

StudyFlow

Group 7 – Kachow

- Hilda Hermunen
- Veera Ruotsalainen
- Patrick Scott

Table of Contents

Introduction

Project Vision and Goals

Application Features and Usage (including a demonstration)

Technologies used

Software Architecture

Localization (UI / DB)

Quality Assurance (Tools and Results)

Functional Testing

Non-functional Testing

Lessons Learned

Relevant Links:

- [GitHub](#)
- [Trello](#)

Introduction

Managing time effectively is a key challenge for students balancing coursework, assignments, and other responsibilities.

Not all courses share the same space to view content from, making it harder to keep track of everything.

The users can enhance their time management through a practical application that visualizes the workload.



Project vision and goals

The project aims to provide an easy to use timetable application to help students organize courses, class schedule, assignments and deadlines.

In long-term, encourages better time management habits and improve academic performance by enabling better and more consistent planning.

Application features and usage, demonstration

Features:

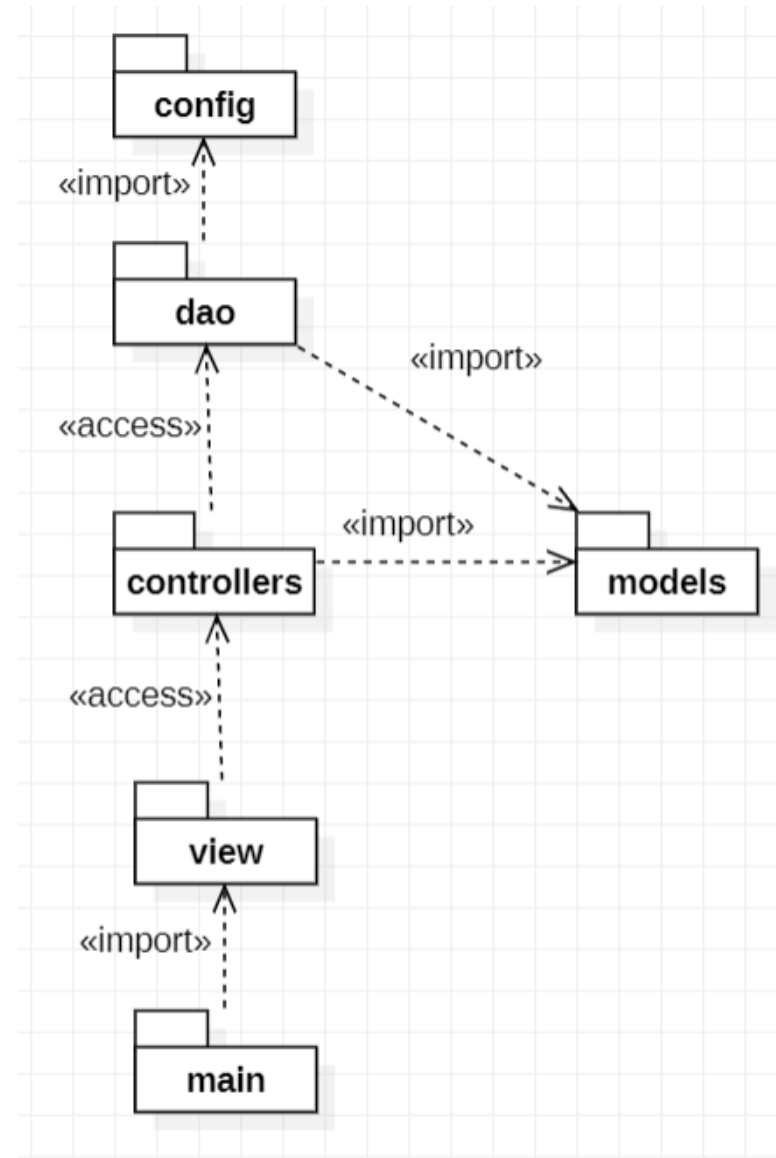
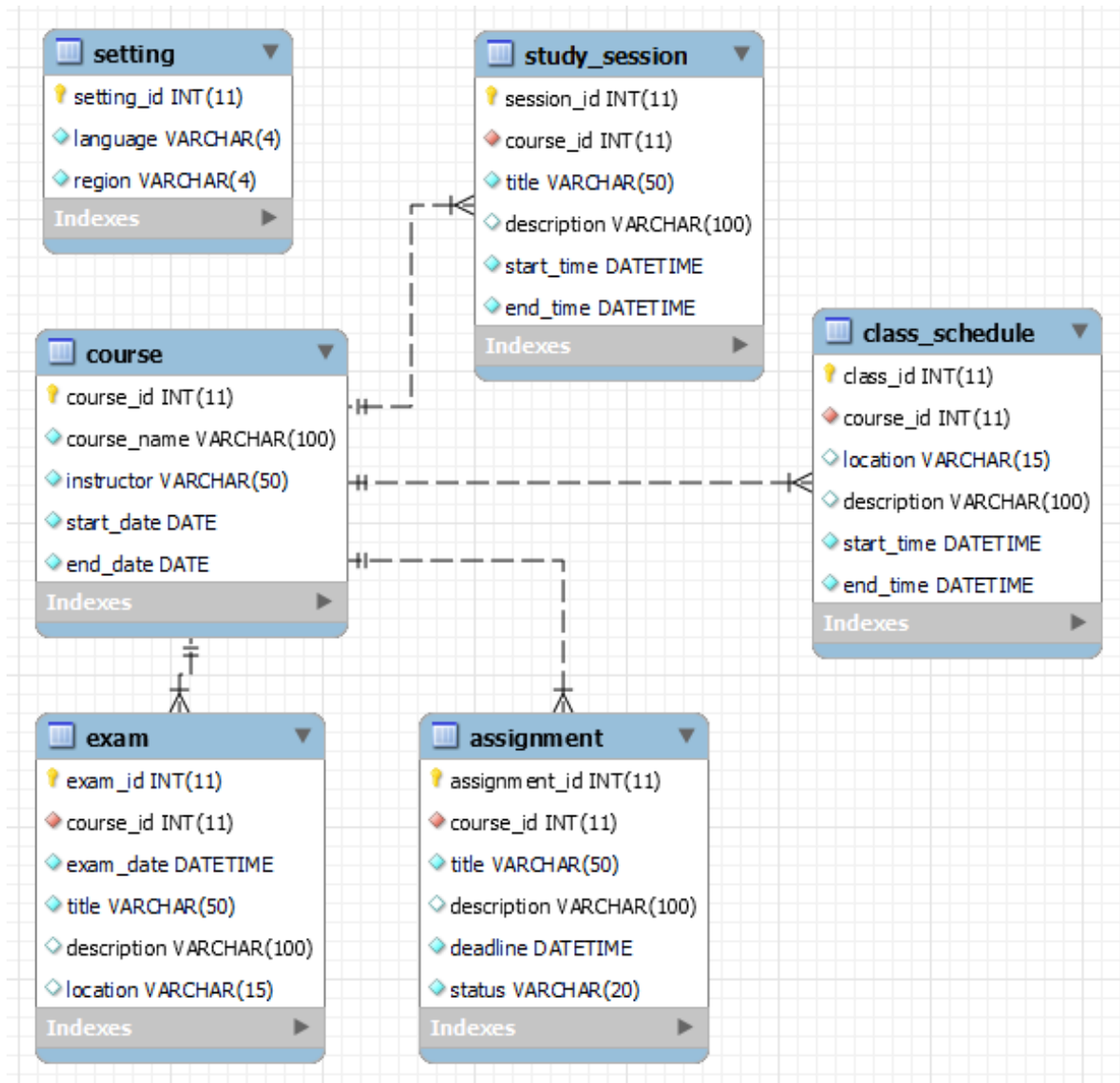
- Adding, updating and deleting courses, class schedule, assignments, exams and study sessions on the timetable.
- Changing and saving a language setting.
 - Supports English, Korean, and Arabic.
 - Includes right-to-left (RTL) layout adaption.
 - The saved language setting is retrieved from the database when the program is started again.

