بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

FATIMA JINNAH WOMEN UNIVERSITY, RAWALPINDI

DATASTRUCTURES AND ALGORITHM (BSE-103)

END SEMESTER PROJECT SUBMITTED TO DR. SIDRA EJAZ

DEPARTMENT OF SOFTWARE ENGINEERING
SECTION B
BY
SYEDA FARWA BATOOL (BSE-2022-071)

RAWALPINDI, PAKISTAN
DECEMBER 17, 2024

Explanation

1. Node Class:

- Represents an account in the bank.
- Contains attributes such as name, PIN, phone number, email, amount, account number, and pointers to left and right child nodes.
- Uses a static variable `accountCounter` to assign a unique account number to each account.
- **Constructor (Node()):**
  - Initializes the attributes of a new Node object. It sets default values for attributes such as name, PIN, phone number, email, amount, and assigns a unique account number using the static accountCounter.

2. Bank Class:

- Manages the banking system with operations on accounts.
- Uses a binary search tree structure to efficiently organize and search for accounts based on their PINs.
- Provides functions for creating an account, depositing money, withdrawing money, updating account information, and displaying account details.
- **Constructor (Bank()):**
  - Initializes the Bank object with a null root, indicating an empty binary search tree.
- **isDuplicate(Node* temp):**
  - Checks for duplicate accounts by traversing the binary search tree.
  - Compares the PIN, phone number, and email of the given temp node with existing nodes in the tree.
  - Returns true if a duplicate is found, indicating that the user needs to provide a different PIN, phone number, or email during account creation.
- **createAccount():**
  - Creates a new account and adds it to the binary search tree.
  - Asks the user for account details (name, PIN, phone number, email) and checks for duplicates using the isDuplicate function.
  - Generates a unique account number and assigns it to the new account.
  - Asks the user if they want to deposit money into the account.
  - Inserts the new account into the binary search tree based on the PIN.
- **depositAmount():**
  - Allows a user to deposit money into their account.
  - Prompts the user to enter their PIN and the amount they want to deposit.
  - Searches for the account with the provided PIN in the binary search tree.
  - If the account is found, updates the account balance by adding the deposited amount.
- **withdraw():**
  - Allows a user to withdraw money from their account.
  - Prompts the user to enter their PIN and the amount they want to withdraw.
  - Searches for the account with the provided PIN in the binary search tree.
  - If the account is found and has sufficient funds, updates the account balance by subtracting the withdrawn amount.
- **update():**
  - Allows a user to update their account information (email or phone number).
  - Prompts the user to enter their PIN and choose the information to update.

- Searches for the account with the provided PIN in the binary search tree.
- If the account is found, updates the specified information.

- **displayInfo():**
  - Displays account information for a user.
  - Prompts the user to enter their PIN and searches for the account with the provided PIN in the binary search tree.
  - If the account is found, prints the account details (account number, name, phone number, email, amount).

3. Static Variable:

- The use of `static` for `accountCounter` ensures that each instance of the `Node` class shares the same counter. It is associated with the class rather than instances, so all objects of the class share the same static member.

4. User Interaction:

- The program interacts with users through a console-based menu.
- Users can choose from options like creating an account, depositing money, withdrawing money, updating information, and displaying account details.
- The program uses loops to allow users to perform multiple operations in a single run.

5. Binary Search Tree:

- The program utilizes a binary search tree to efficiently manage accounts based on their PINs.
- When creating a new account, the program traverses the tree to find the correct position for insertion.
- The tree structure makes it faster to search for and access accounts based on PINs.

6. Input Handling:

- The program handles user inputs using `cin` and checks for duplicate information to maintain unique accounts.
- For inputting strings with spaces (like email), the `getline` function is used after appropriate calls to `cin.ignore` to handle newline characters left in the buffer.

