

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka – 590018



A

Mini Project Report

On

“LIBRARY MANAGEMENT SYSTEM”

Submitted By

SHUBHASHINI PAL (1KT17IS019)

Under the guidance

of

Mrs. TEJASWINI. N

Assistant Professor,

Dept. Of ISE



SRI KRISHNA INSTITUTE OF TECHNOLOGY

Department of Information Science and Engineering

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru – 560090

2019-2020

SRI KRISHNA INSTITUTE OF TECHNOLOGY

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru – 560090

Department of Information Science and Engineering



CERTIFICATE

Certified that the mini project work prescribed in 17ISL68 entitled "**LIBRARY MANAGEMENT SYSTEM**" carried out by **Shubhashini Pal ,1KT17IS019**, bonafied students of **Sri Krishna Institute of Technology**, Bengaluru in partial fulfillment for the award of **Bachelor of Engineering in Information Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2019-20. 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 with respect to project work prescribed for the said Degree.

Signature of the Guide

Mrs.Tejaswini. N

Signature of the HOD

Dr.Hemalatha K.L

EXTERNAL VIVA

Name of the Examiners

1.

2.

Signature with date

DECLARATION

I ,**Shubhashini pal**, students of VI semester in Information Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, hereby declare that the mini project entitled "**Library Management System**" has been carried out by us under the supervision of our guide **Mrs.Tejaswini N Assistant Professor, Dept. of Information Science and Engineering, Sri Krishna Institute of Technology, Bengaluru** and submitted in partial fulfillment for the award of degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during the academic year 2019-20. I, further declare that the report has not been submitted to any other University for the award of any other degree.

Place: Bengaluru

Date: **SHUBHASHINI PAL (1KT17IS019)**

ABSTRACT

An organized and systematic office solution is essential for all universities and organizations. Library Management System is software which is helpful for students as well as the university authorities. There are many departments of administration for the maintenance of college information and student file structure in any institution. All these departments provide various records regarding department library and file system. Most of these track records need to maintain information about the book logs. This information could be the general details like student name, address, date of issue, date of return, availability of books etc. or specific information related to departments like collection of data. In the current system, all the activities are done manually. So, they need to be automated and centralized as, Information from one module will be needed by other modules.

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 which are generally not available in normal library management systems 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 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 from the librarian after logging into his account admin account can generate various reports such as student report , issue report, teacher report and book report Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

ACKNOWLEDGEMENT

The completion of mini project work brings with a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

At the outset, we express our most sincere grateful acknowledgment to the holy sanctum “**Sri Krishna Institute of Technology**”, the temple of learning, for giving us an opportunity to pursue the degree course in Information Science and Engineering and thus helping us in shaping our career.

I extend our deep sense of sincere gratitude to **Dr. A. Manjunatha, Principal**, Sri Krishna Institute of Technology, Bengaluru, for providing us an opportunity to continue our higher studies.

I express our heartfelt sincere gratitude to **Dr. Hemalatha K. L., Professor and HOD, Department of Information Science and Engineering**, Sri Krishna Institute of Technology, Bengaluru, for her valuable suggestions and support.

I also thank our project guide **Mrs.Tejaswini .N, Professor, Department of Information Science and Engineering**, Sri Krishna Institute of Technology, Bengaluru, for her constant support and valuable guidance in completion of project.

I would like to thank all the teaching and non-teaching staff members in our **Department of Information Science and Engineering**, Sri Krishna Institute of Technology, Bengaluru, for their support.

Finally, we would like to thank all our friends and family members for their constant support, guidance and encouragement.

