

Central Online Grading System

Team Members:

Forrest Scott, Daniel Riechers, Zachary Lones, Daniel McDonough, Kathryn Widen

Faculty Adviser/Client:

Professor Thomas Daniels

DEC1521

Introduction

Problem Statement:

Grading Code submitted through blackboard is tedious and time consuming. TAs who are grading student's code must manually download and compile the submissions from each individual student. With a large amount of students in a course, this becomes very time consuming.

Solution:

Create a website where students can submit assignment code, as well as provide a streamlined single page grading process to instructors. The website will include interfaces for students to customize their code arguments and view their compile and run results. Lastly, The website will be equipped with MOSS modules that will aid instructors in cheating detection.

Requirements

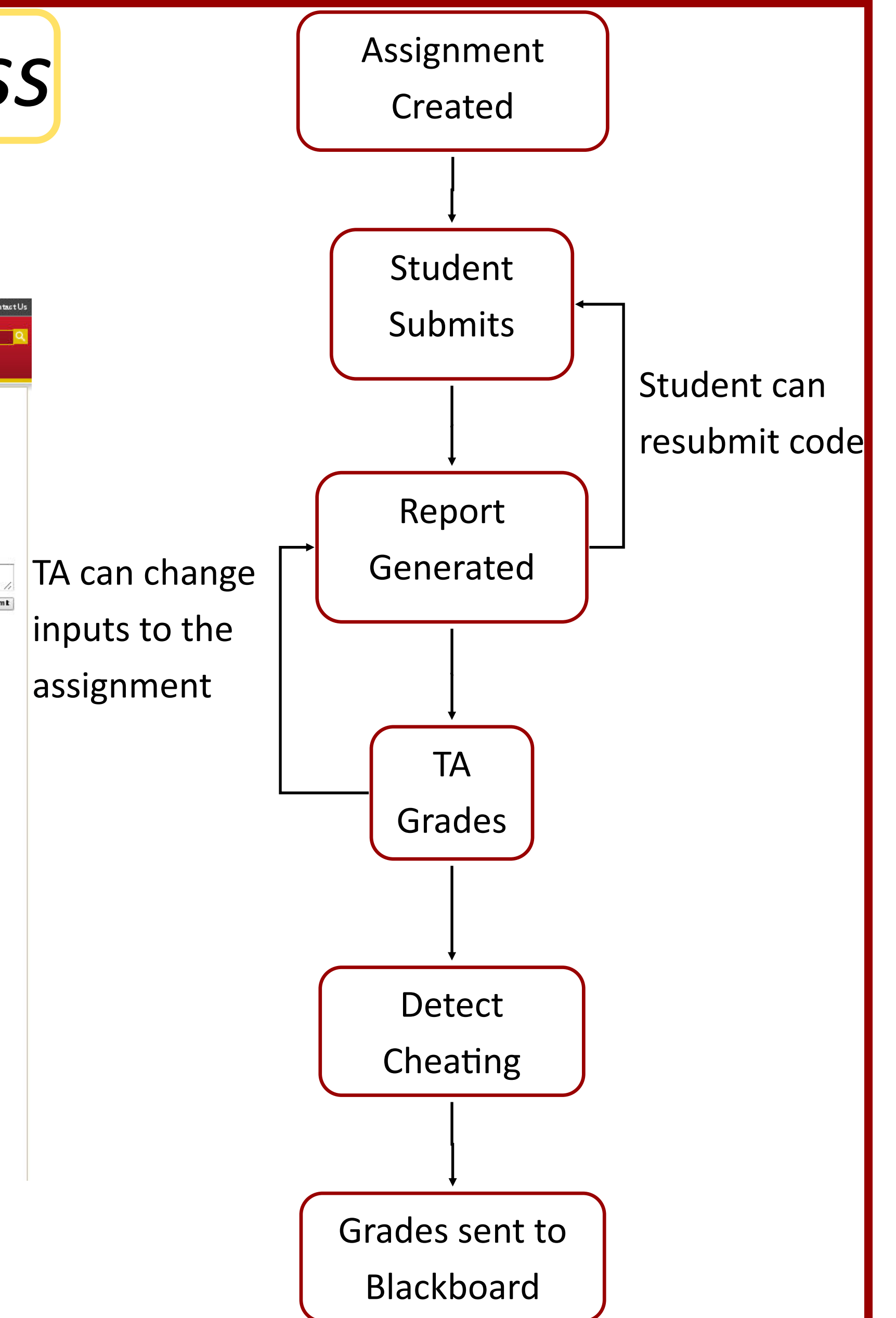
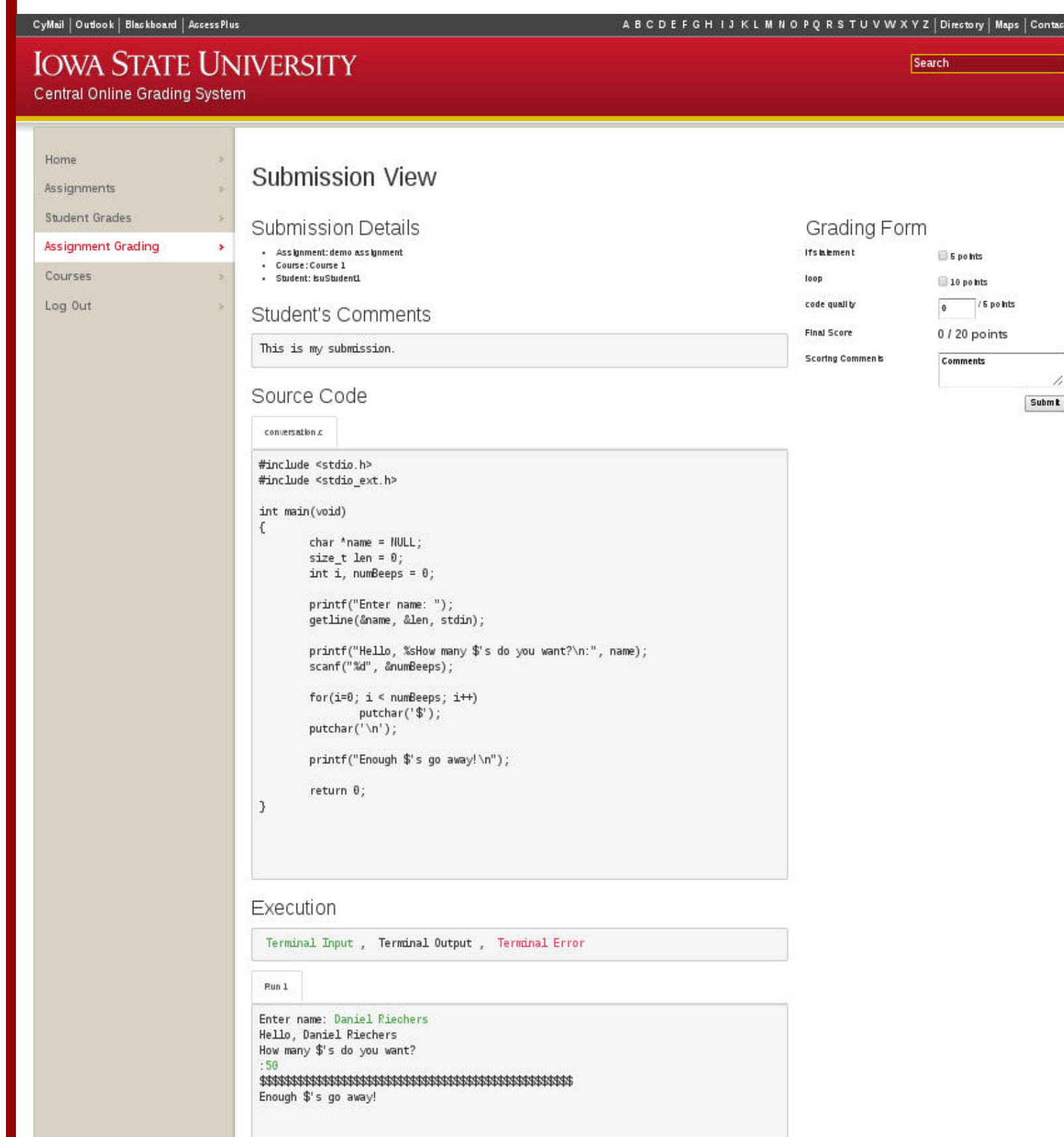
Non-Functional

- System will prioritize efficiency for TA allowing grading of assignments to be done with minimal time
- All COGS software will be verified to be secure, reliable, and easily maintainable

Functional

- Student submitted code will be compiled and students will receive compiler report
- Instructor will be able to create assignments with components necessary to define grading rubric
- The system will include cheating detection that looks for similarity between student's code
- To protect the system, student's code will be run in chrooted environment with SELinux policy

New Grading Process



With Blackboard, TAs were then responsible for downloading the .C file, compiling the code, running the executable, and then grading the code's output and the code itself. Now, these steps are automated by COGS and a TA is given everything needed to grade all on one page. This page includes student's source code and executables, student's comments, tools to assign the student's grade and provide comments to the student.

Testing

Our Back end was designed through test driven development, meaning that unit tests were created before we started coding. This ensures that code reaches full coverage of requirements and gives us a way to test while coding. Our security policy and cheating detection were manually tested.

COGS Overview

