**LICT\_JAVA\_TUP\_BSMRSTU\_01\_CSE\_JAN’17**

**Java Project**

**Library Management System**

|  |  |
| --- | --- |
| **Submitted By:** | **Submitted To:** |
| **Group:1**  **Members: Avijit Bairagi (G018080),**  **Sanjit Majumdar(G018016),**  **Mahdia Tazeem (G018019),**  **Gondhoraj Chandro Sarker(G018092)** | **Mohamed Raffic Rasul.**  **Java Trainer, LICT.** |

**Date of Submission : 31.03.17**

**ACKNOWLEDGEMENT**

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, **Mohamed Raffic Rasul**, for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We also take this opportunity to express a deep sense of gratitude to our class coordinators. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank our friends and family for the support and encouragement they have given us during the course of our work.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Content** | Page No. |
| ABSTRACT | 5 |
| **1. INTRODUCTION** | 6 |
| 1.1 PROJECT AIMS AND OBJECTIVES | 6 |
| 1.2 BACKGROUND OF PROJECT | 7 |
| 1.3 OPERATION ENVIRONMENT | 8 |
| **2. SYSTEM ANALYSIS** | 9 |
| 2.1 SOFTWARE REQUIREMENT SPECIFICATION | 9 |
| 2.2 EXISTING VS PROPOSED | 20 |
| 2.3 SOFTWARE TOOL USED | 21 |
| **3. SYSTEM DESIGN** | 27 |
| 3.1 Table Design | 27 |
| **4. SYSTEM IMPLEMENTATION** | 31 |
| 4.1 MODULE DESCRIPTION | 31 |
| 4.2 User Manual | 32 |
| **5.LIMITATIONS & FUTURE SCOPES** | 39 |
| 5.1 Limitation | 39 |
| 5.2 Future Scopes | 39 |
| **6.CONCLUSION** | 40 |

**ABSTRACT**

This project of “**LIBRARY MANAGEMENT**” of gives us the complete information about the library. We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records and can also check how many books are issued and stock available in the library. In this project, we can maintain the late fine of students who returns the issued books after the due date. Library management system is a project which aims in developing a computerized system to maintain all the daily work of library. This project has many features which are generally not available in normal library management systems like facility of user login and a facility of admin login. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date.

Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

**CHAPTER 1**

**INTRODUCTION**

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

**1.1 PROJECT AIMS AND OBJECTIVES**

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows: 

* Online book issue.
* Request column for librarian for providing new books.
* A separate column for digital library.
* Student login page where student can find books issued by him/her and date of return.
* A search column to search availability of books.
* A teacher login page where teacher can add any events being organized in the college and important suggestions regarding books.
* Online notice board about the workshop.

**1.2 BACKGROUND OF PROJECT**

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system, there will be no loss of book record or member record which generally happens when a non-computerized system is used.

In addition, report module is also included in Library Management System. If user’s position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

**1.3 OPERATION ENVIRONMENT**

|  |  |
| --- | --- |
| PROCESSOR | INTEL CORE PROCESSOR OR BETTER PERFORMANCE |
| OPERATING SYSTEM | ALL TYPES |
| MEMORY | 1GB RAM OR MORE |
| HARD DISK SPACE | 2 GB |
| DATABASE | ORACLE |

**CHAPTER 2**

**SYSTEM ANALYSIS**

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

**2.1 SOFTWARE REQUIREMENT SPECIFICATION**

**2.1.1 GENERAL DESCRIPTION**

**PRODUCT DESCRIPTION:**

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

**PROBLEM STATEMENT:**

The problem occurred before having computerized system includes:

* **File lost:**

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

* **File damaged:**

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

* **Difficult to search:**

record When there is no computerized system there is always a difficulty in searching of records if the records are large in number.

* **Space consuming:**

After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

* **Cost consuming:**

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

**2.1.2 SYSTEM OBJECTIVES**

* **Improvement in control and performance:**

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

* **Save cost:**

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

* **Save time:**

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

**2.1.3 SYSTEM REQUIREMENTS**

**2.1.3.1 NON-FUNCTIONAL REQUIREMENTS**

* Product Requirements

EFFICIENCY REQUIREMENT

When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very fast.

RELIABILITY REQUIREMENT

The system should accurately perform member registration, member validation, book transaction and search.

USABILITY REQUIREMENT

The system is designed for a user-friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

ORGANIZATIONAL REQUIREMENT IMPLEMENTATION

REQUIREMNTS

In implementing whole system, it uses java language & the database part is developed using Oracle.

* + 1. **FUNCTIONAL REQUIREMENTS**

**2.1.4.1 NORMAL USER**

**2.1.4.1.1 STUDENT LOGIN**

Description of feature

This feature used by the student to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

-user id is provided when they register

-The system must only allow user with valid id and password enter the system.

-The system performs authorization process which decides what user level can access to.

-The user must be able to logout after they finished using system.

**2.1.4.1.2 REGISTER NEW USER**

Description of feature

This feature can be performed by all users to register new user to create account.

Functional requirements

-System must be able to verify information.

-System must be able to delete information if information is wrong.

**2.1.4.1.3 SEARCH BOOK**

Description of features

This feature is found in book maintenance part. We can search book based on book name.

Functional requirements

- System must be able to search the database based on select search type.

**2.1.4.1.4 ISSUE BOOKS AND RETURN BOOKS**

Description of features

This feature allows to issue and return books and also view reports of book issued.

Functional requirements

-System must be able to enter issue information in database.

- System must be able to search if book is available or not before issuing books.

-System should be able to enter issue and return date information.

-System can automatically count the fine .

**2.1.4.2 ADMIN USER**

**2.1.4.2.1 ADMIN LOGIN**

Description of feature

This feature used by the admin/librarian to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

-user id is provided when they register

-The system must only allow user with valid id and password enter the system.

-The system performs authorization process which decides what user level can access to.

-The user must be able to logout after they finished using system.

**2.1.4.2.2 SEARCH BOOK**

Description of features

This feature is found in book maintenance part . We can search book based on book name.

Functional requirements

- System must be able to search the database based on select search type.

**2.1.4.2.3 ISSUE BOOKS AND RETURN BOOKS**

Description of features

This feature allows to issue and return books and also view reports of book issued.

Functional requirements

-System must be able to enter issue information in database.

-System must be able to update number of books.

- System must be able to search if book is available or not before issuing books.

-System should be able to enter issue and return date information.

-System can automatically count the fine & admin can see when student return the book.

**2.1.5 SOFTWARE AND HARDWARE REQUIREMENTS**

This section describes the software and hardware requirements of the system.

