



BANKING MANAGEMENT SYSTEM

Mini Project Report

Submitted as mini project in the subject of Data Structures Lab in Computer Engineering

by

**Wahab Shaikh
Rayees Shaikh
Mitul Patel**

Supervisor
Prof. Jignasha Dalal



Department of Computer Engineering

K. J. Somaiya Institute of Engineering and Information Technology

Ayurvihar, Sion, Mumbai - 400022

2019

MINI PROJECT APPROVAL

This project report entitled Banking Management System by

Wahab Shaikh
Rayees Shaikh
Mitul Patel

is approved for the Mini Project work of Data Structures Lab in Computer Engineering.

Supervisor

Prof. Jignasha Dalal

Date:

Place:

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Wahab Shaikh _____

Rayees Shaikh _____

Mitul Patel _____

DATE:

ACKNOWLEDGEMENT

We would like to convey our sincere thanks to many people who guided us throughout the course for this work. First, we would like to express our sincere thanks to our beloved Principal **Dr. SURESH UKARANDE** for providing various facilities to carry out this report.

We would like to express our sincere thanks to **Prof. JIGNASHA DALAL** for her guidance, encouragement, co-operation and suggestions given to us at progressing stages of report.

Finally, we would like to thank our H.O.D. **Prof. SARITA AMBADEKAR** and all teaching, non-teaching staff of the college and friends for their moral support rendered during the course of the report work and for their direct and indirect involvement in the completion of our report work, which made our endeavor fruitful.

Wahab Shaikh

Rayees Shaikh

Mitul Patel

ABSTRACT

In this project, we are attempting to make a Banking Management System using Linked List, Structures, File Handling in C language. We shall be using Singly Linked List to store and work on data.

Linked List is a linear data structure. Unlike arrays, linked list elements are not stored in contiguous memory locations; the elements are linked using pointers. We can also easily access each element of linked list. The linked list is used to store the account number and the name of the account holder, account pin, age of account holder, address and account balance.

When the program is executed, a menu is displayed on the screen using which the user can select whether they want to login as staff or as a customer. This is implemented by using switch case statement.

After selecting the option of their preference, a list of actions that could be taken is displayed to the user. The staff login option includes creating a new account, displaying the created account details, removing an already created account, searching for an account using the account number, editing the account details, depositing or withdrawing a certain amount in an account. The customer login option includes displaying account details and carrying out a transaction. Hence using features from various data types to functions of C language we have created a user-friendly program that solves the real time problems. More features can be added and program can be made to make real enterprise software.

CONTENTS

Sr. No.	Name	Page No.
1	Problem Definition	1
2	Theory	2
3	Functions and Algorithms	6
4	Conclusion	9
5	References	10

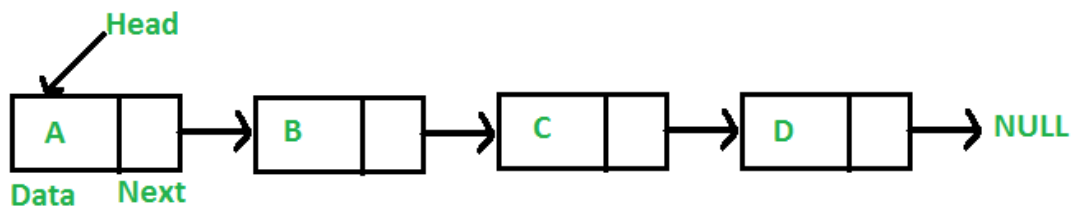
PROBLEM DESCRIPTION

Create a program to display a menu that allows the user to select between staff and customer login option. After selecting staff login option from the main menu, the user will have to enter a pre-assigned password to go ahead. After that the program will continue to display a sub-menu that will allow you to create a new account, display all the existing accounts details, searching for a specific account using account number and logout to exit from staff login. If the user selects customer login option from the main menu, the user will be asked for a password assigned while creating the account to go ahead. Further it will display a sub menu that will allow you to display your account details and carry out a transaction of sending money to some other bank account using account number only if sufficient balance is available and finally logout. The program will continue to allow the user to carry out various banking tasks until they select LOGOUT option, and then only the program will display the main list with the Staff login, customer login and exit option.

THEORY

LINKED LIST:

A linked list is a linear data structure, in which the elements are not stored in contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

Why Linked List?

Arrays can be used to store linear data of similar types, but arrays have the following limitations:

- 1) The size of the arrays is fixed. So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage.
- 2) Inserting a new element in an array of elements is expensive because the room has to be created for the new elements and to create room existing elements have to be shifted.

Representation:

A linked list is represented by a pointer to the first node of the linked list. The first node is called the head. If the linked list is empty, then the value of the head is NULL.

Each node in a linked list consists of at least two parts:

- 1) Data
- 2) Pointer (or Reference) to the next node

In C, we can represent a node using structures.

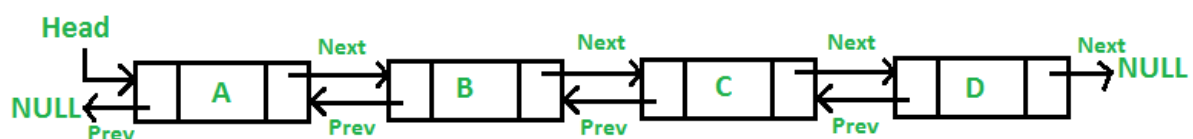
In Java or C#, LinkedList can be represented as a class and a Node as a separate class. The LinkedList class contains a reference of Node class type.

Linked list is mainly divided as:

- Singly linked list
- Doubly linked list
- circular linked list

Singly Linked List: The simplest kind of linked list is a **Singly Linked List (SLL)** which has one link per node. It has two parts, one part contains data and other contains address of the next node.

