**PROJECT MANAGEMENT SYSTEM FOR FINAL YEAR STUDENTS**

**Project Plan**

**Group members: TASK**

1. Safeena Akhter-04072312008 (Team Lead)
2. Haleema Sadia-04072312010 (Team Member 2)
3. Masooma Muhammadi-04072312030 (Team Member 3)

**Submission date:** 3/5/2025

**Supervised by:** Dr. Onaiza Maqbool

**Department of Computer Science**

**QUAID-E-AZAM UNIVERSITY,ISLAMABAD**

**Signature Page**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Signature** |
|  |  |  |
| Safeena Akhter | Team Leader |  |
| Haleema Sadia | Team member |  |
| Masooma Muhammadi | Team member |  |

**Change History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **CHANGES MADE** |
| 1.0 | 3/4/2025 | Safeena Akhter | Initial Draft |
| 1.1 | 3/5/2025 | Safeena Akhter | Review Document and add Member 2 and Member 3 tasks . |
|  |  |  |  |

**Remarks:**



**Preface:**

This document outlines the project plan for the development of a Project Management System for the Department of Computer Science at Quaid-i-Azam University. The plan follows ISO/IEC/IEEE 16326 guidelines and provides a structured approach to project execution, covering objectives, constraints, planning, and risk management. The goal is to facilitate communication between students and supervisors during the final-year project process.

Table of Contents

[1. Project Overview 5](#_Toc192088095)

[1.1 Project Summary 5](#_Toc192088096)

[1.2 Purpose, Scope, and Objectives 5](#_Toc192088097)

[1.2.1 Purpose: 5](#_Toc192088098)

[1.2.2 Scope: 5](#_Toc192088099)

[1.2.3 Objectives: 6](#_Toc192088100)

[1.3 Assumptions and Constraints 6](#_Toc192088101)

[Assumptions: 6](#_Toc192088102)

[Constraints: 7](#_Toc192088103)

[1.4 Project Deliverables 7](#_Toc192088104)

[1.5 Schedule Summary 7](#_Toc192088105)

[2. References 8](#_Toc192088106)

[3. Definitions: 8](#_Toc192088107)

[4. Project Context: 9](#_Toc192088108)

[4.1 Process Model: 9](#_Toc192088109)

[4.2 Methods, Tools and Techniques: 10](#_Toc192088110)

[4.3 Product acceptance plan: 12](#_Toc192088111)

[Acceptance Criteria: 12](#_Toc192088112)

[4.4 Product Acceptance Methods: 12](#_Toc192088113)

[5. Project Planning 12](#_Toc192088114)

[5.1 Project Work Plan 12](#_Toc192088115)

[5.2 Work Activities 15](#_Toc192088116)

[5.3 Schedule Allocation: 15](#_Toc192088117)

[5.4 Resource Allocation: 16](#_Toc192088118)

[8. Supporting Process Plans: 16](#_Toc192088119)

[8.1 Risk Management 16](#_Toc192088120)

[8.1.1 Risk Identification: 16](#_Toc192088121)

[8.1.2 Risk Analysis: 17](#_Toc192088122)

[8.2 Risk Mitigation Strategies 17](#_Toc192088123)

[8.3 Risk Monitoring and Control 18](#_Toc192088124)

**List of Figures:**

Figure1: Project Timeline

Figure2: Resource allocation

|  |
| --- |
|  |

**List of Tables:**

Table 1: Change History

Table2: Schedule Summary

# 1. Project Overview:

## 1.1 Project Summary

The Department of Computer Science at Quaid-e-Azam University seeks to develop a **Project Management System** to facilitate seamless communication and collaboration between final-year students and their supervisors. Every semester, students undertake final-year projects under the guidance of faculty members, requiring continuous feedback, version control, and structured documentation. This system will serve as a centralized platform where students can register their projects, get assigned supervisors, upload work products, receive structured feedback, and track multiple versions of a single work product. Supervisors will be able to monitor progress, review submissions, provide timely feedback, and suggest necessary improvements.

