SITYOG INSTITUTE OF TECHNOLOGY

GROWTH CENTER ,JASOIYA MORE , AURNAGABAD, PIN CODE 824101

Affiliated to aryabhatta university & Approved by AICTE, NEW DELHI

A

PROJECT REPORT ON

"BANKING SERVICES"

Submitted in partial fulfillment for the award of engineering in

BACHELOR OF COMPUTER APPLICATION

SUMMITED BY:-

NAME:-	ROLL NO:-	REG NO:
Deepak kumar Sharma	19BCA22	19303106010
Priti kumari	19BCA06	19303106004
Anuradha rani	19BCA17	19303106001
Anupriya	19BCA02	19303106005

UNDER THE GUIDANCE OF

Mr. Akash Kumar
BCA Department
AURANGABAD

2019-2022

Mr. Ranjeet Pandey
H.O.D BCA DEPARTMENT S.I.T
S.I.T AURANGABAD

SITYOG INSTITUTE OF TECHNOLOGY

GROWTH CENTER, JASOIYA MORE, AURANGABAD

Affiliated to Aryabhatta knowledge university & Approved by AICTE, NEW DELHI

Certificate

This is certify that the project report entitled

"BANKING SERVICES"

Submitted by

NAME	ROLL.NO	REG.NO
DEEPAK KUMAR SHARMA	19BCA22	19303106010
PRITI KUMARI	19BCA06	19303106004
ANURADHA RANI	19BCA17	19303106001
ANU PRIYA	19BCA02	19303106005

Student of BCA final year for the partial fulfillment of Bachelor's in computer application is a reward of bonafide work completed under mysupervi

....

Principal ..... External Examiner

H.O.D (BCA BRANCH)

DECLARATION

This is to certify that report entitled "**BANKING SERVICES**" which is submitted by me in partial of the requirements for the award degree **BCA** to **SITYOG INSTITUTE IF TECHNOLOGY, ARYABHATTA UNIVERSITY ,PATNA (BIHAR)** comprises only my work and due acknowledge has been made in the text all other material used.

ACKNOWLEDGEMENT

i am very thankful to all that have helped me for the project and for providing valuable time and guidance my projrct work.

I owe deep gratitude to a number of people of who have guide, inspired amd motivated me.So I take this opportunity to thank all of them. This endeavor of mine has given me insight into the various aspects of the software implementation in practical application . it has given me a hand on experience , which I could not have gained from all the books of the world.

I am very grateful to my guide **Mr. Akash Kumar** who has endeavored me from the learning and processing the knowledge during the project work.

I am very much thankful to all the STAFF MEMBERS of this college who have proved to be a constant motivation for the knowledge acquision amd morale support during the project.

I would like to thank to **Mr. Ranjeet Pandey ,** head of department of BCA.

I would like to thank our friends and all those persons who directly of indirectly helped me during this project .

And, of course I would not forget our PARENTS who have motivated me very often during this phase of the project by suggesting new ideas towards the project and making us successful.

With Regards

DEEPAK KUMAR SHARMA 19BCA22
PRITI KUMARI 19BCA06
ANURADHA RANI 19BCA17
ANUPRIYA 19BCA02

Banking service

ORIGINALITY REPORT

6%	%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	**Vaishali M. Deshmukh, G.R. Bamnote. "An Empirical Study of XML Parsers across Applications", 2015 International Conference on Computing Communication Control and Automation, 2015** Publication	4%
2	**"Application of Cloud Security Services for the Secured Vendor Registration", International Journal of Engineering Research and Advanced Technology, 2017** Publication	1%
3	**Avelino J. Gonzalez. "Computer Programming in C for Beginners", Springer Science and Business Media LLC, 2020** Publication	
4	**Carmel Domshlak. "", IEEE Transactions on Knowledge and Data Engineering, 4/2007** Publication	<1%
5	**Oda, K.. "Emulsifier content and side effects of oil-based adjuvant vaccine in swine", Research**	<1%

<1%

in Veterinary Science, 200608

Publication

Exclude quotes Exclude matchesOf

Off Exclude bibliography

On

ABSTRACT

The main objective of the project is to develop online Banking system for banks. In present system all banking work is done manually. User have to visit bank to Withdrawal or Deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system, we will automate all the banking process. In our bank management system user can check his balance online and he can also transfer money to other account online. In this Software you can keep record for daily Banking transactions. The main purpose of developing bank management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provides security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user.

Keywords: Pin code, Secure, Simple process.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossiblewithoutthementionofthepeoplewhomadeitpossible,whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon EducationalInstitutionsforprovidingnecessaryinfrastructureand creating goodenvironment.

Itakethisopportunitytoexpressmyprofoundgratitudeto**Dr.Manjunatha,**Principal NHCE, for his constant support and encouragement.

I would also like to thank Dr**. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for herconstant support.

I express my gratitude **Mr. Dinesh G,** Sr. Assistant Professor, department of CSE, NHCE my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing thework.

Raj sagar Kumar (1NH19CS137)

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

INFORMATION CONSISTENCY

The Banking Service Interface is targeted to thefuture banking solution for the Users who is having multiple bank accounts in multiple banks. This interface integrates and provide business solutions for both retail and corporate. This service acts as a standard interface between the clients and all the banks, by using this portal any client who maintain accounts in various banks can directly log on to Multi Banking Service Interface and make any kind of transactions. In the backend, system will take care of the entire obligation requiredinorderto carry on transaction smoothly.

OBJECTIVE

System Involves:

- This interface integrates all existing banks and provides business solutions for both retailers and corporate.
- Thissystem actsasastandard interface betweentheclientsandthe bank.
- Users who have accounts in various banks can login here and make any kind of transactions.
- In the backend, system will take care of the entire obligation required in order tocarry on transaction.

PROJECT ANALYSIS

This application consists following modules

(1) Admin Module
(2) Customer Module
(3) Bank Admin Module
(4) Reports Module

(1) Admin Module:

The admin module will be used by the administrator of this portal, admin can accept or reject the requests from the bankers, and also admin can accept or reject the requests from the users. The requests arein theform of bank registration, customer registration. This module is having following functionalities.

Pending Bankers Requests:

By using this functionality Administrator can give access permeations to all bankers who are registered in this portal

1 Pending User Requests:

By using this functionality Administrator can give access permeations to all users who areregistered in this portal.

(2) Customer Module:

Thismoduledescribeseverythingaboutcustomers,byusingthismoduleany customer can do some operations like create a new account, view the account information, Transfer amount from one account to other account and customer can also see the Transaction Reports. This module consists following functionalities.

Create New Account:

By using this functionality user can create a new account in any bank by selecting bank name option.

View Account Information:

By using thisfunctionality user viewall hisaccount details,thiscan be viewed by users who arehaving account in any bank

Transfer Amount:

By using this functionality user can transfer money from his account to other accounts of same bank or other banks.

Transaction Reports:

By using thisfunctionality user can get allhis transaction reports like accepted transactions, rejected transactions and pending transactions.

(3) Bank Admin Module:

This module deals with alltransactions of bankmanagement. By using this module bank staff can view all details of customers, theycan go for any transactions of their customers and also, they can give access permeations to all customers of that bank This module consists following functionalities.

List of Customers:

By using thisfunctionality Bank admin can gettheirentirecustomers list and their details.

List of Accounts:

By using thisfunctionality Bank admin can gettheirentire customers list based on selected account type likesaving account, current account etc.

Transfer Pending:

By using thisfunctionality Bankadmin can maintain moneytransfer details of customers.

View Account Information:

By using thisfunctionality user viewall hisaccount details,thiscan be viewed by users who arehaving account in any bank

Transfer Amount:

By using this functionality user can transfer money from his account to other accounts of same bank or other banks.

Transaction Reports:

By using thisfunctionality user can get allhis transaction reports like accepted transactions, rejected transactions and pending transactions.

(4) Bank Admin Module:

This module deals with alltransactions of bankmanagement. By using this module bank staff can view all details of customers, theycan go for any transactions of their customers and also, they can give access permeations to all customers of that bank This module consists following functionalities.

List of Customers:

By using thisfunctionality Bank admin can gettheirentirecustomers list and their details.

