Ryhmä 7: Hilda Hermunen, Patrick Scott, Veera Ruotsalainen

Sprint 5 review

Scrum master: Veera Ruotsalainen

Note: Only group members Hilda and Veera participated in Sprint 5

Trello board status

Trello board for Sprint 5 included user stories and backlog related to localization.

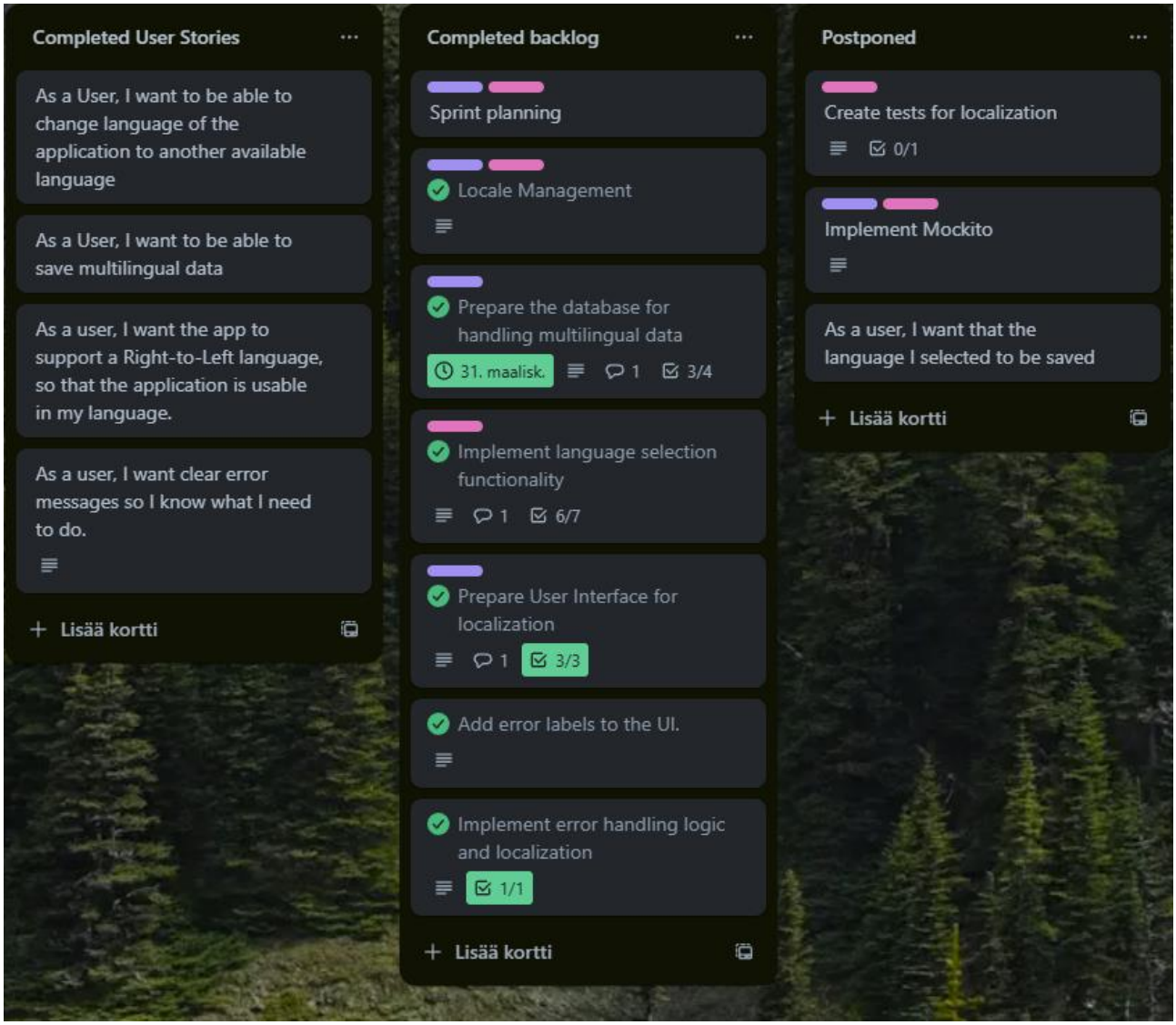


Figure 1: Screenshot of the Sprint 5 Trello board

As shown in the figure 1 most of the planned tasks were completed. Tests for localization, implementing Mockito and functionality where the user's selected language is changed were

postponed. During the sprint a user story, “As a user, I want clear error messages, so I know what I need to do” and backlog, “Add error labels to UI” and “Implement error handling logic and localization” were added. All these were completed during the sprint.

