# Project Report

On

**STUDENT MANAGEMENT SYSTEM**

A Desktop Application

Submitted in partial fulfillment of the requirement

for the award of the three year diploma in

COMPUTER ENGINEERING

Under the guidance of

Mr. H S Bhatia

****

**AMBEDKAR INSTITUTE OF TECHNOLOGY**

Shakarpur Near Nirman Vihar Metro Station Delhi-110092

(SESSION: 2017-2020)

**Submitted by: - Guided by: -**

Suraj Mamgai Mr. H S Bhatia

BTE Roll No: 1702051062 ( Project In-Charge)

(SSPL-DRDO, Intern)

**AMBEDKAR INSTITUTE OF TECHNOLOGY**

SHAKARPUR, NEAR METRO STATION, DELHI-110092

## (DIRECTORATE OF TRAINING & TECHNICAL EDUCATION)

GOVT. OF NCT OF DELHI

**CERTIFICATE**

This is certified by the project report entitled “STUDENT MANAGEMENT SYSTEM” which is submitted by **MR. SURAJ MAMGAI** to **AMBEDKAR INSTITUTE OF TECHNOLOGY,** a record of bonafide project work carried out by him under our supervision and guidance and submitted for partial fulfillment to award of diploma in Computer Engineering

**Mr. H S BHATIA** (SUPERVISOR)

**Acknowledgement**

To get practical knowledge and to make a report on a certain topic without anybody’s help is really a task of great difficulty, fortunately, in preparation of this project report I got the help of many known as well as unknowns for providing the valuable information about each and every stage of this project.

First of all, I am extremely thankful to the **DIPLOMA (Computer Engineering) department of AMBEDKAR INSTITUTE OF TECHNOLOGY, Rohini** for giving me enough time and support to complete my project.

I consider myself privileged having found an opportunity to express our heartfelt thanks to our head of the Computer Science Engineering Department.

I am grateful to my project guide **Mr. H S Bhatia** for the guidance, inspiration and constructive suggestions that were helpful to me in the preparation of this project.

I am also thankful to my colleagues who have helped in successful completion of this project.

**Abstract**

“STUDENT MANAGEMENT SYSTEM” is a desktop-based application to manage the database of all the students in a college within a single system. This project will handle the whole system of the college which is related to the students.

The Student Management System has most of the facilities to keep the records of students, teaching, feeds, notice, library and non- teaching stuff with all their required details along with all required daily activities. It has most of the facilities that a modern college requires to computerize its day-to -day jobs.

It comprises Administrator as well as User panel with different access levels to perform various functions.

**INTRODUCTION**

This project on “Student Management System” is useful for easy user interface. The system utilizes powerful database management, data retrieval and data manipulation. This project provides more ease for managing the data than manually maintaining it in the documents. The project is useful for saving valuable time and reduces the huge paperwork.

It will help educational Institutions like schools and colleges will keep track of their student records like personal details, contact details, marks details, etc. The Internet is rapidly becoming a part of the everyday lives of a majority of people in the world. People perform various activities on the Internet and one of them is storing their data in a data-base where they are interested in. In these databases they can post the queries and they can retrieve the required data. Obviously, there is a need for Student Information System software for management of student’s data.

There are many departments of administration for the maintenance of college information and student databases in any institution. All these departments provide various records regarding students. Most of these track records need to maintain information about the students. This information could be the general details like student name, address, performance, attendance etc. or specific information related to departments like collection of data. All the modules in college administration are interdependent. They are maintained manually. So, they need to be automated and centralized as, Information from one module will be needed by other modules.

**REQUIREMENT ANALYSIS**

1. Introduction

The following subsections are an overview of the entire Software Requirements Specification (SRS).

* 1. Purpose

This document provides the technical description of all software requirements of STUDENT MANAGEMENT SYSTEM.

The document will not only define the product functions, user characteristics, constraints, and specific requirements of the system, but also serve as a basis for the Software Design Document that is prepared according to IEEE Std.

The objective of the software is to maintain information pertaining to the students with the purpose of: -

* Planned approach towards working
* Accuracy
* Reliability
* No Redundancy
* Immediate retrieval of information
* Immediate storage of information
* Easy to Operate
  1. Scope

“STUDENT MANAGEMENT SYSTEM” is a project with a mission of viewing and manipulating student information of Ambedkar Institute of Technology in a Web-based environment. Thus, the overall system will consist of a Student Database System and Web Interface.

