PROJECT PLAN

Event Hub

PRESENTED BY

MEMBER 1: AHMAD MASOOD (LEADER)

MEMBER 2: M. AHMAD HASSAN

MEMBER 3: M. WALEED

MEMBER 4: TALHA WAINS

GROUP 15

SUPERVISED BY

Dr. ONAIZA MAQBOOL

Department of Computer Science

Quaid e Azam University, Islamabad

PROJECT PLAN DOCUMENT APPROVEL SIGNATURES

SIGNATURES:

MEMBER 1(LEADER):

Section 5

Ahmad Masood

_____

MEMBER 2:

Section 1-3

M. Ahmad Hassan

_____

MEMBER 3:

Section 4

Talha Arif Wains

_____

MEMBER 4:

Section 8

Muhammad Waleed

_____

STAKEHOLDER:

DR. ONAIZA MAQBOOL

Professor, QAU Islamabad

CHANGE HISTORY:

VERSION NO.	REVISION DATE	MODIFIED BY	CHANGES	APPROVED BY
1.0	03/03/2025	Ahmad Masood	Original	DR. ONAIZA MAQBOOL

REMARKS:

_____

_____

Signature _____

DR. ONAIZA MAQBOOL

Professor, QAU Islamabad

TABLE OF CONTENTS

List of figures

List of tables

1.Project overview

1.1 Project Summary

The department aims to create an Event Management System to promote the organization, registration, and participation of students for departmental and outside events. The system will enable event postings, student registration, and post-event posting of pictures, ratings, and reviews. The site will automate event monitoring, increase student interaction, and have a structured record of departmental events.

1.2 Project Purpose, Scope and Objectives

Purpose:

The purpose of this project is to design an efficient and user-friendly system that manages student participation in events that are organized by the department or external organizations. The system should automate event registration, information sharing, and feedback collection to make the experience of students and faculty smooth and engaging.

Scope:

>Context:

This is a standalone system managing the event registration process and overseeing the scheduling of student participation in events.

>Functional requirements:

Event Management: External organizers and departments can make, edit, and promote events.

Student Registration: Eligible students can register for events according to eligibility.

Event Updates: Post-event photos and updates can be posted for documentation.

Event Rating & Feedback: Students and teachers can rate and review events.

User Administration: Various roles (admin, faculty, students) will possess various levels of access.

Notification System: Computerized alerts for future occurrences.

Objectives:

Create a centralized system to manage and track departmental events efficiently.	✓
Enable students to easily register for events and allow administrators to monitor participation.	✓
Provide a mechanism for students and teachers to rate events and give feedback for future improvements.	✓
Ensure secure access by defining user roles (admin, student, teacher) with appropriate permissions.	✓

Design the platform to handle multiple users efficiently while maintaining data security.	✓

1.3 Assumptions and Constraints

Constraints:

The software will be built using Java programming language.
User authentication is necessary for event registration and submission of feedback.
The system must support high concurrent users in the event registration process.
The project must be done within the provided semester timeline.

Assumptions:

The system can only be accessed by students and staff of the department.
Events will have pre-defined eligibility requirements (e.g., department, year, batch).
The system will support image uploads for event documentation.
Students and professors can give ratings and feedback after the event.

1.3 Project Deliverables

DELIVERABLE	DESCRIPTION
Event Management System (EMS)	Fully functional event registration and management system.
User Interface (UI)	Web-based interface for students, faculty, and admins.
Database Schema	Structured database for storing events and user information.
Post-Event Features	Image uploads, event ratings, and feedback submission.
Reports & Logs	Event participation statistics and user activity logs.

Documentation	User manual, system design document, and test reports.

1.3 Schedule Summary

PHASE	PROCESS	DURATION (weeks)
1	Requirement	Done
2	Planning	Done
3	Analysis	2
4	Coding	4
5	Testing	3
6	Deployment	2

2.References

1. IEEE Software Construction Guideline. ISO/IEC/IEEE   16326:2019(E)
2. Project Libre manual version 0.1-October 6,2012.