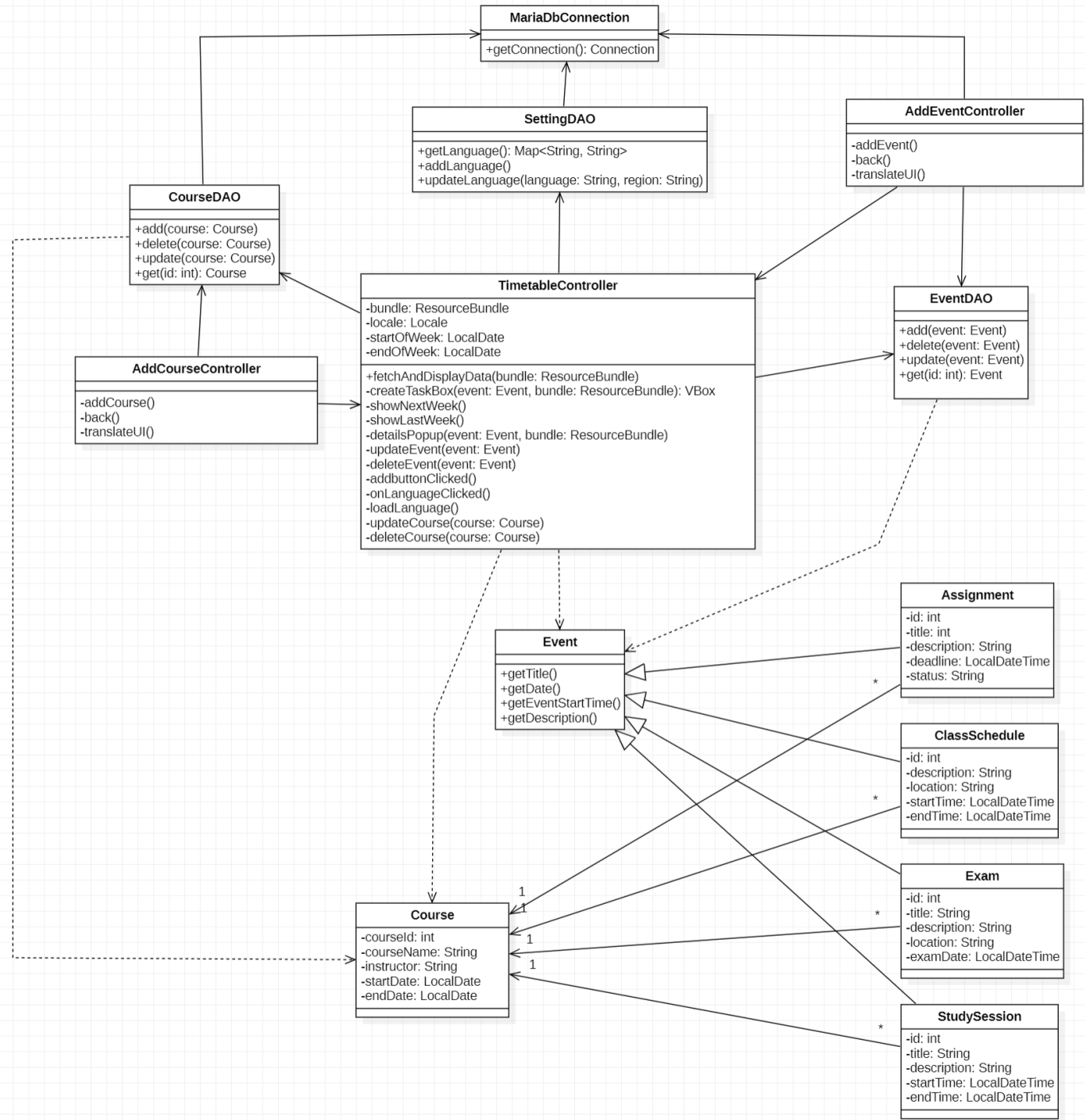


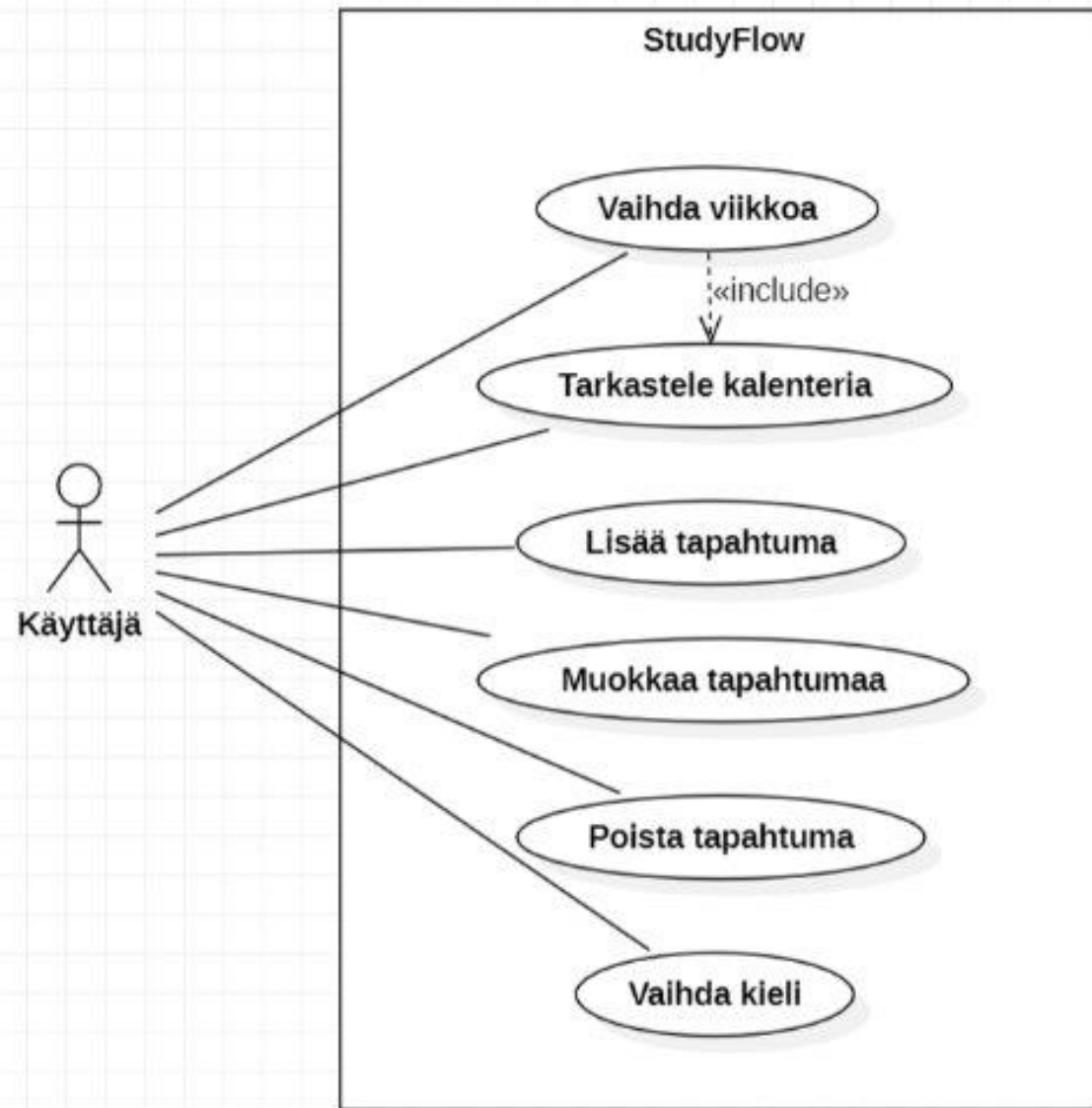
Technologies used

- IDE – IntelliJ
 - UI/UX prototype – Figma
 - UI/UX design – Scene Builder
 - Database – MariaDB (SQL)
 - Backend – Java
 - Frontend – JavaFX, FXML
 - Styling – CSS
 - Database connection – JDBC
 - Dependency management – Maven
 - Unit testing - JUnit
 - Testing automation – Jenkins
 - Code coverage testing – JaCoCo
 - Deployment – Docker
 - Code review & quality analysis – PMD, SonarQube
 - Performance testing – JMeter
-

