

VORMUND

Black Box Testing Plan for Testify

Team 6

Seth Cook
Margarita Lee Li
Samuel Marshall
Adam Rice
Mitchell Stendal
John Vall

Test Plan Requirements

Requirement: User to student client interaction

- **Functionality Test Cases**
 - Login Test 001 - Student account creation
 - **Severity:** 2
 - **Instructions:** Enter a first name and last name for text fields within the system.
 - **Expected Result:** Account creation will proceed normally, and the login window will close, revealing the test window.
 - Login Test 002 - Student account persistence
 - **Severity:** 2
 - **Instructions:** Once a new account has been created, close the window, then open the student window and login again.
 - **Expected Result:** The student should successfully be logged into the system and login window will close, opening the exam.
 - Login Test 003 - Incorrect First name
 - **Severity:** 2
 - **Instructions:** Attempt to login to the system using an incorrect first name.
 - **Expected Result:** The student should be prompted to input another first name.
 - Login Test 004 - Incorrect Last name
 - **Severity:** 2
 - **Instructions:** Attempt to login to the system using an incorrect last name.
 - **Expected Result:** The student should be prompted to input another last name.
- **Equivalence Test Cases**
 - Login Test 005 - Proper first name input
 - **Severity:** 2
 - **Instructions:** Input a first name that isn't in the system.
 - **Expected Result:** Student account creation will proceed as normal, and the login window will close, opening the exam window.
 - Login Test 006 - Proper last name input
 - **Severity:** 2
 - **Instructions:** Input a last name that isn't in the system.
 - **Expected Result:** Student account creation will proceed as normal,

and the login window will close, opening the exam window.

- Boundary Value Test Cases
 - Login Test 007 - Empty first name input
 - **Severity:** 2
 - **Instructions:** Input an empty first name into the system.
 - **Expected Result:** Student will be alerted of the incorrect input and asked to put a correct one.
 - Login Test 008 - Empty last name input
 - **Severity:** 2
 - **Instructions:** Input an empty last name that is in the system.
 - **Expected Result:** Student will be alerted of the incorrect input and asked to put a correct one.

Requirement: User to instructor client interaction

- Functionality Test Cases
 - Open Existing Test 001 - Attempt to open existing test
 - **Severity:** 1
 - **Instructions:** Select “Edit Test” from the home screen and proceed to browse for the existing exam file you wish to open.
 - **Expected Result:** The test properties menu will open allowing you to see the questions and make adjustments to the test data.
 - Open Existing Test 002 - Save test and re-open
 - **Severity:** 1
 - **Instructions:** Select “Edit Test” from the main menu and select the test you would like to edit. Make a minor change to the wording of a question, and save the test. Exit the test after saving and re-open the test to see if the minor change was successfully saved.
 - **Expected Result:** The saved change will be present in the re-opened version of the test file.
 - Create New Test 001 - Attempt to create a new test
 - **Severity:** 1
 - **Instructions:** Select “Create Test” from the home screen and use the menu to begin adding questions to the test.
 - **Expected Result:** The test menu will open allowing you to begin building the questions and answer key.
 - Create New Question 001 - Create short answer question
 - **Severity:** 1
 - **Instructions:** Select “Short Answer” in the menu bar of the program window under “Question”->”New.” Fill in information for the short

