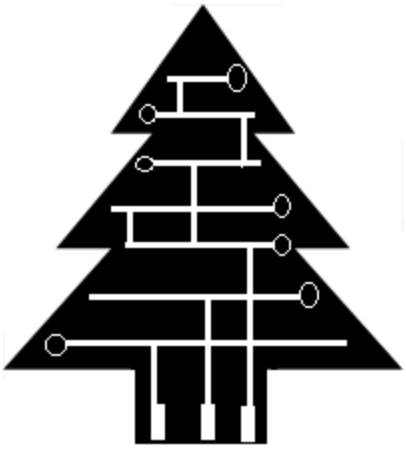

CO₂ SOFTWARE

Software Testing Plan

April 8th, 2018

Version 3

Sponsor

Dr. Andrew Richardson

Dr. Mariah Carbone

Mentor

Ana Paula Chaves Steinmacher

Team

Sam Beals, James Beasley,

Andrew Greene, Joseph Kelroy

Table of Contents

Introduction

Our team, CO2 Software, has been ask to design an application for Northern Arizona University, with the guidance of our sponsors Doctor Andrew Richardson and Doctor Mariah Carbone. The Richardson-Carbone Lab studies carbon cycling in forest ecosystems. In a nutshell, the research is used to understand the balance between carbon uptake (photosynthesis by plants) and release (respiration both by living and growing plants, as well as by microorganisms decomposing dead organic matter in the soil). The application they have asked us to design reads in information from the LI-840A gas analyzer that they use and then help make their data collection process easier. It does this by allowing metadata input for collections, letting the user easily view the collection sets, and letting the user get the collections out of the application.

This application will be tested extensively to ensure the highest quality product for our Sponsors.There will be three types of tests: unit tests, functionality tests and usability tests. The testing of our functionality and unit tests will be done by the team and the testing for usability will be done by random people, with tests that are laid out in this document. Most functionality and unit testing will be done by us showing the durability of each function based on everything the application should be able to handle. This will be done by using the application intentionally giving it many difficult things to handle. It will also be tested by people who have not used it before so it is not only us showing everything it can do but also by others navigating it differently. These users will be a group of five others from another group in capstone. Usability tests will be based on a time system and they are either pass or fail.

Though all functionality is very important some pages have more than others so some pages will not have as many tests. Since it is an application we will be basing tests around worse case scenarios with things that can happen in regular use with it. Also the tests done by people who have not used the application will be important for showing the usability of the application with the average user. These tests will be rigorous and well documented by the team to prove functionality for this application and if any are to fail will show the flaws to help us improve it.

Unit Testing

In this document, unit testing refers to the testing of the methods which perform the main features of the application. For each test, we specify which feature we are testing, how it will be tested, and what will be considered a correct outcome. All of our Unit Tests will be completed by entering in dummy data within the applications classes.

Home Screen

1. **Name:** Metadata Navigation Test
   Description: Starting from a clean start of the application, the user will navigate to the metadata screen using the “Start New Data Set” button. The metadata screen should contain no entered data.
   Expected outcome: Application navigates to the metadata screen with no data entered.

2. **Name:** File Directory Navigation Test
   Description: Starting from a clean start of the application, the user will navigate to the file directory using the “File Directory” button. The file directory screen should be presented with all data sets properly displayed in order of recorded date.
   Expected outcome: Application navigates to the file directory and properly displays all saved data sets.

Metadata Screen

1. **Name:** Field Validation Test
   Description: Certain Fields only allow certain characters. The user will be asked to enter in Operator Name, Site Name, and Sample ID. These are all required fields, so they must be filled in. They must also only allow certain characters. Ie, if the user enters a “-” in operator name, then the conditions are not met. The Field Validation is handled by a TextWatcher which is an active listener. In the main flow, the user will tap on each field, and enter only characters that are allowed. If all conditions are met, the “Finish” button is enabled, and the main flow is complet. In the alternative flow, the user will re-enter new information, to check if the TextWatcher is still watching. The user will also try to type in invalid characters. The conditions will not be met, and the “Finish” button will be

disabled. This flow applies to all fields, required and optional: Operator Name, Site Name, Sample ID, Temperature, Comments, Time, Date, Elevation, Longitude, and Latitude. Our tests will be composed of dummy data and real data. In the dummy data tests, we will be trying to fail the tests. There will be more real data tests, as this will provide better results on how the interface reacts.
Expected outcome: The first three required fields, Operator Name, Site Name, Sample ID, only allow letters and numbers. Temperature only allows numbers. Comments only allows letters, numbers, and select symbols. The “Finish” button will be disabled until all conditions are met.

