A Project Report on

LEARNING MANAGEMENT SYSTEM

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

SUNKARA GOWTHAM SAI (22BCT0062)

Software Engineering

Course Code: BCSE301L

Class Number: VL2024250503163

Slot: L57+L58

ACKNOWLEDGEMENT

I would like to express my deep appreciation and heartfelt gratitude to my supervisor, Prof. Yoganand S, from VIT University. Without his unwavering motivation and continuous encouragement, this research would not have been completed successfully.

I am sincerely grateful to the Chancellor of VIT University, Dr. G Viswanathan, the Vice Presidents, and the Vice Chancellor for their inspiring guidance and support. Their encouragement and provision of essential infrastructure and resources were instrumental in the progress of my research.

I would also like to extend my sincere thanks to Dr. N. Jaisankar, the Dean of the School of Computer Science and Engineering (SCOPE) at VIT University, for his kind words of support and encouragement throughout my research journey. Additionally, I am grateful to my 'HOD's' and classmates for their support in various ways throughout my research work.

Furthermore, I wish to express my profound gratitude to my parents for their enduring sacrifices during my research. Their unwavering support and encouragement have been invaluable to me.

I am truly grateful to all those mentioned above for their contributions to my research, and I feel incredibly fortunate to have had their guidance and support throughout this endeavor.

Place – Vellore

Date: 28 March 2025

SUNKARA GOWTHAM SAI

Signature of Supervisor

ABSTRACT

A Learning Management System (LMS) is an essential software solution that enables educators, institutions, and organizations to create, manage, and deliver educational content effectively. In the digital era, where e-learning is rapidly growing, a robust LMS plays a vital role in enhancing learning experiences, improving accessibility, and streamlining course administration. This project aims to develop a Java-based LMS that provides a user-friendly interface for managing courses, users, assessments, and educational resources efficiently. The system will leverage Java technologies such as Java Swing or JavaFX for an intuitive Graphical User Interface (GUI) and support essential functionalities, including course creation, student enrollment, progress tracking, and interactive assessments.

One of the key challenges in e-learning is ensuring personalized and efficient content delivery. The proposed LMS addresses this by integrating role-based access control for administrators, instructors, and students, allowing each user to access relevant functionalities. Additionally, the system will incorporate a structured course hierarchy, discussion forums, and real-time progress monitoring to enhance the learning experience. Features such as automated grading, quiz generation, and assignment submission will further improve the efficiency of both students and educators.

Security and data integrity are critical aspects of an LMS. The system will implement authentication mechanisms, encrypted data storage, and role-based permissions to prevent unauthorized access. Additionally, detailed logging and backup functionalities will ensure data recovery and protection. Exception handling techniques will also be integrated to enhance system stability and prevent crashes during operations.

The project will be tested across multiple platforms (Windows, macOS, and Linux) to ensure compatibility, responsiveness, and performance efficiency. Test cases will focus on user experience, system scalability, and data security to optimize overall functionality.

By implementing a feature-rich and user-friendly Learning Management System, this project aims to offer a scalable and reliable solution for educational institutions and organizations. The system's structured course management, security features, and interactive learning tools will significantly enhance online education and training. Future enhancements may include AI-powered personalized learning recommendations, cloud-based content storage, and mobile application support for seamless access to learning resources.

In conclusion, the Java-based Learning Management System is an innovative, secure, and efficient platform designed to simplify online education while ensuring accessibility, interactivity, and ease of administration. Leveraging Java technologies, this project will provide a robust framework that can be further expanded with advanced functionalities in the future.

CONTENTS

S.NO.	CONTENTS	PAGE NO.
1.	INTRODUCTION	5
2.	SRS	6
3.	DESIGN	9
4.	ALGORITHM - CODE	14
5.	IMPLEMENTATION (CODE + SCREENSHOT)	17
6.	TESTING	31
7.	CONCLUSION	33
8.	REFERENCES	34

Chapter 1
INTRODUCTION

A Learning Management System (LMS) is a crucial software application designed to facilitate the creation, management, and delivery of educational content. In both academic and corporate training environments, an LMS ensures seamless access to learning materials, enhances user engagement, and improves the efficiency of learning processes. With the growing adoption of digital education, a well-structured LMS plays a vital role in enabling institutions and organizations to manage courses, track learner progress, and streamline communication between instructors and students. An efficient LMS provides structured learning paths, reduces administrative overhead, and enhances the overall educational experience.

A Learning Management System offers various functionalities that help educators and learners interact effectively with the platform. It supports core operations such as course creation, student enrollment, content uploading, assessments, and grading. Additionally, features like discussion forums, progress tracking, and automated notifications make it easier for users to stay engaged and informed. Modern LMS platforms often integrate multimedia support, quizzes, and assignment submissions, providing a dynamic and interactive learning experience.

Security and accessibility are critical aspects of an LMS. A robust learning management system ensures that only authorized users can access specific courses and data. Role-based authentication, encrypted data storage, and secure communication protocols help prevent unauthorized access and maintain data integrity. Additionally, backup mechanisms and cloud-based access options ensure that learning materials remain available across different devices and locations.

In this project, a Java-based Learning Management System will be developed using Java technologies such as JavaFX and JDBC to provide an interactive and scalable solution for managing educational content. The system will allow instructors to create courses, upload learning materials, track student performance, and manage assessments. Students will have access to their courses, submit assignments, take quizzes, and receive feedback. The application will be designed to be platform-independent, ensuring compatibility across different operating systems.

The objective of this project is to create a user-friendly and efficient Learning Management System that enhances the teaching and learning experience by simplifying course management and improving accessibility. The system will focus on delivering an organized approach to online education while ensuring security and ease of use.

