Verification and Validation Report: Digital Twin Forest

Team # 8, Forest Mirror
Bowen Zhang
Tingyu Shi
Jiacheng Wu
Junhong Chen
Yichen Jiang

March 8, 2023

1 Revision History

Date	Version	Notes
March 6th	1.0	First Draft

2 Symbols, Abbreviations and Acronyms

symbol	**description**
Digital Twin	A virtual representation of the real world
FR	Functional Requirement
NFR	Non-Functional Requirement
SRS	Software Requirements Specification
GUI	Graphical User Interface
VnV	Validation and Verification

Contents

List of Tables

List of Figures

3 SRS Verification

The SRS can be found here. This part focuses on the verification of functional and non-functional requirements of SRS.

3.1 Modification to the current functional requirements

- FR1: This requirement is fulfilled.
- FR2: This requirement is fulfilled.
- FR3: This requirement is fulfilled.
- FR4: This requirement is removed. Loading forest model can be completed very fast, so implementing a progress bar is unnecessary.
- FR5: This requirement is removed. This requirement requires building large building very large models, which is not in line with the idea of model reuse.
- FR6: This requirement is fulfilled.
- FR7: We should modify this requirement to "The product shall be able to display environmental data and tree parameters."
- FR8: This requirement is removed. During our implementation, we found out that the lab cannot provide tree parameters for each tree. The lab can only provide tree parameters for each tree type. Therefore, this functional requirement is redundant and was removed from our product.
- FR9: This requirement is fulfilled.
- FR10: This requirement is fulfilled.
- FR11: This requirement is fulfilled.
- FR12: This requirement is fulfilled.
- FR13: This requirement is fulfilled.
- FR14: Since we do not have an overall forest view, we should modify this document to "The product shall allow users to go back to the main page when they are viewing a certain plot."
- FR15: This requirement is fulfilled.
- FR16: We should modify this requirement to "The product shall allow users to update environmental data and tree parameters."
- FR17: This requirement is fulfilled.

3.2 Newly added functional requirements

We have added the following requirements to our product.

- The product shall be able to display the contact information of all the developers.
- The product shall be able to display percentages of different tree types.
- The product shall be able to display correct environmental data when users change the plot.
- The product shall be able to display correct tree parameters when users change tree type.
- The forest model shall be able to reflect correct percentages of different tree types according to tree densities.
- Tree distributions shall be the same as the real forest.
- The forest model shall allow users to change seasons.

3.3 Modification to the current non-functional requirements

- LF1.1: This requirement is removed. This is not related to look and feel.
- LF1.2: This requirement is fulfilled.
- LF2.1: This requirement is fulfilled.
- LF2.2: This requirement is fulfilled.
- UH1.1: This requirement is fulfilled.
- UH2.1: This requirement is fulfilled.
- UH3.1: This requirement is fulfilled.
- UH4.1: This requirement is fulfilled.
- UH4.2: This requirement is fulfilled.
- UH5.1: This requirement is fulfilled.
- UH5.2: This requirement is fulfilled.
- PR1.1: This requirement is fulfilled.
- PR1.2: This requirement is fulfilled.
- PR1.3: This requirement is fulfilled.
- PR3.1: This requirement is fulfilled.
- PR3.2: This requirement is removed. All the tree models are generated from the data, if data are accurate, then we can extrapolate that tree models are accurate. Therefore, this requirement is covered by PR3.1.

- PR4.1: This requirement is fulfilled.
- PR4.2: This requirement is fulfilled.
- PR5.1: This requirement is fulfilled.
- PR6.1: This requirement is fulfilled.
- PR7.1: This requirement is fulfilled.
- PR8.1: This requirement is fulfilled.
- OE1.1: This requirement is fulfilled.
- OE1.2: This requirement is fulfilled.
- OE2.1: This requirement is fulfilled.
- OE3.1: This requirement is fulfilled.
- OE4.1: This requirement is fulfilled.
- OE4.1: This requirement is fulfilled.
- OE4.3: This requirement is removed. This requirement is covered by PR3.1.
- MS1.1: Our team failed to keep all the documents up to data. We will update all the documents for the final documentation.
- MS1.2: This requirement is fulfilled.
- MS1.3: This requirement is fulfilled.
- MS2.1: This requirement is fulfilled.
- MS3.1: This requirement is fulfilled.
- MS3.2: This requirement is fulfilled.
- SR1.1: This requirement is fulfilled.
- SR1.2: This requirement is not fulfilled.
- SR2.1: This requirement is fulfilled.
- SR2.2: This requirement is fulfilled.
- SR2.3: This requirement is fulfilled.
- SR2.4: This requirement is fulfilled.
- SR2.5: This requirement is fulfilled.
- SR2.6: This requirement is fulfilled.
- SR3.1: This requirement is fulfilled.