Doubly Linked List: A **Doubly Linked List (DLL)** contains an extra pointer, typically called *previous pointer*, together with next pointer and data which are there in a singly linked list.



This structure allows for efficient insertion or removal of elements from any position in the sequence. A drawback of linked list is random access is not allowed. We have to access elements sequentially starting from the first node. So we cannot do binary search with linked lists efficiently with its default implementation.

STRUCTURES:

A **struct** in the C programming language (and many derivatives) is a composite data type declaration that defines a physically grouped list of variables to be placed under one name in a block of memory, allowing the different variables to be accessed via a single pointer. To define a structure, you must use the **struct** statement. The struct statement defines a new data type, with more than one member.

To access any member of a structure, we use the **member access operator (.)**. The member access operator is coded as a period between the structure variable name and the structure member that we wish to access. You would use the keyword **struct** to define variables of structure type.

For example:

```
struct address
{
    char name[50];
    char street[100];
    char city[50];
    char state[20];
    int pin;
};
```

FILES IN C:

Files can be accessed in C with the use of FILE pointers. To open a file, use this syntax:

```
FILE *fp = fopen(char filename[], char mode[]);
```

where, 'filename' is the name of the file and 'mode' is the way to open the file, e.g. for reading, writing, appending etc.

Examples of modes are "r" for reading, "w" for writing a fresh file, "a" for appending to an old file.

Text Files

Text files in C are straightforward and easy to understand. All text file functions and types in C come from the **stdio** library.

When you need text I/O in a C program, and you need only one source for input information and one sink for output information, you can rely on **stdin** (standard in) and **stdout** (standard out). There are six different I/O commands in <stdio.h> that you can use with stdin and stdout:

- **printf** - prints formatted output to stdout
- **scanf** - reads formatted input from stdin
- **puts** - prints a string to stdout
- **gets** - reads a string from stdin
- **putc** - prints a character to stdout
- **getc**, **getchar** - reads a character from stdin

The advantage of stdin and stdout is that they are easy to use. Likewise, the ability to redirect I/O is very powerful. For example, to create a program that reads from stdin and counts the number of characters.

FUNCTIONS

List of functions used in this program:

- **main()**

The main menu is displayed in the main function that has various options like staff login, customer login and exit. It consists of switch case which allows the user to make a choice and proceed with the sub-menus.

- **staff_login()**

This function displays the menu of tasks that can be performed by the bank staff like creating, deleting, updating accounts. After selecting staff login from main menu the user will be asked for a pin to enter as a staff.

- **user_login()**

This function displays the menu of tasks that can be performed by the bank account holder. After selecting user login from main menu the user will be asked for account number and a pin assigned to the user's account while creating the account. After successful input of pin the user can carry out monetary transactions or view account details.

- **create_account()**

In the staff login the user gets the option of creating a new account. The user will be allowed to enter a number which will act as account number for the customer and would be used for all further transactions. The user will also have to create a pin for that particular account. Account holder's details like name, age, address

initial deposit will be entered using this function. The details of the newly created account will be stored in linked list. Which is then stored into a file.

- **display_list()**

In the staff login the user gets the option of displaying the list of all the existing accounts and their details. This function creates a file pointer and opens the file in read mode and displays its contents. If the file is empty the function prints as “No records found!”. And later exit the function.

- **search_account()**

In the staff login the user gets the option of searching a particular account using the account number assigned while creating the account as input. If the account number is not found in the file then the function displays an error message.

- **remove_account()**

In the staff login the user gets the option of deleting an account permanently. The function will take account number as the input. Further the pin assigned to that account number while creating the account will be asked, if right pin is entered the account will be deleted permanently else an error message will be prompted.

- **edit_account()**

In the staff login the user gets the option of editing an already created account's details. The function will ask for the account number of which the details are to be edited. It will further show a menu of things that could be edited. After successful

update a success message will be prompted. The updated details will be updated in file as well.

- **deposit()**

In the staff login the user gets the option of depositing amount in a particular bank account. The function will ask for the account number and then the amount to be deposited. After deposition the updated account balance will be displayed. The updated balance will be updated in file as well.

- **withdraw()**

In the staff login the user gets the option of withdrawing amount from a particular bank account. The function will ask for the account number and then the amount to be withdrawn. If the amount to be withdrawn exceeds the current balance an error of insufficient balance will be displayed else the updated balance will be displayed. The updated balance will be updated in file as well.

- **view_account()**

In the customer login the user gets the option of viewing account details.

- **transact()**

In the customer login the user gets to carry out monetary transactions i.e. send money to another bank account holder. The function inputs the account number of the account in which the money is to be sent and also the amount. If the amount to

be sent is greater than the current balance then an insufficient balance error is displayed; else the updated balance will be displayed.

File Handling

- **fopen()**

This function is used to open a file to perform operations such as reading, writing etc.

- **fprintf()**

fprintf() function writes formatted data into a file.

- **fclose()**

fclose() function closes the file that is being pointed by file pointer.

CONCLUSION

The attempt at preparing a Banking Management System was successful, making it very easy for the user to choose between various options. Also the details of all the accounts are saved in a file.

Although all required objectives have been met, the system still has room for improvement. The system is flexible enough for future upgrade using advanced technology and devices.

This project can serve as a base for banking management in any bank.

This project helped us to understand the process of Project Development and the issues faced while working on a team project such as code synchronization, task assignment and inter-member conflicts on the project.

REFERENCES

- 1) "Programming in C" by Reema Thareja.
- 2) Geeks For Geeks.
- 3) stackoverflow.com