

# **Project Helpline**

## **REPORT**

### **“LIBRARY MANAGEMENT SYSTEM”**

## TABLE OF CONTENTS

<b>Contents</b>	<b>Page No.</b>
<b>1. Introduction</b>	<b>6</b>
<b>1.1 Objective</b>	<b>7</b>
<b>2. Project Overview</b>	<b>10</b>
<b>3. Study &amp; Drawbacks of Existing System</b>	<b>14</b>
<b>4. System Analysis</b>	<b>16</b>
<b>5. Hardware &amp; Software Requirements</b>	<b>17</b>
<b>6. Code Efficiency</b>	<b>23</b>
<b>7. Feasibility Study</b>	<b>28</b>
<b>8. System Requirement specification</b>	<b>30</b>
<b>9. Opportunity Statement</b>	<b>31</b>
<b>10. Usage Analysis</b>	<b>32</b>
<b>11. Summary of Project</b>	<b>33</b>
<b>12. Project Plan</b>	<b>35</b>
<b>13. Data Flow Diagram</b>	<b>36</b>
<b>14. Logic Diagram</b>	<b>38</b>
<b>15. ER Diagram</b>	<b>43</b>
<b>16. Source Code</b>	<b>50</b>
<b>17. Report Generation</b>	<b>88</b>
<b>18. Description of Modules</b>	<b>94</b>
<b>19. Security Features</b>	<b>96</b>
<b>20. Optimization</b>	<b>97</b>
<b>21. Testing</b>	<b>98</b>
<b>22. Post Implementation Maintenance &amp; Review</b>	<b>101</b>
<b>23. Scope of Improvement</b>	<b>103</b>
<b>24. Conclusion</b>	<b>103</b>
<b>25. Refrences</b>	<b>104</b>

# 1. INTRODUCTION

## INTRODUCTION

The library management deals in management of records. The records include books which are required by different students accordingly. Maintaining these records manually becomes difficult thus we are providing such system a computerized backend.

In this system each member is provided with a member code and all the records are maintained in a table along with the member's code. Any information can be accessed according to member-code anytime from the table and also at the time of issuing books.

### The Automated System Have Following Benefits Over Manual System

- **Data handling:** It captures the information from different sources, presents it systematically and organizes its storage for efficient retrieval.
- **Quality control:** Paper work would totally be eliminated in the new system as failure data is directly fed into system.
- **System reliability:** System is very reliable as no skipping, missing of data is possible.
- **Maintenance:** No data mismatching is possible due to various checks incorporated in the system.
- **Accuracy:** The data provided by the system will be accurate as all Processing steps are algorithmic and computer based.
- **Centralized Storage:** The data is Processed and stored at central location.

- **Security:** The data is processed and stored using .net framework based application. Hence all the security features related to .net framework are used.

### **SAILENT FEATURES OF NEW PROJECT**

Integrated approach to the package helps in minimizing manual approach.

- User friendly package.
- Timing and retrieval efficiency.
- In-built validation checks to ensure accuracy.
- Elimination of paper work.
- Suitable help/error message for better user interface.
- Updated information.

### **1.1 OBJECTIVES**

With the growing information technology industry, automation of their system and management is desired by all kind of commercial enterprises. As the name suggest **Library Management System** will deal in the entire requirement needed for managing the activities of College Library. It will deal with the process of maintaining data about the books and many other things as well as transactions which are taking place in the library with respect to the Issue, Cataloguing, Searching and Return of the books.

Library Management System maintains the record of books in the library, issue, purchasing and return process of the books in the library. Here we are primarily concerned with management of books of library. In this project we identify the need for computer based Library Management Systems.

Hence this covers the following issues:

- Maintains data about the books of the library.
- Arranging data in logical order for easy maintenance.
- Collection of data about books which are issued and requirement for purchasing.
- To provide various search options to know the availability of books in the Library.
- Data about books which are lost.
- Generation of various reports according to the management request i.e. Cataloguing, Searching etc.

Library is a growing organism that requires constant positive changes to meet the need of its user. The invention of computer has brought in a rapid change in the society. Therefore, automation has become the need of the hour. Library automation not only improves the image of the library staff but also provides additional services to the users with the existing staff. The impact of automation on the library is quite obvious; it creates new environment where each function redefines the traditional organizational structure and transforms it into new institutional entries. In this unit a brief overview is given about library automation.

Automation is defined as a technique, a process, or a system which operates automatically. According to the Encyclopedia of Library and Information Science, “Automation is the technology concerned with a design and development of the process and systems that minimize the necessity of human intervention in their operation.

Swihart Stanley S and Hefley Beryl F have defined the term ‘library automation’ as “the processing of certain routine clerical function in the library with the assistance of

computer or other mechanized or semi automatic equipment”. It may also be defined as a process of mechanization of all the housekeeping operation of a library which is repetitive in nature. The housekeeping operation includes acquisition, cataloguing, circulation, serial control, references and administration work.

Automation is a technique to make a system automated, i.e. self active. For this the electronic machines are used to automate the libraries. By automation, libraries activities such as acquisition, circulation, serial control, information retrieval, cataloguing and indexing can be mechanized by using library software’s.

The Library Automation means:-

- Computerization of the entire house keeping operation of the library.
- Operate a computerization library management system.
- Offer new services based on the technologies and also integrate the traditional library operations in the era.

### **Scope**

This application can be used by any Library to automate the process of manually maintaining the records related to the subject of purchasing of books, selecting and ordering items for the collection and maintaining the accounts, Creating records for material held in the collection and providing access to the catalogue-Via an Online Public Access Catalogue (OPAC).

## **2. PROJECT OVERVIEW**

To develop an automated Library Management System for Aravali International School, Faridabad. As I am working over there as a teacher.

The Intelligent and Supportive system for Library deals in management of records.

Records include books required by students accordingly. Maintaining these records manually becomes difficult. Automated systems help in maintaining these records easily.

Thus a lot of time can be saved and correct information about the borrower can be accessed.

### **Goals**

To ensure the following:

- Ease of maintenance of records of books according to unique accession code.
- Ease of maintenance of records of the borrowers according to unique borrower ID
- To calculate Fine amount of the overdue books.
- To change the fine amount.
- To add more categories of the books in library.
- Searching of books according to ID, Title and Authors.
- To check no. of books issued by a borrower.
- To find no. of books due at a particular date.
- To find books which are overdue.
- Report generation.
- Ease of use.
- Ease of management.
- Ease of upgrade.
- Saving the valuable time of the library.

- All of the above while keeping the system scalable.
- To reduce paperwork.
- Reduced operational time.
- Increased accuracy and reliability.
- Increased operational efficiency.
- Viewing and maintaining stock level.
- Data security.

### **ADVANTAGES**

The **VISUAL BASIC 6.0** based circulation system has the following advantages:

- ✚ Efficient and effective charge and discharge systems.
- ✚ Able to record and access pertinent user information.
- ✚ Automatic maintenance of accurate, up-to-date circulation record.
- ✚ Efficient hold and recall functions.
- ✚ Automatic production of over dues, recalls, and holds notices and bills.
- ✚ Automatic calculation of fines.
- ✚ Able to handle course reserves.
- ✚ Member's queries on the availability of books can be answered quickly.
- ✚ 'Books detail' form will give the sorted list of books by several options like book number, author, title, subject etc.
- ✚ Certain operation like, registration (add new record). Cancellation of membership, weeding out of books by deletion, charging (issuing) and discharging (return) have restricted access in order to prevent any malpractice.
- ✚ Which book is lost, and whose cards are lost and fine calculation etc. information can be known quickly from the transaction form
- ✚ Searching becomes easy.

## **METHODOLOGY**

To develop the Library Management System Software, There we are decided the three Different layers-Presentation Layer, Logical Layer and Database Layer. In Presentation Layer to design the interface of the software. In Logical Layer to decide and write the program for to performing the library task execution under the decide module. In Database layer to analysis and design the database of the Module.

- Carry out web-based surveys of ILS curricula, as they relate to coverage of LMSs and related topics, and of ILS professionals, to determine their workplace needs for education in this specific area
- Invite selected respondents from to test the teaching materials being developed for the range of hosted LMSs
- Carry out a literature review of both the developing demands on LMSs in terms of the search facilities they offer and on practical methods of teaching basic interface design, and develop new teaching materials, using as testers people selected from.
- Evaluate, by means of post-test questionnaires for developers and course participants.

In this to collect the information for the exits Library System. To study about the introduction, Problem of the system. To analyze the aim of the project. To collects the actual information about the library from the original record of the organization. To decide the step of the solution of analyze problem using the decided methodology and

technology of the project. To design the interface of the project using the design interface software (Visual basic 6.0). Designed interface is user-friendly.

**Features of Library Management System:**

- Only basic knowledge of computers is required for operation of Library Management System. As it has user-friendly application interface.
- Library Management System is Customizable and User Configurable.
- An inbuilt settings module makes Library Management System flexibility to cater to diverse organizational needs

### **3. STUDY & DRAWBACKS OF EXISTING SYSTEM**

On studying the existing system and holding student interviews, it was found that a normal system records failures. The complexities involved in the existing system as studied by us are as follows:

➤ **Timing and retrieving problem:**

Retrieval of information is time consuming, because of large volume of data. Thus if it is required to trace a particular record, a large number of records need to be scanned.

➤ **Redundancy:**

In existing system the data is maintained on lib cards. This often leads to considerable redundancy in the stored data, thus leading to wastage of storage space.

➤ **Problem of updating:**

In the existing system since every thing is stored on a card, it is quite difficult and time consuming to update the card once a week.

➤ **Material wastage:**

Too many lib cards are wasted in the manual system, daily many efforts and sources are required to give the information.

➤ **Non-reliability:**

The system is not reliable, as chances of skipping of any card from hundreds of cards are very large. As sometimes information maintained

through cards may be lost, mishandling of cards may also occur sometimes.

➤ **Maintenance of data:**

Maintenance of data is very difficult. As daily many books are to be issued, so it takes time to search and trace the lib cards according to one's requirements when required.

**DRAWBACKS OF EXISTING SYSTEM**

After studying the existing system we come across certain drawbacks, which are discussed below:-

- 1) The records stored manually can be altered by unauthorized person
- 2) Searching any information is difficult, since all the cards are to be checked.
- 3) Generation of fine on late return is difficult to calculate.
- 4) Problem arises if any of the records get misplaced.
- 5) Overload on staff members.

## **4. SYSTEM ANALYSIS**

- **Organizational:** “Library Management” is organizational in nature. The Core component in “LMS” is User module, which is based upon the various other modules.
- **Interaction:** Each component of the system interacts with the one or other component of the system. Like, generation of various report formats interacts with the Subscribers Data.
- **Interdependence:** Components of the “Library Management” are linked together in a planned way to achieve the objective.
- All the modules of the system are developed using modular and object oriented approach, so that the further improvement or modification of the modules is not a big problem.

### **Elements of System Analysis**

There are four basic elements used in “Library Management” considering system analysis. Brief description of each has been given below:

- **Outputs:** Our objective is to achieve the complete automation of “Library management” first goal is to completely automate the
- Training schedule of training of students i.e. registration of the students, assigning projects to students.

- **Inputs:** Right now the inputs of “LMS” are student’s details, area details and daily updates. Utmost care is taken in the form of validations while user is inputting the data so that incorrect or improper information cannot be able to enter in the system.

## **5. HARDWARE & SOFTWARE REQUIREMENTS**

### **Tools/Platform**

### **S/W and H/W requirements**

#### **Hardware Requirements**

Processor : Pentium IV 2GHz and Above

RAM : 2GB RAM

Monitor : 15” Color Monitor

Keyboard

Mouse

#### **Software Requirements**

Operating system : Windows XP /or upgrade

Developing Tool : Visual Basic 6.0

Database : MS Access

## **FRONT END**

Visual programming tools are complete programming environments. It allows programmers to build a GUI program using the various on-screen controls such as buttons, text, menus, boxes etc. These controls are placed on a form and then the processing details related with each control are filled in.

In the business world, competitive strategies have become the order of the day to improve quality, cut costs and provide a high response customer service base. Most organizations today need to be market driven and do a lot of value addition to their products and services. This naturally calls for rational decision making, which requires information. Information Technology or IT provides that effective channel to support and implement this strategy. Client/Server is the technology that empowers the desktop, thus setting a trend for the way successful organizations will use technology in the next decade.

### **Visual Basic (front end)**

Information that is readily available and easily accessible is the key to achieving a globally competitive spirit. Client/Server computing is what will make this happen. The Visual Basic 6.0 is one of the few Programming Languages that takes us on a tour of exactly what a Client/Server systems is all about, its benefits, specialized features and how the technology can be used to its full potential.

Visual Basic VB in short, VB has been developed by Microsoft Corporation, the makers of the widely used Operating Systems (Windows 95 / 98 / NT XP). It helps to create user friendly applications. Microsoft's Visual Basic is a powerful programming language for

Client/Server development. Microsoft's Visual Basic Release 6.0 takes development to an unprecedented level of scalability and productivity, allowing application developers to create highly functional, highly scaleable, enterprise-wide applications.

Visual Basic incorporates an integrated set of builders for creating forms, reports and database objects. Visual Basic sets new standards for ease-of-use and productivity in the development of Client/Server and web applications through the use of the following:

### **Event Driven**

Event Driven means, the user generates a stream of events each time he/she clicks with the mouse or presses a key on the keyboard.

### **Object Orientation**

This means that you can compartmentalize different aspects of your application as objects and develop and test those objects independently of the application.

### **Windows Application Development System**

Visual Basic application will look and behave like other Windows program users might work with.

The Visual Basic interface consists of a powerful and easy-to-use combination of object browsers, tabbed dialog boxes, property palettes and a WYSIWYG (*what you see is what you get*) GUI interface.

Extending Visual Basic applications with client and server-based logic is easy using the ActiveX Data Object (ADO). The ADO is a fast, high-level interface into the OLE DB application-programming interface. ADO's benefits include next-generation, data access specification with the ease of an interface.

## ACCESS

**Access** is the database software in the Microsoft Office Suite. It allows you to order, manage, and search large amounts of information.

### Opening and closing Access.

Lets look at the two basic ways of **initiating** Access.

● **From the Start button** , normally situated at the bottom left corner of the screen. Situate the mouse over the **Start** button, click, and a menu will unfold. On situating the pointer over **Programs**, a list of all the programs installed on your computer will appear; look at **Microsoft Office**, then **Microsoft Access**, click, and the program will initiate.

● **From the Access2003 button on your desktop** 

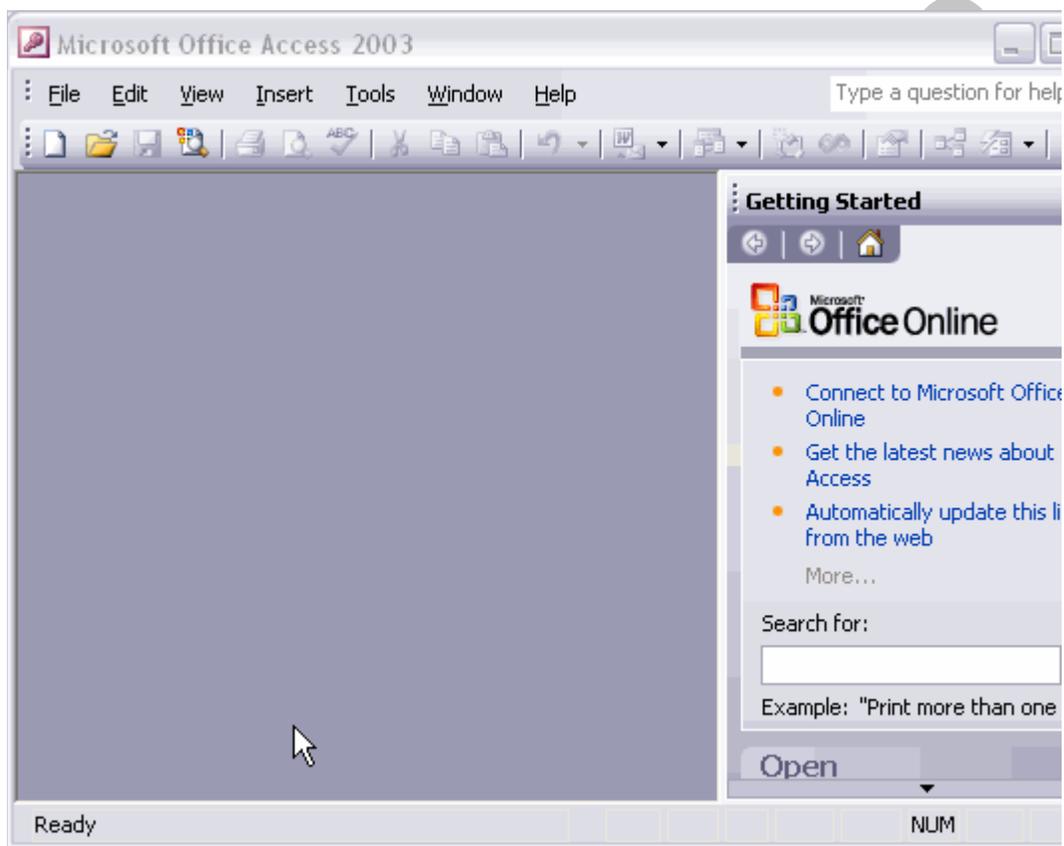
You can now initiate Access2003 to try everything we explain to you.

