

Submit: An Online Submission Platform for Computer Science Courses

Nolan Burfield
Sergiu M. Dascalu

Hardy Thrower

Brandon Worl
Frederick C. Harris, Jr.

Department of Computer Science
University of Nevada, Reno
Reno, Nevada 89503

Abstract

Computer Science class sizes have increased for several years and will continue to grow. Consequently, instructors dedicate more time grading the increased number of programming assignments they receive. Grading programs individually by hand is inefficient and error prone. Submit implements a deployable web platform that alleviates the problem of inefficiency by automating the grading process, simplifying the submission, and providing instant feedback to students. The Submit web platform allows both students and instructors to focus on their work without additional complexities of managing submissions. As class sizes grow, solutions such as Submit will become necessary to ensure a smooth and efficient work-flow.

1 Introduction

Submit is an Internet-based submission platform for computer science classes. The platform targets students, instructors, and graders. Students are able to enroll in courses and submit the source code for their programming assignments. Submit will automatically grade the assignments. The parameters for a grade is derived from test cases defined by the instructor. The success rate of the test case determines the grade of the assignment, but may be modified by the instructor. The automated grading feature reduces the grading time for instructors and graders. Submit allows the students to receive instant feedback regarding their assignment. These simple functions of Submit makes the platform a necessary program with the rise in computer science enrollment.

Based off statistics from the Computing Research Association (CRA) [3] class sizes have been growing since 2007 and this trend is likely to continue. As the number of enrolled students increase in a class, an instructor's spends less time focused on class content and more on grading the assignments. With a scalable online

submission platform, the burden of manually running and examining student code is eliminated. This means that class sizes can grow without forcing instructors to make compromises by removing necessary content from assignments to make more time for instruction. Submit allows Instructors give more instruction and one-on-one time since they will spend less time managing the logistics of potentially hundreds of files turned in every week.

In addition to usability improvements and added functionality, Submit needs security. Submit will supersede an older submission platform that is written with PHP, which is insecure if not implemented correctly. This security is a flaw with outdated software that is a vulnerability in a web site not updated regularly [7]. Submit takes advantage of Ruby on Rails, which is a modern web application framework. Rails has security features for a web platform that is built into the foundation [6]. The security features of rails will protect users data from both outside attacks as well as internal tampering by users trying to find exploits in the application.

2 Similar Platforms

The University of Nevada, Reno currently has a program similar to Submit. The current system grades assignments, handles files, and provides grades. However, the system does not provide an adequate user interface, security, or modern design. These flaws are attributed to ten year old software that is not updated, written in PHP code, and uses a file storage system on a server. Submit implements numerous usability improvements including an efficient user interface, more interactivity, and a secure and organized file storage system. More courses will be able to take advantage of the system due to the improved interface and dynamic back end that submit offers. The old submit platform also has the security errors as discussed in the introduction section.

Blackboard Inc. [8] offers a course management

platform that can be taken advantage of by an entire university. The software developed by Blackboard provides students with online access to their courses, data associated to the courses, and submission options. Instructors are able to make assignments, upload files, and give grades all on the Blackboard site. Students are able to view the assignments and turn in projects, quizzes, and tests. When a student takes quizzes or tests on the platform their grades are calculated and added to the overall grade. This grading feature is only available to multiple choice assignments, and not capable of code handling. Submit differs from Blackboard by having the ability to compile and grade the student programs.

A similar online software for course management is built by Instructure Inc. [9]. This platform offers all the same features as Blackboard. Students are able to turn in assignments, but there is not the same grading option that Submit will offer to users.

3 Requirements Specification

The requirements specification section for Submit is formatted based off Ian Sommerville's book *Software Engineering* [12].

3.1 Functional Requirements

Submit shall provide the following functions and features. (Functional Requirements are labeled as **(F)**):

- F01** Allow users to make accounts.
- F02** Allow users to edit their accounts.
- F03** Allow users to edit files within the web page.
- F04** Allow students to enroll in courses.
- F05** Allow students to submit files to assignments.
- F06** Allow students to view assignment grades and grader comments.
- F07** Allow instructors and admins to create courses.
- F08** Allow instructors to create new assignments.
- F09** Allow instructors to open-close course enrollment.
- F10** Allow instructors to specify the how to compile and run uploaded submissions.
- F11** Allow instructors to specify start and due dates on an assignment.
- F12** Allow instructors to mark students as a grader of the course.
- F13** Allow graders to function as an instructor without the ability to create courses.
- F14** Allow graders to run submissions against test cases.
- F15** Allow graders to create comments on a student's submission.

