**ABSTRACT**

Bank Management System is a project where all the banking operations are performed easily and efficiently by computer. Unlike online banking system this bank management system application covers all the transactions like new account opening, deposit and withdrawal of amount as well as transaction of money for various processes. Bank management system will perform all the tasks that any customer may desire. Our aim is to develop a vb project for managing the entire banking process related to customer accounts and their various transaction processes. Our main purpose is the customer’s satisfaction. For any system, present satisfaction is important, but it is also necessary to see and visualizes the future scope. Future enhancement is necessary for any system as the limitations that cannot be denied by anybody. These limitations can be overcome by better technologies. In my project, records of the customers and transactions are maintained. It will be helpful for the organization and customer.

**CHAPTER 1**

**INTRODUCTION**

Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank. Now a days, managing a bank is tedious job upto certain limit. So software that reduces the work is essential. Also today’s world is a genuine computer world and is getting faster and faster day-by-day. Thus, considering above necessities, the software for bank management has became necessary which would be useful in managing the bank more efficiently.

* Our software will perform and fulfill all the tasks that any customer would desire.
* Our motto is to develop a software program for managing the entire bank process related to customer accounts, employee accounts and to keep each every track about their property and their various transaction processes efficiently.
* Hereby, our main objective is the customer’s satisfaction considering today’s faster world.

In the recent years, computers are included in almost all kind of works and jobs everyone come across in the routine. The availability of the software’s for almost every process or every system has taken the world in its top-gear and fastens the day-to-day life. So, we have tried our best to develop the software program for the Bank Management System where all the tasks to manage the bank system are performed easily and efficiently. It manages all the transactions like new account entry, deposit as well as withdraw entry, transaction of money for various processes, loan entry, managing bills cash or cheque, etc. Thus, above features of this software will save transaction time and therefore increase the efficiency of the system. Requirements definition and management is recognized as a necessary step in the delivery of successful system s and software projects, discipline is also required by standards, regulations, and quality improvement initiatives. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budge. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment.

**CHAPTER 2**

**AIM, OBJECTIVE AND SCOPE OF THE PROJECT**

**2.1 Aim of the project**

To develop a software for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user workspace to have additional functionalities which are not provided under a conventional banking software.

**2.2 Objective of the project**

The main aim of this project is to develop software for bank management system. This project is to develop software for bank management system. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using VB language and. Hence it provides the complete solution for the current management system.

**2.3 Scope of this project**

Bank Management System is based on C++ & file structure and is a major project for students. It is used to Keep the records of clients, employee etc in Bank. The bank management system is an application for maintaining a personal account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present.

* Creating New Accounts- The application can be used to create two different types of accounts by the customers, which are Savings Account and Current Account. It helps save the hustle for the customer to visit the bank physically and create/use these accounts.
* Depositing Money- As the world is moving towards the limited use of paper currency, depositing or transferring money from one bank to the other will become as easy as clicking a few buttons using this application.
* Withdrawing Money- Requests can be sent through the application to ask for money transfer as well.
* Account Holder List- This is a feature for the admin. The admin can view the list of all the account holders.
* Balance Enquiry- The customer can check their balance via this application.
* Changing Passwords/PIN- The customer can easily change the passwords and pin numbers using the application.
* Closing- The customer can close their accounts too using this application.