3.Definitions

Event Management System (EMS): A software system designed to manage the creation, registration, and participation in events.

Event Announcement: The process of publishing event details such as date, time, location, and eligibility criteria.

Student Registration: The process by which students sign up for an event based on predefined requirements.

Event Rating: A feedback mechanism allowing students and faculty to rate and review completed events.

Admin: A user responsible for managing event approvals, monitoring registrations, and maintaining the system.

Faculty: A user role that can create events, approve participants, and provide feedback.

Student: A user role that can register for events, participate, and submit feedback.

External Events: Events hosted outside the department but open for student participation.

4.Project Context

4.1 Process Model

Since requirements might evolve (e.g., new event types, different registration rules), **Agile** allows iterative development.

Continuous user feedback (from students and faculty) can help improve the system.

Features can be prioritized and delivered in sprints.

Why:

Flexibility: Can adapt to changing requirements.

Early Delivery: Usable features are delivered quickly.

User Involvement: Continuous feedback ensures the system meets real needs.

Better Quality: Frequent testing reduces errors and improves performance.

4.2 Methods, Tools and Techniques

Methods:

Agile Methodology for managing development through sprints.

Object Oriented Programming (OOP) for managing software architecture.

Tools & Techniques:

Java as the Programming language.

MS Word for the documentation of our project.

Project Libre for measuring the progress of our project.

Git and GitHub to keep crucial versions of our project.

Firebase as database management to store Event description and Registered Users data.

Apache that will work on server side and used as a deployer for deploying Java Applications to end-users.

4.3 Product Acceptance Plan

Acceptance Criteria:

The system will be considered acceptable when it meets the following criteria:

Successfully allows event creation, modification, deletion and announcements by authorized users.
Enables users to register and receive confirmation and vice-versa.
Supports post-event feedback submission.
Ensures a secure login system with role-based access control.

5.Project Planning

This section describes the project's general management plan. Important points in planning are:

>Process Model (Agile model).

5.1 Project Work Plan.

The project will be executed in **phases**, with each phase focusing on specific functionalities.

>Project phases:

Our project consists of these main phases. They will also be the deliverables:

- Planning
- Analysis
- Design
- Coding
- Testing
- Deployment

5.2 Work activities/tasks in every sprint:

Requirement Gathering: Understanding user needs, defining system features.

System Design: Designing UI, database, and system architecture.

Implementation: Developing core functionalities like event creation, deletion, registration, and feedback.

Testing & Debugging: Ensuring the system is error-free and meets requirements.

5.3 Schedule Allocation:

It includes start and finish dates, dependencies and milestones. A Gantt chart will be used to see task sequences and check the progress. Here time will be given to every activity.

5.4 Resource Allocation:

Resources will be allocated to every phase of our project life cycle.

The screenshot of the projectLibre file is attached at the end of the document.

8.Supporting Process Plans

8.1 Risk management:

The plan for Risk Management uses the approach for Identifying, Analyzing, Prioritizing and Mitigating the risks throughout the lifecycle of the project. By Addressing potential risk factors, our aim is to minimize their impact on project success.

Risk Identification:

The Project team will identify the risks that could impact on the project using the following methods.

Team sessions to discuss potential challenges that can occur.

Stakeholder meetings to gather input from external.

Review the data which is historical from similar projects for common risk patterns.

The following risk categories have been identified:
- o Technological Risks:
  - ▪ Issues in Integrating various technologies like not familiar with Java, GitHub, MySQL, Eclipse, JDBC or problems with platform scalability.
- o Risks in Scheduling and Budget:
  - ▪ Delays in development phases, like in analysis or especially in deployment or testing or cost overruns because of not analyzing one of the important tasks.
- o Personnel Risks:
  - ▪ Unavailability of the team member or turnover, leading to skill gaps or loss of key project knowledge.
- o Size and Complexity Risks:

▪ Complexity of the Project, especially in shortlisting algorithms, could introduce unanticipated challenges.