**2.1.5.1 SOFTWARE REQUIREMENTS**

* Operating system- Any Windows can be used as the operating system as it is stable and supports more features and is more user friendly.
* Database - Oracle 10g is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.
* Development tools and Programming language

- For the ADMIN user page java, swing & JDBC is used.

- For the STUDENT user page servlets & JDBC.

**2.1.5.2 HARDWARE REQUIREMENTS**

* Intel core i5 2nd generation is used as a processor because it is fast than other processors a provide reliable and stable and we can run our pc for longtime. By using this processor, we can keep on developing our project without any worries.
* Ram 2 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing.

**2.2 EXISTING VS PROPOSED SYSTEM**

i. Existing system does not have any facility of teacher’s login or student login whereas proposed system will have a facility of student login as well as Admin login.

ii. Existing system does not have a facility of adding new books so easily whereas proposed system has a facility of it.

iii. Existing system does not have any facility for book request where as in proposed system after logging in to their accounts student can check books as well as check the return date & also the fine.

iv. Existing system is slower where proposed system is very faster & user can easily use it.

**2.3 SOFTWARE TOOLS USED**

The whole Project is divided in two parts the student login part and admin part.

**2.3.1** **Student Login part**

The Student Login part is designed using of Java servlet, jsp, HTML, CSS & JDBC.

Java servlet :

A **Java servlet** is a [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) program that is used to create web application. This program extends the capabilities of a [server](https://en.wikipedia.org/wiki/Server_(computing)). Although servlets can respond to any types of requests, they most commonly implement applications hosted on servers. Such Web servlets are the [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) counterpart to other dynamic Web content technologies such as [PHP](https://en.wikipedia.org/wiki/PHP) and [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET).

Servlets are most often used to process or store a [Java class](https://en.wikipedia.org/wiki/Java_class) in [Java EE](https://en.wikipedia.org/wiki/Java_EE) that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) protocol, but they are most often used with the [HTTP protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol). Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a [software developer](https://en.wikipedia.org/wiki/Software_developer) may use a servlet to add [dynamic content](https://en.wikipedia.org/wiki/Dynamic_web_page) to a [web server](https://en.wikipedia.org/wiki/Web_server) using the [Java platform](https://en.wikipedia.org/wiki/Java_platform). The generated content is commonly [HTML](https://en.wikipedia.org/wiki/HTML), but may be other data such as [XML](https://en.wikipedia.org/wiki/XML). Servlets can maintain [state](https://en.wikipedia.org/wiki/State_(computer_science)) in [session](https://en.wikipedia.org/wiki/Session_(computer_science)) variables across many server transactions by using [HTTP cookies](https://en.wikipedia.org/wiki/HTTP_cookie), or [URL rewriting](https://en.wikipedia.org/wiki/URL_rewriting).

To deploy and run a servlet, a [web container](https://en.wikipedia.org/wiki/Web_container) must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access right.

JSP:

**Java Server Pages** (**JSP**) is a technology that helps [software developers](https://en.wikipedia.org/wiki/Software_developer) create [dynamically generated web pages](https://en.wikipedia.org/wiki/Dynamic_web_page) based on [HTML](https://en.wikipedia.org/wiki/HTML), [XML](https://en.wikipedia.org/wiki/XML), or other document types. Released in 1999 by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems), JSP is similar to [PHP](https://en.wikipedia.org/wiki/PHP), [ASP](https://en.wikipedia.org/wiki/Active_Server_Pages) and [Reacts JSX](https://en.wikipedia.org/wiki/React_(JavaScript_library)#JSX) but it uses the [Java programming language](https://en.wikipedia.org/wiki/Java_(programming_language)).

Architecturally, JSP may be viewed as a high-level [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) of [Java servlets](https://en.wikipedia.org/wiki/Java_servlet). JSPs are translated into [servlets](https://en.wikipedia.org/wiki/Java_Servlet) at runtime, therefore JSP is a Servlet; each JSP servlet is cached and re-used until the original JSP is modified.

JSP can be used independently or as the view component of a server-side [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) design, normally with [JavaBeans](https://en.wikipedia.org/wiki/JavaBeans) as the model and Java servlets (or a framework such as [Apache Struts](https://en.wikipedia.org/wiki/Apache_Struts)) as the controller. This is a type of [Model 2](https://en.wikipedia.org/wiki/JSP_model_2_architecture) architecture.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than [machine code](https://en.wikipedia.org/wiki/Machine_code). Like any other Java program, they must be executed within a [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) that interacts with the server's host [operating system](https://en.wikipedia.org/wiki/Operating_system) to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well.

The [Web container](https://en.wikipedia.org/wiki/Web_container) creates JSP implicit objects like request, response, session, application, config, page, page Context, out and exception. JSP Engine creates these objects during translation phase.

HTML:

HTML or Hyper Text Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

JDBC:

**Java Database Connectivity** (**JDBC**) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition) platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database). A JDBC-to-[ODBC](https://en.wikipedia.org/wiki/ODBC) bridge enables connections to any ODBC-accessible data source in the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) host environment. JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections. Here Oracle 1og database used.

**2.3.2 Admin Login part**

The Student Login part is designed using of Java, Swing & JDBC.

**Java:**

**Java** is a general-purpose [computer programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [byte code](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). As of 2016, Java is one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity), particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (which has since been [acquired by Oracle Corporation](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C++), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

Swing:

**Swing** is a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) [widget toolkit](https://en.wikipedia.org/wiki/Widget_toolkit) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is part of [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)'s [Java Foundation Classes](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC) – an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI [components](https://en.wikipedia.org/wiki/Software_component) than the earlier [Abstract Window Toolkit (AWT)](https://en.wikipedia.org/wiki/Abstract_Window_Toolkit). Swing provides a native [look and feel](https://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](https://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element.

Though Swing is intended to be replaced by [JavaFX](https://en.wikipedia.org/wiki/JavaFX), it will remain part of the Java SE specification for the foreseeable future

JDBC:

**Java Database Connectivity** (**JDBC**) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition) platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database). A JDBC-to-[ODBC](https://en.wikipedia.org/wiki/ODBC) bridge enables connections to any ODBC-accessible data source in the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) host environment. JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager.

