CS 102 *Spring 2020/21*

Project Group **G2C**

Instructor: **Aynur Dayanık**
Assistant: **Haya Shamim Khan Khattak**

Criteria	TA/Grader	Instructor
Presentation		
Overall		

~ LabConnect ~

Group Name

Borga Haktan Bilen 22002733
Vedat Eren Arıcan 22002643
Berkan Şahin 22003211
Berk Çakar 22003021
Alp Ertan 22003912

Requirements Report

(Draft Version (pre-review))

February 23, 2021

1 Introduction

LabConnect facilitates communication between students, TA's, tutors, and instructors. In the background, it is mainly a web application (can be ported to Android possibly) that aims to assist CS introductory courses in terms of organization and communication. Proposed ideas for features include priority queuing for TA zoom rooms. For example, those who have completed their labs can be tested using pre-defined (by TA or instructor) unit tests and ordered from most complete to least, in order to decrease waiting times for students who are done with their labs, and to optimize the process in general. TA's can also use the system to see previous versions of each student's code in a more practical way, similar to real version control managers in spirit. The style guidelines put forth by the instructors can be enforced automatically by parsing the student's sent code files. Much of the repetitive work that course staff need to do can be reduced substantially by automated actions, allowing TA's to allocate time for more hands-on help towards students. The student experience can be improved further by adding helpful features such as personal notes for students and so on.

2 Details

LabConnect is designed to contain three user interfaces for instructors, students, and assistants/graders. It will also contain a server side program where the submissions are stored and tested.

2.1 LabConnect - Instructor Side

2.1.1 Prior to the lab

- The instructor decides upon the name and the language of the assignment
- The instructor uploads the instructions either as a document, or as a Markdown or a plaintext file, in which case it will be rendered and displayed on the website
- The instructor writes the unit tests as input-output pairs and groups them if they wish. Some groups of unit tests can be hidden, in which case they won't be shown to the students prior to submission.
- The instructor can determine a time constraint for unit tests. If the execution of the code takes longer than the determined time, it will fail said test.
- The instructor determines a time frame for submissions. They can determine a seperate deadline for re-submissions if they wish.
- The instructor can assign students to assistants either at random or by hand. They can also choose to not assign assistants at all, in which case the students will be assigned to the assistants at the time of the lab, based upon the length of the queue.

2.1.2 During the Lab

- The instructor will have better control over how code of the students' are tested. The instructor will be able to add more unit tests as the lab progresses. The students' unit tests can be updated within the lab period so any mistakes made on the tests itself can be corrected this way.

2.1.3 After the Lab

- The common errors that are made by the student such as missing documentation of the written code, conventions that aren't followed (naming conventions, styling guidelines), will be detected by LabConnect. This data will be shared with the student and their instructor. The instructor can later on determine to act up on the most important mistakes that are made by the students.
- With the unit tests, instructors will be able to see which end cases of the program that the students mostly failed at. These unit tests can again show the common weaknesses of the programmers.
- The instructor will be able to assess their students properly by considering their performance in the lab. LabConnect will provide detailed performance of the student, based on the mistakes they made, and overall unit test scores. This information can help the

instructor to have more idea about their students since the data is properly organised and accessable.

2.2 LabConnect - Student Side

2.2.1 Prior to the Lab

2.2.2 During the Lab

2.2.3 After the Lab

2.3 LabConnect - Grader/Assistant Side

2.3.1 Prior to the Lab

2.3.2 During the Lab

- As stated, the program will be able to detect common errors made in student code. Also, the unit tests can be arranged in a way they that they test specific skills that are required. During the lab, these properties of the program will increase the efficiency of the graders.

- The Grader can create pre-messages for the students if there is a highly repeated mistake. This way, they won't need to repeat the same small correction for each student, and this way they will find easier time to correct other unique mistakes that the students have made. The graders will find greater motivation on teaching the students proper techniques if they are freed from the repetitiveness of same mistakes. Of course the student can demand further help for their "repeated" mistake.

2.3.3 After the Lab

2.4 LabConnect - Server Side

2.4.1 Prior to the Lab

2.4.2 During the Lab

2.4.3 After the Lab

3 Summary & Conclusions