**Project**



**Fatima Jinnah Women University , Rawalpindi**

**Bank Management System**

**SUBMITED TO:**

“Mrs Sobia Khalid”

**SUBMITTED BY:**

Hala Ali Khan 19-BSE-007

Marukh Rehman 19-BSE-037

Eisha ter raazia mir 19-BSE-005

***Report***

Our Project is about ***"Bank Management System"*** using double linked list. In this program we have used creation of linked list, insertion and deletion of nodes through searching.

**Creation :**

In the creation of linked list the manager will ask the user and input all the required fields for the account to be created.

User will be asked about the type of account.

Moreover, system provides the facility to create more than one accounts at the same runtime.

**Insertion:**

Insertion will be done by user's desire at a specific position in the log. This field also asks for the type of account. After insertion user axcount at a specific location will be made.

**Deletion:**

Deletion is only allowed to specific personnal. Like managers or admins. To delete the account passcode is required. The passcode will allow the deletion of specific account.

**Main:**

In the main body the switch statement is used which will give user the option to select a task and the system will perform it. It asks the user to enter data till the user desires to***.***

***Program***

// bms.cpp : Defines the entry point for the console application.

//Group members : Hala Ali Khan, Eisha ter raazia mir ,Marukh Rehman

#include "stdafx.h"

#include <iostream>

#include<string>

using namespace std;

class Bank //Class

{

private:

struct node //Structure Node

{

int pin;

int id;

char acc\_num [13];

char holder[100];

char branch[100];

node\* next;

node\* prev;

};

node\* head , \* tail;

public:

Bank() //Default Constructor

{

head = NULL;

tail = NULL;

}

int length() //Counts Node in the Linked List

{

node\* len;

len = head;

int count = 0;

while (len != NULL)

{

count++;

len = len->next;

}

return count;

}

void create() //Create Nodes

{

int opt;

char choice;

node \*newnode;

do

{

newnode = new node;

cout << "Enter Account Holder Name :: ";

cin >> newnode->holder;

cout << "Enter Account Holder ID :: ";

cin >> newnode->id;

cout << "Enter Account Number :: ";

cin >> newnode->acc\_num;

cout << "Enter PIN number :: ";

cin >> newnode->pin;

cout << "Enter Branch Name :: ";

cin >> newnode->branch;

newnode->next = NULL;

newnode->prev = NULL;

if (head == NULL)

{

head = tail = newnode;

}

else

{

tail->next = newnode;

newnode->prev = tail;

tail = newnode;

}

cout << "Press 1 for saving account ::\nPress 2 for current account ::\nEnter Choice :: ";

cin >> opt;

if (opt == 1)

{

saving\_account();

}

else if (opt == 2)

{

current\_account();

}

else

cout << "\nINVALID ENTRY!\n";

cout << "\nDo you want to insert more nodes? (Y/N)\n";

cin >> choice;

} while (choice == 'Y' || choice == 'y');

}

void insert\_beg() //Inserts Node at Beginning

{

node\* newnode;

int opt;

newnode = new node;

cout << "Enter Account Holder Name :: ";

cin >> newnode->holder;

cout << "Enter Account Holder ID :: ";

cin >> newnode->id;

cout << "Enter Account Number :: ";

cin >> newnode->acc\_num;

cout << "Enter PIN number :: ";

cin >> newnode->pin;

cout << "Enter Branch Name :: ";

cin >> newnode->branch;

newnode->next = NULL;

newnode->prev = NULL;

if (head == NULL)

{

create();

}

else

{

head->prev = newnode;

newnode->next = head;

head = newnode;

}

cout << "Press 1 for saving account ::\nPress 2 for current account ::\nEnter Choice :: ";

cin >> opt;

if (opt == 1)

{

saving\_account();

}

else if (opt == 2)

{

current\_account();

}

else

cout << "\nINVALID ENTRY!\n";

}

void insertatPos() //Inserts Node at a Specific Position

{

int pos, opt;

cout << "\nEnter the position at which you want to insert the account :: ";

cin >> pos;

if (pos<1 && pos>length())

{

cout << "\nInvalid Position\n";

}

else if (pos == 1)

{

insert\_beg();

}

else

{

node\* newnode, \*ptr;

ptr = head;

newnode = new node;

cout << "Enter Account Holder Name :: ";

cin >> newnode->holder;

cout << "Enter Account Holder ID :: ";

cin >> newnode->id;

cout << "Enter Account Number :: ";

cin >> newnode->acc\_num;

cout << "Enter PIN number :: ";

cin >> newnode->pin;

cout << "Enter Branch Name :: ";

cin >> newnode->branch;

for (int i = 1; i < pos - 1; i++)