List of Accounts:

By using thisfunctionality Bank admin can gettheirentire customers list based on selected account type likesaving account, current account etc

Transfer Declines:

By using this functionality Bank admin can maintain money transfer rejected customer details.

New Accounts Pending:

By using this functionality Bank admin can maintain entire user details who are requesting for new account in that bank

Reports Module:

In this module administrator will get different types of reports read customers like number of customers of this portal and no. of banks registered in this portal. This module is controlled by administrator only.

Software requirements:

- Operating System: Window 7 or above.
- IDE: Visual studio 2015 or above.
- Web Server: Tomcat Database: Oracle

Hardware requirements:

- Processor Any Processor above 500 MHz
- RAM: 512 Mb or above.
- Hard Disk 4GB
- Input Device: Standard keyboard and mouse.

Additional Tools:

- HTML Designing: Dream weaver tools
- Development tool kit my Eclipse.

EXISTING SYSTEM & DISADVANTAGES

Currently we are having lot of banks in the market and any person can do transactors of any individual bank either manually or in online. But no one can do all banks transactions in a single portal or in single bank

PROPOSED SYSTEM & ITS ADVANTAGES

The Banking System Interface is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing bank good provides business solutions for both retail and corporate. This system acts as a standard interface between the clients and all the banks, by using this portal any client who maintain accounts in various banks can directly log onto Multi Banking System Interface and make any kind of transactions. In the backend, system will take care of the entire obligation required in order to carry on transaction smoothly.

Chapter2

DATA STRUCTURE

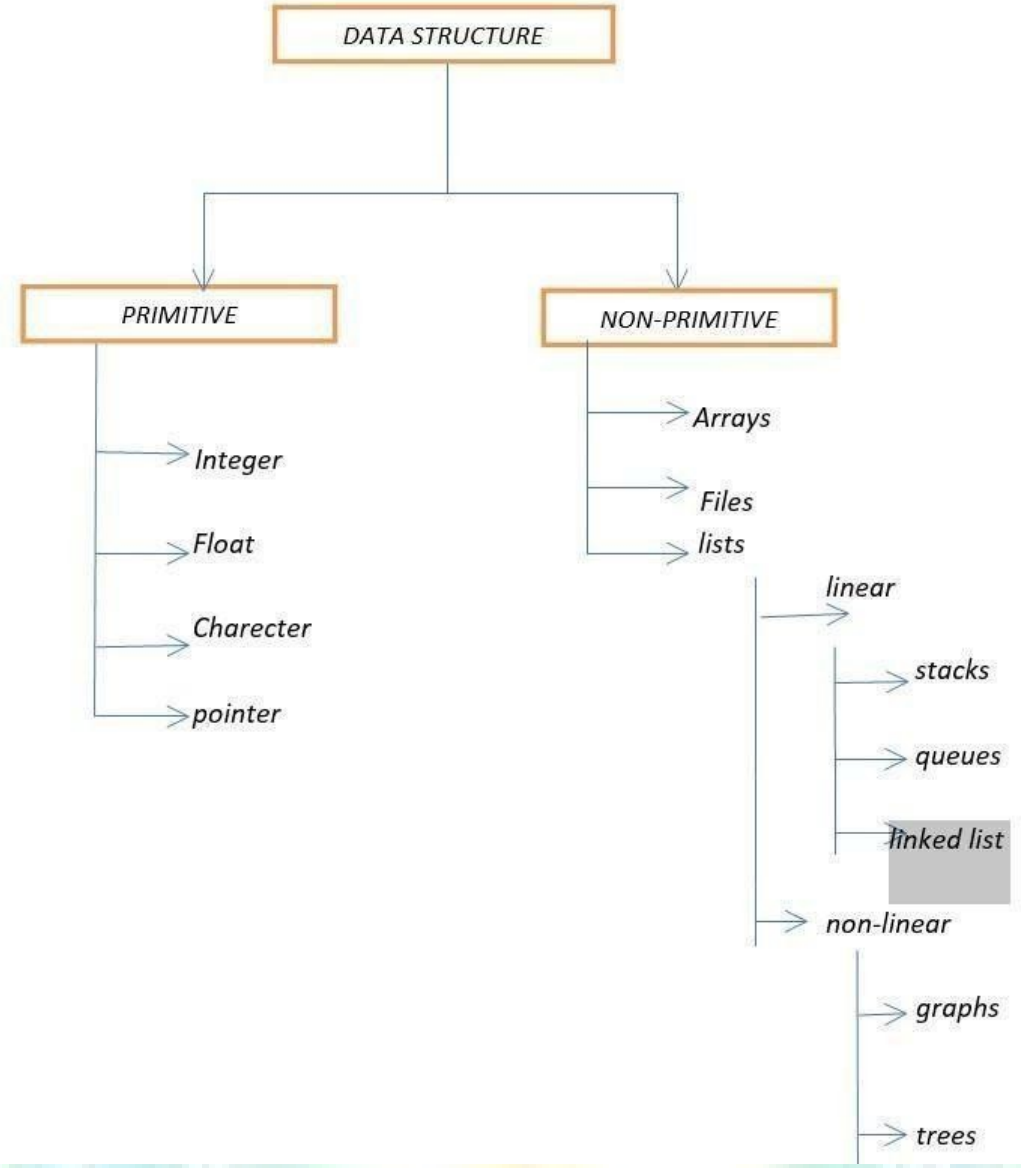

Fig no :- 1 Flowchart of Data Structure

DATA:

Data is a collective of value. e.g., the data of the student includes his USN ,Name, Semester, Section, Address and phone number. Where address and name may have sub data like his/her first name, Middle name, Last name.

Data structure**:**

It is used for storing data in a format.

Data structures are of two types.

- Primitive data structures
- Non-primitive data structures.

1. Primitive data structures can be directly manipulated by machine instructions. Examples :

I. Integer
II. Float
III. Char
IV. Pointer

i. **Integer:** It is a whole number without decimal points; it consists of two bytes.

ii. **Float:** It is a decimal number where it consists of 4 bytes.

iii. **Char:** It consists of all the string variables and it consists of two bytes. *iv.* pointer: It gives the address of the other variable.

2.Non-primitive data structures can not be directly manipulated using machine instructions.

Examples :

1. Arrays
2. List
3. File

1. **Array:** It is a sort of information structure where it is an assortment of comparative information type variables. An exhibit is an information type in C, which is built from basic information of the C language.

2. **Syntax:**

Data type Array-Name[index]; E.g.: -int A[10];

Char S[20]; float F[15];

Types of array

- Single dimensional array
- Two dimensional array
- Multi dimensional array

- **Single dimensional array:** It is a collection of data items which can be stored under a variable name only by using one subscript

  Syntax :

  Data _type array _name[array_size_1] Ex: int a[5]

- **Two dimensional array :** Arrays with two sets of brackets [][] are called two dimensional arrays. It can be used when the data items are arranged in row wise or in column wise in a tabular column.

  Syntax :

  Data _type array _name[array_size_1][array_size_2]

  Ex : int a[5][5]

- **Multi dimensional array :** It is also called a three dimensional array where it is used for representing the total number of tables of matrix. It is used when we want to make the two or more tables of matrix elements for declaring the array elements we can use like this way. Syntax :

Data _type array _name[array_size_1][array_size_2] [array_size_3] Ex :

int a[5][5][5]

2.List :

It is a sequence of links which contains items. Each link is connected to one another link. Lists are again classified into.

- Linear lists
- Non-linear lists.

1. Linear list consists of:

- Stack
- queues
- linked list.

2. Non-linear list consists of :

- trees
- graph.

We have 2 types of Memory Managements/Allocations

- Static Memory Allocation
- Dynamic Memory Allocation

- **Static Memory Management:**If memory is allocated to the variables during compile time then it is called static memory allocation.

  e.g. int a [10]; int *p;

Disadvantages:

I. Memory space is fixed during compilation time. i.e. it can't be altered.
II. If all the memory allocated is not used then it leads to memory wastage.
III. Memory can't be increased later, if we require it to be.