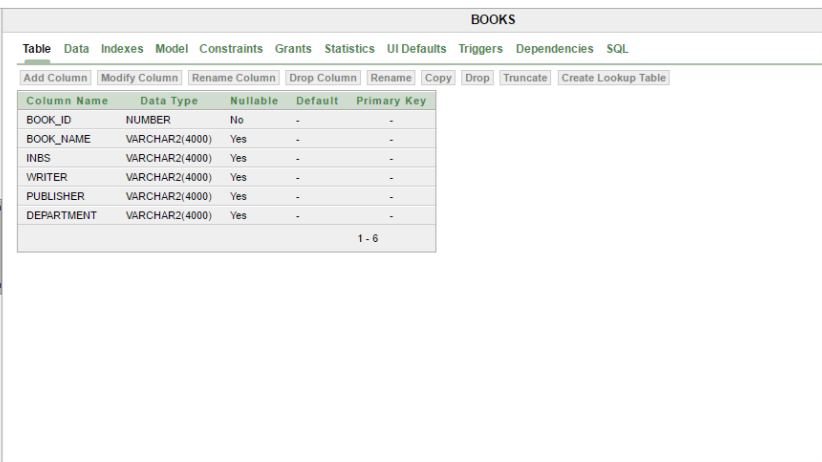
**CHAPTER 3**

**SYSTEM DESIGN**

* 1. **TABLE DESIGN**

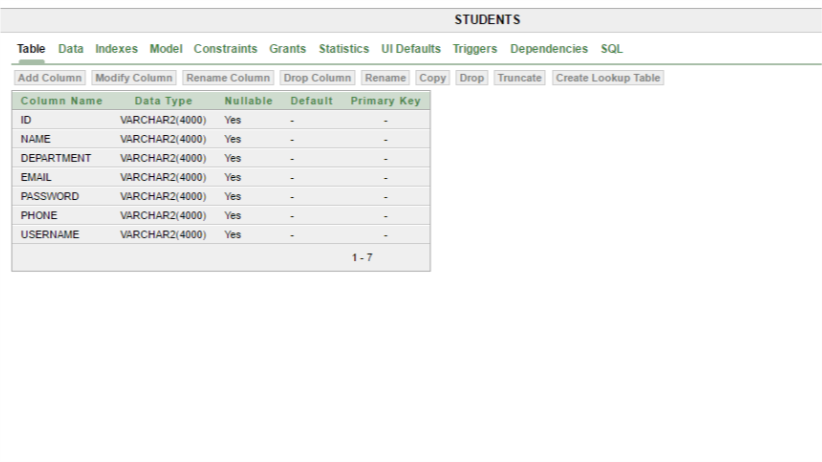
VARIOUS TABELS TO MAINTAIN INFORMATION:

BOOKS TABLE FOR KEEPING TRACK OF BOOKS

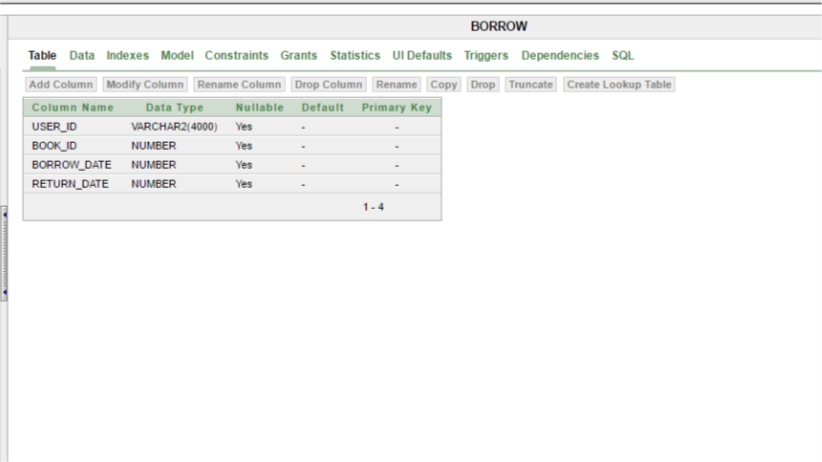




* STUDENT TABLE FOR STUDENT INFORMATION

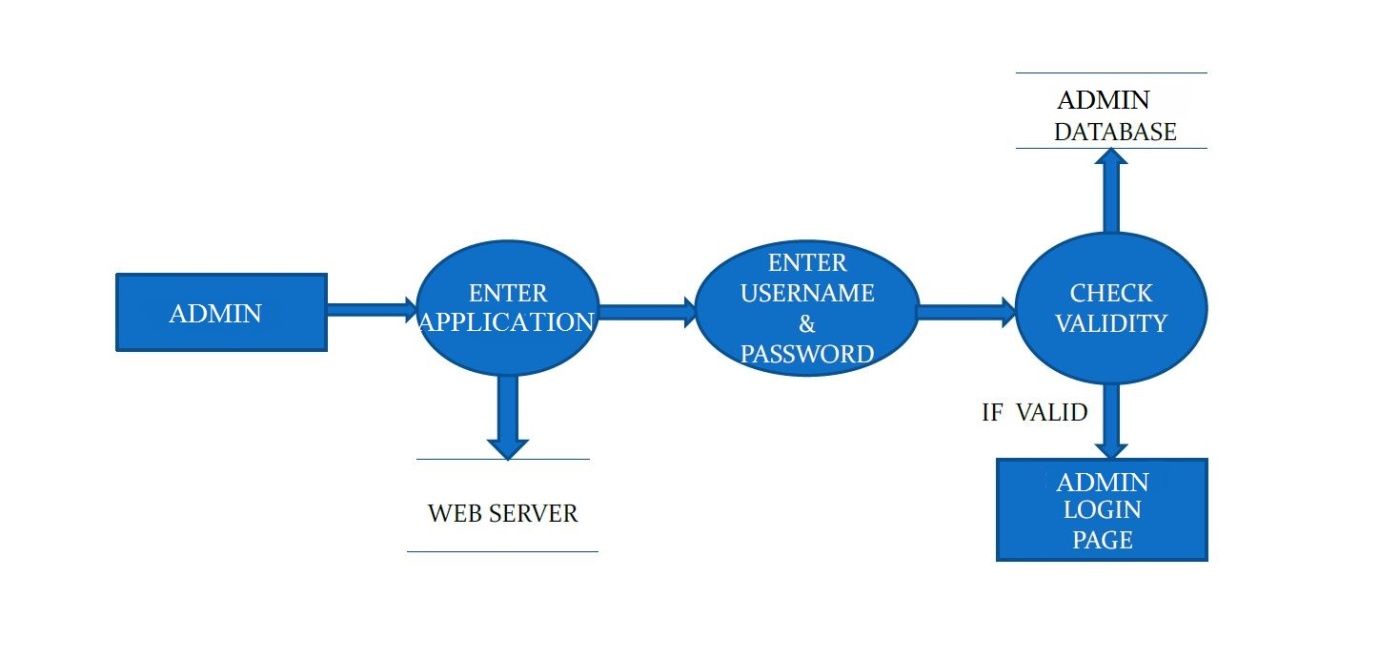


* BORROW TABLE TO KEEP BORROWED BOOK INFOR MATION



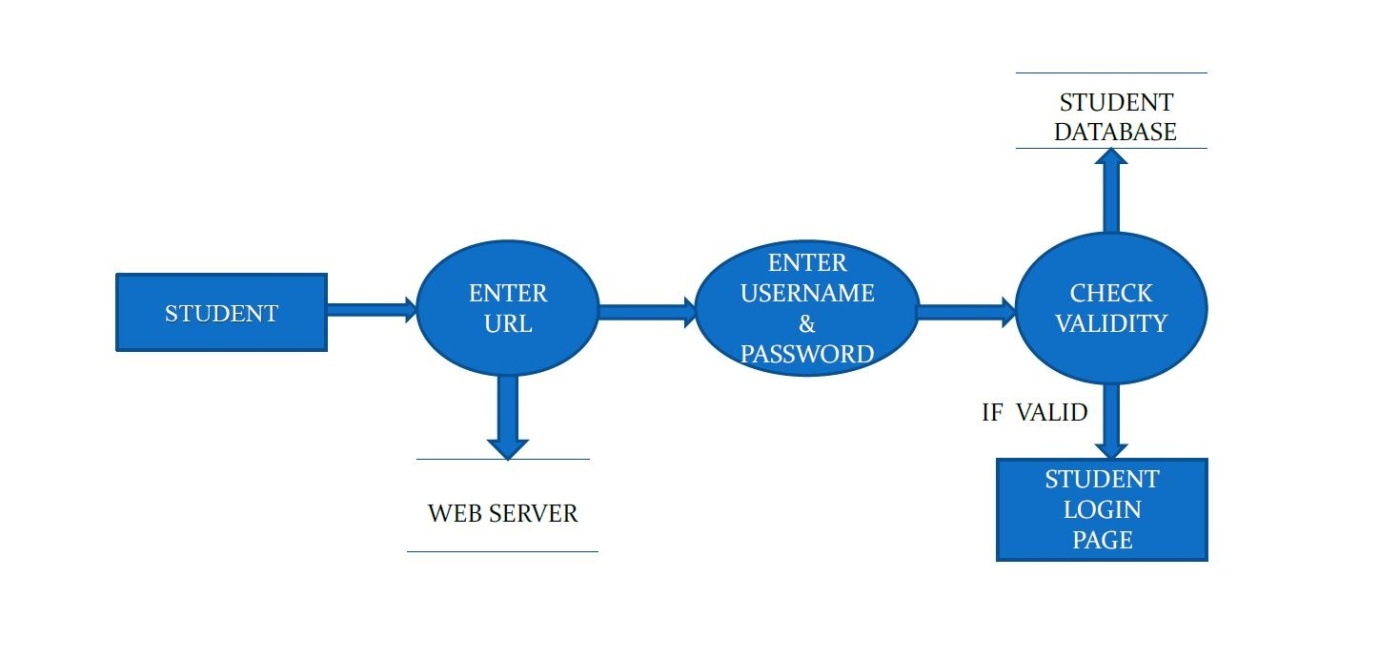
**3.2 DATA FLOW DIAGRAMS**

DATA FLOW DIAGRAM FOR ADMIN LOGIN



After entering to the application ADMIN can choose the ADMIN LOGIN option where they are asked to enter username & password , and if he/she is a valid user then a admin Main page will be displayed.

DATA FLOW DIAGRAM FOR STUDENT LOGIN



After entering to the home page of the website, student can choose the STUDENT LOGIN option where they are asked to enter username & password, and if he/she is a valid user then a student login page will be displayed

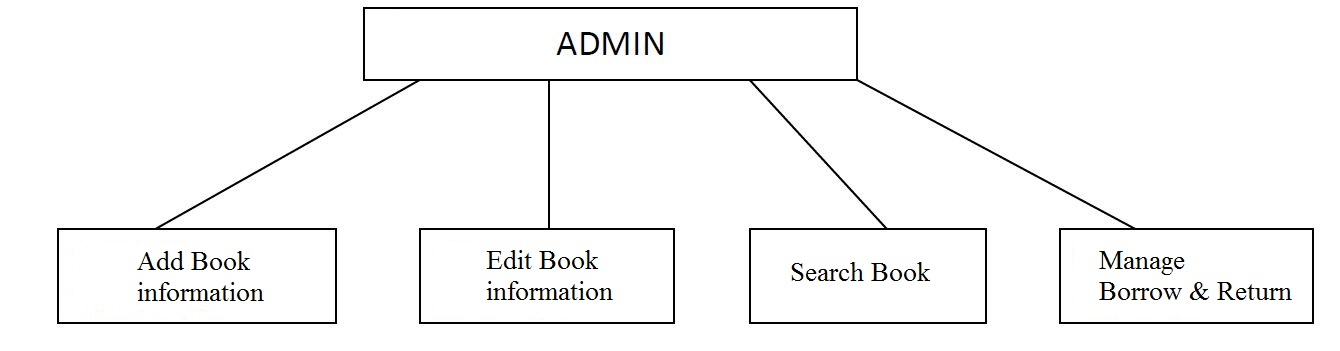
**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

