CS-322 Software Construction

Project Plan

Project Management System

PRESENTED BY

MEMBER 1: Tayyab Mehboob (LEADER) (04072213035)

MEMBER 2: M. Hamad (04072213023)

MEMBER 3: M. Ehtisham (04072213026)

Member 4: M. Rayyan (04072312024)

SUPERVISED BY

Dr. Onaiza Maqbool

Department of Computer Science

Quaid-e-Azam University, Islamabad

Project Plan Document Approval Signatures

Project Manager

_____ _____

Signature Date

Tayyab Mehboob

Student, QAU Islamabad

Member 1

_____ _____

Signature Date

M. Hamad

Student, QAU Islamabad

Member 2

_____ _____

Signature Date

M. Ehtisham

Student, QAU Islamabad

Member 2

_____ _____

Signature Date

M. Rayyan

Student, QAU Islamabad

Stakeholder

_____ _____

Signature Date

Dr. Onaiza Maqbool

Professor, QAU Islamabad

Change History

Version No.	Date	Modified By	Changes
1	**03/03/2025**	**Tayyab Mehboob**	**Original**

Remarks

Signature

Dr. Onaiza Maqbool

Professor, QAU Islamabad

Table of Contents

List of figures……5

List of tables……5

1. Project Overview ……5

1.1 Project Summary ……5

1.2 Purpose, Scope, and Objectives ……5

1.3 Assumptions and Constraints ……6

1.4 Project Deliverables ……...6

1.5 Schedule Summary ……7

2. References ……7

3. Definitions ……...7

4. Project Context ……7

4.1 Process Model ……7

4.2 Methods, Tools, and Techniques ……10

4.3 Product Acceptance Plan ……12

5. Project Planning ……14

5.1 Work Activities ……14

5.2 Schedule Allocation ……15

5.3 Resource Allocation ……15

8. Supporting Process Plans ……...16

8.1 Risk Management ……...16

Ghant Char/ Task Network ……..18

List of figures

List of tables

1. Project Overview

1.1 Project Summary

The main aim of this project is to develop a **Project Management System** that allows students to register their final year projects along with their supervisors on this platform. Students can submit their work products and supervisors can provide feedback on the submission. The system can support multiple versions of work products, ensuring that students can improve and update their submission, and it will provide a centralized platform for project management.

1.2 Project Purpose, Scope and Objectives

Purpose

To develop a system where students can register their semester projects with their supervisors, submit their work product and can receive feedback. The system will streamline the process of project registration, work product submission, and feedback management. It will also support multiple versions of work products, allowing students to update and improve their submissions.

Scope

- **Context**

  This is a **Standalone system** managing the project registration, submission and feedback process. It ensures seamless communication between students and supervisors.

- **Functions**
  - **Functional Requirements**

    The system will allow:

    1. Students register projects and submit the work product for reviews.
    2. Supervisors can provide feedback on submissions.
    3. The system supports multiple versions of work products.

  - **Non-Functional Requirements**
    1. **Usability:** The user interface should be in-built, enabling students and supervisors to easily register projects, submit work products and provide feedback.
    2. **Scalability:** The system architecture should allow for easy scaling to accommodate increased users (students and supervisors) without major changes.

3. **Security:** The system should implement data encryption and user authentication to protect sensitive projects and feedback information.

Objectives

1. To provide a **centralized platform** for students and supervisors to easily communicate and manage project submissions and feedback.
2. To make the process of project registration, submission and feedback efficient.
3. To support multiple versions of work, allowing students to improve and resubmit their work.
4. Automatic management of project tracking and feedback reduce the manual effort for both students and supervisors.

1.3 Project Constraints and Assumptions

Constraints

1. The software will be built in **Java programming language.**
2. The system will develop as an app-based platform.
3. It will be provided as free software.
4. The platform will be available only in **English language.**
5. The project must be completed by 14 weeks of development cycle.
6. The system will have a limited user base which may impact on the efficiency of project registration, submission and feedback processes.

Assumptions

1. Assuming that there is a significant need for a project management system to streamline communication between students and supervisors.
2. Assuming that students and supervisors have the basic knowledge to use and navigate the platform easily.
3. Assuming that students will provide accurate and up-to-date information and work products.
4. Assuming that supervisors will provide clear and relevant feedback on submitted work products.

1.4 Project Deliverables

A detailed document at each phase:

1. **Project Plan:** A document that specifies the overall software development strategy and timeline.
2. **Software Requirements Specification (SRS):** A document detailing the entire functionality of the software and use cases associated with each requirement.
3. **Analysis Document:** A document describing the design prototypes and architectural details of the system.

4. **Fully functional software** that will incorporate all the core features.
5. **Quality assurance and testing reports,** comprehensive reviews of work product, phase or a milestone.
6. **Work products** at each meeting internal to the team.
7. **Design prototypes** representing the system visually.
8. **User guidelines and system manuals** to assist users in navigating the software.

1.5 Schedule Summary

1. **Phase 1** (Planning): 1 week
2. **Phase 2** (Analysis): 3 weeks
3. **Phase 3** (Design): 4 weeks
4. **Phase 4** (Development): 4 weeks
5. **Phase 5** (Testing): 3 weeks

2. References

1. IEEE Software Construction Guideline. ISO/IEC/IEEE 16326:2nd edition 2019-12
2. Project Libre manual version 0.1-October 6,2012.

3. Definitions

1. **IEEE:** Institute of Electrical and Electronics Engineers.
2. **ISO:** The International Organization for Standardization.
3. **Work Product:** A document or file submitted by students (e.g., reports, code) for review by supervisors.
4. **Version Control:** Management of multiple versions of work products, allowing students to update and improve their submissions.
5. **Deliverables:** The outcomes after completion of a particular phase, such as project documentation or functional software.

4. Project Context

4.1 Process Model

The project follows an Iterative and Incremental Process Model inspired by the Scrum framework, ensuring flexibility, adaptability, and continuous stakeholder feedback throughout the 14-week development cycle. This model is divided into sequential phases with iterative sprints, allowing the team to refine deliverables incrementally while maintaining alignment with academic requirements and ISO/IEC/IEEE 16326 guidelines.

1. Project Initiation

- **Objective**: Define the project's scope, stakeholders (students, supervisors, department), and success criteria.
- **Activities**:
    - Conduct kickoff meetings with stakeholders to finalize requirements.
    - Draft the **Project Plan** using **ProjectLibre** to outline timelines, roles, and deliverables.
    - Establish **version control workflows** using **Git/GitHub** to manage code and documentation.

2. Planning Phase

- **Objective**: Break down the project into manageable sprints and assign tasks.
- **Activities**:
    - Define **user stories** (e.g., "Student uploads Report_v1," "Supervisor reviews submission").
    - Create a **Sprint Backlog** prioritizing core features:
        - Project registration module.
        - Multi-version submission logic.
        - Feedback management system.
    - Develop a **Gantt chart** in ProjectLibre to visualize the 14-week timeline.

3. Analysis Phase

- **Objective**: Formalize system requirements and workflows.
- **Activities**:
    - Conduct workshops with students and supervisors to map **use cases** (e.g., "Submit Work Product," "Provide Feedback").
    - Draft the **Software Requirements Specification (SRS)** document, detailing functional and non-functional requirements.
    - Design **Entity-Relationship (ER) diagrams** for the MySQL database, including tables for projects, submissions, and feedback.

4. Design Phase

- **Objective**: Create blueprints for the system's architecture and interface.

- **Activities**:

  1. Develop **UI/UX prototypes** in **Figma** for student and supervisor dashboards.
  2. Define **Java class diagrams** (e.g., Student, Submission, Feedback) using Eclipse.
  3. Finalize the **system architecture** (MVC pattern) with Spring Boot for backend logic and JSP for frontend rendering.

5. Development Phase

- **Objective**: Build and integrate system components iteratively.
- **Activities**:
  - **Sprint 1**
    1. Implement **project registration** (Java forms + MySQL CRUD operations).
    2. Enable supervisors to approve/reject projects via backend logic.
  - **Sprint 2**
    1. Develop **version control** for submissions (e.g., auto-increment version field in MySQL).
    2. Integrate **feedback module** with timestamped comments (Java Local Date Time).
  - Conduct **daily Scrum meetings** to address blockers and track progress.

