1.) INTRODUCTION

Library management system is designed to help the library to manage various details pertaining to books. This is basically developed for the authorities of the school/institute or any independent library to make their task easier or we can say this automate their tasks like maintaining students library details, maintaining book detail and many more. This package helps the administrative & library department in maintaining the students' library and books record as well as details. It provides us a simple interface for maintenance book record and details. This system focuses to present information in an easy and intelligent way. This system is effectively used to fetch data in a very short duration of time.

2.) PURPOSE

This project Library Management System has been developed to help the departments maintaining the book (present in institute or library) details, earlier the records were maintained manually, with the help of this project the concerned departments will be able to improve the productivity, reduce the time, cost factors associated with this system.

The automation of this system will help the organization or libraries in proper maintenance of the record of books, less manpower, less man-days, less cost, proper and accurate functioning.

The basic need for that is to automate the whole procedure of maintaining of books details, earlier it was all done manually. It removes a lot of burden from the departments, which was maintaining the book's details improved the efficiency, reduced the cost, and reduced the time need to do the work manually. With the help of this the past details of the books and the book borrower can assessed and reports can be generated on this details.

In brief we can say that this system was required to automate the processing of books details, which was done manually before the development of this package.

Earlier all the information / data pertaining to the books was maintained manually or we can say on paper, hence it created a problem for the organization / school or libraries, how to manage it properly. With the help of this system the organization/school or libraries are able to maintain the data properly & accurately.

3.) PROBLEM STATEMENT

- Earlier the data pertaining to books was maintained manually.
- Manual system was not efficient.
- Cost of maintaining data manually was bigger and huge.
- Large manpower was required.
- The procedure was error prone, it was not accurate.
- Need large data base to maintain record, which take extra space.
- As the time passes the data become too much, that it becomes harder to maintain it manually.

4.) METHODOLOGY AND IMPLEMENTATION:-

REQUIREMENTS:

Tools & platform:

- Basic used software for making this project are java swing, java database collector (JDBC).
- For saving the data of students, the MySQL database.
- To access the website, we can use any kind of web browser.

Modules used in this project:

- **Users** -> This table consists of the columns {UID, Username, Password, Admin}
- **Books**-> The book's table consists of the columns {BID, Book name, Price, Genre}
- **Issue** -> This table consists of the columns {IID, UID, BID, Issue Date, Period, Return Date, Fine}

Project Category:

- It is a web based application.
- It reduces the man power and manual paper works used to maintain the records.
- Improves management and control of the data.

SPECIFICATIONS:

Software Requirements:

Name of component	Specification
Operating System	Windows 9,8,Windows XP, Windows 7, Linux
Language	Java 2 runtime environment
Database	My SQL Server
Browser	Any of Mozilla, Opera, Chrome

Name of component	Specification
Processor	Pentium III 630MHz
RAM	128MB
Hard Disk	20GB
Monitor	15N color monitor
Keyboard	122 keys

JAVA RUNTIME ENVIRONMENT

The Java Runtime Environment (JRE) is a set of software tools for development of java applications. It combines the Java Virtual Machine (JVM), platform core classes and supporting libraries.

JRE is part of the Java Development Kit (JDK), but can be downloaded separately. JRE was originally developed by Sun Microsystems Inc., a wholly-owned subsidiary of Oracle Corporation.

JRE consists of the following components:

1. Deployment technologies, including deployment, Java Web Start and Java Plug-in.

2. User interface toolkits, including Abstract Window Toolkit (AWT), Swing, Java 2D, Accessibility, Image I/O, Print Service, Sound, drag and drop (DnD) and input methods.
3. Integrating libraries, including interface Definition Language (IDL), Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI), Remote Method Innovation (RMI) and scripting.
4. Lang and util base libraries, including lang and util, management, versioning, zip, instrument, reflection, Collections, Concurrency Utilities, Java Archive (JAR), Logging, Preferences API, Ref Objects and regular expressions.
5. Java Virtual Machine (JVM), including Java Hotspot Client and server virtual machines.

JRE 1.0 has evolved with a variety of class and package additions to the core libraries, and it has grown from a few hundred classes to several thousand in Java 2 Platform, Standard edition (J2SE).

