Software Test Description (STD)

Ted Adams, Victor Konen, Corinne Eide, Hamza Saeed, Glen Taylor

Team 1

DI-IPSC-81439

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 to which this document applies is a UMBC Computer Science 201 Python Automated Grading System. The system has a Dashboard, a Admin Panel, and Instructor Panel, and a Student Panel. An admin will be able to add in instructors and students into the system. These Instructors can add in courses and add students to that particular course. The courses will have multiple assignments created by the instructor. The students can log into the system and submit their code assignments. The system, in return, will auto grade that assignment and give back a score out of the total points to the student. The student gets 3 attempts to each assignment.**

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 make things easier for the graders and teachers in CMSC-201 classes. This software will help by auto grading all the python code assignments in the course. This document applies to the Auto Grade System developed by Team 1. The software project started 2 months ago and the customer is Prof Max. The current operating and testing site is datahole.ddns.net and there are no planned operating sites currently.**

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.

**The purpose of this document is to create a Software Test Plan for Auto Grade System developed by Team 1.**

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

**STP, Usecases**

*http://datahole.ddns.net/cmsc447/main/documents.html*

3. Test preparations. This section shall be divided into the following paragraphs. Safety precautions, marked by WARNING or CAUTION, and security and privacy considerations shall be included as applicable.

**N/A**

3.x (Project-unique identifier of a test). This paragraph shall identify a test by project-unique identifier, shall provide a brief description, and shall be divided into the following subparagraphs. When the information required duplicates information previously specified for another test, that information may be referenced rather than repeated.

3.x.1 Hardware preparation. This paragraph shall describe the procedures necessary to prepare the hardware for the test. Reference may be made to published operating manuals for these procedures. The following shall be provided, as applicable:

**There is no hardware preparation that is not already done**

Software Test Description (STD) (PDF version)
DI-IPSC-81439

3.x.2 Software preparation. This paragraph shall describe the procedures necessary to prepare the item(s) under test and any related software, including data, for the test. Reference may be made to published software manuals for these procedures. The following information shall be provided, as applicable:

1. The specific software to be used in the test
    - **Google Oauth API**
    - **PHPMyAdmin**
2. The storage medium of the item(s) under test (e.g., magnetic tape, diskette)
    - **HDD of the Raspberry pi server**
3. The storage medium of any related software (e.g., simulators, test drivers, databases)
    - **The JSON string returned by Oauth is stored in our database**

1. Instructions for loading the software, including required sequence
    - **A readme.txt will be included with the software, outlining all changes that will need to be made to the software in order to move it to a different server**
    - **The readme.txt will describe changes that must be made to "config.php", a configuration file that determines pathing for .php and .html files used to generate the site, as well as the credentials for the Google Oauth API**

1. Instructions for software initialization common to more than one test case
    - **N/A**

3.x.3 Other pre-test preparations. This paragraph shall describe any other pre-test personnel actions, preparations, or procedures necessary to perform the test.

**There are no other pre test operations not already detailed in the STD or STP**

4. Test descriptions. This section shall be divided into the following paragraphs. Safety precautions, marked by WARNING or CAUTION, and security and privacy considerations shall be included as applicable.

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. When the required information duplicates information previously provided, that information may be referenced rather than repeated.

4.x.y (Project-unique identifier of a test case). This paragraph shall identify a test case by project-unique identifier, state its purpose, and provide a brief description. The following subparagraphs shall provide a detailed description of the test case.

**STP.4.2.1 - test google login**
**STP.4.2.2 - test admin panel**
**STP.4.2.3 - test instructor panel**
**STP.4.2.4 - test student panel and grading functionality**

4.x.y.1 Requirements addressed. This paragraph shall identify the CSCI or system require-ments addressed by the test case. (Alternatively, this information may be provided in 5.a.)

