

NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY
B.E / BS PROGRAMME OF USMAN INSTITUTE OF TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT

Spring 2024

**SE-312 Software Construction and
Development**

Project Summary Report

Semester: 6th

○ **Project Title: University Management System**

○ **Team Members:**

Name	Responsibility/Task Assigned
Hadiqa Mehdi	Backend Development
Musfira Khalid	Front end Development
Saad Ather Ali	Front end Development

PROJECT DESCRIPTION:

University Management System is a versatile, and cost-effective solution for universities, colleges, and schools. It's a whole end-to-end system that takes care of every detail of a university workflow, which is exactly what we needed.

Purpose and Significance:

Our University Management System aims to design and enhance the efficiency of managing university operations. The system facilitates the management of student, teacher information, academic records, fee details, and other administrative tasks. The UMS will reduce the dependence on manual paperwork, minimize errors, and improve accessibility and security. The system is significant as it will contribute to the better resource management, enhance communication among stakeholders, and provide a platform that is to be centralized

Problem Statement:

Universities often face challenges in managing data related to students, teachers, academic records, and administrative tasks. The dependence on manual processes leads to inefficiencies, errors, and delays in data processing. There is a need for a comprehensive system that can handle various aspects of university management, provide real-time data access, and administrative workflows. UMS is developed to monitor, and analyze complicated activities such as student admission, examinations, and much more. It is a program that manages the complete student life cycle up to degree completion.

Aims:

- **Centralized Data Management:** Efficiently add, update, and access information about students, teachers, and their academic records.
- **Automated Workflows:** Streamline administrative tasks like admissions, leave requests, fees, exams, and grading.
- **Enhanced Decision-Making:** Provide real-time data access for informed choices by administrators, faculty, and students.
- **Improved Collaboration:** Foster communication through features designed for teachers, students, and staff to connect.
- **Role-Based Security:** Establish secure access with different user roles (admin, teacher, student) having specific permissions.

Objectives:

- **Data Management:** Develop the system to efficiently add, update, and view information about students and teachers.
- **Leave Management:** Implement a module for handling leave requests and permissions for students and teachers.
- **Exam Management:** Provide the functionalities for managing exam schedules, details, and results.
- **Fee Management:** Create a feature for tracking and managing student fees details and payments.
- **User Roles and Permissions:** Establish different user's roles (e.g., admin, teacher, student) with specific permissions and access levels.

Scope:

- **Student Information System:** Adding, updating, and viewing student details, academic records
- **Teacher Information System:** Managing teacher profiles, scheduling, and academic contributions.
- **Leave Management System:** Handling leave applications and approvals
- **Exam Management System:** Managing exam details, and publishing results.
- **Fee Management System:** Manage fees records, and fees details.
- **User Authentication and Authorization:** Ensuring secure access with role-based permissions.

Constraints and Limitations:

- **Data Privacy and Security:** Ensuring compliance with data protection regulations and safeguarding sensitive information.
- **User Training and Adoption:** Adequate training must be provided to users to ensure smooth adoption of the system.
- **Technical Expertise:** Ongoing maintenance and updates may require in-house technical expertise or reliance on external support.
- **Scalability and Integration:** The system's architecture needs to be adaptable to accommodate future growth and potential integration with other university systems

FRONT-END LANGAUGES/Framework/LIBRARIES:

Languages

Java:

- **Purpose:**
Java is the primary programming language used for the frontend development. It is chosen for its robustness, platform independence, and an extensive library support.
- **Contribution:**
Java provides a strong foundation for building a stable and efficient user interface, facilitating event-driven programming that is essential for interactive applications.

Frameworks and Libraries:

Swing (Java Swing):

- **Purpose:**
Swing is a part of Java Foundation Classes used for building graphical user interfaces
- **Contribution:**
Swing provides a rich set of components (such as buttons, tables, and text fields) and flexible customization options, allowing developers to create sophisticated and create interactive user interfaces.

AWT (Abstract Window Toolkit):

- **Purpose:**

AWT is a core Java library used for building graphical user interfaces (GUIs) for Java programs.

- **Contribution:**

- **Components:** AWT provides a set of classes for creating and managing GUI components such as buttons, text fields, checkboxes, and more.

- **Layout Management:** AWT includes layout managers to arrange components within the container, ensuring that they are displayed correctly.

- **Event Handling:** AWT supports event-driven programming, allowing developers to respond to user actions such as: button clicks and mouse movements

JDate Chooser (com.toedter.calendar.JDateChooser):

- **Purpose:**

This is a date picker component from JCalendar library.

- **Contribution:**

JDate Chooser simplifies the process of selecting dates within application, enhancing user experience by providing an intuitive and user-friendly date selection mechanism.

