# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI-590018, KARNATAKA

A Mini Project Report On

# “ATM MANAGEMENT SYSTEM ”

Submitted in the partial fulfillment of the requirement for the award of degree of

## BACHELOR OF ENGINEERING
in
## INFORMATION SCIENCE AND ENGINEERING

Submitted By

| | |
|---|---|
| BHARGAV S | 1VA20IS006 |
| CHAITRA Y J | 1VA20IS012 |

Under the Guidance of

Prof. Radha.R
Assistant Professor
Department of IS&E

Prof. Divya Mani M K
Assistant Professor
Department of IS&E

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

## SAI VIDYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka | Approved by AICTE, New Delhi)

Accredited by NBA, New Delhi (CSE, ISE, ECE, MECH, CIVIL) & NAAC- ‘A’ Grade

RAJANUKUNTE, BENGALURU – 560 064

2022-23

# SAI VIDYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka | Approved by AICTE, New Delhi)
Accredited by NBA, New Delhi (CSE, ISE, ECE, MECH, CIVIL) & NAAC- 'A' Grade

Rajanukunte, Bengaluru- 560 064

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

## CERTIFICATE

Certified that the mini project work entitled "**ATM MANAGEMENT SYSTEM"** carried out by **Mr. BHARGAV S (1VA20IS006), Ms. CHAITRA Y J (1VA20IS012),** a bonafide students of SAI VIDYA INSTITUTE OF TECHNOLOGY, Bengaluru, in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said Degree.

Prof. Radha.R
Assistant Professor,
Dept. of IS&E, SVIT

Prof. Divya Mani M K
Assistant Professor,
Dept. of IS&E, SVIT

Dr.Vrinda Shetty
Professor &Head
Dept. of IS&E, SVIT

Dr. M S Ganesha Prasad
Principal
SVIT

| Name of External Examiners | Signature with Date |
|---|---|
| 1. | |
| 2. | |

# ACKNOWLEDGEMENT

The completion of project brings with and sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost, we wish to express our deep sincere feelings of gratitude to our Institution, Sai Vidya Institute of Technology, for providing us an opportunity to do our education.

We would like to thank the Management and Prof. M R Holla, Director, Sai Vidya Institute of Technology for providing the facilities.

We extend our deep sense of sincere gratitude to Dr. M S Ganesha Prasad, Principal, Sai Vidya Institute of Technology, Bengaluru, for having permitted to carry out the project work on "ATM MANAGEMENT SYSTEM " successfully.

We are thankful to Dr. A M Padma Reddy, Director (A), Professor and Dean (Student affairs), Department of Computer Science and Engineering, Sai Vidya Institute of Technology, for his constant support and motivation.

We express our heartfelt sincere gratitude to Dr. Vrinda Shetty, Professor and HOD, Department of Information Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for her valuable suggestions and support.

We express our sincere gratitude to Prof. Radha R, Assistant Professor, Project Guide, Department of Information Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for her constant support.

Finally, we would like to thank all the Teaching, Technical faculty and supporting staff members of Department of Information Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for their support.

| | |
|---|---|
| BHARGAV S | 1VA20IS006 |
| CHAITRA Y J | 1VA20IS012 |

# ABSTRACT

An ATM (Automated Teller Machine) C++ menu-based project is a computer program that simulates the functions of an ATM machine. The program allows users to perform various banking transactions such as withdraw, deposit, check balance, and logout, among others. The program starts with a menu screen where users can login or create a new account, after choosing the operation they want to perform. The user is prompted to enter their username and password to access their account. If the username and password entered are correct, the user is taken to the transaction menu. In the transaction menu, the user can select the desired transaction and enter the necessary details such as the amount to be withdrawn or deposited, the account details for transfer, and can logout from it after all operations. The program performs the transaction and updates the account balance accordingly. The program also provides error messages if the user enters incorrect account details or insufficient balance for withdrawal. The user can choose to perform multiple transactions or exit the program at any time. The application of an ATM C++ project can be to provide a platform for users to perform basic banking transactions such as withdrawing money, depositing money, checking account balance, and transferring funds between accounts. Overall, an ATM C++ project can be a valuable tool for students to learn C++ programming concepts, as well as providing a simulated platform for users to perform basic banking transactions with good security.

# Table of Contents

# List of Figures

# CHAPTER 1

## INTRODUCTION

### 1.1 AIM

The “ATM Management System” project is a model Internet Banking Site. This site enables the customers to perform the basic ATM transactions by sitting at their office or at home through PC or laptop. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also view reports of all accounts present. The customers can access the ATM websites for viewing their account details and perform transactions on account as per their requirements.

### 1.2 PROBLEM STATEMENT

This report highlights the design of a standalone application. The Application is managing the details of respective customers and their accounts. Here in this application the customer can create account themselves and every customer can login to their account. The customer can also make transactions online. They modify their account details by themselves without reaching the bank personnel.

### 1.3 SOLUTION TO THE PROBLEM

The ATM management system project is a program that keeps track of a client’s ATM account. This project demonstrates the operation of a banking account system and covers the essential functions of ATM management software. It develops a project for resolving a customer’s financial applications in bank environment to meet the needs of an end banking user by providing multiples ways to complete banking chores. This project’s main goal is to create software for a ATM management system. This project was designed to make it simple and quick to complete previously impossible processes with manual systems which are now possible with this software.

## 1.4 EXISTING TECHNIQUE

In existing technique, there are several complex ways of handling the issues across the world, but this project helps in resolving a customer’s financial applications in a banking environment to meet the needs of an end user by providing multiple ways to complete banking chores online. Whereas in existing system the customer has to visit the atm personally for transactions, modification or changes to be done in their account even in their busy schedule, Also in this system there is a lacking security in a manual management of ATM accounts.

