Dhirubhai Ambani Institute of Information and Communication Technology

DA-IICT

IT206

Project Report

End to end encrypted

Banking Management System

Name:

Jaykumar Dhirubhai Navadiya (202001465)

Devansh Manishkumar Patel (202001262)

Course Instructor:

Prof. Archana Nigam

System Requirements:

- A Machine Running On Windows Operating System with Mingw C++ Compiler, Having Code::Blocks on it.
- Employee login credentials
  Id: ADMIN
  Password: Employee@123
- 2 Accounts have been added to the system for demo purposes though any number of Accounts can be added
  - Account No: 1 Password: Deva#1234
  - Account No: 2 Password: Jayn#1234
- All the files are encrypted hence the credentials are provided for demo accounts

Problem Statement:

- Since everyone is working from home in these pandemic times, we designed a system which can be used by both customers and employees. Before the pandemic, employees did not have any system to work with when they were home. So a system was required which can solve these problems.

- An effective system which can distinguish customers and employees while providing the basic facilities of a manual banking system like transferring money, depositing money, withdrawing money, requisition for credit cards, debit cards and loans for customers and viewing the customers’ information (expect password), reviewing customers’ requests, downloading customers’ data (name, account no, mobile no, current balance) in a .txt file for employees.

- Due to pandemic cashless transactions have increased. A system was required which can transfer money between two person very effectively and in a secure manner.

How we approached the problem?

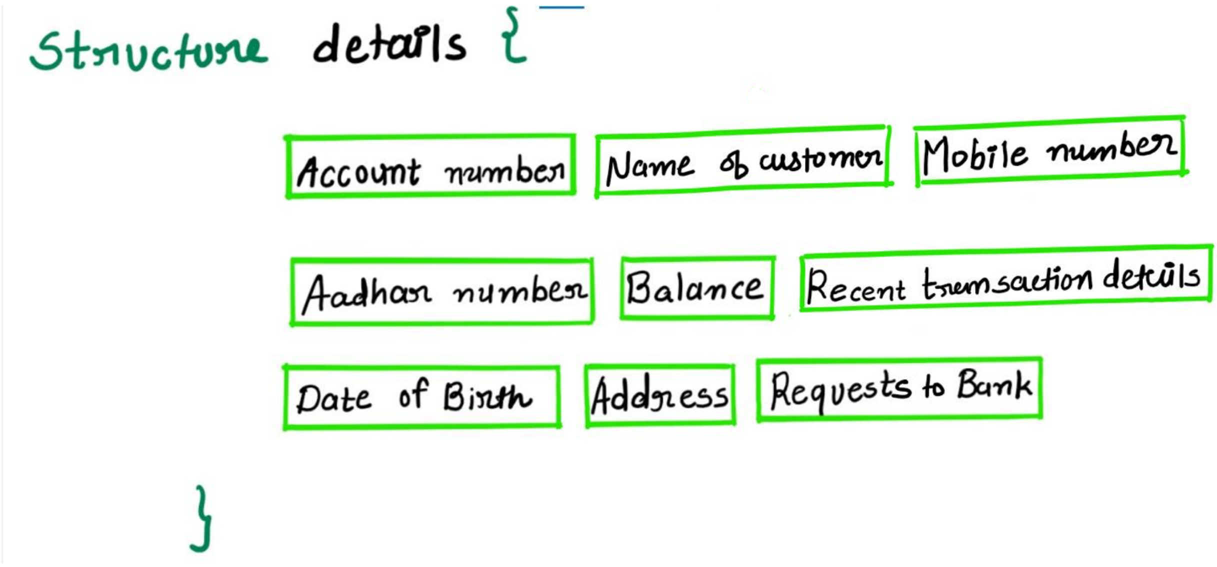
- First of all we decided to use Object Oriented Programming and make class and some of the features as functions of the class.

