CPET 491
(W.I.S.E)

WEATHER INTELLIGENT SENSING EXPERT

Student: Michael Van Auken

Advisor: Prof. Hadi Alasti

Instructor: Prof. Paul I. Lin

12/9/2016

Outline

2

- Problem Statement
- Solution
- System Requirements
- System Design
- System hardware
- Testing
- Validation/Risk
- Project Schedule
- Lessons Learned
- Q/A?
- Demo

Problem Statement

3

- How to be more efficient with water usage in lawn care
  - On average an American family uses 320 gallons of water per day.
  - 30% of that water is devoted to the outdoor use.
  - 50% percent of it is wasted lost due to wind and evaporation
  - High water bills
  - Over watering lawn causing damage to soil and lawn

Source : [1]

Solution

4

- An Intelligent weather monitoring system that controls a irrigation system
  - Reduces the amount of waste water
  - Lower water bill
  - Less maintenance to lawn
  - Reduces lost from wind and other elements

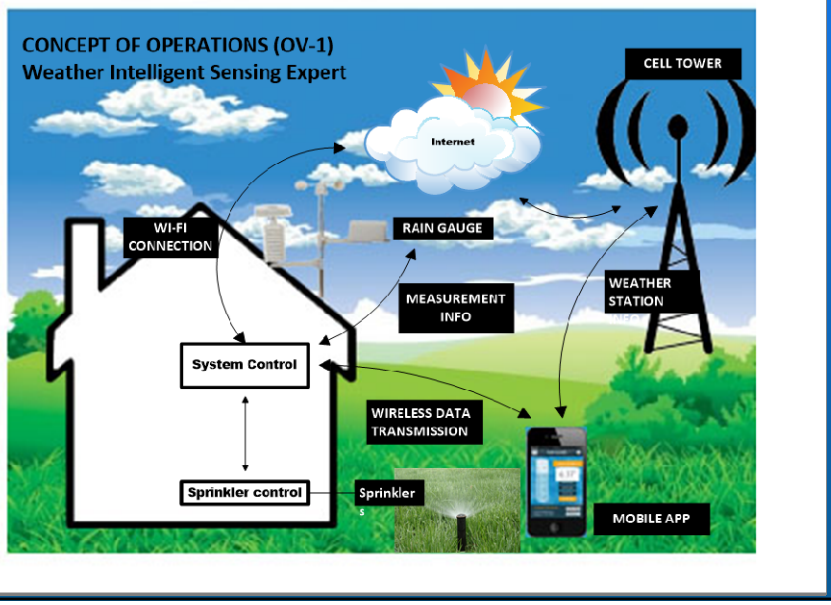
System Requirements

5

- The system shall measure rain fall
- The system shall receive measurements of weather statistics
- The system shall use a rain gauge to measure rain fall
- The system shall use the Internet to get weather statistics
- The system shall measure rain fall to an tolerance of 0.01" inch or better
- The system shall update data at least once per hour
- The system shall use LCD Display to show information
- The system shall display status of irrigation system.
- The system UI shall operate on mobile and tablet devices
- The system shall operate in Android platform version Jelly Bean or higher
- The system shall be developed in Android Studio and Visual Studio 15
- The system shall enable/disable a irrigation system based on a user data inputs
- The system shall enable/disable a irrigation system based on future rainfall prediction
- The system controller shall be cased in an enclosure

