Final Documentation: StudyShelf

Software Engineering Project 2025

Armas Nevolainen, Santtu Saaranen, Jiayue Zheng
Group 6

Table of Contents

Project development Process

1. Applied Methodology

In this project, we adopted Agile development using the Scrum framework. We supported our workflow and delivery process with the following technologies:

- Docker
- Jenkins
- Git
- Trello
- Hibernate
- JPA
- Java
- JavaFX
- MariaDB
- JWT
- Google Drive API
- Google Translate API

2. Planning and 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 bad materials.	Medium	Medium	Implement a rating and review system for evaluation. Allow teachers to manage uploaded resources.

Table 1. Risk analysis conducted at the start of the project

3. Agile implementation Approach

In this project, we followed Scrum, structured around iterative development cycles and continuous collaboration. Development was organized into Sprints, each with a fixed duration of two weeks.

- **Sprint Planning:** At the start of each Sprint, the tasks were organized by the requirements of the deliverables. The Scrum Master would derive tasks from the user stories and assign responsibilities for each task.
- **Daily Communication:** Project was worked on daily through the team's Discord server with fast responses to messages and dedicated channels for different areas of work.
- **Sprint Review**
    - User Stories Status: During each Sprint, user stories moved through clearly defined status stages that reflected their development progress.
    - Time Tracking: The overall conclusion among the team is that everyone spent approximately similar amounts of time and effort this sprint.

4. Scrum team roles and Responsibilities

For each Sprint, the team members acted as developers. Additionally, a Scrum Master was named at the start of each Sprint.

	Sprint1	Sprint2	Sprint3	Sprint4	Sprint5	Sprint6	Sprint7	Sprint8
Santtu Saaranen	Scrum Master	Developer	Developer	Developer	Scrum Master	Developer	Developer	Scrum Master
Jiayue Zheng	Developer	Scrum Master	Developer	Scrum Master	Developer	Developer	Scrum Master	Developer
Armas Nevolainen	Developer	Developer	Scrum Master	Developer	Developer	Scrum Master	Developer	Developer

Table 2. Each member's roles categorized by Sprint

Version control of the project was done at a dedicated GitHub repository. For Sprint planning and tracking progress, Trello was used.

Product documentation

1. Introduction

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.