## 1.5 PROPOSED TECHNIQUE

The technique proposed is to eliminate the traditional way of physically maintaining a records system, we use the concept of primary indexing where each record in particular file is assigned a key which helps us to easily access a particular record. To protect the data form being accessed by anyone only the admin or authorized person is allowed to make any changes in the file system, this way only the person with authority can access to the sensitive information that needs to be stored.

## 1.6 OBJECTIVES

Our objective is to develop a software program for managing the entire bank process related to administration accounts, customer accounts and to keep track of each and every property and transaction process efficiently. Hereby, our main objective is the customer’s satisfaction considering today’s faster growth of technology worldwide. Client can do his operations comfortably without any risk or losing of his privacy. Our software will perform and fulfil all the tasks that any customer would desire. Client doesn’t need to go to the bank to do small operation. It helps the customer to be satisfied and comfortable in his choices, without worrying about protection of his bank account, money, and his privacy or safety.

## 1.7 SCOPE OF PROJECT

Depending on the banks policies, bank personnel can /or customer can utilize the a ATM management system. It can be utilized by multiple employees at the same time if they have the necessary permissions. It is accessed by any web browser with a graphical interface.

## 1.8 INTRODUCTION TO FILE STRUCTURES

File structures is the organization of data in secondary storage devices in such a way that minimizes the access time and the storage space.

### 1.8.1 FILE STRUCTURE:

It Is A Combination Of

- Representation for data in files and
- The operations for accessing the data

It allows applications to read write and modify the data search. An improvement in file structure design makes application hundred times faster. To access the data faster from the storage disk we use file structure. Disks are slow which are used to pack thousands of megabytes of data.

### 1.8.2 RECORDS AND ITS TYPES

Are the collection of fields, possibly of different datatypes, typically in fixed number of sequences.

The fields are also be called members. For example, a data could be sorted as a record containing a numeric year field, a month field represented as string, and a numeric day of month field. Records are distinguished from arrays by the fact that the number of fields is typically fixed, each filed has a name, and that each field may have a different types.

### 1.8.3 WHY WE NEED FILE STRUCTURES

As we know without proper structure of organizing is ordinary system, it generates some types of problems, So to avoid this kind of problems we go for file structure.

## 1.9 PRIMARY INDEXING

A primary index consists of all prime-key attributes of a table and a pointer to physical memory address of the record of data file. To retrieve a record on the basis of all primary key attributes, primary index is used for fast searching. Binary search is done on index table and then directly retrieve that record from physical memory. It may be spares.

Advantages of primary index

- Search operation is very fast.
- Index table record is usually smaller.
- A primary index is guaranteed not to duplicate.

Disadvantages of primary index

- There is only one primary index of a table. To search a record on less than all primekey attributes, linear search is performed on index table.
- To create a primary index of an existing table, record should be in some sequential order otherwise database is required to be adjusted.

In the below fig.1.9 we can observe the indexing concept.

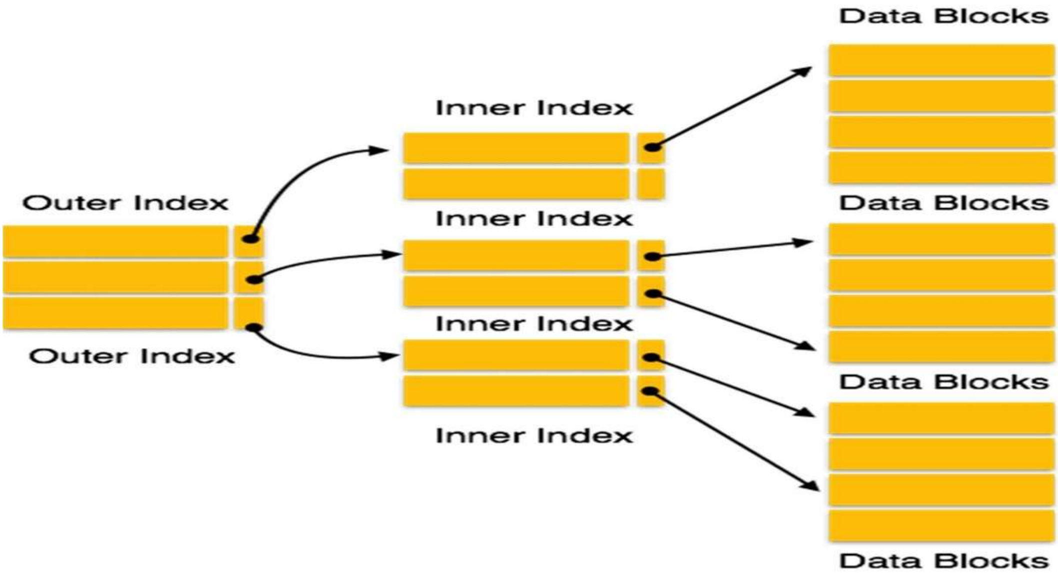


fig: 1.9 Indexing

# CHAPTER 2

## SYSEM AND SOFTWARE REQUIREMENTS AND SPECIFICATIONS

### 2.1 SOFTWARE REQUIREMENTS:

| | |
|---|---|
| Programming language | : C++ |
| Application required | : Dev C++ |
| Operating system | : Windows 7 and later |

### 2.2 HARDWARE REQUIREMENTS:

| | |
|---|---|
| C.P.U | : Pentium IV 2.4 GHz or above |
| Memory (primary) | : 2GB or above |
| Hard disk | : 40 GB, 80 GB, 160 GB or above |
| Monitor | : 15 VGA color |

### 2.3 FUNCTIONAL REQUIREMENTS

A description of this facility or feature required. Functional requirements deal with what the system should do or provide for users, They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

#### 2.3.1 INTERFACE REQUIREMENTS

- The system shall provide an option for customer to register.
- The system should give option for admin login.
- The system shall provide users to request for their services.

