CENG 492

SENIOR PROJECT

Test Specification Report

PDC

(PROJECT DEVELOPMENT CENTER)

PROJECT TITLE: VIRTUAL CLASSROOM

COMPANY STAFF:

Harun Alpak	1249903
Ömer Faruk Aygün	1297506
Ufuk Biçen	1297555
Shahin Abdurrehimov	114068

Table of Contents

Introduction

1. Goals and Objectives

The testing is one of the main steps of developing a good quality software and importance of this process cannot be overemphasized. While developing this document we tried to focus on these objectives:

- Find out undiscovered errors.
- Design a good test plan.
- Try to observe the cases in different aspects.
- Handle all cases

2. Statement of Scope

In this document we mainly focused on black box testing that focuses on the functional requirements of the software. First we described each module ant their functionality and behaviors of the functions than we specified the functions to be tested and derived test cases in a certain simple format as many as possible.

We know that testing cannot show the absence of errors and defects and we keep this rule in mind while developing these test cases.

3. Test Specification Format

Requirement set: This part describes the functionality of the module and how this module works. By using these requirement sets for each module we developed test cases.
Tested Function: Identifies the function that is tested. We gave an index for each tested function of each module.
Case No: The test case number should be a three digit identifier of the following form: c.s.t, where: c- is the module number, s- is the section number (represents the function of the module), and t- is the test case number.
Title: is the title of the test.
Author: is the person who wrote the test specification.
Scenario: is the situation that is possible to occur and that may probably cause error.

Test Specifications:

1. Login Module

Requirements Set:

This module will handle all the interaction with users. This module gets the username, password, user type and the server address from the user and connects to database for checking the validity of the username and password information. When the user is student then this class checks whether any course is available. If any course is available, it sends the username and user type to server for registering. Then this class creates an instance of the Virtual Classroom class. If there is not available course then this class returns a notification message to user. In addition to that when the connection is not established, this class returns error message to user. When the user is admin or instructor then this class creates the instances of the AdminInterface or InstructorInterface respectively.

For this module we will develop and test the following cases for the specified functions of this module:

***Tested Function*:** Check user input (1)
Case No: 1.1.1
Title: No input test
Author: Ömer Faruk AYGÜN
Scenario: The user enters nothing in the user name, password and server address fields and presses ‘Login’ button.
Expected Result: The program should return a notification message

Case No: 1.1.2
Title: Missing input test
Author: Ömer Faruk AYGÜN
Scenario: The user does not fill all the fields of the login form and tries to log in.
Expected Result: The program should notify the user to try properly

Case No: 1.1.3
Title: Format of server address field test

Author: Ömer Faruk AYGÜN

Scenario: The user enters the server address in wrong format (e.g. www.localhost.com or 10.0.0.1983)

Expected Result: The program should notify the user to try properly.

Case No: 1.1.4

Title: Invalid usertype test

Author: Ömer Faruk AYGÜN

Scenario: The user does not select the user type from the drop down list and types an invalid text and tries to login.

Expected Result: The program should detect the situation and notify the user to try properly.

Case No: 1.1.5

Title: Length of username and password test

Author: Ömer Faruk AYGÜN

Scenario: The user enters very long username and password (say 100-200 characters)

Expected Result: There must be limits in the input boxes that are determined by the length of the field in the database.

***Tested Function*:** Connect to database and validate user input (2)

Case No: 1.2.1

Title: Non existent user test

Author: Ömer Faruk AYGÜN

Scenario: The username, password and usertype that are entered by the user do not exist in any records of the users table.

Expected Result: The program should notify the user and return back.

Case No: 1.2.2

Title: Partially wrong user input test

Author: Ömer Faruk AYGÜN

Scenario: The user enters two of the three input fields (username-password, username-usertype, password-usertype) fairly and the remaining field wrongly. Or enters only one of the fields true and the other fields wrong.

Expected Result: The program should notify the user and return back

Case No: 1.2.3
Title: Server not active test.
Author: Ömer Faruk AYGÜN
Scenario: The user tries to login while the server is not active or database is not working.
Expected Result: The program should give an error message to correct the situation.