The Student Database System will supply the fundamental database structure of the entire system whereas Web Interface will provide a secure Web interface between the users and the database.

The Software aims to create a “paperless institute” rather than using a traditional record keeping system.

Although this project is presently being designed specifically for Ambedkar Institute of Technology but there exists the possibility in future to upgrade it to general level.

The software will not only help the following levels of users in viewing the information but also each user can alongside update changes within their respective access limits.

* Administrative Level
* User Level
* Management of Institute
* Faculty
* Students
* Department Staff

1.3 Definition

Paperless Office: refers to an integrated working environment where all the data and documentation is represented in electronic format.

Student Personal Information: refers to personal records of individual student’s biodata along with his performance throughout the course. Traditional Record Keeping System: refers to a manual system where all records are kept on papers by manual in-charge.

* 1. References

Pressman, Roger S., Software Engineering “A Practitioner’ s Approach”, Fifth Edition, McGraw-Hill, 2000.

Software Engineering by K.K Aggarwal and Yogesh Singh

1.5 Overview

This document is prepared in accordance with the IEEE Std, IEEE Recommended Practice for Software Requirements Specifications.

It also provides product perspectives, product functions, user characteristics, general constraints, and assumptions and dependencies of the system.

It will contain functional and performance requirements, design constraints, attributes and external interface requirements for the Software.

**PLATFORM (TECHNOLOGIES/TOOLS)**

**The JAVA and the Swing framework**

Java is one of the most popular and widely used programming languages and platforms. A platform is an environment that helps to develop and run programs written in any programming language.

Java is fast, reliable and secure. From desktop to web applications, scientific supercomputers to gaming consoles, cell phones to the Internet, Java is used in every nook and corner.

Overview of the JAVA LANGUAGE

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

* Mobile applications (especially Android apps)
* Desktop applications
* Web applications
* Web servers and application servers
* Games
* Database connection
* And much, much more!

**Why Use Java?**

* Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
* It is one of the most popular programming languages in the world
* It is easy to learn and simple to use
* It is open-source and free
* It is secure, fast and powerful
* It has a huge community support (tens of millions of developers)

**Java** is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is [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](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) "[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 ["bytecode"](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 the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of Java is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

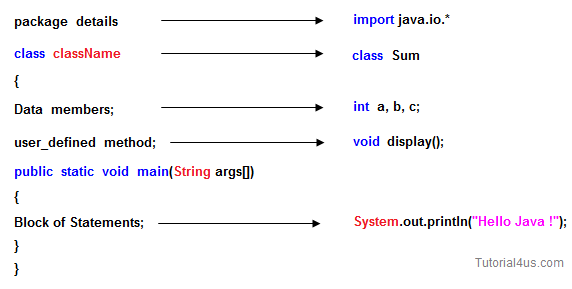
As of 2018, Java was according to [GitHub](https://en.wikipedia.org/wiki/GitHub) one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity), particularly for [client-server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers.

## 

## Structure of Java Program

Structure of a java program is the standard format released by a Language developer to the Industry programmer.

Sun Micro System has prescribed the following structure for the java programmers for developing java application.



* A **package** is a collection of classes, interfaces and sub-packages. A sub package contains a collection of classes, interfaces and sub-sub packages etc. java.lang.\*; package is imported by default and this package is known as default package.
* **Class** is a keyword used for developing user defined data types and every java program must start with a concept of class.
* **"ClassName"** represents a valid variable name treated as a name of the class each and every class name in java is treated as user-defined data type.
* **Data members** representing either instance or static will be selected based on the name of the class.
* **User-defined** methods represent either instance or static; they are meant for performing the operations either once or each and every time.
* Each and every java program starts execution from the main () method. And hence main () method is known as program driver.
* Since the main () method of java is not returning any value and hence its return type must be void.
* Since the main () method of java executes only once throughout the java program execution and hence its nature must be static.
* Since main () method must be accessed by every java programmer and hence whose access specifier must be public.
* Each and every main () method of java must take an array of objects of String.
* **Block of statements** represents a set of executable statements which are in term calling user-defined methods are containing business-logic.
* The file naming conversion in the java programming is that which-ever class is containing the main () method, that class name must be given as a file name with an extension .java.