2. Design

2.1. ER Diagram

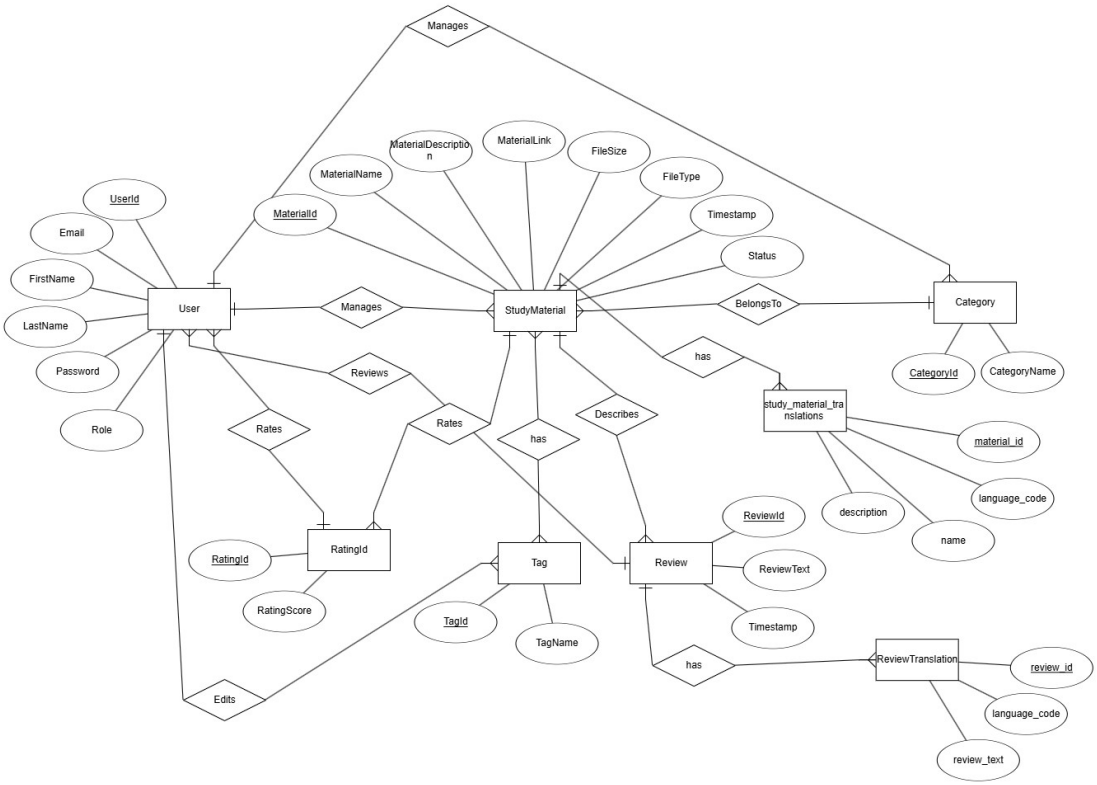

Fig 1. ER Diagram of the StudyShelf database

2.2. UML Diagrams

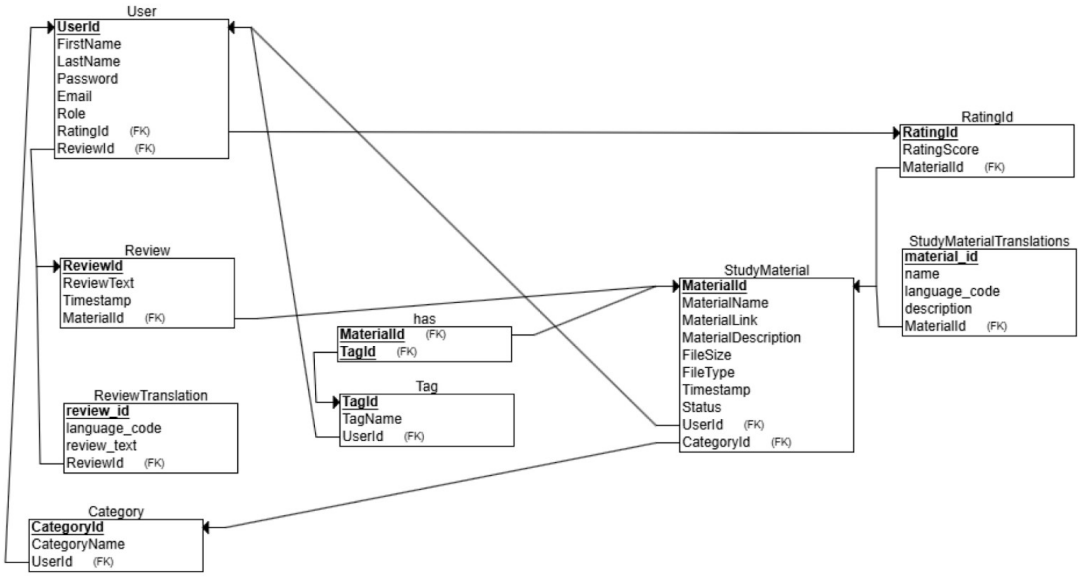

Fig 2. Relational Database Schema

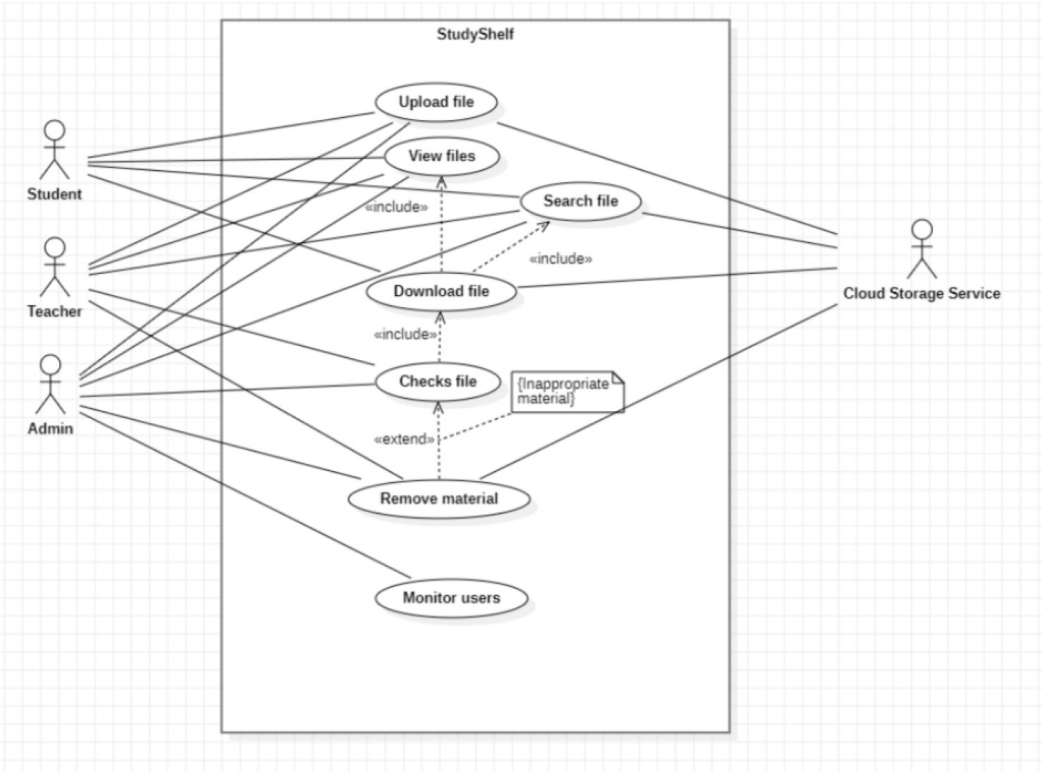

Fig 3. Use Case Diagram

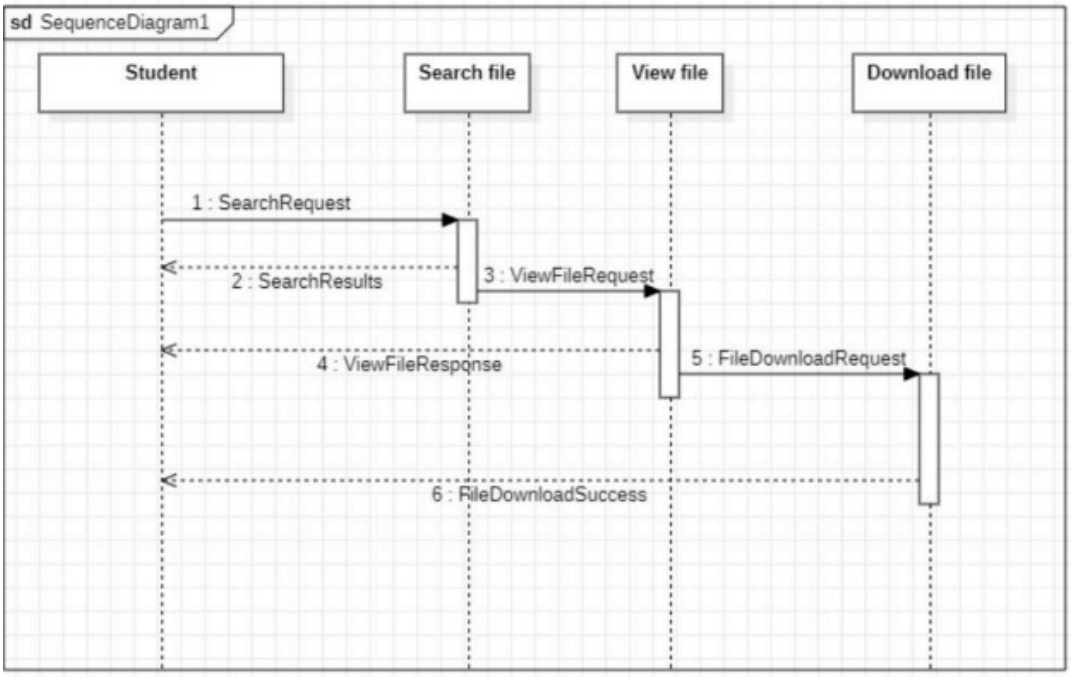

Fig 4. Sequence Diagram

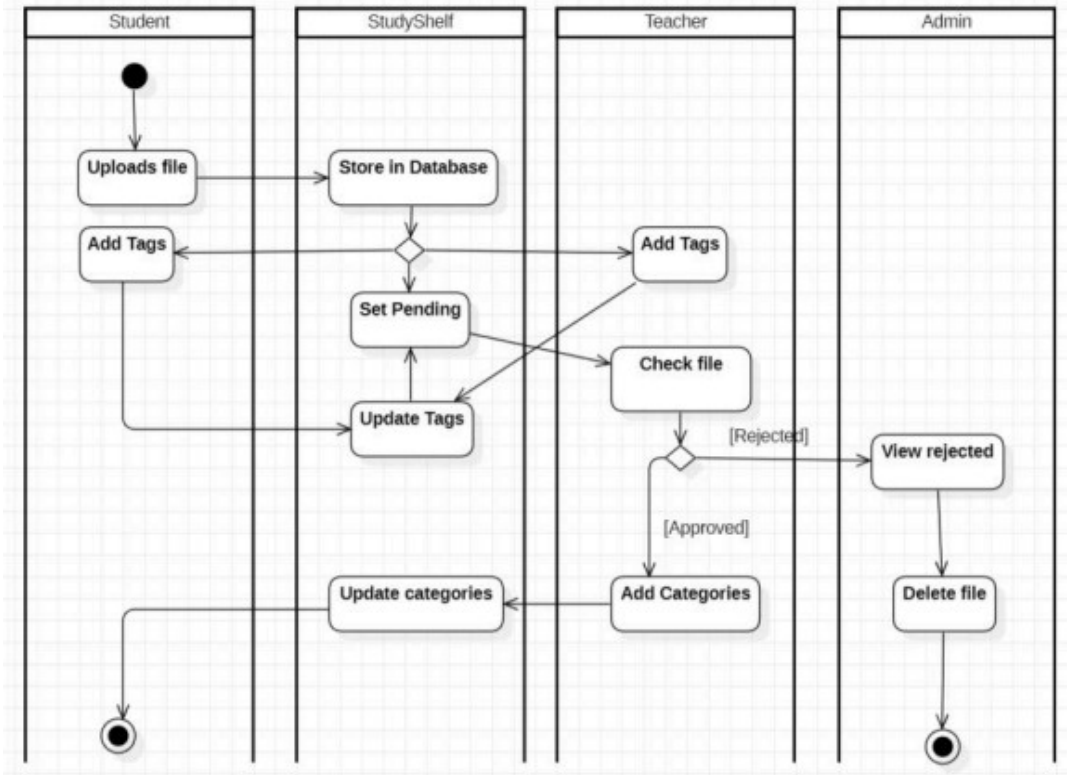

Fig 5. Activity Diagram

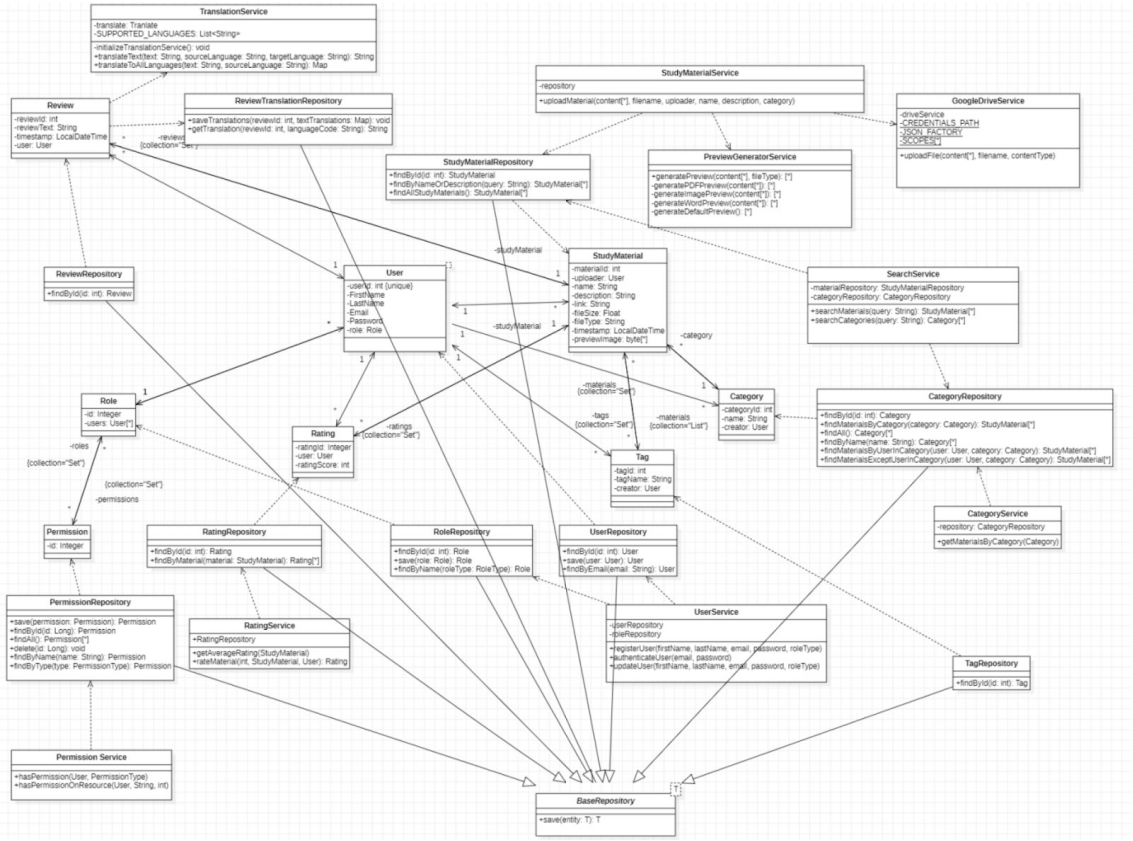

Fig 6. Class Diagram

2.3. UI Mockups

For the UI design, there was a low fidelity prototype that ultimately wasn't used, but worked as a base for the rest of the design.