To **close** Access2003, you can use any of the following methods:

- Click on the **Close** button 
- depress the keys **ALT+F4** .
- Click on the **File** menu and then choose **Exit** option.

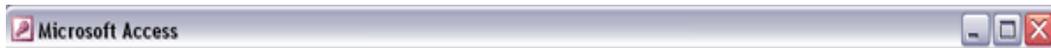
The Inicial screen

On initiating Access2003, an **opening window** will appear (seen below), we will now look at the basic components. This way we will get to know the **names of the different elements** and it will be easier for us to understand the rest of the course. The next screen we will show you (and generally all of those seen on this course) might not coincide exactly with what you will see on your own computer screen as every user can decide which elements to see at any particular time, as we will see further on.



The bars.

● **The title bar.**



**The title bar contains the name of the program.** On the extreme right are the buttons to minimize, maximize/restore and close.

● **The menu bar.**



**The menu bar** contains all **Access2003 commands**, grouped in drop down menus. By clicking on **Insert** for example, we will see the related operations with the different elements that can be inserted.

All operations can be executed from these menus. But the more regular things are executed more quickly from the tool bar that we will see in the next point.

Every option has an underlined letter, this means that we can access the option directly by pressing simultaneously the **Alt** key and the underlined letter, for example, **Alt+F** opens **File** option.

In Access2003 the menu bar has an "intelligent" behaviour, which basically consists of showing the most important and most options used by the user. If you want more information on "intelligent" behaviour" of the drop down menus clic here 

## **6. CODE EFFICIENCY**

For Code efficiency, the following conditions must be fulfilled:

### **1) Option Explicit:**

All variables must be declared with proper data type. Option Explicit keyword must be used, it forces each variable or expression in an application to be defined with a specific data type through the use of the **Dim, Private, Public, Redim or Static keywords**. If **Option Explicit** is not used, undefined variables are automatically defined as **Variant**. **Variant data type** must be avoided because it takes 16 bytes of memory.

### **2) Early Binding Vs Late Binding:**

**Early binding is faster than Late Binding** because the application doesn't have to interrogate the object at runtime to determine the objects properties and methods. In Late Binding the objects are defined as objects. Late binding is slower than Early Binding because the application must interrogate the object to determine its properties and methods.

### **3) Reclaiming Space:**

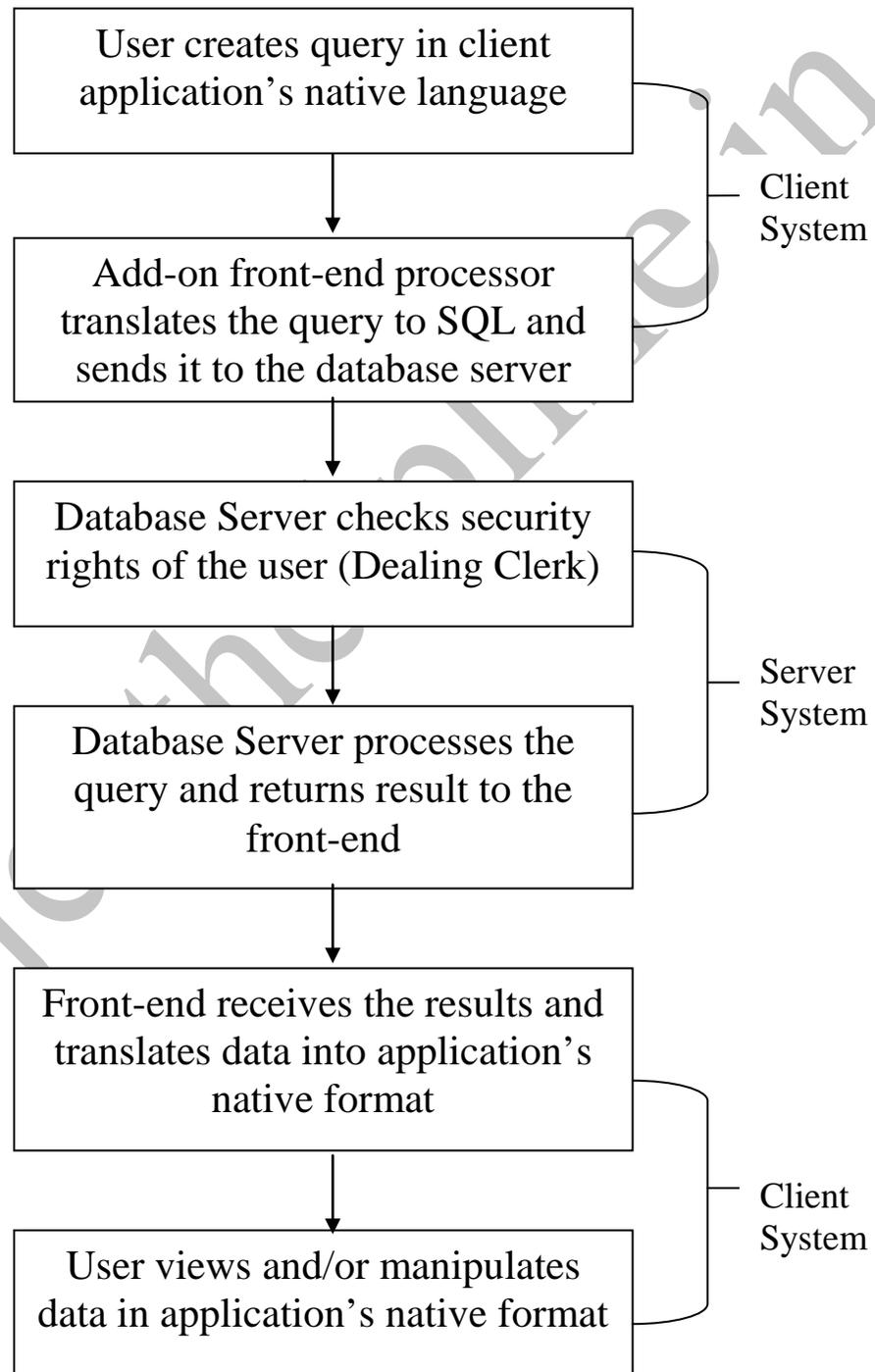
Reclaim space from strings by setting a **string variable to the empty string (“”)** space can also be reclaim from object by setting an **object variable to Nothing**. It frees the space associated with the object if it is the last reference to the object. It also has the advantage of reducing the amount of cleanup work needed when the program terminates.

#### **4) Control Array:**

Complex forms needs more memory. The more controls put on a form, the more memory it will require when it becomes loaded in to memory. Therefore, reducing the number of controls reduces the memory requirements. For this a control array must be used. A control array counts as only one name. Control arrays consumes fewer resources than the equivalent number of independent controls.

Projecthelpline.in

## How Front-End in the system Work



## **ENVIRONMENT / OPERATING SYSTEM**

The age of standalone computing is long gone and it is now a world of networks. In this context, it is imperative for users to understand issues involved in working in a multi-user environment. Windows NT today is the most sought after Multi-user Operating System in the world. This is a 32-bit, powerful, user friendly and robust network operating system from Microsoft.

Windows NT is a big step up from Windows 95 both in terms of price and in terms of capabilities. NT Server is a well-built, fully functional, multi-user, multitasking operating system.

The release of NT Server 4 adds the popular Windows 95 user interface, the Internet Information Server (IIS) – capable of running Web, FTP, and Gopher services - a search engine called Index Server, the FrontPage HTML authoring package, and the Internet Explorer, all of which make Windows NT server a tough act to beat as a robust, capable web server.

The suggested Operating System for the server computer is Windows NT Server and for client computer is Windows 95.

### **Windows 95 (OS for Clients – at branch offices)**

It has a new and improved graphical user interface, which is “document-centric” and not “program-centric”.

Many features have been added to Windows 95 to make the system easier to use.

DOS and Windows are merged into a single program. One can run DOS commands from the Windows environment itself.

The “Plug and Play” feature automates the configuration process. This makes Windows easy to identifying the kind of printer, modem, mouse or other hardware being used.

- It is fully compatible with the existing Windows and DOS programs.
- It provides pre-emptive multitasking.
- It provides free Microsoft Exchange client software. One can use Exchange as the platform for developing GroupWare client/server application.

### **Windows NT Server (OS for Server – for central database)**

It is a part of Microsoft BackOffice. It is a suite of products designed for client/server computing. Some of the features of Windows NT server are as follows: -

- The standard Intel version requires a 486 or Pentium processor Versions of Windows NT are also available for several RISC systems.
- The Windows NT Server supports symmetric multiprocessing, up to 32 processors.
- It also provides global directory services similar to NetWare 4.1.

Windows NT Server is the option to choose if you want a Microsoft client/server solution. Such a solution will involve the following:

- Windows 95 on client computers.
- Windows NT server as the Network Operating System (NOS).
- Microsoft SQL Server as the database Server.

## **7. FEASIBILITY STUDY**

It is a test of a proposed system according to its workability, impact on the organization. It is a test whether the system meet user needs and using the resources properly.

Depending on the results of initial investigation, We came to know that the Library administration wants to manage large amount of records accurately, precisely and quickly with security and integrity. The administration wants a system that can provide for the management of the currently active records anywhere and anytime. To accomplish this task, we need a Relational Database Management System (RDBMS) and a Web based GUI application development environment.

We did three types of feasibility study :

- ✓ Technical feasibility
- ✓ Operational feasibility
- ✓ Economic feasibility

### **Technical feasibility:**

In this feasibility we have to find out weather all the technical resources are available or not for the development of the system. In the College Campus all types of technical facility were available. All the software and hardware were available for the development viewpoint.

There are a number of technical issues which are generally raised during the feasibility stage of the investigation. They are as follows:

- Does the necessary technology exist to do what is suggested and can it be acquired.
- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

**Operational feasibility:**

Here we have to find out whether our system is fulfilling the user requirements or not. For this, discussions were done with the person incharge of library. Proposed projects are beneficial only if they can be turned into information systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed. Some of the important questions that are useful to test the operational feasibility of a project are given below:

- Is there sufficient support for the project from the management? From users?
- Are current business methods acceptable to the users?
- Have the users been involved in the planning and development?
- Will the proposed system cause harm? Will it produce poorer result in any case or area?

**Economic feasibility:**

It is seen whether the expenditure incurred for developing the new system will be cost effective or not.

This basically involves the top level management of the company who are the decision maker. Following decision is made during Economical feasibility:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.

## **8. SYSTEM REQUIREMENT SPECIFICATIONS**

Our school library which is concerned with the management of various types of books for all the students of the college .Currently, all work is done manually which includes -:

- Maintaining of library cards to issue books.
- Generation of fine on late return manually.
- Keep track of no of copies left after issuing the book to large no of Students.

The main objective of our solution is to design an integrated Automated system so as to reduce the paperwork and high mentally demanding work of the Librarian by creating an application which does all the manual work automatically for the Library .

Secondly it also aims at meeting the user`s requirement by allowing him to perform some operations like ADD ,DELETE , VIEW , EDIT etc.

## **9. OPPORTUNITY STATEMENT**

In the today's world of Information Technology, all the work such as obtaining user requirements, preparation of budget estimation, preparation of records, keeping track of the books that are delivered and issued, Detailed reports about the students and also the detailed status report of each book that is being managed by library were all done manually which is a very tedious process.

With the introduction of our solution which is completely backed by information Technology, all the above work that was currently being handled manually will be handled automatically through an automated system, enabling LIBRARY Employees to access information anywhere. Secondly it will also reduce the paperwork of the employees which will in turn reduce their expenditures.

Also the scope for errors and mistakes will be reduced as all the work will be done automatically by the machine.

## **10. USAGE ANALYSIS**

### **User Profiles**

The Administrator is that person who'll be involved in the maintenance and updating of the database in the Library Management System. This person will have all the rights for performing operations like addition, updation and deletion. The Administrator will be trained with the working and details of the application.

The Administrator will have his userid and password for access to the application. Thus supporting AUTHENTICITY ie the assurance that the communicating entity is the one that it claims to be. It also helps in the prevention of unauthorized use of a resource.

Therefore one login provided for the administrator having complete access to all the functions of the project.

## **11. SUMMARY OF PROJECT**

**LIBRARY MANAGEMENT SYSTEM** is one of the crucial assets of the organization.

So up keeping and up gradation of their skills is one of the major operations to be carried out in any organization. To focus on this aspect, some organizations have a separate department which undertakes library development activity. Department of Information Technology has various departments and its tedious job to provide books and skill up gradation of existing students is solved when an automated system has to in-placed instead of our existing manual system. To cope up this the department has manual system which keeps track of the regular issuing of books being arranged for different categories of B.Tech students and employees But this manual system too have some limitations such as

- Lack in security
- Difficulty in handling of various types of information
- Lack of automated management capability

To overcome these problems we propose an automated system which will automate a process of development of human resource in the form of a software package which will named as **LIBRARY MANAGEMENT SYSTEM** .

## **PROFILE OF ASSIGNED PROBLEM**

To analyze a system it is necessary to get information about that System and also see the drawbacks of that system .We have made a project on Library Management. There is a manual database that means all the information is stored in the cards .So there are many difficulties.

.They`ve shared the information about the students in the cards. If we want to find the information about the students, then it is very difficulty to find the records. It is also difficult to remove any record and put it into registers known as dead registers. There is also wastage of time .We have also got there are many difficulties. They make a report about the candidates, which is very difficult, so there are many drawbacks in that system it is also noted that in a manual database system there are many difficulties.

To overcome these types of problems, we have made a project; this project does all the work easily. We go through the information about the students easily. We can also find out the candidates according to requirement on the basis of category , qualification etc very easily ,all which is difficult to do manually. By doing this task using Computer, manpower and time is save. So with this Computerization we can make reports as we need them for e.g. we can have College wise report, Alphabetic report of names of Students, and we can even search as per the ID provided to each candidate and with which much easier task.

## **12. PROJECT PLAN**

### **TEAM STRUCTURE**

We are team of two persons equally handling the project problems.

### **DEVELOPMENT SCHEDULE**

Project development schedule is divided into three parts.

- Identifying the problem
- System analysis and database design
- Testing and implementation

### **PROGRAMMING LANGUAGES AND DEVELOPMENT TOOLS**

**Front End:** VB

**Back End:** Access

## 13. DATA FLOW DIAGRAM

Larry Constantine, as a way of expressing, first developed the DFD system requirements in a graphical form; this led to a modular design. A *DFD* is also known as a “*bubble chart*”. A DFD has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined lines. The bubbles represent data transformations and the lines represent data flows in the system.

### **DFD SYMBOLS**

In the DFD, there are four symbols,

- 1) A *square* defines a source (originator) or destination of system data.
- 2) An *arrow* identifies data flow -- data in motion. It is pipeline through which information flows.
- 3) A *circle* or a “*bubble*” (or an oval bubble) represents a process that transforms incoming data flow(s) into outgoing data flow(s).
- 4) An *open rectangle* is a data store-data at rest, or temporary repository of data.

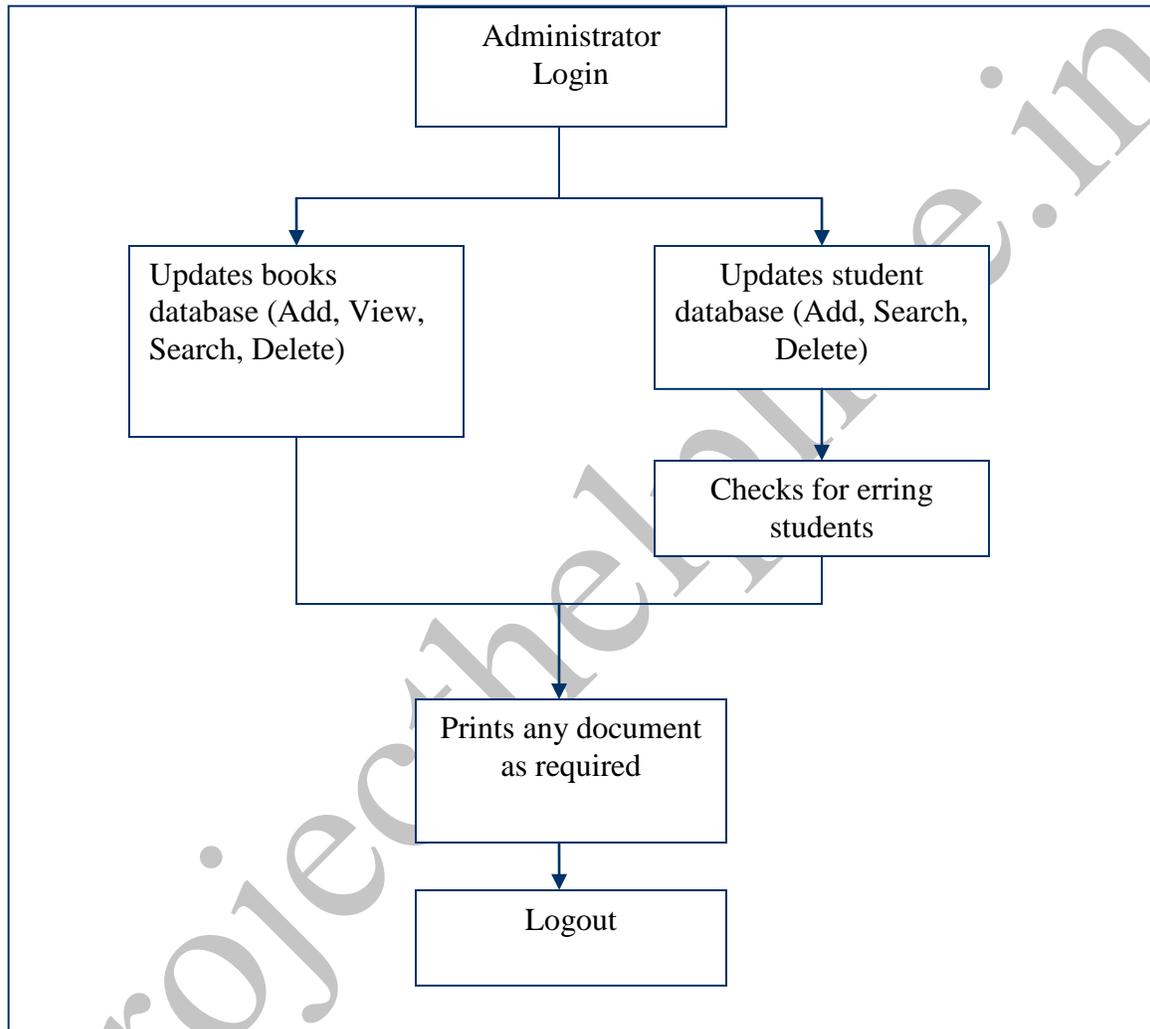
A DFD describes *what* data flows i.e., logical data flow rather than *how* they are processed, and so it does not depend on hardware, software, data structures, or file organization.