## Compile and Run Java Program: It's Two Step Process

Compilation and execution of a Java program is a two-step process. During the compilation phase Java compiler compiles the source code and generates *bytecode*. This intermediate *bytecode* is saved in the form of a .class file. In second phase, Java virtual machine (JVM) also called Java interpreter takes the .class as input and generates output by executing the *bytecode*. Java is an object-oriented programming language; therefore, a program in Java is made of one or more classes. No matter how trivial a Java program is, it must be written in the form of a class.

## Compile Java Program from Command Prompt

Once the Java program is written and saved, first, it has to be compiled. To compile a Java program from the command line we need to invoke the Java compiler by supplying the javac command. Java compiler comes with JDK (Java Development Kit). JDK is a bundle of software needed for developing Java applications. It includes the JRE (Java Runtime Environment), set of API classes, Java compiler, Webstart and additional files needed to write Java applets and applications.

## Run Java Program from Command Prompt

After successful compilation of HelloWorld.java to HelloWorld.class to actually run the program, we use the Java interpreter, called java. To do so, pass the class name HelloWorld as a command-line argument, as shown follows:

[root host ~] # java HelloWorld

The message Hello World! will be printed on the screen as a result of the above command.

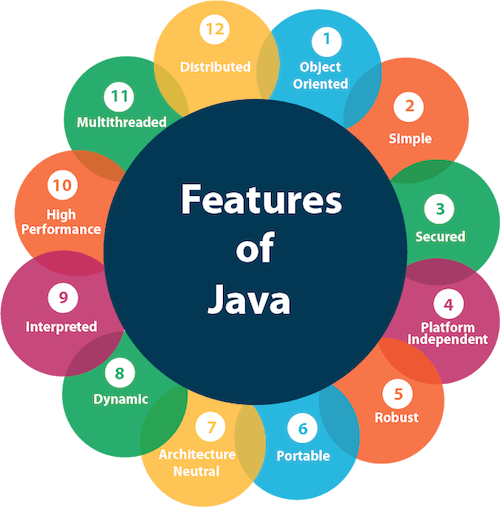
It is important to note that in above command we have omitted the .class suffix of the byte-code file name (that is HelloWorld.class in our case). The java command invokes the *Java Virtual Machine* (will be written JVM hereafter). JVM then loads the specified class mentioned at the command line and invokes the method main of this class and starts executing it, passing it a single argument that is an array of strings. The array of strings passed to main is to receive command line arguments. The JVM generally takes the following steps in order to run a Java program.

# **Features of Java**

The primary objective of [Java programming](https://www.javatpoint.com/java-tutorial) language creation was to make it a portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as java *buzzwords*.

A list of most important features of Java language is given below.

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed
12. Dynamic



**TECHNOLOGY USED**

**Front-end**

**Swing Framework: -**

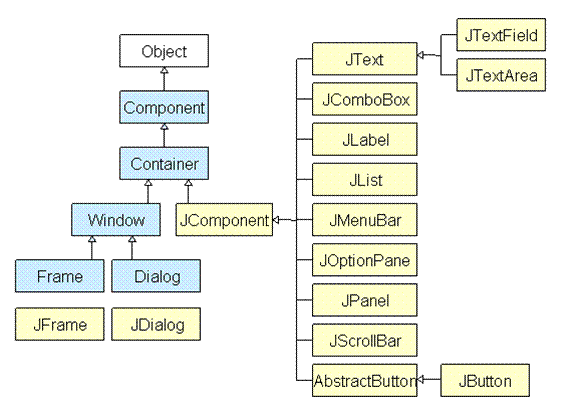
Swing Framework contains a set of classes that provides more powerful and flexible GUI components than those of **AWT**. **Swing** provides the look and feel of modern Java GUI. Swing library is an official Java GUI tool kit released by Sun Microsystems. It is used to create graphical user interfaces with Java.

Swing classes are defined in javax.swing package and its sub-packages.

#### **Main Features of Swing Toolkit: -**

1. Platform Independent
2. Customizable
3. Extensible
4. Configurable
5. Lightweight
6. Rich Controls
7. Pluggable Look and Feel

#### **AWT and Swing Hierarchy**



**BACK END: MySQL- XAMPP SERVER**

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by Sun Microsystems (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)).[[8]](https://en.wikipedia.org/wiki/MySQL#cite_note-sunacquire-8) In 2010, when Oracle acquired Sun, Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB).