Case No: 1.2.4
Title: Invalid usertype test
Author: Ömer Faruk AYGÜN
Scenario: User enters something in the user type field instead of selecting from drop down list.
Expected Result: The program should detect the situation and notify the user to try properly.

Case No: 1.2.5
Title: Student number test
Author: Ömer Faruk AYGÜN
Scenario: Large number of students tries to log in to the virtual classroom
Expected Result: 20 students should log in properly the system should not give permission to the other attempts (e.g. more than 20 students)

Case No: 1.2.6
Title: Instructor number test.
Author: Ömer Faruk AYGÜN
Scenario: After login of an instructor another user tries to login to the virtual classroom as an instructor.
Expected Result: The system should not give permission to login

Case No: 1.2.7
Title: Admin number test.
Author: Ömer Faruk AYGÜN
Scenario: After login of an admin another user tries to login to the virtual classroom as an admin.

Expected Result: The system should not give permission to login

2. Admin Interface Module

Requirements Set: This module provides adding/deleting user and course to database. Only admin can reach and use this module's functions. Admin can add user and course to system. When he/she creates a course in the database its status is set to 'offline' by default and a folder is created in the server side under the folder of VC that is created before, whose name is the created course's name.

***Tested Function*:** Validate the course name input and add to database **(1)**
Case No: 2.1.1
Title: Empty course name test.
Author: Ömer Faruk AYGÜN
Scenario: Admin lefts the course name field empty and clicks 'Create' button.
Expected Result: The system should not give permission to create such course and notify the user

Case No: 2.1.2
Title: Invalid course name test.
Author: Ömer Faruk AYGÜN
Scenario: The admin writes an invalid course name that the program cannot create this named folder in the server side. (E.g. Biology/1 or very long name says 100-200 characters)
Expected Result: The system should not give permission to such an action there must be limit in the database field.

Case No: 2.1.3
Title: Creating an existing course test
Author: Ömer Faruk AYGÜN
Scenario: Admin creates a course that already exists in the database
Expected Result: It should not be permitted and the user should be informed.

***Tested Function*:** Validate the user name, password and user type input and add to database **(2)**

Case No: 2.2.1

Title: Empty fields in the user info test.

Author: Ömer Faruk AYGÜN

Scenario: Admin leaves one or more fields of the user name, user type or password fields of the create user form and clicks ‘Create’ button.

Expected Result: It should not be permitted and the user should be informed

Case No: 2.2.2

Title: Invalid usertype test

Author: Ömer Faruk AYGÜN

Scenario: Admin writes an invalid user type in the user type field instead of selecting from drop down list.

Expected Result: It should not be permitted and the user should be informed

Case No: 2.2.3

Title: Creating an existent user test

Author: Ömer Faruk AYGÜN

Scenario: The admin creates a user that is already exists in the user table.

Expected Result: It should not be permitted and the user should be informed

Case No: 2.2.4

Title: Improper user name or password test

Author: Ömer Faruk AYGÜN

Scenario: The admin gives a very long username or password.

Expected Result: It should not be permitted and the user should be informed

3. Instructor Interface Module

Requirements Set: This module handles all of the instructor operations. By this module instructor can see all of the courses, and select one of them then he/she prepare new document by using our offline tool or can upload new power point slides to this course’s folder in the server side. In addition to that he/she can start the selected course.

Tested Function: Selecting course and starting lecture (1)

Case No: 3.1.1

Title: Try to start non existent course test

Author: Ömer Faruk AYGÜN

Scenario: User writes something invalid or leaves empty the 'Select Course' combo box field instead of selecting from the drop down list.

Expected Result: It should not be permitted and the user should be informed

Case No: 3.1.2

Title: Concurrent lecture test.

Author: Ömer Faruk AYGÜN

Scenario: Instructor starts a lecture and then instructor selects another course from drop down list and tries to start that course.

