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

|  |  |
| --- | --- |
| **Document:** | Testcase |
|  |  |
| **Authors:** | Al Al-Mohaiminul Islam Khan  Mihail Hadzhinikolov  Raima Khan |
|  | Richard Dyer |
|  | Tsanko Hadzhiev |
|  | Zisis Damianidis |
|  |  |
|  |  |
| **Creation Date:** | 29/09/16 |
| **Last Revised:** | 04/10/16 |
| **Group Name:** | Tanks & Co.™ |
| **Version:** | 0.1 |

*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)

[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.

# 

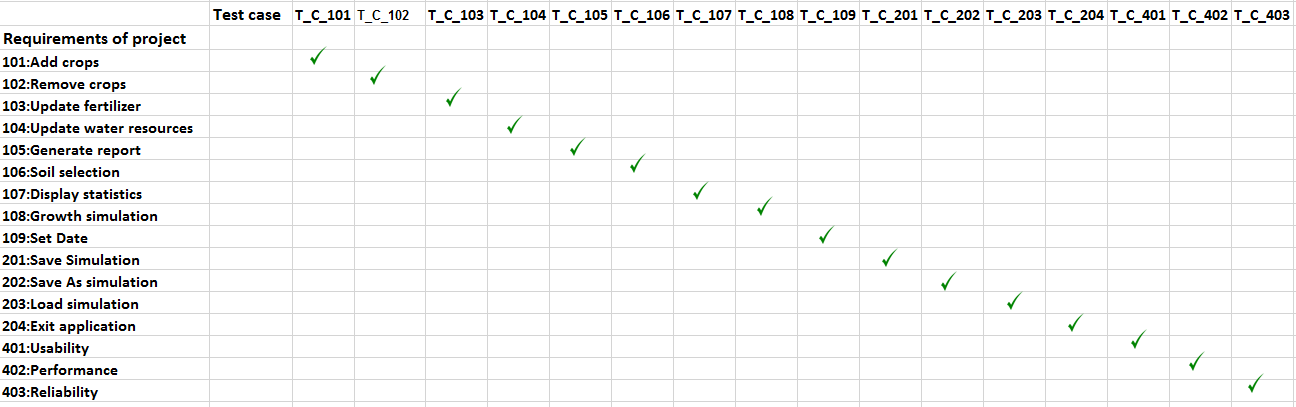
# 

# Scope

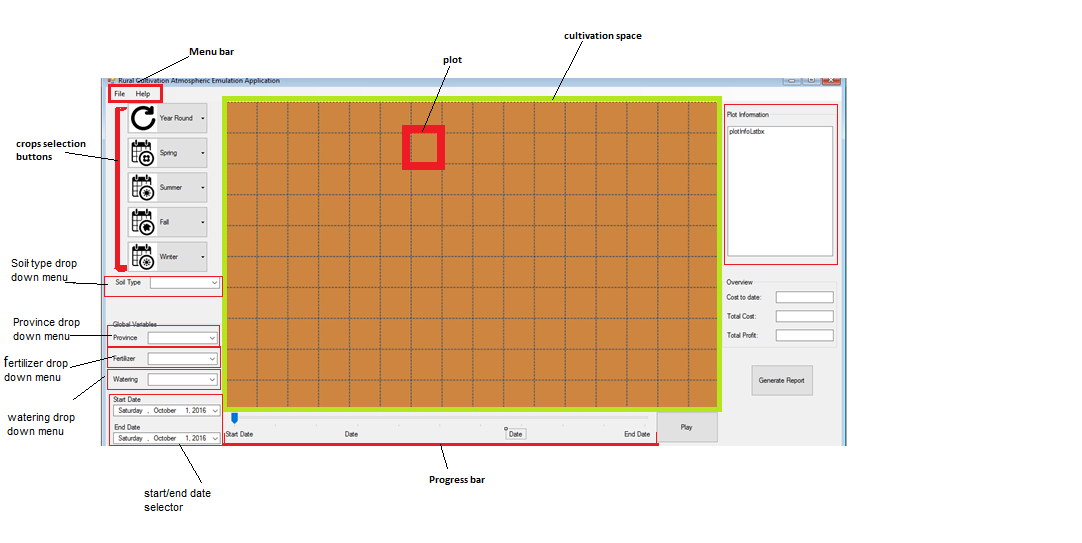
## Functions to be tested:

|  |  |  |  |
| --- | --- | --- | --- |
| 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\_101

Initialization status: Main screen of application is open. A grid of 70 spaces is created.

