LIBRARY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

C. MATHEW KURUSE (953421104025)

A. N. MOHAMED RIYAZ (953421104026)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

V V COLLEGE OF ENGINEERING, TISAIYANVILAI

ANNA UNIVERSITY: CHENNAI 600 025

MARCH 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"LIBRARY MANAGEMENT SYSTEM"** is a bonafide work of **"C. MATHEW KURUSE (953421104025), A.N. MOAHMED RIYAZ (953421104026) "** who carried out final year project work under my supervision.

SIGNATURE

Dr.Muthulakshmi,M.E.,ph.D.,

HEAD OF DEPARTMENT

Computer Science and Engineering,
V V College of Engineering,
Tisayainvilai-627 657.

SIGNATURE

Ms. M.Jancyrani Malli,M.E.,

SUPERVISOR

Computer Science and Engineering,
V V College of Engineering,
Tisaiyanvilai-627 657.

Submitted for the viva-voce held on ……..

INTERNAL EXAMINER

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

First and foremost, we express our gratitude to God Almighty who has showered blessings on us by giving us the strength and determination to complete this project successfully.

Our sincere thanks to the **Management** for giving us good firmware facilities to complete the project. We thank our department faculty members for their support to complete this project.

We sincerely thank our **Principal Dr.K.S.Saji M.E,Ph.D.,** for her valuable guidance and support.

We would like to express our heartfelt gratitude to our Head of the Department **Dr.Muthulakshmi,M.E.,ph.D.,** for being a constant source of inspiration to us.

We are indebted to our project co-ordinator **Ms. M.Jancyrani Malli,M.E.,** for providing moral support and motivation.

Our special thanks to our families and friends who have stood by us during the course of our project.

ABSTRACT

The library management system project aims to streamline the process of managing books and student records within a library setting. By providing a user-friendly graphical interface, users, including librarians and administrators, can efficiently perform various tasks such as issuing books to students and handling book returns. The system allows for easy searching and retrieval of book and student information using unique identifiers like book IDs and student IDs. Through integration with a database using JDBC, the system ensures data integrity and persistence, enabling seamless storage and retrieval of information. Key features include the ability to input, update, and delete book and student records, as well as issuing and returning books with automated date tracking. With its intuitive design and robust functionality, the library management system project aims to enhance the overall efficiency and effectiveness of library operations, ultimately benefiting both library staff and patrons. Additionally, the library management system project incorporates features to manage overdue books and fines, providing automated notifications to remind users of impending due dates and overdue items. This helps promote timely returns and ensures accountability among library patrons.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The library management system (LMS) project represents a comprehensive solution aimed at optimizing the management of books and user records within a library environment. This section serves as an introduction to the project, providing an overview of its aims, objectives, background, and the operational environment in which it functions.

1.1 PROJECT AIMS AND OBJECTIVES

The primary goal of the LMS project is to streamline and modernize the processes involved in managing library resources and user information. Key objectives include:

Streamlining book management processes: The LMS aims to simplify tasks such as book cataloging, tracking, and inventory management, ultimately improving the overall efficiency of library operations.

Enhancing user experience: By providing a user-friendly interface and intuitive features, the LMS seeks to enhance the experience of both library staff and patrons, making it easier to search for and access library resources.

Improving operational efficiency: Through automation and integration with databases and other systems, the LMS aims to reduce manual workload, minimize errors, and optimize resource allocation within the library.

Facilitating data-driven decision-making: By centralizing and organizing library data, the LMS enables librarians and administrators to analyze trends, track usage patterns, and make informed decisions regarding collection development and resource allocation.

1.2 BACKGROUND OF PROJECT

The development of the LMS project was prompted by several factors, including:

Existing challenges in manual book management systems: Traditional methods of cataloging and managing library resources are often time-consuming and prone to errors. The need for a more efficient and accurate solution led to the exploration of automated library management systems.

Technological advancements: The rapid advancements in information technology, particularly in areas such as database management, user interface design, and networking, presented opportunities to develop sophisticated library management solutions that could address the evolving needs of modern libraries.

Modernization in library operations: As libraries increasingly embrace digital resources and online services, there is a growing need for robust, flexible, and scalable management systems that can accommodate these changes while maintaining the core functions of traditional library services.

1.3 OPERATION ENVIRONMENT

The operational environment of the LMS encompasses various factors that influence its functionality and usability, including:

Physical setting of the library: This includes considerations such as the size, layout, and infrastructure of the library facility, as well as the availability of resources such as computers, scanners, and printers.

Target user base: The LMS caters to a diverse user base, including librarians, library staff, students, faculty, and other library patrons. As such, the system must be designed to accommodate the needs and preferences of different user groups.

Technological infrastructure: The LMS relies on a robust technological infrastructure, including hardware such as servers, computers, and networking equipment, as well as software components such as databases, operating systems, and application servers.

External factors: External factors such as regulatory requirements, institutional policies, and industry standards may impact the design, implementation, and operation of the LMS. Additionally, considerations such as data security, privacy, and accessibility must be taken into account to ensure compliance with relevant regulations and best practices.

CHAPTER 2

SYSTEM ANALYSIS

The system analysis phase constitutes a comprehensive investigation into the intricacies of the library management system (LMS), encompassing its requirements, functionalities, and limitations. This section scrutinizes diverse facets of the system, including the delineation of software requirements, a comparative analysis between the prevailing system and the proposed LMS, and an elucidation of the software tools harnessed throughout the development trajectory.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

The software requirement specification (SRS) serves as the cornerstone of the LMS, delineating both its functional and non-functional prerequisites, thereby furnishing a blueprint for the system's conception and maturation. The cardinal constituents of the SRS entail:

Functional Requirements: These delineate the gamut of the system's capabilities and functionalities, encompassing pivotal aspects such as user

authentication, book cataloging, search and retrieval mechanisms, user management functionalities, reporting functionalities, and administrative tasks, ensuring a seamless orchestration of library operations.

Non-functional Requirements: These edify the quality attributes underpinning the system, embracing elements such as performance, reliability, security, usability, scalability, and maintainability. For instance, the system is mandated to adeptly navigate a surfeit of concurrent users sans any performance compromises.

User Requirements: These epitomize the exigencies and anticipations of disparate user cohorts, spanning librarians, administrators, and patrons alike. User requisites may span a spectrum of features including an ergonomic user interface, personalized user profiles, and provisions for accessibility catering to users with disabilities.

2.2 EXISTING VS PROPOSED

This segment juxtaposes the attributes and functionalities of the extant library management system (if extant) vis-à-vis those posited in the proposed LMS. It accentuates the lacunae inherent in the incumbent system whilst expounding upon how the proposed LMS ameliorates these shortcomings whilst furnishing supplementary benefits. Salient areas of comparison may encompass:

Ease of Use: The proposed LMS may proffer an interface imbued with intuitive design paradigms and streamlined workflows vis-à-vis the extant system, thereby facilitating facile execution of tasks such as book retrieval, checkout, and return.

Functionality: The LMS may proffer a surfeit of novel features and functionalities including but not limited to advanced search capabilities, seamless integration with external databases, support for digital resources, and automated notifications pertaining to overdue books.

Performance: The proposed LMS may evince a marked enhancement in performance metrics such as response time, system uptime, and scalability vis-à-vis the incumbent system, enabling it to efficaciously accommodate larger data volumes and user cohorts.

Security: The LMS may embed robust security measures encompassing facets such as role-based access control, data encryption, and audit trails, fortifying the bulwarks safeguarding sensitive information and mitigating the specter of security breaches.

2.3 SOFTWARE TOOL USED

The gestation of the LMS hinges upon an assortment of software tools and technologies culled to engender a fecund milieu for the design, implementation, testing, and deployment phases. Commonly enlisted software tools encompass:

Integrated Development Environments (IDEs): Robust IDEs such as Eclipse, NetBeans, and IntelliJ IDEA furnish an expansive pantheon of functionalities for code composition, debugging, and testing, encompassing features such as auto-completion, syntax highlighting, and seamless integration with version control systems.

Database Management Systems (DBMS): Pervasive DBMS platforms including MySQL, PostgreSQL, and Oracle serve as the linchpin for the storage and management of the library's data milieu, spanning book records, user information, transaction logs, and system configurations.

Version Control Systems (VCS): VCS platforms such as Git and Subversion usher in a paradigm of collaborative development, catalyzing concurrent contributions from disparate developers whilst undergirding functionalities such as versioning, branching, merging, and code review processes.

Issue Tracking Systems (ITS): ITS frameworks including Jira, Bugzilla, and Redmine constitute veritable lynchpins in the orchestration of software development endeavors, efficaciously tracking bugs, feature requests, and tasks, thereby expediting communication and coordination amongst stakeholders.

Documentation Tools: An array of documentation tools spanning the likes of Doxygen, Javadoc, and Sphinx coalesce to engender a corpus of

documentation encompassing API documentation, user manuals, and technical specifications, thereby fostering code readability, maintainability, and knowledge propagation across the development echelons.

The development of the library management system (LMS) heavily relies on Java, Java Swing for the frontend, and SQL for the backend, along with additional tools to support the development process.

Java: Java serves as the primary programming language for developing the library management system. Java is chosen for its platform independence, robustness, and extensive standard libraries. It allows for the creation of modular, scalable, and maintainable code, making it well-suited for large-scale software projects like the LMS. Additionally, Java's object-oriented nature facilitates code reusability and modularity, contributing to the system's overall flexibility and extensibility.

Java Swing: Java Swing is a set of graphical user interface (GUI) components and libraries for building desktop applications in Java. In the context of the LMS, Java Swing is utilized to create the frontend interface, including windows, dialogs, buttons, text fields, and other GUI elements. Java Swing provides a rich set of tools and layouts for designing intuitive and interactive user interfaces, enabling users to navigate, interact with, and manage library resources seamlessly. With Java Swing, developers can create platform-independent GUI applications that run smoothly on various operating systems without compromising performance or functionality.

SQL (Structured Query Language): SQL is used as the backend database language for storing, retrieving, and manipulating data in the library management system. SQL databases such as MySQL, PostgreSQL, or SQLite are commonly employed to manage the system's relational database schema, which includes tables for storing information about books, students, transactions, and other relevant entities. SQL enables efficient data querying, indexing, and transaction management, ensuring the integrity, consistency, and security of the library's data. Through SQL, developers can implement complex database operations such as CRUD (Create, Read, Update, Delete) operations, data aggregation, and joins, providing a solid foundation for the LMS's data management capabilities.

In summary, Java, Java Swing, and SQL play integral roles in the development and implementation of the library management system, providing the necessary tools and frameworks for building a robust, user-friendly, and feature-rich application that meets the needs of librarians, administrators, and library patrons alike.

CHAPTER 3
SYSTEM DESIGN

3.1 TABLE DESIGN

The table design for the library management system (LMS) involves creating structured database tables to store information about accounts, books, students, issued books, and returned books. Each table is designed to efficiently organize and manage specific types of data, ensuring data integrity and facilitating seamless retrieval and manipulation of information. Below is the table design for the LMS:

1. Account Table:

- **`id`:** Primary key auto-incremented integer representing the account ID.
- **`username`:** VARCHAR(50) field storing the username of the account holder.
- **`name`:** VARCHAR(50) field storing the name of the account holder.
- **`password`:** VARCHAR(100) field storing the encrypted password of the account.

- **`sec_q`:** VARCHAR(100) field storing the security question for the account.

- **`sec_ans`:** VARCHAR(100) field storing the encrypted security answer for the account.

2. Book Table:

- **`book_id`:** Primary key auto-incremented integer representing the book ID.

- **`name`:** VARCHAR(100) field storing the name of the book.

- **`isbn`:** VARCHAR(50) field storing the ISBN (International Standard Book Number) of the book.

- **`publisher`:** VARCHAR(100) field storing the publisher of the book.

- **`edition`:** VARCHAR(50) field storing the edition of the book (if applicable).

- **`price`:** DECIMAL(10, 2) field storing the price of the book.

- **`pages`:** INT field storing the number of pages in the book.

3. Student Table:

- **`student_id`:** Primary key auto-incremented integer representing the student ID.

- **`name`:** VARCHAR(100) field storing the name of the student.

- **`father`:** VARCHAR(100) field storing the name of the student's father.

- **`course`:** VARCHAR(50) field storing the course enrolled by the student.

- **`branch`:** VARCHAR(50) field storing the branch of study of the student.

- **`year`:** VARCHAR(20) field storing the academic year of the student.

- **`semester`:** VARCHAR(20) field storing the semester of the student.

4. IssueBook Table:

- **`issue_id`:** Primary key auto-incremented integer representing the issue ID.

- **`book_id`:** Foreign key referencing the `book_id` in the Book table.

- **`student_id`:** Foreign key referencing the `student_id` in the Student table.

- **`bname`:** VARCHAR(255) field storing the name of the issued book.

- **`sname`:** VARCHAR(255) field storing the name of the student to whom the book is issued.

- **`course`:** VARCHAR(50) field storing the course of the student.

- **`branch`:** VARCHAR(50) field storing the branch of the student.

- **`dateOfIssue`:** DATE field storing the date when the book is issued.

5. ReturnBook Table:

- **`book_id`:** VARCHAR(20) field storing the ID of the returned book.
- **`student_id`:** VARCHAR(20) field storing the ID of the student returning the book.
- **`bname`:** VARCHAR(100) field storing the name of the returned book.
- **`sname`:** VARCHAR(100) field storing the name of the student returning the book.
- **`course`:** VARCHAR(50) field storing the course of the student.
- **`branch`:** VARCHAR(50) field storing the branch of the student.
- **`dateOfIssue`:** DATE field storing the date when the book was issued.
- **`dateOfReturn`:** DATE field storing the date when the book was returned.

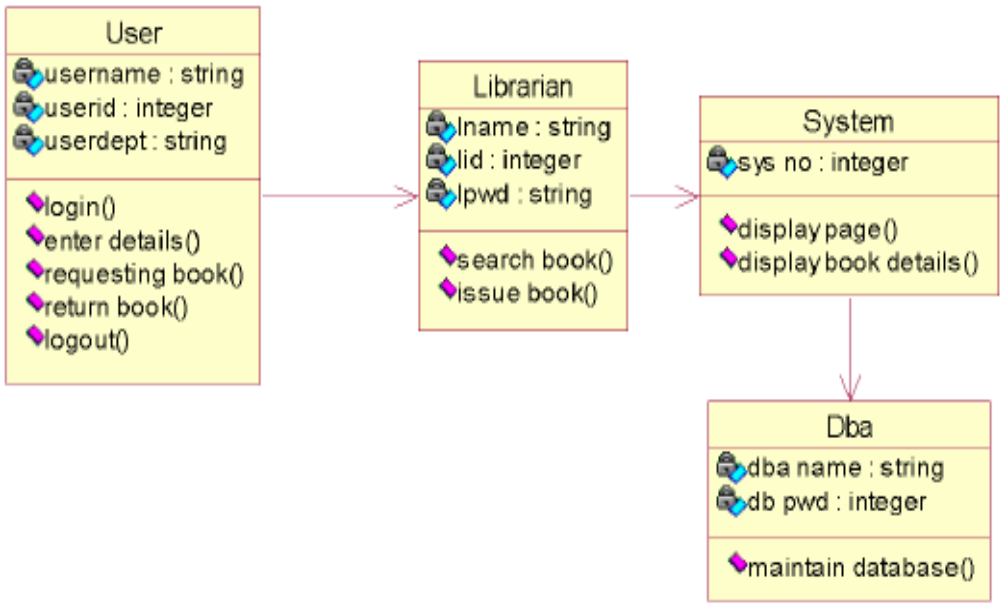
These tables form the backbone of the LMS database, providing a structured framework for storing and managing crucial information related to accounts, books, students, and transactions within the library system.