*STP.4.2.x* **All map to requirements laid out in the Use Cases document**

4.x.y.2 Prerequisite conditions. This paragraph shall identify any prerequisite conditions that must be established prior to performing the test case. The following considerations shall be discussed, as applicable:

**N/A - there are no prerequisite conditions that need to be met to start testing that have not already been fulfilled.**

Software Test Description (STD) (PDF version)
DI-IPSC-81439

4.x.y.3 Test inputs. This paragraph shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

*STP.4.2.1.x*

1. Name, purpose, and description (e.g., range of values, accuracy) of each test input
   **Verifying login and logout procedures**

1. Source of the test input and the method to be used for selecting the test input
   **Source of the test input is from the user and the method is manual**

1. Whether the test input is real or simulated
   **Real input**

1. Time or event sequence of test input
   - **Click login --> check to see if login was successful**
   - **Hit logout --> check success**

1. The manner in which the input data will be controlled to:
   **N/A**

---

*STP.4.2.2.x*

2. Name, purpose, and description (e.g., range of values, accuracy) of each test input
   **Testing the admin panel's functionality**

2. Source of the test input and the method to be used for selecting the test input
   **Source of the test input is from the user and the method is manual**

2. Whether the test input is real or simulated
   **Real input**

2. Time or event sequence of test input
   - **Test adding a proper name and email to the database**
     - **Check null values for each field**
     - **Check all null values**
   - **Test adding a class**
     - **Test null values in all fields**

2. The manner in which the input data will be controlled to:
   *N/A*

---

*STP.4.2.3.x*

3. Name, purpose, and description (e.g., range of values, accuracy) of each test input

**Testing the Instructor panel's functionality**

3. Source of the test input and the method to be used for selecting the test input
   **Source of the test input is from the user and the method is manual**

3. Whether the test input is real or simulated
   **Real input**

3. Time or event sequence of test input
   - **An instructor begins at the dashboard, and will be given the option to click on "Manage Classes"**
   - **Afterwards, they are shown a list of classes they belong to at the privilege level of instructor**
   - **On this page instructors have 3 options: View one of their classes, remove a class, or add a class**
     - **If they view their class by selecting "Manage" in the row belonging to the desired class: they can manage students, add assignments, view submissions per assignment, or remove an assignment**
       - **If they decide to manage students, they can add students by adding an email into a textbox, or remove a student by selecting "Remove" on the row belonging to that student**
       - **If they decide to add an assignment, they will need to enter the name of the assignment, the submission limit, upload a .pdf file containing a description of the assignment, and a .py file with correct output for the assignment. Then, they can click "Add Assignment" and it will be added to the list of assignments for the class**
       - **If they decide to view assignment submissions, they are given a list of submissions made by their students for that assignment - it will display the files, which the instructors can click on to view, and it will display the grade received per submission**
       - **If they decide to remove an assignment, they click on "Remove" for the row pertinent to that assignment and the assignment will be deleted from the list**
     - **To remove a class, they click on "Remove" for the row pertaining to that class**
     - **To add a class, they click "Add Classes", enter a Class name and class descript, then click "Add Class" which will add the class to the list**
   - **Test adding a proper name and email to the students database**
   - **Check null values for each field**
   - **Check all null values**

3. The manner in which the input data will be controlled to:
   **N/A**

---

*STP.4.2.4.x*

4. Name, purpose, and description (e.g., range of values, accuracy) of each test input
   **Testing the Student panel's functionality**

4. Source of the test input and the method to be used for selecting the test input
   **Source of the test input is from the user and the method is manual**

4. Whether the test input is real or simulated
   ***Real input***

4. Time or event sequence of test input
   - **Student clicks on "View Classes" from dashboard**
   - **Student selects a class from list of classers they belong to**
   - **Student selects an assignment that belongs to the class**
   - **Student can assignments under "Previous Uploads", which will show their individual submissions and the grades received per submission**
   - **Student can have an assignment graded from under "Previous Uploads" if they have not graded that submission yet.**
   - **Student can select a file to upload, and then upload it using the button "Upload Attempt"**
   - **If the student does not upload a python file, an error will display informing them the Program file must be of type .py**
   - **Once an attempt is uploaded and the student clicks on "(Grade)" for that submission, the grade is received 1-2 seconds after the action**

4. The manner in which the input data will be controlled to:
   **N/A**

---

1) Test the item(s) with a minimum/reasonable number of data types and values

2) Exercise the item(s) with a range of valid data types and values that test for overload, saturation, and other "worst case" effects

3) Exercise the item(s) with invalid data types and values to test for appropriate handling of irregular inputs

4) Permit retesting, if necessary

4.x.y.4 Expected test results. This paragraph shall identify all expected test results for the test case. Both intermediate and final test results shall be provided, as applicable.