## **CONSTRUCTING A DFD**

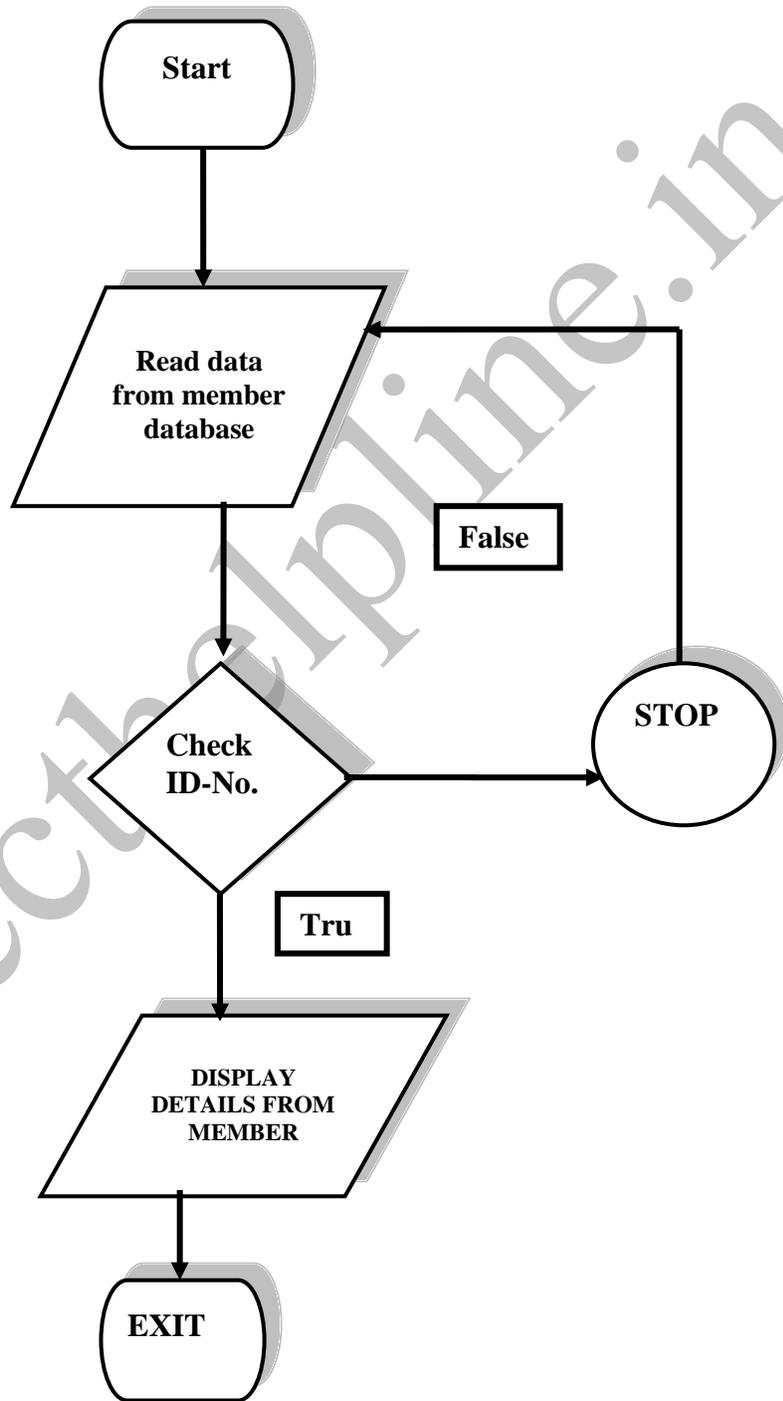
Several rules of thumb are used in drawing DFD:

- Processes should be named and numbered for easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from source (upper left corner) to the destination (lower right corner), although they may flow back to a source. One way to indicate this is to draw along flow back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in a DFD, it is marked with a short diagonal in the lower right corner.
- When a process is exploded into lower-level details, they are numbered.
- The names of data stores, sources, and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

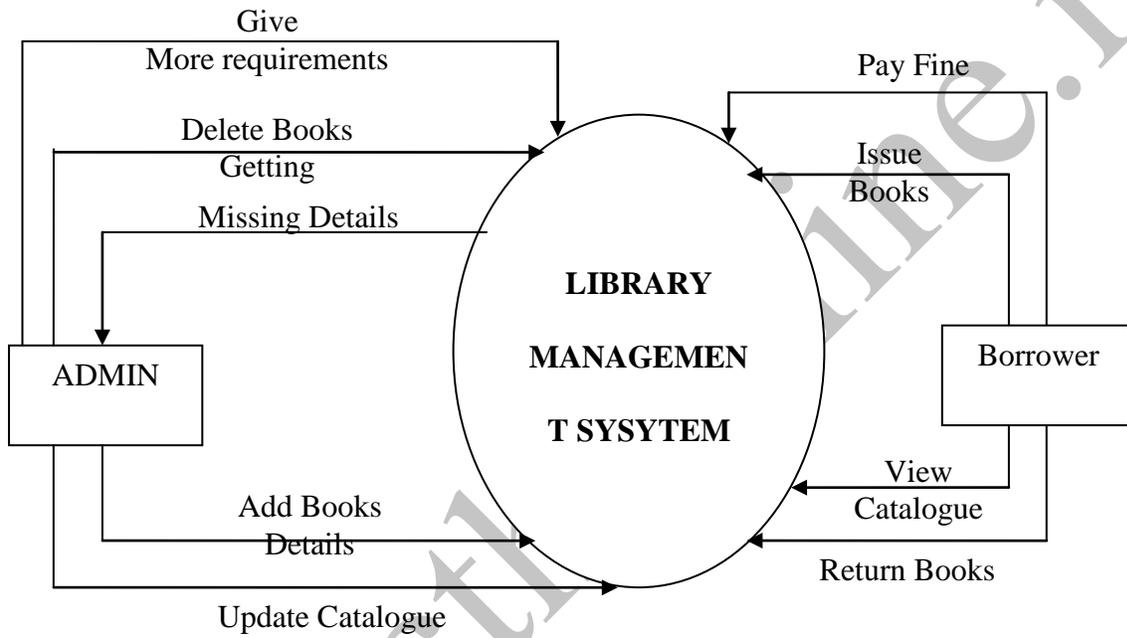
## 14. LOGIC DIAGRAM



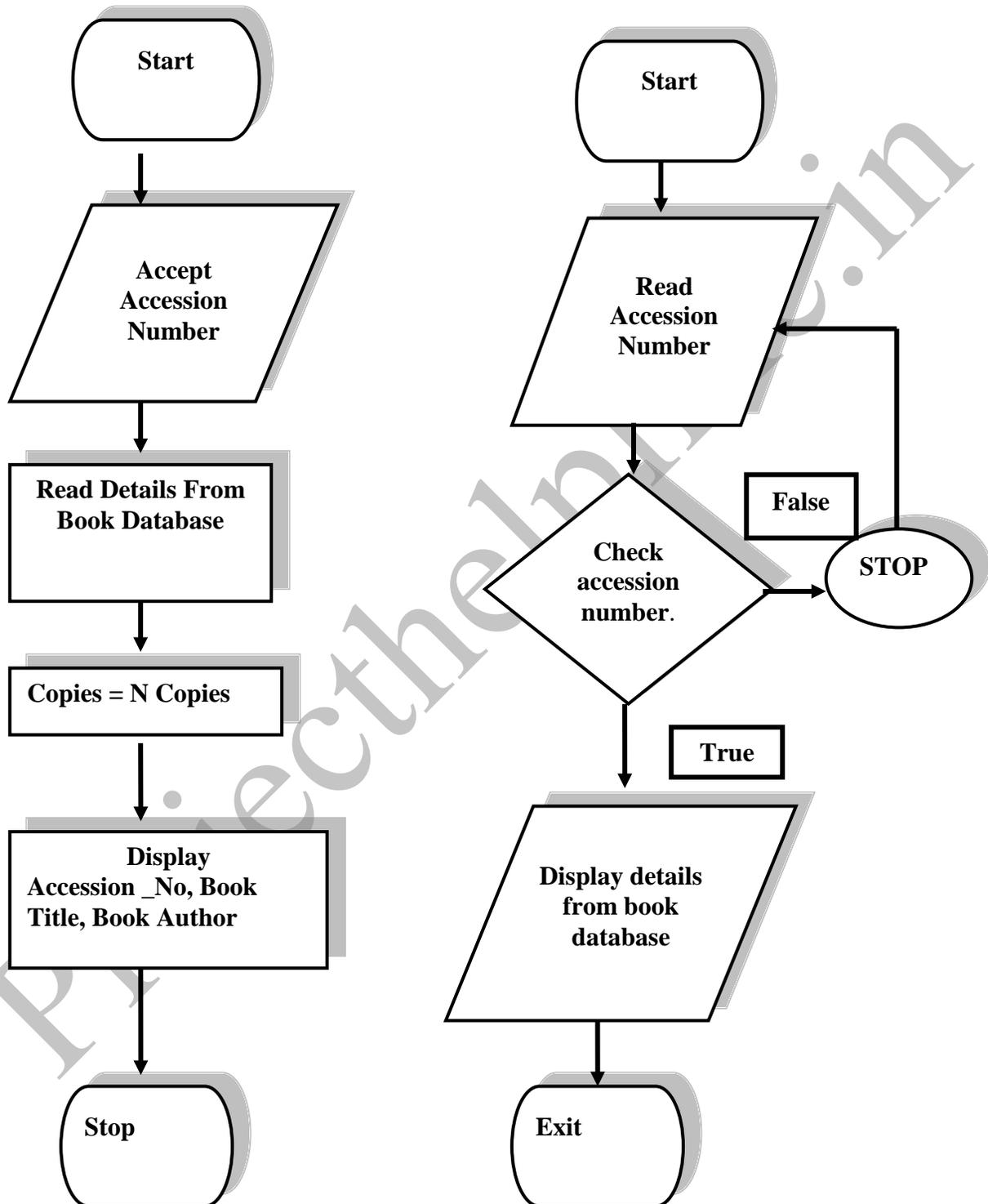
**FLOW CHART FOR**  
**MEMBER CREATION FORM LIBRARY MANAGEMENT SYSTEM**



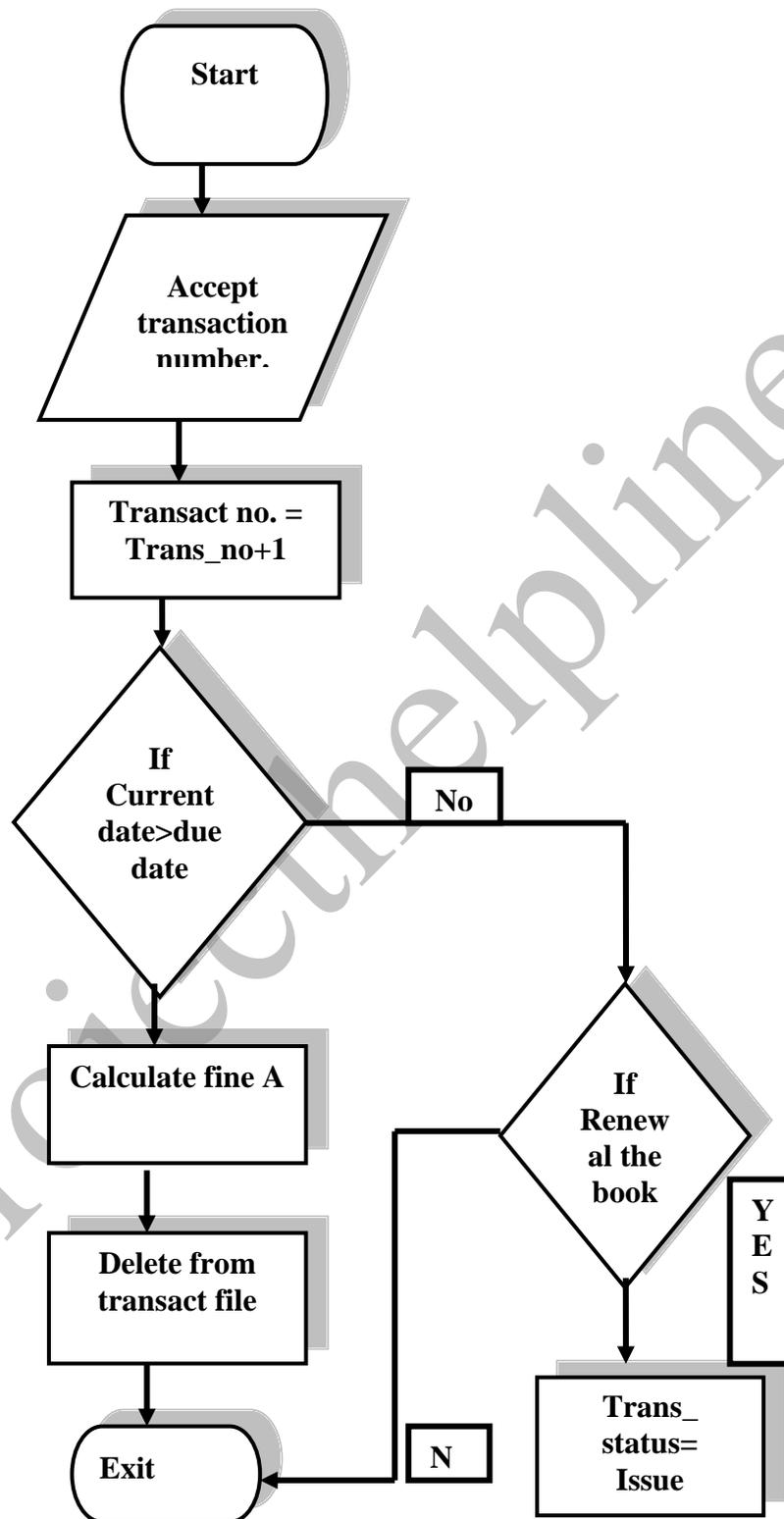
**FLOW CHART**  
**O – LEVEL**  
**(CONTENT LEVEL)**



