DEVELOPMENT OF A COMPREHENSIVE LEARNING MANAGEMENT SYSTEM DESIGNED FOR COMPUTER SCIENCE AND MATHEMATIC DEPARTMENT WITH BUILT-IN AUTOMATIC CODE TESTING

by

Bailin He

Thesis submitted in partial fulfillment of the

requirements for the Degree of

Bachelor of Computer Science with

Honours in Jodrey School of Computer Science

Acadia University

July 2018

This thesis by Bailin He
is accepted in its present form by the
of Jodrey School of Computer Science
as satisfying the thesis requirements for the degree of
Bachelor of Computer Science with Honours

Approved by the Thesis Supervisor

Dr. James Diamond	Date

Approved by the Head of the Department

Dr. Darcy Benoit	Date

Approved by the Honours Committee

(The honours committee person)	Date

I, Bailin He, grant permission to the University Librarian at Acadia University to reproduce, loan, or distrubute copies of my thesis in microform, paper or electronic formats on a non-profit basis. I, however, retain the copyright in my thesis.

_____

Signature of Author

_____

Date

Acknowledgements

Contents

List of Tables

List of Figures

Abstract

This thesis presents a learning management system (LMS) written in Python using Django web framework. This system is designed specifically for computer science and mathematics instructors and students to ease the process of submitting and, more importantly, the process of grading assignments, with a built-in code testing module for programming assignments to build and run all students' submitted solutions and provide informative test results for students and markers, as well as features like syntax highlighting, Markdown input and TEXnotation input for mathematical equation rendering for markers and instructors to comment on the students' assignments.

Chapter 1

Introduction

Many courses in computer science involves programming in several programming languages. The nature of programming assignments evaluation involves process like code testing, debugging, comments, code styles. Which makes the evaluation process of a programming computer science assignment very different than any conventional course assignments. Also, it would surely motivates students to hand in a higher quality assignment if more information were provided shortly after their assignments were submitted. Therefore, a learning management system (LMS) that can auto-test the submitted assignments and have the results provided for both the markers and the students is desired.

1.1 Background

Acadia University currently uses Moodle Learning System (also known as Acorn) as the default LMS. This system works well for most of the courses, however, it gets shorthanded when it comes to computer science programming assignments due to the nature of programming source code files and the lack of certain features, including but not limited to monospace text font for plain text files display, programming language detection and syntax highlighting for source code files display. Also, it is not uncommon for computer science markers and instructors trying to put mathematical equations or formulas into their comments on students' assignments, the only

two ways to do that with the Moodle Learning System are draw the equations or formulas by hand with the pointing device of the computer (i.e. computer mouse, trackpad), or write the equations or formulas in *HTML* code. Both of these two ways are time-consuming and borderline unfeasible. Moreover, there are features in the Moodle Learning System that are rarely used by instructors and students (especially instructors and students in computer science department), those features increased the complexity of the system and resulted in the fact that an easy task often takes a longer time to finish than it was supposed to. This phenomenon is known to be a "Feature Creep" (Boram Lee and Woohun Lee, 2015 [12]), the Microsoft Office Suite is generally considered to be an example of this phenomenon (Des Traynor [13]). In short, the Moodle Learning System lack of features that are essential to computer science students and instructors, at the same time could slow down the process of submitting and evaluating assignments with degree of complexity of the system.

The *Course2000* course system designed by Dr. Rick Giles is a good choice of alternative LMS to the Moodle Learning System, in fact it has been the default LMS for one computer science course, *COMP 2103: Programming 3*, for over a decade, with the *Visual Mark* module designed by former student Tim Cooper integrated, the *Course2000* system has proven itself to be very capable of handling computer science assignment. However, with more and more modifications had applied to the system over the years, maintaining this system has become more and more difficult.

1.2 Course2018

Influenced by the *Course2000* system, the goal of this project is to develop a secure and dynamic LMS with functionalities designed specifically for computer science instructors and students, functionalities that mainly focus on providing informative feedbacks on programming assignments to both students and instructors for the purpose of easing the process of learning as well as evaluation, while maintaining the simplicity of the overall system design to make sure every task the system needs to perform can be done efficiently for both the users and the server, also to make sure that the future development and maintenance will not be too difficult.

1.2.1 Key features

The *Course2018* LMS is mainly characterized by the functionalities described below.