## 1.2 Purpose, Scope, and Objectives

### 1.2.1 Purpose:

The **Project Management System** aims to improve efficiency, transparency, and collaboration in managing final-year projects within the **Department of Computer Science** at Quaid-i-Azam University. This system will:

* Offer a central platform for students and supervisors to organize project-related activities.
* Simplify communication by allowing formalized feedback and progress monitoring.
* Enable secure and systematic submission of work products (documents, reports, code files).
* Support accountability and transparency through a computerized record of submissions and feedback.
* Automate reminders and notifications to update students and supervisors regarding deadlines.

### 1.2.2 Scope:

* User Authentication & Role Management: Secure student, supervisor, and administrator login with varying levels of access.
* Project Registration & Supervisor Assignment: Students register their projects, and supervisors will be allocated accordingly.
* Work Product Submission & Version Control: Students can submit project documents, and previous versions will be archived for reference.
* Supervisor Feedback & Approval System: Supervisors can view submissions, leave comments, and approve work.
* Project Progress Tracking: Student and supervisor dashboards to track deadlines, feedback, and project status.
* Notifications & Alerts: Automated notifications for outstanding submissions, requests for feedback, and critical deadlines.
* Basic Administrative Control: Admins can control users, projects, and system access.

### 1.2.3 Objectives:

The goal is to develop a **functional and user-friendly** platform that improves project management within the university. The key objectives are:

* Develop web-based system that integrates collaboration and built effective communication between students and supervisors.
* Ensure an intuitive interface using simple and responsive design principles. Provide a simple yet effective dashboard for students and supervisors to track progress.
* Implement a document submission system with version control for project tracking. Version control helps in building better system.
* Enable structured feedback so supervisors can easily review and comment on submissions.
* Secure the system with authentication and role-based access control.
* Automate key processes such as reminders and notifications for deadlines.

“By implementing this system, the final-year project process will become more efficient, organized, and transparent, benefiting both students and faculty”.

## 1.3 Assumptions and Constraints

### Assumptions:

* Users (students, supervisors, and administrators) will have reliable internet access to use the system.
* Supervisors will actively engage with the platform and provide timely feedback on project submissions.
* The system will be accessible to students, supervisors, and administrators both on and off campus.
* Students will submit work within the specified deadlines.
* Each project will have a designated supervisor who will oversee progress and evaluations.
* Users will have basic technical knowledge to navigate and interact with the system efficiently.
* Authentication will be required to ensure only authorized users (students, faculty, and administrators) can access the system.

### Constraints:

* Development must be completed within the academic semester, limiting the time for implementation and testing.
* The system must comply with university data security policies and privacy regulations.
* Access will be restricted to registered users (students, faculty, and administrators) via university email authentication rather than limiting access by network.
* The platform should be lightweight and optimized to function efficiently on various devices and internet connections.
* The system will only support project-related document formats (e.g., PDFs, DOCX, and ZIP files for code submissions).
* The scope of features will be limited to core project management functionalities to ensure timely completion and smooth operation.
* Data encryption and secure login mechanisms (e.g., multi-factor authentication) will be implemented to protect user data.

## 1.4 Project Deliverables

Project Deliverables include :

1. Project Plan.
2. Gannt chart for timeline.
3. Software Requirements Specification (SRS) Document.
4. Design Plan Document.
5. Use case diagrams.
6. Final design.
7. Functional Prototype.
8. Final System.
9. Final Presentation.

## 1.5 Schedule Summary

|  |  |  |
| --- | --- | --- |
| **Task or Phase** | **Time Period** | **Deliverables** |
| **Planning** | 3/05/25 | Project Plan Document |
| **Analysis** | 3/27/25 | Use Case Diagrams, SRS Document. |
| **Design** | 4/28/25 | Design Plan Document,  Use case diagrams. |
| **Development** | 5/26/25 | Functional Prototype. |
| **Testing** | 6/5/25 | Refined System |
| **Submission** | 6/2/25 | Final System |
| **Presentation** | 6/3/25 | Final Presentation. |

# 2. References:

* ISO/IEC/IEEE 16326: Systems and Software Engineering – Project Management
* Software Engineering: A Practitioner's Approach – Roger S. Pressman
* Project Management for Software Engineering – Richard H. Thayer

# 3. Definitions:

**Student:** A final-year student who registers a project, submits work products, and interacts with the supervisor through the system.