## 2.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the overall qualities or attributes of the resulting system.

### 2.4.1 USABILITY

Usability is the ease with which a user can learn to operate and get instant information about products and its services.

### 2.4.2 SECURITY

Security requirements are included in a system to ensure:

- Enquires from the end users as well as from any organization are highly secured.

### 2.4.3 RELIABILITY

Reliability is the ability of a system to perform its required functions under stated conditions for a specific period of time. Constraints on the run-time behaviour of the system can be considered under two separate headings:

- Availability: is the system available for service when requested by end-users.
- Failure rate: how often does the system fail to deliver the service as expected by endusers.

### 2.4.4 EFFICIENCY

The comparison of what is actually produced or performed with what can be achieved with the same consumption of clouds (money, time, labour etc.). It is an important factor in determination of productivity.

# CHAPTER 3

# SYSTEM DESIGN

This chapter of the report describes the structure of the project, followed by Data flow Diagram.

## 3.1 DATA FLOW DIAGRAM AND ITS NOTATIONS

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects, A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity work flow diagram, which presents both control and data, flows as a unified model.

All data flow diagrams include four main elements: entity, process, data store and data flow.

### 3.1.1 EXTERNAL ENTITY

It is also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entities and the system being diagrammed. These data flows are the inputs and outputs of the DFD. Since they are external to the system being analysed, these entities are typically placed at the boundaries of the diagram. They can represent another system or indicate a subsystem.

### 3.1.2 PROCESS

An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD. This symbol is given a simple name based on its function, such as “search” rather than being labelled “process” on a diagram. In Ganersarson notation, a rectangular box is used and may be labelled with a reference number, location of where in the system the process occurs and a short title that describes its function. Processes are typically oriented from top to bottom and left to right on a data flow diagram.

## 3.1.3 DATA STORE

A data store does not generate any operations but simply holds data for later access. Data stores could consist of files held long term or a batch of documents stored briefly while they wait to be processed. Input flows to a data store include information or operations that change the stored data. Output flows would be data retrieved from the store.

## 3.1.4 DATA FLOW

Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow. This data could be electronic, written or verbal. Input and output data flows are labelled based on the type of data or its associated process or data store, and this name is written alongside the arrow, fig.3.1 shows the notations which are used in data flow.