Risk Analysis and Prioritization:

Each Risk will be assessed for its probability of occurrence and its impact on the project, using a risk matrix to categorize risks as low, medium, or high priority.

<u>High-priority risks:</u>
- o (e.g., integration failure or major security vulnerabilities) will be addressed immediately.

<u>Medium-priority risks:</u>
- o (e.g., minor delays or user integration feedback) will be monitored closely with contingency plans in place.

<u>Low-priority risks:</u>
 (e.g., minor bugs or less critical delays) will be tracked and mitigated if necessary.

Mitigation of Risks and Contingency Planning:

For each risk that is identified, the following strategies will be used:

<u>Technological Risks:</u>
- o Regular reviews of code, testing must be conducted automatically, and robust security measures will be implemented.

<u>Risks in Scheduling and Budget:</u>
- O Project Libre will be used to track Milestones. Any delays will trigger immediate review and reallocation of resources.

<u>Personnel Risks:</u>
- O Team members will be doing cross-training to play multiple roles and by developing clear documentation, it will minimize the impact of team changes.

<u>Complexity Risks:</u>
- O To break down Complex tasks into manageable parts, a phased, modular development approach will be used, that will reduce the overall risks of system-wide issues.

Tracking Risks and its Evaluation:

Throughout the Project Lifecycle, Risks will be continuously monitored and reassessed, Our Team will:

Maintain a Risk Register to check all Identified Risks, Mitigation Strategies, and their Status.

Conduct Weekly Risk Review Meetings to evaluate Risk Levels.

Use of GitHub and Project Libre will be done to track progress and identify emerging risks.

Risks Communication:

The manager of the Project will be responsible for communicating the status of Risks to the project stakeholders. It will include:
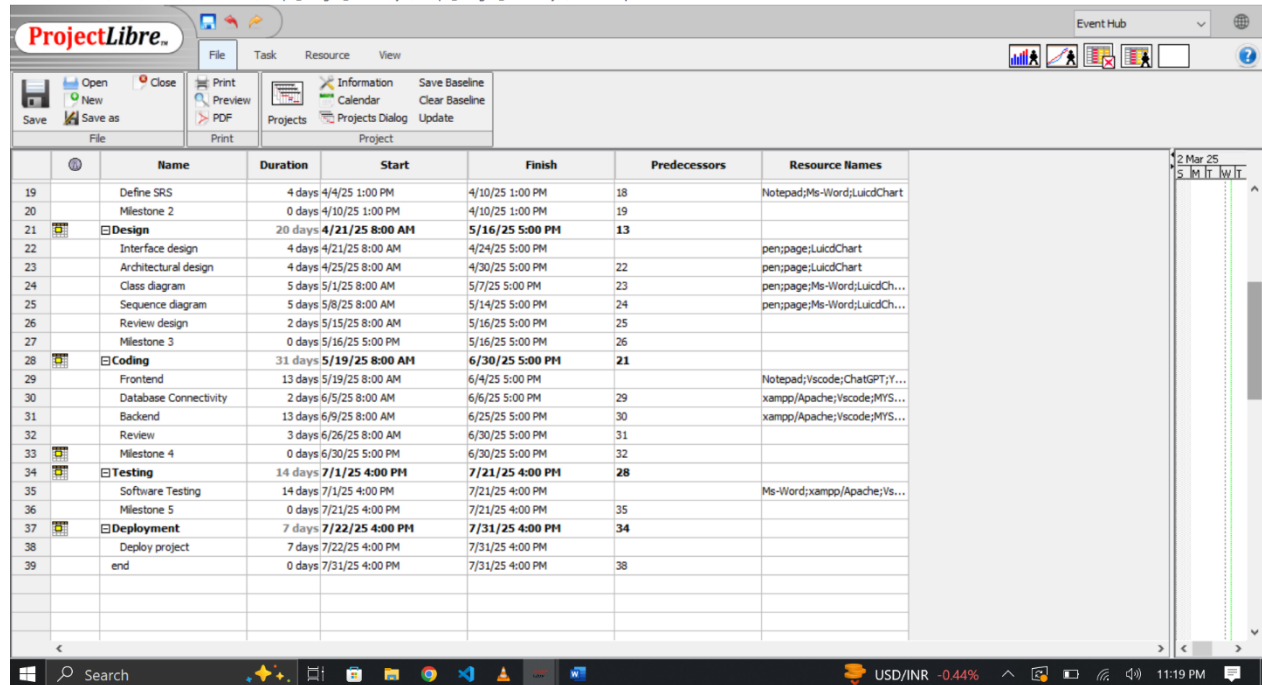
Providing Weekly Reports during team and stakeholders meetings.

