جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية علوم الحاسب وتقنية المعلومات
College of Computer Science and Information Technology

CSC 305: Software Engineering

Term: 1

Academic Year: ٢٠٢٥/٢٠٢٦

Course Coordinator: Dr. Rahma Ahmed

Software Project Management Plan (SPMP)

Based on IEEE 1058 - Adapted for Student Projects

Project Title: Virtual Study Group Platform

Team Name & Members:

Name	Student ID	Role in Project
Masaheer Alshamrani	2240002456	Team Leader
Noon Abdelraziq Abdelrhman	2240009097	Backend Developer
Ranya Ali Alshamrany	2240003330	Frontend Developer
Shahad Nasser Alqahtani	2240001323	QA Tester
Rawan Alhanfoush	2240003762	UX/UI Designer

Advisor: Dr. Maha Asiri

Version: 1.0

Date: 11/10/2025

Table of Contents

1. Project Overview

1.1 Purpose, Scope, and Objectives

Purpose:

This project aims to develop an interactive online platform that enables students to form study groups, collaborate on assignments, share educational resources, and communicate effectively. By integrating chat, scheduling, and resource-sharing features into a single platform, the system seeks to improve collaboration, increase productivity, and enhance the overall study experience.

Scope of the project:

The system will include user registration, group creation, session scheduling, file/resource sharing, and real-time communication features (chat and video). Also, it covers integration with external services. However, advanced data analytics and integration with external e-learning platforms are not included at this stage.

Objectives:

- Facilitate effective peer-to-peer learning through virtual study groups.
- Provide tools for organizing and managing study sessions.
- Enable seamless communication via chat and optional video integration.
- Support easy sharing of documents, links, and educational resources.

1.2 Assumptions, Constraints, and Risks

Assumptions:

- All team members have access to laptops, stable internet connections, and GitHub for collaboration.
- The development environment will support Python, MySQL, HTML, and JavaScript.

Constraints:

- The project must be completed within the academic semester schedule (around 15 weeks).

- All features must be implemented and tested before the final presentation.
- Team members must coordinate remotely, which may affect communication and task management.
- The platform must maintain good performance while handling multiple users simultaneously.

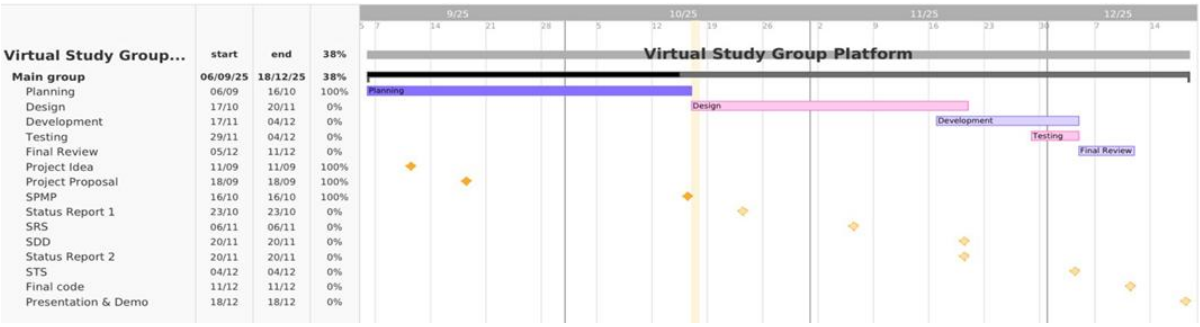
Risks:

- Security and privacy challenges when handling user data.
- API integration or third-party service issues.
- Synchronization problems in real-time sessions.
- Limited time for testing and debugging.
- Risk of missing deadlines due to workload or unexpected technical issues

1.3 Project Deliverables

Deliverable	Format	Deadline
Project Idea	Word	11 September
Project Proposal	Word	18 September
Software Project Management Plan (SPMP)	Word	16 October
Status Report 1	Word	23 October
Software Requirements Specification (SRS)	Word	6 November
Software Design Description (SDD)	Word	20 November
Status Report 2	Word	20 November
Software Test Specification (STS)	Word	4 December
Final Project Code	Code Files	11 December
Project Presentation & Demo	PowerPoint, Pdf	18 December

1.4 Schedule Summary



The Gantt chart shows our project timeline, outlining all major tasks and milestones with their durations. It indicates that the project is 38% complete, with the early planning phases already finished, and the project set to conclude on December 18, 2025.