- **Dynamic Memory Management:**It is the process of allocating memory during execution/run time. It uses predefined functions to allocate and release memory. Memory is allocated to the nodes using dynamic memory allocation functions such as malloc() , calloc () , realloc() and free().

I. **malloc() :**This function is used to allocate a complete single block of memory of the specified size. A pointer is used to store the address returned by malloc.

Syntax – data type*ptr=(datatype*)malloc(size)

ii. **calloc() :** It is a function which allocates a specified size of memory in multiple blocks of the same size. Each block should be assigned to null. A pointer is used to store the address.
Syntax – datatype *ptr =(datatype*)calloc(size, number of blocks)

iii. **realloc() :**For reallocating the allocated memory this function is used. A pointer is used to store the address returned
Syntax – data type*ptr=(datatype*)realloc(ptr,size)

iv. **free() :**It is a function which is used to free the allocated memory. Syntax – free(pointer name)

STACK

A stack is a linear data structure where the elements are inserted and deleted from the same which follows last in first out (LIFO) .

Ex. Stack of books.

The basic primitive operation of stack is

1. **push:**Push inserting an element on the top of stack
2. **pop:**Pop deleting the top most element of the stack

Advantages

- Easy to start
- Less hardware
- requirement It is a cross
- platform

Disadvantages

- Its not flexible
- it is lack of scalability

- its unable for copy and
- paste

QUEUES

A Queue is an ordered collection of data such that data is inserted at one end and deleted from another end.

It follows first in first out(FIFO) order.

Ex: queue in cafeteria.

Queues can be represented using an array and linked list similar to stack

Queue operation

- Enqueue
- Dequeue
- **Enqueue** : The addition of a new element to the queue is called insertion and every time before inserting a new item rear is incremented first then is inserted.

  The condition to be checked in insertion is queue overflow, where reaches maximum size
- **Dequeue** : the removal of the front element from the queue is called deletion after deleting the element .We need to increment the front pointer.

  The condition to be checked in deletion is queue underflow.

LINKED LIST

A linked list is a linear data structure , in which the elements are not sorted at contiguous memory locations , a linked list contains nodes where each node contains a data field and a reference (link) to the next node in the list. The first node is called head. If the linked list is empty, then the value of head is NULL.

Some types of linked list are

- Single linked list
- double linked list
- circular linked list
- header linked list

- **Single linked list:**-Each node contains only one pointer which will point to the next node. accessing from left to right is possible.
- **Double linked list :-**Each node contains two pointers one is the previous node and the other is the next node. Accessing from left to right and right to left is possible.
- **Circular linked list:** In any linked list, if the last node is to be pointed back to the first node then it is called circular linked list.
- **Header linked list:-** header linked list is one or more types of linked list. In these types, we have a special node created in the linked list. This special node is used to store some useful information about the linked list.

ADVANTAGES OF LINKED LIST OVER ARRAYS IS :

1. Size of the array is fixed ,we must know its upper limit in advance. But the linked list size is not fixed.
2. Insertion and deletion is easy compared to array.
3. No memory wastage will be there in linked list

DISADVANTAGES OF LINKED LISTS:

There were so many problems with linked lists .The major disadvantage of linked lists. lists are access time to individual elements .Array is randomly accessed memory ,which means it takes to access any element in the array .Linked lists take for access to an element. In the list in the worst case. Another disadvantage of arrays in access time is special locality in memory .Arrays are defined as contiguous blocks of memory and lots of array elements will be temporarily near its neighbouring nodes .This greatly benefits from the latest CPU holding methods.

TREES

It is non linear data structures ,compared to array ,linked list, stack which are linear data structures .A tree can be empty with no nodes or a tree is a mode consisting of one tree.

Types of tree:

- Binary tree:

  It is a tree which has 2 child nodes.
- Full binary tree:

  It's a tree in which every node should have two child nodes and 2 child nodes except the leaf node.
- Strictly binary tree:

  It is a tree which has 0 or 2 child nodes is called strictly a binary tree.
- Complete binary tree:

  It is a full binary tree till n-1 level. But at the last level all the nodes are filled from left to right.
- Binary search tree:

  It is a binary tree in which all the elements off the left subtree are less than the root node and all the elements in the right subtree are greater than the root node.
- Expression binary tree:

  It is a binary tree on which the internal nodes represents operation and external nodes represents operands

GRAPHS

It is a non linear data structure. It is a pictorial representation of a set of objects where some pairs are connected into links.

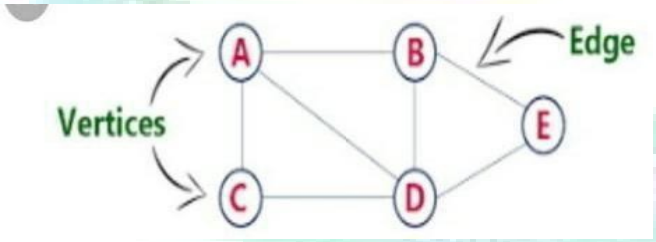

Fig no :- 2 Graph representation

Different types of representing the graph:

1. Adjacency matrix.
2. Adjacency linked list.

1. Adjacency matrix : It is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph.
2. Adjacency linked list : In an adjacency linked list is a collection of unordered lists used to represent a finite graph. Each linked list describes the set of neighbours of a vertex in the graph.

Types of traversal methods:

- Breadth first search
- Depth first search

- Breadth first search : It is searching a tree or graph data structures. It starts at the tree root, and explores all the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level.
- Depth first search **:** It is searching a tree or graph data structures. The algorithm starts at the root node and explores as far as possible along each branch before brackets.

Chapter 3

DESIGN

Data flow diagram will act as a graphical representation of the system in terms of interaction between the system, external entities, and process and how data stored incertain location