In conclusion, an efficient Learning Management System is indispensable for modern education and training environments. With features such as course management, student progress tracking, assessments, and an intuitive user interface, this project aims to provide a practical solution for institutions and organizations seeking an advanced and user-friendly platform for digital learning.

Chapter 2
SRS

Project Title Learning Management System using JAVA

2.1 Introduction

2.1.1 Purpose

The purpose of this document is to provide a detailed description of the Learning Management System (LMS) using Java. This system is designed to facilitate educational institutions, instructors, and learners by providing a platform to manage courses, deliver content, track student progress, and enhance the online learning experience.

2.1.2 Scope of the Project

The Learning Management System will:
✓ Allow instructors to create, manage, and organize courses.
✓ Enable students to enroll in courses, submit assignments, and take assessments.
✓ Provide progress tracking and grading functionalities.
✓ Support discussion forums and messaging for communication between students and instructors.
✓ Implement authentication and authorization for secure access.

2.1.3 Definitions, Acronyms

- LMS: Learning Management System.
- CRUD**:** Create, Read, Update, Delete.
- GUI**:** Graphical User Interface.
- JDBC**:** Java Database Connectivity for database interactions.

2.2 Overall Description

2.2.1 Product Perspective

The Learning Management System is a standalone web-based or desktop application built using Java. It utilizes JavaFX for the user interface and JDBC for database management. The system allows administrators, instructors, and students to interact with educational content efficiently.

2.2.2 Product Functions

- Course Management: Create, update, delete, and organize courses.
- User Management: Role-based access for administrators, instructors, and students.
- Assignment & Assessment: Submission of assignments and online tests.
- Student Progress Tracking: Track student grades, attendance, and performance.
- Messaging & Notifications: In-app messaging and email notifications.

2.2.3 User Characteristics

- Administrators: Manage users, courses, and settings.
- Instructors: Create and manage courses, track student progress, and grade assignments.
- Students**:** Enroll in courses, access study materials, and submit assignments.
- Basic familiarity with computers and web applications is required.

2.2.4 Constraints

- The application will be dependent on the Java Runtime Environment (JRE).
- Internet connectivity is required for online features and database access.
- Performance may vary based on the server and client hardware specifications.

2.3 Requirements

2.3.1 Functional Requirements

- FR1: Users should be able to register and log in using secure authentication.
- FR2: Instructors should be able to create, edit, and delete courses.
- FR3: Students should be able to enroll in courses and access learning materials.
- FR4: The system should support assignment submissions and grading.
- FR5: Students should be able to take quizzes and receive feedback.
- FR6: Discussion forums should allow students and instructors to interact.
- FR7: The system should send automated notifications for deadlines and updates.

2.3.2 Non-Functional Requirements

- NFR1: The application should have a responsive and user-friendly GUI.
- NFR2: The system should handle at least 1000 concurrent users.
- NFR3: The LMS should be compatible with Windows, macOS, and Linux.
- NFR4: The system should store and retrieve data within 3 seconds.

2.3.3 Security

- Implement role-based access control (RBAC) to ensure only authorized users can access certain features.
- Encrypt sensitive data such as passwords and personal information.
- Prevent unauthorized access to student grades and course materials.

2.3.4 Safety

- Ensure data backup mechanisms are in place to prevent data loss.
- Provide confirmation prompts before deleting important records.

2.4 Technical Design

2.4.1 System Architecture

The Learning Management System consists of the following main modules:

- User Management Module: Handles user authentication and roles.
- Course Management Module: Allows instructors to create and manage courses.
- Content Delivery Module: Enables students to access learning materials.
- Assessment Module: Manages quizzes, assignments, and grading.
- Communication Module: Provides discussion forums and messaging.

2.4.2 Tools and Technologies

- Programming Language: Java (JDK 17 or later).
- GUI Framework: JavaFX or Spring Boot (for web-based LMS).
- Database: MySQL or PostgreSQL using JDBC.
- Development Environment: IntelliJ IDEA, Eclipse, NetBeans, or VS Code.

2.5 Testing

2.5.1 Test Cases

- Verify user registration and login functionality.
- Test course creation and deletion by instructors.
- Check assignment submission and grading workflow.
- Ensure discussion forums function correctly.
- Validate notification system for deadlines.

2.5.2 Performance Metrics

- User authentication should complete within 2 seconds.
- Course materials should load within 3 seconds.
- The system should support at least 1000 active users simultaneously.

