

ClickEase

Software Test Description (STD)

Sara Nabulsi, Kyle Nehman, Marc Neisser, Parth Patel, Garrett Porter

Version 1.2

December 10, 2018

Version History

Date	Version	Description	Implemented by
11-29-18	1.0	Initial version from the template provided. Set up paper layout.	Sara
12-9-18	1.1	Added requirements traceability table (Section 5) and completed entering scope (Section 1) information.	Sara
12-10-18	1.2	Added pre-test preparations and test descriptions.	Garrett, Parth
12-10-18	1.2	General edits and corrections	Sara, Parth

Table of Contents

ClickEase	1
Software Test Description (STD)	1
Version History	2
Table of Contents	2
1. Scope.	6
1.1 Identification.	6
1.3 Document overview.	6
3. Test preparations.	7
3.1 Hardware preparation.	7
3.2 Software preparation.	8
3.3 Other pre test preparations.	8
4. Test descriptions.	8
4.1 T-1: Account registration and class configuration.	9
4.1.1 Requirements addressed.	9
4.1.2 Prerequisite conditions.	9
4.1.3 Test inputs.	9
4.1.4 Expected test results.	9
4.1.5 Criteria for evaluating results.	9
4.1.6 Test procedure.	9
4.1.7 Assumptions and constraints.	10
4.2 T-2: Quiz setup.	10
4.2.1 Requirements addressed.	10
4.2.2 Prerequisite conditions.	10
4.2.3 Test inputs.	10
4.2.4 Expected test results.	11
4.2.5 Criteria for evaluating results.	11
4.2.6 Test procedure.	11
4.2.7 Assumptions and constraints.	11
4.3 T-3: Take quiz.	11
4.3.1 Requirements addressed.	11
4.3.2 Prerequisite conditions.	11
4.3.3 Test inputs.	12
4.3.4 Expected test results.	12

4.3.5 Criteria for evaluating results.	12
4.3.6 Test procedure.	12
4.3.7 Assumptions and constraints.	12
5. Requirements traceability.	14
6. Notes.	15
A. Appendixes.	15

1. Scope.

1.1 Identification.

The ClickEase system is designed for operation within the University of Baltimore Maryland (UMBC) lecture system.

1.2 System overview.

The ClickEase system was designed to address a gap area in the current University of Maryland Baltimore County (UMBC) clicker system. The system is a free-to-use application designed for teachers and students for use in the classroom as a compliment to the lecture. It provides hub for teachers to create quizzes for students who can submit answers in real time through their devices during class. The system supplies teachers with data on student accuracy and attendance per quiz and holistically. The goal of the system is to provide a straightforward, free method of facilitating quizzes in class without the hassle of clicker subscriptions and fees.

The ClickEase system was loosely designed based on a suggestion from Michael Neary, and with implementation feedback from Dr. Russell Cain.

1.3 Document overview.

The Software Test Description (STD) document describes the test preparations, test cases, and test procedures to be used to perform qualification testing of ClickEase capabilities specified in the Software Requirements Specification (SRS) and Software Design Description (SDD) documents.

Section 1, Scope, identifies the ClickEase and provides a brief description of the system.

Section 2, Referenced Documents, identifies all documents referenced in or used in the creation of this document.

Section 3, Test Preparations, identifies pre-test preparations performed on the system and provides a brief description including procedures necessary to prepare the hardware and software for the test as well as any other pre-test preparations.

Section 4, Test Descriptions, identifies the individual tests cases performed on the system, detailing addressed requirements, prerequisite conditions, inputs, expected results, criteria for evaluating results, assumptions and constraints, and details the test procedure itself.

Section 5, Requirements Traceability, provides the traceability from each test case identified in this document to the system requirements allocated to it as well as traceability from each system requirement to the test cases that address it.

Section 6, Notes, may contain any additional information pertaining to the ClickEase that may not fit into one of the aforementioned sections.

2. Referenced documents.

Software Test Description (STD) for the Globally Relocatable Navy Tide/Atmospheric Modeling System (PCTides)

May 22, 2002

Report Number: NRL/MR/7320--02-8618

Document Title: Minutia Deviation Tool: Software Test Description (STD), (Version 1.0)

March 17, 2015

Report Number: 249555

3. Test preparations.

3.1 Hardware preparation.

A client hardware may consist of a computer, smartphone, tablet, or any internet capable device that accepts user input. Tests should be run using a variety of different devices to ensure compatibility. The client is in a state of readiness when connected to the internet. Instructions for connecting to the internet vary depending on the type of device. See manufacturer instructions for the device's internet connectivity instructions..

The server is currently being hosted on an AWS free-tier t2.micro EC2 VM and has the capability of being scaled up to a larger server, or to be moved to on premises if desired. The server hardware is in a state of readiness when it is connected to the internet. Server configurations such as cabling are specific to the type of server. Standard server configurations will be sufficient in most cases. See manufacturer or host instructions for setting up internet connectivity on the server.