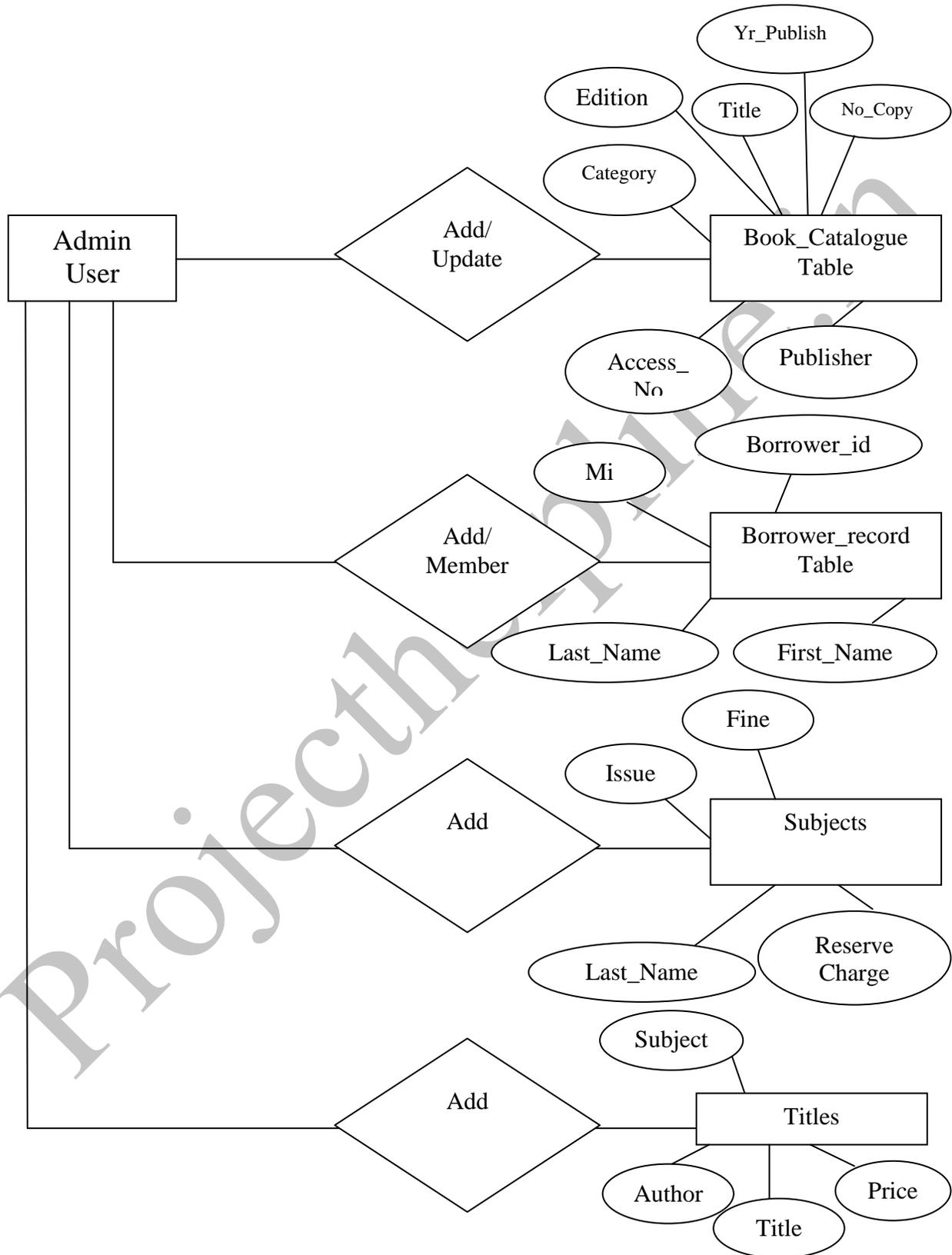
## FLOW CHART FOR BOOK ENTRY PROGRAM



## FLOW CHART FOR TRANSACTION



## 15.ER – DIAGRAM



## **FUNCTIONS**

Circulation is one of the most visible jobs in the library and involves three sections of the library and involves three sections of the library.

- ✚ The circulation desk or point of charging out library materials.
- ✚ The book stacks (closed or open) where the library materials are housed.
- ✚ A circulation records section where clerical routines, such as filling, compiling statistical reports, preparing over dues, and computing fines, are carried out.

**The following functionalities are available with in circulation module**

- ✚ Handles circulation activities such as:
  - Issue/check outs;
  - Returns/checks ins;
  - Reservations;
  - Renewals;
  - Recall process;&
  - Report loss of item process Reports;
- ✚ Manages library materials- circulation type, location and status: patron database patron type, profiles privileges: and other transactions such as computation and payment of overdue fines, lost books, etc.
- ✚ Information about the member.
- ✚ Efficient and effective charge and discharge function.
- ✚ Able to record and access pertinent user information.

- ✚ Automatic maintenance of accurate, up-to-date circulation records.
- ✚ Efficient hold and recall functions.
- ✚ Automatic production of over dues, recall, and hold notices and bills.
- ✚ Automatic calculation of fines.
- ✚ Flexibility in handling increases in collection size, number of users, and number of transaction.

**Modules to cover the general library functions of:**

- Acquisitions–Selecting and ordering items for the collection and maintaining the accounts.
- Cataloguing–Creating records for material held in the collection.
- Providing access to the catalogue-Via an Online Public Access Catalogue (OPAC). This is an Online Computerized Catalogue through which library materials are processed and retrieved.

**Logic Model**

Logic models are narrative or graphical depictions of processes in real life that communicate the underlying assumptions upon which an activity is expected to lead to a specific result. Logic models illustrate a sequence of cause-and-effect relationships-a systems approach to communicate the path toward a desired result.

### **Inputs**

1. Library Management System for maintained the record of books to must have input data from the end user.
2. To enter the record of new books.
3. Prepare the cataloguing as input data is requirement of activity of College Library.
4. To provide various search options to know the availability of books in the Library.
5. To search the data in database.
6. To enter the cost, discount of the relative book.

### **Process Logic**

In Library Management System has Perform the various type of processing. These are following:

1. Selecting and ordering items for the collection and maintaining the accounts.
2. Creating records for material held in the collection.
3. Search the registered books and members-Via an Online Public Access Catalogue (OPAC).
4. Calculate the actual cost of book.
5. Report of available books.
6. To view the requirements of library.

### **Book Catalog Table**

<b>Field Name</b>	<b>Type</b>	<b>Constrains</b>
Access_no	Text	Not Null
Category	Text	Primary Key
Title	Text	Not Null
Edition	Text	Not Null
Author	Text	Not Null
Publisher	Text	Not Null
Yr_publish	Text	Not Null
No_copy	Number	Not Null
Available_copy	Number	Not Null
Borrow_copy	Number	Not Null

### **Borrower record Table**

<b>Field Name</b>	<b>Type</b>	<b>Constrains</b>
borrower_id	Text	Primary Key
First Name	Text	Not Null
Last Name	Text	Not Null
mi	Text	Not Null
status	Text	Not Null
course	Text	Not Null
Add	Text	Not Null
contact	Text	Not Null

### **Fine Table**

<b>Field Name</b>	<b>Type</b>	<b>Constrains</b>
Member ID	Text	Not Null
Fine Out	Number	Number
Pay Date	Date/Time	Not Null

### **Global Variable Table**

<b>Field Name</b>	<b>Type</b>	<b>Constrains</b>
Total Issue Books	Number	Not Null
Renewal Counters	Number	Not Null
Max Fine Ball	Number	Not Null
Membership Duration	Number	Not Null
Membership Fee	Number	Not Null
Renewal Fee	Number	Not Null

### **Current Borrow Table**

<b>Field Name</b>	<b>Type</b>	<b>Constrains</b>
db_no	Text	Primary Key
Access_no	Text	Not Null
Category	Text	Not Null
Title	Text	Not Null
Edition	Text	Not Null
Author	Text	Not Null
Publisher	Text	Not Null
Yr_publish	Date/Time	Not Null
Borrow_date	Date/Time	Not Null
Due_date	Text	Not Null
Borrower_id	Text	Not Null

### Requirements Table

Field Name	Type	Constrains
ID	Text	Primary Ket
TITLE	Number	Not Null
AUTHOR	Number	Not Null
PUBLISHER	Number	Not Null
NOOFBOOKS	Number	Not Null

### User Table

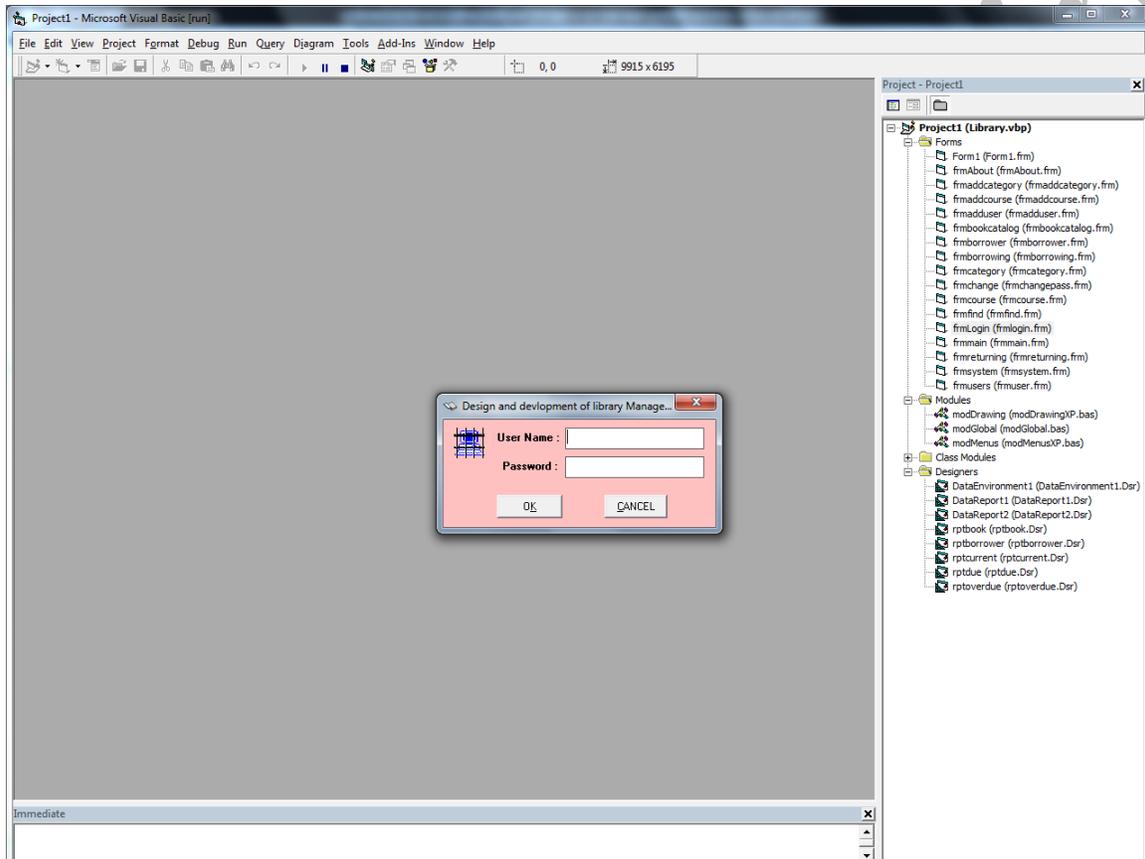
Field Name	Type	Constrains
Username	Text	Primary
Password	Text	Not Null

Projecthelp.in

## 16.Source Code

### CODING

#### Login



```
Dim pwctr As Integer
```

```
Private Sub cmdCancel_Click()
```

```
ReleaseMenus hwnd
```

```
End
```

```
End Sub
```

```
Private Sub cmdOK_Click()
```

```
Set userRS = New ADODB.Recordset
```

```
If txtUserName.Text <> "" Then
```

```

SQLstr = "Select * From userlist Where username = " & Trim(txtUserName.Text) &
""
userRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If txtUserName.Text <> "Administrator" Then
If txtUserName.Text <> "administrator" Then
If txtUserName.Text <> "ADMINISTRATOR" Then
    frmmain.mnusettingsystem.Enabled = False
    frmmain.mnusettinguser.Enabled = False
Else
    frmmain.mnusettingsystem.Enabled = True
    frmmain.mnusettinguser.Enabled = True
End If
Else
    frmmain.mnusettingsystem.Enabled = True
    frmmain.mnusettinguser.Enabled = True
End If
Else
    frmmain.mnusettingsystem.Enabled = True
    frmmain.mnusettinguser.Enabled = True
End If
If Not userRS.EOF And Not userRS.BOF Then
    If txtPassword.Text <> userRS!Password Then
        pwctr = pwctr + 1
        If pwctr = 1 Then
            MsgBox "Invalid password! You have 2 tries remaining!", vbOKOnly +
vbInformation, "Information"
            txtPassword.Text = ""
            txtPassword.SetFocus

            ElseIf pwctr = 2 Then
                MsgBox "Invalid password! You only have 1 try remaining!", vbOKOnly +
vbInformation, "Information"
                txtPassword.Text = ""
                txtPassword.SetFocus
            Else
                ReleaseMenus hwnd
            End
        End If
    Else
        Unload Me
        frmmain.Show
    End If
Else
    MsgBox "Invalid Username!", vbOKOnly + vbExclamation, "Warning.."
    txtUserName.Text = ""
    txtPassword.Text = ""

```

```
        txtUserName.SetFocus
    End If
Else
    MsgBox "Invalid Username and Password!", vbOKOnly + vbExclamation,
    "Warning.."
    txtUserName.SetFocus
End If

End Sub

Private Sub Form_Load()
    dbconnect
    txtUserName.Text = GetSetting(App.EXENAME, "TextBox", txtUserName.Name, "")
End Sub

Private Sub Form_Unload(Cancel As Integer)
    SaveSetting App.EXENAME, "Textbox", txtUserName.Name, txtUserName.Text
End Sub

Private Sub txtPassword_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    If cmdOK.Enabled = True Then
        cmdOK.Value = True
    End If
End If
End Sub

Private Sub txtUserName_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtPassword.SetFocus
End If
End Sub
```

## Main Form



```
'this code is exit button to prompt a message box b4 exiting
Private Const clMsgbxEXITAPP As Long = vbDefaultButton1 + vbQuestion + vbYesNo
Private mbIsDirty As Boolean
```

```
Private Sub Form_Load()
    dbconnect
    SetMenus hwnd, ImageList1
```

```
'this code is exit button to prompt a message box b4 exiting
Debug.Print "Form1::Load"
mbIsDirty = True
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
```

```
'this code is exit button to prompt a message box b4 exiting
Debug.Print "Form1::QueryUnload"
```

```
If mbIsDirty Then
    Cancel = CInt(pExitApp = False)
    If Not Cancel Then
        '-- We are ending the app. Clean up here.
        Debug.Print "Clean Up time..."
        Dim F As VB.Form
        For Each F In Forms
```

```

        Unload F
    Next
    ReleaseMenus hwnd
End If
End If
End Sub

Private Function pExitApp() As Boolean
'this code is exit button to prompt a message box b4 exiting
    Debug.Print "Exit Application"
    pExitApp = (MsgBox("Exit system?", clMsgbxEXITAPP, "Library") = vbYes)
End Function

Private Sub Form_Unload(Cancel As Integer)
'this code is exit button to prompt a message box b4 exiting
    Debug.Print "Form1::Unload"
End Sub

Private Sub mnuabout_Click()
frmAbout.Show vbModal
End Sub

Private Sub mnufilebook_Click()
    frmbookcatalog.Show vbModal
End Sub

Private Sub mnufileborrowerstud_Click()

End Sub

Private Sub mnufileborrower_Click()
    frmborrower.Show vbModal
End Sub

Private Sub mnufileexit_Click()
If MsgBox("Exit system?", vbQuestion + vbYesNo) = vbNo Then
    Exit Sub
End If
ReleaseMenus hwnd
End
End Sub

Private Sub mnufilefind_Click()
    frmfind.Show vbModal
End Sub

```

```

Private Sub mnureportsbook_Click()
    Set bookRS = New ADODB.Recordset
    bookRS.Open "book_catalog", libCON, adOpenKeyset, adLockReadOnly
    Set rptbook.DataSource = bookRS
    rptbook.Show vbModal
End Sub

Private Sub mnureportsborrow_Click()
    Set borrowerRS = New ADODB.Recordset
    borrowerRS.Open "borrower_record", libCON, adOpenKeyset, adLockReadOnly
    Set rptborrower.DataSource = borrowerRS
    rptborrower.Show vbModal
End Sub

Private Sub mnureportscurrent_Click()
    Set currentRS = New ADODB.Recordset
    currentRS.Open "current_borrow", libCON, adOpenKeyset, adLockReadOnly
    Set rptcurrent.DataSource = currentRS
    rptcurrent.Show
End Sub

Private Sub mnureportsdue_Click()
    Set currentRS = New ADODB.Recordset
    SQLstr = "select * from current_borrow where due_date = date()"
    currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    Set rptdue.DataSource = currentRS
    rptdue.Show vbModal
End Sub

Private Sub mnureportsover_Click()
    Set currentRS = New ADODB.Recordset
    SQLstr = "select * from current_borrow where due_date < date()"
    currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    Set rptoverdue.DataSource = currentRS
    rptoverdue.Show vbModal
End Sub

Private Sub mnusettingsystemcategory_Click()
    frmcategory.Show vbModal
End Sub

Private Sub mnusettingsystemcourse_Click()
    frmcourse.Show vbModal
End Sub

Private Sub mnusettingsystemfee_Click()