2. **Name:** Camera Test
   Description: Main Flow: The user can take a picture and it will be saved correctly and relative to the pertaining data set. Alternative Flow: No amount of button presses will get the user stuck outside of the main app, or be able to break the camera functionality. The user can take as many pictures as they like, but only the most recent one is saved, as well as shown in the preview box.
   Expected outcome: When the Camera Button is pressed, the camera API will open. The Camera takes a picture properly, and displays a preview on the Metadata Screen.

3. **Name:** Home Screen Navigation Test
   Description: The user will navigate to the Home screen using the “Cancel” button. All data currently entered in the MetaData screen will be erased. When the user clicks on the “Start a New Dataset” button again, all of the fields will be empty.
   Expected outcome: Application navigates back to the Home Screen

4. **Name:** GPS Unit Test
   Description: Main Flow: If the user is outside, and there is nothing obstructing the GPS signal, GPS will take about 10 seconds to load the values. It is estimated that the Metadata screen will be filled out by a user in about 30 seconds. So, by time the user is finished filling out the other fields, the GPS fields should be filled in automatically. Alternative Flow: No signal is found, and the User may enter the fields manually. (The fields will still be checked by the Field Validation Test) The fields also must be cross referenced with Constants. Ie, if the test is at the SICCS building, then the pulled GPS values should match the actual values of the the building itself.
   Expected outcome: Fields ‘Elevation’, ‘Longitude’, and ‘Latitude’ are all filled in

with correct Values. The fields must be filled in within a time frame of 30 seconds.

5. **Name:** Permission Request Test
   Description: The metadata screen must ask for location services to get a GPS signal. It must also ask for Camera Services Main Flow: During standard operation of the app, the permissions will already be enabled, meaning permissions will not be asked for. Alternative Flow: Upon first installation or reinstallation of the app, the application must ask for Permission to use Location Services and the Camera.
   Expected outcome: Application asks for permission for Camera and Location Services.

Graph Screen

1. **Name:** Disabled Finalize Button
   Description: After entering metadata and navigating to the graph screen, the “Finalize” button should be disabled, and pressing it should do nothing. Only after pressing the “Start Logging” button, and then pressing the “Stop Logging” button should the “Finalize” button be activated.
   Expected outcome: Before a data set has been recorded, the user should be unable able to press the “Finalize” button.

2. **Name:** Duration Test
   Description: After navigating to the graph screen, the gas analyzer should be connected to the device, and the “Start Logging” button should be pressed. One hour of time should elapse before the “Stop Logging” is pressed. The data reading should be viewable on the view screen and no anomalies should be present.
   Expected outcome: A single data reading should be recorded for twenty minutes without any application crashes or unintended side effects.

3. **Name:** Streaming Test
   Description: A data log will be recorded with a duration of at least ten minutes. The log will then be emailed out of the application, and examined for anomalies, or repeat data points. If a miscommunication is encountered with the instrument, then the alternative flow would be for the serial reader to disregard that data point, and not graph it or include it in the saving data set.

Expected outcome: A data stream from the device with a corresponding data log that accurately represents the information collected.

File Directory

1. **Name:** File Deletion Test
   Description: The app will have varying amounts of files added and deleted,testing if there is any underlying bugs that result when certain amounts of files are selected to be deleted.
   Expected outcome: User-Selected Files will be deleted from the file directory display as well the app's internal storage whenever the user presses the delete button. The effects of this function should be immediate and permanent and applicable to any non-zero number of files. Only the files that have been selected and are currently highlighted green should be deleted, and no files should be selected after the deletion is performed.