The MySQL server software itself and the client libraries use a dual licensing distribution. They are offered under GPL version 2 or a proprietary license

Xampp Server: -

XAMPP is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends,[[2]](https://en.wikipedia.org/wiki/XAMPP#cite_note-kaiseidlerinterview-2) consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language).[[3]](https://en.wikipedia.org/wiki/XAMPP#cite_note-x_mariadb-3)[[4]](https://en.wikipedia.org/wiki/XAMPP#cite_note-4) Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a [WAMP](https://en.wikipedia.org/wiki/WAMP) or [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) stack can be installed quickly and simply on an operating system by a developer. With the advantage a number of common add-in applications such as [WordPress](https://en.wikipedia.org/wiki/Wordpress) and [Joomla!](https://en.wikipedia.org/wiki/Joomla!) can also be installed with similar ease using [Bitnami](https://en.wikipedia.org/wiki/Bitnami).

**ER diagram**

An Entity-relationship model is an abstract conceptual representation of structured data. Entity relationship modeling is a relational schema database modeling method, used in Software Project Management to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in top-down fashion.













**Admin ER Diagram**

























**DATA FLOW DIAGRAMS**

A **data-flow diagram** (DFD) is a way of representing the flow of a data of a [process](https://en.wikipedia.org/wiki/Process) or a system (usually an [information system](https://en.wikipedia.org/wiki/Information_system)) The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a [flowchart](https://en.wikipedia.org/wiki/Flowchart).

**Data flow diagram for login Window**



username & verified /

password unverified

**Data flow diagram for registration Window**



Enter details Registration

Successful



saving details to database Checking users existence



**Data flow diagram User Application**











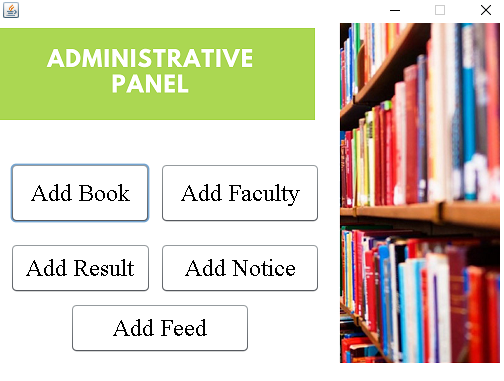


**DESIGN**

There are two different modules of the software:

1. Admin Module

* Welcome Window

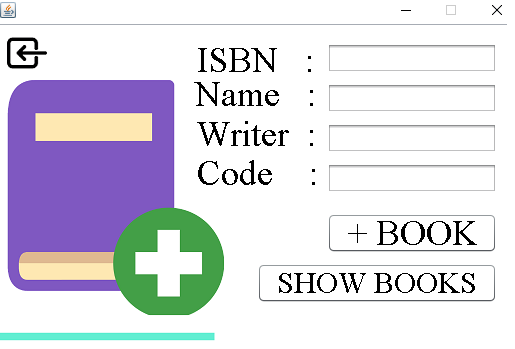


**Welcome Window**

* The Welcome Window in the administrator panel describes various functions that can be performed by the administrator.
* The functions of the administrator includes:
* Add Students Result
* Add a Notice
* Add a Book
* Add New Librarian
* Add Feeds
* The Administrator is solely responsible for populating, manipulating and managing databases connected to the software and the tables related to above mentioned functions.

Admin Module

* Add Book Window



**Add Book Window**

* In order to issue a book to a student, firstly the book has to be a part of the books table of the database.
* The Administrator is the person who adds the books by clicking to the “Add Book” button on the welcome Window.
* To add a book, the administrator has to add the following details about a book. So that the unique book can be identified easily.
* Book’s ISBN
* Book’s Name
* Book’s Writer
* Book’s Code
* After filling all the details the administrator has to click on the “+Book” button.
* After clicking on the button, the book gets added to the database. The administrator can also see the books in the database by clicking on the “Show Books” button.