```

```
    frmssystem.Show vbModal  
End Sub
```

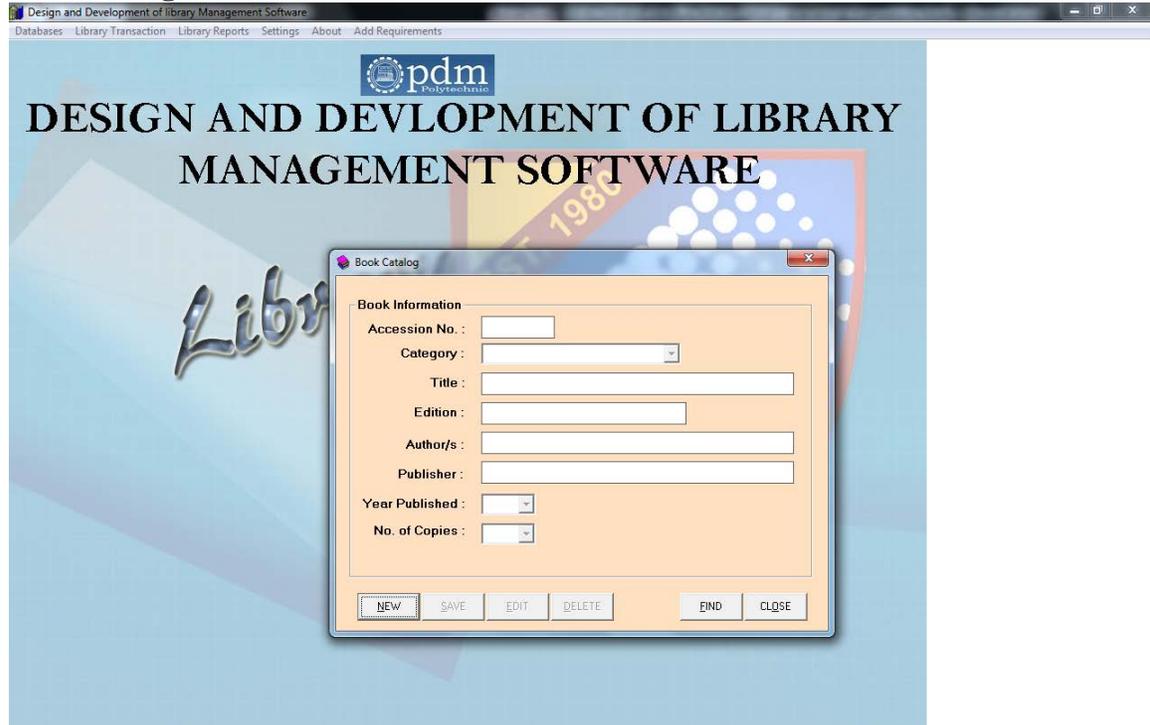
```
Private Sub mnusettinguser_Click()  
    frmusers.Show vbModal  
End Sub
```

```
Private Sub mnutransborrow_Click()  
    frmborrowing.Show vbModal  
End Sub
```

```
Private Sub mnutransreturn_Click()  
    frmreturning.Show vbModal  
End Sub
```

Projecthelpline.in

## Bookcatalog



```
Dim x, y As Integer
```

```
Private Sub clear()
```

```
    Combo1.Text = ""
```

```
    Text1.Text = ""
```

```
    Text2.Text = ""
```

```
    Text3.Text = ""
```

```
    Text4.Text = ""
```

```
    Text5.Text = ""
```

```
    Combo2.Text = ""
```

```
    Combo3.Text = ""
```

```
End Sub
```

```
Private Sub disable()
```

```
    Combo1.Enabled = False
```

```
    Text1.Enabled = False
```

```
    Text2.Enabled = False
```

```
    Text3.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text5.Enabled = False
```

```
    Combo2.Enabled = False
```

```
    Combo3.Enabled = False
```

```
End Sub
```

```

Private Sub enable()
    Combo1.Enabled = True
    Text1.Enabled = True
    Text2.Enabled = True
    Text3.Enabled = True
    Text4.Enabled = True
    Text5.Enabled = True
    Combo2.Enabled = True
    Combo3.Enabled = True
End Sub

Private Sub Command1_Click()
    enable
    Text1.SetFocus
    Command1.Enabled = False
    Command2.Enabled = True
    Command5.Enabled = False
    Command6.Caption = "&CANCEL"
End Sub

Private Sub Command2_Click()
If Text1.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text1.SetFocus
    Exit Sub
End If
If Combo1.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Combo1.SetFocus
    Exit Sub
End If
If Text2.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text2.SetFocus
    Exit Sub
End If
If Text5.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text5.SetFocus
    Exit Sub
End If
If Text3.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text3.SetFocus
    Exit Sub
End If

```

```

If Text4.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text4.SetFocus
    Exit Sub
End If
If Combo2.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Combo2.SetFocus
    Exit Sub
End If
If Combo3.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Combo3.SetFocus
    Exit Sub
End If

If Command2.Caption = "&SAVE" Then
    Set bookRS = New ADODB.Recordset
    SQLstr = "select access_no from book_catalog where access_no=" & Text1.Text & ""
    bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If Not bookRS.EOF And Not bookRS.BOF Then
        MsgBox "Accession No. already exist!", vbExclamation
        Text1.SetFocus
    End If
    Exit Sub
End If
If MsgBox("Save Book Catalog?", vbYesNo + vbQuestion) = vbYes Then
    Set bookRS = New ADODB.Recordset
    bookRS.Open "book_catalog", libCON, adOpenKeyset, adLockOptimistic
    With bookRS
        .AddNew
        !access_no = Text1.Text
        !category = Combo1.Text
        !Title = Text2.Text
        !Edition = Text5.Text
        !Author = Text3.Text
        !Publisher = Text4.Text
        !yr_publish = Combo3.Text
        !no_copy = Combo2.Text
        !available_copy = Combo2.Text
    End With
    .Update
    .Close
    MsgBox "Book Catalog Successfully Saved!", vbInformation
End If
Else
    If MsgBox("Update Book Catalog?", vbYesNo + vbQuestion) = vbYes Then

```

```

Set bookRS = New ADODB.Recordset
SQLstr = "Select * from book_catalog where access_no=" & Text1.Text & """"
bookRS.Open SQLstr, libCON, adOpenKeyset, adLockOptimistic
With bookRS
    !category = Combo1.Text
    !Title = Text2.Text
    !Edition = Text5.Text
    !Author = Text3.Text
    !Publisher = Text4.Text
    !yr_publish = Combo3.Text
    !no_copy = Combo2.Text
    !available_copy = Combo2.Text
    .Update
    .Close
End With
MsgBox "Book Catalog Successfully Updated!", vbInformation
End If
End If
clear
disable
Command1.Enabled = True
Command2.Enabled = False
Command2.Caption = "&SAVE"
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = True
Command6.Caption = "CL&OSE"
End Sub

Private Sub Command3_Click()
    enable
    Text1.Enabled = False
    Combo1.SetFocus
    Command2.Enabled = True
    Command2.Caption = "&UPDATE"
    Command3.Enabled = False
    Command4.Enabled = False
    Command7.Visible = False
End Sub

Private Sub Command4_Click()
    If MsgBox("Sure To Delete Book Catalog?", vbQuestion + vbYesNo) = vbYes Then
        Set bookCMD = New ADODB.Command
        SQLstr = "Delete * from book_catalog where access_no=" & Text1.Text & """"
        With bookCMD
            .ActiveConnection = libCON

```

```

        .CommandType = adCmdText
        .CommandText = SQLstr
        .Execute
    End With
    clear

    MsgBox "Book Catalog Successfully Deleted!", vbInformation

    Command1.Enabled = True
    Command2.Enabled = False
    Command3.Enabled = False
    Command4.Enabled = False
    Command5.Enabled = True
    Command6.Caption = "CL&OSE"
    Command7.Visible = False
End If
End Sub

Private Sub Command5_Click()
    Text1.Enabled = True
    Text1.SetFocus
    Command1.Enabled = False
    Command5.Enabled = False
    Command6.Caption = "&CANCEL"
    Command7.Visible = True
End Sub

Private Sub Command6_Click()
    If Command6.Caption = "CL&OSE" Then
        Unload Me
    Else
        clear
        disable
        Command1.Enabled = True
        Command2.Enabled = False
        Command3.Enabled = False
        Command4.Enabled = False
        Command5.Enabled = True
        Command6.Caption = "CL&OSE"
        Command7.Visible = False
    End If
End Sub

Private Sub Command7_Click()
    Set bookRS = New ADODB.Recordset
    SQLstr = "Select * from book_catalog where access_no=" & Text1.Text & ""

```

```

bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If bookRS.EOF And bookRS.BOF Then
    MsgBox "Accession Number Not Found!", vbExclamation
    Text1.SetFocus
    Exit Sub
End If
With bookRS
    Combo1.Text = !category
    Text2.Text = !Title
    Text3.Text = !Author
    Text4.Text = !Publisher
    Text5.Text = !Edition
    Combo2.Text = !no_copy
    Combo3.Text = !yr_publish
End With
Command3.Enabled = True
Command4.Enabled = True
End Sub

Private Sub Form_Load()
    dbconnect
    clear
    disable

    Set catRS = New ADODB.Recordset
    catRS.Open "category", libCON, adOpenKeyset, adLockReadOnly
    While catRS.EOF <> True
        Combo1.AddItem catRS!category
        catRS.MoveNext
    Wend

    x = 1
    While x <= 10
        Combo2.AddItem x
        x = x + 1
    Wend

    y = 1601
    While y <= 9999
        Combo3.AddItem y
        y = y + 1
    Wend

    Command2.Enabled = False
    Command3.Enabled = False
    Command4.Enabled = False

```

```
Command7.Visible = False
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If Command7.Visible = True Then
    If KeyAscii = 13 Then
        Command7.Value = True
    End If
End If
End Sub
```

### Borrower's record



```
Private Sub clear()
Text5.Text = ""
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Option1.Value = False
Option2.Value = False
Combo1.Text = ""
Text4.Text = ""
MaskedTextBox2.Text = "( ) - "
End Sub
```

```
Private Sub disable()
```

```
Text5.Enabled = False
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Option1.Enabled = False
Option2.Enabled = False
Text4.Enabled = False
Combo1.Enabled = False
MaskedTextBox2.Enabled = False
End Sub
```

```
Private Sub enable()
Text5.Enabled = True
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Option1.Enabled = True
Option2.Enabled = True
Text4.Enabled = True
'Combo1.Enabled = True
MaskedTextBox2.Enabled = True
End Sub
```

```
Private Sub Command1_Click()
enable
Text5.SetFocus
Command1.Enabled = False
Command2.Enabled = True
Command5.Enabled = False
Command6.Caption = "&CANCEL"
End Sub
```

```
Private Sub Command2_Click()
If Text5.Text = "" Then
MsgBox "Complete necessary information", vbExclamation
Text5.SetFocus
Exit Sub
End If
If Text1.Text = "" Then
MsgBox "Complete necessary information", vbExclamation
Text1.SetFocus
Exit Sub
End If
If Text2.Text = "" Then
MsgBox "Complete necessary information", vbExclamation
Text2.SetFocus
```

```

Exit Sub
End If
If Text3.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text3.SetFocus
    Exit Sub
End If
If Option1.Value = False And Option2.Value = False Then
    MsgBox "Complete necessary information", vbExclamation
    Option1.SetFocus
    Exit Sub
End If
If Option1.Value = True And Combo1.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Combo1.SetFocus
    Exit Sub
End If
If Text4.Text = "" Then
    MsgBox "Complete necessary information", vbExclamation
    Text4.SetFocus
    Exit Sub
End If

If Command2.Caption = "&SAVE" Then
    Set borrowerRS = New ADODB.Recordset
    SQLstr = "select borrower_id from borrower_record where borrower_id=" &
Text5.Text & ""
    borrowerRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If Not borrowerRS.EOF And Not borrowerRS.BOF Then
        MsgBox "Borrower ID already exist!", vbExclamation
        Text5.SetFocus
        Exit Sub
    End If

    If MsgBox("Save Borrower Record?", vbYesNo + vbQuestion) = vbYes Then
        Set borrowerRS = New ADODB.Recordset
        borrowerRS.Open "borrower_record", libCON, adOpenKeyset, adLockOptimistic
        With borrowerRS
            .AddNew
            !borrower_id = Text5.Text
            !name = Text1.Text
            !fname = Text2.Text
            !mI = Text3.Text
            If Option1.Value = True Then
                !Status = "Student"
            Else

```

```

        !Status = "Faculty / Employee"
    End If
    !course = Combo1.Text
    !Add = Text4.Text
    !contact = MaskEdBox2.Text
    .Update
    .Close
End With
MsgBox "Borrower Record Successfully Saved!", vbInformation
End If
Else
If MsgBox("Update Borrower Record?", vbYesNo + vbQuestion) = vbYes Then
Set borrowerRS = New ADODB.Recordset
SQLstr = "Select * from borrower_record where borrower_id=" & Text5.Text & ""
borrowerRS.Open SQLstr, libCON, adOpenKeyset, adLockOptimistic
With borrowerRS
    !name = Text1.Text
    !name = Text2.Text
    !mI = Text3.Text
    If Option1.Value = True Then
        !Status = "Student"
    Else
        !Status = "Faculty / Employee"
    End If
    !course = Combo1.Text
    !Add = Text4.Text
    !contact = MaskEdBox2.Text
    .Update
    .Close
End With
MsgBox "Borrower Record Successfully Updated!", vbInformation
End If
End If

clear
disable
Command1.Enabled = True
Command2.Enabled = False
Command2.Caption = "&SAVE"
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = True
Command6.Caption = "CL&OSE"
End Sub

Private Sub Command3_Click()

```

```

enable
Text5.Enabled = False
Text1.SetFocus
Combo1.Enabled = True
Command2.Enabled = True
Command2.Caption = "&UPDATE"
Command3.Enabled = False
Command4.Enabled = False
Command7.Visible = False
End Sub

Private Sub Command4_Click()
    If MsgBox("Sure To Delete Borrower Record?", vbQuestion + vbYesNo) = vbYes
    Then
        Set borrowerCMD = New ADODB.Command
        SQLstr = "Delete * from borrower_record where borrower_id=" & Text5.Text & ""
        With borrowerCMD
            .ActiveConnection = libCON
            .CommandType = adCmdText
            .CommandText = SQLstr
            .Execute
        End With
        clear

        MsgBox "Borrower Record Successfully Deleted!", vbInformation

        Command1.Enabled = True
        Command2.Enabled = False
        Command3.Enabled = False
        Command4.Enabled = False
        Command5.Enabled = True
        Command6.Caption = "CL&OSE"
        Command7.Visible = False
    End If
End Sub

Private Sub Command5_Click()
    Text5.Enabled = True
    Text5.SetFocus
    Command1.Enabled = False
    Command5.Enabled = False
    Command6.Caption = "&CANCEL"
    Command7.Visible = True
End Sub

Private Sub Command6_Click()

```

```

If Command6.Caption = "CL&OSE" Then
    Unload Me
Else
    clear
    disable
    Command1.Enabled = True
    Command2.Enabled = False
    Command3.Enabled = False
    Command4.Enabled = False
    Command5.Enabled = True
    Command6.Caption = "CL&OSE"
    Command7.Visible = False
End If
End Sub

Private Sub Command7_Click()
    Set borrowerRS = New ADODB.Recordset
    SQLstr = "Select * from borrower_record where borrower_id=" & Text5.Text & ""
    borrowerRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If borrowerRS.EOF And borrowerRS.BOF Then
        MsgBox "Borrower ID not valid!", vbExclamation
        Text5.SetFocus
        Exit Sub
    End If
    With borrowerRS
        Text1.Text = !lname
        Text2.Text = !fname
        Text3.Text = !mI
        If !Status = "Student" Then
            Option1.Value = True
        Else
            Option2.Value = True
        End If
        Combo1.Text = !course
        Text4.Text = !Add
        MaskedTextBox2.Text = !contact
    End With
    Command3.Enabled = True
    Command4.Enabled = True
End Sub

Private Sub Form_Load()

    dbconnect
    clear
    disable

```

```
Set courseRS = New ADODB.Recordset
courseRS.Open "course", libCON, adOpenKeyset, adLockReadOnly
While courseRS.EOF <> True
    Combo1.AddItem courseRS!course
    courseRS.MoveNext
Wend
```

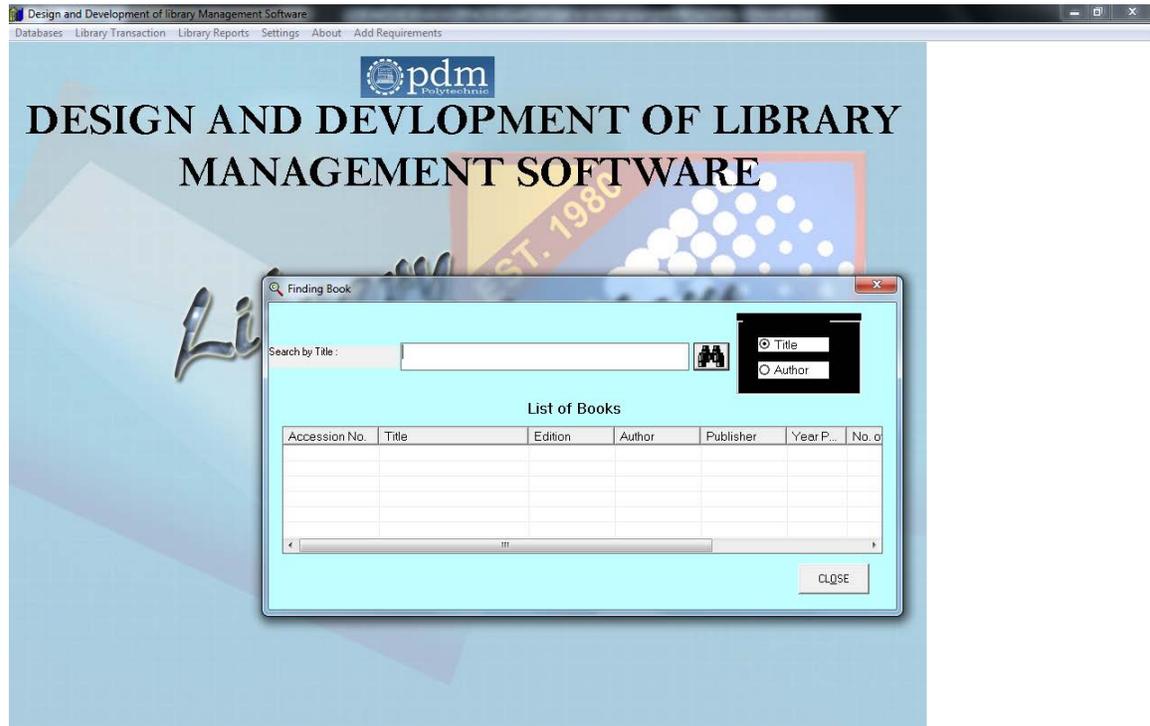
```
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
Command7.Visible = False
End Sub
```

```
Private Sub Option1_Click()
If Option1.Enabled = True Then
    Combo1.Enabled = True
    Combo1.Text = ""
End If
End Sub
```

```
Private Sub Option2_Click()
If Option2.Enabled = True Then
    Combo1.Enabled = False
    Combo1.Text = "n/a"
End If
End Sub
```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
If Command7.Visible = True Then
    If KeyAscii = 13 Then
        Command7.Value = True
    End If
End If
End Sub
```

## Findbook



