**Rural Cultivation & Atmospheric Emulation Application (RCAEA) URS document**

|  |  |
| --- | --- |
| **Document:** | Testcase |
|  |  |
| **Authors:** | Damianidis, Zisis  Dyer, Richard  Khan, Al-Mohaiminul Islam |
|  | Khan, Raima |
|  | Hadzhiev, Tsanko |
|  | Hadzhinikolov, Mihail |
|  |  |
|  |  |
| **Creation Date:** | 29/09/16 |
| **Last Revised:** | 09/10/16 |
| **Group Name:** | Tanks & Co.™ |
| **Version:** | 1 |

*16*

Table of Contents

[Introduction 3](#_Toc463382346)

[Objectives 3](#_Toc463382347)

[Scope 4](#_Toc463382348)

[Functions to be tested: 4](#_Toc463382349)

[Simulating growth of crops 5](#_Toc463382350)

[Selecting start date and end date 5](#_Toc463382351)

[Requirements tracebility matrix 6](#_Toc463382352)

[GUI 6](#_Toc463382352)

[Requirements for Testing 7](#_Toc463382353)

[Deliverables 15](#_Toc463382354)

# Introduction

This document will outline the testing procedures for RCAEA Project to be developed by Tanks & Co.™ The application allows the user to simulate cultivating specific crop(s) in an area of land during a certain length of time. By using this application they can determine when, where, and what crops to place in a specified piece of land. It will help the user make a cultivation plan for a certain area of land based on real land data. It considers regions factors such as weather whereby the user can select which outdoor agricultural crops to place in an area. The simulation will use real data on the crop and simulate its growth based on external and internal determinate factors. RCAEA will take all these factors into account and determine an estimated cost and production outcome. Data will be saved in a file which the user can load or keep for their own records.

# Objectives

Describe the objectives supported by the Master Test Plan, eg., defining tasks and responsibilities, vehicle for communication, document to be used as a service level agreement, etc.

This document specifies the functions that will be tested for the RCAEA. This document has the following objectives.

* To define the functionalities that will be tested.
* Give details of testing procedures.
* Define elements of testing activities.
* Defining the testing strategies to be implemented
* Define deliverable dates for the testing results.