**CHAPTER 3**

**REQIREMENT SPECIFICATION**

**3.1 System Requirements**

The basic requirements for the development of this mini project are as follows

**3.1.1 Hardware Configuration**

1. Processor: Intel core i3 or above
2. Ram:512 MB
3. Hard disk:20 GB

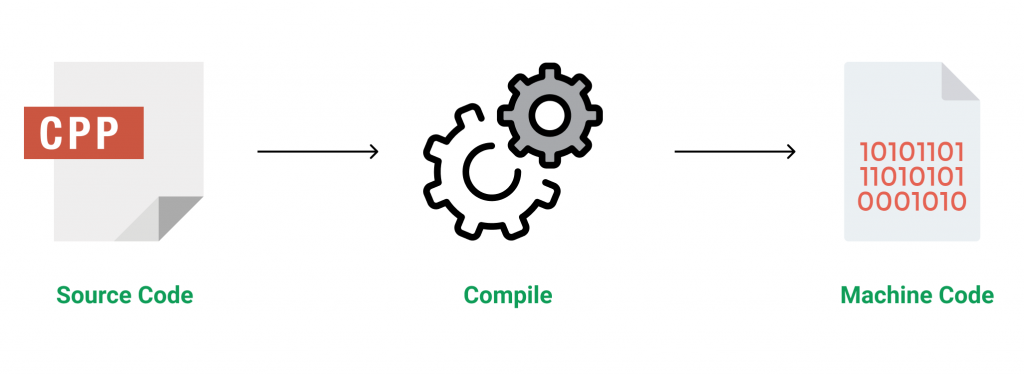
**3.1.2 Software Configuration**

1. Language: C & C++
2. IDE: DEV C++
3. Documentation tool: Microsoft office 2003 or above

**3.2 Development Environment**

**3.2.1 C++**

**C++** is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language[7].



**Figure 3.1 Source code compilation**

C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the features & key-points to note about the programming language are as follows:

* Simple: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
* Machine Independent but Platform Dependent: A C++ executable is not platform-independent (compiled programs on Linux won’t run on Windows), however they are machine independent.
* Mid-level language: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
* Rich library support: Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development.
* Speed of execution: C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as garbage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast.
* Pointer and direct Memory-Access: C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables).
* Object-Oriented: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
* Compiled Language: C++ is a compiled language, contributing to its speed.

**3.2.2 File Structure**

In computing, file system or filesystem (often abbreviated to fs) is a method and data structure that the operating system uses to control how data is stored and retrieved.[1] Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stopped and the next began, or where any piece of data was located when it was time to retrieve it [5]. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way a paper-based data management system is named, each group of data is called a "file”. The structure and logic rules used to manage the groups of data and their names is called a "file system”.

A file system consists of two or three layers. Sometimes the layers are explicitly separated, and sometimes the functions are combined. The logical file system is responsible for interaction with the user application. It provides the application program interface (API) for file operations — OPEN, CLOSE, READ, etc., and passes the requested operation to the layer below it for processing. The logical file system "manage[s] open file table entries and per-process file descriptors".[5] This layer provides "file access, directory operations, [and] security and protection".

The second optional layer is the virtual file system. "This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation". The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. The physical file system interacts with the device drivers or with the channel to drive the storage device.

* + 1. **Files Operation**

A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files. Six basic file operations. The OS can provide system calls to create, write, read, reposition, delete, and truncate files.

* Creating a file: Two steps are necessary to create a file.
  + Space in the file system must be found for the file.
  + An entry for the new file must be made in the directory
* Writing a file: To write a file, we make a system call specifying both the name of the file and the information to be written to the file. The system must keep a write pointer to the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.
* Reading a file: To read from a file, we use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. The system needs to keep a read pointer to the location in the file where the next read is to take place.
* Because a process is usually either reading from or writing to a file, the current operation location can be kept as a per-process current-file-position pointer.
* Both the read and write operations use this same pointer, saving space and reducing system complexity.
* Repositioning within a file: The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seek.
* Deleting a file: To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase the directory entry.
* Truncating a file: The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.

These six basic operations comprise the minimal set of required file operations. These primitive operations can then be combined to perform other file operations (i.e., copying). The OS keeps a small table, called the open-file table, containing information about all open files. When a file operation is requested, the file is specified via an index into this table, so no searching is required. When the file is no longer being actively used, it is closed by the process, and the OS removes its entry from the open-file table. Most systems require that the programmer open a file explicitly with the $open () system call before that file can be used. The $open () operation takes a file name and searches the directory, copying the directory entry into the open-file table.