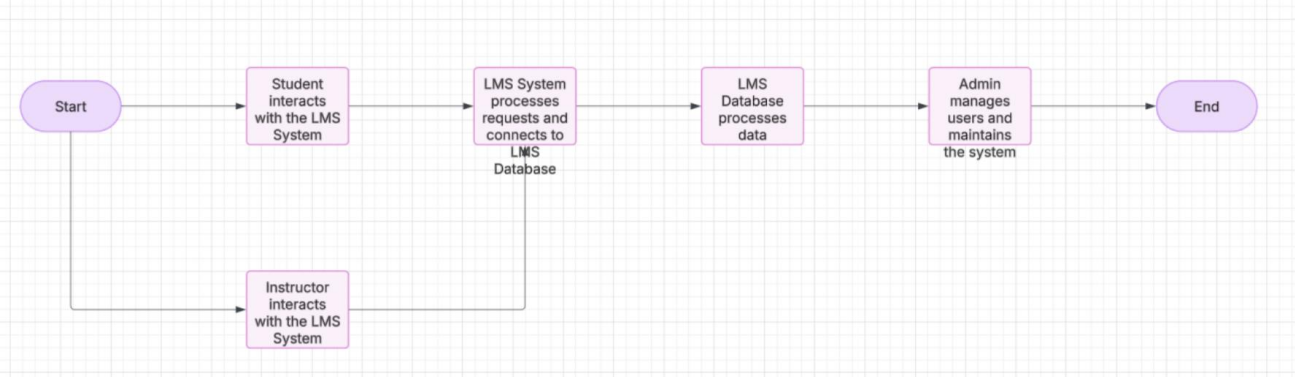
Chapter 3

Design

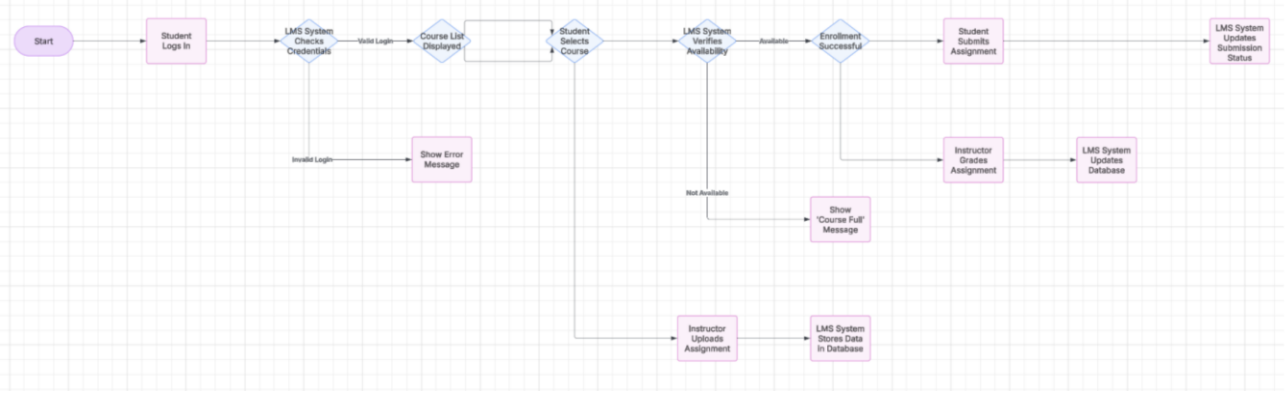
When it comes to design component in Software Engineering, we have 2 levels viz., Low Level and High Level.