Expected Result: It should not be permitted and the user should be informed

Tested Function: Selecting course and uploading document (2)

Case No: 3.2.1

Title: Course name test.

Author: Ömer Faruk AYGÜN

Scenario: The user does not select any course from 'Select Course' drop down list and writes an arbitrary course in the field and tries to upload a file.

Expected Result: The program should detect the situation and notify the user to try properly

Case No: 3.2.2

Title: Not existing course select test.

Author: Ömer Faruk AYGÜN

Scenario: User types a course that does not exist in the database and tries to upload document.

Expected Result: The program should detect the situation and notify the user to try properly

Case No: 3.2.3

Title: File name test

Author: Ömer Faruk AYGÜN

Scenario: User types an invalid filename and clicks 'OK'.

Expected Result: The program should detect the situation and notify the user to try properly

Case No: 3.2.4
Title: File size test
Author: Ömer Faruk AYGÜN
Scenario: The user selects a very large file say 1MB or 2 MB and tries to upload to the server.
Expected Result: The system should make this request that time depends on the connection speed.

Tested Function: Selecting course and creating material (3)
Case No: 3.3.1
Title: Course name test
Author: Ömer Faruk AYGÜN
Scenario: User does not select any course from the drop down (writes an arbitrary name or leaves empty) list and clicks 'Create Material' button.
Expected Result: The program should detect the situation and notify the user to try properly

4. WhiteBoard Module

Requirements Set:

This class controls all of the operations into the whiteboard. Basically this class has two major operations. One of them is about drawing operations and the other is about the controlling the slide show (such as open, pause, resume, next, previous). Our white board module has drawing capabilities, editing capabilities, and also displaying slides capabilities. Besides that our whiteboard is used by one person (instructor or specified student).

In addition, all of the operation is sent to the clients (except instructor). And the clients can see all of the operation. These are provided by some meaningful messages.

The followings are the main drawing functions of the whiteboard:

- Draw line, rectangle and ellipse.
- Clear whiteboard.
- Select line, rectangle, ellipse and text.
- Create text.

- Move line, rectangle and ellipse.
- Delete selected object.
- Undo the last operation(only add and delete operations)
- Copy selected object.
- Paste the copied object.

Note: The following tests are examined both sides of the program which are server & client sides for whiteboard module.

***Tested Function*:** Draw Object (1)

For drawing object firstly we must press the draw button. When it pressed the color of the button is changed (became red). When we press the other button or press the draw button again the color is changed back to old color. After activating the draw mode we can draw object by using mouse. The size and color of pen can be changed using SelectSize or SelectColor combo boxes. Our whiteboard have 3 size and 4 color options. Size options are 2(default), 4 and 6. Color options are black (default), blue, red and green.

Case No: 4.1.1
Title: Test whether the user can draw any object while the draw mode is not activated.
Author: Harun ALPAK
Scenario: The user tries to draw any object while the draw button is not pressed.
Expected Result: Noting is drawn.

Case No: 4.1.2
Title: Test whether the draw mode can be activated and deactivated.
Author: Harun ALPAK
Scenario: The user presses the draw button and tries to draw any object. And then he/she presses draw button again and then tries to draw any object again.
Expected Result: Firstly the color of the draw button became red (which shows that the draw mode is activated) and the object is drawn. After the second press of the draw button, the color of the draw button is changed back to old color and noting is drawn.

Case No: 4.1.3

Title: Test whether the drawing mode is deactivated when presses the clear, select, undo, delete, text button.
Author: Harun ALPAK
Scenario: The user firstly activates the draw mode by pressing the draw button. Then he/she press the clear, select, undo, delete, text button in this order. After each press the user tries to draw any object. The user tries these operations for each button (clear, select, undo, delete, text button) independently.
Expected Result: Noting is drawn. And the color of the draw button is changed back to old color.

Case No: 4.1.4
Title: Test whether the user can draw line.
Author: Harun ALPAK
Scenario: The user draws an arbitrary line after pressing the draw button. (Default drawing mode is pen, pen size is 2 and pen color is black.)
Expected Result: This arbitrary line is drawn with default properties. (Size=2 and color=black).