- answer question and include it in the test.
 - **Expected Result:** The short answer question will be successfully added to the test.
 - Create New Question 002 - Create multiple choice question
 - **Severity:** 1
 - **Instructions:** Select “Multiple Choice” in the menu bar of the program window under “Question”->”New.” Fill in information and answer choices for the multiple choice question and include it in the test.
 - **Expected Result:** The multiple choice question will be successfully added to the test.
 - Create New Question 003 - Create programming question
 - **Severity:** 1
 - **Instructions:** Select “Programming” in the menu bar of the program window under “Question”->”New.” Fill in information for the programming question and include it in the test.
 - **Expected Result:** The programming question will be successfully added to the test.
 - Modify Test 001 - Modify test properties
 - **Severity:** 1
 - **Instructions:** Select “Edit Test” from the main menu of the program. Begin to make modifications to questions, changing answer choices and other test parameters.
 - **Expected Result:** The changes will be reflected in the saved copy of the exam upon program exit.
 - Equivalence Test Cases
 - Open Existing Test 003 - Do not select a file with proper file type
 - **Severity:** 2
 - **Instructions:** When browsing for a test file to open, select a file type that is not compatible with the program implementation.
 - **Expected Result:** A dialog box should appear stating that a file with an incompatible file type was selected. The user will then be able to select a different file.
 - Multiple Choice Test 001 - Create multiple choice question with 3 answer choices
 - **Severity:** 1
 - **Instructions:** Attempt to create a multiple choice question with 3 answer choices, since that falls within the possible equivalence class for number of answer choices.

- **Expected Result:** The multiple choice question should be added successfully to the test.
 - Modify Test 002 - Leave existing test questions incomplete
 - **Severity:** 2
 - **Instructions:** Open an existing test using the “Edit Test” option in the program’s main menu. Then choose a question and modify its parameters/answer choices, leaving it incomplete (e.g., without question text or an answer). Then attempt to save the test.
 - **Expected Result:** A dialog box should appear saying that the test question has not been completed. It should prompt the user to finish inputting the question parameters.
 - Create New Question 004 - Create short answer question
 - **Severity:** 2
 - **Instructions:** Select “Short Answer” in the menu bar of the program window under “Question”->”New.” When filling in the test question fields, leave at least one field blank. Then attempt to add the question to the exam.
 - **Expected Result:** A dialog box will open, prompting the user to include the portions of the question that have yet to be input by the instructor.
 - Create New Question 005 - Create multiple choice question
 - **Severity:** 2
 - **Instructions:** Select “Multiple Choice” in the menu bar of the program window under “Question”->”New.” When filling in the test question fields and answer choices, leave at least one field blank. Then attempt to add the question to the exam.
 - **Expected Result:** A dialog box will open, prompting the user to include the portions of the question that have yet to be input by the instructor.
 - Create New Question 006 - Create programming question
 - **Severity:** 2
 - **Instructions:** Select “Programming” in the menu bar of the program window under “Question”->”New.” When filling in the test question fields, leave at least one field blank. Then attempt to add the question to the exam.
 - **Expected Result:** A dialog box will open, prompting the user to include the portions of the question that have yet to be input by the instructor.
- Boundary Value Test Cases

- Multiple Choice Test 002 - Create multiple choice question with 1 answer choice
 - **Severity:** 3
 - **Instructions:** Attempt to create a multiple choice question with only one possible answer choice.
 - **Expected Result:** The program will open a dialog box prompting the user to include more than one answer choice for a multiple choice question.

Requirement: Instructor client is able to save and open a tests to the file system.

- Functionality Test Cases
 - Save As New Test 001 - Test creation
 - **Severity:** 1
 - **Instructions:** Click “Create Test” on the home screen
 - **Expected Result:** New window opens allowing user to create test questions/answers
 - Save As New Test 002 - Save a new test
 - **Severity:** 1
 - **Instructions:** Once questions/answers have been created on a new test, go to the Test menu at the top left and click “Save As”
 - **Expected Result:** A dialog box should appear with the file system displayed and allow the user to select where to save the test file, as standard with most operating systems
 - Open Existing Test 001 - Open existing test
 - **Severity:** 1
 - **Instructions:** Click “Edit Test” on the home screen
 - **Expected Result:** Dialog box opens with file system in screen, allows user to select a test and have that test data populated into a new window with editable components
 - Save Existing Test 001 - Save an existing test
 - **Severity:** 1
 - **Instructions:** Once a test is open in the dialog box, click Test > Save
 - **Expected Result:** No additional windows/popups should appear. The program overwrites the open file with new data.
- Boundary Value Test Cases
 - Save As New Test 001 - Saving over an existing file name
 - **Severity:** 2
 - **Instructions:** Attempt to save a newly created test over one that previously exists, i.e. same file name under same directory