```
Private Sub SetListViewTo(ByVal xrs As ADODB.Recordset, Optional ByVal  
strSMIcons As String = "", Optional ByVal strLRGIcons As String = "", Optional ByVal  
clmWidth)
```

```
ListView1.ListItems.clear
```

```
xrs.MoveFirst
```

```
While Not xrs.EOF
```

```
Set Item = ListView1.ListItems.Add(, "_ " & xrs.Fields(0).Value, xrs.Fields(0).Value)
```

```
Item.SubItems(1) = xrs!access_no
```

```
Item.SubItems(1) = xrs!Title
```

```
Item.SubItems(2) = xrs!Edition
```

```
Item.SubItems(3) = xrs!Author
```

```
Item.SubItems(4) = xrs!Publisher
```

```
Item.SubItems(5) = xrs!yr_publish
```

```
Item.SubItems(6) = xrs!no_copy
```

```
Item.SubItems(7) = xrs!available_copy
```

```
Item.SubItems(8) = xrs!borrow_copy
```

```
xrs.MoveNext
```

```
Wend
```

```
End Sub
```

```
Public Sub SetSectionViewTitle()
```

```

Set bookRS = New ADODB.Recordset
SQLstr = "SELECT * FROM book_catalog WHERE title='" & Text1.Text & "'"
bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If bookRS.EOF And bookRS.BOF Then
    MsgBox "No title found!", vbInformation
    Exit Sub
End If
SetListViewTo bookRS, 2, 2, clmWidth

End Sub

Public Sub SetSectionViewAuthor()

    Set bookRS = New ADODB.Recordset
    SQLstr = "SELECT * FROM book_catalog WHERE author='" & Text1.Text & "'"
    bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If bookRS.EOF And bookRS.BOF Then
        MsgBox "No Author found!", vbInformation
        Exit Sub
    End If

    SetListViewTo bookRS, 2, 2, clmWidth
End Sub

Private Sub Command1_Click()
    If Option1.Value = True Then
        SetSectionViewTitle
    Else
        SetSectionViewAuthor
    End If
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Option1.Value = True
End Sub

Private Sub Option1_Click()
    If Option1.Value = True Then
        Label1.Caption = "Search by Title :"
    End If

```

End Sub

Private Sub Option2\_Click()

    If Option2.Value = True Then

        Label1.Caption = "Search by Author :"

    End If

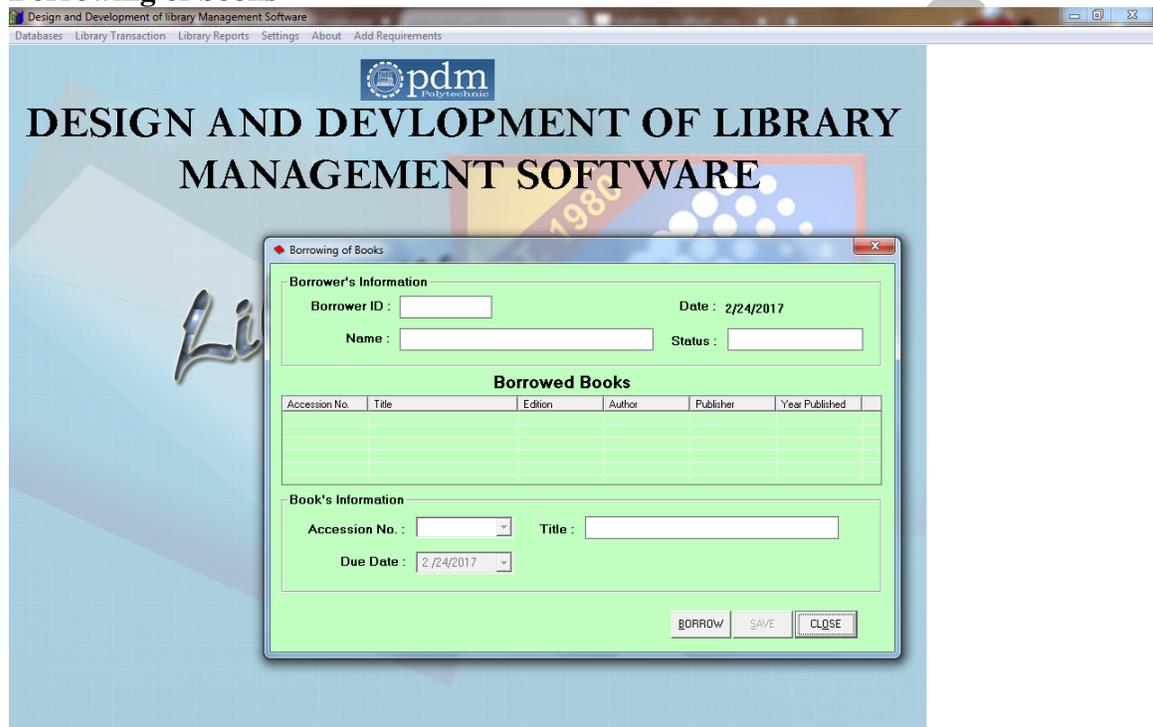
End Sub

Private Sub Text1\_Change()

    ListView1.ListItems.clear

End Sub

### Borrowing of books



Dim xcat, xedition, xauthor, xpublisher, xyr\_publish As String

Dim xavail\_copy As Integer

Private Sub SetListViewTo(ByVal xrs As ADODB.Recordset, Optional ByVal

strSMIcons As String = "", Optional ByVal strLRGIcons As String = "", Optional ByVal  
clmWidth)

    ListView1.ListItems.clear

    xrs.MoveFirst

    While Not xrs.EOF

        Set Item = ListView1.ListItems.Add(, "\_" & xrs.Fields(0).Value, xrs.Fields(0).Value)

        Item.SubItems(1) = xrs!access\_no

        Item.SubItems(2) = xrs!Title

        Item.SubItems(3) = xrs!Edition

        Item.SubItems(4) = xrs!Author

```

Item.SubItems(5) = xrs!Publisher
Item.SubItems(6) = xrs!yr_publish
xrs.MoveNext
Wend
End Sub

Public Sub SetSectionView()

Set currentRS = New ADODB.Recordset
SQLstr = "SELECT * FROM current_borrow WHERE borrower_id=" & Text1.Text
& ""
currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If currentRS.EOF And currentRS.BOF Then
Exit Sub
End If
SetListViewTo currentRS, 2, 2, clmWidth
End Sub

Private Sub clear()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Combo1.Text = ""
DTPicker1.Value = Date
End Sub

Private Sub disable()
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Combo1.Enabled = False
DTPicker1.Enabled = False
End Sub

Private Sub Combo1_Change()
Dim strPart As String, iLoop As Integer, iStart As Integer, strItem As String
If Not auto And Combo1.Text <> "" Then
iStart = Combo1.SelStart
strPart = Left$(Combo1.Text, iStart)
For iLoop = 0 To Combo1.ListCount - 1
strItem = UCase$(Combo1.List(iLoop))
If strItem Like UCase$(strPart & "*") And strItem <> UCase$(Combo1.Text)
Then
auto = True

```

```

        Combo1.SelText = Mid$(Combo1.List(iLoop), iStart + 1)
        Combo1.SelStart = iStart
        Combo1.SelLength = Len(Combo1.Text) - iStart
        auto = False
        Exit For
    End If
Next iLoop
End If

If Combo1.Text = "" Then
    Text4.Text = ""
    xcat = ""
    xedition = ""
    xauthor = ""
    xpublisher = ""
    xyr_publish = ""
End If
If Combo1.Text <> "" And Text4.Text <> "" Then
    Command2.Enabled = True
Else
    Command2.Enabled = False
End If
End Sub

Private Sub Combo1_Click()
    Set bookRS = New ADODB.Recordset
    SQLstr = "Select * from book_catalog where access_no=" & Combo1.Text & """"
    bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    With bookRS
        Text4.Text = !Title
        xcat = !category
        xedition = !Edition
        xauthor = !Author
        xpublisher = !Publisher
        xyr_publish = !yr_publish
        xavail_copy = !available_copy
    End With

    If Combo1.Text <> "" And Text4.Text <> "" Then
        Command2.Enabled = True
    Else
        Command2.Enabled = False
    End If
End Sub

Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)

```

```

If KeyCode = vbKeyBack Or KeyCode = vbKeyDelete Then
    auto = True
    Combo1.SelText = ""
    auto = False
ElseIf KeyCode = vbKeyReturn Then
    Combo1_LostFocus
    Combo1.SelStart = Len(Combo1.Text)
End If
End Sub

```

```

Private Sub Combo1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    Set bookRS = New ADODB.Recordset
    SQLstr = "Select * from book_catalog where access_no=" & Combo1.Text & ""
    bookRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If bookRS.EOF And bookRS.BOF Then
        Exit Sub
    End If
    With bookRS
        Text4.Text = !Title
        xcat = !category
        xedition = !Edition
        xauthor = !Author
        xpublisher = !Publisher
        xyr_publish = !yr_publish
        xavail_copy = !available_copy
    End With

    If Combo1.Text <> "" And Text4.Text <> "" Then
        Command2.Enabled = True
    Else
        Command2.Enabled = False
    End If
End If
End Sub

```

```

Private Sub Combo1_LostFocus()
Dim iLoop As Integer
If Combo1.Text <> "" Then
    For iLoop = 0 To Combo1.ListCount - 1
        If UCase$(Combo1.List(iLoop)) = UCase$(Combo1.Text) Then
            auto = True
            Combo1.Text = Combo1.List(iLoop)
            auto = False
        Exit For
    End If

```

```

    Next iLoop
End If
End Sub

Private Sub Command1_Click()
    Text1.Enabled = True
    Text1.SetFocus
    Command1.Enabled = False
    Command6.Caption = "&CANCEL"
    Command7.Visible = True
End Sub

Private Sub Command3_Click()
    Text1.Enabled = False
    Command7.Visible = False
    frmaddbook.Show vbModal
End Sub

Private Sub Command2_Click()
    If xavail_copy = 0 Then
        MsgBox "There is no available copy of this book!", vbInformation
        Combo1.SetFocus
        Exit Sub
    End If

    Set currentRS = New ADODB.Recordset
    SQLstr = "Select * from current_borrow where access_no=" & Combo1.Text & "" &
" and borrower_id=" & Text1.Text & ""
    currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If Not currentRS.EOF And Not currentRS.BOF Then
        MsgBox "Borrower's cannot borrow 2 same book title!", vbInformation
        Combo1.SetFocus
        Exit Sub
    End If

    Set currentRS = New ADODB.Recordset
    currentRS.Open "current_borrow", libCON, adOpenKeyset, adLockOptimistic
    With currentRS
        .AddNew
        !access_no = Combo1.Text
        !Title = Text4.Text
        !category = xcat
        !Edition = xedition
        !Author = xauthor
        !Publisher = xpublisher
        !yr_publish = xyr_publish
    End With

```

```

!borrow_date = Label4.Caption
If DTPicker1.Enabled = True Then
    !due_date = DTPicker1.Value
End If
!borrower_id = Text1.Text
!Name = Text2.Text
!Status = Text3.Text
.Update
.Close
End With

Set bookRS = New ADODB.Recordset
SQLstr = "select * from book_catalog where access_no=" & Combo1.Text & """"
bookRS.Open SQLstr, libCON, adOpenKeyset, adLockOptimistic
With bookRS
    !available_copy = xavail_copy - 1
    !borrow_copy = !borrow_copy + 1
    .Update
    .Close
End With

SetSectionView
MsgBox "Library Transaction Successfully Saved!", vbInformation

If ListView1.ListItems.Count = 3 Then
    MsgBox "Borrower reach the maximum book that can be borrowed at a time!",
vbExclamation
    clear
    disable
    ListView1.ListItems.clear
    Command1.Enabled = True
    Command2.Enabled = False
    Command6.Caption = "CL&OSE"
    Exit Sub
End If

If MsgBox("Borrow Another Book?", vbQuestion + vbYesNo) = vbYes Then
    Combo1.Text = ""
    Combo1.SetFocus
    Exit Sub
End If
clear
disable
ListView1.ListItems.clear
Command1.Enabled = True
Command2.Enabled = False

```

```

        Command6.Caption = "CL&OSE"
End Sub

Private Sub Command6_Click()
If Command6.Caption = "CL&OSE" Then
    Unload Me
Else
    clear
    disable
    Command1.Enabled = True
    Command2.Enabled = False
    Command6.Caption = "CL&OSE"
    Command7.Visible = False
End If
End Sub

Private Sub Command7_Click()
    Set borrowerRS = New ADODB.Recordset
    SQLstr = "Select * from borrower_record where borrower_id=" & Text1.Text & ""
    borrowerRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If borrowerRS.EOF And borrowerRS.BOF Then
        MsgBox "Borrower ID not valid!", vbExclamation
        Text1.SetFocus
        Exit Sub
    End If
    With borrowerRS
        Text2.Text = !name & ", " & !fname & " " & !mI
        Text3.Text = !Status
    End With

    SetSectionView
    If ListView1.ListItems.Count = 3 Then
        MsgBox "Borrower Already Borrowed 3 Books!", vbExclamation
        clear
        ListView1.ListItems.clear
        Text1.Text = ""
        Text1.SetFocus
        Exit Sub
    End If
    If Text3.Text = "Faculty / Employee" Then
        DTPicker1.Enabled = False
    Else
        DTPicker1.Enabled = True
    End If
    Combo1.Enabled = True
    Combo1.SetFocus

```

```
Text1.Enabled = False
Command7.Visible = False
End Sub
```

```
Private Sub Form_Load()
    dbconnect
    clear
    disable
    Label4.Caption = Date
    Command2.Enabled = False
    Command7.Visible = False
```

```
'accession number
Set bookRS = New ADODB.Recordset
bookRS.Open "book_catalog", libCON, adOpenKeyset, adLockReadOnly
While bookRS.EOF <> True
    Combo1.AddItem bookRS!access_no
    bookRS.MoveNext
Wend
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    Command7.Value = True
End If
End Sub
```

## Returns of books

Design and Development of library Management Software

Databases Library Transaction Library Reports Settings About Add Requirements

pdm

# DESIGN AND DEVELOPMENT OF LIBRARY MANAGEMENT SOFTWARE

### Returning of Books

Book's Information

Accession No. :  Title :

Borrower's Information

Borrower ID :  Name :  Status :

Borrowing Information

Date Borrowed :  Due Date :

In Case of Overdue Books

No. of Days : 0

Charge Fee Per Day : 0.00 Total Charge : 0.00

RETURN SAVE CLOSE

```
Dim xaccess_no, xborrower_id As String
```

```
Private Sub clear()
```

```
    Combo1.Text = ""
```

```
    Text1.Text = ""
```

```
    Combo2.Text = ""
```

```
    Text2.Text = ""
```

```
    Text3.Text = ""
```

```
    Text4.Text = Date
```

```
    Text5.Text = Date
```

```
    Label11.Caption = "0"
```

```
    Label12.Caption = "0.00"
```

```
    Label13.Caption = "0.00"
```

```
End Sub
```

```
Private Sub Combo1_Change()
```

```
    Dim strPart As String, iLoop As Integer, iStart As Integer, strItem As String
```

```
    If Not auto And Combo1.Text <> "" Then
```

```
        iStart = Combo1.SelStart
```

```
        strPart = Left$(Combo1.Text, iStart)
```

```
        For iLoop = 0 To Combo1.ListCount - 1
```

```
            strItem = UCase$(Combo1.List(iLoop))
```

```
            If strItem Like UCase$(strPart & "*") And strItem <> UCase$(Combo1.Text)
```

```
        Then
```

```

        auto = True
        Combo1.SelText = Mid$(Combo1.List(iLoop), iStart + 1)
        Combo1.SelStart = iStart
        Combo1.SelLength = Len(Combo1.Text) - iStart
        auto = False
    Exit For
End If
Next iLoop
End If