2. **Name:** File Selection Functionality
   Description: When multiple files are selected, the "View" button should be disabled, while the delete and email buttons are still able to be used. Whenever there are no files selected the delete, view, and email buttons should be unable to be pressed by the user. The filter button should remain unaffected by the amount of files selected at any point.
   Expected outcome: The user should be able to select any number of files from any parts of the file list (and be able to see which ones are currently selected), and then perform the available functions. The effects of the operations applied to the files should be recognizable and predictable from the user's perspective.

3. **Name:** Email Functionality
   Description: When the user is on the File Directory screen, they can highlight any amount of files to be emailed. We will have varying amounts of files selected and chosen to be emailed out of the application. The email button must be pressable whenever any files are selected, and the email clients available must be choosable every time.
   Expected outcome: The user should be able to select any number of files and be able to attach all associated into the email client that is chosen. When using the file directory, the "Email" Button must only be selectable when at least one file has been selected. Additionally, when the user selects the email option, they should be prompted to choose the email client of their choice based on the apps on their device.

4. **Name:** Filter Functionality
   Description: The test will consist of randomly generated strings of characters being entered into the field in order to see the results. This includes characters that are symbols, i.e. not just numbers and letters.This will allow us to identify any bugs that could be caused by the user inputting unexpected characters,as well as finding any cases where the filter fails include or exclude the proper files.
   Expected outcome: When the user enters an input of characters in the field for the filter, the list will be made to show only files whose names contain the input. When the user presses filter without anything or just spaces in the field, the button will not perform any action. When a valid input is entered, the undo filter button is enabled. When pressed, the undo filter will display the entire list again as it was before the initial filter.

View Screen

1. **Name:** Statistical Test
   Description: All statistics have a function for calculating that statistic. We can give each one of these statistics a graph input of a blank graph, a regular graph, and a very odd graph and all statistics should either be calculated or return a 0 on an error.
   Expected outcome: The statistical functions (r squared, slope, regression line, and standard error) will all be calculated for a specific graph and these outputs should be the same as us calculating them by hand.
   Alternative Flow: With null data sets all functions will return 0.

2. **Name:** Valid Entry Test
   Description: All other entries that are not in that form will not be calculated. If it is in the correct form a graph will be able to be changed.
   Expected outcome: The function that handles valid entries for changing a graph will only allow an entry in the form of x,y.
   Alternative Flow: when a non valid entry the function will call a pop up alerting the user.

3. **Name:** Chop Graph Test
   Description: This function will be tested by looking at what is returned after what is expected to be changed is input. Since the valid entry function checks if the graph can be cut in that way this function will only check that the graph is cut

correctly.
Expected outcome: A new graph will be returned from the chopString function.

Integration Testing

In this document, integration testing refers to the successful mapping of our functions to the user interface. For each screen of the application, we will dictate a series of actions to be taken in order to test if the features have correctly been assigned to the corresponding parts of the user interface.

Home Screen

1. **Name:** New Data Button Mapping
   Description: All navigation in our application is done with button presses. When the “Start New Data Set” button is pressed, its corresponding method should be called from within the home screen java object.
   Expected outcome: Corresponding navigation method is called when button is pressed.

2. **Name:** File Directory Button Mapping
   Description: All navigation in our application is done with button presses. When the “File Directory” button is pressed, its corresponding method should be called from within the home screen java object.
   Expected outcome: Corresponding navigation method is called when button is pressed.

Metadata Screen

1. **Name:** Data Passing Test
   Description: This test applies to all fields, which are EditTexts: Operator Name, Site Name, Sample ID, Temperature, Comments, Time, Date, Elevation, Longitude, and Latitude. Main Flow: Every field is filled in with real data. This tests that the fields are formatted correctly when they are seen in the CSV file or view screen. The Field Validation Test will make sure that any “illegal” characters cannot be passed, and check for strings that are too long. All strings will be identical when seen in the view screen and CSV File. This test will be checking for accuracy and consistency. Alternative Flow: Certain fields are Null, and how they are handled. Every field must be tested for null values, as that is a

very common occurrence of bugs and crashes. If a value is null, then the CSV file and view screen will display “NA”.
Expected outcome: When the “Finish” button is pressed, the function “goGraphScreen” sends all metadata as strings to the next screen. All of the data must be identical when it is viewed on the view screen, as well as the CSV File. Null values (empty fields) will default to “NA”.