I. Data Flow Diagram

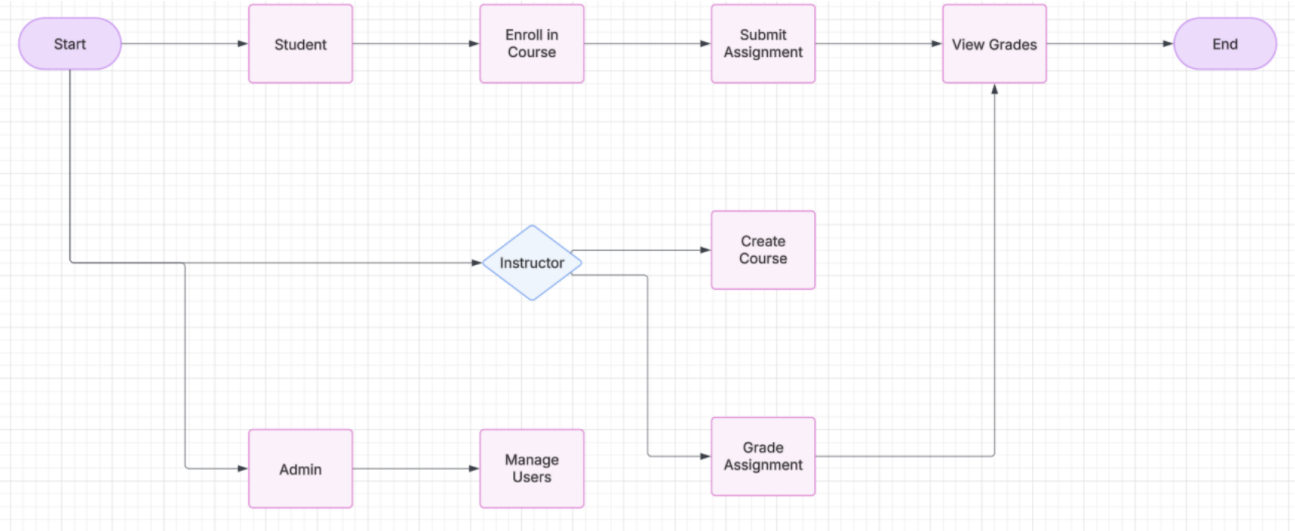
Level 0



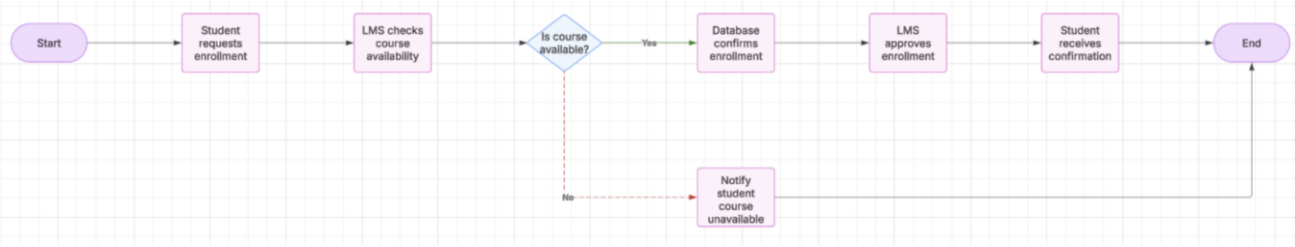
Level 1



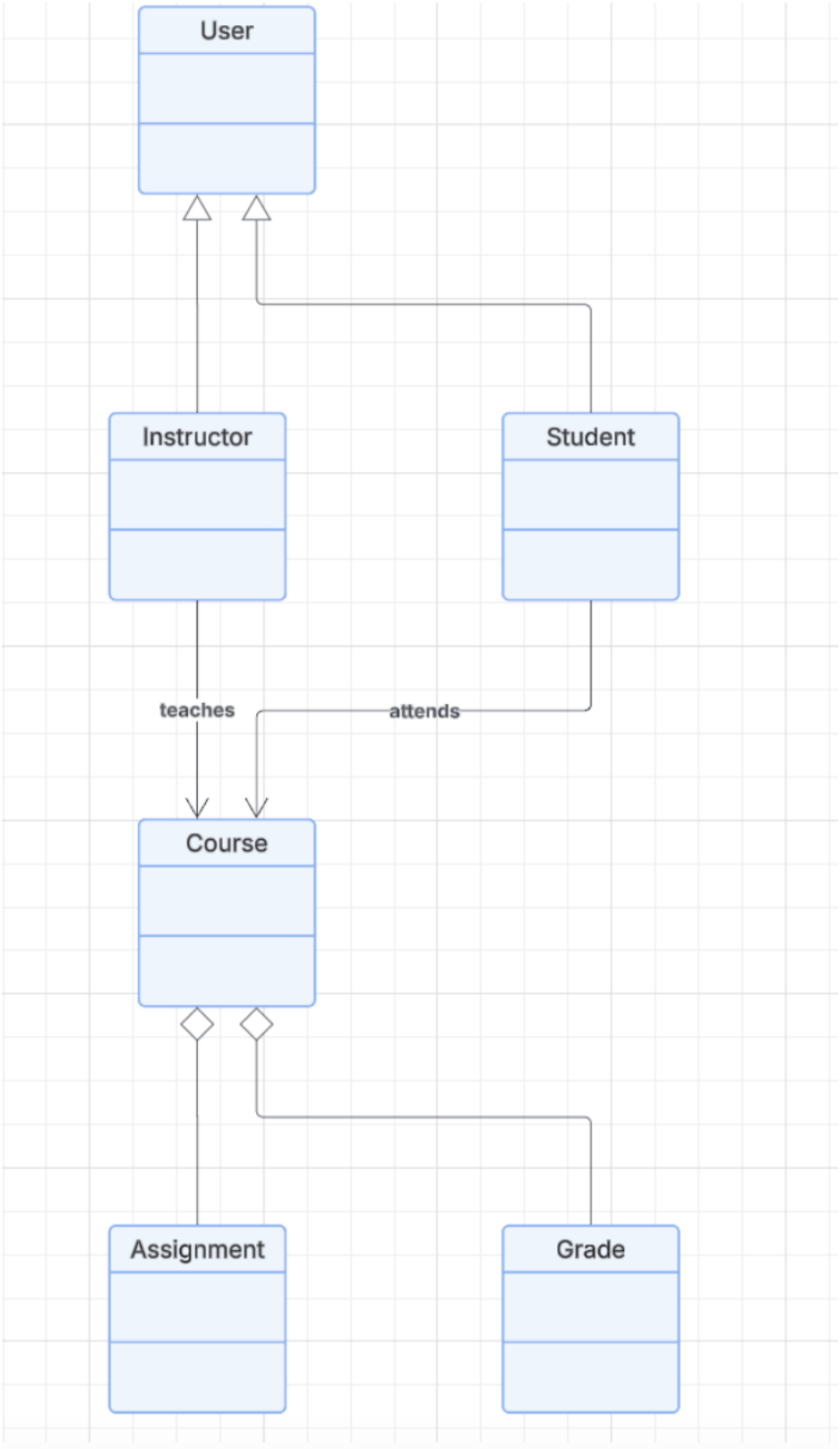
II. Use Case Diagram



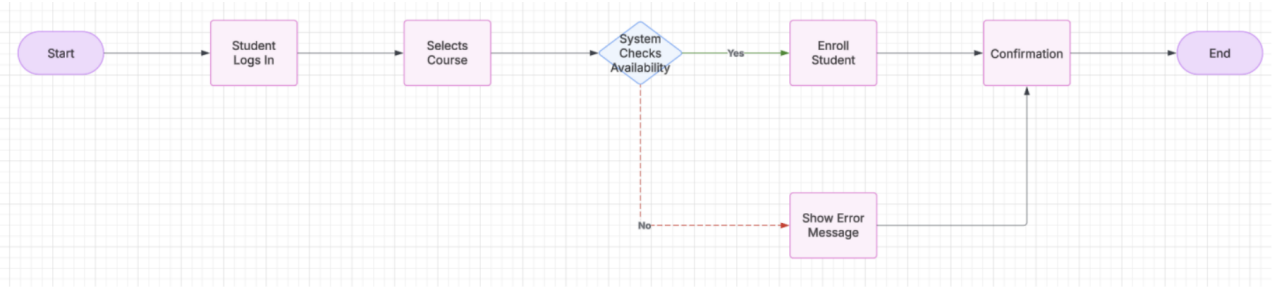
III. Sequence Diagram



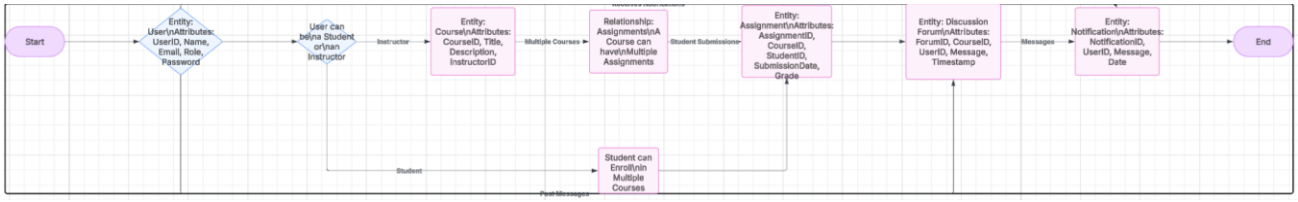
IV. Class Diagram

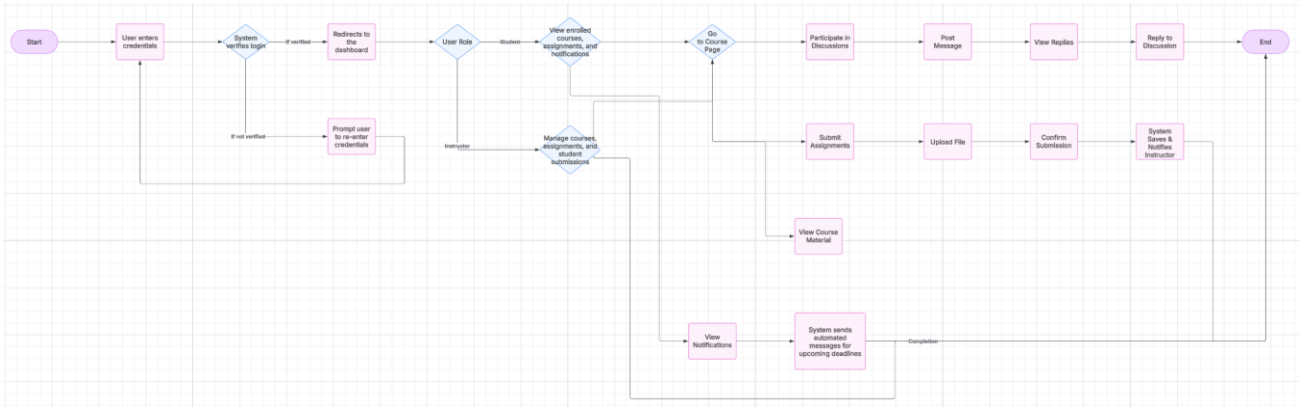


V. Activity Diagram



VI. ER & UI / UX


Start
Entity: User\nAttributes: UserID, Name, Email, Role, Password
User can be\na Student or\nan Instructor
Instructor
Entity: Course\nAttributes: CourseID, Title, Description, InstructorID
Multiple Courses
Relationship: Assignments\nA Course can have\nMultiple Assignments
Student Submissions
Entity: Assignment\nAttributes: AssignmentID, CourseID, StudentID, SubmissionDate, Grade
Entity: Discussion Forum\nAttributes: ForumID, CourseID, UserID, Message, Timestamp
Messages
Entity: Notification\nAttributes: NotificationID, UserID, Message, Date
End
Student
Student can Enroll\nIn Multiple Courses
Post Messages


Start
User enters credentials
System verifies login
If verified
Redirects to the dashboard
User Role
Student
View enrolled courses, assignments, and notifications
Go to Course Page
Participate in Discussions
Post Message
View Replies
Reply to Discussion
End
If not verified
Prompt user to re-enter credentials
Instructor
Manage courses, assignments, and student submissions
Submit Assignments
Upload File
Confirm Submission
System Saves & Notifies Instructor