**Supervisor:** A faculty member responsible for guiding and assessing a student’s project.

**Work Product:** Any document, code, or deliverable submitted by a student for evaluation.

**Version Control**: The process of managing multiple iterations of a work product.

**Project Portal:** The web-based system used for project management.

**Milestone:** A key project phase or deadline that marks significant progress in the project lifecycle.

**Submission System:** The feature that allows students to upload work products and supervisors to provide feedback.

**Feedback Mechanism:** The process through which supervisors review, comment on, and approve student submissions.

**Project Repository:** A centralized storage system within the project portal where all project-related files and documents are maintained.

**Authentication:** The process of verifying a user’s identity via email verification before granting access to the system.

**Dashboard:** The main interface where users (students, supervisors, administrators) can access relevant project details, submissions, and feedback.

**Gantt Chart:** A visual representation of the project schedule, showing tasks, durations, and dependencies.

**System Documentation:** Comprehensive documents outlining the system's functionality, user guides, and technical specifications.

**Access Control:** The system’s mechanism to restrict access to authorized users based on their roles (student, supervisor, administrator).

**Project Evaluation:** The assessment process conducted by supervisors based on submitted work products and adherence to project requirements.

# 4. Project Context:

## 4.1 Process Model:

We will use Scrum, an incremental and iterative agile process, for our project management system.

**Reason for using Scrum model:**

**Frequent Feedback:** Students and supervisor interact often, scrum iterative process allows frequent feedback.

**Incremental development:** Project submission, feedback, and version control aspects can be developed in steps (sprints).

* Sprints: The development shall be divided into brief sprints, each with a focus on one features For example project registration, submission, feedback.

**Flexibility:** As new features or improvements might be needed in the process of development, Scrum iterative approach allows for alterations easily.

* Scrum enables us to manage work products, feedbacks and notifications with continuous development.

**Major Project Activities:**

* Gathering Requirements from students and supervisors.
* System Design to give an effective framework for project management.
* System development, testing, and deployment in multiple iterations (sprints),which is the largest advantage of using scrum model.

**Function Increments based on Sprints will include:**

* Student Project Registration
* Assignment of Supervisors to Projects
* Student Work Product Upload
* Supervisor Feedback
* Project Progress Tracking

Each sprint will be finished with feedback sessions in order to gather feedback and change the system if necessary.

## 4.2 Methods, Tools and Techniques:

**Methods:**

* Agile principles for iterative development and flexibility.
* Version control Git and Github for multiple submissions of work products.

**Tools:**

**VS Code:** Provides an integrated development environment (IDE) for coding, debugging, and testing code.

**MySQL:** Stores student, supervisor, and project data, including multiple versions of submitted work.

**HTML, CSS, JavaScript:** For front-end design and interaction required in any browser-based user interface in order to create an intuitive experience for the students and supervisors.

**Project Management tool:**

* **ProjectLibre**: For creating Gantt chart to set deadlines and track overall project scheduling

**Programming Language:**

* **Java:** Programming language having strong OOP features and wide community support, suitable for secure and scalable development

**Webserver:**

* **Apache Tomcat:** As a web application server online so that students and supervisor can access it online.

**Communication tool:**

* **Microsoft Teams:** Maintained student and supervisor in sync by chat, call, and work product sharing and reviewing.

**Version Control tool:**

* **Git and GitHub:** Manages code changes effectively, enabling collaboration and version tracking.

**Techniques:**

* **Weekly Meetings:**

Provides a specific time to receive feedback from supervisor, discuss project status, and remove any changing requirements and handle unnecessary blockers during project development.

* **Version Indexing**: When students upload multiple versions of a work product, each file is stored with a version index (For example v1,v2,v3) to track changes and easily retrieve previous versions.

