Lab Assignment - Phase 1

**Software Evolution and Maintenance Project**

Team:

Salma Walid Mohammad

Youssef Abdelrahman Youssef

Ahmed Tarek Mahmoud

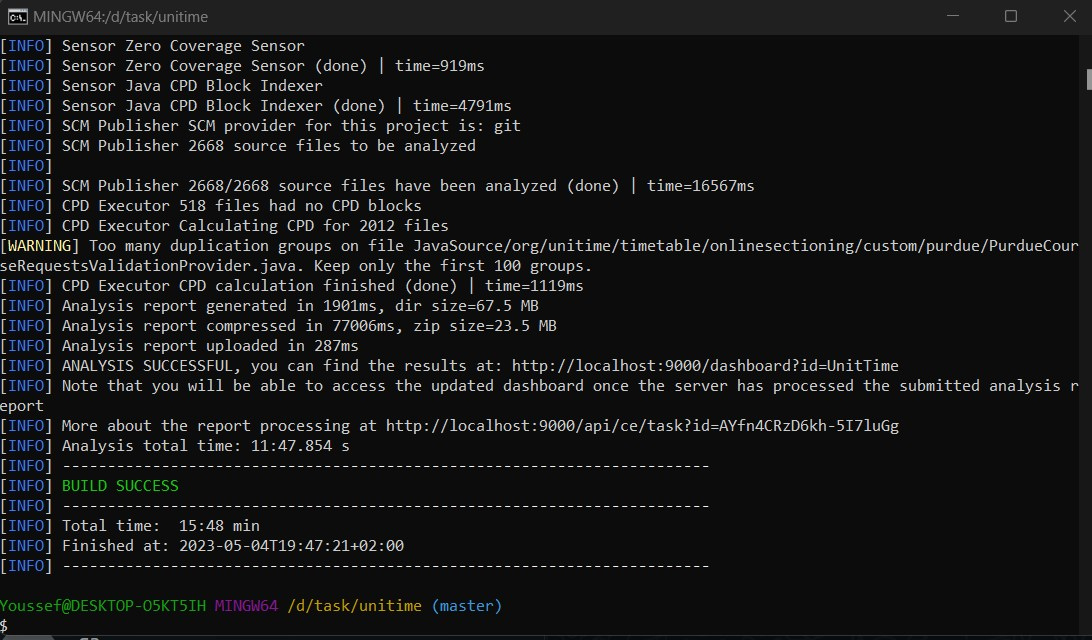
Reham Hassan Abdelazeem

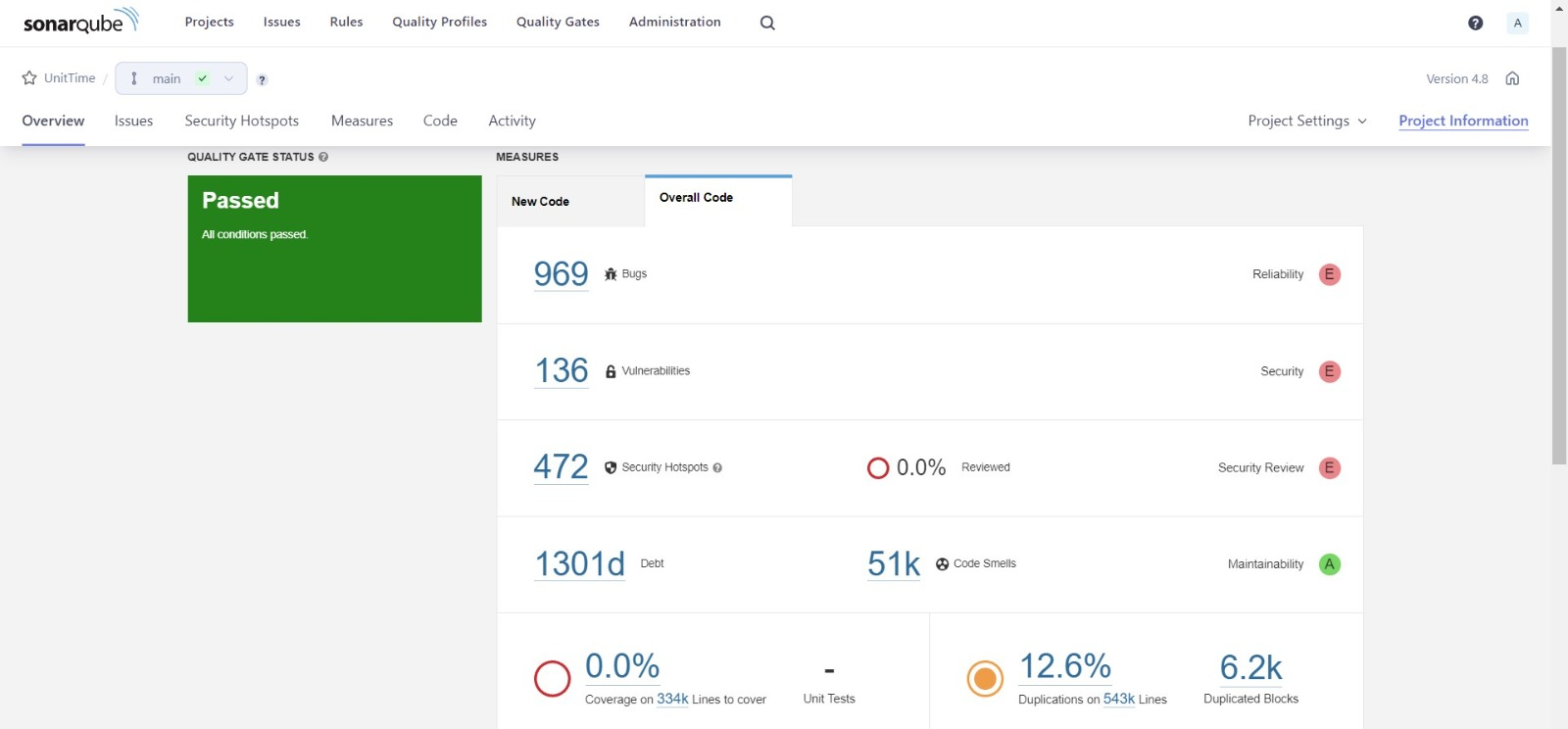
**1- GitHub repository link**:

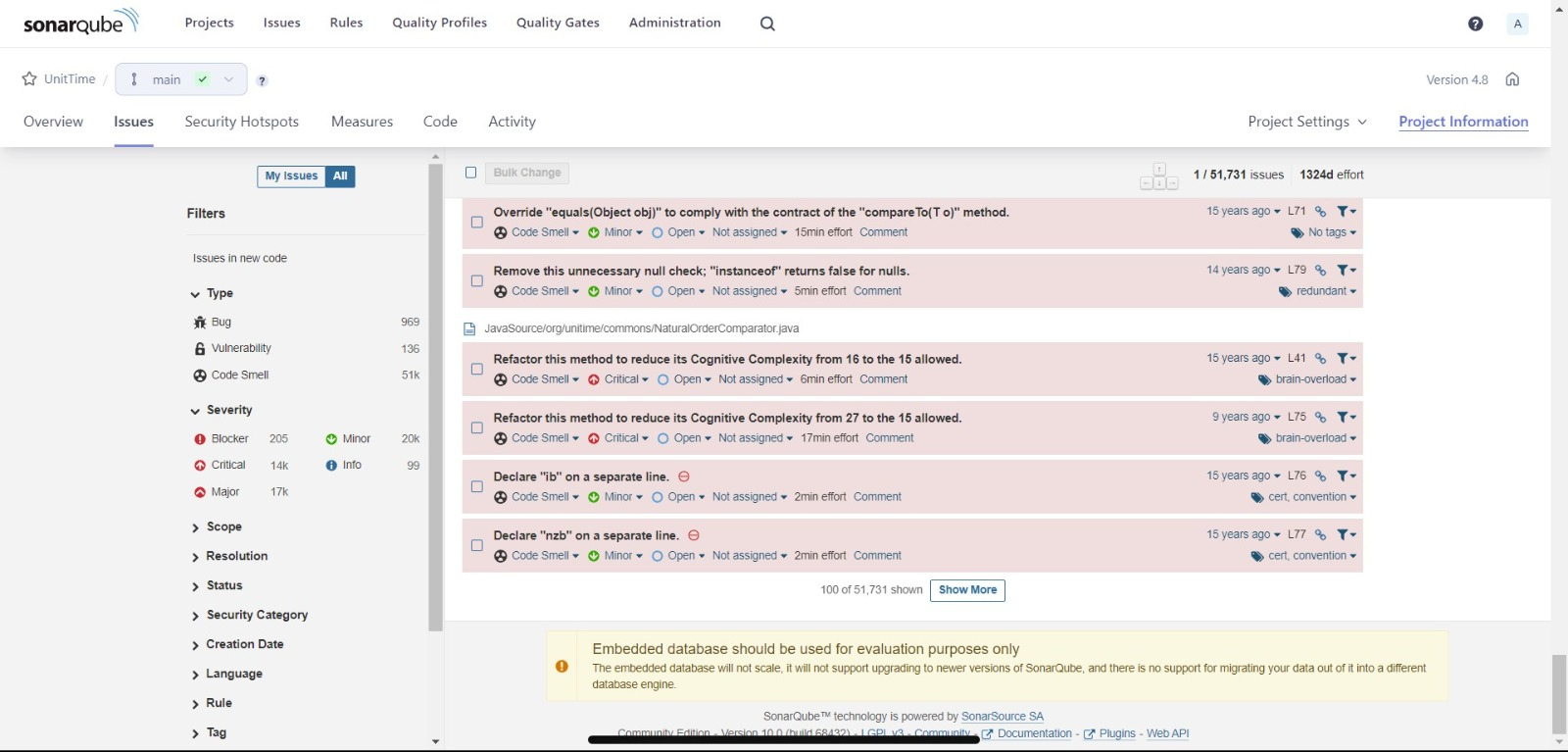
https://github.com/salmawalid/-Uni-Timetabling

**2- Ticketing system link:** <https://trello.com/invite/universitytimetable4/ATTI4ff811a49d08b1d357f7594d0bc1398bBB00B941>

**3- Static analysis report:**

****



****

**4- Description of what the project does. Mention what technique(s) you used to obtain this description:**

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.

[UniTime](http://www.unitime.org/) is an open-source enterprise system for automated construction of  
demand driven [course](http://www.unitime.org/uct_courses.php) and [examination](http://www.unitime.org/uct_exams.php) timetables. It also provides event management capabilities  
and [student scheduling](http://www.unitime.org/uct_students.php) capabilities. 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.

**Its main functionalities are**:

* **Course timetabling**

The primary objective behind course timetabling is to place each course at a time (or set of times) which does not conflict with the time(s) assigned to any other course required by the students attending it.

* **Examination timetabling**

UniTime builds a complete exam schedule each term that minimizes the number of conflicting exam placements for all students. It can also minimize the number of occurrences of back-to-back exams or students with more than a given number of exams in a day. This is particularly useful for colleges and universities that frequently update class offerings or have a large number of multi-section courses that do not fit well into mapped exam timetables.

* **Event management**

Most colleges and universities make use of their facilities for more than scheduled classes. Guest speakers, club meetings, study sessions, and other activities need to be scheduled to a variety of campus spaces. UniTime automatically creates events for all class meetings and examinations in the events calendar for the academic session. Additional events can be added to rooms used by classes, or any other campus facility in the system, using the web-based Events interface.

* **Student scheduling**

The student scheduling process is essentially that of matching the sets of classes required by each student to the available class spaces so that all (or as many as possible) of the students educational requirements are met.

**Techniques:** Top-down approach whereas knowledge is gathered, analyzed, then actionable conclusions is drawn. It is a rigid and structures approach which is useful in projects with many different project parts or any factor that makes processes difficult to keep organized.

