Smart Backpack Sprayer for Small-Scale Agriculture Applications

Design Document

Team Number:
sdmay20-53

Clients:
Taylor Greiner & Tim Andersen

Advisor:
Dr. Daji Qiao

Team Members/Roles:
Kevin Davis - Hardware Research Engineer
Sean Doran - Logistics Engineer
David Hayes - Communication Engineer
Madison Kriege - Communication Engineer
Donald Laracuente - API Research Engineer
Shuangquan Li - iOS Engineer

Team Email:
sdmay20-53@iastate.edu

Team Website:
http://sdmay20-53.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

We do not currently have any circuit design requirements, as we plan on using Arduino. We will follow the ISO/IEC 12207 systems and software development standards. Lastly, we will follow the IEEE 1016 digital design standards. We will follow these standards for each component of our project.

Summary of Requirements

System

- Collect flow rate data
- Collect GPS data
- Record substance in use
- Collect compass data
- Be waterproof
- Be operable in temperatures between 0-100C
- Be under 30 pounds
- Be wearable on one's back

Hardware

- Use a flow sensor with accuracy to 10% of the duty cycle
- Use a GPS sensor with accuracy to 3 meters
- Have a battery life at least 3 hours.
- Be mountable inside backpack sprayer
- Package data in JSON format
- Be able to send data using Bluetooth

Data

- Collected in 1 second intervals
- Collection shall be time-stamped with 24 hour time format

Application

- Be functional on iOS
- Collect data using bluetooth
- Be able to parse data in JSON format
- Map GPS data using Mapbox
- Report substance being sprayed for each spraying event
- Include flow rate in report of spraying, for each received packet
- Include functionality to save spraying events

Applicable Courses from Iowa State University Curriculum

AGRON 160

This class covers water resources and how they can be affected. Our project would help reduce chemical runoff, and material learned in this class can be useful when dealing with the environmental aspects of the project.

COMS 309

Working on a semester long small group project helped us develop good team practices and better understand how to work together effectively. Additionally, the material covered agile practices we will be utilizing throughout this project.

CPRE 288

Our project will be combining a variety of sensors and a microcontroller, similar to the process that we went through in the lab portion of this course. We can build off of the material learned to more productively use the sensor data.

EE 201/230

The basic circuitry learned in these classes will be useful in our project for connecting the Arduino to power and other components.

New Skills/Knowledge acquired that was not taught in courses

Swift Programming Language

The mobile application for the project is utilizing Swift and XCode for iOS. Individual members of the team have had outside exposure to these languages; however, none of us have taken a formal course in it.

Firebase Development

Firebase and Firestore are being utilized for the serverless server portion of the application. Similarly, several team members have had limited experience with the interface before this project.

Arduino Hardware and Device Communication

The hardware components of the project are built on an Arduino board with a variety of sensors. Learning to navigate the device and the communication between the board and other sensors has required several outside resources.

Mapbox API

For the graphing and map aspects of the application, our team is utilizing the Mapbox API recommended by our clients. The documentation for the API has been used to help get more familiar with the skills needed for integration.

Table of Contents

1.0 Introduction

1.1 Acknowledgement

We would like to thank our clients, Taylor Greiner and Tim Andersen, for proposing the project and helping throughout the entirety of the project. They provided us with hardware, financial support to purchase necessary components, and with technical guidance and support as needed for the project. We would also like to thank our mentor, Dr. Qiao, for providing support with our documentation and for providing technical support. Dr. Qiao also ensured our group was focused on the project and that the team was working efficiently and in an organized way.

1.2 Problem and Project Statement

This project is modifying a backpack sprayer used for weed killers, bug spray, or similar chemicals on small farms and acreages. Commercial backpack sprayers currently do not offer a "smart" option, making it difficult for users to keep track of product placement information. We will develop an iOS application, paired with an Arduino device inside the backpack, to meet project requirements and provide users with spray information. This application will store location, flow rate, time of coverage, and other relevant data. The purpose of our project is to simplify the lives of our client base and reduce waste caused by excess chemical coverage.