- External entities
- Data stores
- Process
- Data Flow

FLOWCHART

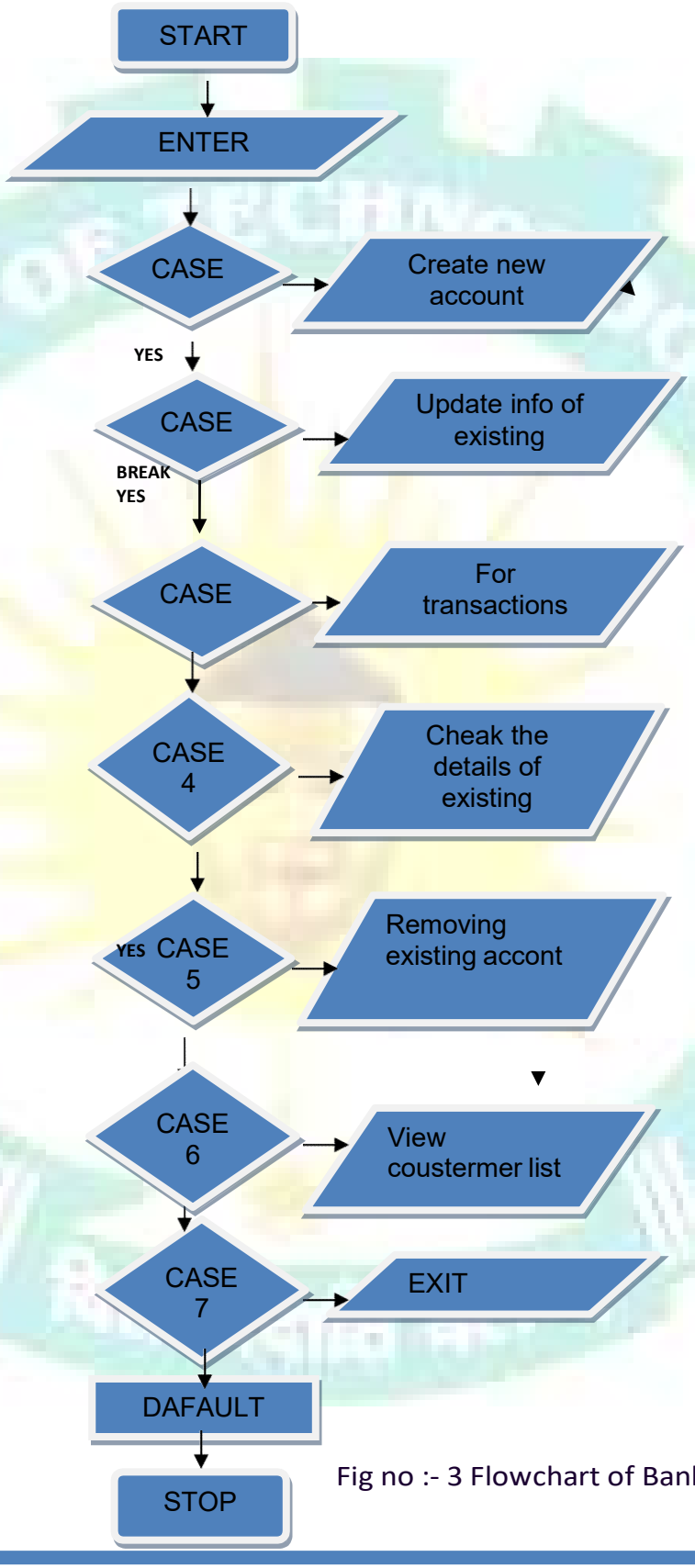

Fig no :- 3 Flowchart of Banking Service

Chapter 4

IMPLEMENTATION

```
#include<stdio.h>
 #include<stdlib.h>
 #include<windows.h>

int main exit;
void menu ();
struct date

int month, day, year;
struct

char name [fi0]; int acc_no, age; char address [60];
char citizenship [15];

double  phone; char acc_type [10]; float amt;
struct date dob; struct date deposit; struct date withdraw;
add, upd, check, rem, transaction;

float interest (float t, float amount, int rate)
float SI; SI=(rate*t*amount)/100.0; return (SI);

void for delay (int j)

{ int i, k;

for(i=0;i<j; i++)
```

```
void new_acc ()

int choice; FILE * ptr;

ptr = f open (”record. dat”,”a+”); account_no:
system(”cls”);

print f("\t\t\t\xB2\xB2\xB2\ ADD RECORD
\xB2\xB2\xB2 \xB2”);
print f (”\n\n\n Enter today’s date(mm/dd/yyyy):”);
scanf(”%d/%d/%d”,&add.deposiLmonth,&add.deposit.da y,&add.deposit.year);
print f(”\n Enter the account number:”); scan f(”%d”,&check.acc_no);
while(fscan f(ptr,”%d %s %d/%d/%d %d %s %s %lf %s
%f

%d/%d/%d\n”,&add.acc_no,add.name,&add.dob.month,&
add.dob.day,&add.dob.year,&add.age,add.address,add.citiz
enship,&add.phone,add.acc_type,&add.amt,&add.deposit. month,&add.deposit.day,&add.deposit.year)!=EOF)

if (check.acc_no==add.acc_no)
{print f(”Account no. already in use!”); For delay(1000000000);
Go to account_no;

add.acc_no=check.acc_no;
print f(”\n Enter the name:”); scan f(”%s”,add.name);
print f(”\n Enter the date of birth(mm/dd/yyyy):”);
scan f(”%d/%d/%d”,&add.dob.month,&add.dob.day,&add.dob. year);
print f("\n Enter the age:”); scan f(”%d”,&add.age);

print f(”\n Enter the address:”); scan f(”%s”,add.address);
```

```
print f(”\n Enter the citizenship number:”); scan f(”%s”,add.citizenship);
print f(”\n Enter the phone number: ”); scan f(”%lfi',&add.phone);
print f(”\n Enter the amount to deposit:$”); scan f(”%fi‘,&add.amt);
print f(”\n Type of account:\n\t#Saving\n\téCurrent\n\t#Fixed1(for 1
year)\n\téFixed2(for 2 years)\n\t#Fixed3(for 3 years)\n\n\t Enter your choice:”);
scan f(”%s”,add.acc_type);

fprint f(ptr,”%d %s 0Zod/ % d/ % d 0Zod %s %s 0Zolf %s 0Zof
%d/%d/%d\n”,add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposiLmonth,add.d eposit.day,add.deposit.year);

fclose(ptr);

print f(”\n Account created successfully!”);

add_invalid:
print f(”\n\n\n\t\t Enter 1 to go to the main menu and 0 to exit:”);
scan f(”%d”,&main_exit); system(”cls”);
if (main_exit==1)
menu();

else if(main_exit==0)
close();
else
Print f(”\n Invalid!\a”);
goto add_invalid;

void view_list 0
```

```
FILE *view;

view=fopen (”record.dat”,”r”);

int test=0;
system(”cls”);

print f(”\n ACC. NO.\tNAME\t\t\tADDRESS\t\t\tPHONE\n”);

while(fscan f(view,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposit.m onth,&add.deposit.day,&add.deposit.year)!=EOF)

Print f(”\n%6d\t
%10s\t\t\t%l0s\t\t%.0lf',add.acc_no,add.name,add.addr ess,add.phone);
test++;

fclose(view); if (test==0)

{ system(”cls”);

Print f(”\n NO RECORDS!!\n”);}

view_list_invalid:

print f(”\n\n Enter 1 to go to the main menu and 0 to exit:”);
scan f(”%d”,&main_exit); system(”cls”);
if (main_exit== l) menu ();
```

```
else if(main_exit==0)

close 0i
else

Print f(”\n Invalid!\a”); goto view_list_invalid;

void edit(void)

int choice,test=0; FILE *old,*newrec;
old=fopen(”record.dat”,”r”);
newrec=fopen(”new.dat”,”w”);

print f(”\n Enter the account no. of the customer whose info you want to change:”);
scan f(”%d”,&upd.acc_no);
while(fscan f(old,”%d %s %d/%d/%d %d %s %s %lf %s
%f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposit.m onth,&add.deposit.day,&add.deposiLyear)!=EOF)

if (add.acc_no==upd.acc_no)
{ test=1;
Print f(”\n Which information do you want to change?\nl.Address\n2.Phone\n\n Enter your choice(l for address
and 2 for phone):”);
Scan f(”%d",&choice);

system(”cis”); if(choice==l)
{print f(”Enter the new address:”); Scan f(”%s”,upd.address);
Fprint f(newrec,”%d %s %d/%d/%d %d %s %s
```

```
%lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,upd.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposiLmonth,add.d eposiLday,add.deposit.year);
system("cls");
print f("Changes saved!");

else if(choice==2)

Print f("Enter the new phone number:"); Scan f("%lfi',&upd.phone);
Fprint f(newrec,"%d %s %d/%d/%d %d %s %s
%lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,

upd.phone,add.acc_type,add.amt,add.deposit.month,add.d eposiLday,add.deposit.year);
system("cls");

print f("Changes saved!");

else

print f(newrec,"%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposit.month,add.d eposiLday,add.deposit.year)

fclose(old); fclose(newrec); remove("record.dat");
rename("new.dat","record.dat");
```

```
if(test!=1)

{ system(”cls”);

Print f(”\n Record not found!!\a\a\a”); edit_invalid:
print f(”\n Enter 0 to try again,1 to return to main menu and 2 to exit:”);
scan f(”%d”,&main_exit); system(”cls”);
if (main_exit==1)
menu ();

else if (main_exit==2) close ();
else if(main_exit==0)

edit ();

else

{print f("\n Invalid!\a”);

goto edit_invalid;}

else

{print f(”\n\n\n Enter 1 to go to the main menu and 0 to exit:”);
Scan f(”%d”,&main_exit);

system(”cls”);

if (main_exit== I) menu ();
else
```

```
close 0:
void transact(void)

{ int choice,test=0; FILE *old,*newrec;
old=fopen(”record.dat”,”r”);

newrec=fopen(”new.dat”,"w”);

print f(”Enter the account no. of the customer:”);

scan f(”%d”,&transaction.acc_no);

while (fscan f(old,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposit.m onth,&add.deposit.day,&add.deposit.year)!=EOF)

if(add.acc_no==transaction.acc_no)

{ test=1;

if(strcmpi(add.acc_type,”fixedl”)==0l lstrcmpi(add .acc_typ e,”fixed2”) == 0l l strcmpi(add.acc_type,”fixed3”) == 0)

Print f(”\a\a\a\n\n YOU CANNOT DEPOSIT OR WITHDRAW CASH IN FIXED ACCOUNTS! !!!!”);
fordelay(1000000000); system(”cls”);
menu ();

Print f(”\n\n Do you want to\n1.Deposit\n2.Withdraw?\n\n Enter your choice(1 for deposit and 2 for
withdraw):”);
```

```
Scan f(”%d”,&choice); if (choice==1)

Print f(”Enter the amount you want to deposit:$

Scan f(”%fi‘,&transaction.amt); add.amt+=transaction.amt;
fprint f(newrec,”%d %s %d/%d/%d %d %s %s
%lf 0Zos %f
%d/%d/%d\n”,add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposit.month,add.d eposiLday,add.deposit.year);
print f(”\n\n Deposited successfully!”);

else

Print f(”Enter the amount you want to withdraw:$ ”);

Scan f(”%fi‘,&transaction.amt); add.amt-=transaction.amt;
fprint f(newrec,”%d %s %d/%d/%d %d %s %s
%lf 0Zos %f
%d/%d/%d\n”,add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposiLmonth,add.d eposiLday,add.deposit.year);
print f (”\n\n Withdrawn successfully!”);

else

Fprint f(newrec,”%d %s %d/%d/%d %d %s %s
%lf 0Zos %f
%d/%d/%d\n”,add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
```

```
add.phone,add.acc_type,add.amt,add.deposiLmonth,add.d eposit.day,add.deposit.year);

Fclose (old); fclose(newrec); remove (" record.dat");
rename(" new.dat","record.dat");

if(test!=1)

Print f("\n\n Record not found!!"); transact_invalid:
print f("\n\n\n Enter 0 to try again,1 to return to main menu and 2 to exit:");
scan f("%d",&main_exit);
system("cls");