- SR3.2: This requirement is fulfilled.
- CP1.1: This requirement is fulfilled.
- LR2.1: This requirement is fulfilled.
- LR2.1: This requirement is removed.

3.4 Newly added non-functional requirements

None.

4 Design Documentation Verification

According to the SRS verification documented above, Design Documentations are revised and modified. The major modifications are listed below:

- Rearrange the data categories and hierarchy, and remove overall forest view. For environmental data, keep data `Humidity`, `Soil Carbon Content`, `Soil Nitrogen Content`, `Temperature`, and `LAI`. Add chart to show percentage of each tree type. For tree parameters, keep data `Density`, `DBH`, `Height`, `Age`. Overall forest view is combined with the models for each plot instead and the related information can still be reached.
- Change the stored models to the instantly generated models. Models for 14 plots are both space-consuming and redundant for our product. Store the constraints and methods to generate models instantly, instead of the models themselves.
- Remove the information for each tree. Instead, show the information about each kind of trees for each plot. This modification is for better accomplishing the scientific purpose, as the scientific work of our stakeholders concerns about the overall information within a certain area, instead of specific individuals. Also, due to limited resources and schedule, precisely reappearing every detail of the target forest in the model is not realistic.

5 VnV Plan Verification

During the test processes the Forest Mirror team performed to verify the implementations of functional requirements, team members found that the Automated Testing and Verification Tool section is not correct because automated testing is not applicable to the application. The back-end code of the application is bound to the game components in Unity, which can not be tested by solely using automated tools and running it so the team used manual tests on Unity to verify the implementations of requirements. After the mistake was discovered, the team decided to delete this section from the VnV plan. The Forest Mirror team gathered some feedback and problems occurring in the VnV plan from the other team members and TAs and modified some contents in the presenting version of the VnV plan according to that feedback.

6 Implementation Verification

Our team deployed both dynamic and static testing to verify the implementation of our project. For dynamic testing, since we implement the software with Unity, we used integral testing and system testing most of the time to monitor each component to operate normally. Also, the tests cases are evaluated to make sure all lines of code are covered. All the dynamic testing was done without any error

During the development of the project, we constantly communicated with our stakeholder —- Dr.Gonsamo to ensure that we are doing the right product. Also, by demonstrating the proof of concept to stakeholders, we can ensure that our clients will be satisfied with our product.

For static testing, our team used code inspection and document inspection to make sure that the design is following our expectation(high cohesion and low couple, easy to modify) and there is no big issue in the code. Our purpose for static testing is to make sure the code have good quality and complete. After our testing, we evaluated our code with good quality and completeness.

7 Automated Testing and Verification Tools

All the tests were conducted manually. So this part is not applicable.

8 Software Validation

After Rev0 demo, we had a meeting with Dr. Gonsamo and he gave us the following advice:

- We need to adjust the initial camera position when users just enter the forest model. The camera position should allow users to have a full view of the plot.
- We should add another GUI for displaying leaf information.
- We should implement seasonal change.
- Tree distribution should follow the real forest tree distribution.

9 Functional Requirements Evaluation

9.1 Presentation

The following testing results of test cases related to the presentation of our products.

9.1.1 Test-FR1

- Inputs: Testers click `Instruction` button for 10 times and reset the system to the initial state(Main page) after each click.
- Excepted Value: Instruction page should appear on the screen 10 times.
- Actual Value: Instruction page appeares on the screen 10 times.
- Result: PASS

9.1.2 Test-FR4.1

This test was about testing the display of the progress bar when switching from the main page to the forest model. During our implementation, loading the forest model is really fast so that the function of the progress bar is removed from our product. The SRS and VnV Plan will be updated accordingly.

9.1.3 Test-FR4.2

This test was about testing the display of the progress bar when switching between different forest models of different plots. During our implementation, switching the forest model between plots is really fast so that the function of the progress bar is removed from our product. The SRS and VnV Plan will be updated accordingly.

9.1.4 Test-FR5

This test was about testing displaying the full view (14 plots) of the target forest. This requires building very large models, which is not in line with the idea of model reuse. This function is removed from our product. The SRS and VnV Plan will be updated accordingly.

