**COMP 2005 – Final Project**

Weicai Zhao

Vladimir Andrade

Uvwie Omafume

Clifford Bowe

**Overview**

The project implements an alternative to the Brightspace provided online quiz application. It supports authentication of various levels of access (student and instructor) and gives different options to the user, based on their current level of access. Instructors will have the option to create a quiz, which will take them through the process of designing their own, multiple choice quiz to be stored and accessed later. The instructor can also view results (specifically, the results that are for the instructor only to view, such as class participation or the grade histogram for a given quiz). The student will have the option to view quizzes, which will take them to a list of quizzes that are within the system and then allow them to take a quiz, if they are permitted to. They will also have the option to view results, specific to them.

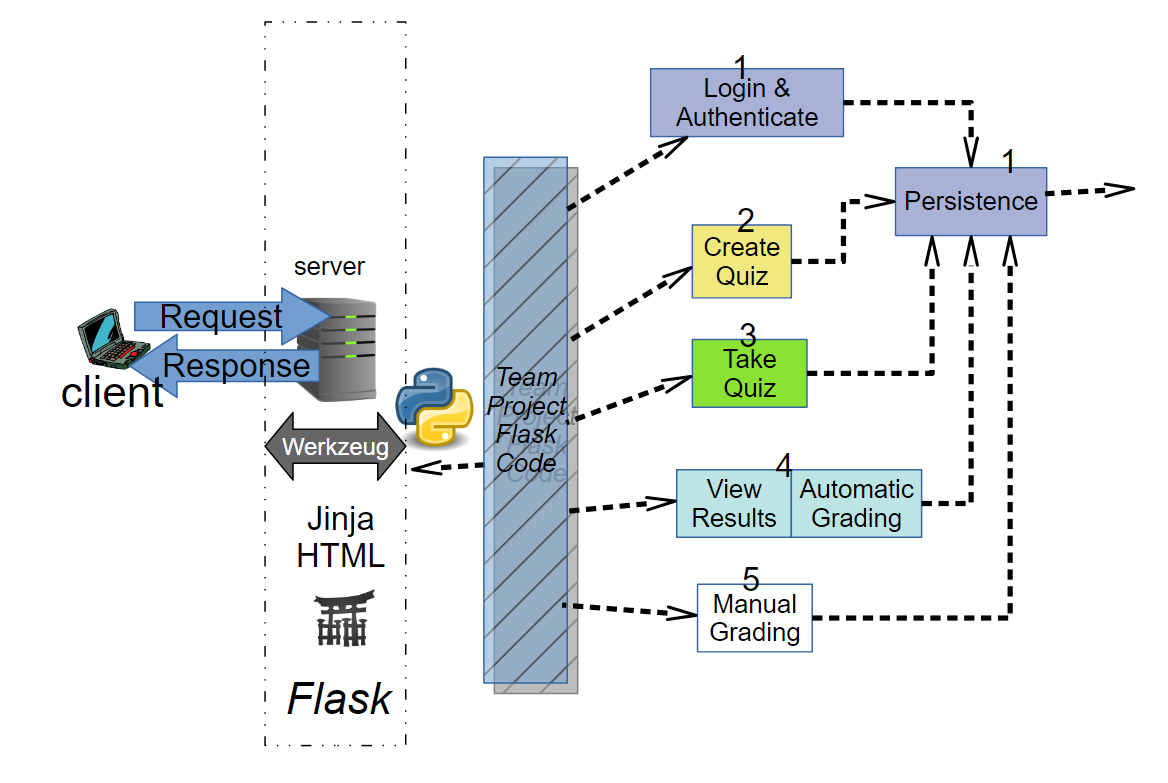
**Main Functionality**

The functionality of each module is listed below; a more in depth description of the methods and functions used can be found within the docstrings of the given modules. Each module fulfills the minimum, mandatory requirements laid out in the requirements listed in the modified “COMP2005 Project Requirements Sheet.html” file included in the project folder.

1. Login
   * Users can create accounts at the log in page (instructor accounts need special permission to be created)
   * Users can log in with existing accounts and passwords
2. Persist
   * Provides methods to store data from the various modules into persistent storage
   * Provides methods to access the data stored by the various modules in persistent storage
3. Create Quiz
   * Allows only a user that’s marked as an “Instructor” to access the “Create Quiz” section of the app.
   * Provides the instructor with the ability to create a quiz, giving them control over all aspects of the quiz’s design (question text, choices, test due date, etc.)
   * Stores the created quiz in persistent storage using a method provided by the Persist module
4. Take Quiz
   * Allows only a user that’s marked as a “Student” to access the “Take Quiz” section of the app.
   * Retrieves the stored, created quiz data from persistent storage
   * Shows a student the list of created quizzes
   * Only allows a student to take quizzes that they are permitted to take
   * Only allows the student to take a quiz before its scheduled end date
   * Allows a student to pause the quiz and resume it at a later date
   * Stores the relevant quiz attempt data in persistent storage using a method provided by the Persist module
   * Allows students to take multiple attempts, up to the attempt limit of the quiz dictated by the Instructor
5. View Results/Automatic Grading
   * Allows the user to view results; results are displayed differently for “Instructors” and “Students”
   * Retrieves the Quiz Attempts information from persistent storage
   * Automatically grades and stores the graded quiz information internally
   * Stores the graded quiz information in persistent storage using a method provided by the Persist module

Each module, when implemented as intended, will interact with Flask and Jinja to display the appropriate information, based on the current web page attempting to be accessed and the user’s information.

**Project Architecture**

****