3.2 Non-Functional Requirements

Submit shall provide the non-functional requirements. (Labeled as **(NF)**):

NF01 Be developed with Ruby on Rails.

NF02 Be fast at serving pages and files.

NF03 Be intuitive and quick to learn.

NF04 Be compatible across all major web browsers.

NF05 Support many students in multiple courses.

4 Use Cases

The use cases of the system outlines the functionalities provided. Use cases specify user(s) and the functionality provided to them. In the Submit system there are four types of users: Administrator, Instructor, Grader, and Student.

UC01 Sign Up: All users must have an account to access Submit. They are required to give an email, name, and password.

UC02 Enroll in Courses: Students and graders need to enroll in a course to access the assignments. Courses have a twelve character registration token that must be given out by the instructor to enroll in the course.

UC03 Grade Submissions: Instructors and graders should be able to grade student submissions, either one at a time or all at once.

UC04 View and Comment on Submissions: Instructors and graders should be able to view code submitted by a student and provide feedback through comments.

UC05 Edit Users: The Submit administrator should be able to view all enrolled users and edit their information and roles.

UC06 Create Courses: Instructors should be able to create new courses, specifying the course name, term, and year.

UC07 Create Assignments: Instructors should be able to create new assignments in courses that they have created.

UC08 Create Test Cases: Instructors should be able to create test cases which consist of inputs and expected outputs, which all submitted code for a given assignment will be tested against.

UC09 Manage Enrolled Students: Instructors should be able to view all students enrolled in a course and edit their roles or remove them from the course.

UC10 Submit Assignments: Students should have the ability to submit their assignment once they are satisfied with their work. Once submitted, files may

