A Gamified Course Visualization, Organization, and Assessment System

Mark Floryan, a Computer Science professor at the University of Virginia, is dissatisfied with the way classes are currently taught. Floryan believes that strict deadlines and one-time assessments are not optimal for learning. Floryan seeks to understand the efficacy of gamification as an alternative method for learning. In his proposed system, students advance through a course as they advance through a video game: at their own pace, with deliberate practice and immediate feedback. Course material will be organized by topic, allowing for class progress to be visualized as a graph. Professors and students alike can see their progress in real time with on-site quizzes and automated grading.

Many classrooms today use the lecture-based exam model. In this model, lectures are the foundation of the course; quizzes, homework, and exams are all contingent on the lecture material. This solution forces students to learn at a controlled pace outside of the classroom, often causing students to move on from a topic before they have mastered it. Floryan's system aims to allow students to work at their own pace outside of the classroom and to make the mastery of a topic feel like beating a difficult boss in a video game.

Requirements are important because they enable clients to express their desires in a clear and unambiguous way. By converting client wishes to requirements and reviewing those requirements with customers, software developers can confirm that the product they will create matches the client's needs.

In the previous academic year, this project was started by a different team under the original set of requirements. Our contribution to the system includes adding security and authentication, improving performance at scale, aligning the system with Floryan's course, and adding quizzes and auto-grading. Based on several meetings with Floryan, we assembled this list of requirements for our contribution.

Minimum Requirements:

- The system must secure student data and grades such that they are only accessible by that student and by the course staff
- As a professor, I want students to only be able to see course topics that have been unlocked
- As a professor, I want to be able to lock and unlock topics from within the client
- As a professor, I want the students to be able to see their grade for level of competency per topic
- As a professor, I want to be able to upload grades in csv format
- As a professor, I want to be able to upload large amounts of data quickly (50,000 instances)
- As a user, I want the system front end to not experience notable lag when the database contains large amounts of data

Desired Requirements:

- As a TA, I want to be able to access and modify student grades from the frontend
- The system must be secure at the network level by encrypting traffic with HTTPS
- As a professor, I want the system to be able to store arbitrary assignment grades associated with a topic
- As a professor, I want to be able to toggle between cutoff grades and percentile grades per course and per topic
- As a professor, I want to be able to customize the thresholds for cutoff grades per the course and per topic
- As a professor, I want to be able to import grades from Bloomfield's new 2150 system
- As a professor, I want to be able to administer multiple choice questions
- As a professor, I want to be able to auto grade quiz submissions and provide immediate feedback
- As a professor, I want to be able to administer parson's problem questions
- As a professor, I want to be able to create quizzes from a question bank and to specify how the quizzes are to be generated
- As a professor, I want to be able to administer short answer questions
- As a staff member, I want to be able to grade short answer questions

Optional Requirements

- As a professor, I want to be able to administer short answer questions
- As a staff member, I want to be able to grade short answer questions
- As a professor, I want to be able to administer and grade coding questions

Figure 1.