Supporting Team Submissions and Peer Grading within Submitty

Matthew Peveler, Samuel Breese, Evan Maicus, Andrew Aikens, Timothy Cyrus, Elizabeth Dinella, James Anderson, Josh Barthelmess, Marisa Lee, Leon Montealegre, Jessica Wang, Buster Holzbauer, Barbara Cutler, and Ana Milanova

Rensselaer Polytechnic Institute
What is Submitty?

- Students upload code (and resubmit) for auto-grading
- TAs review and add additional grading/feedback
- Configurable number of late days per gradeable
- **Open-source, free to use**
- **Installed on your own hardware or VPS**
  - Instructors have ssh access to files & logs for debugging
- **Support for any language / tool** installed on your server
  - We use Python, C, C++, Java, Scheme, Prolog, and SPIM
  - JUnit, Emma code coverage, Dr Memory, static analysis, ...
- **Supports dozen of courses with thousands of users** at RPI
  - 500-700+ students in Computer Science I and Data Structures
Server Performance

Weekly Average Homework and Lecture Exercise Submissions

- # homework submissions
- # lecture exercise submissions
- average wait time
- average grading time
More Information

● All material from this demo available at
  https://submitty.org/tutorial
  https://github.com/Submitty

● New users are welcome! Ask us questions:
  submitty-admin@googlegroups.com

● New developers are welcome:
  Rensselaer Center for Open Source Software (RCOS)
  Sponsored by RedHat Software
  Google Summer of Code 2018
  To access our Slack server:
  http://submitty.org/developer/
Today’s Demo

- Support for team assignments  new for Fall 17!
- Student, grader, and instructor views
- Manual grading interface with common marks  new for Fall 17!
- Support for Peer Grading  planned for Summer 18!
- Submitty Discussion Forum  new for Spring 18!
- Autograding with External Tools
- Static Analysis for Autograding  poster at 2pm!
- Flexible late day policy, plagiarism, student stress  poster at 2pm!
- Sandbox/Docker  poster at 2pm!
- Future Goals
Team Assignments

Instructors specify maximum size and “team lock date”

<table>
<thead>
<tr>
<th>What is the type of the gradeable? (Required)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic File</td>
</tr>
</tbody>
</table>

Is this a team assignment? *Team assignments are new as of Fall 2017. Email questions/bugs/feedback to: submitty@cs.rpi.edu.*

- Yes
- No

What is the maximum team size? [2]

What is the **Team Lock Date**? (Instructors can still manually manage teams): [2016-02-22 16:23:34-0500]

What is the **Submission Open Date**? (submission available to students): [1971-01-01 23:59:59-05] must be >= TA Beta Testing Date

What is the **Due Date**? [9996-12-31 23:59:59-05] must be >= Submission Open Date

How many late days may students use on this assignment? [2]

Are students uploading files or submitting to a Version Control System (VCS) repository?

- Upload File(s)
- Version Control System (VCS) Repository

**Full path to the directory containing the autograding config .json file:**

See samples here: Submitty GitHub sample assignment configurations

- `/usr/local/submitty/more_autograding_examples/upload_only/config` (an assignment without autograding)
- `/var/local/submitty/private_course_repositories/My_Course_Name/My_Homework_Name/` (for a custom autograded homework)
- `/var/local/submitty/courses/s18/sample/config_upload/#` (for an uploaded configuration)

Should students be able to view submissions?
Create a new team and invite teammates

Or accept an invitation from a classmate
Instructor View of Teams

Instructors monitor team formation, and can create/edit teams
Team Submissions

- Anyone on a team can submit or resubmit
- All team members share one sequence of versions
- TAs add manual grades for teams just like single students
Student STDOUT.txt
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -3 and -3

Expected STDOUT.txt
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -3 and -3

Student STDOUT.txt
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -2 and -12

Expected STDOUT.txt
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -0.5 and -3

Standard Error (STDERR)
WARNING: This file should be empty
1 ERROR: -2 is not a root of this formula.
2 ERROR: Unable to verify one or both roots.
Git Integration

- Can be configured with an internal git server
  - Installed to /var/local/submitty/vcs
  - Students get a web path to access
    ex: https://submitty-vcs.cs.rpi.edu/git/

- Can use external sources as well, such as Github
Instructor Configuration

What is the Submission Open Date? (submission available to students): 2018-02-23 07:58:38-05  must be >= TA Beta Testing Date

What is the Due Date? 2018-03-02 23:59:59-05  must be >= Submission Open Date

How many late days may students use on this assignment? 2

Are students uploading files or submitting to a Version Control System (VCS) repository?

