PROJECT REPORT

ON

LIBRARY MANAGEMENT SYSTEM

FOR

SUPER BRILLIANT PVT LTD.

BY

DINESH SAKHARAM THANGE

UNIVERSITY OF PUNE

MASTER OF COMPUTER APPLICATION

M.E.S's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411029

2012-13

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am more than pleasure to thank **Dr. V. H. Inamdar (Director IMCC), Mrs. ManasiBhate (Training & Placement Officer IMCC), Dr. SantoshDeshpande (Head Of The Department, IMCC)** and **Mrs. AshwiniPatil (Co-ordinator, MCA)** for giving me such an important opportunity. I express my sincere gratitude to Mrs. ManasiBhate for their valuable time and constant support. I thank all the teaching and non - teaching staff of IMCC without whom the completion of this project in time would have been a remote possibility.

Dinesh Thange

INDEX

	Chapters	Page No.
	1:INTRODUCTION	
1.1	Company Profile	1
1.2	Existing System and Need for System	3
1.3	Scope of Work	5
1.4	Operating Environment- Hardware and software	10
1.5	Details Description of Technology Used	11
	2:PROPOSED SYSTEM	
2.1	Proposed System	23
2.2	Objective of System	24
2.3	User Requirements	25
	3:ANALYSIS & DESIGN	
3.1	Data Flow Diagram(DFD)	28
3.2	Functional Decomposition Diagram (FDD)	33
3.3	Entity Relationship Diagram(ERD)	34
3.4	Data Dictionary	35
3.5	Table Design	37
3.6	Code Design	42
3.7	Menu Tree	44
3.8	Menu Screen	45
3.9	Input Screen	50
3.10	Report Format	55
3.11	Test Procedures And Implementation	58

	4:USER MANNUAL	
4.1	User Manual	66
4.2	Operations Manual	68
4.3	Program Specification/Flow Charts	
4.4	Forms and Report Specifications (For Commercial Project)	69
	Drawbacks and limitation	71
	Proposed Enhancements	72
	Conclusions	73
	Bibliography (Write Web & Book Bibliography separatly)	
	ANNEXURES :	
	ANNEXURE 1 : USER INTERFACE SCREENS	
	ANNEXURE 2 : OUTPUT REPORTS WITH DATA (Data in forms must be reflected in report) (If Report or form contains any calculation field check the calculation)	
	ANNEXURE 3 : SAMPLE PROGRAM CODE (which will prove sufficient development is done by the student)	

1.1 Company Profile:

Super Brilliant Pvt Ltd is an emerging organization with a vision to tap the untapped mid and low-end market. It has expertise in consulting, technology and end to end solutions for different verticals. We have excellent exposure this market segment and we understand need of ‘No Frill’ quality solutions. We aim to develop customized IT solutions to match the requirements.

Super Brilliant to equip clients with most futuristic of IT tools for information infrastructure, application and solutions, needed to respond to business dynamics. It believes in ever evolving market scenario.

Vision

To aim to be leading global player with unique products proposition working towards total customer satisfaction through customer oriented service.

Mission

To design and deliver value added solution and support to improve productivity of businesses at an absolutely affordable price.

Quality Policy

We at Super Brilliant synchronize our efforts to build quality into the products and services that we deliver to internal as well as external customers. We aim to nurture an environment that creates sense of ownership for value creation.

Values

Growth at Super Brilliant is driven by the values that we stand by

1. To attract and nurture best of talent.
2. To create boundary free environment by including flexibility.
3. To promote research and development through industry-institution partnership.

4. To provide customer oriented quick response support mechanism for total customer satisfaction.
5. To provide conductive atmosphere to boost the respect for individuals.

1.2 Existing System Need for System:

Various problems of physical system are described below:-

- If one is not very careful then there is a possibility of issuing more than one book to a user.
- There is a possibility of issuing a book to a user, whose membership is not there.
- When a user requests for the a book, one has to physically check for the presence of a book in the library Answering management query is a time consuming process.
- Daily keeping a manual record of changes taking place in the library such as book being issued, book being returned etc can become some if the Library size is bigger.

1.3 Scope of work:

Generally this project is divided into ten main Modules.

- **Login Module**
- **Books Module**
- **Barrowers Module**
- **Search Module**
- **Returning Module**
- **Barrower Module**
- **Course Module**
- **Category Module**
- **Help**

Login Module:

This module is just an entry point, through which any user/employee can log into the system and can operate on the system provided they know the **Administrator name**and **password**. In order to log into the system, user/employee has to type the **Administrator name**and**password**in the field provided.

Books Module:

These modules basically focuson storing the database information of available books, new books, Category books, etc. This module contain following feature

a) Add books
b) Edit books
c) Search books
d) List available books
e) Remove books

Borrowers Module:

This module basically focus on storing the database information of available Borrowers, Add Borrowers, Edit, Searchetc. This module contain following feature

a)AddBorrowers

b) List all Borrowers

c) Edit Borrowers

d) Remove Borrowers

e)Borrowers Information

Search Module:

This utility provide an efficient mechanism for searching different entry on database its provide fast and quick result accession. We can search the detail by using different categories like

a) Search by bookID.

b) Search by BorrowersID

Returning Module:

The Whole process for returning the Books on candidate side is maintained by this module. Using for following features

a) Search Book

b) Previous Book

c) Refresh Book

d) Cancel Book

Borrower Module:

The whole process for issuingthe books on candidate side is maintain by this module. Using for following features

a)Add Books

b) Edit Books

c) Return Book

d) Search Book

e) Remove Book

f) Refresh Book

g) Cancel Book

Course Module:-

This Module process to save the information in Course wise data. Using for following features.

a) Add new

b) Edit Course

c) Search Course

d) Remove Course

Category Module:-

This module process to save the information in Category wise data. Using following features.

a) Add new

b) Edit Category

c) Remove Category

d) Search Category

Help Module:

All the basic information which improve usability of application for user which make application user friendly are available on help module. Its provide instants help to user.

1.4 Operating Environment- Hardware and Software

Hardware Requirements:-

- Pentium-IV(Processor).
- 256 MB Ram
- Hard disk 40 GB

Software Requirements: -

- **Operating System :** Windows 7
- **Programming language:** Java
- **Database:** MS Access2007

1.5 Detail Description of technology Used

Java Features:-

Platform Independent

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is more closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

Simple

There are various features that makes the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we cannot say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory allocation and deal location system.

Object Oriented

To be an Object Oriented language, any language must follow at least the four characteristics.

Inheritance :It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.

Encapsulation: It is the mechanism of combining the information and providing the abstraction.

Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the

Functions having the same name based on the signatures of the methods.

Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfills the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java, it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand-alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

Robust

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features make the java language robust.

Distributed

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

Portable

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. This feature makes the java as a portable language.

Dynamic

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

Secure

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The byte code Verifier checks the classes after loading.

Performance

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

Multithreaded

As we all know several features of Java like Secure, Robust, Portable, dynamic etc; you will be more delighted to know another feature of Java which is Multithreaded.

Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer.

Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

Interpreted

We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code.

The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program.

The versatility of being platform independent makes Java to outshine from other languages. The source code to be written and distributed is platform independent.

Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

This project is mainly developed using the Java Swing. There is some introduction about Java Swing.

Java swing:-

To create a Java program with a graphical user interface (GUI), you'll want to learn about Swing.

The Swing toolkit includes a rich set of components for building GUIs and adding interactivity to Java applications. Swing includes all the components you would expect from a modern toolkit: table controls, list controls, tree controls, buttons, and labels.

Swing is far from a simple component toolkit, however. It includes rich undo support, a highly customizable text package, integrated internationalization and accessibility support. To truly leverage the cross-platform capabilities of the Java platform, Swing supports numerous look and feels, including the ability to create your own look and feel. The ability to create a custom look and feel is made easier with Synth, a look and feel specifically designed to be customized. Swing wouldn't be a component toolkit without the basic user interface primitives such as drag and drop, event handling, customizable painting, and window management.

Swing is part of the Java Foundation Classes (JFC). The JFC also include other features important to a GUI program, such as the ability to add rich graphics functionality and the ability to create a program that can work in different languages and by users with different input devices.

The following list shows some of the features that Swing and the Java Foundation Classes provide.

Swing GUI Components

The Swing toolkit includes a rich array of components: from basic components, such as buttons and check boxes, to rich and complex components, such as tables and text. Even deceptively simple components, such as text fields, offer sophisticated functionality, such as formatted text input or password field behavior. There are file browsers and dialogs to suit most needs, and if not, customization is possible. If none of Swing's provided components are exactly what you need, you can leverage the basic Swing component functionality to create your own.

Java 2D API

To make your application stand out; convey information visually; or add figures, images, or animation to your GUI, you'll want to use the Java 2D API. Because Swing is built on the 2D package, it's trivial to make use of 2D within Swing components.

Adding images, drop shadows, compositing — it's easy with Java 2D.

Pluggable Look-and-Feel Support

Any program that uses Swing components has a choice of look and feel. The classes shipped by Oracle provide a look and feel that matches that of the platform. The Synth package allows you to create your own look and feel. The GTK+ look and feel makes hundreds of existing look and feels available to Swing programs.

A program can specify the look and feel of the platform it is running on, or it can specify to always use the Java look and feel, and without recompiling, it will just work. Or, you can ignore the issue and let the UI manager sort it out.

Accessibility API

People with disabilities use special software — assistive technologies — that mediates the user experience for them. Such software needs to obtain a wealth of information about the running application in order to represent it in alternate media: for a screen reader to read the screen with synthetic speech or render it via a Braille display, for a screen magnifier to track the caret and keyboard focus, for on-screen keyboards to present dynamic keyboards of the menu choices and toolbar items and dialog controls, and for voice control systems to know what the user can control with his or her voice. The accessibility API enables these assistive technologies to get the information they need, and to programmatically manipulate the elements that make up the graphical user interface.

Undo Framework API

Swing's undo framework allows developers to provide support for undo and redo. Undo support is built in to Swing's text component. For other components, Swing supports an unlimited number of actions to undo and redo, and is easily adapted to an application. For example, you could easily enable undo to add and remove elements from a table.

Flexible Deployment Support

If you want your program to run within a browser window, you can create it as an applet and run it using Java Plug-in, which supports a variety of browsers, such as Internet Explorer, Firefox, and Safari. If you want to create a program that can be launched from a browser, you can do this with Java Web Start. Of course, your application can also run outside of browser as a standard desktop application.

2.1 Proposed System

Proposed system is an automated Library Management System. Through our software user can add members, add books, search members, search books, update information, edit information, borrow and return books in quick time. Our proposed system has the following advantages.

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Look and Feel Environment
- Quick transaction

All the manual difficulties in managing the Library have been rectified by implementing computerization.

2.2 Objectives of System

With the growing Information Technology industry, Automation of their system and management is desired by all kind of commercial enterprises.

Library Management System maintains the record of Books in the library, issue and return process of the books in the library. Here we are primarily concerned with management of books of library.

- To maintain details of the books, borrower etc.
- To maintain proper record of the all borrowed books.
- If we want to find out the exact information about books if it issued or not etc.
- If due date is greater than the date than that records automatically added to the due books table.
- After returning the book that record goes to returned book table.
- To provide the short cut keys for all things like showing the user using ctrl + U, borrowers records using ctrl + B etc.

2.3 User Requirements

The System is made for the prospective borrowers method is followed during system design. Following user requirements were outlined during the Library.

Borrowers Information Entry

- System should provide Borrowers Information Entry form.
- Borrowers id should be automatically generated by the system.

Book Information Entry

- System should provide separate facility for addition, updation of books.
- Provision should be made for automatic book id generation.
- Addition of book should keep control on validity of publisher and author id.

Borrow Book Entry

- There should be transaction form for borrowing the book.
- There should be a check that one Borrowers can only borrow one book.
- System should display current date for borrow date
- Expected return date should be automatically generated by syetem.

Book Return Entry

- The System should provide a book return entry form.
- Actual return date should be generated by the system automatically.

Search Facility

- System should provide search facility to navigate the master information.

Fine Pay Entry

- There should be form for fine pay entry.
- System should generate payment date automatically.

Course Entry

- System should provide Course facility to related to master information .
- There should be form for Course Entry.

Category Entry

- System should provide Category of book in which is yours admitted.
- There should be form for Category Entry.

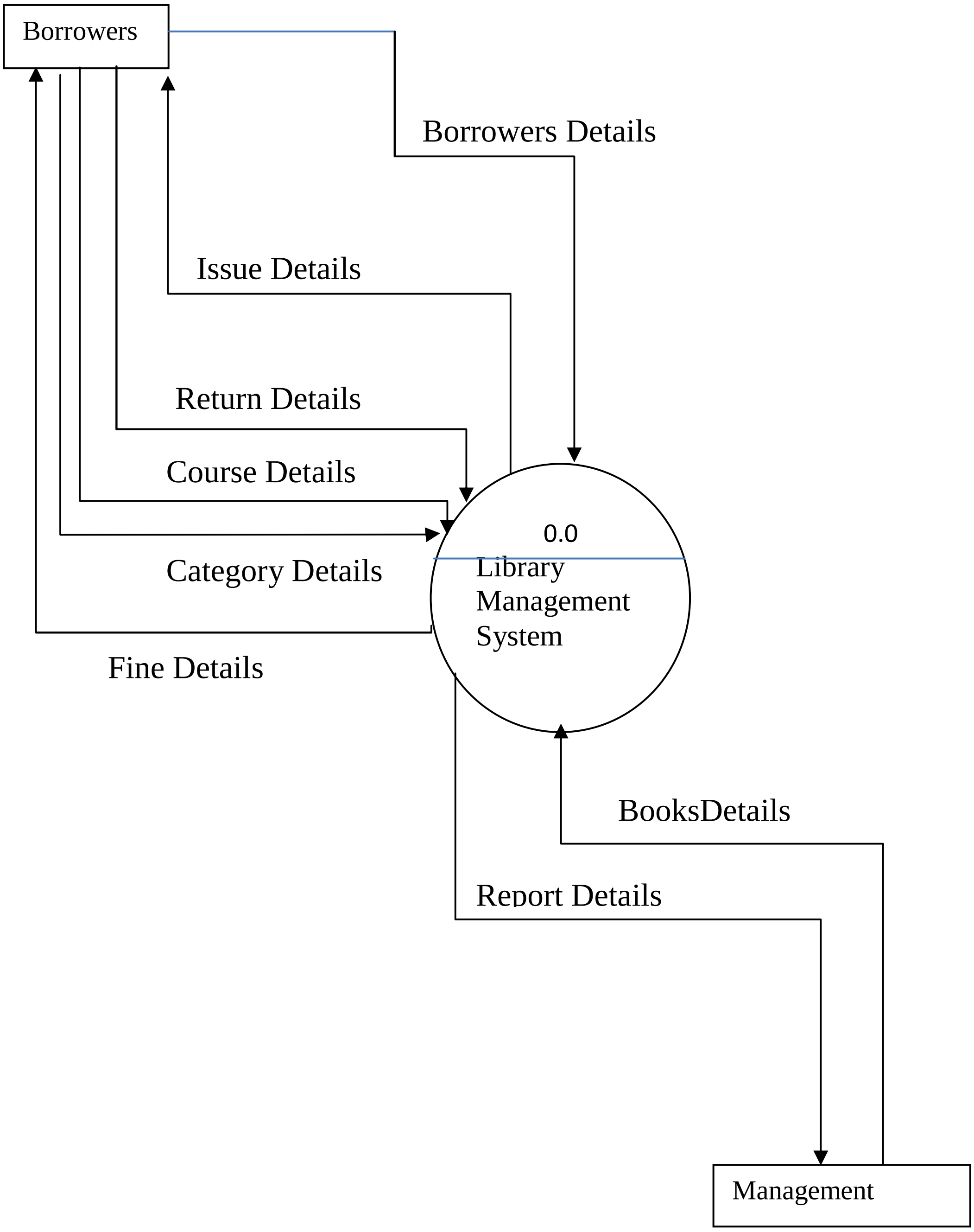
Report Generation

System should generate following reports

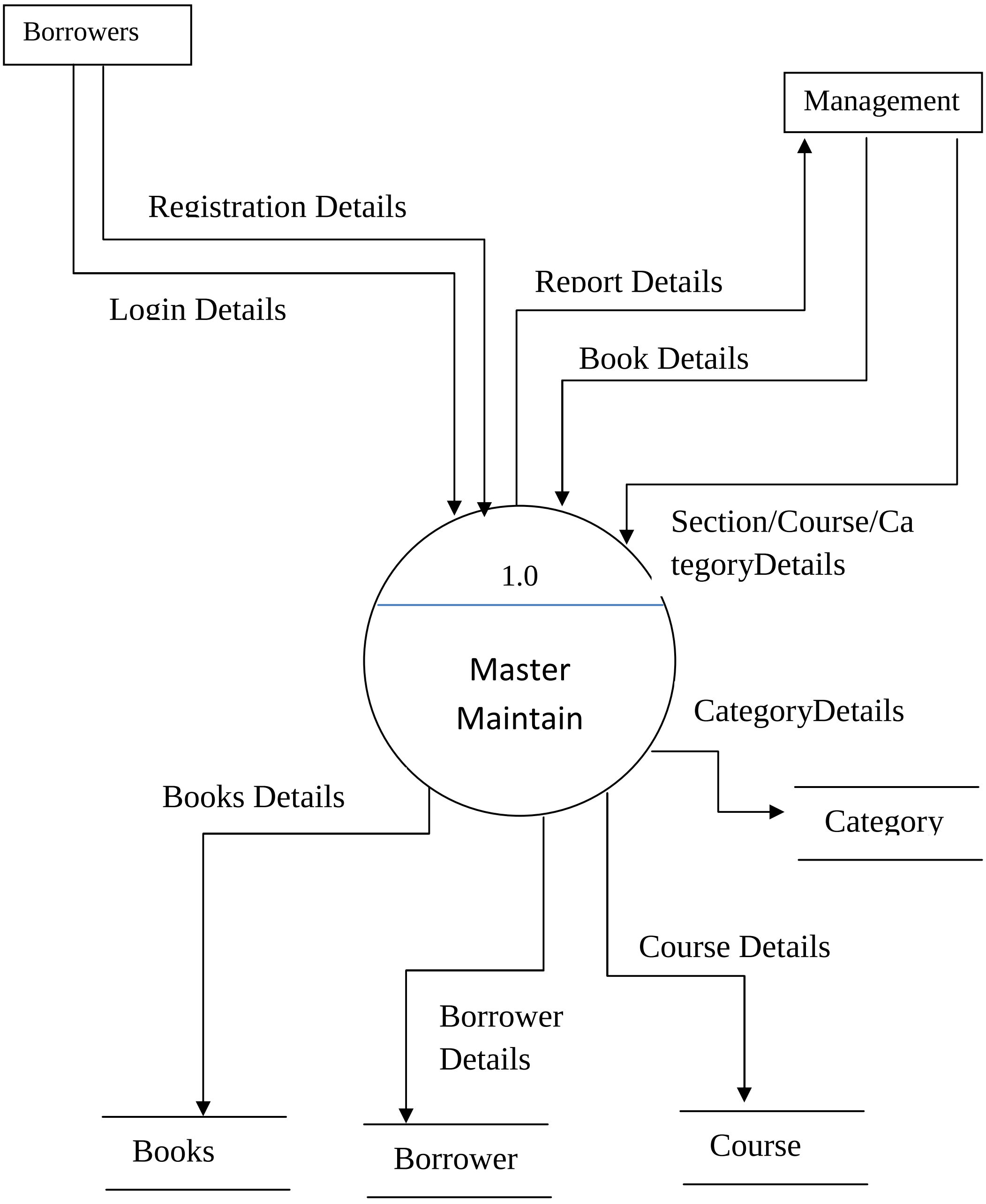
- All Borrowers List
- All Books List
- Available Books List
- All Return Books List
- All Borrow Book List
- List of Course wise
- List of Category wise

