Software Testing Plan (STP)
for the
Computer Science Question Input and Delivery Template Manager (C-SQID-TM)

Document Number: CMSC447-FA2020-G02-STP-01B
Revision B

12 December 2020

Authors:
Ryan Appleby
Charles Varga
Aleya Mayo
Amrithya Balasubramanian
Chris DeVoe
Danada Amalage Don

Version History

Date	Version	Description	Authors
11/12/2020	1.0	Initial Version for Submission	Team
12/12/2020	1.1	Updated CST_F_5 traceability	Ryan Appleby

1. Scope 5
1.1 Identification. 5
1.2 System overview. 5
1.3 Document overview. 6
1.4 Relationship to other plans. 6
2. Referenced Documents 6
3. Software test environment. 7
3.1 Microsoft Edge Testing Environment 7
3.1.1 Software Items 7
3.1.2 Hardware and Firmware Items 7
3.1.3 Other Materials 7
3.1.4 Proprietary nature, acquirer’s rights, and licensing. 8
3.1.5 Installation, testing, and control. 8
3.1.6 Participating organizations. 8
3.1.7 Personnel. 8
3.1.8 Orientation plan. 8
3.1.9 Tests to be performed. 8
3.2 Google Chrome Testing Environment 8
3.2.1 Software Items 8
3.2.2 Hardware and Firmware Items 9
3.2.3 Other Materials 9
3.2.4 Proprietary nature, acquirer’s rights, and licensing. 9
3.2.5 Installation, testing, and control. 9
3.2.6 Participating organizations. 9
3.2.7 Personnel. 9
3.2.8 Orientation plan. 9
3.2.9 Tests to be performed. 9
3.3 Apple Safari Testing Environment 10
3.3.1 Software Items 10
3.3.2 Hardware and Firmware Items 10
3.3.3 Other Materials 10
3.3.4 Proprietary nature, acquirer’s rights, and licensing. 10
3.3.5 Installation, testing, and control. 10
3.3.6 Participating organizations. 10
3.3.7 Personnel. 11
3.3.8 Orientation plan. 11
3.3.9 Tests to be performed. 11
3.4 Mozilla Firefox Testing Environment 11
3.4.1 Software Items 11

3.4.2 Hardware and Firmware Items 11
3.4.3 Other Materials 11
3.4.4 Proprietary nature, acquirer's rights, and licensing. 12
3.4.5 Installation, testing, and control. 12
3.4.6 Participating organizations. 12
3.4.7 Personnel. 12
3.4.8 Orientation plan. 12
3.4.9 Tests to be performed. 12

4. Test identification 12

4.1 General Information 12
4.1.1 Test levels. 12
4.1.2 Test classes. 13
4.1.3 General test conditions. 13
4.1.4 Test progression. 13
4.1.5 Data recording, reduction, and analysis. 13
4.2 Planned tests. 14
4.2.1 Security 14
4.2.1.1 CST-Sec-1 14
4.2.1.2 CST-Sec-2 14
4.2.2 Connectivity 14
4.2.2.1 CST-Con-1 14
4.2.2.2 CST-Con-2 14
4.2.3 Uploading 14
4.2.3.1 CST-Upl-1 15
4.2.3.2 CST-Upl-2 15
4.2.4 Retrieving 15
4.2.4.1 CST-Ret-1 15
4.2.4.2 CST-Ret-2 15
4.2.5 Modify 15

5. Test schedules. 16

6. Requirements traceability. 16

1. Scope

1.1 Identification.

This Software Testing Plan pertains to the development of the Computer Science Question Input and Delivery Template Manager (C-SQID-TM) application and backend question database powering said application. This document contains information on the testing procedures for the software. The intended audience includes the engineers tasked with developing the application and its database as well as the test conductors responsible for the verification and acceptance of the application.

1.2 System overview.

The C-SQID-TM application can interface with the backend C-SQID-TM database to store, retrieve, and update the information within the database. The user can manually enter information to both store and retrieve information. The user may also submit a local document to be parsed and stored in the database. In addition, the application allows the user to input values into retrieved questions and save said questions locally without storing them in the database, and will allow the user to export this output to a separate file. The primary user for this application would be Computer Science professors and teaching assistants.