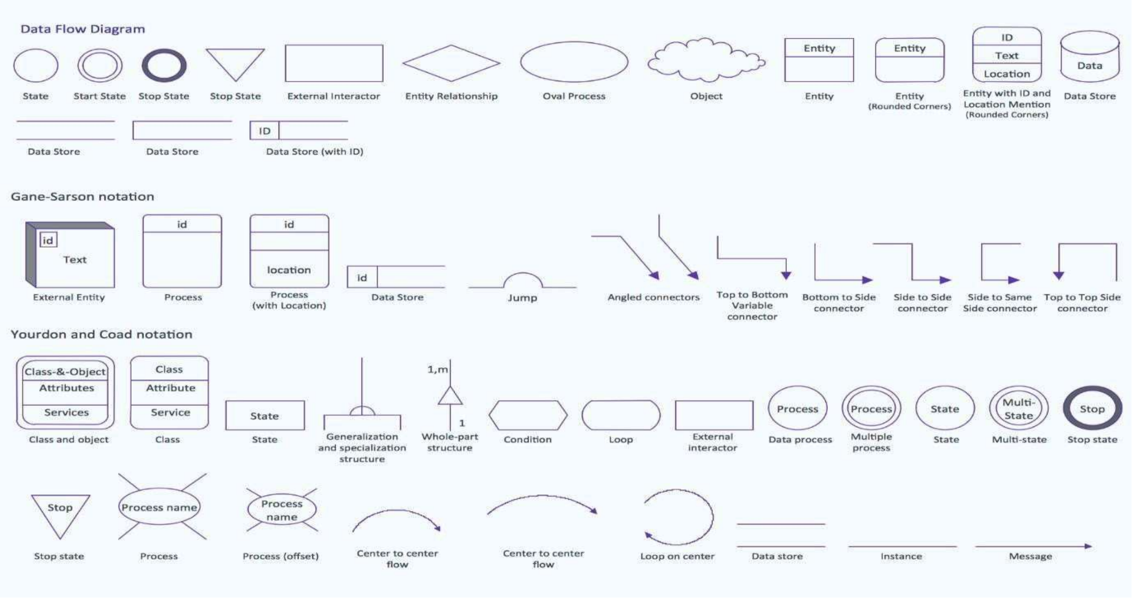


fig: 3.1 Notations

### 3.2 DATA FLOW DIAGRAM EXAMPLE

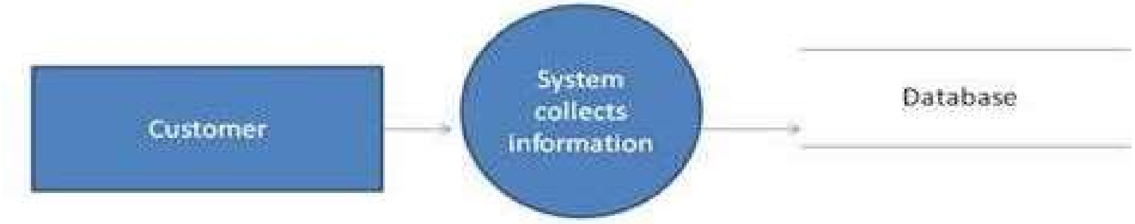


fig: 3.2 DFD example

## 3.3 DATA FLOW DIAGRAM:

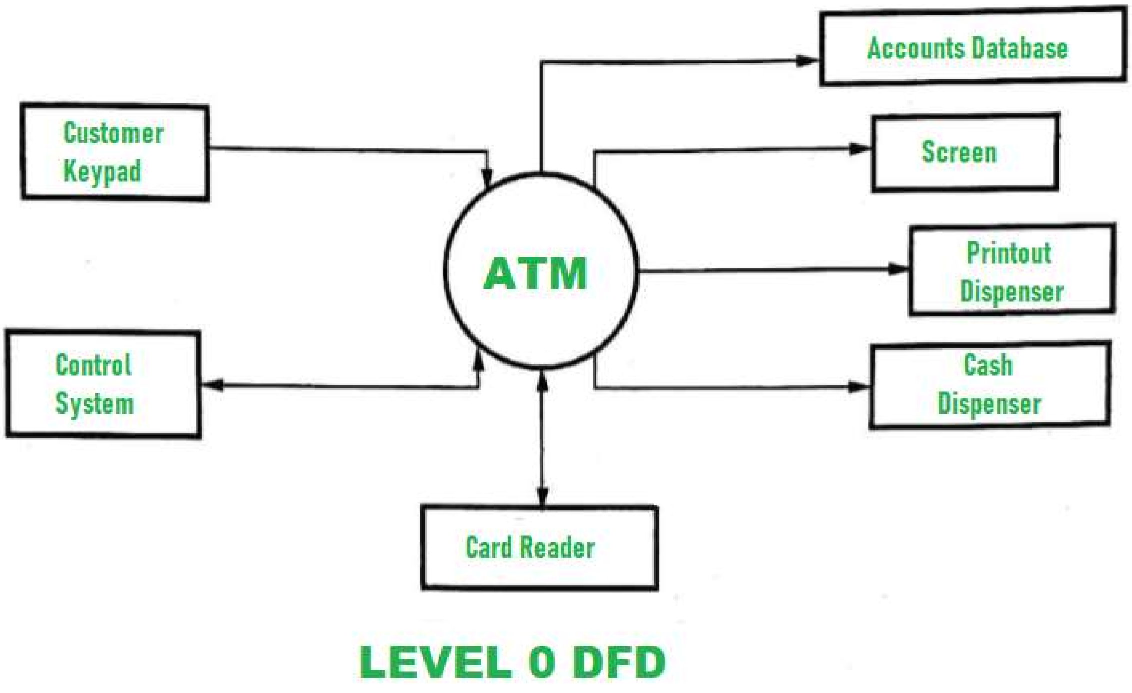


fig: 3.3 Design structure

Above fig.3.2 shows the data flow of the database and fig.3.3 shows the data flow of the atm mini project.

# CHAPTER 4

## IMPLEMENTATIONS

This chapter of the report describes the functions, packages and modules used in the project.

## 4.1 LIBRARIES AND FRAMEWORKS

### 4.1.1 GCC COMPLIER

The GNU Complier Collection (GCC)is an optimizing complier product by the GNU project supporting various programming languages, hardware architectures and opertaing systems.GCC includes front ends for C (gcc), C++ (g++), objective-C, Fortran (gfortran), go (gccgo) and D (gdc, since 9.1) programming languages, with the openMP and openACC parallel language extensions being supported since GCC 5.1 versions prior to GCC 7 also supported Javav(gcj), allowing compilatin of java to antive machine code. To installGCC on ububutu follow along these steps:

- $sudo apt update
- $ sudo apt install build-essential

### 4.1.2 LINUX BASED SYSTEM

Linux is a UNIX=like, open source and community-developed operating sytsem(os) for computers, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform, including x86, ARM and SPARC, making it one if the most widely supported operating systems. Every version of the Linux is OS manages hardware resources, launches and handles applications, and provides some distributions means that a Linux versions is available for almost any task, and Linux has penetrated many areas of the computing. The Linux OS can be fond is many different settings, supporting many different use cases.

## 4.2 PROGRAMMING LANGUAGE SELECTION

C++ is a general purpose, case-senstive, free-form programming language that supports object oriented, procedural and generic programming.

## 4.3 FUNCTIONAL MODULES

The functional modules included in the project are listed below:

### 4.3.1 INSERT MODULE:

This module provides the functionality of collecting the required data from the designed interface and transmitting it to the appropriate record present in the file designed for this project. If the provide data does not satisfy the given constraints, it must refrain from storing it into the file. Customers can only able to create account.

### 4.3.2 DISPLAY MODULE:

This module is used to display all the customers which are applied for accounts. This display function displays the records in the order the customers applied accounts. This functionality is only accessed by admin.