1.3 Operational Environment

The operational environment for this project is an agricultural environment. This means that the solution will be exposed to rain, wind, dirt, dust, chemicals, and other elements. The solution must be robust such that it will function in a wide variety of conditions.

1.3.1 Requirements

1.3.1.1 Functional Requirements

- Read sensor data
    - Includes GPS location, rate of application
- Backpack sprayer must communicate with mobile app
- App should analyze data
- Create map of spray amount and time of application

1.3.1.2 Economical Requirements

Initial Budget of $750 from the clients. The clients will provide us with what is needed for the completion of the project. This includes hardware such as the compass, gps module, Arduino Mega 2560, flow sensor, and other components as needed. The clients will also supply accounts for software/backend as needed.

1.3.1.3 Non-Functional Requirements

- Portable
  - Easily worn and carried on one's back
- Scalable
- Reliable
- Intuitive and easy to learn/understand/use by a wide variety of users.

1.3.1.4 Environmental Requirements

- Waterproof: IP67
- Signal Range: 10m
- Temperature: -25 ~ 100 °C
- Altitude: 0 ~ 2500m
- Wind Speed: below 25 km/h

1.4 Intended Users and Uses

The intended user is one that works in a small scale agricultural environment. The user has a need to spray the land somewhat frequently, where learning about each spray application would be beneficial in order to maximize efficiency of spraying and to learn from previous applications. The use of this product is for a small scale operation, so that the battery life and amount of spray is sufficient to cover the desired area. The user will typically wear the product while using it.

1.5 Assumptions and Limitations

1.5.1 Assumptions

- Limited Number of Users
- The end product will be used in spring and summer
- The end product will not be used in the rain
- The end product will be self contained inside of the backpack
- Then end product will measure in imperial and metric units

1.5.2 Limitations

- GPS Measurement Accuracy within a meter
- Flow Rate Measurements based off of what the sensors allow
- Battery-life must meet Realistic usage times
- Expenses are within the $750 initial budget

1.6 Expected End Product and Deliverables

The deliverables are a backpack sprayer solution and a mobile application. These two need to be able to communicate together and create the “smart” backpack sprayer.

The backpack sprayer shall be a portable/wearable solution that will allow the user to spray substances on the small-scale agricultural operation. This will also include sensors that are able to collect GPS coordinates, directional data, and flow data from the backpack sprayer. An Arduino housed within the backpack will be a component of this solution. The arduino will use Bluetooth to send data to the user’s mobile device.

The mobile application will be developed for iOS. It shall be able to receive data from the smart backpack using Bluetooth and then parse this data and represent it in a clear and effective way to the user. This will include a map of the locations sprayed as well as the flow rate at each location, the time of the spraying, direction of where the spray was directed, and what was being sprayed from the backpack. The application will be able to record historical data from previous sprays.