Admin Module

* Add Faculty Window

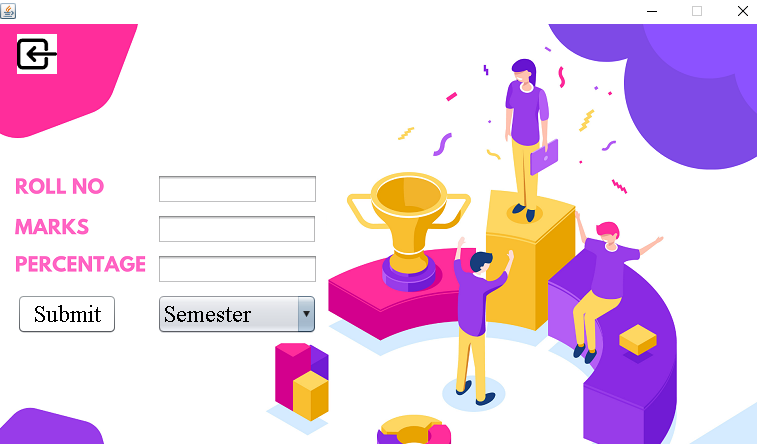
****

**Add Faculty/Librarian Window**

* In order to issue a book to a student, there should be a faculty/librarian in the librarian table of the database.
* The Administrator is the person who creates a new librarian profile by clicking to the “Add Faculty” button on the welcome Window.
* To add a book, the administrator has to add the following details about the person who is going to be assigned as the librarian.
* Faculty’s Name
* Faculty’s ID No.
* Faculty’s Password
* After filling all the details the administrator has to click on the “Register” button.
* After clicking on the button, the faculty gets added to the database.
* The faculty can now login through his credentials in the “User Application” of the Software.

Admin Module

* Add Result Window



**Add Result Window**

* The administrator can also add the result of the students to the database.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

Admin Module

* Add Book Window



**Add Notice Window**

* The administrator can also add the notice for the students to the database.
* The Administrator is the person who adds the notice for the students to the database by clicking to the “Add Notice” button on the welcome Window.
* To add the notice for the students, the administrator has to choose the file which contains the notice which has to be displayed on the student dashboard and get added to the database.
* The Administrator must ensure that the chosen file must be of correct format.
* The administrator also has to keep in mind that the size of the file should be smaller.
* After choosing the file, the administrator has to also provide the date on which this notice will be displayed.
* At last, the administrator has to click on the “Update” button in order to update the notice of a particular day.
* After clicking on the button, the notice gets added to the database and now it can be accessed by the students through User Application.

Admin Module

* Add Feed Window

****

**Add Feed Window**

* The administrator can also add the feed update for the student’s awareness.
* The Administrator is the person who adds the feeds for the students to the database by clicking to the “Add Feed” button on the welcome Window.
* While adding the feed for the students, the administrator can add the following details for the students.
* Upcoming Events
* Daily Hunt
* Innovation Feed
* After filling all the details the administrator has to click on the “Update” button.
* After clicking on the button, the feed gets updated and the student can now access it through the User Application.

1. User Module

* Welcome Window

****

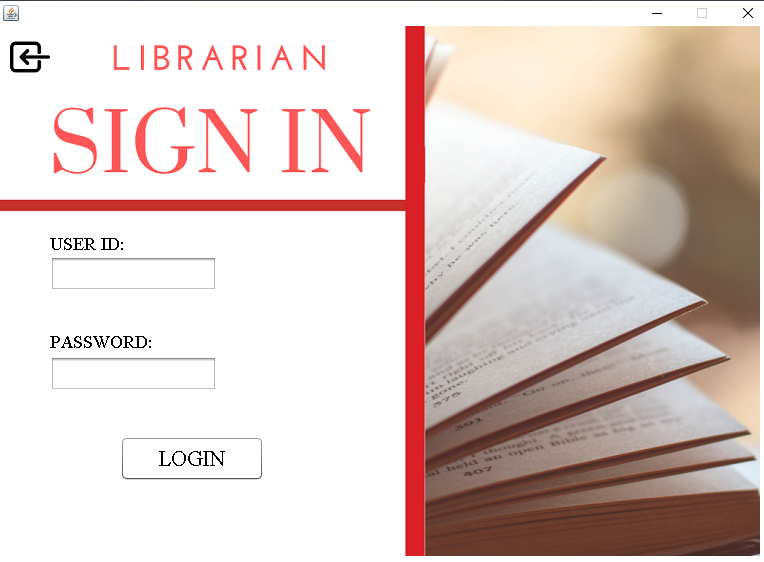
**Welcome Window**

* The Welcome Window in the User Application provides two different panels available for the users through which they can go to their dashboard.
* The two different panels includes:
* Faculty Panel
* Student Panel
* Both the users, either faculty or student can signin to their profile by their credentials.
* The faculty is the person who was added by the administrator as a faculty/librarian in the Administrator Application.
* The Student is the person who uses the application in order to get various details from the administrator about him for eg. Marks, Issued Books etc.