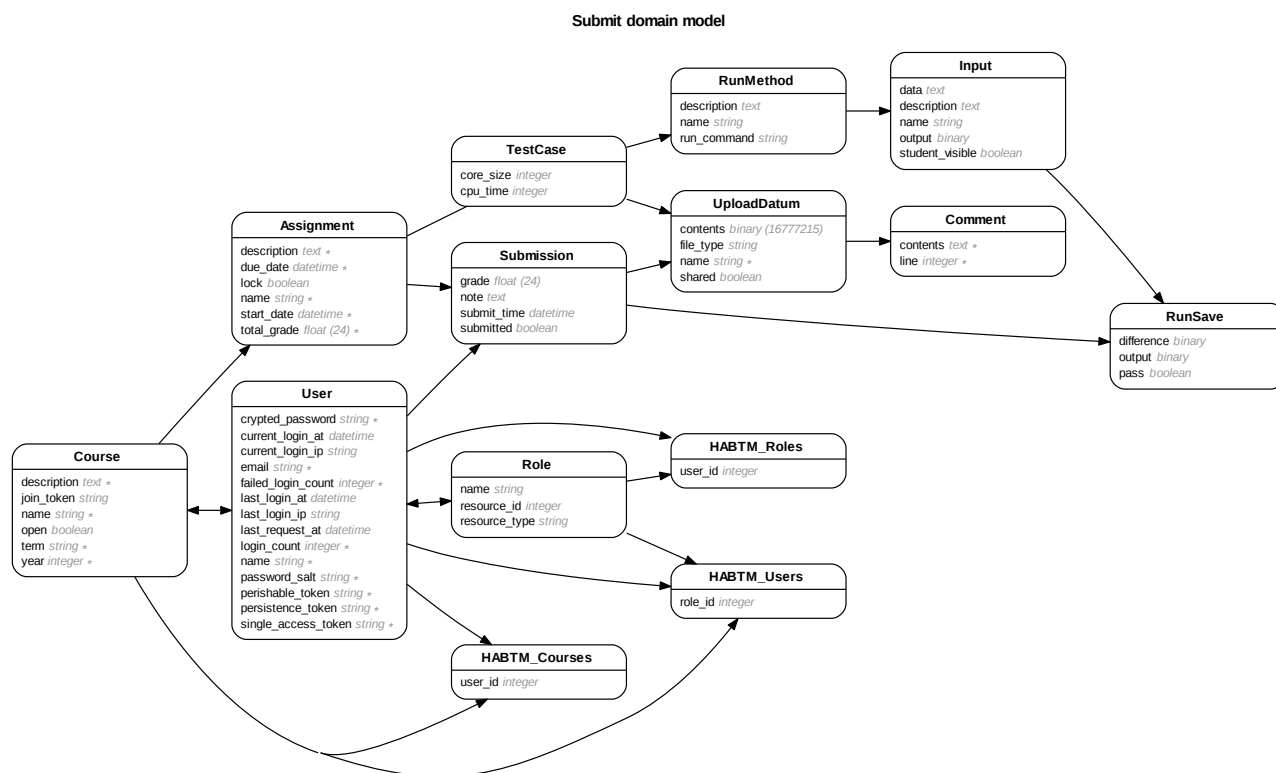


Figure 1: A graphical representation of the Submit database and table relationships.

not be edited any more, but grader comments will be visible to the student.

UC11 Run Test Cases: Students, graders, and instructors should be able to run a submission's code against the set of test cases at any time. Certain test cases may be marked as hidden, and will only be visible to graders and instructors.

UC12 Edit Submissions: Students should be able to edit their submission by adding, removing, or modifying their uploaded files. Additionally, files may be edited directly with Submit's built in text editor.

5 Design Overview

Submit is a web platform built with Ruby on Rails. Rails is a web application framework that uses model-view-controller (MVC) architecture. Rails offers simple integration with a MySQL database, Syntactically Awesome Stylesheets (SASS), jQuery, and asynchronous JavaScript and XML (Ajax) [5]. The MVC design aids in the separation of tasks for similar page requests, user interaction, and database implementation [4]. The views section holds the information on how to display the web page, the controllers handle requests, and the

models handle data. The MySQL database is what the model integrates with and stores all the data associated with the Submit web site.

5.1 View Design

A view generates an output for the user. Submits views include a visually appealing graphical design and simple user interface. Each view is connected to a controller action that will communicate data between the two. A Rails view is built in a HTML/Embedded Ruby file, which allows both HTML/CSS integration and Ruby logic. Rails builds the web display based off the view file and the controller specified data.

5.2 Controller Design

Submit is driven by a set of controllers that are responsible for linking the front-end views with the back-end models and database. Each controller is responsible for a particular component of Submit, such as user sessions, courses, or uploaded files. These controllers are further divided into actions which are linked to the application's routes. When a particular route is taken, the associated controller action locates

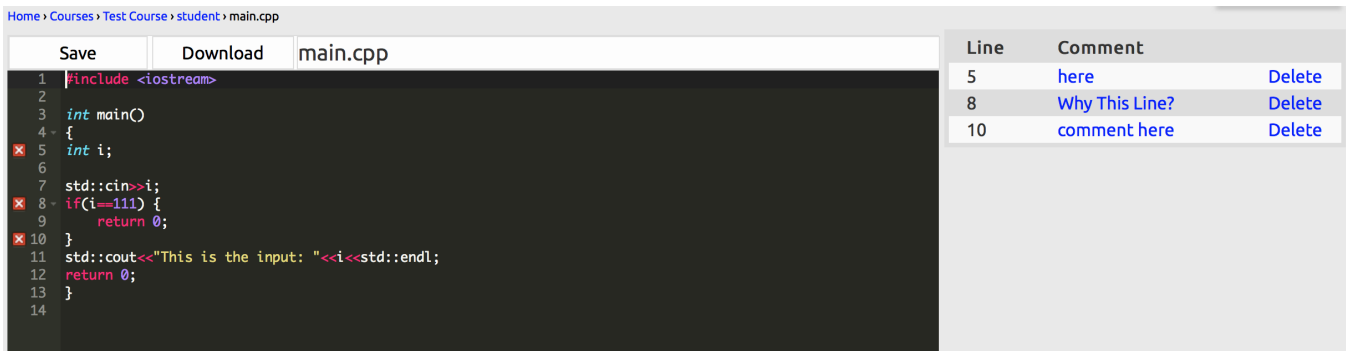


Figure 2: The file editor showing a simple program with comments made on the page.

the required models and passes them to a specific view. The controller uses Ajax to remove the need for reloading page content for simple tasks such as deleting a file; Ajax will call the necessary controller and update the page with JavaScript [11].