3.1 Data Flow Diagram(DFD)

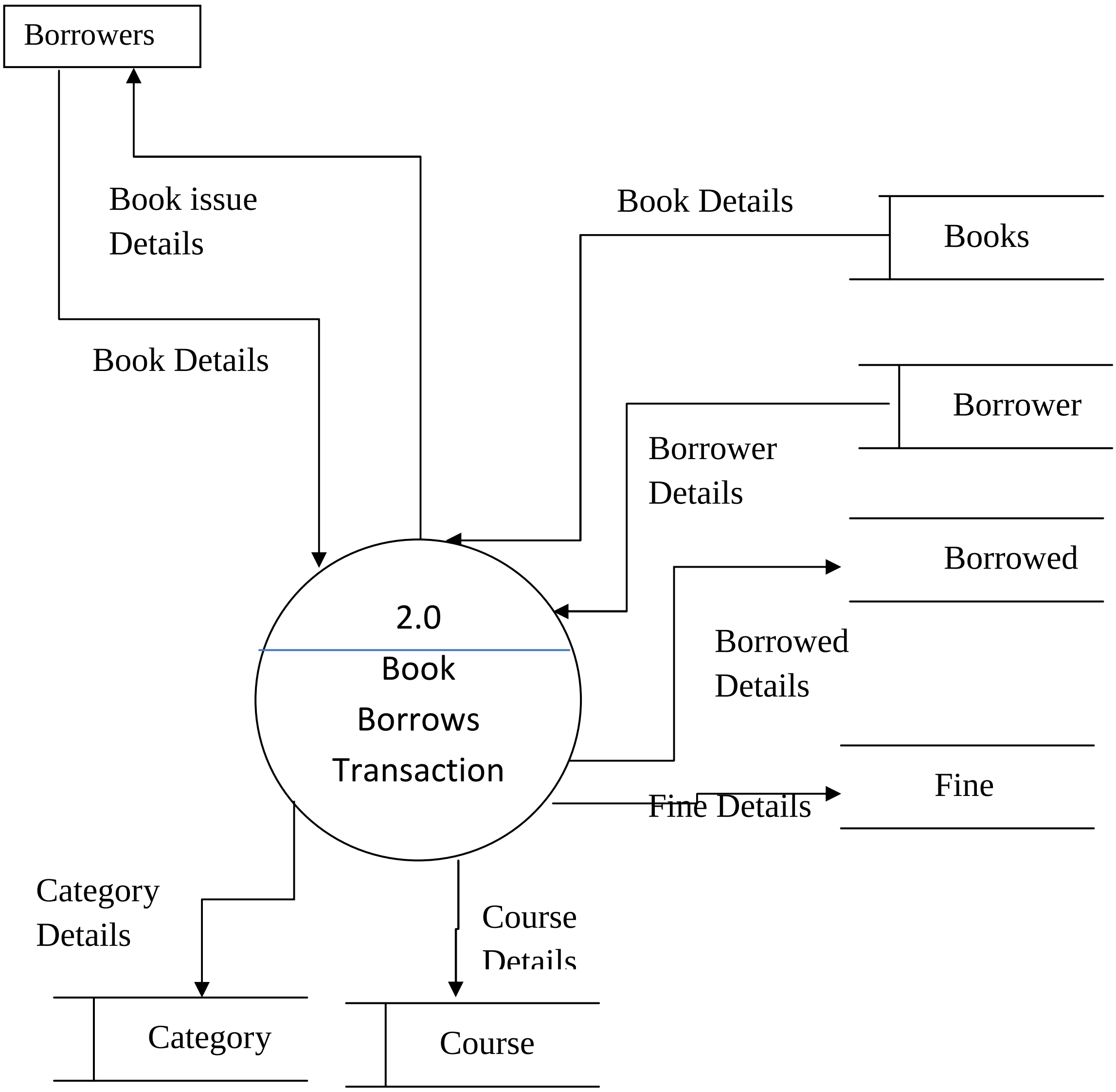
1) Context Level Diagram



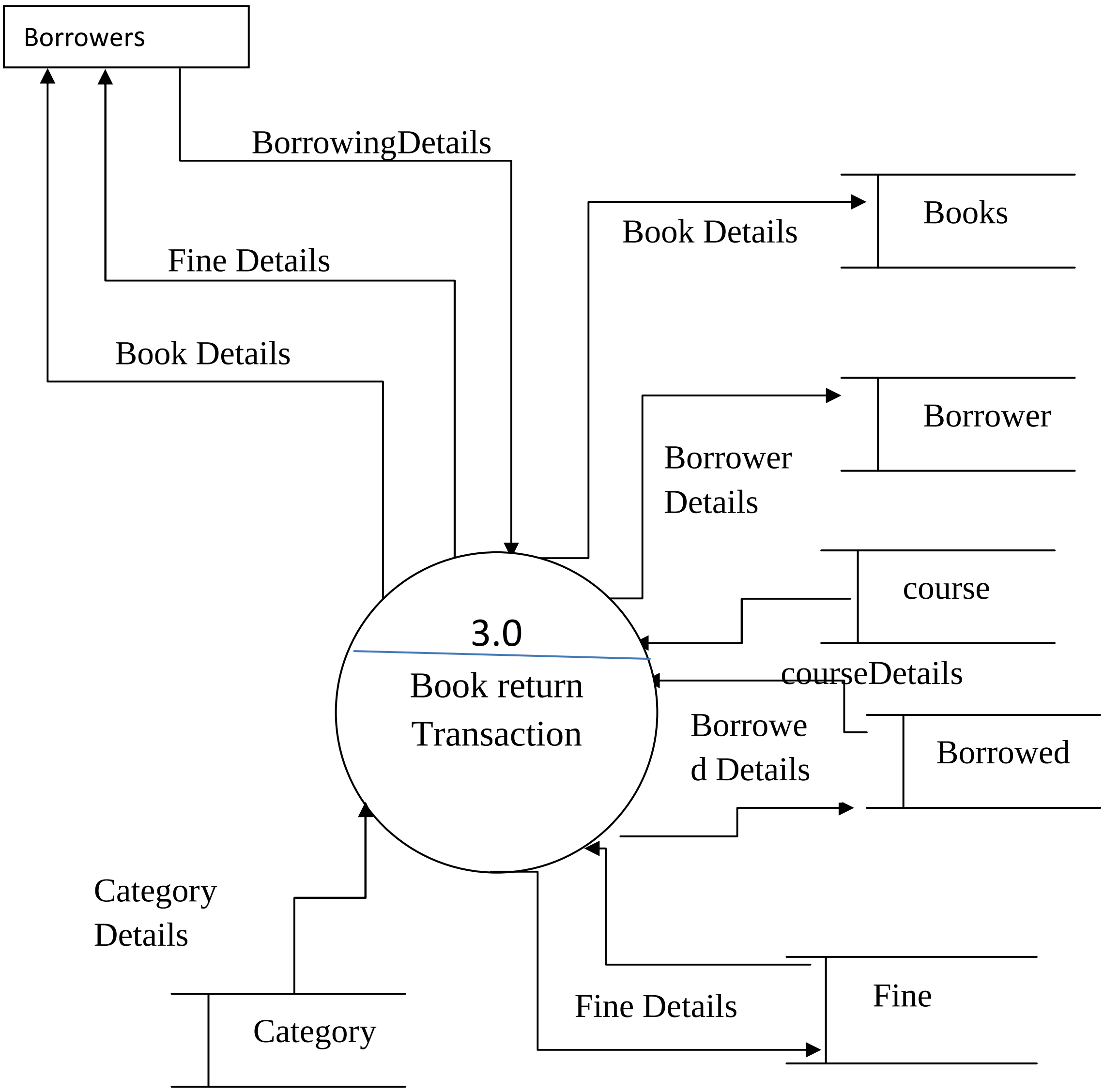
2)First Level Data Flow Diagram



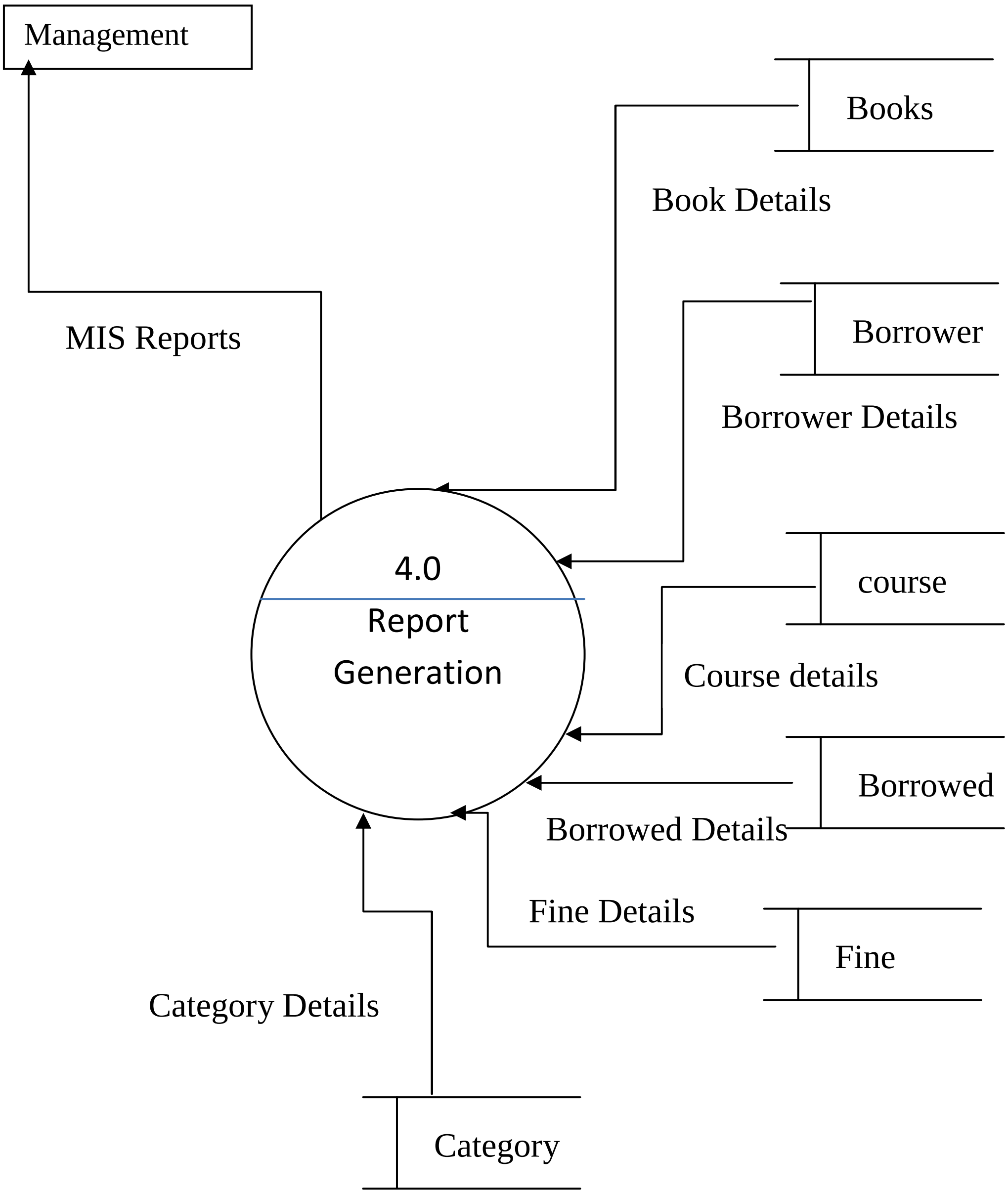
2) 2.0 Level Data Flow Diagram



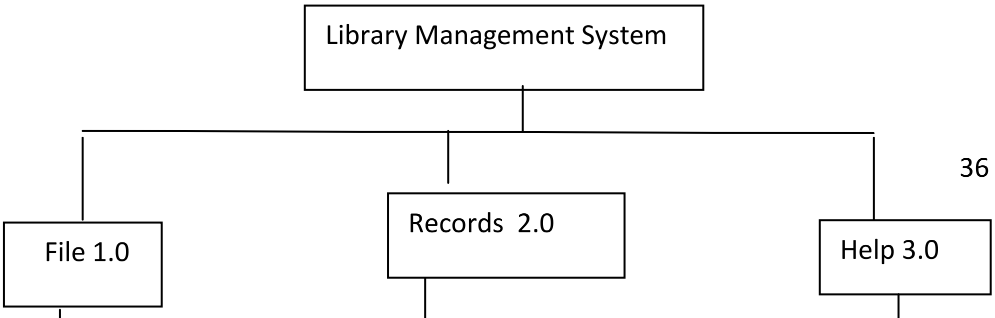
3) 3.0 Level Data Flow Diagram



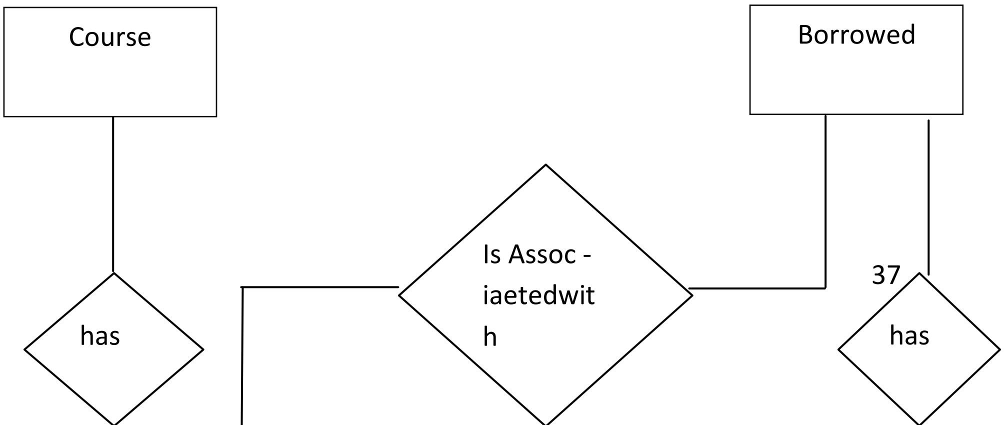
4) 3.0 Level Data Flow Diagram

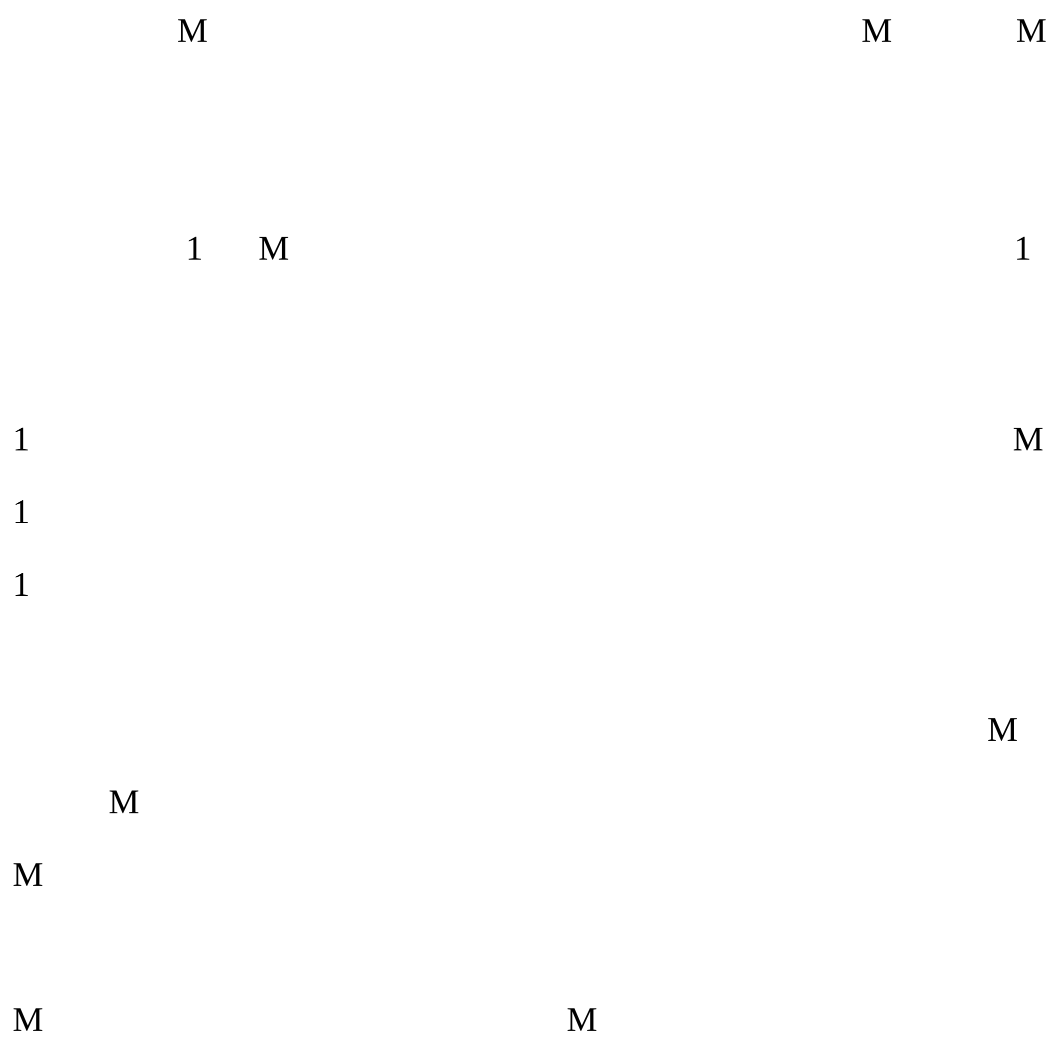


3.2 Functional Decomposition Diagram(FDD)



3.3 E-R Diagram





3.4 Data Dictionary

Sr.No	Field Name	Description	Datatype	Size
1	Address	Barrowers Address	Text	20

2	Author	Author Name	Text	40
3	BarrowerID	BarrowerId	Number	5
4	BarrowersName	Barrowers Name	Text	20
5	Book_No	Book_No	Number	5
6	Category	Name of Category	Text	20
7	CategoryID	Category	Text	20
8	Course	Name of Course	Text	10
9	CurrentYear	Year of Barrowers	Text	10
10	DateBarrowed	Date Barrowed Books	Boolean	Yes/No
11	DateDue	Date Due to Books	Boolean	Yes/No
12	DateReturned	Date in Return books	Boolean	Yes/No
13	Description	Number of Category	Text	30
14	Fine_id	Fine id	Number	5
15	Fines	Fine in Books	Number	5
16	ISBN	ISBN_No	Number	20
17	NO due date	No of days after due date	Number	5
18	Password	User Password	Text	20
19	Price	Book Price	Number	5

20	Quantity	Number of Books	Number	5
21	Remaining	Available or not	Boolean	Yes/No
22	Section	Number of Section	Number	5
23	Title	Title of Book	Text	40
24	UserName	Name of User	Text	20
25	Year	Name of Year	Text	10
26	Year Publisher	Publisher Name	Text	40

3.5 Table Design

Book

Field Name	Description	Data Type	Size	Constraint
BookNo	Book_No	Number	5	Primary key
ISBN	ISBN_No	Number	20	
Title	Title of Book	Text	40	
Author	Author Name	Text	40	
Year Publisher	Publisher Name	Text	40	
CategoryID	Category	Text	20	
Price	Book Price	Number	5	
Quantity	Number of Books	Number	5	
Remaining	Available or not	Boolean	Yes/ No	

BarrowedBooks(Return Books)

Barrowers				
Field Name	**Description**	**Data Type**	**Size**	**Constraint**
BarrowerID	BarrowerId	Number	5	Primary key
BarrowersName	Barrowers Name	Text	20	
Address	Barrowers Address	Text	20	
CurrentYear	Year of Barrowers	Text	10	
Course	Name of Course	Text	10	
Section	Number of Section	Number	5	

Field Name	**Description**	**Data Type**	**Size**	**Constraint**
BookNo	BookNo	Number	5	Primary Key
BarrowersID	BarrowersId	Number	5	Foreign key
DateBarrowed	Date Barrowed Books	Boolean	Yes/ No	
DateDue	Date Due to Books	Boolean	Yes/ No	
DateReturned	Date in Return books	Boolean	Yes/ No	
NO due date	No of days after due date	Number	5	
Fines	Fine in Books	Number	5	

Category				
Field Name	**Description**	**Data Type**	**Size**	**Constraint**
CategoryId	Book Catergory Id	Number	5	Primary key
Category	Name of Category	Text	20	
Description	Number of Category	Text	30	

Course				
Field Name	**Description**	**Data Type**	**Size**	**Constraint**
Course	Course Name	Text	20	
Year	Name of Year	Text	10	
Section	Number Section	Number	5	
Adviser	Name of Adviser	Text	20	

Fine				
Field Name	**Description**	**Data Type**	**Size**	**Constraint**
Fine_id	Fine id	Number	5	Primary key
Book_No	Book_No	Number	5	Foreign key
Fines	Rs. Fine in books	Number	5	

Users				
Field Name	**Description**	**Data Type**	**Size**	**Constraint**
UserName	Name of User	Text	20	
Password	User Password	Text	20	

