



**Department of Computer Science**

**Quaid-e-Azam University, Islamabad**

**PROJECT**

**PLAN**

**Student Course Registration System**

Prepared by:

**HURR MEHDI**

**M. SEHAL BILAL**

**M. ASAD**

**ZOHAIB AHMED**

Submitted to:

**DR. ONAIZA MAQBOOL**

05 March 2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signature page **Team Member 1/Team Leader:** | |  | **Team Member 2** | |
| **By:** |  |  | **By:** |  |
|  | *(Signature)* |  |  | *(Signature)* |
| **Name:** | Hurr Mehdi |  | **Name:** | Muhammad Sehal Bilal |
|  |  |  |  |  |
| **Reg No. :** | 04072312048 |  | **Reg No. :** | 04072312015 |
| **Date:** | 05-03-2025 |  | **Date:** | 05-03-2025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Team Member 3:** | |  | **Team Member 4** | |
| **By:** |  |  | **By:** |  |
|  | *(Signature)* |  |  | *(Signature)* |
| **Name:** | Muhammad Asad |  | **Name:** | Zohaib Ahmed |
|  |  |  |  |  |
| **Reg No. :** | 04072312023 |  | **Reg No. :** | 04072213001 |
| **Date:** | 05-03-2025 |  | **Date:** | 05-03-2025 |

|  |  |
| --- | --- |
| **Supervisor:** | |
| **By:** |  |
|  | *(Signature)* |
| **Name:** | Dr. Onaiza Maqbool |
|  |  |
| **Position:** | Associate Professor |
| **Date:** |  |

# Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed By | Description |
| **1.0** | 05 March 2025 | Hurr Mehdi | Project Plan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Remarks:

|  |  |  |
| --- | --- | --- |
| Approved By |  | Date |

# Preface:

*This document outlines the project management plan for developing the Student Course Registration System for the CS Office. The system automates course registration, enforces course prerequisites, and enables coordinators to track student progress. The plan adheres to ISO/IEC/IEEE 16326 guidelines.*

Table of Contents:

[**PROJECT PLAN** 5](#_Toc192101587)

[**1. PROJECT OVERVIEW (Hurr Mehdi) 5**](#_Toc192101588)

[1.1 Project Summary 5](#_Toc192101589)

[1.2 Purpose, Scope, and Objectives 5](#_Toc192101590)

[1.3 Assumptions and Constraints 6](#_Toc192101591)

[1.4 Project Deliverables 6](#_Toc192101592)

[1.5 Schedule Summary 7](#_Toc192101593)

[**2. REREFERNCES (Hurr Mehdi) 7**](#_Toc192101594)

[**3. DEFINITIONS (Hurr Mehdi) 7**](#_Toc192101595)

[**4. PROJECT CONTEXT (Sehal Bilal) 8**](#_Toc192101596)

[4.1 Process Model 8](#_Toc192101597)

[4.2 Methods, Tools, and Techniques 8](#_Toc192101598)

[4.3 Product Acceptance Plan 8](#_Toc192101599)

[**5. PROJECT PLANNING (M. Asad) 9**](#_Toc192101600)

[5.1 Project Work plans 9](#_Toc192101601)

[5.2 Work Activities 9](#_Toc192101602)

[5.3 Schedule Allocation 10](#_Toc192101603)

[5.4 Resource Allocation 10](#_Toc192101604)

[**8. SUPPORTING PROCESS PLANS (Zohaib Ahmed) 11**](#_Toc192101605)

[8.1. Risk Management 11](#_Toc192101606)

List of Figures:

[Schedule Allocation (Project Libre) 10](#_Toc191992775)

[Resource Allocation (Project Libre) 10](#_Toc191992776)

List of Tables:

Project [Deliverables 6](#_Toc191992775)

[Schedule Summary 7](#_Toc191992776)

|  |  |
| --- | --- |
| PROJECT PLAN | **Student Course Registration System** |
| Created by | Group 8 |
| Date Created | 05-03-2024 |