Case No: 4.1.5
Title: Test whether the user can draw rectangle.
Author: Harun ALPAK
Scenario: The user draws an arbitrary rectangle after pressing the draw button and selects rectangle draw mode. (Default pen size is 2 and pen color is black.)
Expected Result: This arbitrary rectangle is drawn with default properties. (Size=2 and color=black).

Case No: 4.1.6
Title: Test whether the user can draw ellipse.
Author: Harun ALPAK
Scenario: The user draws an arbitrary ellipse after pressing the draw button and selects ellipse draw mode. (Default pen size is 2 and pen color is black.)
Expected Result: This arbitrary ellipse is drawn with default properties. (Size=2 and color=black).

Case No: 4.1.7

Title: Test whether pen size (for line) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 3 lines with pen sizes 2, 4 and 6 respectively and compares them.

Expected Result: 3 lines are drawn with pen sizes 2, 4 and 6 respectively.

Case No: 4.1.8

Title: Test whether pen size (for rectangle) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 3 rectangles with pen sizes 2, 4 and 6 respectively and compares them.

Expected Result: 3 rectangles are drawn with pen sizes 2, 4 and 6 respectively.

Case No: 4.1.9

Title: Test whether pen size (for ellipse) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 3 ellipses with pen sizes 2, 4 and 6 respectively and compares them.

Expected Result: 3 ellipses are drawn with pen sizes 2, 4 and 6 respectively.

Case No: 4.1.10

Title: Test whether pen color (for line) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 4 lines with pen color black, blue, red, green respectively and compares them.

Expected Result: 4 lines are drawn with pen color black, blue, and red, green respectively.

Case No: 4.1.11

Title: Test whether pen size (for rectangle) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 4 rectangles with pen color black, blue, red, green respectively and compares them.

Expected Result: 4 rectangles are drawn with pen color black, blue, red, green respectively.

Case No: 4.1.12

Title: Test whether pen size (for ellipse) mode works properly.

Author: Harun ALPAK

Scenario: The user draws an arbitrary 4 ellipses with pen color black, blue, red, green respectively and compares them.

Expected Result: 4 ellipses are drawn with pen color black, blue, red, green respectively.

Case No: 4.1.13

Title: Robustness line test.

Author: Harun ALPAK

Scenario: The user draws a line with pen size is 6 and color red which covers all of the whiteboard.

Expected Result: This line is drawn and whole whiteboard is covered by this line.

Case No: 4.1.14

Title: Robustness rectangle test.

Author: Harun ALPAK

Scenario: The user draws 50 rectangles (crowded) with pen size is 6 and color red which covers all of the whiteboard.

Expected Result: These 50 rectangles are drawn and whole whiteboard is covered by these rectangles.

Case No: 4.1.15

Title: Robustness ellipse test.

Author: Harun ALPAK

Scenario: The user draws 50 arbitrary ellipses.

Expected Result: These 50 ellipses are drawn.

***Tested Function*:** Select & Move & Undo & Delete & Clear Object (2)

Case No: 4.2.1

Title: Test whether the user can select any object while the select mode is not activated.

Author: Harun ALPAK

Scenario: The user draws 1 line, 1 rectangle and 1 ellipse then tries to select any object.
Expected Result: Noting is selected.

Case No: 4.2.2
Title: Test whether the user can select any object while the select mode is activated.
Author: Harun ALPAK
Scenario: The user draws 1 line, 1 rectangle and 1 ellipse then tries to select each object object respectively.
Expected Result: Line, rectangle and ellipse are selected respectively and the color of the selected object became yellow.

Case No: 4.2.3
Title: Test whether the color of deselected object is equal to old color of this object.
Author: Harun ALPAK
Scenario: The user draws 4 line (with color black, blue, red, green respectively), 4 rectangle (with color black, blue, red, green respectively) and 4 ellipse (with color black, blue, red, green respectively) then selects and deselects each of the object.
Expected Result: The original color of each object is preserved.