SHUBHASHINI PAL (1KT17IS019)

TABLE OF CONTENTS

Abstract	i	
Acknowledgement	ii	
Table of contents	iii	
List of figures	v	
CHAPTER NO.	CHAPTER NAME	PAGE No
Chapter 1.	INTRODUCTION	1
	1.1 Different Types of Access Method	2
	1.2 B-Tree	2
	1.3 Properties of B-Tree	2
	1.4 Search	3
	1.5 Insertion	4
	1.6 Deletion	4
Chapter 2.	SYSTEM ANALYSIS	6
	2.1 Existing System	6
	2.2 Proposed System	6
	2.3 Software requirement specification	7
	2.3.1 General description	7
	2.3.2 System objectives	8
	2.4 Advantages of proposed system	8
	2.5 Hardware requirements	9
	2.6 Software requirements	9
Chapter 3.	SYSTEM DESIGN	10
	3.1 ER Diagram	10
	3.2 Class Diagram	12
	3.3 Use Case Diagram	13
Chapter 4.	SYSTEM IMPLEMENTATION	14
	4.1 Pseudo code	14

4.2 Programming language	14
4.3 Local server	14
Chapter 5. TESTING	15
5.1 System testing	15
5.2 Unit testing	15
5.3 Integration testing	16
Chapter 6. RESULTS	17
Chapter 7. CONCLUSION AND FUTURE ENHANCEMENT	18
BIBLIOGRAPHY	
Appendix	Appendix A: Snapshots

		14
4.2 Programming language		14
4.3 Local server		15
Chapter 5.	TESTING	15
5.1 System testing		15
5.2 Unit testing		16
5.3 Integration testing		17
Chapter 6.	RESULTS	18
Chapter 7.	CONCLUSION AND FUTURE ENHANCEMENT	
	BIBLIOGRAPHY	
Appendix	Appendix A: Snapshots	

LIST OF FIGURES

FIGURE NO.	FIGURE DISCRIPTION	PAGE NO.
Fig 1.1	B-Tree	3
Fig 1.2	Deletion in B-Tree	5
Fig 3.1	ER Diagram	10
Fig 3.2	Class Diagram	12
Fig 3.3	Use Case diagram	13

CHAPTER 1

INTRODUCTION

FILE STRUCTURES

A File structure is a combination of representations for data in files and of operations for accessing the data. A File structure application allows us to read, write, and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order. An improvement in file structure design may make an application hundreds of times faster. The details of representation of data and the implementation of the operations determine the efficiency of the file structure for particular application. The fundamental operation of file systems: open, create, close, read, write, and seek. Each of these operations involves the creation or use of a link between a physical file stored on a secondary device and a logical file that represents a program's more abstract view of the same file. When the program describes an operation using the logical file name, the equivalent physical operation gets performed on the corresponding physical file. Disks are very slow compared to memory.

On the other hand, disks provide enormous capacity at much less cost than memory. They also keep the information stored on them when they are turned off.

The tension between a disk's relatively slow access time and its enormous, nonvolatile capacity is the driving force behind file structure design. Good file structure design will give us access to all the capacity without making our applications spend a lot of time waiting for the disk. A tremendous variety in the types of data and in the needs of applications makes file structure design very important. The problems that researchers struggle will reflect the same issues that one confronts in addressing any substantial file design problem. Working through the approaches to major file design issues shows one a lot about how to approach new design problems.

Goals of researched development in file structures are:

- Get the information with one access to the disk.
- Structures that allow to find the target information with as few accesses as possible.
- File structures to group information so that to get everything we need with only one trip to the disk.

1.1 Different Types Access Methods

Different types of access methods in file structure are:

- Indexing
- Co sequential processing model
- AVL trees
- B-trees
- B+ trees
- Hashing

1.2 B-TREE

B-Tree is a self-balancing search tree. In most of the other self-balancing search trees (like AVL and Red Black Trees), it is assumed that everything is in main memory. To understand use of B-Trees, we must think of huge amount of data that cannot fit in main memory. When the number of keys is high, the data is read from disk in the form of blocks. Disk access time is very high compared to main memory access time. The main idea of using B-Trees is to reduce the number of disk accesses. Most of the tree operations (search, insert, delete, max, min etc.) require $O(h)$ disk accesses where h is height of the tree. B-tree is a fat tree. Height of B-Trees is kept low by putting maximum possible keys in a B-Tree node. Generally, a B-Tree node size is kept equal to the disk block size. Since h is low for B-Tree, total disk accesses for most of the operations are reduced significantly compared to balanced Binary Search Trees like AVL Tree, Red Black Tree, etc.

1.3 Properties of B-Tree

All leaves are at same level.