**5- Description for the feature/bug fix that you will start working on:**

**Class: BaseUser**

Each user's Username must have at least 5 characters.

Each user's Username can't be null.

Each user's Password must have at least 8 characters.

Each user's Password can't be null or empty.

Each user's externalUniqueId can't be null or empty.

**Class: BaseSubjectArea**

Adding validations to check input data before saving to the database.

More Data Validation: Additional checks can be added to ensure that the data entered into each field is valid and of the correct format:

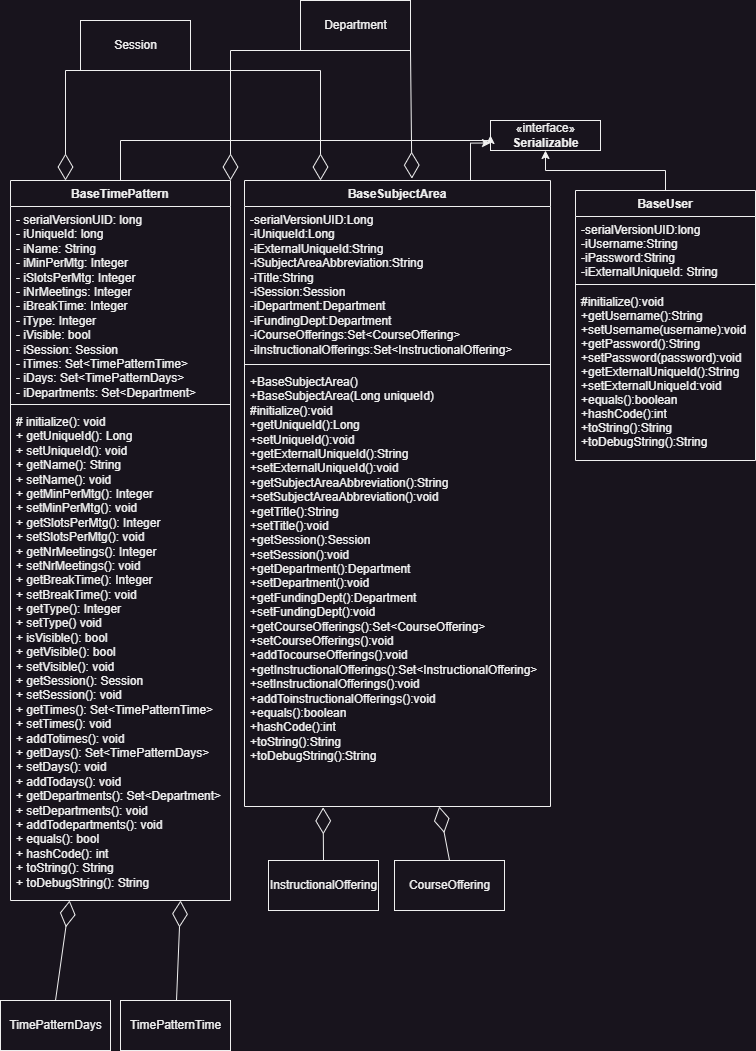
1- Validate Subject Area Abbreviation: You can validate that the Subject Area Abbreviation field is not empty, does not contain any special characters, and has a specific length using regular expressions.