If Combo1.Text = "" Then
    Text1.Text = ""
    Combo2.Text = ""
End If
If Combo1.Text <> "" And Text1.Text <> "" Then
    Combo2.Enabled = True
Else
    Combo2.Enabled = False
End If
End Sub

Private Sub Combo1_Click()
    Set currentRS = New ADODB.Recordset
    SQLstr = "Select * from current_borrow where access_no=" & Combo1.Text & ""
    currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If currentRS.EOF And currentRS.BOF Then
        Exit Sub
    End If
    With currentRS
        Text1.Text = !Title
        Combo2.clear
        xborrower_id = ""
        While .EOF <> True
            If !borrower_id = xborrower_id Then
                .MoveNext
            Else
                Combo2.AddItem !borrower_id
                xborrower_id = !borrower_id
                .MoveNext
            End If
        Wend
        Text2.Text = ""
        Text3.Text = ""
        Text4.Text = Date
        Text5.Text = Date
    End With

```

```

If Combo1.Text <> "" And Text1.Text <> "" Then
    Combo2.Enabled = True
Else
    Combo2.Enabled = False
End If
End Sub

Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyBack Or KeyCode = vbKeyDelete Then
        auto = True
        Combo1.SelText = ""
        auto = False
    ElseIf KeyCode = vbKeyReturn Then
        Combo1_LostFocus
        Combo1.SelStart = Len(Combo1.Text)
    End If
End Sub

Private Sub Combo1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    Set currentRS = New ADODB.Recordset
    SQLstr = "Select * from current_borrow where access_no=" & Combo1.Text & ""
    currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
    If currentRS.EOF And currentRS.BOF Then
        Exit Sub
    End If
    With currentRS
        Text1.Text = !Title
        Combo2.clear
        xborrower_id = ""
        While .EOF <> True
            If !borrower_id = xborrower_id Then
                .MoveNext
            Else
                Combo2.AddItem !borrower_id
                xborrower_id = !borrower_id
                .MoveNext
            End If
        Wend
        Text2.Text = ""
        Text3.Text = ""
        Text4.Text = Date
        Text5.Text = Date
    End With

```

```

If Combo1.Text <> "" And Text1.Text <> "" Then
    Combo2.Enabled = True
Else
    Combo2.Enabled = False
End If
End If
End Sub

Private Sub Combo1_LostFocus()
Dim iLoop As Integer
If Combo1.Text <> "" Then
    For iLoop = 0 To Combo1.ListCount - 1
        If UCase$(Combo1.List(iLoop)) = UCase$(Combo1.Text) Then
            auto = True
            Combo1.Text = Combo1.List(iLoop)
            auto = False
            Exit For
        End If
    Next iLoop
End If
End Sub

Private Sub combo2_Change()
Dim strPart As String, iLoop As Integer, iStart As Integer, strItem As String
If Not auto And Combo2.Text <> "" Then
    iStart = Combo2.SelStart
    strPart = Left$(Combo2.Text, iStart)
    For iLoop = 0 To Combo2.ListCount - 1
        strItem = UCase$(Combo2.List(iLoop))
        If strItem Like UCase$(strPart & "*") And strItem <> UCase$(Combo2.Text)
Then
            auto = True
            Combo2.SelText = Mid$(Combo2.List(iLoop), iStart + 1)
            Combo2.SelStart = iStart
            Combo2.SelLength = Len(Combo2.Text) - iStart
            auto = False
            Exit For
        End If
    Next iLoop
End If

If Combo2.Text = "" Then
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = Date
    Text5.Text = Date

```

```
End If
End Sub
```

```
Private Sub Combo2_Click()
Set currentRS = New ADODB.Recordset
SQLstr = "select * from current_borrow where borrower_id=" & Combo2.Text & ""
currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If currentRS.EOF And currentRS.BOF Then
Exit Sub
End If
With currentRS
Text2.Text = !Name
Text3.Text = !Status
Text4.Text = !borrow_date
On Error Resume Next
Text5.Text = !due_date
End With
End Sub
```

```
Private Sub combo2_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = vbKeyBack Or KeyCode = vbKeyDelete Then
auto = True
Combo2.SelText = ""
auto = False
ElseIf KeyCode = vbKeyReturn Then
combo2_LostFocus
Combo2.SelStart = Len(Combo2.Text)
End If
End Sub
```

```
Private Sub Combo2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Set currentRS = New ADODB.Recordset
SQLstr = "select * from current_borrow where borrower_id=" & Combo2.Text & ""
currentRS.Open SQLstr, libCON, adOpenKeyset, adLockReadOnly
If currentRS.EOF And currentRS.BOF Then
Exit Sub
End If
With currentRS
Text2.Text = !Name
Text3.Text = !Status
Text4.Text = !borrow_date
On Error Resume Next
Text5.Text = !due_date
End With
End If
```

```
End Sub
```

```
Private Sub combo2_LostFocus()
```

```
Dim iLoop As Integer
```

```
    If Combo2.Text <> "" Then
```

```
        For iLoop = 0 To Combo2.ListCount - 1
```

```
            If UCase$(Combo2.List(iLoop)) = UCase$(Combo2.Text) Then
```

```
                auto = True
```

```
                Combo2.Text = Combo2.List(iLoop)
```

```
                auto = False
```

```
            Exit For
```

```
        End If
```

```
    Next iLoop
```

```
End If
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    If currentRS.EOF And currentRS.BOF Then
```

```
        MsgBox "There is no books currently borrowed!", vbExclamation
```

```
    Exit Sub
```

```
End If
```

```
Combo1.Enabled = True
```

```
Combo1.SetFocus
```

```
Command1.Enabled = False
```

```
Command6.Caption = "&CANCEL"
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Set bookRS = New ADODB.Recordset
```

```
    SQLstr = "Select * from book_catalog where access_no=" & Combo1.Text & """"
```

```
    bookRS.Open SQLstr, libCON, adOpenKeyset, adLockOptimistic
```

```
    With bookRS
```

```
        !available_copy = !available_copy + 1
```

```
        !borrow_copy = !borrow_copy - 1
```

```
    .Update
```

```
    .Close
```

```
    End With
```

```
    Set currentCMD = New ADODB.Command
```

```
    SQLstr = "Delete * from current_borrow where access_no=" & Combo1.Text & """" &  
" and borrower_id=" & Combo2.Text & """"
```

```
    With currentCMD
```

```
        .ActiveConnection = libCON
```

```
        .CommandType = adCmdText
```

```
        .CommandText = SQLstr
```

```
    .Execute
```

```

    End With
    clear
    Combo1.Enabled = False
    Command1.Enabled = True
    Command2.Enabled = False
    Command6.Caption = "CL&OSE"

    MsgBox "Library Transaction Successfully Saved!", vbInformation

    Set currentRS = New ADODB.Recordset
    currentRS.Open "current_borrow", libCON, adOpenKeyset, adLockReadOnly
    Combo1.clear
    xaccess_no = ""
    While currentRS.EOF <> True
        If currentRS!access_no = xaccess_no Then
            currentRS.MoveNext
        Else
            Combo1.AddItem currentRS!access_no
            xaccess_no = currentRS!access_no
            currentRS.MoveNext
        End If
    Wend
End Sub

Private Sub Command6_Click()
    If Command6.Caption = "CL&OSE" Then
        Unload Me
    Else
        clear
        Combo1.Enabled = False
        Combo2.Enabled = False
        Command1.Enabled = True
        Command2.Enabled = False
        Command6.Caption = "CL&OSE"
    End If
End Sub

Private Sub Form_Load()
    dbconnect
    clear
    Combo1.Enabled = False
    Combo2.Enabled = False
    Command2.Enabled = False

    Set currentRS = New ADODB.Recordset
    currentRS.Open "current_borrow", libCON, adOpenKeyset, adLockReadOnly

```

```

xaccess_no = ""
While currentRS.EOF <> True
    If currentRS!access_no = xaccess_no Then
        currentRS.MoveNext
    Else
        Combo1.AddItem currentRS!access_no
        xaccess_no = currentRS!access_no
        currentRS.MoveNext
    End If
Wend
End Sub

Private Sub Text2_Change()
If Text2.Text = "" Then
    Command2.Enabled = False
Else
    Command2.Enabled = True
End If