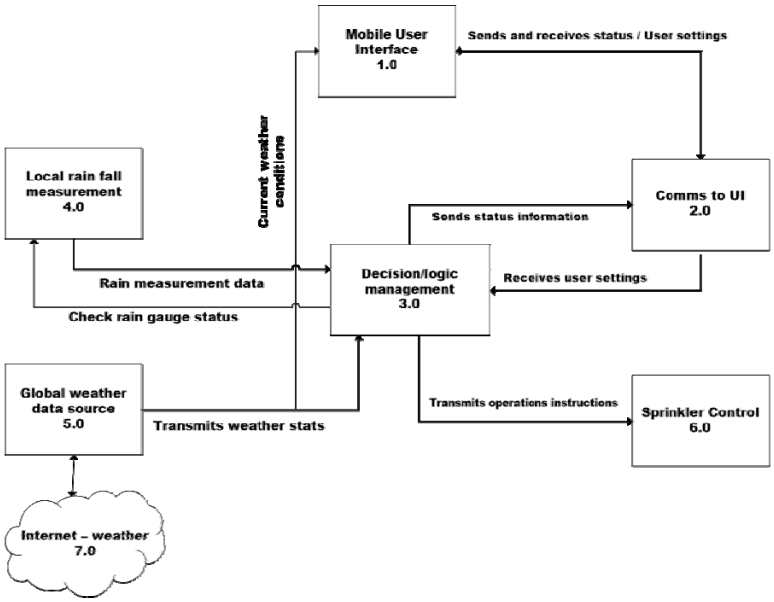
System Design

6

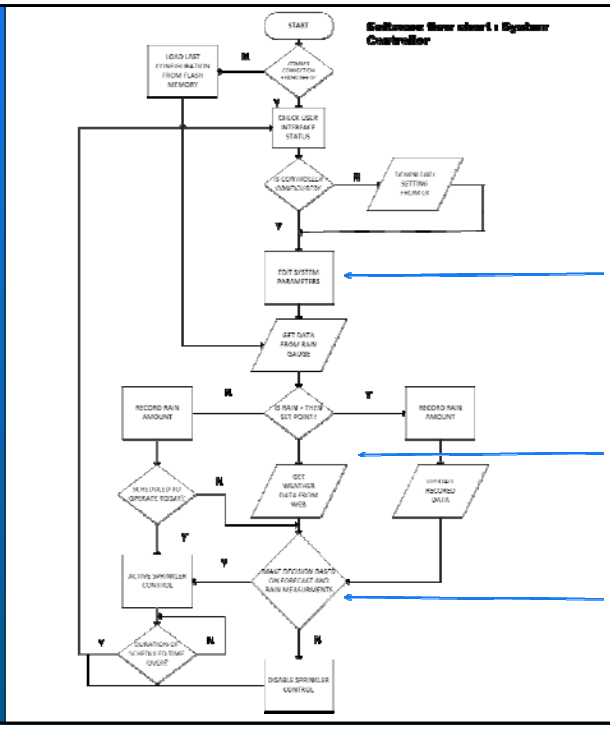


Functional Diagram : Weather Intelligent Sensing Expert

7



8



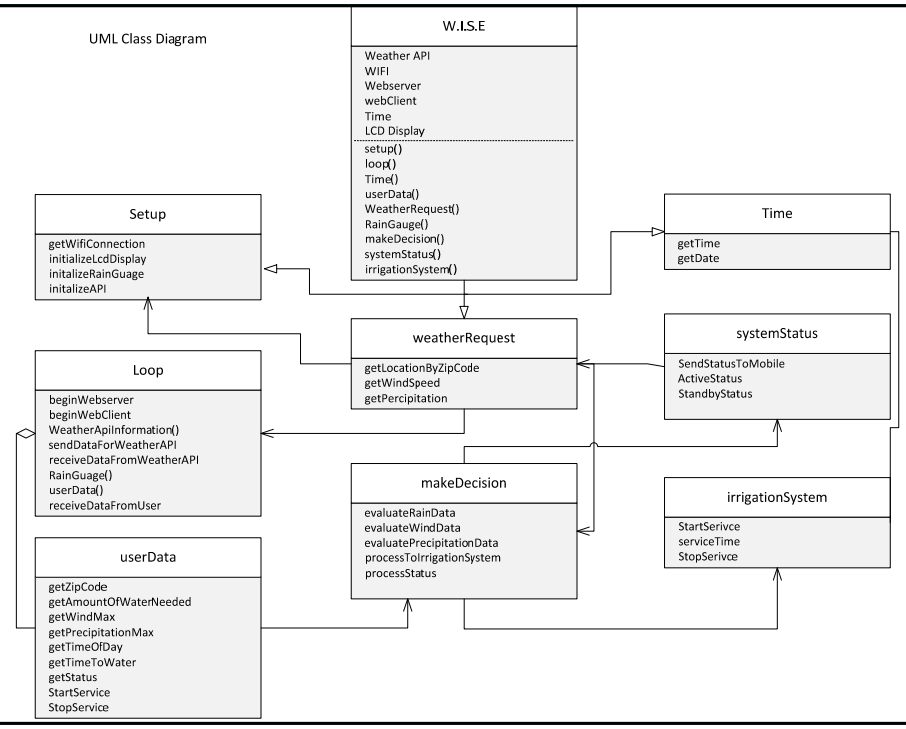
Get data from Mobile App

Get data from weather API

Make decision

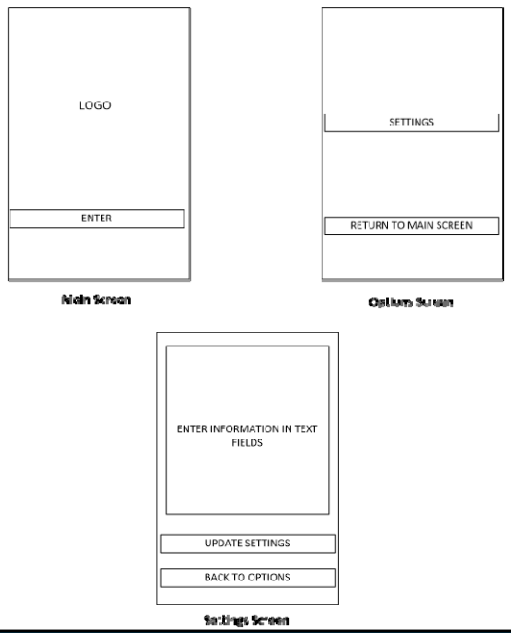
9

UML Class Diagram

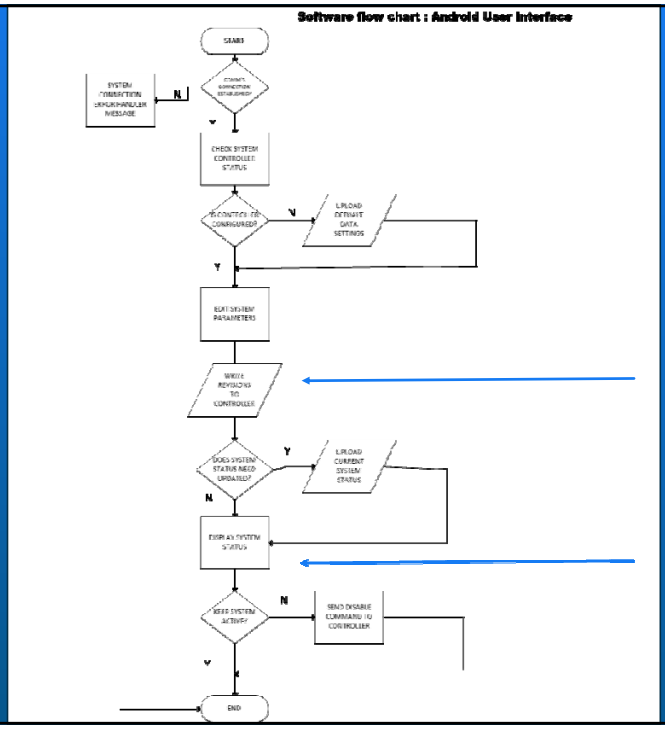


10

USER INTERFACE SCREENS



11

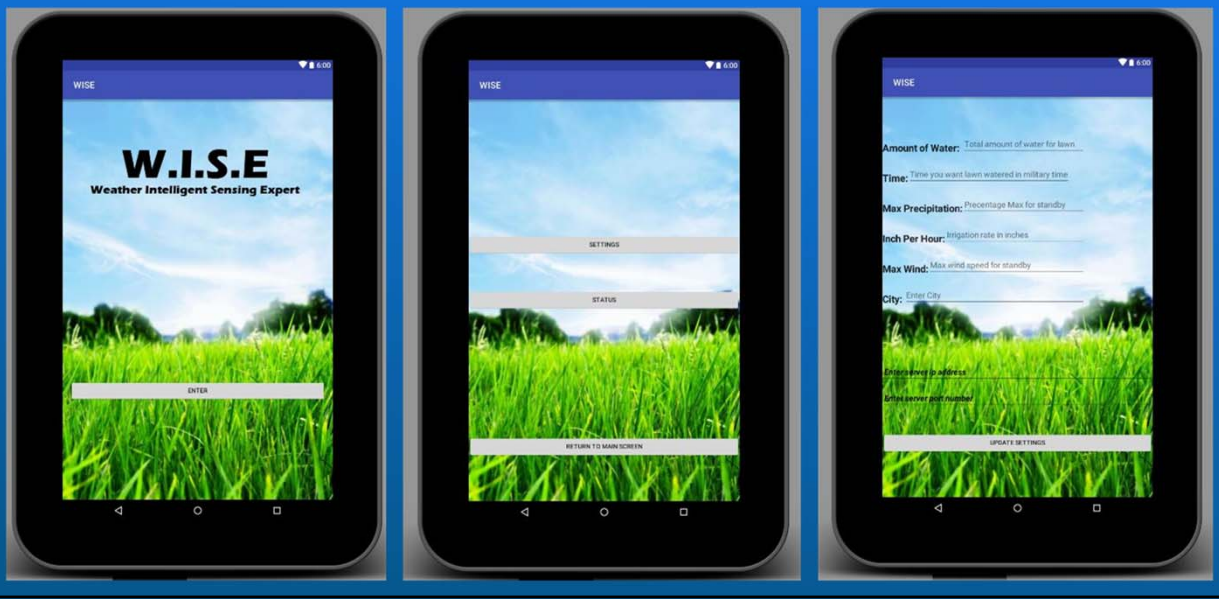


Write data to system hardware

Display System Status

12

GUI created in Android Studio showing the different activity screens



13

AccuWeather

- The AccuWeather API provides subscribers access to location based weather data via a simple web interface
- Access to the AccuWeather API requires an API key.
- More than 3.5 million searchable locations, including 1.7 million
- Data responses are returned in JSON
