

**Team 6 - Sprint 7 review report**  
**(Ilkka Sinkonen, Daniel Juntunen, Veikko Jokinen)**

**What has been done:**

- Store user's preferred language to the database. Preferred language is hard coded to the database at the moment.
- Code cleanup and refactor
- Docker image has been recreated and functions as intended
- SonarQube report has been refreshed and added into the github repository
- JMeter performance test report metrics has been created
- Acceptance testing criteria has been tested and successfully met
- Heuristic testing has been conducted (Implementations not made yet)
- Finished BCrypt hashing for passwords

**User Stories completed in specific:**

- Users should be introduced to the application in their last chosen language, which is fetched from the database

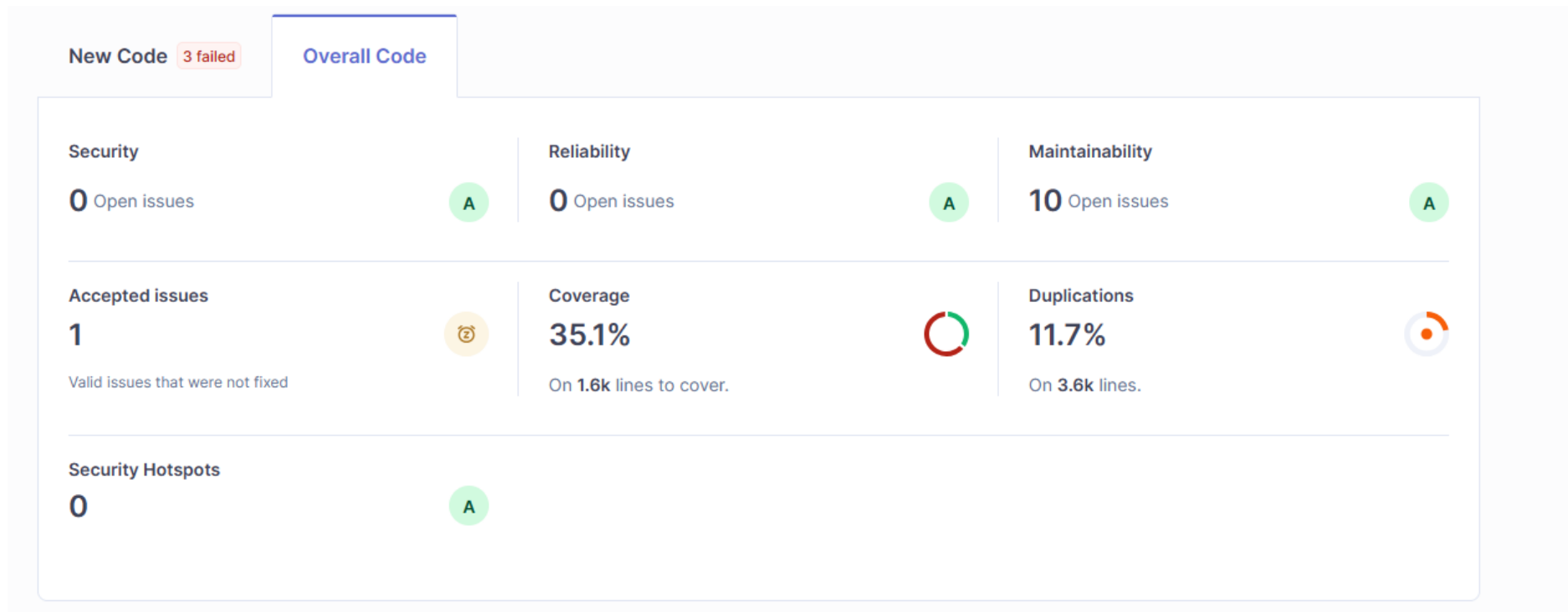
**What is in progress:**

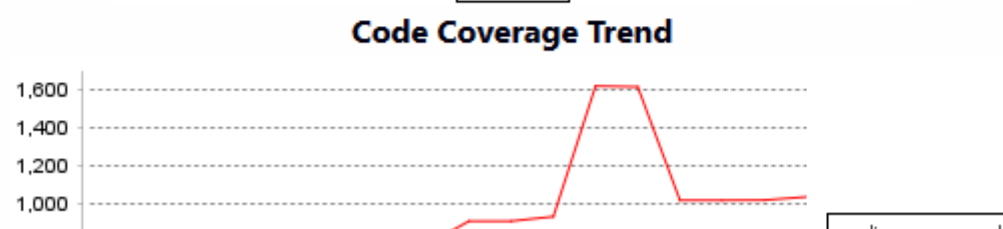
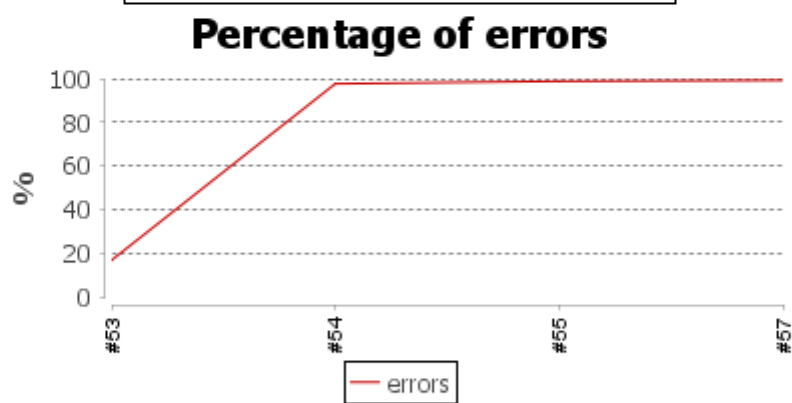
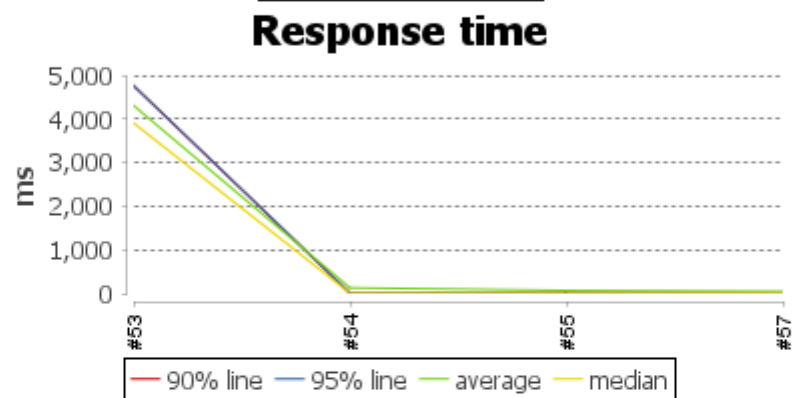
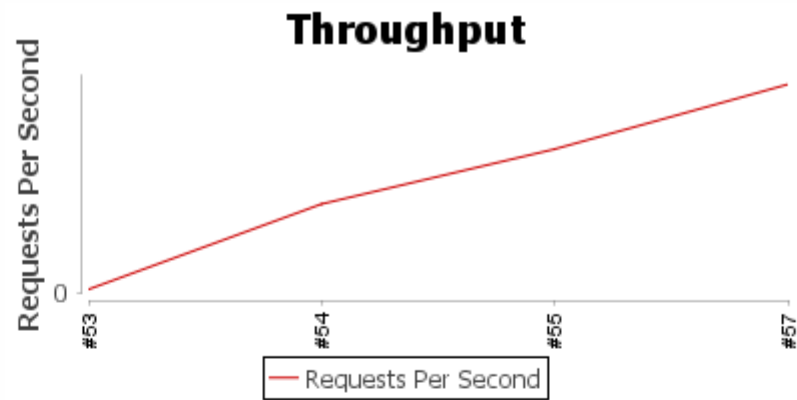
- Implementing localization to the database, when user selects preferred language
- Localize login page from system language, if language is supported
- Refactoring DAO to be properly aligned with database elements

**Obstacles we faced:**

- Applying preferred language, when user's log in

- Due to BCrypt hashing, this process locked most of our accounts out due to the passwords not being previously hashed
- JMeter seems to be pretty unusable when conducting performance testing (Except database load), currently in discussion about using JMS and/or TestFX with unit testing to get load times.