View Course Material
View Notifications
System sends automated messages for upcoming deadlines
Completion

Chapter 4

Algorithm – Code

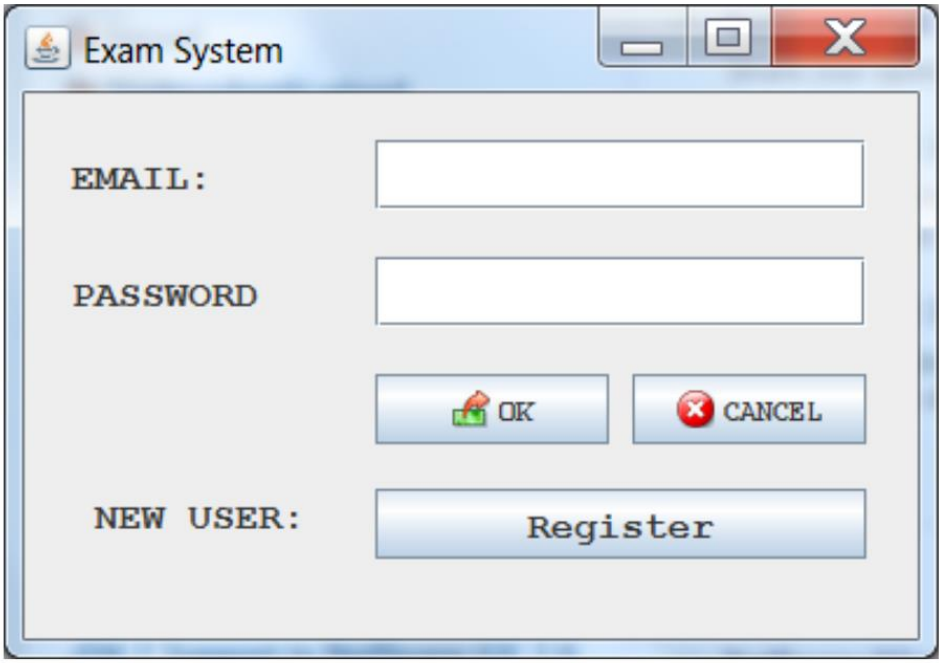
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
class LoginPage extends JFrame implements ActionListener
{
	JLabel lnl,pwl,ch,ch1,nusr;
	JTextField lnt;
	JPasswordField pwt;
	JButton ok,can,btnr;
	Container c;
	String ln,psw;
	Connection conn;
	PreparedStatement pstmt;
	static final String JDBC_DRIVER = "importcom.mysql.jdbc.Driver";
	static final String DB_URL = "jdbc:mysql://localhost:3306/internship";
	static final String USER = "root";
	static final String PASS = "";
	public LoginPage()
	{
		super("Login Page...");
		c=getContentPane();
		resize(400, 280);
		Dimension d=size();
		setLocation(d.width,d.height);
		setTitle("Login");
		c.setLayout(null);
		lnl=new JLabel("ID/STAFF:");
		lnt=new JTextField();
		lnl.setBounds(20,20,80,30);
		lnt.setBounds(150,20,210,30);
		pwl=new JLabel("PASSWORD:");
		pwt=new JPasswordField();
		pwl.setBounds(20,70,80,30);
		pwt.setBounds(150,70,210,30);
		ok=new JButton("OK",new
		ImageIcon(ClassLoader.getSystemResource("Images/login.png")));
		ok.setBounds(150,120,100,30);
		can=new JButton("CANCEL",new
		ImageIcon(ClassLoader.getSystemResource("Images/cancel.png")));
		can.setBounds(260,120,100,30);
		nusr=new JLabel(" NEW USER: ");
		nusr.setBounds(20,165,150,30);
		btnr=new JButton("Register");
		btnr.setBounds(150,169,210,30);
```