6. Testing Phase

- **Objective**: Validate functionality, performance, and usability.
- **Activities**:

  1. **Unit Testing**: Use **JUnit** to validate Java classes (e.g., Submission Service, Feedback Controller).
  2. **Integration Testing**: Simulate end-to-end workflows (e.g., submit → feedback → resubmit) using **Selenium**.
  3. **User Acceptance Testing (UAT)**: Invite 15 students and 5 supervisors to test the system. Metrics include task success rate (>90%) and bug severity (zero critical issues).

7. Deployment & Closure

- **Objective**: Launch the system and finalize documentation.
- **Activities**:
  - Deploy the application on **Apache Tomcat** using WAR files.
  - Archives completed projects automatically (Java Scheduled Executor Service).

- Submit **final deliverables**:
    1. Fully functional system with version control and feedback workflows.
    2. User manuals for students and supervisors.
    3. Test reports and **ISO/IEC/IEEE 16326-compliant** documentation.

8. Reviews & Retrospectives

1. **Sprint Reviews**: Bi-weekly meetings with stakeholders to demo features (e.g., version control logic) and gather feedback.
2. **Final Review**: A formal session with Dr. Onaiza Maqbool to sign off on deliverables.
3. **Retrospectives**: Team reflection meetings to identify improvements (e.g., code quality, communication).

4.2 Methods, Tools and Techniques

Methods

The project employs a combination of **structured software engineering methodologies** to ensure systematic development, scalability, and compliance with academic requirements:

1. **Scrum Methodology**:
    - **Bi-weekly sprints** to deliver incremental features (e.g., project registration, version control).
    - **Daily standups** to track progress and resolve blockers.
    - **Sprint reviews** with stakeholders (students, supervisors) to validate deliverables and gather feedback.
2. **Object-Oriented Programming (OOP)**:
    1. Java classes model real-world entities (e.g., Student, Project, Submission, Feedback).
    2. **Encapsulation**: Secure data access via getters/setters (e.g., get Submission Version ()).
    3. **Inheritance**: Reusable code for common functionalities (e.g., User superclass for Student and Supervisor).
3. **Model-View-Controller (MVC) Architecture**:
    1. **Model**: Java entities (e.g., Submission.java with fields submission Id, version, filePath).
    2. **View**: JSP (JavaServer Pages) + HTML/CSS for rendering dashboards and forms.
    3. **Controller**: Spring Boot controllers (e.g., SubmissionController.java) to handle HTTP requests.

4. **Modular Development**:

   Break the system into independent modules:

   1. **Registration Module**: Java forms + MySQL CRUD operations.
   2. **Version Control Module**: Logic to manage multiple submissions (e.g., v1, v2).
   3. **Feedback Module**: Timestamped comments linked to submissions.

Tools

1. **Java & Eclipse**:
   - Core language for backend/frontend logic.
   - Eclipse IDE for coding, debugging, and Git integration.
2. **MySQL & JDBC**:
   - Relational database for structured data storage (projects, submissions).
   - JDBC API for secure Java-MySQL connectivity.
3. **Git/GitHub**:
   - Version control for codebase and documentation.
4. **ProjectLibre**:
   - Gantt charts for tracking the 14-week schedule and milestones.
5. **Figma**:
   - UI/UX design for student/supervisor dashboards.
6. **Apache Tomcat**:
   - Deployment of the Java web application via WAR files.
7. **MS Word**:
   - ISO-compliant documentation (SRS, user manuals).

Techniques

1. **Use Case Modeling**:
   - Define user interactions using UML diagrams:
     - **Student**: "Upload Work Product v1", "View Feedback".
     - **Supervisor**: "Review Submission", "Add Feedback".
   - Tools: **Eclipse UML Plugins** for diagram design.
2. **Entity-Relationship (ER) Modeling**:

- Design database scheme with relationships:
  - **One-to-Many**: One project → Many submissions.
  - **One-to-One**: One submission → One feedback entry.
- Tool: **MySQL Workbench** for ER diagrams.

3. **Version Control Logic**:
   - **Auto-increment versions**: Java logic to append _v1, _v2 to filenames.
   - **Timestamping**: Use Java LocalDateTime to track submission and feedback times.
4. **Security Implementation**:
   - **Role-Based Access Control (RBAC)**:
     - Students: Access to submission forms and feedback view.
     - Supervisors: Access to approval/rejection and feedback modules.
   - **Spring Security**: Secure login/logout workflows and password encryption.
5. **Data Validation**:
   - **Java Annotations**: Not-Null, @Size to validate form inputs (e.g., project titles, feedback comments).
   - **MySQL Constraints**: UNIQUE for project IDs, NOT NULL for deadlines.
6. **Feedback Loop Management**:
   - **Email Notifications**: JavaMail API to alert students when feedback is added.

   Dashboard Updates: Real-time status changes (e.g., "Feedback Received").

4.3 Product Acceptance Plan

The system will be accepted only if it fulfills the following criteria:

User Registration and Authentication
✓ Students and supervisors must register with valid credentials (name, university email, password).
✓ Role-based authentication ensures only registered users (students/supervisors) access their dashboards.

Project Registration and Approval
✓ Students can submit project proposals with details (title, description, deadline).
✓ Supervisors can approve/reject proposals and set deadlines.
✓ Projects auto-archive after deadlines.

Work Product Submission
Students can upload multiple versions of work products (e.g., Report_v1, Report_v2).
Submissions include timestamps and version history.
Students receive confirmation notifications upon successful uploads.

Feedback Management

✓ Supervisors can provide timestamped feedback linked to specific submission versions.
✓ Students can view feedback and resubmit improved work.
✓ Automatic notifications alert students when feedback is added.

Data Security and Performance

✓ Role-based access controls (students: submit/view work; supervisors: approve/review).
✓ MySQL encryption protects sensitive data (e.g., project details, feedback).
✓ System handles 50+ concurrent users during peak submission periods.

2. Testing Strategy

Unit Testing

- Validate Java classes (e.g., SubmissionService, FeedbackController) using **JUnit**.
- Ensure version control logic increments submissions correctly (e.g., v1 → v2).

Integration Testing

- Test end-to-end workflows (e.g., project registration → submission → feedback) using **Selenium**.
- Verify MySQL connectivity via **JDBC** (e.g., CRUD operations for projects and submissions).

User Acceptance Testing (UAT)

- Conduct UAT with **15 students** and **5 supervisors** from the department.
- **Success Metric**: 95% of users must complete core tasks (submit work, view feedback) without errors.

3. Sign-Off Procedures

Approvers

- **Dr. Onaiza Maqbool** (Supervisor)

Conditions for Approval

- All functional requirements are met (zero critical bugs).
- Final deliverables submitted:

- Fully functional system with version control and feedback workflows.
- SRS, user manuals, and test reports compliant with ISO/IEC/IEEE 16326.

4. Compliance & Documentation

- **Traceability Matrix**: Link requirements (e.g., "version control") to test cases and code modules.
- **Key Documents**:
  - **SRS**: Lists functional/non-functional requirements.
  - **Test Reports**: Include JUnit/Selenium results and UAT feedback.
  - **User Manuals**: Step-by-step guides for students/supervisors.

5. Post-Deployment Support

- **Feedback Portal**: Java-based form for users to report issues or suggestions.

7. Supporting Process Plans

5. Project Planning

This section describes a complete plan for creating and implementation of the project. We will discuss various stages involved in the planning, which includes work activities, schedule allocation and resource allocation.

5.1 Work Activity

Each sprint involves key activities to ensure successful delivery:

1. **Sprint Planning**: Define tasks and goals for the sprint.
2. **Analysis**: Gather requirements for the sprint.
3. **Design**: Create solutions for the tasks before development.
4. **Development**: Build the features or functionalities.
5. **Testing**: Test the deliverables to ensure quality.

Reviews and Feedback:

Regular reviews and feedback sessions are key to ensuring continuous improvement and aligning the project with stakeholders' expectations. These include:

- **Sprint Review**: Conducted at the end of each sprint to present the completed work, gather feedback, and ensure alignment with project goals.
- **Stakeholder Feedback**: Ongoing collection of input from stakeholders to refine project deliverables and ensure the system meets their needs.
- **After-Action Review**: A retrospective meeting to reflect on what went well, identify challenges, and discuss areas for improvement in the next sprint or phase.

Milestones:

After completing each module, it should be delivered to the customer.

5.2 Schedule Allocation:

In this section, a proper time will be allocated to each activity based on its complexity and priority, ensuring that all tasks are completed within the set timeframe and that the project stays on schedule. A **Gantt chart** will be utilized to visualize task sequences, dependencies, and the overall project timeline, providing a clear overview of the project's progress. It will also allow the team to make adjustments if delays or issues arise.

