

BASIC BANKING APPLICATION

~Batch 11

Team 1

Amol Mandhane
Aditya Joshi
Parul Varma
Swati Kharole

Team 2

Saqib
Akash Bansal
Jeet Juneja
Savi

Team 3

Aditya Patil
Mahesh Avasare
Kanishka Kayathwal.
Saurbh Suman
Anil Babu

INTRODUCTION

The definition of a bank varies from country to country.

A **bank** is a financial intermediary that accepts deposits and channels those deposits into lending activities, either directly or through capital market. A bank connects customers with capital deficits to customers with capital surpluses.

A banker is defined as a person who carries on the business of banking, which is specified as

- conducting current accounts for his customers
- paying cheques drawn on him, and
- collecting cheques for his customers.

Banking is generally a regulated industry, and government restrictions on financial activities by

banks have varied over time and location.

Although using a bank is the most common method of storing and accessing your money, there are some alternatives you should consider. If you feel that your bank isn't giving you what you want, then perhaps it is time for a change. Here are some banking alternatives that might be able to offer you the features and services that you require.

Why use a bank?

Of course, the main reason to use a bank is the fact that banks are widely available, and they are the first option that comes to mind when dealing with finances. In fact, some people aren't even aware that there are alternatives to banking apart from keeping your money at home. Although banking has its uses, it can cost you money for day-to-day financial matters that you can get for less. Bank fees can be extremely expensive, but there are some alternatives.(source : Internet)

Our program facilitates all most all the basic bank operations like creating new account , modifying it , basic transactions, interest calculations , deleting account. Our program also used modification with authentication i.e. it ask the user to input the correct information and if it is verified then it allows to update the file.

TEAM 3 (Handling module 1)

Our part of the program facilitates the user to create new account and update the user's account .Data input taken from the users are name, account number, age,phone, balance,fathers name,address and other functions providing the facilities to add or delete one's account or modify the personal information of the user. For modification we first confirm the users information and then modify it. This ensures the user doesn't enter wrong information.

The program also takes into account that the minimum balance of 1000 bucks. And if not it does not allow the transactions and informs the user to increase their balance, such is our tentative plan as of now.

TEAM 2 (Handling module 2)

Our part of the program is related to handling the records of the bank. We have written the functions which use fstream objects to establish links with the file and then read from or write into the file. So far, we have written functions to perform specific tasks, which would be mentioned in detail in the latter part of this report. What we have done to assign uniqueness to each entry as well as to get its position in the file instantaneously, we have taken a global variable ACCNO, which gets assigned to a new entry and after that gets incremented by 1. This ensures that we know the position of the records in our file as well as get the records in a sorted and uniform fashion, thereby increasing the efficiency of the program.

Team 1(Handling Module 3)

The Module 3 of the project was the part of the job to be done by team 1. The module provides several functions to the main program that complete the tasks. The module 3 deals mainly with the solutions to the banking problem rather than the database structure. Details of the functions are as follows:

Task 1:

Transactions:

Transaction is the soul of banking operations . So, a banking application must provide a strong function for transaction. The problem with doing transaction is first verification and then money transfer. The transaction must not low down the balance of the account lower than the minimum balance limit provided by bank management system. And also , the transactions must have an upper limit. The program comes with solution to these problems . It deals separately with withdrawal and deposit. If the transaction is to deposit , it verifies the upper limit of transaction and skips the minimum balance limit constraint. But if it is the

withdrawal , it checks for both. If transaction is invalid , it gives user an error message . But if it is valid , it modifies the database accordingly. The verification is done by a separate function block.

Task 2:

Delete Account

If it is a bank , it has to create and delete account. The deletion was covered in this module of the project. First the program ask the user his account number. The it reconfirms the user whether he is sure to delete the account or not. If yes, it makes all the fields, name ,age balance etc. null . And thus account is deleted, and the record is of no longer use in the database.

Task 3.

Interest

The bank account has interest associated with it and any banking application should have a code to calculate the interest. The program comes up with a function that calculates the interest on six monthly basis with rate of 5% p.a. The function requires an attribute to every account holder as the last updated date. The program gets the date from the system. If six months are over since the last updated date , it calculates the interest and adds to the balance of the account holder. The comparison of dates is done by a function `verify_dates`.

FUNCTIONAL SPECIFICATIONS

TEAM 2

The functions written by us are basically to perform data updation. The functions we wrote are:

- 1. transaction()*

2. *modification()*

3. *check()* //which gets called by the function modification to simplify the user choice of modification

4. *search()* and *view()*

The algorithms associated with the functions are fairly simple ones.

1. *transaction()* would be taking the choice from the user whether he/she wants to perform deposit or withdrawal. If the balance is zero, then only deposit can be performed. This function validates the balance of the user.

2. *Modification()* is basically for doing modifications to the existing information of the record.

It would be calling check which would be asking user whether he/she wants to modify that particular entry. A character pointer is passed as a parameter to this function.

3. *The next function is search() and view(). It would be used to display the data of a particular record. Search() would find the position of the record in the file. With this as the parameter to view(), we will display that particular record.*

Team 1 :

Function Blocks

Transaction()

Verify_Transaction()

Delete_account()

Interest()

Date()

Compare_Dates()

TEAM 3 :

Here we define basic functions for mainly addition. We are yet to write the data validation parts.

1.ADDITION OF AN ACCOUNT

a) It is designated by “ addition()”

b) This adds information of user i.e. It creates his or her account.

c) Its calls the function getdata() which takes input from the user and stores it in the global object obj of class bank.

d) Variables used are “actno”, “accno”, ”name”, ”phone”, ”balance”, ”age”, "gname", ”fname”, ”address”.

e)

"actno" - *it is 4 digit number unique for each customer....*

(staring from 1001)

"name" - *it is char sting of max 30 characters...*

"phone" - *as we are allowing international customers phone no must be 12 digit no...*

"balance" - *it is float type variable with lower limit of Rs 1000...*

"age" - *it is simple integer type variable with lower limit of 18 years (for normal account) and if age is below 18 then there must be introduction of guardian...*

"gname" - *it stands for guardians name . it is a sting of 30 char max...*

"fname" - *it stand for father's name of customer. Again it is char sting of max 30 characters...*

"address" - *it is a char string of max 50 characters...>*

f) The other inbuilt functions used in the program are

ofstream fout;

getdata()

fout.open

fout.write

fout.close

cin.getline

DATA FUNCTIONAL

Our program require certain basic information of user account , mainly -

- 1)Name
- 2)Age
- 3)Address
- 4)Father name
- 5)Phone

There will be 3 basic options -

- 1)Make new account
- 2)Edit account
- 3)Exit

Along with this there will be also options for the user for basic Transactions and Deletion of account.

APPENDICES

wikipedia.com – for general introduction of banking

Google.com – for sample program

C++.com - understanding class,structure.

www.gamingeeks.org- for abstracting Data.