**Code TA**

Project Proposal

For

**ENGT 4050**

**Senior Technology Capstone**

Prof. Richard Springman

`

Project Team B-12

Brandon Collins, Team Leader

Stephen Wambui

Sierra Linn

Ben Morris

February 5, 2014

The University of Toledo

College of Engineering

Engineering Technology Department

**Introduction**

Computer programs, no matter how simple or complex, all have one thing in common: each had to be coded. Every computer function is performed by some sort of program which was written in a programming language. This has increased the demand for programmers and other computer-based careers, which in turn increases the number of computer-based, technological courses in schools worldwide. All these courses have assignments that have to be graded, which can be time consuming, especially with complex assignments or a large number of students. Even with the help of a teaching assistant (TA), the work load can still be overwhelming and cumbersome. In order to reduce the work load and make grading code easier, Code TA is being designed as a program to read submitted code, grade it and give feedback in a timely manner. Code TA will help professors evaluate and grade student-written code, making the process more efficient and time-effective. This document will describe the motivation of the project, give a summarized view of the system, provide various details of the system, and give a list of referenced material for further reading at the end.

**Motivation**

The current system of grading has been proven to be time-consuming. Instructors have been faced with the task of grading the code of hundreds of students, and at the same time are expected to give meaningful feedback, especially if errors occur. For open-open ended assignments, such as those involving coding, this increases the work of the instructor substantially. Some professors may compensate for this by not completely evaluating submitted code, which cripples students’ learning abilities when they fail to receive critical feedback. A heavy workload on professors also restricts the available classes and the sizes of classes, despite the fact that demand is high. Automated grading systems could be a breakthrough that helps both professors and students by reducing the workload, allowing for increased professor availability and providing students with feedback.

The problem of a heavy workload for professors has prompted other attempts to automatically grade assignments. Recently, a Stanford University research team came up with a program that could evaluate program code called “Codewebs” [1]. The program is used to grade assignments as a part of what they call “massive open online courses” (MOOCs) [2]. However, this system does not organize content in a way that resembles a typical university class structure, and it comes coupled with a database of previous solutions to which it compares new ones for the sake of finding common mistakes and trends among students. It also revolves around a set of predefined assignments that are coded into the system. A tool better suited for professor use, however, would aim to tackle the problem in a simplified manner, by grading a wider variety of submissions on an individual basis. It would also organize and store assignments and students in a way that is consistent with a traditional class.

**Project Summary**

Code TA is a web-based application that will be used to grade student-created programs. Users will go to the Code TA website and organize their programming-related classes and assignments, as well as upload code that needs to be graded. A server-side program then compiles the code and assigns it a grade. It will generally be used for classes in which there is a need to test and grade long or complex programs. In all, this project will be useful for professors who need to grade programs for an entire class of students efficiently.

**Project Details**

**Project Scope and Elements**

Code TA is a web application that provides automated program source code grading. Code TA lets instructors or students submit code that will be automatically compiled, run, and graded based on output of students' programs. There are a few components of Code TA: the web application front-end, the web application back-end, the server-side code grader, and the database to hold all the data produced by the application. Figure 1 shows how these elements will interact with one another.

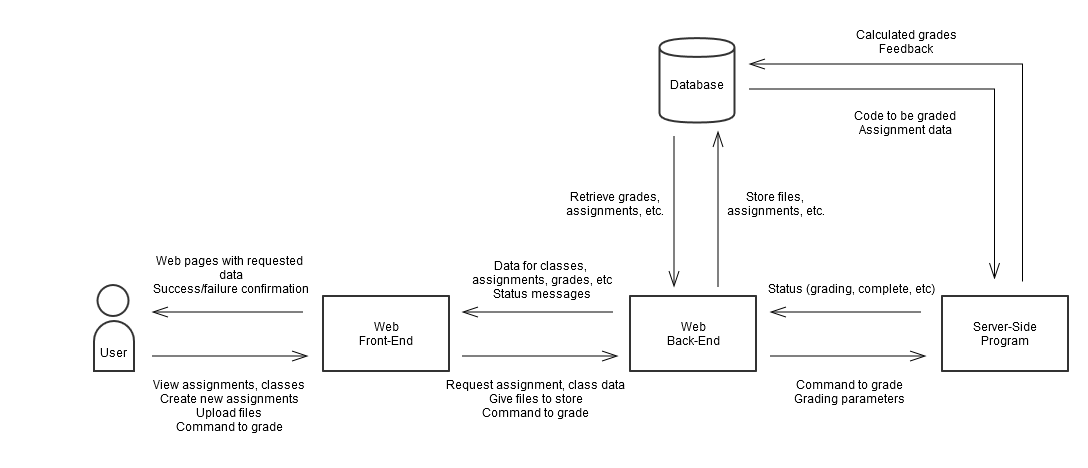


Figure 1. Data flow between Code TA components

Combining these elements of Code TA will allow instructors to post assignments and users to submit corresponding code to be graded.

Code TA will be built using agile software development practices. This allows Code TA to be built in incremental steps and functionality kept modular. Code TA will be run on a GNU/Linux server running the Debian distribution. The hardware for the server can scale up with demand. If many students are going to be using Code TA, the server hardware can be upgraded. For use by a handful of instructors, the server could be run on a small $500 server. Code TA will be built around Python. The web application front-end and back-end will be built using the web framework Python-Flask, and the server-side grader will also be built using Python. The database used to store all the data will be PostgreSQL. This combination of GNU/Linux, Python, and PostgreSQL gives us a cheap and flexible platform to build Code TA on.

**Implementation Issues and Challenges**

The four main elements of Code TA, the web application front-end, the back-end, the database, and the server-side grader, all have their own unique implementation challenges.

