**Software Evolution and Maintenance**

**FCAI-CU-SWE 2022**

You are given a Java project University Timetabling (<https://sourceforge.net/projects/unitime/>) as a base version, you are the maintenance team that needs to implement changes to the existing software. We will only focus on the Java backend.

The change request to the project can either be a bug fix, change in an existing feature or addition of a new requirement. Motivate the need for the change that you have proposed.

**Project Implementation**

* You should work in teams of no more than 4 and no less than 3 From the same lab.
* **The project is organized into two phases**

Phase 1:

1. Create a GitHub repository and all team members should be contributors. contributors (Note: All of your work should be pushed to GitHub as pull requests and reviewed by another team member.
2. Provide a description of what the project does.
3. Run static code analysis tool and generate an analysis report (You may use SonarQube as we did in the lab)
4. Provide a description for the feature/bug fix that you will start working on it.
5. Register the change request you defined in step 4 to a free ticketing system or software such as (Bugzilla, Jira, Agiloft, ...),
6. You are requested to update it through the lifetime of the change request to indicate it’s status, and any changes that are related.

* **Deliverables**

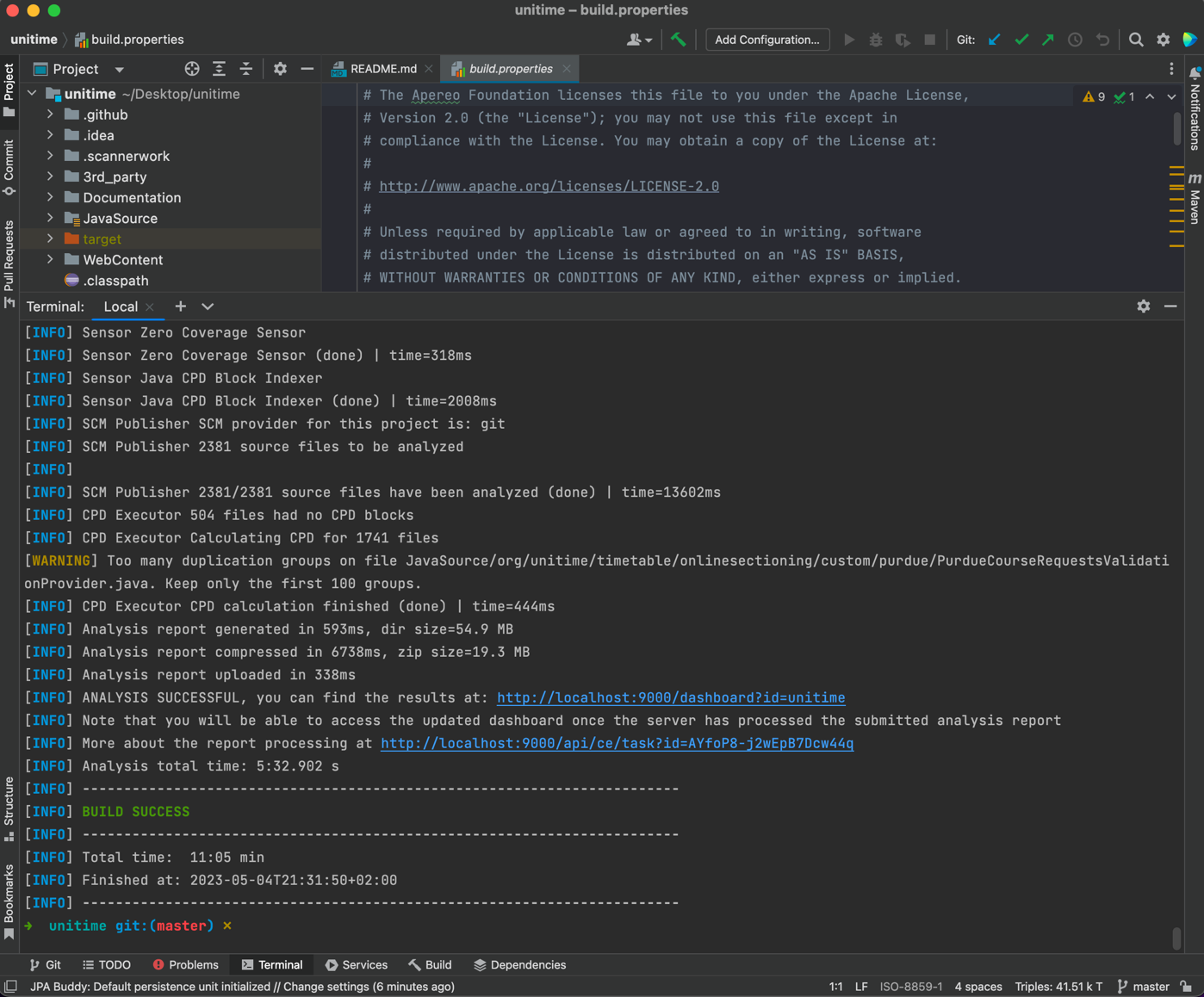
1. A link to your Github repo
2. A link to your ticketing system
3. Static Analysis Report For java backend code
4. text file (word or pdf) containing the description of what the project does. Mention what technique(s) you used to obtain this description
5. A description for the feature/bug fix that you will start working on it.