This call can also accept access-mode information (create, read-only, read-write, append-only, and so on). This mode is checked against the file's permissions. If the request mode is allowed, the file is opened for the process. The $open () system call typically returns a pointer to the entry in the open-file table. This pointer, not the actual file name, is used in all I/O operations. The implementation of the $open () and $close () operations is more complicated in an environment where several processes may open the file at the same time. This may occur in a system where several different applications open the same file at the same time.

Typically, the OS uses two levels of internal tables:

* A per-process table. The per-process table tracks all files that a process has open. For instance, the current file pointer for each file is found here. Access rights to the file and accounting information can also be included.
* A system-wide table. Each entry in the per-process table in turn points to a system-wide open-file table. The system-wide table contains process-independent information, such as the location of the file on disk, access dates, and file size. Once a file has been opened by one process, the system-wide table includes an entry for the file.

Typically, the open-file table also has an open count associated with each file to indicate how many processes have e the file open. Each $close () decreases this open count, and when the open count reaches zero, the file is no longer in use, and the file's entry is removed from the open-file table. In summary, several pieces of information are associated with an open file.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

Here is a project we developed as mini project in C++ bank management system. This project is focused on customer account services in bank, so it is named “Customer Account Bank Management System”. Here, you can create a new account, update information of an existing account, view and manage transactions, check the details of an existing account, remove existing account and view customers’ list. Overall, with this project, you can perform banking activities like in a REAL bank. Bank management mini project in C is a console application without graphics. It is compiled in Code::Blocks with gcc compiler.

The source code for Customer Account Bank Management System is relatively short and easy to understand. I have divided this C mini project into many functions, most of which are related to different banking activities. Listed below are some of the more important functions which may help you understand the project better.

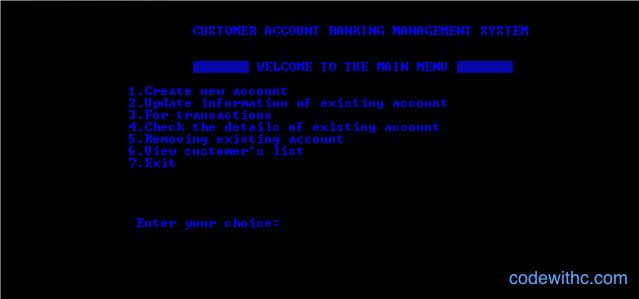
* menu() – This function displays the menu or welcome screen to perform different banking activities mentioned below.
* new\_acc() – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, citizenship number, address and phone number. You can enter the amount to deposit and choose one type of deposit account – saving, current, fixed for 1 year, fixed for 2 years or fixed for 3 years.
* view list() – With this function, you can view the customer’s banking information such as account number, name, address and phone number provided while creating the account.
* edit() – This function has been used for changing the address and phone number of a particular customer account.
* transact() – With this function, you can deposit and withdraw money to and from a particular customer account.
* erase() – This function is for deleting a customer account.
* see() – This function shows account number, name, date of birth, citizenship number, age, address, phone number, type of account, amount deposited and date of deposit. It also displays the amount of interest corresponding to a particular account type.

In this bank management system project in C, file handling has been used for almost all functions. File has been used to store data related to new account, transaction, editing of account information and viewing of account information. I haven’t used file handling for the menu, interest calculation and password.