**101.1 Adding carrots on empty plot**

**Test steps:**

1. User clicks on year round crops selection button
2. List of crops that can grow all round year will be displayed.
3. User clicks carrots.
4. System shows picture and name “carrots” on split button as currently selected crop
5. User clicks on first plot in second column where he wants to grow carrots.

**Test result:**

The system will show carrots cultivated on that plot.

**101.2 Carrots added to already cultivated plot exception**

**Test steps:**

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

**Test results:**

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

**Removing crops**

Id: T\_C\_102

* 1. **Removing carrots from currently selected plot**

**Test steps:**

1. User will right click on first plot in second column, where carrots are cultivated.
2. System will show right clicked menu with delete option.
3. User clicks delete option from menu.

**Test results:**

System will remove carrots from first plot in second column. System will remove statistics info and will change status of plot as empty plot.

* 1. **Empty plot selected for removing crops**

1. User will right click on uncultivated plot.

**Test result:**

System will show right clicked menu with disabled delete option.

**Updating fertilizers to crops**

Id: T\_C\_103

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

**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 sufficient as the amount of fertilizer that he wants to add from the drop down options.
4. System changes the fertilizer attribute to ‘sufficient’.
5. System will change the fertilizer option to ‘sufficient’.

**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**

**Test steps:**

1. User will click on the watering drop down menu.
2. System will display the watering drop down menu ‘minimal’,’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 sufficient.
5. System will change the watering drop down option to ‘sufficient’.

**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 at least one field, cultivated with crops.

**Test steps:**

1. User will click generate button on right side of main form (according to users’ point of view).
2. System will open new form window with the report.
3. System report page displays the correct statistics and details of the simulation.

**Test results:**

When the user clicks on the generate report page, the user will be able to view a page that displays an overview of all the statistics from the simulation.

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

**Test steps:**

1. User clicks generate button on right side of main form (according to users’ point of view)
2. System will show message “Please cultivate any crop before generating report”.

**Test result:**

**Soil Selection**

Id:T\_C\_106

**106.1 Selecting soil of currently selected plot**

**Test steps:**

1. User will select 3rd plot in second column from cultivation space
2. System with 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 3rd plot in second column. Information panel will update soil type of this plot.

**Display statistics**

Id: T\_C\_107

**Initialization status:** The screen must have at least one field, cultivated with crops.

**107.1 Displaying statistics for currently selected plot**

**Test steps:**

1. User selects first plot in second column from cultivation space.

**Test results:**

System displays statistics for the selected plot in the plot information panel.

**107.2User is selecting empty plot**

**Test steps:**

1. User selects empty plot in second column from cultivation space.

Test results:

Information panel will be displayed empty.

**Simulating growth of crops:**

Id: T\_C\_108

**Initialization status:** The screen must have at least one field, cultivated with crops. User has entered in start and end date values.

**108.1 Using the play button**

**Test steps:**

1. User clicks the play button.
2. System starts the simulation.
3. System disables crops selecting buttons and watering/fertilizer/soil selection dropdown.
4. System changes the start button to stop button.
5. System runs the simulation at rate of 1 second/per week.
6. User clicks the stop button before simulation has finished.
7. System pauses the simulation.
8. System enables crop selection.
9. User adds two crops and replaces a crop with another crop.
10. User clicks the play button.
11. System continues the simulation from before time point.
12. System finishes the simulation.
13. System changes start button to “reset” button.
14. System enable crop selecting buttons and watering/fertilizer/soil selection dropdown.

**Test results:**

System will start the simulation run without errors. The user can pause the simulation edit the land and continue the simulation without errors.

**108.2 Using the timeline scroll to manually scroll through the simulation**

**Test steps:**

1. User moves process bar to the middle of the timeline.
2. System disables watering/fertilizer/soil selection dropdown.
3. System goes skips to the middle of the simulation time.
4. User removes a crop.
5. User returns the progress bar to the beginning of the timeline.
6. System shows the correct crops.
7. System restores watering/fertilizer/soil selection dropdown.

**Test Results:**

User can scroll through the timeline of the simulation make changes and those changes will be applied to that point of time.

**Selecting start and end date**

Id:T\_C\_109

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

**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**

**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**

**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**

**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.Saving a simulation**

**Test steps:**

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

2. System displays ”File” options.

3. User selects “Save” option.

**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.Save as simulation**

**Test steps:**

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

2. System displays “File” options.

3. User selects “Save As” option.

**Test results:**

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

**Loading a simulation**

Id: T\_C\_203

**Initialization status:** The User has started on a new project.

**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 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

**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. User data is saved.
4. System prompts users asking if he wants to quit the application.
5. User clicks “Yes”

**Test result** :System closes the application.

**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 “Cancel”
5. System closes the application.

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

# Deliverables

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