3.6 Code Design

Table Name: Books.
Primary Key: Book_No
Data Type: Number
Width: 5
Design: Unique book registration number
Description: For uniquely identification of book details

Example: 2

Table Name: Borrowers.
Primary Key: Borrowers_Id
Data Type: Number
Width: 5
Design: Unique Borrowers identity
Description: For uniquely identification of Borrowers details

Example: 1111

Table Name: Borrow.
Primary Key: Borrow_Id
Data Type: Number
Width: 5
Design: Unique Borrow (issue) Books
Description: For uniquely identification of Borrow details

Example: 3

Table Name: Category.
Primary Key: CategoryID
Data Type: Number
Width: 5
Design: Unique Category registration
Description: For uniquely identification of Category details

Example: IT, Management,

Table Name: Fine.
Primary Key: Fine_id
Data Type: Number
Width: 5
Design: Unique   Fine Amount
Description: For uniquely identification of Fine details

Example: Rs5

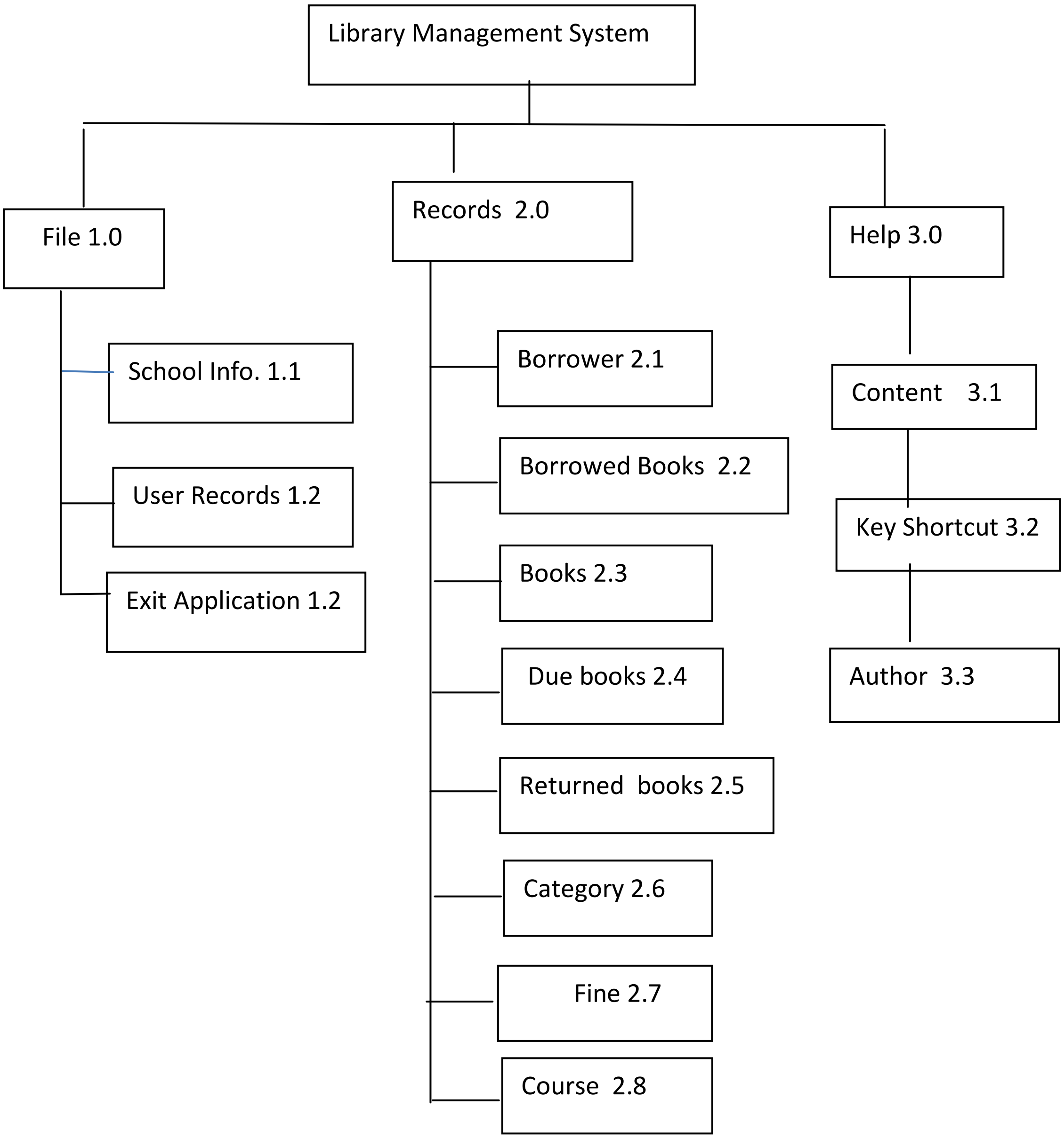
Table Name:Course.
Primary Key: Course_Id
Data Type: Number
Width: 5
Design: Unique Course registration number
Description: For uniquely identification of Course details

Example: BCA,MCA

Table Name:BorrowedBook(return).
Primary Key: BorrowedID
Data Type: Number
Width: 5
Design: Unique return registration number
Description: For uniquely identification of return details

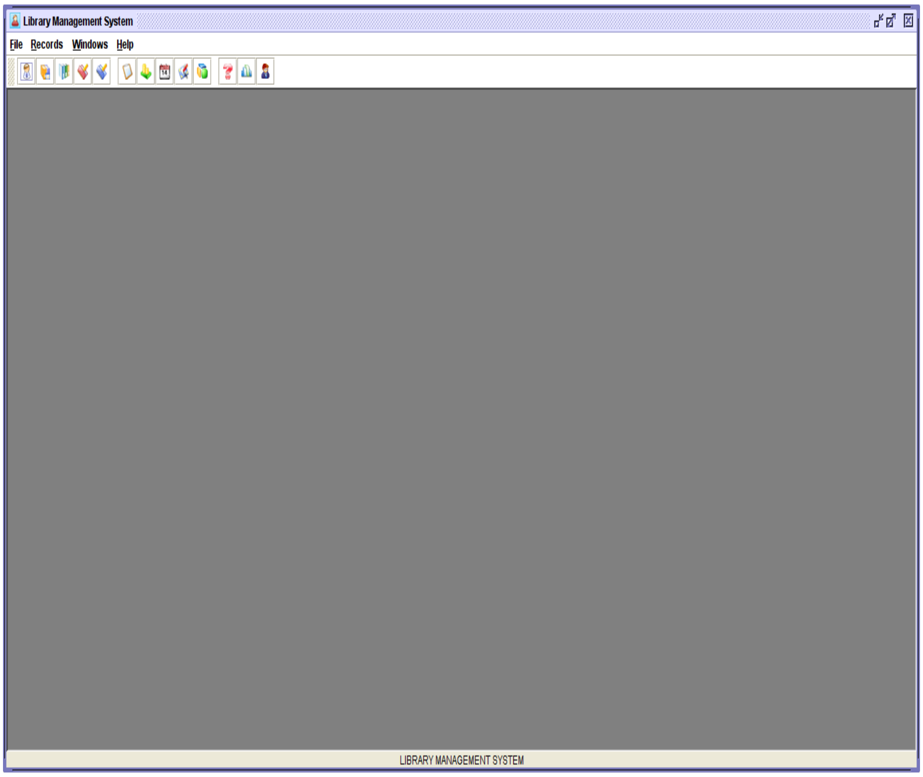
Example: 22

3.7Menu Tree:

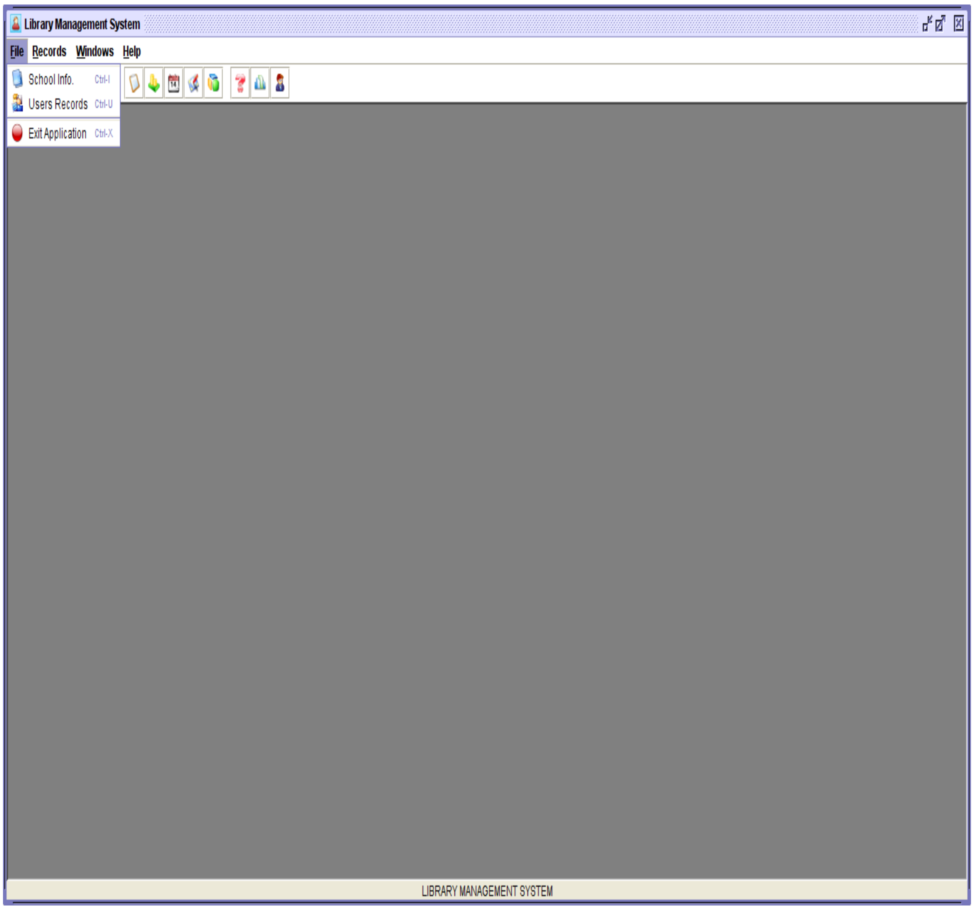


Library Management System
File 1.0
Records 2.0
Help 3.0
School Info. 1.1
User Records 1.2
Exit Application 1.2
Borrower 2.1
Borrowed Books 2.2
Books 2.3
Due books 2.4
Returned books 2.5
Category 2.6
Fine 2.7
Course 2.8
Content 3.1
Key Shortcut 3.2
Author 3.3

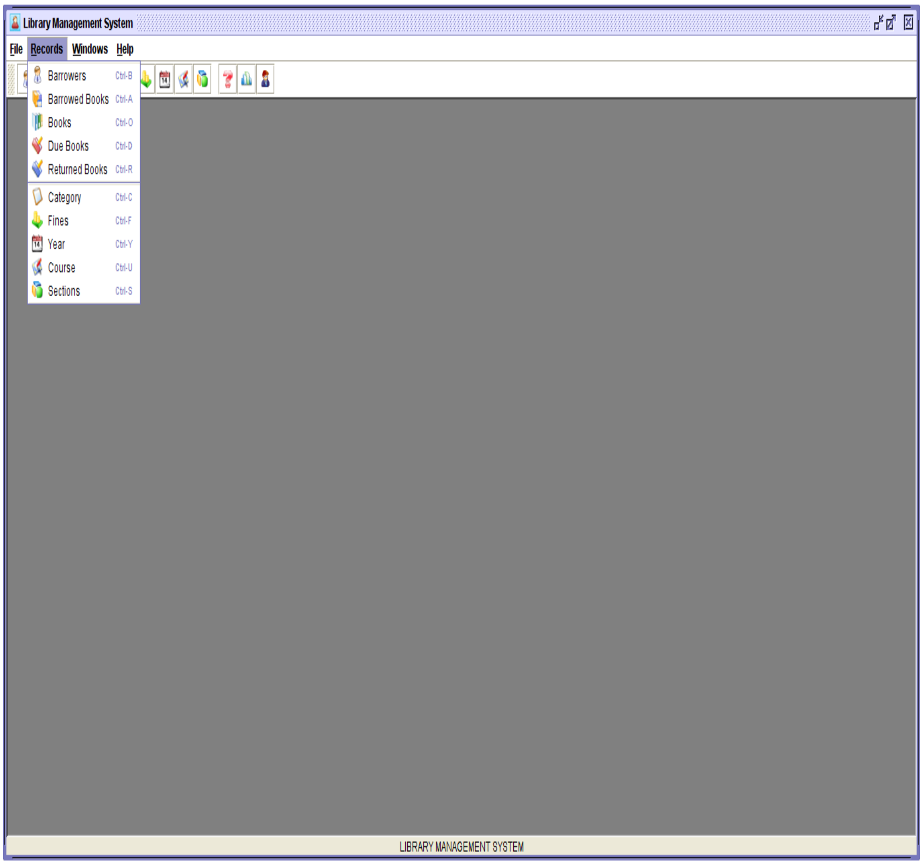
3.8 MenuScreen:



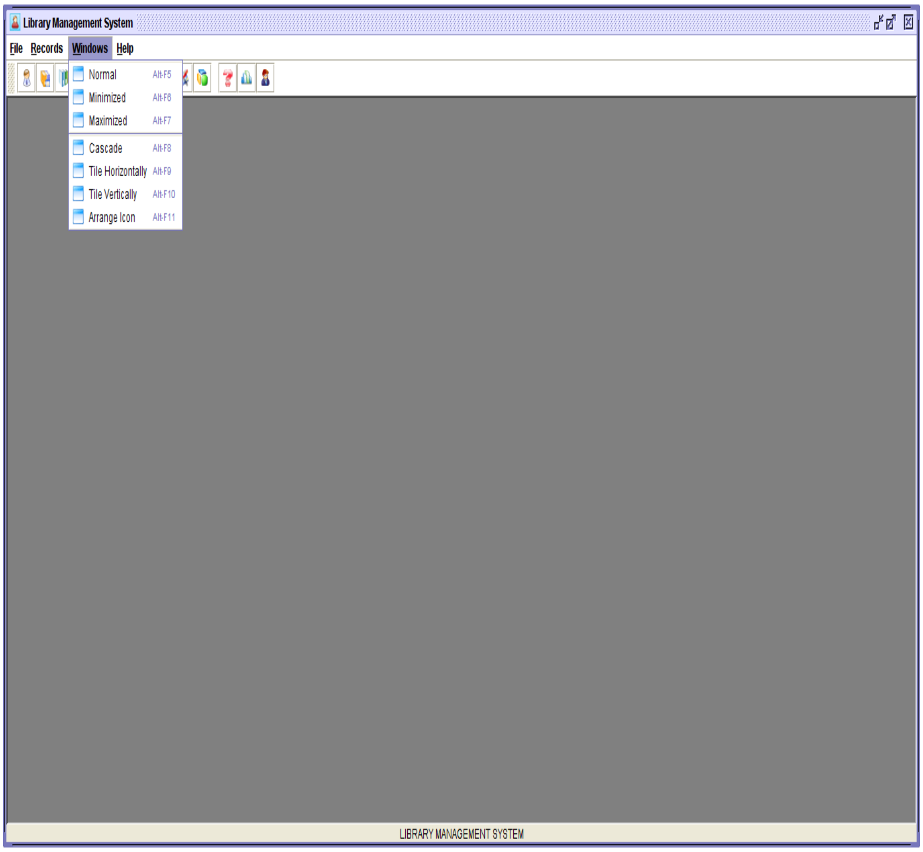
File:


Library Management System
File
Records
Windows
Help
School Info. Ctrl-I
Users Records Ctrl-U
Exit Application Ctrl-X
LIBRARY MANAGEMENT SYSTEM

Record:-

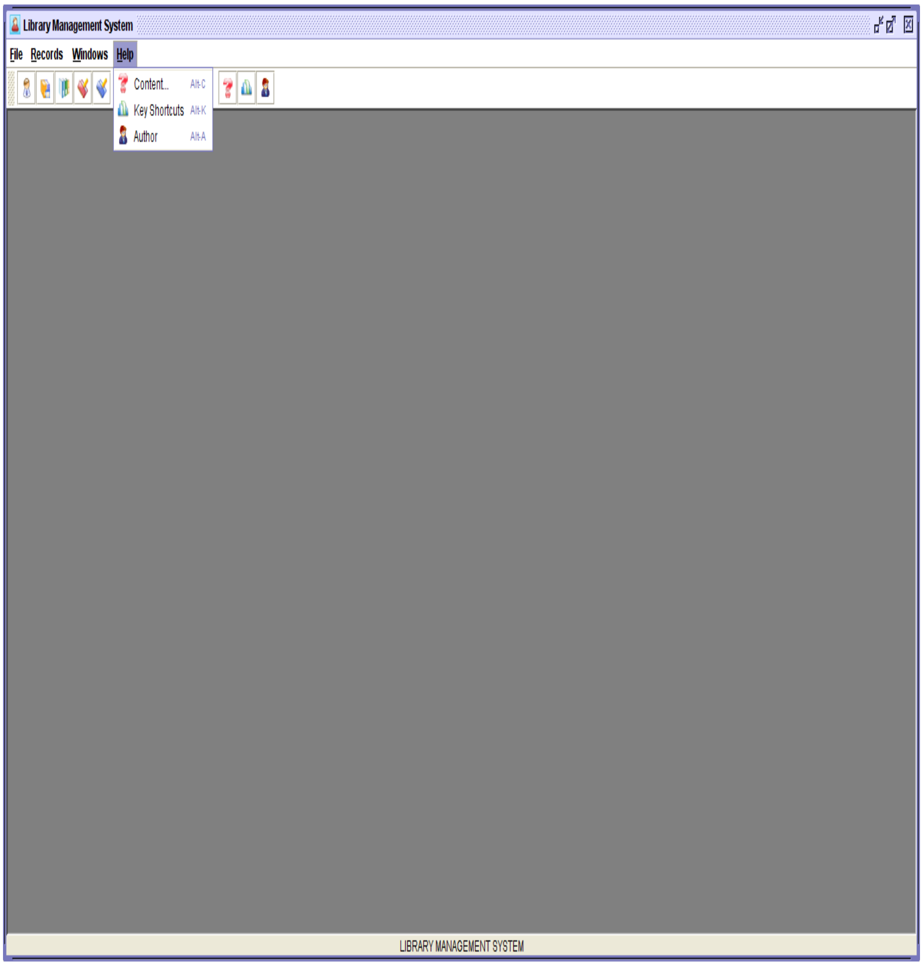

Library Management System
File
Records
Windows
Help
Barrowers Ctrl-B
Barrowed Books Ctrl-A
Books Ctrl-O
Due Books Ctrl-D
Returned Books Ctrl-R
Category Ctrl-C
Fines Ctrl-F
Year Ctrl-Y
Course Ctrl-U
Sections Ctrl-S
LIBRARY MANAGEMENT SYSTEM

Windows:

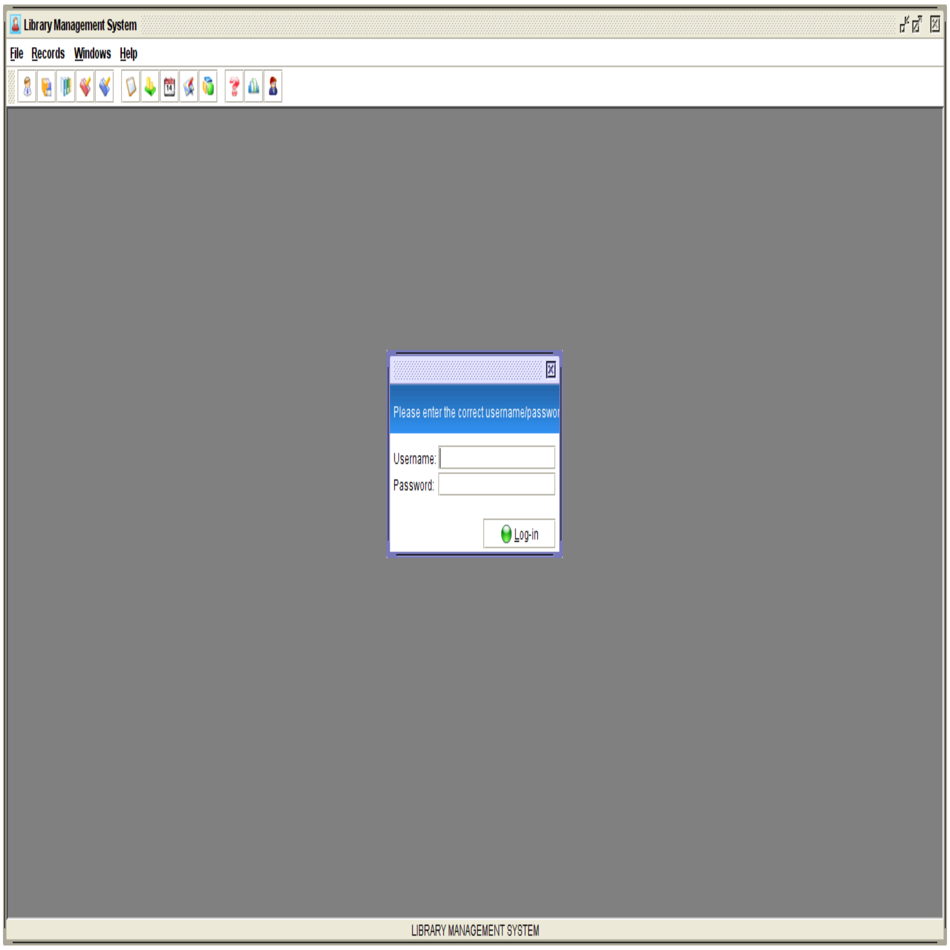


Library Management System
File
Records
Windows
Help
Normal Alt-F5
Minimized Alt-F6
Maximized Alt-F7
Cascade Alt-F8
Tile Horizontally Alt-F9
Tile Vertically Alt-F10
Arrange Icon Alt-F11
LIBRARY MANAGEMENT SYSTEM