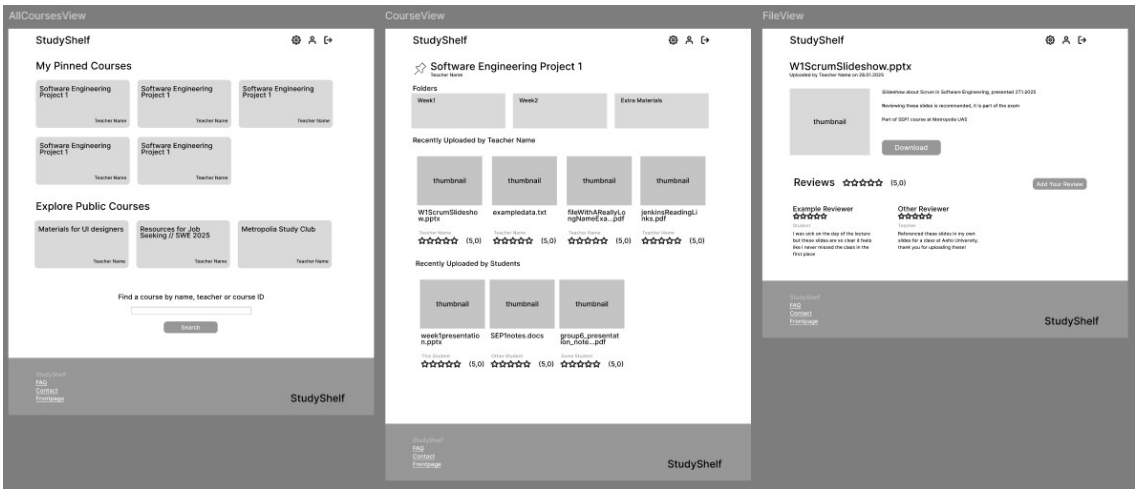

Fig 7. Early low fidelity prototype

The prototype was created before there was a common understanding of the exact features of the platform. In addition, a style guide was made with Figma.

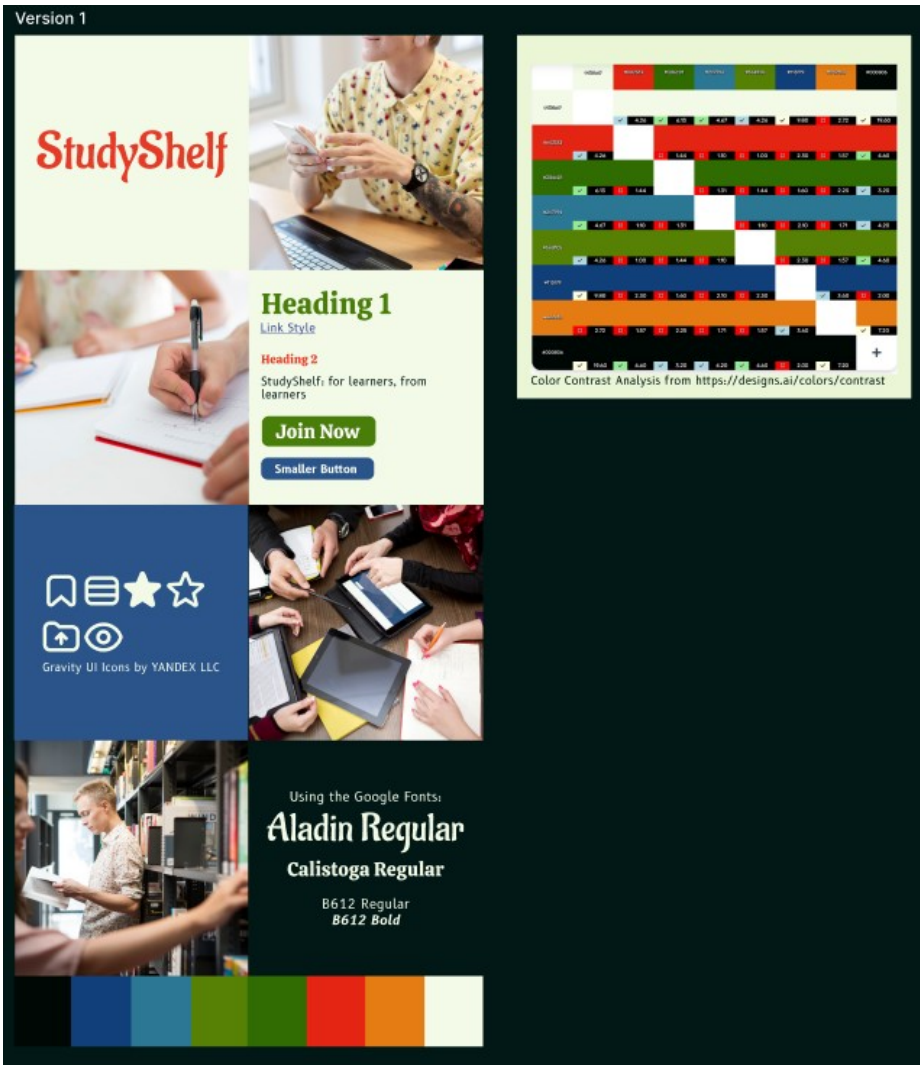

Fig 8. StudyShelf style guide

The style guide defined the fonts, colors and button style for the rest of the project. It also defined that the icons should be from the Gravity UI iconset whenever possible. Both of the documents can be viewed from this Figma file.