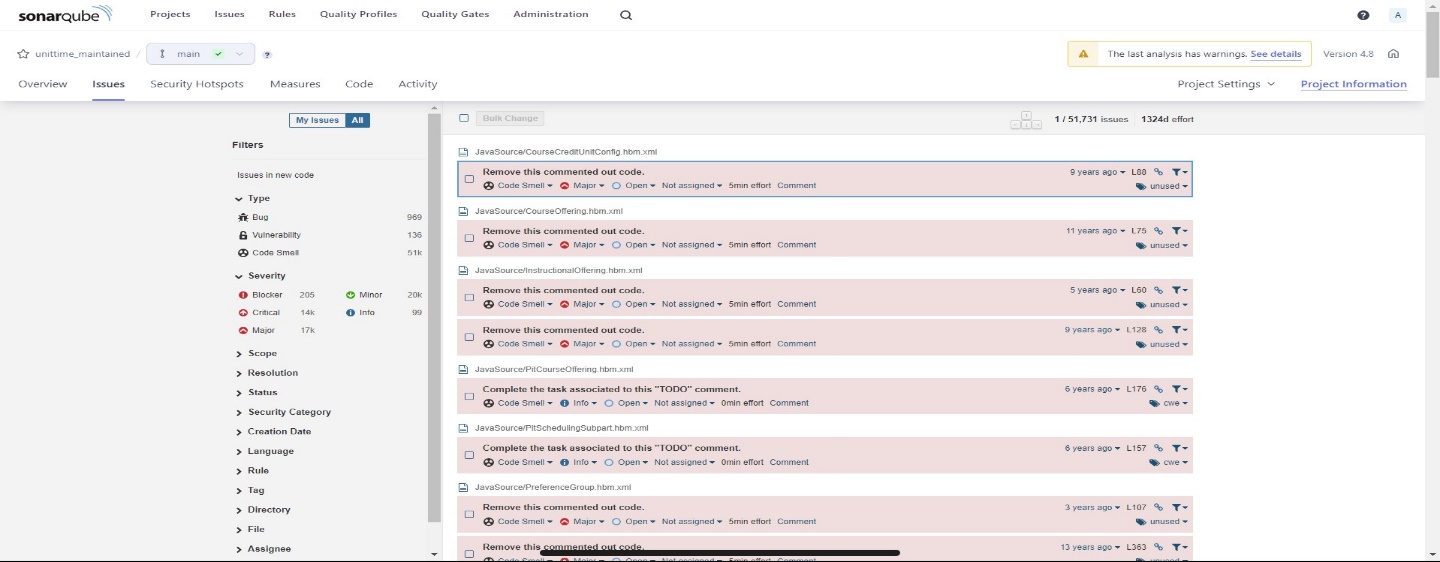
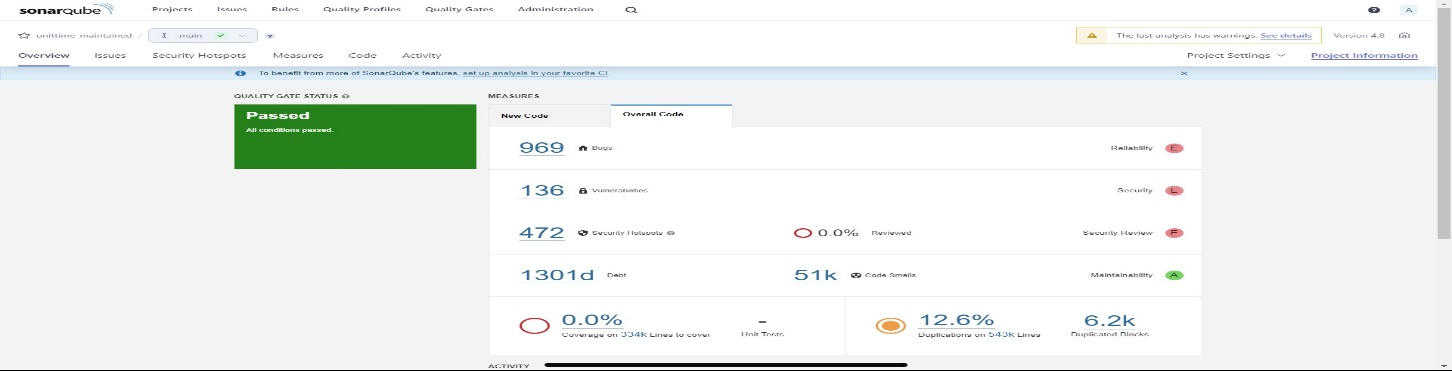
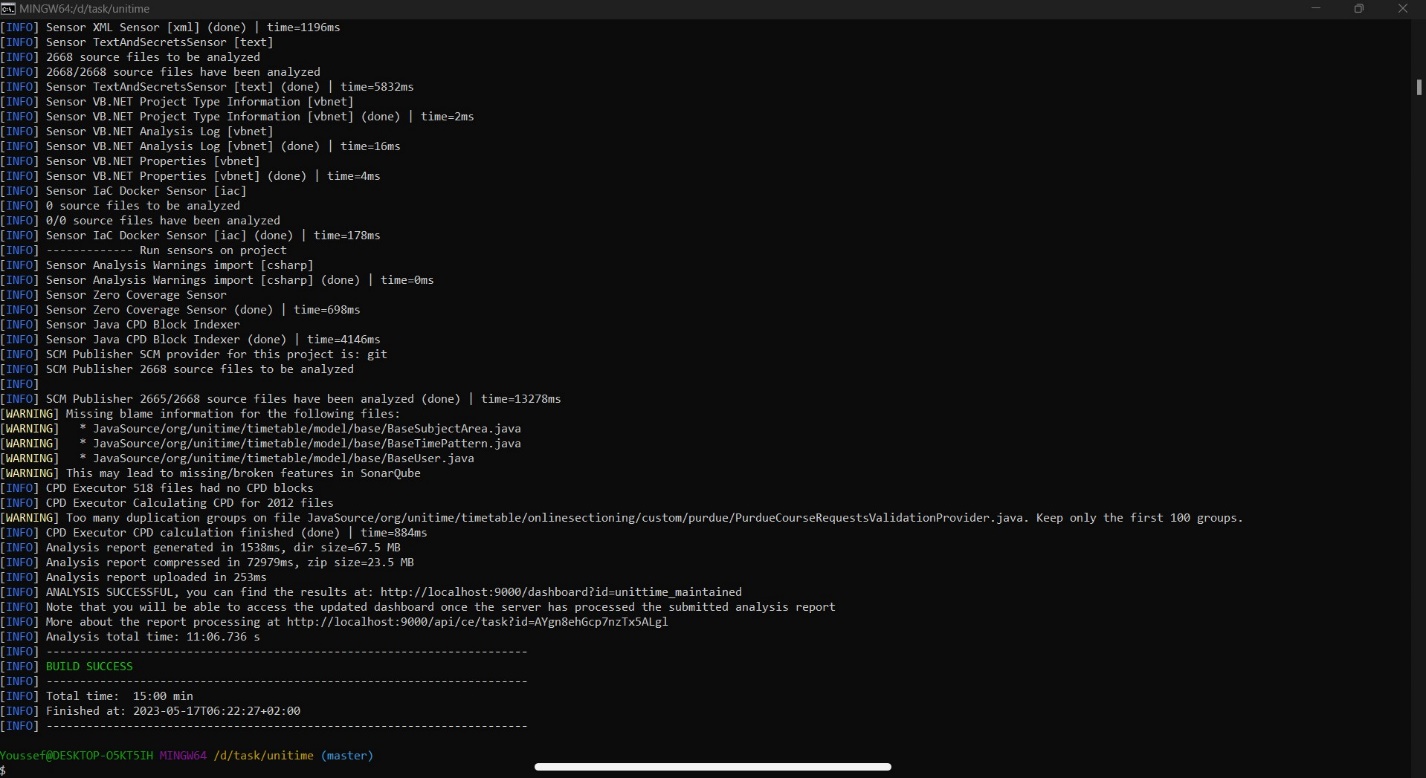
2- Validate Title: You can validate that the Title field is not empty and has a maximum length using regular expressions.

**Class: BaseTimePattern**

Validation for the input values. For example, checks could be added to ensure that the minimum number of minutes per meeting and slots per meeting are positive integers, and that the number of meetings and break time are non-negative integers. Similarly, the type attribute could be constrained to an enumerated set of values, and the visible attribute could be required to be either true or false. These checks can help to prevent invalid inputs from being persisted to the database and causing errors later on in the application's operation.

**6- Class Diagram:**



**7- SonarQube after adding features:**