2.0 Specifications and Analysis

2.1 Proposed Design

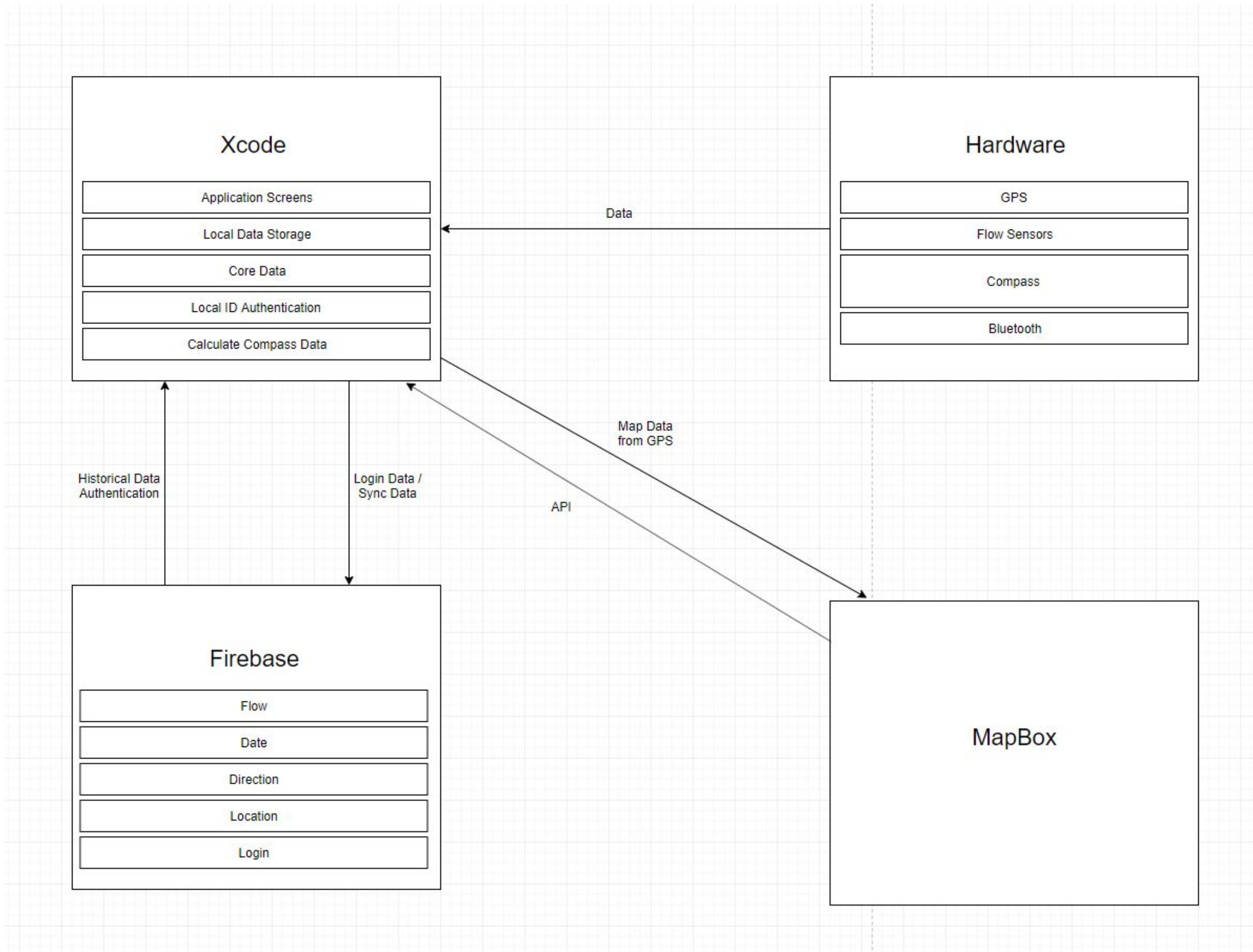
So far, the proposed design can be summarized by the following diagrams/figures. We will use an iOS application, and the screens that will be in this app can be shown in the diagram as well.


Login
Login
Sign Up
Forgot Password
Login Successful
Sign Up
Button
Pressed
Submitted
Sign Up
First Name:
Last Name:
Email Address:
Password:
Confirm Password:
Switching
Between
Screens
Date
Location
Flow
ETC
Name
Bio

