



Project: Banking System

Names: -

- 1- Mostafa Mahmoud Mohamed Salem (Section 3)
- 2- Nurhan Allam Mohammed Mohammed (Section 3)
- 3- Wa'el Ahmad "Muhammad Wahid El-Din" Ahmad (Section3)

Description:

Banking system software will be provided as a tool for all employees in the bank. This software will work for Accounts Information. Employees will be able to help our Clients in many different things as:

- Create a new account: creating an account by taking all the details and the information needed from the user, so every and each client will have his own account with a unique username and password for security purposes.
- Withdraw: Our Clients will be able to withdraw from their bank accounts by taking the original balance into consideration.
- Deposit: an option to deposit amount to the account.
- Transfer Money: They will be able as well to transfer money from their bank account.
- Take a Loan: Depending on the original balance that the client has , he will have the chance to request for a loan in specific range.
- Search: the option to search for the user details by a given account number.
- Update Account and close account.

Data Structures and Algorithms Usage:

The system will need some data structure to store the Accounts of clients and their internal details including:

- Account holder name.
- Password.
- Phone Number.
- Date of Birth.
- Current Balance.
- Account number.
- Account type (Checking or Savings).

The data structure to be used for this purpose is a **Doubly Linked List** which is implemented generically using the concept of **Template Classes** and **Template Functions**, so it can hold any type of data including primitive types and objects (instances of classes).

The reasons for this choice are:

- It can hold unlimited number of elements, which is a needed functionality, to create and delete accounts for example. No need to define an initial size.
- Elements of the linked list are not stored in contiguous memory locations. Instead, they are stored in disperse locations each accessed by pointers of next or previous elements.
- Inserting and erasing an element are easily implemented.
- No space is wasted compared to static and dynamic arrays.
- It is a dynamic structure, meaning it can allocate and deallocate memory during runtime.

Another list is needed to store employees and clients' data.

The algorithms to be used in the project include **Searching** and **sorting** algorithms, some **recursive** algorithms.

Classes used:

- Employee
- Account

Class Employee

```
class Employee {
private:
    string username;
    string password;

public:
    // Constructors and destructor

    Employee() : username(""), password("") {}

    Employee(const string &username, const string &password) : username(username), password(password) {}

    // Getters and setters

    const string &getUsername() const {
        return username;
    }

    void setUsername(const string &username) {
        this->username = username;
    }

    const string &getPassword() const {
        return password;
    }

    void setPassword(const string &password) {
        this->password = password;
    }
}
```

Class Account

```
class Account {  
    private:  
        string accNumber;  
        char accType;  
        string firstName;  
        string surName;  
        string lastName;  
        string phoneNum;  
        string username;  
        string password;  
        string dateOfBirth;  
        double balance;  
  
    public:  
        // Constructors and destructor (create_account() will replace constructor)  
  
        Account();  
  
        Account(const string &accNumber, char accType, const string &firstName, const string &surName,  
                const string &lastName, const string &phoneNum, const string &username, const string &password,  
                const string &dateOfBirth, double balance);  
  
        //////////////////////////////////////  
        // Setters and getters  
  
        const string &getAccNumber() const;  
  
        void setAccNumber(const string &accNumber);  
  
        char getAccType() const;  
  
        void setAccType(char accType);  
};
```

```
void setFirstName(const string &firstName);  
  
const string &getSurName() const;  
  
void setSurName(const string &surName);  
  
const string &getLastName() const;  
  
void setLastName(const string &lastName);  
  
const string &getPhoneNum() const;  
  
void setPhoneNum(const string &phoneNum);  
  
const string &getUsername() const;  
  
void setUsername(const string &username);  
  
const string &getPassword() const;  
  
void setPassword(const string &password);  
  
const string &getDateOfBirth() const;  
  
void setDateOfBirth(const string &dateOfBirth);  
  
double getBalance() const;  
  
void setBalance(double balance);
```

➤ Test Cases: -

- Login as an employee
Get username and password from "Employees File.txt"
This menu will appear: -

```
*****
                                Bank Management System
*****

Enter your login data:
Username:      rana
Password:      11114444

Login Successful!

Choose one of the following:
  1. Create Account
  2. Deposit
  3. Withdraw
  4. Transfer Money
  5. Inquire Balance
  6. Display Account Info
  7. Show Transactions History
  8. Close Account
  9. Quit Program
```

