Resilient Smart Garden
Test Specifications

Test ID#	1
Test level	System (1 of 5)
Quality criterion / attribute	External and Internal Quality: Reliability
Description of test	A new user attempts to register a user that already exists
Requirements reference	Usability
Steps of the test case	1. A new user runs the install.py script to begin installation. 2. A test enters ‘testUser’, a username known to exist as the user name they would like to register. 3. A user enters any matching password twice to advance registration.
Expected outcome	The user receives a message stating that the user name they requested already exists and the user is allowed to start the registration process over again.
Actual test outcome	Expected behavior happens.
Pass/Fail	Pass
Defect(s)	Standardized failure messages do not exist.

Test ID#	2
Test level	System (2 of 5)
Quality criterion / attribute	External and Internal Quality: Portability
Description of test	A new user is able to complete install and setup on a Raspberry Zero instead of a Raspberry Pi B+
Requirements reference	Usability
Steps of the test case	1. A new user downloads and extracts the Smart Gardens piDir zip file to their home directory. 2. The user runs the install.py script using the following command: *sudo python setup.py* 3. The user completes the user registration 4. The user completes Garden and sensor setup 5. The user completes the job scheduler setup
Expected outcome	The user will be able to complete the install of the Smart Gardens software regardless of the Pi Model (A, B, 2, 3, or Zero) so long as their OS is Raspbian.
Actual test outcome	Installation on Raspberry Pi Zero succeeded.
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	3
Test level	Unit
Quality criterion / attribute	Quality in Use: Efficiency
Description of test	A user will be able to leave the SmartGarden system unmonitored for at least two weeks without having the intervene, restart the software or hardware, and have daily uploads completed without fail
Requirements reference	Usability
Steps of the test case	1. Complete the setup process in a stable location with a constant power source and good wifi. 2. Register a test account, setup a garden with sensors, and schedule twice a day readings. 3. Leave the system running without intervention for 2 weeks 4. Check that readings were taken each day.
Expected outcome	There will be 28 garden readings over the 14-day period with no user interaction.
Actual test outcome	All readings were taken at the appropriate times.
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	4
Test level	Integration
Quality criterion / attribute	External and Internal Quality: Functionality
Description of test	Configuration settings can be downloaded remotely
Requirements reference	Usability
Steps of the test case	1. An administrator changes the configText field in the login table. 2. An administrator changes the configTime field to reflect the time changed. 3. The user's system attempts to take a reading 4. System compares remote configTime and downloads new configText 5. System stores the new configuration as config.json 6. System loads the new configuration and takes the readings with the new configuration.
Expected outcome	The system will update the config.json and display a line that a new configuration has been found.
Actual test outcome	The new configuration was detected, the config.json was updated, and the display line was printed.
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	5
Test level	Unit
Quality criterion / attribute	External and Internal Quality: Functionality
Description of test	The system will be able to support more than one Garden as part of a single configuration.
Requirements reference	Scalability
Steps of the test case	1. The user will add a complete garden with at least one of each sensor. 2. The user will then add a second garden with at least one of each sensor. 3. The user will run the takeReadings job.
Expected outcome	The readings for both gardens will be completed with no errors.
Actual test outcome	Both gardens were added and worked normally
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	6
Test level	Unit
Quality criterion / attribute	Usability
Description of test	Test the statistics report if the website run without enter any data
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Account Management Page 2. Press Statistics Report 3. Press Submit Button
Expected outcome	Error Message displayed
Actual test outcome	A dialog tell user to enter the user name
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	7
Test level	Unit
Quality criterion / attribute	Usability
Description of test	Test the website will it register the same name two times
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Account Management Page 2. Enter the Username and password 3. Press the submit button twice
Expected outcome	The first time, user is allow to register. The second will give a error message for it.
Actual test outcome	The first time, user is allow to register. The second will give a error message for it.
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	8
Test level	Unit
Quality criterion / attribute	Usability
Description of test	Testing can I use invalid user name and password
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Account Management Page 2. Enter an invalid Username and Password 3. Press Submit
Expected outcome	Error Message will be displayed
Actual test outcome	Error Message was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	9
Test level	Unit
Quality criterion / attribute	Usability