- Upload File(s)
- Version Control System (VCS) Repository

Path for the Version Control System (VCS) repository:
VCS base URL: http://192.168.56.102/git/s18/sample
The VCS base URL is configured in Course Settings. If there is a base URL, you can define the rest of the path below. If there is no base URL because the entire path changes for each assignment, you can input the full path below. If the entire URL is decided by the student, you can leave this input blank.
You are allowed to use the following string replacement variables in format ${...}
- gradeable_id
- user_id OR team_id OR repo_id (only use one)
EX. /${gradeable_id}/${user_id} OR https://github.com/test-course/${gradeable_id}/${repo_id}

VCS URL: http://192.168.56.102/git/s18/sample/${gradeable_id}/${user_id}

Full path to the directory containing the autograding config.json file:
See samples here: Submitty GitHub sample assignment configurations
/usr/local/submitty/more_autograding_examples/upload_only/config (an assignment without autograding)
/var/local/submitty/private_course_repositories/MY_COURSE_NAME/MY_HOMEWORK_NAME/ (for a custom autograded homework)
/usr/local/submitty/courses/s18/sample/config_upload/# (for an web uploaded configuration)

Should students be able to view submissions?
- Yes
- No (Select 'No' during grading of a bulk upload pdf quiz/exam.)
New submission for: Final Project

No automatic grading for this assignment

To access your Repository:
Note: There may be a delay before your repository is prepared, please refer to assignment instructions.
git clone http://192.168.56.102/git/s18/sample/final_project/student SPECIFY_TARGET_DIRECTORY

Grade My Repository

No submissions for this assignment.
• **Components** represent top level rubric items
• “Marks” add and/or subtract points (“count up” or “count down”)
• Instructor can “**publish**” certain marks (student sees the grading criteria, even if the mark was not selected for them)
● **Components** represent top level rubric items
● “Marks” add and/or subtract points (“count up” or “count down”)
● Instructor can “publish” certain marks (student sees the grading criteria, even if the mark was not selected for them)
- **Components** represent top level rubric items
- "Marks" add and/or subtract points ("count up" or "count down")
- Instructor can "publish" certain marks (student sees the grading criteria, even if the mark was not selected for them)

![Manual/TA/Peer Grading Rubric](image_url)

<table>
<thead>
<tr>
<th>Read Me</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message to TA/Grader (seen only by TAs/Graders)</td>
<td></td>
</tr>
<tr>
<td>Message to Student (seen by both students and graders)</td>
<td></td>
</tr>
<tr>
<td><strong>0</strong> Full Credit</td>
<td></td>
</tr>
<tr>
<td><strong>-1</strong> Minor errors in Read Me</td>
<td></td>
</tr>
<tr>
<td><strong>-2</strong> Major errors in Read Me or Read Me missing</td>
<td></td>
</tr>
<tr>
<td><strong>Add Common Deduction/Addition</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manual/TA/Peer Grading Rubric</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message to TA/Grader (seen only by TAs/Graders)</td>
<td></td>
</tr>
<tr>
<td>Message to Student (seen by both students and graders)</td>
<td></td>
</tr>
<tr>
<td><strong>0</strong> Full Credit</td>
<td></td>
</tr>
<tr>
<td><strong>-5</strong> Code is unreadable</td>
<td></td>
</tr>
<tr>
<td><strong>-3</strong> Code is very difficult to understand</td>
<td></td>
</tr>
</tbody>
</table>
Grading With Common Marks

Customize panel layout
Add/edit marks during grading
Improves consistency between graders
We support bulk pdf upload (test/quiz), name matching and manual grading
Peer Review/Grading

● Instructor configures some components for peer review
● Instructor specifies # of peer review assignments
  ○ Assigned randomly
  ○ Or assigned by .csv upload
● Peers use same interface as TAs to review/grade
  ○ Access restricted to relevant files and dates
  ○ Student identities are randomized
● Average scores given by multiple peer graders
  ○ Compute consistency of grades — “grade the grader”
Submitty Forum

