Teacher View

# Functionality:

* Allows instructors to view students’ progress and grades.
* Displays a comprehensive table with all exercises and students listed.
* Includes each student’s score and status on a particular exercise: green for proficient, red for not yet proficient, and gray for not yet started.
* Allows instructors to export table to a CSV file for easy storage.
* Allows instructor to view breakdown of students for each exercise.

# Technical Details:

Using the Django web framework for the development of views for the OpenDSA project, each view uses the Model-View-Controller architecture. In the repository: /Aalto--/aaltoplus/, the OpenDSA data is housed in /opendsa/.

## Models

The models used for the teacher view are defined in models.py. The model classes used by the teacher view are:

* Exercise
* BookModuleExercise
* UserData
* UserExercise

## Views

The views for the teacher view are defined in /Aalto--/aaltoplus/templates/teacher\_view/. The views are html files that the controller uses to render data from the models. The views included for the teacher view are:

* exercise\_summary.html
* progress\_summary.html

These pages display the dynamic data from the models in different ways. The exercise\_summary page is of more detail, using Javascript and a JQuery plugin stored in /Aalto--/aaltoplus/assets/ to display an advanced table showing each student’s score and status in each exercise. The progresss\_summary page simply displays a table with columns for the three statuses for each exercise, along with which students fall in which category.

## Controllers

The controller for the teacher view is defined back in /opendsa/ in views.py. The relevant classes for the teacher view are:

* userValue
* exerciseProgress

While the view definitions for the teacher view are:

* exercise\_summary
* export\_csv
* progress\_summary

### exercise\_summary:

This view definition uses the BookModuleExercise model that corresponds to the opendsa\_bookmoduleexercise table in the database. The controller uses this table to build a list of exercises in the correct order as they appear in the book.

This view definition also uses the UserData model that corresponds to the opendsa\_userdata table in the database. The controller uses this table to build a unique list of users along with their points.

The UserExercise model corresponds to the opendsa\_userexercise table in the database. This table has an entry for each exercise a user begins, and if they were proficient or not in it. The controller uses this model to render the statuses for each student in each exercise.

The controller then builds a list of userValue objects that essentially hold a unique user, the user’s score, and the statuses of the user for each exercise in the correct order based on the exercise list.

Lastly, the controller renders the exercise\_summary.html view using the list of exercises for headers of the table, and the list of userValue objects for each row in the table.

To improve performance in this view, fetching of each model uses select\_related() to return the QuerySet. This allows the QuerySet to follow as far as it can, any foreign-key relationships, so that it will not have to continuously hit the database but instead will already have those other models built.

### export\_csv:

This view definition renders the same information as the exercise\_summary view definition, except that instead of rendering the view page, it renders it in an attachment of type CSV.

### progress\_summary:

This view definition uses the BookModuleExercise model to build a list of exercises in the correct order as they appear in the book.

It also uses the UserExercise model to filter out which entries in the table correspond to the specific exercise.

The controller then builds a list of exerciseProgress objects that hold a unique exercise and three lists of proficient, not yet proficient, and not yet started users of the exercise.

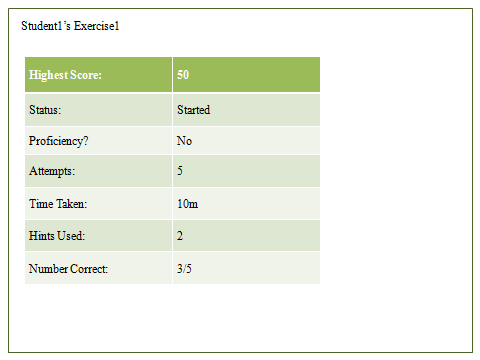
Lastly, the controller renders progress\_summary.html using the list of exerciseProgress items to write each row in the table.

## Future Work

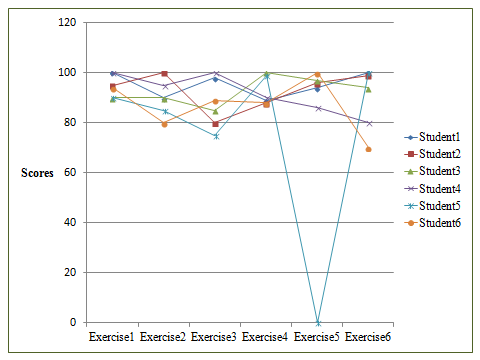
Performance improvements are necessary for each view. Currently, the UserData, BookModuleExercise, and UserExercise tables are fetched and then used to build lists containing information in the right format for display.

Display can also be improved for better readability and usability. The exercise\_summary.html view could add rotated table headers to save space. This has proved difficult with the JQuery plugin used to fix the table headers in place for scrolling, Fixed-Header-Table. Information for the plugin can be found at: <http://fixedheadertable.com/>.

Additional views for the teacher might also prove necessary in the future to expand on current views. Mockups for some possible views are included below:



An expansion view for exercise\_summary. Every table cell expands to this view for additional student information in regards to the particular exercise.



A different view that displays the progress of scores for each student. It might be more useful for displaying this view based on the types of exercises, since each are scored differently.