

# Software Engineering Project Plan

StudyShelf: Classroom Resource Sharing Platform by Group 6

## 1. Introduction

### 1.1 Project Description

This project's title is StudyShelf and it is a Java-based classroom resource sharing platform. The aim of the project is for StudyShelf to become a platform that teachers and students can use to easily share useful educational resources for their studies. The platform's goal is to increase learning efficiency and resource gathering in educational environments.

### 1.2 Objectives

#### 1.2.1 Goals

- Users can upload, access and download resources
- The resources have ratings and reviews to show the quality of resources
- The platform has categories and tags for the resources to find related resources more easily
- The platform has search capabilities to find specific resources
- Users can register and log in with role-based authentication, where teachers and students have different permissions for managing resources and categories

#### 1.2.2 Intended Outcomes

- Users can efficiently share high-quality educational resources
- The rating and review system helps users identify valuable resources
- Categorization and tagging enhance the discoverability of related resources
- Search functionality improves accessibility to specific content
- Teachers and students have different permissions based on their roles

### 1.3 Scope

#### 1.3.1 Included features

- Uploading, downloading and managing of resources.
- Resource categorization and tagging.
- Rating and reviewing system for resource quality.
- Search capabilities for resources.

#### 1.3.2 Excluded features

- Live chat functionality
- Gamification

## 2. Project Organization

### 2.1 Team Structure

The team consists of three software developers, Armas Nevolainen, Jiayue Zheng and Santtu Saaranen. Armas Nevolainen is acting as a database developer, with a focus on user authentication. Jiayue Zheng is acting as a database designer and developer. Santtu Saaranen is acting as a UI/UX designer and is responsible for its implementation.

Amir Dirin will be involved with discussing the progress at the end of each sprint and acting as the product owner.

The role of the team's Scrum Master is rotating, with a new Scrum Master being assigned at the start of each sprint. The first sprint's Scrum Master is Santtu Saaranen.

### 2.2 Communication Plan

The developer team's main communication tool for quick questions and solutions is the shared Discord server. This server is used daily. Additionally, the developers meet twice a week to discuss the project in person.

The development time is split into a total of four sprints, each lasting two weeks. At the end of each sprint, there's a dedicated Sprint Review where the project is discussed with all of the team members, including the project owner.

### 2.3 Stakeholder Involvement

- Teachers and students: provide requirements, use the platform, give feedback on improvements
- Development team: handle development, conduct testing and perform maintenance
- Educational institutions: support platform promotion and provide infrastructure

### 3. Risk Analysis

Risk Description	Likelihood	Impact	Mitigation Strategies
Possible Delay in Development: Unforeseen technical issues.	High	High	Use Trello for progress tracking. Plan realistic Sprint backlogs with buffer time for possible delays. Conduct daily Scrum meetings to discuss and address technical issues early.
Data Security: Unauthorized users may delete or modify resources.	High	High	Implement role-based authentication: Only teachers can manage all resources in their course; users can only manage their own uploads.
Software Bugs: Errors may impact platform stability and functionality.	High	High	Conduct Unit Tests before deployment to identify and fix bugs.
Copyright Infringement: Users may upload copyrighted resources without permission	Medium	High	Implement a mandatory agreement checkbox before uploads, confirming that resources are original or legally authorized. Allow teachers to delete infringing resources.
User Adoption Issues: Users may not actively use the platform.	Medium	High	Implement a user-friendly UI/UX design to improve user engagement.
Poor Quality of Uploaded Resources: Users may upload irrelevant or low-quality materials.	Medium	Medium	Implement a rating and review system for evaluation. Allow teachers to manage uploaded resources.

## 4. Hardware and Software Resource Requirements

These requirements are aimed at the software developers of the project.

### 4.1 Hardware

RAM	4 GB
Free Disk Space	20 GB
OS	Windows 10/11

### 4.2 Software

Java IDE	IntelliJ IDEA
Build Automation	Apache Maven
Version Control	Git
Testing	JUnit
DB Management System	MariaDB
Database Interactions	JDBC
Storing Files	Google Drive
UI Layout Tool	SceneBuilder
UI Build Library	JavaFX

## 5. Work Breakdown

As according to the team structure, each team member has main responsibilities. Santtu Saaranen is primarily responsible for designing and implementing the UI/UX elements, Armas Nevolainen is responsible for user authentication and Jiayue Zheng is mainly responsible for other parts of the database.