- A B-Tree is defined by the term *minimum degree 't'*.
- The value of t depends upon disk block size.
- Every node except root must contain at least $t-1$ keys. Root may contain minimum 1 key.
- All nodes (including root) may contain at most $2t - 1$ keys.
- Number of children of a node is equal to the number of keys in it plus 1.
- All keys of a node are sorted in increasing order. The child between two keys k_1 and k_2 contains all keys in range from k_1 and k_2 .
- B-Tree grows and shrinks from root which is unlike Binary Search Tree. Binary Search Trees grow downward and also shrink from downward.
- Like other balanced Binary Search Trees, time complexity to search, insert , delete is $O(\log n)$.

Following is an example B-Tree of minimum degree 3. Note that in practical B-Trees, the value of minimum degree is much more than 3.

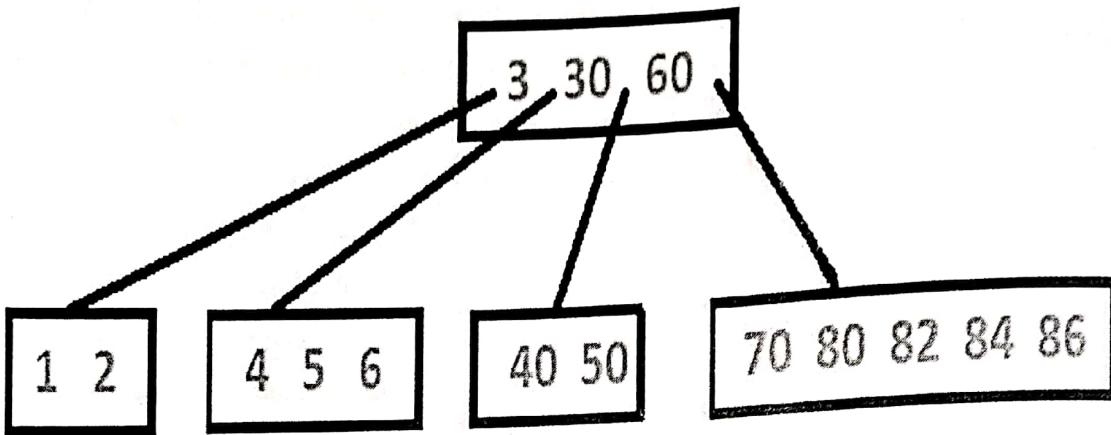


Figure 1.1:b-tree

1.4 Search

Search is similar to search in Binary Search Tree. Let the key to be searched be k . We start from root and recursively traverse down. For every visited non-leaf node, if the node has key, we simply return the node. Otherwise we recur down to the appropriate child (The child which is just before the first greater key) of the node. If we reach a leaf node and don't find k in the leaf node, we return NULL. **Traverse**

Traversal is also similar to In order traversal of Binary Tree. We start from the leftmost child, recursively print the leftmost child, then repeat the same process for remaining children and keys. In the end, recursively print the rightmost child.

1.5 Insertion

Initialize x as root

- While x is not leaf, do following
- Find the child of x that is going to be traversed next. Let the child be y .
- If y is not full, change x to point to y .
- If y is full, split it and change x to point to one of the two parts of y .
- If k is smaller than mid key in y , then set x as first part of y . Else second part of y .

1.6 Deletion

Deletion from a B-tree is more complicated than insertion, because we can delete a key from any node—not just a leaf—and when we delete a key from an internal node, we will have to rearrange the node's children. As in insertion, we must make sure the deletion doesn't violate the B-tree properties. Just as we had to ensure that a node didn't get too big due to insertion, we must ensure that a node doesn't get too small during deletion (except that the root is allowed to have fewer than the minimum number $t-1$ of keys). Just as a simple insertion algorithm might have to back up if a node on the path to where the key was to be inserted was full, a simple approach to deletion might have to back up if a node (other than the root) along the path to where the key is to be deleted has the minimum number of keys.