3.2 UML Diagrams

In the system design phase of the library management system (LMS) project, Unified Modeling Language (UML) diagrams are instrumental in visualizing the architecture, structure, and interactions within the system. These diagrams serve as blueprints for developers and stakeholders, providing a clear understanding of the system's design and functionality. The following UML diagrams are created for the LMS:

1. Class Diagram:

- The class diagram illustrates the static structure of the system by representing classes, attributes, methods, and their relationships. It depicts the entities involved in the LMS, such as books, students, accounts, and transactions, along with their attributes and associations.

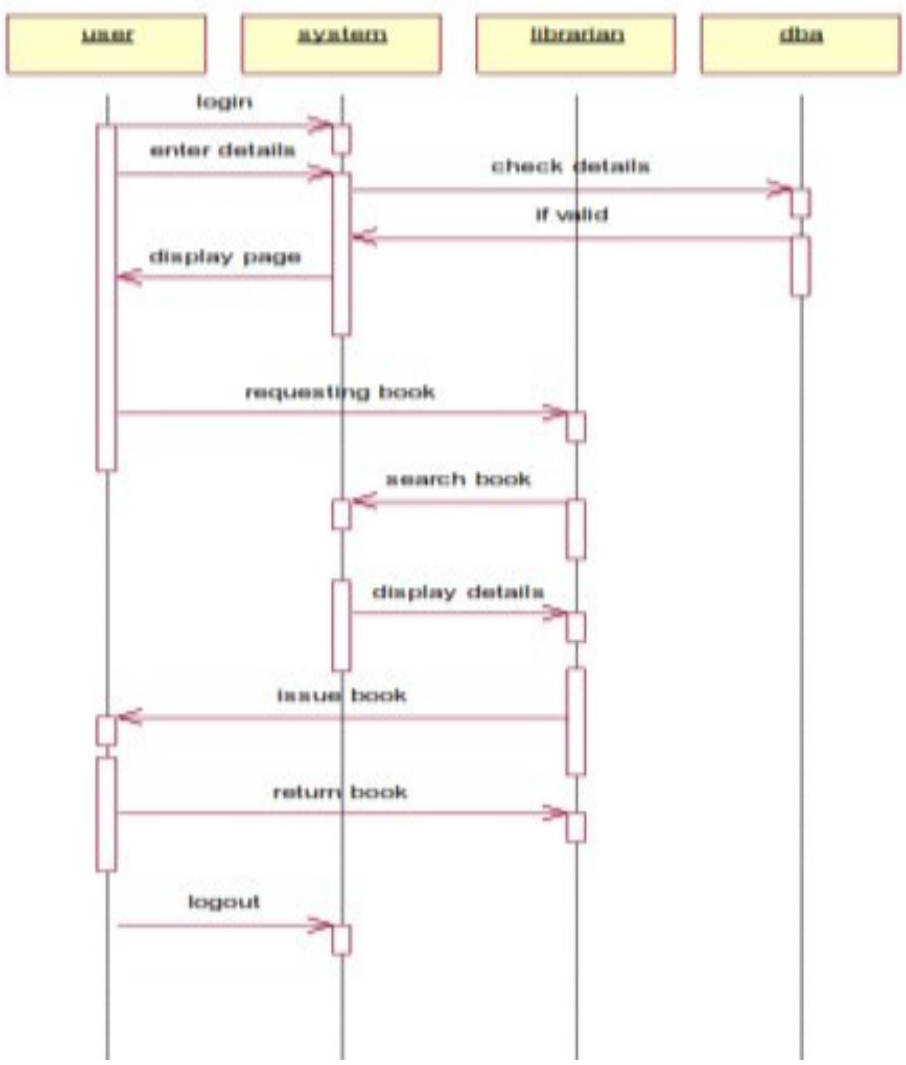


2. Use Case Diagram:

- The use case diagram identifies the system's functionalities from the user's perspective. It showcases various actors, such as librarians, administrators, and students, interacting with the system to perform tasks like book search, checkout, return, user management, and reporting. Each use case represents a specific functionality of the system.

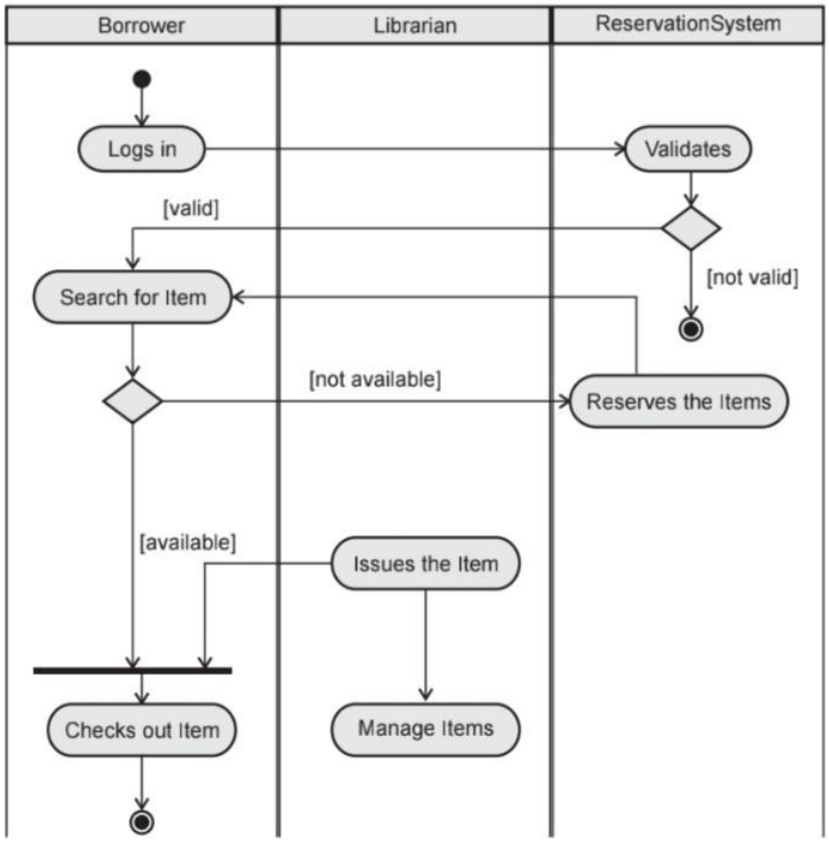
3. Sequence Diagram:

- The sequence diagram visualizes the interactions between objects or components in a sequential manner, depicting the flow of messages exchanged during runtime. It illustrates how different system components collaborate to fulfill a specific use case scenario, such as issuing a book to a student or returning a book to the library.



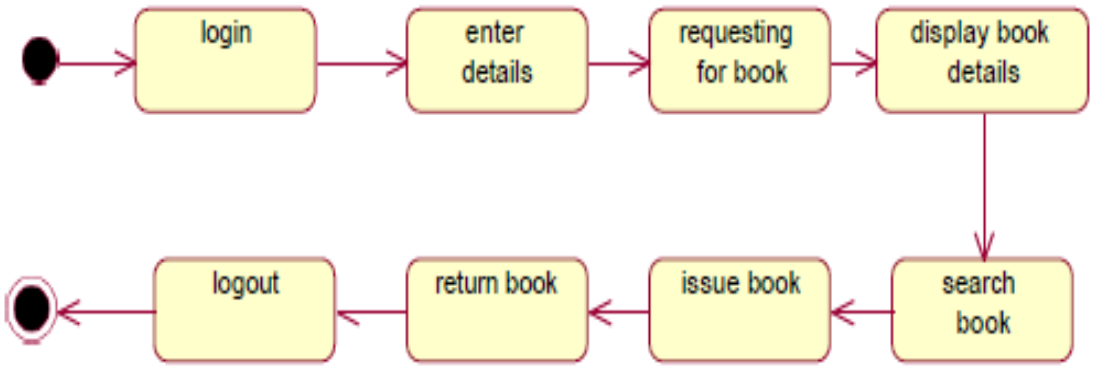
4. Activity Diagram:

- The activity diagram represents the workflow or business processes within the system, showcasing the sequence of activities, decision points, and transitions. It helps in understanding the control flow and logic of various operations performed by the system, such as user authentication, book cataloging, and transaction processing.



5. State Diagram:

- The state diagram models the lifecycle of objects or entities within the system, depicting their different states and transitions triggered by events. It illustrates the behavior of entities like books, students, and transactions as they progress through various states, such as available, issued, overdue, and returned.



These UML diagrams collectively provide a comprehensive overview of the LMS design, enabling stakeholders to visualize system components, functionalities, interactions, and behaviors in a structured manner. They serve as invaluable tools for communication, analysis, and validation throughout the development lifecycle of the LMS project.

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

For Library Management System it is divided into the following

4.1.1 Code for Homepage:

```
import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;

public class Home extends JFrame implements ActionListener{

    private JPanel contentPane;
   private JButton b1,b2,b3,b4,b5,b6;

    public static void main(String[] args) {
     new Home().setVisible(true);
    }
```

```
    public Home() {
        super("Library Management System");
        setBounds(400, 150, 1000, 800);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);


        JMenuBar menuBar = new JMenuBar();
        menuBar.add(Box.createRigidArea(new Dimension(400,100)));
          menuBar.setBorder(new EtchedBorder(EtchedBorder.LOWERED,
new Color(0, 128, 0), new Color(128, 128, 128)));
        menuBar.setBackground(Color.white);
        menuBar.setBounds(0, 10, 1000, 40);
        contentPane.add(menuBar);

        JMenu mnExit = new JMenu("Exit");
        mnExit.setFont(new Font("Trebuchet MS", Font.BOLD, 17));


        JMenuItem mntmLogout = new JMenuItem("Logout");
        mntmLogout.setBackground(Color.white);
        mntmLogout.setForeground(Color.red);
        mntmLogout.addActionListener(this);
        mnExit.add(mntmLogout);
```

```
        JMenuItem mntmExit = new JMenuItem("Exit");
        mntmExit.setForeground(Color.red);
        mntmExit.setBackground(Color.white);
        mntmExit.addActionListener(this);
        mnExit.add(mntmExit);



//          JMenu mnHelp = new JMenu("Help");
//          mnHelp.setFont(new Font("Trebuchet MS", Font.BOLD, 17));
//
//
//          JMenuItem mntmReadme = new JMenuItem("Read Me");
//          mntmReadme.setBackground(Color.white);
//          mntmReadme.setForeground(Color.red);
//          mnHelp.add(mntmReadme);
//
//          JMenuItem mntmAboutUs = new JMenuItem("About Us");
//          mntmAboutUs.setForeground(Color.red);
//          mntmAboutUs.setBackground(Color.white);
//          mntmAboutUs.addActionListener(this);
//          mnHelp.add(mntmAboutUs);

        JMenu mnRecord = new JMenu("Record");
        mnRecord.setFont(new Font("Trebuchet MS", Font.BOLD, 17));
```

```
        JMenuItem bookdetails = new JMenuItem("Book Details");
        bookdetails.addActionListener(this);
        bookdetails.setBackground(Color.white);
        bookdetails.setForeground(Color.red);
        mnRecord.add(bookdetails);

        JMenuItem studentdetails = new JMenuItem("Student Details");
        studentdetails.setBackground(Color.white);
        studentdetails.setForeground(Color.red);
        studentdetails.addActionListener(this);
        mnRecord.add(studentdetails);

        menuBar.add(mnRecord);
//          menuBar.add(mnHelp);
        menuBar.add(mnExit);


        JLabel l1 = new JLabel("Library Management System");
        l1.setForeground(new Color(204, 51, 102));
        l1.setFont(new Font("Segoe UI Semilight", Font.BOLD, 30));
        l1.setBounds(268, 30, 701, 80);
        contentPane.add(l1);

        JLabel l2 = new JLabel("");
```

```
        l2.setVerticalAlignment(SwingConstants.TOP);
                                                ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icons/addbooks.png"));
                        Image i2 = i1.getImage().getScaledInstance(150,
150,Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        l2 = new JLabel(i3);
        l2.setBounds(60, 140, 159, 152);
        contentPane.add(l2);

        JLabel l3 = new JLabel("");
                                                ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("icons/stats.png"));
                        Image i5 = i4.getImage().getScaledInstance(200,
200,Image.SCALE_DEFAULT);
        ImageIcon i6 = new ImageIcon(i5);
        l3 = new JLabel(i6);
        l3.setBounds(300, 160, 134, 128);
        contentPane.add(l3);

        JLabel l4 = new JLabel("");
                                                ImageIcon i7 = new
ImageIcon(ClassLoader.getSystemResource("icons/addstudent.png"));
                        Image i8 = i7.getImage().getScaledInstance(150,
150,Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
```

```
l4 = new JLabel(i9);
l4.setBounds(530, 140, 225, 152);
contentPane.add(l4);

b1 = new JButton("Add Books");
b1.addActionListener(this);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
b1.setBounds(60, 320, 159, 44);
contentPane.add(b1);

b2 = new JButton("Statistics");
b2.addActionListener(this);
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
b2.setBounds(313, 320, 139, 44);
contentPane.add(b2);

b3 = new JButton("Add Student");
b3.addActionListener(this);
b3.setBackground(Color.BLACK);
b3.setForeground(Color.WHITE);
b3.setBounds(562, 320, 167, 44);
contentPane.add(b3);

JPanel panel = new JPanel();
```

```
            panel.setBorder(new TitledBorder(new LineBorder(new Color(250,
128, 114), 2), "Operation", TitledBorder.LEADING,
                                    TitledBorder.TOP, null, new Color(220, 20, 60)));
            panel.setBounds(20, 120, 750, 260);
            panel.setBackground(Color.WHITE);
            contentPane.add(panel);

            b4 = new JButton("Issue Book");
            b4.addActionListener(this);
            b4.setBackground(Color.BLACK);
            b4.setForeground(Color.WHITE);
            b4.setBounds(76, 620, 143, 41);
            contentPane.add(b4);

            b5 = new JButton("Return Book");
            b5.addActionListener(this);
            b5.setBackground(Color.BLACK);
            b5.setForeground(Color.WHITE);
            b5.setBounds(310, 620, 159, 41);
            contentPane.add(b5);

//          b6 = new JButton("About Us");
//          b6.addActionListener(this);
//          b6.setBackground(Color.BLACK);
//          b6.setForeground(Color.WHITE);
//          b6.setBounds(562, 620, 159, 41);
```

```
//          contentPane.add(b6);

        JLabel l5 = new JLabel("");
                                                ImageIcon i10 = new
ImageIcon(ClassLoader.getSystemResource("icons/issue.png"));
                        Image i11 = i10.getImage().getScaledInstance(150,
150,Image.SCALE_DEFAULT);
        ImageIcon i12 = new ImageIcon(i11);
        l5 = new JLabel(i12);
        l5.setBounds(60, 440, 159, 163);
        contentPane.add(l5);

        JLabel l6 = new JLabel("");
                                                ImageIcon i13 = new
ImageIcon(ClassLoader.getSystemResource("icons/return.png"));
                        Image i14 = i13.getImage().getScaledInstance(150,
150,Image.SCALE_DEFAULT);
        ImageIcon i15 = new ImageIcon(i14);
        l6 = new JLabel(i15);
        l6.setBounds(332, 440, 139, 152);
        contentPane.add(l6);

        JLabel l7 = new JLabel("");
                                                ImageIcon i16 = new
ImageIcon(ClassLoader.getSystemResource("icons/logo_r.png"));
```

```
                    Image i17 = i16.getImage().getScaledInstance(150,
150,Image.SCALE_DEFAULT);
        ImageIcon i18 = new ImageIcon(i17);
        l7 = new JLabel(i18);
        l7.setBounds(562, 465, 157, 152);
        contentPane.add(l7);

        JPanel panel2 = new JPanel();
        panel2.setBorder(new TitledBorder(new LineBorder(new Color(205,
133, 63), 2), "Action", TitledBorder.LEADING,
                        TitledBorder.TOP, null, new Color(233, 150,
122)));
        panel2.setBounds(20, 420, 750, 270);
        panel2.setBackground(Color.WHITE);
        contentPane.add(panel2);

        getContentPane().setBackground(Color.WHITE);
        contentPane.setBackground(Color.WHITE);
       }


    public void actionPerformed(ActionEvent ae){
        String msg = ae.getActionCommand();
        if(msg.equals("Logout")){
           setVisible(false);
              new Login_user().setVisible(true);
```

```
}else if(msg.equals("Exit")){
   System.exit(ABORT);

}else if(msg.equals("Read Me")){

}else if(msg.equals("About Us")){
   setVisible(false);
       new aboutUs().setVisible(true);

}else if(msg.equals("Book Details")){
   setVisible(false);
       new BookDetails().setVisible(true);
}else if(msg.equals("Student Details")){
   setVisible(false);
   new StudentDetails().setVisible(true);

}

if(ae.getSource() == b1){
   this.setVisible(false);
   new AddBook().setVisible(true);
}
if(ae.getSource() == b2){
   this.setVisible(false);
   new Statistics().setVisible(true);
}
```

```
        if(ae.getSource() == b3){
            this.setVisible(false);
            new AddStudent().setVisible(true);
        }
        if(ae.getSource() == b4){
            this.setVisible(false);
            new IssueBook().setVisible(true);
        }
        if(ae.getSource() == b5){
            this.setVisible(false);
            new ReturnBook().setVisible(true);

        }
        if(ae.getSource() == b6){
            this.setVisible(false);
            new aboutUs().setVisible(true);

        }

    }
}
```