- Web client used to get data from web service

[{"Version":1,"Key":"328790","Type":"City","Rank":35,"LocalizedName":"Fort Wayne","EnglishName":"Fort Wayne","PrimaryPostalCode":"46825","Region":{"ID":"NAM","LocalizedName":"North America","EnglishName":"North America"},"Country":{"ID":"US","LocalizedName":"United States","EnglishName":"United States"},"AdministrativeArea":{"ID":"IN","LocalizedName":"Indiana","EnglishName":"Indiana","Level":1,"LocalizedType":"State","EnglishType":"State","CountryID":"US"},"TimeZone":{"Code":"EST","Name":"America/Indiana/Indianapolis","GmtOffset":-5.0,"IsDaylightSaving":false,"NextOffsetChange":"2017-03-12T07:00:00Z"},"GeoPosition":{"Latitude":41.131,"Longitude":-85.129,"Elevation":{"Metric":{"Value":243.0,"Unit":"m","UnitType":5},"Imperial":{"Value":797.0,"Unit":"ft","UnitType":0}}},"IsAlias":false,"SupplementalAdminAreas":[{"Level":2,"LocalizedName":"Allen","EnglishName":"Allen"}]},{"Version":1,"Key":"2225810","Type":"City","Rank":385,"LocalizedName":"Fort Wayne","EnglishName":"Fort Wayne","PrimaryPostalCode":"48209","Region":{"ID":"NAM","LocalizedName":"North America","EnglishName":"North America"},"Country":{"ID":"US","LocalizedName":"United States","EnglishName":"United States"},"AdministrativeArea":{"ID":"MI","LocalizedName":"Michigan","EnglishName":"Michigan","Level":1,"LocalizedType":"State","EnglishType":"State","CountryID":"US"},"TimeZone":{"Code":"EST","Name":"America/Detroit","GmtOffset":-5.0,"IsDaylightSaving":false,"NextOffsetChange":"2017-03-12T07:00:00Z"},"GeoPosition":{"Latitude":42.299,"Longitude":-83.096,"Elevation":{"Metric":{"Value":184.0,"Unit":"m","UnitType":5},"Imperial":{"Value":603.0,"Unit":"ft","UnitType":0}}},"IsAlias":false,"SupplementalAdminAreas":[{"Level":2,"LocalizedName":"Wayne","EnglishName":"Wayne"}]}]