5.3 Resource Allocation:

Resource allocation involves strategically assigning the necessary resources—such as team members, technology, and infrastructure to each task to ensure the project's smooth progress. Efficient resource allocation is key to minimizing risks, avoiding bottlenecks, and optimizing project performance.

Following are the required resources for developing a system:

1. **Human Resources**:

   Project Manager, Developers (Frontend, Backend), Designers, QA Engineers.

2. **Technical Resources**:

   Servers for hosting, development tools, databases.

3. **Tools**:

- Version control (GitHub/Bitbucket).
- Project management tools (Trello/Jira).
- Communication tools (Slack, Zoom).

8. Supporting Process Plans

8.1 Risk Management

The Risk Management Plan systematically addresses risks through identification, analysis, prioritization, and mitigation to ensure project success and compliance with ISO/IEC/IEEE 16326.

Risk Identification

Risks are identified via:
✓ **Team brainstorming sessions** to anticipate challenges.
✓ **Stakeholder consultations** (students, supervisors, IT department).
✓ **Historical data review** from similar academic projects.

Risk Categories:

1. **Technological Risks**:
    - Integration issues with **Java, MySQL, JDBC, or Git**.
    - Security flaws in authentication or data encryption.
    - Scalability challenges during peak usage.
2. **Scheduling & Budget Risks**:
    - Delays in critical phases (e.g., version control development, UAT).
    - Cost overruns due to unplanned tasks (e.g., feedback module refinement).
3. **Personnel Risks**:
    - Team member unavailability or skill gaps in **Spring Boot/JSP**.
    - Knowledge loss due to turnover.
4. **Complexity Risks**:
    - Challenges in multi-version submission workflows or feedback tracking.

Risk Analysis & Prioritization

Risks are prioritized as **High**, **Medium**, or **Low** based on likelihood and impact:

- **High-Priority Risks**:
    - MySQL-Java integration failures.

    - Critical security vulnerabilities (e.g., unauthorized data access).
- **Medium-Priority Risks**:
    - Minor UI/UX delays or stakeholder feedback delays.
- **Low-Priority Risks**:
    - Cosmetic bugs (e.g., formatting issues in dashboards).

Risk Mitigation & Contingency Planning

Technological Risks:

- Conduct regular code reviews and **automated testing** (JUnit/Selenium).
- Implement **Spring Security** for secure authentication and MySQL encryption.

Scheduling Risks:

- Track progress via **ProjectLibre** and maintain a 1-week schedule buffer.
- Relocate tasks during sprint meetings if delays occur.

Personnel Risks:

- **Crosstrain** team members on critical modules (e.g., feedback logic).
- Document workflows in **GitHub Wiki** to preserve knowledge.

Complexity Risks:

- Break complex tasks (e.g., version control) into smaller sprints.
- Use **modular Java development** (e.g., separate Submission Service and Feedback Service).

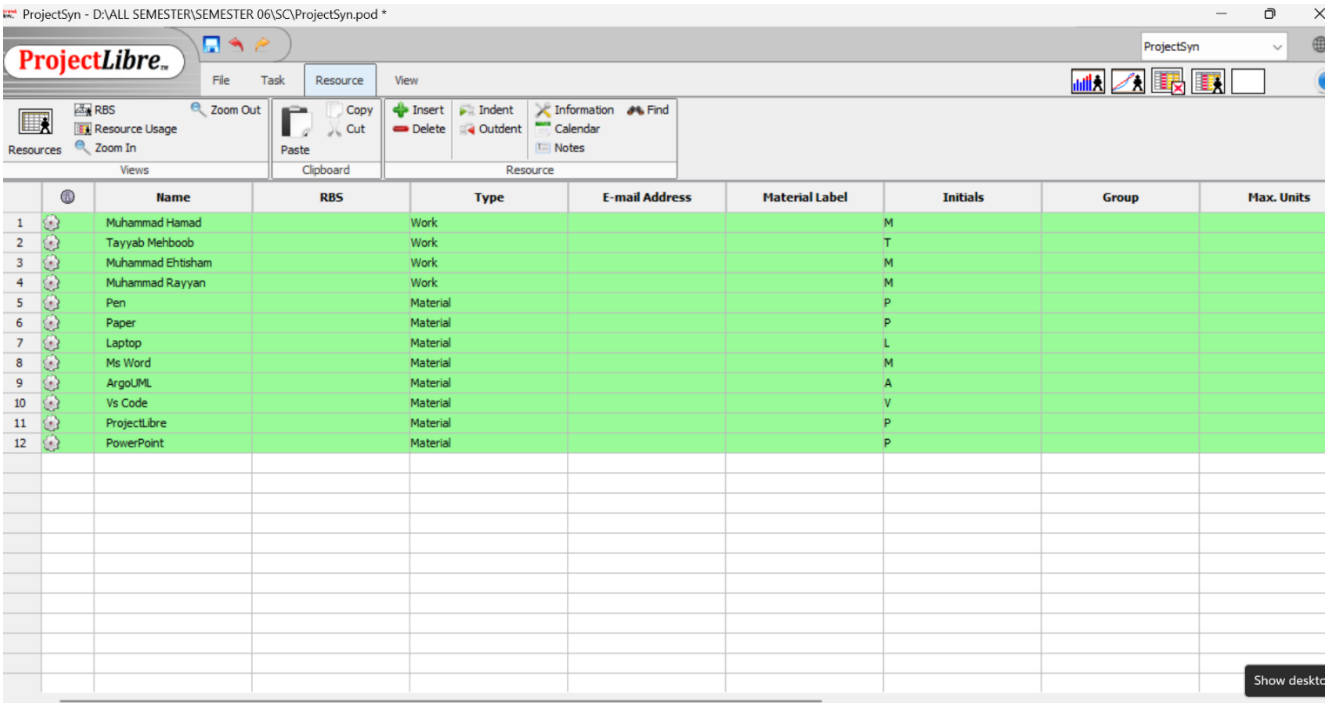
Risk Tracking & Evaluation

✓ **Risk Register**: Maintain a log of all risks, mitigation actions, and status updates.
✓ **Weekly Reviews**: Assess risks during team meetings and adjust priorities.
✓ **Tool Integration**: Monitor code progress via **GitHub** and timelines via **ProjectLibre**.