## 4.3 Product acceptance plan:

The product acceptance plan will include:

### Acceptance Criteria:

Defined for each feature and functionality ensure they meet the defined requirement

* **Project Registration:** Students should be able to register their projects.
* **Submission of Work:** Multiple versions of work product must be submitted by students.
* **Feedback by Supervisor:** Supervisors must be able to see and comment on student submissions.
* **Authentication of Users:** Secure login for supervisors and students.
* **Integration with the Database:** All project-related data must be stored and fetched in the correct way.

## 4.4 Product Acceptance Methods:

* **Unit Testing:** Tests individual modules to ensure each component works correctly.
* **Integration Testing:** After individual modules are verified, they are tested together to ensure smooth interaction between components For example: how the submission module communicates with the feedback module.
* **User Acceptance Testing (UAT):** Involves actual end-users final year students and supervisors to ensure that the system fulfill real-world needs and check the system's functionality and usability. Because the system is utilized intensively by supervisors and students, involving them in UAT ensures that the interface is easy to use and the functionalities like multiple version uploads, feedbacks works as intended

# 5. Project Planning

## 5.1 Project Work Plan

**Features:  
  
Student Project Registration:**

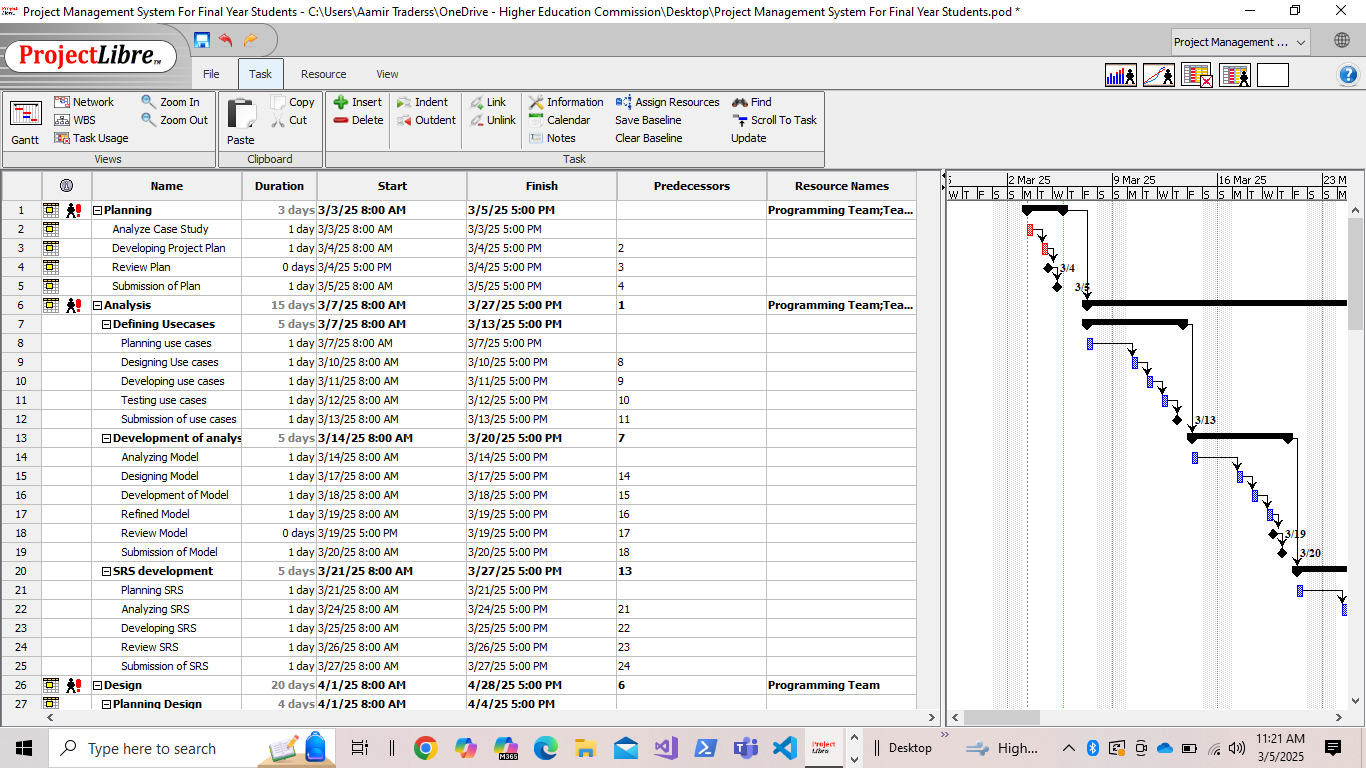
* Students can sign up for projects and select project supervisors.