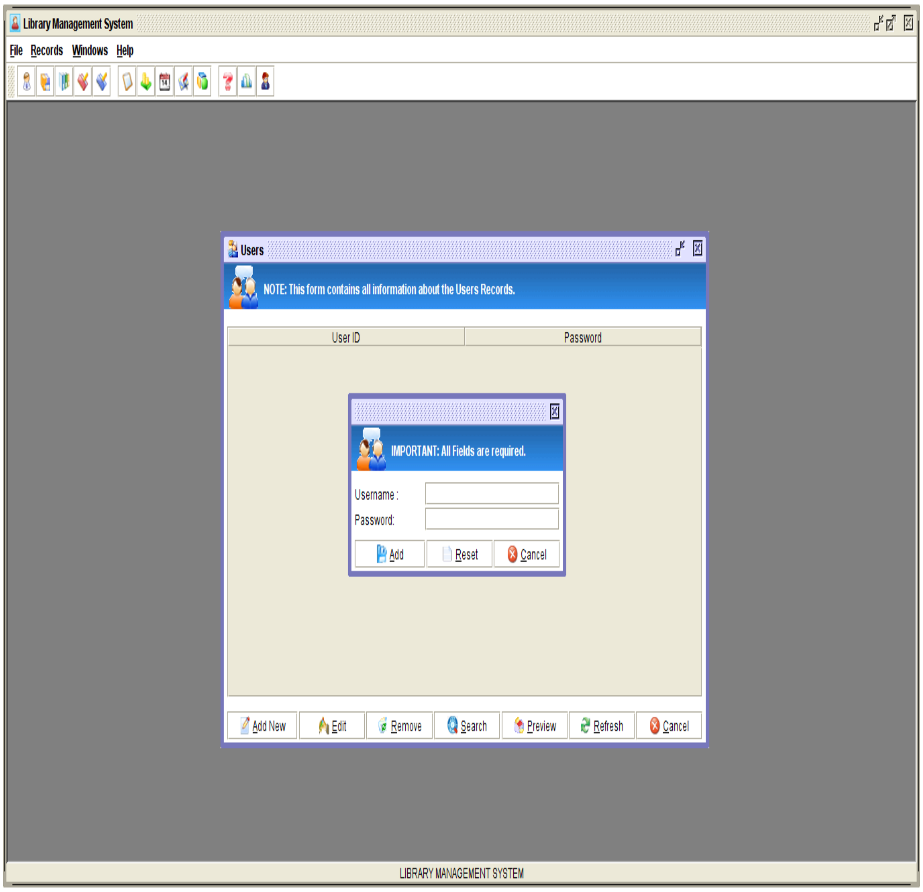
Help:


Library Management System
File
Records
Windows
Help
Content... Alt-C
Key Shortcuts Alt-K
Author Alt-A
LIBRARY MANAGEMENT SYSTEM

3.9 Input Screen:

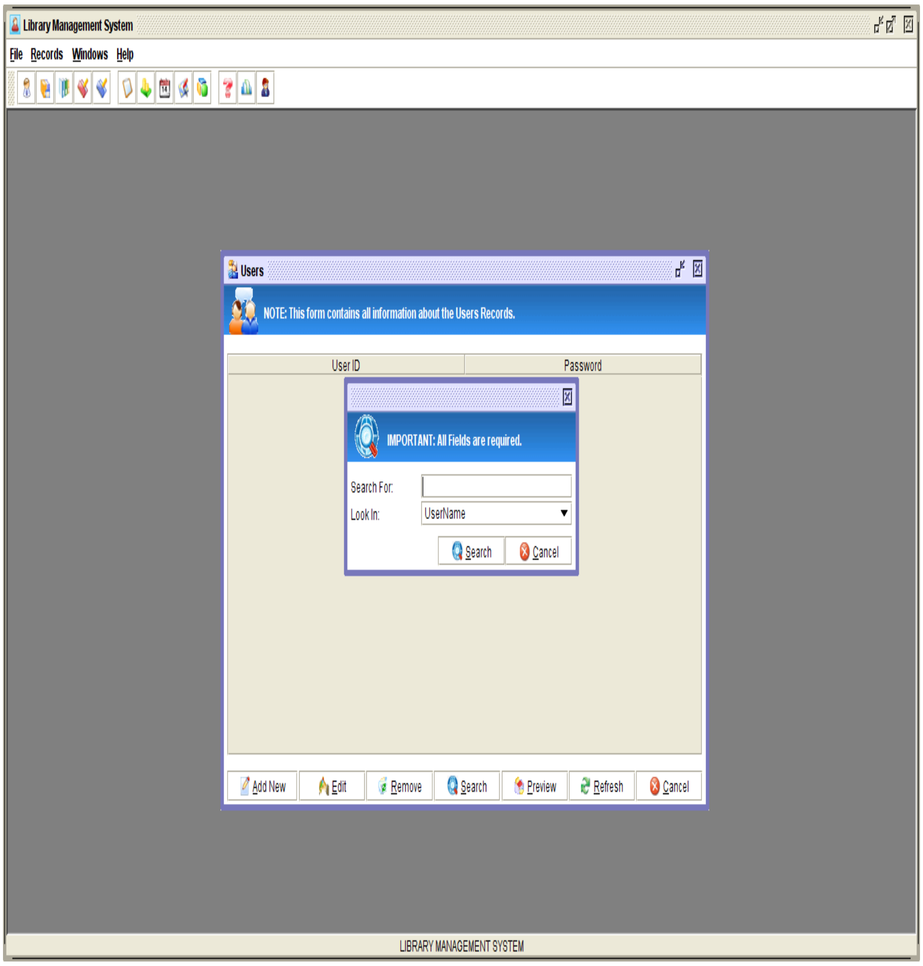


Please enter the correct username/passwor



NOTE: This form contains all information about the Users Records.

IMPORTANT: All Fields are required.



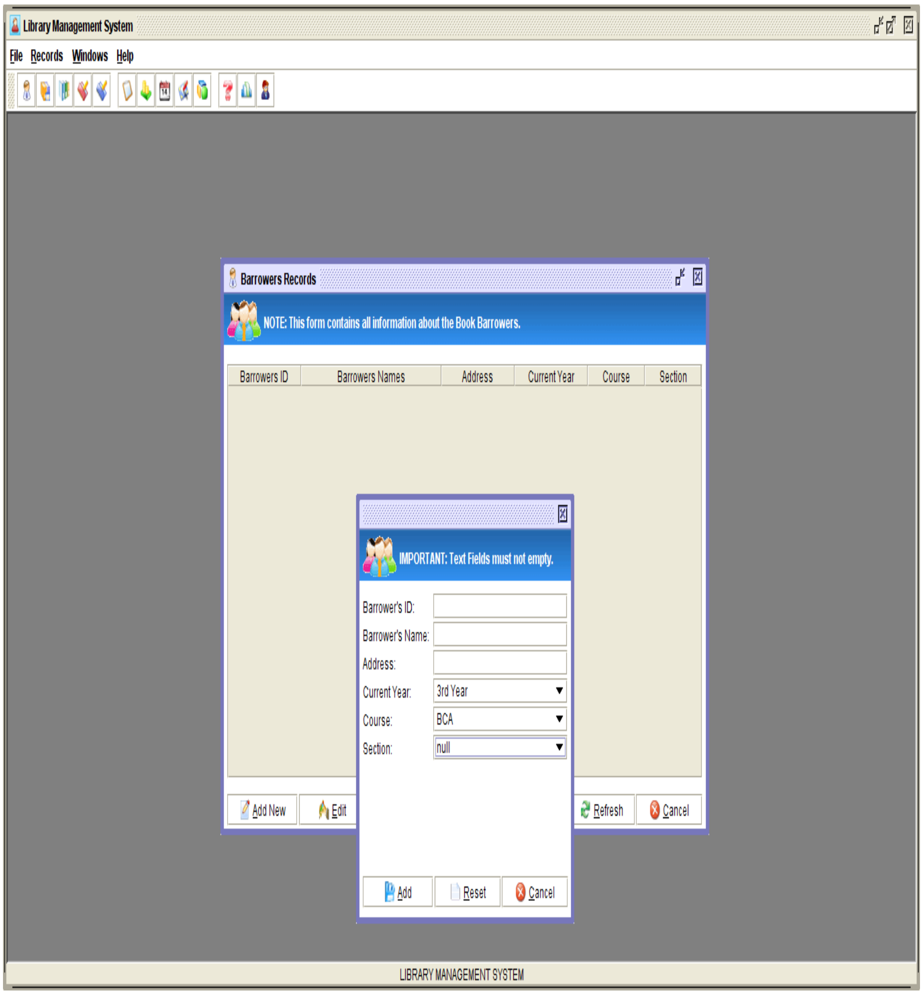
Library Management System
File Records Windows Help
Users
User ID
Password
Add New
Edit
Remove
Search
Preview
Refresh
Cancel
LIBRARY MANAGEMENT SYSTEM

NOTE: This form contains all information about the Users Records.

IMPORTANT: All Fields are required.

Search For:

Look In: UserName



NOTE: This form contains all information about the Book Barrowers.

Barrowers ID	Barrowers Names	Address	Current Year	Course	Section

IMPORTANT: Text Fields must not empty.

Barrower's ID:

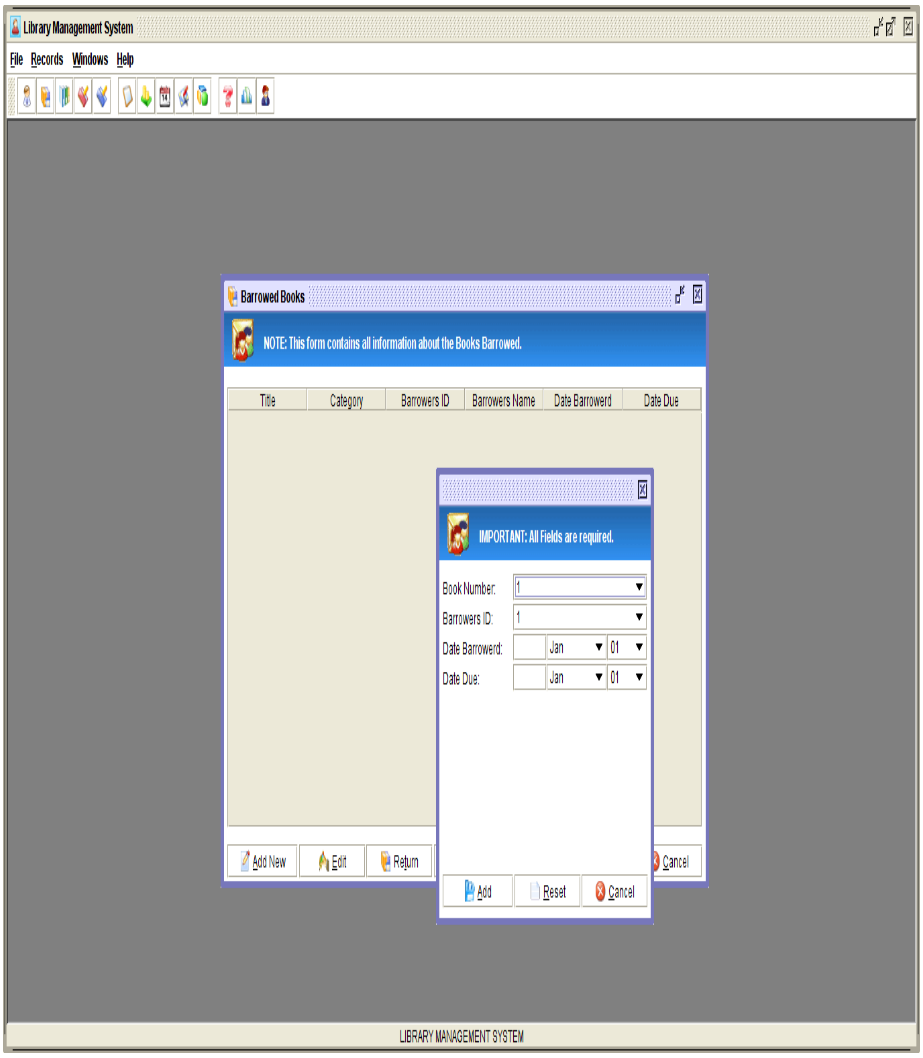
Barrower's Name:

Address:

Current Year: 3rd Year

Course: BCA

Section: null


Library Management System
File Records Windows Help
Barrowed Books

NOTE: This form contains all information about the Books Barrowed.

Title	Category	Barrowers ID	Barrowers Name	Date Barrowerd	Date Due

IMPORTANT: All Fields are required.

Book Number: 1

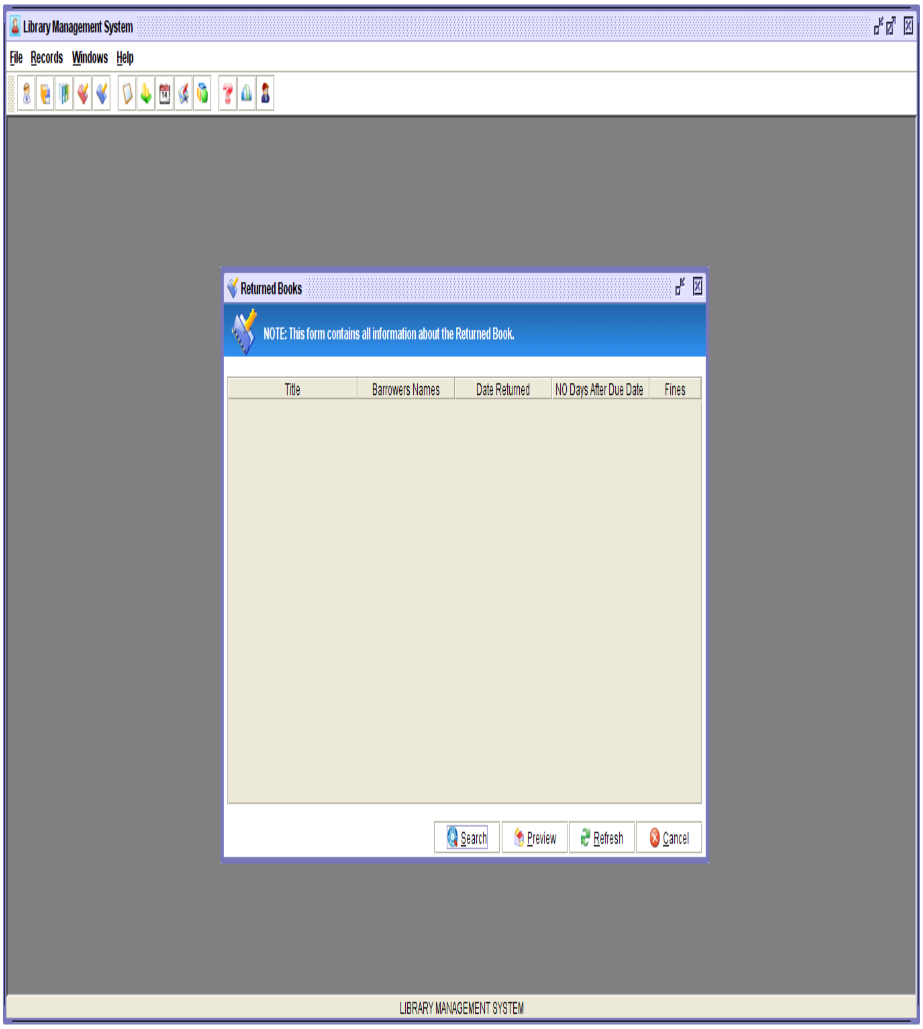
Barrowers ID: 1

Date Barrowerd: Jan 01

Date Due: Jan 01

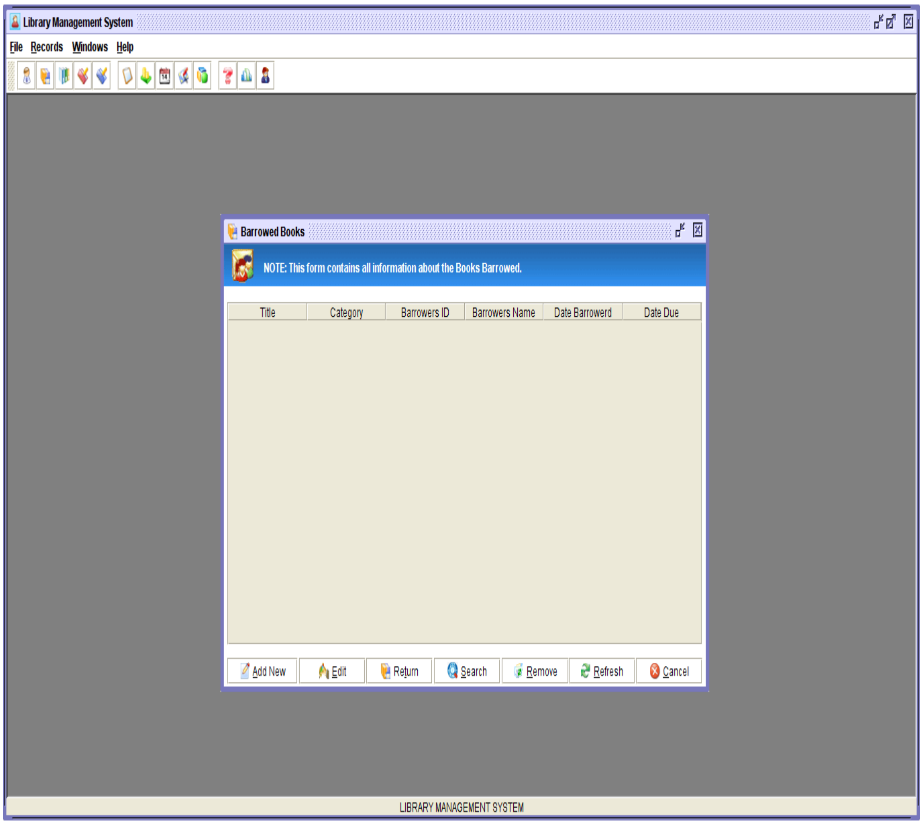
Add New
Edit
Return
Cancel
Add
Reset
Cancel
LIBRARY MANAGEMENT SYSTEM

3.10 Report Format:



NOTE: This form contains all information about the Returned Book.

