Assessment Management System for Maties CS

Michael Shepherd, 19059019

September 25, 2019

1 Project Description

At Stellenbosch University, computer science students spend a lot of time doing practical work. Managing all of this is a very large task, and the existing solutions are very frustrating and have led to a number of unofficial solutions being made by different people. Maintaining and using the current solutions is time consuming for lecturers and demis that need to manage and mark large amounts of submissions that come in many different shapes and forms.

This project aims to solve this problem by creating a submission platform that can be easily used from both the submission and submission handling ends. This platform should be able to work through the US single sign on system and the actions on uploads will be configurable from the submission handling end. The configurations should allow for easy testing and marking of submissions via scripts and frameworks, reporting of results as well as calling on plagiarism checkers.

There is also a lack of assessment environments at Stellenbosch university. This project will also try to provide a way to create an assessment environment that is also highly configurable and easy to use. This assessment environment should be able to limit computing resources, such as CPU cores and memory and network access. The assessment environment should also provide a configurable set of programs and services that can be used by the student.

The plan of action for the implementation of this project is to create a web-app with a Python backend that is easy to use for lecturers, demis, students and maintainers. This backend should be usable from both the web-app and from a command line interface.

2 Human Interaction

2.1 Submission System

The submission system will allow a lecturer to configure a project to their requirements and allow students to submit files that fit the configurations, after logging in with their student numbers. The submissions will then be available to the lecture and teaching assistants for marking purposes. The marks will be stored in a google document and the presentation of these marks will be up to the lecturer. This is intended as a clear and unambiguous alternative to the sunlearn submission system to make the process easier for the students as well as the markers. The only data that will be kept by this project will be student numbers, the submitted assignments and the marks associated with the submission. The submitted assignments will be stored on the university servers and backed up in accordance to the standard data handling protocols of the Stellenbosch university computer science department. Only the lecturers and teaching assistants will have access to the marks and assignments.

2.2 Assessment Environment

The assessment environment will be a way to limit a students access to their computer during a test or assignment. This will mean that the computer will look the same, but the students will only have access to approved websites, tools, documents, environments, libraries and programs. The assessment environment will be configurable by the lecturer to fit the needs of the current test or assignment. There will be technical demonstrations before the tests and assignments as well as technical assistance during to ensure that the students are not negatively affected by the new environment.

2.3 Real World Testing

The project will be tested with the help of the students of Scientific Computing 272, which is a course presented by this projects supervisor, Willem Bester. The testing will take place during the assessments scheduled for the 17th and 25th of Ocober 2019. These assessments count for 14% of the students' year mark when put together. The submission on the 17th of October will be an assignment under test conditions and the submission on the 25th of October is an open submission that is open for a week.

2.4 Dealing With Issues

In the case of an issue with the project or the students during or after the assignment, the existing submission system will be available at all times. That is to say that if the system were to fail, there is an alternative submission that the students are familiar with. The students will not be penalised in any way if they were to submit through the existing submission system rather than this project. There will also be a voluntary feedback form for the students, lecturers and teaching assistants to fill out to give me an indication of the success of the project.

2.5 Protecting Data

The assignments and associated marks will not be publicly available. The only people that will have access to this information will be the lecturers and teaching assistants involved with the marking. The marks will be made available to the students by the lecturer in any way that they see fit.

The access that I will have to the data will be equivalent to the access that any teaching assistant would have. This will be so that I can asses whether the project is working correctly or not.

3 Goals

The Goals of this project can be grouped around three requirements:

3.1 Secure Submission

1. There will be secure upload functionality for submissions.[9] [7]

2. There will be the ability to run tests on uploaded submissions. [7] [5]

3. There will be the ability to run marking scripts on uploaded submissions. [7] [5]

4. There will be the ability to call plagiarism tests on the uploaded submissions. [7] [5]

5. The results of actions on the submissions will be inputted into online services like Google sheets.[2]

6. There should be continuous integration of the submissions with the use of Git hooks.

3.2 Assessment Environment

1. There will be the ability to spawn a secure and scriptable assessment environment. [5] [4] [3] [1]

2. The environment should have access to a curated set of programs and services.[5] [4] [3]

3. The environment should have a configurable limited network connection.[5] [4] [3]

4. The environment should have a configurable limited computing resources.[5] [4] [3]

4 Constraints

- The project should be built on standard, freely-available software, so that it will be possible and convenient for the technical staff of the university to support. [5] [4] [3]
- There project must be usable from both a web-app interface and a command line interface. [1] [8] [6]
- Both of the interfaces must be simple and intuitive. [6] [7]
- Ethical clearance will need to be obtained to involve the students in marking.

5 Expected Difficulties

5.1 Meeting Goals

For the assessment environment, it may not be possible to curate accessible programs or the level of network connection. The security goals are also quite involved, as the project will have to account for the possibility of malicious submissions in both the secure submissions and the assessment environment.

5.2 Implementation

Bringing all of these sections together into one user friendly and powerful project looks to be where struggles may be encountered. Using the US single sign on system correctly to ensure security of the submissions may also present a diffuculty.

5.3 Technologies

This project will involve working in a way in which I have no experience. It would appear that learning how to integrate technologies such as Git and Docker with a Python backend will be needed to achieve the goals of this project. This will require a depth of knowledge of these, and related, technologies that I do not have yet.

6 Reading List

[1] Joe Coburn. *How to Create Your Own Command Line Programs in Python With Click.* 2018. URL: `https://www.makeuseof.com/tag/python-command-line-programs-click/`.

[2] Google. *Python Quickstart.* 2019. URL: `https://developers.google.com/sheets/api/quickstart/python`.

[3] Jeff Hale. *Learn Enough Docker to be Usefull.* 2019. URL: https://towardsdatascience.com/learn-enough-docker-to-be-useful-b7ba70caeb4b.

[4] Ocelot Uproar Ltd. *Learn Docker & Containers using Interactive Browser-Based Scenarios.* 2019. URL: https://www.katacoda.com/courses/docker.

[5] Asad Memon. *How we used Docker to compile and run untrusted code.* 2016. URL: https://blog.remoteinterview.io/how-we-used-docker-to-compile-and-run-untrusted-code-2fafbffe2ad5.

[6] Thomas Stringer. *The easy (and nice) way to do CLI apps in Python.* 2016. URL: https://medium.com/@trstringer/the-easy-and-nice-way-to-do-cli-apps-in-python-5d9964dc950d.

[7] Pallets Team. *Uploading Files.* 2010. URL: http://flask.pocoo.org/docs/1.0/patterns/fileuploads/.

[8] Scott Torborg. *Command Line Scripts.* 2012. URL: https://python-packaging.readthedocs.io/en/latest/command-line-scripts.html.

[9] tutorialspoint.com. *RESTful Web Services - Introduction.* 2019. URL: https://www.tutorialspoint.com/restful/restful_introduction.htm.