**ABSTRACT**

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library. This project has many features like facility of user login and a facility of teachers login . It also has a facility of admin login through which the admin can monitor the whole system. It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account ie admin account can generate various reports such as student report, issue report, teacher report and book report

This mini project in C Library Management System is a console application without graphic developed using the C programming language. It is compiled in Code Blocks with the GCC compiler. In this, you can perform book-related operations like in a REAL library management system with computer.

Here, you can perform functions such as add books, return books, issue books, delete record of books issued, view record of books issued, search for books, and more. File handling has been extensively used in this project for almost all functions. So, this project can definitely guide you to understand C mini projects in a better way.

**CHAPTER 1**

**INTRODUCTION**

Library management systems are designed to manage the movement of books and maintain records of the members in a library. The software solution is designed based on the system requirements, the people involved, the content of the operation and the activity to be performed. The purpose of a library management system is to operate a library with efficiency and at reduced costs. The system being entirely automated streamlines all the tasks involved in operations of the library. The activities of book purchasing, cataloging, indexing, circulation recording and stock checking are done by the software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors.

The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs. The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books. Adding, removing or editing the database is a simple process. Adding new members or cancelling existing memberships can be done with ease.

Stock checking and verification of books in the library can be done within a few hours. The automated system saves a considerable amount of time as opposed to the manual system. The library management system software makes the library a smart one by organizing the books systematically by author, title and subject. This enables users to search for books quickly and effortlessly. Students need access to authentic information. An advanced organized library is an integral part of any educational institution. In this digital age a web based library management system would be ideal for students who can access the library’s database on their smartphones.

The system requirement in library management focuses on the possibility of search for books by title, author or subject by the member.

They should be able to locate a book physically by the unique identification code and the rack number for each book. The system should provide details on the books held by the members. The system should limit the number of books that can be taken and the number of days that a book can be kept for. The system should generate fines when due from the member. The next step focuses on the functions of the librarian, the member and the system.

**CHAPTER 2**

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

**2.1 Aim of the project**

The "Library Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

**2.2 Objective of the project**

The main objective of the Project on Library Management System is to manage the details of Student, Books, Issues, Librarian, Member. It manages all the information about Student, Address, Member, Student. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Student, Books, Address, Issues. It tracks all the details about the Issues, Librarian, Member.

Functionalities provided by Library Management System are as follows:

* Provides the searching facilities based on various factors. Such as Student, Issues, Librarian, Member
* Library Management System also manage the Address details online for Librarian details, Member details, Student.
* It tracks all the information of Books, Address, Librarian ect
* Manage the information of Books
* Shows the information and description of the Student, Issues
* To increase efficiency of managing the Student, Books
* It deals with monitoring the information and transactions of Librarian.
* Manage the information of Student
* Editing, adding and updating of Records is improved which results in proper resource management of Student data.
* Manage the information of Librarian
* Integration of all records of Member.

**2.3 Scope of this project**

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Library Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Library Management System.

* In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
* In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
* To assist the staff in capturing the effort spent on their respective working areas.
* To utilize resources in an efficient manner by increasing their productivity through automation.
* The system generates types of information that can be used for various purposes.
* It satisfies the user requirement
* Be easy to understand by the user and operator
* Be easy to operate
* Have a good user interface
* Be expandable
* Delivered on schedule within the budget.

**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].

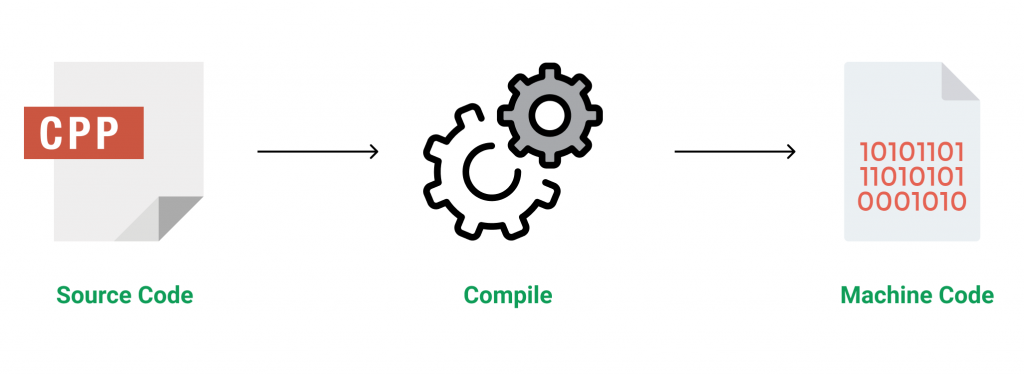


Figure 3.1 Source code compilation

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**

For library management, this project considers six departments – Computer, Electrical, Civil, Electronics, Mechanical and Architecture. These departments work simultaneously with the operations mentioned above. You can add a book to the Civil section, delete a book from the Electrical section or view issued book details of the Mechanical department.