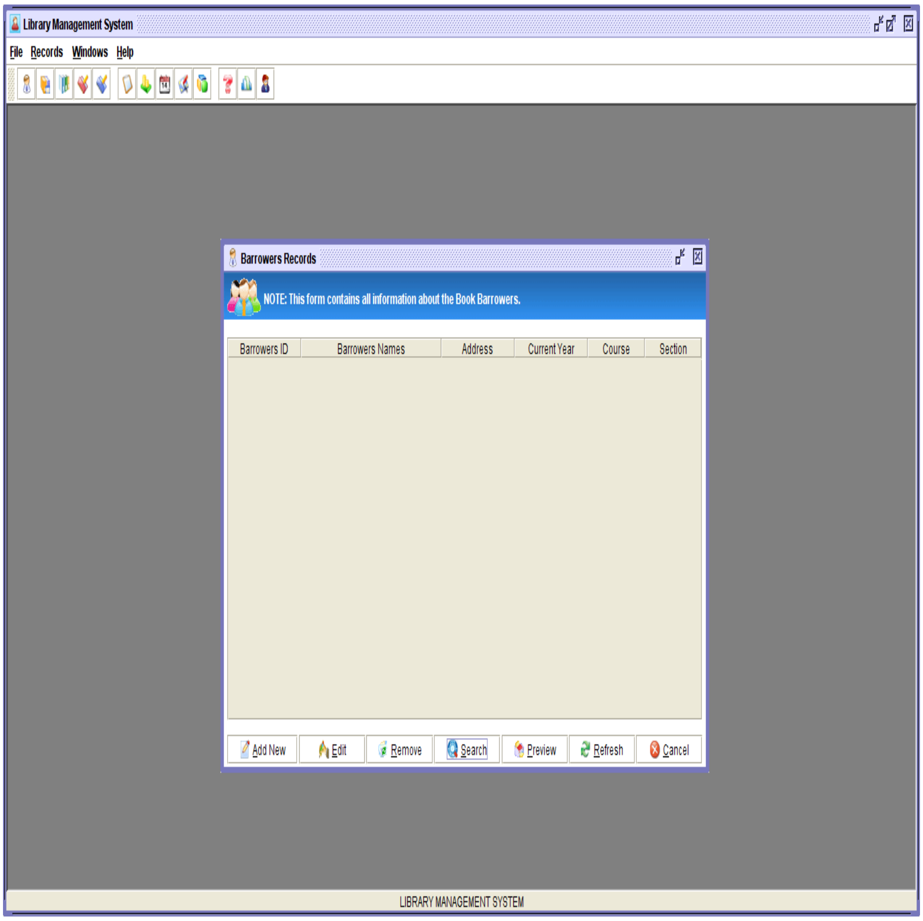
Title	Barrowers Names	Date Returned	NO Days After Due Date	Fines


Library Management System
File Records Windows Help
Barrowed Books

NOTE: This form contains all information about the Books Barrowed.

Title	Category	Barrowers ID	Barrowers Name	Date Barrowerd	Date Due

Add New
Edit
Return
Search
Remove
Refresh
Cancel
LIBRARY MANAGEMENT SYSTEM



NOTE: This form contains all information about the Book Barrowers.

Barrowers ID	Barrowers Names	Address	Current Year	Course	Section

3.11 Test Procedures and Implementation:

What is software testing?

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. It is a process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specifications and in the way the end user expects. This can be done by various software testing techniques which provide a systematic guidance for designing tests that exercise the input and output domains of the program to uncover errors in program function, behavior and performance. The following software testing techniques were used in order to uncover errors in the system:

- Unit testing
- Integration testing
- White box testing
- Black box testing

- Acceptance tests (Alpha and Beta testing)

1: Unit Testing

Unit testing is normally considered as an adjunct to the coding step. It is the test for the small units of code, e.g. programs, modules or procedures, in order to ensure that they perform their intended functions. All possible paths through the control structure are exercised to ensure that all statements in a program are executed at least once. Unit testing is also done to test the data flow across a module interface.

The following errors are uncovered during unit testing:

- Comparison of different data types.
- Incorrect logical operators or precedence.
- Incorrect comparison of variables.
- Improper or nonexistent loop termination.
- Improperly modified loop variable.

2: Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. During this activity, unit tested components are taken and a program structure is built as per the design. Then incremental integration is performed on the system. This means that programs are constructed and tested in small increments instead of testing the entire program as a whole. This is done because correction of errors becomes difficult in case of whole program testing as many errors were detected and it is not easy to correct them at one go. Thus, through incremental integration testing, any error uncovered could be easily noted and corrected and interfaces are tested completely.

3: White Box Testing

White box testing is also called as glass box testing. It is related with the structure (internal logic) of the program. It helps in uncovering many errors that black box testing cannot. During white box testing activity, every statement of programs is executed at least once. All independent paths are also executed. Every logical decision is executed to check both true and false conditions. All loops are executed at their boundaries and within their operational bounds. Validation checks are also done during this process.

4: Black Box Testing

Black box testing, also known as behavioral testing, focuses on the functional requirements of the software. It is related to input and output only and not related with the internal structure of the program. This testing is also done so as to find errors such as:

- Initialization and termination errors
- Behavior and performance errors
- Incorrect or missing functions
- Interface errors
- Errors in data structures and external database access
- Performance errors

5: Acceptance Testing (Alpha & Beta Testing)

An acceptance test is a test carried out by the customer or end user rather than the developer in order to enable the customer to validate all requirements. Alpha testing and beta testing are two types of acceptance tests that are conducted.

6: Alpha Testing

Alpha test is conducted in a controlled environment. As a matter of fact, the end user conducts alpha test at the developer’s site. During the course of the system development, the end user is operating the software in front of the developer and the errors and other problems are recorded. Rectification is made accordingly.

7: Beta Testing

Beta testing is also conducted by the end user, but in the absence of the developer. Here, the end user himself records all the problems that he encounters during testing the system and then reports them to the developer at regular intervals. As a result of problems reported during beta testing, modifications are made to overcome the problem

Test case

Test case for: Login Form

Description: This test case is used to check out the validity of

User Name and Password.

Id	Description	Test Values	Expected Result	Actual Result	Pass/Fail
L1	User id	User id = “”	Error message pop up	Error Message	Pass
L2	User id	User id = 001	Error message pop up	Error Message	Pass
L3	User id	User id = 1001	Main screen will appears	Main Page	Pass
L4	Password	Password = “”	Error message pop up	Error Message	Pass
L5	Password	Password = @#$	Error message pop up	Error Message	Pass
L6	Password	Password length less than 6 or greater than 10	Error message pop up	Error Message	Pass
L7	Password	Password = ctr2004	Main Screen will appears	Main Page	Pass

Test case for: Borrowers Maste

Description: This test case is used to test add or Edit of Borrowers Details.

Id	Description	Test Values	Expected Result	Actual Result	Pass/Fail
S1	Name	Enter Null Values	Error message pop up	Error Message	Pass
S2	Course having Selected Category	Enter Null values	Error message pop up	Error Message	Pass
S3	Category having selected Section	Enter particular section	Error message pop up	Error Message	Pass

Test case for: Borrow Details

Description: This test case is used to test add or Edit of Borrow details.

Id	Description	Test Values	Expected Result	Actual Result	Pass/Fail
R1	Borrow Book having Name	Enter Null Values	Error message pop up	Error Message	Pass
R2	Borrow book having available list	Enter alphabets	Error message pop up	Error Message	Pass
R3	Borrow book having success	Enter alphabets	Main form	Successfully	Pass

4.1User Manual:

User Manual acts as a guide to any user of the system. It can be describes everything about the system from user’s point of view. It also helps the user to operate system easily and efficiently.

1. **How to login?**

   The login screen is displayed by the system where the user has to login by entering the correct credentials.

   If the credentials are given by the user are correct then the respective screen with navigation menu is displayed by the system.

2. **How to use Menu?**

   The menu is provided to operate various screens & reports depending upon operations to be performed. The user can select particular menu by using mouse by navigating the items & clicking it. This will invoke the item & display the respective screens. User can navigate through the pages by clicking items of menu.

3. How to use Screens?

In the menu, various textboxes and buttons are provided to enter the data into the system. The screen contains various data grid views to display the data on the screen.

4. How to use Reports?

The user can see the different analytical information from report details and transaction data. The necessary data values must be provided for the reports.

4.2 Operations Manual

Login window is for the user authentication. This is for the company employee who will be entering the transaction of Add Books,AddBorrowers,Borrow, Book,Return Books, etc.

User:

1. User generated Account.
2. Add, Edit and search the Books details
3. Add, Edit and search the Borrow details
4. Add, Edit and search the Return details
5. Add, Edit and search the Borrowers details
6. Add, Edit and search the Category details
7. Add, Edit and search the Fine details
8. Add, Edit and search the Course details
9. Generate reports.

In the current system all of the operations are collected under standard menu format. Overall menu structure is arranged in such a way that ordinary user would easily and completely grasp the overall scope of the work in very small amount of time.

4.3 Forms and Report Specification

- **Form Name:** Books

  Table Names: Books.

  Form description:

  This form is used to store the add books data for the further use. In this form all the book information is stored to the system. The book details are stored in the book table.

- **Form Name:** Borrowers

  Table Names: Borrowers

  Form description:

  This form is used to store the add borrowers Details data for the further use. In this form all the borrowers information is stored to the system. The borrowers details are stored in the borrowers table.

- **Form Name:** Borrow

  Table Names: BorrowedBook

  Form description:

  This form is used to store the add issue Details data for the further use. In this form all the borrow information is stored to the system. The borrowedBook details are stored in the borrowedBook table.

- **Form Name:** Return Book

  Table Names: Borrowes

  Form description:

  This form is used to store the add return Details data for the further use. In this form all the Return Book information is stored to the system. The Return Book details are stored in the borrowedBook table.

Drawbacks and Limitations

This application avoids the manual work which is commonly done in the any College library. The manual work is much more tedious work. The manual work also requires much more man power. But using this system we can avoid this problem easily.

There are some drawbacks and limitations works.

- The current system is only desktop application in future I will try to implement it as a web application so from any corner of the world you can access it easily.
- If it is implemented as web application it provides the facility to users to check the availability of the books.
- In future I will go to add the billing module for the fines. Whatever fines of the students will printable form.

Proposed Enhancements

- Standalone project of the Library Management system.
- Fine will be calculate in day wise to due date
- Printing to report all details

Conclusion

The system is designed to manage any college library. It provides the facilities

To maintain all the records of books, barrowers, borrowed books etc. We can easily add, edit and update any records. This project provides simple GUI. There are some conclusions from whole projects are given below:

- Provides the friendly graphical user interface which is easily used by the naïve user.
- It also provides easy way to the maintain the data of borrower, books and borrowed books etc.
- We can easily update the fines by just change in the fine table.
- It provides the facility of shortcut keys with combinations of the ctrl key.

Bibliography

Books:

The Complete Reference J2EE.

JSP Complete Reference.

MS Access by Dusanpetkovic

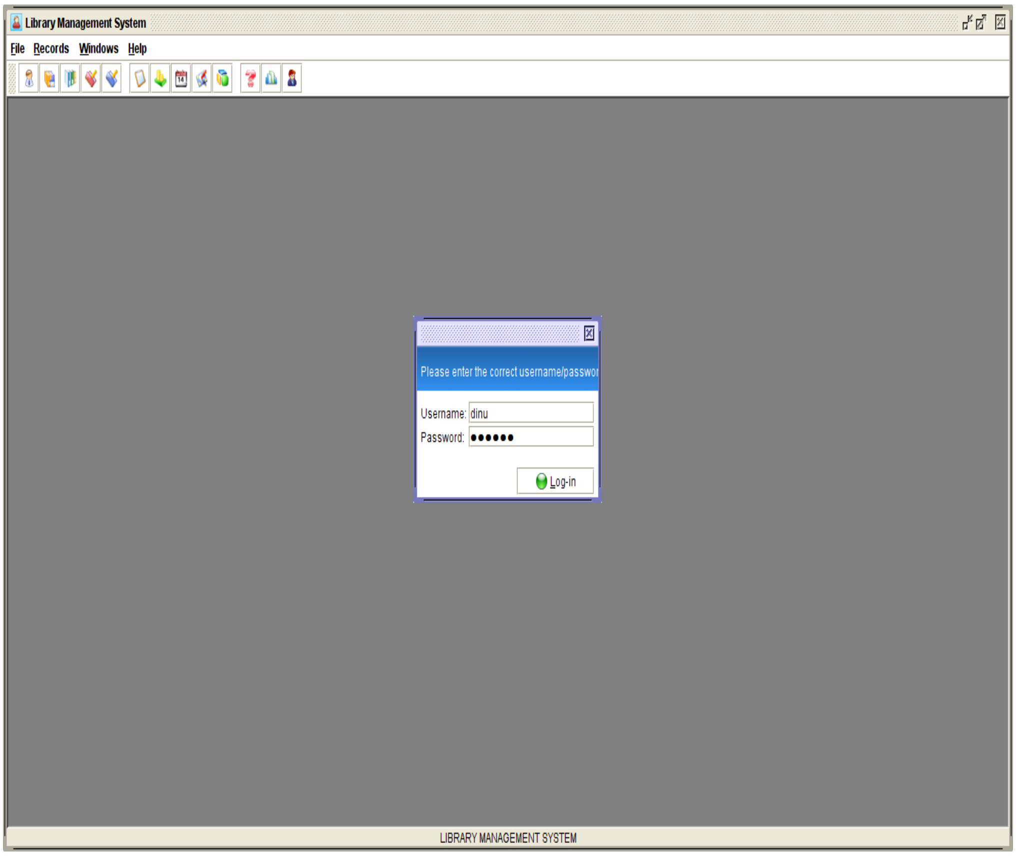
URLs:

www.sun.java.com

www.google.com

Annexure 1: User InterFace Screens:

Login Window:-



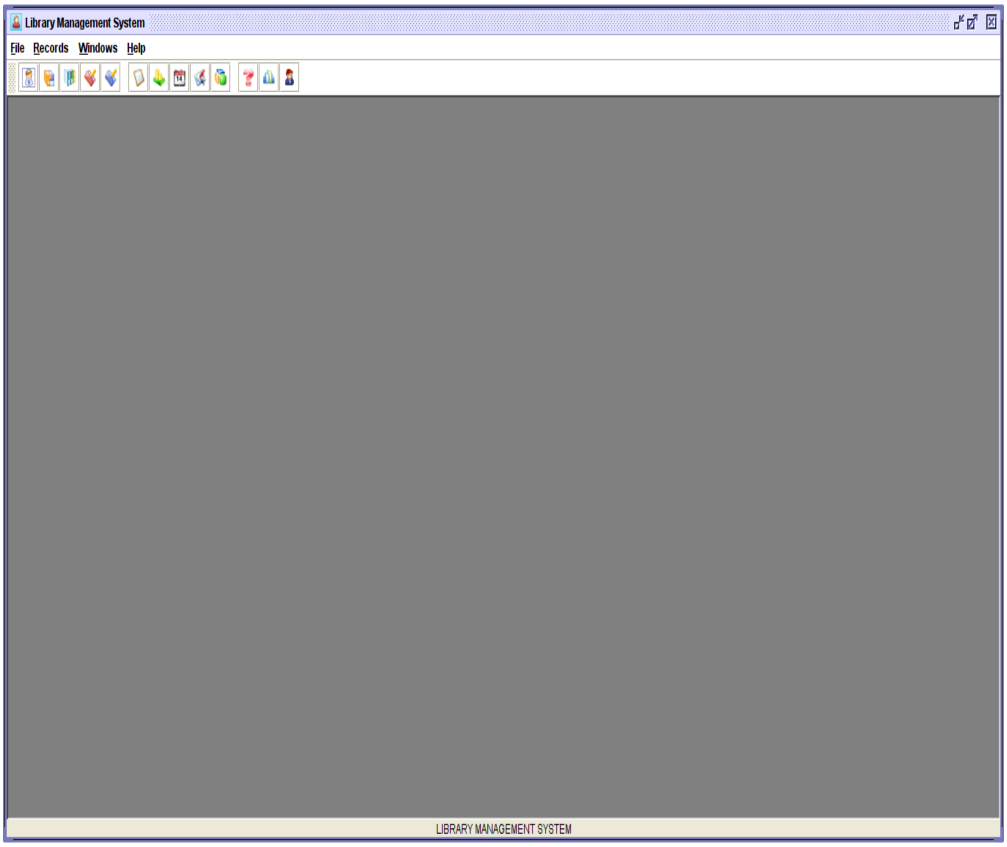
Library Management System
File Records Windows Help
Log-in
LIBRARY MANAGEMENT SYSTEM

Please enter the correct username/passwor

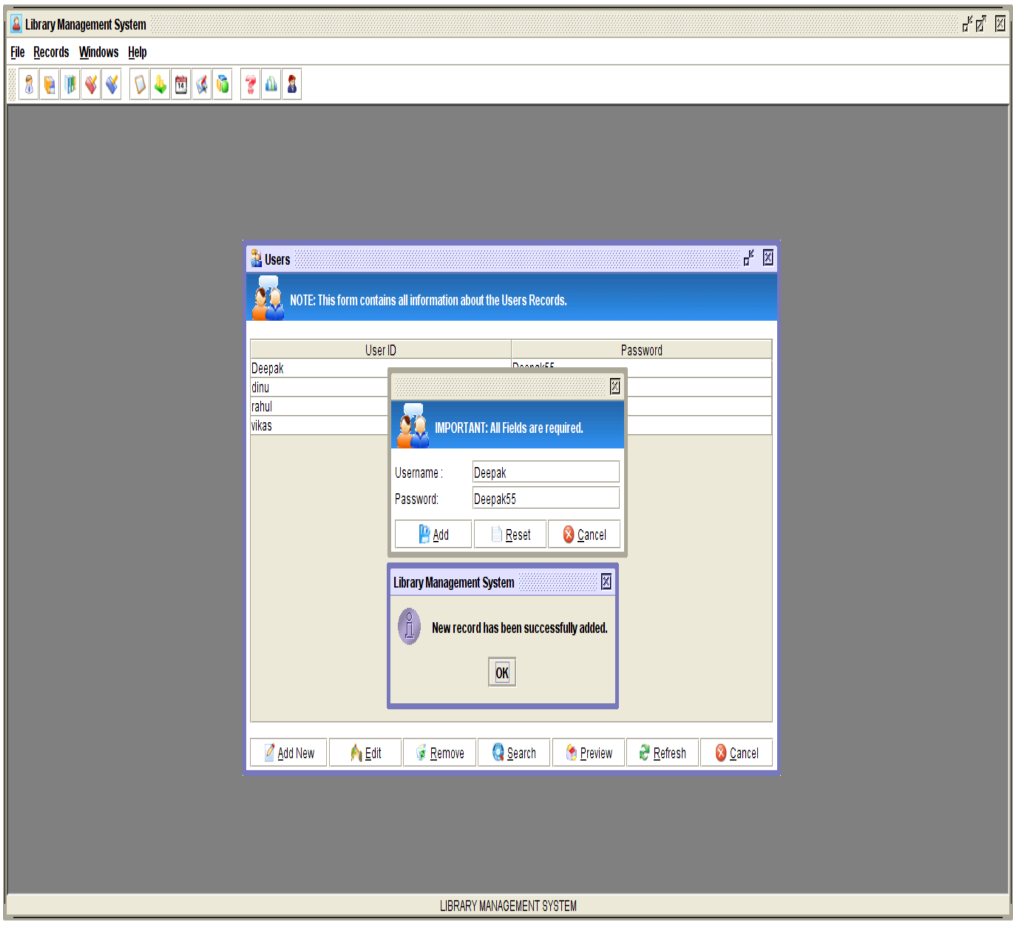
Username: dinu

Password: ●●●●●●

MDI From:-


Library Management System
File
Records
Windows
Help
LIBRARY MANAGEMENT SYSTEM

User Create:-



NOTE: This form contains all information about the Users Records.