**4.1 MODULE DESCRIPTION**

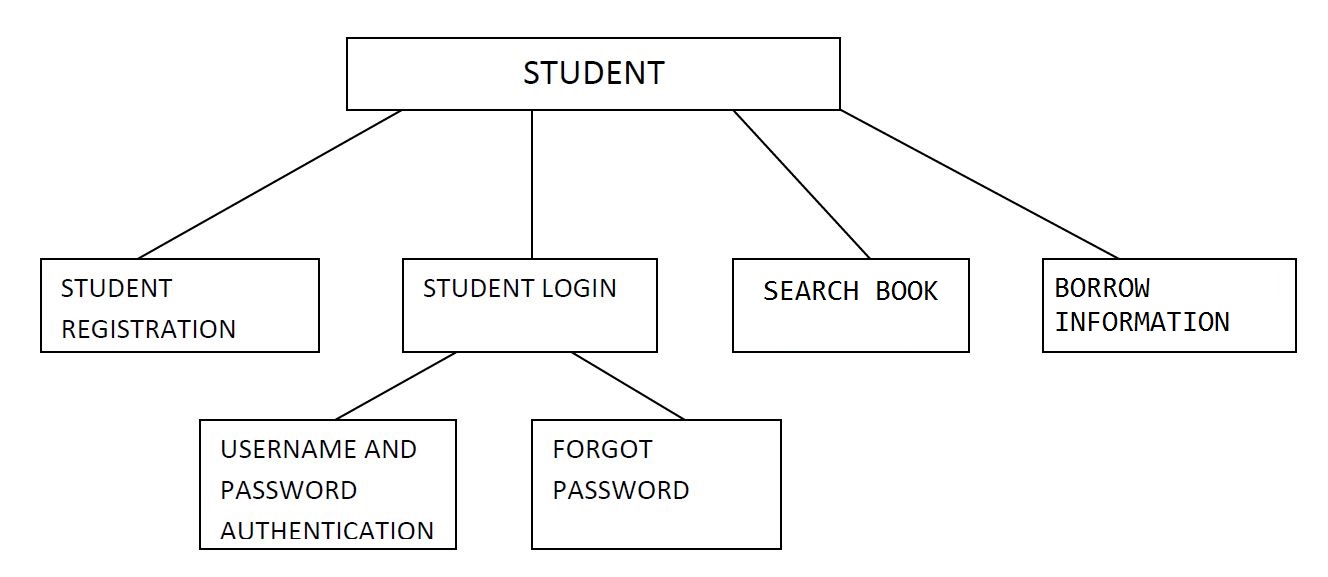
For Library Management System, it is divided into the following Modules:

**4.1.1 Admin Module**



The following module contains various facilities like add book information, update book information, searching books and maintain borrow & return procedures.

**4.1.2 Student Module**



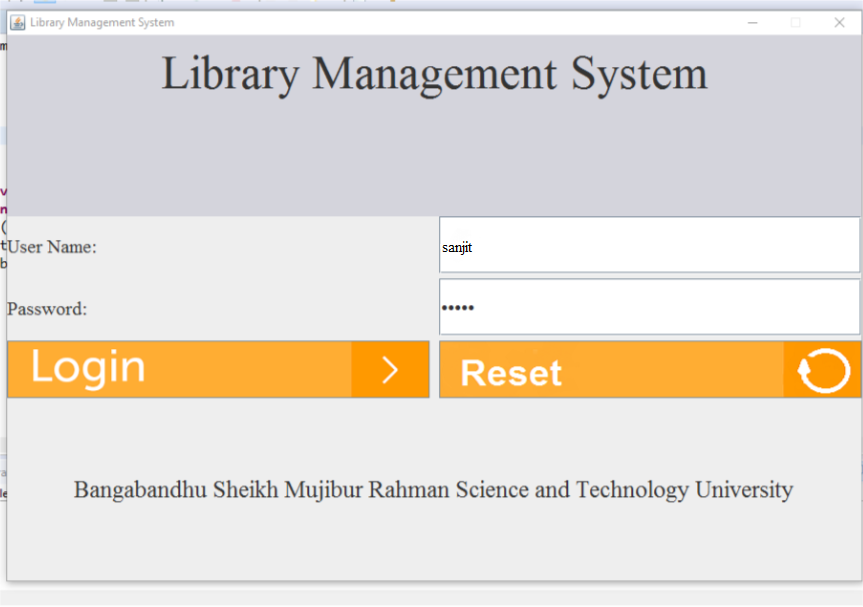
**4.2 User Manual**

For Library Management System it is divided into two user :

1. Admin login.
2. Student login.

**4.2.1 Admin Login:**

* At first admin has to login. The front page for admin is..



* After login, the admin will see this page.

