

**🔧 Technologies Used**

Language: C++

Libraries: iostream, fstream

Compiler: Visual Studio

Data Storage: File Handling (customers.txt, transactions.txt)

No External Libraries Used

**✅ Instructions Followed Strictly**

❌ No global variables used.

❌ No C++ built-in string functions (strcmp, strcpy, etc.)

❌ No use of string type – only C-style character arrays used.

✅ All features are implemented in separate functions (1 function = 1 task).

✅ File handling is used for storing and retrieving persistent data.

✅ Code is memory efficient – no extra space or memory leaks.

✅ 100% compatible with Visual Studio C++ Compiler.

✅ Code uses modular structure and function-based logic.

**🧩 Project Modules**

**1. Main Menu**

Displays 3 options:

Login as Customer

Login as Employee

Exit

**A. Customer Login:**

Customer enters:

Account Number

PIN

System matches credentials from file.

On success → Goes to Customer Panel.

On failure → Displays "Invalid credentials".

**B. Employee Login:**

Employee enters:

Username (must contain @)

PIN

If @ is missing → Login is rejected.

On success → Goes to Employee Panel.

1. **Customer Panel – Features & Explanations**

Once a customer logs in successfully, the system opens the Customer Panel with the following options:

**1. ✅ Check Balance**

Displays the current available balance of the logged-in customer.

Helps the user keep track of their funds.

Data is fetched from the file and shown accurately.

**2. 💰 Deposit**

Allows the customer to deposit money into their account.

The entered amount is added to their balance.

The transaction is saved in transactions.txt for record-keeping.

**3. 🏧 Withdraw**

Lets the customer withdraw money from their account.

Withdrawal only succeeds if:

The balance is sufficient.

The withdrawn amount is deducted from the customer’s balance.

Transaction is saved in the transaction history.

**4. 🧾 View Account Info**

Shows all details of the customer's account:

Name

Account Number

CNIC

Age

Income

Balance

Data is read directly from the customer's stored record.

**5. 📝 Change Notes**

Lets the user write or update personal notes.

Notes can be simple text stored in a file (optional feature if implemented).

Helps customers store reminders or banking-related notes.

**6. 🤖 Chatbot Assistance**

A basic chatbot that responds to customer queries.

Replies to commands like:

“How to deposit?”

“What can I do here?”

Provides help about available features and how to use them.

**7. 💱 Currency Converter**

Converts PKR to other currencies like:

USD

EUR

GBP

AED

Uses fixed conversion rates.

Helpful for customers dealing in foreign exchange or sending money abroad.

**8. 🏦 Loan Eligibility Checker**

Analyzes:

Age

Income

If age is greater then 19 and monthly income is more then 30000 or equal loan will granted

Based on the above, system checks if user is eligible for a bank loan.

Shows message:

"Eligible for up to X PKR"

Or "Not eligible due to low income/balance/age"

**9. 🔁 Transfer Money**

Customer can send money to another existing account.

Requires:

Recipient's account number

Amount to send

System checks:

Both accounts exist

Sufficient balance in sender’s account

Amount is deducted from sender and added to recipient

Transaction is saved in transactions.txt

**10. 📄 Transaction History**

Shows a list of past transactions by the customer.

Includes:

Deposit

Withdrawal

Transfers

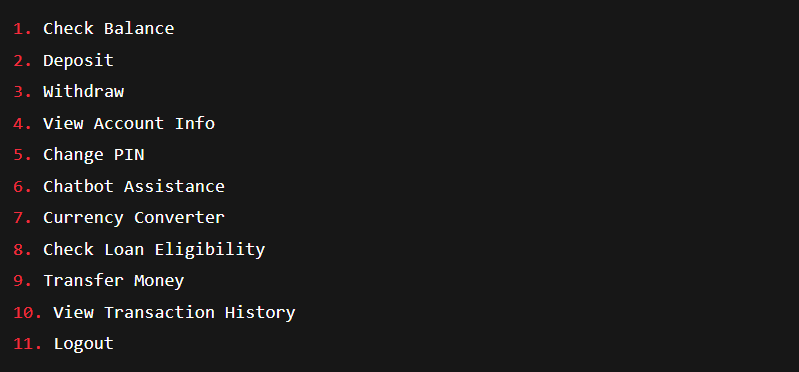
Displays type, amount, and recipient/sender account (if any)

**11. 🚪 Logout**

Ends the customer session.

Returns to the main menu.

Ensures account data is saved before exit.



1. **Employee Panel – Features & Explanations**

When an employee logs in (with a username containing @), they access the Employee Panel, which contains administrative functionalities:

**1. ➕ Create New Account**

Used by employees to register new customers.

Inputs:

Name, CNIC, Age, Income, PIN, etc.