User ID	Password
Deepak	Deepak55
dinu	
rahul	
vikas	

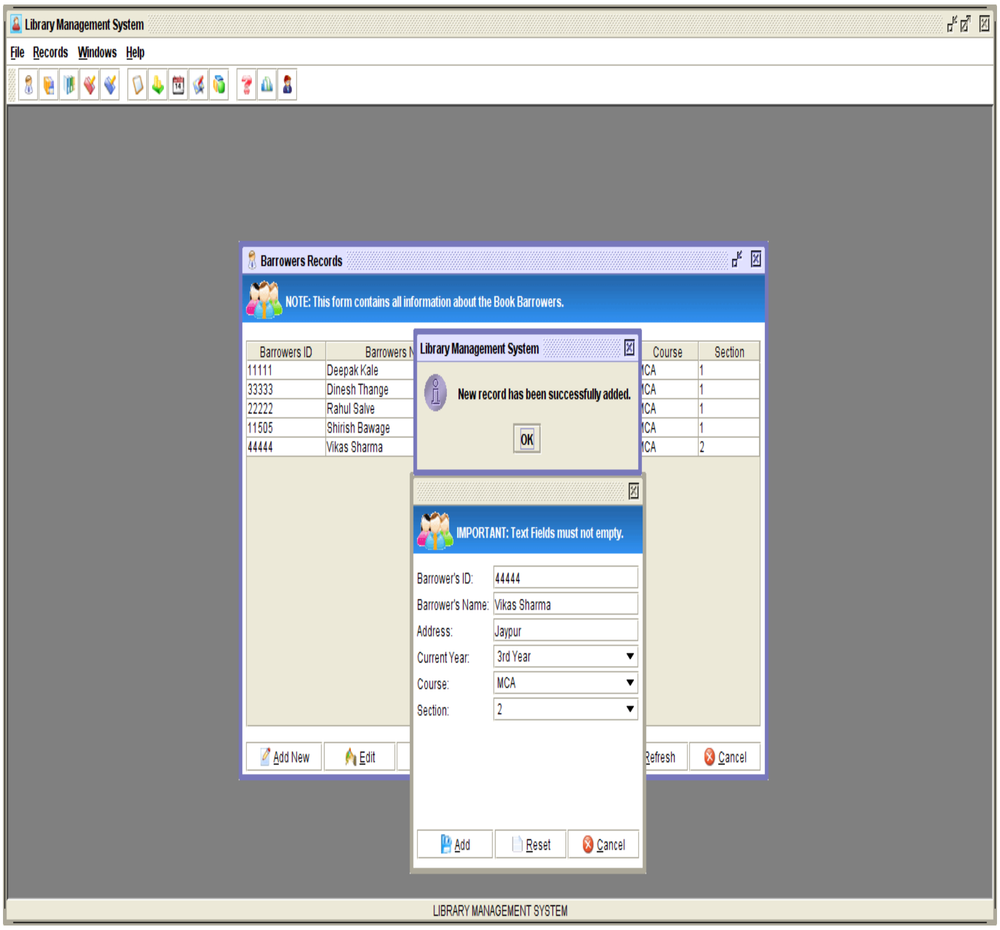
IMPORTANT: All Fields are required.

Username : Deepak

Password: Deepak55

New record has been successfully added.

Barrowers Add From:-


Library Management System
File Records Windows Help
Barrowers Records
Add New
Edit
Refresh
Cancel
Add
Reset
Cancel
LIBRARY MANAGEMENT SYSTEM

NOTE: This form contains all information about the Book Barrowers.

Barrowers ID	Barrowers N	Course	Section
11111	Deepak Kale	CA	1
33333	Dinesh Thange	CA	1
22222	Rahul Salve	CA	1
11505	Shirish Bawage	CA	1
44444	Vikas Sharma	CA	2

Library Management System

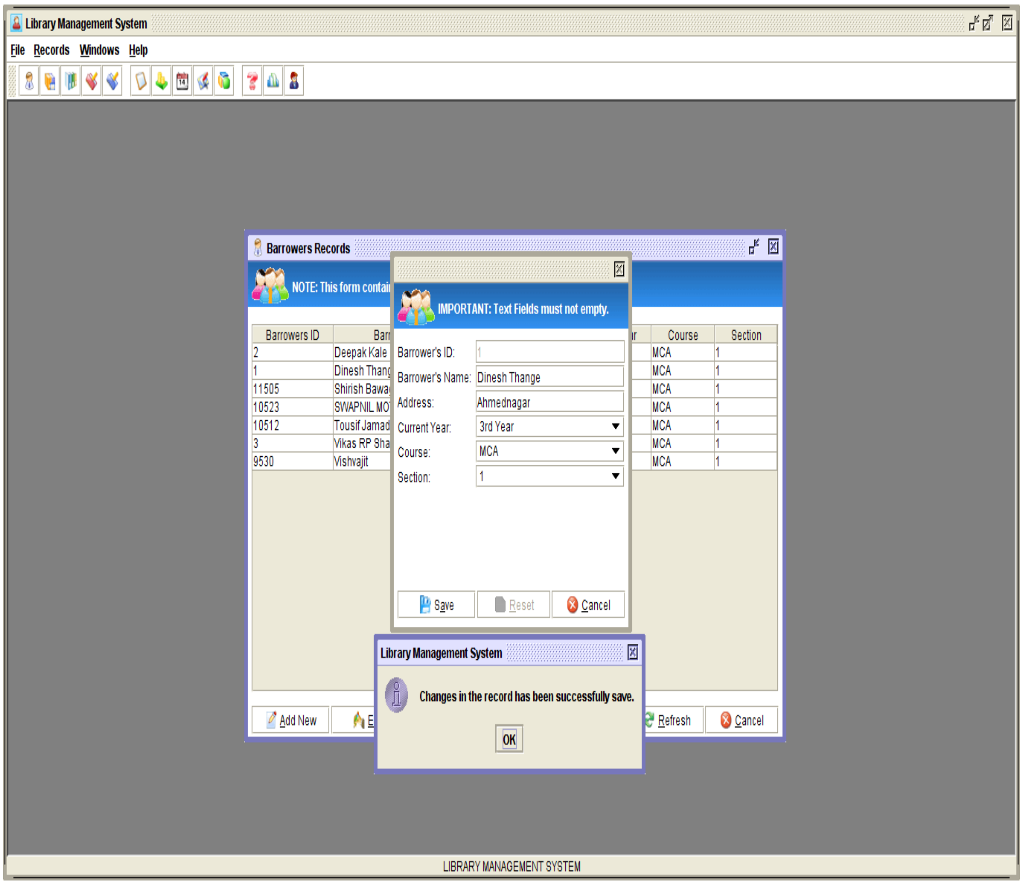
New record has been successfully added.

OK

IMPORTANT: Text Fields must not empty.

Barrower's ID:	44444
Barrower's Name:	Vikas Sharma
Address:	Jaypur
Current Year:	3rd Year
Course:	MCA
Section:	2

Barrowers Edit From:-



Barrowers ID	Barr	r	Course	Section
2	Deepak Kale		MCA	1
1	Dinesh Thang		MCA	1
11505	Shirish Bawa		MCA	1
10523	SWAPNIL MO		MCA	1
10512	Tousif Jamad		MCA	1
3	Vikas RP Sha		MCA	1
9530	Vishvajit		MCA	1

IMPORTANT: Text Fields must not empty.

Barrower's ID: 1

Barrower's Name: Dinesh Thange

Address: Ahmednagar

Current Year: 3rd Year

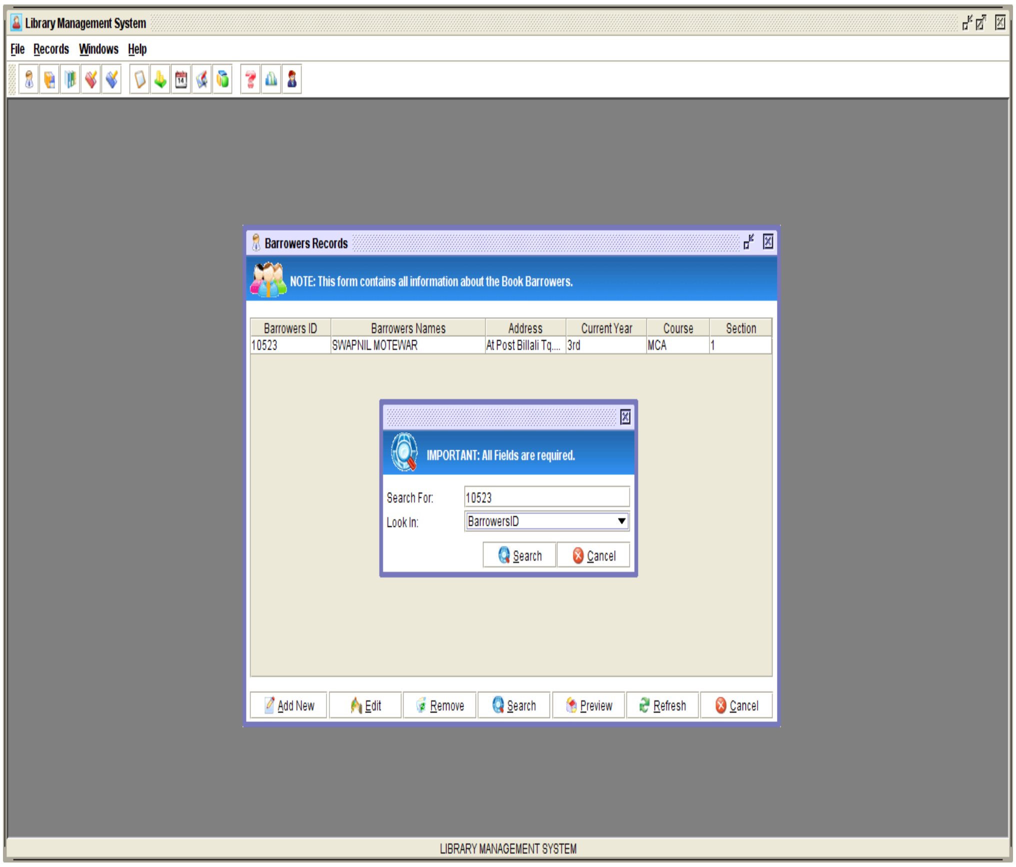
Course: MCA

Section: 1

Library Management System

Changes in the record has been successfully save.

Barrowers Search From:-



Library Management System
File Records Windows Help
Barrowers Records
NOTE: This form contains all information about the Book Barrowers.

Barrowers ID	Barrowers Names	Address	Current Year	Course	Section
10523	SWAPNIL MOTEWAR	At Post Billali Tq....	3rd	MCA	1

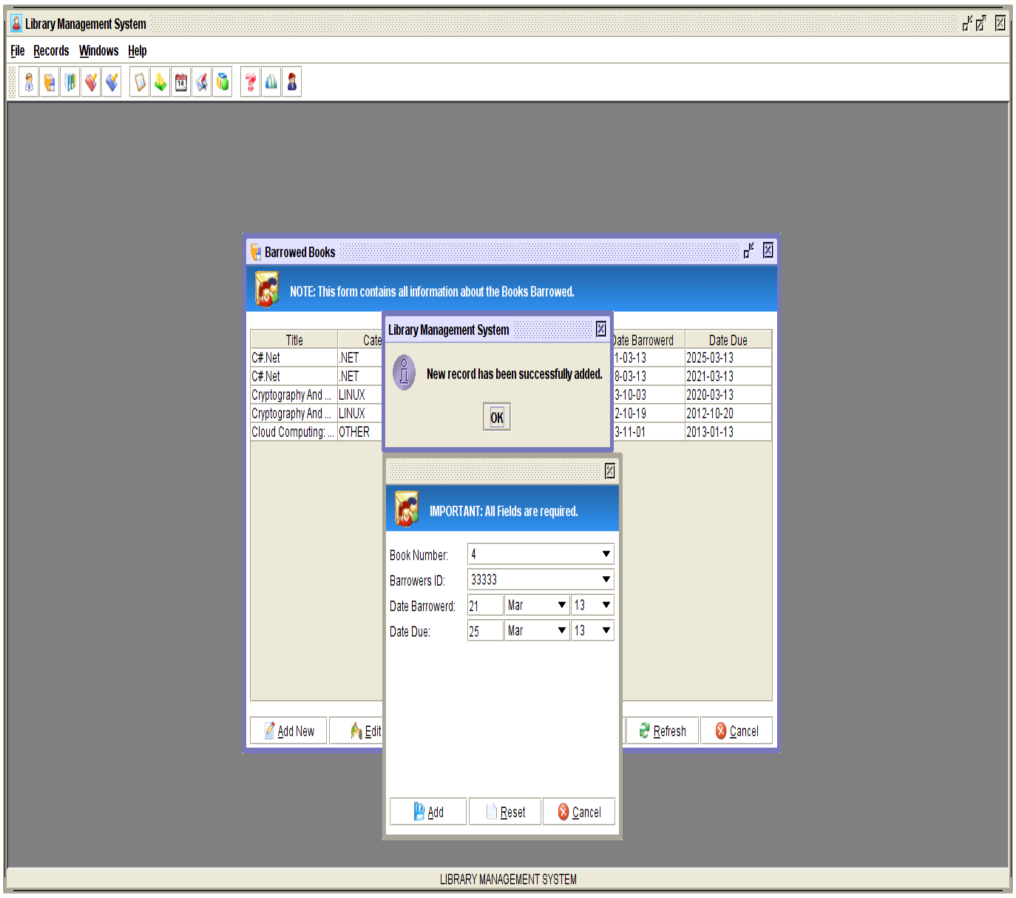
IMPORTANT: All Fields are required.

Search For: 10523

Look In: BarrowersID

Search
Cancel
Add New
Edit
Remove
Search
Preview
Refresh
Cancel
LIBRARY MANAGEMENT SYSTEM

Barrowed Add From:- (issue)



Barrowed Books

NOTE: This form contains all information about the Books Barrowed.

Title	Cate	Date Barrowerd	Date Due
C#.Net	.NET	1-03-13	2025-03-13
C#.Net	.NET	8-03-13	2021-03-13
Cryptography And ...	LINUX	3-10-03	2020-03-13
Cryptography And ...	LINUX	2-10-19	2012-10-20
Cloud Computing: ...	OTHER	3-11-01	2013-01-13

Library Management System

New record has been successfully added.

IMPORTANT: All Fields are required.

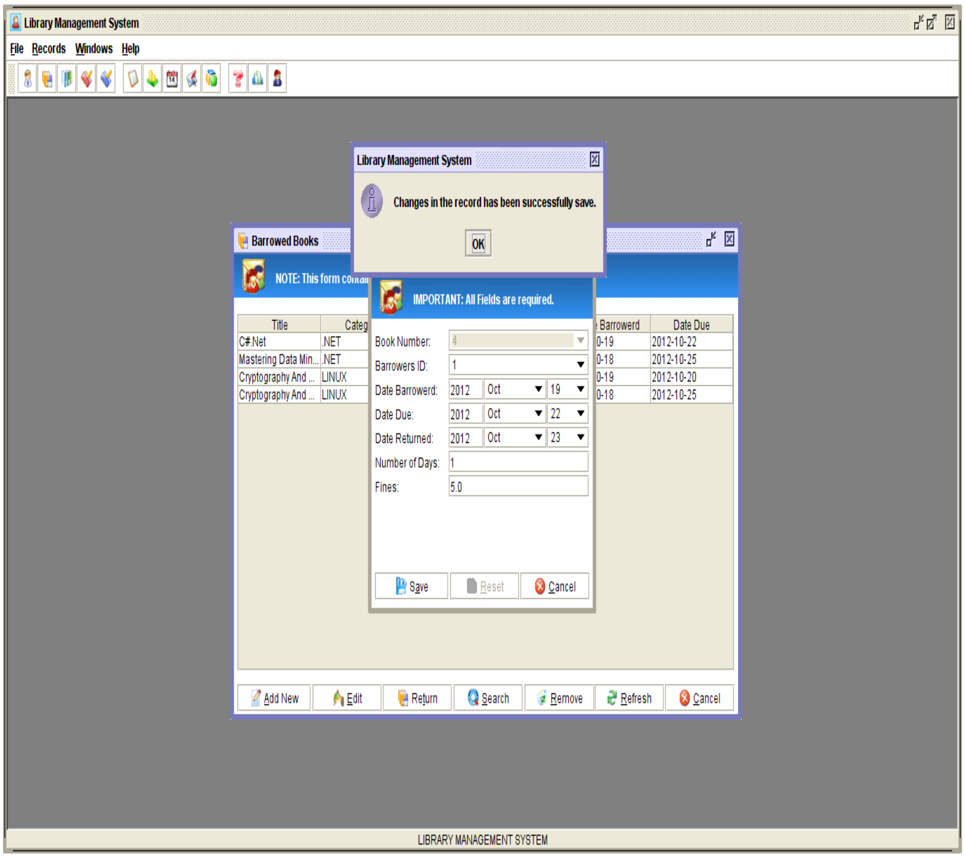
Book Number: 4

Barrowers ID: 33333

Date Barrowerd: 21 Mar 13

Date Due: 25 Mar 13

Barrowed Edit From:-


Library Management System
File Records Windows Help
Barrowed Books
Add New
Edit
Return
Search
Remove
Refresh
Cancel
Save
Reset
Cancel
OK
LIBRARY MANAGEMENT SYSTEM

Library Management System

Changes in the record has been successfully save.

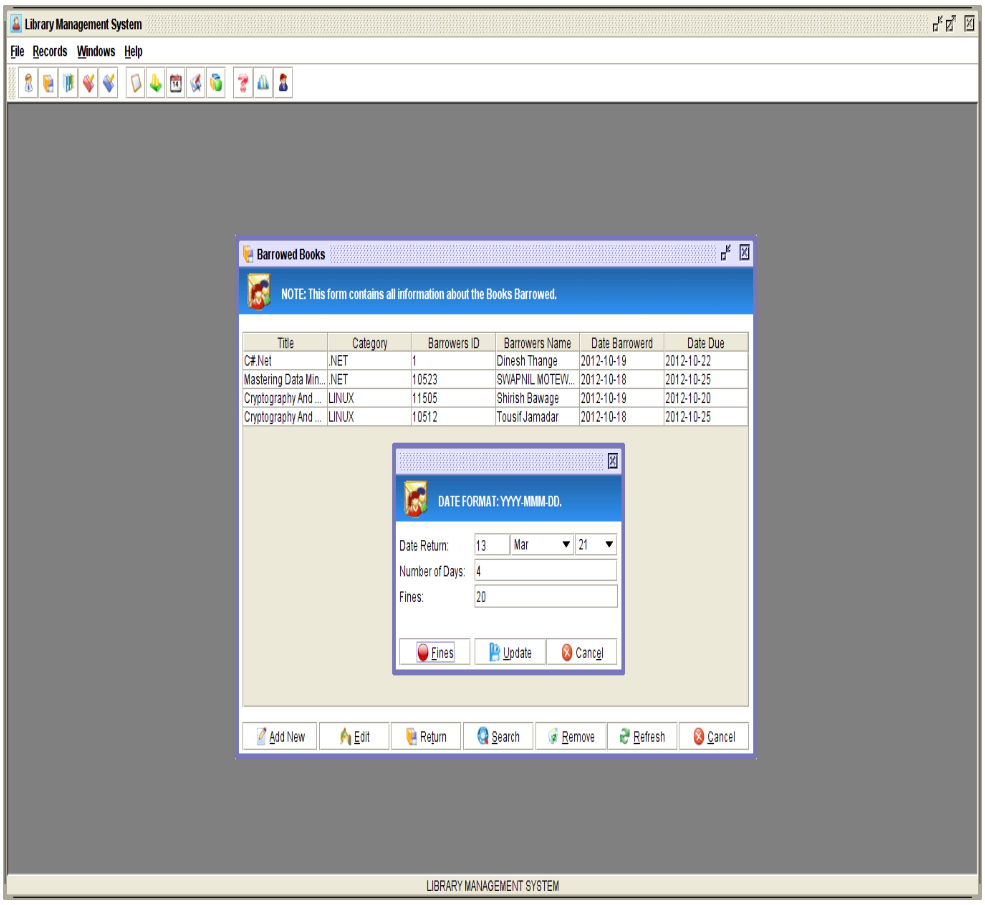
NOTE: This form contai

Title	Categ	e Barrowerd	Date Due
C#.Net	.NET	0-19	2012-10-22
Mastering Data Min...	.NET	0-18	2012-10-25
Cryptography And ...	LINUX	0-19	2012-10-20
Cryptography And ...	LINUX	0-18	2012-10-25

IMPORTANT: All Fields are required.

Book Number:	4
Barrowers ID:	1
Date Barrowerd:	2012 Oct 19
Date Due:	2012 Oct 22
Date Returned:	2012 Oct 23
Number of Days:	1
Fines:	5.0

Barrowed Fine From:-



NOTE: This form contains all information about the Books Barrowed.

