Jason Bruce, Travis Burns, Aiden Leach

CS 246 - System Design

Requirements Document - Code Review Management System

Purpose and Scope

The objective is to develop a web application that facilitates the peer review process for academic assignments. The system will streamline the code review workflow and eliminate the need for manual partner assignments while maintaining review group integrity. The system will implement document management, automated partner assignment, review processing, and administrative interfaces. The project will support multiple file formats including direct code uploads, GitHub repository links, ZIP files, and general document formats.

Stakeholders

The internal stakeholders consist of instructors who will serve as administrators with full system access for their courses. Teaching assistants will handle assignment creation and management, along with review process oversight. Students will manage assignment submissions and peer reviews. A development team later on ‘could’ handle system administration, reporting, and technical oversight.

External stakeholders
include educational institutions that will implement the system, open-source contributors who may extend the system's functionality, and potential integration partners such as learning management system providers. Future stakeholders may include Moodle integration specialists for LTI implementation.

Functional Requirements

- System must be able to handle the upload of a variety of document types (pdf, word, ect) from students.
- System must have a database to store all documents, user information, course, and institution information
- Students and instructors will have separate dashboards:
    - Students will be able to view their courses, reviews that they have to do as well as ones already completed, and submit their code or view previous submissions.
    - Instructors will have access to reviews submitted by their students, courses and groups they have created, as well as upload documents such as new code review forms
- There will also be a dashboard for administrators which will be able to access any page within the site, as well as some additional analytics
- Instructors can create courses, assignments, and groups

    - Groups allow students in a class to be split, so that instructors can have different versions of the same assignment if desired.
    - Assignments will be tracked by the due date for code to be in for review, as well as the due date of the peer review
- Students will be able to register an account on their own, and then will be placed into courses/groups by their instructor
- System must automatically assign students within the same group to each other when ready to review.
- Students will ideally fill out a form in the web app, which is then stored and grades are uploaded to moodle via LTI (If time allows)
- System will be designed to work in a web browser, but should have basic viewing abilities on mobile devices
- Must be written in .NET 8

Non-Functional Requirements

- Needs to be scalable with at least several hundred users
- Usability: Site should be simple to navigate, without too many features that are hidden away on different pages
- Performance: must load in a timely manner and use asynchronous code whenever accessing the database
- Security: Secure login provided through .NET Identity with strong password policies

Assumptions and Dependencies

The primary technical constraint is the requirement for SmarterASP.NET hosting compatibility, with MySQL database support being beneficial for users. The system assumes users will have reliable internet access and basic technical proficiency. The core dependency is consistent hosting service availability for system operation.

Data Requirements

The system will process and store several types of data. Input data includes student submissions, review forms, assignment specifications, and user information. The system outputs organized review materials and assignment status updates. All data will be stored in a primary database with cloud storage maintaining system backups.

Interface Requirements

Aside from a home page and navigation, necessary interfaces will include a grading page that only instructors can access, a page for students to swap codes and submit forms, a register/login page, and administrative controls. There may also be a mobile layout design.

External Systems include Moodle LTI.

Risk Analysis and Mitigation

The key risks for this system involve technical operations during high-usage periods. System downtime during critical submission periods presents a significant risk, which will be mitigated through reliable hosting services and regular backups. Large assignment files could cause upload and download issues, which will be addressed through proper file size limits and robust error handling. Data loss or corruption poses another risk that will be managed through secure database management and backup systems.

Acceptance Criteria

Final system acceptance requires completion of all core functionality, successful deployment on the specified hosting platform, comprehensive documentation delivery, and verification of the partner assignment system. The system must demonstrate reliable file handling, proper user role management, and pass all security and performance requirements before handover.