Total Area Method (from Hunter Industries)

14

Calculation to determine rate of precipitation in Inches Per Hour for a irrigation system

$$P_r = \frac{96.25 \times \text{Total GPM}}{\text{Total Area}}$$

- ● Full-circle sprinkler at 8 GPM
- ◐ Half-circle sprinkler at 4 GPM
- ○ Quarter-circle sprinkler at 2 GPM

P_r is the precipitation rate in inches per hour.

96.25 is a constant that converts gallons per minute to inches per hour. It is derived from 60 min/hr divided by 7.48 gallons per cu ft times 12 inches per foot.

Total GPM is the cumulative flow from all sprinklers in the specified area, in gallons per minute.

Total Area is the area irrigated, in square feet.

Example:

Total GPM

● 2 Full Circle x 8 GPM	= 16 GPM
◐ 6 Half Circle x 4 GPM	= 24 GPM
○ 4 Quarter Circle x 2 GPM	= 8 GPM
Total	= 48 GPM

Total Area

40 ft. x 60 ft. = 2400 sq. ft.

$$P_r = \frac{96.25 \times \text{Total GPM}}{\text{Total Area}}$$

$$= \frac{96.25 \times 48}{2400}$$

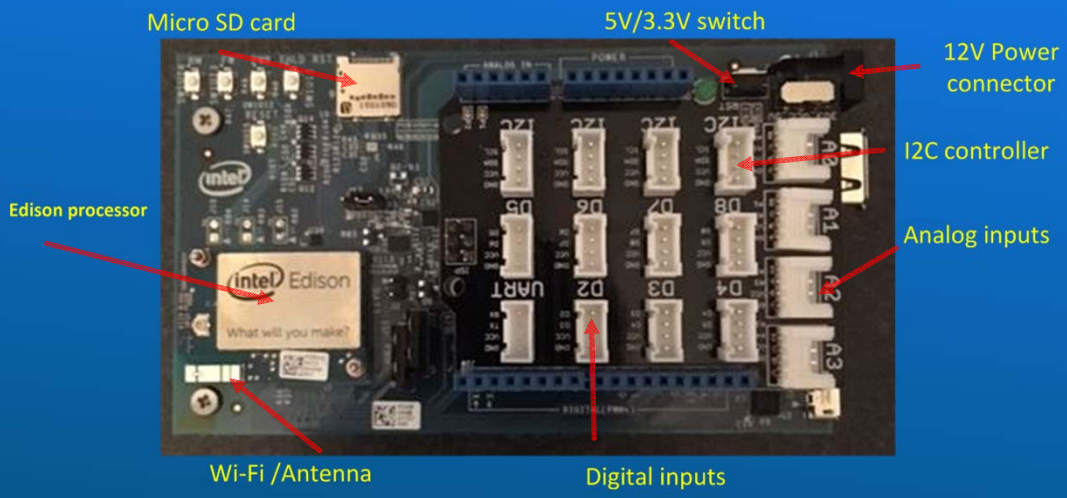
$$= 1.93 \text{ in./hr}$$

Source: [2]

System Hardware

15

Intel Edison w/ Arduino breakout Board

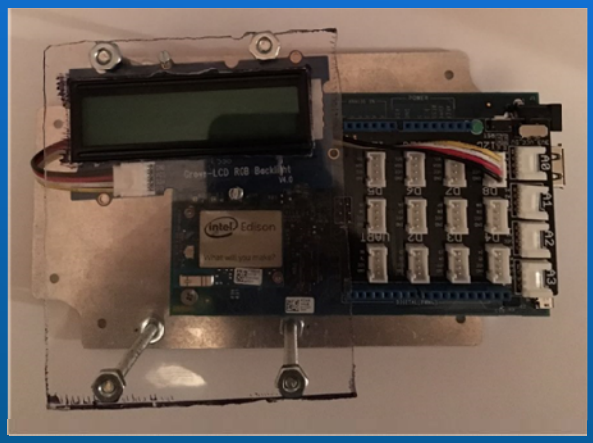


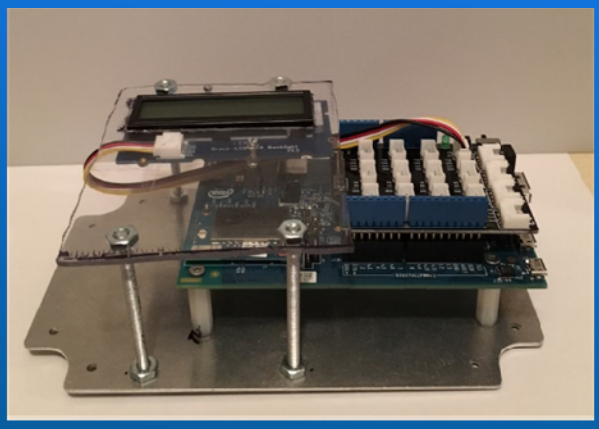
16

Rain Gauge tipping bucket

17

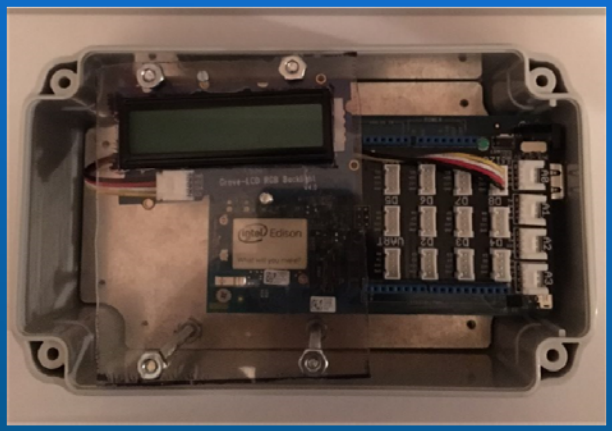
Hardware Construction





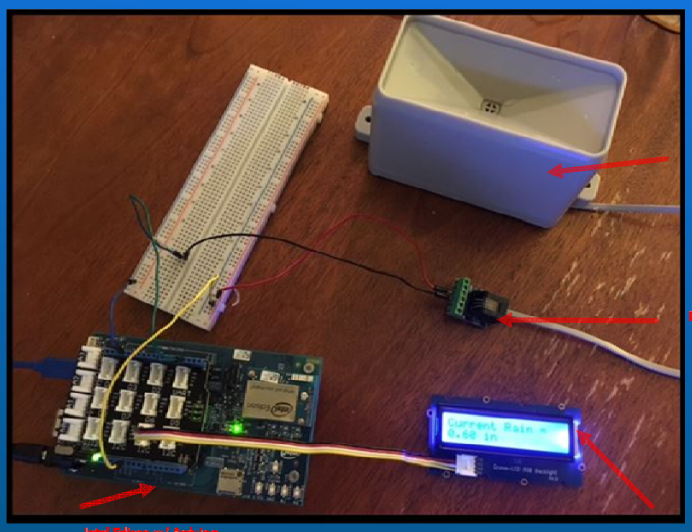
18

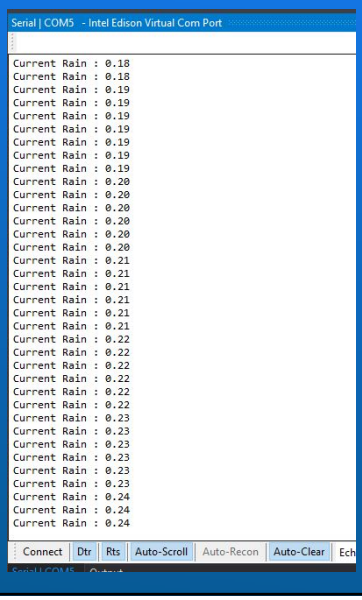
Inside enclosure



Testing

19





System Status Display's

20

21

Testing connection to home router over Wi-Fi.