The AWS free-tier EC2 Ubuntu server hardware configuration includes:

- 3.3 GHz Intel Xeon CPU

- 1 GiB Ram
- 8 GB HDD storage

3.2 Software preparation.

The client software shall be loaded through the ClickEase web app on a modern web browser. The client is in a state of readiness when it has a line of communication to the server. Tests should be run using a variety of modern browsers (i.e., Chrome 71, Safari 12, Firefox 63, IE 11) to insure compatibility.

The server software is in a state of readiness when it is listening for incoming requests from the client, and is able to send a response.

3.3 Other pre test preparations.

Logging should be set up to capture incoming and outgoing traffic from both the client(s) and the server. This requires adding logging code.

Information used to implement tests will be sample account information and questions/answers provided by the developers. This information will not be pulled from any outside sources.

4. Test descriptions.

Test cases for specific capabilities and features are included in later sections. System requirements not explicitly addressed by the test case, but satisfied through general installation and use of the system are included in the table below (Table 1).

Table 1: Requirements Traceability Matrix

Requirement	Description of Requirement	Test Case
SR-29	The system shall have a database to store user information	T-00

4.1 T-1: Account Registration and Class Configuration

4.1.1 Requirements addressed.

Requirements are addressed in section 5. However for the specific requirements that are addressed by this section (4.1) on account registration and class configuration are the following. Requirements SR-1, SR-2, SR-3, SR-9, SR-18, SR-19.

4.1.2 Prerequisite conditions.

The prerequisite conditions are to have properly configured clients and a server that are able to communicate with each other. Sections 3.1 and 3.2 provide additional details on proper configuration.

4.1.3 Test inputs.

The test will require unique user registration credentials for the student(s) and the teacher accounts. This input may be alphanumeric with symbols. It will be used initially for creation of the accounts, then as to login thereafter.

Student(s) must input a registration code that is generated by the teacher account. This is used to authenticate that they are allowed in the class.

4.1.4 Expected test results.

The intermediate test result is the creation of a teacher and student account(s) that can be logged into. The end result is that the student(s) should be added to the teacher's class. Student(s) should be able to view the class that they are in, and the teacher should be able to view all students who are in their class. To further delve into what is expected, is a database full of accounts that are registered as either a professor or student. As well as a table filled with ~150 students for a specific course.

4.1.5 Criteria for evaluating results.

The test is passed only if all students are added to the class. No errors should be encountered.

4.1.6 Test procedure.

- Step 1. A 'fake' professor account will be created that will simulate the possible real life professor creating an account.
- Step 2. This account then will create the class (or multiple) and create a key (generated for them) for students to use to log in. As stated in the assumptions below (Section 4.7), with limited availability of UMBC credited emails, students will just register through the code provided.
- Step 3. Student accounts will be generated and register for said course. 150 student accounts will be generated to stimulate the stress of a real large lecture. I.e Phys 121, Phys 122 and so on.
- Step 4. The database will be checked to see if all possible students were properly enrolled into the fake course that was created in step 1.

- Step 5. In the case that step 4 resulted in failure due to bugs, fixed will be made and steps 1-4 will be completed again to properly check the functionality of this test.

4.1.7 Assumptions and constraints.

The successful replication of creating real-life scenarios with how much load that the server can take when multiple students are creating accounts at once will be assumed to be overlooked. Constraints also are limited to not having emails sent out to students, since UMBC emails are required and only have access to 5 of said accounts.

4.2 T-2: Quiz setup

4.2.1 Requirements addressed.

Requirements are addressed in section 5. However for the specific requirements that are addressed by this section (4.2) on quiz creations and edits are the following. Requirements SR-5, SR-6, SR-7, SR-20.

4.2.2 Prerequisite conditions.

The prerequisite conditions are to have properly configured clients and a server that are able to communicate with each other. Sections 3.1 and 3.2 provide additional details on proper configuration. Another prerequisite requirement would be to have a successful completed test case 1 (T-1) in which a professor was successfully able to create an account and create a course. Having the course populated with student accounts is not required for this specific test case since no interaction comes from student accounts when creating quizzes.

4.2.3 Test inputs.

The test will require a professor to create an account and have a course set up. From this the professor should be able to create a quiz. This quiz should be editable, in the sense that questions can be removed and added prior to when students take it.

4.2.4 Expected test results.

The execution of Test Case 2 should result in either one or multiple quizzes within a specific course. The quiz should have a questions and possible correct answer based on the slides they want to use in class.

4.2.5 Criteria for evaluating results.

The test is passed only if a professor was able to create and edit a quiz. As well as to make sure the creation and amendments are present in the associated table within the database.