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter / loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

Java developers are initially presented with two JDK tools, java and javac. Both are run from the command prompt. Java source files are simple text files saved with an extension of .java. After writing and saving Java source code, the javac compiler is invoked to create .class files. Ones the .class files are created, the 'java' command can be used to run the java program. For developers who wish to work in an integrated development environment (IDE), a JDK bundled with Net beans can be downloaded from the Oracle website. Such IDEs speed up the development process by introducing point-and-click and drag-and-drop features for creating an application.

Java Database Connectivity (JDBC)

JDBC is an API for the java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. From a

technical point of view, the API is a set of classes in the java.sql package. To use JDBC with a particular database, we need a JDBC driver for that database.

JDBA is a cornerstone for database programming in java. Today, it is considered to be very low-level and prone to errors. Solutions such as MyBatis or JDBC Templates were created to ease the burden of JDBC Programming. JDBC is part of the java Standard Edition platform.

JDBC manages these three main programming activities:

- Connecting to a database;
- Sending queries and update statements to the database;
- Retrieving and processing the results received from the database in answer to the query.

MYSQL J Connector:

To connect to MySQL in java, MySQL provides MySQL Connector/J, a driver that implements the JDBC API. MySQL Connector/J is a JDBC Type 4 driver. The type 4 designation means that the driver is a pure java implementation of the MySQL

protocol and does not rely on the MySQL client libraries.

NETBEANS

Net Beans is an integrated development environment (IDE) for Java. Net Beans allows applications to be developed from a set of modular software components called *modules*. Net Beans runs on Windows, mac, OS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on Net Beans, including the net Beans IDE, can be extended by third party developers

The Net Beans Platform is a framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plugins and Net Beans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (carries out efficient storage)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated development tools

Java Swing

Swing is a GUI widget toolkit for Java.[1] It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a 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

Since early versions of Java, a portion of the Abstract Window Toolkit (AWT) has provided platform-independent APIs for user interface components. In AWT, each component is rendered and controlled by a native peer component specific to the underlying windowing system.

By contrast, Swing components are often described as *lightweight* because they do not require allocation of native resources in the operating system's windowing toolkit. The AWT components are referred to as heavyweight components.

SOURCE CODE:

The source code can be divided into 5 main parts:

1.) Login

2.) Connect

3.) Create/Reset

4.) User Menu

5.) Admin Menu

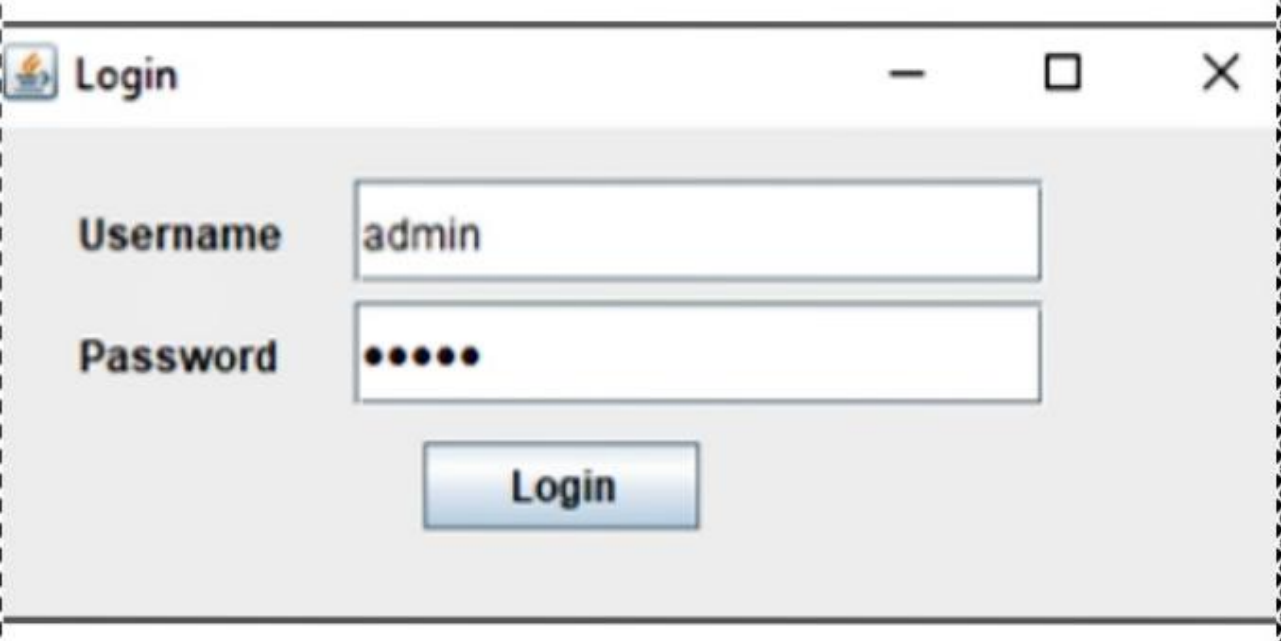
In our project (java) we are going to use the following packages in our source code:

- **Import java.awt.event.ActionEvent;**
- **Import java.awt.event.ActionListener;**
- **Import java.sql.*;**
- **Import java.text.DateFormat;**
- **Import java.text.ParseException;**
- **Import java.text.SimpleDateFormat;**
- **Import java.util.ArrayList;**
- **Import java.util.Date;**
- **Import java.util.Locale;**
- **Import java.util.concurrent.TimeUnit;**
- **Import javax.swing.*;**
- **Import net.proteanit.sql.DbUtils;**

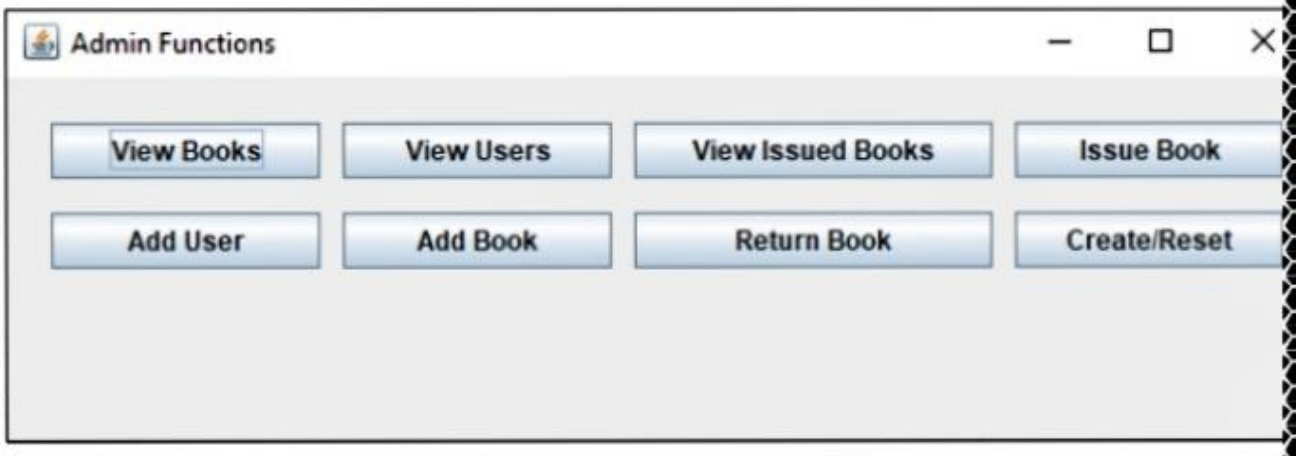
5.) RESULT:

The project will help us to maintain the book records and details and the output will resembles as follows:

a.) Login: here the user will login



b.) Admin functions:
After login admin functions will appear:



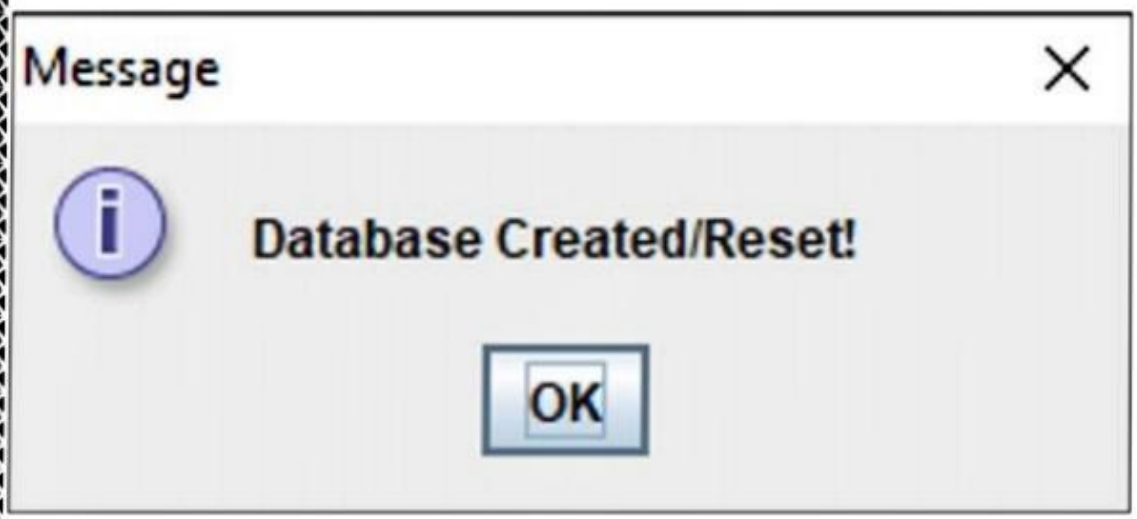
c.) After that chooce any function as per the requirements,all other outputs resembels as follows:

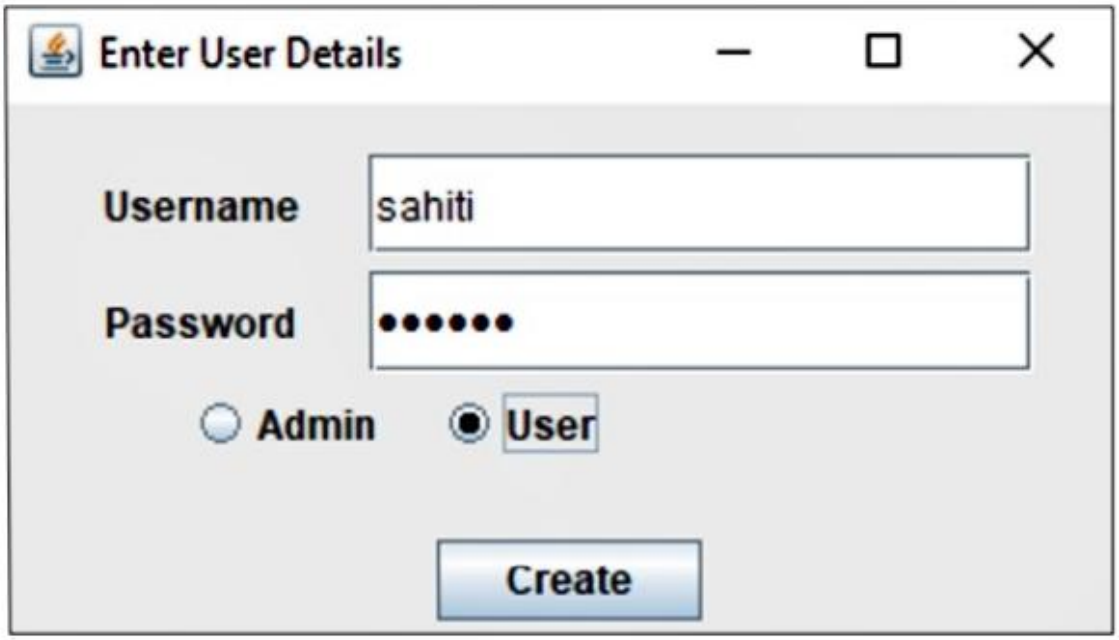
BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