**STP.4.2.1.x**

**STP.4.2.1.1 - User is taken to the dashboard and displays their privilege level**

**STP.4.2.1.2 - Use is taken back to the login page**

**STP4.2.2.x**

**STP.4.2.2.1 - Instructor can be added to the database**

**- Instructor can not be added twice to the database**

**- Instructors can be removed from the database**

**STP.4.2.2.2 - Same tests as STP.4.2.2.1**

**STP.4.2.2.3 - Same as STP.4.2.2.2**

**- The system should still function normally with 1000 added students**

**STP.4.2.3.x**

**STP.4.2.3.1 - Assignment should be able to be added properly with all parameters**

**STP.4.2.3.2 - Student added successfully with no duplicate students**

**STP.4.2.3.3 - Instructor should be able to see all grades in their class**

**STP.4.2.4.x**

**STP.4.2.4.1 - Student can view individual assignment grades**

**- Student can upload assignments and view them**

**- Student's file will be rejected if it has the wrong file extension**

**- Student will use one of their tries and the assignment will be regraded**

4.x.y.5 Criteria for evaluating results. This paragraph shall identify the criteria to be used for evaluating the intermediate and final results of the test case. For each test result, the following information shall be provided, as applicable:

1. The range or accuracy over which an output can vary and still be acceptable

   **All tests from STP.4.2.x.y are pass/fail**

1. Minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result

1. Maximum/minimum allowable test duration, in terms of time or number of events

   **STP.4.2.1.x - 1 for each**

   **STP.4.2.2.x - Minimum of 2 per and maximum of 7 (all tests are symmetric), events are clicks or inputs**

   **STP.4.2.3.1 - Minimum or 2 events max of 11 events**

   **STP.4.2.3.2 - Minimum of 2 events and a max of 7 events**

   **STP.4.2.3.3 - Three events always**

   **STP.4.2.4.1 - Minimum of 3 events max of around 10 events**

1. Maximum number of interrupts, halts, or other system breaks that may occur
   **None**
1. Allowable severity of processing errors
   **ALL - Inputs are not time sensitive**
   **STP.4.2.1.x - No room for error in login process and logout**
   **STP.4.2.2.x - Medium allowance for error, admins can reverse any change so it can be fixed promptly**
   **STP.4.2.3.x - Medium, instructors can fix any errors that occur with the database within their domain**
   **STP.4.2.4.x - Low tolerance since students cannot modify the database directly**
1. Conditions under which the result is inconclusive and re-testing is to be performed
   ***N/A all testing will be conclusive as all testing is pass/fail***
1. Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database/data files, or in test procedures
   **Failure in the pass/fail tests will indicate and irregularity**
1. Allowable indications of the control, status, and results of the test and the readiness for the next test case (may be output of auxiliary test software)

1. Additional criteria not mentioned above.
   **N/A**

Software Test Description (STD) (PDF version)
DI-IPSC-81439

4.x.y.6 Test procedure. This paragraph shall define the test procedure for the test case. The test procedure shall be defined as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed. For convenience in document maintenance, the test procedures may be included as an appendix and referenced in this paragraph. The appropriate level of detail in each test procedure depends on the type of software being tested. For some software, each keystroke may be a separate test procedure step; for most software, each step may include a logically related series of keystrokes or other actions. The appropriate level of detail is the level at which it is useful to specify expected results and compare them to actual results. The following shall be provided for each test procedure, as applicable:

1. Test operator actions and equipment operation required for each step, including commands, as applicable, to:

1) Initiate the test case and apply test inputs

2) Inspect test conditions

3) Perform interim evaluations of test results

4) Record data

5) Halt or interrupt the test case

6) Request data dumps or other aids, if needed

7) Modify the database/data files

8) Repeat the test case if unsuccessful

9) Apply alternate modes as required by the test case

10) Terminate the test case

1. Expected result and evaluation criteria for each step

1. If the test case addresses multiple requirements, identification of which test procedure step(s) address which requirements. (Alternatively, this information may be provided in 5.)