1.5 References

- Software Engineering Course
- https://wildart.github.io/MISG5020/standards/IEEE-1058-1998.pdf
- https://app.teamgantt.com/projects/gantt?ids=4392388

1.6 Definitions and Acronyms

1. SPMP: Software Project Management Plan
2. SRS: Software Requirements Specification
3. SDS: Software Design Specification
4. STS: Software Test Specification
5. UI/UX: User Interface / User Experience
6. QA: Quality Assurance
7. API: Application Programming Interface
8. IDE: Integrated Development Environment
9. WBS: A hierarchical decomposition of the total scope of work to be carried out by the project team.
10. Agile: A flexible development approach that supports teamwork, frequent communication, and building and testing the system step-by-step
11. Unit Testing: Testing small, isolated parts of the code

2. Project Organization

2.1 External Interfaces

The Virtual Study Group Platform project interacts with multiple external stakeholders who provide oversight, feedback, and technical support. Their collaboration ensures alignment with the academic requirements of the College of Computer Science and Information Technology (CCSIT) and the project's technical feasibility.

External Stakeholders and Roles

Stakeholder	Role in Project	Type of Interaction
Course Coordinator (Dr.Maha Asiri)	Acts as the client; reviews project proposal and SPMP, provides feedback, approves milestones, and evaluates final deliverables.	Formal reviews, grading, progress meetings.
College of Computer Science & Information Technology (CCSIT)	Represents institutional needs and ensures the project aligns with CCSIT academic and administrative objectives.	Provides academic guidance and ensures compliance with college policies.
Students (EndUsers)	Primary beneficiaries of the platform; provide input on usability, communication tools, and study group functionality.	Surveys, user testing, feedback forms.
University IT Services / LMS Team	Provides technical feedback on system compatibility, authentication integration, and security standards.	Consultations, technical validation, access permissions.

Third-Party Services (GitHub, Google APIs, MySQL, Flask, Figma)	Provide technical infrastructure for hosting, database management, real-time communication, and UI/UX design.	API integration, repository management, version control.

2.2 Internal Structure

The Virtual Study Group Platform team consists of five students from the College of Computer Science and Information Technology. The team is organized to ensure clear communication, equal task distribution, and effective progress tracking using Agile (Scrum) methodology.

a) Team Leader – Masaheer Alshamrani: Responsible for overall coordination, sprint planning, scheduling weekly meetings on Microsoft Teams, and managing GitHub commits and project documentation.

b) Backend Developer – Noon Abdelrazig Abdelrhman: Implements server-side functionality using Python (Flask) and manages the MySQL database integration for user registration, chat, and scheduling modules.

c) Frontend Developer – Ranya Ali Alshamrany: Designs and develops the user interface using HTML, CSS, and JavaScript, ensuring responsiveness and accessibility across browsers.

d) Quality Assurance (QA) Tester – Shahad Nasser Alqahtani: Develops test plans, executes functional and non-functional testing, documents bugs, and ensures all project features meet acceptance criteria.

e) UI/UX Designer – Rawan Alhanfoush: Creates UI layouts and prototypes using Figma, ensuring the platform is visually appealing, intuitive, and consistent across all devices.

The team communicates daily through WhatsApp for task updates and holds weekly Scrum meetings on Microsoft Teams to review sprint progress, assign new tasks, and address any issues. All project documentation and code are shared via GitHub and Google Drive for transparency and collaboration.

2.3 Roles and Responsibilities

Team Member	Role	Responsibilities
Masaheer Alshamrani	Team Leader	Oversees project progress, manages deadlines, facilitates communication, maintains GitHub repository, and reviews deliverables.
Noon Abdelrazig Abdelrhman	Backend Developer	Implements server-side logic, manages MySQL database, and ensures API functionality for chat and scheduling features.
Ranya Ali Alshamrany	Frontend Developer	Designs and codes the platform interface, integrates frontend with backend APIs, and ensures cross-browser compatibility.
Shahad Nasser Alqahtani	QA Tester	Conducts software testing, logs bugs, verifies fixes, and ensures system reliability before submission.
Rawan Alhanfoush	UI/UX Designer	Creates wireframes and prototypes, designs visual elements, and ensures the user experience is consistent and engaging.