## PROJECT OVERVIEW

## Project Summary

|  |  |
| --- | --- |
|  | The Student Course Registration System is a platform designed to assist academic administration for the CS Department. The system will automate course enrollment processes, enforce prerequisite validations, and provide real-time tracking of student academic progress. The platform will replace manual record keeping with a digital solution. Key functionalities include dynamic course scheduling, prerequisite rule enforcement, and automated reporting for academic coordinators, ensuring alignment with institutional policies and reducing administrative overload. |

## Purpose, Scope, and Objectives

**Purpose:**

*The current manual system for course registration and progress tracking is prone to delays, errors, and inconsistencies. This project aims to replace it with a secure, scalable web application that:*

* *Eliminates manual data entry errors.*
* *Enforces academic policies (e.g., prerequisites) programmatically.*
* *Provides stakeholders (students, coordinators, and faculty) with instant access to up-to-date academic records.*

**Scope:**

*The system will encompass:*

* *Study Scheme Management: Upload and update curriculum structures for student batches.*
* *Automated Prerequisite Validation: Rule-based checks during course registration to ensure compliance with academic requirements.*
* *Progress Tracking Dashboard: Real-time visibility into courses completed, grades, and pending requirements for students and coordinators.*
* *Reporting Module: Generate PDF/Excel reports on enrollment statistics, pass/fail rates, and prerequisite violations.*
* *Role-Based Access Control: Secure login tiers for students, faculty, and administrators.*

**Objectives:**

* *Ensure 100% adherence to prerequisite rules during registration.*
* *Provide coordinators with instant access to student academic histories.*
* *Automate deletion of outdated course entries after semester deadlines.*
* *Simplify course registration for students with intuitive interfaces.*
* *Eliminate manual data entry for coordinators through batch uploads and system-generated reports.*
* *Maintain accurate, up-to-date records of student progress and course prerequisites.*

## Assumptions and Constraints

**Assumptions:**

* *Students and coordinators have internet access.*
* *Course registration data (e.g., student IDs, course codes) will be manually input.*
* *Prerequisite data is accurately maintained.*
* *Students will register for courses before deadlines.*
* *The system will support multiple academic batches.*

**Constraints:**

* *Must use Java for backend development.*
* *Project completion within the semester 16 weeks.*
* *Only authorized users (Students, Coordinators, Timetable Coordinator) can access relevant data.*

## Project Deliverables

|  |  |
| --- | --- |
| Deliverable | Description |
| **Project Plan** | *A formal document outlining the project’s scope, timelines, resource allocation, and risk management strategies.* |
| **SRS Documentation** | *A Software Requirements Specification detailing functional/non-functional requirements, use cases, and system constraints.* |
| **System Prototype** | *A mock-up UI design demonstrating core workflows (e.g., course registration, prerequisite checks).* |
| **Functional Application** | *A working system with all core features: prerequisite validation, pass/fail tracking, coordinator dashboards, and report generation.* |
| **Source Code** | *Well-commented Java code, configuration files, and dependencies for reproducibility.* |
| **User Manual** | *A brief guide explaining how students/coordinators can use the system (e.g., registration steps, report generation).* |
| **Test Plan and Reports** | *Documentation of test cases (e.g., prerequisite validation scenarios) and results to verify system accuracy.* |
| **Final Presentation** | *A PowerPoint and live demo summarizing the project’s phases, challenges, and outcomes for academic evaluation.* |

## Schedule Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Start Date | End Date | Deliverables |
| **Planning** | 19/02/2025 | 05/03/2025 | Project Plan |
| **Analysis** | 06/03/2025 | 26/03/2025 | SRS Documentation |
| **Design** | 27/03/2025 | 23/04/2025 | Prototype |
| **Development** | 24/04/2025 | 21/05/2025 | Functional System/User Manual/Source Code |
| **Testing** | 22/05/2025 | 28/05/2025 | Test Reports |
| **Presentation** | 29/05/2025 | 11/06/2025 | Presentation |

## REREFERNCES

* ISO/IEC/IEEE 16326:2019
* Case Study 3 – Problem Description
* University's Course Registration Guidelines
* Relevant Software Engineering Textbooks/Papers

## DEFINITIONS

* **Prerequisite Checking:** Automated validation of required courses before registration.
* **Scheme of Study:** Curriculum structure for a student batch.
* **SRS:** Software Requirements Specification. .
* **Coordinator:** Manages course registration and tracks student progress.
* **Timetable Coordinator:** Views student course registrations for scheduling.
* **Passed Courses:** Courses successfully completed by a student.
* **Skipped Courses:** Courses a student has not taken.

