

Sprint 3 Plan

Product Name: Smart-Irrigation Frontend

Team Name: Smart-Irrigation Frontend

Sprint Completion Date: November 25th, 2015

Actions to Stop Doing

Stop spending too much time working on only one task:

If there is a specific task, like learning about the rails security plugin, don't spend too much time trying to figure out how to make it work. Communicate with team members and try to all figure out a solution to the tasks that are more complex (tasks with higher number of points). Tasks that have a lower number of points tend to be easier tasks like doing independent research for data displays.

Actions to Start Doing

Develop a better system of communicating with the people in charge of the smart irrigation project since we did not have any live data.

As we approach the end of the sprint, we do not have live data from the sensors so we had to start working with the backend team to generate dummy data to be presented in our application. Communicating with the backend team is essential to this project since they provide all the data analysis needed to be presented to the users as visible data. Having a strong communication source makes complex tasks more flexible.

Actions to keep doing:

Always have a good communication source to share ideas with everybody in the group:

Continue to keep on working with the backend team to keep making Unit Tests to see if all of the files created work. The reason for the unit tests is to see and check the functionality of the code before we start displaying live data from the sensor data stored in the databases. Not all the APIs implemented by the backend team were shown on the rails application since it was a very difficult process to get familiar with the rails framework.

Work completed/not completed:

1. 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. Created some html templates using the Google Chart tools
 - b. We are now able to pull live data from MySQL
 - c. Generate a summary window for history of the garden
 - d. Develop a controller window for user to be able to control their garden
 - e. Provide live feed of garden
2. As a developer, I want to be able to create a secure website where the user can access their garden.
 - a. Learned about configuring the Spring Security Core on the rails-app

- b. Developed a grails security system that secures the website using a Spring Security Core
- 3. 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. Created a simple animation using Processing
 - b. Learned about adding Processing code to the grails-app
 - c. Styled the website using Cascading Style Sheets
- 4. As an administrator, I want to be able to have access to the secure website and make necessary changes to modify the data represented on the website.
 - a. Configured the Grails application with battle-hardened and proven Spring Security Core to have an administrator account
 - b. Learned about adding an administrator account (It should be different from a user account)
 - c. Failed to create an administrator account, for now user and administrator look the same.
- 5. 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. Learned 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

Work completion rate:

Sprint Goals: 80%
Total Project 80%