Case No: 4.2.4
Title: Test whether the selected object can be moved.
Author: Harun ALPAK
Scenario: The user draws 1 line, 1 rectangle and 1 ellipse then selects and moves each object object respectively.
Expected Result: The Line, rectangle and ellipse are selected and moved respectively.

Case No: 4.2.5
Title: Test whether whiteboard can be cleared correctly.
Author: Harun ALPAK
Scenario: The user draws 10 lines, 10 rectangles and 10 ellipses arbitrarily. Then presses the clear button.
Expected Result: All of the objects are cleared.

Case No: 4.2.6

Title: Undo Test

Author: Ufuk BICEN

Scenario: The user presses the undo button.

Expected Result: The last delete/add action is undone. (If there is no such action then nothing should happen. This test can be repeated 10+n times (where n >=0 and n is an integer) to see that at most 10 undo actions can be done).

Case No: 4.2.7

Title: Delete Test

Author: Harun Alpak

Scenario: The user draws 1 rectangle, 1 line and 1 ellipse then selects each of them and deletes them.

Expected Result: All objects are deleted.

***Tested Function*:** Add Text (3)

Case No: 4.3.1

Title: Insert test

Author: Shahin Abdurahimov

Scenario: Instructor clicks text button, clicks left button on whiteboard and enters string.

Expected result: Input text must appear on whiteboard.

Case No: 4.3.2

Title: Insert test

Author: Shahin Abdurahimov

Scenario: Instructor changes properties such as color, font, size and style of text using dialog box and enters string. (Try 5 different combinations)

Expected result: Text must be displayed on whiteboard at a given format.

Case No: 4.3.3

Title: Insert test

Author: Shahin Abdurahimov

Scenario: Instructor inserts text to five different places of whiteboard.

Expected result: Text must be displayed on whiteboard at a given place.

Case No: 4.3.4

Title: Robustness test

Author: Shahin Abdurahimov

Scenario: Instructor enters 10 very long texts (say 100-200 characters).

Expected result: Displayed text number and length must be equal to the input's number and length.

***Tested Function*:** LoadSlide & Next & Previous Functions (4)

Case No: 4.4.1

Title: Test whether the user can load any presentation.

Author: Harun ALPAK

Scenario: The user first selects presentation from SelectFile combo box and then presses the LoadSlide button.

Expected Result: The selected slide is loaded into whiteboard as an image.

Case No: 4.4.2

Title: Test whether the user can show the next slide.

Author: Harun ALPAK

Scenario: After loading the presentation the user presses the next button.

Expected Result: The next slide in this presentation is shown.

Case No: 4.4.3

Title: Test whether the user can show the previous slide.

Author: Harun ALPAK

Scenario: After loading the presentation and pressing next slide the user presses the previous button.

Expected Result: The previous slide in this presentation is shown.

Case No: 4.4.4

Title: Test whether the user warned when it is at the first slide and the user presses the previous button.

Author: Harun ALPAK

Scenario: After loading the presentation the user presses the previous button.

Expected Result: The warning message is shown.

Case No: 4.4.5
Title: Test whether the program shows the first slide when it is at the last slide and the user presses the next button.
Author: Harun ALPAK
Scenario: After showing the last slide show the user presses the next button.
Expected Result: The first slide is shown.

Case No: 4.4.6
Title: Test whether each slide has its own drawing.
Author: Harun ALPAK
Scenario: After loading the presentation, the user draw line for the first slide and presses the next button. And draw a rectangle for second slide and presses the next button. And draw an ellipse for third slide and presses the next button.
Expected Result: When he/she presses the previous button the user should see the 3rd slide with an ellipse. And when he/she presses the previous button the user should see the 2nd slide with a rectangle. And when he/she presses the previous button the user should see the 1st slide with a line.