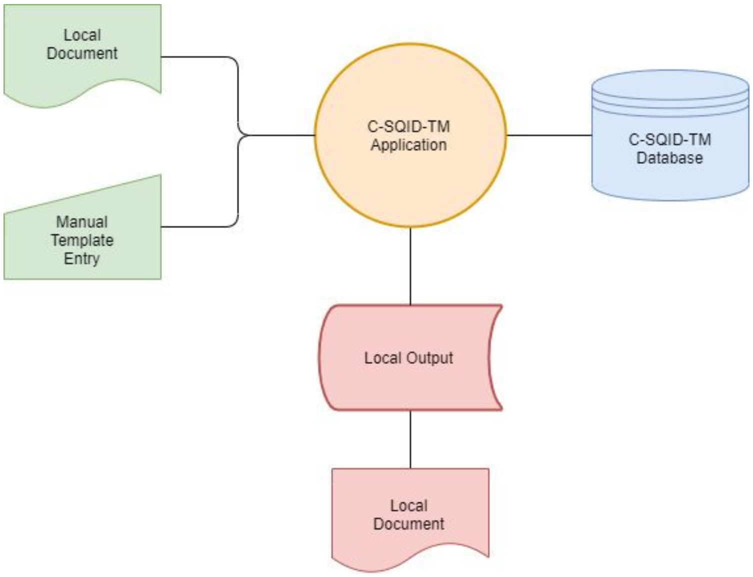

Figure 1. C-SQID-TM Interface

The software outlined in this STP is responsible for interfacing with the user's local machine as well as the C-SQID-TM database in order to store and retrieve information. The software's overall state transition is shown in Figure 2.

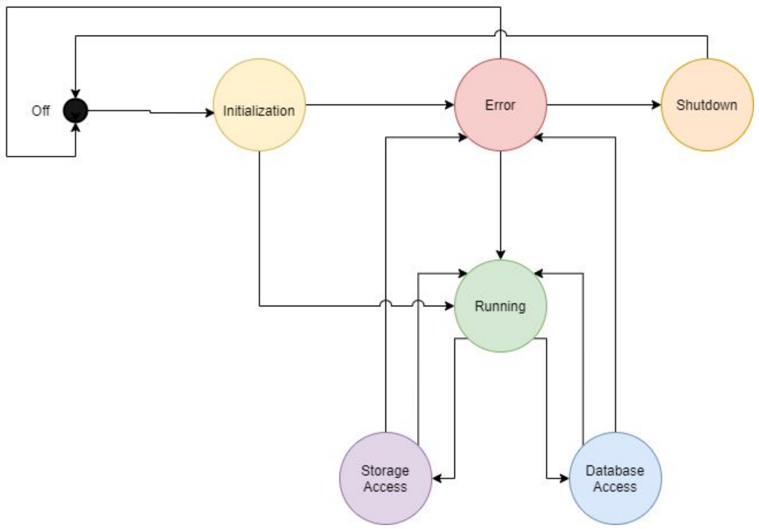

Figure 2. C-SQID-TM State Diagram

1.3 Document overview.

Section 1 describes the scope of the project. Section 2 provides the documents references. Section 3 details our planned testing environment, while Section 4 details the tests themselves. Section 5 will provide a brief test schedule. Section 6 will provide a traceability matrix, mapping each test to one of our requirements.

1.4 Relationship to other plans.

This STP relates to other software development plans for the C-SQID-TM software. These plans include a Software Development Proposal, Software Requirements Specification, and Software Design Document. These documents can be found in the Documents folder of our GitHub repository, located here:
https://github.com/rappleb1/software-development-447-project/tree/master/Documents

2. Referenced Documents

MIL-STD-498	Military Standard Software Development and Documentation
CMSC447-FA2020-G02-SDP-01A	Computer Science Question Input Template and Delivery Manager - Software Development Proposal
CMSC447-FA2020-G02-SRS-01B	Computer Science Question Input Template and Delivery Manager - Software Requirements Specification
CMSC447-FA2020-G02-SDD-01B	Computer Science Question Input Template and Delivery Manager - Software Design Document

3. Software test environment.

3.1 Microsoft Edge Testing Environment

3.1.1 Software Items

The software items that would be necessary to perform testing in this test environment will be the browser application Microsoft Edge. The team will also be using the most current version of that software item on a Windows 10 operating system.