The deletion procedure deletes the key k from the subtree rooted at x . This procedure guarantees that whenever it calls itself recursively on a node x , the number of keys in x is at least the minimum degree t . Note that this condition requires one more key than the minimum required by the usual B-tree conditions, so that sometimes a key may have to be moved into a child node before recursion descends to that child. This strengthened condition allows us to delete a key from the tree in one downward pass without having to “back up” (with one exception, which we'll explain).

Library Management System

The steps and process of deletion in b-trees.

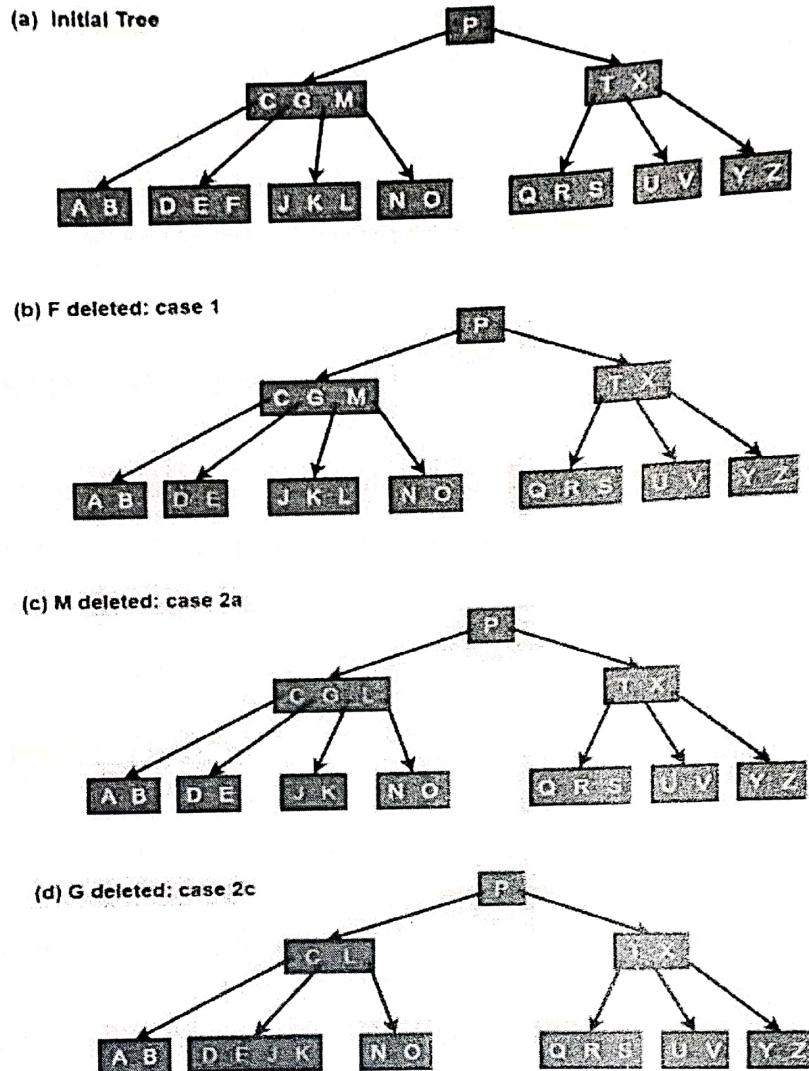


Fig 1.6: deletion of a b-tree.

CHAPTER 2

SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

2.1 EXISTING SYSTEM

System analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is what all problems exists in the present system. Analysis begins when a user or manager begins a study of the program using existing system.

2.2 PROPOSED SYSTEM

Proposed system is an automated Library Management System. Through our software user can add books, search books, renewal, update, information, edit information, and return books in quick time. Our proposed system has the password protection accessibility system.

2.3 SOFTWARE REQUIREMENT SPECIFICATION

2.3.1 GENERAL DESCRIPTION

. Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time-saving.

Library Management System

PROBLEM STATEMENT:

Space consuming the problem occurred before having computerized system includes:

- File lost

When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.

- File damage

When a computerized system is not their file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage the files.

- Difficult to search record

When there is no computerized system there is always a difficulty in searching of records if the records are large in number. After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

- Cost consuming

As there is no computerized system the to add each record paper will be needed which will increase the cost for the management of library.