## PROJECT CONTEXT

## Process Model

*For the Student Course Registration and Status Tracking System, the V-model is adopted to ensure rigorous testing at each stage aligning with ISO/IEC/IEEE 16326 standards. Each development phase is paired with a corresponding testing phase to verify that all requirements from course registration rules to prerequisite enforcement are met.*

*Key Phases:*

* *Requirements Analysis: Define all functional and non-functional requirements.*
* *System Design: Outline a high-level architecture including registration, course management, and reporting modules.*
* *Detailed Design: Develop detailed specifications and prepare unit test cases.*
* *Implementation: Code the system components according to the design.*
* *Unit & Integration Testing: Test individual modules and their interactions.*
* *System & Acceptance Testing: Validate the complete system in a simulated real-world environment.*
* *Deployment & Maintenance: Deploy the system and provide ongoing support.*

## Methods, Tools, and Techniques

***Frontend****: HTML, CSS, JavaScript*

***IDE****: Eclipse*

***Backend:*** *Java*

***Database:*** *MySQL*

***Framework:*** *Spring Boot / JSF*

***Version Management:*** *Github*

***Project Management:*** *Project Libre, Canva, PowerPoint*

## Product Acceptance Plan

* *Acceptance Criteria: All features meet SRS requirements.*
* *User Acceptance Testing (UAT): Performed by students and coordinators*
* *Pre-requisite Checking Validation: Ensures only eligible courses can be registered.*
* *Performance Testing: Evaluates system efficiency under high load*

## PROJECT PLANNING

## Project Work plans

*The project work plan will be divided into discrete phases (Planning, Analysis, Design, Development, Testing, and Presentation) with clearly defined tasks and milestones. Each phase will conclude with a review meeting to assess progress and prepare for the next phase.*

## Work Activities

***Planning:***

* *Analyzing Case Study*
* *Developing Plan*
* *Create Work Plan*
* *Create Project Document*

***Analysis:***

* *Figure out Use Cases*
* *Develop Analysis Model*
* *Create SRS Model*

***Design:***

* *Database Schema Design (ERD)*
* *UI/UX Mockups (Prototype)*
* *Detailed System Design*

***Development:***

* *Backend Setup (Java)*
* *Frontend Development (HTML/CSS/JS)*
* *Database Implementation (MySQL)*
* *Module Integration*
* *Create User Manual*