2.1.1 Proposed Application Views/Layout

In order to gather data about the spraying application, several sensors will need to be used during the time of the spray. An Arduino Mega 2560 has been purchased and will be used to control all of the sensors, gather data, and send data to the mobile application. The sensors that have been purchased are a GPS module(Neo 6m), a Bluetooth module, a flow meter, and a compass. The GPS module will be used to gather the coordinates of the sprayer, which will then be used to map which area has been sprayed. THe compass will show direction or spray and give a more accurate description of what was sprayed. The flow sensor will be attached to the hose of the sprayer and record how much spray has been used and where the spray was used. The Bluetooth module will be used to send the aggregated data to the mobile application. This will cover the functional requirements of the hardware. Accuracy testing is necessary to ensure that all of the requirements are being fulfilled. If it is discovered that a requirement is not being satisfied, different hardware components will be evaluated as part of finding a solution. The need for a PCB that will allow for easy connection of the hardware to the Arduino Mega 2560 is currently being evaluated, and if necessary will be further developed. This would make it simple to hold the necessary hardware on the backpack sprayer when in use.

2.1.2 Proposed Hardware/Backpack Design



2.1.3 Proposed Architecture Design

2.2 Design Analysis

Researching hardware components and purchasing those that best fit the needs of the project was the initial step in analyzing the project's design. Finding and verifying that hardware exists for the needs of the project proves that the overall design is feasible. After purchasing appropriate hardware, completing testing of each component to ensure the accuracy and quality of the component will satisfy both the functional and non-functional requirements is necessary. THe GPS module has been tested outdoors and shown to have accuracy within 2-3 meters. This will be adequate for the project's needs. The flow meter has also been tested and shown to be accurate when running water through it. Running about 3L of water resulted in a reading of 3.03L, which is accurate enough for the requirements captured. Some strengths we have learned in the design analysis is how precise our equipment is. It has been higher than expectations and should help provide accurate data to the user. A few weaknesses we have is not getting different types of hardware to test the difference. We did get a couple of GPS sensors, but only one type of each of the other sensors.

2.3 Development Process

As the project becomes more mature and the necessary tasks are known, the development process that will be followed is Agile. This process will be utilized in all components of the project, from the mobile application, to the APIs used, to the hardware development.
…(expand)

2.4 Design Plan

Use Case	***Requirement***	***Dependency***
Spray the fields	Portable/Backpack Sprayer	Purchase and modify sprayer/hardware.
Collect the data	Hardware sensors for GPS, flow, direction, bluetooth, Arduino	Purchase appropriate hardware and connect to Arduino. Be able to spray fields.
Upload the data to phone	Bluetooth connection between hardware and mobile application	Use bluetooth module, ability to connect to mobile application, be able to spray fields.
Upload the data from phone to database	Store historical data about spraying application	Have spray data, have a database setup, enable communication between mobile application and database.

View the data on Mapbox on phone	Display spray data on map for user.	Ability to use Mapbox/create map, ability to get data from backpack in readable format on iOS.

2.4.1 Use case table

3.0 Statement of Work

3.1 Previous Work and Literature

While we are not directly following any other products, similar solution do exist for large scale agricultural operations. These solutions allow to collect smart data about spraying applications, such as flow, time, location, etc. - similar to the goals of this project. The purpose of this project is to use similar technology, but for a small scale operation. The usage of these technologies is similar, but the solution developed for this project is unique and not based on any particular existing work.

3.2 Technology Considerations

There is a lot of technology available for this project. There are many microcontrollers out there that we could use. Some of the strengths because we have so many options is the controllers are very cheap. One weakness is that you cannot test out to see what microcontroller would work the best. Some of the trade-offs we make are we just got a microcontroller that was pretty cheap and did not do too much research into it. There are a lot of considerations that need to be made with the hardware. We are still working out what all of these are and will expand this section. For the mobile application, both Android and iOS was considered. Given group experience and opportunity for growth, iOS was decided upon. As for the mapping ability, the main considerations were Google Maps and Mapbox. Ultimately Mapbox was chosen for cost and feature reasons.

