Project Plan: Personalized Study Planner EduSync

Group 3: Aaro Jylhämäki, Juhana Hänninen, Leevi Rinnetmäki & Nea Lukumies

31.8.2025

1. Project Overview

This project is part of the course Software Engineering Project 1 in Metropolia University of Applied Sciences. The purpose of this project is to use agile software engineering methods when developing an application for personalized study planning. The problem is that students struggle with organizing their studies as all existing solutions are divided into different applications. This makes it difficult to visualize the overall workload and deadlines. The EduSync application will combine all the necessary features, making study planning more efficient.

Our target audience consists mainly of students in higher education, but the application may also be used by younger students who need a detailed study planner. The main features of EduSync are creating and tracking study plans. In addition, students can integrate their class schedule, set assignment deadlines and reminders in the application.

2. Project Objectives

These are the objectives of this project:

- Develop the EduSync application with the planned main features described in the Project Overview.
- Design an easy-to-use user interface that provides students with a clear view of their study plans.
- Implement a reliable backend database to securely store and prove access to students' study plans.

In line with the course requirements the project should also apply agile software development methods such as sprint planning and sprint reviews.

3. Scope and Deliverables

The scope of this project includes the following features:

- Tools for creating, editing and monitoring personalized study plans.
- Deadline reminders for assignments, exams and projects.
- Intuitive and easy-to-use interface.

- Multi-language translations.

The following are excluded from the scope:

- Multi-user collaboration or real-time group planning.
- Integration with cloud services.

Planned deliverables:

- EduSync Application: A fully functioning desktop application built with JavaFX that supports study plan management, deadline tracking and reminders.
- EduSync Database: A well-structured database to manage schedules, tasks and plans, built with MariaDB.
- EduSync documentation: Installation and user guides for the EduSync application.

4. Project Timeline

This is the planned timeline for the project.

Sprint	Time	Scrum Master	Milestones and Key Objectives
Sprint 1	19.8.-1.9.25	Nea Lukumies	- Requirement gathering completed. - Initial backlog created. - Project scope and deliverables defined. - Development environment setup.
Sprint 2	2.-15.9.25	Aaro Jylhämäki	- Database created. - Basic functionality (Study plan creation, editing and monitoring) implemented. - Rudimentary interface created
Sprint 3	16.-29.9.25	Juhana Hänninen	- User interface refined. - Functionality developed further. - Unit testing and bug fixing conducted.
Sprint 4	30.9.-5.10.25	Leevi Rinnetmäki	- Final testing and bug fixing completed. - Project documentation prepared. - Final presentation prepared.

5. Resource Allocation

The team members for this project are Nea Lukumies, Aaro Jylhämäki, Juhana Hänninen and Leevi Rinnetmäki. There are currently no specialized roles for each member as those roles usually arise naturally as the project continues. The task of Scrum Master will alternate each sprint so that each team member will be Scrum Master once during Software Engineering Project 1.

The IDEs we will use will be IntelliJ and Visual Studio Code. We will program the application using Java and MySQL. We will utilize frontend, backend and databases in the project to ensure a useful application. There will be no specialized hardware used in the project, standard laptops are sufficient for development and testing.

6. Risk Management

There are many risks that could impact on the success of our project, but there are three that stand out. The first notable risk is running out of time, due to other schoolwork. The chance of this having at least a minor impact is very high, but it can have a catastrophic impact if not dealt with accordingly. To ensure that doesn’t happen, the group must maintain a clear sense of what needs to be done and when.

The second notable risk is someone getting sick. The likelihood of this happening is not that high, and the impact should be low, because sick group members can usually still work at home. It may make it more difficult to communicate and plan, because group members are temporarily unable to communicate face to face, but that can be fixed with good online communication.

The third notable risk is misunderstanding something about the project, e.g., when the deadlines are, what the goals of the project are, etc. The potential impact is quite severe, because the project might not meet its requirements or deadlines, which is why it's important to try to mitigate this as much as possible. This can be done by thoroughly looking over the material and by asking clarifying questions when appropriate.

7. Testing and Quality Assurance

Testing will be accomplished on two fronts: automated and manual. Manual testing is to be performed by our developers, emulating different sorts of users. Its purpose is to gauge the user friendliness of the app, functionality and performance. Automated testing will be mainly done to stress test our infrastructure and to ensure that general requirements are met in the app’s design.

Manual testing will require no significant tools as it will be done by hand. Ideally, we would have outside users help in later testing phases. Automated testing will use the following technologies/frameworks:

- JUnit (for unit testing)
- JavaFX Test (for UI testing)

User acceptance testing will be performed near the end of our initial development cycle, and will include rigorous testing by our developers, and the introduction of random participants to determine if our design will work in the real world. The secondary purpose of outside testing is to see how intuitive the user experience is for a first-time user. This phase of testing should be completed in time to still make significant improvements and redesigns, if serious problems are found in the current version of our app.

Quality Assurance (QA) testing will be performed continuously throughout the development process. Its main purpose is to ensure our project meets requirements and standards set by the product owner. This will consist of unit testing, as well as integration testing.

The purpose of unit testing will be to ensure that new components and other parts of the app follow the same standard and do not break any existing parts of the app. Integration testing will be done to compare different areas of the app and to see if communication between the front and backend is clear.

We will use docker containers to streamline development environments across all developers. This will decrease the number of bugs caused by developers relying on their own environment and hopefully eliminate the infamous "it works on my machine" issue.

8. Documentation and Reporting

Documentation as a process will be done throughout the project, beginning as soon as a clear outline for the app has been finished. Technical documentation will be written as soon as the first lines of code are created and will expand to include other parts of project infrastructure. User manuals will be made separately for every relevant part of the user experience and compiled together into a definitive user guide before the product is released.

Progress reports will be reported weekly by standard, during a meeting scheduled by the Scrum master. Reports can also be made during sprint reviews and individual parts can also be reported on through intermittent communications, like our developer WhatsApp group. A general idea of our progress will be gained from discussing our individual parts as a group and communicating about any impediments to the project.