2.3.2 SYSTEM OBJECTIVES

- Improvement in control and performance

The system is developed to cope up with the current issues and problems of library the system can add user, validate user and is also bug free.

- Save cost

After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.

- Save time

Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- It is very fast and clever.
- No need more user admin, always extra manual effort.
- Not easy for data loss.
- Approaching students to learn the requirements of the university.
- Short time to knowledge and learning to use operate the system.

2.5 HARDWARE REQUIREMENTS

Processor: Intel Pentium 4

RAM: 512MB

Hard Disk: 40GB

MONITOR : 15 VGA color

2.6 SOFTWARE REQUIREMENTS

Operating System: windows 10

Back End : Apache tomcat server, MS Access.

Front End : turbo C++

Tools: Turbo C++ 3.0

Technologies: DOS 7.0

CHAPTER 3

SYSTEM DESIGN

3.1 ER DIAGRAM

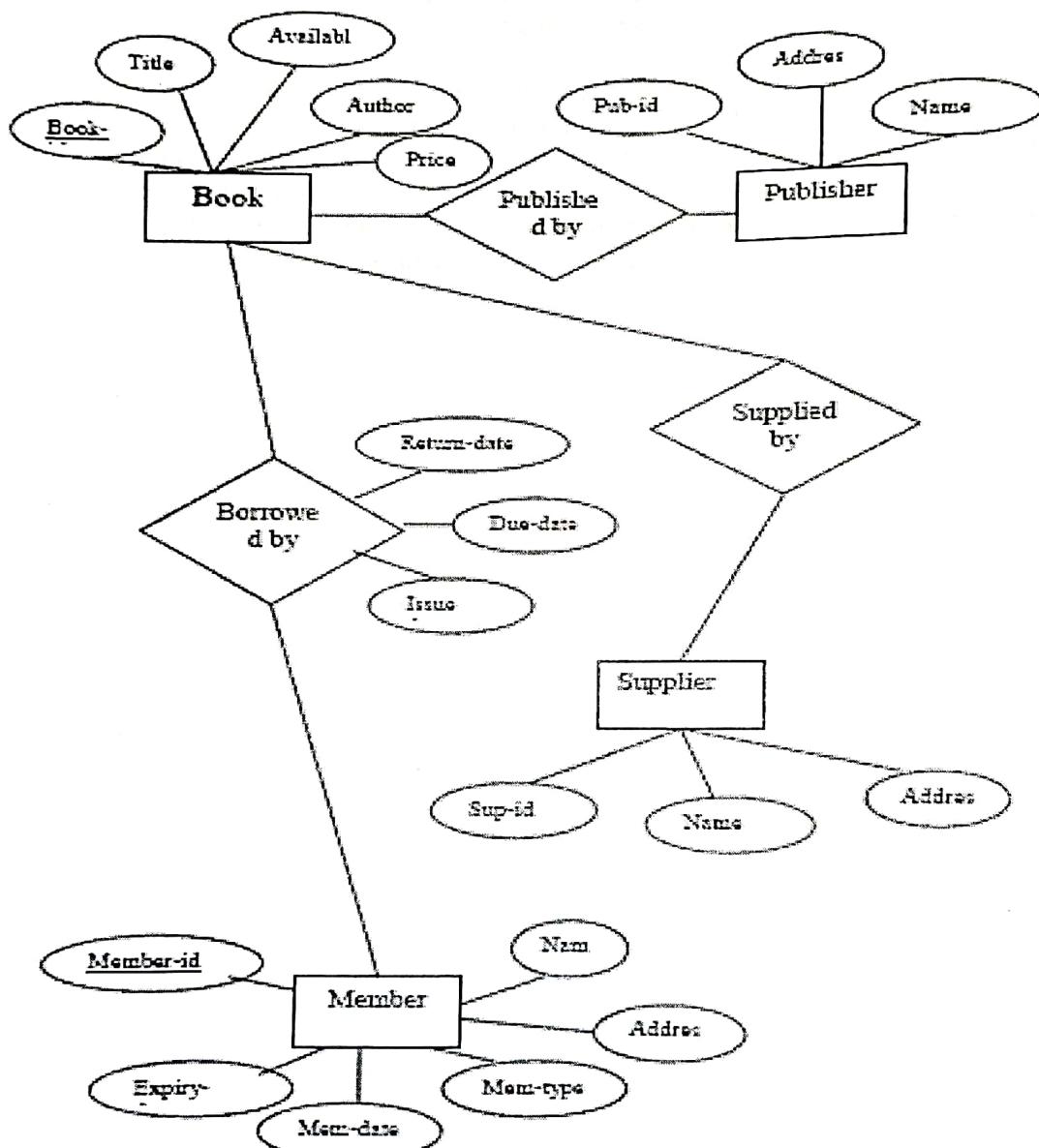


FIG 3.1 : library Management System ER Diagram

Library Management System

An **entity–relationship model** (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database.