- Create Account

```
1
Account Creation Form:
First Name: Walied
Sur Name: Ahmad
Last Name: Muhammad
Choose Account Type:   'C' for Checking Account or 'S' for Savings Account: s
Birth Date:           (DD/MM/YYYY): 14/01/1887
Mobile Number: 01006294274
Enter Starting Balance to Create Account:
(Note: Minimum Starting balance is $5000) 7000
```

- Show details about this account
by entering account number

```
Enter Account Number: 507307552
Account Number: 507307552
Account Type: S
Name: Walied Ahmad Muhammad
Phone Number: 01006294274
Username: waliedmuhammad
Date of Birth: 14/01/1887
Balance: 20000
```

- Transfer money from client to another
By entering account number for each client and the value of
money

```
Enter Account Number to transfer from:
215597403
Enter Account Number to transfer into:
507307552
Enter value of money to transfer:
5000
```

- Show transactions history

```
Enter Account Number: 874169465
>Sun May 14 15:45:28 2017 Account #874169465 created | Employee Username: ahmed
>Sun May 14 17:01:38 2017 Account #874169465 made a $2000 deposit | Employee Username: ahmed
>Sun May 14 20:55:10 2017 Account #896096250 made a $2000 transfer to Account #874169465 | Employee Username: rana
>Sun May 14 20:55:10 2017 Account #874169465 received a $2000 transfer from Account #896096250 | Employee Username: rana
>Sun May 14 21:07:54 2017 Account #891111281 made a $5000 transfer to Account #874169465 | Employee Username: rana
>Sun May 14 21:07:54 2017 Account #874169465 received a $5000 transfer from Account #891111281 | Employee Username: rana
>Sun May 14 21:08:36 2017 Account #874169465 made a $40000 deposit | Employee Username: rana
>Sun May 14 21:08:52 2017 Account #874169465 made a $30000 withdraw | Employee Username: rana
```

```
>Sun May 14 15:45:28 2017 Account #874169465 created | Employee Username: ahmed
>Sun May 14 17:01:38 2017 Account #874169465 made a $2000 deposit | Employee Username: ahmed
```


- Make deposit

By entering account number and value of money to deposit

```
Enter Account Number:
112315075
Enter value of money to deposit:
1200
```

- Make withdraw

By entering account number and value of money to withdraw

```
Enter Account Number:
201809734
Enter value of money to withdraw:
600
```

- Balance inquiry

By entering account number, we get the balance available in the account

```
Enter Account Number: 201809734

Account:          #201809734
Balance Available:    $12400
```

- Close account

If we search it by entering account number we cannot find it

```
Enter Account Number: 813592725
Are you sure you want to close this account? (y/n)
y
```

```
Enter Account Number: 813592725
Error: Account doesn't exist
```

- Login as a client

Get username and password from “Employees File.txt”

This menu will appear: -

```
*****
                        Bank Management System
*****

Enter your login data:
Username:      waelmuhammad
Password:      21559740

Login Successful!

Choose one of the following:
  1. Inquire Balance
  2. Display Account Info
  3. Show Transactions History
  4. Quit Program
```

- Show transactions history about this account

```
>Sun May 14 19:20:59 2017 Account #206077215 created | Employee Username: ahmed
>Sun May 14 19:30:40 2017 Account #896096250 made a $50000 deposit | Employee Username: rana
>Sun May 14 19:35:18 2017 Account #896096250 made a $2500 withdraw | Employee Username: rana
>Sun May 14 19:38:28 2017 Account #896096250 made a $9000 withdraw | Employee Username: rana
>Sun May 14 20:53:52 2017 Account #896096250 made a $300 withdraw | Employee Username: rana
>Sun May 14 20:55:10 2017 Account #896096250 made a $2000 transfer to Account #874169465 | Employee Username: rana
>Sun May 14 20:55:10 2017 Account #874169465 received a $2000 transfer from Account #896096250 | Employee Username: rana
>Sun May 14 21:07:54 2017 Account #891111281 made a $5000 transfer to Account #874169465 | Employee Username: rana
>Sun May 14 21:07:54 2017 Account #874169465 received a $5000 transfer from Account #891111281 | Employee Username: rana
>Sun May 14 21:08:36 2017 Account #874169465 made a $40000 deposit | Employee Username: rana
>Sun May 14 21:08:52 2017 Account #874169465 made a $30000 withdraw | Employee Username: rana
>Sun May 14 21:22:11 2017 Account #738466290 closed | Employee Username: rana
```