

ADVANCED SUBMISSION SYSTEM

REGRESSION AND INCREMENTAL TESTING LOG SPRINT 2

Team 15

Krutarth Rao

Anthony Goeckner

Harold Smith

Saurav Khanna

Max Molnar

CLASSIFICATION OF COMPONENTS

Module description

Grade Configuration Manager

Module in charge of creating/updating/removing entries in configuration files that the system uses to keep track of student grades and instructor feedback. Automated tests for the module can be found in *TestGradeConfigManager.py* in the test directory. The module was tested using the *unittest* framework in Python.

Inputs: Items to be added/updated in the grade configuration file.

Output: A confirmation on implemented changes or a warning on invalid changes.

Course Manager

Module in charge of creating/updating/removing course directories and assignment directories. Also handles entering/updating grades

Inputs: Courses/Assignments to be created/deleted/updated, Grades to be entered/updated

Output: A confirmation of whether or not the changes were successful

Submission Interface

The Submission User Interface allows students to easily submit files to assignments in their course.

Inputs: Course name, assignment name, and files, all selected using UI.

Output: A confirmation of whether or not the submission was successful.

Interface Component - Button

The Button component is used to create selectable buttons in the user interface.

Inputs: Button label and callback function.

Output: Callback function is called when button is pressed.

Interface Component - Picker

The Picker component allows for display of single or multi-selectable lists or trees. This is used frequently throughout the UI.

Inputs: Options list, maximum number of selected options, title.

Output: A list/tree UI widget that allows user interaction.

Interface Component - Input Manager

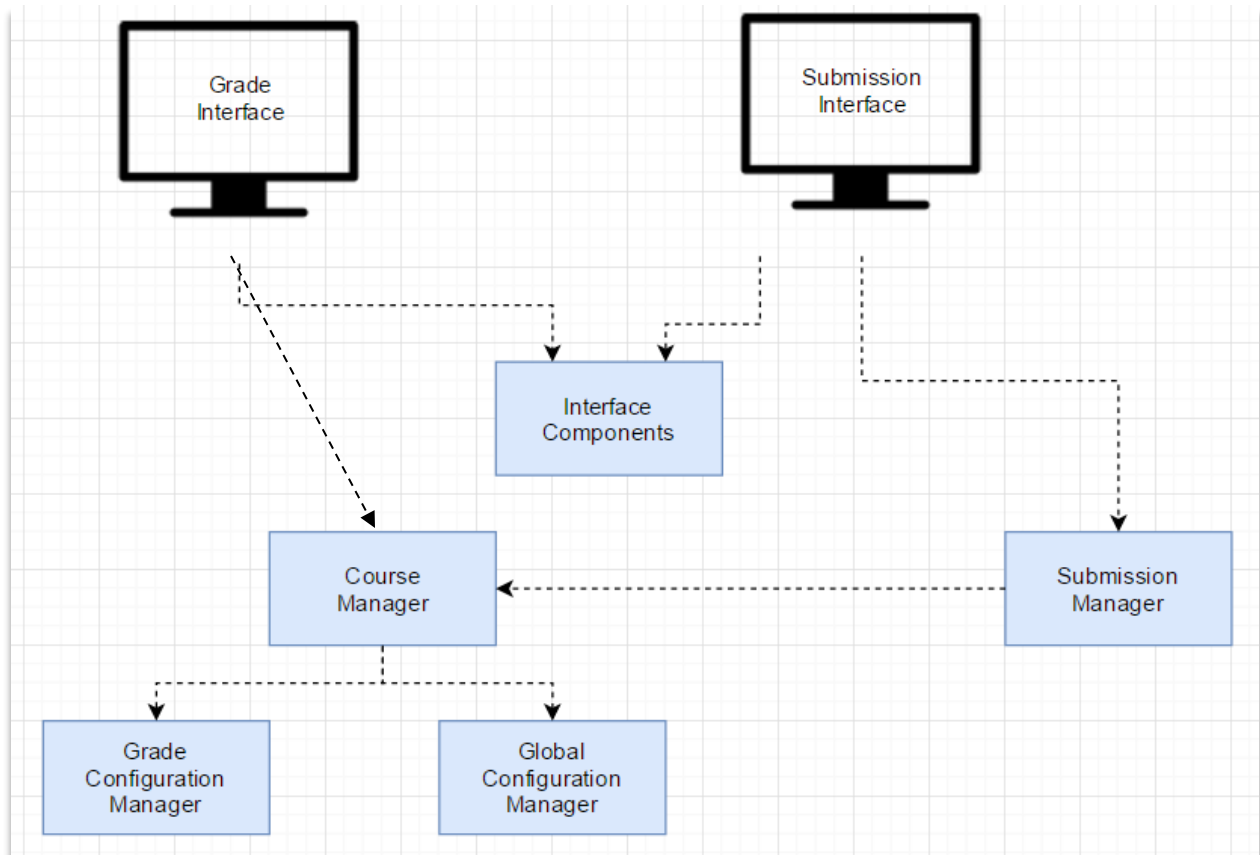
The Input Manager directs user input to specified components in the user interface, and allows the user to “tab” through components to select them.