An **entity** is a thing that exists either physically or logically. An entity may be a physical object such as a house or a car (they exist physically), an event such as a house sale or a car service, or a concept such as a customer transaction or order (they exist logically—as a concept). Although the term entity is the one most commonly used, following Chen we should really distinguish between an entity and an entity-type.

An **entity-type** is a category. An entity, strictly speaking, is an instance of a given entity type. An entity is a thing that exists either physically or logically. An entity may be a physical object such as a house or a car (they exist physically), an event such as a house sale or a car service, or a concept such as a customer transaction or order (they exist logically—as a concept). Although the term entity is the one most commonly used, following Chain we should really distinguish between an entity and an entity-type.

Library Management System

3.2 CLASS DIAGRAM

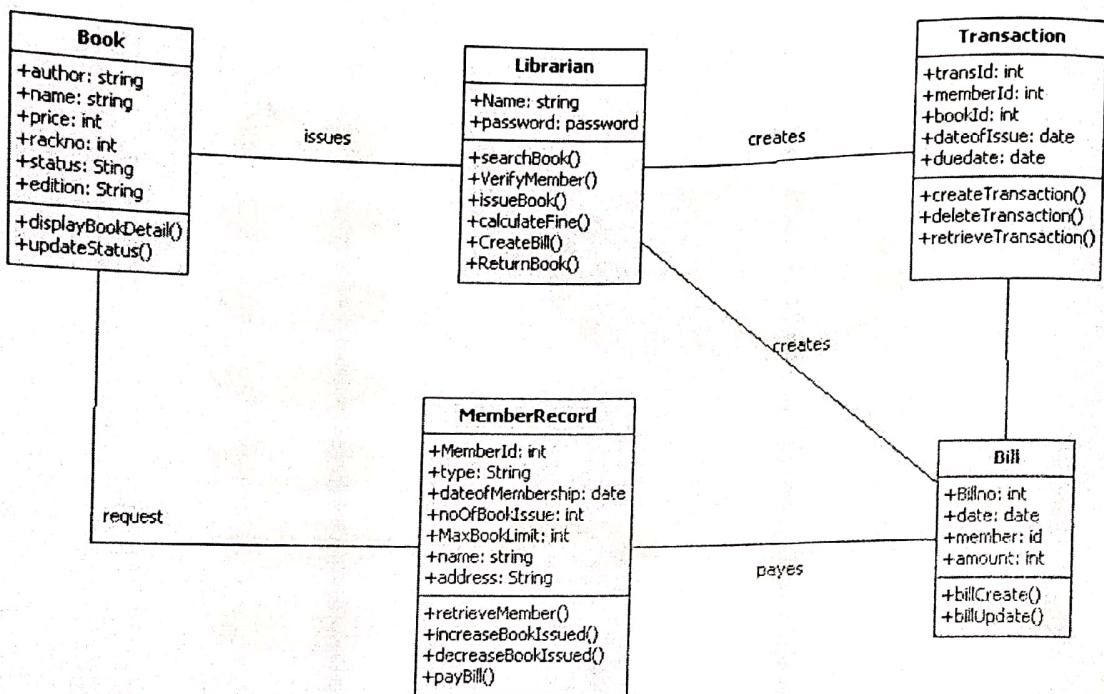


Fig 3.2: Library Management System Class Diagram.

