**Documentation for TAAP**

**GitHub Project Link:** [**https://github.com/nidjosep/TAAP**](https://github.com/nidjosep/TAAP)

**Overview**

**T**eaching **A**ssistant **A**ssistance **P**latform – TAAP, is a web application that assists teaching assistants and professors in evaluating students' performance in lab sessions that are conducted in-person at the labs.

**Objectives**

We made use of the following objectives to implement the system:

* The system should allow the creation of new lab sessions and generate tokens for teaching assistants and students to join the active lab session.
* The system should allow students to raise a review request to the teaching assistant when they finish their lab exercise.
* The teaching assistant should be able to see the number of students in the queue, and the system should assign review requests to teaching assistants on a first-come, first-serve basis.
* The system supports multiple lab sessions and multiple teaching assistants to be registered with a lab session so that the review requests will be split based on availability.
* Once the teaching assistant taps on the "ready to accept request" button, a request will be assigned to the teaching assistant from the queue.
* The system should allow the teaching assistant to grade the students and submit the evaluation.
* While in the queue, the students should be able to see their queue status on the page.
* The lab session should have a set time duration, and once the session expires, the evaluation results should become available in the dashboard for download for the next 24 hours.

**Technologies Used**

We have made use of the following technologies to implement the system:

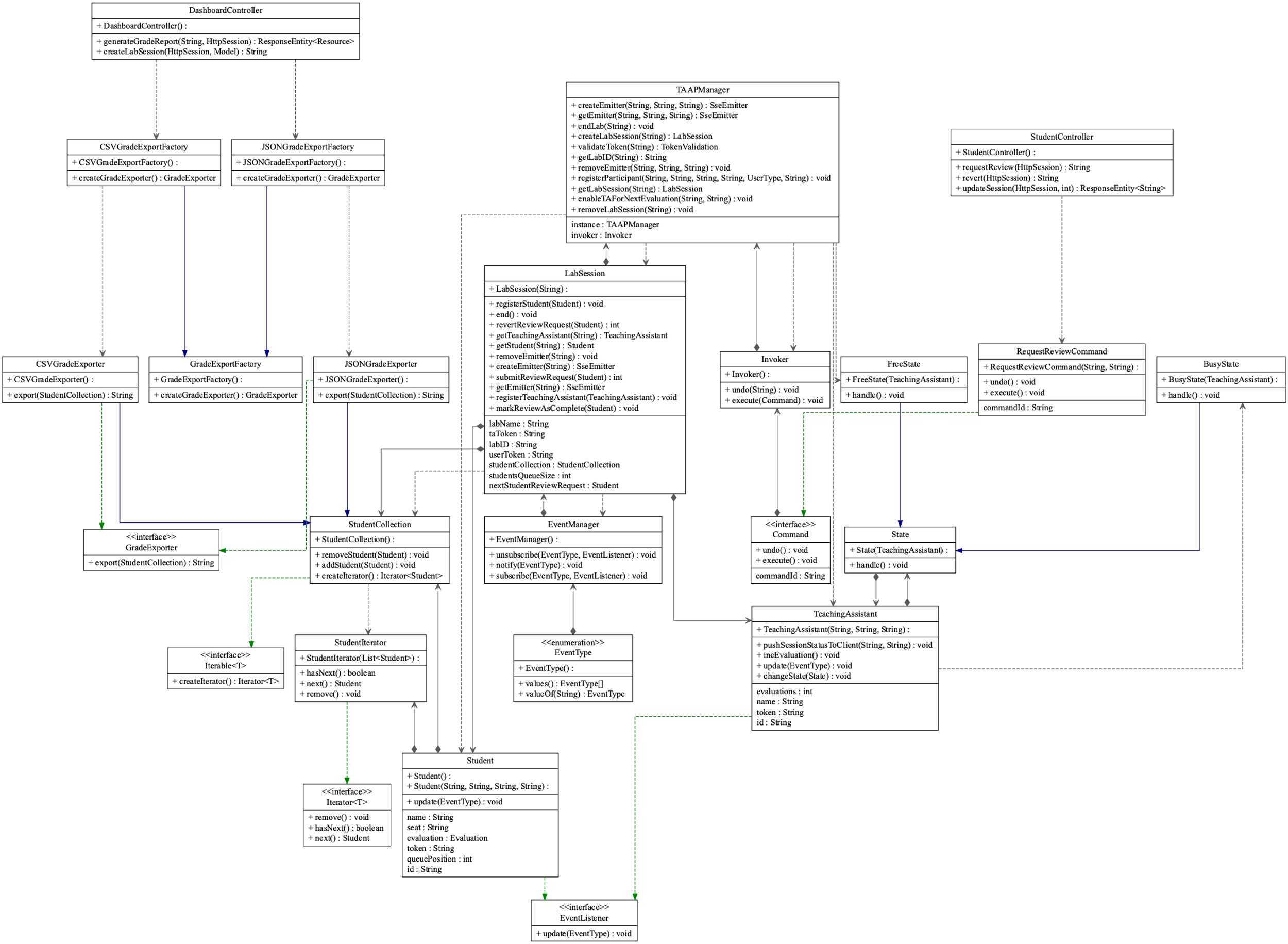
* Java Spring Boot
* Server-Sent Events (SSE)
* Thymeleaf
* HTML
* CSS
* Bootstrap
* JavaScript
* jQuery

