Project Plan: Agri Drain

Team 1602

1. Introduction

a. Project Statement

Our project's aim is to create a web application as a user interface to allow farmers to remotely control the amount of water that is being used to irrigate crops and to observe flow rates, nutrient levels, water levels, and rainfall totals from either a computer, tablet, or smartphone. The project will also save user preferences in terms of draining scheduling; for example, how much water is to be drained at what times of the day during different weekdays.

b. Project Purpose

The purpose of this project is to allow remote communication between the farmers and their irrigation systems that are mechanically implemented. It will allow more freedom and ease for the farmers to control and monitor water levels in their crops by using a user-intuitive graphical user interface. The web application that will stem from the success of this project will be have a low learning curve in order to make it convenient and easy to use. The purpose is also to bridge communication between data sent from the user interface to a server and how that data can be interpreted by the server, and vice-versa.

c. Goals

We aim to create several mockups of potential layouts of the web application and present them to the client for feedback. We also aim to create a proof of concept prior to the end of this term. Overall, we would like to achieve all statements previously outlined in parts 1A and 1B. In order to achieve these goals we will have a biweekly meeting with our advisor to ensure we are staying on track as well as frequent feedback from the client to eliminate misunderstanding or to account for change of desired outcomes.

2. Project Deliverables

The primary deliverable that will follow this course's design project include a user-intuitive web application that is accessible via computer and smartphone, where the user can effectively interact with the application's controls for effectively sending out useful data for controlling irrigation levels in the user's crops. The project will also allow the user to monitor the levels of flow rates, nutrient levels, water levels, and rainfall totals in order to effectively conserve soil and

water usage. The website will present such data in a visually pleasing graphical format, and can be exported to an excel spreadsheet to record these levels.

3. Design

a. Previous Work

Most of the team has previous experience working with web applications and mobile/host communication

Griffen Clark, our webmaster, has worked on a web application for his 309 project which dealt with JavaScript logic handling, CSS libraries to improve user experience in terms of a graphic user interface, and PHP programming to interpret data sent from clients to a server

Rodney Barto, our Communications Liaison, has a measurable amount of experience in creating web applications. He has assisted in the development of a personal home media server that uses a web application for viewing content stored on an attached network device. More importantly, he has a current internship where he is responsible for creating a web application that allows a user to configure and control a networked LED lighting system. Rodney also has experience in various other languages.

Adam Wolter, one of our Key Concept Holders, has created and assisted in creating several web interfaces and applications. At his current internship, he works with graphically displaying and analyzing data, as well as designing the user interface. In his free time he has created and designed a home media server web application for streaming media on a local network.

b. Proposed System Block Diagram

<This will be filled in later, after meeting again with our client.>

c. Validation

We plan on designing acceptance criteria for all modular aspects to this project. Ideally, we would like to set up a continuous integration server, such as Jenkins, to make deliverable builds and schedule daily automated tests to ensure our product meets that acceptance criteria. These scheduled builds will be based on a strict repository workflow that we as a team will abide by. This will be done by using a git repository, with an integration branch that will merge all of our individual code changes. Once this integration branch meets our testing acceptance criteria, it will be merged into a master branch in which a deliverable build can be made. The automated tests will be partitioned based on the modules of the total project. These tests will be based on user-story driven test case scenarios that will be developed earlier on in the process. These can be categorized by issuing Black-Box testing (having testing criteria with specified user inputs and checking that the outputs are correct), and White-Box testing (testing on an design level to make sure these modules and interfaces communicate correctly and efficiently with each other).

We will appoint a quality assurance team member to continuously ensure that our code implementations align with our client's initial requirements and specifications.

4. Project Requirements/Specification

a. Functional

This project needs to allow users to remotely change the settings (e.g. water level setpoint) on an Agri Drain device. The users need to be able to see sensor data (e.g. data to calculate flow rate) from the device at the current moment and historically with timestamps.

b. Non-functional

This project will be a web application, but it needs to be easily usable on cell phones. The historical data should be in a format that is easy to back up. The code should be documented well enough that this can be modified in the future by another team. We don't know yet what kind of security the application needs.

5. Challenges

One of the biggest challenges/risks that we will be facing is the fact that none of us have any real world experience with irrigation or water drainage systems. It will be hard for us to put ourselves in an end-user's shoes because we don't really know what kind of features would be good for a system like this, or what we could add to help make the entire user experience less complicated. To reduce/mitigate this risk, we will have to work with Agri Drain to create or go over sample user interfaces that highlight essential features.

Another risk is that receiving timely information. Our client is not someone we frequently see in person, and we need to communicate with them regularly to create this application. We will mitigate this risk by setting up weekly meetings with our client and by preparing the topics we may need to discuss ahead of time.

Most potential risks cannot be predicted at this point, as we haven't delved deeply into our project.

6. Timeline

a. First Semester

First semester will consist of moderate communication between the team/advisor and the client. Much of the first portion of the first semester will be continually refining requirements and specifications for verification. This will entail design mockups with feedback from client,

architecture designs, a backlog of items to implement in the project, process tools for agile work process such as Jira, Crucible for code reviewing, and Confluence as a wiki for design and requirement documents so we will have everything prepared to start implementing the project. We also hope to be able to have a proof-of-concept that we can demonstrate to our client by the end of the first semester.

b. Second Semester

Second semester will mostly consist of the bulk of implementation work. We decided to use an agile work process. This will include bi-weekly sprints with a list of backlog items that will be a collection of minimum viable deliverables, bi-weekly sprint planning to determine how much work will be put into the sprint and who will be assigned what issues, bi-weekly sprint review and demos with client and advisor, bi-weekly backlog grooming to point estimate items in the backlog, quick 15-30 daily meetings of sprint updates among the team.

7. Conclusions

<To be filled in with future versions.>

8. References

<To be filled in with future versions.>

9. Appendices

<To be filled in with future versions.>