4.1.2 Code For SQL Connection :

```
import java.sql.*;

public class conn{
    Connection c;
    Statement s;
    public conn(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
                                                                    c
=DriverManager.getConnection("jdbc:mysql:///project2","root","1234");
            s =c.createStatement();



        }catch(Exception e){
            System.out.println(e);
        }
    }
    public static void main(String args[])
    {
        new conn();
```

```
        System.out.println("Database Connected");
    }
}
```

4.1.3 Code For Statistics:

```
import java.awt.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.DbUtils;
import java.awt.event.*;

public class Statistics extends JFrame{

    private JPanel contentPane;
    private JTable table;
    private JTable table_1;

    public static void main(String[] args) {
        new Statistics().setVisible(true);
    }

    public void issueBook() {
```

```
      try {
       conn con =  new conn();
       String sql = "select * from issueBook";
       PreparedStatement st = con.c.prepareStatement(sql);
       ResultSet rs = st.executeQuery();

       table.setModel(DbUtils.resultSetToTableModel(rs));

      } catch (Exception e) {
                  // TODO: handle exception
      }
}

public void returnBook() {
   try {
       conn con = new conn();
       String sql = "select * from returnBook";
       PreparedStatement st = con.c.prepareStatement(sql);
       ResultSet rs = st.executeQuery();
       table_1.setModel(DbUtils.resultSetToTableModel(rs));
      } catch (Exception e) {
                  // TODO: handle exception
      }
}

public Statistics() {
```

```
        setBounds(400, 200, 810, 594);
          contentPane = new JPanel();
        contentPane.setBackground(Color.WHITE);
          contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
          setContentPane(contentPane);
          contentPane.setLayout(null);

          JScrollPane scrollPane = new JScrollPane();
          scrollPane.setBounds(40, 51, 708, 217);
          contentPane.add(scrollPane);

        table = new JTable();
          table.setBackground(new Color(224, 255, 255));
          table.setForeground(new Color(128, 128, 0));
          table.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
          scrollPane.setViewportView(table);

          JPanel panel = new JPanel();
          panel.setBorder(new TitledBorder(new LineBorder(new Color(47, 79,
79), 2, true), "Issue-Book-Details",
                                  TitledBorder.LEADING,  TitledBorder.TOP,  null,
new Color(0, 128, 128)));
          panel.setForeground(new Color(0, 128, 128));
          panel.setBounds(26, 36, 737, 240);
        panel.setBackground(Color.WHITE);
          contentPane.add(panel);
```

```
        JLabel l1 = new JLabel("Back");
        l1.addMouseListener(new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent arg0) {
                setVisible(false);
                Home home = new Home();
                home.setVisible(true);
         }
        });
        l1.setForeground(new Color(204, 0, 102));
        l1.setFont(new Font("Tahoma", Font.BOLD, 18));
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/tenth.png"));
                Image i2 = i1.getImage().getScaledInstance(20, 20, Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      l1.setIcon(i3);
        l1.setBounds(690, 13, 96, 27);
        contentPane.add(l1);

        JScrollPane scrollPane_1 = new JScrollPane();
        scrollPane_1.setBounds(40, 316, 717, 217);
        contentPane.add(scrollPane_1);

        table_1 = new JTable();
```

```
        table_1.setBackground(new Color(204, 255, 255));
        table_1.setForeground(new Color(153, 51, 0));
        table_1.setFont(new Font("Sitka Small", Font.BOLD, 12));
        scrollPane_1.setViewportView(table_1);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new TitledBorder(new LineBorder(new Color(0,
204, 153), 2, true), "Return-Book-Details",
                TitledBorder.RIGHT, TitledBorder.TOP, null, new Color(0,
102, 51)));
        panel_1.setBounds(22, 299, 741, 240);
       panel_1.setBackground(Color.WHITE);
        contentPane.add(panel_1);



        issueBook();
        returnBook();
    }
}
```

4.1.4 Code For Login_user:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
```

```
import java.sql.*;

public class Login_user extends JFrame implements ActionListener{

        private JPanel panel;
        private JTextField textField;
        private JPasswordField passwordField;
      private JButton b1,b2,b3;



        public Login_user() {
         super("Login/SignUp");

      //Color class --> Swing
      //background --> Jframe
        setBackground(Color.ORANGE);
      setBounds(600, 300, 600, 400);

      panel = new JPanel();
        panel.setBackground(Color.gray);
        setContentPane(panel);
        panel.setLayout(null);

//      JLabel l = new JLabel(new
ImageIcon(ClassLoader.getSystemResource("icons/logo.png")));
//      l.setBounds(300, 10, 400, 300);
```

```
//        panel.add(l);



          JLabel l1 = new JLabel("Username --> ");
          l1.setBounds(124, 89, 95, 24);
        l1.setForeground(Color.black);
          panel.add(l1);

          JLabel l2 = new JLabel("Password --> ");
          l2.setBounds(124, 124, 95, 24);
        l2.setForeground(Color.black);
          panel.add(l2);

          textField = new JTextField();
          textField.setBounds(210, 93, 157, 20);
          panel.add(textField);

          passwordField = new JPasswordField();
          passwordField.setBounds(210, 128, 157, 20);
          panel.add(passwordField);

          JLabel l3 = new JLabel("");
```

```
l3.setBounds(377, 79, 46, 34);
panel.add(l3);

JLabel l4 = new JLabel("");
l4.setBounds(377, 124, 46, 34);
panel.add(l3);

b1 = new JButton("Login");
b1.addActionListener(this);

b1.setForeground(Color.black);
b1.setBackground(Color.green);
b1.setBounds(149, 181, 113, 39);
panel.add(b1);

b2 = new JButton("SignUp");
b2.addActionListener(this);

b2.setForeground(Color.black);
b2.setBackground(Color.CYAN);
b2.setBounds(289, 181, 113, 39);
panel.add(b2);

b3 = new JButton("Forgot Password");
b3.addActionListener(this);
```

```
        b3.setForeground(Color.black);
          b3.setBackground(Color.red);
          b3.setBounds(199, 231, 179, 39);
          panel.add(b3);



//                JPanel panel2 = new JPanel();
//                panel2.setBackground(Color.YELLOW);
//                panel2.setBounds(24, 40, 434, 263);
//                panel.add(panel2);
        }

        public void actionPerformed(ActionEvent ae){
           if(ae.getSource() == b1){
              Boolean status = false;
                 try {
                 conn con = new conn();
                 String sql = "select * from account where username=? and
password=?";
                 PreparedStatement st = con.c.prepareStatement(sql);

                 st.setString(1, textField.getText());
                 st.setString(2, passwordField.getText());

                 ResultSet rs = st.executeQuery();
```

```
                if (rs.next()) {
                    this.setVisible(false);
                    new Home().setVisible(true);
                } else
                        JOptionPane.showMessageDialog(null, "Invalid
Login...!.");

                } catch (Exception e2) {
                e2.printStackTrace();
                }
            }
            if(ae.getSource() == b2){
                setVisible(false);
                    Signup su = new Signup();
                    su.setVisible(true);
            }
            if(ae.getSource() == b3){
                setVisible(false);
                    Forgot forgot = new Forgot();
                    forgot.setVisible(true);
            }
        }

          public static void main(String[] args) {
              new Login_user().setVisible(true);
          }
```

```
}
```

4.1.5 Code For Forget Password:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.sql.*;
import java.awt.event.*;

public class Forgot extends JFrame implements ActionListener{

   private JPanel contentPane;
   private JTextField t1,t2,t3,t4,t5;
   private JButton b1,b2,b3;

   public static void main(String[] args) {
        new Forgot().setVisible(true);
   }

   public Forgot() {
```

```
setBounds(500, 200, 650, 500);
  contentPane = new JPanel();
  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
  setContentPane(contentPane);
contentPane.setBackground(Color.WHITE);
  contentPane.setLayout(null);

  JLabel l1 = new JLabel("Username");
  l1.setFont(new Font("Tahoma", Font.BOLD, 13));
  l1.setBounds(109, 83, 87, 29);
  contentPane.add(l1);

  JLabel l2 = new JLabel("Name");
  l2.setFont(new Font("Tahoma", Font.BOLD, 13));
  l2.setBounds(109, 122, 75, 21);
  contentPane.add(l2);

  JLabel l3 = new JLabel("Your Security Question");
  l3.setFont(new Font("Tahoma", Font.BOLD, 13));
  l3.setBounds(109, 154, 156, 27);
  contentPane.add(l3);

  JLabel l4 = new JLabel("Answer");
  l4.setFont(new Font("Tahoma", Font.BOLD, 13));
  l4.setBounds(109, 192, 104, 21);
```

```
contentPane.add(l4);

JLabel l5 = new JLabel("Password");
l5.setFont(new Font("Tahoma", Font.BOLD, 13));
l5.setBounds(109, 224, 104, 21);
contentPane.add(l5);

t1 = new JTextField();
t1.setFont(new Font("Tahoma", Font.BOLD, 13));
t1.setForeground(new Color(105, 105, 105));
t1.setBounds(277, 88, 139, 20);
contentPane.add(t1);
t1.setColumns(10);

t2 = new JTextField();
t2.setEditable(false);
t2.setFont(new Font("Tahoma", Font.BOLD, 13));
t2.setForeground(new Color(165, 42, 42));
t2.setColumns(10);
t2.setBounds(277, 123, 139, 20);
contentPane.add(t2);

t3 = new JTextField();
t3.setEditable(false);
t3.setFont(new Font("Tahoma", Font.BOLD, 12));
t3.setForeground(new Color(72, 61, 139));
```

```
  t3.setColumns(10);
  t3.setBounds(277, 161, 221, 20);
  contentPane.add(t3);

  t4 = new JTextField();
  t4.setFont(new Font("Tahoma", Font.BOLD, 13));
  t4.setForeground(new Color(205, 92, 92));
  t4.setColumns(10);
  t4.setBounds(277, 193, 139, 20);
  contentPane.add(t4);

  t5 = new JTextField();
  t5.setEditable(false);
  t5.setFont(new Font("Tahoma", Font.BOLD, 13));
  t5.setForeground(new Color(50, 205, 50));
  t5.setColumns(10);
  t5.setBounds(277, 225, 139, 20);
  contentPane.add(t5);

  b1 = new JButton("Search");
  b1.addActionListener(this);
  b1.setFont(new Font("Tahoma", Font.BOLD, 12));
  b1.setBounds(428, 83, 80, 29);
  b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
contentPane.add(b1);
```

```
        b2 = new JButton("Retrieve");
        b2.addActionListener(this);
        b2.setFont(new Font("Tahoma", Font.BOLD, 12));
        b2.setBounds(426, 188, 85, 29);
        b2.setBackground(Color.BLACK);
      b2.setForeground(Color.WHITE);
        contentPane.add(b2);

        b3 = new JButton("Back");
        b3.addActionListener(this);
        b3.setFont(new Font("Tahoma", Font.BOLD, 13));
        b3.setBounds(233, 270, 101, 29);
      b3.setBackground(Color.BLACK);
      b3.setForeground(Color.WHITE);
        contentPane.add(b3);

        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(new LineBorder(new Color(139,
69, 19), 2), "Forgot-Panel",
                        TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(178, 34, 34)));
        panel.setBounds(47, 45, 508, 281);
      panel.setBackground(Color.WHITE);
        contentPane.add(panel);
    }
```

```
public void actionPerformed(ActionEvent ae){
   try{
      conn con = new conn();
      if(ae.getSource() == b1){
         String sql = "select * from account where username=?";
            PreparedStatement st = con.c.prepareStatement(sql);

            st.setString(1, t1.getText());
            ResultSet rs = st.executeQuery();

            while (rs.next()) {
           t2.setText(rs.getString("name"));
           t3.setText(rs.getString("sec_q"));
            }

      }
      if(ae.getSource() == b2){
         String sql = "select * from account where sec_ans=?";
            PreparedStatement st = con.c.prepareStatement(sql);

            st.setString(1, t4.getText());
            ResultSet rs = st.executeQuery();

            while (rs.next()) {
            t5.setText(rs.getString("password"));
```

```
                }

            }
            if(ae.getSource() == b3){
                this.setVisible(false);
                    new Login_user().setVisible(true);

            }
        }catch(Exception e){

        }
    }

}
```