The current version of Microsoft Edge is marked as 86.0.622.63 as of 2020-11-06. The Windows 10 operating system will be at the current version which is marked as 19042.610.

The purpose of using Microsoft Edge in this testing environment is to confirm that the web application can work properly on the browser application. By using Windows 10 as the operating system for these tests, it will help to focus on the audience who would most be using the browser application on this operating system.

3.1.2 Hardware and Firmware Items

Extensive hardware and ground-rooted firmware are not a part of this test plan. No interaction with the web application shall occur outside of the Flask web framework and the front end web browser interface. As such no additional hardware is required. The only software required is the Microsoft Edge web browser and IDE related debugging tools and extensions necessary for testing procedures.

3.1.3 Other Materials

Any documentation including manuals on the software items will not be provided to the team members, so it will be up to the project's team members to locate and use any documentation on the software items as needed.

3.1.4 Proprietary nature, acquirer’s rights, and licensing.

There are no assumed rights or licensings for this test plan.

3.1.5 Installation, testing, and control.

The team members are responsible for downloading and installing any of the software items mentioned above as well as confirming that the software items are up to date for testing.

3.1.6 Participating organizations.

The projects software development team is responsible for developing the application which will be tested by the projects testing team.

3.1.7 Personnel.

The personnel for this environment will be the members of the team who are responsible for testing the project. As this project is taking place remotely it will be up to the members of the testing team to have testing done on time and based on the schedule. The Project Manager will be responsible for keeping the testing team on track.

3.1.8 Orientation plan.

All team members are expected to learn and understand the software items on their own as there will be no orientation or training provided.

3.1.9 Tests to be performed.

All of the tests described in this STP can be performed in all of the four testing environments.

3.2 Google Chrome Testing Environment

3.2.1 Software Items

The software items that would be necessary to perform testing in this test environment will be the browser application Google Chrome. The team will be using the most current version of that software item on a Windows 10 operating system.

The current version of Google Chrome is marked as 86.0.4240.193 as of 2020-11-10. The Windows 10 operating system will be at the current version which is marked as 19042.610.

The purpose of using Google Chrome in this testing environment is to confirm that the web application can work properly on the browser application. By using Windows 10 as the operating system for these tests it will help to focus on the audience who would most be using the browser application on this operating system.

3.2.2 Hardware and Firmware Items

Extensive hardware and ground-rooted firmware are not a part of this test plan. No interaction with the web application shall occur outside of the Flask web framework and the front end web browser interface. As such no additional hardware is required. The only software required is the Google Chrome web browser and IDE related debugging tools and extensions necessary for testing procedures.

3.2.3 Other Materials

Any documentation including manuals on the software items will not be provided to the team members, so it will be up to the project's team members to locate and use any documentation on the software items as needed.

3.2.4 Proprietary nature, acquirer’s rights, and licensing.

There are no assumed rights or licensings for this test plan.

3.2.5 Installation, testing, and control.

The team members are responsible for downloading and installing any of the software items mentioned above as well as confirming that the software items are up to date for testing.

3.2.6 Participating organizations.

The projects software development team is responsible for developing the application which will be tested by the projects testing team.

3.2.7 Personnel.

The personnel for this environment will be the members of the team who are responsible for testing the project. As this project is taking place remotely it will be up to the members of the testing team to have testing done on time and based on the schedule. The Project Manager will be responsible for keeping the testing team on track.

3.2.8 Orientation plan.

All team members are expected to learn and understand the software items on their own as there will be no orientation or training provided.

3.2.9 Tests to be performed.

All of the tests described in this STP can be performed in all of the four testing environments.

