Software Test Report (STR)
DI-IPSC-81440
Ted Adams, Victor Konen, Corinne Eide, Hamza Saeed, Glen Taylor
Team 1

1. Scope. This section shall be divided into the following paragraphs.

1.1 Identification. This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

The software is a UMBC Computer Science Python 201 Automated Grading System. It contains a dashboard with an Administrator Panel, an Instructor Panel, and a Student Panel. Administrators can choose and remove multiple Instructors, and Instructors can choose and remove multiple students. Instructors can upload the assignments to the system where students will have three chances to submit the assignment before the deadline. The software will return a grade to the student, and give them the highest grade out of the three attempts.

1.2 System overview. This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.

The purpose of the system is to provide an environment that will make grading easier for 201 Instructors, where an administrator can add multiple instructors, instructors can add multiple students, and instructors can add multiple assignments. It will then allow students to submit their assignments to the directory where said assignments will then be graded by the software, returning the grade to the student. More specifically, this system is aimed towards students in a Computer Science 201 Python class. The Automated Grading System will not allow submissions for a specific assignment more than three times, and will not allow submissions for an assignment after the given deadline. Even though this software was created with the intention of being used only by our customer, Maksym Morawski, it was designed to keep of multiple administrators and multiple instructors, and in turn, their multiple students.

This system was designed and developed by the members of Team One of Edward Birrane's 447 Software Engineering I course. The users of the software will be Computer Science administrators, instructors, and students. There are no sponsors or support agencies involved with the development and deployment of this software.

This software has been in development since the beginning of the Spring 2016 semester and shall be completed by the end of the Spring 2016 semester.

1.3 Document overview. This paragraph shall summarize the purpose and contents of this document and shall describe any security or privacy considerations associated with its use.

This Software Test Report is used to summarize the results of testing certain aspects of Team One's Automated Grading Software. Showing the areas that Team One decided to focus on and improve with test cases will give the reader a better understanding of what the team was trying to accomplish with their testing.

2. Referenced documents. This section shall list the number, title, revision, and date of all documents referenced in this report. This section shall also identify the source for all documents not available through normal Government stocking activities.

STD, STP
http://datahole.ddns.net/cmsc447/main/documents.html

3. Overview of test results. This section shall be divided into the following paragraphs to provide an overview of test results.

3.1 Overall assessment of the software tested. This paragraph shall:

a. Provide an overall assessment of the software as demonstrated by the test results in this report

The Automated Grading Software met all of the basic requirements we set, such as Administrators being added, Administrators adding Instructors, Instructors adding Students and Assignments, and Students uploading assignments.

b. Identify any remaining deficiencies, limitations, or constraints that were detected by the testing performed. Problem/change reports may be used to provide deficiency information.

The grading algorithm for the assignments could be improved. At the moment, the software grades the student's program by feeding it a pre-selected input and checking that the output it provides in comparison to the expected output given by the instructor. If the output is the same between the student's and the instructor's, then the student passes the assignment.

c. For each remaining deficiency, limitation, or constraint, describe:

1) Its impact on software and system performance, including identification of requirements not met

The deficiency affects the software since it is not a thorough grading of the Student's program, which is the entire purpose of the software.

2) The impact on software and system design to correct it

We will attempt to change the program so that it does a more in depth analysis of the Student's program by checking specific areas of the program instead of checking the differences between the Students' and Instructors' output.

3) A recommended solution/approach for correcting it
See above.

3.2 Impact of test environment. This paragraph shall provide an assessment of the manner in which the test environment may be different from the operational environment and the effect of this difference on the test results.

The test environment is different from possible operational environments because there are factors that could affect the result of the software's functionality in its operational environment. There could be multiple students submitting their assignments at the same time, which could cause the system to slow down. A student can also submit an assignment that returns the expected results, but the student's software might not what be what the instructor asked for.

3.3 Recommended improvements. This paragraph shall provide any recommended improvements in the design, operation, or testing of the software tested. A discussion of each recommendation and its impact on the software may be provided. If no recommended improvements are provided, this paragraph shall state "None."