if (main_exit==0) transact ();
else if (main_exit== 1) menu ();
else if (main_exit==2)

close (); else

Print f("\n Invalid!"); goto transact_invalid;

else

Print f("\n Enter 1 to go to the main menu and 0 to exit:");
Scan f("%d",&main_exit); system("cls");
if (main_exit== 1) menu ();
else

close 0:

void erase(void)
```

```
FILE *old,*newrec; int test=0;
old=fopen(”record.dat”,”r”);
newrec=fopen(”new.dat”,”w”);
print f(”Enter the account no. of the customer you want to delete:”);
scan f(”%d”,&rem.acc_no);
while (fscan f(old,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposit.m onth,&add.deposit.day,&add.deposiLyear)!=EOF)

if(add.acc_no!=rem.acc_no)
fprint f(newrec,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d\n”,add.acc_no,add.name,add.dob.month,add.
dob.day,add.dob.year,add.age,add.address,add.citizenship,
add.phone,add.acc_type,add.amt,add.deposit.month,add.d eposiLday,add.deposit.year);

else

{test+ +;
Print f(”\n Record deleted successfully!\n”);

Fclose (old); fclose(newrec); remove(”record.dat”);
rename(”new.dat”,”record.dat”);

if(test== 0)
```

```
Print f(”\n Record not found!!\a\a\a”); erase_invalid:
print f(”\n Enter O to try again,l to return to main
menu and 2 to exit:”);
scan f(”%d",&main_exit);

if (main_exit==1) menu ();
else if (main_exit==2)

close ();

else if(main_exit==0) erase ();
else

{print f(”\n Invalid!\a”); goto erase_invalid;}

else

{print f(”\n Enter 1 to go to the main menu and 0 to exit:”);
Scan f(”%d”,&main_exit);
system(”cls”);

if (main_exit== l) menu ();
else

close 0:

void see(void)

FILE *ptr;
```

```
int test=0,rate; int choice; float time; float intrst;
ptr=fopen(”record.dat”,”r”);

print f(”Do you want to check by\n1.Account no\n2.Name\n Enter your choice:”);
scan f(”%d”,&choice); if (choice==1)
{ print f(”Enter the account number:”);

Scan f(”%d”,&check.acc_no);

while (fscan f(ptr,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposiLm onth,&add.deposit.day,&add.deposiLyear)!=EOF)

if(add.acc_no==check.acc_no)

{ system(”cls”); test=1;

print f(”\n Account NO.:%d\nName:%s \n DOB:%d/%d/%d \n Age:%d \n Address:%s \n Citizenship No:%s \nPhone
number:%.0lf \n Type Of Account:%s \n Amount deposited:$ %.2f \n Date Of
Deposit:%d/%d/%d\n\n”,add.acc_no,add.name,add.dob.
month,add.dob.day,add.dob.year,add.age,add.address,add. citizenship,add.phone,

add.acc_type,add.amt,add.deposiLmonth,add.deposit.day,a dd.deposit.year);
if(strcmpi(add.acc_type,”fixed1”)==0)

time=l.0; rate=9;
intrst=interest(time,add.amt,rate);
```

```
print f(”\n\n You will get $%.2f as interest on
%d/%d/%d”,intrst,add.deposiLmonth,add.deposit.day,ad d.deposit.year+1);

else if(strcmpi(add.acc_type,”fixed2”)==0)

time=2.0;

rate=ll; intrst=interest(time,add.amt,rate);
print f(”\n\n You will get $.%.2f as interest on
%d/%d/%d”,intrst,add.deposiLmonth,add.deposit.day,ad d.deposit.year+2);

else if(strcmpi(add.acc_type,”fixed3”)==0)

time=3.0;

rate=ll; intrst=interest(time,add.amt,rate);

print f(”\n\n You will get $.%.2f as interest on
%d/%d/%d”,intrst,add.deposiLmonth,add.deposit.day,ad d.deposit.year+3);

else if(strcmpi(add.acc_type,”saving”)==0)

time=(1.0/12.0); rate=8;
intrst=interest(time,add.amt,rate);

print f(”\n\n You will get $.%.2f as interest on
%d of every month”,intrst,add.deposiLday);

else if(strcmpi(add.acc_type,”current”)==0)
```

```
Print f("\n\n You will get no interest\a\a”);

else if (choice==2)
{ print f(”Enter the name:”); Scan f(”%s”,&check.name);
while (fscan f(ptr,”%d %s %d/%d/%d %d %s %s %lf
%s %f
%d/%d/%d”,&add.acc_no,add.name,&add.dob.month,&ad
d.dob.day,&add.dob.year,&add.age,add.address,add.citize
nship,&add.phone,add.acc_type,&add.amt,&add.deposit.m onth,&add.deposit.day,&add.deposiLyear)!=EOF)

if(strcmpi(add.name,check.name)==0)
{system(”cls”); test=1;
print f(”\n Account No.:%d\n Name:%s \n DOB:%d/%d/%d \n Age:%d \n Address:%s \nCitizenship No:%s \n
Phone number:%.0lf \n Type Of Account:%s \n Amount deposited:$%.2f \n Date Of
Deposit:%d/%d/%d\n\n”,add.acc_no,add.name,add.dob.

month,add.dob.day,add.dob.year,add.age,add.address,add. citizenship,add.phone,

add.acc_type,add.amt,add.deposiLmonth,add.deposit.day,a dd.deposit.year);
if(strcmpi(add.acc_type,”fixedl”)==0)

time=1.0; rate=9;
intrst=interest(time,add.amt,rate);

print f(”\n\n You will get $.%.2f as interest on
%d/%d/%d”,intrst,add.deposiLmonth,add.deposiLday,ad d.deposit.year+l);

else if(strcmpi(add.acc_type,”fixed2”)==0)

time=2.0;
```

```
rate=ll;
intrst=interest(time,add.amt,rate);

%d/%d/%d",intrst,add.deposiLmonth,add.deposit.day,ad d.deposit.year+2);

else if(strcmpi(add.acc_type,"fixed3")==0)

time=3.0;

rate=13;

intrst=interest(time,add.amt,rate);

print f("\n\n You will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposiLday,ad d.deposit.year+3);
else if(strcmpi(add.acc_type,"saving")==0)

time=(1.0/12.0);                rate=8;
intrst=interest(time,add.amt,rate);

%d of every month",intrst,add.deposiLday);

else if(strcmpi(add.acc_type,"current")==0)

Print f("\n\n You will get no interest\a\a");

fclose(ptr); if(test!=1)
{ system("cls");
Print f("\n Record not found!!\a\a\a");
```

```
see_invalid:

print f(”\n Enter O to try again,l to return to main menu and 2 to exit:”);
scan f(”%d”,&main_exit); system(”cls”);
if (main_exit==1) menu ();
else if (main_exit==2)
close ();

else if(main_exit==0)

see ();
else

system(”cls”);

print f(”\n Invalid!\a”);

goto see_invalid;}

else

{print f(”\n Enter 1 to go to the main menu and 0 to exit:”);

BANKING SERVICE
Scan f(”%d”,&main_exit);}
if (main_exit== l)

system(”cis”);
menu ();
```

```
else

system(”cls”);
close 0:
close(void)

Dept of CSE, N H CE

Print f(”\n\n\n\n This C Mini Project is developed by Raj Sagar ”);

void menu(void)
{ int choice; system(”cls"); system(”color 9”);
print f(”\n\n\t\t\t CUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM");
print f(”\n\n\n\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO THE MAIN MENU
\xB2\xB2\xB2 \xB2\xB2\xB2\xB2”);
print f(”\n\n\t\t1.Create newaccount\n\t\t2.Update information of existing account\n\t\t3.For
transactions\n\t\t4.Check the details of existing account\n\t\t5.Removing existing account\n\t\tfLView
customer’s list\n\t\t7.Exit\n\n\n\n\n\t\t Enter your choice:”);
scan f(”%d”,&choice);

system(”cls”);

switch(choice)

case 1:new_acc0:
break;
case 2:edit0:
break;

case 3:transact(); break;
```

```
case 4:see(); break;
case 5:erase(); break;
case 6:view_list();

break;

case 7:close(); break;
 int main()

char pass[10],password[10]=”admin”; int i=0,
print f(”\n\n\t\t Enter the password to login:”),

scan f(”%s”,pass);

/‘°•

//if (pass[i]!=13&&pass[i]!=8)

Print f(”*”); pass[i]=getch 0.

}while (pass[i]!=13);
pass[10]=’\0’,*/
if (strcmp(pass,password)==0)

{print f(”\n\n Password Match!\n LOADING”); for(i=0;i<=6;i++)

fordelay(100000000);
print f(”.”);
```

```
system(”cls”); menu ();

else

{ print f(”\n\n Wrong password!!\a\a\a”); login_try:
print f(”\n Enter 1 to try again and 0 to exit:”); scan f(”%d”,&main_exit);
if (main_exit==1)

system(”cls”);

main ();

else if (main_exit==0)

system(”cls”);

close();
else
{printf(“i invalid!”); fordelay(`1`1000000000); system(”cls”);
goto login_try;}

return 0;
}
```

