

High Level Design & Low-Level Design

**Document Control:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
|  | 1.0 | Group 2 | - | | | |  | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

**Index**

1. Introduction ------------------------------------------------ 3

1.1 Intended audience ------------------------------------------------ 3

1.2 Acronyms/Abbreviations -------------------------------------------------3

1.3 Project purpose ------------------------------------------------ 3

1.4 Key project objective ------------------------------------------------ 3

1.5 Project scope and limitation ------------------------------------------------ 3

1.6 Functional overview ------------------------------------------------ 3

1.6.1Header files ------------------------------------------------ 4

1.6.2 Functions ------------------------------------------------ 4

2. Design overview ------------------------------------------------ 4

2.1 Design objective ------------------------------------------------ 5

2.2 Design alternative ------------------------------------------------ 5

2.3 User interface paradigms ------------------------------------------------ 5

2.4 Error detection/ Exceptional Handling ------------------------------------------------ 5

2.5 Performance ------------------------------------------------ 6

2.6 Maintenance ------------------------------------------------ 6

3. System architecture ------------------------------------------------ 6

3.1 Structure ------------------------------------------------6

3.2 Flow chart -------------------------------------------------7

3.3 ER diagram -------------------------------------------------8

4. Environment description -------------------------------------------------9

4.1 Time zone support -------------------------------------------------9

4.2 Language support -------------------------------------------------9

4.3 User desktop requirement -------------------------------------------------9

4.4 Server-side requirement -------------------------------------------------9

4.4.1Deployment consideration -------------------------------------------------9

4.4.2 Integration requirements -------------------------------------------------9

4.4.3 Network -------------------------------------------------9

4.5 Configuration -------------------------------------------------9

4.5.1 Operating system --------------------------------------------------9

5. Reference ------------------------------------------------- 9

1. **Introduction: -**
   1. **Intended Audience: -**

The audience set for this project would include employees working in banking organisations as well as for their clients which are the customers that use services from these organisation.

## Acronyms/Abbreviations: -

|  |  |
| --- | --- |
| ac | account |
| pswd | password |

* 1. **Project Purpose: -**

The banking system is a simple console application without graphics, specifically developed for banking application with facilities of account opening, deposit & withdrawal. Secondly, the application also allows only the authorized bank personnel to transfer money from one account in the same bank to another. Whereas the customers can use this system to create an account , deposit or withdraw amount as well as view balance.

* 1. **Key Project Objectives: -**
* Allow customer to create account.
* Allow customer to do transactions (Withdraw & Deposit Amount).
* Allow customer to check Balance.
* Allow Admin to view, update and delete customer.
* Allow Admin to do transfer among respective accounts.
* Allow Admin to search customer’s account by account number and by name.
  1. **Project scope and limitation: -**

Primarily, the scope of the banking application features to ensure smooth banking operations. The main aim of the application is to automate records on the system. It provides primary functions which are required by the bank in order to run a stable system.One can also check their present transactions that are in process and keep a check on their accounts via this application. It’s not only useful for the customers but also for the admin.

**1.6 Functional Overview: -**

1.6.1 Following header files are included in the program:

* #include <stdio.h>
* #include <string.h>
* #include <stdlib.h>
* #include <pthread.h>

1.6.1 Login

* Customer: Customer logins by entering customer’s account number, phone number & email address.
* Admin: Admin logins by Admin id & phone number as login password.
  + 1. Customer

1.6.2.1 Create Account

The customer can create account by entering name, email address and phone number.

1.6.2.2 Do\_Transaction

1. Deposit: This function will add the deposited amount to the current balance.
2. Withdraw: This function will deduct the withdrawal amount from the current balance.

1.6.2.3 View\_Balance

This function will display the details from customer file using account number.

1.6.3Admin

1.6.3.1 Edit\_Customer\_Details

The Admin can edit the customer’s name, details and balance.

1.6.3.2 Delete\_Customer\_Details

The entire customer record is deleted from database.

1.6.3.3 Do\_Transfer

The banker transfers the amount from source account to destination account.

# 2.Design Overview: -

Banking Application comprises of the following modules:

## Design Objectives: -

* Allow customer to create a new account
* Do Transaction
* View Balance
* Allow admin to edit, update and delete customer.
* Allow admin to do transfer.
* Allow admin to search account by account number and by name.
  1. **Design Alternative****: -**

We have used linked list instead of stack & queue as Insertion and Deletions operations are fast and easier in linked list. Memory allocation is done during run-time. (i.e., no need to allocate any fixed memory.

### User Interface Paradigms: -

The Banking System gives a user an option to have its personal banking application stored on a system file. A system always works faster than a person can. User is given an interface to create a new account in bank, an option to deposit and transfer amount in the account & view balance. A specific set of users are given interface to edit details of the accounts & delete the account, to transfer an amount among respective accounts.