None.

Software Test Report (STR) (PDF version)
DI-IPSC-81440

4. Detailed test results. This section shall be divided into the following paragraphs to describe the detailed results for each test. Note: The word "test" means a related collection of test cases.

4.x (Project-unique identifier of a test). This paragraph shall identify a test by project-unique identifier and shall be divided into the following subparagraphs to describe the test results.

Reference STP 3.x.9 for the Project-Unique Identifiers.

4.x.1 Summary of test results. This paragraph shall summarize the results of the test. The summary shall include, possibly in a table, the completion status of each test case associated with the test (for example, "all results as expected," "problems encountered," "deviations

required"). When the completion status is not "as expected," this paragraph shall reference the following paragraphs for details.

STP.4.2.1.x - Passed all tests
STP.4.2.2.x - Most tests passed only 1 failed test
STP.4.2.3.x - Problems encountered
STP.4.2.4.x - Most tests passed

4.x.2 Problems encountered. This paragraph shall be divided into subparagraphs that identify each test case in which one or more problems occurred.

4.x.2.y (Project-unique identifier of a test case). This paragraph shall identify by project-unique identifier a test case in which one or more problems occurred, and shall provide:

Steps are referenced from the STD Testing procedures

STP.4.2.2.y

- **Admins were able to remove themselves, locking them out of the system**
  - **Step 6 STD 4.2.2.x**
  - **1 attempt resolved the test failure**
  - **Failure was unanticipated, database backups made beforehand were used**

STP.4.2.3.y

- **Not able to edit assignments**
  - **No official test cases (not implemented)**
- **Not able to remove assignments**
  - **Step 35 STD 4.2.3.x**
  - **3 attempts resolved the test failure**
  - **Backup data made via phpmyadmin interface**
- **Files were able to uploaded with incorrect file names (turned out to not matter)**
  - **Step 31 STD 4.2.3.x**
  - **1 attempt resolved the test failure**
  - **Backups of test data were not required**
- **File contents could get messed up in some scenarios**
  - **Step 31 STD 4.2.3.x**
  - **Approximately 10 attempts resolved the test failure**
  - **Backups of test data were not required**
- **Does not protect against malware.**
  - **No official test cases**

- **No attempt to fix (time is a thing)**
- **Backups of database are in the repository**

STP.4.2.4.y

- **The grading only gives 0 or 100**
  - **No official test cases**
  - **No attempt to fix**
  - **No backups needed**
- **See 4.2.3.3 and 4.2.3.4**

4.x.3 Deviations from test cases/procedures. This paragraph shall be divided into subparagraphs that identify each test case in which deviations from test case/test procedures occurred.

There are no deviations from the test cases.

4.x.3.y (Project-unique identifier of a test case). This paragraph shall identify by project-unique identifier a test case in which one or more deviations occurred, and shall provide:

Software Test Report (STR) (PDF version)
DI-IPSC-81440

5. Test log. This section shall present, possibly in a figure or appendix, a chronological record of the test events covered by this report. This test log shall include:

a. The date(s), time(s), and location(s) of the tests performed

4.2.1.x - Tests performed on April 19th, at 9:00 AM
4.2.2.x - Tests performed on April 19th, at 11:00 AM
4.2.3.x - Tests performed on April 19th, at 1:00 PM
4.2.4.x - Tests performed on April 19th, at 3:00 PM

b. The hardware and software configurations used for each test including, as applicable, part/model/serial number, manufacturer, revision level, and calibration date of all hardware, and version number and name for the software components used

The hardware being used during testing was a *Raspberry Pi 2 Model B+ running Raspbian. The software being used during testing was Google Oauth API and PHPMyAdmin.*

c. The date and time of each test-related activity, the identity of the individual(s) who performed the activity, and the identities of witnesses, as applicable.

.**All of the members of Team One were involved with each test-related activity. The members performed the testing together during a Google Hangouts call.**

6. Notes. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

N/A

A. Appendixes. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

N/A