- **Purpose**: For instructor/TA announcements, student question/answers, offline discussion of assigned readings, etc.
- **Stay within Submitty environment** (same login, moderated by TAs/instructor)
- **Threads/posts/replies/hierarchy**
- **Multiple Image attachments/links/prose vs. code segments**
- **Current/Future work**: search, notifications (incl. email), direct message/regrade requests
Create a New Thread

Homework 1 has been posted on the course website. You can view it [url=http://example.com]here.[/url]

Hint: In this homework you will have to use a switch statement. I have included an example of one below.

```php
switch ($_REQUEST['page']) {
    case 'create_thread':
        $this->showCreateThread();
        break;
    case 'make_announcement':
        $this->makeAnnouncement();
        break;
    case 'view_thread':
        $this->viewThread();
        break;
    default:
        $this->showThreads();
        break;
}
```
Announcements, Code Segments, and Links

Hello Alyssa P (Logout)
Submitty > sample > Discussion Forum

Quiz Reminder
In addition to the homework that was just released. Please don't forget that we have a...
02/22/2018 11:44 PM

Hw 1 Released
Homework 1 has been released on the course website. You can view it here...
02/22/2018 11:42 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...
02/22/2018 11:42 PM

Hw1, switch statement not working...
I created what I believe is the correct switch statement but it seems to not be working. Can...
02/22/2018 11:33 PM

Hw 1 Released
Homework 1 has been released on the course website. You can view it here.

Reminder: make sure to use a switch statement in your code. I have provided an example below:

```javascript
public function run() {
  switch (REQUEST['page']) {
    case 'create thread':
      $this->showCreateThread();
      break;
    case 'publish thread':
      $this->publishThread();
      break;
    case 'make announcement':
      $this->alterAnnouncement();
      break;
    case 'publish post':
      $this->publishPost();
      break;
    case 'delete post':
      $this->deletePost();
      break;
    case 'remove announcement':
      $this->alterAnnouncement();
      break;
    case 'view thread':
      default:
      $this->showThreads();
      break;
  }
}
```

Quinn I 02/22/2018 11:27 PM

Enter your reply here...
Quiz Reminder

In addition to the homework that was just released, please don't forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

Jill T 02/22/2018 11:44 PM

Thank you for the reminder!!!

Anonymous 02/22/2018 11:47 PM

What sections are on the quiz again?

Alyssa P H 02/22/2018 11:56 PM

Enter your reply here...

Uploaded Attachment

Anonymous (to class)?  Reply

Quiz Reminder

In addition to the homework that was just released, please don't forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

Jill T 02/22/2018 11:44 PM

Thank you for the reminder!!!

Ben Bitiddle (bitiddle) 02/22/2018 11:47 PM

What sections are on the quiz again?

Alyssa P H 02/22/2018 11:56 PM

Enter your reply here...

Uploaded Attachment
Hello Quinn (Logout)
Submitty > sample > Discussion Forum

Quiz Reminder
In addition to the homework that was just released. Please don't forget that we have a...
02/22/2018 11:44 PM

Hw 1 Released
Homework 1 has been released on the course website. You can view it...
02/22/2018 11:27 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...
02/22/2018 11:42 PM

Hw1 switch statement not working...
I received an error message in the previous switch statement but it seems to not be functioning.
02/23/2018 01:03 PM

Hw1 switch statement not working
I created what I believe is the correct switch statement but it seems to not be working. Can someone take a look? I believe that it is in my switch condition.
1 switch(color) {

Anonymous 02/22/2018 11:33 PM

It seems like you are getting an error similar to what I have attached. Make sure to use, assuming that it has been previously instantiated:
1 scalar

Quinn I 02/23/2018 12:27 AM

Enter your reply here...

Upload Attachment

Anonymous (to class)?

Reply
Edit Posts

Hello Quinn (Logout)

Edit Posts

Submitty > sample > Discussion Forum

Editing a post by: ta on 02/22/2018 11:44 PM

In addition to the homework, please don’t forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

Homework 1 Now Posted
Homework 1 has been posted on the course website. You can access it...

02/23/2018 1:29 AM

Quiz Reminder
In addition to the homework, please don’t forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

02/23/2018 1:44 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...

02/22/2018 11:42 PM

Hw1 switch statement not working...
I created what I believe is the correct switch statement but it seems to not be working. Can...