9.1.5 Test-FR9.1

- Inputs: Testers click `Environmental Data` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but environmental data are not displayed
- Excepted Value: Environmental data should appear on the left side of the screen 10 times
- Actual Value: Environmental data appeares on the left side of the screen 10 times.
- Result: PASS

9.1.6 Test-FR9.2

- Inputs: Testers click `Tree Parameters` button 10 times and reset the system to the initial state after each click. The initial state is defined as the forest model presented but tree parameters are not displayed
- Excepted Value: Tree parameters should appear on the right side of the screen 10 times
- Actual Value: Tree parameters appeares on the right side of the screen 10 times.
- Result: PASS

9.1.7 Test-FR12.1

- Inputs: Testers record all the environmental data from the GUI. This includes environmental data for 14 plots and the overall forest.
- Expected Value: All the environmental data recorded from the GUI should match the JSON files stored here.
- Actual Value: All the environmental data recorded from the GUI matches the JSON files.
- Result: PASS

9.1.8 Test-FR12.2

- Inputs: Testers record all the tree parameters from the GUI. This includes tree parameters of 7 types of trees from 14 plots and the overall forest.
- Expected Value: All the tree parameters recorded from the GUI should match the JSON files stored here.
- Actual Value: All the tree parameters recorded from the GUI matches the JSON files.
- Result: PASS

9.2 Users' interaction with the product

9.2.1 Test-FR2&3

- Inputs: Testers click `Start` button 10 times and reset the system to the initial state(main page) after each click.
- Expected Value: Forest model view should appear on the screen 10 times.
- Actual Value: Forest model view appears on the screen 10 times.
- Result: PASS

9.2.2 Test-FR6.1

- Inputs: Testers click `Environmental Data` button 10 times and reset the system to the initial state after each click. The initial state is defined as “environmental data are displayed”
- Expected Value: Environmental data display should disappear 10 times.
- Actual Value: Environmental data display disappeares 10 times.
- Result: PASS

9.2.3 Test-FR6.2

- Inputs: Testers click `Tree Parameters` button 10 times and reset the system to the initial state after each click. The initial state is defined as “tree parameters are displayed”
- Expected Value: Tree parameters display should disappear 10 times.
- Actual Value: Tree parameters display disappeares 10 times.
- Result: PASS

9.2.4 Test-FR7

Test-FR7 can be considered as a subset of Test-FR12.1 and Test-FR12.2. Since our system passed Test-FR12.1 and Test-FR12.2, this test was not performed. VnV Plan will be updated accordingly.

9.2.5 Test-FR8

This test was about testing clicking on a tree model and related tree parameters should be displayed. During our implementation, we found out that the lab cannot provide tree parameters for each tree. The lab can only provide tree parameters for each tree type. Therefore, this functional requirement is redundant and was removed from our product. The SRS and VnV Plan will be updated accordingly.

9.2.6 Test-FR10

- Inputs: Testers press `W` key 5 times in succession and each key press should last 2 seconds. After this, testers press `S` key 5 times in succession, and each key press should last 2 seconds.
- Excepted Value: Tree models should be zoomed in after each `W` key press. Tree models should be zoomed out after each `S` key press. and zoomed out after
- Actual Value: Tree models are zoomed in after each `W` key press. Tree models are zoomed out after each `S` key press.
- Test Result: PASS

9.2.7 Test-FR11.1

- Inputs: Testers press `A` key and `D` key.
- Expected value: Users' first view should move left after pressing `A` key. Users' first view should move right after pressing `D` key.
- Actual value: Users' first view moves left after pressing `A` key. Users' first view should move right after pressing `D` key
- Result: PASS

9.2.8 Test-FR11.2

- Inputs: Testers move the mouse in 4 different directions, which are left, right, forward, and backward.
- Expected value: The user's point of view should rotate according to the input.
  - Move the mouse forward: The user's first point of view should rotate up.
  - Move the mouse backward: The user's first point of view should rotate down.
  - Move the mouse left: The user's first point of view should rotate left.
  - Move the mouse right: The user's first point of view should rotate right.
- Actual value: The user's point of view rotates according to the input.
  - Move the mouse forward: The user's first point of view rotates up.
  - Move the mouse backward: The user's first point of view rotates down.
  - Move the mouse left: The user's first point of view rotates left.
  - Move the mouse right: The user's first point of view rotates right.
- Result: PASS

9.2.9 Test-FR13.1

- Inputs: Let testers click the `turn page` button on the window showing environmental data 10 times. Before, each click, testers need to make sure that the system is in the initial state.
- Expected value: A pie chart indicating percentages of different tree types should appear in the same window 10 times.
- Actual value: A pie chart indicating percentages of different tree types appears in the same window 10 times.
- Result: PASS

9.2.10 Test-FR13.2

- Inputs: Let testers click `turn page` button on the window showing tree parameters 10 times. Before, each click, testers need to make sure that the system is in the initial state.
- Expected value: Leaf information should appear in the same window 10 times.
- Actual value: Leaf information appears in the same window 10 times.
- Result: PASS

9.2.11 Test-FR14

This test was about testing the presentation of the overall view. During our implementation, we removed this view from our product. The SRS and VnV Plan will be updated accordingly.