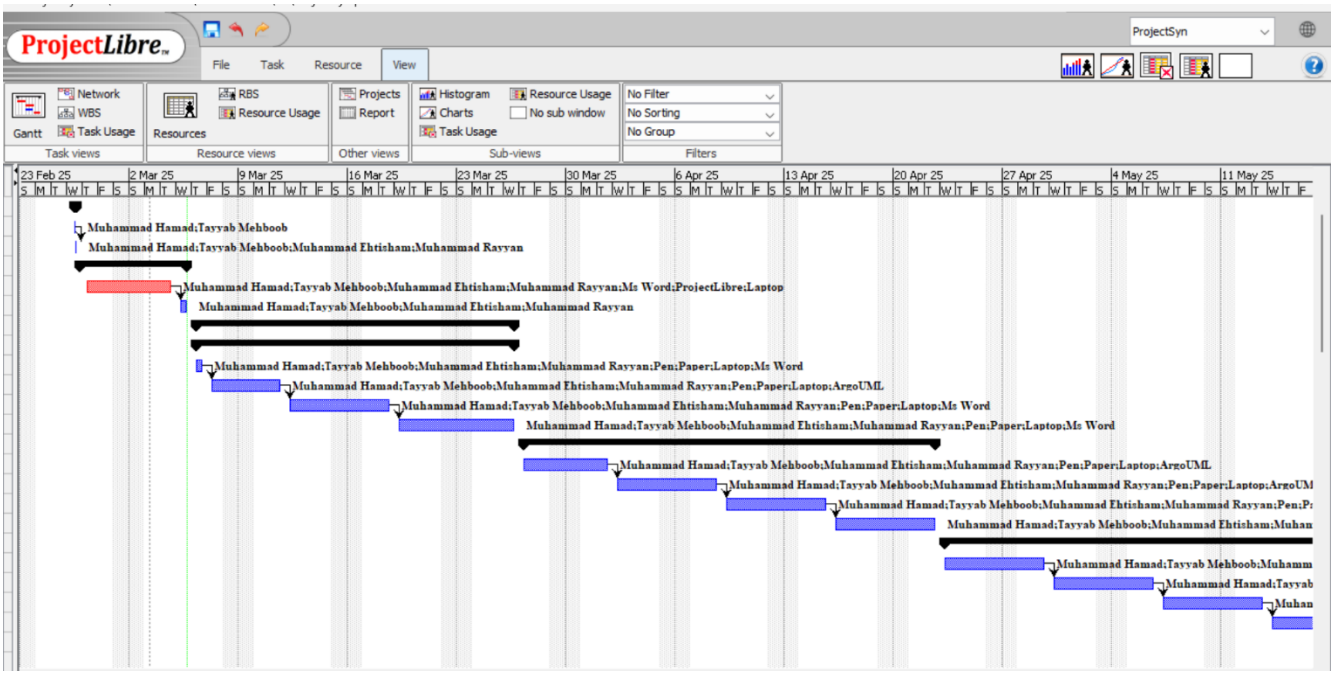
Risk Communication

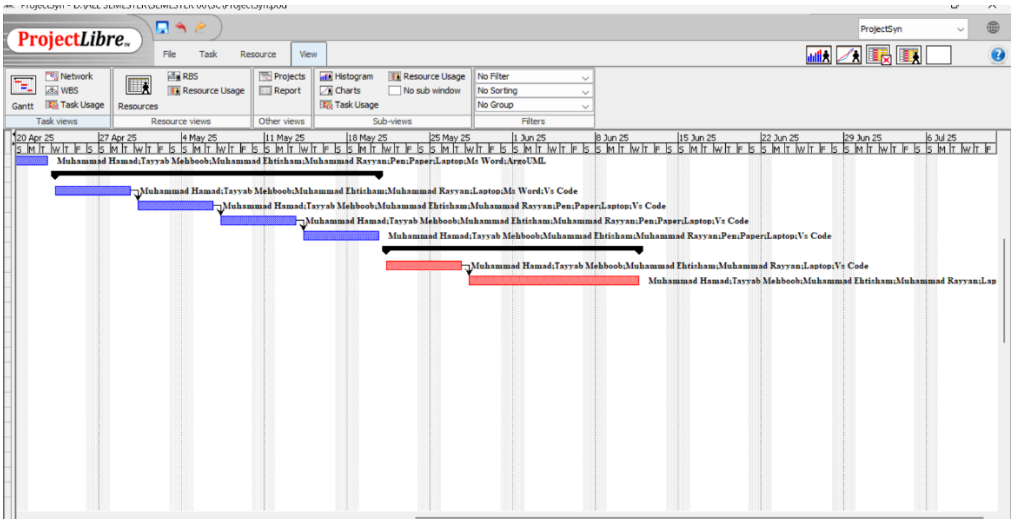
The Project Leader will:
✓ Share **weekly risk reports** in stakeholder meetings.
✓ Send real-time alerts for high-priority risks via **email/MS Teams**.
✓ Update the **Risk Register** for transparency across the team.

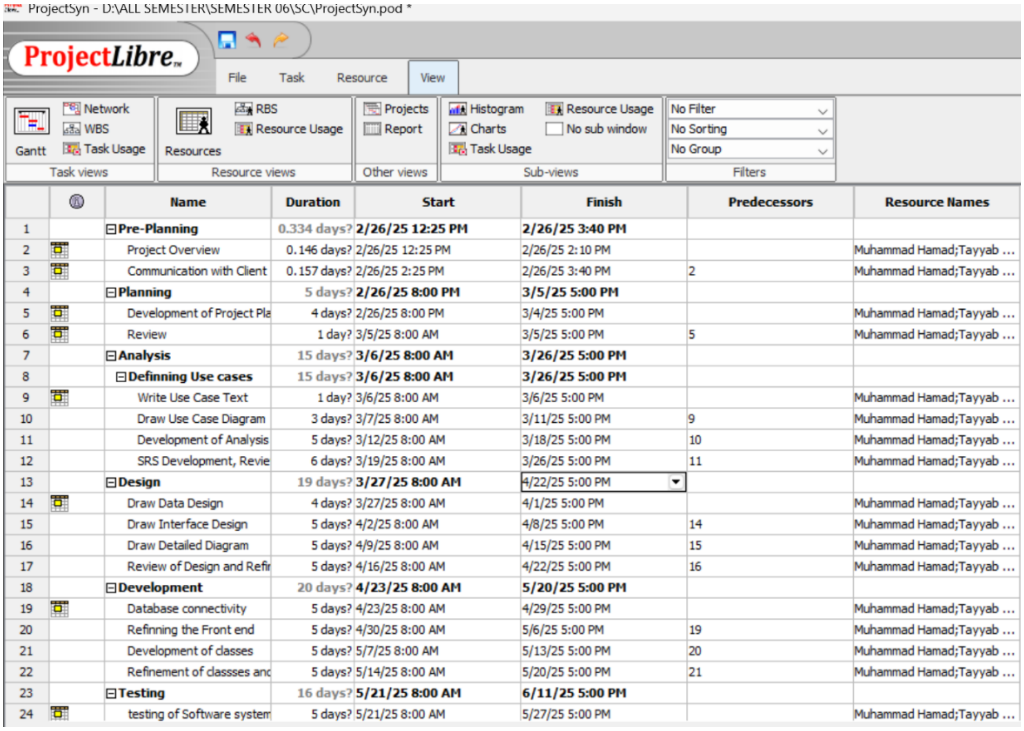
- **Ghant Char/ Task Network**



	Name	RBS	Type	E-mail Address	Material Label	Initials	Group	Max. Units
1	Muhammad Hamad		Work			M		
2	Tayyab Mehboob		Work			T		
3	Muhammad Ehtisham		Work			M		
4	Muhammad Rayyan		Work			M		
5	Pen		Material			P		
6	Paper		Material			P		
7	Laptop		Material			L		
8	Ms Word		Material			M		
9	ArgoUML		Material			A		
10	Vs Code		Material			V		
11	ProjectLibre		Material			P		
12	PowerPoint		Material			P		







ProjectSyn - D:\ALL SEMESTER\SEMESTER 06\SC\ProjectSyn.pod *

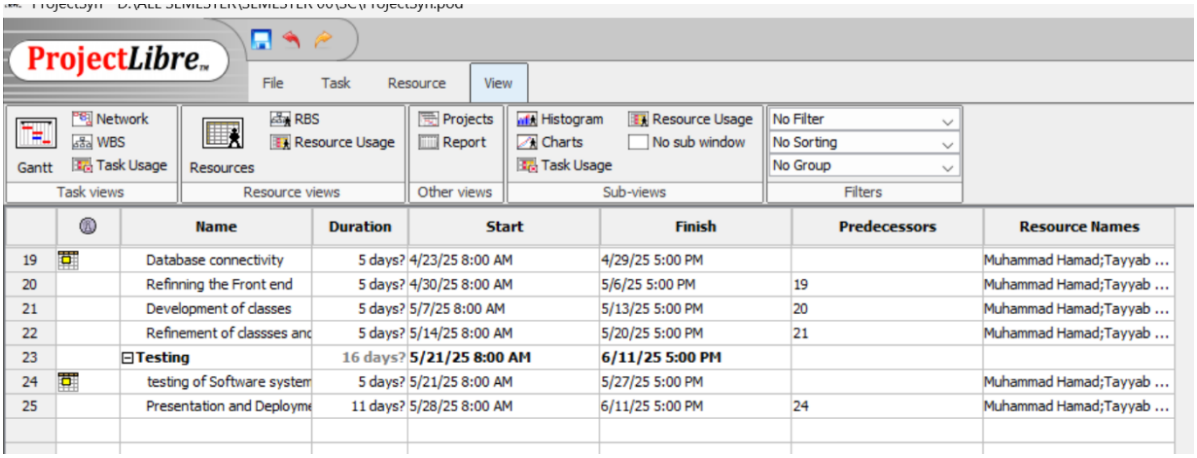
ProjectLibre

File Task Resource View

Gantt Network WBS Task Usage | Resources RBS Resource Usage | Projects Report | Histogram Charts Task Usage Resource Usage No sub window | No Filter No Sorting No Group