4.1.6 Code For Signup:

```
import java.awt.*;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;

public class Signup extends JFrame implements ActionListener{

    private JPanel contentPane;
```

```
private JTextField textField;
private JTextField textField_1;
private JTextField textField_2;
private JTextField textField_3;
private JButton b1, b2;
private JComboBox comboBox;


public static void main(String[] args) {
   new Signup().setVisible(true);
}

public Signup() {
   setBounds(600, 250, 606, 406);
     contentPane = new JPanel();
     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
     setContentPane(contentPane);
   contentPane.setBackground(Color.WHITE);
     contentPane.setLayout(null);

     JLabel lblUsername = new JLabel("Username :");
     lblUsername.setForeground(Color.DARK_GRAY);
     lblUsername.setFont(new Font("Tahoma", Font.BOLD, 14));
     lblUsername.setBounds(99, 86, 92, 26);
     contentPane.add(lblUsername);
```

```
        JLabel lblName = new JLabel("Name :");
        lblName.setForeground(Color.DARK_GRAY);
        lblName.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblName.setBounds(99, 123, 92, 26);
        contentPane.add(lblName);

        JLabel lblPassword = new JLabel("Password :");
        lblPassword.setForeground(Color.DARK_GRAY);
        lblPassword.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblPassword.setBounds(99, 160, 92, 26);
        contentPane.add(lblPassword);

        JLabel lblAnswer = new JLabel("Answer :");
        lblAnswer.setForeground(Color.DARK_GRAY);
        lblAnswer.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblAnswer.setBounds(99, 234, 92, 26);
        contentPane.add(lblAnswer);

        comboBox = new JComboBox();
        comboBox.setModel(new DefaultComboBoxModel(new String[]
{ "Your NickName?", "Your Lucky Number?",
                        "Your child SuperHero?", "Your childhood Name ?" }));
        comboBox.setBounds(265, 202, 148, 20);
        contentPane.add(comboBox);

        JLabel lblSecurityQuestion = new JLabel("Security Question :");
```

```
  lblSecurityQuestion.setForeground(Color.DARK_GRAY);
  lblSecurityQuestion.setFont(new Font("Tahoma", Font.BOLD, 14));
  lblSecurityQuestion.setBounds(99, 197, 140, 26);
  contentPane.add(lblSecurityQuestion);

textField = new JTextField();
  textField.setBounds(265, 91, 148, 20);
  contentPane.add(textField);
  textField.setColumns(10);

  textField_1 = new JTextField();
  textField_1.setColumns(10);
  textField_1.setBounds(265, 128, 148, 20);
  contentPane.add(textField_1);

textField_2 = new JTextField();
  textField_2.setColumns(10);
  textField_2.setBounds(265, 165, 148, 20);
  contentPane.add(textField_2);

  textField_3 = new JTextField();
  textField_3.setColumns(10);
  textField_3.setBounds(265, 239, 148, 20);
  contentPane.add(textField_3);

  b1 = new JButton("Create");
```

```
        b1.addActionListener(this);
        b1.setFont(new Font("Tahoma", Font.BOLD, 13));
        b1.setBounds(140, 289, 100, 30);
      b1.setBackground(Color.BLACK);
      b1.setForeground(Color.WHITE);
        contentPane.add(b1);

        b2 = new JButton("Back");
        b2.addActionListener(this);
        b2.setFont(new Font("Tahoma", Font.BOLD, 13));
        b2.setBounds(300, 289, 100, 30);
        b2.setBackground(Color.BLACK);
      b2.setForeground(Color.WHITE);
        contentPane.add(b2);

        JPanel panel = new JPanel();
        panel.setForeground(new Color(34, 139, 34));
        panel.setBorder(new TitledBorder(new LineBorder(new Color(128,
128, 0), 2), "Create-Account",
                        TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(34, 139, 34)));
        panel.setBounds(31, 46, 476, 296);
      panel.setBackground(Color.WHITE);
        contentPane.add(panel);
    }
```

```
    public void actionPerformed(ActionEvent ae){
        try{
            conn con = new conn();

            if(ae.getSource() == b1){
                    String sql = "insert into account(username, name, password,
sec_q, sec_ans) values(?, ?, ?, ?, ?)";
                PreparedStatement st = con.c.prepareStatement(sql);

                st.setString(1, textField.getText());
               st.setString(2, textField_1.getText());
                st.setString(3, textField_2.getText());
                st.setString(4, (String) comboBox.getSelectedItem());
                st.setString(5, textField_3.getText());

                int i = st.executeUpdate();
                if (i > 0){
               JOptionPane.showMessageDialog(null, "successfully Created");
               }

               textField.setText("");
               textField_1.setText("");
                textField_2.setText("");
                textField_3.setText("");
            }
            if(ae.getSource() == b2){
```

```
            this.setVisible(false);
                new Login_user().setVisible(true);

          }
       }catch(Exception e){

       }
   }
}
```

4.1.7 Code For AddBook :

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class AddBook extends JFrame implements ActionListener{

   private JPanel contentPane;
   private JTextField t1,t2,t3,t4,t5,t6;
   private JButton b1,b2;
   JComboBox comboBox;
```

```
public static void main(String[] args) {
      new AddBook().setVisible(true);
}

public void random() {
    Random rd = new Random();
      t1.setText("" + rd.nextInt(1000 + 1));
}

public AddBook() {
    setBounds(600, 200, 518, 442);
      contentPane = new JPanel();
      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
      setContentPane(contentPane);
      contentPane.setLayout(null);

      JLabel l1 = new JLabel("Name");
      l1.setForeground(new Color(47, 79, 79));
      l1.setFont(new Font("Tahoma", Font.BOLD, 14));
      l1.setBounds(73, 84, 90, 22);
      contentPane.add(l1);

      JLabel l2 = new JLabel("ISBN");
      l2.setForeground(new Color(47, 79, 79));
      l2.setFont(new Font("Tahoma", Font.BOLD, 14));
```

```
l2.setBounds(73, 117, 90, 22);
contentPane.add(l2);

JLabel l3 = new JLabel("Publisher");
l3.setForeground(new Color(47, 79, 79));
l3.setFont(new Font("Tahoma", Font.BOLD, 14));
l3.setBounds(73, 150, 90, 22);
contentPane.add(l3);

JLabel l4 = new JLabel("Price");
l4.setForeground(new Color(47, 79, 79));
l4.setFont(new Font("Tahoma", Font.BOLD, 14));
l4.setBounds(73, 216, 90, 22);
contentPane.add(l4);

JLabel l5 = new JLabel("Pages");
l5.setForeground(new Color(47, 79, 79));
l5.setFont(new Font("Tahoma", Font.BOLD, 14));
l5.setBounds(73, 249, 90, 22);
contentPane.add(l5);

JLabel l6 = new JLabel("Book_id");
l6.setForeground(new Color(47, 79, 79));
l6.setFont(new Font("Tahoma", Font.BOLD, 14));
l6.setBounds(73, 51, 90, 22);
contentPane.add(l6);
```

```
JLabel l7 = new JLabel("Edition");
l7.setForeground(new Color(47, 79, 79));
l7.setFont(new Font("Tahoma", Font.BOLD, 14));
l7.setBounds(73, 183, 90, 22);
contentPane.add(l7);

t1 = new JTextField();
t1.setForeground(new Color(47, 79, 79));
t1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t1.setBounds(169, 54, 198, 20);
contentPane.add(t1);
t1.setColumns(10);

t2 = new JTextField();
t2.setForeground(new Color(47, 79, 79));
t2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t2.setColumns(10);
t2.setBounds(169, 87, 198, 20);
contentPane.add(t2);

t3 = new JTextField();
t3.setForeground(new Color(47, 79, 79));
t3.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t3.setColumns(10);
t3.setBounds(169, 120, 198, 20);
```

```
        contentPane.add(t3);

        t4 = new JTextField();
        t4.setForeground(new Color(47, 79, 79));
        t4.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t4.setColumns(10);
        t4.setBounds(169, 153, 198, 20);
        contentPane.add(t4);

      t5 = new JTextField();
        t5.setForeground(new Color(47, 79, 79));
        t5.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t5.setColumns(10);
        t5.setBounds(169, 219, 198, 20);
        contentPane.add(t5);

        t6 = new JTextField();
        t6.setForeground(new Color(47, 79, 79));
        t6.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t6.setColumns(10);
        t6.setBounds(169, 252, 198, 20);
        contentPane.add(t6);

        comboBox = new JComboBox();
        comboBox.setModel(new DefaultComboBoxModel(new String[]
{ "1", "2", "3", "4", "5", "6", "7", "8", "9" }));
```

```
        comboBox.setBounds(173, 186, 194, 20);
        contentPane.add(comboBox);

      b1 = new JButton("Add");
        b1.addActionListener(this);
        b1.setBorder(new CompoundBorder(new LineBorder(new Color(128,
128, 128)), null));
      b1.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        b1.setBounds(102, 300, 100, 33);
      b1.setBackground(Color.BLACK);
      b1.setForeground(Color.WHITE);
        contentPane.add(b1);

        b2 = new JButton("Back");
        b2.addActionListener(this);
        b2.setBorder(new CompoundBorder(new LineBorder(new Color(105,
105, 105)), null));
        b2.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        b2.setBounds(212, 300, 108, 33);
      b2.setBackground(Color.BLACK);
      b2.setForeground(Color.WHITE);

        contentPane.add(b2);

        JPanel panel = new JPanel();
```

```
        panel.setBorder(new TitledBorder(new LineBorder(new Color(138,
43, 226), 2), "Add Books", TitledBorder.LEADING,
                        TitledBorder.TOP, null, new Color(0, 0, 255)));
        panel.setBounds(10, 29, 398, 344);
        contentPane.add(panel);

      panel.setBackground(Color.WHITE);
      contentPane.setBackground(Color.WHITE);
        random();

   }

   public void actionPerformed(ActionEvent ae){
      try{
         conn con = new conn();
         if(ae.getSource() == b1){
                String sql = "insert into book(book_id, name, isbn, publisher,
edition, price, pages) values(?, ?, ?, ?, ?, ?, ?)";
               PreparedStatement st = con.c.prepareStatement(sql);
            // st.setInt(1, Integer.parseInt(textField.getText()));
               st.setString(1, t1.getText());
               st.setString(2, t2.getText());
               st.setString(3, t3.getText());
               st.setString(4, t4.getText());
               st.setString(5, (String) comboBox.getSelectedItem());
               st.setString(6, t5.getText());
```

```
                st.setString(7, t6.getText());

                int rs = st.executeUpdate();
                if (rs > 0)
                JOptionPane.showMessageDialog(null, "Successfully Added");
                else
                JOptionPane.showMessageDialog(null, "Error");
            t1.setText("");
            t2.setText("");
            t3.setText("");
            t4.setText("");
            t5.setText("");
            t6.setText("");
                st.close();
            }
            if(ae.getSource() == b2){
                this.setVisible(false);
                    new Home().setVisible(true);

            }
            con.c.close();
        }catch(Exception e){

        }
    }
}
```

4.1.8 Code For Add Student:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class AddStudent extends JFrame implements ActionListener{

   private JPanel contentPane;
   private JTextField t1,t2,t3;
         private JComboBox comboBox, comboBox_1, comboBox_2, comboBox_3;
   JButton b1,b2;

   public static void main(String[] args) {
      new AddStudent().setVisible(true);
```

```
}

public void random() {
    Random rd = new Random();
      t1.setText("" + rd.nextInt(10000 + 1));
}

public AddStudent() {
    super("Add Student");
      setBounds(700, 200, 550, 450);
      contentPane = new JPanel();
      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
      setContentPane(contentPane);
      contentPane.setLayout(null);

      JLabel l1 = new JLabel("Student_id");
      l1.setForeground(new Color(25, 25, 112));
      l1.setFont(new Font("Tahoma", Font.BOLD, 14));
      l1.setBounds(64, 63, 102, 22);
      contentPane.add(l1);

      JLabel l2 = new JLabel("Name");
      l2.setForeground(new Color(25, 25, 112));
      l2.setFont(new Font("Tahoma", Font.BOLD, 14));
      l2.setBounds(64, 97, 102, 22);
      contentPane.add(l2);
```

```
JLabel l3 = new JLabel("Father's Name");
l3.setForeground(new Color(25, 25, 112));
l3.setFont(new Font("Tahoma", Font.BOLD, 14));
l3.setBounds(64, 130, 102, 22);
contentPane.add(l3);

JLabel l4 = new JLabel("Branch");
l4.setForeground(new Color(25, 25, 112));
l4.setFont(new Font("Tahoma", Font.BOLD, 14));
l4.setBounds(64, 209, 102, 22);
contentPane.add(l4);

JLabel l5 = new JLabel("Year");
l5.setForeground(new Color(25, 25, 112));
l5.setFont(new Font("Tahoma", Font.BOLD, 14));
l5.setBounds(64, 242, 102, 22);
contentPane.add(l5);

JLabel l6 = new JLabel("Semester");
l6.setForeground(new Color(25, 25, 112));
l6.setFont(new Font("Tahoma", Font.BOLD, 14));
l6.setBounds(64, 275, 102, 22);
contentPane.add(l6);

t1 = new JTextField();
```

```
//      t1.setEditable(false);
        t1.setForeground(new Color(47, 79, 79));
        t1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t1.setBounds(174, 66, 156, 20);
        contentPane.add(t1);
        t1.setColumns(10);

        t2 = new JTextField();
        t2.setForeground(new Color(47, 79, 79));
        t2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t2.setColumns(10);
        t2.setBounds(174, 100, 156, 20);
        contentPane.add(t2);

        t3 = new JTextField();
        t3.setForeground(new Color(47, 79, 79));
        t3.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t3.setColumns(10);
        t3.setBounds(174, 133, 156, 20);
        contentPane.add(t3);

        comboBox = new JComboBox();
        comboBox.setModel(new DefaultComboBoxModel(new String[]
{ "Mechanical", "CSE", "IT", "Civil", "Automobile", "EEE", "Other" }));
        comboBox.setForeground(new Color(47, 79, 79));
        comboBox.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
```

```
        comboBox.setBounds(176, 211, 154, 20);
        contentPane.add(comboBox);

        comboBox_1 = new JComboBox();
        comboBox_1.setModel(new DefaultComboBoxModel(new String[]
{ "First", "Second", "Third", "Four" }));
        comboBox_1.setForeground(new Color(47, 79, 79));
        comboBox_1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        comboBox_1.setBounds(176, 244, 154, 20);
        contentPane.add(comboBox_1);

        comboBox_2 = new JComboBox();
        comboBox_2.setModel(
                        new DefaultComboBoxModel(new String[] { "1st",
"2nd", "3rd", "4th", "5th", "6th", "7th", "8th" }));
        comboBox_2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        comboBox_2.setForeground(new Color(47, 79, 79));
        comboBox_2.setBounds(176, 277, 154, 20);
        contentPane.add(comboBox_2);

        b1 = new JButton("ADD");
        b1.addActionListener(this);
        b1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        b1.setBounds(64, 321, 111, 33);
       b1.setBackground(Color.BLACK);
       b1.setForeground(Color.WHITE);
```

```
        contentPane.add(b1);

        b2 = new JButton("Back");
        b2.addActionListener(this);
        b2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        b2.setBounds(198, 321, 111, 33);
       b2.setBackground(Color.BLACK);
       b2.setForeground(Color.WHITE);
        contentPane.add(b2);

        JLabel l7 = new JLabel("Course");
        l7.setForeground(new Color(25, 25, 112));
        l7.setFont(new Font("Tahoma", Font.BOLD, 14));
        l7.setBounds(64, 173, 102, 22);
        contentPane.add(l7);

        comboBox_3 = new JComboBox();
        comboBox_3.setModel(new DefaultComboBoxModel(
                        new String[] { "B.E", "B.Tech", "M.Tech", "MBA",
"BBA", "BCA", "B.Sc", "M.Sc", "B.Com", "M.Com" }));
        comboBox_3.setForeground(new Color(47, 79, 79));
        comboBox_3.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        comboBox_3.setBounds(176, 176, 154, 20);
        contentPane.add(comboBox_3);

        JPanel panel = new JPanel();
```

```
        panel.setBorder(new TitledBorder(new LineBorder(new Color(102, 205, 170), 2, true), "Add-Student",
                        TitledBorder.LEADING, TitledBorder.TOP, null, new Color(30, 144, 255)));
        panel.setBackground(new Color(211, 211, 211));
        panel.setBounds(10, 38, 358, 348);

      contentPane.setBackground(Color.WHITE);
      panel.setBackground(Color.WHITE);

        contentPane.add(panel);
        random();
    }

   public void actionPerformed(ActionEvent ae){
      try{

         if(ae.getSource() == b1){
            try{
            conn con = new conn();
              String sql = "insert into student(student_id, name, father, course, branch, year, semester) values(?, ?, ?, ?, ?, ?, ?)";
                PreparedStatement st = con.c.prepareStatement(sql);
                st.setString(1, t1.getText());
                st.setString(2, t2.getText());
                st.setString(3, t3.getText());
```

```
                st.setString(4, (String) comboBox_3.getSelectedItem());
                st.setString(5, (String) comboBox.getSelectedItem());
                st.setString(6, (String) comboBox_1.getSelectedItem());
                st.setString(7, (String) comboBox_2.getSelectedItem());

                int i = st.executeUpdate();
                if (i > 0){
                JOptionPane.showMessageDialog(null, "Successfully Added");
                this.setVisible(false);
                new Home().setVisible(true);
            }
                else
                JOptionPane.showMessageDialog(null, "error");
            }catch(Exception e){
                e.printStackTrace();
            }
        }

        if(ae.getSource() == b2){
            this.setVisible(false);
                new Home().setVisible(true);
        }
    }catch(Exception e){

    }
}
```