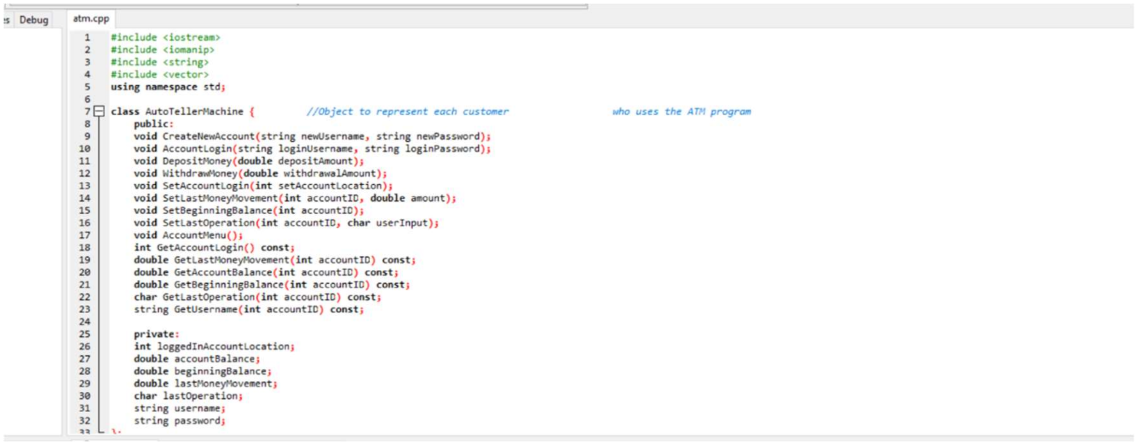
### 4.3.3 SEARCH MODULE:

The search module has a basic functionality of accessing the record in the data file using customers name and displaying it in the text field. The data is searched using the sequential order searching.

### 4.3.4 MODIFY MODULE:

The modify module has an ability to modify certain record and place it before position. This functionality is accessible only to custom.

## 4.4 IMPORTANT CODE SNIPPETS



```cpp
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>
using namespace std;

class AutoTellerMachine {          //Object to represent each customer                    who uses the ATM program
    public:
    void CreateNewAccount(string newUsername, string newPassword);
    void AccountLogin(string loginUsername, string loginPassword);
    void DepositMoney(double depositAmount);
    void WithdrawMoney(double withdrawalAmount);
    void SetAccountLogin(int setAccountLocation);
    void SetLastMoneyMovement(int accountID, double amount);
    void SetBeginningBalance(int accountID);
    void SetLastOperation(int accountID, char userInput);
    void AccountMenu();
    int GetAccountLogin() const;
    double GetLastMoneyMovement(int accountID) const;
    double GetAccountBalance(int accountID) const;
    double GetBeginningBalance(int accountID) const;
    char GetLastOperation(int accountID) const;
    string GetUsername(int accountID) const;

    private:
    int loggedInAccountLocation;
    double accountBalance;
    double beginningBalance;
    double lastMoneyMovement;
    char lastOperation;
    string username;
    string password;
};
```

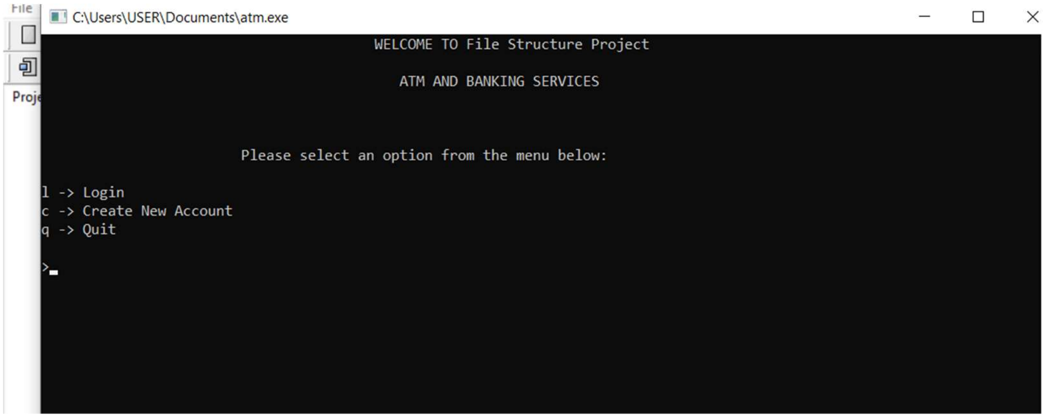
fig: 4.4 A class called AutoTellerMachine with its function definition

A class named AutoTellerMachine is defined where we define the functions CreateNewAccount(),Accountlogin(),DepositeMoney(),WithdrawMoney(),SetAccountLogin(),SetBeginningBalance(),SetLastOperation(),GetAccountLogin(),GetAccountBalance(),GetBeginningBalance(),GetUsername(). In fig 4.4 we can observe the code.

- CreateNewAccount(): This function is used to create the primary index for each record that has to be inserted.
- Accountlogin (): This function is used to login to atm.
- DepositeMoney (): This function is used deposit the amount into the bank account
- WithdrawMoney (): This function is used to withdraw the money from the user account.
- SetAccountLogin (): this function is used to set the login account of the user.
- SetBeginningBalance (): this function is used to set the initial balance to zero.
- SetLastOperation (): this function is used to set the last operation performed.
- GetAccountLogin (): this function returns the account login details.
- GetAccountBalance (): this function returns the balance amount in account.