3. Features Overview

3.1 Account Management

All of the actions in StudyShelf are done under a registered StudyShelf -account. The login uses e-mail address and password, keeping the process simple. Registration is free and can be done from the StudyShelf application.

The application offers some account setting management, meaning the user can change their account details when logged in. The account can also be deleted, which wipes out all of the data associated with it.

All of the accounts have an associated role either as a teacher or a student. The teacher accounts can create and manage courses, but other than that the features are largely the same. The user chooses the appropriate role at registration.

3.2 Study Materials

The main focus of the application is the Study Material functionality. Study Materials are stored files with associated tags, title and description. They are categorized under Courses, which are created by teacher accounts. The Study Materials are completely user generated content.

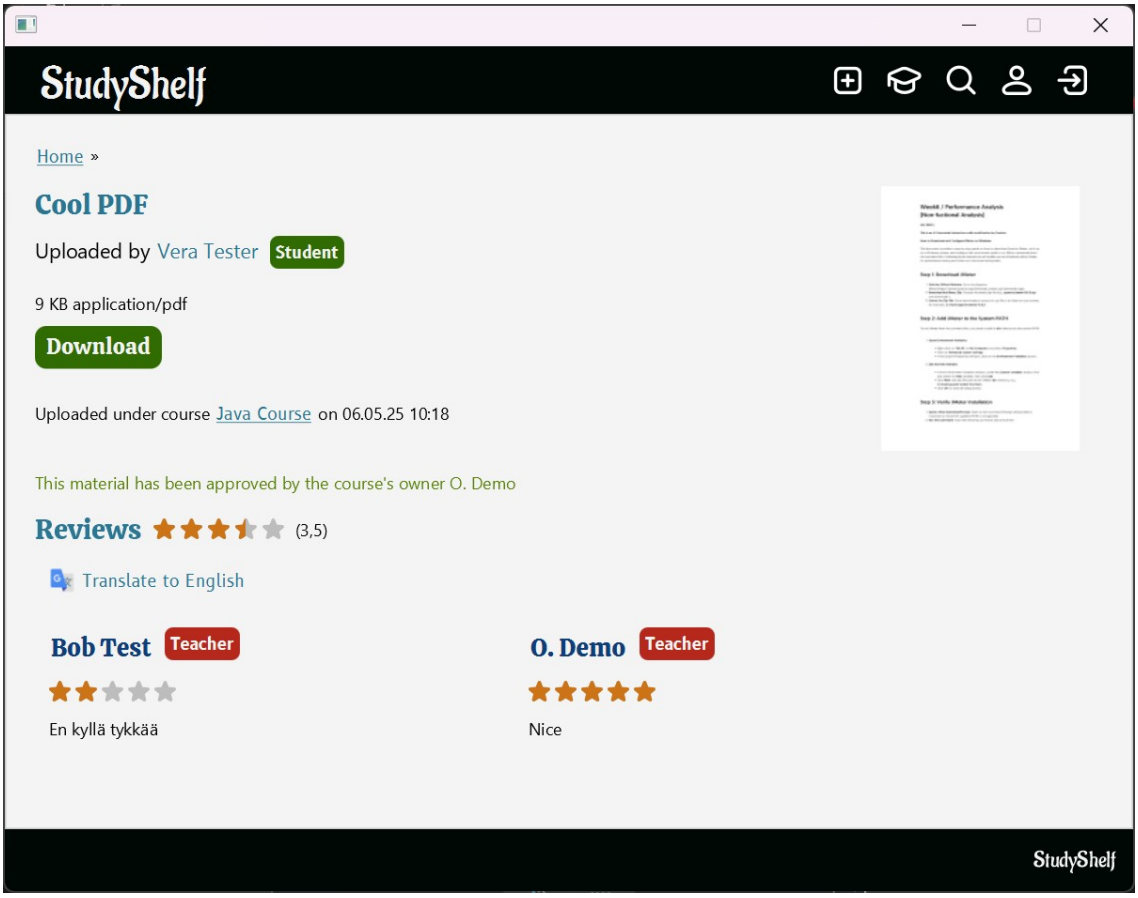

Fig 9. Example of a Study Material

3.3 Courses

Accounts with the teacher role can create and manage their own Courses. All of the study materials are uploaded under some Course. The teacher account that owns the course can manually pick and choose which Study Materials are appropriate for the course with StudyShelf's approval and rejection system.

3.4 Reviews and Ratings

In addition to the Course owner's approval, the Study Materials may receive ratings from all of the users on the platform. The reviews are visible to other users, enabling the community

to develop dialogue around which Study Materials are the most helpful. In practice, the ratings are given as 1-5 stars, with an optional text field to write to.

The average amount of stars is calculated and displayed along the Study Material. Using this value, the best reviewed materials are displayed on the home dashboard of every user.

3.5 Search

The application has a separate search page that allows the user to find Courses, Study Materials or tags by a specific query.

3.6 Home Dashboard

The landing page of StudyShelf displays relevant information to the user. If there are pending materials on a teacher account's course, they are displayed on top of the page. There are rows for both the newest and best rated materials on all of StudyShelf, as well as a row of materials that have recently been reviewed by the logged in user.

