Test Cases and Requirements Test Coverage Report for Plantit

Lab group : TS4

Team Members	**Role**
Ryan Tan Jinn En	Project Manager
Neo Yong Tai	Development Lead
Lee Yu Sheng Daniel	Back-End Developer
Mamuduri Paulani	Front-End Developer
Frankie Ye Htet Myat	Release Engineer / Manager
Chen Xueyao	QA Manager
Kundu Koushani	QA Engineer

Name of Design Team: **Team Syan**
School of Computer Science and Engineering, Nanyang Technological University

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Mamuduri Paulani, Lee Yu Sheng Daniel, Neo Yong Tai	18/10/2021	Chen Xueyao	19/10/2021	Initial Version
1.1	Mamuduri Paulani, Lee Yu Sheng Daniel, Neo Yong Tai	19/10/2021	Chen Xueyao	20/10/2021	Final Version

Table of Contents

Version History **2**

Table of Contents **3**

1 Unit Testing **4**

1.1 Test Case 1 4

1.2 Test Case 2 5

1.3 Test Case 3 6

1.4 Test Case 4 7

2 Integration Testing **8**

2.1 Test Case 1 8

2.2 Test Case 2 9

2.3 Test Case 3 10

2.4 Test Case 4 11

2.5 Test Case 5 12

2.6 Test Case 6 13

3 System Testing **15**

3.1 Test Case 1 15

3.2 Test Case 2 16

3.3 Test Case 3 17

4 Summary **18**

1. Unit Testing

1.1 Test Case 1

Test Case #: 1	Test Name: Add Plant
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing the adding of a new plant	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Click on the "+" icon	User is redirected to the plant creation form page for them to add plant details.
2	Click on Add Plant button	User is redirected to the dashboard page with the new plant record reflected.

Post-Conditions	
1	A new plant is displayed in the dashboard page.

1.2 Test Case 2

Test Case #: 2	Test Name: Navigation Bar Links
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing individual links of navigation bar	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Click on Dashboard in the navbar	The user is redirected to the dashboard page.
2	Click on Notifications icon in the navbar	The user is redirected to the notifications page.
3	Click on Logout icon in the navbar	The user is redirected to the sign-in page.

Post-Conditions	
1	The navigation bar is properly redirecting the user to the correct pages.

1.3 Test Case 3

Test Case #: 3	Test Name: Delete Plant
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing the deletion of a plant entry from dashboard table row	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Click on Trash icon	The plant entry is deleted from the table row of the dashboard.

Post-Conditions	
1	The respective plant is deleted, and changes are reflected in the dashboard.

1.4 Test Case 4

Test Case #: 4	Test Name: Edit Plant
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing editing of a plant entry in dashboard table row	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Click on Pen icon	The user is redirected to plant edit form with fields populated and allows them to change values of each field.
2	Click on Edit Plant button	The user is redirected to the dashboard page, reflecting the changes in the updated plant.

Post-Conditions	
1	The plant is successfully edited and changes are reflected in the dashboard page.

2. Integration Testing

2.1 Test Case 1

Test Case #: 1	Test Name: Sign In
System: Planit	Sub-System: App
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing sign-in	

Pre-Conditions	
1	User must have an internet connection.
2	User must be on the login page.

Step	**Action**	**Expected System Response**
1	Enter Email and Password	N/A
2	Click on Sign-in button	If credentials are correct: ● Redirected to the dashboard page. If user credentials are incorrect: ● Error Message: Username / Password incorrect.

Post-Conditions	
1	The correct page is displayed based on the input provided by the user.

2.2 Test Case 2

Test Case #: 2	Test Name: Sign Up
System: Planit	Sub-System: App
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing sign-up	

Pre-Conditions	
1	User must have an internet connection.
2	User must be on the sign-up page.

Step	**Action**	**Expected System Response**
1	Enter Name, Email and Password	If credentials are correct: ● N/A If user credentials are incorrect: ● Error message: Invalid name, email, or password
2	Click on Sign-up button	The user is redirected to the dashboard page.

Post-Conditions	
1	The user successfully created an account and is able to use the application.

2.2 Test Case 3

Test Case #: 3	Test Name: Dashboard
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing whether the dashboard display plants	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Load the page	The dashboard page should correctly display plant entries with name, age, species, and date columns.
2	Use the search function	The dashboard page should return valid search results of plant entries.
3	Use the sort function	The dashboard page should return entries which are sorted on a column correctly.
4	Change number of rows per page	The dashboard page should display the maximum number of specified plant entries.

Post-Conditions	
1	Details about plants are shown.

2.4 Test Case 4

Test Case #: 4	Test Name: Add Plant Details
System: Planit	Sub-System: Plant Creation Form
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing filling in of the plant details	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Enter name of plant	Successfully add a name for a plant in string format.
2	Enter age of plant	Successfully add an age for a plant in numerical format.
3	Enter species of plant	Successfully add species for a plant in string format.
4	Insert image of plant	Successfully upload an image file for the plant in jpg or png format.
5	Click on submit button	Successfully creates a new plant with entered details.

Post-Conditions	
1	Creates a plant and adds it to the dashboard.

2.5 Test Case 5

Test Case #: 5	Test Name: Notifications.js
System: Plantit	Sub-System: Individual Plant
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing system to notify user of upcoming tasks	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	Action	Expected System Response
1	Loads the page	The notifications should correctly display all upcoming tasks with the relevant information i.e notification title, notification subtitle, status, date, priority and plant name.
2	Use search function	System shows relevant notifications based on the search entry.
3	Select the number of notifications per page to be displayed	System correctly displays the selected numbers of notifications per page.
4	Click on filter function	Correctly displays the notifications based on priority.

Post-Conditions	
1	Notifications are shown.

2.6 Test Case 6

Test Case #: 6	Test Name: Individual Plant
System: Planit	Sub-System: Dashboard
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing the display of an individual plant	

Pre-Conditions	
1	User must have an internet connection.
2	User must be logged in.

Step	**Action**	**Expected System Response**
1	Click on a row of plant in the dashboard page.	User is redirected to the individual plant page of the selected plant.
2	Check that name is displayed correctly.	The individual plant page should display the name of the selected plant.
3	Check that the uploaded image is displayed correctly.	The individual plant page should display the image of the selected plant.
4	Check that age is displayed correctly.	The individual plant page should display the age of the selected plant.

5	Check that species are displayed correctly.	The individual plant page should display the species of the selected plant.
6	Check that notifications are displayed correctly.	The individual plant page should display the notifications of the selected plant.
7	Check that measurements are displayed correctly.	The individual plant page should display the measurements of the selected plant.

Post-Conditions	
1	Individual plant details are displayed correctly.

3. System Testing

System testing includes functional and non-functional testing. It is the process of testing all the integrated hardware and software components of our system to verify that the application meets its specified requirements.

3.1 Test Case 1

Test Case #: 1	Test Name: Functional Testing for Entire System
System: Plantit	Sub-System: App
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing if entire system functions as intended	

Pre-Conditions	
1	Have an existing account.
2	User opens the Plantit application.

Step	**Action**	**Expected System Response**
1	Clicking on respective buttons	The functionalities work as expected based on the integration and unit tests done previously.

Post-Conditions	
1	Desired action occurs based on the user’s actions.

3.2 Test Case 2

Test Case #: 2	Test Name: Performance Testing
System: Planit	Sub-System: App
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing response time of the database	

Pre-Conditions	
1	Have an existing account.
2	User opens the Planit application.

Step	**Action**	**Expected System Response**
1	The user clicks on various navbar tabs	MongoDB server response time should be a maximum of 0.2 seconds for read operations, with the information loaded and displayed on the page.
2	The user does add, delete, and edit operations in various pages	MongoDB server response time should be a maximum of 0.2 seconds for create, delete, and update operations.

Post-Conditions	
1	Desired outcome executed in time based on the user’s actions.

3.3 Test Case 3

Test Case #: 3	Test Name: Scalability Testing
System: Plantit	Sub-System: App
Designed By: Quality Assurance Team	Design Date: 18 Oct 2021
Executed By: Quality Engineer	Execution Date: 19 Oct 2021
Short Description: Testing if application can support multiple applications without crashing	

Pre-Conditions	
1	Have an existing account.
2	User opens the Plantit application.

Step	**Action**	**Expected System Response**
1	100 users are operating the website simultaneously. Users must add plant or edit plant details.	Database should update the plant details and add the plants for all users simultaneously without crashing.

Post-Conditions	
1	The database is updated based on the website usage of all 100 users simultaneously

4. Summary

Test Case	Pre-conditions	Test Steps	Test Data	Expected Result	Actual Result	Status
Add Plant	1.Have internet connection 2. Must be logged in	1. Click on the “+” icon 2. Click on Add Plant button	Name, age, species, image	Creates plant	Creates plant with specified details	Pass
Navigation Bar Links	1.Have internet connection 2. Must be logged in	1. Click on dashboard in nav bar 2. Click on notification icon in navbar 3. Click on logout icon on navbar	-	User is redirected to the correct pages	User is redirected to the correct pages	Pass
Delete Plant	1.Have internet connection 2. Must be logged in	1. Click on Trash icon	Plant	Plant successfully deleted	Plant was removed from the dashboard page	Pass
Edit Plant	1.Have internet connection 2. Must be logged in	1. Click on Pen icon 2. Click on Edit Plant button	Plant	Plant successfully updated	Plant changes are reflected in the dashboard page	Pass
Sign In	1. Have internet connection 2. Must be on Login page	1. Enter email and password 2. Click on sign-in button	Email and password	If credential are correct, redirect to dashboard If credentials are incorrect, display error message	Redirected to dashboard with correct credentials. Otherwise, displayed error message	Pass
Sign Up	1.Have internet connection	1. Enter Name, Email and	Name, email and	Account is successfully	User is able to	Pass

	2. Must be on sign-up page	Password 2. Click on Sign-up button	password	created	login and use the application	
Dashboard	1.Have internet connection 2. Must be logged in	1. Load the page 2. Use the search function 3. Use the sort function 4. Change number of rows per page	Plant	The functions work as they should	Dashboard functions are working properly	Pass
Add Plant Details	1.Have internet connection 2. Must be logged in	1. User enters relevant information about plant 2. Click on submit button	Plant	Creates a plant and adds to dashboard	The dashboard is updated with the plant	Pass
Notifications	1.Have internet connection 2. Must be logged in	1. Load page 2. Use search function 3. Select no. of notifications to be displayed 4. Click on filter function	Notification	Notifications are displayed correctly. Functions should work properly.	Notifications functions are working correctly	Pass
Individual Plant	1.Have internet connection 2. Must be logged in	1. Click on a row of a plant in the dashboard page. 2. Check that name is displayed correctly. 3. Check that the uploaded image is displayed correctly. 4. Check that age is displayed correctly. 5. Check that species are displayed correctly.	Specific plant (name, image, measurements, etc)	Details of a plant is displayed correctly	Details of a plant is displayed correctly	Pass

		6. Check that notifications are displayed correctly. 7. Check that measurements are displayed correctly.				
Functional Testing for Entire System	1. Have existing account 2. User opens app	1. Click various buttons	-	Buttons functionalities are as implemented in unit or integration testing	Buttons function as intended	Pass
Performance Testing	1. Have existing account 2. User opens app	1. Click on various navbars 2. Add, delete, or update operations	Various	MongoDB server response time <= 0.2 seconds	Response time is within limit	Pass
Scalability Testing	1. Have existing account 2. User opens app	1. Have 100 users using the website concurrently	-	Website works without crashing or slowing down	Website displays as per normal	Pass