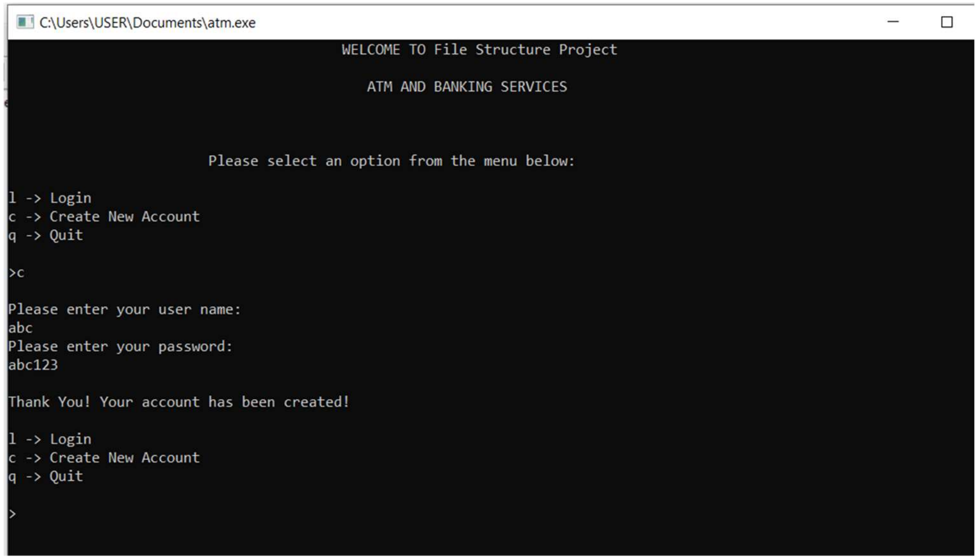
# CHAPTER 5

## RESULTS



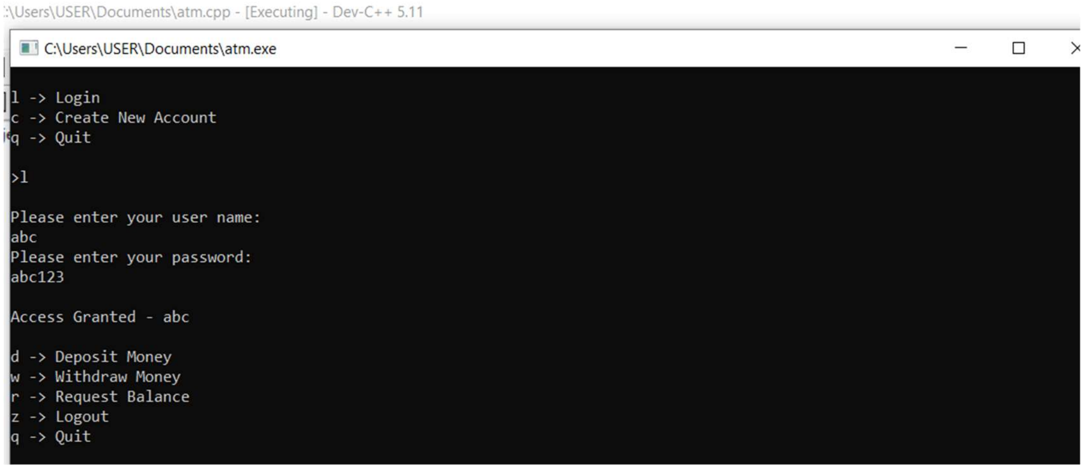
```
                                    WELCOME TO File Structure Project

                                       ATM AND BANKING SERVICES



                      Please select an option from the menu below:

l -> Login
c -> Create New Account
q -> Quit

>
```

fig:5.1 Home page



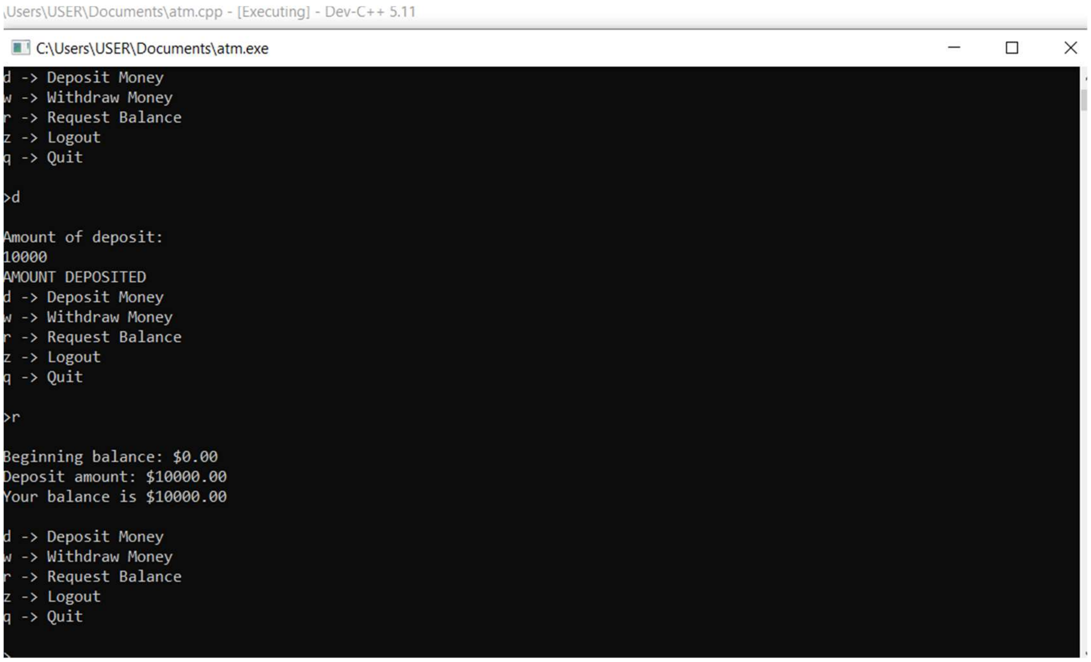
```
                                    WELCOME TO File Structure Project

                                       ATM AND BANKING SERVICES



                      Please select an option from the menu below:

l -> Login
c -> Create New Account
q -> Quit

>c

Please enter your user name:
abc
Please enter your password:
abc123

Thank You! Your account has been created!

l -> Login
c -> Create New Account
q -> Quit

>
```

fig: 5.2 Creating new account


:\Users\USER\Documents\atm.cpp - [Executing] - Dev-C++ 5.11
C:\Users\USER\Documents\atm.exe