A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Examples: an owns relationship between a company and a computer, a supervises relationship between an employee and a department, a performs relationship between an artist and a song, a proves relationship between a mathematician and a conjecture, etc. The model's linguistic aspect described above is utilized in the declarative database query language ERROL, which mimics language constructs. ERROL's semantics and implementation are based on reshaped relational algebra (RRA), a relational algebra that is adapted to the entity–relationship model and captures its linguistic aspect. A File structure is a combination of representations for data in files and of operations for accessing the data. A File structure application allows us to read, write, and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order. An improvement in file structure design may make an application hundreds of times faster.

3.3 USE CASE DIAGRAM

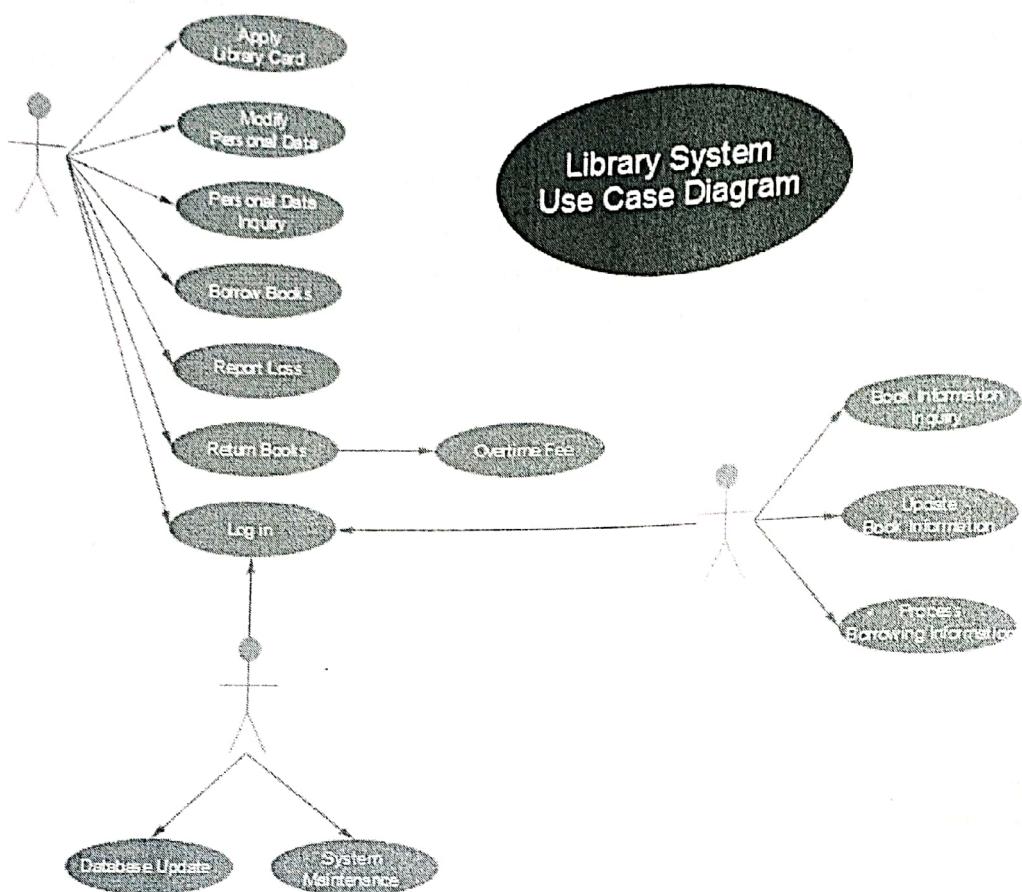


Fig 3.3: Library Management System Use Case Diagram

Entities and relationships can both have attributes. Examples: an employee entity might have a Social Security Number (SSN) attribute, while a proved relationship may have a date attribute. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's key. Entity-relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets (all entities of the same entity type) and relationship sets (all relationships of the same relationship type). Examples: a particular song is an entity; the collection of all songs in a database is an entity set; the eaten relationship between a child and her lunch is a single relationship; the set of all such child-lunch relationships in a database is a relationship set.