# 

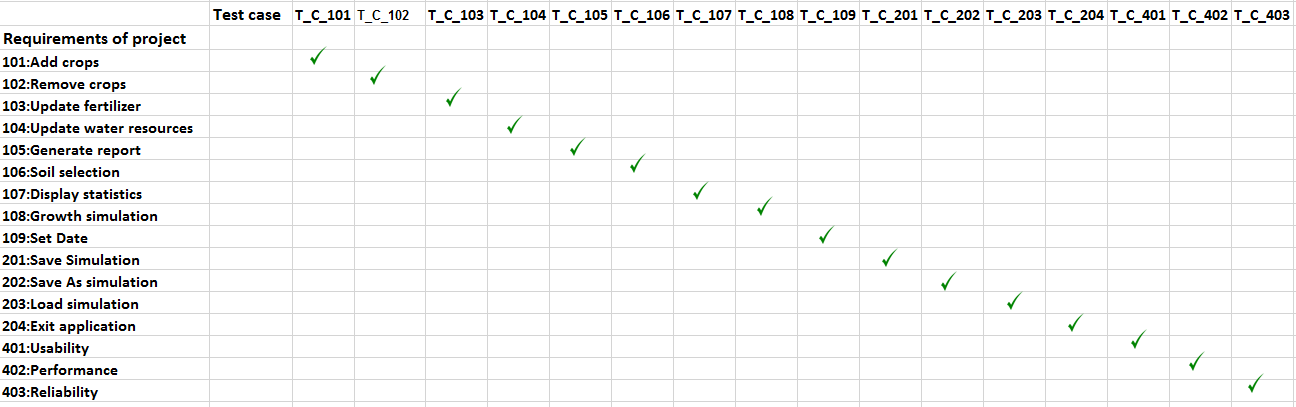
# 

# Scope

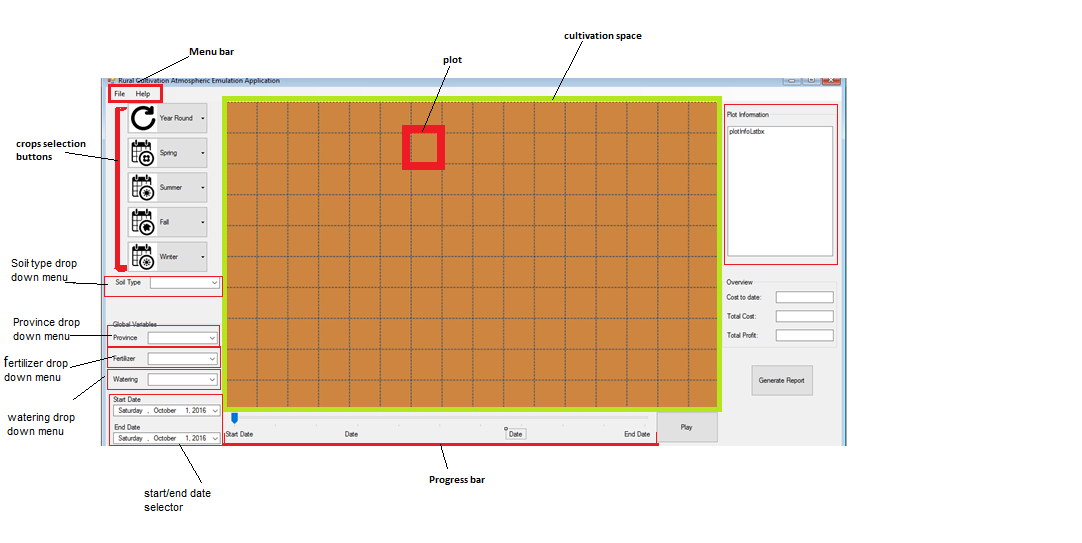
## Functions to be tested:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Function | Objective | Description |
| T\_C\_101 | Adding Crops | * Ensure crop is added to plot without error * Crop is added to plot within 1500 milliseconds * Exception message must be displayed in case of error. | Adding Crop testing tests the system’s ability to fill a plot with a crops characteristic. |
| T\_C\_102 | Removing Crops | * Ensure crop is removed from selected plot without error * Exception message must be displayed in case of an error without crashing | Removing Crop testing tests the system’s ability to remove a crop and its characteristics from a plot while also reseting the plot’s status. |
| T\_C\_103 | Updating fertilizers to crops | * Ensure fertilizer is updated correctly * Information panel must be updated accordingly | Updating fertilizers to crops testing tests the system’s ability to correctly update the quantity of the global variables. |
| T\_C\_104 | Updating water resources | * Ensure water resources are applied correctly to all cultivated crops * Information panel must be updated accordingly | Updating water resources testing tests the system’s ability to correctly update the quantity of the global variables on all cultivated crops. |
| T\_C\_105 | Generate Report | * Ensure the report contains all the information about the current simulation * Exception message must be displayed in case of an error without crashing * Report is generated within 4000 milliseconds | Report generation testing tests the system’s ability to generate a description of all the crops involved in the simulation and the overall costs. |
| T\_C\_106 | Soil Selection | * Ensure that the soil type can be correctly selected and updated for every plot * Ensure that the “default” soil type and characteristic is based on the region * Information panel must be updated accordingly | Soil Selection testing tests the system’s ability to update the soil properties of the selected plot as well as the information panel. |
| T\_C\_107 | Display Statistics | * Ensure the statistics of the selected plot are displayed correctly in the information panel | Displays Statistics testing tests the system’s ability to update the information panel depending on the specific plot that is selected. |
| T\_C\_108 | Simulating growth of crops | * Ensure the simulation runs smoothly and without crashing * Ensure that the properties of the soil and global variables cannot be changed if the simulation is not paused * Simulation must be running at 1000 milliseconds per simulated week * Ensure that the simulation ends at the allocated end date and that the correct number of weeks have been simulated * Ensure the cultivation field is updated correctly if the manual timeline scroll is used * “Play” button must change to “Stop” when the simulation is initiated * “Play” button changes to “Re-play” when simulation finishes. | Simulating growth of crops testing tests the simulation run of the cultivated crops while factoring in all the external and internal factors.  It also tests the manual timeline scroll that is used to change the current progression of the simulation, which allows for quick editing of the land when simulation is paused. Otherwise, crops cannot be changed or updated and global variables remain “locked” during simulation runtime.  The growth simulation must run only until the specified end date.  The system must correctly calculate the number of weeks between the start date and the end date and update the information panel accordingly during the simulation. |
| T\_C\_109 | Selecting start date and end date | * Ensures that the selected start and end dates are updated on the timeline. * Exception message must be displayed in case the dates are outside of the possible time scope. | Start and end date selection testing test the system’s ability to update the timeline with the selected dates and must check that the selected end date must be longer than 3 months and less than 3 years. |
| T\_C\_201 | Saving a simulation | * Ensure that the simulation is saved into the database and the correct values are overwritten. | Saving a simulation testing tests the system’s ability to store the current simulation in the database by overwriting an existing saved simulation. All related variables, attributes and conditions that have been changed must be overwritten. |
| T\_C\_202 | Save as simulation | * Simulation is saved into the database * If the same name is found in the database, must ask if User want to overwrite the file of the same name. * If application, cannot connect to the database must prompt user of error, and continue without crashing. | The working simulation will be saved into the database. All attributes and variables involved must be stored so that it is easily loaded when needed. |
| T\_C\_203 | Load simulation | * Must prompt user if they want to save if current simulation is unsaved * Must load the correct previously saved data into the working simulation. * User needs to be able to find desired project without difficulty. | Simulation will be loaded from the database with all the previously placed crops, in the correct position with the correct data, and variable set for the saved simulation. |
| T\_C\_204 | Exit application | * The application must prompt user to save if working simulation is unsaved. * Application closes without delay. | It is an essential function of any application, to be able to close process and free it from the computer’s memory. |