Task views Resource views Other views Sub-views Filters

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Pre-Planning**	0.334 days?	**2/26/25 12:25 PM**	**2/26/25 3:40 PM**		
2		Project Overview	0.146 days?	2/26/25 12:25 PM	2/26/25 2:10 PM		Muhammad Hamad;Tayyab ...
3		Communication with Client	0.157 days?	2/26/25 2:25 PM	2/26/25 3:40 PM	2	Muhammad Hamad;Tayyab ...
4		**Planning**	5 days?	**2/26/25 8:00 PM**	**3/5/25 5:00 PM**		
5		Development of Project Pla	4 days?	2/26/25 8:00 PM	3/4/25 5:00 PM		Muhammad Hamad;Tayyab ...
6		Review	1 day?	3/5/25 8:00 AM	3/5/25 5:00 PM	5	Muhammad Hamad;Tayyab ...
7		**Analysis**	15 days?	**3/6/25 8:00 AM**	**3/26/25 5:00 PM**		
8		**Defenning Use cases**	15 days?	**3/6/25 8:00 AM**	**3/26/25 5:00 PM**		
9		Write Use Case Text	1 day?	3/6/25 8:00 AM	3/6/25 5:00 PM		Muhammad Hamad;Tayyab ...
10		Draw Use Case Diagram	3 days?	3/7/25 8:00 AM	3/11/25 5:00 PM	9	Muhammad Hamad;Tayyab ...
11		Development of Analysis	5 days?	3/12/25 8:00 AM	3/18/25 5:00 PM	10	Muhammad Hamad;Tayyab ...
12		SRS Development, Revie	6 days?	3/19/25 8:00 AM	3/26/25 5:00 PM	11	Muhammad Hamad;Tayyab ...
13		**Design**	19 days?	**3/27/25 8:00 AM**	4/22/25 5:00 PM		
14		Draw Data Design	4 days?	3/27/25 8:00 AM	4/1/25 5:00 PM		Muhammad Hamad;Tayyab ...
15		Draw Interface Design	5 days?	4/2/25 8:00 AM	4/8/25 5:00 PM	14	Muhammad Hamad;Tayyab ...
16		Draw Detailed Diagram	5 days?	4/9/25 8:00 AM	4/15/25 5:00 PM	15	Muhammad Hamad;Tayyab ...
17		Review of Design and Refi	5 days?	4/16/25 8:00 AM	4/22/25 5:00 PM	16	Muhammad Hamad;Tayyab ...
18		**Development**	20 days?	**4/23/25 8:00 AM**	**5/20/25 5:00 PM**		
19		Database connectivity	5 days?	4/23/25 8:00 AM	4/29/25 5:00 PM		Muhammad Hamad;Tayyab ...
20		Refinning the Front end	5 days?	4/30/25 8:00 AM	5/6/25 5:00 PM	19	Muhammad Hamad;Tayyab ...
21		Development of classes	5 days?	5/7/25 8:00 AM	5/13/25 5:00 PM	20	Muhammad Hamad;Tayyab ...
22		Refinement of classses anc	5 days?	5/14/25 8:00 AM	5/20/25 5:00 PM	21	Muhammad Hamad;Tayyab ...
23		**Testing**	16 days?	**5/21/25 8:00 AM**	**6/11/25 5:00 PM**		
24		testing of Software system	5 days?	5/21/25 8:00 AM	5/27/25 5:00 PM		Muhammad Hamad;Tayyab ...



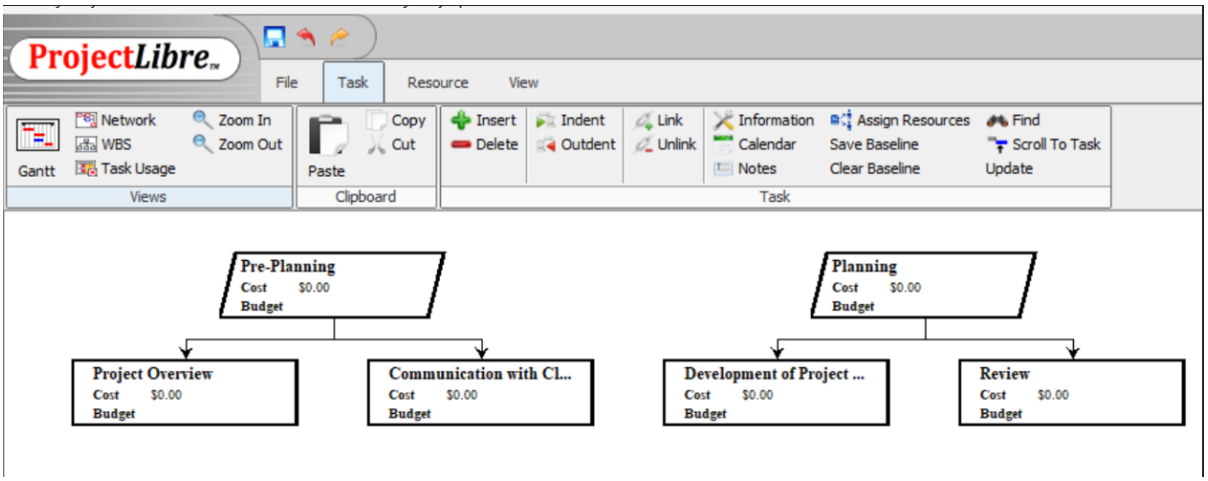
ProjectLibre

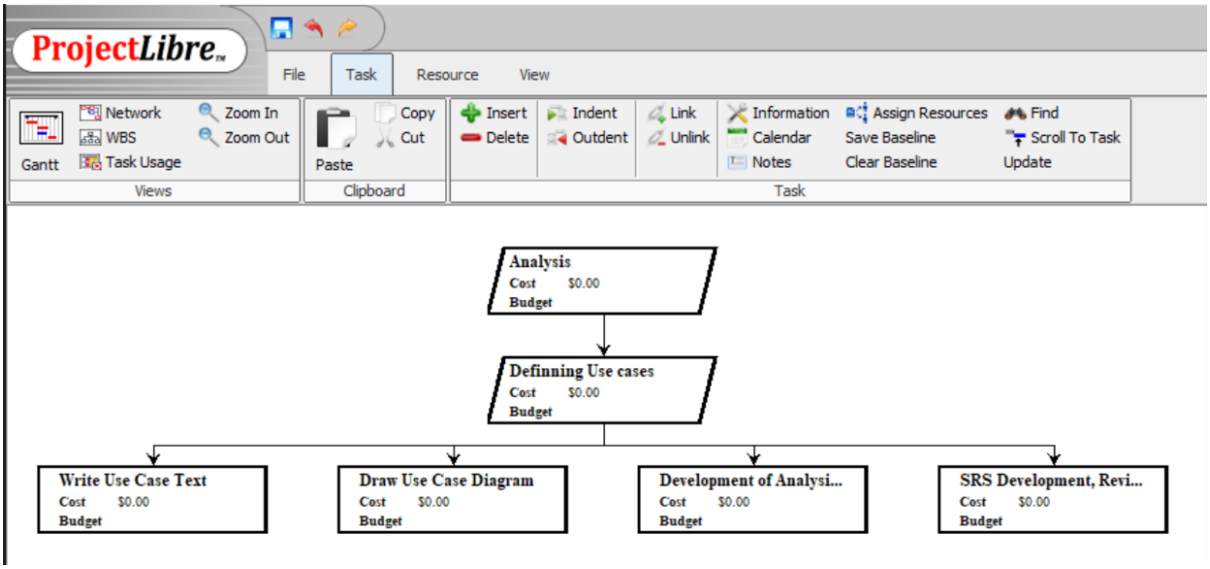
File Task Resource View

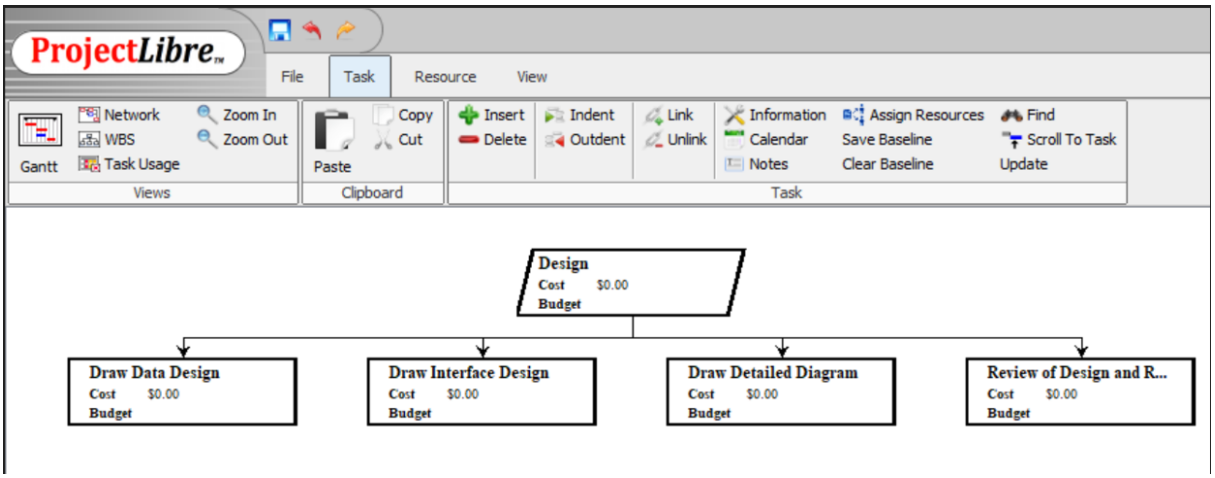
Gantt Network WBS Task Usage | Resources RBS Resource Usage | Projects Report | Histogram Charts Task Usage Resource Usage No sub window | No Filter No Sorting No Group