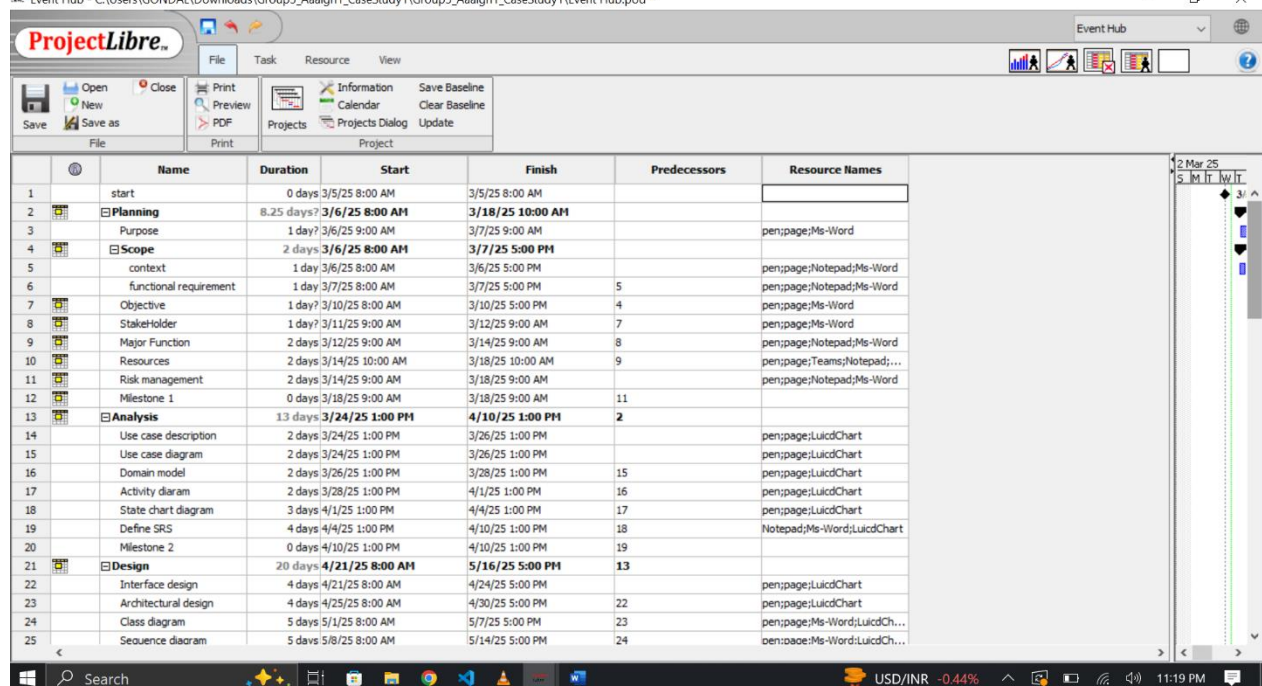Sharing real-time updates on critical risks via Project Communication channels (e.g., email address).
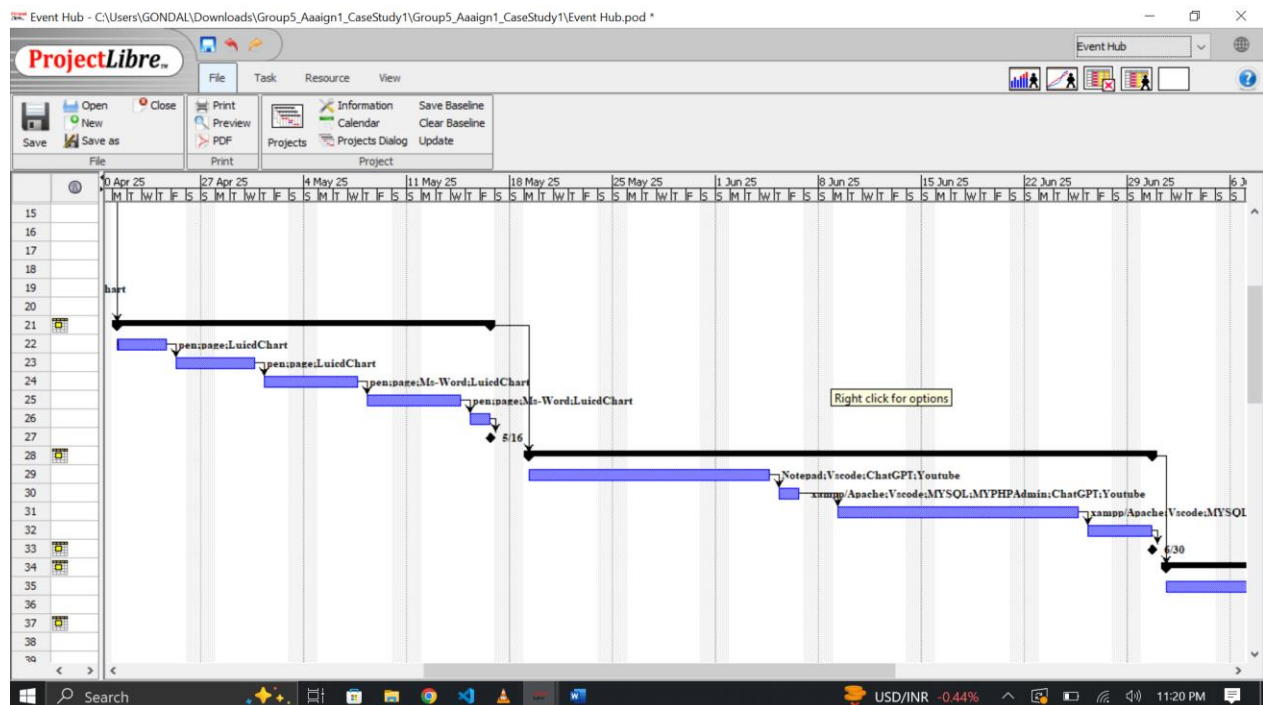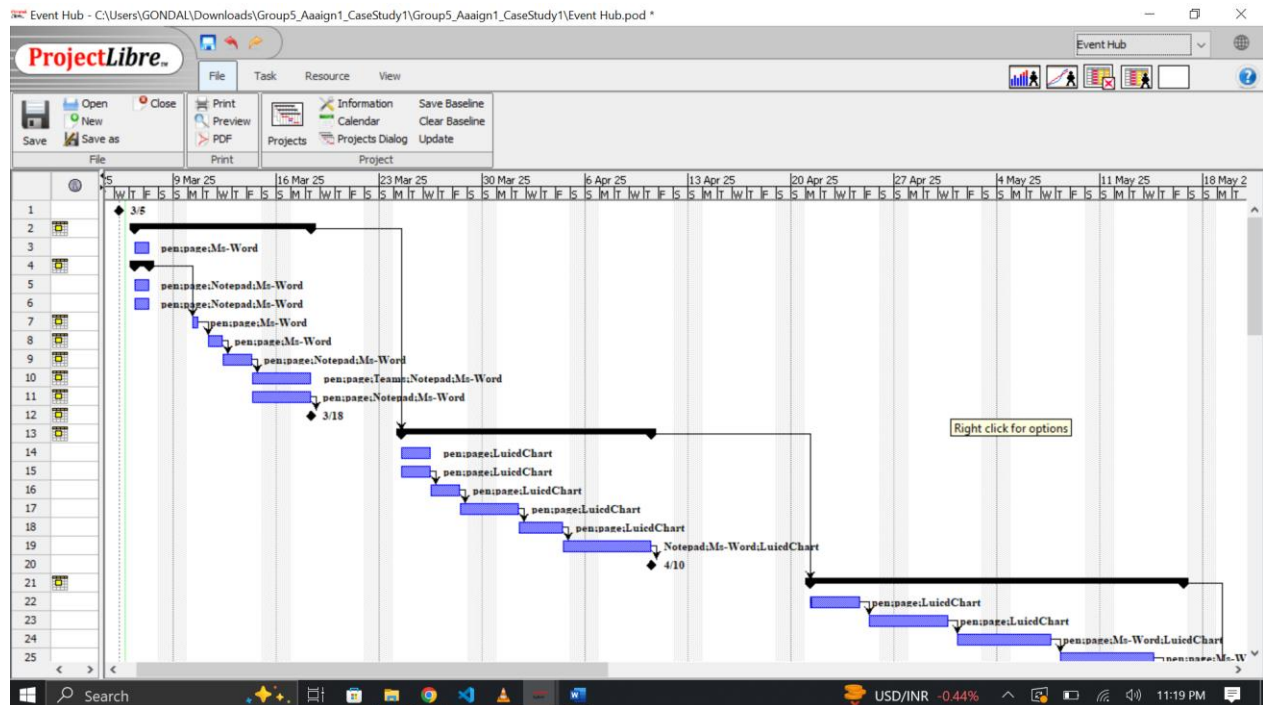
Project Schedule:

> Gantt Chart:

ProjectLibre screenshot showing task list rows 19-39:

		Name	Duration	Start	Finish	Predecessors	Resource Names
19		Define SRS	4 days	4/4/25 1:00 PM	4/10/25 1:00 PM	18	Notepad;Ms-Word;LuicdChart
20		Milestone 2	0 days	4/10/25 1:00 PM	4/10/25 1:00 PM	19	
21		⊟Design	20 days	4/21/25 8:00 AM	5/16/25 5:00 PM	13	
22		Interface design	4 days	4/21/25 8:00 AM	4/24/25 5:00 PM		pen;page;LuicdChart
23		Architectural design	4 days	4/25/25 8:00 AM	4/30/25 5:00 PM	22	pen;page;LuicdChart
24		Class diagram	5 days	5/1/25 8:00 AM	5/7/25 5:00 PM	23	pen;page;Ms-Word;LuicdCh...
25		Sequence diagram	5 days	5/8/25 8:00 AM	5/14/25 5:00 PM	24	pen;page;Ms-Word;LuicdCh...
26		Review design	2 days	5/15/25 8:00 AM	5/16/25 5:00 PM	25	
27		Milestone 3	0 days	5/16/25 5:00 PM	5/16/25 5:00 PM	26	
28		⊟Coding	31 days	5/19/25 8:00 AM	6/30/25 5:00 PM	21	
29		Frontend	13 days	5/19/25 8:00 AM	6/4/25 5:00 PM		Notepad;Vscode;ChatGPT;Y...
30		Database Connectivity	2 days	6/5/25 8:00 AM	6/6/25 5:00 PM	29	xampp/Apache;Vscode;MYS...
31		Backend	13 days	6/9/25 8:00 AM	6/25/25 5:00 PM	30	xampp/Apache;Vscode;MYS...
32		Review	3 days	6/26/25 8:00 AM	6/30/25 5:00 PM	31	
33		Milestone 4	0 days	6/30/25 5:00 PM	6/30/25 5:00 PM	32	
34		⊟Testing	14 days	7/1/25 4:00 PM	7/21/25 4:00 PM	28	
35		Software Testing	14 days	7/1/25 4:00 PM	7/21/25 4:00 PM		Ms-Word;xampp/Apache;Vs...
36		Milestone 5	0 days	7/21/25 4:00 PM	7/21/25 4:00 PM	35	
37		⊟Deployment	7 days	7/22/25 4:00 PM	7/31/25 4:00 PM	34	
38		Deploy project	7 days	7/22/25 4:00 PM	7/31/25 4:00 PM		
39		end	0 days	7/31/25 4:00 PM	7/31/25 4:00 PM	38	