Time spent by each member

The total hours spent during the sprint 5 was 24 hours. The tasks were divided evenly between team members.

Team member	**Assigned Task**	**Time Spent**
Veera	Locale Management	1 hours
Veera	Database Localization	4 hours
Veera	Prepare User Interface for localization	5 hours
Veera	Implement error handling logic and localization	4 hours
Hilda	Locale Management	2 hours
Hilda	Implement language selection functionality	8 hours

Sprint backlog status

Completed backlog:

- Locale Management
    - Identified and created resource bundles for different languages.
    - Example: Added messages_ar_AE.properties for Arabic localization.
    - Mapped localized strings to their corresponding keys in the resource bundles.
- Prepare database for handling multilingual data (partially completed)
    - Ensured that database tables and columns support UTF-8 encoding to handle multiple languages.
    - Updated database_creation.sql to use utf8mb4 character set and collation for all tables and columns.
    - Verified that the database can store and retrieve localized data correctly.
    - Updated course primary key to be an auto increment id.
- Implement language selection functionality
    - Modified controllers to load and apply localized strings dynamically.

    - Example: AddTaskController now uses a ResourceBundle to set UI text based on the selected language.
- Prepare user interface for localization
    - Implemented localization for the user interface by adding support for multiple languages.
    - Updated FXML files to use localized strings from resource bundles.
- Add error labels to UI
- Implement error handling logic and localization

Postponed backlog:

- Implement Mockito
- Database localization (partially)
- Create tests for localization

Obstacles encountered

The team had difficulties understanding the level of required localization. The misunderstanding led to the team not completely fulfilling the task database localization. This will be corrected at the beginning of the next sprint.

Sprint 6 planning

Scrum master: Patrick Scott

Goals

The goal for sprint 6 is to finish, review and clean-up the code, and plan acceptance testing.

1. Code Review and Quality Analysis

   Using static code analysis tools, such as Checkstyle, PMD or FindBugs to analyze codebase. The goal is to have the least amount of code violations and potential errors as possible, and to improve readability and optimization.

   Analyzing the metrics such as complexity, duplicate code and potentially unreachable blocks of code.