# Requirement traceability matrix



## GUI



# Requirements for Testing

The underlying items are the identified targets for testing, every item below will be tested in the proposed manner.

**Adding crops:**

ID:T\_C\_100

**100.1 Adding Cauliflower on empty plot**

**Initialization status:**Plot at position (2,3) is empty .And no simulation is running.

**Test steps:**

1. User clicks on year round crops
2. List of crops that can grow all round year will be displayed.
3. User clicks cauliflower.
4. System shows picture and name “Cauliflower” on split button as currently selected crop
5. User clicks  on plot position (2,3) where he wants  to grow cauliflower.

**Test result:**

The system will show cauliflower cultivated on plot position (2,3).

**100.2 Cauliflower added to already cultivated plot exception**

**Initialization status:**Plot at position (2,3) is already cultivated. And no simulation is running.

**Test steps:**

1. User clicks on year round crops
2. List of crops that can grow all round year will be displayed.
3. User clicks potatoes.
4. System shows picture and name “potatoes” on split button as currently selected crop
5. User clicks on plot position (2,3), which is already cultivated with cauliflower, where he wants to grow potatoes.

**Test results:**

System will display an exception message “Selected plot is cultivated.”

**Removing crops**

Id: T\_C\_102

**102.1Removing A Crop From Plot**

**Initialization status:** The screen will have plot position 3,4 and plot position 3,5 filled with a cauliflower crop. And no simulation is running.

**Test steps:**

1. User will right click on first plot position 3,4.
2. System will show the right clicked menu with ‘delete’, and ‘delete all’ options.
3. User clicks ‘delete’ option from the right click menu.

**Test results:**

System will remove cauliflower crop from plot position 3,4. System will remove statistics info and will change status of plot as empty plot.

**102.2 Empty plot selected for removing crops**

**Initialization status:** Plot at position (3,4) in cultivation space is empty. And no simulation is running.

**Test steps:**

1. User will right click on plot at position (3,4) in cultivation space.

**Test result:**

System will show the right clicked menu with disabled ‘delete’, and ‘delete all’ options .

**102.3 Removing a group of same crop type cultivated on different plots.**

**Initialization status:** The screen will have plot positions (3,4),(3,5),(4,4)are filled with the cauliflower crop. Plot position(5,6),( 3,6) is filled with the Cauliflower Crop. And no simulation is running.

**Test steps:**

1. User will right click on first plot position 3,4.
2. System will show the right clicked menu with ‘delete’, and ‘delete all’ options.
3. User clicks ‘delete all’ option from the right click menu.

**Test result:**

System will remove the all cauliflower crops and change the status of the plots with the position (3,4),(3,5),(4,4) to empty. The Plot positions (5,6), and (3,6) will remain cultivated.