3. Managerial Process Plans

3.1 Estimates & Staffing

Phase	Duration	Team Members
Planning & Requirements	2 weeks	All members
UI/UX Design	2 weeks	1-2 members
Backend Development	3 weeks	Noon (Backend), Team Leader
Frontend Development	3 weeks	Ranya (Frontend), Rawan (UI/UX)
Testing & Integration	2 weeks	Shahad (QA), All members
Documentation & Final Demo	Throughout + 1 week final	Team Leader, All

3.2 Work Plan

Work Breakdown Structure (WBS): The project is divided into major phases. Each phase includes resources, deliverables, acceptance criteria, responsible members, and duration. Risks are identified with mitigation strategies.

Phase	Description	Predecessor/Successor	Responsible Members	Resources	Deliverables	Duration
System Design & Requirements	**Define system requirements, architecture, and UI/UX. Risks: incomplete requirements. Mitigation: reviews with stakeholders.**	**None → Backend Tasks**	**Team Leader, UI/UX Designer, Backend Developer**	**Figma, GitHub, Google Drive**	**Finalized requirements document, wireframes, system architecture diagrams**	**2 weeks**
Backend Development	**Implement Flask backend, MySQL database, and chat/video APIs. Risks: integration errors. Mitigation: unit testing,**	**Design → Frontend**	**Noon (Backend), Team Leader**	**VS Code, Flask, MySQL, API keys**	**Working backend with database and API integration**	**3 weeks**

	early API tests.					
Frontend Development	**Develop UI with HTML, CSS, JS, integrate with backend. Risks: mismatched UI. Mitigation: peer reviews, early testing.**	**Backend → Integration**	**Ranya (Frontend), Rawan (UI/UX)**	**VS Code, Browsers, GitHub**	**Responsive frontend connected with backend services**	**3 weeks**
Integration & Testing	**Combine frontend and backend, perform QA, fix bugs. Risks: integration conflicts. Mitigation: version control, bug tracking.**	**Frontend/Backend → Final Demo**	**Shahad (QA), All Members**	**GitHub, Trello, PyTest, Browser Dev Tools**	**Integrated system with test reports**	**2 weeks**

3.3 Project Tracking Plan

A. Monitoring Progress

- Weekly sprint review meetings to assess completed tasks and blockers.
- GitHub commits and pull requests used to track development.
- Trello board for task management.

B. Change Control Process

- Any proposed changes must be documented and reviewed by the Team Leader.
- Approved changes are updated in Trello and GitHub milestones.

C. Metrics for Tracking

- % of tasks completed (weekly).
- Number of issues resolved.
- Requirement coverage vs. implemented features.
- Test coverage during QA phase.

D. Risk Monitoring

- Risks reviewed weekly and mitigation plans updated.
- New risks documented and assessed during sprint reviews.

3.4 Risk Management Plan

Risk	Likelihood	Impact	Mitigation
Missed deadlines	Medium	High	Implement strict sprint deadlines and monitor weekly.
Integration failure	Low	High	Test API components early using mock data.
Team member unavailability	Medium	Medium	Redistribute tasks promptly among members.
Data loss	Low	High	Use GitHub and daily cloud backups.
Technical challenges in Chat/Video APIs	Medium	High	Provide early research, tutorials, and fallback options.

3.5 Closeout Plan

The project will formally conclude after verifying all deliverables meet acceptance criteria. Upon completion of testing and integration, the team will submit all documentation (SRS, SDD, STS), source code, and final presentation slides. The system will be archived on GitHub and Google Drive. A final sprint review meeting will be conducted to confirm project completion. Team members will prepare a reflection report summarizing lessons learned, challenges, and recommendations.

4. Technical Process Plans

4.1 Process Model

We chose Agile because it is flexible and fits our project needs. The Virtual Study Group Platform has many features that may change or improve as we work, so Agile helps us adapt easily.

1. Agile supports teamwork and communication, which is important for our group project.
2. It allows us to build and test parts of the system step by step, so we can see results early and make changes if needed.
3. Since our project includes different parts like chat, scheduling, and design, Agile helps us work on them at the same time and combine everything smoothly.