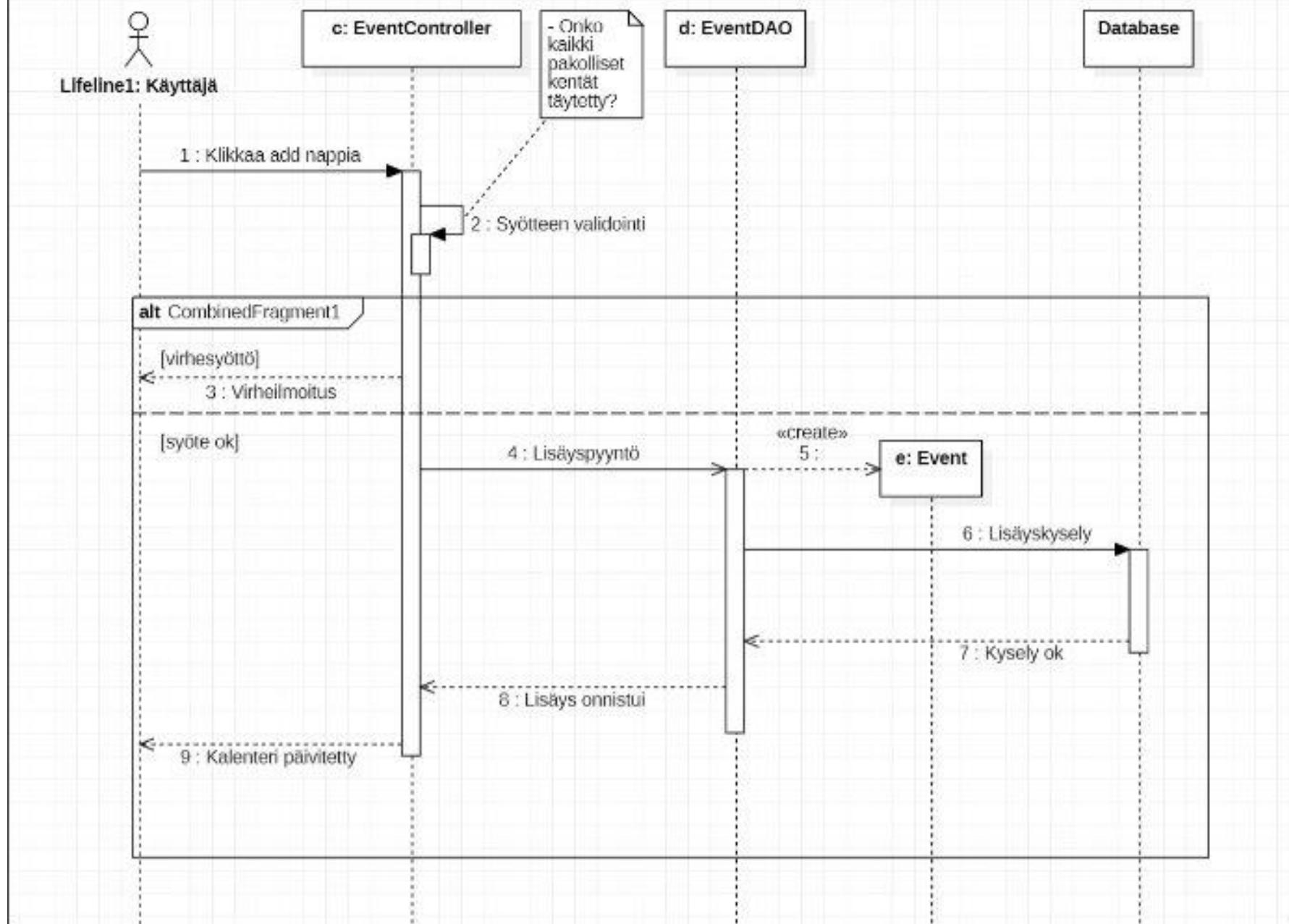
Architectural Design







sd lisää tapahtuma



Localization

- Localization was implemented by translating the application's text.
 - Implemented multilingual support: English, Korean, Arabic.
 - Right-to-left (RTL) layout adaption for Arabic.
 - The project uses ResourceBundle to load language-specific properties files.
- The application saves user's selected language preference to the database.
 - This allows the application to load the user's preferred language on startup.

Quality assurance – Tests, Tools & Issues found

Unit tests

- JUnit
- Focused on model classes. All tests passed

Testing Automation

- Jenkins

Coverage Testing

- JaCoCo
- Results: 6,8%

Code Review and analysis

- SonarQube
 - 119 Maintainability issues, 33 Reliability issues & 1 Security issue
- PMD
 - Priority level 1 violations: 2
 - Priority level 3 violations: 81
 - Priority level 4 violations: 25

Performance testing

- JMeter
- Load testing showed no significant issues



Quality assurance - Solutions

- Code refactoring and changes were made based on SonarQube & PMD reports.
 - SonarQube: Initial 157 issues were reduced to 43.
 - PMD: Initial 108 issues were reduced to 4.
 - Refactoring was implemented to improve code maintainability and reduce code duplication.
-

Functional testing

- 10 tests cases
 - Adding, updating and deleting a course
 - Adding, updating and deleting an assignment
 - Adding an event month ahead
 - Changing the application language
 - User being able to receive error messages for invalid or missing input fields
- 2 tests failed at first. The user was not able to modify or delete a course.
- After fixing the issues all the tests passed.

Non-functional testing

- Heuristic evaluation was done by all team members, and the results were combined.
- 4 different issues were prioritized. These issues were resolved by adding missing functionality to the application.

No	Heuristic	Description	Severity
1	H1-5 Feedback	No clear feedback message after an event is added, updated or deleted.	3
2	H1-6 Clearly marked exits	No confirmation dialogue when user deletes or updates an event.	3
3	H1-9 Prevent errors	No possibility to delete or update a course after it's added to timetable.	4
4	H1-10	No help tooltips or user guide.	2

Lessons learned

- The importance and implementation of UI and DB localization
- Value of functional testing and creating test cases
- Non-functional testing
- Code analysis using SonarQube and PMD
- Performance testing using JMeter

Challenges

- Database localization created challenges. These were overcome by creating an alternative solution with teacher.
- SonarQube installation

Relevant Links

- [Github](#)
- [Trello](#)

Questions

