Department of Computer Science and Engineering
The University of Texas at Arlington

System Test Plan

Project: Home Irrigation Control System

Team Members:
Belachew Haile-Mariam
GautamAdhikari
Jeremiah O'Connor
Tung Vo

Last Updated: 3/30/2015

TABLE OF CONTENTS

Document Revision History

Revision Number	Revision Date	Description	Rationale
0.1	3/21/15	STP first draft	Created the skeleton for the sections
0.2	3/30/15	STP first draft	Section integration and review
0.3	4/8/15	STP baseline	Integrate changes from peer review
1.0	4/9/15	STP baseline	Final review for baseline submittion

List of Figures

List of Tables

1. Introduction

This System Test Plan document will provide information about the testing aspects of the Home Irrigation Control System (HICS) project. Here we will breakdown our project into its smallest components and describe all of their corresponding testing details, namely the features to be/not to be tested, possible risks, testing approaches, item pass/fail criteria, test deliverables, and test schedule. This document will expand upon the layers proposed in our ADS: our Sensor Layer, Hardware I/O Layer, Interface Layer, and Server Layer. It will also make references to our previous documents: the System Requirements Specification, Architecture Design Specification, and Detailed Design Specification.

1.1 Product Concept

HICS is an intelligent home irrigation control system that utilizes soil moisture sensors to measure the amount of moisture present in the user's lawn and use this information to water the lawn in an efficient way. The rain sensors and the temperature sensors help prevent watering when unnecessary. The sensors, hardware and their proper integration and programming make HICS a machine that's smart enough to save people time, effort, and money on their home irrigation system. Its purpose is to replace an existing sprinkler control system and allow users to control their home irrigation remotely through a web application that will scale to fit on a computer or mobile device.

HICS uses a central control unit to read soil moisture levels and set watering schedules accordingly by interfacing directly with the users existing irrigation valves. The control unit allows users to remotely schedule or monitor their lawns by communicating with the HICS web application over the Internet. Through the web application, users will be granted complete access to all the features that HICS offers while providing an intuitive and easy to use interface.

1.2 Testing Scope

The system testing for HICS is a critical phase for ensuring our prototype has met all the critical standards set forth in previous document. In order to maintain a realistic schedule for deployment our team has decided to focus testing in a limited and controlled environment. By focusing on testing HICS in our mockup environment it will allow us to streamline a large portion of our hardware tests. By not having to setup the product in a large scaled system and allowing us to test indoors it will allow the team more flexible and throughput when testing the controller and sensor devices.

2. References

This section provides a reference for the System Test Plan to the System Requirements Specification, Architectural Design Specification, and Detailed Design Document. These documents were created prior to this Test Plan and have been used when evaluating the teams approach to testing HICS.

2.1 System Requirements Specification

The SRS outlines the requirements for the HICS project. It goes over the Customer, Packaging, Performance, Safety, Maintenance and Support, and Other Requirements.

2.1.1 – Customer requirements

Req #	Requirement	Description	Priority
3.1	Central Control Unit	The central control unit is responsible for all communication between the web application, soil moisture sensors, and water valves. The control unit will transmit the readings from the soil moisture sensors and report the data to the web application. The control unit also controls the water valves, which sets the flow of water to the sprinkler system.	1 – Critical Priority
3.2	Soil Moisture Sensors	In-ground sensors that monitor and report soil moisture levels.	1 – Critical Priority
3.3	Web Application	The web application will be used to interface with the central control unit. The application is responsible for scheduling watering times as well as interfacing with the central control unit to control the water valves. The web application will be built with a scalable interface to enable the same functionality on a mobile device as it would on a desktop computer while maintaining a similar look and feel.	1 – Critical Priority

3.4	Watering Scheduler	The web application can schedule watering times and durations for the sprinkler system. The scheduler will display upcoming watering schedules and allow users to create, edit, or delete them.	1 – Critical Priority
3.5	Soil Moisture Reports	The web application will generate visual reports of the current soil moisture levels. The reports will be generated as graphic representations of the data and will be filterable by zone, year, month, and day.	3 – Moderate Priority
3.6	User Login	The web application will have a status indicator on the home dashboard to display whether a zone is actively being watered or not.	1 – Critical Priority
3.7	Active Status	The web application will have a status indicator on the home dashboard to display whether a zone is actively being watered or not.	4 – Low Priority
3.8	Rain Sensor	An external rain sensor will monitor the weather conditions and will send an alert to the central control unit in the event of rain. The user preferences will dictate whether this event will turn the sprinklers off or not.	2 – High Priority
3.9	Web Hosting Server	There will be a server hosting our web application with an associated URL for domain user access.	1 – Critical Priority
3.10	Database Management System	There will be a DBMS that will store all data from soil sensor readings as well as all the information for the web application. This information includes user account information, settings, reports, and weather data necessary for the application to perform as expected.	1 – Critical Priority

3.11	Device Registration	A user must register their HICS device with their account in order to interface with it via the web application. The central control unit will have a unique serial number on it that will be used to register the device to a user account.	2 – High Priority
3.12	Region Grouping	A user may group their sprinkler zones into regions through the web application. This functionality will allow the user to setup watering schedules for multiple zones at once without having to configure each individual zone.	3 – Moderate Priority
3.13	Auto Off	The user can set an auto off timer via the web application that terminates watering if the duration lasts longer than the set time.	3 – Moderate Priority

Table 2-1 Customer Requirements Table

2.1.2 – Packing requirements

Req #	Requirement	Description	Priority
4.1	Control Unit Housing	The components of the control unit, including its accessories, will be contained inside a mountable hard plastic container.	1 – Critical Priority
4.2	Soil Moisture Sensor Packaging	(4) Soil sensors will be included in the final packaging of HICS. They will be located within their own boxes and will have corresponding documentation that describes their proper installation. The user will be able to purchase additional soil moisture sensors and add them to their system.	3 – Moderate Priority
4.3	Rain Sensor Packaging	(1) Rain sensor will be included in the final packaging of HICS. It will be located within its own box and will have corresponding documentation that describes its proper installation.	3 – Moderate Priority

4.4	Connecting Cables	(1) Power cable and (1) Ethernet cable will be included in the final packaging of HICS. Their intended purpose is to be used with the central control unit.	3 – Moderate Priority
4.5	Software Application	The HICS software should be a stand-alone web application that requires no set up from the user. The user must register an account with a valid product serial number to access the web application. Once they are logged into their account and their device has been connected, they will be able to access the features of the HICS.	1 – Critical Priority
4.6	User Manual	A user manual will be presented to the user that details the use and operation of HICS. It will be provided in the final packaging as a CD-ROM and will also be made available online via the web application.	3 – Moderate Priority

Table 2-2 Packing Requirements Table

2.1.3 – Performance Requirements

Req #	Requirement	Description	Priority
5.1	Sensor Accuracy	Proper sensor readings of soil moisture levels must be captured to ensure accurate and efficient watering schedules. This requires solid construction of the sensor modules and proper software analysis of the data provided by the sensors.	2 – High Priority
5.2	Rain Detection	In the event of rainy weather, the rain sensor must quickly transmit an alert to the control unit to interrupt or delay any active watering.	3 – Moderate Priority

5.3	Communication Between the Web Application and Central Control Unit	The user will be able to set watering schedules through the web application to send a signal to the central control unit to initiate watering. This must happen in a timely manner to ensure consistency between the scheduled start/end times and the actual start/end times.	2 – High Priority
5.4	Communication Between the Soil Moisture Sensors and Central Control Unit	The soil moisture sensors will communicate with the central control unit to relay information about soil moisture levels. This must be accomplished periodically in an accurate and timely manner in order to avoid errors and to ensure consistency throughout the system.	2 – High Priority
5.5	Device Power Malfunction	If the server fails to receive communication from the central control unit for a specified period of time, it will notify the user that the system is offline.	3 – Moderate Priority

Table 2-3 Performance Requirements Table

2.1.4 – Safety requirements

Req #	Requirement	Description	Priority
6.1	Soil Moisture Sensor Insulation	The electronic components of the soil moisture sensors must be properly insulated against external environmental conditions. This is to ensure that they do not malfunction while in use.	2 – High Priority
6.2	Central Control Unit Temperature	Efficient air flow and low temperatures must be established while the central control unit is in operation. Computer fans and proper ventilation in the housing container will be used to cool the device and prevent it from overheating.	2 – High Priority

6.3	Proper Wiring of Central Control Unit	The central control unit will be powered from an external wall plug. The source will be 120v and improper wiring could leave the user exposed to currents in the range of 15-20 amps. It is crucial that the control unit is wired properly to accept this input and cause no harm to the device or user.	1 – Critical Priority

Table 2-4 Safety Requirements Table

2.1.5 – Maintenance and Support Requirements

Req #	Requirement	Description	Priority
7.1	Sensor Maintenance	The soil and rain sensors will be replaceable in the event of hardware failure.	2 – High Priority
7.2	Software Maintenance	All source code files and relevant documentation must be available for software maintenance and troubleshooting. The required files and documentation will be available via the GitHub repository. The software will be split into loosely coupled parts so it would be easier to make changes and improvements to the system over time.	3 – Moderate Priority
7.3	Source Code Documentation	All source code files required for HICS will be extensively commented to support future updates and troubleshooting.	3 – Moderate Priority
7.4	Scalability	The central control unit will support adding additional valves or soil sensors.	2 – High Priority

Table 2-5 Maintenance and Support Requirements Table

2.1.6 – Other Requirements

Req #	Requirement	Description	Priority
8.1	Mapping the Soil Moisture Sensors to Irrigation Valves	HICS requires that the user properly maps the soil moisture sensors to the irrigation valves. The central control unit will feature a diagram showing the one-to-one relationship between soil moisture sensor inputs and irrigation valve inputs.	1 – Critical Priority
8.2	Browser Support	The web application should be scalable and responsive in all of the latest versions of Mozilla Firefox, Google Chrome, and Internet Explorer.	3 – Moderate Priority
8.3	Central Control Unit Mounting	The central control unit must be mounted in a location where it can interface with wires from the water valves and soil moisture sensors. It must also be within range of Ethernet and power outlets.	2 – High Priority
8.4	Rain Sensor Mounting	The rain sensor must be placed outdoors in a location where there are no obstructions directly above it. The best placement for this sensor is on the roof of the user's home in a location where other water sources will not interfere with it.	3 – Moderate Priority
8.5	Application Security and Privacy	All user data and account information will be stored on a secure server. Users will only be allowed to access their HICS devices and account information if they provide the correct login credentials. In addition, all user passwords will be encrypted and stored only as hash values.	3 – Moderate Priority
8.6	User Administration Account	The web application will allow for administrative accounts to create, edit, and delete user accounts and their associated devices.	2 – High Priority

Table 2-6 Other Requirements Table

2.2 Architecture Design Specification

The Architecture Design Specification for HICS defines the overall layout and structure of the system. The ADS breaks down each major component of HICS into Layers which are comprised of Subsystems that handle a specific functionality.