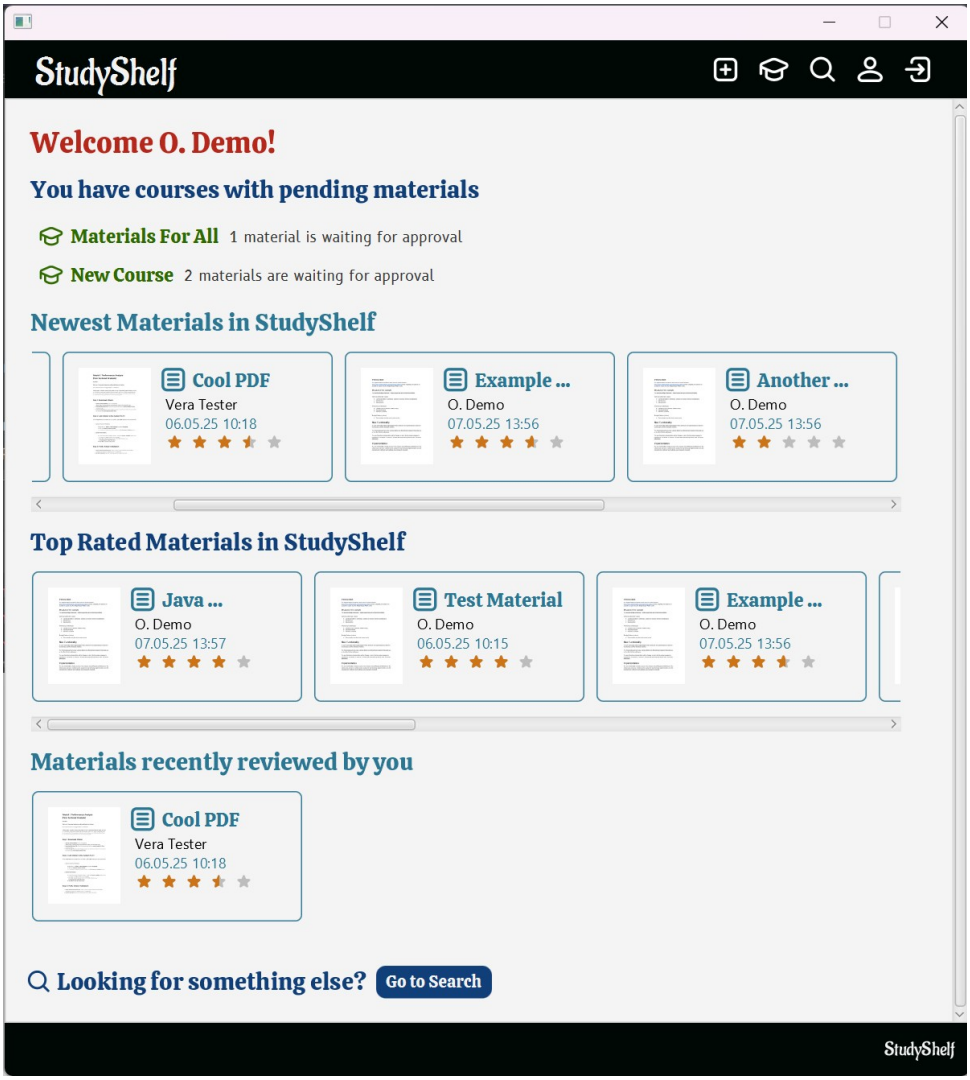

Fig 10. StudyShelf Front Page

3.7 Localization

As can be seen from figure 9, all reviews in another language can be translated to the language of the UI by Google Translate API. As for the UI, there are four supported languages for StudyShelf right now: English, Russian, Chinese and Finnish.

4. Implementation Details

4.1 Database implementation

4.1.1 Overview

StudyShelf's database implementation provides persistent storage for educational resources, user data, course information, and interaction metrics. The implementation follows a repository pattern with JPA/Hibernate for object-relational mapping.

StudyShelf uses MariaDB for its database needs. The database schema is designed to efficiently store and retrieve educational resources, user information, reviews, ratings, and course categorization.

4.1.2 Architecture

The database layer is structured around a base repository pattern with specialized repositories for each domain entity:

- **BaseRepository** - Provides generic CRUD operations for all entities
- **Entity-specific repositories** - Extend the base repository with specialized queries and operations

All repositories leverage Jakarta Persistence API (JPA) with Hibernate as the implementation provider, connecting through a centralized DatabaseConnection configuration.

4.1.3 Repository Structure

The repository layer consists of the following components:

- **BaseRepository** - Abstract class providing common operations like save, findById, findAll, update, and delete
- **CategoryRepository** - Manages course categories and their associated study materials
- **PermissionRepository** - Handles system permissions for role-based access control
- **RatingRepository** - Stores and retrieves user ratings for study materials
- **ReviewRepository** - Manages textual reviews of study materials
- **ReviewTranslationRepository** - Handles multilingual translations of reviews
- **RoleRepository** - Manages user roles and their associated permissions
- **StudyMaterialRepository** - Core repository for educational resources
- **TagRepository** - Manages tags for categorizing study materials
- **UserRepository** - Handles user account data and authentication

4.1.4 Relationship Management

The repositories handle complex entity relationships, including:

- One-to-many relationships (for example, User to StudyMaterials)
- Many-to-many relationships (for example, StudyMaterials to Tags)
- Cascading operations for entity deletion

4.1.5 Specialized Operations

Beyond basic CRUD, the repositories implement domain-specific operations:

- Finding materials by search terms
- Retrieving materials by status
- Managing user-specific content
- Handling multilingual content through translations

This database implementation provides the foundation for the StudyShelf platform, enabling secure and efficient management of educational resources and user interactions.

4.2 UI Implementation

4.2.1 Overall

The UI is coded with JavaFX 20.0.1. A few components (such as the header and footer) utilize .fxml files for display, but most of it is done purely programmatically. The main flow is controlled by a singleton SceneManager class, which handles moving between different screens.

In practice, the pages themselves are implemented as subclasses of a PageController class, in which they all share the same setPage() -method. SceneManager calls this method when needed. The SceneManager instance stores the current scene, which is a BorderPane with the header component on top, footer component on the bottom and the ever-changing center that can be changed by setPage().

4.2.2 Style

To ensure consistency, all of the styling follows the StudyShelf Style Guide (see 2.3) and most of the styling is handled with a single CSS -file. After code review, the CSS classes are never called by strings, but rather by specified variables of the new StyleClasses -utility class. This way, class names are never mispelled or misused.

4.2.3 Localization

The StudyShelf UI supports four languages: English, Russian, Chinese and Finnish. To ensure that the application knows which one to use, there is a LanguageManager singleton class that stores the current locale. As for now, the language can only be changed from the login screen.

The different languages themselves are handled by Java’s ResourceBundle.

5. Testing Strategy and results

We've implemented a multi-layered testing strategy that combines automated testing frameworks, manual evaluation techniques, and continuous code quality monitoring to identify and resolve issues throughout the development lifecycle.

StudyShelf testing strategy:

- Unit Testing
- Functional Testing
- Heuristic Evaluation
- Static Code analysis

5.1. Unit Testing

The JUnit tests cover domain and infrastructure layers. It consists of a total of 274 tests.

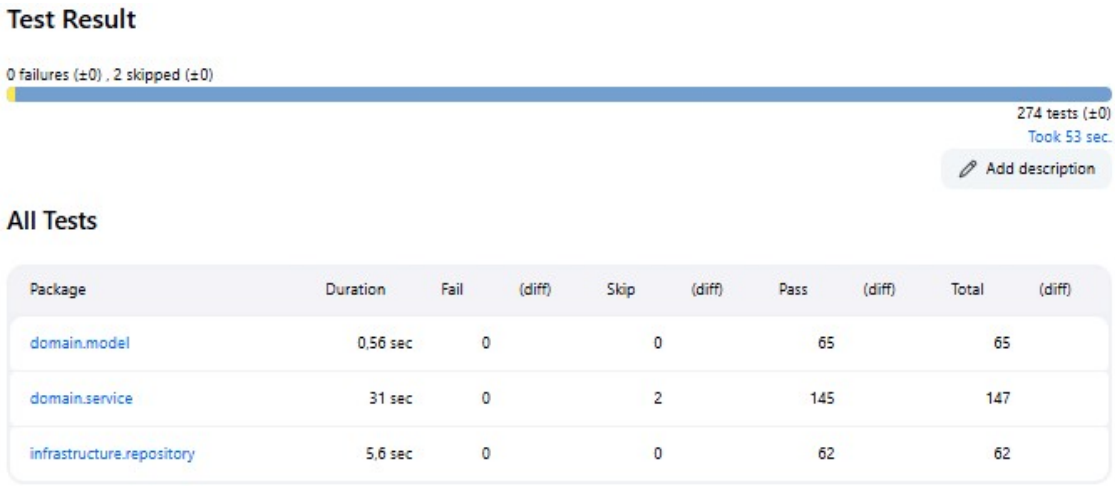

Fig 11. Test Report from Jenkins

5.2. Functional Testing

10 test cases were implemented, covering User Authentication and Profile Management, Study Material Management, Course Management, Localization and Permission System.

In test cases TC04 (Student downloads study materials of various sizes and checks system responsiveness and error handling) and TC05 (Student completes full account lifecycle with error handling and security validation) two major bugs were found.

- BUG-01 - Duplicate email registration attempt lacks proper error message and incorrectly redirects user to login page without explanation

- BUG-02 - System becomes unresponsive during large file download

Both were fixed during Sprint 8.