- Establishing connection
- Verifying connection
- Show Device IP Address that the user will use in the mobile application

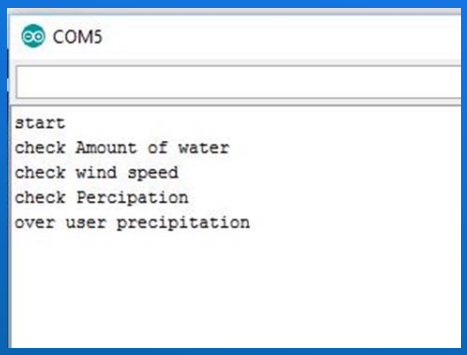
22

Decision test

```
//weather data
double precipData = 68;
double windData =24;

// user inputs
double userDesiredWater = 7.00;
String userTime = "";
double userPercipation = 60;
double userGph = 1.50;
String city = "";
double userWind = 25;

//total amount of water
double totalAmountWater = 6.00;
```



In this test it's shows that the precipitation from the weather data is higher than the user wants and makes decision not to enable the irrigation system .

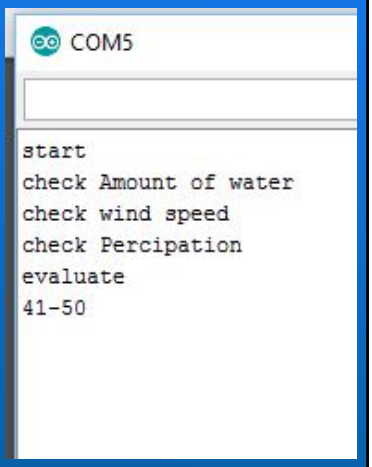
Decision test Cont'd ...

- Test shows that it's evaluating Precipitation being at 41-50 % chance.
- The run time is based off one hour increments. Breaking the hour into four run times.

```
//weather data
double precipData = 45;
double windData =25;

// user inputs
double userDesiredWater = 7.00;
String userTime = "";
double userPercipation = 70;
double userGph = 1.50;
String city = "";
double userWind = 226;

//total amount of water
double totalAmountWater = 6.00;
```



Run times : 0-25% = 45 mins , 26-50% = 30 mins, 51-75% = 15 mins
76-100% = 0 mins

Validation / Risk

Requirement Type	Requirement (Shall or Should statements)	Verification Method	Date Verified	Verification Report
Operational	The system shall measure rain fall	Analysis	30-Aug-16	Passed
Operational	The system shall receive measurements of weather statistics	Test	13-Sep-16	Passed
Functional	The system shall use a rain gauge to measure rain fall	Inspection	14-Jul-16	Passed
Functional	The system shall use the internet to get weather statistics	Test		Failed
Performance	The system shall measure rain fall to an tolerance of 0.01" inch or better	Test	5-Oct-16	Passed
Performance	The system shall update data at least once per hour	Test	25-Oct-16	Passed
Physical	The system shall use LCD Display to show information	Inspection	5-Oct-16	Passed
Physical	The system shall display status of irrigation system.	Inspection	20-Nov-16	Passed
Environmental	The system UI shall operate on mobile and tablet devices	Inspection		Failed
Environmental	The system shall operate in Android platform version Jelly Bean or higher	Inspection	22-Nov-16	Passed
Environmental	The system shall be developed in Android Studio and Visual Studio 15	Analysis	14-Jul-16	Passed
Operational	The system shall enable/disable a irrigation system based on a user data inputs	Test	25-Nov-16	Passed
Operational	The system shall enable/disable a irrigation system based on future rainfall prediction	Test	25-Nov-16	Passed
Operational	The system shall activate or deactivate sprinkler system based on manual shutdown by user	Test		Failed
Physical	The system shall weigh no more then 3 lbs	Inspection	25-Oct-16	Passed
Physical	The system controller shall be cased in an enclosure	Inspection	25-Oct-16	Passed

Risk

25

1. If programming the system hardware takes longer then expected.
2. If decision function is too difficult to design
5. If rain gauge doesn't work with the intel Edison board