3.3 Task Decomposition

To solve the problems the team faces and accomplish the outlined goals, the team has split into 3 teams. One team focuses on the mobile application, one focuses on the API and the backend, and the third team focuses on hardware. While all teams must communicate and work together,

each team is able to focus on specific challenges and collaboration amongst the team to progress the project.
…(expand)

3.4 Possible Risks and Risk Management

The main risks that the project faces are related to the technology being used. The hardware sensors, mapping tools/API, and communication techniques. If these end up not working after considerable time is devoted towards a particular solution, the quality of the final product may be hindered. This is why it is important to continuously test and consider possible impacts on the project throughout the process.
…(expand)

3.5 Project Proposed Milestones and Evaluation Criteria

Key milestones can be for each team - the iOS team, the API team, and the hardware team - as well as looking at the project as a whole. Creating a running application that outlines the necessary functionality is a milestone for the iOS team. Talking with the backend and establishing proper communication with APIs is another milestone. Creating a backpack sprayer that can hold the necessary hardware and collect data is a hardware milestone. Sending real data from the backpack to the application is a milestone too. Putting everything together and displaying smart spraying data to the user is the ultimate milestone for the project.

For each milestone, test cases that align with the requirements and ensure that the solution is properly made, by looking at efficiency, user experience, accuracy, etc. are some of the ways that these milestones will be evaluated.

3.6 Project Tracking Procedures

We will use GitLab Issues to track the project's tasks and progress throughout the course of the project, including both this semester and next semester. We will be able to centralize all of the work with this tool and effectively track progress at all times, as well as have the ability to go back and look at issues and work that has been completed. It will allow for planning of the project too.