Inputs: All user input passes through this component.

Output: If the user input is a “tab”, the next component is marked as selected. Otherwise, all user input is passed to the selected component.

Incremental testing technique

Testing approach is bottom up where the modules below were implemented and tested, integrated with modules above them and tested again until the interfaces reflected the changes/bug-fixes. Older modules from sprint 1 were subjected to more testing along with the testing applied to the new modules. Instructor and Student interfaces from sprint 1 were merged to form the Grade interface for simplicity and provide the users with easy to understand choices.



Due to the bottom up approach, the team used drivers for the first phase of testing to verify the functionality of the terminal modules. Later, upper level modules were added as testing progressed with passing unit tests for each module on the way up.

Newer modules from sprint 2 were unit tested to verify isolated functionality and merged with the existing integrated system.

INCREMENTAL AND REGRESSION TESTING

Automation

Grade Configuration Manager

Module testing was done using the unittest framework provided by the Python library. The tests included the regression tests used to verify that previously fixed bugs stay out of the system and confirm that bug fixes did not create new bugs that affect functionality.

The automated tested functionality also includes checks to verify the reused methods from the global config manager were kept out of the system in not replicated in the new module.

Submission Interface

Submission Interface testing uses the built-in Python unittest library. Tests are designed to cover edge cases and a variety of inputs.

Interface Components

Testing of interface components uses the built-in Python unittest library. Tests are designed to cover edge cases and a variety of inputs, including different configurations and combinations of components.

Course Manager

Module testing was done via automated test cases with valid inputs. The tests included incremental tests and regression tests to account for existing bugs and to verify that new bugs were not created by bug fixes.

Defect log

Module	<u>Grade Configuration file manager</u>		
Defect No.	Description	Severity	Solution
1	Boolean return values in the get Course Grades method caused the GUI to crash.	2	Course grades method was changed to return an empty list to warn the caller instead of a Boolean return.

Module	<u>Configuration file manager</u>		
Defect No.	Description	Severity	Solution
1	The module did not contain the functionality to keep track of the instructor user group that is permitted to modify grade files.	1	Add Instructor method created to include an instructor section to the global config file.
2	Add Instructor method added a section to the global config file that was read as a course name in get course list	2	Get course list method removes the "Instructor" section before returning the list.
3	No functionality exists to fetch the user group of instructors	1	Get_setting reused to fetch the user_group setting from global config.

Module	<u>Course Manager</u>		
Defect No.	Description	Severity	Solution
1	An error is thrown when attempting to create an assignment that already exists	1	Put a check to see if the assignment exists
2	An error is thrown when attempting to delete an assignment that does not exist	1	Put a check to see if the assignment exists
3	An error is thrown when attempting to edit a grade that does not exist	1	Put in a check to verify that the grade exists

Module	<u>Picker UI Component</u>		
Defect No.	Description	Severity	Solution
1	Issue identified in sprint 1, where selection was incorrect after updating the list of options.	1	After changing options list, selection is now cleared. Wrote regression test.

Module	Student Interface		
Defect No.	Description	Severity	Solution
1	If default assignment is selected with out any input, "final-practice" is selected as opposed to "Lab1" which is highlighted.	3	Fixed issue in Picker class which caused this error and wrote regression test.

Module	Submission Manager		
Defect No.	Description	Severity	Solution
1	Course Config parsing was only correct locally and not in the server environment	1	Course Config parsing was made more generic
2	Calling "getProjectInfo" originally assumed the config files were created in the same hard coded order	1	Changed it to search dynamically for specific settings