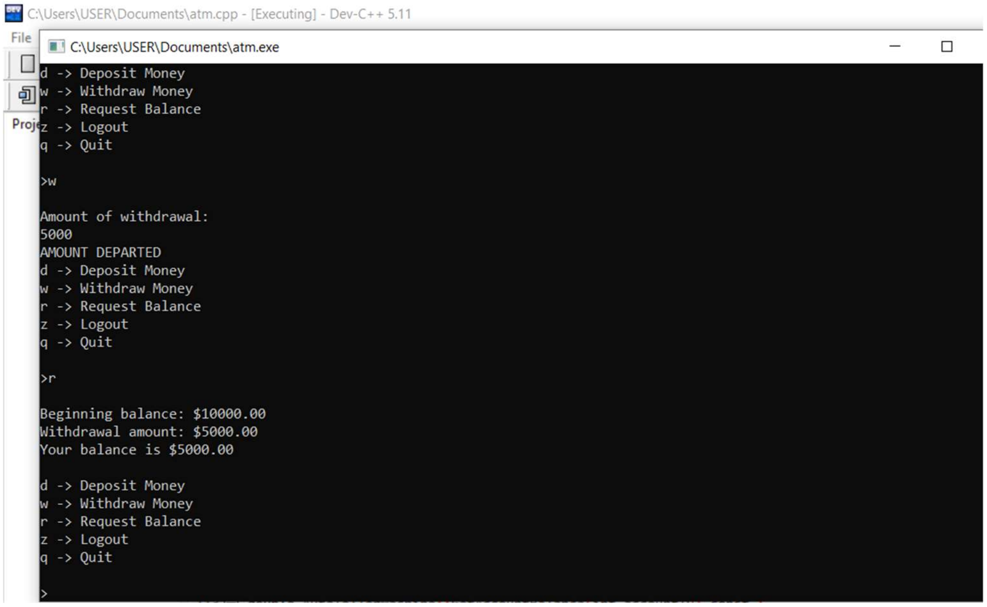
```
l -> Login
c -> Create New Account
q -> Quit

>l

Please enter your user name:
abc
Please enter your password:
abc123

Access Granted - abc

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit
```

fig: 5.3 login page


\Users\USER\Documents\atm.cpp - [Executing] - Dev-C++ 5.11
C:\Users\USER\Documents\atm.exe

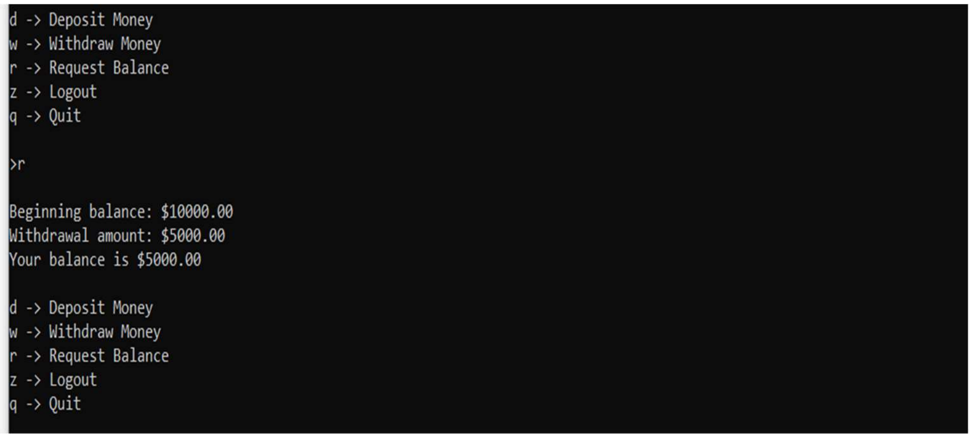
```
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>d

Amount of deposit:
10000
AMOUNT DEPOSITED
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>r

Beginning balance: $0.00
Deposit amount: $10000.00
Your balance is $10000.00

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit
```

fig: 5.4 Deposit amount


C:\Users\USER\Documents\atm.cpp - [Executing] - Dev-C++ 5.11
File
Proje
C:\Users\USER\Documents\atm.exe