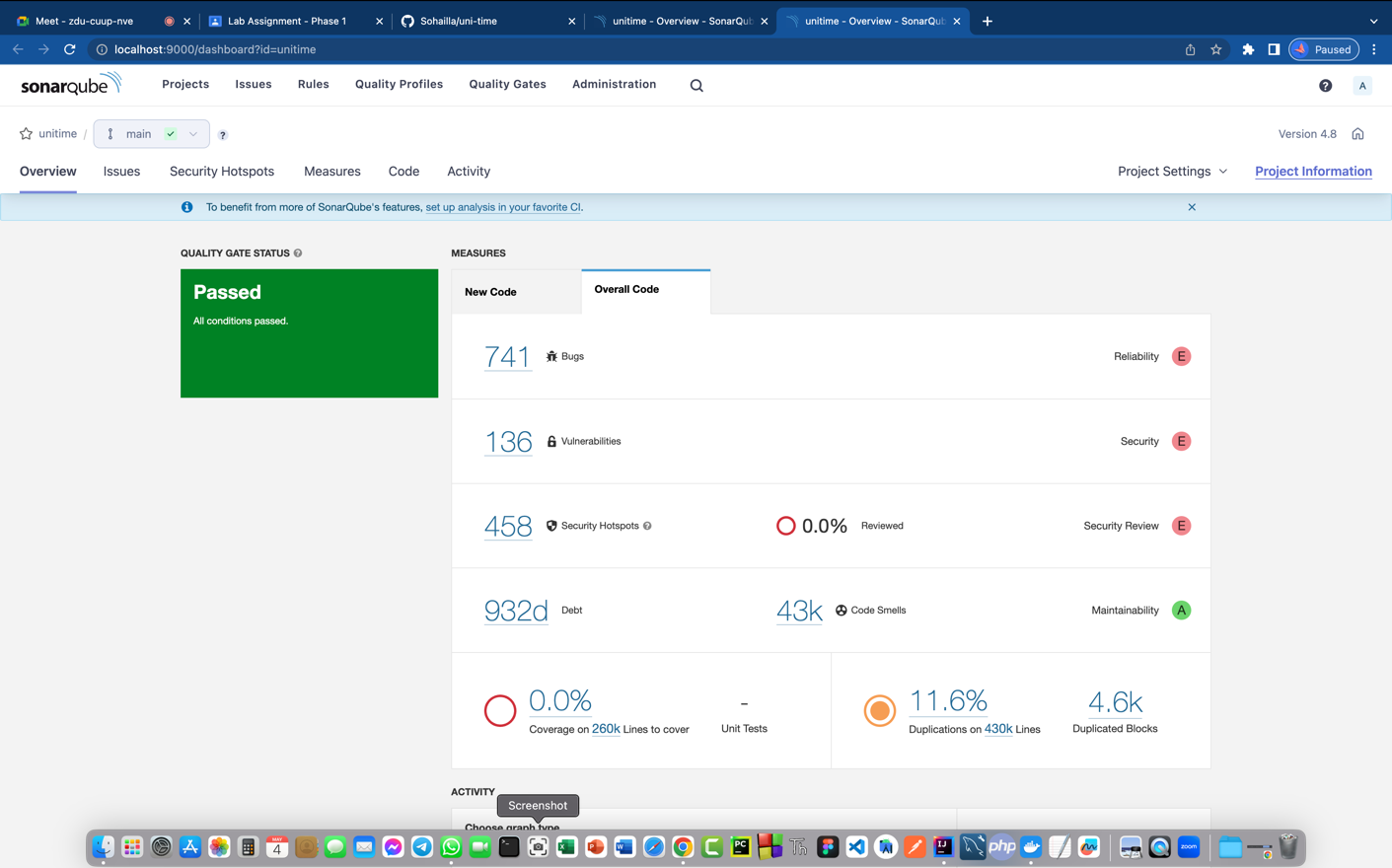
* **Notes:**
  + All of the required deliverables will be pushed to your Github repository, all team members must use Github to push their tasks.
  + You should submit a text or pdf file with your names and github repo (don’t push anything after deadline),
  + the file should have the following format file name: Group\_1stStudentId\_2ndStudentId\_3rdStudentId\_4thStudentId