Title	Category	Barrowers ID	Barrowers Name	Date Barrowerd	Date Due
C#.Net	.NET	1	Dinesh Thange	2012-10-19	2012-10-22
Mastering Data Min...	.NET	10523	SWAPNIL MOTEW...	2012-10-18	2012-10-25
Cryptography And ...	LINUX	11505	Shirish Bawage	2012-10-19	2012-10-20
Cryptography And ...	LINUX	10512	Tousif Jamadar	2012-10-18	2012-10-25

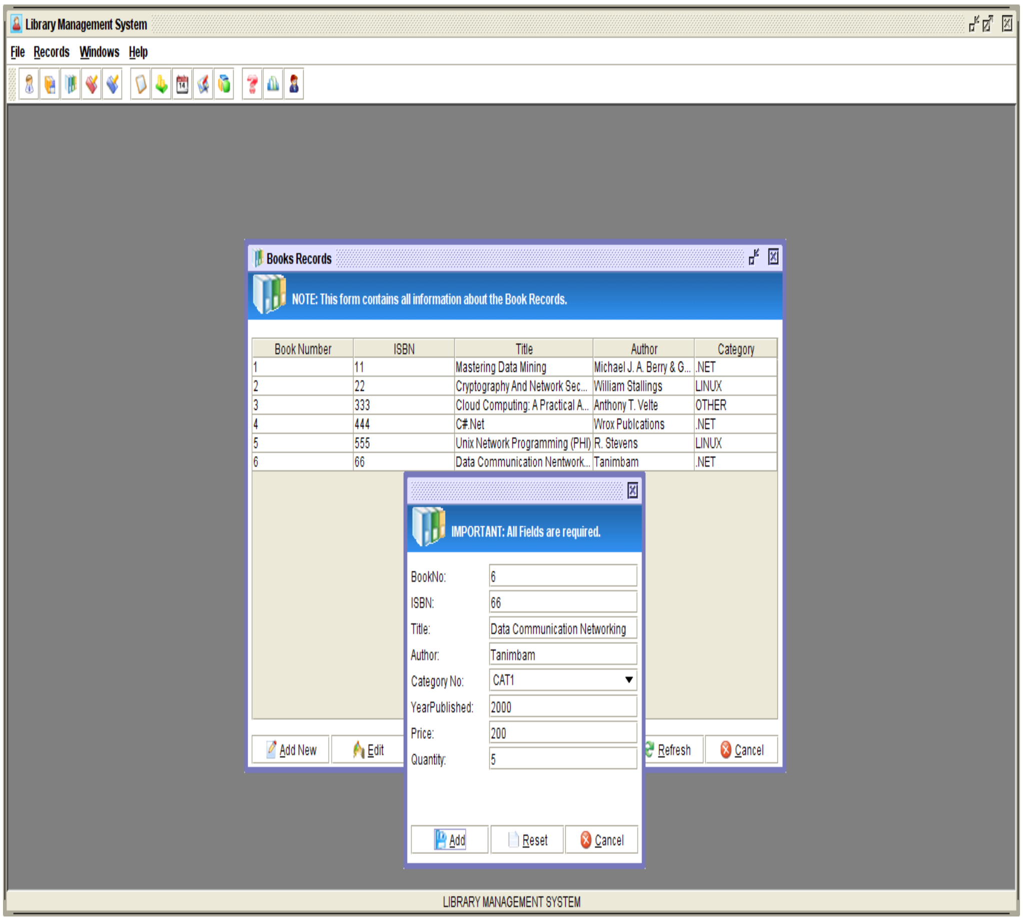
DATE FORMAT: YYYY-MMM-DD.

Date Return: 13 Mar 21

Number of Days: 4

Fines: 20

Books Add From:-



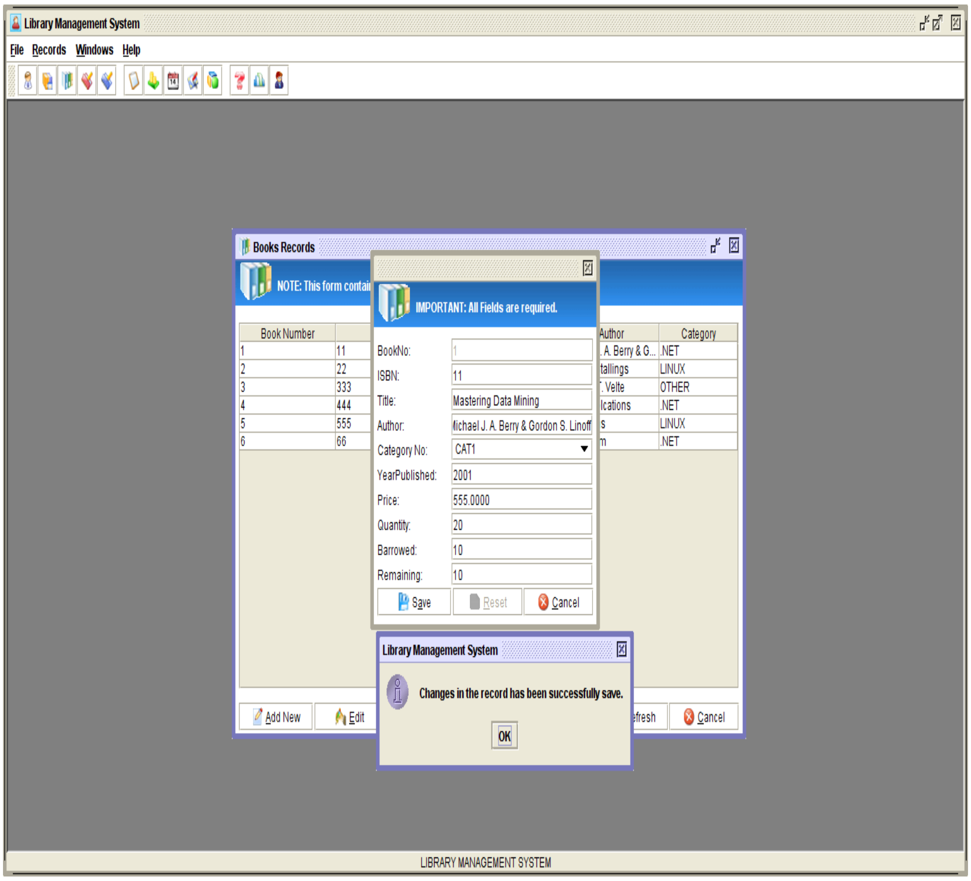
NOTE: This form contains all information about the Book Records.

Book Number	ISBN	Title	Author	Category
1	11	Mastering Data Mining	Michael J. A. Berry & G...	.NET
2	22	Cryptography And Network Sec...	William Stallings	LINUX
3	333	Cloud Computing: A Practical A...	Anthony T. Velte	OTHER
4	444	C#.Net	Wrox Publcations	.NET
5	555	Unix Network Programming (PHI)	R. Stevens	LINUX
6	66	Data Communication Nentwork...	Tanimbam	.NET

IMPORTANT: All Fields are required.

BookNo:	6
ISBN:	66
Title:	Data Communication Networking
Author:	Tanimbam
Category No:	CAT1
YearPublished:	2000
Price:	200
Quantity:	5

Book Edit From:-



Library Management System
File Records Windows Help
Books Records
NOTE: This form contai
Save
Reset
Cancel
Add New
Edit
efresh
Cancel
OK
LIBRARY MANAGEMENT SYSTEM

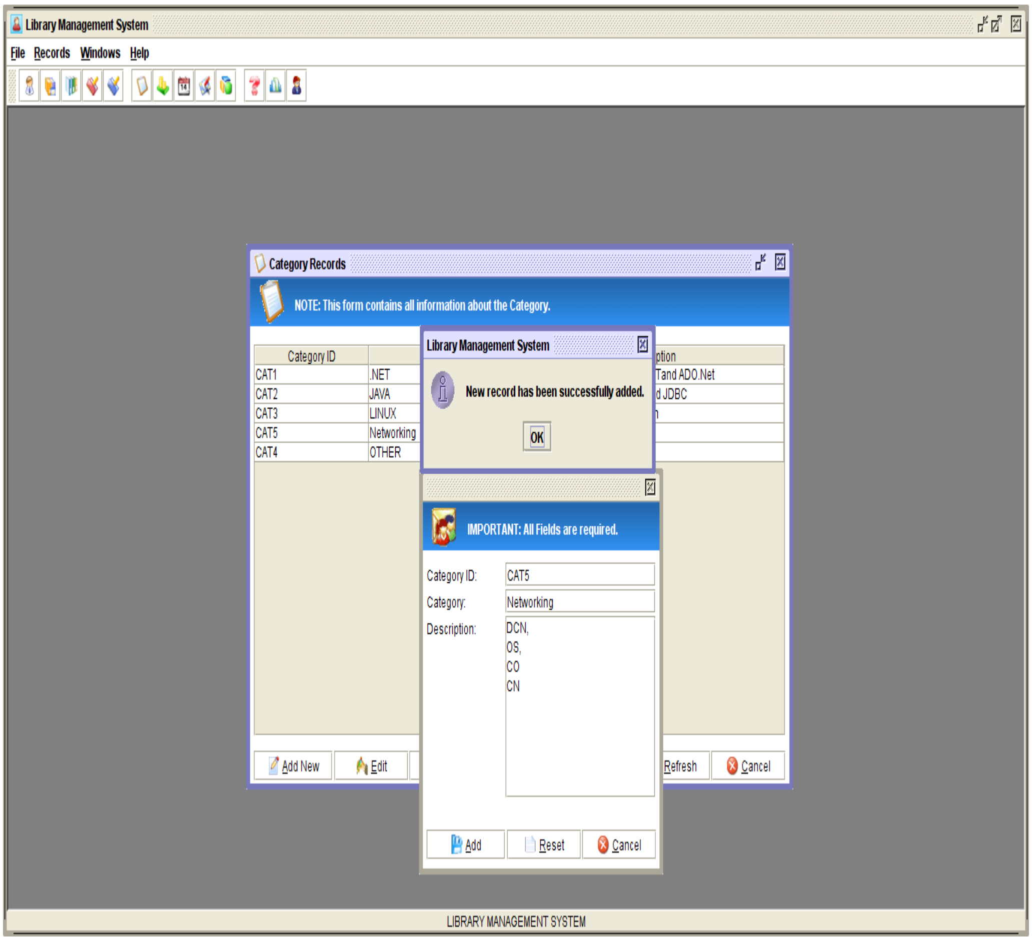
Book Number		Author	Category
1	11	. A. Berry & G...	.NET
2	22	tallings	LINUX
3	333	. Velte	OTHER
4	444	lcations	.NET
5	555	s	LINUX
6	66	m	.NET

IMPORTANT: All Fields are required.

BookNo:	1
ISBN:	11
Title:	Mastering Data Mining
Author:	lichael J. A. Berry & Gordon S. Linoff
Category No:	CAT1
YearPublished:	2001
Price:	555.0000
Quantity:	20
Barrowed:	10
Remaining:	10

Changes in the record has been successfully save.

Category Add From:-



Library Management System
File Records Windows Help
Category Records
Add New
Edit
Refresh
Cancel
Add
Reset
Cancel
OK
LIBRARY MANAGEMENT SYSTEM

NOTE: This form contains all information about the Category.

Category ID		ption
CAT1	.NET	T and ADO.Net
CAT2	JAVA	d JDBC
CAT3	LINUX	n
CAT5	Networking	
CAT4	OTHER	

Library Management System

New record has been successfully added.

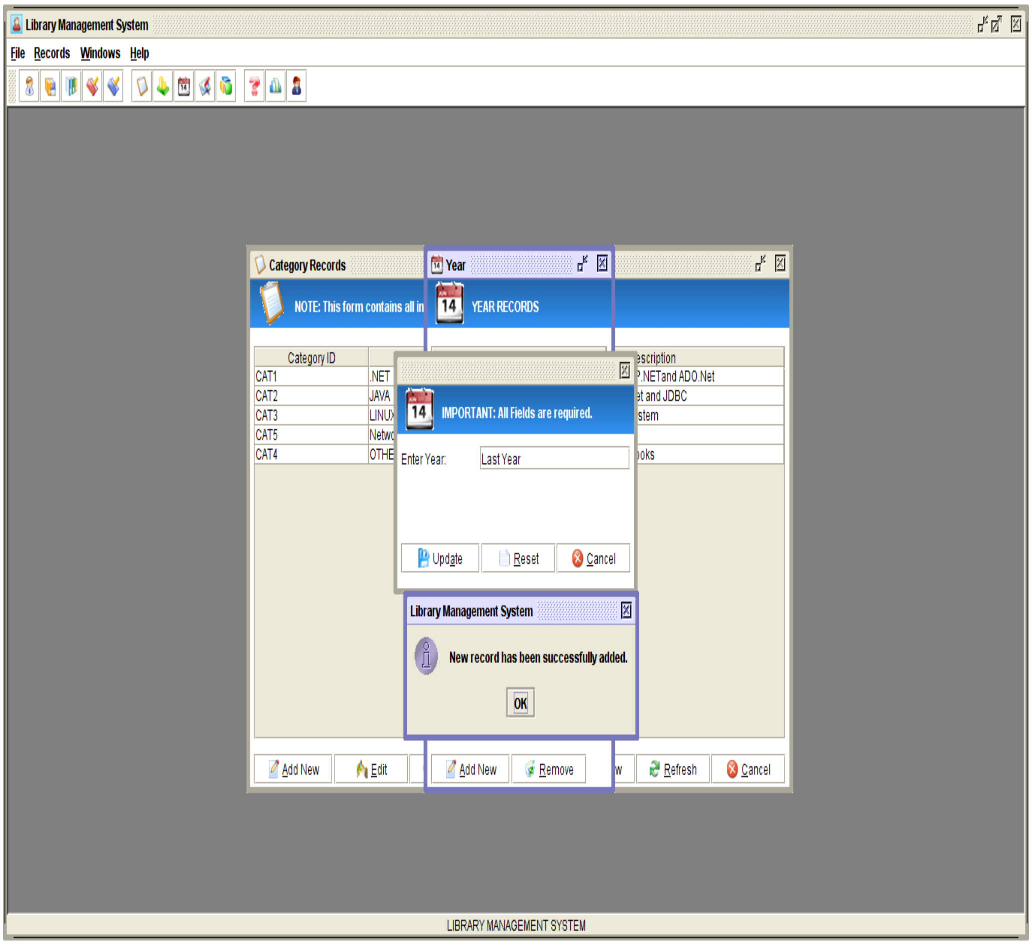
IMPORTANT: All Fields are required.

Category ID: CAT5

Category: Networking

Description: DCN,
OS,
CO
CN

Category Records:



NOTE: This form contains all in

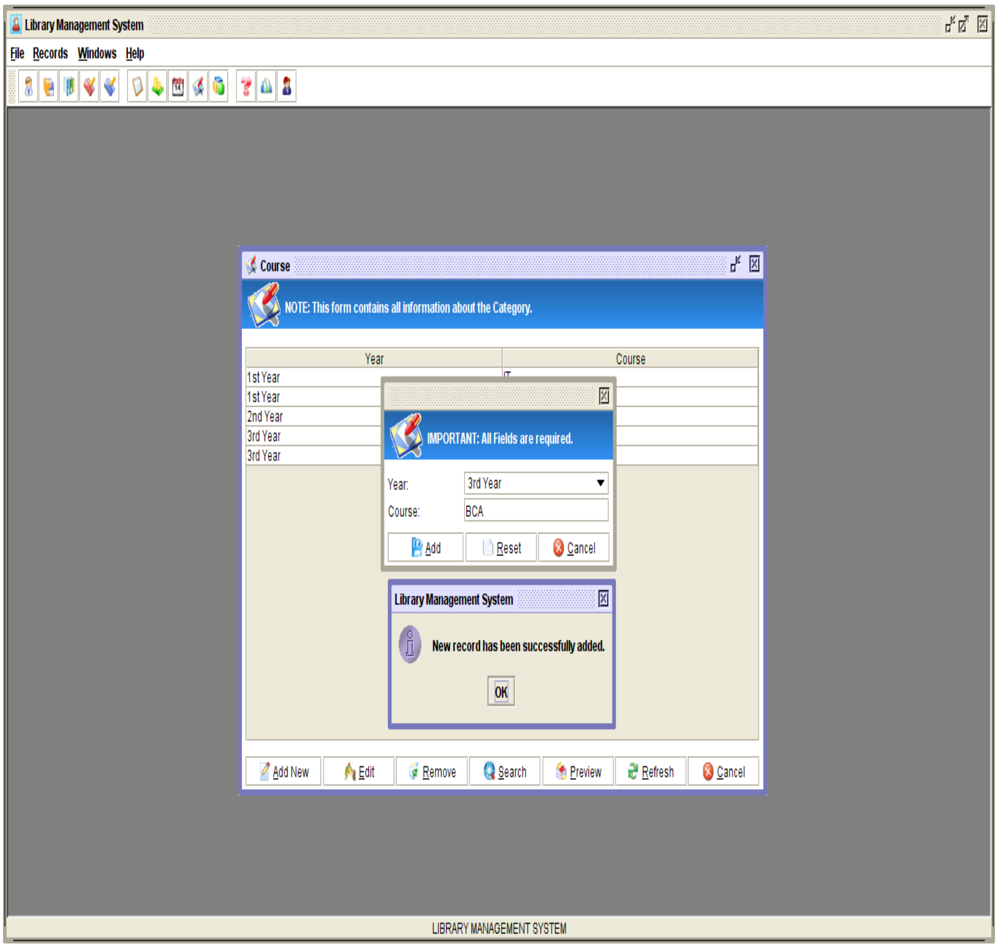
Category ID		escription
CAT1	.NET	P.NETand ADO.Net
CAT2	JAVA	et and JDBC
CAT3	LINUX	stem
CAT5	Netwo	
CAT4	OTHE	ooks

IMPORTANT: All Fields are required.

Enter Year: Last Year

New record has been successfully added.

Course Add From:-



NOTE: This form contains all information about the Category.

Year	Course
1st Year	IT
1st Year	
2nd Year	
3rd Year	
3rd Year	

IMPORTANT: All Fields are required.

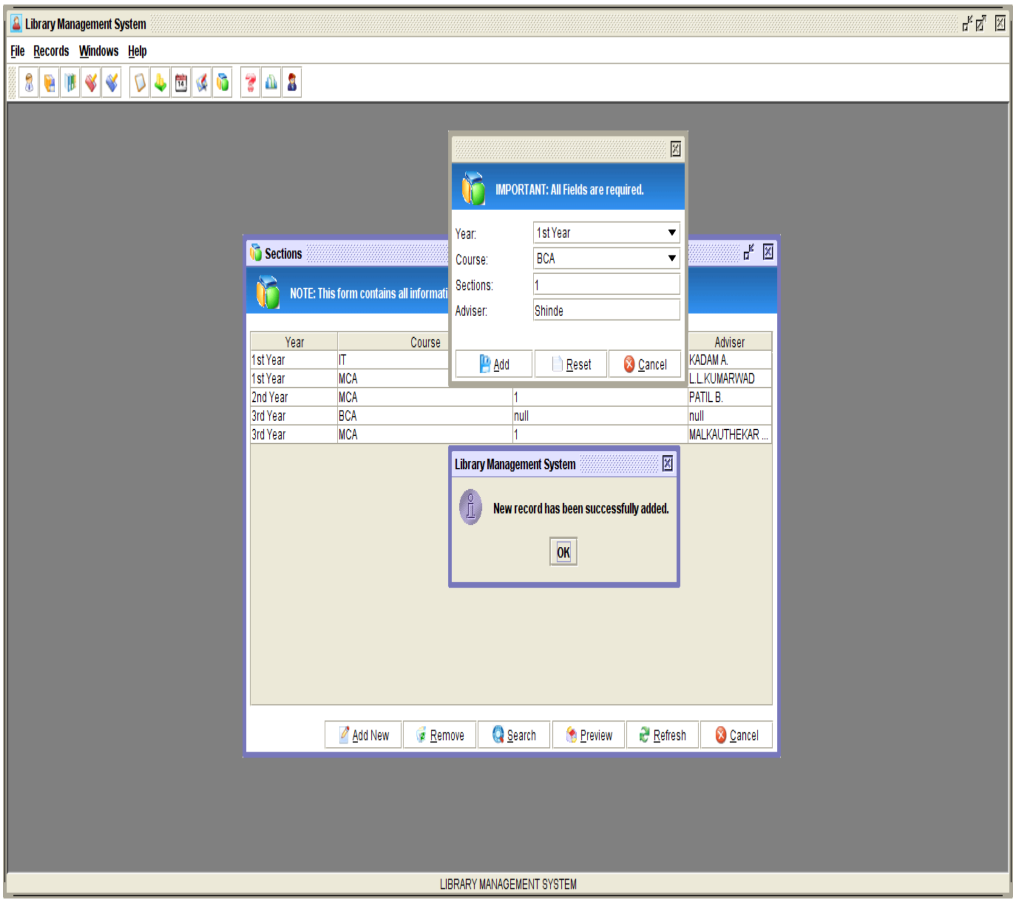
Year: 3rd Year

Course: BCA

Library Management System

New record has been successfully added.

Section Add From:-


Library Management System
File Records Windows Help
Sections
Add New
Remove
Search
Preview
Refresh
Cancel
LIBRARY MANAGEMENT SYSTEM

IMPORTANT: All Fields are required.