3.3 Apple Safari Testing Environment

3.3.1 Software Items

The software items that would be necessary to perform testing in this test environment will be the browser application Safari. The team will be using the most current version of that software item on a macOS.

The current version of Safari is marked as version 14 as of 2020-10-22. The macOS will be at the current version which is marked as Catalina version 10.15.7.

The purpose of using Safari in this testing environment is to confirm that the web application can work properly on the browser application. By using macOS as the operating system for these tests it will help to focus on the audience who would most be using the browser application on this operating system.

3.3.2 Hardware and Firmware Items

Extensive hardware and ground-rooted firmware are not a part of this test plan. No interaction with the web application shall occur outside of the Flask web framework and the front end web browser interface. As such no additional hardware is required. The only software required is the Apple Safari web browser and IDE related debugging tools and extensions necessary for testing procedures.

3.3.3 Other Materials

Materials such as manuals, software listings are not assigned by the team to conduct testing at the test site. Teammates have the option to choose necessary software items to conduct appropriate testing of the test site.

3.3.4 Proprietary nature, acquirer’s rights, and licensing.

There are no assumed rights or licensings for this test plan.

3.3.5 Installation, testing, and control.

The team members are responsible for installing necessary software items for the testing environment. The items should be up to date and should correctly function on the testing environment.

3.3.6 Participating organizations.

The projects software development team is responsible for developing the application which will be tested by the projects testing team.

3.3.7 Personnel.

Personnel who are tasked with testing the software project are responsible for working in the test site. Testing done by the software testing team will be done in a timely manner and based on the schedule. The project manager shall check the status of the testing phase.

3.3.8 Orientation plan.

There will be no orientation or training given prior or during the testing. Team members are expected to know and understand how to conduct testing on their own.

3.3.9 Tests to be performed.

All of the tests described in this STP can be performed in all of the four testing environments.

3.4 Mozilla Firefox Testing Environment

3.4.1 Software Items

The software items that would be necessary to perform testing in this test environment will be the browser application Mozilla Firefox. The team will be using the most current version of that software item on a Linux operating system through a Virtual Machine (VM) environment.

The current version of Mozilla Firefox is marked as 82.0.3 as of 2020-11-10. The Linux operating system will be at the current version which is marked as 5.9.8. The Virtual Machine that the team will use to run linux will be Oracle VM VirtualBox, of which version 6.1.16 is the most recent version.

The purpose of using Mozilla Firefox in this testing environment is to confirm that the web application can work properly on the browser application. By using Linux as the operating system for these tests it will help to focus on the audience who would most be using the browser application on this operating system.

3.4.2 Hardware and Firmware Items

Extensive hardware and ground-rooted firmware are not a part of this test plan. No interaction with the web application shall occur outside of the Flask web framework and the front end web browser interface. As such no additional hardware is required. The only software required is the Mozilla Firefox web browser, IDE related debugging tools and extensions necessary for testing procedures, and VirtualBox.

3.4.3 Other Materials

Any documentation including manuals on the software items will not be provided to the team members so it will be up to the project's team members to locate and use any documentation on the software items as needed.

3.4.4 Proprietary nature, acquirer’s rights, and licensing.

There are no assumed rights or licensings for this test plan.

3.4.5 Installation, testing, and control.

The software team is responsible for downloading and installing any of the software items mentioned above as well as confirming that the software items are up to date for testing.

3.4.6 Participating organizations.

The projects software development team is responsible for developing the application which will be tested by the projects testing team.

3.4.7 Personnel.

The personnel for this environment will be the members of the team who are responsible for testing the project. As this project is taking place remotely it will be up to the members of the testing team to have testing done on time and based on the schedule. The Project Manager will be responsible for keeping the testing team on track.

3.4.8 Orientation plan.

The test conductors will need to familiarize themselves with the Mozilla Firefox environment and navigate a Linux Virtual Machine through VirtualBox. The test conductors may also need to install necessary supplementary extensions and software in order to use this web browsing platform as the testing site for the web application.

3.4.9 Tests to be performed.

All of the tests described in this STP can be performed in all of the four testing environments.