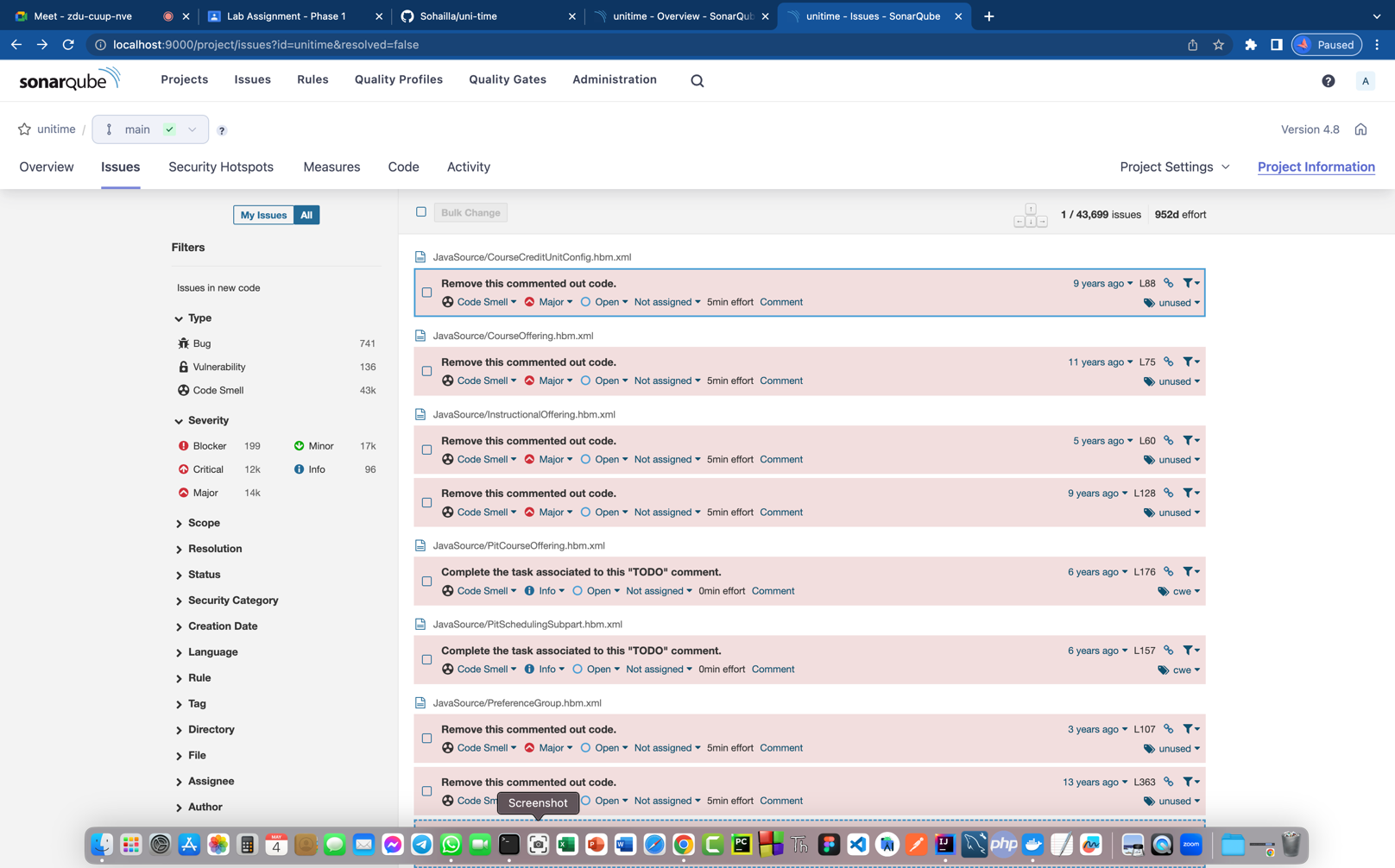
**Team Member:**

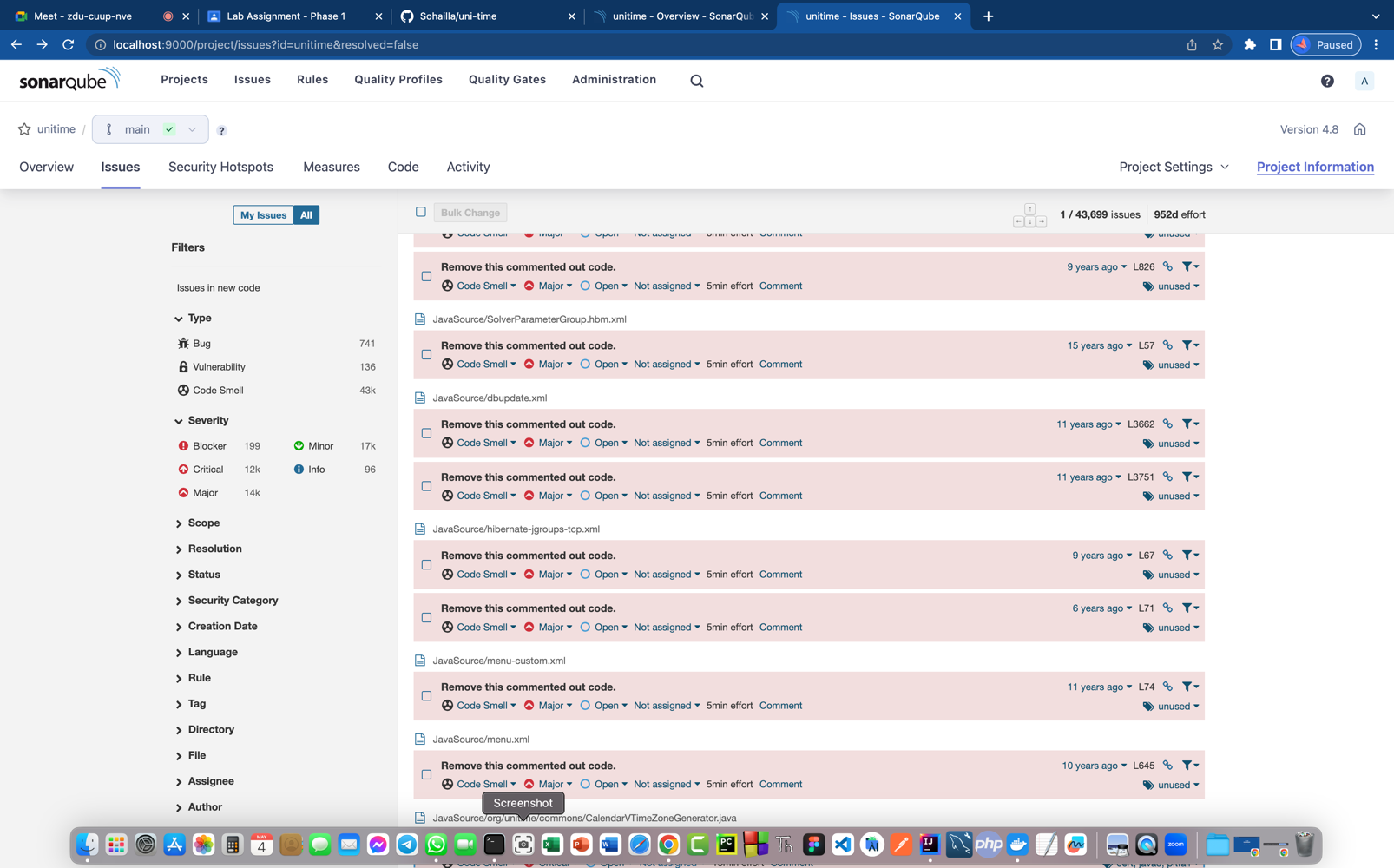
|  |  |  |
| --- | --- | --- |
| ID | Name | GitHub Username |
| 20196037 | Amr Halaby | AmrHalaby |
| 20196026 | Sohaila Gamal | Sohailla |
| 20206023 | Dalia Gamal | DaliaGamal11 |
| 20196003 | Ahmed Gamal | Ahmedgemy2002@gmail.com |