4.2.6 Test procedure.

- Step 1. A 'fake' professor account will log in to their account and be able to click on the create a quiz button for a specific course.
- Step 2. This account will then go and create a quiz with multiple questions. When in the quiz creation step the 'fake' professor will be asked for a question and a correct answer prior to moving along. When all the questions are added the professor can finish creating quiz.
- Step 3. Once quiz is created, the 'fake' processor can view. Step 3 will test the edit feature and have the test case remove and add questions.
- Step 4. The database will be checked to see if the most up-to-date version of the quiz is saved and ready for distribution.
- Step 5. In the case that step 4 resulted in failure due to bugs, fixed will be made and steps 1-4 will be completed again to properly check the functionality of this test.

4.2.7 Assumptions and constraints.

The assumptions for this test case are non-existent. Since a quiz creation has no real constraints.

4.3 T-3: Taking a Quiz

4.3.1 Requirements addressed.

Requirements are addressed in section 5. However for the specific requirements that are addressed by this section (4.2) on actually taking the quiz are the following. Requirements SR-8, SR-21, SR-22, SR-25.

4.3.2 Prerequisite conditions.

The prerequisite conditions are to have properly configured clients and a server that are able to communicate with each other. Sections 3.1 and 3.2 provide additional details on proper configuration. Another prerequisite requirement would be to have a successful completed test case 1 and test case 2. To test the ability for students to take a quiz it should require accounts for students to be enrolled in courses and for professors to have quizzes ready for the class.

4.3.3 Test inputs.

The test will require a professor to create an account and have a course set up. From this the professor should be able to create a quiz. This quiz should be editable, in the sense that questions can be removed and added prior to when students take it.

4.3.4 Expected test results.

The execution of Test Case 2 should result in either one or multiple quizzes within a specific course. The quiz should have a questions and possible correct answer based on the slides they want to use in class.

4.3.5 Criteria for evaluating results.

The test is passed only if a professor was able to start the session for a quiz and have students be able to answer the question within the time limit provided by the professor. The criteria for evaluating the results would be another database check to see if the final answer selected for the specific question within the time limit was recorded in the table. This should be true for all students for all questions within the quiz.

4.3.6 Test procedure.

- Step 1. Have the 'fake' professor account start the quiz session.
- Step 2. Have multiple student accounts (4 in this case) login and join the active quiz.
- Step 3. Within the active session have student accounts check to see the correct selected answer is highlighted within the app. Also have students check to see that they can change the selected answer while they are within the active time limit for a specific question.
- Step 4. Have the 'fake' professor account be able to end the quiz session, and this will not allow students from viewing anymore questions.
- Step 5. Check to see if database holds correct info for selected answers for each student that was present for said quiz. In the case that step 4 resulted in failure due to bugs, fixed will be made and steps 1-4 will be completed again to properly check the functionality of this test.

4.3.7 Assumptions and constraints.

The assumptions for this test case are that the number of students responding to each question will be the entire registered population, not just the five accounts that we stimulate. Constraints are added on this because it is possible to get 150 people together to test such a real life scenario.

5. Requirements traceability.

Requirement	Description of Requirement	Test Case
SR-1	The application shall allow users to create personal, non overlapping, accounts	T-1
SR-2	The application shall provide separate user interfaces for student and teacher accounts	T-1
SR-3	Teacher accounts shall generate a registration code upon the creation of a new class	T-1
SR-5	Teachers shall be able to create quizzes for each of their classes	T-2
SR-6	Teachers shall be able to add questions and answer choices for every quiz	T-2
SR-7	Teachers shall be able to define a specified period of time for each question to remain open during a quiz session, after which the question will end and answers will no longer be accepted.	T-2
SR-8	Teachers shall be able to start and end a quiz session	T-3
SR-9	Teachers shall be able to view a roster of students for each class	T-1
SR-18	Students shall be able to register for a course using a registration code provided by the Teacher.	T-1
SR-19	Students shall be able to view a list of all their registered courses	T-1
SR-20	Students shall be able to enter a quiz session by selecting a course with an active quiz session	T-2
SR-21	Students shall be able to respond to quiz questions by selecting the answer options provided on their user interface.	T-3
SR-22	Students shall be able to change their answer to a quiz question after a cooldown period in which no answers will be accepted.	T-3

SR-25	Student's current answer choice shall be highlighted on the user interface.	T-3
--------------	---	-----

6. Notes.

Removed SR 4, 10-17, 23, 24, 26-30 from the project scope. Metrics that were going to be a huge portion of the web app have been since removed and the overall project scope has been reworked as instructed by Dr. Haque.

While our initial scope has changed, we retain the original unique identifiers for our requirements that were stated in the SRS to maintain consistency across all documents.

A. Appendixes.