The front-end will be facing the end-user, so the interface must be well-designed and easy to use. A big challenge with front-end development is making sure the front-end UI looks good and performs well on a variety of platforms and devices. With screen sizes of devices ranging from 3.5 inches to 27+ inches, it will important to carefully construct the front-end to be responsive to most devices.

The database needs to be both simple and efficient. Relationships between the various entities should be straightforward and normalized as much as possible. Having multiple intertwined relationships could lead to inconsistencies in data, so it must be simplified when designed.

The server-side grader and back-end of the web application also face some challenges, mostly regarding maintainability and scalability. The code for the back-end is going to be complex, and it is important to make sure the code is clean and well designed. In the future, Code TA might be expanded with additional capabilities, and it is very important to make sure that the source code for both the program and the back-end are easy to maintain and as modular as possible. Keeping the complexity of the code down will be a huge challenge.

The project will be constructed on the distributed source control platform Github. Github provides decentralized source code version control that makes it easy to build the software. With Github and Git, the only tools needed to build Code TA will be a Linux machine and a code editor.

Skills in software engineering and system administration will be required. Some of these include designing and coding software, developing and maintaining a website and web server, and managing source code across multiple workstations.

**Deliverables**

Code TA is made up of three parts which will be delivered on completion of the project. These three parts are the website (made up of the front-end and the back-end), the database, and the grading program. Each of these pieces works in conjunction with one another to provide the entire functionality of the project, but they can be separated to be used individually as well.

The website is the interface by which users are able to access the data and use the functions of Code TA. It can be further divided into two parts: the front-end and the back-end. The front-end encompasses everything that the user can see and interact with. The back-end is the web-based application that makes the website run behind the scenes and interacts with the other components of the larger project such as the program and the database. Using the website, users will log in, which will give them access to their available classes, assignments, and grades. They can also add new classes and assignments. Users will be able to upload code to be graded for each assignment, or view previously graded code, along with comments as to why the grade was received. Code can be uploaded for each student on each assignment. Instructors will be able to issue commands to grade the assignment and set the parameters on which the code should be evaluated. Doing so will produce a grade that is associated with each individual student. Initially, all users are planned to be instructors, who will upload all code for a particular assignment. If extra time is available during the course of the project, student users will also be introduced, which will allow students to upload the code themselves before the professor issues the command to grade the assignment.

The database is the storage for all information associated with classes, students, grades, assignments, and users for Code TA. It organizes all data placed into the system in a manner so that the website can easily retrieve and display it. Users will not interact with the database directly, but will be able to see and modify its contents by interfacing with the website. Classes will be made up of a list of students, as well as all assignments associated with them. Students are always associated with a class, but can be associated with multiple classes. Assignments are also associated with an individual class. Submissions for an assignment are designated by a combination of a student and assignment, and have a grade and comments that are automatically generated by the grading program. Submissions have a .zip file of code associated with them, and assignments have the option of having files associated with them as well (such as any files necessary to complete the assignment). User data is all stored into the database as well, such as email and password. Users may be stored as instructors, students, or potentially both.

The grading program is the portion of the project which automatically grades and evaluates student written code. It is a Python-based program which is able to examine, compile, run, and evaluate programs, as well as assign them a grade based on the output, or optionally the source code. It is designed to run on a Linux-based server, and can interface with both the website and the database. A user will use the website to send a command to grade code and adjust the grading criteria. The program will receive this input from the website and retrieve the necessary data from the database (such as project files). After running and evaluating the code, the program will then store a grade and comments as to why the grade was received into the database.

Using all of these components together will produce a single system that is able to organize programming classes and assignments in a user friendly way, and gives instructors a centralized, efficient way to grade code for an entire class all at once.

**Timeline**

Code TA must be completed by April 25th. Development has been divided into seven separate tasks, which will be completed in increments of several weeks each. The list of tasks and time to complete each is as follows:

* Database schema – 1 week
* Web-end – 4 weeks
* Database implementation – 1 week
* Web back-end – 3 weeks
* Server-side grading program – 4 weeks
* Database management and support code – 2 weeks
* User testing – 2 weeks
* Integration testing – 2 weeks

Figure 2 lays out each of these tasks on a rough schedule in the form of a Gantt chart.

Figure 2. Gantt Chart for Code TA

**Budget**

The estimated budget of Code TA is quite small. Most of the tools needed come in the form of free or already owned software, so the cost of development is negligible. However, there will be two things needed in order to deliver a final product: web hosting and a domain name. Both can be obtained relatively cheaply, depending on the source. Web hosting for the purpose of this project can be purchased for roughly $5 a month [3], and a domain name will cost about $10. Assuming the immediate purchase of the necessary materials, the total cost of this project will be approximately $30. With these low costs, members of the group will be able to fund this project out of pocket.

**Conclusion**

Due to the need for an automated grading system, Code TA will be a useful tool for professors, as it provides an easy and efficient way to grade code and provide feedback for an entire class all at once. Using the web front-end, web back-end, database, and server-side program grader in conjunction with one another, Code TA will be a system that makes grading student-written code simpler and more time-effective.

**References**

# [1] H. Hodson: Robotutor marks the homework of a class of thousands. In NewScientist Magazine issue 2944, November 2013.

[2] J. Huang, C. Piech, A. Nguyen, and L. Guibas: Syntactic and Functional Variability of a Million Code Submissions in a Machine Learning MOOC. In Proceedings of the 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on Artificial Intelligence in Education, 2013.

# [3] T. Trapani: Most Popular Reliable and Affordable Web Hosts. Lifehacker, 6 January 2009. http://lifehacker.com/5124856/most-popular-reliable-and-affordable-web-hosts