Description of test	Delete a Garden
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Account Management Page 2. Add a new Garden to a user 3. Delete the Garden
Expected outcome	The Garden will be added then deleted from the database
Actual test outcome	The Garden was added then deleted from the database
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	10
Test level	Unit
Quality criterion / attribute	Accessibility
Description of test	Create a new User in the Smart Garden App
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Smart Garden App 2. Enter a Username and Password 3. Click on Register
Expected outcome	A Message telling the user how to login and visit the startup guide
Actual test outcome	The Message was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	11
Test level	Unit
Quality criterion / attribute	Usability
Description of test	Testing the Statistics Report with empty valid user name, password, garden name, sensor name but without data.
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Account Management Page 2. Create a new Account and add a garden 3. Go to the Statistics Report Page 4. Use the account to retrieve a report
Expected outcome	A message will appear saying that no data currently exists
Actual test outcome	The Message was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	12
Test level	Integration
Quality criterion / attribute	Modifiability
Description of test	The User can create a new garden from the App
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Smart Garden App 2. Login using an existing user 3. Enter a Garden name and description 4. Click Add Garden
Expected outcome	The Garden will pop up on the list of gardens
Actual test outcome	The newly added garden was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	13
Test level	System
Quality criterion / attribute	Accessibility
Description of test	The user can view all the data from each sensor
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Smart Garden App 2. Enter a Username and Password 3. Click on a garden 4. Click on a sensor 5. Enter a date range and view the data
Expected outcome	The graph will show the correct data from the database
Actual test outcome	The correct data was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	14
Test level	Integration
Quality criterion / attribute	Accessibility
Description of test	The User can access the startup guide from the App
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Smart Garden App 2. Click on "Startup Guide"
Expected outcome	The user will be taken to the online instructional
Actual test outcome	The user was taken to the website
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	15
Test level	Unit
Quality criterion / attribute	Accessibility
Description of test	Login to the App with an incorrect Username
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Open the Smart Garden App 2. Enter an incorrect Username and Password 3. Click on Login
Expected outcome	A Message telling the user the credentials are incorrect
Actual test outcome	The Message was displayed
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	16
Test level	Integration
Quality criterion / attribute	Reliability
Description of test	The Pump is activated at the correct time
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Insert the Moisture sensor into dry soil 2. Initiate the TakeReading.py script
Expected outcome	The script will return a reading to console then the pump will fire
Actual test outcome	The pump turned on
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	17
Test level	System
Quality criterion / attribute	Precision
Description of test	The Console reading matches data sent to the database
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Turn on the Pi and insert the Sensor into the soil 2. Initiate the TakeReadings.py script
Expected outcome	The values displayed in console should match the values inserted into the database
Actual test outcome	They were identical
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	18
Test level	System
Quality criterion / attribute	Responsiveness
Description of test	The Raspberry Pi will shut off the pump after the set amount of time
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Turn on the Raspberry Pi 2. Initiate the TakeReadings.py script
Expected outcome	The Pump will stay on for the amount of time the user designated. 3 as default
Actual test outcome	The pump turned on for 3 seconds then shut off
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	19
Test level	System
Quality criterion / attribute	Customizability
Description of test	The Raspberry Pi's current Moisture limit and readings schedule can be changed
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Turn on the Raspberry Pi 2. Initiate the Setup.py script 3. Follow instructions and set a new Schedule and Moisture Limit
Expected outcome	The console will output the new values saved to the configuration file
Actual test outcome	The Configuration file was changed successfully
Pass/Fail	Pass
Defect(s)	N/A

Test ID#	20
Test level	Unit
Quality criterion / attribute	Portability
Description of test	The website can be ran on mobile, and different desktop web browsers
Requirements reference	External and Internal Quality: Functionality
Steps of the test case	1. Navigate to the Smart Garden website on multiple devices and browsers.
Expected outcome	The website should display normally on all browsers and devices
Actual test outcome	The website was displayed normally
Pass/Fail	Pass
Defect(s)	N/A