The controllers that Submit uses are outlined below with a description of the controller functionalities. All the controllers unless specified handle the functionality of create, read, update, and destroy (CRUD).

Application: This is the base controller created by Rails. All other controllers extend this Application controller. There are no views or models associated with it.

Assignments: This module handles everything directly relating to assignments. The Assignments model runs several validations to make sure due dates make sense, all required fields are set, and that all students get a submission object made within the assignment object.

Comments: This module handles the creation and deletion of the comments. The placement of comments in a file is handled by the Upload Data controller.

Courses: This module manages courses. The course model runs several validations to make sure all required data is set and within acceptable value ranges. Courses do not perform much logic, they are mostly used as a container of assignments and for enrollment purposes.

Inputs: This module processes an input for a test case on an assignment.

Run Methods: This module is a container for multiple input objects and the run command. The run command specifies how to run the compiled program.

Submissions: This module contains all of the files and grade information for a student’s submission to an assignment. Submissions can be locked, at which point they will be graded and can no-longer be modified.

Test Cases: This module processes the test cases that get configured by the course instructor for an

assignment. The controller sets constraints on how an assignment is ran, such as the max core dump size and runtime in seconds. This object also contains a collection of run method objects. Test case objects contain the instructor’s source code, which the controller uses to automatically generate correct outputs for input objects.

Upload Data: This module handles uploaded files. It runs several validations to ensure that names are not duplicated, both amongst the upload data in a submission and the shared upload data in a test case.

User Sessions: This module handles user authentication, and is part of the AuthLogic gem, which handles the session logic.

Users: This module handles user account information, such as user name and email.

5.3 Model Design

A Model represents a singular object in Rails. They are responsible for validating data and then saving data to the database. They can also have methods attached to them to perform certain actions [10]. For example, the assignment model has a method that can clear all run-saves associated with itself, which is useful when the assignment’s test cases change, thus rendering old run-saves obsolete.

5.4 Database Design

Submit uses a MySQL database to store all data associated with the website. This includes user data with encrypted and salted passwords, uploaded file data, course information, and more. The database structure is relational, meaning that relationships exist between database tables, such as “has-many”, “belongs-to”, and “has-and-belongs-to-many” [2]. The database relationships can be seen in Figure 1.