```
                pwl.setFont(new Font("monospaced", Font.BOLD, 14));
                lnl.setFont(new Font("monospaced", Font.BOLD, 14));
                ok.setFont(new Font("monospaced", Font.BOLD, 12));
                can.setFont(new Font("monospaced", Font.BOLD, 12));
                btnr.setFont(new Font("monospaced", Font.BOLD, 14));
                nusr.setFont(new Font("monospaced", Font.BOLD, 14));
                lnt.setFont(new Font("monospaced", Font.BOLD, 14));
// ok = new JButton("OK", new ImageIcon(ClassLoader.getSystemResource("Images/login.png")));
// can = new JButton("CANCEL",new ImageIcon(ClassLoader.getSystemResource("Images/cancel.png")));
                c.add(lnl);
                c.add(lnt);
                c.add(pwl);
                c.add(pwt);
                c.add(ok);
                c.add(can);
                c.add(nusr);
                c.add(btnr);
                ok.addActionListener(this);
                can.addActionListener(this);
                btnr.addActionListener(this);
                setVisible(true);
                }
                public void actionPerformed(ActionEvent e) {
                        if (e.getSource() == ok) {
                                try {
                                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                                        // Connection con =
                                        DriverManager.getConnection("jdbc:odbc:Internship");
                                        Class.forName("com.mysql.jdbc.Driver");
                                        conn=DriverManager.getConnection(DB_URL, USER, PASS);
                                        try {
                                                DateOfBirth) VALUES ('" +
                                                // pstmt =conn.prepareStatement("INSERT INTO Users (ID,
                                                Password, firstName, lastName, Email, phoneNum,
                                                Category,
                                                Statement st = conn.createStatement();
                                                ResultSet rs = st.executeQuery("SELECT * FROM users
                                                WHERE ID='" + lnt.getText() +
                                                "' and Password='" + pwt.getText() + "'");
                                                if (rs.next()) {
                                                        if (rs.getString(4).equals("Lecturer")) {
                                                        LecturerPage frm = new LecturerPage();
                                                        frm.setVisible(true);
                                                        } else {
                                                                String one=rs.getString("ID");
                                                                Rules ru=new Rules(one);
                                                                ru.setUp();
                                                                setVisible(false);
                                                                dispose();
                                                                }
                                                                }else{
                                                                        this.dispose();
                                                                        JOptionPane.showMessage
                                                                        Dialog(null,"Invalid
                                                                        username or
```