Tester	TC ID	Title	Status	Notes
Santtu	TC01	Student uploads study material, material gets rejected by teacher, student re-uploads a revised version of material, revised material receives approval from teacher; material remains hidden from others before approval	Pass	
Santtu	TC02	Student searches for study material using various filter combinations (Courses, Materials, Tags)	Pass	
Santtu	TC03	Student switches interface language and verifies review content translation accuracy and persistence	Pass	
Santtu	TC04	Student downloads study materials of various sizes and checks system responsiveness and error handling	Fail	to be clear, after reconnecting, it does respond normally: but it doesnt communicate that the download ever failed
Santtu	TC05	Student completes full account lifecycle with error handling and security validation	Fail	doesn't inform, yet also doesn't register duplicate account and lets proceed
Santtu	TC06	Student rates and comments on study materials, and verifies average rating updates across multiple actions	Pass	
Santtu	TC07	Student completes study material lifecycle: finds a specific course, uploads a material, edits its title and description, then deletes it	Pass	
Santtu	TC08	Teacher creates a new course and uploads study material	Pass	
Santtu	TC09	Teacher approves or rejects uploaded materials with repeated actions; material remains hidden from others before approval	Pass	
Santtu	TC10	Teacher finds, downloads, approves, and leaves a review on a pending material under their course	Pass	

Armas	TC01	Student uploads study material, material gets rejected by teacher, student re-uploads a revised version of material, revised material receives approval from teacher; material remains hidden from others before approval	Pass	
Armas	TC02	Student searches for study material using various filter combinations (Courses, Materials, Tags)	Pass	
Armas	TC03	Student switches interface language and verifies review content translation accuracy and persistence	Pass	
Armas	TC04	Student downloads study materials of various sizes and checks system responsiveness and error handling	Fail	Application window is unresponsive during download
Armas	TC05	Student completes full account lifecycle with error handling and security validation	Fail	System doesn't inform that the e-mail is already taken
Armas	TC06	Student rates and comments on study materials, and verifies average rating updates across multiple actions	Pass	
Armas	TC07	Student completes study material lifecycle: finds a specific course, uploads a material, edits its title and description, then deletes it	Pass	
Armas	TC08	Teacher creates a new course and uploads study material	Pass	
Armas	TC09	Teacher approves or rejects uploaded materials with repeated actions; material remains hidden from others before approval	Pass	
Armas	TC10	Teacher finds, downloads, approves, and leaves a review on a pending material under their course	Pass	
Jiayue	TC01	Student uploads study material, material gets rejected by teacher, student re-uploads a revised version of material, revised material receives approval from teacher; material remains hidden from others before approval	Pass	
Jiayue	TC02	Student searches for study material using various filter combinations (Courses, Materials, Tags)	Pass	

Jiayue	TC03	Student switches interface language and verifies review content translation accuracy and persistence	Pass	
Jiayue	TC04	Student downloads study materials of various sizes and checks system responsiveness and error handling	Fail	System becomes unresponsive during large file download. Download must restart from beginning after network interruption.
Jiayue	TC05	Student completes full account lifecycle with error handling and security validation	Fail	Duplicate email registration attempt lacks error message and incorrectly redirects to login page
Jiayue	TC06	Student rates and comments on study materials, and verifies average rating updates across multiple actions	Pass	
Jiayue	TC07	Student completes study material lifecycle: finds a specific course, uploads a material, edits its title and description, then deletes it	Pass	
Jiayue	TC08	Teacher creates a new course and uploads study material	Pass	
Jiayue	TC09	Teacher approves or rejects uploaded materials with repeated actions; material remains hidden from others before approval	Pass	
Jiayue	TC10	Teacher finds, downloads, approves, and leaves a review on a pending material under their course	Pass	

Table 3. Functional Testing Results

5.3. Heuristic Evaluation

The Heuristic Evaluation has provided insights in usability issues that weren't captured during functional testing, such as:

H1-5: Feedback

- When a search query doesn't return any results, it is not communicated to the user

H1-6: Clearly marked exits

- Application has no backtracking. When visiting any page, there is no way to go back to the previous page the user was on.

Most issues found during heuristic evaluation were fixed during Sprint 8.

Overall Heuristic Report			
Heuristic	**Total Amount of Issues Found**	**Total Severity of Issues Found**	**Overall Passability**
H1-1: Simple & Natural Dialog	0	0	Passed Well
H1-2: Speak the User's Language	1	0	Passed Well
H1-3: Minimize users' memory load	1	2	Passable
H1-4: Consistency	0	0	Passed Well
H1-5: Feedback	4	9	Poor
H1-6: Clearly Marked Exits	1	3	Not Good
H1-7: Shortcuts	2	4	Not Good
H1-8: Precise & Constructive Error Messages	5	6	Poor
H1-9: Prevent Errors	3	6	Poor
H1-10: Help & Documentation	2	3	Not Good

Table 4. Heuristic Evaluation Report Summary

5.4. Static Code Analysis

To monitor code quality metrics, SonarQube was used.

Key takeaways from SonarQube analysis

- Moderate test coverage across the codebase, with stronger coverage in domain models
- Some minor code duplication
- A few code smells
- No security vulnerabilities detected

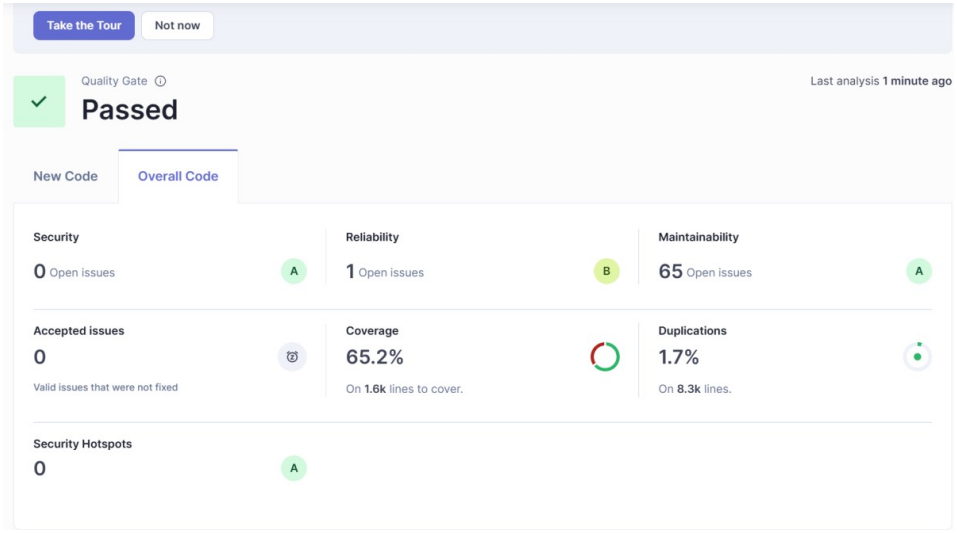

Fig 12. Sonar Qube Static Code Analysis

6. Installation Guide

6.1. Prerequisites

Before installing StudyShelf, ensure you have the following:

- Docker Desktop installed and running
- Windows X Server (for Windows users)
- Java Development Kit (JDK) 17
- Google API credentials for application authentication and translation functionality

6.2. Installation

Step 1: Set Up Windows X Server (Windows Users Only):

Download and install VcXsrv from SourceForge

Launch VcXsrv using the "XLaunch" application:

- Select "Multiple windows" and click "Next"
- Choose "Start no client" and click "Next"
- Ensure "Disable access control" is checked and click "Finish"

Set the DISPLAY environment variable in your terminal as such:

```
set DISPLAY=”Your-IP-Address”:0.0
```

Step 2: Set Up Google API Credentials - Configure OAuth authentication

Ensure you have a valid credentials.json file in the credentials folder
This file is used for Google OAuth authentication

Step 3: Set Up Google API Credentials - Configure translation functionality

Create or verify the translate-api.properties file in the credentials folder
The file should contain your Google Translation API key in the following format:

```
google.translate.api.key=YOUR KEY HERE
```

If you don't already have these credentials:

- Go to the Google Cloud Console (https://console.cloud.google.com/)
- Create a new project
- For OAuth: Enable the Google OAuth API and download credentials
- For Translation: Enable the Cloud Translation API and create an API key
- Place the files in the appropriate locations as described above

Step 4: Start the Application

Ensure Docker Desktop is running

Start the application by running the provided batch file: run-remote-studyshelf-full.bat
Wait for the application to initialize. This may take a few minutes on the first run as Docker downloads and builds the necessary images.