9.2.12 Test-FR15

- Inputs: Let testers click `Quit` button located in the main page.
- Expected value: The product should terminate.
- Actual value: The product terminates.
- Result: PASS

9.2.13 Test-FR16

- Inputs: Testers enter new data in the input box and click `Update` button.
- Expected value: A text should pop up indicating if the update was successful. And environmental data display and tree parameters display should reflect the newly entered data.
- Actual value: A text pops up indicating if the update was successful. And environmental data display and tree parameters display reflect the newly entered data.
- Result: PASS

NOTE: The above test was conducted for 14 plots and 7 tree types. All tests passed.

9.2.14 Test-FR17

- Inputs: Let testers go to GitHub and download a new version.
- Expected value: The downloaded new version of software should work properly.
- Actual value: The downloaded new version of software works properly.
- Result: PASS

10 Nonfunctional Requirements Evaluation

10.1 Look and Feel Requirements Evaluation

10.1.1 Test-NFR-LF1.1

We found that this requirement is neither necessary or related to the look and feel requirement, so we decided to remove the requirement and test cases for that.

10.1.2 Test-NFR-LF1.2

- Testing Process: We conducted interviews to random people for the feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the first question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 10 chose A and 7 chose B in this question.(17/20)
- Result Analysis: PASS

10.1.3 Test-NFR-LF2.1

- Testing Process: Testers will collect all the data and parameters of our forest and compare the data to the physical measurements.
- Expected Result: All environmental data and tree parameters have relative errors less than 0.1 comparing to the actual data our clients provide us
- Actual Result: All environmental data and tree parameters have relative errors less than 0.1.
- Result Analysis: PASS

10.1.4 Test-NFR-LF2.2

- Testing Process:We conducted interviews to random people for the feedback by providing the questionnaire in the Appendix,
- Expected Result: Over 80 percent of the users choose A or B in the second question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 9 chose A and 9 chose B in this question.(18/20)
- Result Analysis: PASS

10.2 Usability and Humanity Requirements Evaluation

10.2.1 Test-NFR-UH1.1

- Testing Process:We conducted interview random people for the feedback by providing the questionnaire in the Appendix.

- Expected Result: Over 80 percent of the users choose A or B in the third question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 5 chose A and 12 chose B in this question.(17/20)
- Result Analysis: PASS

10.2.2 Test-NFR-UH2.1

- Testing Process: Every member in our team inspect all the UI components that contain texts in this software to see whether there is any non-English texts.
- Expected Result: 100 percent of the texts should be used in English except for number.
- Actual Result: 100 percent of the texts are in English except for numbers.
- Result Analysis: PASS

10.2.3 Test-NFR-UH3.1

- Testing Process: Testers will launch the system and go to main page. After that, the testers will click instructions option on that page
- Expected Result: Instructions are displayed on the screen.
- Actual Result: Instructions are displayed on the screen.
- Result Analysis: PASS

10.2.4 Test-NFR-UH4.1

- Testing Process: We conducted interview random people for the feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the fourth question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 12 chose A and 7 chose B for this question.(19/20)
- Result Analysis: PASS

10.2.5 Test-NFR-UH4.2

- Testing Process: We conducted interview random people for the feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the fifth question in the questionnaire
- Actual Result: Out of 20 interviewed users, 11 chose A and 9 chose B for this question.(20/20)
- Result Analysis: PASS

10.2.6 Test-NFR-UH5.1

- Testing Process: We asked 5 users to use mouse and keyboard to navigate through our system
- Expected Result: All of them will go through the system without any trouble
- Actual Result: All of them will went through the system without any trouble
- Result Analysis: PASS

10.2.7 Test-NFR-UH5.2

- Testing Process: We conducted interview random people for the feedback by providing the questionnaire in the Appendix.
- Expected Result: Over 80 percent of the users choose A or B in the sixth question in the questionnaire.
- Actual Result: Out of 20 interviewed users, 15 of them chose A and 1 of them chose B(16/20)
- Result Analysis: PASS

10.3 Performance Requirements Evaluation

10.3.1 Test-NFR-PR1.1

- Testing Process: Testers launch the software, set a timer, and record the time it takes to respond to the events from the mouse and the keyboard.
- Expected Result: The system should respond to any requests within 2 seconds.
- Actual Result: The average response time is less than 2 seconds.
- Result Analysis: PASS

10.3.2 Test-NFR-PR1.2

- Testing Process: Testers launch the software, move, click the mouse, and press the keyboard to give all kinds of inputs for actions. At the same time, testers record the FPS displayed on the screen.
- Expected Result: 80 percent of the time the software should have at least 30 FPS.
- Actual Result: The software has over 30 FPS all the time when it is running.
- Result Analysis: PASS

10.3.3 Test-NFR-PR1.3

- Testing Process: Testers click on the start option on the main page and simultaneously record the time the software takes to show all the models.
- Expected Result: The system should load all the models within 10 seconds.
- Actual Result: The system loads all the models in less than 10 seconds.
- Result Analysis: PASS

10.3.4 Test-NFR-PR3.1

- Testing Process: Testers record all the data and parameters displayed in the application and compare the data with the actual data collected by Dr.Gonsamo and his lab members. After that, testers compute the relative error for each data.
- Expected Result: All the data of the forest should have relative errors less than 0.05.
- Actual Result: All the data of the forest have relative errors less than 0.05.
- Result Analysis: PASS

10.3.5 Test-NFR-PR4.1

- Testing Process: Testers click on the application icon and wait for the application to open.
- Expected Result: The application should open normally every time when testers click on the application icon.
- Actual Result: The application opens normally every time when testers click on the application icon.
- Result Analysis: PASS

10.3.6 Test-NFR-PR4.2

- Testing Process: Testers launch the software and run the application in the background for 24 hours.
- Expected Result: The application should not crash in 24 hours.
- Actual Result: The application runs normally and does not crash in 24 hours.
- Result Analysis: PASS

10.3.7 Test-NFR-PR5.1

- Testing Process: Testers run the application without an internet connection and see if the system responds normally to every input action.
- Expected Result: The application should respond to all the input actions normally without an internet connection.
- Actual Result: The application responds to all the input actions normally without an internet connection.
- Result Analysis: PASS

10.3.8 Test-NFR-PR6.1

- Testing Process: Testers open the property window of the application and check its size.
- Expected Result: The size of the software should be less than 10 GB.
- Actual Result: The size of the software is around 250MB.
- Result Analysis: PASS

10.4 Operational and Environmental Requirements Evaluation

10.4.1 Test-NFR-OE1.1

- Testing Process: Testers run the application on both desktops and laptops to check if it works normally.
- Expected Result: The application should run normally on desktops and laptops.
- Actual Result: The application runs normally on desktops and laptops.
- Result Analysis: PASS

10.4.2 Test-NFR-OE1.2

- Testing Process: Testers run the software and perform all kinds of actions in the software using a mouse and keyboard.
- Expected Result: All actions should be accomplished without any problems.
- Actual Result: All actions are accomplished without any problems.
- Result Analysis: PASS

10.4.3 Test-NFR-OE2.1

- Testing Process: Testers run the software on Windows 10 or later version or MacOS 12 or later version and perform operations on the software.
- Expected Result: All the operations should be accomplished normally.
- Actual Result: All the operations are accomplished normally
- Result Analysis: PASS

10.4.4 Test-NFR-OE3.1

- Testing Process: Testers download and install the application from GitHub and check it can run normally.
- Expected Result: The application should be downloaded successfully from GitHub to the computer and it can run without any errors and bugs.
- Actual Result: The application is downloaded successfully from GitHub to the computer and it can run without any errors and bugs.
- Result Analysis: PASS

10.4.5 Test-NFR-OE4.1

- Testing Process: Testers look at the git commit history and see if there are weekly updates.
- Expected Result: There should be some weekly updates in the git commit history.
- Actual Result: There are some weekly updates in the git commit history
- Result Analysis: PASS

10.4.6 Test-NFR-OE4.2

- Testing Process: Testers perform tests on the previous functions after each update and record the test results in the test report.
- Expected Result: All the previous functions should be unaffected and all the test cases should be passed.
- Actual Result: All the previous functions are unaffected and all the test cases are passed.
- Result Analysis: PASS