**Updating fertilizers to crops**

Id: T\_C\_103

**103.1Updating fertilizer to all cultivated crops at the moment**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click on the fertilizer drop down menu.
2. System will display the fertilizer drop down options ‘minimal’, ‘sufficient’, ‘plentiful’.
3. User will click on ‘minimal’ as the amount of fertilizer that he wants to add from the drop down options.
4. System changes the fertilizer attribute to ‘minimal’.
5. System will change the fertilizer option to ‘minimal’.

**Test results:**

System will update the fertilizer variable in the system. The system will also update the amount of fertilizer accordingly in the information panel.

### **Updating water resources**

Id: T\_C\_104

**104.1 Updating water resources to all cultivated crops at the moment**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click on the watering drop down menu.
2. System will display the watering drop down menu ‘low, ‘sufficient’, ‘plentiful’.
3. User will click on sufficient as amount of water that he wants to add from the drop down options.
4. System changes the watering attribute to ‘minimal’.
5. System will change the watering drop down option to ‘minimal’.

**Test results:**

User can change the Watering options and the system will process this request without errors.

**Generate report**

Id: T\_C\_105

**105.1Generating report for current simulation**

**Initialization status:** The screen must have one plot cultivated (Plot (2,2) seeded with peas, on “clay” soil, in Noord Brabant with fertilizer and water set to max). There’s a valid start (25/02/2017) and end date (25/02/2018) set in.And no simulation is running.

**Test steps:**

1. User will click the Generate Report button.
2. System will open new form window with the report.

**Test results:**

The user will be able to view a page that displays an overview of all the statistics from the simulation. It will show that Plot (2,2) is seeded with peas, on “clay” soil, in Noord Brabant with fertilizer and water set to max). There start date is (25/02/2017) and end date is (25/02/2018).

**105.2 Report button is clicked before any cultivation**

**Initialization status:** All plots in cultivation space are empty at the moment. And no simulation is running.

**Test steps:**

1. User clicks generate button on right side of main form (according to users’ point of view)

**Test result:**

System will show message “Please cultivate any crop before generating report”.

**Soil Selection**

Id:T\_C\_106

**106.1 Selecting soil of currently selected plot**

**Initialization status:** Plots can be either cultivated or they can be empty. And no simulation is running.

**Test steps:**

1. User clicks  on plot position (2,3) from cultivation space
2. System will display soil type of selected plot in soil type drop down box.
3. User clicks on drop down arrow on soil type drop down box.
4. System Displays soil type options.
5. User selects “Fertile” option from the drop down box options.

**Test results:**

System will set “Fertile” as soil type of plot at position (2,3). Information panel will update soil type of this plot.

**Display statistics**

Id: T\_C\_107

**107.1 Displaying statistics for currently selected plot**

**Initialization status:** The screen must have one plot at position (1,2), cultivated with crops. And no simulation is running.

**Test steps:**

1. User selects plot at position (1,2 ) column from cultivation space.

**Test results:**

System displays statistics for the plot at position (1,2) in the plot information panel.

**107.2User is selecting empty plot**

**Initialization status:** The screen must have one plot at position (2,2), cultivated with crops. And no simulation is running.

**Test steps:**

1. User selects plot at position(2,2), which is an empty plot from cultivation space.

Test results:

Information panel will be displayed empty.

**Simulating growth of crops:**

Id: T\_C\_108

**108.1 Using the play button**

**Initialization status:** Plot position (1,1) is filled with a cauliflower, the start date 12/12/2017 and end date 03/04/2018 has been entered.

**Test steps:**

1. User clicks the play button.
2. System starts the simulation.
3. System disables crops selection buttons and watering/fertilizer/soil selection dropdown.
4. System changes the start button to a stop button.
5. System runs the simulation at rate of 1 second/per week.
6. System updates progress bar based on time passed.
7. User clicks the stop button at the halfway point.
8. System pauses the simulation.
9. System enables crop selection.
10. User adds cauliflowers to position (1,2) crops.
11. User clicks the play button.
12. System continues the simulation from halfway point.
13. System finishes the simulation.
14. System changes start button to “restart” button.
15. System enable crop selecting buttons and watering/fertilizer/soil selection dropdown.
16. User drags progress bar to 06/02/2017.
17. System only displays on plot 1,1 a cauliflower crop.
18. User drags the progress bar to 20/02/2016.