4.2 Methods, Tools, and Techniques

Frontend: HTML, JavaScript.

Backend: Python.

Database: MySql.

API Integration: connect with external services such as chat, and calendar scheduling.

Version Control: GitHub for code management and team collaboration.

Testing Tools: Manual testing and debugging tools in the IDE.

4.3 Infrastructure

The Virtual Study Group Platform will be developed using laptops or computers running on Windows, macOS, or Linux with at least 8GB of RAM and a stable internet connection. The programming languages used are Python, HTML, and JavaScript. MySQL is used for the database, while GitHub and Google services are used for code management, and online testing.

4.4 Product Acceptance

The final Virtual Study Group Platform will be accepted when all main features are completed and tested successfully. Each part of the system should work properly, have no major errors, meet the project goals, and be easy for users to use. The system should also be reliable and responsive.

All the following features must work as intended before final approval:

- User registration and profile management
- Study group creation and management
- Resource sharing (documents, links, and files)
- Real-time chat
- Session scheduling with reminders
- Dashboard and activity insights
- To-do list and timer features
- Notification system and integrated calendar

Final approval of the product will be granted by Dr. Maha Asiri after verifying that all system features function correctly and meet the project requirements.

5. Supporting Process Plans

5.1 Documentation Plan

Storage Location	Reviewed by	Prepared by	Document
Google Drive	Instructor	Team Leader(Masaheer Alshamrani)	Project Proposal
Google Drive	Team Leader	Ranya Alshamrany & Noon Abdelrazig	SPMP
Google Drive / GitHub	QA Tester (Shahad Alqahtani)	Noon Abdelrazig	SRS (Software Requirements Specification)
Google Drive / GitHub	Team Leader	Ranya Alshamrany	SDS (Software Design Specification)
Google Drive / GitHub	Instructor	Shahad Alqahtani	STS (Software Test Specification)
Figma + PDF	Team Leader	Rawan Alhanfoush	UI/UX Design Report
Google Drive	Instructor	All Members	Final Report & Presentation

5.2 Quality Assurance

Coding Standards:

- Backend (Python / Flask): clear names, proper indentation, and readable code.
- Frontend (React): same structure and design style for all pages.
- Database schemas must use clear naming conventions (lowercase with underscores).

Peer Review Policy:

- All code must be reviewed via **GitHub Pull Requests** before merging.
- At least **two approvals** are required per merge.
- The QA Tester verifies key functionality after each merge to the dev branch.

Testing Coverage:

- **Unit Testing:** testing small parts of the code (functions, APIs).
- **Integration Testing:** verifying that all system parts (frontend + backend) work together correctly.
- **UI Testing:** checking design behavior and user interactions (buttons, forms, links).
- **Acceptance Testing:** final testing round before submitting the completed system.

Quality Metrics:

- No major bugs in the final version.
- At least 80% of the project functions tested successfully.
- All team members must review their code before submission.

5.3 Configuration Management

Version Control Tool:

- The project uses **GitHub** to manage and track all source code, documents, and design files.

Branching Policy:

- main: stable and approved version for submission.
- dev: active development branch for testing new features.
- feature/*: individual branches for each new task or feature.

Commit Rules:

- Each commit must have a clear and descriptive message (e.g., feat(login): add login authentication).
- Only the Team Leader merges changes into the main branch after full review and testing.
- **Release Plan:**

Target Date	Description	Version
Week 4	Proposal and SPMP Submission	v1.0
Week 9	Prototype Release	v2.0
Week 15	Final Project Delivery	v3.0

5.4 Problem Resolution

Reporting Issues:

- All bugs or problems will be reported as **GitHub Issues** with the following details:
    - Issue title and description
    - Steps to reproduce
    - Priority level (Critical / High / Medium / Low)
    - Assigned developer

Resolution Process:

1. The issue is created by the member who found the problem.
2. The Team Leader assigns the issue to a responsible developer.
3. The developer fixes the issue and submits it for testing.
4. The QA Tester confirms that the issue is resolved before closing it.

Communication Channels:

- **Discord / Microsoft Teams:** used for quick team communication.
- **GitHub Projects:** used for tracking progress and managing issues.

Resolution Timeline:

- Critical issues: within 24 hours
- High: within 48 hours
- Medium: within 3 days
- Low: before the next milestone