

Release Plan

Product Name: Smart-Irrigation Frontend

Team Name: Wubadubdubs

Release Name: Smart Irrigation V2.0

Release Date: November 25th, 2015

Revision Number: 2

Revision Date: October, 24th, 2015

High Level Goals:

Our frontend team goals for release include the following:

- Be able to display live data from the garden with Google Charts
- Be able to control the garden watering controllers with the application
- Be able to access a live watering window with predictions of next watering cycle
- Be able to set up system in different gardens
- Web Application will have accounts for users and also for administrators (Both should be different.
- Web Application will have pop-up buttons to show different display charts (Moisture, Temperature, etc.)
- Web Application will be secured using the rails secure tools.

User stories for release:

Sprint 1:

1. As a developer, I want a stable webserver to host dynamic content on.
 - a. Setup CentOS server and configure for remote access and testing.
 - b. Install Java and Grails for basic webserver hosting.
 - c. Install Apache Tomcat to contain the server securely.
2. As a developer I want remote access to the server for developing in a real server environment.
 - a. Create a dynamic hostname smart-irrigation.no-ip.info to access the server from.
 - b. Setup SSH for remote development.
 - c. Configure router NAT to allow server connections on ports 22 and 8080

Sprint 2:

1. As a developer, I want a finalized layout of the website so that data pushed from the back end can be displayed in intuitive ways.
 - a. Semi- detailed layout of website's pages on paper
 - b. Skeleton of paper layout implemented in HTML/GSP
 - c. Fill in skeleton with back end data or dummy data
2. As a developer, I want to be able to use the API created from the backend team to pull data from database and use represent the data with a google chart.
 - a. Learn the basics of Google Chart tools
 - b. Create some html templates using the Google Chart tools
 - c. Be able to pull live data from MySQL
 - d. Generate a summary window for history of the garden

Sprint 3:

1. As a developer, I want to be able to create a secure website were the user can access their garden.
 - a. Learn about configuring the Spring Security Core on the grails-app
 - b. Develop a grails security system that secures the website using a Spring Security Core
2. As a developer, I want to be able to add a simple animation that will help in the design of the website. Also, I want to be able to style the website using Cascading Style Sheets (CSS).
 - a. Create a simple animation using Processing
 - b. Learn about adding Processing code to the grails-app
 - c. Style the website using Cascading Style Sheets
3. As an administrator, I want to be able to be able to have access to the secure website and make necessary changes to modify the data represented on the website.
 - a. Configure the Grails application with battle-hardened and proven Spring Security Core to have an administrator account
 - b. Learn about adding an administrator account (It should be different from a user account)
4. As a user, I want to be able to see a graphic description that contains information about the temperature and moisture sensor readings. As a user, I also want to see a summary generated by the grails-app that should tell me when to water my garden again.
 - a. Learn about the post_watering API method to add to the website and learn how to modify it.
 - b. Summary of next watering window will tell the user when the water the garden again using

Product Backlog:

- Automated Weather Adjustment given cross-referencing rainfall data window
It is really difficult to create an algorithm that can predict the rainfall using weather channels. We would require more time to do more independent research and learn about creating automated weather adjustments since it could become a
- Controllers for automated watering
Due to a lack of communication with IDEASS, we were unable to work with the sensors and controllers until really late in the quarter to create a controller for automated watering. For now, the system is manual and water data has to be inputted manually into the database. Sensor information is now being stored in a database which is accesses by the web application.

Project Presentation:

Final project submission and acceptance tests:
Final project class presentation:

Tuesday, December 1st, 2015
Wednesday, December 2nd, 2015