4. Test identification

4.1 General Information

4.1.1 Test levels.

The application shall be tested within different segments of the design, with gradual integrations for additional testing. For example, the database shall be tested independently initially, then the web frontend shall be integrated to the database testing by attempting to add questions to the database from the web frontend. In addition, any backend scripts will be tested independently before integrating the web frontend into an interconnected test.

4.1.2 Test classes.

This subsection shall describe the classes of tests that shall be performed.

The tests that will be performed on the application correspond to the requirements defined in the Software Requirements Specification, and shall aim to cover the following matters:

- Question retrieval - The application will be tested for its ability to store question templates in a manner that makes them retrievable by professors utilizing the web frontend.
- Question upload - The application will be tested for its capability to allow users to upload questions through multiple methods.
- Connectivity - The web frontend will be tested for its ability to connect to and communicate with the database.
- Security - The application shall be tested for immunity to user inputs designed to exploit systems that are vulnerable to injection attacks.

4.1.3 General test conditions.

Some general testing conditions include the following:

- Each test shall be performed from development team members' personal machines to ensure cross-platform accessibility of the application.
- Each test shall include minimum and maximum quantities of values or characters, if applicable.
- Each test shall not affect other systems connected to the same network as that of the application.

4.1.4 Test progression.

The progression of the testing stages are organized in order of testing as follows: front-end display/responsiveness, backend database functionality, and middle-end flask framework robustness. These three testing premises will be incrementally developed and tested in order of lowest to highest priority.

4.1.5 Data recording, reduction, and analysis.

Frontend testing will incorporate little to no numerical or tangible data recordings. Screenshots of the front end display being run across different browsers will be taken as data items to analyze the robustness of the display with respect to varying web browsing platforms. The main metric used to test and analyze the flask framework will be the time taken to deliver the different html pages in response to user interaction with the frontend html objects. Efficiency of file parsing from user input to regular expression format within the flask framework will also be tested by acquiring exact runtime measurements. Database connectivity measurements such as query response times and packet size measurements will be taken in order to leverage the quality of backend functionality and ensure that no data is being lost in the database sessions.

4.2 Planned tests.

The subsystems of the software application will be tested. These tests include security of the login, functionality of a database that stores template files, and the main functions that occur within the application, such as retrieval, storage, modification and upload of template files, as well as test-taking features.

4.2.1 Security

Testing should be conducted to ensure that no user with wrong credentials should be allowed access to the database.

4.2.1.1 CST-Sec-1

There will be low level testing to analyze if a token from the user account is sent to the authorized_users table of the CST database and token is verified correctly. This test is essential for the security requirements of the application.

4.2.1.2 CST-Sec-2

There will be high level testing with wrong and correct credential user accounts to see if the login process is functioning properly. This test is essential for the security requirements of the application.

4.2.2 Connectivity

There will be testing to make sure database that interact with user to store, and retrieve template files is working well and is capable of performing main functions

4.2.2.1 CST-Con-1

There should be testing done to test if the database that contains template files properly links with the application. This test is essential for the functional and performance requirements of the application.

4.2.2.2 CST-Con-2

The database will be inspected and observed to ensure that all required data is properly formatted and stored. Storing this information properly is essential for the functional and performance requirements of the application.

4.2.3 Uploading

There needs to be testing conducted on the functionality of the uploading of template items from the local machine of the user.

4.2.3.1 CST-Upl-1

Testing shall be conducted to make sure the user interface in the upload page of the application is functioning properly. Also make sure the side navigation panel is functioning properly. This test is essential for the functional and performance requirements of the application.

4.2.3.2 CST-Upl-2

Testing shall be done to make sure that multiple methods that can be used to upload template files are functioning properly. Each method should be tested individually to make sure no error exists in any method. This test is essential for functional requirements of the application.

4.2.4 Retrieving

There should be testing done on the retrieving function of the application where users can retrieve template files from the database.

4.2.4.1 CST-Ret-1

The retrieve page, rather the user interface should be tested to make sure each component such as side nav works properly. This test is essential for the functional and performance requirements of the application.