5. Offline Module

Requirements Set:

This module gives instructor the options of creating and editing existing slides for the selected course. He/she can download an existing document and make some changes on that document such as adding new slides, adding some annotations to the existing slides or he/she can create a new slide. This tool can use the whiteboard's drawing functions so it will help the instructor to create annotations on the slides. Only the instructor can use the functions of this module. Note that, the instructor can create and edit slides not only when he/she is offline but also he/she is online.

For this module we will develop and test the following cases for the specified functions of each module:

***Tested Function*:** Check user login (1)

Case No: 5.1.1

Title: Correct User Test

Author: Ufuk BICEN

Scenario: The user logs in the system.

Expected Result: If the user is student he/she cannot use the buttons of this class, however if he/she is the instructor he can use them.

***Tested Function*:** Create Material (2)

Case No: 5.2.1

Title: Course Selection Test

Author: Ufuk BICEN

Scenario: The instructor tries to create a material without selecting a course.

Expected Result: A warning message saying that the user should first select which course he/she wants to create the material for should be shown.

Case No: 5.2.2

Title: Create Offline Material Button Test

Author: Ufuk BICEN

Scenario: The instructor selects the course and then presses the button.

Expected Result: The Virtual class module should be opened with limited functionality. Here, streaming, chat and Q/A Box modules should not be shown to the user.

Case No: 5.2.3

Title: Insert Slide to the End Test

Author: Ufuk BICEN

Scenario: The instructor presses the insert a slide button without entering a slide number.

Expected Result: The slide should be inserted to the end of the slideshow.

Case No: 5.2.4

Title: Insert Slide Test

Author: Ufuk BICEN

Scenario: The instructor presses the insert slide button with the specified slide number.

Expected Result: The slide should be inserted to the material and all the slides numbers should be updated. (If there exists a slide with that number, that slide and the slides after it should be incremented by one).

Case No: 5.2.5
Title: Load Slide Test
Author: Ufuk BICEN
Scenario: The instructor presses the Load Slide button.
Expected Result: The saved slideshow should be loaded and the first slide of the show should be opened.

Case No: 5.2.6
Title: Next Slide Test
Author: Ufuk BICEN
Scenario: The instructor presses the Next Slide button.
Expected Result: The next slide should be opened. (If the last slide is already open, a warning message saying that the user is at the end of the slideshow should be shown).

Case No: 5.2.7
Title: Previous Slide Test
Author: Ufuk BICEN
Scenario: The instructor presses the Previous Slide button.
Expected Result: The previous slide should be opened. (If the first slide is already open, a warning message saying that the user is at the first slide should be shown).

Case No: 5.2.8
Title: Go to Next Slide Test
Author: Ufuk BICEN
Scenario: The instructor presses the Go button without entering a number.
Expected Result: The next slide should be opened. (If the last slide is already open, a warning message saying that the user is at the end of the slideshow should be shown).

Case No: 5.2.9
Title: Go to Slide Test

Author: Ufuk BICEN

Scenario: The instructor presses the Go button entering the slide number.

Expected Result: The slide with the specified number should be opened. (If there exists no slide with that number the user should be warned and the last slide should be opened).

Case No: 5.2.10

Title: Go to Slide Test

Author: Ufuk BICEN

Scenario: The instructor presses the Goto Slide button without entering a number.

Expected Result: The next slide should be opened. (If the last slide is already open, a warning message saying that the user is at the end of the slideshow should be shown).

Case No: 5.2.11

Title: Bullet Test

Author: Ufuk BICEN

Scenario: The instructor presses the Bullet button.

Expected Result: The textbox should be placed as a next bulleted format waiting for the user insert text.

Case No: 5.2.12

Title: Increase Indent Test

Author: Ufuk BICEN

Scenario: The instructor presses the Increase Indent button.

Expected Result: The textbox should be placed as an increased indent waiting for the user insert text.

Case No: 5.2.13

Title: Decrease Indent Test

Author: Ufuk BICEN

Scenario: The instructor presses the Decrease Indent button.

Expected Result: The textbox should be placed as a decreased indent waiting for the user insert text.

6. Chat Module

This module will handle all the interaction between users. Every student and Instructor can use this module to talk with each other.