10.5 Maintenance and Support Requirements Evaluation

10.5.1 Test-NFR-MS1.1

- Testing Process: Testers check the documentation of the product to see if the new features, functions, and any modifications are added to the documentation.
- Expected Result: All features, functions and modifications should be on the documentation.
- Actual Result: New features, functions and modifications after February are not on the documentation.
- Result Analysis: FAIL

10.5.2 Test-NFR-MS1.2

- Testing Process: Testers read the documentation of the product to see if the documentations specifies the functions clearly.
- Expected Result: All functions on the documentation should be clearly documented.
- Actual Result: All functions on the documentation are clearly documented.
- Result Analysis: PASS

10.5.3 Test-NFR-MS1.3

- Testing Process: Record the time from the bug occurs to the time when the bug is fixed. Measure the time to see if it is within three days.
- Expected Result: All bugs should be fixed within three days.
- Actual Result: All bugs are fixed within three days.
- Result Analysis: PASS

10.5.4 Test-NFR-MS2.1

- Testing Process: Testers first look for the contact method on the contact page, then use the contact method to contact the developers to see if they can contact the developers successfully or not.
- Expected Result: Testers should be able to reach the developers successfully.
- Actual Result: Testers are able to reach the developers successfully.
- Result Analysis: PASS

10.5.5 Test-NFR-MS3.1

- Testing Process: Testers try to launch the application on more than one operating system to see if it can be run successfully on a different system or not.
- Expected Result: The application should be able to launch on more than one operating system.
- Actual Result: The application is able to launch on more than one operating system.
- Result Analysis: PASS

10.5.6 Test-NFR-MS3.2

- Testing Process: Testers try to launch the application on the devices located indoors and outdoors respectively to see if the application can be used both indoors and outdoors.
- Expected Result: The application should be able to launch on the devices located both indoors and outdoors.
- Actual Result: The application is able to launch on the devices located both indoors and outdoors.
- Result Analysis: PASS

10.6 Security Requirements Evaluation

10.6.1 Test-NFR-SR1.1

- Testing Process: Testers try to find and download the product from GitHub and any other website.
- Expected Result: Testers can not download the product in other websites other than GitHub.
- Actual Result: Testers can not download the product in other websites other than GitHub.
- Result Analysis: PASS

10.6.2 Test-NFR-SR1.2

- Testing Process: Testers try to update the data of trees and forests via the interface and anywhere else to see if they can update the data successfully or not.
- Expected Result: Testers should not be able to update the data except through the interface.
- Actual Result: Testers can delete or modify .json file to modify the data.
- Result Analysis: FAIL

10.6.3 Test-NFR-SR2.1

- Testing Process: Testers inject 100 errors on purpose in our application and see if the scan results of the computer security application will detect the errors in the computer system or not.
- Expected Result: The number of errors that the product propagates should less than 2.
- Actual Result: 0 error is propagated.
- Result Analysis: PASS

10.6.4 Test-NFR-SR2.2

- Testing Process: Testers use the software and perform some complicated tasks such as zooming in and zooming out for a long time to see if the application will crash or not.
- Expected Result: The application should not crash.
- Actual Result: The application does not crash.
- Result Analysis: PASS

10.6.5 Test-NFR-SR2.3

- Testing Process: Testers input some invalid data in the `Update Data` page.
- Expected Result: Fail to update the data. And a warning message should pop up indicating invalid update operation.
- Actual Result: Fail to update the data. And a warning message pops up indicating invalid update operation.
- Result Analysis: PASS

10.6.6 Test-NFR-SR2.4&2.6

- Testing Process:Testers manually update some data into the product, take records, and find the data in the display.
- Expected Result: The data should be consistent with the data just updated.
- Actual Result: The data are consistent with the data just updated.
- Result Analysis: PASS

10.6.7 Test-NFR-SR2.5

- Testing Process: Testers manually compare the GUI and data files.
- Expected Result: Each data should have one unique position to display in the GUI and each GUI should correspond to a unique data.
- Actual Result: Each data has one unique position to display in the GUI and each GUI corresponds to a unique data.
- Result Analysis: PASS

10.6.8 Test-NFR-SR3.1

- Testing Process: Use questionnaire to check if the users are asked to provide personal information when using the product.
- Expected Result: No user should be asked to provide personal information.
- Actual Result: No user was asked to provided personal information.
- Result Analysis: PASS

10.6.9 Test-NFR-SR3.2

- Testing Process: Use questionnaire to check if the product sends notification without permission.
- Expected Result: No user should receive notification without permission.
- Actual Result: No user received notification without permission.
- Result Analysis: PASS

10.7 Cultural and Political Requirements Evaluation

10.7.1 Test-NFR-CP1.1

- Testing Process: Use questionnaire to check if any user is offended by the product.
- Expected Result: No user should feel offended by the product.
- Actual Result: No user feels offended by the product.
- Result Analysis: PASS

10.8 Legal Requirements Evaluation

10.8.1 Test-NFR-LR2.1

- Testing Process: Testers ask the users that if any part appears lawfully unreasonable when spread the questionnaires.
- Expected Result: Users should not notice any part lawfull unreasonable.
- Actual Result: No user notices any part lawfull unreasonable.
- Result Analysis: PASS

11 Comparison to Existing Implementation

This section will not be appropriate for every project.