User Module

Faculty Panel

* Faculty Login Window

****

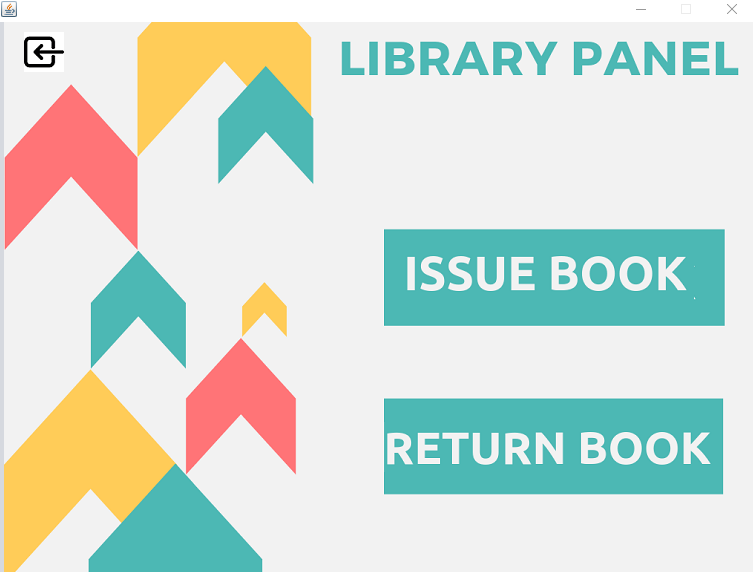
**Faculty Login Window**

* The faculty has to login through his credentials in order to issue a book to any of the students.
* To enter the faculty login Window, firstly the person has to click on the “Faculty” button on the welcome window.
* To login to the application, the faculty has to provide the following details about himself.
* Faculty’s ID
* Faculty’s Password
* After filling all the details the faculty has to click on the “Login” button.
* After clicking on the button, the system checks whether the entered details match with any entry in the faculty’s table in the database.
* If the entered details match with an entry in the database, the user gets logged in to the library panel.
* If the entered details don't match with any of the entries in the database, the user gets an error message.

User Module

Faculty Panel

* Library Panel Window

****

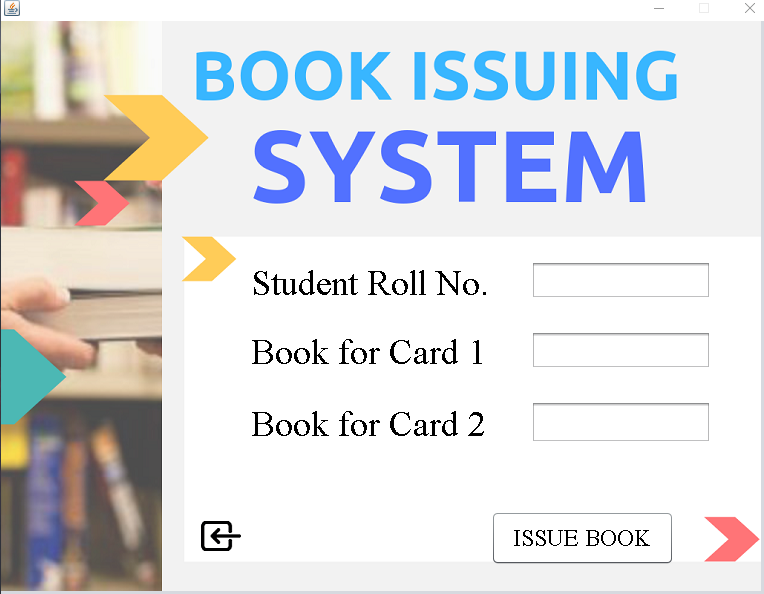
**Library Panel Window**

* The library panel Window provides the functions that can be performed by a faculty after logging in.
* The various functions that can be performed by a faculty/librarian includes:
* Issue Books
* Return Books
* If the Student asks to issue a book, the faculty can do so by going to the issue book section of the User Application.
* If the Student asks to return a previously issued book, the faculty can do so by going to the return book section of the User Application.