			Risk Ranking				
			1 Insignificant: minor problem easily handled by day to day processes	2 Minor: some disruption possible	3 Moderate: significant time / resources required	4 Major: operations severely damaged	5 Catastrophic: project survival is at risk
5 Almost Certain: >90% chance	Likelihood	5					
4 High: 50 - 90% chance		4					
3 Moderate: 10 - 50% chance		3					
2 Unlikely: 3 - 10% chance		2	3		6	1,2,5	
1 Rare: <3% chance		1			4	7	
			1	2	3	4	5
			Severity				

Project Schedule break down

26

Phase II Start	0 days	Mon 8/22/16	Mon 8/22/16
Android UI Complete	0 days	Tue 11/22/16	Tue 11/22/16
System Controller Complete	0 days	Mon 11/21/16	Mon 11/21/16
Functional Prototype Complete	0 days	Thu 12/1/16	Thu 12/1/16
Verification Testing Complete	0 days	Tue 11/8/16	Tue 11/8/16
Final Report Submitted	0 days	Thu 12/8/16	Thu 12/8/16
Presetantion Delivered	0 days	Fri 12/9/16	Fri 12/9/16
End of Semester	0 days	Fri 12/9/16	Fri 12/9/16
Prototype Development	**74 days**	**Mon 8/22/16**	**Thu 12/1/16**
Android User Interface	**26 days**	**Wed 10/26/16**	**Wed 11/30/16**
Program coding and design	16 days	Tue 10/18/16	Tue 11/8/16
Design error testing	6 days	Wed 11/9/16	Wed 11/16/16
Program debugging	3 days	Thu 11/17/16	Mon 11/21/16
OpenweatherMap API coding	1 day	Tue 11/22/16	Tue 11/22/16
System Controller Development	**66 days**	**Mon 8/22/16**	**Mon 11/21/16**
Parts Procurement	2 days	Mon 8/22/16	Tue 8/23/16
Device coding and design	4 days	Thu 11/3/16	Tue 11/8/16
Debugging Device	5 days	Wed 11/9/16	Tue 11/15/16
Testing device	3 days	Wed 11/16/16	Fri 11/18/16
OpenweatherMap API coding	1 day	Mon 11/21/16	Mon 11/21/16
Hardware Development	**8 days**	**Tue 11/22/16**	**Thu 12/1/16**
Rain Guage Configuration	2 days	Tue 11/22/16	Wed 11/23/16
Arduino board Configuration	2 days	Thu 11/24/16	Fri 11/25/16
Bluetooth Configuration	2 days	Mon 11/28/16	Tue 11/29/16
Wi-Fi Configuration	2 days	Wed 11/30/16	Thu 12/1/16
Interagation and Testing	**20 days**	**Wed 10/26/16**	**Tue 11/22/16**
Assemble system	2 days	Wed 10/26/16	Thu 10/27/16
Connection testing of UI to Controller	2 days	Fri 10/28/16	Mon 10/31/16
Verify Requirements	2 days	Tue 11/1/16	Wed 11/2/16
Test Rain Guage connection	1 day	Thu 11/3/16	Thu 11/3/16
Test Complete system Operation	3 days	Fri 11/4/16	Tue 11/8/16
Final Report Development	**29 days**	**Tue 11/1/16**	**Fri 12/9/16**
Presentation Development	**19 days**	**Tue 11/15/16**	**Fri 12/9/16**

Lessons Learned

27

- Time Management, follow the WBS and Gantt chart made for the project.
- Get a better understanding of Networking, do more research.
- Realize that you need to ask for help sometimes. Don't get in your own way.
- I now have a better understanding of what I'm capable of and opportunities I need to work on to be a better computer engineer.

Work Cited

28

[1] "Water Sense," United States Environmental Protection Agency, [Online]. Available: https://www3.epa.gov/watersense/pubs/outdoor.html. [Accessed September 2016].

[2] H. Industries, "Precipitation Rates and Sprinkler Irrigation," [Online]. Available: http://infohouse.p2ric.org/ref/53/52027.pdf. [Accessed November 2016].

29

Q/A?

30

DEMO