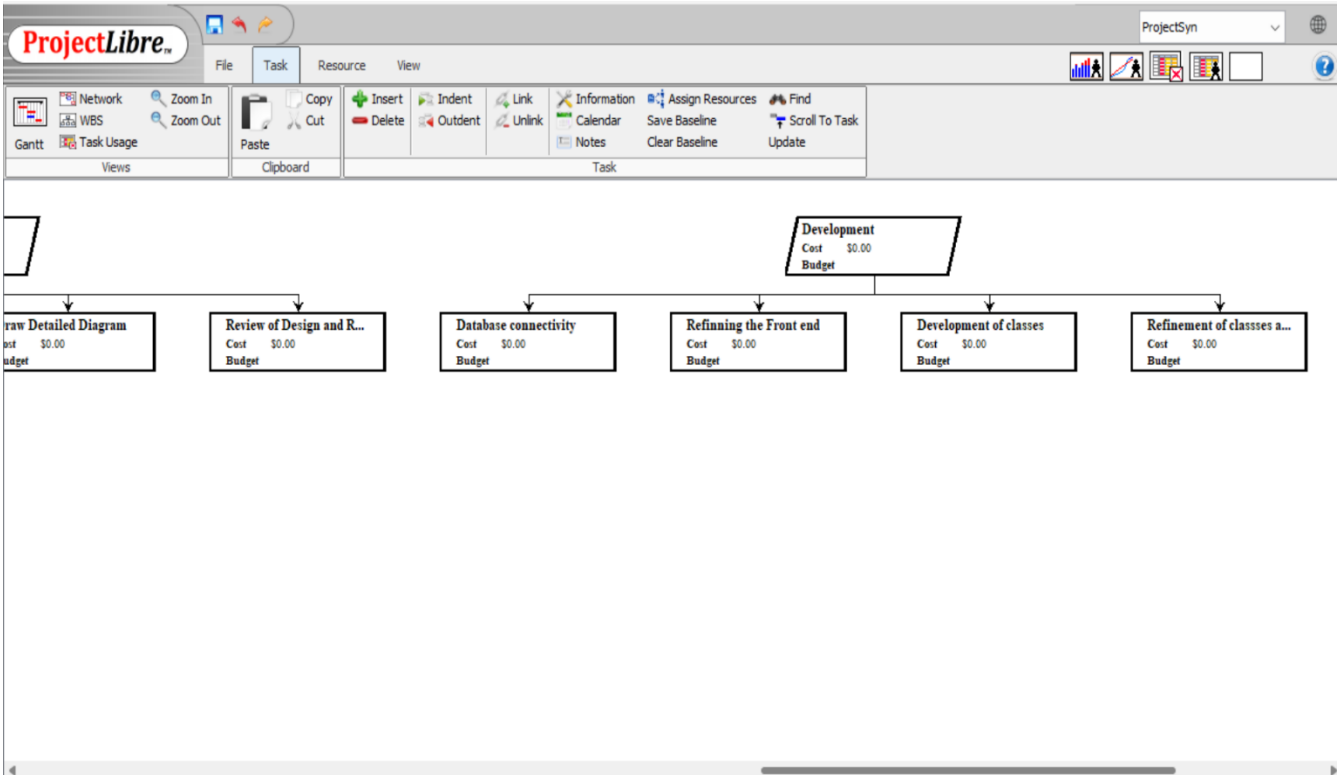
Task views Resource views Other views Sub-views Filters

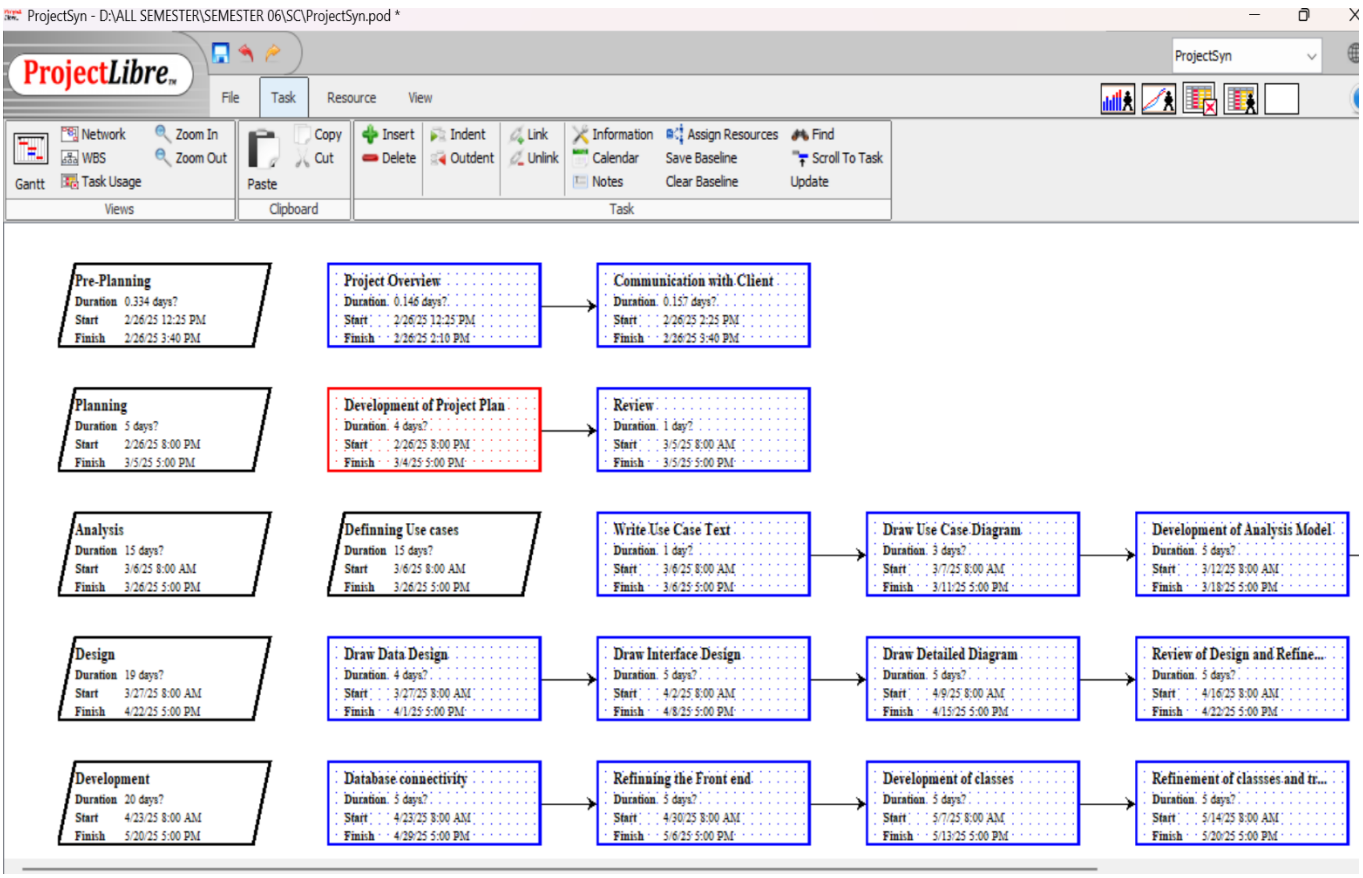
		Name	Duration	Start	Finish	Predecessors	Resource Names
19		Database connectivity	5 days?	4/23/25 8:00 AM	4/29/25 5:00 PM		Muhammad Hamad;Tayyab ...
20		Refinning the Front end	5 days?	4/30/25 8:00 AM	5/6/25 5:00 PM	19	Muhammad Hamad;Tayyab ...
21		Development of classes	5 days?	5/7/25 8:00 AM	5/13/25 5:00 PM	20	Muhammad Hamad;Tayyab ...
22		Refinement of classses anc	5 days?	5/14/25 8:00 AM	5/20/25 5:00 PM	21	Muhammad Hamad;Tayyab ...
23		**Testing**	16 days?	**5/21/25 8:00 AM**	**6/11/25 5:00 PM**		
24		testing of Software system	5 days?	5/21/25 8:00 AM	5/27/25 5:00 PM		Muhammad Hamad;Tayyab ...
25		Presentation and Deployme	11 days?	5/28/25 8:00 AM	6/11/25 5:00 PM	24	Muhammad Hamad;Tayyab ...