Code

```
#include <iostream>
#include <string>

using namespace std;

class Node
{ public:
   string name;
   int PIN;
   int    ph_num;
   string    email;
   float amount;
   int accountNumber;
   Node* left;
   Node* right;

   static int accountCounter;

   /*
The use of static for accountCounter in this context is to ensure that each instance of the Node
class shares the same counter. A static member variable is associated with the class rather than
with instances of the class. This means that all objects of the class share the same static member
            */
   Node() : name("N/A"), PIN(0), ph_num(0), email("N/A"), amount(0.0), left(NULL), right(NULL)
      { accountNumber = ++accountCounter;
   }
};
//initializing static member
int Node::accountCounter = 0;
```

```
class Bank
{ private:
   Node* root;

   bool isDuplicate(Node* temp)
      { Node* current = root;

      while (current != NULL) {
         if (temp->PIN == current->PIN || temp->ph_num == current->ph_num || temp->email == current->email)
            { return true;  // Duplicate found
         }

         if (temp->PIN < current->PIN)
            { current = current->left;
         } else {
             current = current->right;
         }
      }

       return false;  // No duplicate
   }

public:
   Bank() : root(NULL) {}

   void createAccount()
   { Node* temp = new Node;

   cout << "Enter your Name: ";
   cin.ignore();
   getline(cin, temp->name);
```

```
cout << "Enter your PIN: ";
cin >> temp->PIN;

while (isDuplicate(temp)) {
   cout << "This PIN is being used by another account. Please change your PIN: ";
   cin >> temp->PIN;
}

cout << "Enter your Phone Number: ";
cin >> temp->ph_num;

while (isDuplicate(temp)) {
   cout << "This Phone Number is being used by another account. Please change your Phone Number: ";
   cin >> temp->ph_num;
}

cout << "Enter your Email Address: ";
cin.ignore(); // Ignore the newline character left in the buffer
getline(cin, temp->email);

while (isDuplicate(temp)) {
   cout << "This Email is being used by another account. Please change your Email: ";
   getline(cin, temp->email);
}

cout << "Your Account has been created. Account Number: " << temp->accountNumber << endl;

int choice;
cout << "Do you want to deposit amount?\n1. Yes\n2. No\n";
cin >> choice;
```

```
if (choice == 1) {
   cout << "Enter the amount: ";
   cin >> temp->amount;
   cout << "Your Amount has been deposited!\nThank You!\n";
} else {
   cout << "OK. Thank You!\n";
}

// Insert directly within the createAccount function
if (root == NULL) {
   root = temp;
} else {
   Node* current = root;
   while (current != NULL) {
        if (temp->PIN < current->PIN)
          { if (current->left == NULL)
             { current->left = temp;
              break;
          } else {
             current = current->left;
          }
       } else {
          if (current->right == NULL)
             { current->right = temp;
             break;
          } else {
             current = current->right;
          }
       }
   }
```

```
    }
}



    void depositAmount()
        { int pin;
        cout << "Enter your PIN: ";
        cin >> pin;

        Node* temp = root;
        while (temp != NULL) {
            if (temp->PIN == pin)
                { int newAmount;
                cout << "Enter the amount you want to deposit: ";
                cin >> newAmount;
                temp->amount += newAmount;
                cout << "Your Amount has been deposited!\nTotal Amount in your Account= " << temp->amount << "\nThank
You!\n";
                return;
            }
            if (pin < temp->PIN)
                { temp = temp->left;
            } else {
                temp = temp->right;
            }
        }

        cout << "Sorry, we could not find your account.\n";
    }

    void withdraw() {
```

```
    int pin;
    cout << "Enter your PIN: ";
    cin >> pin;

    Node* temp = root;
    while (temp != NULL) {
      if (temp->PIN == pin)
        { int withdrawAmount;
        cout << "Enter the amount you want to withdraw: ";
        cin >> withdrawAmount;

        if (temp->amount >= withdrawAmount)
          { temp->amount -= withdrawAmount;
          cout << "Your Amount has been withdrawn!\nTotal Amount in your Account= " << temp->amount <<
"\nThank You!\n";
        } else {
          cout << "You don't have sufficient amount in your account.\n";
        }
        return;
      }
      if (pin < temp->PIN)
        { temp = temp->left;
      } else {
        temp = temp->right;
      }
    }

    cout << "Sorry, we could not find your account.\n";
  }

  void update() {
```

```
int pin;
cout << "Enter your PIN: ";
cin >> pin;

Node* temp = root;
while (temp != NULL) {
   if (temp->PIN == pin)
      { int choice;
      cout << "What do you want to update:\n1. Email\n2. Phone Number\n";
      cin >> choice;

      if (choice == 1) {
         cout << "Enter new email: ";
         cin >> temp->email;
         cout << "Your email address has been updated\n";
      } else if (choice == 2) {
         cout << "Enter new Phone number: ";
         cin >> temp->ph_num;
         cout << "Your phone number has been updated\n";
      } else {
         cout << "Invalid Entry\n";
      }
      return;
   }
   if (pin < temp->PIN)
      { temp = temp->left;
   } else {
      temp = temp->right;
   }
}
```