Here options are

-Add book

-View book

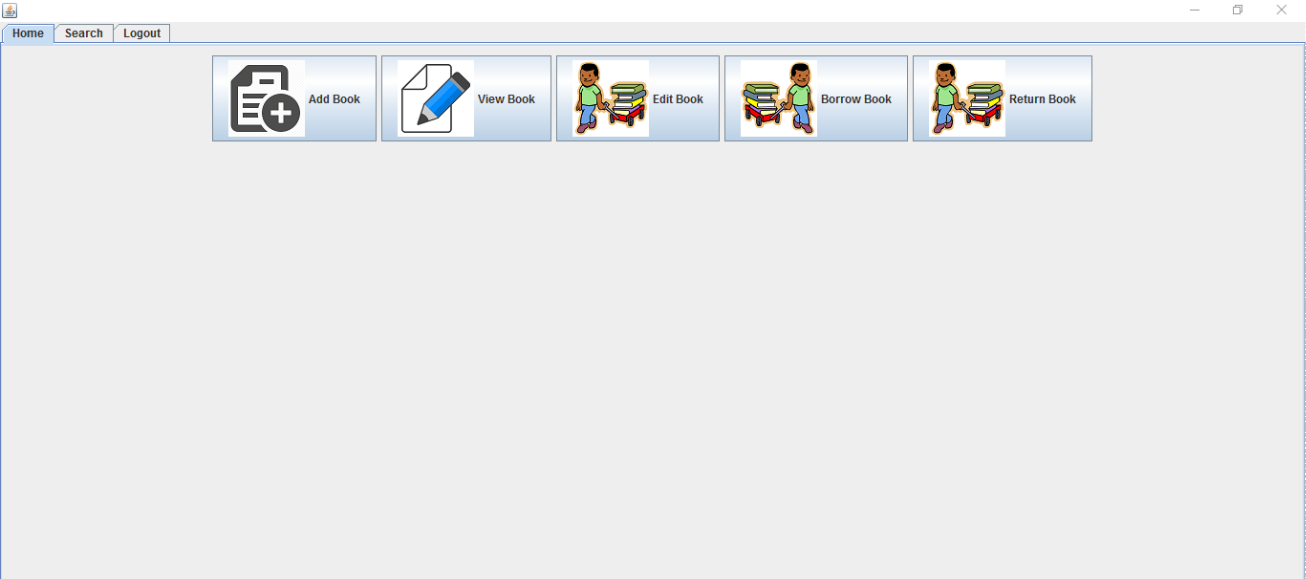
-Edit book

-Borrow book

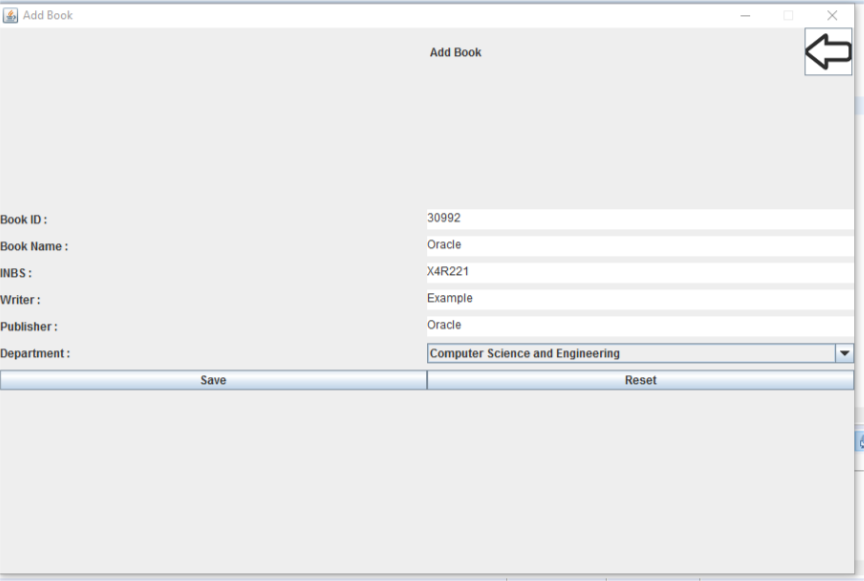
-Return book

-Search book

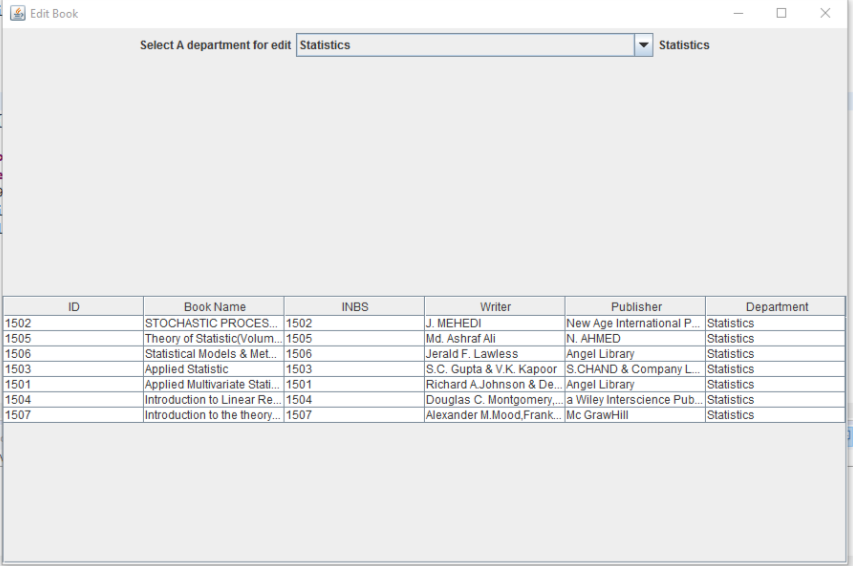
-Logout



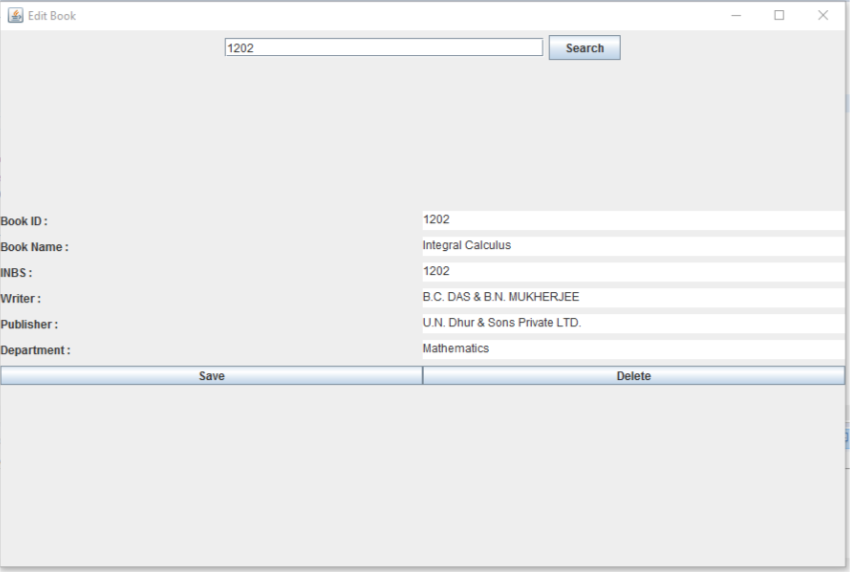
* Add book



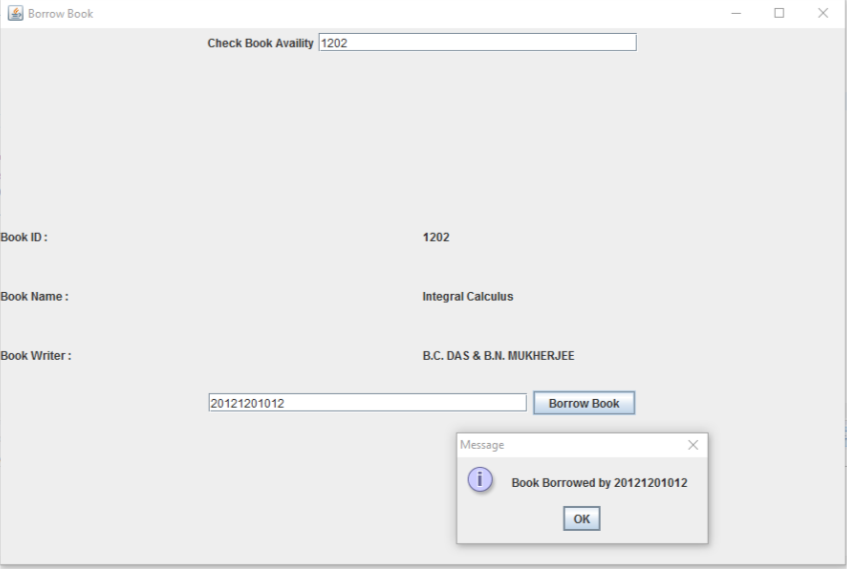
* View book



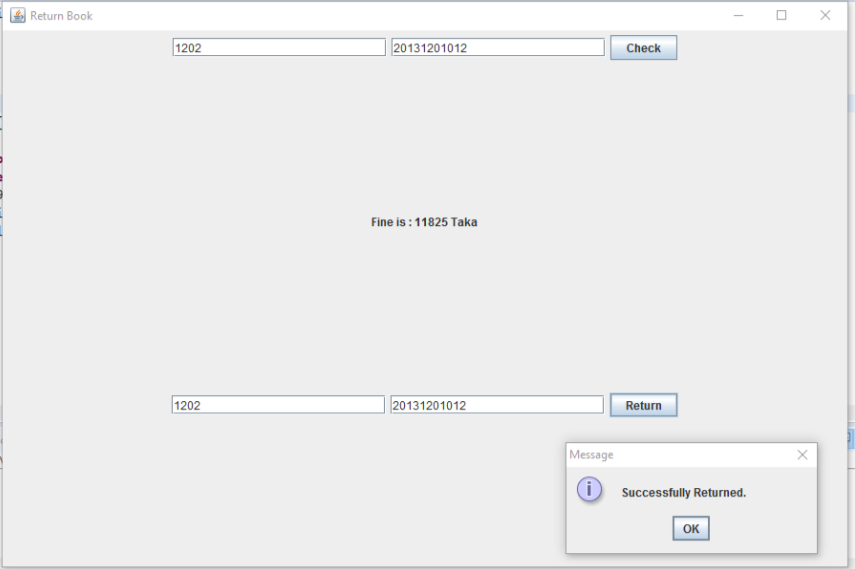
* Edit book



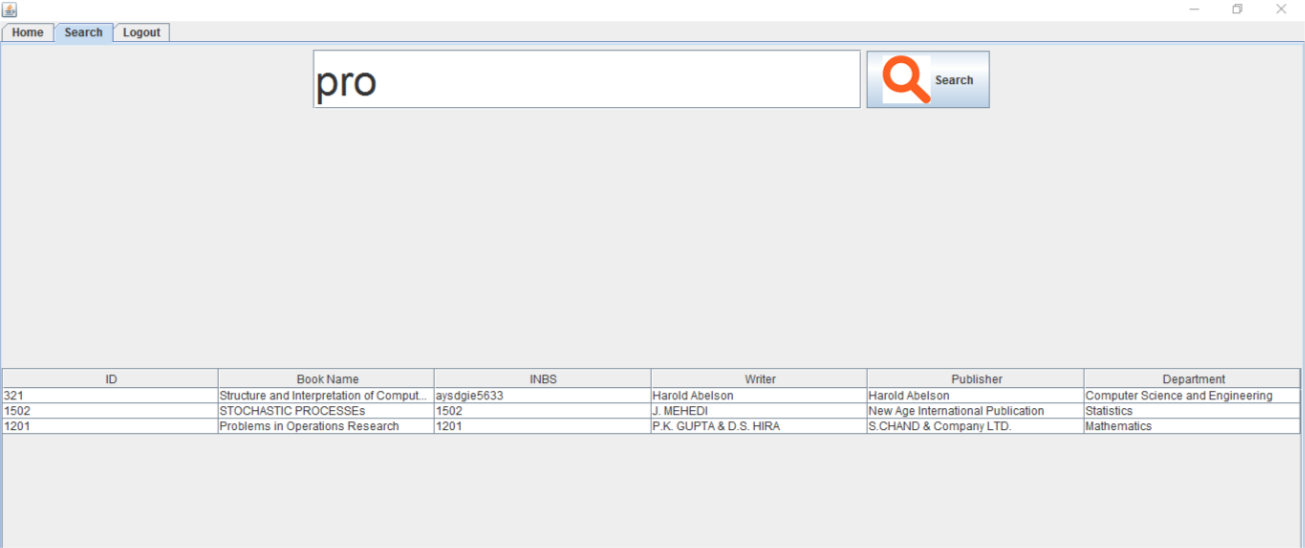
* Borrow book



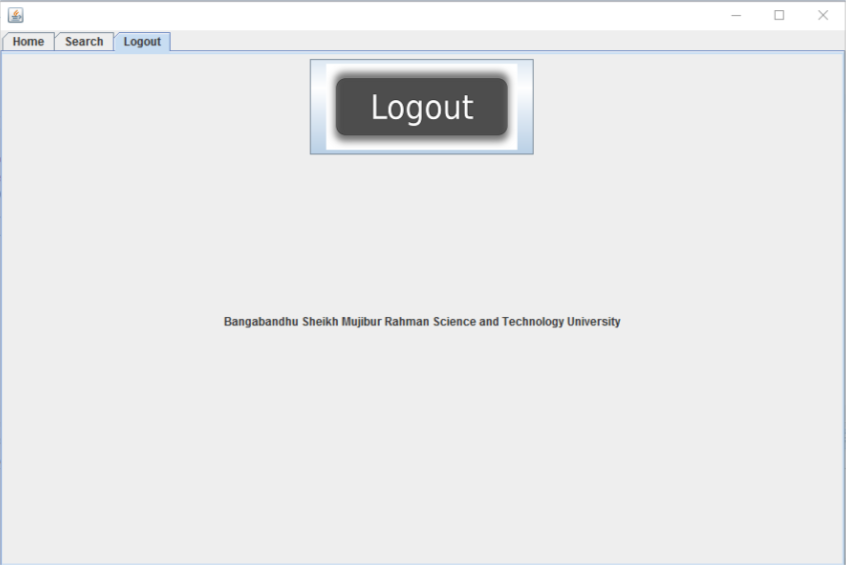
* Return book



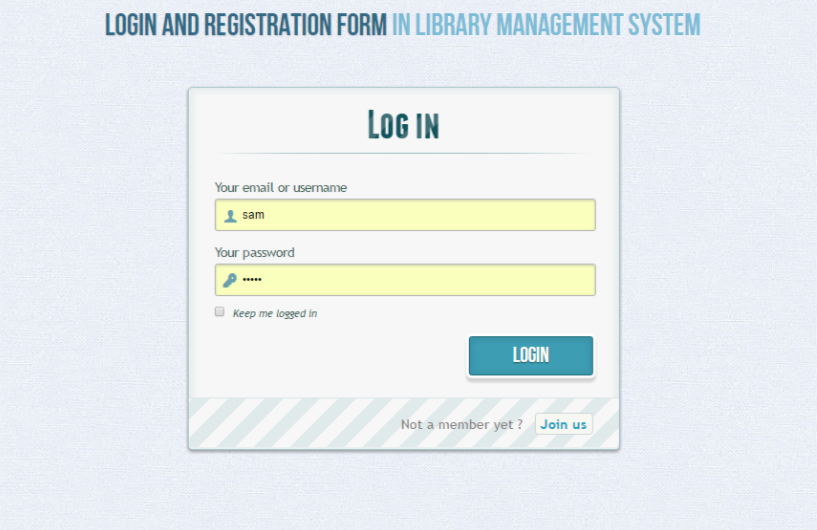
* Search book



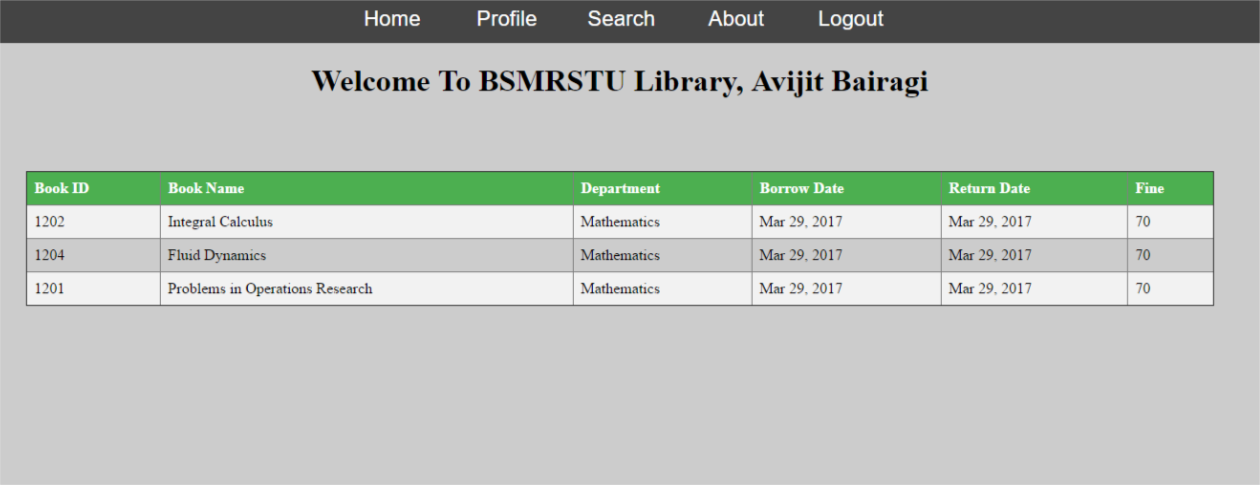
* Logout

**4.2.2 Student Login:**

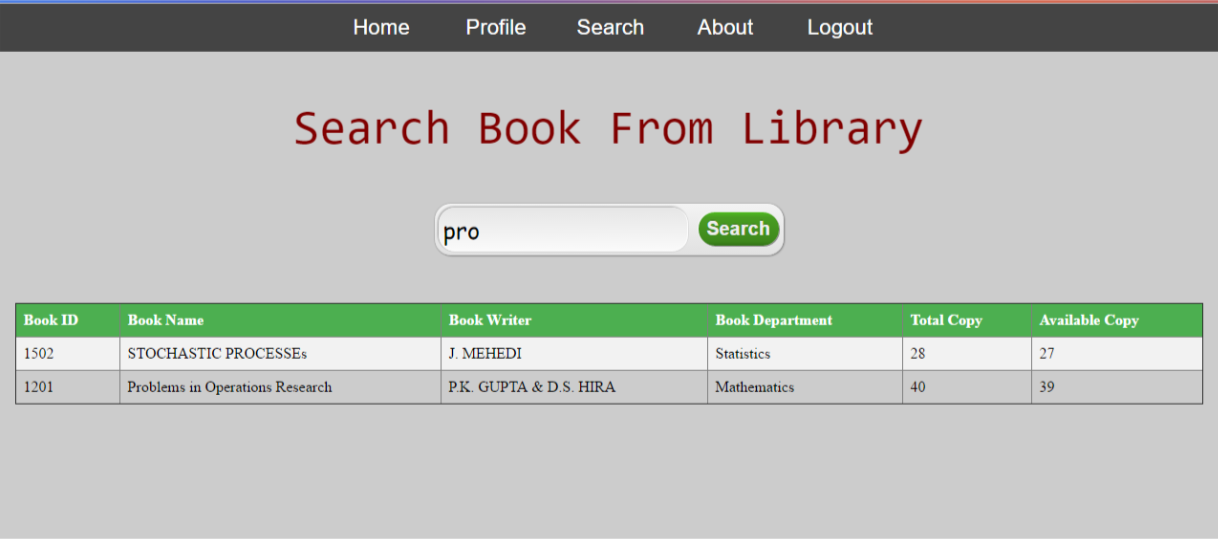