Year: 1st Year

Course: BCA

Sections: 1

Adviser: Shinde

NOTE: This form contains all informati

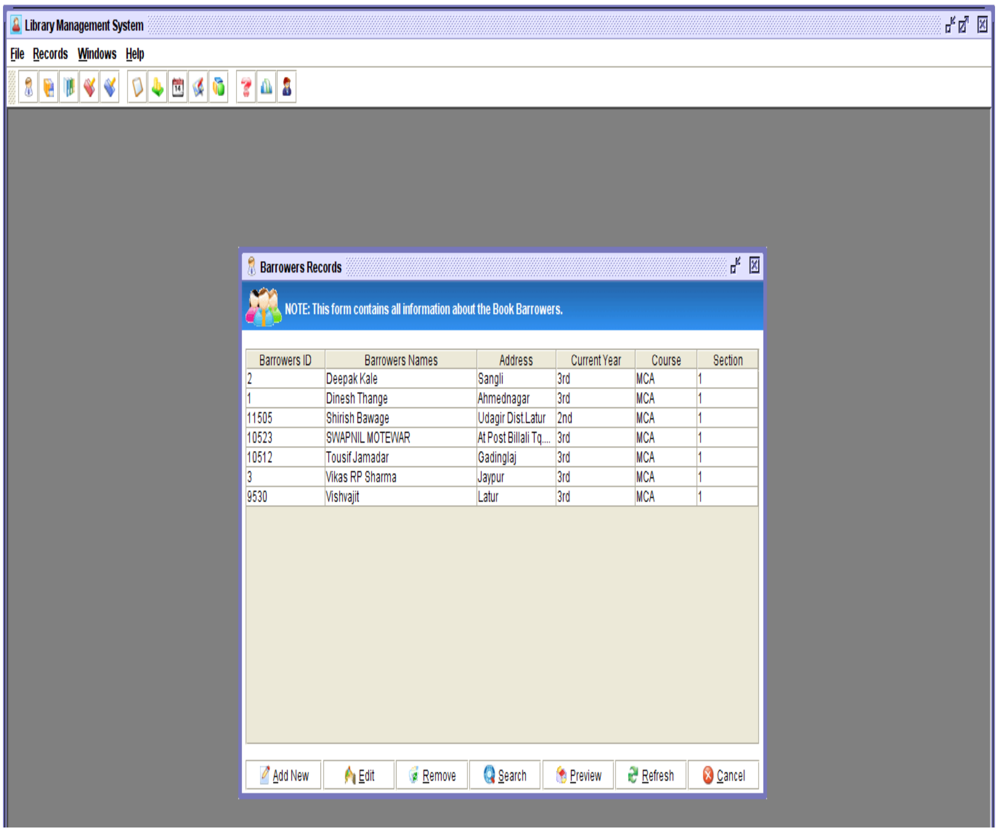
Year	Course		Adviser
1st Year	IT		KADAM A.
1st Year	MCA		L.L.KUMARWAD
2nd Year	MCA	1	PATIL B.
3rd Year	BCA	null	null
3rd Year	MCA	1	MALKAUTHEKAR ...

Library Management System

New record has been successfully added.

Annexure 2: Output Reports With Data:

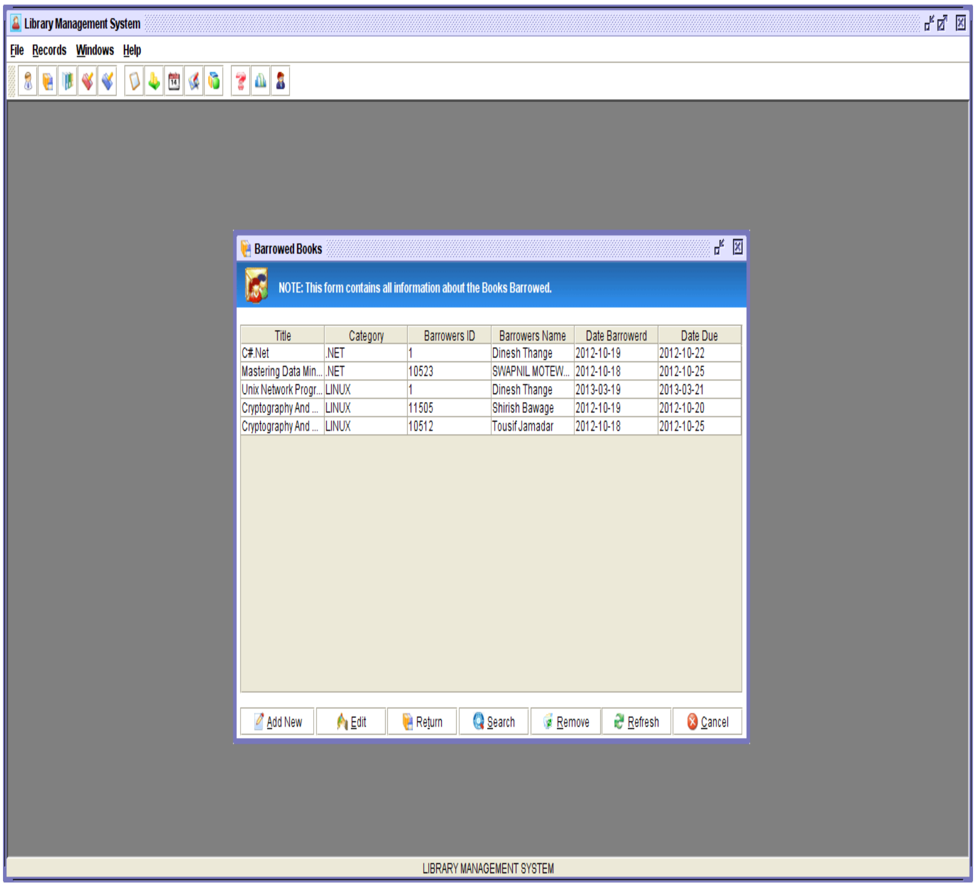
Barrowers Records:-



NOTE: This form contains all information about the Book Barrowers.

Barrowers ID	Barrowers Names	Address	Current Year	Course	Section
2	Deepak Kale	Sangli	3rd	MCA	1
1	Dinesh Thange	Ahmednagar	3rd	MCA	1
11505	Shirish Bawage	Udagir Dist.Latur	2nd	MCA	1
10523	SWAPNIL MOTEWAR	At Post Billali Tq....	3rd	MCA	1
10512	Tousif Jamadar	Gadinglaj	3rd	MCA	1
3	Vikas RP Sharma	Jaypur	3rd	MCA	1
9530	Vishvajit	Latur	3rd	MCA	1

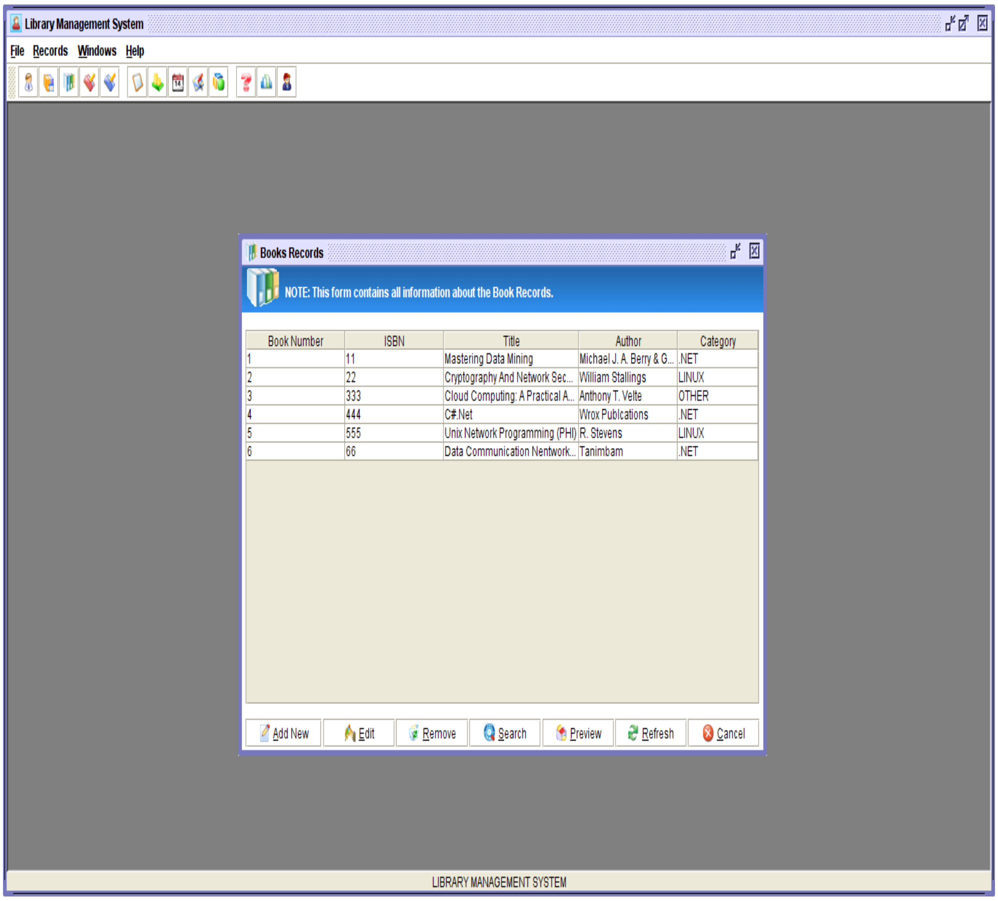
Barrowers Books Records :-



NOTE: This form contains all information about the Books Barrowed.

Title	Category	Barrowers ID	Barrowers Name	Date Barrowerd	Date Due
C#.Net	.NET	1	Dinesh Thange	2012-10-19	2012-10-22
Mastering Data Min...	.NET	10523	SWAPNIL MOTEW...	2012-10-18	2012-10-25
Unix Network Progr...	LINUX	1	Dinesh Thange	2013-03-19	2013-03-21
Cryptography And ...	LINUX	11505	Shirish Bawage	2012-10-19	2012-10-20
Cryptography And ...	LINUX	10512	Tousif Jamadar	2012-10-18	2012-10-25

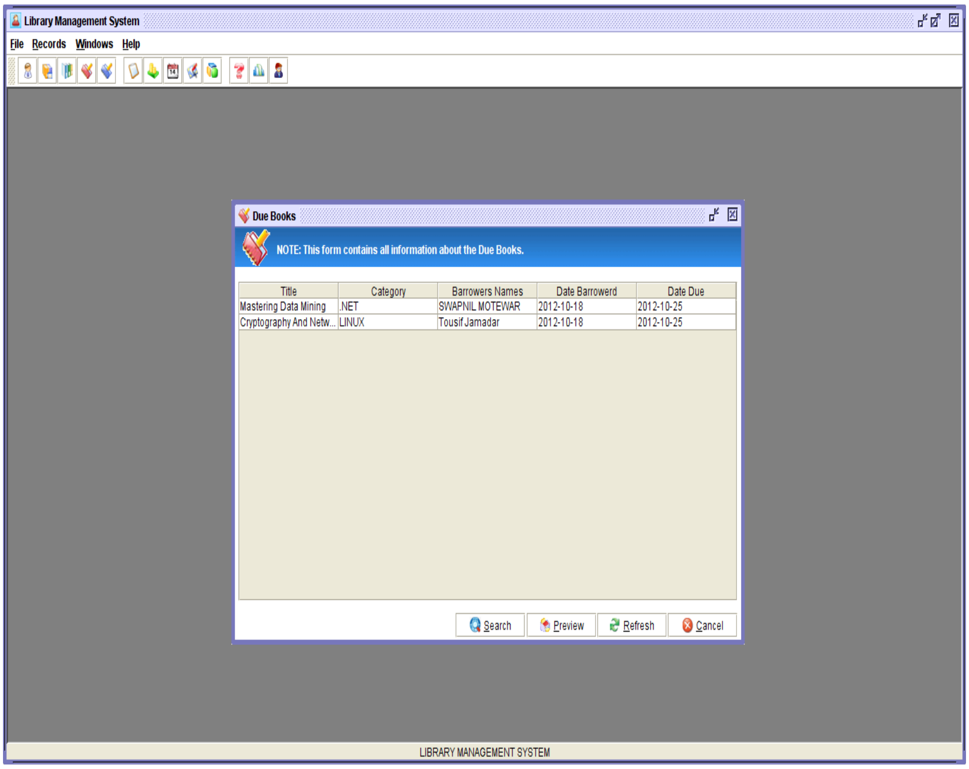
Books Record:-



NOTE: This form contains all information about the Book Records.

Book Number	ISBN	Title	Author	Category
1	11	Mastering Data Mining	Michael J. A. Berry & G...	.NET
2	22	Cryptography And Network Sec...	William Stallings	LINUX
3	333	Cloud Computing: A Practical A...	Anthony T. Velte	OTHER
4	444	C#.Net	Wrox Publcations	.NET
5	555	Unix Network Programming (PHI)	R. Stevens	LINUX
6	66	Data Communication Nentwork...	Tanimbam	.NET

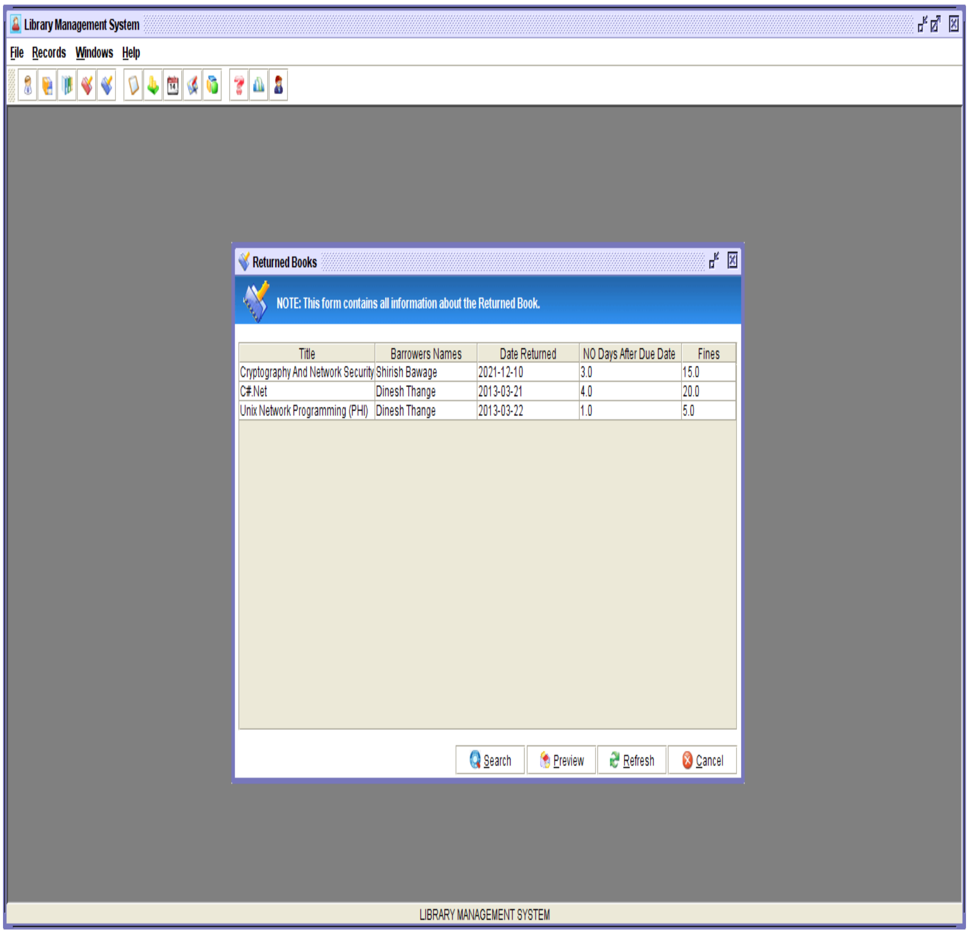
Due Book Record:-



NOTE: This form contains all information about the Due Books.

Title	Category	Barrowers Names	Date Barrowerd	Date Due
Mastering Data Mining	.NET	SWAPNIL MOTEWAR	2012-10-18	2012-10-25
Cryptography And Netw...	LINUX	Tousif Jamadar	2012-10-18	2012-10-25

Return Books Record:-



NOTE: This form contains all information about the Returned Book.

Title	Barrowers Names	Date Returned	NO Days After Due Date	Fines
Cryptography And Network Security	Shirish Bawage	2021-12-10	3.0	15.0
C#.Net	Dinesh Thange	2013-03-21	4.0	20.0
Unix Network Programming (PHI)	Dinesh Thange	2013-03-22	1.0	5.0

Annexure 3:Simple Programing code

Program Codeing

Sample Code for frmFines.java

```
import java.util.*;

import java.text.*;

import java.sql.*;

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.event.*;

import javax.swing.border.*;

import javax.swing.table.*;

import javax.swing.plaf.metal.*;

public class frmFines extends JInternalFrame implements ActionListener

{

        public static JPanel            jpnlMain       = new JPanel();

        Connection cnFines;

        public static Statement stmtFines;

        public static ResultSet rsFines;     //Recordset
```

```
    public static String sSQL;

    Dimension screen    =
    Toolkit.getDefaultToolkit().getScreenSize();

    //JButton Variables

    JButton    bttnUpdate       =    new    JButton("Update",new
ImageIcon("@imgs/add new.gif"));

    //JLabel Variables

    JLabel  lblHeader     = new JLabel();

    JLabel  lblIcon       = new JLabel();

    JLabel  lblCaption    = new JLabel("FINES");

    JLabel  lblFines      = new JLabel("Fines:");

    JTextField txtFines = new JTextField();

    JFrame JFParentFrame;

    mdlFunctions module_func = new mdlFunctions();

    mdlSQLStatements module_sql    =new
mdlSQLStatements();

    public        frmFines(Connection        conn,        JFrame
getParentFrame) throws SQLException

    {

        super("Fines",false,true,false,true);

        cnFines = conn;

        sSQL = "SELECT * FROM tblFines";

        try
        {
        stmtFines =
cnFines.createStatement(ResultSet.TYPE_SCROLL_INSENSITI
VE,ResultSet.CONCUR_UPDATABLE);
        }
        catch(SQLException sqlEx){}
        try
        {
            rsFines = stmtFines.executeQuery(sSQL);
            rsFines.next();
            txtFines.setText("" +
rsFines.getInt("Fines"));
        }
        catch(SQLException
sqlEx){System.out.println(sqlEx.getMessage());}
    module_func.setJButton(bttnUpdate,125,60,100,25,"updat
e","Update");
        bttnUpdate.setMnemonic(KeyEvent.VK_A);

        bttnUpdate.addActionListener(JBActionListener);
        lblHeader.setIcon(new
ImageIcon("@imgs/Barrowers Records.gif"));
        lblIcon.setIcon(new ImageIcon("@imgs/fines
list.gif"));
```

```
		module_func.setJTextField(txtFines,60,60,60,25);
		module_func.setJLabel(lblHeader,0,0,750,40);
		module_func.setJLabel(lblIcon,5,2,50,40);
		module_func.setJLabel(lblCaption,60,2,500,40);
		module_func.setJLabel(lblFines,5,60,100,20);
		lblCaption.setFont(new Font("Dialog", Font.BOLD, 12));
		lblCaption.setForeground(new Color(255,255,255));
		jpnlMain.setBackground(Color.WHITE);
		jpnlMain.setLayout(null);
		//Add Labels
		jpnlMain.add(lblCaption);
		jpnlMain.add(lblIcon);
		jpnlMain.add(lblHeader);
		jpnlMain.add(lblFines);
		//Add TextField
		jpnlMain.add(txtFines);
		//Add Buttons
		jpnlMain.add(bttnUpdate);
		getContentPane().setLayout(new BorderLayout(0,0));
```

```
        getContentPane().add(BorderLayout.CENTER, jpnlMain);
        setFrameIcon(new ImageIcon("@imgs/fines.gif"));
        setSize(240,140);

    setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        setLocation((screen.width  -  240)/2,((screen.height-140)/2)-14);
    }
    ActionListener JBActionListener = new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            String srcObj = e.getActionCommand();
            if(srcObj=="update")
            {
                try
                {

    stmtFines.executeUpdate("UPDATE tblFines SET Fines = '" + txtFines.getText() + " '");

    JOptionPane.showMessageDialog(null,"Changes in the
record has been successfully saved","Library Management System",JOptionPane.INFORMATION_MESSAGE);
                                        dispose();
                                }
                                catch(SQLException sqlEx){System.out.println(sqlEx.getMessage());}
                        }
                }
        };
        public void actionPerformed(ActionEvent event)
        {
                setVisible(false);
                dispose();
        }

}
```