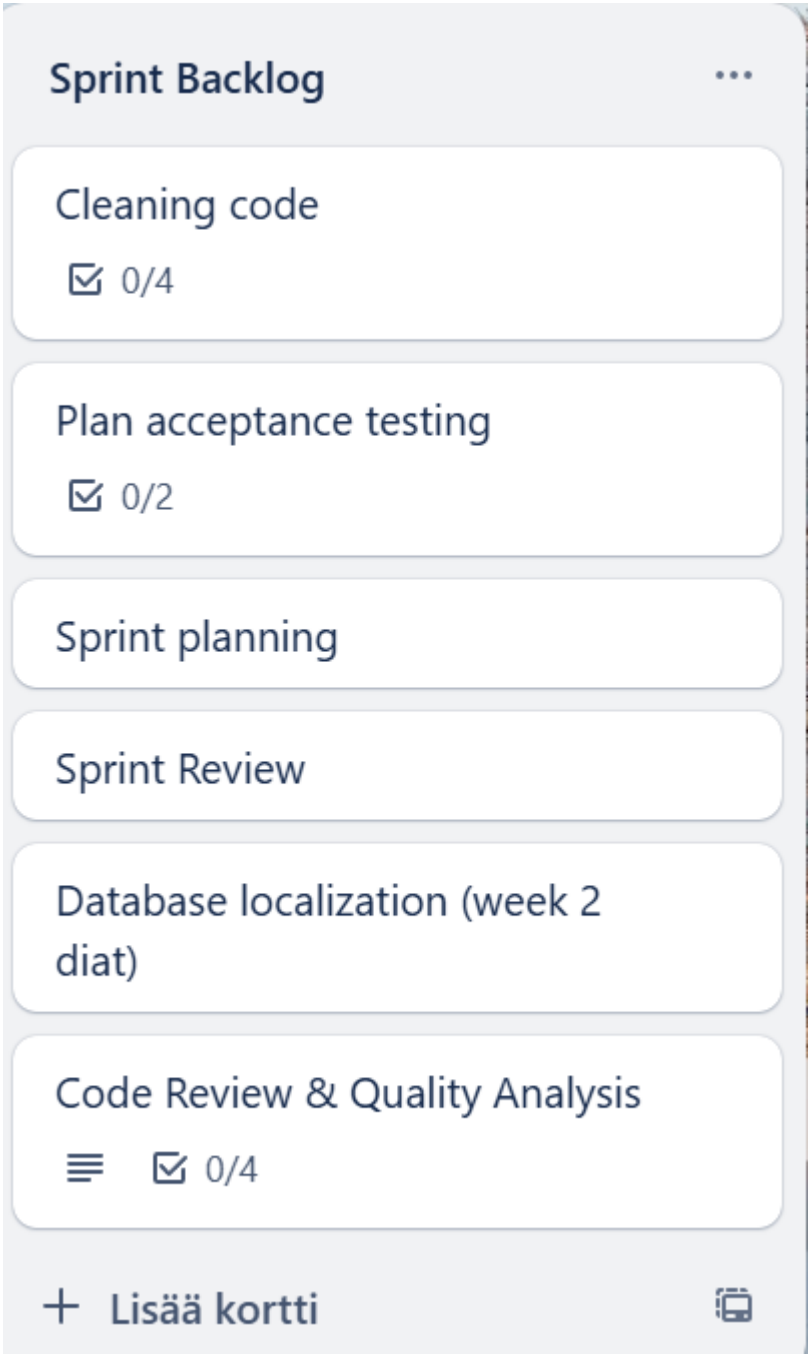
Documenting the findings in a document that includes a summary of key metrics, explaining for any significant findings and screenshots or code snippets for illustration.

2. Code clean-up

   Improving code according to the findings from the previous step. Refactoring any functions with high complexity into smaller, more manageable units. Removing redundant and duplicate code. Adding comments and documentation to code.

3. Planning Acceptance Testing

   Creating a list of acceptance criteria, which will be used as a point of reference for testing. Designing the acceptance tests that validate the core functionalities of the program. Writing a formal report of the plan.



Trello board includes the goals for the sprint. The goals are split into smaller tasks.

Tasks

Statistical Code Review (Hilda, Patrick, Veera)

Estimated time: 5-8 hours

1. Run Static Code Analysis Tools: Use tools such as Checkstyle, PMD, or FindBugs (or another instructorapproved static analysis tool) to analyze your codebase for
    - Assigned to all team members
2. Analyze the Metrics:
    - Extract and document key metrics
    - Assigned to all team members

3. Document Findings:
    - Summarize the results from your analysis tool, listing any areas of concern and suggested changes
    - Assigned to all team members

Code Clean-Up (Hilda, Patrick, Veera)

Estimated time: 6-12 hours

Code Clean-Up

1. Refactor Based on Review:
    - Refactor any functions with high complexity into smaller, more manageable units.
        - Divided evenly between team members
    - Remove redundant or duplicate code.
        - Assigned to Patrick
    - Simplify variable names and method signatures where applicable.
        - Assigned to Hilda
    - Ensure consistent code formatting (spacing, indentation, etc.). Use tools such as lint
        - Veera
    - Add comments and Documentation
        - Assigned to Veera and Hilda

2. Check Code Quality Standards:
    - Ensure that your code adheres to Java coding standards and best practices, focusing on
    - Assigned to all team members

3. Verify Code Changes
    - Run unit tests to ensure that the code refactoring did not introduce any new bugs or errors.
    - Assigned to all team members

Acceptance Testing (Hilda, Patrick, Veera)

No need to conduct the actual test just plan.

Estimated time: 5 hours

1. Define Acceptance Criteria
    - Review the original project requirements and any additional specifications you have established with your team. Use these to create a list of acceptance criteria, which will serve as the benchmark for testing.
    - Done together as a team

2. Design Acceptance Tests
    - Functional tests (testing core features and interactions)
        - Assigned to Hilda
    - Usability tests (ensuring that the user interface is intuitive)
        - Assigned to Veera
    - Performance tests (verifying response times and efficiency)
        - Assigned to Patrick