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

|  |  |
| --- | --- |
| **Document:** | Testcase |
| **Sprint :** | 3 |
| **Authors:** | Al Al-Mohaiminul Islam Khan  Mihail Hadzhinikolov  Raima Khan |
|  | Richard Dyer |
|  | Tsanko Hadzhiev |
|  | Zisis Damianidis |
|  |  |
|  |  |
| **Creation Date:** | 25/12/16 |
| **Last Revised:** | 26/12/16 |
| **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)

[Implementin the run simulation 5](#_Toc463382350)

[Save Simulation 5](#_Toc463382351)

[Exit Simulation 5](#_Toc463382351)

[Load Simulation 5](#_Toc463382351)

[Implementing the information panel 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 on 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 stimulate 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.

# 

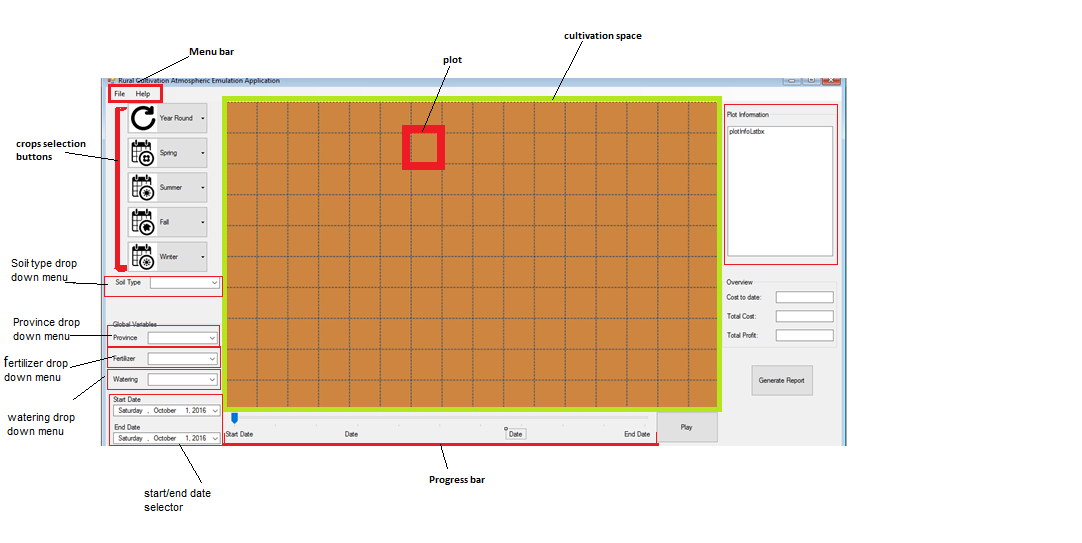
# 

# Scope

## Functions to be tested:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | | Objective | | Description | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
| 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. | |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |

## GUI



# Requirements for Testing

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

**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.

**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

**107.1 Displaying statistics on information panel**

**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 scroll manually .**

**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.

**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.

# T