### Error Detection / Exceptional Handling: -

* If the user doesn’t have any pre-existing account, the user has to create one else it won’t perform any functions and would give “not found” or “Invalid entry” error.
* While creating the account, user should first enter the name followed by phone number else it will display “Already exist” and “Invalid length” error for the respective cases. We check the validity of the name, email address & phone number entered with the help of exception handling. If the name entered has the length less than 5 or greater than 25 or the phone number entered is either already existing or of not length 10 digits, an error message will be flashed and asks customer to enter the valid details.

### Performance: -

### The system will work on the user’s terminal. The performance shall depend upon hardware components of the banker/customer and the internet connection

### Maintenance: -

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system’s parts may be required, and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may result in downtime.

**3.SYSTEM ARCHITECTURE: -**

**3.1 Structure Details:**

The system consists of two structures:

* Customer

This structure contains all the definition of all the variables that are present in the Customer Corner Submenu.

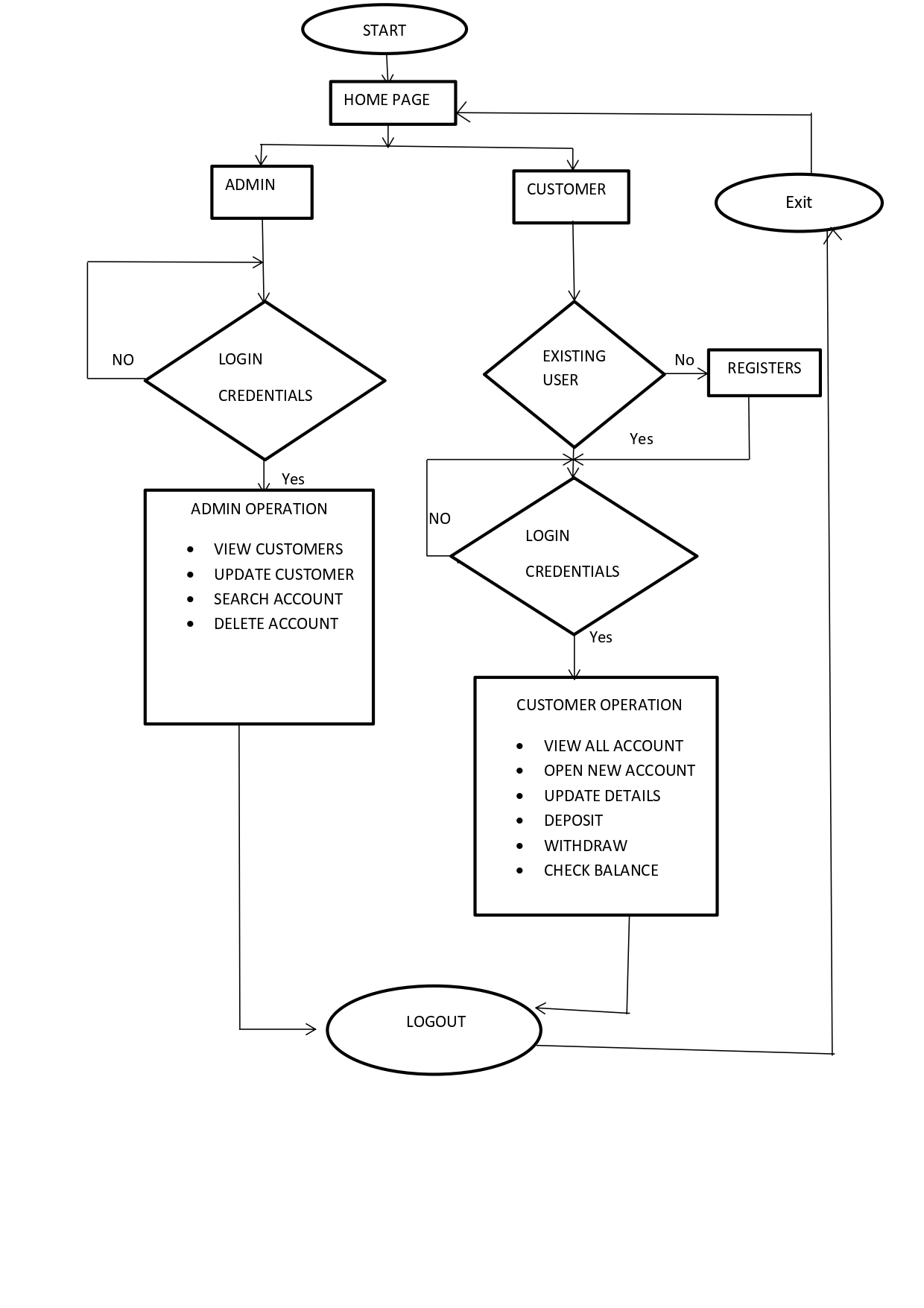
The customer\_id, name have the char data type whereas balance has double and phone number with int data type.