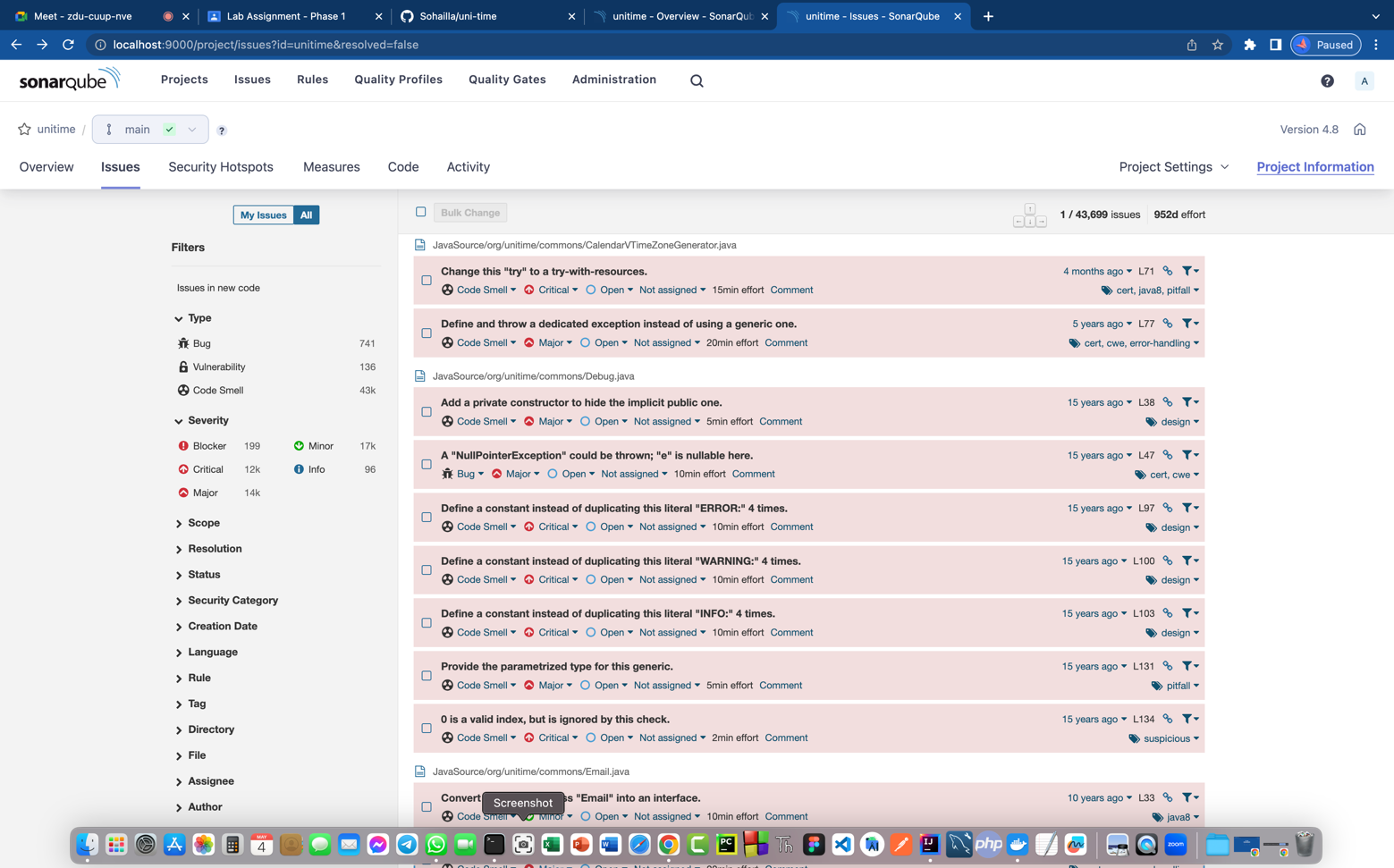
* **GitHub Repository:** https://github.com/Sohailla/uni-time
* **Ticketing System [Trello]:** <https://trello.com/b/nGqGdbd8/uni-time>
* **Static Analysis Report For java backend code**