7. Usage Instructions

7.1 Registration and Logging In

Step 1.
Launch StudyShelf, the application opens on the Login Page.

Step 2.
If you already have a StudyShelf account with available credentials, skip to Step 4. Otherwise, click the “New Here?” link beneath the Login button to be redirected to the registration page.

Step 3.
Fill in the form. Each field is required. The E-Mail address has to be valid and can’t be already registered. The passwords need to match.

The Role options are Teacher and Student. Generally, if you’re unsure and not a teacher by occupation, you should pick the Student option.

Click the Sign Up -button to proceed. You’ll be brought back to the Login page.

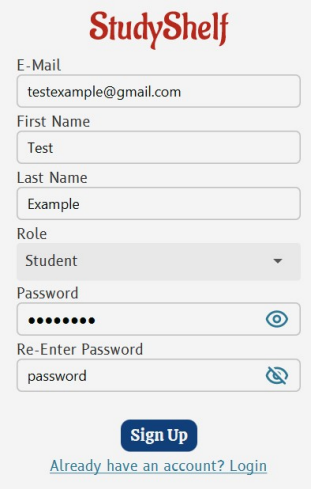

Fig 13. The Signup Form

Step 4.
On login page, enter the registered E-Mail address and the associated password. Click the Login -button. If the credentials are right, you'll be redirected to the Home Page.

7.2 Creating a Course (Teacher Account Only)

Step 1.
From the navigation bar, click the Create -icon. You'll be redirected to the Create -page.

Fig 14. Create -icon.

Step 2.
If you're logged in on a Teacher account, there's a Create Course -form. The only field is for the course's name. Fill this in. The character limit is 50 characters, and the course name can't be the exact same as another course owned by you.

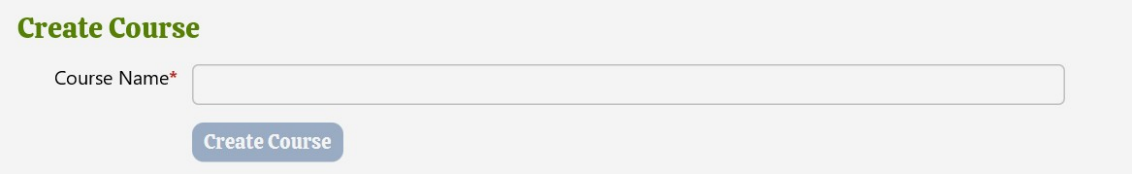

Fig 15. Create course -panel

Step 3.
Click the Create Course -button. You'll be redirected to the new course.

7.3 Creating a Study Material

Step 1.
There are two ways to navigate to the upload page: either by the Create -icon of the navigation bar (see figure 14), or clicking an “Add Your Material to this Course” -button on any course page. Click either. You’ll be redirected to the file upload page.

Step 2.
Fill in the form. The required fields are marked by a red asterisk.

First, you have to upload a file with the Choose File button. This will redirect you to your system’s file chooser.

Fill in the title and an optional description.

If you traversed to this page by a “Add Your Material to this Course” -button, that course is autofilled as the option for the Course field. You can change it if you wish. If you navigated from the navigation bar, you’ll have to choose the right course from the drop down menu.
To add tags, type in a tag you want to add and press enter. The tags that have been entered appear as chips below the field, and can be deleted from there.

To make the Upload -button available, all the required fields have to be filled and the final checkbox has to be checked. Click it to upload the material. You’ll be redirected to the new material.

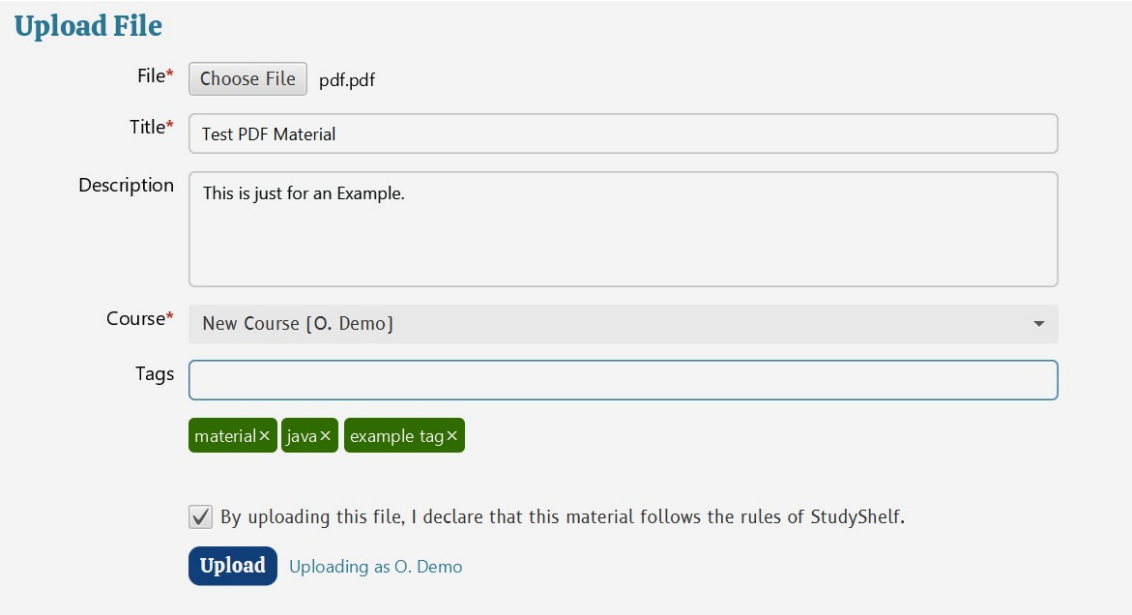

Fig 16. Example of a filled out Upload form

7.4 Downloading Materials

Step 1.
Navigate to a Study Material Page

Step 2.
Click the green Download -button over the description. A pop-up to choose the file destination will appear: choose where to download the file. After choosing the destination, a pop-up window with the download process will appear.

Step 3.
Once the download bar is full and turns green, the material has been downloaded to your system.

7.5 Approving or Rejecting Materials (Teacher Account Only)

Step 1.
When someone else uploads a material to your course, it's status is Pending. To approve or reject materials, you can see all the courses with pending materials on the front page. Click one of these courses to start.

Fig 17. Example of Courses with Pending Materials pop-up on the front page

Step 2.
You'll be redirected to the course page and see a list of the pending materials. You can either approve and reject them from here with the icons, or click the title to be redirected to the Study Material -page for further inspection.

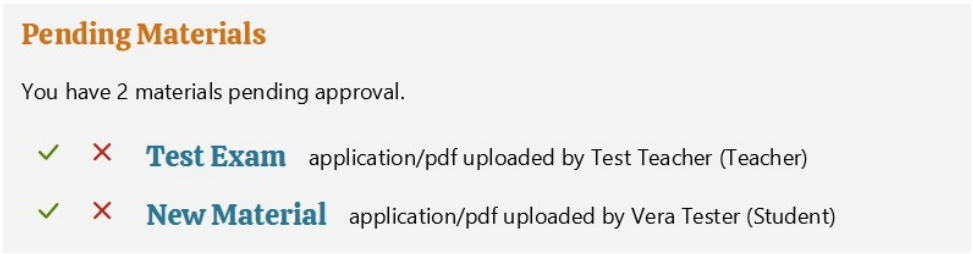

Fig 18. Example of Pending Materials pop-up on a Course Page

Step 3.
On the course page of a pending material, there is a pop-up to approve or reject the material. Click one of these buttons to change the status. Rejected materials aren't removed from the system, but won't be visible on the course page. Approved materials become visible on the course page.