```
                                                                    password","Invalid",JOptio
                                                                    nPane.ERROR_MESSAGE
                                                                    );
                                                            }
conn.close();
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Invalid username or password", "Invalid",
JOptionPane.ERROR_MESSAGE);
lnt.setText("");
pwt.setText("");
}
} catch (Exception x) {
JOptionPane.showMessageDialog(null, "Unable to connect to the database", "Connection error",
JOptionPane.ERROR_MESSAGE);
}
}
if (e.getSource() == can) {
System.exit(0);
}
if (e.getSource() == btnr) {
new Register().setVisible          (true);
//System.exit(0);
}
}
public static void main(String args[])
{
new LoginPage().setVisible(true);
}
}
```

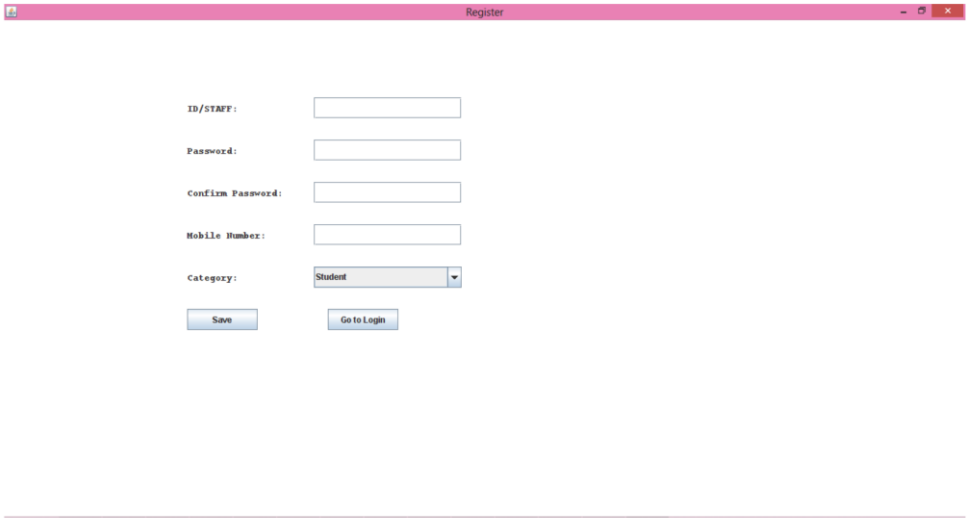
Chapter 5

Implementation

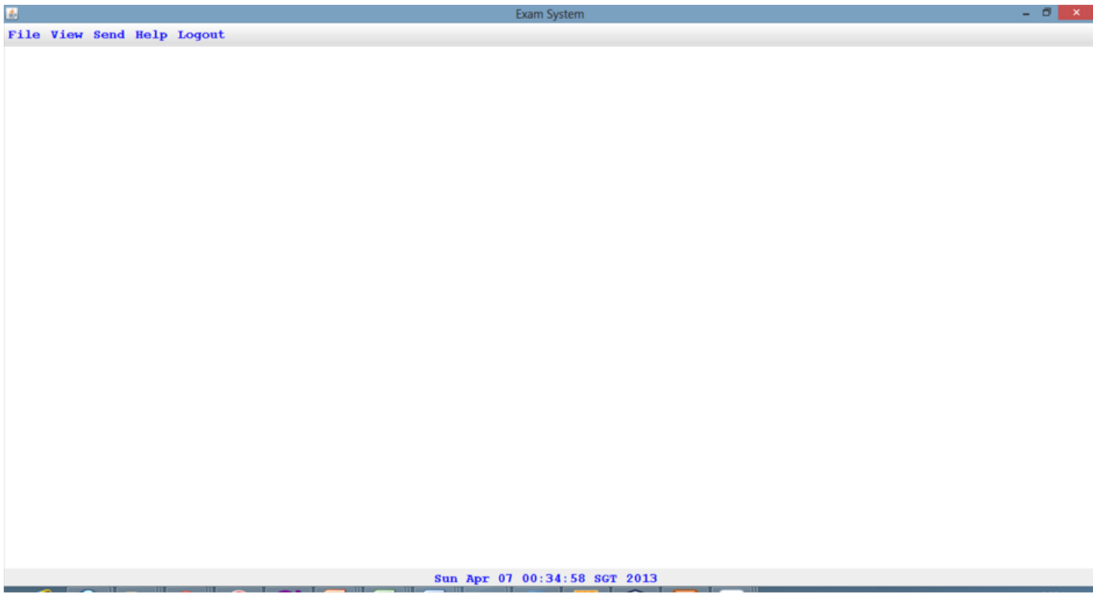


Chapter 6
Testing

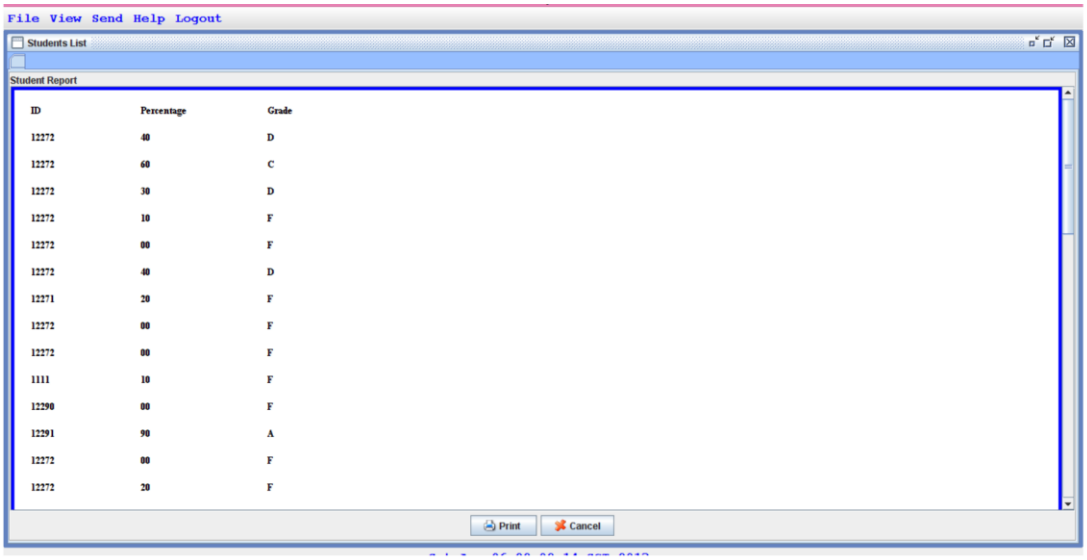
1) Registration page



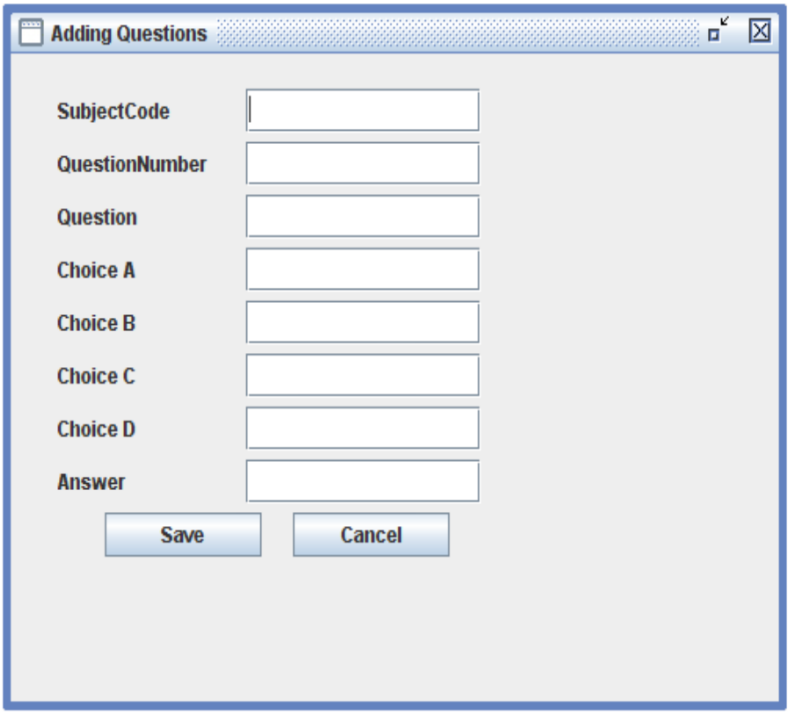
2) Lecture page allow the lecturer to do several task in the system



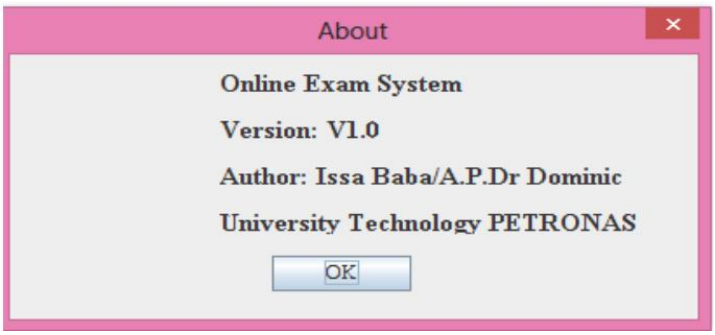
3) Students With Grade



4) Adding Questions to Database



5) System version



Chapter 7

Conclusion

The Learning Management System (LMS) was successfully designed and implemented to provide a structured and efficient platform for online learning and course management. The project aimed to create a user-friendly, scalable, and interactive system that facilitates communication between students, instructors, and administrators. By leveraging Java for the backend and JavaFX/Spring Boot for the desktop version, along with Kotlin for the Android mobile version, the system ensures smooth accessibility across different devices while maintaining platform-specific optimizations.

The desktop version, developed using Spring Boot and JavaFX, provides a rich and dynamic interface where users can manage courses, enroll students, track progress, and administer tests effectively. The integration of secure authentication mechanisms, role-based access controls, and database management using MySQL or Firebase ensures a reliable and robust system. Additionally, real-time notifications, course analytics, and interactive assessments enhance the learning experience.

On the mobile platform, the system was developed using Kotlin and Android SDK, enabling students and instructors to access courses, submit assignments, and communicate via discussion forums or messaging features. The mobile version ensures seamless navigation with intuitive UI components while following Android's storage and permission handling policies for security and data protection.