i) Books available

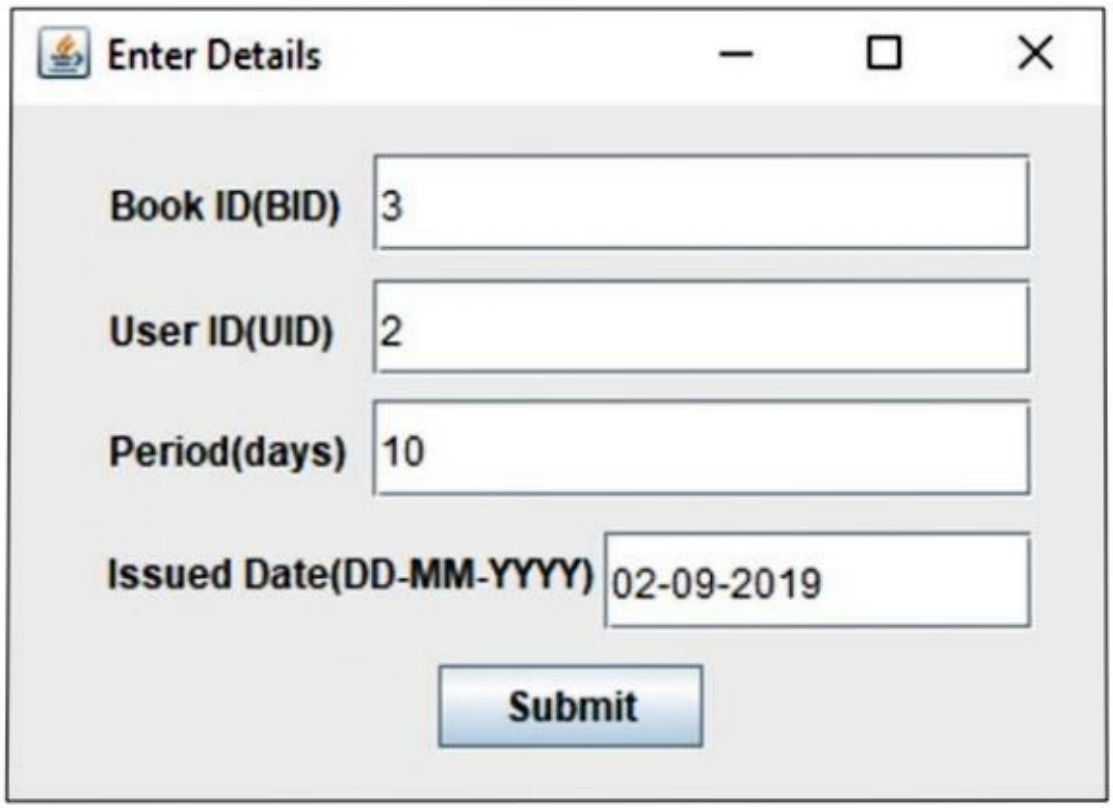
ii) View users

iii) Create/reset

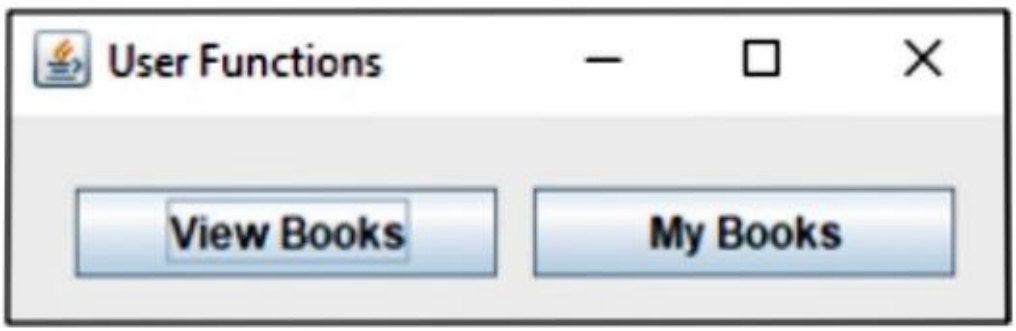



Enter User Details
Username
sahiti
Password
Admin
User
Create

iv) Add user



v) Issue book



vi) User Functions

vii) Add book/book details
Etc.

6.) USES

a.) It increases efficiency
b.) It reduces the cost of managing a library
c.) It saves time
d.) It increases the productivity of library workers
e.) It enhances the presentation of the library

7.) CONCLUSION:

This paper assists in automating the existing manual system. This is a paperless work. It reduces the man power requirement. It provides accurate information always. Malpractice can be reduced. All years together gathered information can be saved and can be accessed at any time. The data which is stored in the repository helps in taking the intelligent decisions by the librarians/users. So it is better to have a web based information management system. All the user and admin can get the required information without delay. This system is essential in colleges / universities/ libraries.

As for the conclusion, the objectives for this project were achieved and functioned well as the desired target. This system will help the Book Information System database works systematically and will make ease the user in order to manage all the books records/details data in the system. This system will give a better performance in arranging a book information without having to do it manually. As for the future recommendation, the project is

recommended to be built with the fully functional software that fulfills all the criteria needed and also applied with more complicated algorithm to the system.

8.) FUTURE SCOPE:

- In the future, users can also be able to lend as well as borrow book online.
- We will see the entire system more accurate and also be able to give statics data about book records.
- It provides a secure system for the admin (mainly librarians) to store the data.
- It maintains and store the user, admin and book information as well as record in a secure accessible system.
- Provide an easy method to the admin to edit the details of the books and users whenever needed.

a) https://www.javatpoint.com/
b) **Herbert Schildt**, Java: A Beginner's Guide, published by oracle
c) E Balagurusamy , Programming With Java A Primer, published by Mc Graw Hill