* At first student have to login. The front page for admin is..



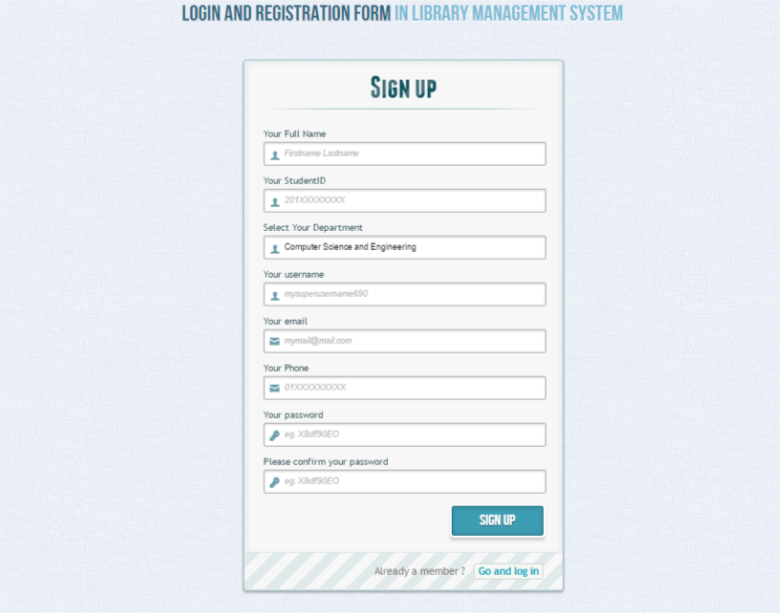
* After login the student will see this page



* Student can search book.



If any student don’t have user id then ,he will have to sign up.



**CHAPTER 5**

**LIMITATIONS & FUTURE SCOPES**

**5.1 Limitations**

Our project “**Library Management**” is recently made. It is updated but it has same limitations also. Those notifications are given below:

* “BAR CODE” scanning has not been added to this system.
* Notification system. Which is very much important for this system. It has not added yet. Student will get Email notification to return the book if late.
* JDBC is used instead of Hibernate.

**5.1 Future scopes**

In future, we will try to remove its limitations. We will also try to modify our projects & it will be updated, newest system. We will also try to add new features for the student & admin.

**CHAPTER 6**

**CONCLUSION**

This system provides a computerized version of library management system which will benefit the students as well as the staff of the library.

It makes entire process online where student can search books; staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of admin login where they can search book, add book, update book etc.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility, a feature of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfills each user need in the best way possible.