}

4.1.9 Code For Student Details:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.DbUtils;

public class StudentDetails extends JFrame implements ActionListener{

   private JPanel contentPane;
   private JTable table;
   private JTextField search;
   private JButton b1,b2;

   public static void main(String[] args) {
        new StudentDetails().setVisible(true);
   }

   public void student() {
      try {
         conn con = new conn();
```

```
        String sql = "select * from student";
        PreparedStatement st = con.c.prepareStatement(sql);
        ResultSet rs = st.executeQuery();

        table.setModel(DbUtils.resultSetToTableModel(rs));
        rs.close();
        st.close();
        con.c.close();
    } catch (Exception e) {

    }
}

public StudentDetails() {
    setBounds(350, 200, 890, 475);
    contentPane = new JPanel();
      contentPane.setBackground(new Color(220, 220, 220));
      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
      setContentPane(contentPane);
    contentPane.setBackground(Color.WHITE);
      contentPane.setLayout(null);

      JScrollPane scrollPane = new JScrollPane();
      scrollPane.setBounds(79, 133, 771, 288);
      contentPane.add(scrollPane);
```

```
        table = new JTable();
        table.addMouseListener(new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent arg0) {
            int row = table.getSelectedRow();
                search.setText(table.getModel().getValueAt(row, 1).toString());
         }
        });
        table.setBackground(new Color(240, 248, 255));
        table.setForeground(Color.DARK_GRAY);
        table.setFont(new Font("Trebuchet MS", Font.BOLD, 16));
        scrollPane.setViewportView(table);

        b1 = new JButton("Search");
        b1.addActionListener(this);
        b1.setBorder(new LineBorder(new Color(255, 20, 147), 2, true));
        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/eight.png"));
                    Image i2 = i1.getImage().getScaledInstance(20, 20, Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      b1.setIcon(i3);
        b1.setForeground(new Color(199, 21, 133));
        b1.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
        b1.setBounds(564, 89, 138, 33);
        contentPane.add(b1);
```

```
        b2 = new JButton("Delete");
        b2.addActionListener(this);
        ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("icons/nineth.png"));
                    Image i5 = i4.getImage().getScaledInstance(30, 30, Image.SCALE_DEFAULT);
      ImageIcon i6 = new ImageIcon(i5);
      b2.setIcon(i6);
        b2.setForeground(new Color(199, 21, 133));
        b2.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
        b2.setBorder(new LineBorder(new Color(255, 20, 147), 2, true));
        b2.setBounds(712, 89, 138, 33);
        contentPane.add(b2);

        JLabel l1 = new JLabel("Student Details");
        l1.setForeground(new Color(102, 205, 170));
        l1.setFont(new Font("Trebuchet MS", Font.BOLD | Font.ITALIC, 26));
        l1.setBounds(250, 20, 400, 47);
        contentPane.add(l1);


        search = new JTextField();
        search.setBackground(new Color(255, 240, 245));
        search.setBorder(new LineBorder(new Color(255, 105, 180), 2, true));
```

```
        search.setForeground(new Color(47, 79, 79));
        search.setFont(new Font("Trebuchet MS", Font.BOLD | Font.ITALIC, 17));
        search.setBounds(189, 89, 357, 33);
        contentPane.add(search);
        search.setColumns(10);

        JLabel l2 = new JLabel("Back");
        l2.addMouseListener(new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent e) {
            setVisible(false);
            Home home = new Home();
                home.setVisible(true);
         }
        });
        l2.setForeground(Color.GRAY);
        l2.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
        ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource("icons/tenth.png"));
                    Image i8 = i7.getImage().getScaledInstance(20, 20, Image.SCALE_DEFAULT);
       ImageIcon i9 = new ImageIcon(i8);
       l2.setIcon(i9);
         l2.setBounds(97, 89, 72, 33);
         contentPane.add(l2);
```

```
        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(new LineBorder(new Color(95,
158, 160), 3, true), "Student-Deatails",
                TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(72, 209, 204)));
        panel.setBounds(68, 59, 790, 370);
       panel.setBackground(Color.WHITE);
        contentPane.add(panel);

        student();
   }

  public void actionPerformed(ActionEvent ae){
   try{
      conn con = new conn();
      if( ae.getSource() == b1){
         // Search operation
           String sql = "select * from student where concat(name, student_id)
like ?";
          PreparedStatement st = con.c.prepareStatement(sql);
          st.setString(1, "%" + search.getText() + "%");
          ResultSet rs = st.executeQuery();

          table.setModel(DbUtils.resultSetToTableModel(rs));
          rs.close();
```

```
            st.close();
        }

        if(ae.getSource() == b2){
            // Delete operation
            String studentNameToDelete = search.getText();
            String sql = "delete from student where name = ?";
            PreparedStatement st = con.c.prepareStatement(sql);
            st.setString(1, studentNameToDelete);

            // Confirmation dialog
            JDialog.setDefaultLookAndFeelDecorated(true);
             int response = JOptionPane.showConfirmDialog(null, "Do you want to continue?", "Confirm",
                                                    JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
            if (response == JOptionPane.YES_OPTION) {
                try {
                    int rowsAffected = st.executeUpdate();
                    if (rowsAffected > 0) {
                            JOptionPane.showMessageDialog(null, "Student deleted successfully!");
                    } else {
                            JOptionPane.showMessageDialog(null, "No student found with the provided name!");
                    }
```

```
                // Commit the transaction
                con.c.commit();
            } catch (SQLException ex) {
                ex.printStackTrace();
                    JOptionPane.showMessageDialog(null, "SQL Exception occurred: " + ex.getMessage());
            }
        } else {
            // Do nothing if user chooses not to continue
        }
        st.close();
    }
    con.c.close();
  } catch(SQLException ex) {
    // Handle SQL exceptions
    ex.printStackTrace();
     JOptionPane.showMessageDialog(null, "SQL Exception occurred: " + ex.getMessage());
  } catch(Exception e){
    // Handle other exceptions
    e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Exception occurred: " + e.getMessage());
  }
}
```

```
}
```

4.1.10 Code For Book Details:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import java.awt.event.*;

public class BookDetails extends JFrame implements ActionListener{

   private JPanel contentPane;
   private JTable table;
   private JTextField search;
   private JButton b1,b2,b3;

   public static void main(String[] args) {
      // Add a print statement to verify that the main method is executed
      System.out.println("Main method executed");
      new BookDetails().setVisible(true);
   }

   public void Book() {
      try {
         conn con = new conn();
```

```
        String sql = "select * from book";
        PreparedStatement st = con.c.prepareStatement(sql);
        ResultSet rs = st.executeQuery();

        table.setModel(DbUtils.resultSetToTableModel(rs));
        rs.close();
        st.close();
        con.c.close();
    } catch (Exception e) {
        // Print stack trace for any caught exceptions
        e.printStackTrace();
    }
}

public BookDetails() {
    try {
        // Initialize frame components
        setBounds(350, 200, 890, 475);
        contentPane = new JPanel();
        contentPane.setBackground(Color.WHITE);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(79, 133, 771, 282);
```

```
        contentPane.add(scrollPane);

        table = new JTable();
        table.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                int row = table.getSelectedRow();
                search.setText(table.getModel().getValueAt(row, 1).toString());
            }
        });
        table.setBackground(new Color(240, 248, 255));
        table.setForeground(Color.DARK_GRAY);
        table.setFont(new Font("Trebuchet MS", Font.BOLD, 16));
        scrollPane.setViewportView(table);

        JButton b1 = new JButton("Search");
        b1.addActionListener(this);
        b1.setBorder(new LineBorder(new Color(255, 20, 147), 2, true));
                                        ImageIcon i1 = new
ImageIcon(getClass().getResource("icons/eight.png"));

                        Image i2 = i1.getImage().getScaledInstance(20, 20,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        b1.setIcon(i3);
        b1.setForeground(new Color(199, 21, 133));
```

```
        b1.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
        b1.setBounds(564, 89, 138, 33);
        contentPane.add(b1);

        b2 = new JButton("Delete");
        b2.addActionListener(this);
                                                ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("icons/nineth.png"));
                        Image i5 = i4.getImage().getScaledInstance(30, 30,
Image.SCALE_DEFAULT);
        ImageIcon i6 = new ImageIcon(i5);
        b2.setIcon(i6);
        b2.setForeground(new Color(199, 21, 133));
        b2.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
        b2.setBorder(new LineBorder(new Color(255, 20, 147), 2, true));
        b2.setBounds(712, 89, 138, 33);
        contentPane.add(b2);

        JLabel l1 = new JLabel("Book Details");
        l1.setForeground(new Color(107, 142, 35));
          l1.setFont(new Font("Trebuchet MS", Font.BOLD | Font.ITALIC,
30));
        l1.setBounds(300, 15, 400, 47);
        contentPane.add(l1);

        search = new JTextField();
```

```
        search.setBackground(new Color(255, 240, 245));
            search.setBorder(new LineBorder(new Color(255, 105, 180), 2,
true));
        search.setForeground(new Color(47, 79, 79));
                    search.setFont(new Font("Trebuchet MS", Font.BOLD |
Font.ITALIC, 17));
        search.setBounds(189, 89, 357, 33);
        contentPane.add(search);
        search.setColumns(10);

        JLabel l3 = new JLabel("Back");
        l3.addMouseListener(new MouseAdapter() {
           @Override
           public void mouseClicked(MouseEvent e) {
              setVisible(false);
              Home home = new Home();
              home.setVisible(true);
           }
        });
        l3.setForeground(Color.GRAY);
        l3.setFont(new Font("Trebuchet MS", Font.BOLD, 18));
                                                ImageIcon i7 = new
ImageIcon(ClassLoader.getSystemResource("icons/tenth.png"));
                        Image i8 = i7.getImage().getScaledInstance(20, 20,
Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
```

```
        l3.setIcon(i9);
        l3.setBounds(97, 89, 72, 33);
        contentPane.add(l3);

        JPanel panel = new JPanel();
            panel.setBorder(new TitledBorder(new LineBorder(new Color(0,
128, 128), 3, true), "Book-Details",
                        TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(0, 128, 0)));
        panel.setBounds(67, 54, 793, 368);
        contentPane.add(panel);
        panel.setBackground(Color.WHITE);
        Book(); // Load book data
      } catch (Exception e) {
        // Print stack trace for any caught exceptions
        e.printStackTrace();
      }
   }

   public void actionPerformed(ActionEvent ae){
      try{

        conn con = new conn();
        if(ae.getSource() == b1){
```

```
            String sql = "select * from book where concat(name, book_id) like
?";
            PreparedStatement st = con.c.prepareStatement(sql);
            st.setString(1, "%" + search.getText() + "%");
            ResultSet rs = st.executeQuery();

            table.setModel(DbUtils.resultSetToTableModel(rs));
            rs.close();
            st.close();

        }
        if(ae.getSource() == b2){
            String sql = "delete from book where name = ?";
            PreparedStatement st = con.c.prepareStatement(sql);
            st.setString(1, search.getText());

            JDialog.setDefaultLookAndFeelDecorated(true);
                int response = JOptionPane.showConfirmDialog(null, "Do you
want to continue?", "Confirm",
                                                JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
            if (response == JOptionPane.NO_OPTION) {
                // Do nothing
            } else if (response == JOptionPane.YES_OPTION) {
                int rs = st.executeUpdate();
                JOptionPane.showMessageDialog(null, "Deleted !!");
```

```
            } else if (response == JOptionPane.CLOSED_OPTION) {
                // Do nothing
            }
            st.close();
        }

        con.c.close();
    }catch(Exception e){
        // Print stack trace for any caught exceptions
        e.printStackTrace();
    }
  }
}
```