The material's status is updated on the Material Page and is visible for other users as well.

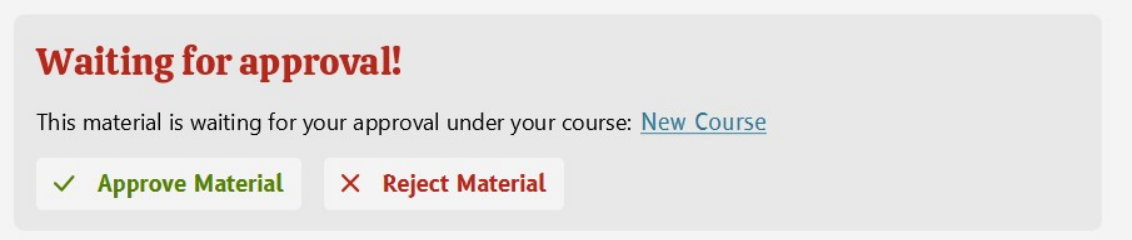

Fig 19. Example of approval pop-up on a Material Page

7.6 Reviewing Materials

Step 1.
Navigate to the page of the Study Material you want to review. You can use the search functionality. Scroll to the bottom, there is a Review Section. Click the Stars to set the right amount of stars. The lowest amount is 1 star, the highest 5 stars. Clicking a Star will make the Send Review -button usable.

Step 2.
If you want, you can add a text review by writing something to the text field. If you type your review in the language of the UI, it can be translated by other users. Click Send Review -button when you're happy with your review.

Fig 20. Example of a correctly filled Review form

Step 3.
You can verify that your review got sent correctly by seeing it appear under the Review section.

Fig 21. Example of a review

7.7 Deleting Your Content

Step 1.
Navigate to the Profile -page by clicking on the Profile -icon of the navigation bar.

Fig 22. Profile Icon

Step 2.
All of your content is displayed under the links under My Profile. In the list view, each entry has a red trash can next to it. By clicking it, the content will be removed from the system. This can be done on any kind of content you've left: Materials, Courses and Ratings. Click the deletion button.

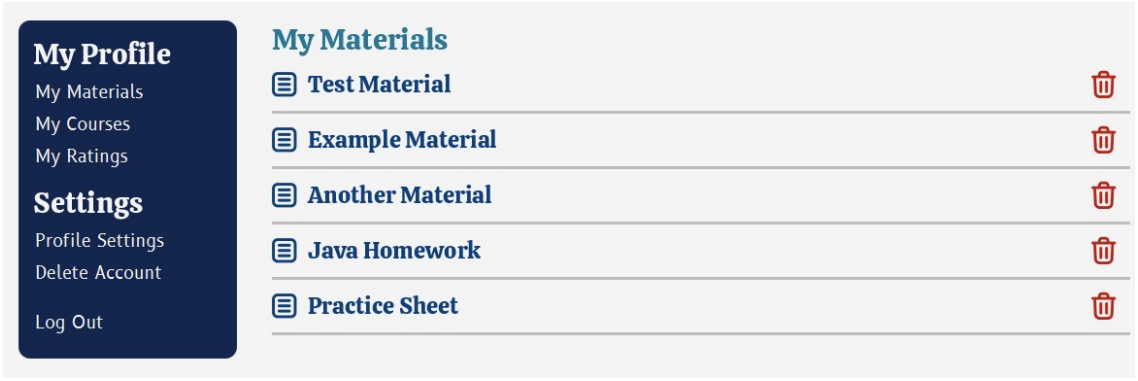

Fig 23. The My Materials -page with visible deletion buttons

Step 3.
When deleting something, there's always a confirmation panel. Reread the prompt to check that you're deleting the right thing. Click the Yes -button if so. Note that none of the deleted items can be recovered: Once you've deleted them, they're gone forever.

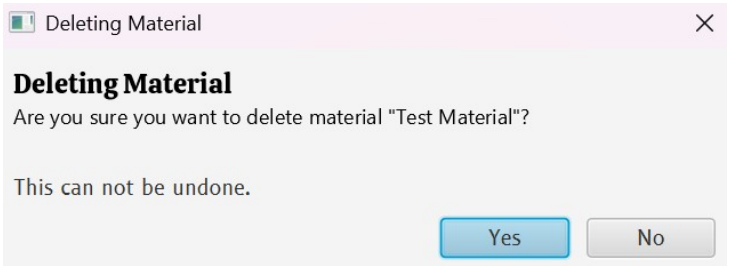

Fig 24. Deletion prompt message

7.8 Deleting Your Account

Step 1.
Navigate to the Profile -page by clicking on the Profile -icon of the navigation bar (see figure 22).

Step 2.
From the side menu, select "Delete Account". You'll be redirected to a Delete Account -page.

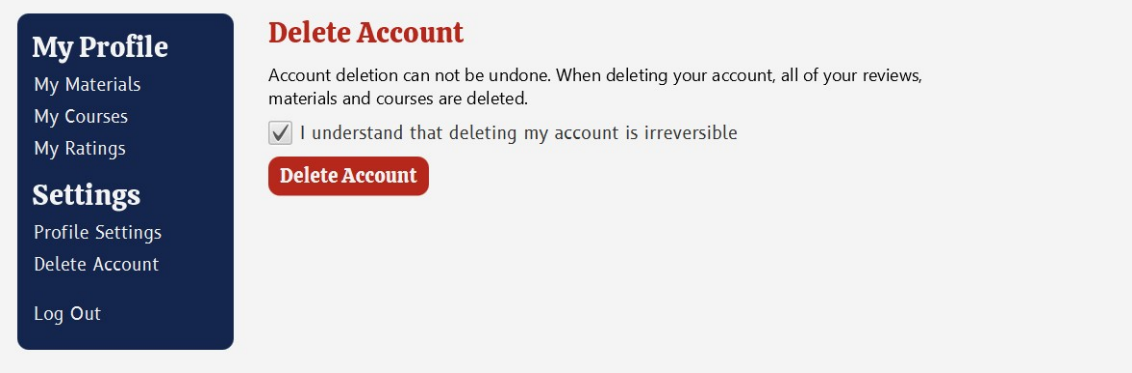

Fig 25. Delete Account page

Step 3.
If you're sure you want to delete the account, check the checkbox and click the Delete Account -button. StudyShelf won't retain any of your data, all of your content will be permanently deleted. You'll be redirected to the Login -page.

8. Troubleshooting

8.1 Environment Setup Issues

8.1.1 VcXsrv Connection Problems

Issue: Application fails to display GUI or shows "Cannot connect to X server" errors

Solution:
Set the DISPLAY environment variable in your terminal as such:

```
set DISPLAY="Your-IP-Address":0.0
```

Ensure VcXsrv is running with "Disable access control" checked.

8.1.2 Docker Container Not Starting

Issue: Docker container fails to start or exits immediately
Solution: Check Docker Desktop daemon is running

8.1.3 Credentials Path Issues

Issue: "Failed to initialize Google Drive service" error message
Solution: Ensure credentials.json exists in the correct location:

```
dir src\main\resources\credentials\credentials.json
```

If missing, obtain valid Google API credentials and place them in this directory (see installation guide 6.2.)

8.2 Application Runtime Issues

8.2.1 File Upload/Download Failures

Issue: "Failed to upload/download file" or timeout errors
Solution: Check your internet connection

8.2.2 Authentication Issues

Issue: "You do not have permission" errors
Solution: JWT token may have expired. Try logging out and back in

9. Support and Contact Information

In case any questions regarding the project arise, feel free to contact one of our developers

Armas Nevolainen, Backend Developer and DevOps
Armas.Nevolainen@metropolia.fi

Santtu Saaranen, UI developer and designer
SanttuAmi.Saaranen@metropolia.fi

Jiayue Zheng, Backend Developer
Jiayue.Zheng@metropolia.fi