Source code management

The *Course2018* LMS provides the abilities for students to manage their source code more efficiently and more professionally, abilities like source code version control, namely, students and instructors are now able to view the history of every source code file, and more importantly, compare the changes between different versions of each file. This is achieved by integrating one greatest open-source version control system, *Git*, which was originally developed by *Linus Torvalds* (Anthony Scopatz & Kathryn D. Huff, 2015 [11]), creator and principal developer of the *Linux* operating system kernel (Linus Torvalds & David Diamond, 2001 [10]), as well as *GitLab*, an open-source Git-repository manager developed by GitLab Inc [3].

Souce code display

A good source code display system with syntax highlighting and line numbering can significantly improve the readability of source code files and thereby ease the process of evaluating. In *Course2018* LMS, this is achieved with two steps: the first step is programming language detection using a *Unix* library called *libmagic* [5], once the programming language is determined, a highlighted text will be generated with one most widely used syntax highlighter, *Pygments* [6], in an *HTML* format.

Auto testing programming assignments

When a programming assignment is submitted by a student with *Course2018*, the source code files will be pushed to a remote *GitLab* instance, then those files will be built (if required) and executed immediately by one script defined by the instructor prior to the assignment was published that contains test data like sample input and output, time limit and sample program return code in a *GitLab* subsystem, *Gitlab CI/CD* [4], finally, an informative test report will be available for the student who submitted the assignment, markers, and the instructor after the execution is finished.

Informative assignment feedback

The key to providing informative assignment feedback comments in computer science assignments (especially programming assignments) is to make sure the comments are well formatted. The *Moodle Learning System* provides what it is called a "WYSIWYG" (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et, *"referring to what is displayed on the screen being the same as what will be printed out"* – Collins English Dictionary [8]) text editor to achieve this goal, however, WYSIWYG editors are generally inefficient when the formatting gets complicated especially if mathematical equations are involved.

Alternatively, an easy-to-use plain text markup language called *Markdown* is introduced to the *Course2018* marking system. This markup language was designed in a way that it can be converted to an *HTML* format, which is exactly what this project needs, and it has been widely used by famous websites including *GitHub* [2] and *Stack Overflow* [7]. Moreover, a *Javascript* library call *MathJax* is also integrated to the marking system so that TEX and LATEX mathematical notation can be recognized and displayed.

Powerful built-in text editor

With an enhanced assignment feedback solution being mentioned above, a powerful text editor that supports syntax highlighting, and custom key-bindings is also desired to ease the actual process of typing for the markers and instructors. A web-based code editor called *Ace* [1] is chosen to be integrated to the marking system for the reasons that firstly it satisfies all requirements mentioned above, and secondly it has been the default editor for a lot of famous projects like the *Cloud9 IDE*, which is now part of the *Amazon Web Services* (Ruben Daniels, 2016 [9]).

Bibliography

[1] Ace editor. Ajax.org. URL https://ace.c9.io/.

[2] Mastering markdown. GitHub Inc., . URL https://guides.github.com/features/mastering-markdown/.

[3] About us. Gitlab Inc., . URL https://about.gitlab.com/about/.

[4] Gitlab continuous integration (gitlab ci/cd). Gitlab Inc., . URL https://docs.gitlab.com/ee/ci/.

[5] magic(4). The Linux man-pages project. URL http://man7.org/linux/man-pages/man4/magic.4.html.

[6] Pygments faq. pygments.org. URL http://pygments.org/faq/.

[7] Markdown help. Stack Exchange, Inc. URL https://stackoverflow.com/editing-help.

[8] *Collins English Dictionary Complete and Unabridged edition*. HarperCollins, 2014. ISBN 0007522746.

[9] Ruben Daniels. Cloud9 now runs on and integrates with aws. c9.io education community, 2016. URL https://c9.io/announcement.

[10] Linus Torvalds & David Diamond. *Just for fun : the story of an accidental revolutionary*. HarperBusiness, New York, NY, 2001. ISBN 0066620724.

[11] Anthony Scopatz & Kathryn D. Huff. *Effective Computation in Physics: Field Guide to Research with Python*, page 351. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2015. ISBN 1491901535.

[12] Boram Lee & Woohun Lee. Feature creep in design students' works: Why and how it happens in student design processes. *The Design Journal*, 18(3):345–365, 2015. doi: 10.1080/14606925.2015.1059603. URL `https://doi.org/10.1080/14606925.2015.1059603`.

[13] Des Traynor. When you think about feature creep and bloated products what comes to mind? endless tabs, toolbars, settings, and preferences, right? Inside Intercom. URL `https://www.intercom.com/blog/what-does-feature-creep-look-like/`.