***Tested Function*: Chat Send** (1)
Case No: 6.1.1
Title: Connection test
Author: Shahin Abdurahimov
Scenario: You enter VirtualClass interface.
Expected result: In the chat box there must be written “You have joined the chat”

Case No: 6.1.2
Title: Connection test
Author: Shahin Abdurahimov
Scenario: Any other user enters VirtualClass interface.
Expected result: In the chat box there must be written name of user and “have joined the chat”

Case No: 6.1.3
Title: Input test
Author: Shahin Abdurahimov
Scenario: The user enters string as an input.
Expected result: Input string must be displayed on all clients chat box.

Case No: 6.1.4
Title: Turkish character test
Author: Shahin Abdurahimov
Scenario: The user enters string as an input including Turkish characters.
Expected result: It must be displayed correctly on connected user’s side.

Case No: 6.1.5
Title: Length of input test

Author: Shahin Abdurahimov

Scenario: The user enters very long input (say 100-200 characters)

Expected result: Displayed text length must be equal to the input's length.

7. Question Module

This module is for students to ask questions. And Instructor answers these questions by this module.

***Tested Function*: Question Send** (1)

Case No: 7.1.1

Title: Ask test

Author: Shahin Abdurahimov

Scenario: Student enters string to question box.

Expected result: Student name + "asked " + input string must be displayed for all connected students and Instructor.

Case No: 7.1.2

Title: Ask robustness test

Author: Shahin Abdurahimov

Scenario: Student enters 10 strings to question box.

Expected result: Input strings must be displayed on all clients' side.

Case No: 7.1.3

Title: Ask robustness test

Author: Shahin Abdurahimov

Scenario: The user enters very long input (say 100-200 characters).

Expected result: Displayed text length must be equal to the input's length.

***Tested Function*: Question Answer** (2)

Case No: 7.2.1

Title: Answer test

Author: Shahin Abdurahimov

Scenario: Instructor enters string to answer box without selecting user.

Expected result: No action will be done.

Case No: 7.2.2

Title: Answer test

Author: Shahin Abdurahimov

Scenario: Instructor enters string to answer box with selecting user.

Expected result: “Instructor replied to”+ user name + input string will be displayed.

Case No: 7.2.3

Title: Ask robustness test

Author: Shahin Abdurahimov

Scenario: Instructor enters 10 strings to answer box.

Expected result: Input strings must be displayed on all clients’ side.

Case No: 7.2.4

Title: Ask robustness test

Author: Shahin Abdurahimov

Scenario: Instructor enters very long input (say 100-200 characters).

Expected result: Displayed text length must be equal to the input’s length.

8. Control Box (User list)

This module is for all users to see connected users to the system. And also teacher can select student to answer to him/her or to allow the selected student to use whiteboard.

Case No: 8.1.1

Title: List users

Author: Shahin Abdurahimov

Scenario: List users button clicked.

Expected result: All user names must be listed correctly.

Case No: 8.1.2

Title: List robustness test

Author: Shahin Abdurahimov

Scenario: List users button clicked when there are 10 users and some users have more than 30 characters long user name.

Expected result: All user names must appear.

Case No: 8.1.3

Title: End course test

Author: Shahin Abdurahimov

Scenario: Instructor clicks end course button.

Expected result: Instructor will be removed from VirtualClass interface.

9. Streaming Module

This module is for capturing and playing audio/video from web cam and microphone.

Case No: 9.1.1

Title: Enter test

Author: Shahin Abdurahimov

Scenario: Instructor enters VirtualClass interface.

Expected result: Video and audio must be captured and played at video area.

Case No: 4.1.2

Title: Enter test

Author: Shahin Abdurahimov

Scenario: Student enters VirtualClass interface.

Expected result: Video and audio must be played at video area, getting streaming data from server. There must not be any delay playing stream at student's side.

Case No: 4.1.3

Title: Robustness test

Author: Shahin Abdurahimov

Scenario: 10 Student enters VirtualClass interface.

Expected result: There must not be any delay playing stream at students' side.