### 5.1 Project Planning and Documentation

Since the team is small, working together on some of the tasks is entirely possible. Most of the tasks in this stage are mainly assigned to the whole team to ensure that everyone is involved in the planning stage and is on board when the programming starts.

- Trello Set Up
  - Creating boards for each sprint and assigning tasks
- Project Plan and Project Vision Documents
  - Documents for defining the idea and scope of the project

### 5.2 Database

- Design Stage
  - Clear plan of the database entities done together with the whole team
  - Plan out some possible queries that the final software will include
  - Only a plain text editor draft
  - **Assigned Team Members:** Jiayue Zheng, Armas Nevolainen
- ER-Model
  - ER-model based on the plan made using ERDPlus
  - **Assigned Team Member:** Jiayue Zheng
- Building Stage
  - Create database in MariaDB with SQL exported from ERDPlus
  - **Assigned Team Member:** Jiayue Zheng
- Database Testing
  - Make various test cases to test that the database is working as intended, using the drafted queries
  - **Assigned Team Member:** Jiayue Zheng
- Database Deployment
  - Deploy database to when the local version is tested to work well enough
  - **Assigned Team Members:** Jiayue Zheng, Armas Nevolainen

### 5.3 User Interface

- Planning Stage
  - Using the first database design and the features mentioned in the project plan, make a text draft of everything the UI has to have
  - **Assigned Team Member:** Santtu Saaranen
- Design Stage
  - Make a draft of the general interface design concept using Figma

- **Assigned Team Member:** Santtu Saaranen
- SceneBuilder
  - **Dependency:** This part is best to be left when the Figma draft and the text plan are done
  - Make a SceneBuilder document and work on it until there's an usable .fxml file
  - **Assigned Team Member:** Santtu Saaranen

## 5.4 User Authentication

- Database implementation
  - Store hashed passwords using BCrypt
  - Implement user roles table
  - **Assigned Team Member:** Armas Nevolainen
- JavaFX Components
  - Login Form
  - Password field with encryption
  - Error handling display
  - **Assigned Team Member:** Santtu Saaranen
- Security Features
  - JWT token implementation
  - Session timeout handling
  - Access based on user role
  - **Assigned Team Member:** Armas Nevolainen

## 5.5 Overall Programming

- GitHub repository
  - Make two GitHub repositories for the project, with all of the team members and the instructor as collaborators
  - **Assigned Team Member:** Santtu Saaranen
- **Dependency:** The following tasks should only be done after all parts of User Interface and most of Database is done
- Controller
  - Following the concept of the MVC-model, connect the moving parts together.
  - **Assigned Team Member(s):** The whole team will end up working on this together, with Armas Nevolainen and Jiayue Zheng holding main responsibility for the Model and Santtu Saaranen holding main responsibility for the View

## 6. Project Schedule

### 6.1 Timeline

#### 6.1.1 Sprint 1 (Week 1-2)

- Team setup
- Project Selection
- Trello setup
- GitHub setup
- Definition of product backlog
- Project UI wireframe

#### 6.1.2 Sprint 2 (Week 3-4)

- Database design and implementation
- UI design and implementation
- JUNIT Test
- User authentication
- Basic upload functionality

#### 6.1.3 Sprint 3 (Week 5-6)

- Material review and rating mechanisms
- Search functionality
- Jenkins integration for unit testing

#### 6.1.4 Sprint 4 (Week 7-8)

- UI refinement
- Testing and debugging
- Project documentation

### 6.2 Milestones

#### 6.2.1 Sprint 1

- Project foundation complete
- UI design wireframe complete
- Development environment ready
- Project Backlog ready
- GitHub repository is set up

#### 6.2.2 Sprint 2

- Database implementation complete
- Basic UI ready
- Authentication system working
- Basic upload feature working

### 6.2.3 Sprint 3

- Review system implemented
- Search function working
- Rating system working

### 6.2.4 Sprint 4

- Final UI version complete and polished
- All tests pass
- Documentation delivered