2. **Name:** Image Passing Test
Description: This test is similar to the Data passing test, but it has more components. In this test, we are passing data, and not a string. The image should not lose image quality due to compression. When the Camera “Intent” is finished, the data for the image will be compressed. Main Flow: When the “Finish” button is pressed, the function “goGraphScreen” sends the image data to the View Screen. The image should be saved correctly with it’s respective data set. It should also have no issues being exported via email (See Email Data Transference Test). Alternative Flow: No picture is taken. The user is aware that no picture has been taken. View Screen and Emailing files out of the app should not be affected.
Expected outcome: When the image is passed to the view screen or file directory, it will be identical to the image on the metadata screen.

Graph Screen

1. **Name:** Graph Switching Buttons
Description: In order to switch between each of the four graphs, methods in the graph screen java object are called to manipulate elements of the UI. We must ensure that the correct UI modification methods are called when the buttons are pressed to switch to the correct graph.
Expected outcome: UI modification methods are called when their corresponding buttons are pressed.

2. **Name:** Start / Stop Logging Button Mapping
Description: All actions pertaining to the graphs are handled by the graph manager, whose methods are called by the graph screen java object. When the start and stop logging buttons are pressed, the graph screen java object must then make the correct method calls to the graph manager object. If the application is not currently connected to the gas analyzer, then the alternative flow would be the recording starting, but disregarding all data points until

communication is established.
Expected outcome: The correct graph manager methods are called when buttons are pressed.

3. **Name:** Finalize Button Mapping
   Expected outcome: The correct graph screen methods are called and the correct graph manager methods are called. If the finalize button is currently disabled, then the alternative flow would be no event occuring.
   Description: When the finalize button is pressed, two things happen: the graph screen object

File Directory

1. **Name:** View Screen Data Transference
   Description: When the view button is pressed with varying files selected, the view screen should be display the correct data every time. It must display the image and graph associated if they exist in the same data set. The view option should only be available when exactly one file is selected.
   Expected outcome: When a file is selected to be viewed in the view screen, the files that is currently selected has its name saved by the app. It then takes the name of the metadata file, along with the names of the graph and image files, and sends them to the view screen. This allows the screen to know which data to display to the user.

2. **Name:** Email Data Transference
   Description: We will have varying amounts of files selected and chosen to be emailed. The files that are attached in the email must be confirmed to be the ones chosen in the application prior. The User should receive the email at the address specified with all of files properly attached with no data loss or corruption.
   Expected outcome: When any amount of files is selected, the clients will have the files attached into an email. The files attached should always be the files that are associated with the files selected in the application. If there is not an image file associated with any of the selected data sets, then the files will be attached without any bugs or crashes. The email should be sent with all attachments chosen to the address specified.

View Screen

1. **Name:** Navigation To Screen
   Description: We will enter in a series of different information including all fields and not including all fields and taking a one second graph and a twenty minute graph to ensure the screen does not have trouble pulling up a maximum amount of data and a minimum amount of data. Also all graphs, metadata, and the image will be looked at to ensure the correct information is passed to this screen.
   Expected outcome: Regardless of the information entered or graphs taken, big or small the view screen will pull up all corresponding information without crashing.
   Alternative Flow: If there is any error in pulling up the file information the screen will present null data and alert the user the file is corrupt.

2. **Name:** Sub Graph Test
   Description: A subgraph will be applied to a graph and the subgraph file will be looked at on the view screen to make sure all information is correct.
   Expected outcome: When a graph is changed, with the subgraph function, the file will get overwritten and will be the correctly cut subgraph to a new file.

Usability Testing

In this document, usability testing refers to the user's ability to locate and utilize the features of the application. This ability will be measured numerically for each test. Each test will have conditions that the application must meet in order to be considered a success. Our test will be done by a first time users of the application, who have no prior knowledge of our software. This will ensure that if our tests are successful, it shows that the application is intuitive enough to be understood by the average end user.