2.2.1 – Architect Design Diagram

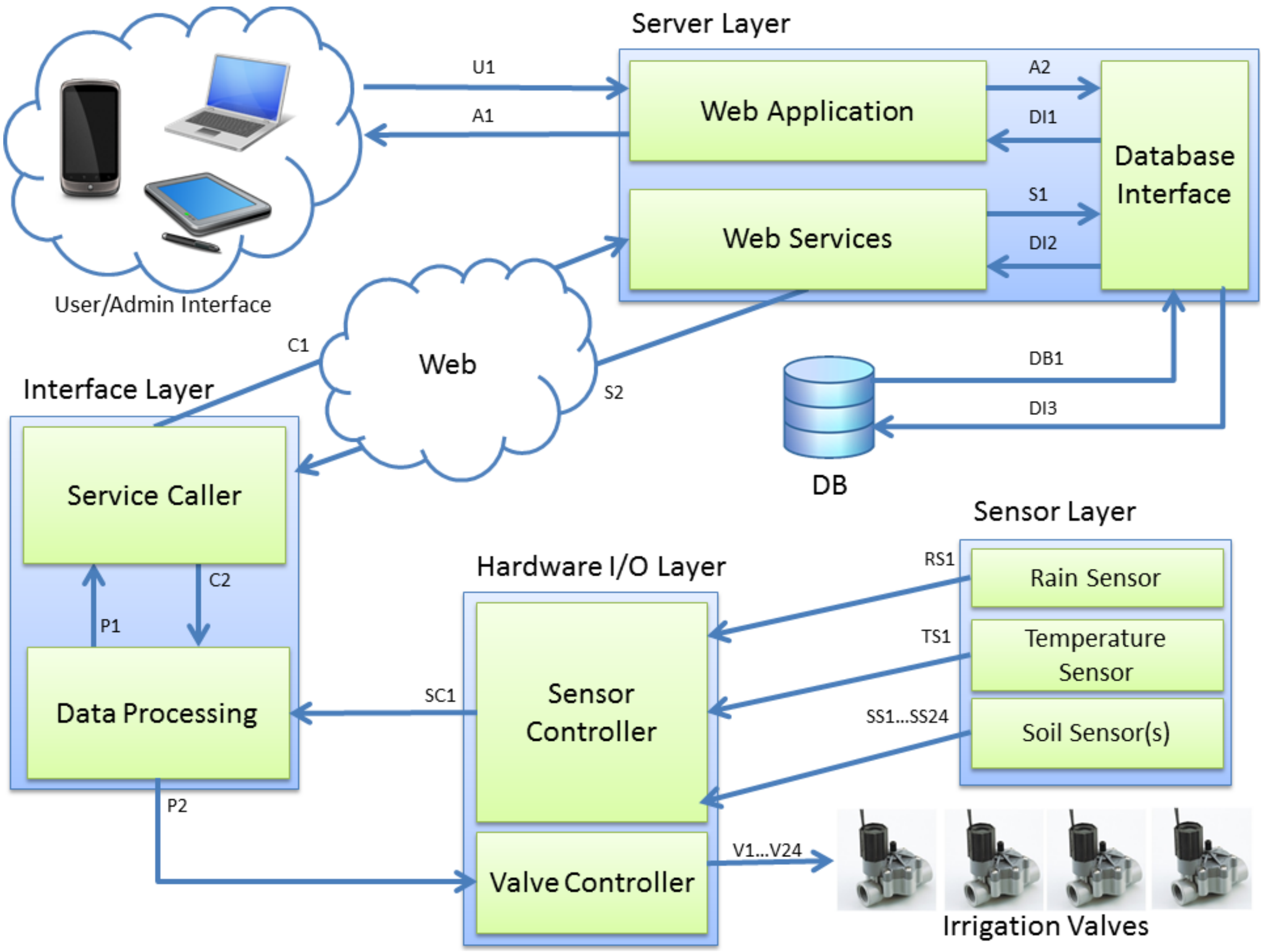

Figure 2-1 Architecture Design Diagram

2.2.2 – Data Flow Definitions

Element	Description
U1	User enters input into the web application or interacts with the applications GUI.
A1	User requested data is presented through the application GUI on the user's web or mobile browser.
A2	Data is received from the user to either update or request information about their system.
DB1	Data is queried from the database and passed along to the Database Interface subsystem.
DI1	Requested data from the web application query is located and returned.
DI2	Requested data from the web service query is located and returned.
DI3	The Database Interface stores data into the DB.
S1	The data requested from the interface is queried or data is sent to the database for storing.
S2	The service caller receives a response from a web service with requested information or a response status code.
C1	A request sent to the web service for either a valve switch command (on/off), data update response, or the caller sends new sensor readings for storage.
C2	Data returned by the web service is relayed to the data processor to check for valve switch commands to turn the water valves on or off.
P1	Data from the sensors is processed and formatted then passed to the service caller.
P2	Response data from the service caller is processed and an operation command is sent to the valve controller.
SC1	Parsed data from the sensors is sent to the data processing subsystem.
IV1+	Control signals are parsed and sent to the irrigation valves to turn them on or off.

RS1	An alert signal is sent from the rain sensor if a specified amount of rain has been detected.
TS1	A temperature reading from the temperature sensor is sent to the sensor controller for parsing.
SS1+	Soil moisture readings are sent from the soil sensors to the sensor controller for parsing.

Table 2-7 Data Flow Definitions Table

2.2.3 – Sensor Layer

The Sensor Layer has three sub systems: Rain Sensor, Temperature Sensor, and Soil Sensor(s). The Sensor Layer is responsible sending all the different environment condition readings to the Hardware I/O Layer. Each Sensor Layer subsystem communicates independently of the other subsystems.

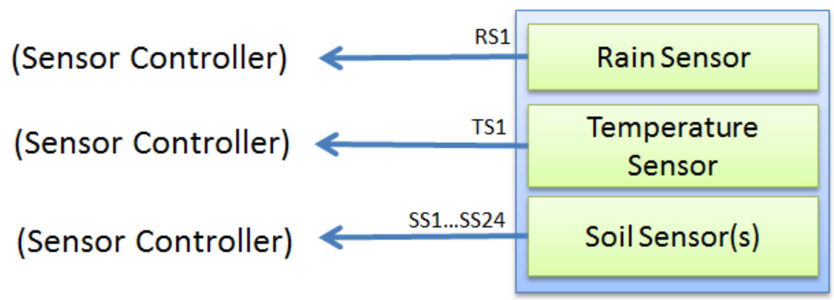

Figure 2-2 Sensor Layer Diagram

2.2.4 – Hardware I/O Layer

Hardware I/O Layer consists of two sub systems: Sensor Controller and the Valve Controller. This layer is responsible for exchanging data between the Interface Layer and the Sensor Layer. The Sensor Controller takes software input from the sensors and sends it to the Data Processing sub-system on the Interface Layer. The Valve Controller takes the input command from the Data Processing to control (switch on/off) the irrigation valves.

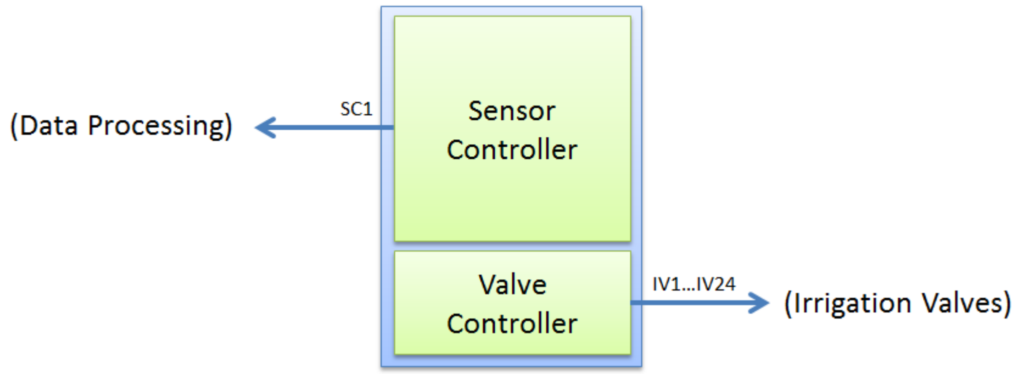

Figure 2-3 HW I/O Layer Diagram

2.2.5 – Interface Layer

The Interface Layer consists of two sub-systems: the Service Caller and Data Processor. The Interface Layer is responsible for processing all communication between the Hardware I/O and the Server Layer. The communication between the two layers consists of input readings from the Hardware I/O Layer as well as control commands from the Server Layer.

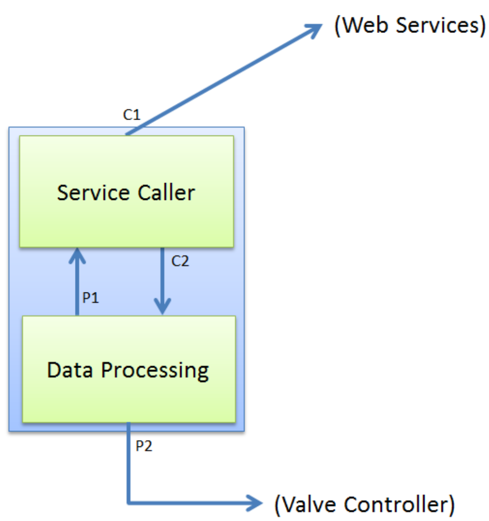

Figure 2-4 Interface Layer Diagram

2.2.6 – Server Layer

The Server Layer has three sub systems: Web Application, Web Services, and Database. The Server Layer handles communication and control between the web application/web services and the Interface Layer. The Server Layer has access to the database and uses the internet to exchange data between all interfaces.

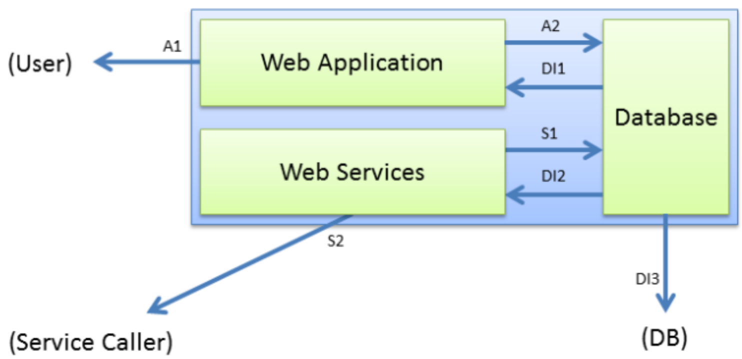

Figure 2-5 Server Layer Diagram

2.2.7 – Requirements Traceability

Requirements No.	Requirements Name	Sensor Layer			Hardware I/O Layer		Interface Layer		Server Layer		
		Rain Sensor	Temperature Sensor	Soil Sensor(s)	Sensor Controller	Valve Controller	Data Processing	Service Caller	Web Services	Database	Web Application
3.1	Central Control Unit					✔	✔	✔			
3.2	Soil Moisture Sensors			✔	✔						
3.3	Web Application				✔				✔		✔
3.4	Water Scheduler	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
3.5	Soil Moisture Reports			✔			✔	✔	✔	✔	✔
3.8	Rain Sensor	✔			✔						
3.14	Temperature Sensor		✔		✔						
5.1	Sensor Accuracy	✔	✔	✔	✔		✔				
5.2	Rain Detection	✔			✔	✔	✔				

Table 2-7 Requirements Traceability Matrix

2.3 Detailed Design Specification

The Detail Design Specification (DDS) document describes the low-level design of the Home Irrigation Control System (HICS) project. It breaks down every piece of the project into its lower components and describes all of their data flows and the roles. Additionally, The DDS document expands upon the layers proposed in our Architecture Design Specification document: the Sensor Layer, Hardware I/O Layer, Interface Layer, and Server Layer. It provides the actual data types, data flows, dependencies, and features along with some pseudo code that demonstrates the functionality of the module.

2.3.1 – Module Decomposition Chart

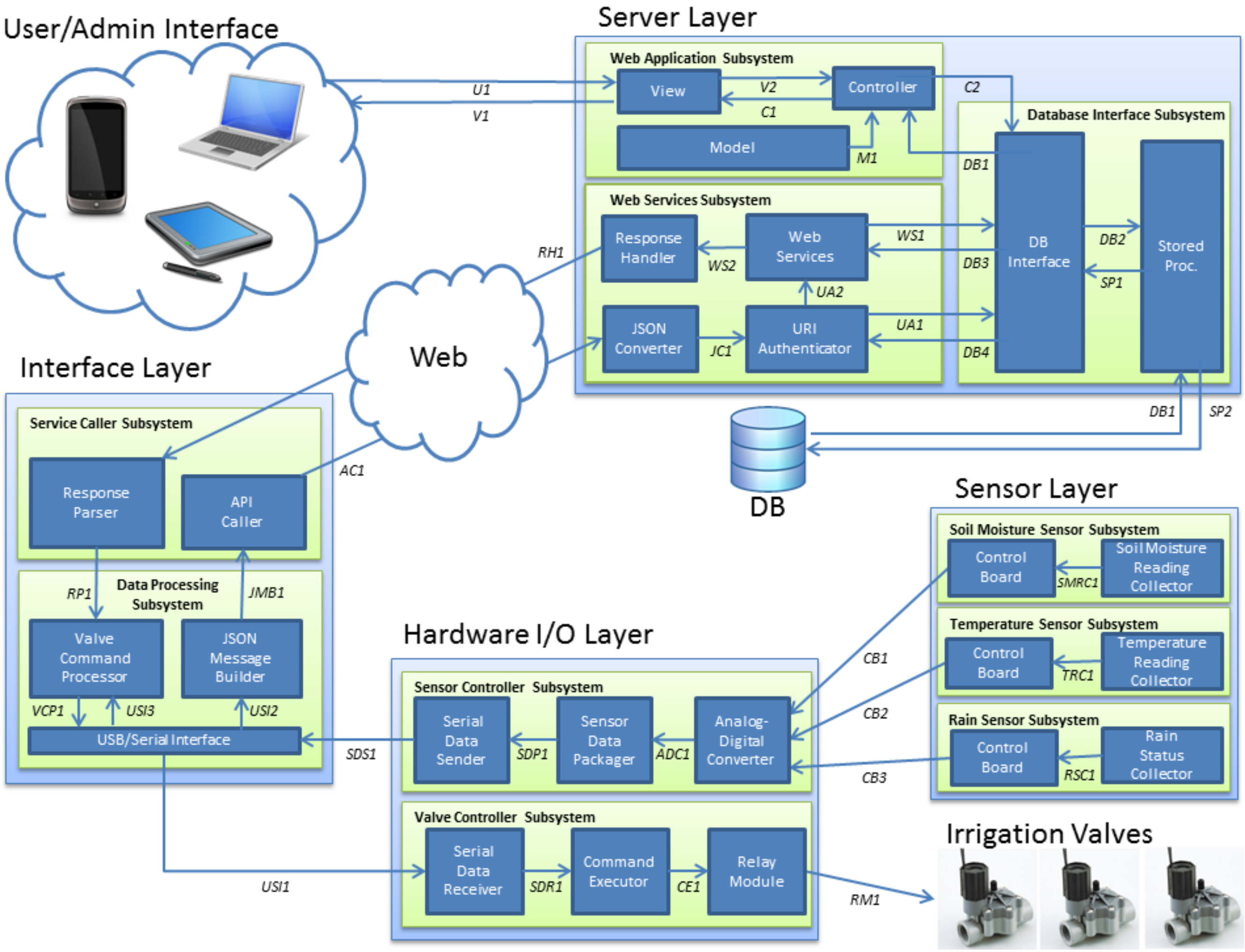

Figure 2-6 Module Decomposition Diagram

2.3.2 – Module Definitions

2.3.2.1 — Sensor Layer

2.3.2.1.1 — Rain Sensor Subsystem

2.3.2.1.1.1 — Rain Status Collector: The Rain Signal Collector is responsible for detecting the presence of rain. If the collector collects enough precipitation, the signal collector transmits an alert signal to the rain sensor Control Board module.

2.3.2.1.1.2 — Control Board: The rain sensor Control Board is responsible interpreting the analog voltage signal that transmits from the rain collector. Once the signal has been interpreted it is relayed to the ADC module.

2.3.2.1.2 — Temperature Sensor Subsystem

2.3.2.1.2.1 — Temperature Reading Collector: The Temperature Reading Collector reads the temperature of the environment and reports the signal to the temperature sensor Control Board module.

2.3.2.1.2.2 — Control Board: The temperature sensor Control Board interprets the analog signals being produced by the temperature sensor and reports the information to the ADC module.

2.3.2.1.3 — Soil Moisture Sensor Subsystem

2.3.2.1.3.1 — Soil Moisture Reading Collector: The Soil Moisture Signal Collector reads the amounts of moisture present in the soil and sends the signal to the soil moisture sensor Control Board module.

2.3.2.1.3.2 — Control Board: The soil moisture sensor Control Board is responsible for interpreting the analog signals being produced its corresponding soil moisture sensor and reports the information to the ADC module.

2.3.2.2 — Hardware I/O Layer

2.3.2.2.1 — Sensor Controller Subsystem

2.3.2.2.1.1 — Analog-Digital Converter: This module collects all the readings from the sensors and converts their signals from analog to digital so that the information can be further processed and packed.

2.3.2.2.1.2 — Serial Data Packager: This Serial Data Packager module is responsible for converting and packaging the raw sensor readings into readable data.

2.3.2.2.1.3 — Serial Data Sender: This module sends the collected sensor data over a USB cable to the Raspberry pi. This connection is the primary link between the Arduino and Raspberry Pi.

2.3.2.2.1 — Valve Controller Subsystem

2.3.2.2.1.1 — Serial Data Receiver: This module receives the serial data sent by USB/Serial Interface and converts it into usable command data.

2.3.2.2.1.2 — Command Executer: This module executes the command that was received from the Serial Data Receiver module to the relay module.

2.3.2.2.1.3 — Relay Module: The Relay Module converts the command from the Command Executer to an electronic signal that will toggle the irrigation valve(s) on or off.

2.3.2.3 — Interface Layer

2.3.2.3.1 — Data Processing Subsystem

2.3.2.3.1.1 — USB/ Serial Interface: This module initiates the connection between the Interface Layer and the Hardware I/O layer. The connection and communication between the layers is done over a USB cable.

2.3.2.3.1.2 — JSON Message Builder: The JSON Message Builder module converts the data received from the USB/Serial Interface into a JSON Object and formats the URI request as well.

2.3.2.3.1.3 — Valve Command Processor: This Valve Command Processor module parses the JSON object to determine what command is being relayed. The module also maps the command to a serial data command that is to be transferred to Valve Controller Subsystem through the USB/Serial Interface.

2.3.2.3.2 — Service Caller Subsystem

2.3.2.3.2.1 — API Caller: This module builds the HTTP request and makes the appropriate call to the HICS API. The JSON sensor readings object is attached to the request in the body and the HICS system identifier is attached in the parameters.

2.3.2.3.2.2 — Response Parser: This module parses the HTTP response sent back by the HICS API and creates a JSON object from the body that is sent to the Valve Command Processor.

2.3.2.4 — Server Layer

2.3.2.4.1 — Web Services Subsystem

2.3.2.4.1.1 — JSON Converter: This module converts the JSON data from the HTTP request into corresponding domain objects that the web services will understand. It also parses out the HICS system id from the request parameters for authorization.

2.3.2.4.1.2 — URI Authenticator: The URI Authenticator module is responsible for authenticating the HICS system id that was attached to the API call. If authorization is successful if routes the call to the appropriate Web Services method.

2.3.2.4.1.3 — Web Services: The Web Services module contains all the API methods needed for the Service Caller Subsystem to communicate back and forth between the database.

2.3.2.4.1.4 — Response Handler: This module takes the updated command state or status code and relays it to back to the system that initiated the request. It returns the data or the

2.3.2.4.2 — Web Application Subsystem

2.3.2.4.2.1 — Model: This module is defines the blueprint for the database tables. It provides a new model or loads a model with existing data and set rules on how that data is accessed.

2.3.2.4.2.2 — Controller: The Controller module handles all the dataflow between the HTML views and the database. It is also responsible for loading views and handling all page routings.

2.3.2.4.2.3 — View/UI: This module is the graphical interface that the user interacts with to control or view the status of their HICS system.

2.3.2.4.3 — Database Interface Subsystem

2.3.2.4.3.1 — DB Interface: The DB Interface module is the interface that is responsible for retrieving and storing data in that database. The module contains all the domain services that handle the logic for the CRUD functionality of each database table.

2.3.2.4.3.2 — Stored Procedures: This module acts like a middleman for communication with the database and is responsible for authenticating pre-requesting queries. The definitions inside the module are made to ensure the security of the data being communicated to and from the database.

2.3.3 – Producer-Consumer Matrix

	Consumer	Soil Moisture Reading Collector	Soil Moisture Sensor Control Board	Temperature Reading Collector	Temperature Sensor Control Board	Rain Status Collector	Rain Sensor Control Board	Analog-Digital Converter	Sensor Data Packager	Serial Data Sender	Relay Module	Irrigation Valve(s)	Command Executor	Serial Data Receiver	USB/Serial Interface	JSON Message Builder	Valve Command Processor	API Caller	Response Parser	JSON Converter	URI Authenticator	Web Services	Response Handler	DB Interface	Stored Procedures	Database	Controller	Model	View	User/Admin
Producer																														
Soil Moisture Reading Collector			SMRC1																											
Soil Moisture Sensor Control Board								CB1																						
Temperature Reading Collector					TRC1																									
Temperature Sensor Control Board								CB2																						
Rain Status Collector							RSC1																							
Rain Sensor Control Board								CB3																						
Analog-Digital Converter									ADC1																					
Sensor Data Packager										SDP1																				
Serial Data Sender															RDS1															
Relay Module												RM1																		
Irrigation Valve(s)																														
Command Executor											CE1																			
Serial Data Receiver													SDR1																	
USB/Serial Interface										USI1						USI2														
JSON Message Builder																		JMB1												
Valve Command Processor															VCP1															
API Caller																				AC1										
Response Parser																	RP1													
JSON Converter																					JC1									
URI Authenticator																						UA2		UA1						
Web Services																							WS2	WS1						
Response Handler																			RH1											
DB Interface																					DB4	DB3			DB2		DB1			
Stored Procedures																								SP1		SP2				
Database																									D1					
Controller																								C2					C1	
Model																											M1			
View																											V2			V1
User/Admin																													U1	

Table 2-8 Producer-Consumer Matrix

2.3.4 – Requirements Traceability

The following sections will detail, layer by layer, how each key requirement from our SRS is fulfilled by a combination of modules. From the tables below you can see just how integrated the entire HICS system is and how requirements can span across multiple modules. The Database Management System is a requirement that incorporates a large amount of modules into its functionality. This is because the DBMS of HICS is the component that allows users to communicate with their systems and vice versa. Other requirements that rely on multiple modules are the Central Control Unit, Web Application, and Water Scheduler. Each one of these requirements is fundamental to how HICS will function and where its value comes from. The components reflect that because we've designed this system to implement everything we emphasized during our SRS and ADS.

2.3.4.1 — Sensor Layer

Requirements No.	Requirements Name	Soil Moisture Sensor S/S		Temperature Sensor S/S		Rain Sensor S/S	
		Voltage Comparator	Soil Moisture Reading Collector	Voltage Comparator	Temperature Reading Collector	Voltage Comparator	Rain Status Collector
3.1	Central Control Unit	✔	✔	✔	✔	✔	✔
3.2	Soil Moisture Sensors	✔	✔				
3.3	Web Application						
3.4	Water Scheduler	✔	✔			✔	✔
3.5	Soil Moisture Reports	✔	✔				
3.6	User Login						
3.8	Rain Sensor					✔	✔
3.10	DB Management System	✔	✔	✔	✔	✔	✔
3.14	Temperature Sensor			✔	✔		
5.1	Sensor Accuracy	✔	✔			✔	✔
5.2	Rain Detection					✔	✔
5.3	Comm. Between Web and Unit						
8.2	Browser Support						

Table 2-9 Sensor Layer Requirements Table

2.3.4.2 — Hardware I/O Layer

Requirements No.	Requirements Name	Sensor Controller S/S			Valve Controller S/S		
		Analog-Digital Converter	Sensor Data Packager	Serial Data Sender	Relay Module	Command Executor	Serial Data Receiver
3.1	Central Control Unit	✔	✔	✔	✔	✔	✔
3.2	Soil Moisture Sensors	✔					
3.3	Web Application						
3.4	Water Scheduler	✔	✔	✔	✔	✔	✔
3.5	Soil Moisture Reports	✔	✔	✔			
3.6	User Login						
3.8	Rain Sensor	✔	✔	✔			
3.10	DB Management System	✔	✔	✔	✔	✔	✔
3.14	Temperature Sensor	✔	✔	✔			
5.1	Sensor Accuracy	✔	✔	✔			
5.2	Rain Detection	✔	✔	✔			
5.3	Comm. Between Web and Unit						
8.2	Browser Support						

Table 2-10 Hardware I/O Layer Requirements Table

2.3.4.3 — Interface Layer

Requirements No.	Requirements Name	Data Processing S/S			Service Caller S/S	
		USB/Serial Interface	Valve Command Processor	JSON Message Builder	Response Parser	API Caller
3.1	Central Control Unit	✔	✔	✔	✔	✔
3.2	Soil Moisture Sensors	✔		✔		✔
3.3	Web Application					
3.4	Water Scheduler	✔	✔	✔	✔	✔
3.5	Soil Moisture Reports	✔		✔		✔
3.6	User Login					
3.8	Rain Sensor	✔		✔		✔
3.10	DB Management System	✔	✔	✔	✔	✔
3.14	Temperature Sensor	✔		✔		✔
5.1	Sensor Accuracy					
5.2	Rain Detection	✔		✔		✔
5.3	Comm. Between Web and Unit				✔	✔
8.2	Browser Support					

Table 2-11 Interface Layer Requirements Table

2.3.4.4 — Server Layer

Requirements No.	Requirements Name	Web Services S/S				Database Interface S/S		Web Application S/S		
		JSON Converter	URI Authenticator	Web Services	Response Handler	DB Interface	Store Procedures	Controller	Model	UI/View
3.1	Central Control Unit									
3.2	Soil Moisture Sensors									
3.3	Web Application	✔	✔	✔	✔	✔	✔	✔	✔	✔
3.4	Water Scheduler			✔	✔	✔	✔	✔	✔	✔
3.5	Soil Moisture Reports					✔	✔	✔	✔	✔
3.6	User Login					✔	✔	✔	✔	✔
3.8	Rain Sensor									
3.10	DB Management System					✔	✔			
3.14	Temperature Sensor									
5.1	Sensor Accuracy									
5.2	Rain Detection									
5.3	Comm. Between Web and Unit	✔	✔	✔	✔					
8.2	Browser Support									✔

Table 2-12 Server Layer Requirements Table

3. Test Items

This section will detail the steps we are going to take to test HICS as a whole. We will first test each of the hardware components separately to ensure that the parts work as intended. We will then work from the ground up by testing the modules then moving on to integration testing and the testing of each layer. Aside from the hardware section every testing phase will be testing with the components already integrated and will work on top of a reliable, connected system.

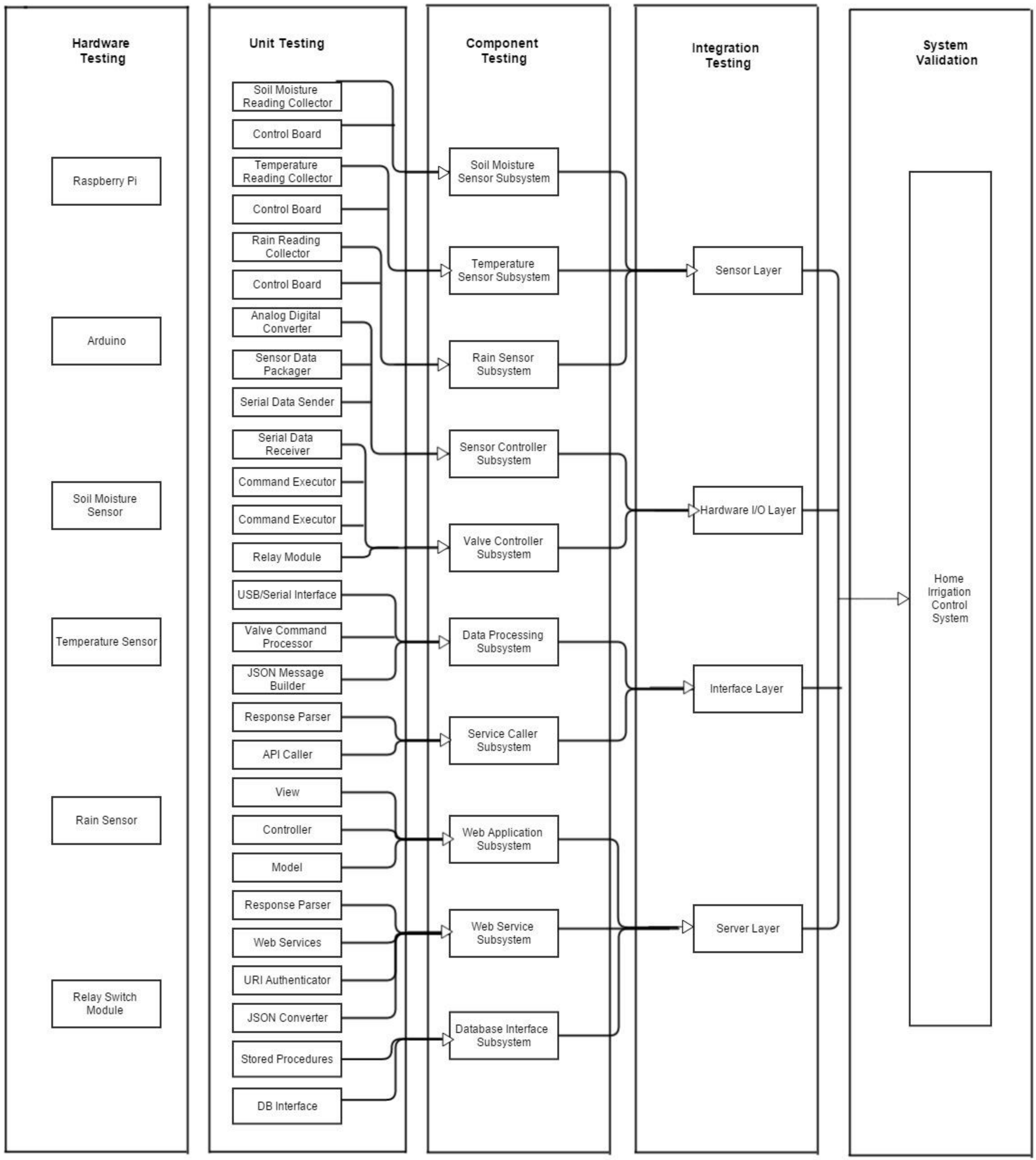

Figure 3-1 Testing Composition Diagram

3.1 Hardware Tests

Test ID	Hardware	Input	Output	Test	Priority
H1	Raspberry Pi B+	JSON response after sending HTTP request, sensor data from Arduino	Commands to the Arduino through USB interface, sensor data through an API call	We will test the functionality of the operating system to ensure that the program on the Raspberry Pi can process all communication between the web services and Arduino.	Critical
H2	Arduino	Commands from Raspberry Pi, environment readings from sensors, valve control statuses	Sensor data to Raspberry Pi, control signal to turn on and turn off the irrigation valves	We will test that we can send the sensor data to Raspberry Pi. We will also test that the Arduino can control all the valve switches.	Critical
H3	Soil moisture sensor	Soil moisture levels from the environment	Soil moisture reading to Arduino	We will test that the data received from the soil moisture sensors reflects the changes in the environment.	High
H4	Temperature sensor	Temperature reading from environment	Temperature reading to Arduino	We will test that the temperature reading received from the sensor is correct to the reading in the thermostat.	High

Table 3-1 Hardware Test Table

3.2 Unit Tests

3.2.1 – Soil Moisture Sensor Subsystem

Test ID	Module	Input	Output	Test	Priority
USSS1	Soil Moisture Reading Collector	Soil moisture sensor level from environment	Voltage difference to control board processed	We will test that the collector gives a voltage difference.	Moderate
USSS2	Control Board	Voltage difference from collector	Voltage level responding to the moisture reading	We will test that control board can power on and is giving out the voltage difference.	Moderate

Table 3-2 Soil Moisture Sensor Subsystem Unit Test

3.2.2 –Temperature Sensor Subsystem

Test ID	Module	Input	Output	Test	Priority
USTS1	Temperature Reading Collector	Rain condition from environment	Voltage difference to control board processed	We will test that the collector give a voltage difference	Medium
USTS2	Control Board	Voltage difference from collector	Voltage level responding to the temperature	We will test that control board can power on and giving out the voltage difference	Medium

Table 3-3 Temperature Sensor Subsystem Unit Test

3.2.3 – Rain Sensor Subsystem

Test ID	Module	Input	Output	Test	Priority
USRS1	Rain Reading Collector	Rain conditions from the environment	Voltage difference to control board processor	We will test that the collector gives a voltage difference.	Moderate
USRS2	Control Board	Voltage difference from the collector	Voltage level responding to the water volume on the surface	We will test that the control board can power on and give out the voltage difference.	Moderate

Table 3-4 Rain Sensor Subsystem Unit Test

3.2.4 – Sensor Control Subsystem

Test ID	Module	Input	Output	Test	Priority
USSC1	Analog Digital Converter	Voltage from sensors	Digital values based on the analog signal	We will test that the analog digital converter gives out a numeric value response to the analog signal.	High
USSC2	Sensor Data Package	Digital sensor data from Analog Converter	Meaningful data about environment condition	We will test that the output data is correct to the environment condition.	High
USSC3	Serial Data Sender	Communication rate of a serial connection	A serial connection is ready for transmitting data	We will test that at this point the data will be transmitted to Raspberry Pi.	High

Table 3-5 Temperature Sensor Subsystem Unit Test

3.2.5 – Valve Control Subsystem

Test ID	Module	Input	Output	Test	Priority
UHVC1	Serial Data Receiver	Communication rate of a serial connection	A serial connection is ready for receiving data	We will test that at this point the data can be received through a serial connection.	High
UHSC2	Command Executor	Control command being sent through serial connection	Control signals to the digital pins	We will test that this module will set the digital pins on the Arduino.	High
UHVC3	Relay Module	Command signals from Arduino	High voltage control signal to turn the valve on or off	We will test that we can turn on or off the relay switches by applying the correct signal.	High

Table 3-6 Valve Control Subsystem Unit Test

3.2.6 – Data Processing Subsystem

Test ID	Module	Input	Output	Test	Priority
UIVDP1	USB/Serial Interface	Communication rate of a serial connection	A serial connection is ready for sending and receiving data	We will test that at this point the data can be sent and received from a serial connection.	High
UIDP2	Valve Command Processor	Control command as an array	Correct control command to single local variables	We will test that this module assigns correct values to local variables based on an input array.	High
UIDP3	JSON Message Builder	Sensor data received through a serial connection	A JSON message to be sent to a web service	We will test that the sensor data is built into the correct JSON format.	High

Table 3-7 Data Processing Subsystem Unit Test

3.2.7 – Service Caller Subsystem

Test ID	Module	Input	Output	Test	Priority
UISC1	Response Parser	API response from the Server Layer	An array in JSON format	We will test this module will receive a response and parse it into a JSON format after a HTTP request is returned.	High
UISC2	API Caller	JSON message to be sent to web services	An URI format data being sent to web service	We will test that this module will send a HTTP request in a correct URI format to a web service to be processed.	High

Table 3-8 Service Caller Subsystem Unit Test

3.2.8 – Web Application Subsystem

Test ID	Module	Input	Output	Test	Priority
USWA1	View	User interaction with the web application	User event object specifying the events the user set off	We will test all page inputs including correct and incorrect information and verify the system outputs the correct output.	High
USWA2	Controller	Event object from a user interaction	Query on database system based on event object	We will test that this module will be able to handle the data transfer from database and UI.	High
USWA3	Module	Controller request	A new view model or a model that represents existing data	We will test with different kinds of view models to check against run time errors and null reference exceptions.	High

Table 3-9 Web Application Subsystem Unit Test

3.2.9 – Web Services Subsystem

Test ID	Module	Input	Output	Test	Priority
USWS1	Response Parser	Command data from data base	A response to send back to the Interface Layer based on the database query and received data.	We will test by sending a HTTP request, and then we verify the change on the database, also to check the response to the Interface layer.	High
USWS2	Web Services	URI request from API caller	A response that contains control commands	We will test by sending a HTTP request with different values in the fields of the JSON body, then we verify the response.	High
USWS3	URI Authenticator	HTTP request	A confirm whether the URI is authenticated	We will test by sending a correct URI and an invalid UIR to see how each is handled.	High
USWS4	JSON Converter	HTTP request	Extracted sensor data from JSON body message	We will test by sending a HTTP request with a JSON body, then verify the information saved on database.	High

Table 3-10 Web Service Subsystem Unit Test

3.2.10 – Database Interface Subsystem

Test ID	Module	Input	Output	Test	Priority
USSI1	Stored Procedures	Data queries for the Web Service Module	A SQL table data	We will test this module by using all function calls to request the data, then we verify the database.	High
USSI2	DB Interface	Data request from web services and controllers	Data responding to different requests	We will test this module by sending a HTTP request and making requests from the UI, then we verify how this module responds.	High

Table 3-11 Database Interface Subsystem Unit Test

3.3 Component Tests

3.3.1 – Sensor Layer

Test ID	Subsystem	Input	Output	Test	Priority
CSS1	Soil Moisture Sensor Subsystem	Soil moisture levels from the environment	Soil moisture reading to the Sensor Controller subsystem	We will test that the data received from the soil moisture sensors reflects the changes in the environment.	High
CSR1	Rain Sensor Subsystem	Rain condition from the environment	Rain condition reading to the Sensor Controller subsystem	We will test that the data received from the rain sensor reflecting the changes in the environment.	High
CST1	Temperature Sensor Subsystem	Temperature reading from the environment	Temperature reading to the Sensor Controller subsystem	We will test that the temperature reading received from the sensor is correct to the reading on the thermostat.	High

Table 3-12 Sensor Layer Component Tests

3.3.2 – Hardware I/O Layer

Test ID	Subsystem	Input	Output	Test	Priority
CHS1	Sensor Controller Subsystem	Data from environment conditions	Sensor data as electric signals	We will test this subsystem is able to interface and collect data from all sensors.	High
CHV1	Valve Controller Subsystem	Control Command	Digital signal to turn the valves on or off	We will send some command control through USB connect to this subsystem in order to turn on and turn off the valves.	High

Table 3-13 Hardware I/O Layer Component Tests

3.3.3 – Interface Layer

Test ID	Subsystem	Input	Output	Test	Priority
CID1	Data Processing Subsystem	Sensor data through USB connection	Sensor data in JSON format	We will test that this subsystem receives data from USB connection successfully and verify the JSON format of data after being processed.	High
CSS1	Service Caller Subsystem	JSON format data	JSON message attached with URI	We will use the web service to check the message from the URI.	High

Table 3-14 Interface Layer Component Tests

3.3.4 – Server Layer

Test ID	Subsystem	Input	Output	Test	Priority
CSA1	Web Application Subsystem	User interaction with the web application	User event object specifying the events the user set off	We will test all pages inputs including correct and incorrect information and verify the system outputs the correct output.	High
CSS1	Web Service Subsystem	URI request from API caller	A response to send back to the Interface Layer based on the database query and received data	We will test by sending a HTTP request, and then we verify the change on database, also to check the response to the Interface Layer.	High
CSD1	Database Interface Subsystem	Data request from web services and controllers	Data responding to different requests	We will test this module by sending a HTTP request and making some request from the UI, then we verify how this subsystem responds.	High

Table 3-15 Server Layer Component Tests

3.4 Integration Tests

Test ID	Layer	Input	Output	Test	Priority
IS1	Sensor Layer	Changes in environment conditions	Electric signal on the pins of Arduino	We will test by changing the environment conditions by spilling water on the header of the rain sensor, sinking the header of the soil sensor into water, and pull the temperature of thermal sensor at the same time. Then, we will verify the change of the values of sensor readings.	High
IH1	Hardware I/O Layer	Control command from Raspberry Pi	Operation of the valves and reading values of sensors	We will use Raspberry Pi to send a command to turn the valves on or off, then read sensor values being sent from the Arduino	High
II1	Interface Layer	Change in local variables in the programming running on Raspberry Pi	Change on HTTP request to a web service	We will test by assigning different values of local variables in the programming running on Raspberry Pi. Then, we verify the change on the web service.	High
IW1	Server Layer	User Interaction	Information display to user	We will test by clicking on some objects on UI display and check the information display reflecting the action.	High

Table 3-16 Server Layer Component Tests

3.5 System Validation

Test ID	Validation	Input	Output	Test	Priority
V1	Verify User Account Register	User registers a HICS system in the website	User should be able to activate a HICS system and be able to log into the system	We will test to see that all user information is accepted in the Register User screen and that the user can register and log into the system.	Critical
V2	Verify Valve Control	User interaction with the website to control the valves	The system will control the valves according to user's action	We will test to make sure that the user is able to turn on and turn off the valves correctly.	Critical
V3	Verify Check Status	User interaction with the website to check the status of their system	The website will show all information about their system that the user wants to check	We will test to make sure that user is able check the status of their system and we will verify all information to make sure it is correct.	High
V4	Verify Watering Schedule	User interaction with the website to set the watering schedule	A new watering schedule will be set for system	We will test by setting up water scheduling and waiting to see if it works as scheduled.	High

Table 3-16 Server Layer Component Tests

4. Risks

The following is a list of possible risks that may be encountered while testing the HICS product. The purpose of this section is to identify possible risks early so that they might be avoided during the testing phase.

4.1 Risk Table

Risk ID	Risk	Impact	Severity	Mitigation Strategy
R1	Raspberry Pi can't connect to Internet/Web Server	HICS Central Control Unit will not be able to interface with the web.	High	Ensure that the Raspberry Pi can connect to the internet early in the development process
R2	Raspberry Pi not receiving power	HICS unit will not be able to function	High	Replace any faulty power adapters if necessary
R3	Arduino's not receiving power from Raspberry Pi	HICS unit will be partially or completely unable to function	High	Ensure that the Raspberry Pi can supply enough power over USB to power the Arduino devices
R4	Raspberry Pi/Arduino data communication over serial port not functioning properly	HICS unit will be partially or completely unable to function	High	Thoroughly test the serial port communication between these devices
R5	Relay Boards not wired correctly to irrigation valves	Commands to the irrigation valves will not be able to be processed	Moderate	Double check relay board connections
R6	The web server is down/not responding to requests	HICS unit will not be able to function	High	Ensure that the web server is set up properly and use a dependable service provider

Table 4-1 Risks Table

5. Features To Be Tested

This sections details which requirements will be tested during the testing phase of HICS. The requirements listed below correspond to the various requirements specified in our SRS.

5.1 Customer Requirements

This section describes all the customer requirements listed on our System Requirement Specification document and gives a brief description of how each feature will be tested to meet the acceptance criteria of the requirement.

5.1.1 - Central Control Unit

Description: The central control unit is responsible for all communication between the web application, soil moisture sensors, and water valves. The control unit will transmit the readings from the soil moisture sensors and report the data to the web application. The control unit also controls the water valves, which sets the flow of water to the sprinkler system.

Test Approach: We will test the Raspberry Pi for all physical and functional tests. We will connect all the ports like USB, Ethernet, power, audio, and HDMI and test it to make sure it works correctly. We will also test it for its communication with the web server and the Hardware I/O Layer to ensure its functioning as well.

5.1.2 - Soil Moisture Sensors

Description: In-ground sensors that monitor and report soil moisture levels.

Test Approach: We will provide different soil moisture environments to the sensor and measure its accuracy. It will also be tested against the different physical conditions like pressure and temperature such that it won’t break or malfunction.

5.1.3 - Web Application

Description: The web application will be used to interface with the central control unit. The application is responsible for scheduling watering times as well as interfacing with the central control unit to control the water valves. The web application will be built with a scalable interface to enable the same functionality on a mobile device as it would on a desktop computer while maintaining a similar look and feel.

Test Approach: We will open our web browser on smartphone, tablet and computer. We must be able to see the moisture readings and current schedule for watering on the web application. We will give command such as watering and schedule creation; we must be able to see the updated information.

5.1.4 - Watering Scheduler

Description: The web application can schedule watering times and durations for the sprinkler system on a soil moisture percentage basis. These schedules are determined by other environmental conditions as well such as temperature and time of year. The scheduler will display upcoming watering schedules and allow users to manually create, edit, or delete them.

Test Approach: We will create, edit, and delete watering schedules and test the respond time and efficiency. When we schedule a watering against the city's schedule, the system must warn the user and suggest user to choose different time available.

5.1.5 - Soil Moisture Reports

Description: The web application will generate visual reports of the current soil moisture levels. The reports will be generated as graphic representations of the data and will be filterable by zone, year, month, and day.

Test Approach: Readings collected from each soil moisture sensor will be saved on the database. To test this, we will note down some previous data and compare it with the one we see on the soil moisture report. We will test if the sensor reading corresponds with the specific zone, time, and day.

5.1.6 - User Login

Description: The user must login to the web application using the correct credentials to be able to remotely control features of HICS.

Test Approach: To test the login feature, we will try to login with various right and wrong login credentials. If the right credentials are entered, then only the system should allow the user to login and use the HICS personalized features.

5.1.7 - Active Status

Description: The web application will have a status indicator on the home dashboard to display whether a zone is actively being watered or not.

Test Approach: The active status bar will be tested by comparing the instant status of the HICS system with the active status bar. When not in use, the active status bar should be inactive and while operating, the status bar must be active.

5.1.8 - Rain Sensor

Description: An external rain sensor will monitor the weather conditions and will send an alert to the central control unit in the event of rain. The user preferences will dictate whether this event will turn the sprinklers off or not.

Test Approach: We will put some water on the rain sensor, which must stop watering at that moment. We will also match if the rain sensor has rain; the soil moisture sensor on the ground must start increasing soil moisture.

5.1.9 - Web Hosting Service

Description: There will be a web service that hosts our web application with an associated URL for domain user access.

Test Approach: We will test our application on the web. After we have hosted our web application, we must be able to run our application from the internet.

5.1.10 - Database Management System

Description: There will be a DBMS provided by our web hosting service provider that will store all data from soil sensor readings as well as all the information for the web application. This information includes user account information, settings, reports, and weather data necessary for the application to perform as expected.

Test Approach: We will test our web service by making queries to the database. Our queries should be able to give us result with the information that we ask for.

5.1.11 - Device Registration

Description: A user must register their HICS device with their account in order to interface with it via the web application. The central control unit will have a unique serial number on it that will be used to register the device to a user account.

Test Approach: On the registration page, we will enter a valid serial number and register our product. If the serial number is not valid, our system should not allow registration.

5.1.12 - Region Grouping

Description: In HICS each zone is individually controlled however a user may group their sprinkler zones into regions through the web application. This functionality will allow the user to setup watering schedules for multiple zones at once without having to configure each individual zone.

Test Approach: We will have our first valve and sensor grouped as Zone A, and the two valves and sensors as Zone B. When we look at the moisture on zone A, we should see only one reading. If we turn on Zone A, only the first valve must run.

5.1.13 - Auto Off

Description: The user can set an auto off timer via the web application that terminates watering if the duration lasts longer than the set time.

Test Approach: For testing, we will put few seconds, e.g. 10 seconds as maximum time to water soil. We won't dip the soil moisture sensor on water and let the watering continue, the watering must stop at 10 seconds.

5.1.14 - Temperature Sensor

Description: A temperature sensor will monitor the temperature conditions at the central control unit's location. If the sensor detects temperatures around freezing (34° F), the HICS will shut off the watering valves and notify the user to prevent freeze hazards.

Test Approach: We will test the temperature for freezing conditions. We will put ice on the temperature sensor, in which the watering should stop automatically.

5.2 Packaging Requirements

This section describes all the packaging requirements listed on our System Requirement Specification document and gives a brief description of how each feature will be tested to meet the acceptance criteria of the requirement.

5.2.1 - Control Unit Housing

Description: The components of the control unit, including its accessories, will be contained inside a mountable hard plastic container.

Test Approach: All the components will be tested if it fits inside the housing or not. It must be tight enough such that it won't move anywhere else.

5.2.2 - Soil Moisture Sensors Packaging

Description: (4) Soil sensors will be included in the final packaging of HICS. They will be located within their own boxes and will have corresponding documentation that describes their proper installation. The user will be able to purchase additional soil moisture sensors and add them to their system.

Test Approach: Soil moisture sensors will be tested with the package that we provide. The sensors must be easy to place on the ground. We will make sure that the proper documentation gives good knowledge about installing sensors.

5.2.3 - Rain Sensor Packaging

Description: (1) Rain sensor will be included in the final packaging of HICS. It will be located within its own box and will have corresponding documentation that describes its proper instillation.

Test Approach: Rain sensor comes with its own box. We will test that the box properly fits the sensor and secures it.

5.2.4 - Connecting Cables

Description: (1) Power cable and (1) Ethernet cable will be included in the final packaging of HICS. Their intended purpose is to be used with the central control unit.

Test Approach: The cables in the pack will be tested before putting inside the final box. We will check the quantity as well as the quality of cables.

5.2.5 - Temperature Sensor Packaging

Description: (1) Temperature sensor will be included in the final packaging of HICS. It will be located within its own box and will have corresponding documentation that describes its proper instillation.

Test Approach: Temperature sensor box will be tested for its perfect fit. We will make sure it the housing won’t affect the temperature reading.

5.3 Performance Requirements

This section describes all the performance requirements listed on our System Requirement Specification document and gives a brief description of how each feature will be tested to meet the acceptance criteria of the requirement.

5.3.1 - Sensor Accuracy

Description: Proper sensor readings of soil moisture levels must be captured to ensure accurate and efficient watering schedules. This requires solid construction of the sensor modules and proper software analysis of the data provided by the sensors.

Test Approach: All sensors will be tested for their accuracy. To get the accurate reading from the soil moisture sensor, the sensors will be placed on the right depth of soil. Rain sensor and temperature sensor will be on the rooftop. While coding, we will make sure the calibration will be exact.

5.3.2 - Device Power Malfunction

Description: If the web application fails to receive communication from the central control unit for a specified period of time, it will notify the user that the system is offline.

Test Approach: We will test each device and make sure there won't be any power malfunction under our specified criteria. In case of damage to device, we will test the device to make sure it will go to fail-safe mode.

5.3.3 - Response Time To Watering

Description: Watering grass when dry is important. But delay of watering for few seconds does no harm to the grass or anyone. After getting command to start watering, HICS may take few seconds to actually have the sprinkles dispensing water. So, the response time is nothing to worry about.

Test Approach: We will initiate a web request for watering instantly. We will note the time difference between the service call done from the web and the actual watering. We will also schedule a watering and note the time for the watering to happen by itself. We will test this multiple times and under different environments like varying temperature, changing moisture and different internet speed to make sure our readings are consistent. It should not take more than 10 seconds for our system to respond.

5.4 Safety Requirements

This section describes all the safety requirements listed on our System Requirement Specification document and gives a brief description of how each feature will be tested to meet the acceptance criteria of the requirement.

5.4.1 - Soil Moisture Sensor Insulation

Description: The electronic components of the soil moisture sensors must be properly insulated against external environmental conditions. This is to ensure that they do not malfunction while in use.

Test Approach: The wires connecting the sensor must not touch any conductor or water. We will test that the insulation is waterproof on the sensor.

5.4.2 - Central Control Unit Temperature

Description: Efficient airflow and low temperatures must be established while the central control unit is in operation. Computer fans and proper ventilation in the housing container will be used to cool the device and prevent it from overheating and possibly starting a fire.

Test Approach: We will test the HICS central control unit under different temperatures. We will put it in different environment temperatures ranging from 10-130 degrees Fahrenheit and make sure each device is functioning correctly.

5.4.3 - Proper Wiring of Central Control Unit

Description: The central control unit will be powered from an external wall plug. The source will be 120v and improper wiring could leave the user exposed to currents in the range of 15-20 amps. It is crucial that the control unit is wired properly to accept this input and cause no harm to the device or the user.

Test Approach: We will test that the wires coming and going to the central control unit are not bare anywhere and are connected properly.

5.5 Maintenance and Support Requirements

This section details the requirements for ongoing use of HICS after its final delivery to the user. Achieving maximum long-term performance is the key to reducing product maintenance. These requirements aim to minimize system errors and hardware failures.

5.5.1 - Sensor Maintenance

Description: The soil and rain sensors will be replaceable in the event of hardware failure.

Test Approach: We will add replace a sensor and test if our system recognizes the new sensor. We will also add and remove few sensors to make sure that sensor mapping works correctly.

5.5.2 - Scalability

Description: The central control unit will support adding additional valves or soil sensors.

Test Approach: To test the scalability, we will add one more relay module and control the valves. If we can take moisture reading and control the valve for one set, we can do it for multiple set of valves and sensors.

5.6 Other Requirements

This section describes all the other requirements listed on our System Requirement Specification document and gives a brief description of how each feature will be tested to meet the acceptance criteria of the requirement.

5.6.1 - Mapping the Soil Moisture Sensors to Irrigation Valves

Description: HICS requires that the user properly maps the soil moisture sensors to the irrigation valves. The central control unit will feature a diagram showing the one-to-one relationship between soil moisture sensor inputs and irrigation valve inputs.

Test Approach: To test the mapping, we will take the reading of the specific soil sensor and turn on the valve associated with it. We will also make sure it's in the right zone by turning on the valves of specific zones individually.

5.6.2 - Browser Support

Description: The web application should be scalable and responsive in all of the latest versions of Mozilla Firefox, Google Chrome, and Internet Explorer.

Test Approach: We will open our web application on all supporting browsers and make sure we can use all features without any trouble. We will walk through every process that our system allows through web.

5.6.3 - Application Security and Privacy

Description: All user data and account information will be stored in a secured DBMS. Users will only be allowed to access their HICS devices and account information if they provide the correct login credentials. In addition, all user passwords will be encrypted and stored only as hash values.

Test Approach: We will try to login with incorrect credentials. The system must not allow access. We will make sure the passwords are encrypted and not available to be seen by anyone else.

5.6.4 - User Administration Account

Description: The web application will allow for administrative accounts to create, edit, and delete user accounts and their associated devices.

Test Approach: We will test all the functionalities that an admin can do. We will try to access the database and look up user details. We will change information on our self-made user account and compare the change by logging as that user for that account again.

6. Features Not To Be Tested

This section will detail the requirements of HICS that will not be tested during the testing phase. Most of the requirements are not to be tested either cannot be tested or their functionality has no impact on how the system operates.

6.1 Customer Requirements

All customer requirements will be tested.

6.2 Packing Requirements

6.2.1 - User Manual

Description: A user manual will be presented to the user that details the use and operation of HICS. It will be provided in the final packaging as a CD-ROM and will also be made available online via the web application.

Reasoning: User manual needs no testing. It is just typed instructions for the user.

6.2.2 - Software Application

Description: The HICS software should be a stand-alone web application that requires no set up from the user. The user must register an account with a valid product serial number to access the web application. Once they are logged into their account and their device has been connected, they will be able to access the features of the HICS.

Reasoning: We will test the application by registering the serial number.

6.3 Performance Requirements

All safety requirements will be tested.

6.4 Safety Requirements

All safety requirements will be tested.

6.5 Maintenance and Support Requirements

6.5.1 - Software Maintenance

Description: All source code files and relevant documentation must be available for software maintenance and troubleshooting. The required files and documentation will be available via the GitHub repository. The software will be split into loosely coupled parts so it would be easier to make changes and improvements to the system over time.

Reasoning: Time constraint. Our software is made from one of the modern and compatible technologies, so there will likely be no issues with our software for a number of years. Also this is a year term project so there is no need to maintain the software.

6.5.2 - Source Code Documentation

Description: All source code files required for HICS will be extensively commented to support future updates and troubleshooting.

Reasoning: Cannot test documentation.

6.5.3 - Troubleshooting Guide

Description: A troubleshooting guide will be presented to the user that details some possible problems that they might encounter in typical use and their appropriate solutions. Similar to the user manual, it will be provided in the final packaging as a CD-ROM and will also be made available online via the web application.

Reasoning: Time constraint. We assume our Guide and User Manual will be clear enough that it will need no reviewing again.

6.6 Other Requirements

6.6.1 - Central Control Unit Mounting

Description: The central control unit must be mounted in a location where it can interface with wires from the water valves and soil moisture sensors. It must also be within range of Ethernet and power outlets.

Reasoning: People may mount their central control unit anywhere they want.

7. Testing Approaches

This section will detail the approaches our team will take for testing the HICS product. These tests are to ensure that each critical requirement specified in our SRS is met and tested to completion. This section will also detail the tools and methods we will be using throughout the testing phase project.

7.1 Strategy

The overall strategy approach for HICS's implementation is to start testing from the beginning of the prototype phase. This means that each component will be tested as it is finished. This approach is to prevent integration of untested and incomplete components.

Once component tests are successful we will then begin integration testing each completed component one by one. This means that once a component is integrated with another we test the functionality of the integration before adding in another component. Each integration will be tested to ensure each sub component functions as it should. We will also test as much overall functionality that is possible at the point of integration to provide a more throughout understanding before system testing.

Once integration is successful the team will then begin the overall testing of the system. The strategy is test all functionality of the system as a whole to ensure that every requirement has been met and works properly. The system testing phase is where we will also begin stress testing as well as environment testing our sensors.

7.2 Tools

The following is a list of tools to be used for development testing in HICS:

- Glimse
- RESTconsole
- JSLint
- Web Performance Load Tester (LITE)
- Ninject
- Entity Framework 6.0
- Multi-Meter

7.3 Core Functionality

The following is a list of core functions that are paramount to HICS functionality and are to be focused during testing.

- Register User
- Login/logout
- Collect and view sensor readings (soil moisture, temperature, and rain)
- Start/stop watering from web application
- Create/edit/delete watering schedule
- Stop watering for rain or freezing conditions

7.4 Test Metrics

Priority	Description	Pass Criteria	Fail Criteria
1 - Critical	Features that the system relies on for functionality. These features must be completed for the system to function.	100%	Less than 100%
2 – High	Features that are highly important to the system but are not completely required for the system to function.	90%+	Less than 90%
3 - Moderate	Features that improve the quality of the system but are not required for the system to function.	75%+	Less than 75%
4 - Low	Features reserved for future requirements or extras that do not impact the system functionality.	0%+	N/A

Table 7-1 Test Metrics Table

8. Item Pass/Fail Criteria

This section describes in detail how pass and fail criteria for each individual test will be regulated. This will help our team to identify the acceptable or not acceptable standards for each portion in the system. It will also clearly define the success or failure of each individual test.

8.1 Hardware Tests

Test ID	Hardware	Pass Criteria	Fail Criteria
H1	Raspberry Pi B+	• Store data • Processes HTTP request • Relay commands to the Arduino	• Does not store data • Does not process all HTTP requests • Does not relay commands to the Arduino
H2	Arduino	• Interface with sensors • Send sensor values to Raspberry Pi • Relay commands to the Relay Switch Module	• Does not interface with sensor • Does not send sensor values to the Raspberry Pi • Does not relay commands to the Relay Switch Module
H3	Soil moisture sensor	• Measure the moisture level of the soil correctly	• Does not measure the soil moisture sensor correctly
H4	Temperature sensor	• Measure temperature of the environment correctly	• Does not measure temperature of the environment correctly
H5	Rain sensor	• Report the rain condition of the environment correctly	• Does not report the rain condition of the environment correctly
H6	Relay Switch Module	• Open/close circuit for all 8 channels	• Does not open/close circuit for any channel

Table 8-1 Hardware Tests Criteria

8.2 Unit Tests

Test ID	Hardware	Pass Criteria	Fail Criteria
USSS1	Soil Moisture Reading Collector	• Return a voltage difference reflecting the change in soil moisture	• Does not return a voltage difference reflecting the change in soil moisture
USSS2	Control Board	• Return a voltage level responding to the moisture reading	• Does not return a voltage level responding to the moisture reading
USTS1	Temperature Reading Collector	• Return a voltage difference to control board processed	• Does not return a voltage difference to control board processed
USTS2	Control Board	• Return a voltage level responding to the temperature change	• Does not return a voltage level responding to the temperature change
USSS2	Control Board	• Return a voltage level responding to the moisture reading	• Does not return a voltage level responding to the moisture reading
USRS1	Rain Reading Collector	• Return a voltage difference reflecting the change in rain condition	• Does not return a voltage difference reflecting the change in rain condition
USSC1	Analog Digital Converter	• Return a digital value response to the analog signal	• Does return a digital value response to the analog signal
USSC2	Sensor Data Package	• Return correct reading about the environment condition	• Does not return correct reading about the environment condition
USSC3	Serial Data Sender	• Initiate a serial connection to send data to the Raspberry Pi	• Does not send data to the Raspberry Pi
UHVC1	Serial Data Receiver	• Initiate serial connection to receive data from the Raspberry Pi	• Does not receive data from the Raspberry Pi

UHSC2	Command Executor	• Set the level at digital pins on the Arduino	• Does not set the level at digital pins on the Arduino
UHVC3	Relay Module	• Open/close circuit for all 8 channels	• Does not open/close circuit for any channel
UIVDP1	USB/Serial Interface	• Initiate a serial connection to transmit data to the Arduino	• Unable to send/receive data from the Raspberry to the Arduino
UIDP2	Valve Command Processor	• Assign correct values to local variables based on a input array	• Does not assign/assign wrong value based on a input array
UIDP3	JSON Message Builder	• Build data as a JSON object	• Does not build/build a wrong JSON object
UISC1	Response Parser	• Receive responses from the Web service • Parse response into an JSON format object	• Unable to receive response from the Web service • Does not parse response into an JSON format object
UISC2	API Caller	• Build an URI request • Send an URI request to the Web service	• Unable to build an URI request • Unable to send an URI request to the Web service
USWA1	View	• Show UI objects for user to interact	• Unable to show
USWA2	Controller	• Handle the data transfer from database and UI	• Unable to handle the data transfer from database and UI
USWA3	Module	• Set up a view model as requirement	• Unable to set up a view model as requirement
USWS1	Response Parser	• Create a JSON object containing control commands • Send a response back to the Raspberry Pi	• Unable to create JSON object containing control commands • Unable to send the response back to the Raspberry Pi

USWS2	Web Services	• Receive JSON object of sensor data • Update sensor data on database • Retrieve new control commands from database and relaying to the Response Parser	• Unable to receive JSON object of sensor data • Unable to update sensor data on database • Unable to retrieve new control commands from database and relaying to the Response Parser
USWS3	URI Authenticator	• Verify a URI request	• Unable to verify a URI request
USWS4	JSON Converter	• Receive URI request • Convert sensor data in URI request	• Unable to receive URI request • Unable to convert sensor data in URI
USSI1	Stored Procedures	• Store all database query functions • Response correctly when database functions are called	• Unable to store all database query functions • Unable to response correctly when functions are called
USSI2	DB Interface	• Set up an interface between database and web services or controllers	• Unable to set up an interface between database and web services or controllers.

Table 8-2 Unit Tests Criteria

8.3 Component Tests

Test ID	Hardware	Pass Criteria	Fail Criteria
CSS1	Soil Moisture Sensor Subsystem	• Measure the moisture level of the soil correctly	• Does not measure the soil moisture sensor correctly
CSR1	Rain Sensor Subsystem	• Report the rain condition of the environment correctly	• Does not measure temperature of the environment correctly
CST1	Temperature Sensor Subsystem	• Measure the temperature of the environment correctly	• Does not report the rain condition of the environment correctly
CHS1	Sensor Controller Subsystem	• Read the analog data form sensor modules • Convert data to digital values • Send digital values to the Raspberry Pi	• Unable to read data form sensor module • Unable to convert data to digital values • Unable to send digital values to the Raspberry Pi
CHV1	Valve Controller Subsystem	• Receive control commands form the Raspberry Pi • Execute commands to turn on/off the valves	• Unable to receive control commands form the Raspberry Pi • Unable to execute commands to turn on/off the valves
CID1	Data Processing Subsystem	• Map control commands to local variable • Convert sensor data to JSON object	• Unable to map control commands to local variable • Unable to convert sensor data to JSON object
CSS1	Service Caller Subsystem	• Attach JSON object of data sensor to URI • Make an URI request to the Web service	• Unable to attach JSON object of data sensor to URI • Unable to make an URI request to the Web service
CSA1	Web Application Subsystem	• Display GUI object for user to interact	• Unable to show GUI object or show a wrong object

CSS1	Web Service Subsystem	• Process HTTP request with sensor data • Return response with control commands • Update or retrieve data from database	• Unable to process HTTP request • Unable to return response with control commands • Unable to update or retrieve data from database
CSD1	Database Interface Subsystem	• Store database functions • Store information about sensor data, user profile, control commands	• Unable to store DB functions • Unable to information about sensor data, user profile, control commands

Table 8-3 Component Tests Criteria

8.4 Integration Tests

Test ID	Hardware	Pass Criteria	Fail Criteria
IS1	Sensor Layer	• Receive information about environment conditions to electric signal including soil moisture, temperature, rain condition	• Unable to convert information about environment conditions to digital values • Report wrong information about environment conditions
IH1	Hardware I/O Layer	• Convert the electric signals from sensors to digital values • Execute control commands to turn on/off the valves • Interface with Interface Layer	• Convert the electric signals from sensors to digital values • Unable to execute control commands to turn on/off the valves • Fail to send and receive data with Interface Layer
II1	Interface Layer	• Interface with Hardware I/O Layer • Parse data into proper format • Communicate with Server Layer	• Fail to send and receive data with Interface Layer • Fail to parse data into format • Unable to send a HTTP request or receive response

IW1	Server Layer	• Show GUI for user to interact • Store information about sensor, command and user profile • Process information based on update from sensor data	• Unable to show GUI object or show a wrong object corresponding user's actions. • Unable to store information about sensor, command and user profile • Fail to process information based on update from sensor data. • Fail to receive data.

Table 8-4 Integration Tests Criteria

8.5 System Validation

Test ID	Hardware	Pass Criteria	Fail Criteria
V1	Verify User Account Register	• User should be able to activate HICS system and able to log in to the system	• Fail to activate HICS system • Fail to store user's information • User is unable to log in to the system
V2	Verify Valve Control	• The system will control the valve according to user's action	• System is unable to control the valves responding to user's action, user's schedule, and change in environment conditions
V3	Verify Check Status	• User interaction with the website to check the status of system	• Fail to show the correct information of current status of system
V4	Verify Watering Schedule	• User interaction with the website to set the watering schedule	• Unable to store the water setting • System is unable to follow user's water settings

Table 8-5 System Validation Tests Criteria

9. Testing Deliverable

This section shortly explains what will be the output of the product after completion of the product. It will explain the expected deliverables and the method we will use to have the tests operated. It will document all the necessary tests performed.

9.1 System Test Plan

System test plan briefly explains what components of HICS are to be tested and how to carry out the test.

9.2 Test Case Specification

Each test case will consist of following terms:

- **Test Case ID:** A unique ID associated with the test case.
- **Description:** Details of the test performed and description of the reason behind the test.
- **Pre-Condition:** The condition the system is in while performing the test.
- **Post Condition:** The condition the system is expected to be in after the test is performed.
- **Input:** The input given to the test case.
- **Expected Output:** The expected output from the test case with the specific input.
- **Process:** Step by step walkthrough of the test performed.

9.3 Test Case Results

Each test case results will be documented with following properties:

- **Test Case ID:** A unique ID associated with the test case.
- **Tester Name:** Name of the person who perform the test.
- **Test Date:** Date the test was performed**.**
- **Inputs:** All inputs provided to perform the test case.
- **Expected Output:** The output that was expected with the entered input.
- **Actual Output:** The actual output you see.
- **Result of Test:** Pass/ Fail.
- **Defect ID:** A unique ID associated with the specific type of defect.
- **Description:** Details explaining the reason behind pass/fail of the test case.

9.4 Defects

Each defect will be documented with following properties:

- **Test Case ID:** A unique ID associated with the test case.
- **Defect ID:** A unique ID associated with the specific type of defect.
- **Result ID:** A unique ID associated with the specific type of defect.
- **Error Log:** A warning message or information related with the specific defect.
- **Description:** Description of the defect explaining the reason of the bug.
- **Defect Fix Status:** The condition of the bug if it's been fixed or not.

10. Test Schedule

10.1 MS Project Plan-System Phase

WBS	Task Name	Planned Start	Planned Finish	BCWS (hrs)
2.5	**System Testing Phase**	**4/6/15**	**5/8/15**	**137**
2.5.1	**System Testing Phase I (Hardware Testing)**	**4/6/15**	**4/10/15**	**7**
2.5.1.1	Raspberry Pi B+	4/6/15	4/10/15	1
2.5.1.2-3	Arduino	4/6/15	4/10/15	1
2.5.1.4	Soil moisture sensor	4/6/15	4/10/15	1
2.5.1.5	Temperature sensor	4/6/15	4/10/15	1
2.5.1.6	Rain sensor	4/6/15	4/10/15	1
2.5.1.7	Relay Switch Module	4/6/15	4/10/15	1
2.5.2	**System Testing Phase II (Unit Testing)**	**4/6/15**	**4/17/15**	**62**
2.5.2.1	Soil Moisture Reading Collector	4/6/15	4/17/15	1
2.5.2.2	Control Board	4/6/15	4/17/15	1
2.5.2.3	Temperature Reading Collector	4/6/15	4/17/15	1
2.5.2.4	Control Board	4/6/15	4/17/15	1

2.5.2.5	Rain Reading Collector	4/6/15	4/17/15	1
2.5.2.6	Control Board	4/6/15	4/17/15	1
2.5.2.7	Analog Digital Converter	4/6/15	4/17/15	1
2.5.2.8	Sensor Data Package	4/6/15	4/17/15	2
2.5.2.9	Serial Data Sender	4/6/15	4/17/15	2
2.5.2.10	Serial Data Receiver	4/6/15	4/17/15	2
2.5.2.11	Command Executor	4/6/15	4/17/15	3
2.5.2.12	Relay Module	4/6/15	4/17/15	3
2.5.2.13	USB/Serial Interface	4/6/15	4/17/15	2
2.5.2.14	Valve Command Processor	4/6/15	4/17/15	3
2.5.2.15	JSON Message Builder	4/6/15	4/17/15	3
2.5.2.16	Response Parser	4/6/15	4/17/15	3
2.5.2.17	API Caller	4/6/15	4/17/15	2
2.5.2.18	View	4/6/15	4/17/15	4
2.5.2.19	Controller	4/6/15	4/17/15	5
2.5.2.20	Model	4/6/15	4/17/15	3
2.5.2.21	Response Parser	4/6/15	4/17/15	3

2.5.2.22	Web Services	4/6/15	4/17/15	5
2.5.2.23	URI Authenticator	4/6/15	4/17/15	2
2.5.2.24	JSON Converter	4/6/15	4/17/15	3
2.5.2.25	Stored Procedures	4/6/15	4/17/15	3
2.5.2.26	DB Interface	4/6/15	4/17/15	2
2.5.3	**System Testing Phase III (Component Testing)**	**4/13/15**	**4/24/15**	**37**
2.5.3.1	Soil Moisture Sensor Subsystem	4/13/15	4/24/15	3
2.5.3.2	Temperature Sensor Subsystem	4/13/15	4/24/15	3
2.5.3.3	Rain Sensor Subsystem	4/13/15	4/24/15	3
2.5.3.4	Sensor Controller Subsystem	4/13/15	4/24/15	4
2.5.3.5	Valve Controller Subsystem	4/13/15	4/24/15	4
2.5.3.6	Data Processing Subsystem	4/13/15	4/24/15	4
2.5.3.7	Service Caller Subsystem	4/13/15	4/24/15	4
2.5.3.8	Web Application Subsystem	4/13/15	4/24/15	4
2.5.3.9	Web Service Subsystem	4/13/15	4/24/15	5
2.5.3.10	Database Interface Subsystem	4/13/15	4/24/15	3

2.5.4	**System Testing Phase IV (Integration Testing)**	**4/20/15**	**5/1/15**	**17**
2.5.4.1	Sensor Layer	4/20/15	5/1/15	2
2.5.4.2	Hardware I/O Layer	4/20/15	5/1/15	3
2.5.4.3	Interface Layer	4/20/15	5/1/15	6
2.5.4.4	Server Layer	4/20/15	5/1/15	6
2.5.5	**System Testing Phase V (System Validation)**	**5/4/15**	**5/8/15**	**14**
2.5.5.1	Verify User Account Register	5/4/15	5/8/15	3
2.5.5.1	Verify Valve Control	5/4/15	5/8/15	3
2.5.5.1	Verify Check Status	5/4/15	5/8/15	4
2.5.5.1	Verify Watering Schedule	5/4/15	5/8/15	4

Table 10-1 Test Phase Schedule Table

11. Approval

Name	Role	Signature	Date
Keith Aholt	Project Sponsor		
Mike O’Dell	Project Supervisor		
Billy Haile-Mariam	Team Leader		
Tung Vo	Team Member		
Gautam Adhikari	Team Member		
Jeremiah O’Connor	Team Member		

Table 11-1 Approval Table