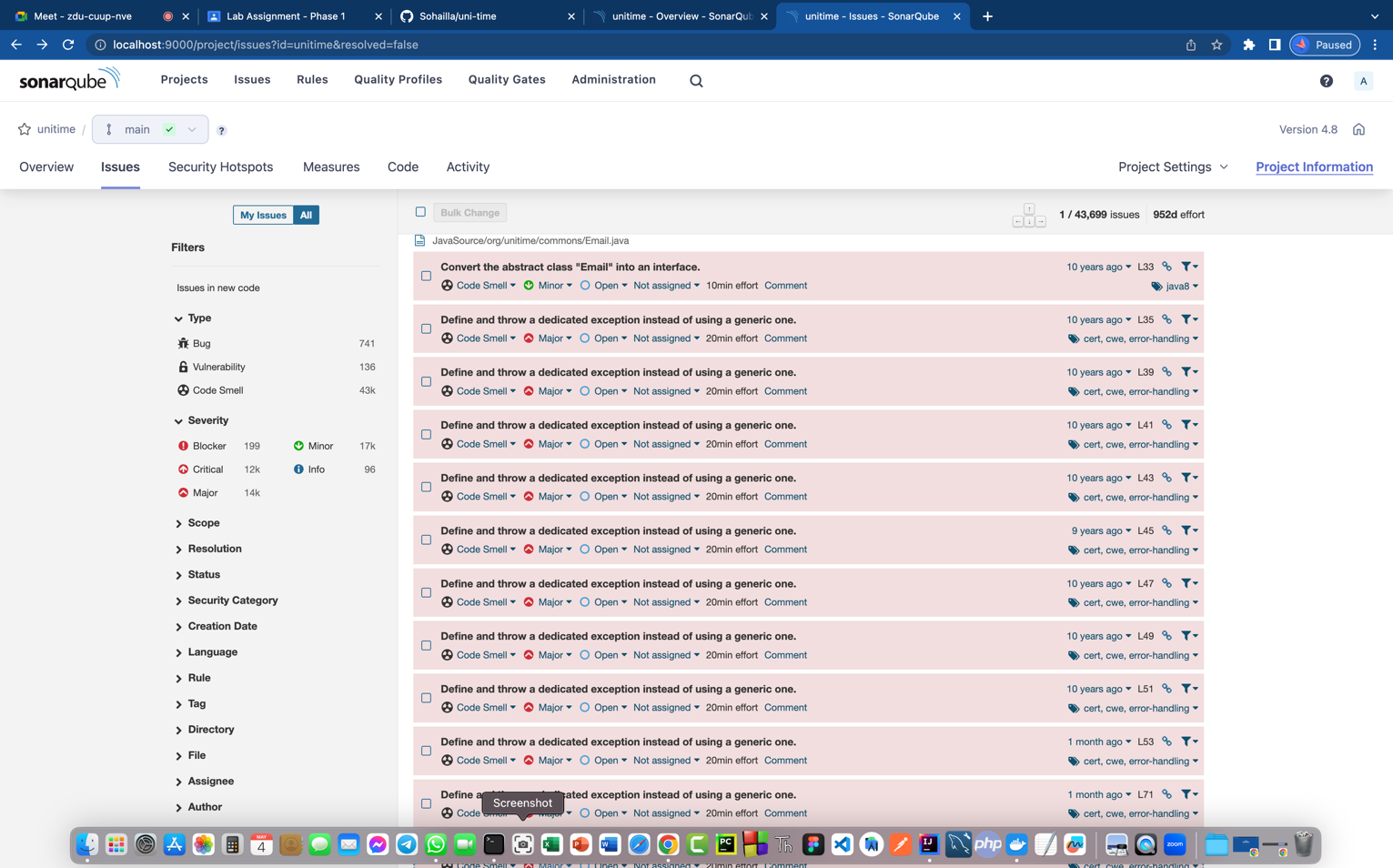
****

****

****

****

****

****

* **The description of what the project does. Mention what technique(s) you used to obtain this description**

1. **What the project does:**

UniTime is a comprehensive educational scheduling system that supports developing course and exam timetables, managing changes to these timetables, sharing rooms with other events, and scheduling students to individual classes. It is a distributed system that allows multiple university and departmental schedule managers to coordinate efforts to build and modify a schedule that meets their diverse organizational needs while allowing for minimization of student course conflicts. It can be used alone to create and maintain a school's schedule of classes and/or exams, or interfaced with an existing student information system.

**Features**

* course timetabling
* examination timetabling
* event management
* student scheduling

1. **Maintenance Technique: “**Opportunistic Approach”

Opportunistic Approach is a hybrid of the two

1. Begin with top-down, gain an overview of the functions of the program
2. Then selectively apply bottom-up strategies when nearing “code level”
3. Presence of beacons can indicate opportunity for change of strategy

* **A description for the feature/bug fix that you will start working on it.**

We will fix the Email.java and ToolBox.java classes.

In Email.java:

We will modify the code into two classes to support solid principles

In ToolBox.java:

Add new features to class to support availability