Software Project Management Plan (SPMP)

Course: CSC 305 – Software Engineering  
Term 1, 2025/2026  
Instructor: Dr. Maha Asiri  
  
Project Title: Virtual Study Group Platform  
Process Model: Agile (Scrum)  
Team Leader: Masaheer Alshamrani  
Team Members: Noon Abdelrazig Abdelrhman, Ranya Ali Alshamrany, Shahad Nasser Alqahtani, Rawan Alhanfoush  
  
Version: 1.0  
Date: October 27, 2025

# 1. Project Overview

## 1.1 Purpose, Scope, and Objectives

The purpose of the Virtual Study Group Platform project is to provide university students with a collaborative environment where they can connect, form study groups, share learning resources, and communicate efficiently through chat and video. The scope includes developing core features such as user registration, group creation, resource sharing, real-time chat, and video conferencing. Objectives are to enhance peer learning, simplify study coordination, and improve productivity in a virtual educational context.

## 1.2 Assumptions, Constraints, and Risks

Assumptions:  
• Each team member has reliable internet access and a functional computer.  
• The team collaborates using GitHub and online communication tools.  
  
Constraints:  
• Project duration is limited to 15 weeks.  
• Only open-source frameworks are allowed.  
  
Top Risks:  
• Missed sprint deadlines due to academic workload.  
• Technical challenges integrating chat/video APIs.  
• Inconsistent participation or communication delays.

## 1.3 Project Deliverables

• Project Proposal – Week 4  
• SPMP – Week 8  
• SRS – Week 11  
• SDD – Week 13  
• STS – Week 15  
• Final Code and Presentation – Week 16

## 1.4 Schedule Summary

The project follows a 15-week Agile schedule with bi-weekly sprints. Key milestones:  
• Week 3: Project Definition  
• Week 4: Proposal Submission  
• Week 8: SPMP Delivery  
• Week 11: SRS  
• Week 13: SDD  
• Week 15: STS  
• Week 16: Demo & Final Submission

## 1.5 References

IEEE Std 1058-1998, Software Project Management Plan. Sommerville, I. (2016). Software Engineering, 10th Edition.

# 2. Project Organization

## 2.1 External Interfaces

The project team reports to the course instructor, Dr. Maha Asiri, who provides feedback and evaluates deliverables. The College’s IT lab and online learning system act as supporting environments.

## 2.2 Internal Structure

The project follows an Agile team structure consisting of a Team Leader, Developers, a QA Tester, and a UI/UX Designer. The team meets weekly for sprint planning and reviews through virtual meetings.

## 2.3 Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Team Member | Role | Responsibilities |
| Masaheer Alshamrani | Team Leader | Oversees project progress, assigns tasks, leads meetings. |
| Noon Abdelrazig Abdelrhman | Backend Developer | Implements server-side logic and database integration. |
| Ranya Ali Alshamrany | Frontend Developer | Develops user interface using HTML, CSS, and JavaScript. |
| Shahad Nasser Alqahtani | QA Tester | Designs and executes test cases, ensures quality assurance. |
| Rawan Alhanfoush | UX/UI Designer | Designs user experience flow, creates mockups and wireframes. |

# 3. Managerial Process Plans

## 3.1 Estimates & Staffing

The project workload is distributed across 15 weeks with 6 sprints. Estimated total effort: 200 hours. Team size: 5 members. Roles are divided equally among development, design, and testing.

## 3.2 Work Plan

Main phases:  
1. Planning and Requirements Gathering  
2. UI/UX Design  
3. Backend and Frontend Development  
4. Testing  
5. Integration  
6. Final Demo  
Each sprint includes development, review, and testing activities following Agile principles.

## 3.3 Project Tracking Plan

Progress is tracked through GitHub commits, Trello boards, and weekly sprint review meetings. Metrics include task completion percentage and issue resolution rate.

## 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. |
| Member unavailability | Medium | Medium | Redistribute tasks promptly among members. |
| Data loss | Low | High | Use GitHub and cloud backups daily. |

## 3.5 Closeout Plan

After final presentation, team members will submit documentation, archive code on GitHub, and write a final reflection report summarizing lessons learned.

# 4. Technical Process Plans

## 4.1 Process Model

The Agile (Scrum) model is adopted to allow iterative development and continuous feedback. Each sprint delivers incremental features for testing and evaluation.

## 4.2 Methods, Tools, and Techniques

Backend: Python (Flask)  
Frontend: HTML, CSS, JavaScript  
Database: MySQL  
Version Control: GitHub  
Project Management: Trello  
Testing: PyTest  
Communication: Microsoft Teams

## 4.3 Infrastructure

The project uses laptops with Windows or macOS, local MySQL server, and GitHub for cloud-based code management.

## 4.4 Product Acceptance

The product will be accepted if all core functionalities—user login, group creation, resource sharing, chat, and video—are functional and tested. Approval by Dr. Maha Asiri confirms completion.

# 5. Supporting Process Plans

## 5.1 Documentation Plan

All documents (Proposal, SPMP, SRS, SDD, STS) are maintained in a shared Google Drive folder. Each deliverable will be reviewed by the team leader before submission.

## 5.2 Quality Assurance

Code reviews are conducted after each sprint. The QA tester verifies all features using functional and non-functional tests. Peer reviews ensure high code quality.

## 5.3 Configuration Management

GitHub will manage version control. The team follows a branching strategy: main branch for stable code, feature branches for development, and pull requests for integration.

## 5.4 Problem Resolution

Issues are reported on Trello and assigned to responsible team members. Weekly meetings address unresolved bugs or blockers.

# 6. Additional Plans

Training Plan: Team members will watch online tutorials for Python Flask and WebRTC integration.  
Security Plan: Passwords and user data are encrypted using hashing algorithms.  
Maintenance Plan: Future versions may include mobile compatibility and integration with university systems.