4.1.11 Code For ReturnBook :

```
import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import com.toedter.calendar.JDateChooser;
import java.awt.event.*;
import java.sql.*;
import java.text.SimpleDateFormat;

public class ReturnBook extends JFrame implements ActionListener{

  private JPanel contentPane;
```

```
private JTextField textField;
private JTextField textField_1;
private JTextField textField_2;
private JTextField textField_3;
private JTextField textField_4;
private JTextField textField_5;
private JTextField textField_6;
private JButton b1,b2,b3;
private JDateChooser dateChooser;

public static void main(String[] args) {
   new ReturnBook().setVisible(true);
}

public void delete() {
   try {
      conn con = new conn();
      String sql = "delete from issueBook where book_id=?";
      PreparedStatement st = con.c.prepareStatement(sql);
      st.setString(1, textField.getText());
      int i = st.executeUpdate();
      if (i > 0)
         JOptionPane.showConfirmDialog(null, "Book Returned");
      else
         JOptionPane.showMessageDialog(null, "error in Deleting");
   } catch (SQLException e) {
```

```
            JOptionPane.showMessageDialog(null, e);
            e.printStackTrace();
        }
    }


    public ReturnBook() {
        setBounds(450, 300, 617, 363);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setBackground(Color.WHITE);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Book_id");
        lblNewLabel.setForeground(new Color(0, 0, 0));
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblNewLabel.setBounds(52, 52, 87, 24);
        contentPane.add(lblNewLabel);

        JLabel lblStudentid = new JLabel("Student_id");
        lblStudentid.setForeground(Color.BLACK);
        lblStudentid.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblStudentid.setBounds(243, 52, 87, 24);
        contentPane.add(lblStudentid);
```

```
JLabel lblBook = new JLabel("Book");
lblBook.setForeground(Color.BLACK);
lblBook.setFont(new Font("Tahoma", Font.BOLD, 14));
lblBook.setBounds(52, 98, 71, 24);
contentPane.add(lblBook);

JLabel lblName = new JLabel("Name");
lblName.setForeground(Color.BLACK);
lblName.setFont(new Font("Tahoma", Font.BOLD, 14));
lblName.setBounds(300, 98, 71, 24);
contentPane.add(lblName);

JLabel lblCourse = new JLabel("Course");
lblCourse.setForeground(Color.BLACK);
lblCourse.setFont(new Font("Tahoma", Font.BOLD, 14));
lblCourse.setBounds(52, 143, 87, 24);
contentPane.add(lblCourse);

JLabel lblBranch = new JLabel("Branch");
lblBranch.setForeground(Color.BLACK);
lblBranch.setFont(new Font("Tahoma", Font.BOLD, 14));
lblBranch.setBounds(303, 144, 68, 24);
contentPane.add(lblBranch);

JLabel lblDateOfIssue = new JLabel("Date of Issue");
lblDateOfIssue.setForeground(Color.BLACK);
```

```
lblDateOfIssue.setFont(new Font("Tahoma", Font.BOLD, 14));
lblDateOfIssue.setBounds(52, 188, 105, 29);
contentPane.add(lblDateOfIssue);

JLabel lblDateOfReturn = new JLabel("Date of Return");
lblDateOfReturn.setForeground(Color.BLACK);
lblDateOfReturn.setFont(new Font("Tahoma", Font.BOLD, 14));
lblDateOfReturn.setBounds(52, 234, 118, 29);
contentPane.add(lblDateOfReturn);

textField = new JTextField();
textField.setForeground(new Color(105, 105, 105));
textField.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
textField.setBounds(128, 56, 105, 20);
contentPane.add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setForeground(new Color(105, 105, 105));
textField_1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
textField_1.setBounds(340, 56, 93, 20);
contentPane.add(textField_1);
textField_1.setColumns(10);

b1 = new JButton("Search");
b1.addActionListener(this);
```

```
b1.setBounds(443, 52, 105, 29);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
contentPane.add(b1);

textField_2 = new JTextField();
textField_2.setEditable(false);
textField_2.setForeground(new Color(0, 100, 0));
textField_2.setFont(new Font("Trebuchet MS", Font.BOLD, 13));
textField_2.setBounds(128, 102, 162, 20);
contentPane.add(textField_2);
textField_2.setColumns(10);

textField_3 = new JTextField();
textField_3.setEditable(false);
textField_3.setForeground(new Color(0, 100, 0));
textField_3.setFont(new Font("Trebuchet MS", Font.BOLD, 13));
textField_3.setColumns(10);
textField_3.setBounds(369, 102, 179, 20);
contentPane.add(textField_3);

textField_4 = new JTextField();
textField_4.setEditable(false);
textField_4.setForeground(new Color(0, 100, 0));
textField_4.setFont(new Font("Trebuchet MS", Font.BOLD, 13));
textField_4.setColumns(10);
```

```
textField_4.setBounds(128, 147, 162, 20);
contentPane.add(textField_4);

textField_5 = new JTextField();
textField_5.setForeground(new Color(0, 100, 0));
textField_5.setFont(new Font("Trebuchet MS", Font.BOLD, 13));
textField_5.setEditable(false);
textField_5.setColumns(10);
textField_5.setBounds(369, 147, 179, 20);
contentPane.add(textField_5);

textField_6 = new JTextField();
textField_6.setForeground(new Color(0, 100, 0));
textField_6.setFont(new Font("Trebuchet MS", Font.BOLD, 13));
textField_6.setEditable(false);
textField_6.setColumns(10);
textField_6.setBounds(167, 194, 162, 20);
contentPane.add(textField_6);

dateChooser = new JDateChooser();
dateChooser.setBorder(new LineBorder(new Color(0, 0, 0), 0, true));
dateChooser.setBounds(167, 234, 172, 29);
contentPane.add(dateChooser);

b2 = new JButton("Return");
b2.addActionListener(this);
```

```
        b2.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        b2.setBorder(new LineBorder(new Color(0, 0, 0), 0, true));
        b2.setBounds(369, 179, 149, 30);
        b2.setBackground(Color.BLACK);
        b2.setForeground(Color.WHITE);
        contentPane.add(b2);

        b3 = new JButton("Back");
        b3.addActionListener(this);
        b3.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
        b3.setBorder(new LineBorder(new Color(0, 0, 0), 0, true));
        b3.setBounds(369, 231, 149, 30);
        b3.setBackground(Color.BLACK);
        b3.setForeground(Color.WHITE);
        contentPane.add(b3);

        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(new LineBorder(new Color(255, 69,
0), 2, true), "Return-Panel",
                TitledBorder.LEADING, TitledBorder.TOP, null, new Color(220,
20, 60)));
        panel.setBounds(10, 24, 569, 269);
        panel.setBackground(Color.WHITE);
        contentPane.add(panel);
    }
```

```
   public void actionPerformed(ActionEvent ae){
      try{
         conn con = new conn();
         if(ae.getSource() == b1){
               String sql = "select * from issueBook where student_id = ? and
book_id =?";
            try (PreparedStatement st = con.c.prepareStatement(sql)) {
               st.setString(1, textField_1.getText());
               st.setString(2, textField.getText());
               try (ResultSet rs = st.executeQuery()) {
                  if (rs.next()) {
                     textField_2.setText(rs.getString("bname"));
                     textField_3.setText(rs.getString("sname"));
                     textField_4.setText(rs.getString("course"));
                     textField_5.setText(rs.getString("branch"));
                     // Format the date before setting it to the text field
                        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-
MM-dd");
                                                   String formattedDate =
sdf.format(rs.getDate("dateOfIssue"));
                     textField_6.setText(formattedDate);
                  } else {
                      JOptionPane.showMessageDialog(null, "No records found
for the given student ID and book ID");
                  }
               }
```

```
            } catch (SQLException e) {
                                JOptionPane.showMessageDialog(null, "Error: " +
e.getMessage());
                e.printStackTrace();
            }
        }
        if(ae.getSource() == b2){
            String sql = "insert into returnBook(book_id, student_id, bname,
sname,course, branch, dateOfIssue, dateOfReturn)
values(?, ?, ?, ?, ?, ?, ?, ?)";
            try {
                PreparedStatement st = con.c.prepareStatement(sql);
                st.setString(1, textField.getText());
                st.setString(2, textField_1.getText());
                st.setString(3, textField_2.getText());
                st.setString(4, textField_3.getText());
                st.setString(5, textField_4.getText());
                st.setString(6, textField_5.getText());
                st.setString(7, textField_6.getText());

                // Format the date before setting it to the prepared statement
                  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-
dd");
                String formattedDate = sdf.format(dateChooser.getDate());
                st.setString(8, formattedDate);
```

```
                int i = st.executeUpdate();
                if (i > 0) {
                    JOptionPane.showMessageDialog(null, "Processing..");
                    delete();
                } else {
                    JOptionPane.showMessageDialog(null, "error");
                }
            } catch (SQLException e) {
                        JOptionPane.showMessageDialog(null,  "Error:  "  +
e.getMessage());
                e.printStackTrace();
            }
        }
        if(ae.getSource() == b3){
            this.setVisible(false);
            new Home().setVisible(true);
        }
    } catch(Exception e){
        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
        e.printStackTrace();
    }
  }
}
```

4.1.12 Code For IssueBook:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import com.toedter.calendar.JDateChooser;
import java.awt.event.*;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.sql.Date;
import java.text.ParseException;



public class IssueBook extends JFrame implements ActionListener{

   private JPanel contentPane;
   JDateChooser dateChooser;
   private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14;
   private JButton b1,b2,b3,b4;

   public static void main(String[] args) {
        new IssueBook().setVisible(true);

   }
```

```
public IssueBook() {
    setBounds(300, 200, 900, 500);
      contentPane = new JPanel();
      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
      setContentPane(contentPane);
    contentPane.setBackground(Color.WHITE);
      contentPane.setLayout(null);

      JLabel l1 = new JLabel("Book_id");
      l1.setFont(new Font("Tahoma", Font.BOLD, 14));
      l1.setForeground(new Color(47, 79, 79));
      l1.setBounds(47, 63, 100, 23);
      contentPane.add(l1);

      JLabel l2 = new JLabel("Name");
      l2.setForeground(new Color(47, 79, 79));
      l2.setFont(new Font("Tahoma", Font.BOLD, 14));
      l2.setBounds(47, 97, 100, 23);
      contentPane.add(l2);

      JLabel l3 = new JLabel("ISBN");
      l3.setForeground(new Color(47, 79, 79));
      l3.setFont(new Font("Tahoma", Font.BOLD, 14));
      l3.setBounds(47, 131, 100, 23);
      contentPane.add(l3);
```

```
JLabel l4 = new JLabel("Publisher");
l4.setForeground(new Color(47, 79, 79));
l4.setFont(new Font("Tahoma", Font.BOLD, 14));
l4.setBounds(47, 165, 100, 23);
contentPane.add(l4);

JLabel l5 = new JLabel("Edition");
l5.setForeground(new Color(47, 79, 79));
l5.setFont(new Font("Tahoma", Font.BOLD, 14));
l5.setBounds(47, 199, 100, 23);
contentPane.add(l5);

JLabel l6 = new JLabel("Price");
l6.setForeground(new Color(47, 79, 79));
l6.setFont(new Font("Tahoma", Font.BOLD, 14));
l6.setBounds(47, 233, 100, 23);
contentPane.add(l6);

JLabel l7 = new JLabel("Pages");
l7.setForeground(new Color(47, 79, 79));
l7.setFont(new Font("Tahoma", Font.BOLD, 14));
l7.setBounds(47, 267, 100, 23);
contentPane.add(l7);

t1 = new JTextField();
t1.setForeground(new Color(47, 79, 79));
```

```
  t1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
  t1.setBounds(126, 66, 86, 20);
  contentPane.add(t1);

  b1 = new JButton("Search");
  b1.addActionListener(this);
  b1.setBorder(new LineBorder(new Color(192, 192, 192), 1, true));
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
  b1.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
  b1.setBounds(234, 59, 100, 30);

  contentPane.add(b1);

  t2 = new JTextField();
  t2.setEditable(false);
  t2.setForeground(new Color(47, 79, 79));
  t2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
  t2.setBounds(126, 100, 208, 20);
  contentPane.add(t2);
  t2.setColumns(10);

  t3 = new JTextField();
  t3.setEditable(false);
  t3.setForeground(new Color(47, 79, 79));
  t3.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
```

```
t3.setColumns(10);
t3.setBounds(126, 131, 208, 20);
contentPane.add(t3);

t4 = new JTextField();
t4.setEditable(false);
t4.setForeground(new Color(47, 79, 79));
t4.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t4.setColumns(10);
t4.setBounds(126, 168, 208, 20);
contentPane.add(t4);

t5 = new JTextField();
t5.setEditable(false);
t5.setForeground(new Color(47, 79, 79));
t5.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t5.setColumns(10);
t5.setBounds(126, 202, 208, 20);
contentPane.add(t5);

t6 = new JTextField();
t6.setEditable(false);
t6.setForeground(new Color(47, 79, 79));
t6.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t6.setColumns(10);
t6.setBounds(126, 236, 208, 20);
```

```
        contentPane.add(t6);

        t7 = new JTextField();
        t7.setEditable(false);
        t7.setForeground(new Color(47, 79, 79));
        t7.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
        t7.setColumns(10);
        t7.setBounds(126, 270, 208, 20);
        contentPane.add(t7);

        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(new LineBorder(new Color(47, 79,
79), 2, true), "Issue-Book",
                                TitledBorder.LEADING,  TitledBorder.TOP,  null,
new Color(34, 139, 34)));
        panel.setFont(new Font("Tahoma", Font.BOLD, 14));
        panel.setBounds(10, 38, 345, 288);
       panel.setBackground(Color.WHITE);
        contentPane.add(panel);

        JLabel l8 = new JLabel("Student_id");
        l8.setForeground(new Color(47, 79, 79));
        l8.setFont(new Font("Tahoma", Font.BOLD, 14));
        l8.setBounds(384, 63, 100, 23);
        contentPane.add(l8);
```

```
JLabel l9 = new JLabel("Name");
l9.setForeground(new Color(47, 79, 79));
l9.setFont(new Font("Tahoma", Font.BOLD, 14));
l9.setBounds(384, 103, 100, 23);
contentPane.add(l9);

JLabel l10 = new JLabel("Father's Name");
l10.setForeground(new Color(47, 79, 79));
l10.setFont(new Font("Tahoma", Font.BOLD, 14));
l10.setBounds(384, 147, 100, 23);
contentPane.add(l10);

JLabel l11 = new JLabel("Course");
l11.setForeground(new Color(47, 79, 79));
l11.setFont(new Font("Tahoma", Font.BOLD, 14));
l11.setBounds(384, 187, 100, 23);
contentPane.add(l11);

JLabel l12 = new JLabel("Branch");
l12.setForeground(new Color(47, 79, 79));
l12.setFont(new Font("Tahoma", Font.BOLD, 14));
l12.setBounds(384, 233, 100, 23);
contentPane.add(l12);

JLabel l13 = new JLabel("Year");
l13.setForeground(new Color(47, 79, 79));
```

```
  l13.setFont(new Font("Tahoma", Font.BOLD, 14));
  l13.setBounds(384, 284, 100, 23);
  contentPane.add(l13);

  JLabel l14 = new JLabel("Semester");
  l14.setForeground(new Color(47, 79, 79));
  l14.setFont(new Font("Tahoma", Font.BOLD, 14));
  l14.setBounds(384, 336, 100, 23);
  contentPane.add(l14);

  t8 = new JTextField();
  t8.setForeground(new Color(47, 79, 79));
  t8.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
  t8.setColumns(10);
  t8.setBounds(508, 66, 86, 20);
  contentPane.add(t8);

  b2 = new JButton("Search");
  b2.addActionListener(this);
  b2.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
  b2.setBorder(new LineBorder(new Color(192, 192, 192), 1, true));
  b2.setBounds(604, 59, 100, 30);
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
  contentPane.add(b2);
```

```
t9 = new JTextField();
t9.setForeground(new Color(47, 79, 79));
t9.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t9.setEditable(false);
t9.setColumns(10);
t9.setBounds(508, 106, 208, 20);
contentPane.add(t9);

t10 = new JTextField();
t10.setForeground(new Color(47, 79, 79));
t10.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t10.setEditable(false);
t10.setColumns(10);
t10.setBounds(508, 150, 208, 20);
contentPane.add(t10);

t11 = new JTextField();
t11.setForeground(new Color(47, 79, 79));
t11.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t11.setEditable(false);
t11.setColumns(10);
t11.setBounds(508, 190, 208, 20);
contentPane.add(t11);

t12 = new JTextField();
t12.setForeground(new Color(47, 79, 79));
```

```
t12.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t12.setEditable(false);
t12.setColumns(10);
t12.setBounds(508, 236, 208, 20);
contentPane.add(t12);

t13 = new JTextField();
t13.setForeground(new Color(47, 79, 79));
t13.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t13.setEditable(false);
t13.setColumns(10);
t13.setBounds(508, 286, 208, 20);
contentPane.add(t13);

t14 = new JTextField();
t14.setForeground(new Color(47, 79, 79));
t14.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
t14.setEditable(false);
t14.setColumns(10);
t14.setBounds(508, 338, 208, 20);
contentPane.add(t14);

JPanel panel_1 = new JPanel();
panel_1.setBorder(new TitledBorder(new LineBorder(new Color(70, 130, 180), 2, true), "Student-Deatails",
```

```
                    TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(100, 149, 237)));
      panel_1.setForeground(new Color(0, 100, 0));
      panel_1.setBounds(360, 38, 378, 372);
    panel_1.setBackground(Color.WHITE);
      contentPane.add(panel_1);

      JLabel l15 = new JLabel(" Date of Issue :");
      l15.setForeground(new Color(105, 105, 105));
      l15.setFont(new Font("Trebuchet MS", Font.BOLD, 15));
      l15.setBounds(20, 340, 118, 26);
      contentPane.add(l15);

      dateChooser = new JDateChooser();
      dateChooser.setBorder(new LineBorder(new Color(0, 0, 0), 1, true));
      dateChooser.setForeground(new Color(105, 105, 105));
      dateChooser.setBounds(137, 337, 200, 29);
      contentPane.add(dateChooser);

      b3 = new JButton("Issue");
      b3.addActionListener(this);
      b3.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
      b3.setBorder(new LineBorder(new Color(192, 192, 192), 1, true));
      b3.setBounds(47, 377, 118, 33);
      b3.setBackground(Color.BLACK);
    b3.setForeground(Color.WHITE);
```

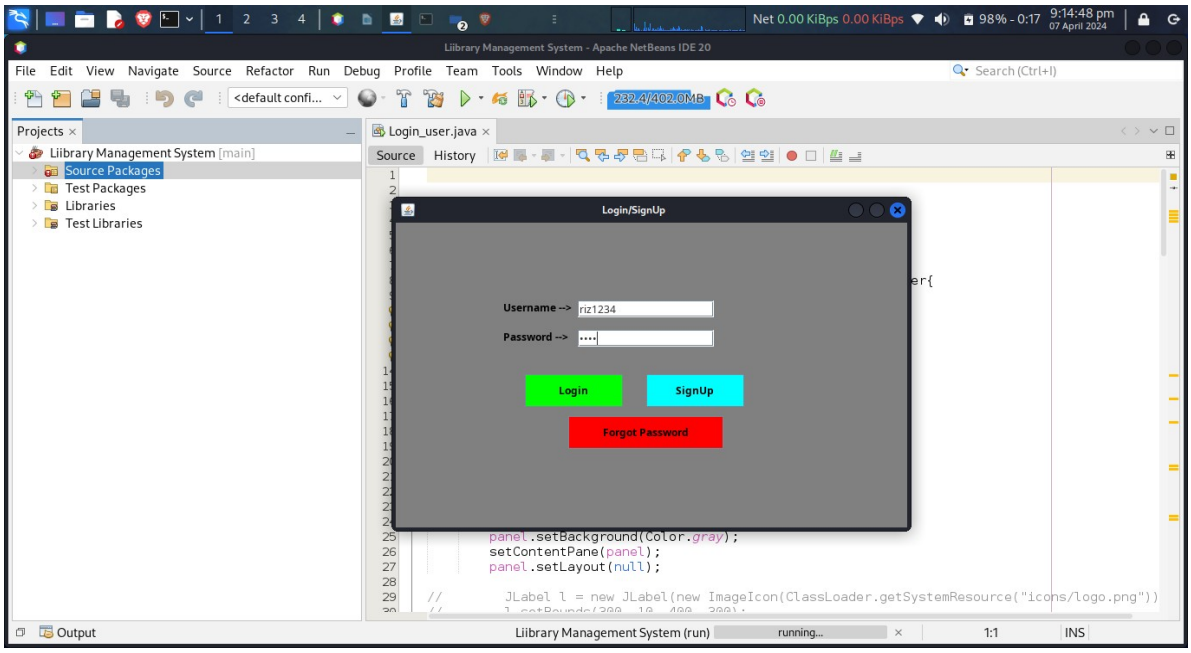
```
    contentPane.add(b3);

      b4 = new JButton("Back");
      b4.addActionListener(this);
      b4.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
      b4.setBorder(new LineBorder(new Color(192, 192, 192), 1, true));
      b4.setBounds(199, 377, 100, 33);
      b4.setBackground(Color.BLACK);
    b4.setForeground(Color.WHITE);
    contentPane.add(b4);
  }