<i><b>Tiimiläisen Nimi</b></i>	<i><b>Annetut tehtävät</b></i>	<i><b>Aikaa käytetty</b></i>
<b>Veikko Jokinen</b>	Persian language localisation, Jenkins pipeline implementation, refreshed docker image, refreshed SonarQube report, JMeter performance test.	10 tuntia
<b>Daniel Juntunen</b>	Refactoring, bug fixing, userCourseJoin dao + controller implementation, show user enrolled courses, minor localization work, password decoding for user & staff	10 tuntia
<b>Ilkka Sinkonen</b>	Code cleanup, bug fixing, dao refactoring, Database Localization	10 tuntia

<b>No</b>	<b>Heuristic</b>	<b>Description of the Issue</b>	<b>Screenshot</b>	<b>Severity</b>	<b>Suggested Improvement</b>
1	H1-1: Simple & natural dialog	In the admin panel, the “View course” button is misleading because it shows all courses, not only one.		1	It should be renamed to “View courses”
2	H1-2: Speak the users’ language	The user must manually select their preferred language every time.		2	Store the user’s selected language and automatically apply it when user logs in.
3	H1-3: Minimize users’ memory load	Currently when checking attendance, students are identified by ID. Admins then have to remember this ID and then navigate to students list to match ids to get students name.		3	In attendance list, add student names.
4	H1-4: Consistency	Main interface profile and course page have empty space which is not consistent and		1	Add alternating row colors (e.g., light gray for every other row) to improve readability