**Test results:**

System will have started the simulation and ran without errors. The user should have paused the simulation edited the land and continue the simulation without errors. The end result is position (1,1) will be cultivated with a cauliflowers and plot position (1,2) will be cultivated with a cauliflower.

**Selecting start and end date**

Id:T\_C\_109

**109.1 Selecting start date and end date for simulation**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click start date selector.
2. System displays small calendar with possible dates.
3. User selects a 24/07/2017 as start date.
4. User will click end date selector.
5. System displays small calendar with possible dates.
6. User selects a 24/07/2018 as end date.

**Test results:**

System will show the selected date into the start date field and selected end date into the end date field. System will show 24/07/2017 on start of progress bar as start date.24/07/2018 will be displayed at end of progress bar as end date.

**109.2 Selecting end date less than 3 months**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click start date selector.
2. System displays small calendar with possible dates.
3. User selects a 24/07/2017 as start date.
4. User will click end date selector.
5. System displays small calendar with possible dates.
6. User selects a 24/08/2017 as end date.

**Test results:**

System will show message ”End date should not be less than 3 month from start date “

**109.3 Selecting end date more than 3 years**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click start date selector.
2. System displays small calendar with possible dates.
3. User selects a 24/07/2017 as start date.
4. User will click end date selector.
5. System displays small calendar with possible dates.
6. User selects a 24/07/2020 as end date.

**Test results:**

System will show message ”End date should not be more than 3 years from start date “

**109.4 Selecting end date less than start date**

**Initialization status:** No simulation is running.

**Test steps:**

1. User will click start date selector.
2. System displays small calendar with possible dates.
3. User selects a 24/07/2018 as start date.
4. User will click end date selector.
5. System displays small calendar with possible dates.
6. User selects a 24/07/2017 as end date.

**Test results:**

System will show message ”End date should not be less than start date “

**Saving simulation**

Id:T\_U\_201

**201.1 Saving a unsaved simulation**

**Initialization status:** No simulation is running.

**Test steps:**

1.    User clicks on “File” from menu bar.

2.    System displays ”File”  options.

3.    User selects “Save” option.

4.    System pop up a new window asking “File name”

5.    User will enter “saveTesting” on the field of “File name”

6.    User will click on “Save” button

**Test results:**

System will save simulation and all related variables and conditions into the database for later use.Database will be updated with new values

**201.2 Saving a saved simulation**

**Initialization status:** No simulation is running.

**Test steps:**

1.    User clicks on “File” from menu bar.

2.    System displays ”File”  options.

3.    User selects “Save” option.

4.    System will display a pop-up message of successful saving

**Test results:**

1. System will save changes to already saved simulation and all related variables and conditions into the database for later use.
2. Database will be updated with new values

**Save As simulation**

Id: T\_C\_202

**202.1Save as simulation**

**Initialization status:** No simulation is running.

**Test steps:**

1.    User clicks on “File” from menu bar.

2.    System displays “File” options.

3.    User click on “Save as” options

4.    System pop up a new window asking “File name” and “Save as type”

5.    User will enter “testing” on the field of “File name”

6.    User will click on “Save as type”

7.    System will show scroll down menu

8.    User will select “Text Documents (\*.txt)”

9.    User will click on “Save” button

**Test results:**

System brings up the list from database and system saves data and simulation in the database.

**Loading a simulation**

Id: T\_C\_203

Id: T\_C\_203

**Initialization status:** The User has started on a new project. And no simulation is running.

**203.1 Loading a simulation**

Test steps:

1. User clicks on File from menu bar.
2. System displays DropDown menu with options.
3. User clicks on “Load” option.
4. System displays a new windows form which has a list of all the saved simulations in the database.
5. Users selects  11.08.2016 - 22.12.2016 from the list with saved Simulations.
6. System prompts user if he would like to save his current work as it will be overwritten by the loaded simulation.
7. User selects no.
8. System loads saved simulation into the current working simulation.

**Test result**:

System loads and displays simulation “11.08.2016- 22.12.2016”.