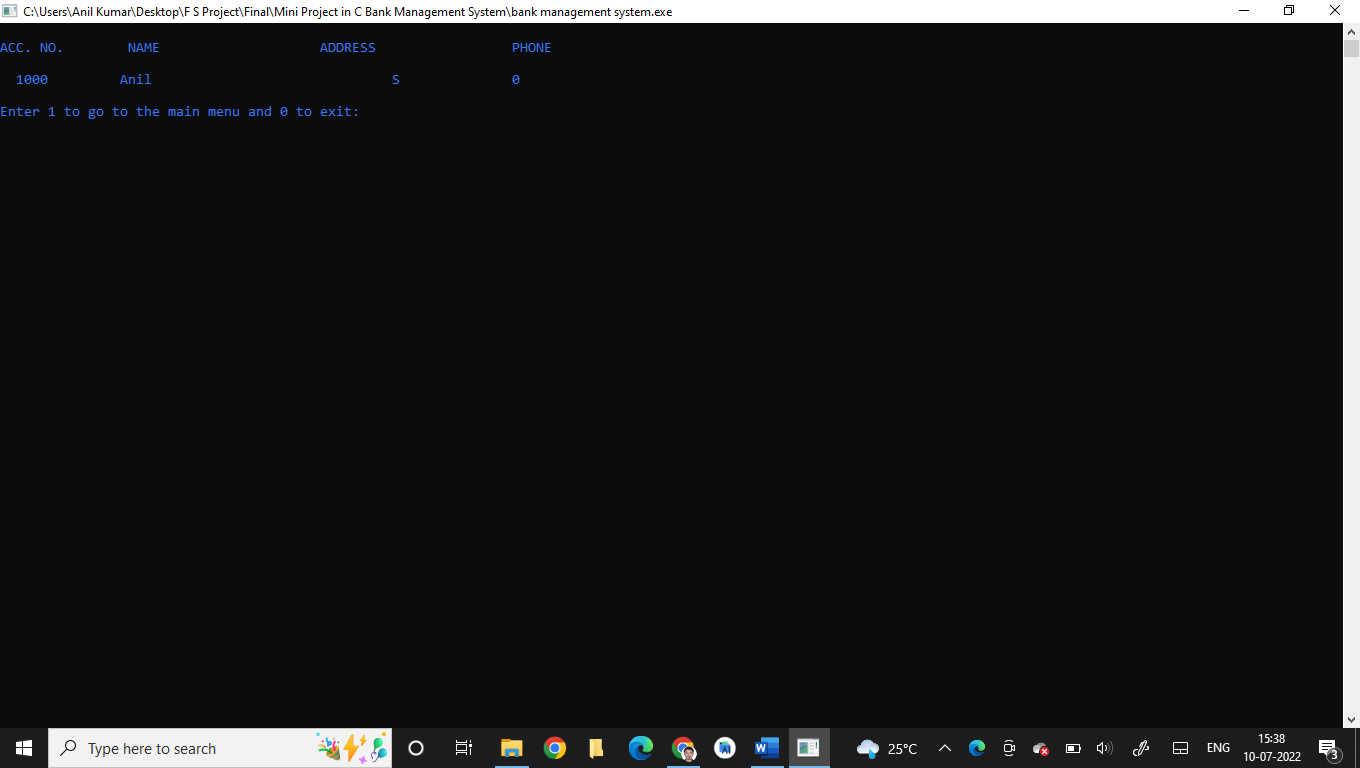
**CHAPTER 5**

**SAMPLE OUTPUT**

**5.1 Screenshots**



**Figure 5.1 Main Menu**



**Figure 5.2 Customer Information**

**CHAPTER 6**

**CONCLUSION**

Bank management system is a virtualization of transactions in banking system. The banking system are used manual working but when we used online banking system it is totally virtualization process which avoid manual process and converts it in automatic process. If user can make a transaction in bank management system it is available in any where also user can link aadhar with account, change branch location easily. Bank management system is saving the time

with accuracy than bank manual system.

The project entitled “Bank management system” is a computerized telecommunications device that provides the customers of a financial institution with access to financial transactions in a public space without the need for a human clerk or bank taller (manpower).

**BIBLIOGRAPHY**

[1] https://www.codewithc.com/mini-project-in-c-medical-store-management-system/

[2] https://www.codewithc.com

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.quora.com](http://www.quora.com)

[5] [www.wikipedia.org](http://www.wikipedia.org)

[6] <https://www.w3schools.com/cpp/cpp_files.asp>

[7] https://www.geeksforgeeks.org/introduction-to-c-programming-language/