- **Expected Result:** A modal/popup box will open asking the user to confirm if they want to overwrite the existing file OR cancel and try again
 - Save as Existing Test 001 - Clicking “Save As” while having an existing test open
 - **Severity:** 3
 - **Instructions:** Open a test, click Test > Save As
 - **Expected Result:** Save As dialog should open and operate as normal.
 - Open Non-existing test 001 - Try to open a filename that isn't in the system
 - **Severity:** 1
 - **Instructions:** Click “Edit a Test” on the home screen, modal should appear displaying filesystem. Enter in a filename that doesn't exist.
 - **Expected Result:** Error message should inform user that file does not exist.
- Equivalence Class Cases
 - *Not applicable for this requirement*

Requirement: Student application is able to open and save tests to/from the file system on their computer.

- Functionality Test Cases
 - New Test 001 - Start a new test an instructor created
 - **Severity:** 1
 - **Instructions:** After entering a first/last name on the home screen, dialog box will prompt user to find test file on file system created by instructor
 - **Expected Result:** First question will display on window, allow user to move to next questions and previous questions
 - New Test 002 - Save a started test test
 - **Severity:** 1
 - **Instructions:** Once questions/answers have been filled out, student will click “Submit” when last question is reached
 - **Expected Result:** Test file will be saved in the same directory as binaries using format firstname_lastname
 - Existing Test 001 - Open a partially completed test
 - **Severity:** 2
 - **Instructions:** Enter first and last name in main window
 - **Expected Result:** System should detect that a test is already there, prompts if user wants to open old test or start over
- Boundary Value Test Cases

- New Test 001 - Saving test with same name as another person in class
 - **Severity:** 2
 - **Instructions:** Have two student accounts with the same names in each class save a test.
 - **Expected Result:** After firstname_lastname is saved, it should start appending incrementing numbers at the end, such as firstname_lastname1, 2, 3, 4, etc.
- New Test 002 - Save a completely blank (totally incomplete) test.
 - **Severity:** 3
 - **Instructions:** Open a test the instructor created, click "Next" until the last problem, "Submit" without having done anything else.
 - **Expected Result:** Should save test results as normal.
- Open Non-existing test - Try to open a test that isn't in the system/instructor never created
 - **Severity:** 2
 - **Instructions:** Type in any name of a test that doesn't exist in the file system
 - **Expected Result:** Error message should inform user that file does not exist.
- Equivalence Class Cases
 - *Not applicable for this requirement*

Requirement: Interaction between instructor client and student client

- Functionality Test Cases
 - Load Student Test 01 - Accessing a test from the student client
 - **Severity:** 2
 - **Instructions:** Open the student client. Enter your name, then press "next". Select an existing student test file, press "open".
 - **Expected Result:** The dialog should disappear and the program should load the file.
 - Load Student Test 02 - Accessing a student test from the instructor client to grade
 - **Severity:** 2
 - **Instructions:** Open the instructor client, then press "Grade Test". Select an existing test file through the open file dialog that appears. Press "Open".
 - **Expected Results:** The dialog should disappear and the program should load the file.
 - Compare Answers 01 - User viewing student answers and test answers from

student client

- **Severity: 2**
- **Instructions:** Open the student client, enter your name, press “next”, then use the dialog to open a student test. Press “Open”.
- **Expected Result:** The student answers should appear beside the actual answers from the test.
- Compare Answers 02 - User viewing student answers and test answers from instructor client
 - **Severity: 2**
 - **Instructions:** Open the instructor client, press “Grade Test”, and select a student test through the file dialog.
 - **Expected Result:** The student answers should appear beside the actual answers from the test.
- Student Load Test 01 - Load a test made by an instructor through the student client.
 - **Severity: 2**
 - **Instructions:** Open the student client, enter a name, press “next”, then select an instructor made test.
 - **Expected Result:** The client loads the test and displays the questions.
- Boundary Value Test Cases
 - Load Student Test 03 - User loading test instead of a student test in instructor client
 - **Severity: 2**
 - **Instructions:** Open the instructor client, press “Grade Test”, and select a regular test through the file dialog.
 - **Expected Result:** A dialog should appear informing you selected the wrong file type.
- *Equivalence Test Cases Not Applicable*

Requirement: The system should be storing confidential data on a secure file system.