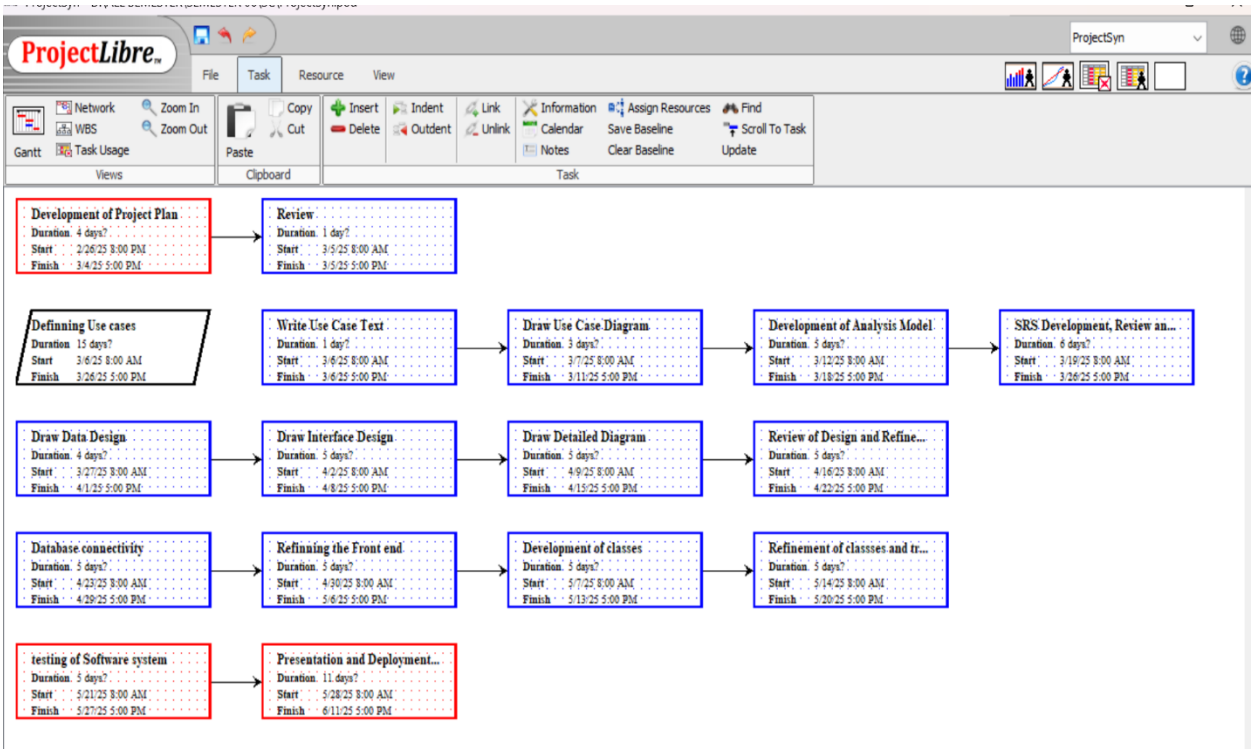

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Copy
Cut
Paste
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Pre-Planning
Cost $0.00
Budget
Planning
Cost $0.00
Budget
Project Overview
Cost $0.00
Budget
Communication with Cl...
Cost $0.00
Budget
Development of Project ...
Cost $0.00
Budget
Review
Cost $0.00
Budget

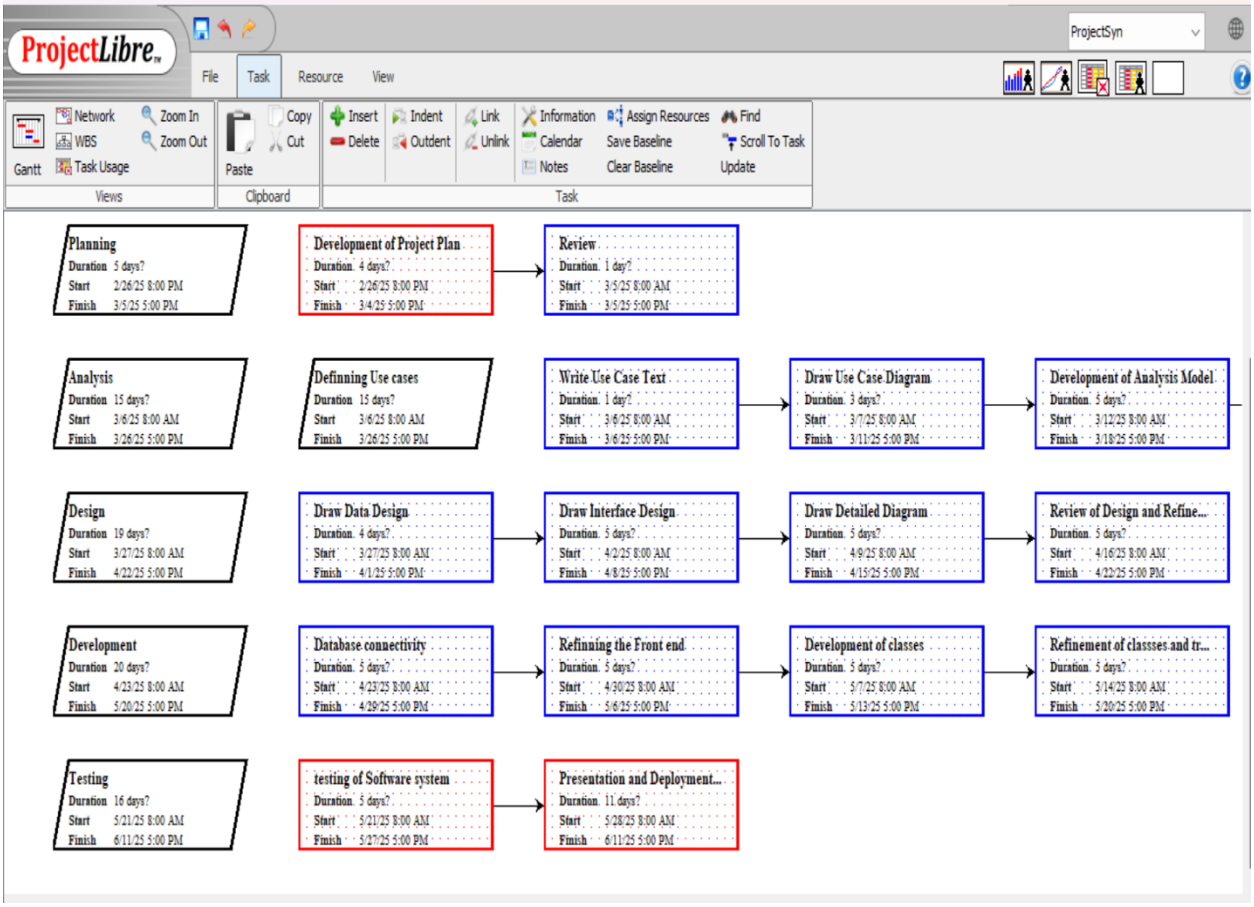

ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Copy
Cut
Paste
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Analysis
Cost $0.00
Budget
Definning Use cases
Cost $0.00
Budget
Write Use Case Text
Cost $0.00
Budget
Draw Use Case Diagram
Cost $0.00
Budget
Development of Analysi...
Cost $0.00
Budget
SRS Development, Revi...
Cost $0.00
Budget


ProjectLibre
File
Task
Resource
View
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Copy
Cut
Paste
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Design
Cost $0.00
Budget
Draw Data Design
Cost $0.00
Budget
Draw Interface Design
Cost $0.00
Budget
Draw Detailed Diagram
Cost $0.00
Budget
Review of Design and R...
Cost $0.00
Budget


ProjectLibre
File
Task
Resource
View
ProjectSyn
Network
WBS
Task Usage
Zoom In
Zoom Out
Gantt
Views
Paste
Copy
Cut
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Development
Cost $0.00
Budget
raw Detailed Diagram
Review of Design and R...
Database connectivity
Refinning the Front end
Development of classes
Refinement of classses a...