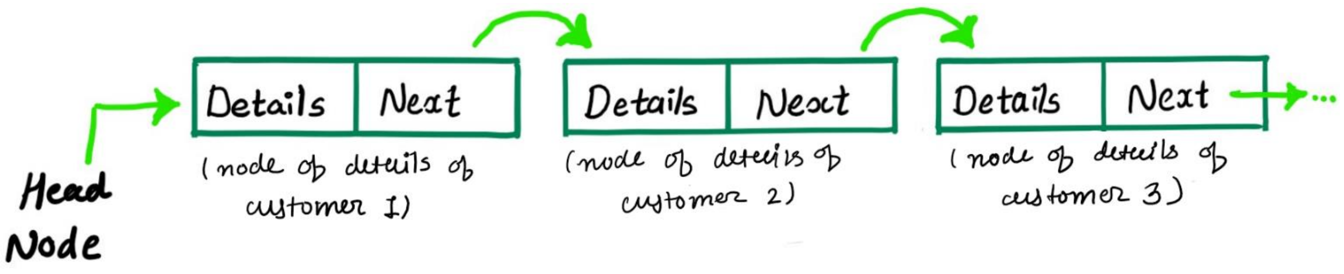
- At first we decided to use a structure which can store data of the customer. The structure contains the following information: Name of the customer, Mobile number of the customer, Account number of the customer, Aadhar number of the customer, Balance information of the customer, Recent transactions' details of the customer, Date of birth of the customer, Address of the customer and other details like if he has requested for loan or cards (credit and debit cards).

- After making the structure our next job was to store data of every customer in a manner that we can access any account any time with its corresponding information. For that Linked List Implementation was decided to be used in the Project.

- One node of the Linked List will store the structure of a particular customer and the pointer to the next node.

- This is visualized below :




Head Node
Details
Next
(node of details of customer 1)
Details
Next
(node of details of customer 2)
Details
Next
(node of details of customer 3)
...

- We made a class named **Bank** on which we implemented the Linked list that we talked about above. Private part of the class Bank contains structure details and a pointer to the next node and also the pointer to the head node.

- The public part of the class contained all the functions which needed access to the customer data.

- For traversing the Linked List, we created a function called **login** which will ask the user for their account number and password and return the node of that particular customer and we will get access to all the details of that customer required in other functions.

- Basically the login function will be called inside the particular function which will require the details of the particular customer.

- One other concern that we had was to take the password from the user in a manner that no one else can see what the customer is entering at the moment. So we thought about displaying an asterisk (*) whenever the customer enters any character of the password. For this purpose we used the conio.h library and the getch() function of conio.h and made our function for taking the password.

- Also with the basic functionality like transfer money, deposit money we added some other features like, recent transactions, card requests and loan requests from customers.

- Also We gave separate functionality to the saving A/C and Current A/C. Account holders having a savings account holder can calculate the interest which will be given by the bank after particular years. Considering Current Balance. If current account holders don't maintain a minimum balance in his/her account, they can check how much penalty they will get after No. of months.

- The problem with our current method of storing the data was whenever anyone (customer/employee) exits the programme, the data will be lost, As we stored the data dynamically in the linked list. So we thought of storing the data in the file at the end of the programme and reading the data at the start of the programme (also inserting the previous data in the linked list at the start while reading the data from file.)

- Our main goal was to secure the data of the user so that no one can misuse the data (even the bank). By using a file to store the data, the details were exposed to everyone (the password of the user was also visible in the file). So we thought of encrypting the file and decrypting while reading the data and then encrypting the data again when the reading process is done. It makes the system end-to-end encrypted. That's how we solved the problem of customers' data security.

Things We learned during the Project :

- We learned about the concepts of object oriented programming.
- We learned about how to work with objects in a class.
- We learned about file implementation for storing the data.
- We also learned about taking the password while showing asterisks (*) in the terminal.
- We learned about a header library conio.h, which provides the functions that can help while running the programme in the console like getch() and system().
- Also we learned about how to keep track of the different versions of the project. For this we used Github, so that we can know every time what we had done the last time and what went wrong and when.
- We learned about coordinating with each other while making the project and how to efficiently work in a group.
- We learned more about Linked list like how to store and use structure objects as data part in the linked list.

Limitations of our Project :

- As our project is not working with the current time and has no knowledge of the time, we can't actually charge a penalty or give interest to the customers.

- We can't keep track of the number of days, so we cannot know if the customer has paid EMI on time or not. (We do have the function of accepting the loan requests, but not the function of keeping track whether the money has been paid back or not).

- Our encryption is not so strong. The probability of getting the key for decoding the data file is 1/256. So, it can be guessed easily and the data of the customers will be exposed.

- If someone gets the access to employee id and employee password, he/she can easily access the data of customers (except password). (Generally banking systems are designed in such a way that customers have different applications than employees so they don't have such kind of problems.)

- In the recent transactions function, if the customer has transferred money to any other account, he/she will be able to know how much money was deducted from his/her account while transferring to another account, but he will not be able to know to whom he/she gave the money. And while receiving the money also he/she will not be able to know from whom he/she received the money, but she will be able to know how much money he/she got.

THANK YOU