DbUtils (net.proteanit.sql.DbUtils):

- **Purpose:** DbUtils is a library from Apache Commons that simplifies database interaction by converting ResultSet objects into table models for use with JTable.

- **Contribution:** It streamlines the process of displaying and managing database records in the application's UI, allowing for efficient and dynamic data presentation in table format.

Java Database Connectivity (JDBC):

- **Purpose:**

JDBC is an API that allows Java applications to interact with relational databases.

- **Contribution:**

JDBC facilitates database operations such as querying and updating records, which are essential for managing student, teacher, exam, and fee details within the UMS.

Random (java.util.Random):

- **Purpose:** This is a utility class for generating random numbers.

- **Contribution:** Random can be used for generating unique IDs or passwords, ensuring variety and unpredictability in elements like user credentials.

Event Handling (java.awt.event):

- **Purpose:** This package contains classes for event handling in Java.

- **Contribution:** Event handling is crucial for creating interactive and responsive applications. It enables the application to respond to user actions like button clicks, mouse movements, and other interactions.

BACKEND-END LANGAUGES/Framework/LIBRARIES:

Languages

Java:

- **Purpose:** Java is the primary programming language is used for backend development in the University Management System. It is chosen for its robustness, scalability, and cross-platform capabilities.
- **Contribution:** Java provides the stable and efficient environment for implementing business logic, handling data processing, and ensuring smooth communication between frontend and backend components.

MySQL Workbench:

- **Purpose:** MySQL Workbench is a visual database designs tool and database management tool for MySQL databases.
- **Contribution:**
- **Database Design:** Allows developers to visually design and model their MySQL databases, including creating tables, defining relationships, specifying data types.
- **SQL Development:** Provides an integrated environment for SQL development, enabling developers to write and executes SQL queries, scripts, and stored procedures.
- **Database Administration:** Offers tools for a managing MySQL server instance, including user administration, backup and restore, and server monitoring. Java Database Connectivity

JDBC:

- **Purpose:** JDBC is an API that allows Java applications to interact with the relational databases.
- **Contribution:** JDBC facilitates database operations such that querying, updating, and managing records. It is fundamental for database connectivity and execution of SQL commands.

UNIQUE FEATURES IN THE SYSTEM:

Built-in notepad: Enhance teacher and student productivity by providing a readily accessible platform for note-taking within the system.

Built-in calculator: Offer a convenient tool for on-the-go calculations, particularly useful for students during exams or teachers for quick grade calculations.

Dynamic Dashboard with Real-Time Analytics:

Description:

The standout feature of our University Management System is its dynamic dashboard with real-time analytics. This dashboard provides administrators, teachers, and students with a comprehensive view of various aspects of the university, including student enrollment, academic performance, financial statistics, and administrative activities. The dashboard updates in the real-time, ensuring all stakeholders have access to the most current information.

Significance:

- **Enhanced Decision-Making:** The real-time analytics empower university administrators to make the data-driven decisions swiftly.
- **Efficient Resource Allocation:** By monitoring financial data and administrative activities in the real-time, the university can allocate resources more efficiently and effectively.
- **Personalized Experience:** The dashboard can be customized to display the relevant information based on their user's role (admin, teacher, student). This personalized experience enhances usability and user satisfaction.
- **Predictive Analytics:** Utilizing historical data, the system can provide predictive analytics to the forecast trends in student enrollment, course popularity, and academic performance.

Adds Values:

- **Competitive Advantage:** The real-time dashboard with analytics sets our UMS apart from others by providing a cutting-edge tool for the university management. It enhances operational efficiency, improves decision-making, and offers a user-friendly experience.
- **Scalability:** The system's architecture supports scalability, allowing the university to grow and adapt to changing needs and technological advancements.
- **Compliance and Security:** The system ensures compliance with the data protection regulations and provides robust security measures to protect sensitive university data.

PROJECT AND LAB TOPICS SYNCHRONIZATION:

Core Java Concepts

Data Types, Variables, Operators: Utilized for managing student, teacher, and course data efficiently.

Packages: Organize code for better modularity and reuse. `javax.swing`, `java.awt`, `event`, `java.util`.

Object-Oriented Programming (OOP) Concepts

- **Inheritance:** Establish hierarchical relationships between entities like students and teachers.
- **Polymorphism:** Achieved through method overriding and dynamic binding.
- **Abstraction:** Hide implementation details and focus on functionality.

Advanced Topics

- **Multithreading:** Enable concurrent processing for tasks such as student enrollment and exam scheduling.
- **Exception Handling:** Ensure robustness and manage errors during database interactions.

Abstract Data Types (ADT):

Organize and manipulate complex data structures effectively MAP ADT is used in action listener.

GUI Development

Swing: Develop user-friendly interfaces for administrators, teachers, and students.