**Submission of Work Products:**

* Students are permitted to submit more than one version of their work.  
  Supervisors could give feedback on students' work products using a feedback system that allows for several revisions.

**Project tracking:**

* Students and supervisors can monitor the project's progress.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## 5.2 Work Activities

## 5.3 Schedule Allocation:

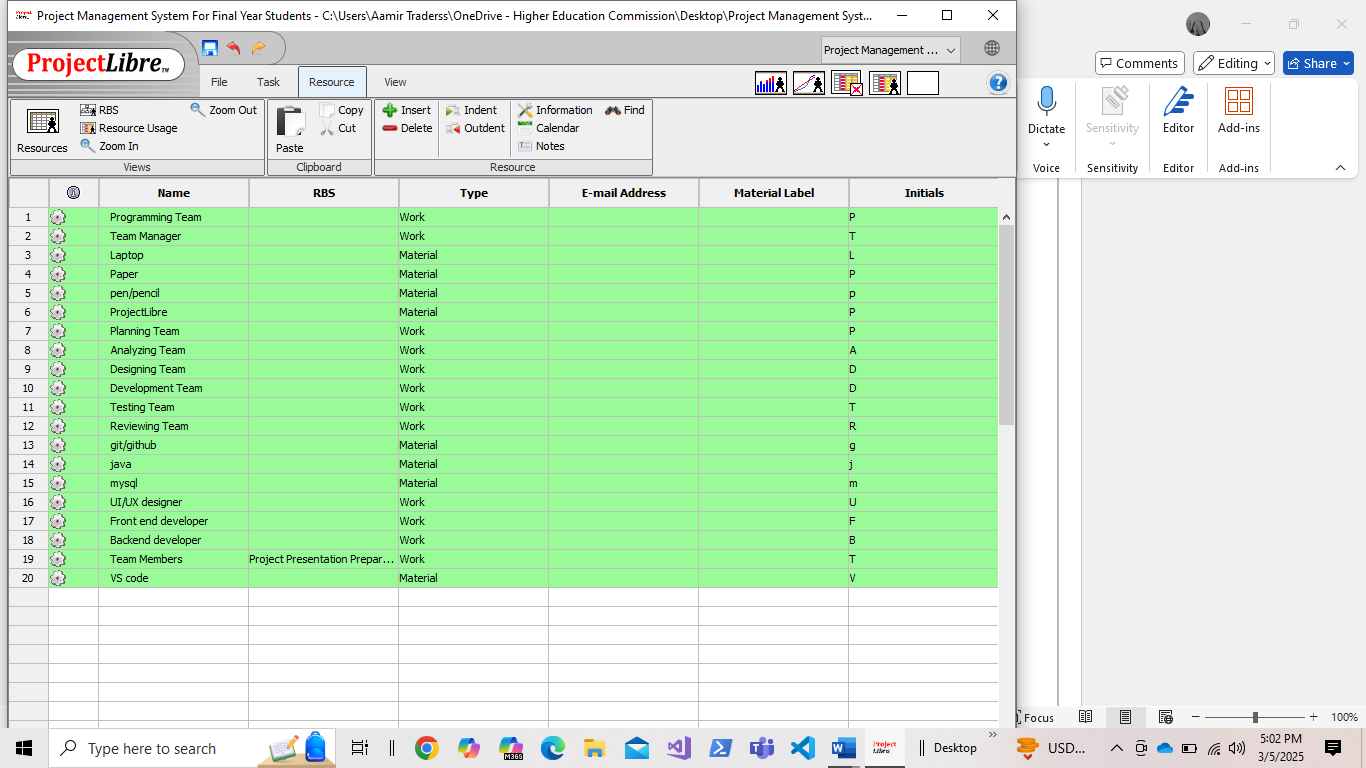
* 1. **Project Setup & Initial Planning (Duration: 1 Week)**

Define project scope, set up the development environment, and prepare the project management tools.

