

# **Assignment 3: Requirements Analysis**

**Spring 2020: CS4320/7320 Software Engineering**

**Connor Travers**

# Problem

---

The system shall provide A method for easily submitting and collecting programming assignments.

- Purpose: To create a system which allows students in programming classes to submit programming work. Additionally, the TAs must be able to collect assignments. Finally, the instructors need to be capable of managing the course, its sections, TAs, and assignments.
- Client Base: CS Department – faculty, staff, and students

## Users and User Requirements

---

The user base shall consist of four main categories:

- **Students**
  - Shall be able to
    - Log-in to the student-side system using their University credentials
    - Navigate between enrolled classes
    - View assignments/grades
    - Submit assignments according to the instructor's restrictions (web link , file upload, text)
    - View the names of students in their section
- **Faculty**
  - Shall be able to
    - Log in to the staff-side system using their University credentials
    - Manage the level of access granted to certain users (i.e. promoting a student to TA/staff status and removing this status)
- **Staff**
  - Shall be able to
    - Log in to the staff-side system using their University credentials
    - Manage student submissions and grades
    - Manage assignments for their sections/courses
    - Manage the level of access granted to certain users (i.e. promoting a student to TA status, removing this status, adding/removing students from sections)
    - Manage their student roster
- **TAs**
  - Shall be able to
    - Perform all actions a student can for sections in which they are not a TA
    - Manage student submissions and grades
    - View the student roster

# Functional Requirements

---

The system's functional requirements can be divided into six main categories, and important data shall be underlined.

- **Log in**
  - The system shall be able to
    - Verify a user's credentials by comparing them to user/pass entries in a "**Users**" table
    - Direct the user to their dashboard *with a failure rate of less than .2% with 500 uses online*
- **Navigation**
  - The system shall display on the dashboard
    - A user's enrolled/managed courses by key searching their username in an "**Access**" table *with a failure rate of less than .01%*
- **Assignment Control**
  - The system shall be able to
    - Display the assignments for a course by referencing a course-specific "**\_Assignments**" table
  - Users shall be able to
    - View assignment properties through reference of the course-specific "**\_Assignments**" table
  - Staff shall be able to
    - Modify (due date, requirements, upload type, additional documents), Create, Remove assignments for a section/course through the course-specific "**\_Assignments**" table
    - Zip and download all submissions for an assignment, which will be housed in an assignment-specific "**\_Submissions**" table *with a total failure rate of less than 2%*
    - Manage assignment availability to students through the course-specific "**\_Assignments**" table
  - TA's shall be able to
    - Download all submissions for an assignment, which will be housed in an assignment-specific "**\_Submissions**" table *with a*
  - Student shall be able to
    - View the due date and requirements of upcoming assignments through the course-specific "**\_Assignments**" table
- **Assignment Grading**
  - The system shall
    - Be able to lock submissions to an assignment at the due date through a flag in the course-specific "**\_Assignments**" table
  - Student shall be able to
    - Upload their submission for an assignment to the assignment's "**\_Submissions**" table with a failure rate of *less than .1% at 500 attempted uploads within 2 minutes and at least 1Mbps up*
    - View their grade, which will be housed in the assignment's "**\_Submissions**" table

- View reuploaded graded submissions, which will be housed in the assignment's "**\_Submissions**" table
- Staff and TA's shall be able to
  - Assign grades to individual assignment submissions through the assignment's "**\_Submissions**" table
  - Upload graded programming work to the assignment's "**\_Submissions**" table with a *failure rate of less than 1% at 2Mbps Up*
  - Attach comments to specific students' grades in the "**\_Submissions**" table
  - View grades and submissions of all students in a specific course, pulling from appropriate "**\_Assignments**" and "**\_Submissions**"
- **Course Control**
  - Staff and Faculty shall be able to
    - Add and remove students from a section/course by editing the student's access in the "**Access**" table *with a 99.9% success rate*
    - Create/Remove a section/course through the "**Access**" table
    - View details of students in their course through query of "**Access**" and "**Users**" tables

## System Constraints and Requirements

---

### The system will require

- An internet connection
- A stable internet connection for uploads/downloads
- An internet-capable device with browser application