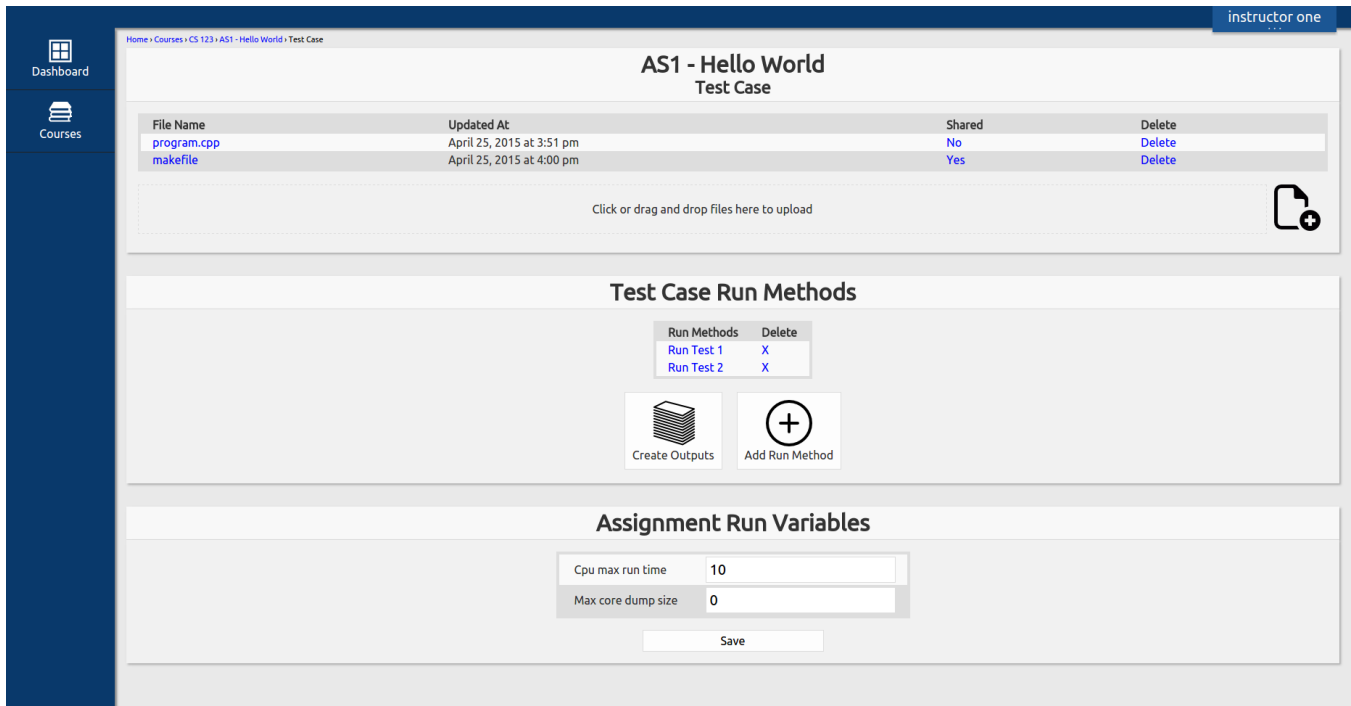


Figure 3: An instructor view of the test cases. From this page an instructor uploads a program, creates run methods, adds inputs to the run methods, and sets program parameters.

6 User Interface

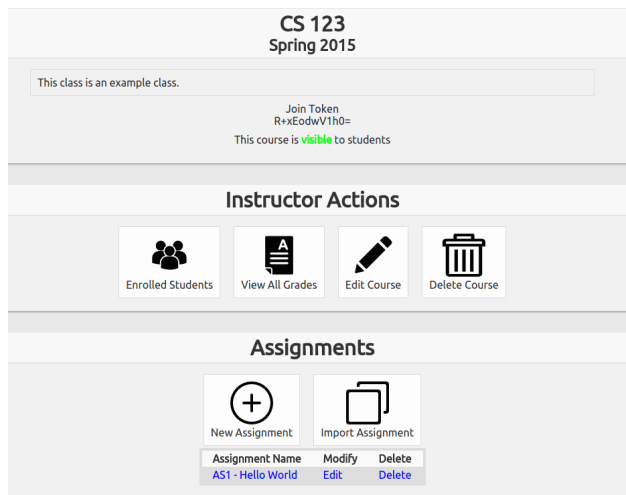


Figure 5: This is a cutout section of the instructor course management page. In this view the student join token can be seen, the enrolled students can be accessed, all student grades, and the assignments for the course are managed.

The interface for all four users is designed to be intuitive, and remove tedious web site navigations. This is the main improvement that was made from the original Submit website. The interface can be broken down into two different views: one for students, and the other for instructors. The student view is the default view for users that are not assigned a role; this is due to the restrictive use of the site a student user is given. As an instructor role this allows the creation of classes, an instructor class management page can be found in Figure 5. Students are given enrollment tokens which are specific to each class, these tokens allow students to enroll in the class.

Automatically grading assignments is Submits main feature. An instructor and grader create assignments in a course. Each assignment contains test cases that specify how to run a program. A view of the test case page is in Figure 3. In a test case, the instructor will upload a working program that will be used to generate correct outputs. These outputs are derived from the inputs created within a run method. A run method specifies the command line call to run the programs for an assignment. There are two parameters that will set the max run time on a CPU the program will run for to prevent infinite loops, and the max core dump size

Program Output

1 Tests Out OF 2 Passed

Quick Links

main

input_1 input_2

main

this will run the main program
main

[Back To Top](#)

input_1	
give number 12	
Input Data	
12	
Your Output	Expected Output
This is the input: 12	This is the input: 12
Difference	
No Difference	

[Back To Top](#)

input_2	
input the number 111	
Input Data	
111	
Your Output	Expected Output
	This is the input: 111
Difference	
+This is the input: 111	

Figure 4: This shows the output view for running a program. The input data, output, expected output, and difference are shown. If an output is incorrect a red box is placed around the input name and the output result, as shown in bottom box.

from a program to prevent large files being stored on the server.