02/22/2018 11:33 PM

What sections are on the quiz again?

02/22/2018 11:56 PM

Enter your reply here...
Flexible Late Day Policy -- Reduces Student Stress

Integrated Plagiarism Detection

Poster “Correlation of a Flexible Late Day Policy with Student Stress and Programming Assignment Plagiarism” Today 3-5pm
Autograding w/ External Tools

- JUnit, Python unittest
- Memory Debugging (Dr. Memory / Valgrind)
- Code Coverage (EclEmma, JaCoCo)
- Verification-Aware Programming Language (Dafny)
- MPI / OpenMP
- OpenGL/GLFW
- TkInter
- Matlab
- Anything you can install & run on GNU/Linux!
Static Analysis

● In use:
  ○ Count print / assignments / multiplication
  ○ Forbid use of goto / auto
  ○ Verify use of for vs. while
  ○ Verify use of dictionary

● Current work
  ○ Loop depth -- naive complexity analysis
  ○ Function calls itself -- naive recursion check
  ○ Forbid STL Vector erase
  ○ Confirm all exceptions are caught
  ○ Check that all member variables are private

● Future
  ○ Reverse engineer UML diagram -- design pattern check
Security via Jailed Sandbox

- Database access done through the PHP Data Objects (PDO) library which protects against malicious and malformed inputs
- Instructor configures appropriate resource limits (GNU Linux `rlimit`) to sandbox testing of electronically-submitted student code and prevent issues like infinite loops, runaway output, and excessive use of other system resources
- Before running the student code, we switch from a privileged system user to an untrusted user using GNU Linux `setresuid`
- Careful design of file and directory permissions and database access maintains confidentiality of student work and grades
- Uses secure computing mode (GNU Linux `seccomp`) to prevent use of sockets, fork, and other unnecessary system calls by student code

*Thanks to RPISEC (our undergraduate Computer Security Club) for helping find & patch vulnerabilities*
Docker for Autograding Isolation

Customize docker images per course, per assignment

Poster
“Analysis of Container Based vs. Jailed Sandbox Autograding Systems”
Today 3-5pm
Future Work

● Make it more mobile friendly
● Expand usage of Submitty beyond RPI
● Support SQL autograding
● PDF/Code annotation for TA grading
● Improved Docker integration/support
● Remote graders for advanced topic classes
  ○ Graphics
  ○ Parallel Computing
  ○ Distributed Systems and Algorithms
Thanks!

- All material from this demo available at
  https://submitty.org/tutorial
  https://github.com/Submitty

- New users are welcome! Ask us questions:
  submitty-admin@googlegroups.com

- New developers are welcome:
  Rensselaer Center for Open Source Software (RCOS)
  Sponsored by RedHat Software
  Google Summer of Code 2018
  To access our Slack server:
  http://submitty.org/developer/
examples/01_simple_python

{
  "testcases": [
    {
      // Student-visible testcase name.
      "title": "Python - Simple Grading",
      // Commands to run (in order). These are not shell commands, although
      // they support some common shell wildcards. This can either be a
      // list or a single string.
      "command": [ "python *.py" ],

      // Point value of this testcase.
      "points": 10,

      "validation": [
        {
          // Grade by "diffing" the student output with an
          // instructor-provided file.
          "method": "diff",
          // The student's output.
          "actual_file": "STDOUT.txt",
          // The title seen by students.
          "description": "Program Output",
          // The instructor-provided file (the correct answer).
          "expected_file": "output.txt"
        }
      ]
    }
  ]
}
// For compiled languages, typically two testcases are used to allow points
// to be assiged independently for compilation and execution.
"testcases" : [
    
    // Indicate that this is a compilation step.
    "type" : "Compilation",
    "title" : "C++ - Compilation",
    "command" : "clang++ -Wall -o a.out -- * .cpp",
    // Name of the result of compilation.
    "executable_name" : "a.out",
    // Point value of compilation.
    "points" : 5

    ],

    
    "title" : "C++ - Execution",
    "command" : "./a.out",
    // Point value of correct output.
    "points" : 15,
    "validation" : [

    "method" : "diff",
    "actual_file" : "STDOUT.txt",
    "description" : "Program Output",
    "expected_file" : "test1_output.txt"

    ]

]}
}