		<p>there is no empty space in the admin panel</p> <p>In create courses view, input “Attendance code” doesn’t match with course lists “Attendance password”</p> <p>In the Administrator panel, rows are not visually separated from each other</p>			<p>Center the elements in maininterface or add details to fill the empty space</p> <p>Rename input field to “Attendance password”</p>
5	H1-5: Feedback	Creation pages do not prompt with anything that the item has been created, only refreshes the datalist		2	Give the user feedback if the action was succesful or has failed. This could be a popup or a simple text.
6	H1-6: Clearly marked exits	The application lacks a logout option, preventing the user from securely ending their session.		2	Implement a logout function that allows the user to safely sign out while maintaining security.
7	H1-7: Shortcuts	Login page doesn’t allow users to use the “Enter” key on the password field to immediately login		2	Add “Enter” functionality to login page’s password field for immediate login
8	H1-8: Precise & constructive error messages	Admin panel has virtually no errors to show the user if there have been issues with the item creation		2	Implement a custom error handler, which prompts the admin which fields have issues/what's wrong with the attempt to create an item
9	H1-9: Prevent errors	When the user creates a new user or course, the interface does not validate the form fields. Required fields can be left empty without any prevention		2	Implement form-validation that prevents saving empty fields to the database
10	H1-10: Help and documentation	The program lacks documentation that provides		3	Create documentation

		instructions on how to use the software.			
--	--	--	--	--	--

## Acceptance Criteria:

- Students can login
  - Staff can login (Teachers)
  - Users can review their own information
  - Information labels change depending on the user (Student/Staff)
  - Staff can review the attendance page for a course
- 
- Staff with admin credentials are able to view the Admin Panel
  - Administrators (Staff with admin credentials) are able to create courses
  - Administrators are able to create new students
  - Administrators are able to create new teachers
  - Administrators are able to create new degrees (Roles)
  - Administrators are able to view raw data about courses, staff, students, degrees and attendances
  - Administrators can modify data about staff, students, courses, degrees

<b>User Story</b>	<b>Acceptance Criteria</b>	<b>Test Case</b>	<b>Expected Outcome</b>
Student login	Students can login to the application	Enter valid username/password	Student is directed to the main interface
Teacher login	Teacher can login to the application	Enter valid username/password	Teacher is directed to the main interface
Teacher student registering	Teacher can register a new student	Validate inputs and check if the new student is added to the database	A new student is added to the database

Teacher course creation	Teacher can create a new course	Validate inputs and check if the course is added to the database	A new course is added to the database
Profile view	Users can view profile information	Login with user credentials, click course tab and validate that all user information is displayed	All user information is displayed in the profile view
Course view	Users can view all courses on the course tab	Login with student credentials, click course tab and validate that all courses are visible	All courses are visible
Perfomance	The dashboard in administrator view loads raw data and updates the view, when creating, deleting or updating data and loads in < 2s	Load raw data, update data, Delete data, Update dashboard	Data is visible in dashboard and its updating, when creating, deleting or updating data

### Acceptance Testing After Summary:

All the criteria that were previously mentioned have been successfully met by conducting all test cases. All the requirements set are met to the best of our abilities.

### Summary of security testing findings:

Using SonarQube, it was found that we have a single security issue in our application. The issue being the lack of protection for the database password. For the sake of the project and how we have been instructed to use Jenkins' pipeline, we have deemed this issue to be void. The issue is now listed as “Accepted” in all of our SonarQube reports.

## **UI/UX Improvements and optimizations:**

Refactored mainInterfaceController's functions to use controllers.



## Links:

Trello Board:

<https://trello.com/b/5WAXql7T/product-backlog-board>

Project Plan:

[https://docs.google.com/document/d/1O\\_g-zJz1wV71\\_avTI1PDZ-5DXelolw9V7ZhQsQWpBLM/edit?tab=t.0](https://docs.google.com/document/d/1O_g-zJz1wV71_avTI1PDZ-5DXelolw9V7ZhQsQWpBLM/edit?tab=t.0)

Product Vision:

<https://docs.google.com/document/d/1O3U6CPSf3AjObtdmFEt4F05RRHxHQHc2p4wla8N3JY/edit?tab=t.0>

Figma UI Blueprint/Prototype

<https://www.figma.com/design/2FSC3hXvOkFGg2zG9coUL4/SchoolApp?node-id=0-1&p=f&t=2K1rck9SdNKwUYTN-0>

Project repository:

<https://github.com/TheZink/OTP1>