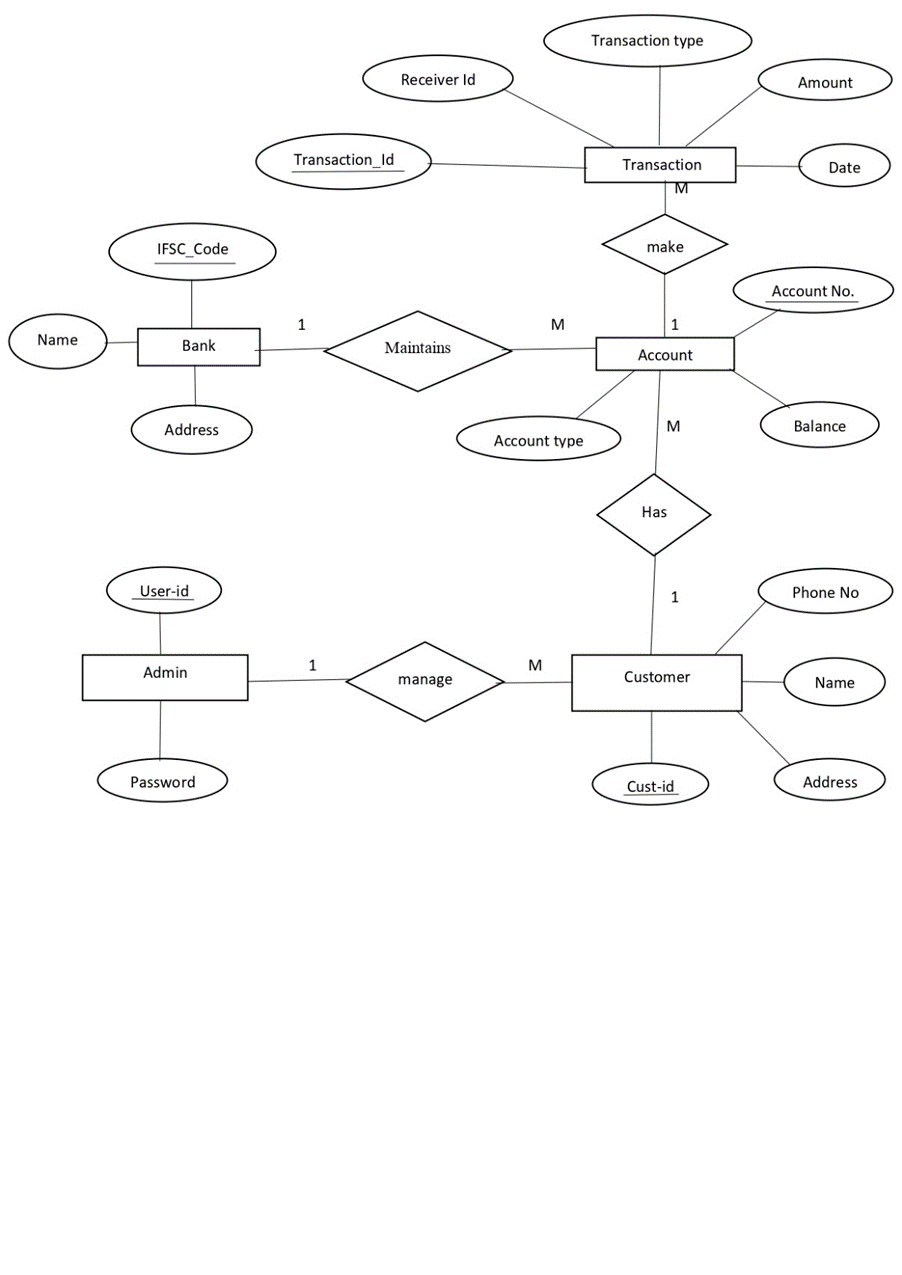
* Transaction

This structure contains all the definition of variables need in Transaction report.The amount

The amount variable has the double data type.

**3.2 Flow chart : -**



**3.3 ER Diagram : - **

1. **Environment Description: -**
   1. **Time Zone Support: -** IST- Kolkata
   2. **Language Support: -** English
   3. **User Desktop Requirements: -**

* 64-bit processor, 1 GHz or faster
* At least 10 GB free hard drive space
* At least 1 GB RAM **Server**
  1. **-Side Requirements: -**
* 64-bit processor, 1 GHz or faster
* At least 2 GB free hard drive space
* At least 1GB RAM

4.4.1.Deployment Considerations: -

* + Local storage is used
  + No network latency to consider
  + To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware

**4.4.2**. Integration Requirements: -

* Language: - C
* Tools: - Valgrind ,Makefile ,Strace
* Complier: - gcc
* Linux Environment

**4.4.3**. Network: - End to End

**4.4 Configuration: -**

**4.4.1**. Operating System: - Linux environment

1. **Reference: -**

The references are:

* https://www.programiz.com/dsa/linked-list
* https://www.javatpoint.com/file-handling-in-c
* https://www.educative.io/answers/how-to-create-a-simple-thread-in-c