{

ptr = ptr->next;

}

newnode->prev = ptr;

newnode->next = ptr->next;

ptr->next = newnode;

newnode->next->prev = newnode;

}

cout << "Press 1 for saving account ::\nPress 2 for current account ::\nEnter Choice :: ";

cin >> opt;

if (opt == 1)

{

saving\_account();

}

else if (opt == 2)

{

current\_account();

}

else

cout << "\nINVALID ENTRY!\n";

}

void del\_beg() //Deletes Node from beginning

{

node\* ptr;

if (head == NULL)

{

cout << "\nLIST IS EMPTY!\n";

}

else

{

ptr = head;

head = head->next;

head->prev = NULL;

ptr->next = NULL;

cout << "\nThe deleted ID is :: " << ptr->id << endl;

delete ptr;

}

}

void delfromPos() //Deletes Node from a Specific Position

{

int pos;

cout << "\nEnter the position from which you want to delete the account :: ";

cin >> pos;

if (pos<1 && pos>length())

{

cout << "\nInvalid Position\n";

}

else if (pos == 1)

{

del\_beg();

}

else

{

node \* ptr;

ptr = head;

for (int i = 1; i < pos; i++)

{

ptr = ptr->next;

}

ptr->prev->next = ptr->next;

ptr->next->prev = ptr->prev;

cout << "\nThe deleted ID is :: " << ptr->id << endl;

delete ptr;

}

}

void traverse() //Displays the Linked List

{

node\* ptr;

ptr = head;

while (ptr!= NULL)

{

cout<< ptr->holder;

cout << endl;

cout << ptr->id;

cout << endl;

cout << ptr->acc\_num;

cout << endl;

cout << ptr->pin;

cout << endl;

cout << ptr->branch;

cout << endl;

cout << endl;

ptr = ptr->next;

}

}

void saving\_account()

{

double amount, interest;

cout << " Enter the amount :: ";

cin >> amount;

if (amount < 10000)

{

amount = amount + (300);

cout << " Your amount after a year would be ::" << amount << endl;

}

else

{

amount = amount + (600);

cout << " Your amount after a year would be ::" << amount << endl;

}

}

void current\_account()

{

int ch;

double amount = 0;

cout << " This is a newly opened account. " << endl;

cout << " Your balance is 0. " << endl;

cout << " Press 1 if you want to add amount in the bank " << endl;

cout << " Choice :: ";

cin >> ch;

if (ch == 1)

{

cout << " Enter the amount :: ";

cin >> amount;

cout << " Now your account balance is :: " << amount << endl;

}

else

{

cout << " Invalid choice " << endl;

}

}

};

int main()

{

Bank b1;

int opt;

int passcode, enter = 13579;

char ch;

cout << "\*\*WELCOME TO BANK MANAGEMENT SYSTEM\*\*\n";

xyz:

cout << "Select from the below given options to start!\n";

cout << "\nPress 1 to Create your Account ::\nPress 2 to Place your Account on the desied place of the log ::\nPress 3 to delete your Account (Only For Admins)::\nPress 4 to EXIT ::\n";

cout << "Enter your Choice to Start!\n";

cin >> opt;

switch (opt)

{

case 1:

{

cout << "\nCREATE YOUR ACCOUNT\n";

b1.create();

b1.traverse();

break;

}

case 2:

{

cout << "\nINSERT YOUR ACCOUNT\n";

b1.insertatPos();

b1.traverse();

break;

}

case 3:

{

cout << "\nDELETE ACCOUNT\n";

cout << "Enter the Passcode ::\n";

cin >> passcode;

if (passcode == enter)

{

b1.delfromPos();

b1.traverse();

}

else

cout << "SECURITY ALERT! YOU HAVE ENTERED WRONG PASSCODE!\n";

break;

}

case 4:

{

cout << "\nEXIT FROM MENU\n";

cout << "THANK YOU!:)\n";

break;

}

default:

{

cout << "INVALID ENTRY!\n";

}

}

cout << "Do you want to return to the Menu? (Y/N)\n";

cin >> ch;

if (ch == 'Y' || ch == 'y')

goto xyz;

else

cout << "THANK YOU FOR YOUR VISIT!\n";

}

**Output :**