12 Unit Testing

Unit testing does not accommodate Unity's workflow. Each script is automatically compiled by Unity. If the team wants to test a particular module, they still have to run the whole system and check the Unity console. As a result, all unit testing are included in the system testing. Please check section 3 and 4 instead.

13 Changes Due to Testing

- Feedback from the Rev 0 demo: It does not make sense that there are so much grass on the forest floor. The tree heights are unbalanced because no tree shall grow in the shade of other trees.
  The team change the ground texture to show more details of the soil and fallen leaves instead of grass for realism. The team measures the accurate height of the tree model, later the models are resized to fit the data collected by drones.

- Feedback from the Rev 0 demo: There are spelling and grammar errors in user interface and codes.
  The team goes through the user interface and comments to fix the typo in project.

- Feedback from the Rev o demo: The user interface is not straight forward. It is hard to learn how to update forest data.
  The team rearranges the user interface to make it easier to learn.

- Feedback from Dr. Gonsamo's lab: The species and tree distribution shall be consistent with the real forest.
  The team checked the species counted by Dr. Gonsamo's lab and updated the models. The team later writes a script to dynamically control the distribution of tree models based on the satellite photos.

14 Automated Testing

Automated testing does not accommodate Unity's workflow, everything is tested manually in Unity, such as UI, scene manager, and terrain.etc.

15 Trace to Requirements

Function Requirement	Test Case ID
FR1	Test-FR1
FR2	Test-FR2&3
FR3	Test-FR2&3
FR4	Test-FR4.1, Test-FR4.2
FR5	Test-FR5
FR6	Test-FR6.1, Test-FR6.2
FR7	Test-FR7
FR8	Test-FR8
FR9	Test-FR9.1, Test-FR9.2
FR10	Test-FR10
FR11	Test-FR11.1, Test-FR11.2
FR12	Test-FR12.1, Test-FR12.2
FR13	Test-FR13.1, Test-FR13.2
FR14	Test-FR14
FR15	Test-FR15
FR16	Test-FR16
FR17	Test-FR17

Table 1: Traceability between Functional Requirements and Tests

Non-functional Requirement	Test Case ID
NFR-LF1.1	Test-NFR-LF1.1
NFR-LF1.2	Test-NFR-LF1.2
NFR-LF2.1	Test-NFR-LF2.1
NFR-LF2.2	Test-NFR-LF2.2
NFR-UH1.1	Test-NFR-UH1.1
NFR-UH2.1	Test-NFR-UH2.1
NFR-UH3.1	Test-NFR-UH3.1
NFR-UH4.1	Test-NFR-UH4.1
NFR-UH4.2	Test-NFR-UH4.2
NFR-UH5.1	Test-NFR-UH5.1
NFR-UH5.2	Test-NFR-UH5.2
NFR-PR1.1	Test-NFR-PR1.1
NFR-PR1.2	Test-NFR-PR1.2
NFR-PR1.3	Test-NFR-PR1.3
NFR-PR3.1	Test-NFR-PR3.1
NFR-PR4.1	Test-NFR-PR4.1
NFR-PR4.2	Test-NFR-PR4.2
NFR-PR5.1	Test-NFR-PR5.1
NFR-PR6.1	Test-NFR-PR6.1
NFR-PR7.1	Test-NFR-PR7.1
NFR-PR8.1	Test-NFR-PR8.1

