DEC1609
Colin Hurd
Swarmy Ponpandi
Alex Clark - Team Leader
Alex Peck - Technical Lead
Chandler Heisler - Communications
Kurry Watson - Webmaster
DEC1609@iastate.edu

http://dec1609.sd.ece.iastate.edu

Revised:11-15-16/2.0

SAVI Software Platform

DESIGN DOCUMENT

Contents

1 Introduction

1.1 Project statement

To create an intuitive and effective UI to allow the user to control farming equipment around their farm. This includes monitoring the equipment, taking in and serving up data in real time, and being able to have complete control over their equipment at all times.

1.2 purpose

As the need for more and more food is constantly on the rise. Farmers need to be able to produce more food, which means to either increase yields or increase the amount of land being farmed. More farm land means more time needing to be spent in the field. But with the new robot and software we are developing farmers will be able to remotely control their tractors. This will open up the possibility for farmers to have multiple tractors in multiple feeds reducing the time the farmer needs to spend in each field.

1.3 Goals

Establish a server and database to collect and store data.

- store and serve data from fields
- store and serve data on tractors
- store and serve data from UI to Tractors

Create a web based UI to interface with the farm equipment

- Show real time updates on equipment progress in fields
- Allow user to change and update information on fields and machines
- Allow user to command equipment to move/stop/sync/nudge

Continuous development on UI as client adds requirements/changes

2 Deliverables

To meet the goals outlined in the introduction we will need the following deliverables:

- A functioning UI
  - monitor the status of their tractor
  - send the tractor to staging and unloading
  - sync tractor and nudge it when synced
  - Draw, save, and load fields
  - Adjust speed of tractor
  - Must be easy to use, intuitive, and look appealing
- Create access to cloud database that allows UI to store and load data
- Add functionality to UI, server, or database as needed at clients request
  - This project has been a very iterative so concrete deliverables were often growing and changing throughout the year.
  - This was our most important deliverable

3 Design

3.1 System specifications

Research was conducted about all forms of software architect. Advantages and disadvantages from various software designs were weighed. Some research included research about client server networking (Client Server Networking), and research about Multitier architecture (Multitier Architecture). Countless other uncertainties and confusions were also cleared up through research.

3.1.1 Non- functional

1. Simple, easy to use GUI
2. Everything should be secure and encrypted
3. Cross Platform compatible (Windows, OS, Android)
4. Data portability for easy integration with precision ag data
5. UI should be easy to add new features to
6. UI should be easy to maintain when we are gone

3.1.2 Functional

1. Create, load and save existing field maps
2. Should be able to control tractor without internet connection

3. UI needs to provide clear way to control tractor (speed, movement, etc.)
4. Possibility to be hosted on cloud
5. Store and present all tractor information in real time as needed
6. Allow multiple views for different farms/sets of equipment

3.2 Proposed design/method

Our team along with our client has decided to most of our work in JavaScript(JS). We plan on making our application in JS so that it is easily portable and can be used on many different operating systems. We will also have a database in Amazon Web Service S3. This will interface well with the Node server that we have.

3.3 Design analysis

After working with the half the semester we have found the JS design to be working very well. The web application that we have built is working well with the AWS S3 database. We have also started to put some of our code on the cloud. This has been working well so far and we have not run into any major problems.

4 Testing/Development

4.1 Interface specifications

The hardware we are using is a robot that our client is providing. We will be using a REST api to communicate with the robot. For our database we will be using AWS S3 and for our application code we will be using React which is a JS library.

4.2 Hardware/Software

We had our code makes calls to the python API. Because we cannot connect to the tractor when we are coding, we made a mock python server that acted just like a tractor should. Our mock server just returns static (or randomly generated) data so we can test the UI is calling the python API properly. Testing heading and how the tractor actually would move around we had to modify the python server so it would run through different scenarios and then ensure our code would perform accordingly.

4.3 Process

Our current method of testing is in two parts. The first part is the developer self tests his code by running it and do manual smoke testing to make sure everything is functioning. They will then write regression tests for their code that can be run and make sure it is working later in the project.

The second method of testing is peer review. We plan on having other members run smoke tests on the software. Bugs fall through the cracks during the development cycle, and the more eyes you get on code, the less bugs that will develop.

5 Results

So far all testing has been done with opening our JavaScript on a web browsers and we have had mostly positive results. We have developed a logger that helps us to debug any problems. We also made a simple test python server that acts like the tractor so we can test more of the UI without having a tractor present. We are currently working on building a more in depth test that will cover more functionality.

6 Conclusions

The goal of Smart- Ag is to create a system of Autonomously controlled farming equipment to help with the day to day tasks around the farm. We had the task of creating the user interface and cloud storage. The UI has complete control over the tractor and is pulling tractor diagnostic data to display to the user. Our motivation is to give a farmer more time to do tasks a human needs to do, by helping him do tasks a robot can do. We believe our solution is an efficient and intuitive way to help farmers.

7 References

1. Project proposal: https://drive.google.com/open?id=0B5SAoPDouSRRTWo1U3BRUUtuQ00
2. Client specifications document: https://docs.google.com/document/d/1bHd2PPzdyn1devf1dfDf61sgbQtK4nlkZIRZU2UioSc/edit#
3. Client minimum viable product + precision ag document: https://docs.google.com/document/d/1xUuwwhmpM4v9Q-5Mc_mBep0YnUH2rYn2KQPdsRt9FG4/edit
4. Client Server Network: Advantages and Disadvantages: http://www.ianswer4u.com/2011/05/client-server-network-advantages-and.html#axzz40OsiBTgL
5. Multitier Architecture: https://en.wikipedia.org/wiki/Multitier_architecture

8 Appendices

N/A currently