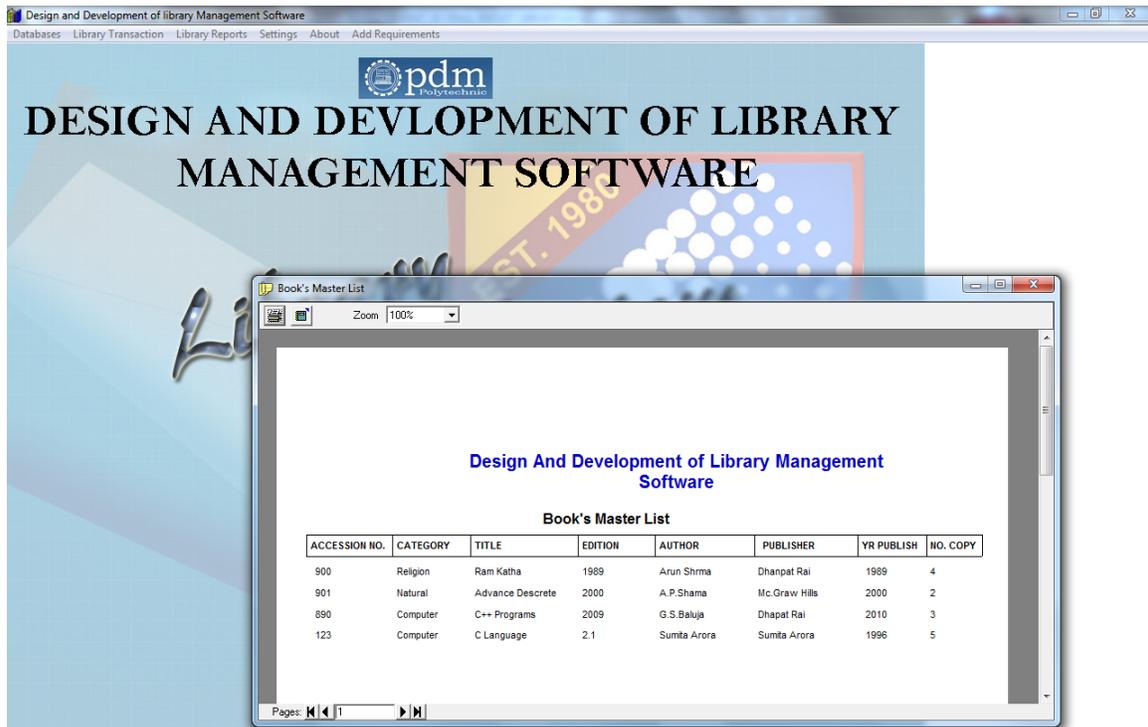
End Sub

Private Sub Text5_Change()
If Date > DateValue(Text5.Text) Then
    Label11.Caption = Date - DateValue(Text5.Text)
    Set feeRS = New ADODB.Recordset
    feeRS.Open "charge", libCON, adOpenKeyset, adLockReadOnly
    Label12.Caption = feeRS!charge_fee
    Label13.Caption = Val(Label11.Caption) * Val(Label12.Caption)
    Label12.Caption = Format(Label12.Caption, "###.00")
    Label13.Caption = Format(Label13.Caption, "###.00")
Else
    Label11.Caption = "0"
    Label12.Caption = "0.00"
    Label13.Caption = "0.00"
End If
End Sub

```

## 17. REPORT GENERATION

### Bookmaster list

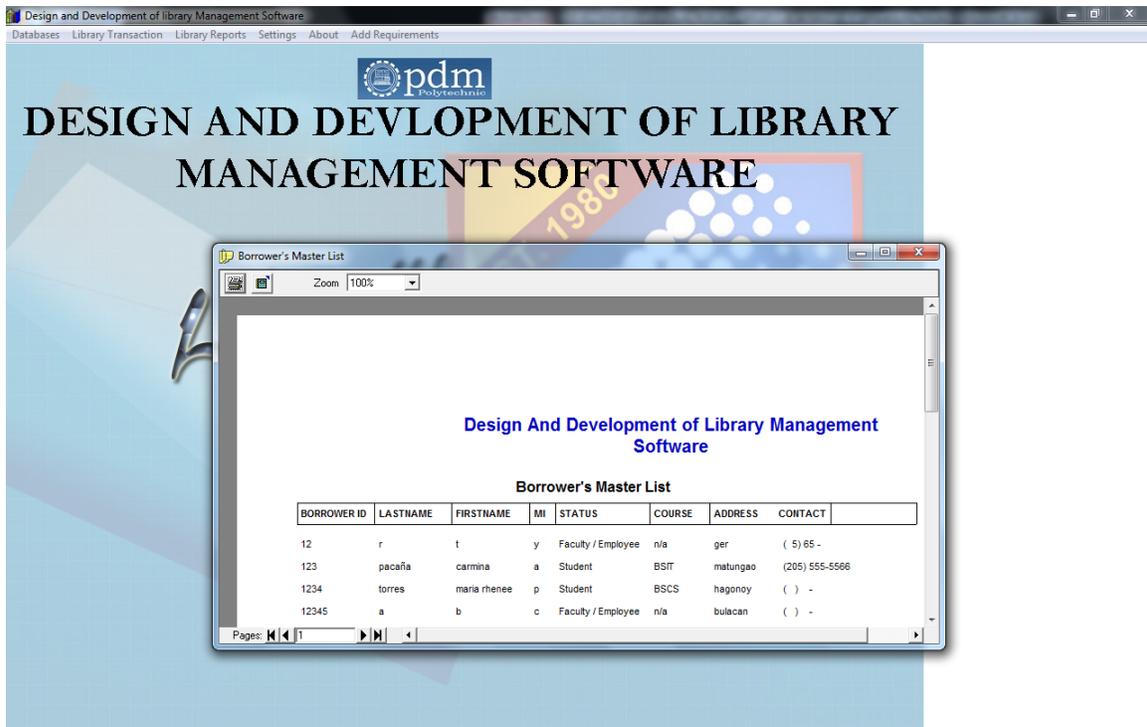


The screenshot displays the 'Design and Development of library Management Software' application. The main window shows a report titled 'Book's Master List' with the following data:

ACCESSION NO.	CATEGORY	TITLE	EDITION	AUTHOR	PUBLISHER	YR PUBLISH	NO. COPY
900	Religion	Ram Katha	1989	Arun Shirma	Dhanpat Rai	1989	4
901	Natural	Advance Descrete	2000	A.P.Shama	McGraw Hills	2000	2
890	Computer	C++ Programs	2009	G.S.Baluja	Dhapal Rai	2010	3
123	Computer	C Language	2.1	Sumita Arora	Sumita Arora	1996	5

Project

## Borrower's master list



Design and Development of library Management Software

Databases Library Transaction Library Reports Settings About Add Requirements

**DESIGN AND DEVELOPMENT OF LIBRARY MANAGEMENT SOFTWARE**

**Borrower's Master List**

BORROWER ID	LASTNAME	FIRSTNAME	MI	STATUS	COURSE	ADDRESS	CONTACT
12	r	t	y	Faculty / Employee	n/a	ger	( 5) 65 -
123	pacaña	carmina	a	Student	BSIT	matungao	(205) 555-5566
1234	torres	maria rhenee	p	Student	BSCS	hagonoy	( ) -
12345	a	b	c	Faculty / Employee	n/a	bulacan	( ) -

Pages: 1

Project

## Currently borrowed books

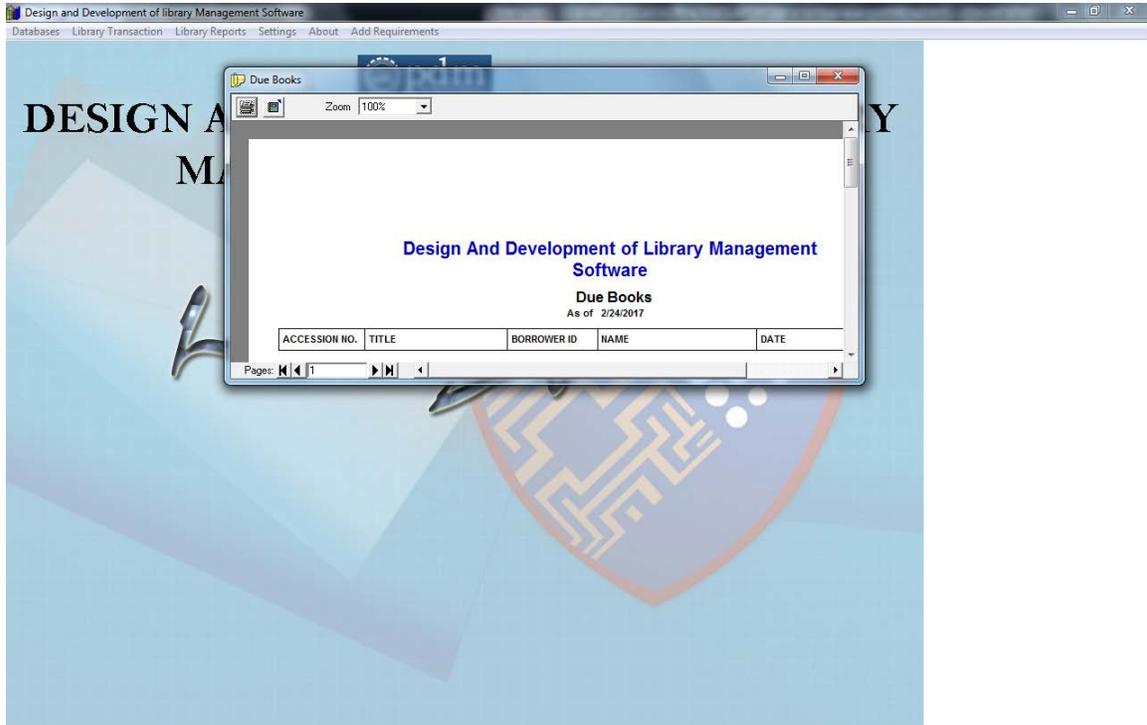
The screenshot displays a web browser window titled "Design and Development of library Management Software". The browser's address bar shows "Databases Library Transaction Library Reports Settings About Add Requirements". The main content area features a logo for "pdm Polytechnic Institute" and the text "DESIGN AND DEVELOPMENT OF LIBRARY M". A modal window titled "Currently Borrowed Books" is open, displaying a table with the following data:

ACCESSION NO.	TITLE	BORROWER ID	NAME	DATE
1	w	1234	torres, maria rhenee p	12/8/2014
2	r	1234	torres, maria rhenee p	12/8/2014
123	kaht ano	123	pacaña, carmina a	8/4/2014
1	w	12	r, ty	9/14/2014
120	k	Bo8	Baluja, Gurmeet Kau	8/28/2014
900	Ram Katha	Bo8	Baluja, Gurmeet Kau	8/28/2014

The modal window also includes a "Zoom | 100%" control and a "Pages: 1" indicator at the bottom.

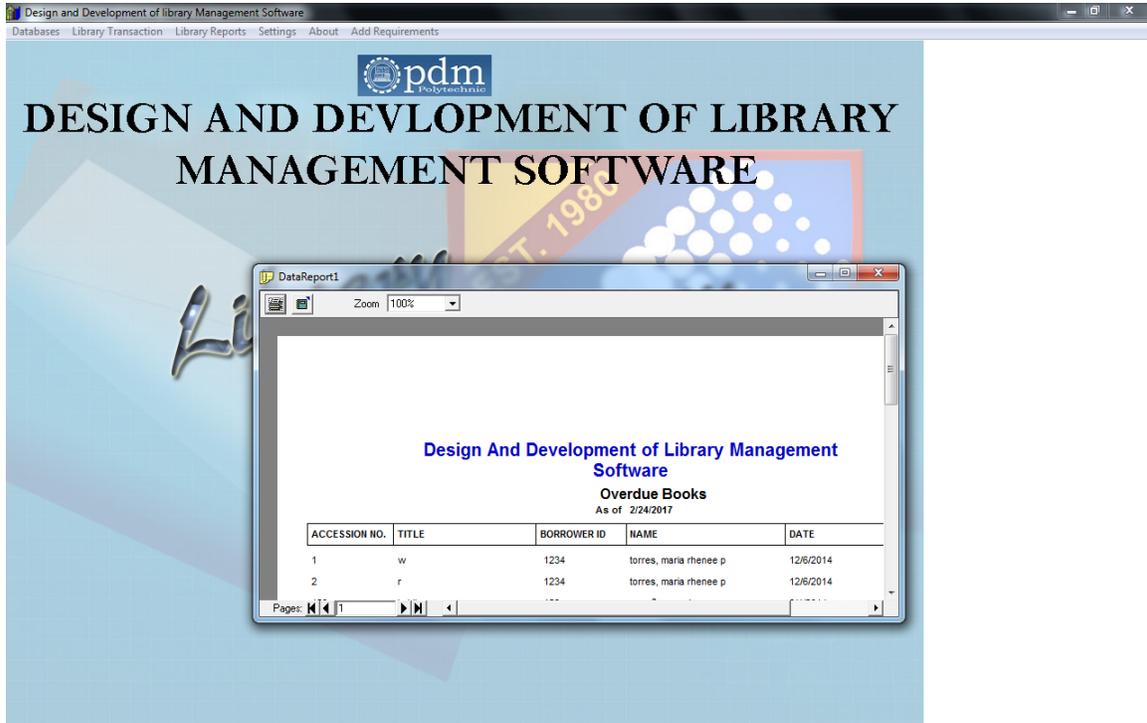
Project

## Due books



Projec

## Overduebooks

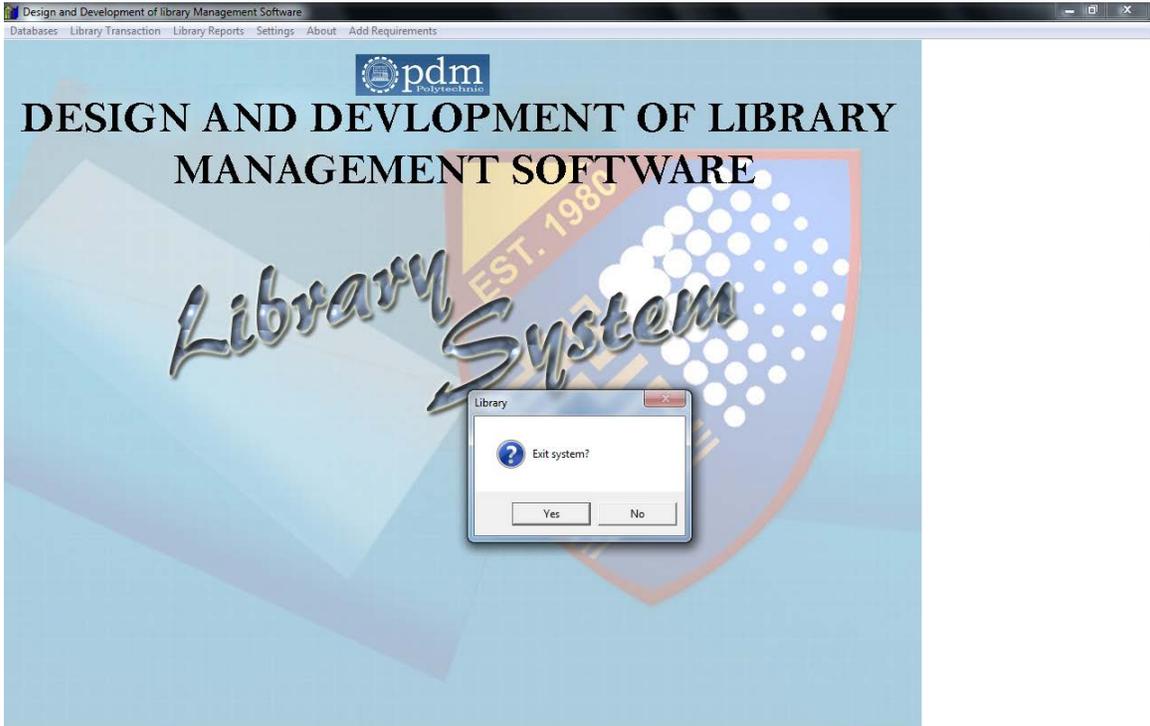


The screenshot displays a web browser window titled "Design and Development of library Management Software". The browser's address bar shows "Databases Library Transaction Library Reports Settings About Add Requirements". The main content area features a blue header with the "pdm" logo and the title "DESIGN AND DEVELOPMENT OF LIBRARY MANAGEMENT SOFTWARE". A "DataReport1" window is overlaid on the page, showing a report titled "Design And Development of Library Management Software Overdue Books As of 2/24/2017". The report contains a table with the following data:

ACCESSION NO.	TITLE	BORROWER ID	NAME	DATE
1	w	1234	torres, maria rhenee p	12/6/2014
2	r	1234	torres, maria rhenee p	12/6/2014

The report window also includes a "Pages:" indicator at the bottom left and a "Zoom: 100%" dropdown at the top left.

Projec



Project

## **18. DESCRIPTION OF MODULES**

### **Module 1: Password Module**

In this module, this website is for multiple users. If a User enters a password and the software checks its validity. If the password is valid then option is given to change the password, otherwise “Invalid

User/Password” message is displayed. There is an option for password recovery, log out, login, new users sign in. The Administrator can also update changes in the site after login.

### **Module 2: Creating new Entities ( Books, Members ,Courses ,Category of books)**

In this module, whenever a new entity is required to be added the corresponding forms are opened and the database is manipulated to check whether the data is already existing or not. If it already exists, then it prompts that “Entry already existing” and if not than the data is entered with the various validation checks.

### **Module 3: Modifying / Updating Existing Entities**

In this module, whenever an existing entity is required to be modified the corresponding forms are opened and the database is manipulated and the data is fetched. Now the administrator can made the required changes and then accordingly, he updates the data. Again, the checks are followed in case there is any invalid entry.

### **Module 4: Searching**

In this module, whenever an existing entity is required to be searched the corresponding forms are opened and the database is manipulated and the data is fetched. Again, the checks are followed in case there is any invalid entry.

### **Module 5: Fine generation**

Our school issues book for a week. One has to return the book one week after the date on which it was issued. The administrator issues the book again. So the administrator needs to maintain the records regarding the issue date, return date, so that the appropriate fine could be easily generated on the return of the book after the date allotted.

### **Module 6: Issue**

Before issuing a book to a borrower, it is checked whether the borrower is valid person and how many books have already issued by him. A borrower can borrow only 3 books. After issue of a book, the no. of copies of that book will be decreased by one in the library. There is also a change in borrower table, as no. of books issued by him will increase by one.

### **Module 7: Return**

When a borrower returns a book, fine can be calculated in case a book is returned after due date. After returning the book, no. of copies of that book in the library will be increased by one and borrower can issue more books.

### **Module 8: Reports**

This module is concerned with the various reports generation based on the admin request.

Following types of reports are generated-

- ❖ Books Report
- ❖ Fine Report
- ❖ Members Report
- ❖ Books due on the current date
- ❖ Overdue Books Details

❖ Books Available

### **Module 9: Validation of Data Entered by the User & Error Handling**

In this module, the validity of data entered by the user during the various business processes is checked through various validation checks. For example, there should not be any characters entered in the numeric fields, likewise if there is any error occurs than it should handle that particular error and give the required messages.

## **19. SECURITY FEATURES**

The following are the security features that are provided by our System

1. Authentication of Users through Strong username password checking mechanism.
2. Use of Sessions for maintaining User Session so that illegal users cannot access the System.
3. Use of Web.Config file for authentication so that nobody can access files directly without logging into the system.

## **20. OPTIMIZATION**

The following factors are identified while analyzing the performance analysis/Optimization of the current system.

### **Design of database**

The design of the database for the project is done using the principle of Normalization and using the Relational model for System Design. This has done to reduce the data redundancy and risk of data integrity.

### **Front end tool used**

To make a faster and easy web based project VB is used as a front-end tool. The data is connected by using ADO.NET control, which is reliable and fast system with SQL Server 2000 as a back end.

### **Environment on which the system will work**

This project works well within the Windows family of operating Systems and will be supported by all major browsers.

### **Data volumes**

As there is a possibility that the volume of data can reach up to lack in few years ACCESS is used as a back end. As ACCESS is a reliable database for large database maintenance.

### **User profile**

The data that is processed or entered into the present project will be done by the software people who are already familiar with computers functionality so the user profile includes educated class of people.

## **21. TESTING**

Testing plays a critical role in quality assurance of the software. Testing is a dynamic method for verification and validation. With the help of testing we observe the failure of the system in terms of logical and runtime errors. The testing process can deduce the presence of fault in the system; however, separate activities have to be performed to identify the faults.

There are two methods of testing: functional & structural. In functional testing, the internal logic of the system under testing is not considered and the test cases are decided from the specification or the requirements. It is often called "Black Box Testing". In structural testing, the test cases are decided entirely on the internal logic of the program or module being tested.

As the goal of testing is to detect any errors in the programs different types of testing are often used. Unit testing is used to test a module or a small collection of modules and the focus is on detecting coding errors in modules. During integration testing modules are combined into sub-systems, which are then tested. The goal here is to test the system design. In system testing and acceptance testing, the entire system is tested. The goal here is to test the requirements. Structural testing can be used for unit testing while at higher level mostly functional testing is used.

In Our Project Event Management System all the above levels of testing were done. Following are the list of errors that were encountered at each level of testing

### **Unit Testing**

- Errors in Database Design: During coding it was discovered that some of the tables in the database didn't have all the attributes needed to implement some of the functionalities of the project also different naming conventions were being used for accessing the same Identifier. The database tables were then subsequently modified.
- Errors in Designing Queries: Here it was found that some Queries have not been properly written as the Query returned unexpected values for some of our test cases.
- Errors in Date Format: These errors cropped up because of poor knowledge of system setting of Date Field as we were using System date format in some of our files.
- Error in Database connectivity: This error was frequently encountered when the code was migrated from one terminal to another. The root cause of this error was the difference in server names at the respective terminals.

### **Integration Testing**

- Errors in working of Links: This was the major problem that was encountered often. This was due to use of different file names as given in the link and the actual file name. These bugs were then easily resolved.
- Errors in passing variables across pages: This problem was also resolved by using session variables, global variables and passing parameters along with Url.

- Error in code planning: There was some code planning errors like applying transactions at every place where database updating or insertion was made. These transactions were later inserted at proper places.

### **System Testing**

- Error in maintaining User Session : This problem occurred when the user was configuring his event the during configuration the session expired too early as its time out period was set to around 15 minutes ,which was then increased to about 60 minutes for convenience.
- List of Service Providers and Services were not displayed properly in increasing order of their Id's, this issue was the resolved by using Order By command in the respective Queries.

## **22. POST IMPLEMENTATION MAINTENANCE & REVIEW**

### **Maintenance**

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. Like any system, there is an aging process that requires periodic maintenance of hardware and software. If the new information is inconsistent with the design specifications, then changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. The importance of maintenance is to continue to bring the new system to standards.

User priorities, changes in organizational requirements, or environmental factors also call for system enhancements.

### **Project Termination**

A system project may be dropped at any time prior to implementation although it becomes more difficult (and Loan) when it goes past the design phase. Generally projects are dropped if after a review process, it is learned that:

- ❑ Changing objectives or requirements of the user cannot be met by the existing design.
- ❑ Benefits realized from the candidate system do not justify commitment to implementation.
- ❑ There is a sudden change in the user's budget or an increase in design Loan beyond the estimate made during the feasibility study.
- ❑ The project greatly exceeds the time and Loan schedule.

In each case, a system project may be terminated at the user's request. In contrast to project termination is new system failure.

There are many reasons a new system does not meet user requirements:

User requirements were not clearly defined or understood. The user was not directly involved in the crucial phases of system development.

- ❑ The analyst, programmer, or both were inexperienced.
- ❑ The systems analyst (or the project team) had to do the work under the stringent time constraints. Consequently, not enough thought went into the feasibility study and system design.
- ❑ User training was poor.
- ❑ Existing hardware proved deficient to handle the new application.
- ❑ The new system left users in other departments out of touch with information that the old system had provided.
- ❑ The new system was not user-friendly.
- ❑ Users changed their requirements.
- ❑ The user staff was hostile.

The list can be expanded to include many more causes. The important point is that although advances in computer systems and software make life easier for the analyst, the success of the system project depends on the experience, creative ability, and knowledge of the analyst and the support from the user staff. This suggests that the analyst be skilled in the state of the art (hardware and software) as well as in dealing with people.

## **23. SCOPE OF IMPROVEMENT**

This project can be used in the Aravli School Library with adding some more useful modules in the project.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the project.

- A module is modifiable without affecting other modules.
- Integration of modules as per requirement.
- Can be implemented on the internet.
- Extendable (addition of a module).

## **24. CONCLUSION**

As we have decided this project, we are sure the problems in the existed system would overcome. The “**LIBRARY MANAGEMENT SYSTEM**” process is computerized to reduce human errors and to increase the efficiency. The main focus of this project is to lessen human efforts. The maintenance of the records is made efficient, as all the records are stored in the ACCESS database, through which data can be retrieved easily. The navigation control will be provided in all the forms to navigate through the large amount of records. If the numbers of records are very large then user has to just type in the search string and user gets the results immediately. The editing is also made simpler. The user

has to just type in the required field and press the update button to update the desired field.

The problems, which existed in the earlier system, have been removed to a large extent. And it is expected that this project will go a long way in satisfying user's requirements. The computerization of the Library Supportive System will not only improve the efficiency but will also reduce human stress thereby indirectly improving human resources.

## **25. REFERENCES**

Fouty, Kathleen G.

**Implementing an automated circulation system.**

New York: Neal-Schumann

Publisher, Inc. 1994.

Cohn, John M., Kelsey, Ann L., and Fiels, Keith Michael.

**Planning for automation.**

New York: Neal-Schumann Publisher, Inc. 1992.

TEREESA HENNIG, ROB COOPER, GEOFFREY L. GRIFFITH.

**access 2007 programmer's reference.**

Wrox publisher.

PERRY (Grey) and HETTIHEWA (Sanjava) Sams.

**TEACH YOURSELF VISUAL BASIC 6 IN 24 HOURS.**

TECHMEDIA, NEW DELHI.