public void actionPerformed(ActionEvent ae) {
  try {
    conn con = new conn();
    if (ae.getSource() == b1) {
      String sql = "select * from book where book_id = ?";
      try (PreparedStatement st = con.c.prepareStatement(sql)) {
        st.setString(1, t1.getText());
        try (ResultSet rs = st.executeQuery()) {
          if (rs.next()) {
            t2.setText(rs.getString("name"));
            t3.setText(rs.getString("isbn"));
            t4.setText(rs.getString("publisher"));
            t5.setText(rs.getString("edition"));
            t6.setText(rs.getString("price"));
```

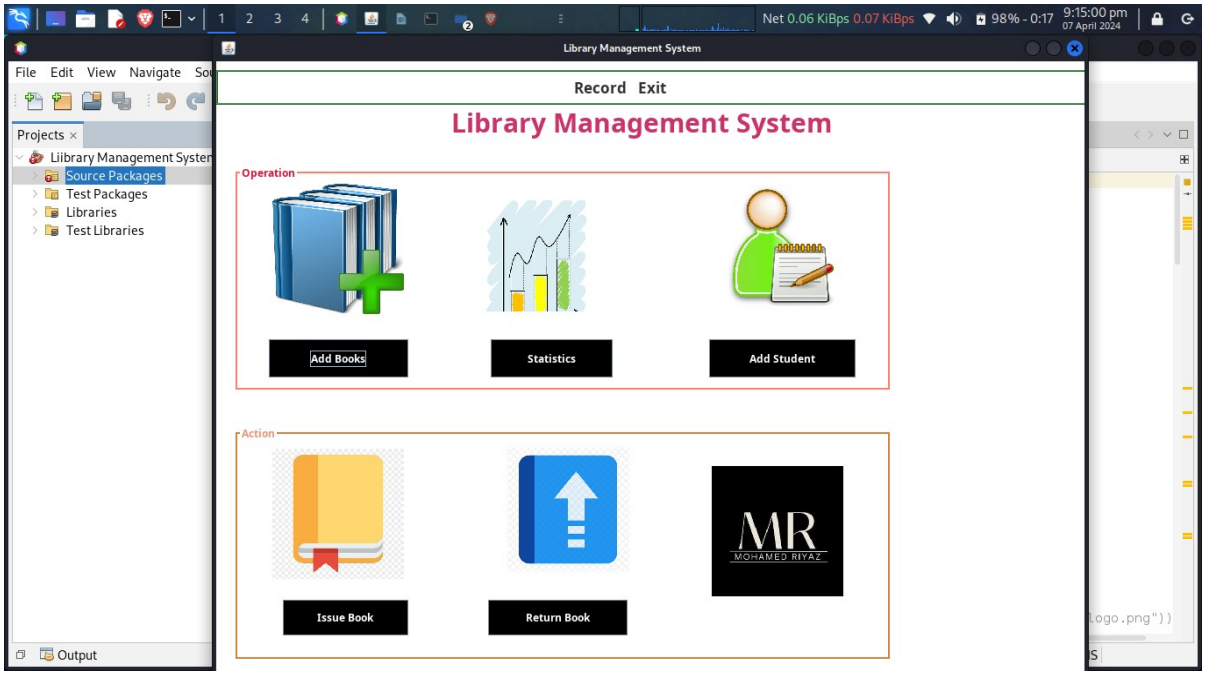
```
                t7.setText(rs.getString("pages"));
            } else {
                    JOptionPane.showMessageDialog(null, "Book ID not
found");
            }
        }
    } else if (ae.getSource() == b2) {
        String sql = "select * from student where student_id = ?";
        try (PreparedStatement st = con.c.prepareStatement(sql)) {
            st.setString(1, t8.getText());
            try (ResultSet rs = st.executeQuery()) {
                if (rs.next()) {
                    t9.setText(rs.getString("name"));
                    t10.setText(rs.getString("father"));
                    t11.setText(rs.getString("course"));
                    t12.setText(rs.getString("branch"));
                    t13.setText(rs.getString("year"));
                    t14.setText(rs.getString("semester"));
                } else {
                        JOptionPane.showMessageDialog(null, "Student ID not
found");
                }
            }
        }
    } else if (ae.getSource() == b3) {
```

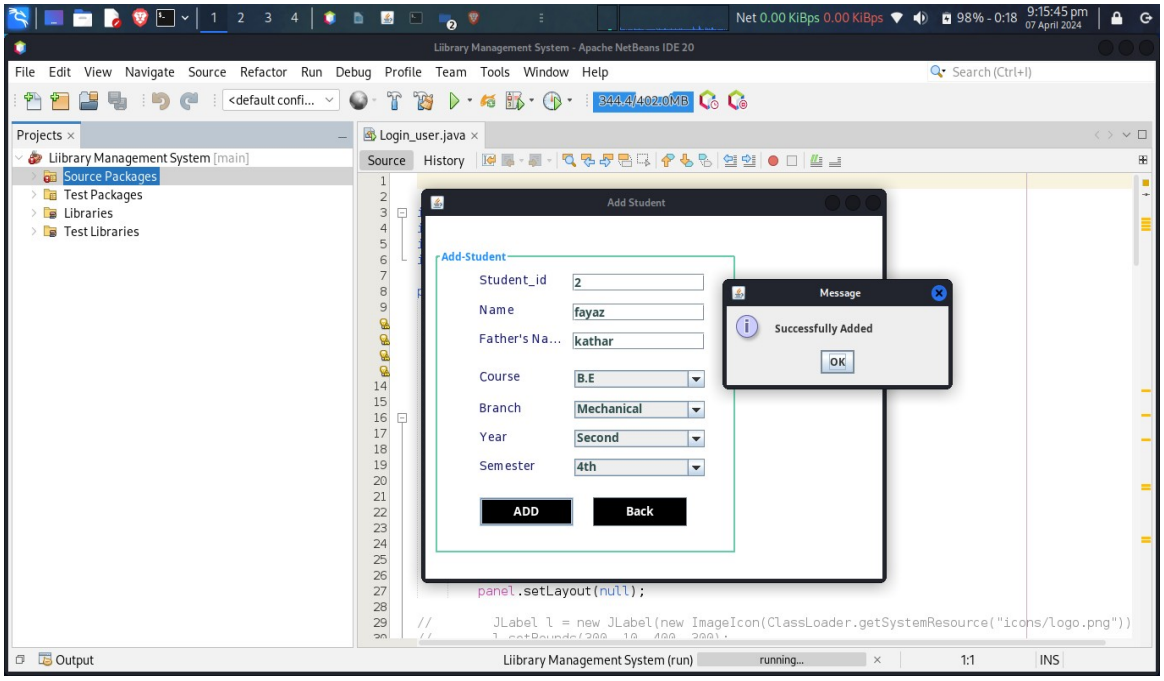
```
                                String dateOfIssue = ((JTextField) dateChooser.getDateEditor().getUiComponent()).getText();
        if (dateOfIssue.isEmpty()) {
            JOptionPane.showMessageDialog(null, "Please select a date of issue");
        } else {
            String sql = "insert into issueBook(book_id, student_id, bname, sname, course, branch, dateOfIssue) values(?, ?, ?, ?, ?, ?, ?)";
            try (PreparedStatement st = con.c.prepareStatement(sql)) {
                st.setString(1, t1.getText());
                st.setString(2, t8.getText());
                st.setString(3, t2.getText());
                st.setString(4, t9.getText());
                st.setString(5, t11.getText());
                st.setString(6, t12.getText());
                java.util.Date selectedDate = dateChooser.getDate();
                long milliseconds = selectedDate.getTime();
                java.sql.Date sqlDate = new java.sql.Date(milliseconds);
                st.setDate(7, sqlDate);
                int i = st.executeUpdate();
                if (i > 0) {
                    JOptionPane.showMessageDialog(null, "Successfully Book Issued..!");
                } else {
                        JOptionPane.showMessageDialog(null, "Error issuing book");
```

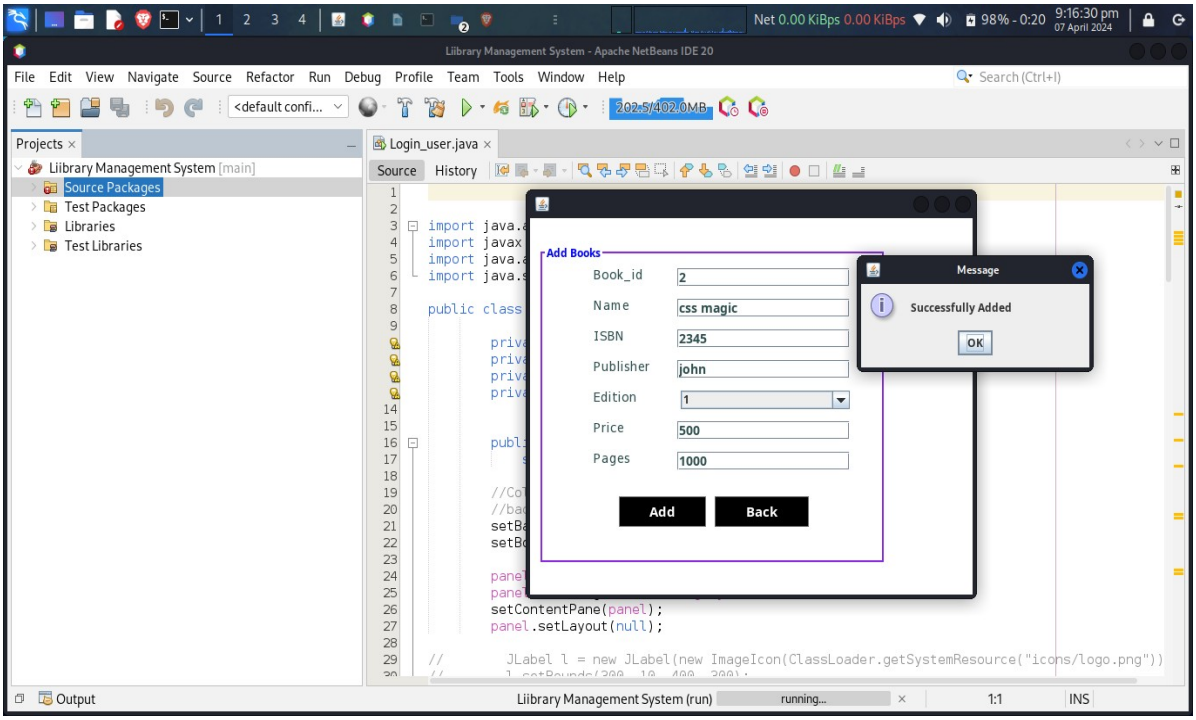
```
                }
            }
        }
    } else if (ae.getSource() == b4) {
        this.setVisible(false);
        new Home().setVisible(true);
    }
    con.c.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
```

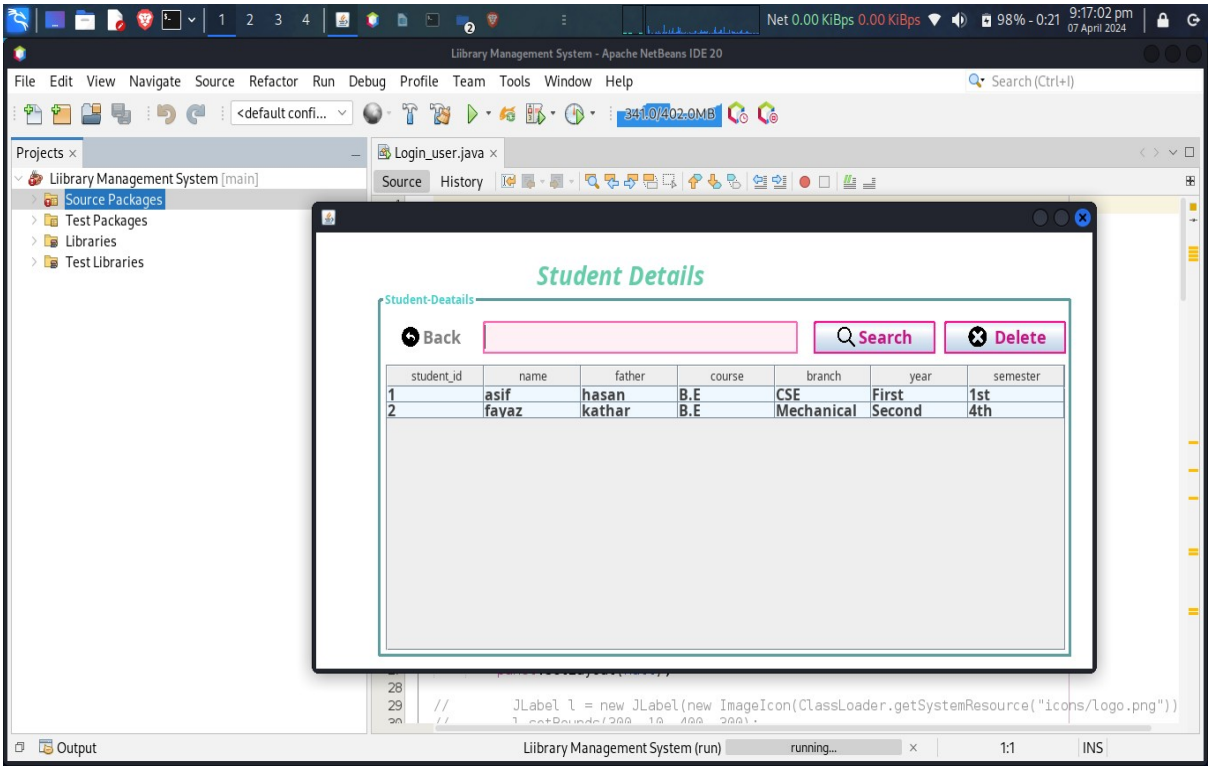
4.2 SCREEN SHOTS:

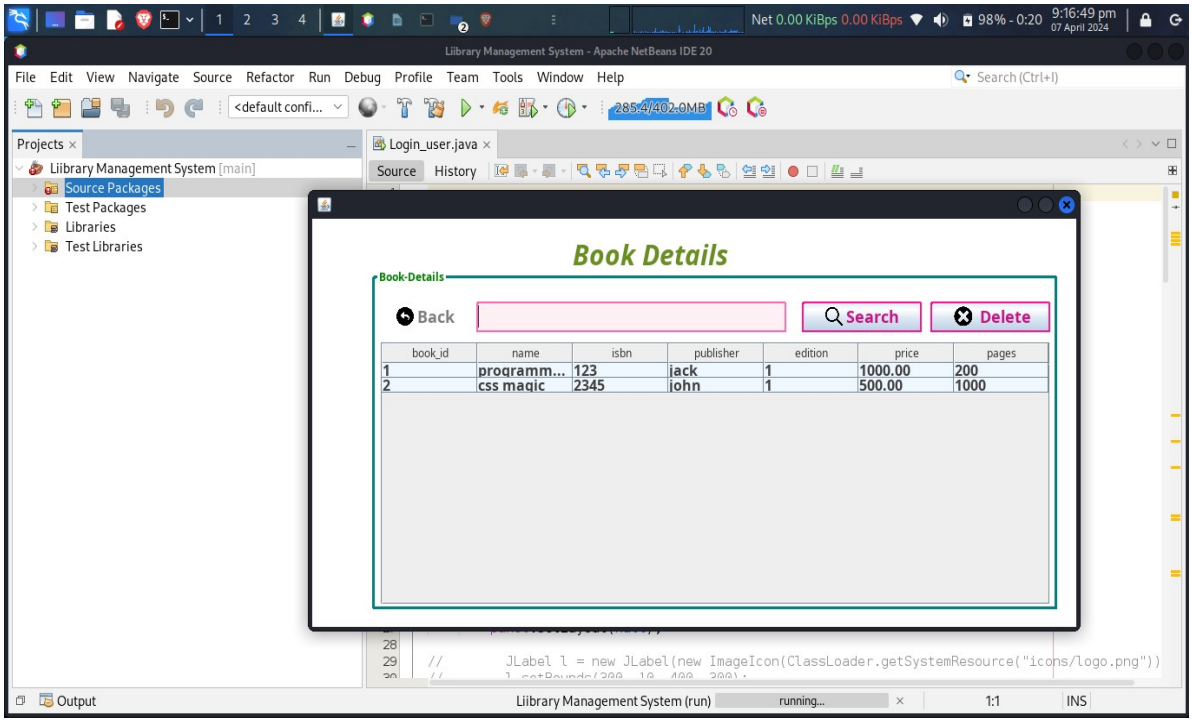


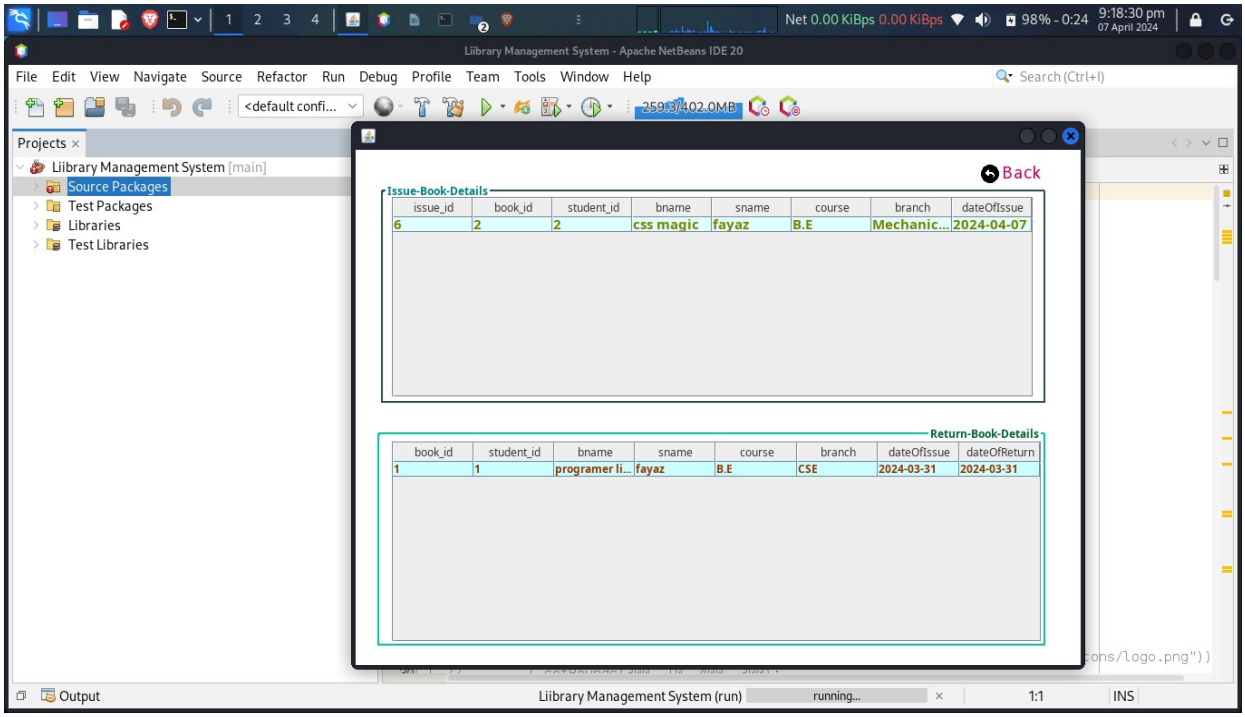



Liibrary Management System - Apache NetBeans IDE 20
Add Student
Add-Student
Student_id
2
Name
fayaz
Father's Na...
kathar
Course
B.E
Branch
Mechanical
Year
Second
Semester
4th
ADD
Back
Message
Successfully Added
OK


Liibrary Management System - Apache NetBeans IDE 20
Add Books
Book_id
2
Name
css magic
ISBN
2345
Publisher
john
Edition
1
Price
500
Pages
1000
Add
Back
Message
Successfully Added
OK


Liibrary Management System - Apache NetBeans IDE 20
Student Details
Student-Deatails
Back
Search
Delete
student_id | name | father | course | branch | year | semester
1 | asif | hasan | B.E | CSE | First | 1st
2 | fayaz | kathar | B.E | Mechanical | Second | 4th


Liibrary Management System - Apache NetBeans IDE 20
Book Details
Book-Details
Back
Search
Delete
book_id | name | isbn | publisher | edition | price | pages
1 | programm... | 123 | jack | 1 | 1000.00 | 200
2 | css magic | 2345 | john | 1 | 500.00 | 1000


Liibrary Management System - Apache NetBeans IDE 20
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Liibrary Management System [main]
Source Packages
Test Packages
Libraries
Test Libraries
Back
Issue-Book-Details
issue_id book_id student_id bname sname course branch dateOfIssue
6 2 2 css magic fayaz B.E Mechanic... 2024-04-07
Return-Book-Details
book_id student_id bname sname course branch dateOfIssue dateOfReturn
1 1 programer li... fayaz B.E CSE 2024-03-31 2024-03-31
Output
Liibrary Management System (run)
running...
1:1
INS

CHAPTER 5
SYSTEM TESTING

5.1 UNIT TESTING:

Unit testing is a crucial phase in the software development life cycle, focusing on testing individual units or components of the library management system (LMS) to ensure their functionality, reliability, and correctness. Each unit is tested in isolation, verifying that it performs as expected and meets the specified requirements. The unit testing process typically involves the following steps:

1. Test Case Design:

Test cases are designed to cover various scenarios and functionalities of each unit. Test cases should encompass typical use cases as well as edge cases and boundary conditions to ensure thorough testing.

2. Test Environment Setup:

A dedicated test environment is set up to execute unit tests. This environment may include mock objects, stubs, and other testing tools to simulate the behavior of dependencies and external systems.

3. Test Execution:

The unit tests are executed, and the actual outcomes are compared against expected results. Test results are logged, and any discrepancies or failures are identified for further analysis.

4. Test Result Analysis:

Test results are analyzed to determine the root causes of failures or deviations from expected behavior. Developers debug and troubleshoot issues to rectify defects and improve unit performance.

5. Regression Testing:

- Regression tests are conducted to ensure that modifications or enhancements to the system do not introduce new defects or regressions in existing functionality. Existing unit tests are rerun to validate the integrity of the system after changes are made.

6. Code Coverage Analysis:

- Code coverage metrics are used to assess the effectiveness of unit testing. Developers measure the percentage of code paths covered by unit tests, aiming for high code coverage to minimize the risk of undiscovered defects.

7. Documentation:

- Comprehensive documentation of unit tests is maintained, including test cases, test results, and any relevant observations or insights. Documentation serves as a reference for future development efforts and system maintenance.

By conducting thorough unit testing, developers can identify and address defects early in the development process, leading to higher software quality, improved reliability, and reduced time and cost of software maintenance. Unit testing also fosters code modularity, reusability, and maintainability, contributing to the overall success of the LMS project.

5.2 INTEGRATION TESTING:

Integration testing is a critical phase in the software development lifecycle, focusing on testing the interactions and interfaces between different modules or components of the library management system (LMS). The primary objective of integration testing is to ensure that individual modules work together as intended, effectively exchanging data and communicating with each other. The integration testing process typically involves the following steps:

1. Integration Strategy Selection:

Developers choose an integration strategy based on the system architecture and design. Common integration strategies include top-down, bottom-up, and incremental integration, each with its own advantages and trade-offs.

2. Test Environment Setup:

A dedicated test environment is established to facilitate integration testing. This environment closely resembles the production environment and includes all necessary hardware, software, and network configurations.

3. Integration Test Planning:

- Integration test cases are designed based on system requirements, use cases, and integration points between modules. Test scenarios cover various integration paths, data flows, and error handling mechanisms.

4. Component Integration:

- Modules are integrated incrementally according to the chosen integration strategy. Integration points are identified, and interfaces between modules are tested to ensure compatibility and interoperability.

5. Data Exchange Testing:

- Integration tests verify the accuracy and integrity of data exchanged between modules. Input data is passed through the system, and output data is validated against expected results to detect any data corruption or loss.

6. Interface Validation:

- Interfaces between modules are rigorously tested to ensure proper data transmission and error handling. This includes testing data formats, protocols, message structures, and error codes defined in the system interface specifications.

7. Error Handling and Recovery:

- Integration tests assess the system's ability to handle errors, exceptions, and unexpected scenarios gracefully. Error recovery mechanisms are tested to ensure that the system can recover from failures and maintain data consistency.

8. Integration Test Execution:

Integration test cases are executed systematically, and test results are recorded. Any failures or discrepancies are documented, and developers investigate the root causes of integration issues for resolution.

9. Regression Testing:

Regression tests are conducted to validate the stability of integrated modules after changes or fixes are applied. Existing integration tests are rerun to verify that modifications do not introduce new integration defects.

Integration testing ensures that the library management system functions seamlessly as a unified whole, meeting performance, reliability, and scalability requirements. By identifying and resolving integration issues early in the development process, integration testing contributes to the overall quality and success of the LMS project.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

In conclusion, the development and implementation of the library management system (LMS) represent a significant milestone in improving the efficiency, effectiveness, and user experience within library environments. Through the analysis, design, and implementation phases, key objectives of the project have been achieved, including streamlining book management processes, enhancing user accessibility, and ensuring data integrity.

The LMS project has addressed several challenges faced by traditional manual systems, such as time-consuming book tracking, limited search capabilities, and cumbersome administrative tasks. By leveraging modern technologies and best practices in software engineering, the system offers a user-friendly interface, robust functionality, and scalability to accommodate future growth and expansion.

Looking ahead, there are several avenues for future enhancement and refinement of the library management system. Some potential areas of focus include:

1. Enhanced Reporting and Analytics: Integrating advanced reporting and analytics capabilities into the LMS can provide valuable insights into library usage patterns, popular book genres, and student borrowing behaviors. This data-driven approach can inform collection development strategies and resource allocation decisions.

2. Mobile Application Development: Developing a mobile application for the LMS can extend its reach and accessibility to users beyond the confines of the library. A mobile app would enable students to search for books, place holds, and receive notifications on their smartphones, enhancing convenience and user engagement.

3. Integration with Digital Libraries: Integrating the LMS with digital libraries and online repositories can broaden access to digital resources, e-books, and multimedia content. Seamless integration would allow users to access and borrow digital materials directly through the LMS interface, promoting digital literacy and lifelong learning.

4. User Feedback and Iterative Improvement: Soliciting feedback from library staff and patrons and incorporating their suggestions into future iterations of the LMS is essential for continuous improvement. Regular

usability testing, surveys, and focus groups can identify usability issues, pain points, and feature requests that can guide system enhancements.

5. Accessibility and Inclusivity: Ensuring that the LMS complies with accessibility standards and guidelines, such as the Web Content Accessibility Guidelines (WCAG), is essential for providing an inclusive user experience. Improving support for assistive technologies and accommodating diverse user needs can enhance accessibility for all users, including those with disabilities.

In conclusion, the library management system project has laid a strong foundation for modernizing library operations and enhancing user services. By embracing innovation and collaboration, the LMS has the potential to become a central hub for knowledge dissemination, collaboration, and lifelong learning in academic and community settings. Through ongoing development and iteration, the LMS will continue to evolve to meet the changing needs of libraries and their patrons in the digital age.

CHAPTER 7

REFERENCES

1. Somerville, Ian. "Software Engineering." Pearson Education, 2015.

2. Pressman, Roger S. "Software Engineering: A Practitioner's Approach." McGraw-Hill Education, 2014.

3. Ambler, Scott W. "The Object Primer: Agile Model-Driven Development with UML 2.0." Cambridge University Press, 2004.

4. Fowler, Martin. "UML Distilled: A Brief Guide to the Standard Object Modeling Language." Addison-Wesley Professional, 2003.

5. Oracle. "Java SE Documentation." Available online: https://docs.oracle.com/en/java/javase/index.html

6. Oracle. "Java Swing Documentation." Available online: https://docs.oracle.com/javase/tutorial/uiswing/index.html

7. MySQL Documentation. "MySQL 8.0 Reference Manual." Available online: https://dev.mysql.com/doc/refman/8.0/en/

8. W3Schools. "SQL Tutorial." Available online: https://www.w3schools.com/sql/