These are the functions used in this mini project:

* void mainmenu(void) – This function is used to display the main menu of this project. Scroll down to Output to view the main menu.
* void returnfunc(void) – Inside this function, the main menu function (mentioned above) is called when the user presses a key. With this, the user can return back to the main menu.
* void addbooks(void) – This function adds books in a file. For that, you need to mention the department to which you want to add the book. The record is saved in a file. And, it is similar for the following functions as well.
* void deletebooks(void)
* void editbooks(void)
* void searchbooks(void)
* void issuebooks(void)
* void viewbooks(void)
* void issuerecord() – With this function, you can keep record of the student to whom the book has been issued.
* void closeapplication(void) – This function is for closing the application.
* int getdata() – This function asks for data input from the user.
* int checkid(int) – This function is used to check whether the ID of a book entered by a user exists in file or not.
* void Password() – Due to this function, the user is asked to input password to run the application after it is opened. You can’t change the password upon running the application.
* void gotoxy (int x, int y) – You need to understand this function as it is one of the most important one used in mini project on Library Management System in C. This function allows you to print text in any place of screen. Using this function in Code::Blocks requires coding, but it can be directly used in Turbo C. Here is a code for this function in Code::Blocks.

COORD coord = {0, 0}; // sets coordinates to (0,0) as global variables

void gotoxy (int x, int y)

{

coord.X = x; coord.Y = y; // X and Y are the coordinates

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

delay(unsigned int mseconds) – This function delays the execution. It can be used directly in Turbo C, but requires coding in Code::Blocks. The code is given below:

void delay(unsigned int mseconds)

{

clock\_t goal = mseconds + clock();

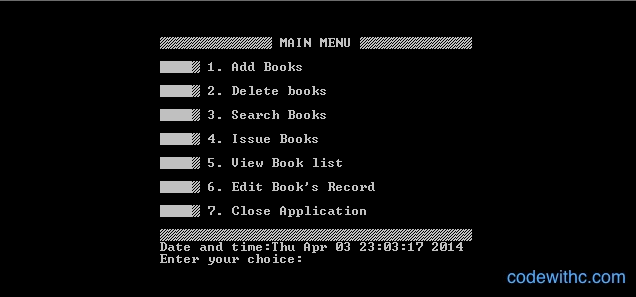
while (goal > clock());

}

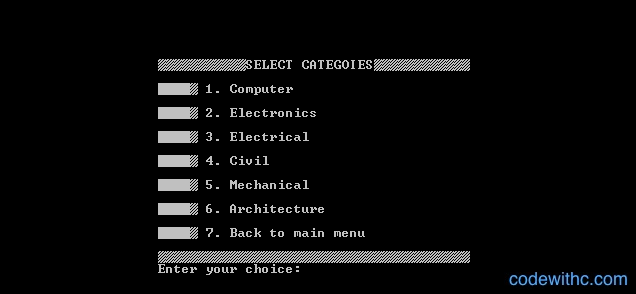
**CHAPTER 5**

**SAMPLE OUTPUT**

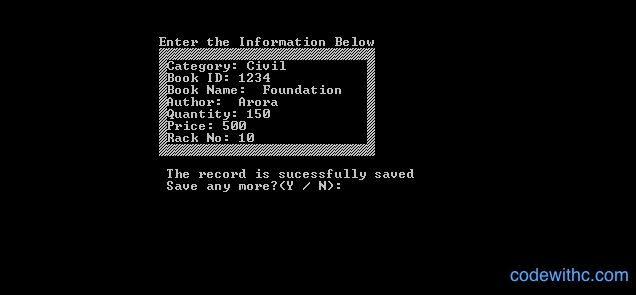
**5.1 Screenshots**



**Figure 5.1 Welcome Screen**



**Figure 5.2 Select Categories**



**Figure 5.3 Information Menu**

**CHAPTER 6**

**CONCLUSION**

To conclude the description about the project, this was developed using C++ and File structure is based on the requirement specification of the users and analysis of the existing system, with flexibility for future enhancement. Library Management System is very useful for college library department. This software is designed for people who want to keep the record of the books that are circulated to the students and it also helps the library head to keep track the new book and book bank books. This also helps in keep records of all the book, journals, e journals and database in college.

**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/