ProjectSyn - D:\ALL SEMESTER\SEMESTER 06\SC\ProjectSyn.pod *
ProjectLibre
File
Task
Resource
View
ProjectSyn
Gantt
Views
Paste
Clipboard
Task
Pre-Planning
Duration 0.334 days?
Start 2/26/25 12:25 PM
Finish 2/26/25 3:40 PM
Project Overview
Duration 0.146 days?
Start 2/26/25 12:25 PM
Finish 2/26/25 2:10 PM
Communication with Client
Duration 0.157 days?
Start 2/26/25 2:25 PM
Finish 2/26/25 3:40 PM
Planning
Duration 5 days?
Start 2/26/25 8:00 PM
Finish 3/5/25 5:00 PM
Development of Project Plan
Duration 4 days?
Start 2/26/25 8:00 PM
Finish 3/4/25 5:00 PM
Review
Duration 1 day?
Start 3/5/25 8:00 AM
Finish 3/5/25 5:00 PM
Analysis
Duration 15 days?
Start 3/6/25 8:00 AM
Finish 3/26/25 5:00 PM
Definning Use cases
Duration 15 days?
Start 3/6/25 8:00 AM
Finish 3/26/25 5:00 PM
Write Use Case Text
Duration 1 day?
Start 3/6/25 8:00 AM
Finish 3/6/25 5:00 PM
Draw Use Case Diagram
Duration 3 days?
Start 3/7/25 8:00 AM
Finish 3/11/25 5:00 PM
Development of Analysis Model
Duration 5 days?
Start 3/12/25 8:00 AM
Finish 3/18/25 5:00 PM
Design
Duration 19 days?
Start 3/27/25 8:00 AM
Finish 4/22/25 5:00 PM
Draw Data Design
Duration 4 days?
Start 3/27/25 8:00 AM
Finish 4/1/25 5:00 PM
Draw Interface Design
Duration 5 days?
Start 4/2/25 8:00 AM
Finish 4/8/25 5:00 PM
Draw Detailed Diagram
Duration 5 days?
Start 4/9/25 8:00 AM
Finish 4/15/25 5:00 PM
Review of Design and Refine...
Duration 5 days?
Start 4/16/25 8:00 AM
Finish 4/22/25 5:00 PM
Development
Duration 20 days?
Start 4/23/25 8:00 AM
Finish 5/20/25 5:00 PM
Database connectivity
Duration 5 days?
Start 4/23/25 8:00 AM
Finish 4/29/25 5:00 PM
Refinning the Front end
Duration 5 days?
Start 4/30/25 8:00 AM
Finish 5/6/25 5:00 PM
Development of classes
Duration 5 days?
Start 5/7/25 8:00 AM
Finish 5/13/25 5:00 PM
Refinement of classses and tr...
Duration 5 days?
Start 5/14/25 8:00 AM
Finish 5/20/25 5:00 PM


ProjectLibre
File
Task
Resource
View
ProjectSyn
Network
Zoom In
WBS
Zoom Out
Task Usage
Gantt
Views
Paste
Copy
Cut
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Development of Project Plan
Duration 4 days?
Start 2/26/25 8:00 PM
Finish 3/4/25 5:00 PM
Review
Duration 1 day?
Start 3/5/25 8:00 AM
Finish 3/5/25 5:00 PM
Definning Use cases
Duration 15 days?
Start 3/6/25 8:00 AM
Finish 3/26/25 5:00 PM
Write Use Case Text
Duration 1 day?
Start 3/6/25 8:00 AM
Finish 3/6/25 5:00 PM
Draw Use Case Diagram
Duration 3 days?
Start 3/7/25 8:00 AM
Finish 3/11/25 5:00 PM
Development of Analysis Model
Duration 5 days?
Start 3/12/25 8:00 AM
Finish 3/18/25 5:00 PM
SRS Development, Review an...
Duration 6 days?
Start 3/19/25 8:00 AM
Finish 3/26/25 5:00 PM
Draw Data Design
Duration 4 days?
Start 3/27/25 8:00 AM
Finish 4/1/25 5:00 PM
Draw Interface Design
Duration 5 days?
Start 4/2/25 8:00 AM
Finish 4/8/25 5:00 PM
Draw Detailed Diagram
Duration 5 days?
Start 4/9/25 8:00 AM
Finish 4/15/25 5:00 PM
Review of Design and Refine...
Duration 5 days?
Start 4/16/25 8:00 AM
Finish 4/22/25 5:00 PM
Database connectivity
Duration 5 days?
Start 4/23/25 8:00 AM
Finish 4/29/25 5:00 PM
Refinning the Front end
Duration 5 days?
Start 4/30/25 8:00 AM
Finish 5/6/25 5:00 PM
Development of classes
Duration 5 days?
Start 5/7/25 8:00 AM
Finish 5/13/25 5:00 PM
Refinement of classses and tr...
Duration 5 days?
Start 5/14/25 8:00 AM
Finish 5/20/25 5:00 PM
testing of Software system
Duration 5 days?
Start 5/21/25 8:00 AM
Finish 5/27/25 5:00 PM
Presentation and Deployment...
Duration 11 days?
Start 5/28/25 8:00 AM
Finish 6/11/25 5:00 PM


ProjectLibre
File
Task
Resource
View
ProjectSyn
Network
Zoom In
WBS
Zoom Out
Task Usage
Gantt
Views
Paste
Copy
Cut
Clipboard
Insert
Delete
Indent
Outdent
Link
Unlink
Information
Calendar
Notes
Assign Resources
Save Baseline
Clear Baseline
Find
Scroll To Task
Update
Task
Planning
Duration 5 days?
Start 2/26/25 8:00 PM
Finish 3/5/25 5:00 PM
Development of Project Plan
Duration 4 days?
Start 2/26/25 8:00 PM
Finish 3/4/25 5:00 PM
Review
Duration 1 day?
Start 3/5/25 8:00 AM
Finish 3/5/25 5:00 PM
Analysis
Duration 15 days?
Start 3/6/25 8:00 AM
Finish 3/26/25 5:00 PM
Definning Use cases
Duration 15 days?
Start 3/6/25 8:00 AM
Finish 3/26/25 5:00 PM
Write Use Case Text
Duration 1 day?
Start 3/6/25 8:00 AM
Finish 3/6/25 5:00 PM
Draw Use Case Diagram
Duration 3 days?
Start 3/7/25 8:00 AM
Finish 3/11/25 5:00 PM
Development of Analysis Model
Duration 5 days?
Start 3/12/25 8:00 AM
Finish 3/18/25 5:00 PM
Design
Duration 19 days?
Start 3/27/25 8:00 AM
Finish 4/22/25 5:00 PM
Draw Data Design
Duration 4 days?
Start 3/27/25 8:00 AM
Finish 4/1/25 5:00 PM
Draw Interface Design
Duration 5 days?
Start 4/2/25 8:00 AM
Finish 4/8/25 5:00 PM
Draw Detailed Diagram
Duration 5 days?
Start 4/9/25 8:00 AM
Finish 4/15/25 5:00 PM
Review of Design and Refine...
Duration 5 days?
Start 4/16/25 8:00 AM
Finish 4/22/25 5:00 PM
Development
Duration 20 days?
Start 4/23/25 8:00 AM
Finish 5/20/25 5:00 PM
Database connectivity
Duration 5 days?
Start 4/23/25 8:00 AM
Finish 4/29/25 5:00 PM
Refinning the Front end
Duration 5 days?
Start 4/30/25 8:00 AM
Finish 5/6/25 5:00 PM
Development of classes
Duration 5 days?
Start 5/7/25 8:00 AM
Finish 5/13/25 5:00 PM
Refinement of classses and tr...
Duration 5 days?
Start 5/14/25 8:00 AM
Finish 5/20/25 5:00 PM
Testing
Duration 16 days?
Start 5/21/25 8:00 AM
Finish 6/11/25 5:00 PM
testing of Software system
Duration 5 days?
Start 5/21/25 8:00 AM
Finish 5/27/25 5:00 PM
Presentation and Deployment...
Duration 11 days?
Start 5/28/25 8:00 AM
Finish 6/11/25 5:00 PM