CHAPTER 4

IMPLEMENTATION

4.1 PSEUDO CODE

Write the pseudocode and draw a flowchart that depicts the registration process for new members. This process should cater to multiple registrations at one instance. All the relevant information needs to be supplied, and all the information being accepted need to be validated as described above. Write a pseudocode for the requisition of students that are unavailable in the database. Draw a flowchart showing the entire system of students in the database. Prepare test data for the algorithms and their results. Prepare a System Manual comprising of the problem statement, all the flowcharts and pseudocode, some test data, and their results.

4.2 Programming Language

Borland first introduced Turbo C++ in 1990. - It is a C++ compiler and integrated development environment (IDE) and computer language. - Turbo C++ provides an environment called IDE (Integrated Development Environment). - The editor is used to create the source file, compile it, link it and then execute it.

4.3 Local Server

Local sever is a server that is running in a local or a mounted folder and whose document root is not the parent of the project root. To configure access to the server in this set-up you need to specify the following:

- The server configuration root folder and the URL address to access it.
- A local server gives you exclusive access to data and object in a set of windows folders called data directions.

The localhost is the default name describing the local computer address also known as the loopback address.

CHAPTER 5

TESTING

5.1 SYSTEM TESTING

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2. Integration testing

5.2 UNIT TESTING

1. Test for the admin module

- Testing admin login form- This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.
- Student account addition- In this section the admin can verify student details from student academic info and then only add student details to main library database it contains add and delete buttons if user click add button data will be added to student database and if he clicks delete button the student data will be deleted.
- Book Addition- Admin can enter details of book and can add the details to the main book table also he can view the books requests.

2. Test for Student login module

- Test for Student login Form- This form is used for log in of Student .In this we enter the library id, username and password if all these are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for library id, username and password.
- Test for account creation- This form is used for new account creation when student does not fill the form completely it asks again to fill the whole form when he fills the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.

3. Test for teacher login module

- Test for teacher login form- This form is used for log in of teacher .In this we enter the username and password if all these are correct teacher login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.

5.3 INTEGRATION TESTING

In this type of testing, we test various integration of the project module by providing the input. The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module. Integration testing is testing in which a group of components are combined to produce output.

Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

The two types of integration testing are top-down integration testing which is testing that takes place from top to the bottom of the system. Bottom-up integration testing which is a testing that takes place from bottom the control flow upwards.

CHAPTER 6

RESULTS

The development of the project is not an easy process as it involves lot of challenges in different stages of software analysis, design, coding and testing.

Having understood the requirements properly and implementing the solutions as per the expectation as brought to the closure of the project. We have tried our best to make this project very realistic, so that the user does not face any trouble when switching over from any real-life project to this highly useful one.

CHAPTER 7

CONCLUSION AND ENHANCEMENT

7.1 CONCLUSION

Library management system needs to be computerized to reduce human errors and to increase the efficiency. The proposed Library management system in this proposal will be a computerized management system developed to maintain all the daily work in library. Library management system are designed to store all the information about books and members. The main focus of this project is to lessen human work and encourage efficient record Keeping. Though we are making an offline version and also a student viewing section creating account on this system.

7.2 FUTURE ENHANCEMENT

The future enhancement that can be done:

- Access to all the members in the institute.
- Accessing using Id card.
- SMS alerts to the users.

BIBILOGRAPHY

Websites:

- <https://en.wikipedia.org/wiki/C%2B%2B>
- <http://www.icbse.com>
- <http://www.filestructure/google/c%./chrome..>

Books:

- Michael J.Folk, Bill Zoellick, Greg Richardi, 3rd Edition, pearson education, 1998 Textbook.
- K.R Venugopal, K.G. Srinivas, P.M Krishnaraj: File structure using C++, Tata Mc Graw-Hill, 2008 textbook.
- Scot Robert Ladd: C++ Components and Algorithms, BPB publications, 1993.
- Raghu Ramakrishna and Johannes Gehrek: Database management system, 3rd Edition, Mc Graw-Hill, 2003.