Submit focuses on instant student feedback. To achieve this, an output of student code is compared to the expected output from the test cases. Submit calculates the difference between the two outputs and generates a form for the user to see correct results or erroneous results. A view of the output with a correct result and incorrect result can be found in Figure 4.

A major improvement Submit makes upon its predecessor is the ability to edit code directly online. A user has access to Ace, a high performance code editor. Ace has built in functionality that makes code editing easy [1]. This feature adds additional support to the student and grader use. A student is able to make minor modifications to code in order to fix output errors. The instructor/grader is able to comment directly on students code as well. This enables the student to remove the need for tedious uploads, and for quicker and more thorough grading of code. The editor and comments is displayed in Figure 2.

7 Conclusion

The Submit web platform brings features to the computer science classes that enables simple code submission as a student and quick grading as an instructor. Submit is built with Ruby on Rails and takes advantages of its security, simplicity, and the easy integration with Ajax, MySQL database, and SASS. Submit enables students to join classes, manage programming assignments, and test their programs online. Instructors benefit from Submits automatic grading feature and course management system. Submit offers efficient grading and submitting of programming assignments.

8 Future Work

Submit is capable of automatically testing student code, but there are many more features that can be added to it. Anonymous peer-review of submitted code is one planned future feature that would allow students to anonymously comment upon another student's code to provide feedback and suggestions. When it comes to assignments cheating needs to be spotted. The ability to flag any submissions that appear to be cheating would be a necessary function. This could include

submitted code that matches other submissions for that same assignment, or code that appears to have been copied from another location on the Internet. To allow the management of local programs and Submits database is a useful function for students.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under grant no. IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] AlloyUI. Tutorial-ace editor, 2013. <http://alloyui.com/tutorials/ace-editor/> Last Accessed 4/21/2015.
- [2] David Aragon. Introduction to database design on rails, 2012. <https://quickleft.com/blog/introduction-to-database-design-on-rails/> Last Accessed 4/21/2015.
- [3] Computer Research Association. Undergrad computer science enrollments rise for fifth straight year, 2013. <http://cra.org/govaffairs/blog/2013/03/taulbeereport/> Last Accessed 4/21/2015.
- [4] Kalid Azad. Intermediate rails: Understanding models, views and controllers, 2007. <http://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/> Last Accessed 4/21/2015.
- [5] Ryan Bigg. Rails guides, 2014. http://guides.rubyonrails.org/getting_started.html Last Accessed 4/21/2015.
- [6] Ryan Dewhurst. Ruby on rails security basics, 2014. <https://www.netsparker.com/blog/web-security/ruby-on-rails-security-basics/> Last Accessed 4/21/2015.
- [7] Anthony Ferrara. Is php secure? “it is if you do it right”, 2015. <http://www.acquia.com/nl/resources/podcasts/acquia-podcast-69-php-security-anthony-ferrara> Last Accessed 4/21/2015.
- [8] Blackboard Inc. Blackboard, 2015. <http://www.blackboard.com/> Last Accessed 4/21/2015.
- [9] instructure Inc. instructure, 2015. <http://www.instructure.com/> Last Accessed 4/21/2015.
- [10] Will Little. How to design and prep a ruby on rails model architecture, 2013. <http://www.startuprocket.com/blog/how-to-design-and-prepare-a-ruby-on-rails-model-architecture> Last Accessed 4/21/2015.
- [11] Tutorials Point. Ajax on rails tutorial, 2014. <http://www.tutorialspoint.com/ruby-on-rails/rails-and-ajax.htm> Last Accessed 4/21/2015.
- [12] Ian Sommerville. *Software Engineering*. Addison-Wesley, Harlow, England, 2010.