Chapter 5

RESULT

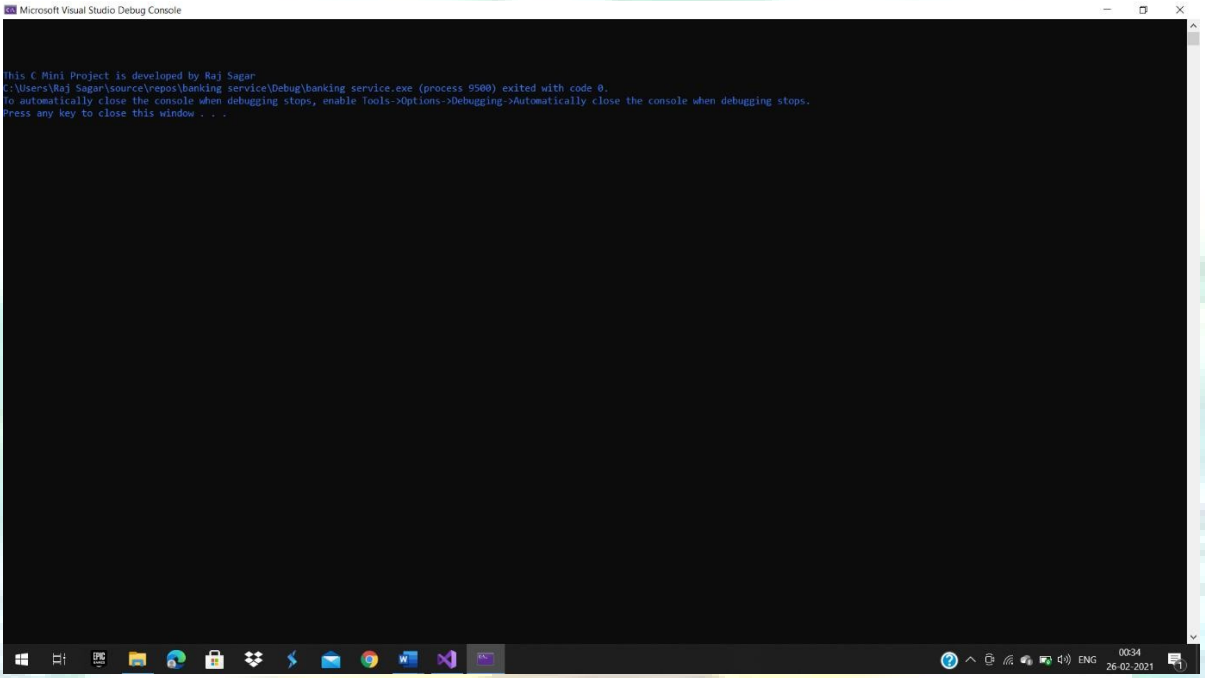

Fig no: -4 Introduction

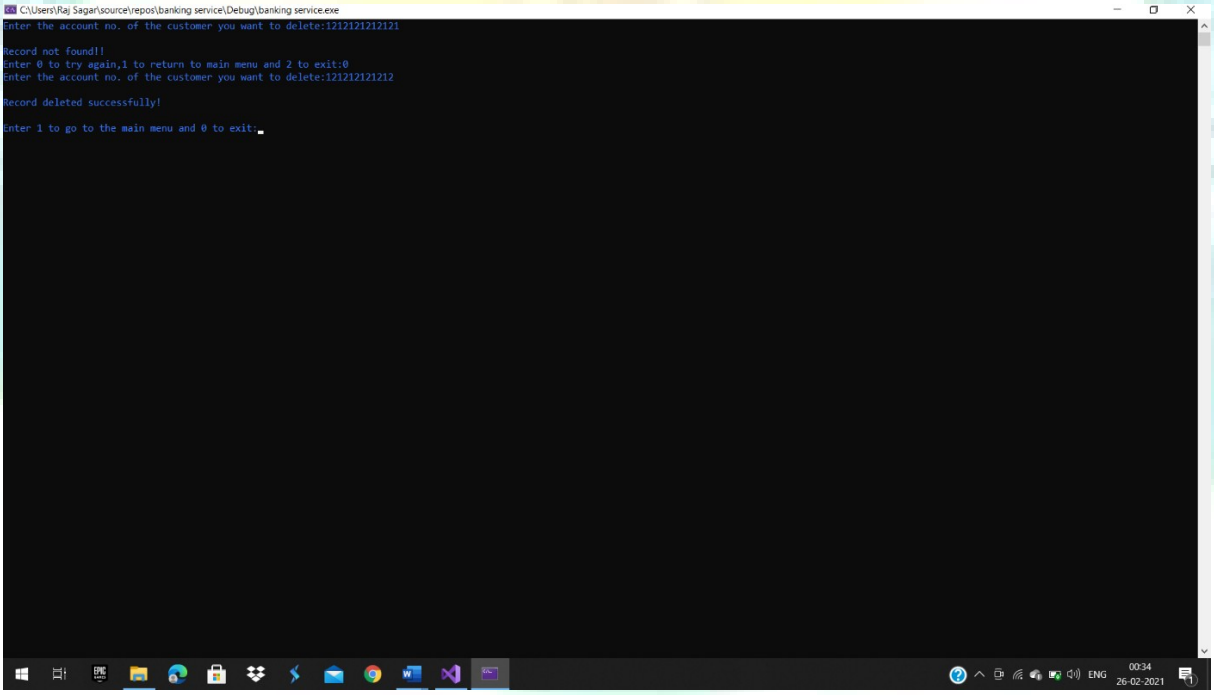

Fig no: - 5 Account Deletion

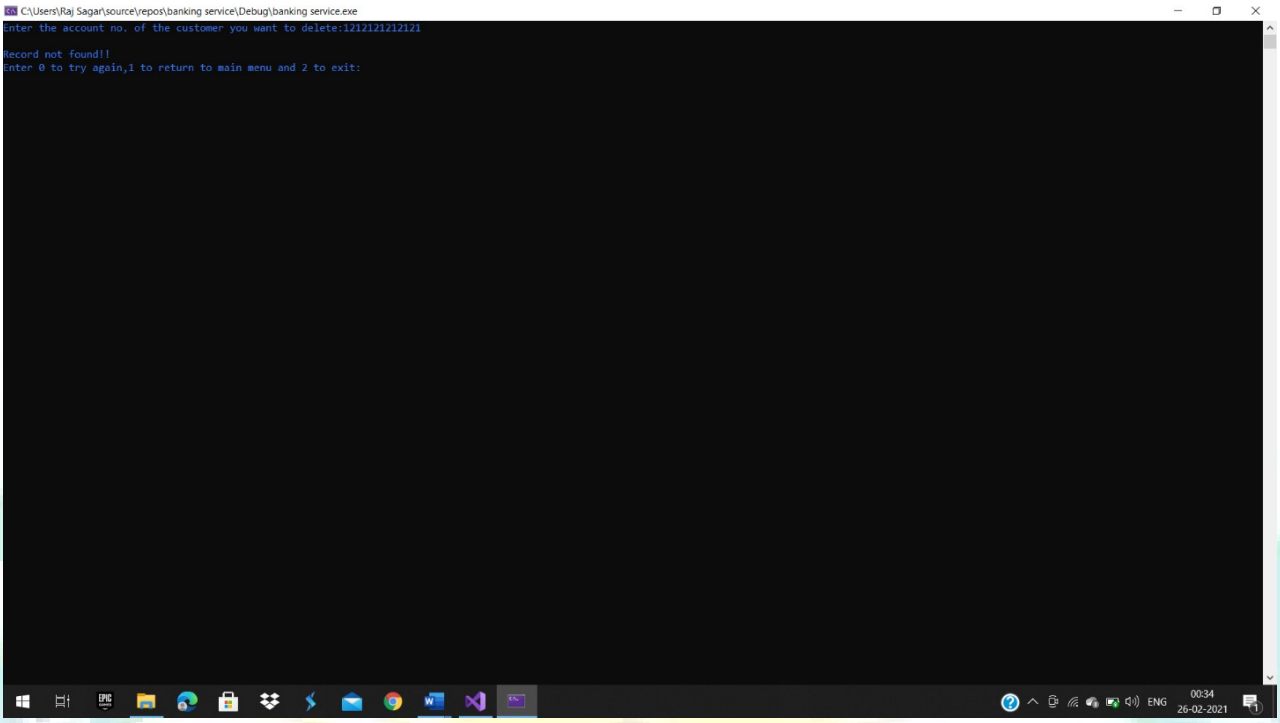

Fig no: - 6 Error

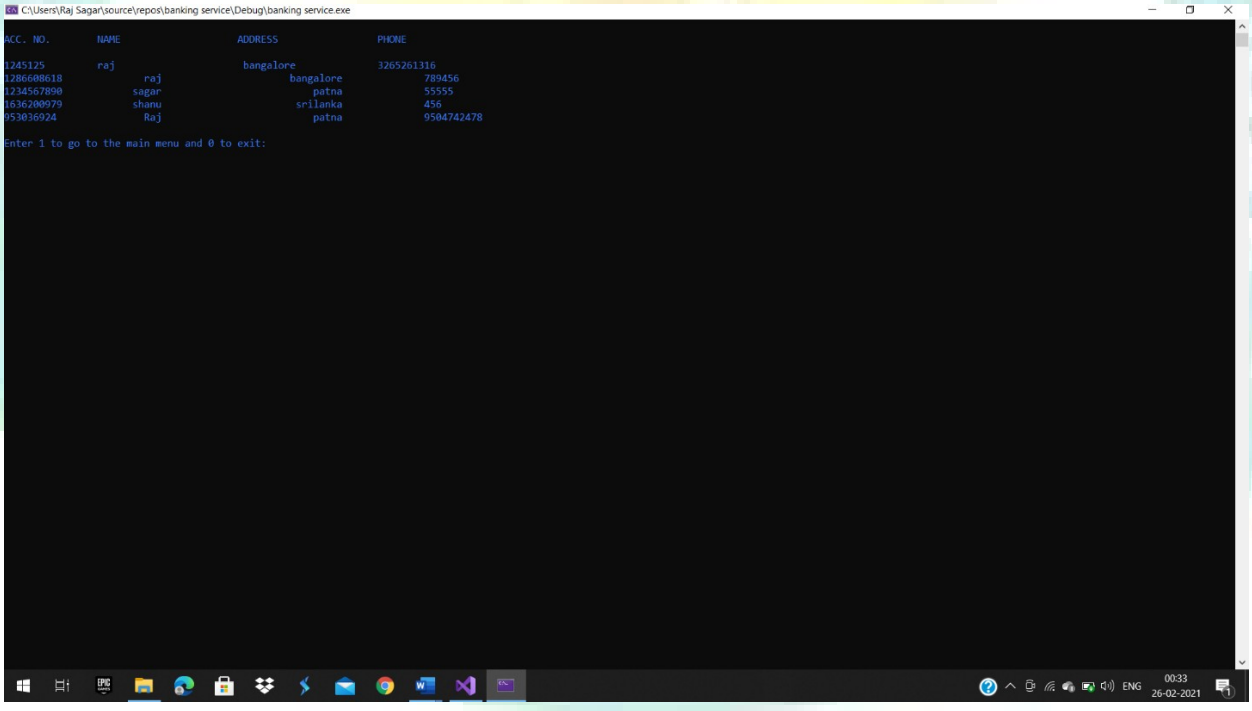

Fig no: - 7 No Of Account

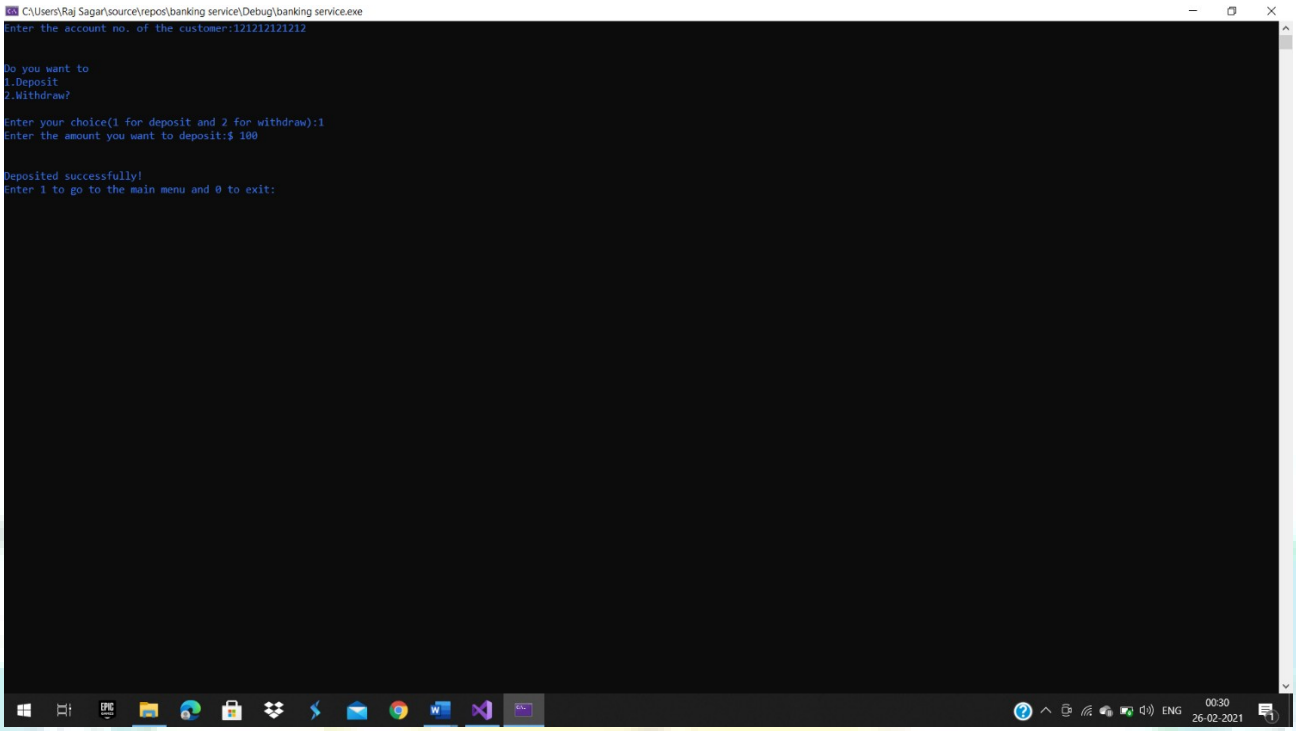

Fig no: - 8 Deposit And Withdraw

