

COSC 4P02 - Project Proposal

Team:

- Tennyson Demchuk - td16qq@brocku.ca - 6190532
- Mutaz Fattal - mf17lg@brocku.ca - 6362156
- Joel Gritter - jg17uy@brocku.ca - 6331763 (leader)
- Kindeep Singh Kargil - kk17xg@brocku.ca - 6329817
- Aditya Rajyaguru - ar18xp@brocku.ca - 6582282
- Daniel Sokic - ds16sz@brocku.ca - 6164545

Proposal Title: **COSC LMS (Learning Management System)**

Base level description: A web platform for computer science students to take tests, submit programming assignments, with automated testing and compiling to aid TA marking.

Why this matters: A platform like this would streamline assignment submission and marking, increasing the marking efficiency for TAs and professors, which reduces turnaround time for handing back marks and could allow for larger class sizes. Containerization of submission allows for decoupling of technologies used in courses with the technologies supported by sandcastle. Current submission platforms do not adapt to modern technology and computer science needs, such as integrated code compilation, our platform provides the foundation to support the ever growing department needs. This platform will also be able to support online learning a lot better than existing methods.

Core components/features:

- Web interface for submitting coding assignments (as opposed to ssh'ing into sandcastle or submitting a ZIP to Sakai)
- Automated testing - allows students to see some of the tests that they are being tested against, and TAs to get a quick understanding of the performance of the code.
- Automatic compiling - the code is compiled and a web interface with a terminal is provided to TAs, meaning that TAs don't need to manually copy-paste over all the code from Sakai into sandcastle
- Submission templates - Custom docker images can be provided by instructors for any kind of submission, and they will be able to select from preset templates, such as Java code submissions.

Potential extra features:

- Plagiarism detection (potentially Turnitin integration)
- Online code editing: Ability to just edit and run within the web interface. Eliminating complexity of different installations, reducing the need for labs.
- Ability to create examinations with coding questions, students can use the above online code editor with predefined starter code and compile/run bindings that can be customized by the test maker. Presets for popular languages like Java, Python, C++ can be included. Test cases for marking can be supported similar to assignments.

- Customizable layout for a course landing page: Allows professors to prioritize certain tabs, functions, and provide custom buttons (i.e. A button that directly takes students to a specific test or assignment)

Target users:

- COSC students
- Marker-graders / TAs
- COSC professors / instructors

Skills used:

- UI/UX
- Full-Stack development
- Containerization
- Storage management

Technologies proposed to be used (subject to change):

- React
- Express
- MongoDB
- NodeJS
- TypeScript
- Docker
- Kubernetes

Github repository: <https://github.com/JoelGritter/4P02>

Timeline:

(Weekly meetings, rough bi-weekly goals)

Sunday, January 17, 2021 - Proposal Submission

Wednesday, January 20, 2021

- Requirements specifications complete
- UI wireframe proposals

Wednesday, January 27, 2021

- Use-case and sequence diagrams complete
- UI wireframe finalization

Wednesday, February 3, 2021

- System architecture diagrams complete
- Deployment diagrams complete

Sunday, February 7, 2021 - Requirements + Overall design submit

- Requirements specifications
- Use-case and sequence diagrams
- System architecture diagrams
- UI wireframes
- Deployment diagrams

Wednesday, February 17, 2021

- Basic project setup, hosting setup, deployment automation and integrations
- Basic web development complete (i.e website skeleton done)
- Template dockerfiles for container templates complete

Sunday, March 7, 2021 - Mandatory Report 1

Monday, March 8, 2021 - Friday, April 2, 2021

- Complete MVP and backlog

Saturday, April 3, 2021 - Mandatory Report 2

Sunday, April 4, 2021 - Sunday, April 25, 2021

- Refactoring, cleanup, documentation, potentially implement stretch goals

Monday, April 26, 2021 -> Friday, April 30, 2021 - Final Demonstration + Presentation + Report

- Report with updated requirements doc, user manual, installation manual, etc.
- 15 minute presentation about the software
- 5 minute demo of the software

Task Management Methodology: Kanban

We will be using Kanban to facilitate the development of this project, this will include designing a kanban board which will have user stories. There will be stages for the user stories which range from backlog, to user-stories for work today, user-stories being done currently, blocked user-stories, and completed user-stories. There will also be a limit to how many user stories can be in progress at a time.

For more information: <https://www.atlassian.com/agile/kanban>

Weekly meeting time: Wednesdays, 2:00pm-3:00pm