Architecture & Design Specification

1. Scope.

1.1 Identification.

The system and software contained within this document belong to the Automatic Grading System. No previous versions or forms of identification exist.

1.2 System overview.

The purpose of the software is to provide a method by which instructors can evaluate students through assignments and provide feedback remotely. The software will be accessed by all users over the WWW through a web page interface. The project sponsor is Maksym Morawski. The acquirer of the software is UMBC. The users are instructors, students, and administrative staff. The developers are Team 1 of Dr. Birrane's Software Engineering I class for the Spring 2016 semester. The software will be hosted on a web server provided by one of the developers during construction. The software will then be deployed on a server of the customer's choosing.

1.3 Document overview.

This document will outline the architecture and design specifications. Criteria for construction of the software will be provided. Dependencies and process ordering will be established to organize characteristics of the project.

2. Referenced documents.

http://datahole.ddns.net/cmsc447/main/documents.html

3. System-wide design decisions.

Administrator accounts will need to be created manually and they will be able to create instructor accounts and classes. They will also be able to associate an instructor with a class and be able to make additional administrator accounts. These administrator accounts will not be linked to University credentials through Google+.

A student or instructor will use their University credential information to log into the system. A University ID and password will be provided by the user. If the credentials are correct, they will be given access to their respective account on the server and shown a home screen.

Instructors can create classes and invite users to join the class as students. When creating a class, they will be prompted for a class name and description. After they have followed the prompt, they can finalize the class. Once finalized, an option is shown to invite users. To invite users, they will add UMBC accounts to a list, and then confirm the list. The users will be informed of the invite and automatically added to the class via the database.

A student will upload an assignment by choosing a file from their system. Once the student submits the assignment, the system processes the script and uploads a grade. The student can view this grade and re-submit again if more submission chances are available.

The assignment grading process will take a student submission as an input. It will also use an output file provided by the instructor and then compare the output between the student

submission and the instructor. Using either specific rubric criteria or number of matches, a pass or fail indication will be given to the student for their submission.

Instructors will be presented with a list of classes they are associated with. Once selected, a list of the assignments they have created for a class with be displayed, along with an option to create an additional assignment. When an assignment is selected, a page will be displayed listing all of the students and their submissions, grades for those submissions, and any other relevant metrics.

Instructors will create assignments under a specific class with the following information necessary: name of the assignment, description of the assignment, number of attempts, any necessary input files, one or more correct sample output files.

4. System architectural design.

4.1 System components.

Backend Components:

Web Server:

The web server would handle the PHP requests from the website and manage the user accounts. When the students submit the assignment, the server would run the program and return a grade. The web server would connect to the mySQL Database server in order to handle store and retrieve data requests. The web server would also manage the classes created on the website along with the assignments that would go with each class. It would basically handle the assignment creation requests and the account relations to these components.

mySQL Database Server:

This component will serve as a relational database for the software. All records and information will be held in this database. The software will use records stored within the database to refer to and provide pages for users. This database will be used for most of the system-wide design decisions. The database will be updated to reflect system-wide decisions including: class creation, student entrance into a class, assignment creation, assignment grading, and file submissions. Each action will require storage and retrieval of entries within the database such that the web server can manipulate information relevant to user actions.

Abstract Components:

Classes:

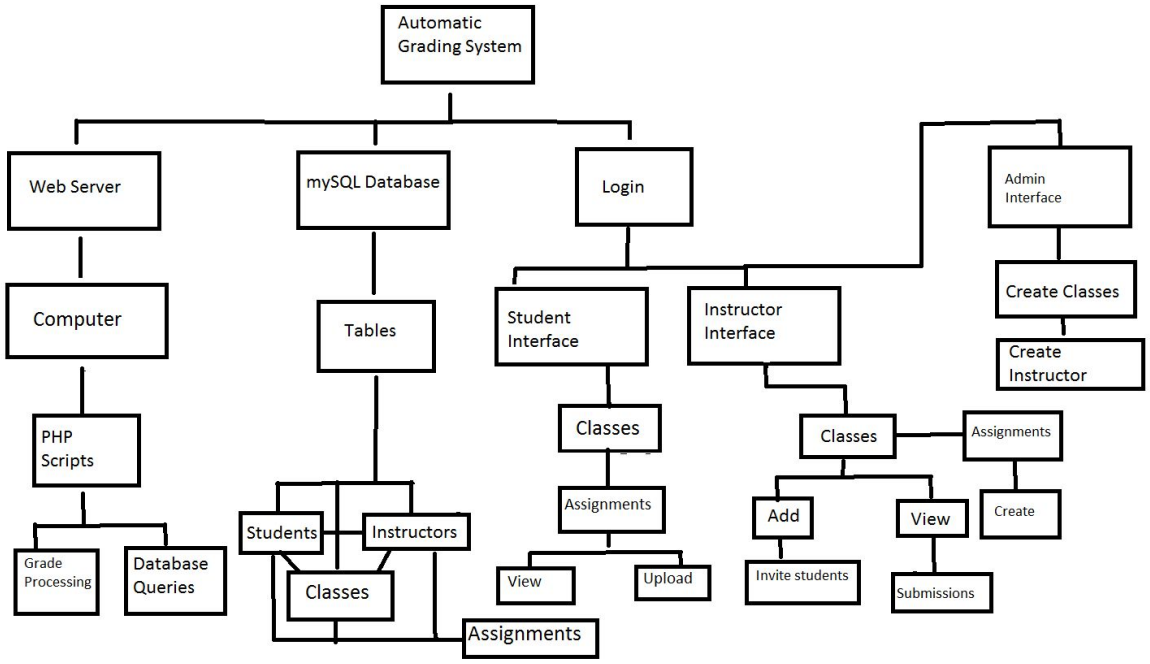
Classes will contain one or more Instructors, one or more Students, and any number of Assignments.

Assignments:

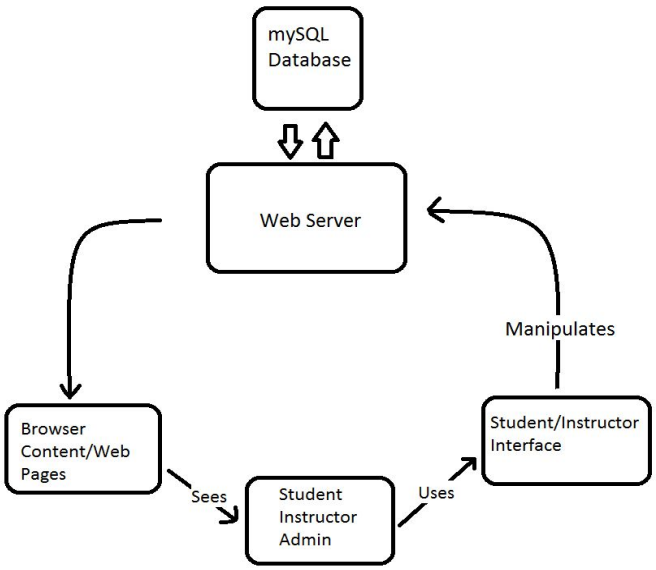
Assignments will contain a name, a description, number of attempts, any necessary input files, a deadline, and one or more correct sample output files. Assignments will be created by the instructors.

Login Component:

The login-page would be the same for all of the users (students, instructors, and administrators alike). Google API would be used for the authentication of the users. Upon success of the login, the users would be granted access to the components for which they have permission and access.



4.2 Concept of execution.



The user that is either a student, instructor, or administrator will use the web interface to access components that will be processed on the web server. The web server will store and retrieve information from the mySQL database to facilitate the functionality of its components. The web server will also update the user's view and provided information and/or additional controls.

4.3 Interface design.

4.3.1 Interface identification and diagrams.

Student Interface
Instructor Interface
Administration Interface

The interfaces will operate through HTTP/HTTPS over the WWW with typical client/server behavior.

See section 4.2 for interface interaction diagram.

4.3.2 Student Interface

The student interface will allow the user to select classes that they are assigned to and then select available assignments within that class. For a given assignment, the student will be able to view past submissions and the grade given by the system. They will also be able to upload additional submissions up to the limit allowed for the assignment.

4.3.3 Instructor Interface

The instructor interface will allow the user to select classes that they are assigned to and add students to that class. They can then create assignments as per the Assignment component, review their assignments, and provide feedback on student submissions to that assignment.

4.3.4 Administration Interface

The administration interface would include the admin features. These admin features consist of an option to create classes, and the ability to associate the UMBC accounts with either the students or the instructors. The admins would also be able to associate the instructors with the classes.

5. Requirements traceability.

See section 4.1 for traceability from each system component identified in this SSDD to the system requirements allocated to it.

See Software Requirements Specification for traceability from each system requirement to the system components to which it is allocated.

6. Notes.

Not Applicable

A. Appendixes.

Not Applicable