Home Screen

1. **Name:** Navigation
   Description: This test will be observed throughout the duration of the application. The user should quickly identify where he wants to go as well as where he/she is in the application. The user should not be able to access different screens in an order that does not follow the flow diagram. If the user wants to view a data set, then it should take less than 10 seconds to get to the

respective screen.
Expected outcome: The user is able to get to any screen with only 2 button presses. This process should only take 10 seconds.

Metadata Screen

1. **Name:** MetaData Entry
   Description: The user can easily select which field to fill in, and type with ease. If the user does not know why the “Finish” button is disabled, he will be prompted with a Toast message explaining why. For example, “Operator Name is a Required Field”. This test will provide feedback on whether or not the hints and messages are accomplishing the expected outcome.
   Expected outcome: The user is able easily figure Required Fields within 5 seconds. User Should know which fields are optional and which are required. The user should be able to fill out the required fields in under 30 seconds. Filling out all fields should take less than 2 minutes.

2. **Name:** Camera Ease of Use
   Description: The goal here is to keep the process of the camera as simple as possible. The user should not be able to get stuck outside of the application, or crash the application. All previously filled in metadata should stay the same as well.
   Expected outcome: The user is able take a picture in 10 seconds. All of the necessary prompts are apparent to the user, and vice versa no unnecessary prompts are available.

3. **Name:** GPS Information Test
   Description: The Fields Longitude, Latitude, and Elevation are all filled in automatically. However, there is a chance that a GPS signal cannot be found. In this instance, the User can enter the fields manually. The user should be aware of whether or not the signal has been found.
   Expected outcome: User should know that the Fields Longitude, Latitude, and Elevation are trying to fill in automatically. However the Option is Available for manual entry.

Graph Screen

1. **Name:** Data Logging
   Description: After entering metadata, the user will be instructed to record a data set to the device using the gas analyzer. This will require the user to connect the gas analyzer to the device, press the “Start Logging” button, wait some time, press the “Stop Logging” button, and then press the “Finalize” button. This test will be failed if a user is not able to successfully navigate to the file directory with a newly recorded data set.
   Expected outcome: The user is able to successfully connect the gas analyzer, record a data set, and finalize that data set.

2. **Name:** Graph Switching
   Description: After entering metadata, the user will be presented with the graph screen, where they will be instructed to view all four values being streamed in from the gas analyzer.
   Expected outcome: The user is able to successfully switch between all four different graphs while data is being streamed out.

File Directory

1. **Name:** File Emailing
   Description: We will ask the user to access the file directory screen. From there they will be told to select exactly three separate files and to email them to an address of their choice. This will test how intuitive the emailing functionality is.
   Expected outcome: The user is able to successfully email three different data sets out of the application

2. **Name:** File Filtering/ Unfiltering
   Description: The user will be told to filter by the first three letters of the current month. They will then be told to unfilter the file directory. At that point, the file directory should look exactly as it did when the user first opened the window.
   Expected outcome: The user is able to easily filter the file directory by the string they are told to enter.

3. **Name:** File Deletion
   Description: The user will be told to delete every file from a specific day among the file directory. They should be able to highlight the files and delete them without making any mistakes, e.g. accidental deletion of other files.
   Expected outcome: The user is able to select files that they are told to delete without any accidents or confusion.

View Screen

1. **Name:** Apply New Graph
   Expected outcome: A user can apply changes to a graph based off given input within 10 seconds.
   Description: A user will be asked to change a graph based on a x start and end point. If the user is not able to change the graph in 10 seconds this test will be failed.

Conclusion

Ensuring the integrity of our software is just as important as implementing all required features. Through the use of this document, we hope to guarantee that all application features are work as intended, and that the application uses an intuitive user interface so that it can be easily adopted by other ecologist around the globe. The tests in this document describe how the team can properly test features, integration of those features, and the interface which allows the end user to access those features, covering the three main areas of software testing. After testing is complete, and corresponding adjustments have been made to the application interface and functionality, we will be able to confidently deliver the final application to the client, on time.