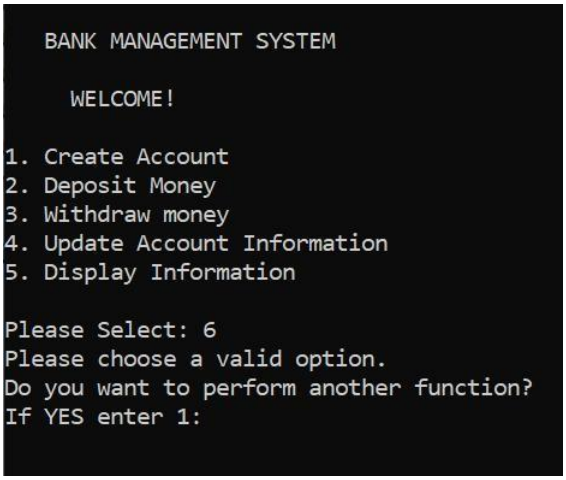
```
        cout << "Sorry, we could not find your account.\n";
    }

    void displayInfo()
        { int pin;
        cout << "Enter your PIN: ";
        cin >> pin;

        Node* temp = root;
        while (temp != NULL) {
            if (temp->PIN == pin) {
                cout << "Account Number: " << temp->accountNumber << endl;
                cout << "Name: " << temp->name << endl;
                cout << "Phone Number: " << temp->ph_num << endl;
                cout << "Email: " << temp->email << endl;
                cout << "Amount: " << temp->amount << endl;
                return;
            }
            if (pin < temp->PIN)
                { temp = temp->left;
            } else {
                temp = temp->right;
            }
        }

        cout << "Invalid PIN\n";
    }
};

int main()
    { int
    input;
```

```
Bank b;

cout << "\n   BANK MANAGEMENT SYSTEM\n\n";
cout << "     WELCOME!\n\n";

do {
   cout << "1. Create Account\n";
   cout << "2. Deposit Money\n";
   cout << "3. Withdraw money\n";
   cout << "4. Update Account Information\n";
   cout << "5. Display Information\n\n";
   cout << "Please Select: ";
   cin >> input;

   switch (input)
      { case 1:
         b.createAccount();
         break;
      case 2:
         b.depositAmount();
         break;
      case 3:
         b.withdraw();
         break;
      case 4:
         b.update();
         break;
      case 5:
         b.displayInfo();
         break;
      default:
```

```
            cout << "Please choose a valid option.\n";
        }

        int choice;
        cout << "Do you want to perform another function?\nIf YES enter 1: ";
        cin >> choice;

        if (choice != 1)
           { break;
        }
    } while (true);

    return 0;
}
```

Output

When choose invalid option



Create Account

```
   BANK MANAGEMENT SYSTEM

     WELCOME!

1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 1
Enter your Name: sobia
Enter your PIN: 123
Enter your Phone Number: 0300
Enter your Email Address: sobia@gmail.com
Your Account has been created. Account Number: 1
Do you want to deposit amount?
1. Yes
2. No
1
Enter the amount: 10
Your Amount has been deposited!
Thank You!
Do you want to perform another function?
If YES enter 1:
```

Two accounts can not have same PIN, Phone number or Email address

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 1
Enter your Name: noor
Enter your PIN: 123
This PIN is being used by another account. Please change your PIN: 321
Enter your Phone Number: 0300
This Phone Number is being used by another account. Please change your Phone Number: 0321
Enter your Email Address: sobia@gmail
This Email is being used by another account. Please change your Email: noor@gmail
Your Account has been created.
```

Deposit Money

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 2
Enter your PIN: 123
Enter the amount you want to deposit: 30
Your Amount has been deposited!
Total Amount in your Account= 50
Thank You!
Do you want to perform another function?
If YES enter 1:
```

Withdraw Money Function when you don't have sufficient amount

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 3
Enter your PIN: 321
Enter the amount you want to withdraw: 321
You don't have sufficient amount in your account.
```

Withdraw Function in normal case

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 3
Enter your PIN: 123
Enter the amount you want to withdraw: 10
Your Amount has been withdrawn!
Total Amount in your Account= 40
Thank You!
```

Updating Email Address

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 4
Enter your PIN: 123
What do you want to update:
1. Email
2. Phone Number
1
Enter new email: sobia@gmail.com
Your email address has been updated
```

Updating Phone Number

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 4
Enter your PIN: 432
What do you want to update:
1. Email
2. Phone Number
2
Enter new Phone number: 0200
Your phone number has been updated
```

Displaying Information

```
1. Create Account
2. Deposit Money
3. Withdraw money
4. Update Account Information
5. Display Information

Please Select: 5
Enter your PIN: 123
Account Number: 1
Name: khush bakht
Phone Number: 121
Email: Sobia@
Amount: 10
Do you want to perform another function?
If YES enter 1:
```