Validates:

Unique CNIC and account number

PIN rules (length ≥ 8 and must contain @, #, or $)

Saves the new customer data in customers.txt

**2. 👁️ View All Accounts**

Displays a list of all registered customer accounts.

Includes:

Name, Account Number, CNIC, Age, Income, Balance

Helpful for monitoring overall customer data.

**3. 🔍 Search Customer**

Allows employee to search a customer by:

Name

OR Account Number

On match, shows complete details of that customer.

**4. ✏️ Update Info**

Enables the employee to edit customer details like:

Name

Age

Income

Used when a customer requests to update their records.

After changes, file is updated immediately.

**5. ❌ Delete Account**

Deletes a customer's record using their account number.

Used if:

Customer closes their account.

Account is permanently removed from customers.txt

**6. 💵 View Total Bank Balance**

Calculates and displays the sum of all customer balances.

Gives an idea of how much total money the bank is holding.

**7. 🧮 Sort Accounts by Balance**

Sorts all accounts:

In ascending or descending order of balance.

Useful to analyze top savers and low balance accounts.

**8. 🔠 Sort Accounts Alphabetically**

Sorts all customers by their names (A-Z).

Helps employee locate users easily when printed.

1. **📈 Maximum Balance Holder**

Finds and displays the customer with the highest balance.

Shows their full account details.

Useful for statistics and rewards.

**10. 📉 Minimum Balance Holder**

Finds and displays the customer with the lowest balance.

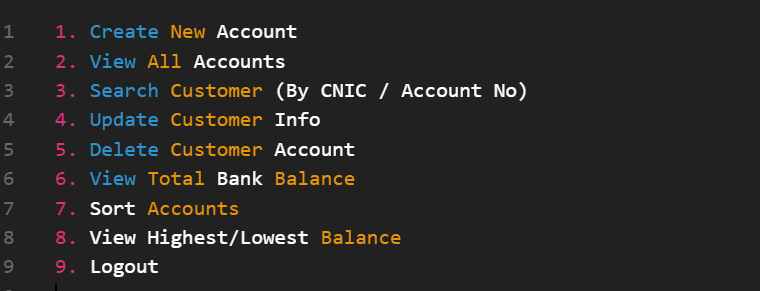
Helps in identifying low-balance or dormant accounts.

**11. 🚪 Logout**

Ends the employee session.

Returns to the main menu.

Ensures all updates are saved to file before exiting.

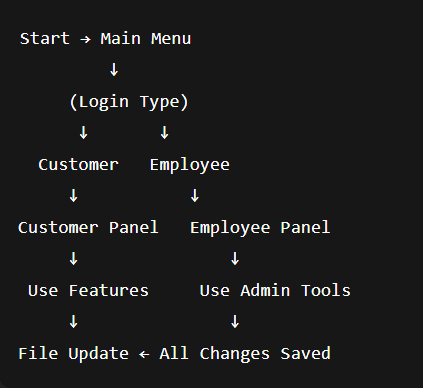


1. **Data Validations**

PIN must be:

* ≥ 8 characters
* Must include @, #, or $
* Employee Username must include @
* CNIC must be unique for each customer
* Account number must be unique
* Withdrawals allowed only if balance is sufficient
* Transfers allowed only between existing accounts.

1. **Project Logic Flow**

****

1. **Key Highlights**

* File-based login (no hardcoded credentials)
* Multiple users supported (100+)
* Secure login with validations
* Real banking operations simulation
* Transaction tracking and account statistics
* Smart chatbot assistant
* Currency conversion system
* Loan suggestion engine
* Search and Sort features for admin

1. **Concepts Used in ZAQ Bank Bank Management System**
2. Variables and Data Types
3. Character Arrays (C-style strings)
4. 1D Arrays
5. 2D Arrays
6. if, else if, else statements
7. switch statements
8. for loops
9. while loops
10. do-while loops
11. User-defined Functions
12. Function Decomposition (modular code)
13. Input and Output (cin, cout)
14. File Handling (ifstream, ofstream, fstream)
15. Data Persistence (storing in and loading from files)
16. Menu-driven Programming
17. User Authentication
18. Custom Validation (e.g. PIN rules)
19. String Comparison (manual, character by character)
20. Linear Search
21. Sorting (by balance and names)
22. Transaction Logging
23. Conditional Logic
24. No use of built-in string functions
25. No use of global variables

# ✅ 7. Conclusion

The ZAQ Bank Bank Management System is a feature-rich and secure banking simulator created using basic to intermediate level C++ concepts. It reflects real-world banking workflows and demonstrates strong command over file handling, data validation, user management, and logic building. It also introduces simple but powerful security validations, making it an impressive second-semester