- Functionality Test Cases
 - Security Test 001 - Account Security
 - **Severity: 2**
 - **Instructions:** The user tries to log into another users account
 - **Expected Result:** New window opens allowing user to create test questions/answers
 - Key Security Test 001 - Key Security Test
 - **Severity: 2**

- **Instructions:** Log in as a student and try to open a key
 - **Expected Result:** You should be denied access
 - Answers Security Test 001 - Regular Password Field
 - **Severity:** 2
 - **Instructions:** Log in as a student and try to open another student's answers
 - **Expected Result:** You should not be able to open them
 - Teacher Security Test 001 - Open Another's Test
 - **Severity:** 2
 - **Instructions:** Log in as a teacher and try to open another teachers test.
 - **Expected Result:** You should not be able to open them.
 - Teacher Security Test 002 - Open Another's Key
 - **Severity:** 2
 - **Instructions:** Log in as a teacher and try to open another teachers Key.
 - **Expected Result:** You should not be able to open them.
 - Teacher Security Test 003 - Open Another Teacher's Student's Answers
 - **Severity:** 2
 - **Instructions:** Log in as a teacher and try to open another teacher's student's answers.
 - **Expected Result:** You should not be able to open them.
- Boundary Value Test Cases
 - Account Security Test 001 - Null Password Field
 - **Severity:** 2
 - **Instructions:** Try to log into another users account by entering a null password
 - **Expected Result:** The current user is denied access to account
 - Account Security Test 002 - Long Password Field
 - **Severity:** 2
 - **Instructions:** Try to log into another users account by entering an extremely long password
 - **Expected Result:** The current user is denied access to account
 - Account Security Test 003 - Regular Password Field
 - **Severity:** 2
 - **Instructions:** Try to log into another users account by entering a normal length password
 - **Expected Result:** The current user is denied access to account
- *Equivalence Test Cases Not Applicable*

Requirement: The system should be reliable and function under all reasonable circumstances.

- Functionality Test Cases
 - Reliability Test 001 - General
 - **Severity:** 2
 - **Instructions:** Simply use the program in the way it was designed
 - **Expected Result:** The program should always run without crashing
- *Equivalence Test Cases Not Applicable*
- *Boundary Value Test Cases Not Applicable*

Requirement: Users should be able to use the system without having any training, and program should be very usable overall.

- Functionality Test Cases
 - Usability Test 001 - Complete Student Test
 - **Severity:** 3
 - **Instructions:** Open test file with the student client, complete the test, and save the test back to the file
 - **Expected Results:** User should be able to complete the test without having spent an inordinate amount of time figuring out how to record their answers using the student client software. Ideally, completion of the test takes less time than if test had been taken by hand.
 - Usability Test 002 - Instructor Test Grading
 - **Severity:** 3
 - **Instructions:** Open test file with the instructor client, grade the test, and record the test score
 - **Expected Results:** User should be able to grade the assignment while having spent a minimal amount of time dealing with the learning curve of the software. Ideally, completion of grading is less than if grading had been done manually/by hand.
 - Usability Test 003 - Instruct Test Creation
 - **Severity:** 3
 - **Instructions:** Open instructor client, create new test, and save the newly created test to a file to be distributed to the students in the course
 - **Expected Results:** The professor should be able to create the test without issue and without encountering a step in the process where the next step required to proceed was unclear to the user.
- Equivalence Class Cases

- *Not applicable for this requirement*
- Boundary Value Test Cases
 - *Not applicable for this requirement*