Table 2: Traceability between Non-Functional Requirements and Tests Part 1

Non-functional Requirement	Test Case ID
NFR-OE1.1	Test-NFR-OE1.1
NFR-OE1.2	Test-NFR-OE1.2
NFR-OE2.1	Test-NFR-OE2.1
NFR-OE3.1	Test-NFR-OE3.1
NFR-OE4.1	Test-NFR-OE4.1
NFR-OE4.2	Test-NFR-OE4.2
NFR-MS1.1	Test-NFR-MS1.1
NFR-MS1.2	Test-NFR-MS1.2
NFR-MS1.3	Test-NFR-MS1.3
NFR-MS2.1	Test-NFR-MS2.1
NFR-MS3.1	Test-NFR-MS3.1
NFR-MS3.2	Test-NFR-MS3.2
NFR-SR1.1	Test-NFR-SR1.1
NFR-SR1.2	Test-NFR-SR1.2
NFR-SR2.1	Test-NFR-SR2.1
NFR-SR2.2	Test-NFR-SR2.2
NFR-SR2.3	Test-NFR-SR2.3
NFR-SR2.4	Test-NFR-SR2.4&2.6
NFR-SR2.5	Test-NFR-SR2.5
NFR-SR2.6	Test-NFR-SR2.4&2.6
NFR-SR3.1	Test-NFR-SR3.1
NFR-SR3.2	Test-NFR-SR3.2
NFR-CP1.1	Test-NFR-CP1.1
NFR-LR2.1	Test-NFR-LR2.1

Table 3: Traceability between Non-Functional Requirements and Tests Part 2

16 Code Coverage Metrics

Code coverage does not accommodate our project since we did not use automated testing.

17 Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:
In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

1. Bowen Zhang: In general, there can be differences between the VnV plan and the actual activities conducted for VnV due to the testing process of Unity project. In the team's VnV plan, it is assumed that there will be unit testing and automated testing after completing MG and MIS. However, the team found that each script has been automatically tested by Unity during implementation. When a script is finished, Unity compiles it directly to see if there are any bugs. In addition, the tester needs to run the Unity project to use the script. As a result, the unit testing has been combined into the system testing. The team shall not use automated testing, everything could be tested manually in Unity, otherwise it is just duplication of work. In this way, the team decides to modify the VnV plan. The unit testing section and automated testing section are deleted. The team adds more details in the testings for functional and non-functional requirements to make them more comprehensive. In future projects, this problem can be anticipated by designing a flexible VnV plan that can accommodate Unity workflow. The team should regularly review and update the plan to reflect any changes happened in Unity. In this way, the team can solve the difference between the VnV plan and the actual activities conducted for VnV and ensure project success.

2. Yichen Jiang: The major difference between VnV plan and the activities that were actually conducted for VnV is brought by the change on the design. The VnV report was completed months ago, and within the time till now, our supervisor clarified more ideas that he would love to have in our product. There are many more features now in our product, so the verification and validation activities were adapted for the changes. Besides, the workflow is modified as well. We completed the VnV plan before we worked on MIS and MG, of course also before the real implementation. Our plan used to be do the testing after all the implementation work's done, while when we implement it, we found that the testing is actually everywhere and anytime. We have to compile the scripts in Unity and check if everything's right all the time, and that is actually how our testing works. This method also influences the unit testing part of our testing plan. Any test has to be done with launching whole system. Therefore you may notice that we put almost all efforts in system testing in the verification and validation report.

3. Junhong Chen: There are some differences between the VnV plan and the actual activities conducted for Vnv. In the VnV plan, we assumed that we can perform unit tests on a single function or a single module. However, in the actual test process, we can't perform unit tests for most of the functions because the implementations of the functions require launching the entire system. For example, it is hard to perform unit tests on the onClick() functions bound to some buttons which capture the click event from the mouse device and opens a new page just by simply running the code itself without launching the entire software. In addition,

after several meetings with Dr.Gonsamo and his lab members, a few new features were added to the product, which causes changes in some requirements, leading to the actual activities conducted for VnV differing from the VnV plan.

4. Jiacheng Wu: There are a few differences in the two V&V documents. First one is that the V&V plan is more about designing the test case and V&V is about conducting the designed test cases. The second difference is the time. At the time we wrote the V&V plan, the software was native and we didn't have a clear view to our product but for the report we have our product completed. Also, during our development, our requirements were changing constantly because we kept contacting with our clients. Due to the change of the requirements, we have to change SRS as well as test plan. Our anticipated changes are probably UI and seasonal change of the model. During this process, I realized that we should constantly change the our SRS and V&V documents every time we contact with our clients because our requirements are changing constantly during the communication.

5. Tingyu Shi: VnV Plan is different from the activities that we conducted for VnV report. In VnV plan, we documented that we will use automated testing. However, our product involves a lot of GUI and interactions between users and the software; therefore, manual testing is more feasible and efficient. So, we need to modify the automated testing part for the final document. For the future project, if it involves many GUI and user interactions, manual testing may be a better way to test.