4.2.4.2 CST-Ret-2

Testing shall be done to make sure if the correct template file is retrieved by the database. There should be testing of whether there is a connection to the template file database, and retrieve functionality should also be analyzed to make sure if the application does retrieve the correct template file. This test is essential for the functional requirements of the application.

4.2.4.3 CST-Ret-3

Testing shall be done to make sure that retrieved question templates can be filled out correctly and exported to a user’s device. Filling out these templates with data within the web interface is a core part of the application’s functionality. This test is essential for the functional requirements of this application.

4.2.5 Modify

Testing shall be performed on the modify function of the application, where users can modify the template files.

4.2.5.1 CST-Mod-1

Testing should be done on the modify page of the application especially the user interface and make sure all components are functioning as intended. This test is essential for functional requirements of the application.

4.2.5.2 CST-Mod-2

Testing shall be done on the fnctionality of the modification of the template files. There needs to be testing to see if once the user makes any modifications to the template file, those modifications are saved and can be stored in the database. This test is essential for functional requirements of the application.

5. Test schedules.

The general testing schedule is provided in the table below.

Requirement	Site	Week of 11/10	Week of 11/16	Week of 11/23	Week of 11/30	Responsible
Orientation	All	X	X			Team
Conduct Tests	Edge		X	X	X	Edge Test Team
Conduct Tests	Firefox		X	X	X	Linux Test Team
Conduct Tests	Chrome		X	X	X	Chrome Test Team
Conduct Tests	Safari		X	X	X	Safari Test Team
Prepare Draft STR				X		Team
Finalize STR					X	Team

6. Requirements traceability.

Requirement	**Description**	**Test Identifier**
CST_F_1	The application shall store question templates in a database for later access and use.	CST-Con-2
CST_F_2	The application shall allow the user to query the database to retrieve question templates.	CST-Ret-2

CST_F_3	The application shall categorize and sub-categorize question templates to allow the user to query the database for a specific type of question.	CST-Con-2
CST_F_4	The application shall provide a way to store question difficulty for each question template.	CST-Upl-2
CST_F_5	The application shall allow the user to upload a file of question templates, specifically .txt files, that the software will then parse through and add each question to the database.	CST-Upl-2
CST_F_6	The application shall allow the user to save question templates to their devices.	CST-Ret-3
CST_F_7	The application shall allow the user to fill out question templates with data.	CST-Ret-3
CST_F_8	The application shall provide a way to manually enter a new question template in a text box and add it to the database.	CST-Upl-2
CST_F_9	The application shall allow the user to modify question templates already in the database.	CST-Mod-2
CST_F_10	The application shall recommend restrictions for assignments, based on the type of question chosen, in an effort to prevent students from using concepts that may make the problem trivial in lower-level courses.	CST-Ret-2
CST_P_1	Application shall be able to perform main tasks such template retrieval and database management without letting the application hang for more than ten seconds on an average internet connection.	CST-Con-1
CST_P_2	Application shall access the correct user's choice of template and let the user make	CST-Ret-2

	appropriate changes without changing the template document.	
CST_C_1	The web application shall connect to the database automatically whenever the user accesses the web interface.	CST-Con-1
CST_S_1	The application shall possess user authentication capabilities to maintain the confidentiality of the question templates written to the interface.	CST-Sec-1
CST_S_2	The application shall transmit user credentials between the client and the server in a manner that preserves confidentiality thereof.	CST-Sec-1
CST_S_3	Availability of the question template management and test creation features of the application shall be reserved to professors and TAs employed at the receiving university.	CST-Sec-2
CST_E_1	The application shall operate continuously, notwithstanding attempted denials of service.	CST-Con-1
CST_E_2	The application shall continue to accept requests for question template retrieval in the event of an internal error.	CST-Con-1
CST_E_3	The application shall continue to allow question template uploads in the event of an internal error.	CST-Con-1
CST_E_4	The application shall dispose of unused memory in a secure manner that prevents leakage.	All tests
CST_E_5	The application shall raise exceptions in the event of errors that compromise confidentiality, integrity, or availability.	All tests