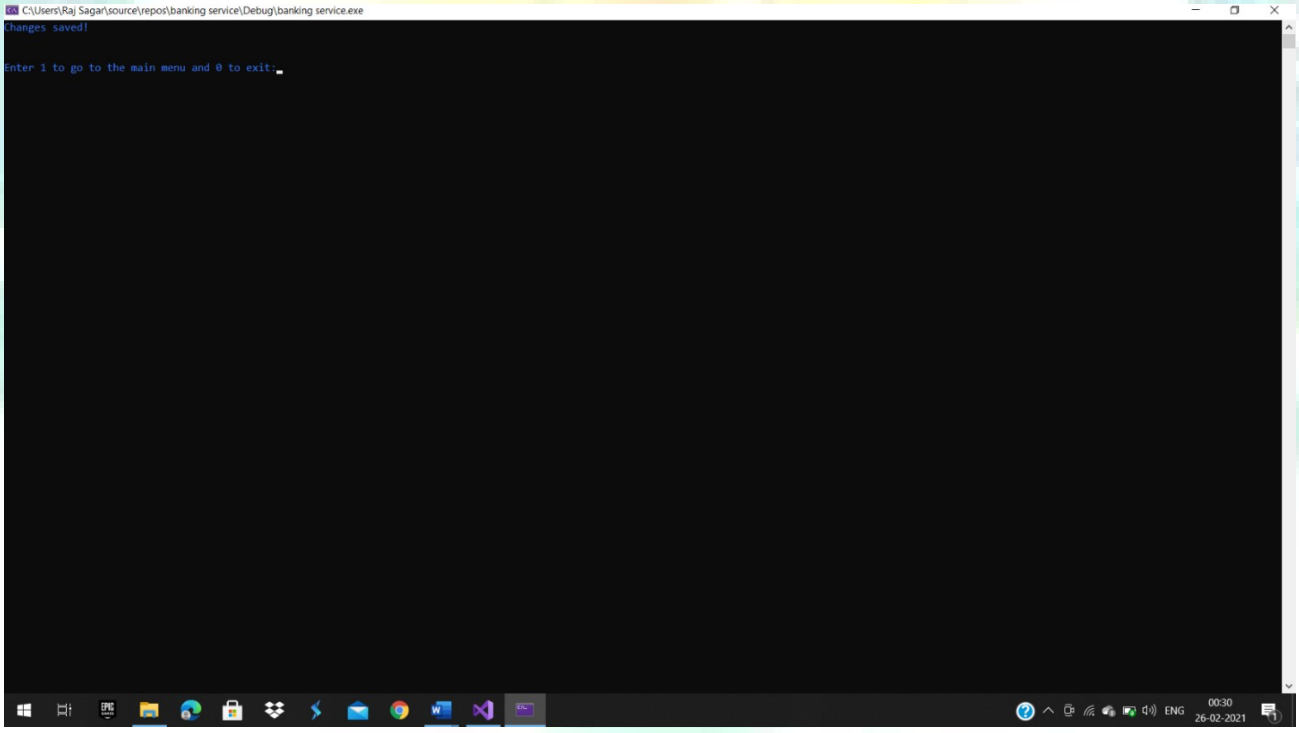

Fig no: - 9 changes

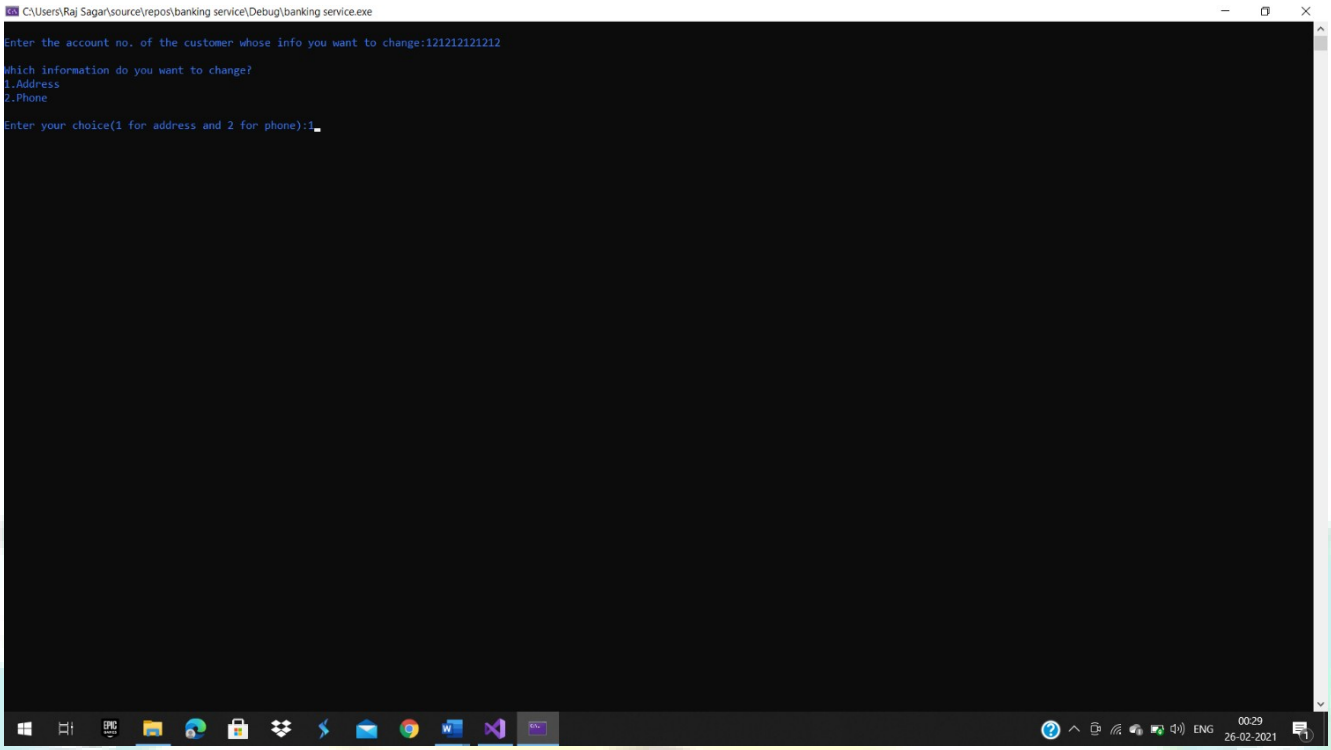

Fig no: - 10 Edit Information

Chapter 6

CONCLUSION

"Banking service" keeps the day-by-day tally record as a complete banking. It can keep the information of account type, account opening form, Deposit, Withdrawal and Searching the transaction, transaction report, individual account opening form, Group account. The exciting part of this project is; it displays transaction reports, statistical summary of account type and interest information.

REFERENCE

https://www.edureka.co/blog/c-data-structures/

https://www.tutorialspoint.com/data_structures_algorithms/linked_list_program_in_c.htm