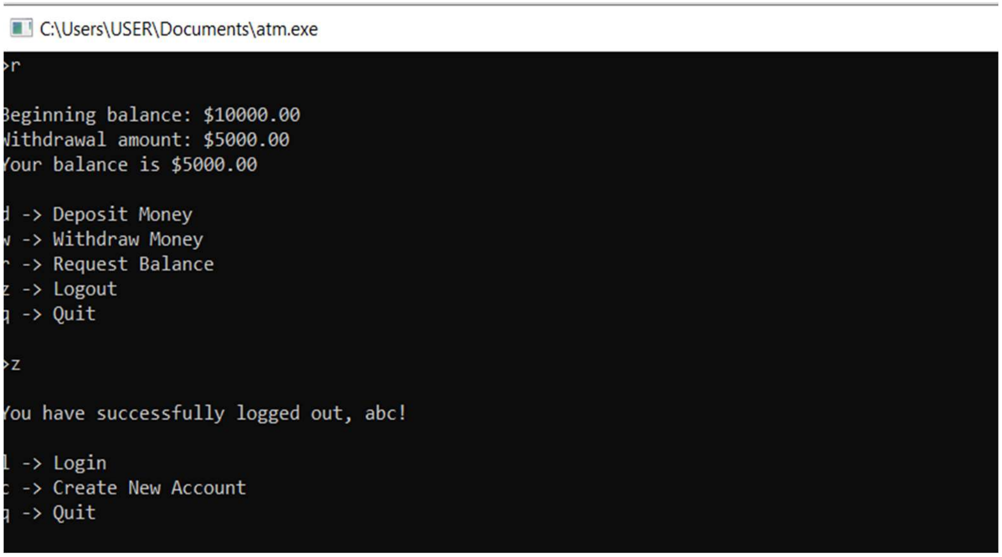
```
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>w

Amount of withdrawal:
5000
AMOUNT DEPARTED
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>r

Beginning balance: $10000.00
Withdrawal amount: $5000.00
Your balance is $5000.00

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>
```

fig: 5.5 Amount withdrawal



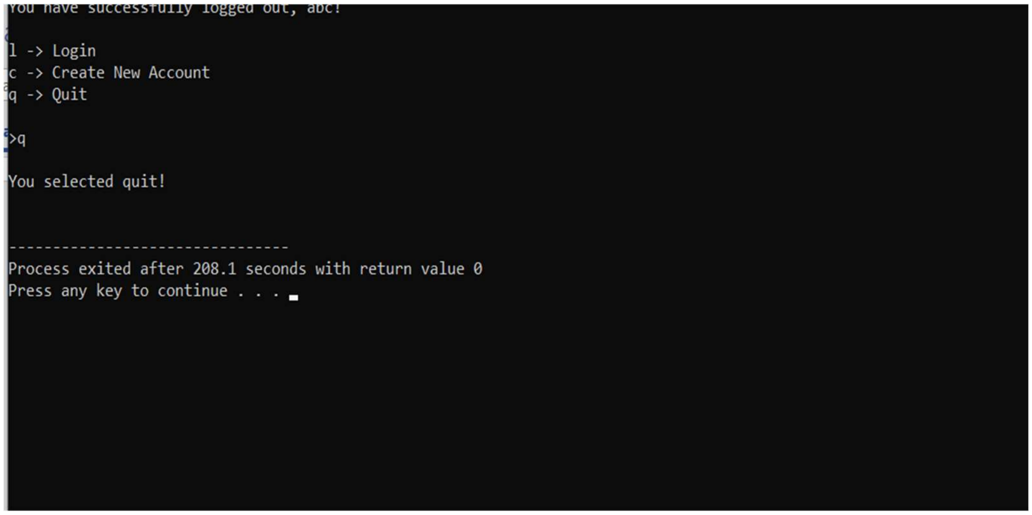
```
d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>r

Beginning balance: $10000.00
Withdrawal amount: $5000.00
Your balance is $5000.00

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit
```

fig: 5.6 check the balance



```
>r

Beginning balance: $10000.00
Withdrawal amount: $5000.00
Your balance is $5000.00

d -> Deposit Money
w -> Withdraw Money
r -> Request Balance
z -> Logout
q -> Quit

>z

You have successfully logged out, abc!

l -> Login
c -> Create New Account
q -> Quit
```

fig: 5.7 Logout page



```
You have successfully logged out, abc!

l -> Login
c -> Create New Account
q -> Quit

>q

You selected quit!


--------------------------------
Process exited after 208.1 seconds with return value 0
Press any key to continue . . .
```

fig: 5.8 Exit page

# CHAPTER 6

## CONCLUSION

The application certainly has some striking feature over manual system. user queries have become quite accurate and efficient. Lot of paper work has been eliminated. Future modification and enhancements have become quite easier now in comparison to the previous manual system. Last but one of the most important advantages of the ATM system is that, through this system the whole procedure will take too less time in comparison of the manual system. No doubt ATM will be helpful for institutes in all procedures, which will be monitoring through account. At the first step ATM will only be installed in the bank.

The main advantage of ATM is that, it will become a powerful tool in establishment of better system in comparison of the existing system. It helps to protect the system from corruption. After installation of ATM in the bank, there is a greater possibility of stabilization at clear and fair system, which will be accurate, update and fast. There is no doubt that there always remains some scope of improvement. The important thing is that the system developed should be flexible to accommodate any future enhancements. This system can be used to provide some enhancement without rewriting of existing code.

The system comprises of the following features:

- Management of customers details.
- Management of customer accounts.
- Management of changing customers account Details.
- Management of Displaying all the account details.
- Management of reaching for specific account details.
- Management of checking all the transaction.

# SCOPE OF ENHANCEMENT

- There are also few features which can be integrated with the system to make it more flexible.
- Below lists shows the future points to be considered:
- View the account details only for the user.
- Creating the account is done by customer.
- Adding the user and admin privilege for the accessing the account and for the security.
- Furthermore with its implementation very large amount of data will be secure and editing and addition or deletion of data is done very easily.
- In future according to the user requirement it can be updated so that to reach the user specification.

# REFERENCES

**BOOKS:**

- File structures, An Object Oriented approach with C++ (by Michale J Folk, Bill Zoellick, Greg Riccardi).
- File structures using C++ (by K R Venugopal, K G Srinivas, P M Krishnaraj, Tata McGraw Hill)

**WEBSITES:**

- WWW.stackoverflow.com
- WWW.geeksforgeeks.com
- WWW.W3Schools.com
- WWW.youtube.com