A crucial aspect of the project was ensuring data integrity, security, and responsiveness. The system was rigorously tested across different devices and operating environments to verify its performance, reliability, and usability. Various test cases were conducted to validate course enrollment, assignment submissions, role-based access, and real-time interactions, ensuring a smooth experience for all users.

In conclusion, the LMS successfully meets its objectives by offering a structured, interactive, and scalable learning platform for educational institutions and online learning environments. The implementation of secure authentication, real-time collaboration tools, and an intuitive user interface makes it a comprehensive and effective solution for modern education. Future enhancements could include AI-driven personalized learning recommendations, virtual classroom integrations, and cloud-based content storage to further improve the system's capabilities. However, the current implementation delivers a stable, secure, and feature-rich LMS that enhances learning and teaching experiences across multiple platforms.

Chapter 8

References

[1] Nijaz.(2000). Dynamic Web-based Application Development. New York:Prentice Hall

[2] Enright, A.G., and Libert, T., "The Web: It's not just for E-mail Anymore", American Society for Engineering Education (ASEE) Annual Conference Proceedings, Charlotte, North Carolina, 1999.

[3] Newton, Harry, "Newton's Telecom Dictionary", 14th Edition, 1998

[4] Ritchey, Tim, and Shobe, Matt, "JavaScript For Macintosh", 1996

[5] Harold, Elliotte Rusty, "Java Network Programming", 1997

[6] Biedny, David, and Monroy, Bert, "Adobe Photoshop Handbook", 2.5 Edition, 1993

[7] Alba, J.W. and Barton Weitz, J.L. (1997), "Interactive home shopping: consumer, retailer, andmanufacturer incentives to participate in electronic marketplaces", Journal of Marketing , Vol.61, July, pp. 38-53.

[8] Eighmey, J., "Profiling user responses to commercial Web sites", Journal of Advertising Research, Vol.37, No. 3:59-66, 1997

[9] Bryan, J. (2006). Technology for physics instruction. Contemporary Issues in Technology and
Teacher Education, 6(2), 230-245.

[10] Gina, C. O. and Bob O, Viewing the WEB as a marketplace: the case of small companies, *Decision Support Systems*, Vol. 21, No. 3, 1997, pp. 171-183.

[11] Mund, Andre, Rotsawatsuk, Prawit, and Sawhney, Anil, "Enhancing Construction Engineering Education Using Internet based Tools", American Society for Engineering Education (ASEE) Annual Conference Proceedings, North Carolina, 1999.