Key Activities:

* Set up the GitHub repository for version control.
* Design the system's database schema in MySQL.
* Set up VS Code as the development environment.
* Develop a detailed Gantt chart in ProjectLibre to outline the overall project timeline and dependencies
  1. **System Design (Duration: 2 Weeks)**
* Designing the system architecture (how data flows from front-end to back-end).
* To create database schema and plan the structure of tables in MySQL.
* Designing the UI for the registration page, submission system, feedback form, and progress tracking page.
  1. **.Development Phase:**
* Sprint 1: Student Registration & Supervisor Assignment*(Duration: 2 Weeks)*
* Sprint 2: Work Product Submission*(Duration: 2 Weeks)*
* Sprint 3: Feedback System *(Duration: 2 Weeks)*
* Sprint 4: Project Progress Tracking*(Duration: 2 Weeks)* 
  1. **Testing Phase*(Duration: 2 Weeks)***
  + Perform integration testing on all system features to ensure everything works together.
  + Run user acceptance testing (UAT) with a few students and supervisors to verify that the system meets their needs.

## 5.4 Resource Allocation:

****

# 8. Supporting Process Plans:

## 8.1 Risk Management

### 8.1.1 Risk Identification:

**Technical Risks**

**Lack of Java experience:** We risk generating poor or faulty code because we're not yet experienced with Java, which could present developmental challenges and raise the risk of errors.

|  |  |
| --- | --- |
| **Probability** | **Impact** |
| Medium | High |

**Data Security:** Storing multiple versions of work products could lead to security risks if access is not controlled properly

|  |  |
| --- | --- |
| **Probability** | **Impact** |
| Medium | High |

**Operational Risks**

**Development Delays:** Academic responsibilities (exams, other projects and assignments) may slow the project progress

|  |  |
| --- | --- |
| **Probability** | **Impact** |
| High | Medium |

**Resource Availability:** Unavailable tools, equipment, or limited hardware and Supervisor unavailability cause schedule disturb.

|  |  |
| --- | --- |
| **Probability** | **Impact** |
| Medium | Medium |

**Scope Changes:** When new functionality is requested, it has the potential to increase the project scope and create rework.

|  |  |
| --- | --- |
| **Probability** | **Impact** |
| Medium | Medium |

### 8.1.2 Risk Analysis:

**Probability and Impact Assessment:** Every risk is considered for how likely it is to happen (low, medium, high) and how bad the consequences would be (low, medium, high).For example: Data security can be medium probability but high impact, so it calls for strict security measures.

**Risk Prioritization:** Prioritize risks by their severity .Critical security issues and high-severity risks get addressed first, with medium or low-severity risks being watched over and treated when needed.

## 8.2 Risk Mitigation Strategies

**Technical Risks**

**Code Reviews & Pair Programming:** Error detection at an early stage and peer-to-peer learning help to solve java programming problems efficiently.

**Security Measures:** Use role-based access, encryption, and safe login to protect student data.

**Module Testing:** Test each component (registration, submission, feedback) separately before they are integrated, to prevent massive integration problems.

**Version Indexing:** Each version of a work product will be clearly labeled (For example: v1, v2) and secured so that no files are accidentally overwritten or mixed up.

**Operational Risks**

**Clearance Project Planning:** Use a Gantt chart to put in deadlines and divide up the tasks equally.

**Regular Progress Checks:** Have weekly meetings to find out if there are delays or shortages of resources and correct them.

**Change Control Process:** Record new feature requests and see how they impact the timeline and resources before implementing them.

## 8.3 Risk Monitoring and Control

**Regular Risk Reviews:** At the end of each sprint or during project updates, we’ll look over our risk list to see if anything has changed or if new risks have popped up.

**Risk Register:** We'll keep one overall file like MS Word with a list of all risks, the seriousness of them, and how we are addressing each one. In that way, everybody knows exactly where to obtain the up-to-date details.

**Regular Communication:** We will keep communicating regularly with students, supervisors so that we can recognize any new issues easily and resolve them before they become big problems.