1. Actions to follow in the event of a program stop or indicated error, such as:

1) Recording of critical data from indicators for reference purposes

2) Halting or pausing time-sensitive test-support software and test apparatus

3) Collection of system and operator records of test results

1. Procedures to be used to reduce and analyze test results to accomplish the following, as applicable:

1) Detect whether an output has been produced

2) Identify media and location of data produced by the test case

3) Evaluate output as a basis for continuation of test sequence

4) Evaluate test output against required output

**STP.4.2.1.x -**

1. **go to website**
2. **click login**
3. **enter credentials**
4. **check to see if your dashboard pops up**

5. **check dashboard credentials**
6. **click logout**
7. **check to see if you are not signed in anymore**

**STP.4.2.2.x -**

1. **Add an initial administrator via phpmyadmin following guidelines in the readme**
2. **Select manage users**
3. **Ensure management buttons for admins, instructors, and users display**
4. **Select manage admins**
5. **Ensure own account displays**
6. **Select to remove on self to ensure this action is blocked**
7. **Select add administrators**
8. **Select add administrators with no input to ensure validation is working**
9. **Provide input that is not a valid email address**
10. **Provide several valid email addresses that are associated with google+ accounts (arbitrary)**
11. **Select add administrators**
12. **Ensure the page redirects back to manage administrators**
13. **Select edit on an administrator**
14. **Ensure all data fields that are not primary key or time based are editable and required by supplying variable input**
15. **Select save**
16. **Ensure edited information has been updated correctly for user**
17. **Login as an added user to ensure privileges are correct**
18. **Repeat these procedures for instructors and students and ensure users can be added as all three types of users (modularity)**

**STP.4.2.3.x -**

1. **Select manage classes**
2. **Select add classes**
3. **Select add class with no input to ensure validation is working**
4. **Provide a course name and description (arbitrary)**
5. **Select add class**
6. **Ensure page redirects back to manage class**
7. **Ensure class that has just been added displays**
8. **Select remove class**
9. **Ensure page redirects back to manage class**

10. **Ensure class that has just been removed no longer displays**
11. **Follow procedure to add another class**
12. **Select manage class**
13. **Select manage students**
14. **Ensure no students are displayed**
15. **Select add students**
16. **Select add students with no input to ensure validation is working**
17. **Provide input that is not a valid email address**
18. **Provide several valid email addresses that are associated with google+ accounts (arbitrary)**
19. **Select add students**
20. **Ensure the page redirects back to manage students**
21. **Ensure the students that were just added are displayed**
22. **Select remove on a student**
23. **Ensure the page redirects back to manage students**
24. **Ensure the student that was just removed is no longer displayed**
25. **Select dashboard**
26. **Select manage classes**
27. **Select manage on the class**
28. **Select add assignments**
29. **Select add assignment**
30. **Select add assignment with no input to ensure validation is working**
31. **Provide inputs to ensure validation**
    a. **Non pdf file to pdf upload field**
    b. **Non py file to py upload field**
    c. **Excessively large files sizes (> 10MB pdf | > 30KB py)**
    d. **Negative submission limit (should default to 3)**
32. **Provide valid inputs and select add assignment**
33. **Ensure the page redirects to edit class**
34. **Ensure the class that was just added is displayed correctly**
35. **Test assignment removal analogous to class removal and re-add the assignment**
36. **Proceed to student testing of assignment submission and grading**
37. **Afterwards select submissions on the assignment**
38. **Ensure student submissions and grades are displayed for the assignment**

**STP.4.2.4.x**

1. **Upload assignment as student**

2. **Upload an assignment with a bad file extension**
3. **Attempt to grade assignment**
4. **Review grade**
5. **Attempt to get assignment regraded**
6. **Try to get re-graded too many times**

4.x.y.7 Assumptions and constraints. This paragraph shall identify any assumptions made and constraints or limitations imposed in the description of the test case due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and database/data files. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this paragraph shall address their effects and impacts upon the test case.

**We assume all email addresses are associated with google and that all python files submitted will terminate**

5. Requirements traceability. This paragraph shall contain:

a. Traceability from each test case in this STD to the system or CSCI requirements it addresses. If a test case addresses multiple requirements, traceability from each set of test procedure steps to the requirement(s) addressed. (Alternatively, this traceability may be provided in 4.x.y.1.)

**Documented in 4.x.y.1**

b. Traceability from each system or CSCI requirement covered by this STD to the test case(s) that address it. For CSCI testing, traceability from each CSCI requirement in the CSCI's Software Requirements Specification (SRS) and associated Interface Requirements Specifications (IRSs). For system testing, traceability from each system requirement in the system's System/Subsystem Specification (SSS) and associated IRSs. If a test case addresses multiple requirements, the traceability shall indicate the particular test procedure steps that address each requirement.

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**