AN AGRICULTURAL GUIDE USING CLOUD COMPUTING AND BIG DATA ANALYSIS

Software Test Documentation

Group-06
Maj. Ruhul Ambia Rimon (201514065)
Rishad Islam (201514168)
Shahir Rahman (201514099)
Mahdiul Islam (201514084)
Hosnee Mobashir (201514054)
Helalur Rahman Khan (201614048)

Version-01, 5. August 2019

Contents

1 Introduction

Testing documentation involves the documentation of artifacts which should be developed before or during the testing of Software. Documentation for Software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing etc. This section includes the description of some commonly used documented artifacts related to Software testing such as: 1) Test Plan, 2) Test Scenario, 3) Test Case, 4) Traceability Matrix. The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan must identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

1.1 Objective

In the project of Agricultural Guide, apart from hardware resources, software resources will be equally demanding. To fulfill the process the following software or applications will be required:

- We will a build an android based application for the specific users and there will be an web based application for the system developers.
- The android applications will provide the aforementioned services to the user through their smartphones in an organized way and the web-based application will help the developers to monitor the performance of the system and different sensors and remotely control different functions of the system.
- The ultimate objective is to provide a cultivator with a full package of information and control over his field still keeping the human factors of the cultivation intact. It will provide user the necessary time to be prepared for upcoming disasters so that the damage will be reduced to a great length.

1.2 Testing Strategies

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. We will follow the following testing strategies:

- Methodical testing strategy: It tests the functions and status of software according to the checklist, which is based on user requirements. This strategy is also used to test the functionality, reliability, usability, and performance of the software.
- Process-oriented testing strategy: It tests the software according to already existing standards such as the IEEE standards. In addition, it checks the functionality of the software by using automated testing tools.
- Model-based testing strategy: This strategy tests the functionality of the software according to the real world scenario (like software functioning in an organization). It recognizes the domain of data and selects suitable test cases according to the probability of errors in that domain.

2 Test Items

We will mainly focus to ensure the following items to be tested:

- Requirements specification
- Design specification
- Features (availability, response time)

3 Features To be Tested

3.1 Sign Up

- Input: Provide valid mobile no.
- Output: Receive confirmation text through the mobile no.

(a) Testing Type: Integration Testing. User information is taken from system and using that system a verification is done through confirmation text message. User input and confirmation text, two individual units combine to produce output.

(b) Criteria Assessment: Reliability and security. This testing verifies the user and allows authorizes user to use the system.

3.2 Log In

- Input: Provide email id and password.
- Output: Login to system.

(a) Testing Type: Integration Testing. User information is provided to system. System checks authorization of the information and validates the user.

(b) Criteria Assessment: Security and reliability. This testing allows authorized user to use the system.

3.3 Weather Data

- Input: Sensor data from hardware interface and weather database.
- Output: Display weather information.

(a) Testing Type: Integration Testing. Data is retrieved from the database and displayed.

(b) Criteria Assessment: Reliability, availability and correctness. This testing checks whether the data displayed.

3.4 Retrieving Data from Firebase

- Input: Sensor data from hardware interface and user information.
- Output: Retrieved data is used for system purposes.

(a) Testing Type: Integration Testing. Data is retrieved from the database and utilised.

(b) Criteria Assessment: Reliability, availability, correctness, response time and Data integrity. This testing checks whether the data is stored in Firebase properly and can be retrieved without any error.

3.5 Controlling Servo Motor

- Input: Turning on/off motor switch in system.
- Output: Switching on/off motor in hardware interface.

(a) Testing Type: Integration Testing. Integration between software and hardware components is being tested.

(b) Criteria Assessment: Reliability,response time and efficiency. This testing checks whether the servo motor in the hardware interface can be operated smoothly through the software.

3.6 Pump Control Button

- Input: Toggling Control Button
- Output: Controlling the action of pump.

(a) Testing Type: Unit Testing. Checking if the button is working properly through the action of the pump in hardware interface.

4 Software Testing

4.1 Development Testing

We have done the following unit testing for our system:

4.1.1 Pump Control Button

- Input: Toggling Control Button
- Output: Controlling the action of pump.

(a) Testing Type: Unit Testing. Checking if the button is working properly through the action of the pump in hardware interface.

4.2 Release Testing

We have done the following integration testing for our system:

4.2.1 Sign Up

- Input: Provide valid mobile no.
- Output: Receive confirmation text through the mobile no.

(a) Testing Type: Integration Testing. User information is taken from system and using that system a verification is done through confirmation text message. User input and confirmation text, two individual units combine to produce output.

(b) Criteria Assessment: Reliability and security. This testing verifies the user and allows authorizes user to use the system.

4.2.2 Log In

- Input: Provide email id and password.
- Output: Login to system.

(a) Testing Type: Integration Testing. User information is provided to system. System checks authorization of the information and validates the user.

(b) Criteria Assessment: Security and reliability. This testing allows authorized user to use the system.

4.2.3 Weather Data

- Input: Sensor data from hardware interface and weather database.
- Output: Display weather information.

(a) Testing Type: Integration Testing. Data is retrieved from the database and displayed.

(b) Criteria Assessment: Reliability, availability and correctness. This testing checks whether the data displayed.

4.2.4 Retrieving Data from Firebase

- Input: Sensor data from hardware interface and user information.
- Output: Retrieved data is used for system purposes.

(a) Testing Type: Integration Testing. Data is retrieved from the database and utilised.

(b) Criteria Assessment: Reliability, availability, correctness, response time and Data integrity. This testing checks whether the data is stored in Firebase properly and can be retrieved without any error.

4.2.5 Controlling Servo Motor

- Input: Turning on/off motor switch in system.
- Output: Switching on/off motor in hardware interface.

(a) Testing Type: Integration Testing. Integration between software and hardware components is being tested.

(b) Criteria Assessment: Reliability,response time and efficiency. This testing checks whether the servo motor in the hardware interface can be operated smoothly through the software.

5 Hardware Testing

6 Pass/Fail Criteria

Pass Fail Criteria		
Test Items	Pass	Fail

7 Testing Schedule

Using Microsoft Project we have calculated the follwoing test schedule:

✓	**20 days**	**Unit Testing**	**Mon 7/15/19**	**Fri 8/9/19**	
✓	2 days	Review modular code	Mon 7/15/19	Tue 7/16/19	
✓	2 days	Test components module	Wed 7/17/19	Thu 7/18/19	
✓	2 days	Modify code	Fri 7/19/19	Mon 7/22/19	62
✓	4 days	Re Test modify code	Tue 7/23/19	Fri 7/26/19	63
✓	10 days	Unit Testing complete	Mon 7/29/19	Fri 8/9/19	64
✓	**10 days**	**Integration testing**	**Wed 8/14/19**	**Tue 8/27/19**	
✓	1 day	Test module integration	Thu 8/15/19	Thu 8/15/19	
✓	1 day	Modify code	Wed 8/21/19	Wed 8/21/19	67
✓	3 days	Re Test modify code	Thu 8/22/19	Mon 8/26/19	68
✓	1 day	Completion of the integration testing	Tue 8/27/19	Tue 8/27/19	69

Figure 1: Testing Schedule

8 Environmental Requirements

8.1 Hardware

For implementation of our Hardware interface we used following products:

1. Arduino Mega
2. Node MCU
3. BME 280 Sensor.
4. Soil Moisture Sensor

5. Servo Motor
6. Water Pump
7. 4 Channel Relay Module
8. Servo Controller Module
9. Rain Meter
10. Air Speed Sensor
11. Hardware Prototype

8.2 Software

We have implemented a mobile application for using our system. To use the app we will require a Android device which supports at least Android 4.1 JellyBean and network connection.

8.3 Tools

For implementation of the software part of our system we have used the following tools:

1. Android Studio for building the mobile application.
2. Firebase as our database.
3. Putty for interfacing the Arduino

9 Risks and Assumptions

We have analysed the following risks for our project:

Risk Id.	Risk Description	Mitigation Plan (what to do to avoid the risk occurring)	Contingency Plan (what to do if the risk occurs)	Impact (what the impact will be to the project if the risk occurs)	Likelihood of occurrence (e.g., %, or high / medium / low)
1.	Unavailability of enough large data set to predict the weather and agricultural tips correctly.	Save the data received from the sensors and save enough data in the database.	Take help from external dataset like Accuweather and Google Dataset	The output of the project may become faulty.	45%
2.	Inability to provide enough awareness among the people regarding flood detection	Allow users to provide emergency contact and current place properly.	If users do not provide the information then ask them to provide the info as soon possible.	The flood detection feature will lose its purpose.	30%
3.	Unable to Integrate between hardware and software portion of the system	Give enough effort to to provide easy and smooth integration between hardware and software	Have to find a wat to integrate the system	The project may not fulfill its aim and objectives properly	40%
4.	The project may miss the deadline	The members of the group must be more robust and agile to complete the project within the given schedule	The basic features and initial structure of the system must be established first later on advanced features can be established.	The client can be unsatisfied.	50%

Figure 2: Project Risks