**CS313 - TA Management System**

**Requirements Document**

**Contributors:** Abigail Byram, Matthew Pointer, Matthew Tran, Neil Watson

**Advisors:** Caroline Budwell, Debra Duke

**Introduction**

The Computer Science department at Virginia Commonwealth University is quickly expanding to accommodate an increasing number of undergraduates. As class sizes grow, it is vital to develop efficient systems of managing and communicating with students to support student outcomes and retention rates. Undergraduate teaching assistants (TAs) can play a major role in facilitating learning by helping students on an individual basis, grading, and handling other tasks. A management system is needed to most effectively utilize the resource of TAs, encourage student outcomes, and allow courses to run smoothly. The purpose of this project is to design a system to solve the existing problems described below.

**The problems and what is needed to solve them**

* Communication
  + TAs often don’t know what’s going on in the lectures or what the professor told the students is expected
    - Option 1: create a detailed, accessible resource of course content
    - Option 2: include TA participation/assistance in lectures
  + Professors often don’t know the topics with which students are struggling
    - Maintain records of grades and the topics associated
    - Also maintain records of students asking for help or explanation
    - Allow professors to sign up for notifications indicating the most recent trends of student confusion points before class
  + Students don’t know when other students are having the same problems
    - Report student-accessible (anonymous) analytics of average performance in topics, visualizations and graphs
    - Offer a supervised discussion forum where students can ask questions and learn from each other (the discussion function of Blackboard or Slack may or may not be the most appropriate choice)
* Recruitment
  + Students don’t know where/how to sign up to be a TA:
    - Formalize TA application and responsibilities
    - Ensure that professors can screen TA applicants to select the most qualified students
  + Not enough TAs to help students during labs
    - Submit a formal proposal to the administration requesting funding for more TAs
    - Design consistent policies to most effectively use
  + TAs start without any training and often don’t know what to emphasize or how to effectively teach low-level programmers
    - Design training and orientation information for new TAs including responsibilities and common challenges for students
* Tutors are not being utilized as a resource
  + Include tutors into communication channel and ask that they also report the topics of student questions
* TA grading is inconsistent/possibly untrustworthy
  + Integrate grading into an online system which is viewable by the professor
  + Allow online assignment of grading by question instead of by student to minimize inconsistencies
  + Allow notes to be added dynamically to rubrics so graders can specify how they handled specific mistakes
  + Utilize a UI similar to Jupyter notebooks to integrate file handling, running code, reviewing code, and entering project grades into a single interface
* Students need individualized help with practical application of concepts and question-answering
  + Formalize TA duties to include more interaction with students such as in help sessions
  + Create an interface for students to schedule meetings with TAs or professors based on their reported availabilities
  + Divide duties among TAs based on experience and availability (e.g. grading TAs, office hour TAs, and lab TAs)

**The Proposed Solution**

Our proposed solution will consist of the following three components:

1. **A database** of teaching assistants, course materials, rubrics, and records of topics with which students struggle
2. **A web portal** to facilitate communication between students, TAs, and professors
3. **A design proposal** for department policies regarding Teaching Assistants

The details of these components are elaborated below.

**Database**

The database will contain information about current, past, and potential TAs, course topics, requirements, and rubrics, and listings of student questions, suggestions, (possibly anonymous) grades, and course evaluations. This database may be used to populate functions in the web application described below or for direct access by professors. Database will be hosted on \_\_\_

Entities:

* Course
* Professor
* Teaching Assistant
* Student
* Assignment
* Learning Objective (will take the function of what we call “topics” in the problem statements)

ER Diagram: TODO

**Web Application**

A web-based portal with the following functions: NOTE these tasks will need to be categorized

* Login function separated by professors, TAs, administrators, and students which each have a view of the database appropriate for that user’s access permissions
* Interact with the above database by accessing, updating, and adding entries based on permissions of the user’s view
* Allow professors to assign and review grading tasks
* Integrate grading into application (e.g. by overlaying gradescope or a Jupyter notebook-like method of grading code)
* Allow students to publicly (for the course) report topics they find confusing or have questions on, e.g. in a discussion board
* Allow private channel for students to submit problems, personal questions, or regrade requests (so these can be better managed than by email to multiple people)
* Scheduling function for students to see when professors and TAs (who have been assigned meeting duties) have listed themselves as available for meetings
* Live rubrics for each graded assignment where graders can make notes for how they dealt with special cases while grading (for grading consistency)
* Access settings alterable by professor of the course (e.g. granting more or fewer permissions to TAs)
* Output reports of average grades and current areas of difficulty (professors can choose whether to allow students to view this)
* Allow professor subscription to notifications based on the data reports (e.g. what topics students are currently more confused about)

**Policy Proposal**

The information and policies will be described in detail and organized into a single document, subject to departmental approval. The sections of this document will be as follows:

* **TA-Student Ratio:** The optimal number of TAs in a course to more efficiently support student needs - based on research of what other schools implement and publications in CS education, the amount of help outside of class time needed per course
* **TA Recruitment:** Describes method of applying to be a TA, how applicants will be evaluated, and how to match approved TAs to appropriate courses
* **TA Training:** Details on information all TAs will need before interacting with students, guidelines for dealing with difficult situations, distribution of responsibilities between new and experienced TAs
* **TA Duties and Expectations:** Describes the types of TA duties (grading, lab, classroom, help-session etc), maximum and minimum hours per week
* **Department Resources and Budget:** A clear description of the resources needed from the department to maintain this system. Includes justifications for each item
* **Professor Advisorship**: Suggested contributions from the course professor to act as an advisor to TAs and to maintain communication (\*mentorship?\*)
* **Management Responsibilities**: Describing who is responsible to maintain the management system (database, web app, and policy document)

The specific approvals needed to enact this proposal are:

* Budget approval from the CS Department and/or College of Engineering
* Collaboration with Engineering tutoring services to include CS tutors into course communication systems
* CS Undergraduate Committee approval to adopt these policies for TA management
* (Informal) Professor approval to ensure professors feel comfortable to enact these policies in their own courses, and that they believe this will make teaching easier