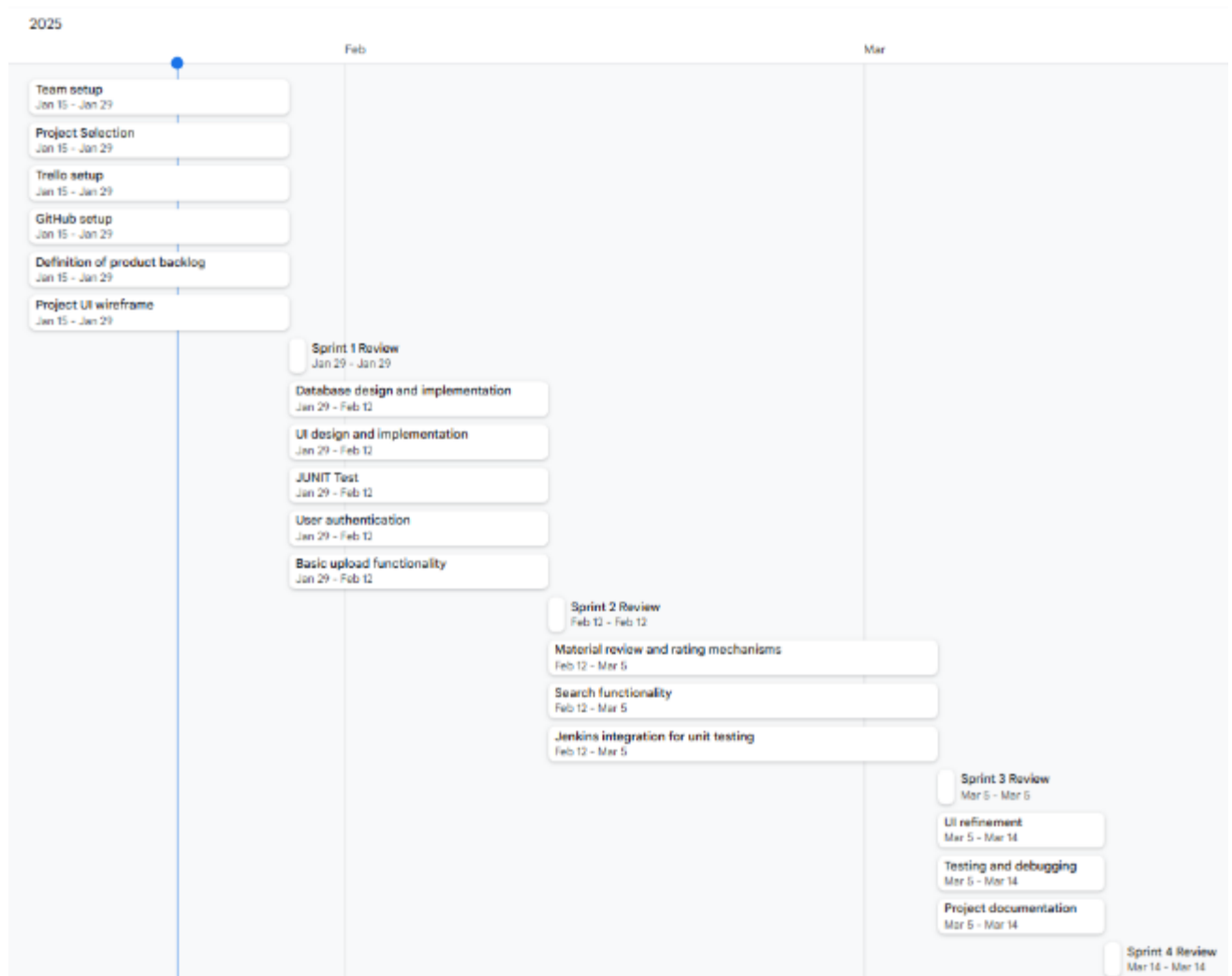


Figure 1. Project Timeline



## 7. Monitoring and Reporting Mechanisms

### 7.1 Progress tracking

#### 7.1.1 Daily progress monitoring.

- Trello board task status updates
- Daily SCRUM meetings

#### 7.1.2 Sprint progress

- Story points completion tracking
- Feature completion tracking

#### 7.1.3 Milestones tracking

- Weekly milestones status review
- Timeline adherence monitoring

### 7.2 Reporting structure

#### 7.2.1 Team communication

- Daily SCRUM
- Real-time updates via Discord
- Sprint planning every two weeks
- Sprint retrospectives

#### 7.2.2 Stakeholder updates

- Sprint review with product owner at the end of each sprint
- Presentation of the full product at the end of the fourth sprint