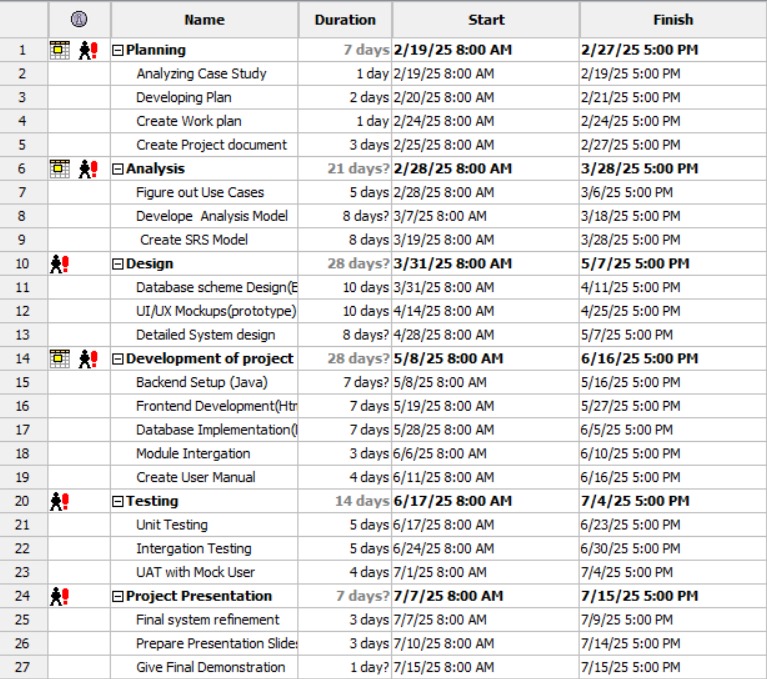
***Testing:***

* *Unit Testing*
* *Integration Testing*
* *UAT with Mock Users*

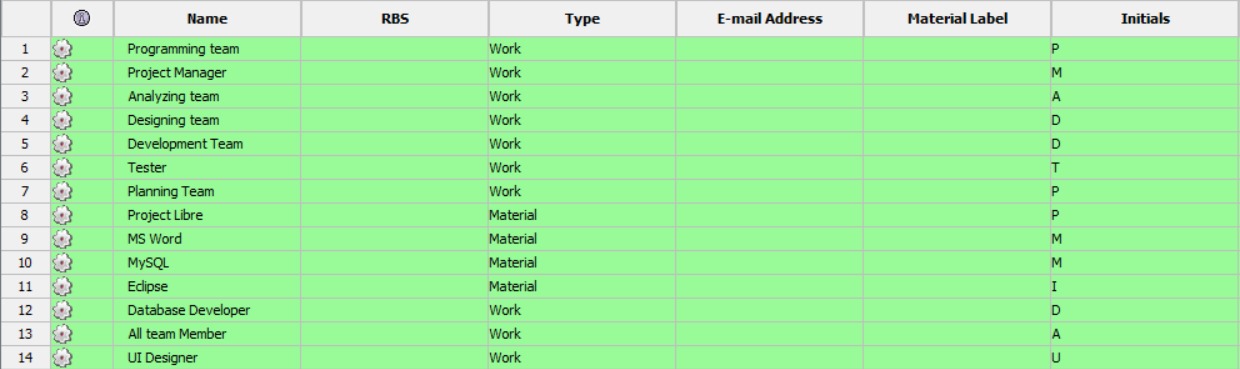
***Presentation:***

* *Final System Refinement*
* *Prepare Presentation Slides*
* *Give Final Demonstration*

## Schedule Allocation



## Resource Allocation



## 8. SUPPORTING PROCESS PLANS

## 8.1. Risk Management

***:***

***Technical Risks:***

* *Inadequate Java Proficiency*

*Impact: Delays in backend development and potential code errors.*

*Mitigation: Conduct weekly peer code reviews and allocate time for team members to upskill via online Java tutorials.*

* *Module Integration Failures:*

*Impact: Incompatibility between registration, database, and UI modules.*

*Mitigation: Use modular design principles and test integrations incrementally during development.*

* *Time-Consuming Tool Learning Curve*

*Impact: Delays due to unfamiliarity with tools like Spring Boot, MySQL, or Eclipse.*

*Mitigation: Dedicate the first week of each phase to tool training, use guided tutorials, and leverage pair programming for knowledge sharing.*

***Operational Risks:***

* *Team Member Overload*

*Impact: Delays due to conflicting academic commitments.*

*Mitigation: Assign tasks based on availability and use a shared Gantt chart for progress tracking.*

* *Time Constraints*

*Impact: Incomplete features by submission deadlines.*

*Mitigation: Prioritize core functionalities (prerequisite checks, progress tracking) over nice-to-haves.*

* *Team Member Unavailability (Sickness/Personal Issues)*

*Impact: Critical tasks stalled due to absenteeism.*

*Mitigation: Cross-train team members on all modules (e.g., frontend, backend), Maintain detailed documentation to enable easy handovers, Build buffer time into the schedule for unexpected absences.*