ProjectLibre screenshot showing task list rows 1-25:

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		start	0 days	3/5/25 8:00 AM	3/5/25 8:00 AM		
2		⊟Planning	8.25 days?	3/6/25 8:00 AM	3/18/25 10:00 AM		
3		Purpose	1 day?	3/6/25 9:00 AM	3/7/25 9:00 AM		pen;page;Ms-Word
4		⊟Scope	2 days	3/6/25 8:00 AM	3/7/25 5:00 PM		
5		context	1 day	3/6/25 8:00 AM	3/6/25 5:00 PM		pen;page;Notepad;Ms-Word
6		functional requirement	1 day	3/7/25 8:00 AM	3/7/25 5:00 PM	5	pen;page;Notepad;Ms-Word
7		Objective	1 day?	3/10/25 8:00 AM	3/10/25 5:00 PM	4	pen;page;Ms-Word
8		StakeHolder	1 day?	3/11/25 9:00 AM	3/12/25 9:00 AM	7	pen;page;Ms-Word
9		Major Function	2 days	3/12/25 9:00 AM	3/14/25 9:00 AM	8	pen;page;Notepad;Ms-Word
10		Resources	2 days	3/14/25 10:00 AM	3/18/25 10:00 AM	9	pen;page;Teams;Notepad;...
11		Risk management	2 days	3/14/25 9:00 AM	3/18/25 9:00 AM		pen;page;Notepad;Ms-Word
12		Milestone 1	0 days	3/18/25 9:00 AM	3/18/25 9:00 AM	11	
13		⊟Analysis	13 days	3/24/25 1:00 PM	4/10/25 1:00 PM	2	
14		Use case description	2 days	3/24/25 1:00 PM	3/26/25 1:00 PM		pen;page;LuicdChart
15		Use case diagram	2 days	3/24/25 1:00 PM	3/26/25 1:00 PM		pen;page;LuicdChart
16		Domain model	2 days	3/26/25 1:00 PM	3/28/25 1:00 PM	15	pen;page;LuicdChart
17		Activity diaram	2 days	3/28/25 1:00 PM	4/1/25 1:00 PM	16	pen;page;LuicdChart
18		State chart diagram	3 days	4/1/25 1:00 PM	4/4/25 1:00 PM	17	pen;page;LuicdChart
19		Define SRS	4 days	4/4/25 1:00 PM	4/10/25 1:00 PM	18	Notepad;Ms-Word;LuicdChart
20		Milestone 2	0 days	4/10/25 1:00 PM	4/10/25 1:00 PM	19	
21		⊟Design	20 days	4/21/25 8:00 AM	5/16/25 5:00 PM	13	
22		Interface design	4 days	4/21/25 8:00 AM	4/24/25 5:00 PM		pen;page;LuicdChart
23		Architectural design	4 days	4/25/25 8:00 AM	4/30/25 5:00 PM	22	pen;page;LuicdChart
24		Class diagram	5 days	5/1/25 8:00 AM	5/7/25 5:00 PM	23	pen;page;Ms-Word;LuicdCh...
25		Sequence diagram	5 days	5/8/25 8:00 AM	5/14/25 5:00 PM	24	pen;page;Ms-Word;LuicdCh...

>Resources Allocation:

>Task Network: