Assignment 3 : Requirements Analysis

Brandon Greer
BCGBR5 12423059
CS4320
3 Feb 2020

Step 1: Potential Users

Professors
Students
TAs

Step 2: User Activities

Professors:
- Create/Change/Delete Assignments
- Upload Documents
- Retrieve Documents either by assignment or by student, with some type of indication added to uploaded document to associate them with individual students
- Submit Grades
- Create Grade Calculations
- Leave comments on assignments
- Create automated tests/checks
- Delegate Responsibilities to TAs

TAs:
- All of the above responsibilities if delegated by the professor

Students:
- Submit assignments
- View assignments and requirements
- View documents uploaded by TA/professors
- View graded documents
- View automated test results

Step 3: Activity Details

- **Create/Change/Delete Assignments:**Instructors will need the ability to create an assignment, set a due date, upload some type of requirements page, upload potential supplemental documents, and change or delete any of these items
- **Retrieve Documents:** either by assignment,i.e. Downloading all assignment 1s from every student or by student, with some type of indication added to uploaded document to associate them with individual students
- **Submit Grades:** Submit grades on a per document basis, logically this should also have a per student or per assignment view
- **Create Grade Calculations:** Teachers should be able to input the final percentage contribution of each category of assignments and assign those categories to assignments

Leave comments on assignments: Teachers should be able to leave comments either on specific parts of an assignment (think Google Docs) or on a page or assignment overall.

Delegate Responsibilities to TAs: Most of these tasks could also be performed by a TA, however since not all professors utilize TAs in the same way and, even within a single class you might have a head TA or something similar, it would make sense to give professors some ability to delegate responsibilities on an individual basis to each TA.

Create automated tests/checks: Since we're looking at coding assignments, depending on the requirements of the assignment, teachers could presumably create automated checks on assignments, either as part of the grading process or as a guide to students on how well the current submission fulfills the requirements.

Submit assignments: Students should be able to submit responses to assignments, potential integration with common source control methods(gitlab/hub or git in general) would both encourage student usage of those tools and potentially make collaboration on group assignments easier

View assignments and requirements: Students should be able to view a list of assignments, their due dates, requirements set forth within the system by the professor and any supplemental documents.

View graded documents: Students will need to be able to view the grades they recieve on any submissions as well as view any comments or feedback left on specific portions or on the document in general

View automated test results: Students should be able to access automated test results if the tests have been set up by the instructor. Depending on the use case this may be part of the submission process (CS1050- make sure it compiles) or may be part of the grade feedback process (Instructors may not want the tests to be public to prevent the code being built to the tests)

Step 4: Constraints and Requirements

Hardware will need to be capable of:

- hosting the website/service users access the service through
- performing any automated checks(either those run by the professor or standard spam/virus checks) on uploaded files
- storing the instructor provided documents and student submissions
- Managing system access and authentication, or being linked into campus wide authentication
- Managing class membership
- Managing class permissions and enabling instructors to add or remove permissions from TAs