User Module

Faculty Panel

* Issue Book Window

****

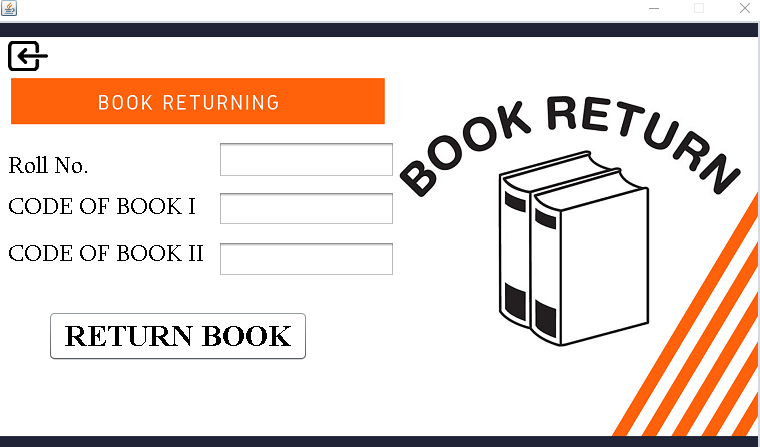
**Issue Book Window**

* The faculty can also issue books to the students through the Issue Book Window.
* To enter the Issue Book Window, firstly the person has to click on the “Issue Book” button on the library window.
* To issue a book to a student, the faculty has to provide the following details about the student and the book.
* Student’s Roll No.
* Book Code (for Card 1)
* Book Code (for Card 2)
* After filling all the details the faculty has to click on the “Issue Book” button.
* After clicking on the button, the system checks whether the entered details match with any entry in the book’s table in the database. It also checks whether another book is issued or not.
* If the entered details match with an entry in the database, the books get issued to the student with corresponding roll no.
* If the entered details don't match with any of the entries in the database, the user gets an error message.

User Module

Faculty Panel

* Return Book Window

****

**Return Book Window**

* The faculty has to accept returned books from the students through the Return Book Window.
* To enter the Return Book Window, firstly the person has to click on the “Return Book” button on the library window.
* To return a book from a student, the faculty has to provide the following details about the student and the book.
* Student’s Roll No.
* Book Code (for Card 1)
* Book Code (for Card 2)
* After filling all the details the faculty has to click on the “Return Book” button.
* After clicking on the button, the system checks whether the books corresponding to the entered details are issued to the students.
* If the entered details match with an entry in the database, the books get returned from the student with corresponding roll no.
* If the entered details don't match with any of the entries in the database, the user gets an error message and has to again enter the correct code of book or roll no. of the student returning the book.

User Module

Student Panel

* Profile Window

****

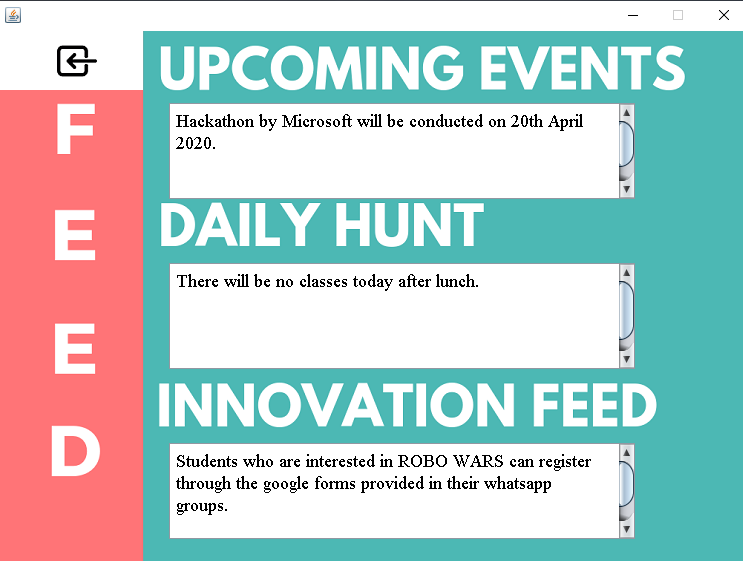
**Profile Window**

* The student can login to his profile through his credentials provided at the time of registration.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