**Exit application**

Id: T\_C\_204

**Initialization status:** The User has cultivated plot (1,1) with a cauliflower, and has not saved the new project.

**204.1 Exiting saved application**

Test steps:

1. User clicks the “X” button located on the top right corner of the application.
2. Systems checks if user has unsaved data.
3. System prompts user if he would like to save unsaved data giving the user ‘yes’ or ‘no’  options.
4. User clicks yes.
5. System displays the save dialog.
6. User saves the project as “Test\_Exit”.
7. System closes application.

**Test result** :System closes the application. Checking the database “Test\_Exit” will be the most recently saved project.

**204.2 Exiting unsaved application**

Test steps:

1. User clicks the “X” button located on the top right corner of the application.
2. Systems checks if user has unsaved data.
3. System prompts users asking if he wants to quit the application.
4. User clicks  “Save”
5. System saves the current simulation into the database.
6. System closes the application.

**Test result** : System saves the unsaved simulation and closes the application.

**204.3 Exiting unsaved application**

Test steps:

1. User clicks the “X” button located on the top right corner of the application.
2. Systems checks if user has unsaved data.
3. System prompts users asking if he wants to quit the application.
4. User clicks  “Quit”
5. System closes the application.

**Test result** : System closes the application without saving current simulation.

**Platform Compatibility**

Id: T\_C\_301

**301.1 Platform Compatibility Test**

**Initialization status:** application has already been installed on windows operating system and “windows Task Manager” running

Test steps:

1. User will perform the following tasks 5 times to confirm platform compatibility
   1. User will click on application “.exe” file
   2. System open the application
   3. User will observe application running details for 120 seconds from “windows Task Manager”
   4. User will exit the application

**Test result** :

1. Application runs on windows normally without any crash each time out of 5 times

**Performance**

Id: T\_C\_302

**302.1 Performance Test**

**Precondition:**”testLoad” is a saved file , where every plot is cultivated.

**Initialization status:** application is running

Test steps:

1. User will click on “File” from the navigation menu
2. System will display the options under “File”
3. User will click on “Load”
4. System will pop-up a new window
5. User will select “testLoad” and click on open
6. System will load “testLoad”

**Test result** :

1. Application will load the selected simulation within 3 seconds

**Usability**

Id: T\_C\_303

**Initialization status:** 10 Testers are available for Guerilla testing.

**303.1 Guerilla Usability Test**

Test steps:

1. Test giver gives introduction to Testers. (See introduction)
2. User explains tasks for perform. (See Guerilla Test Tasks)
3. User administers Guerilla Test to tester one at a time.
4. Tester completes test
5. User records performance and feedback

**Test result** :

Testers should find it easy to navigate the application and perform all the tasks presented to them.

**Reliability**

Id: T\_C\_304

**304.1 Reliability Test**

Test steps:

**Test result** :

# **Guerilla Test Tasks**

|  |  |  |
| --- | --- | --- |
| ***Task*** | ***Description*** | ***Expected Results*** |
| Place a cauliflower crop on the field | The Tester would need to figure out how to select a cauliflower crop and then fill a plot with that selected crop. | The Tester should be able to navigate the crop selection panel and then insert that crop into a plot. |
| Changing the fertilizer giving during the simulation to minimum. | The Tester needs to find and locate the fertilizer dropdown and select ‘minimum’ | The Tester should find it extremely easy to change the fertilizer to minimum, and therefore should take no more than 3seconds. |
| Fill in start date 12-12-2016 and end date 03-04-2017. | The Tester has to find the start and end date buttons select them, and the proceed to navigate the popup calendar in order to select the dates. | The Tester is familiar with the popup calendar functionality and navigates it with easy, should not take more than 10 seconds to fill in dates. |
| Selects the previously placed cauliflower crop and reads crop statistics | Tester has to find the cauliflower crop on the field, select it, then locate the plot information box to be able to find and read the statistics of that plot. | After tester selects the crop, the plot information box should be visible to user and should not be a problem to find. This should take the Tester no more than 3 seconds to find the statistics after crop selection. |
| Plays simulation until end. | The Tester would have to determine that to run the simulation they need to find the play button and press play. | Determining that the play button starts the simulation should not be challenging. Finding the play button should be extremely easy this should be performed within 5 seconds. |

# Deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Date** |
| Test Plan | 09-10-2016 |
| Test Cases | 09-10-2016 |
| Execution | 16-12-2016 |