**Design Patterns**

To satisfy the requirements for the Software Architecture and Design Patterns course, we have implemented the system using the following design patterns:

* Command Pattern
* Factory Pattern
* Iterator Pattern
* Observer Pattern
* Singleton Pattern
* State Pattern

The application follows the architectural style - Model View Controller (MVC).

**UML Diagram**

**Application in Use**

*Homepage*

**Table

Description automatically generated with low confidence**

The Homepage is the starting point of the application. Here, a user can either join an active lab session that was already created or create a new lab session.

As an example, we can create a new lab session named “SoftwareArchitecture”. (A professor can create a lab session)

Graphical user interface

Description automatically generated with medium confidence

Now that the lab session is created, we have given the option to generate the token which can be shared to the respective Teaching Assistants (TA) in the lab.

Graphical user interface

Description automatically generated

The TA can now join the session with the TA token and by filling out the required details as shown above.

**Graphical user interface, Teams

Description automatically generated**

The TA is taken to TA Dashboard where they can do the following:

* Fetch a token to share to the students joining the same lab session.
* Click the “TAP” button to start the evaluation.
* View the number of evaluations already completed and pending evaluations.
* View the Report of evaluated students.
* Terminate the lab session.

Examples of Students Joining the session with the TA generated code “8341”:

Graphical user interface

Description automatically generated

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generated**

**A picture containing diagram

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

The students Ashwant M and Nidhin J have joined the queue in the respective order by tapping the “Request Review” button. Students can also revert the review request by clicking on the “Revert Review Request” button.

**Graphical user interface, Teams

Description automatically generated**

Now, the TA can view that there are 2 students in the queue who are ready to be evaluated.

Graphical user interface, application

Description automatically generated

When the TA is ready to evaluate the students, he/she clicks on the TAP button on the respective screen and the details of the student who is first in the queue will be displayed on his/her screen.

Graphical user interface, application

Description automatically generated

**Graphical user interface, application, Teams

Description automatically generated**

After the evaluation is complete the TA can be ready and wait for new review requests from students.

Graphical user interface

Description automatically generated with medium confidence

The students can view their result by clicking on the “Report” button. Note that here the students cannot view the other students’ grades.

Table

Description automatically generated

Similarly, the TA can view the Report for all the students he/she has evaluated in the lab session.

There is the option for the TA to export the Grade Report in JSON and CSV formats.

Table

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

The TA can also search for individual student using the Search bar.

Graphical user interface

Description automatically generated

After the lab session and evaluations are over the TA can terminate the lab session.