User Module

Student Panel

* Feed Window

****

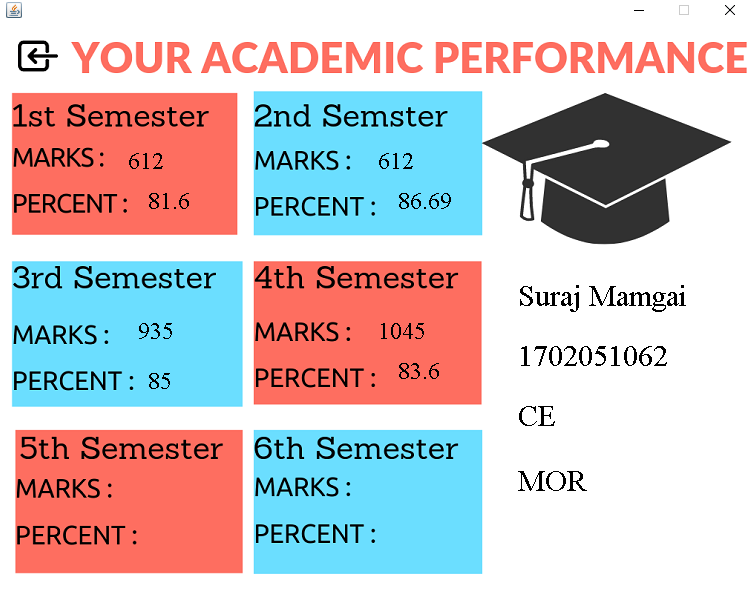
**Feed Window**

* The administrator can also add the result of the students to the database.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

User Module

Student Panel

* Academic Window

****

**Academics Window**

* The administrator can also add the result of the students to the database.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

User Module

Student Panel

* Notice Window

****

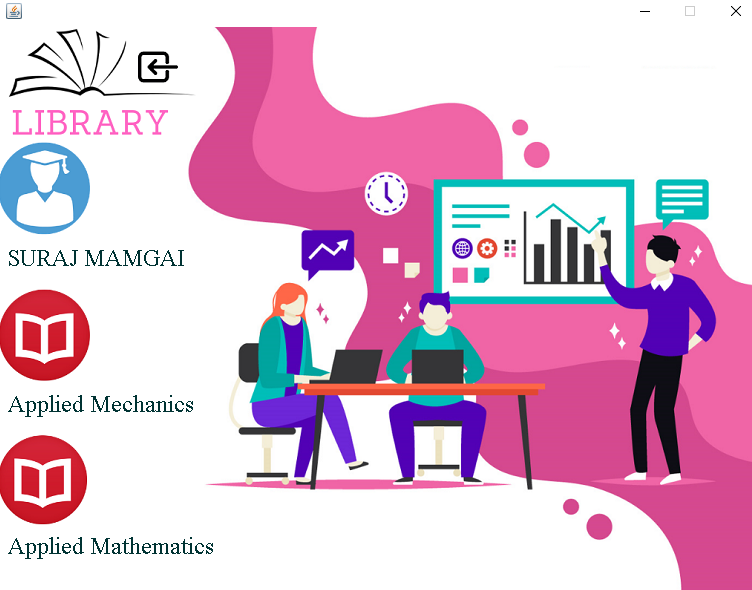
**Notice Window**

* The administrator can also add the result of the students to the database.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

User Module

Student Panel

* Library Window

****

**Library Window**

* The administrator can also add the result of the students to the database.
* The Administrator is the person who adds the result of the students to the database by clicking to the “Add Result” button on the welcome Window.
* To add the result of a student, the administrator has to add the following details about the student whose result has to be added to the database.
* Student’s Roll No.
* Student’s Marks
* Student’s Percentage
* Semester
* After filling all the details the administrator has to click on the “Submit” button.
* After clicking on the button, the result of the student gets added to the database.
* The student can now see his marked performance in the “User Application” of the Software.

**Future Scope**

Such a Desktop based platform has immense potential for future growth not only for students but also for faculty as well as departments concerned in improving upon their efficiency of work with no or less burden. Some of the areas where future scope lies and if implemented would help a great deal to the students at large.

* In the Future, Students can also download and upload notes.
* Also, it would be possible for students to watch lectures in the form of video content which would facilitate easy access to study material.
* Further enhancement would help in Statistics tracking and analytics of student’s performance according to his/her result.

**CONCLUSION**

* The system provides excellent support to the colleges because they can directly manage their student databases.
* Computerized surveying systems provide easy, fast access and support for the user.
* The usage of software increases efficiency, decreases effort.
* It has been thoroughly tested and implemented.