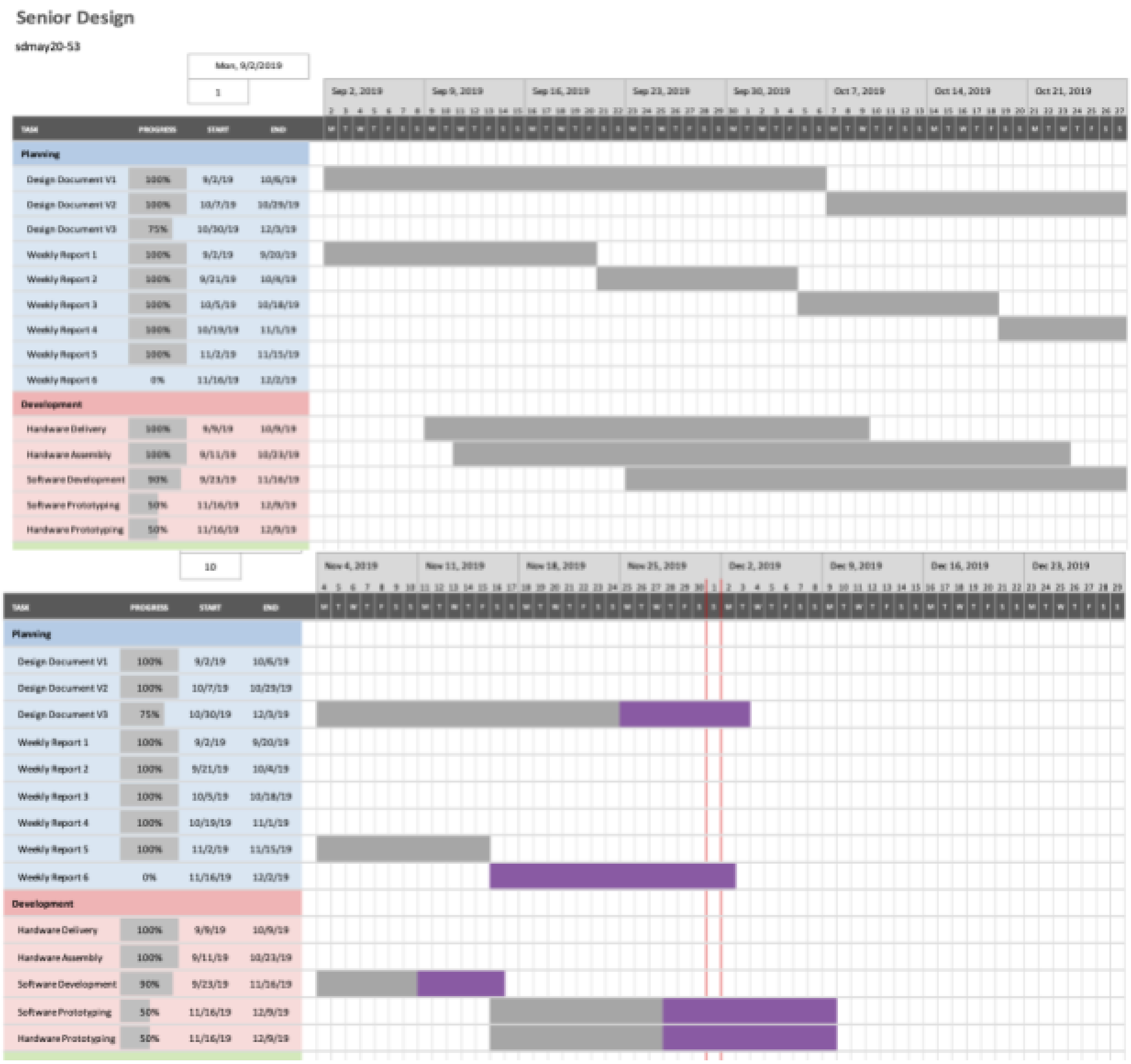
3.7 Expected Results and Validation

Along with meeting the functional and non-functional requirements, the desired outcome of this project is to be able to report spraying information back to a mobile device. The sprayer should be able to wirelessly communicate with an app to share data such as chemical, flow rate, time, GPS coordinates of spray, and direction. This information should be able to be displayed in an easy to understand map for the user. Users should be able to store historical data related to spraying applications.

4.0 Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

1. Design Documentation V1- Due Oct 6, 2019
2. Weekly Reports Interspersed throughout Semester
3. Hardware Delivery - Oct 9, 2019
4. Assembling Hardware - Oct 23, 2019
5. Design Documentation V2 - Due Oct 29-31, 2019
6. Software Completion - Nov 16, 2019
7. Integration - Dec 2, 2019
8. Design Documentation V3 (Final For Semester 1) - Due Dec 3-5, 2019
9. Prototype for Software and Hardware - Due Dec 09, 2019
10. Receive Physical Backpack - Due Jan 31, 2020
11. Testing of Prototype v1 - January 13-February 29 2020
12. Prototype v1- Due Feb 28, 2020
13. Testing of Prototype v2 - March 2020
14. Prototype v2 - Due March 31, 2020
15. Testing for final Prototype - March-May 7, 2020
16. Final Prototype - Due May 7, 2020



4.1.1 Project Gantt Chart for Fall Semester

4.2 Feasibility Assessment

The end product of this project will be a backpack sprayer that has been modified to collect data, such as GPS location, direction of spray, and flow of spray and can communicate this data, along with other data, to a mobile application. Given existing hardware components, it is realistic to accomplish this goal, and the team has experience with iOS application development which will help in this project. The main challenges will be using new tools and APIs and getting these technologies to work together and result in a successful project.

4.3 Personnel Effort Requirements

Task	***Hours***	***Resource***	***Textual Reference/Explanation***
Research Hardware	8	David and Kevin	Satisfy functional requirements. Find hardware that will give necessary data.
Research APIs	8	Sean and Donald	Determine if Mapbox is suitable. Determine how the project will use APIs - logistics between app, etc.
Research iOS	8	Shuangquan and Madison	Determine platform/tools that will be used for iOS application.
Develop Hardware	64	David and Kevin	Use each hardware component, gather data from Arduino.
Develop APIs	64	Sean and Donald	Begin to create map - how map will be used for project - use mock data until real data is available from application
Develop iOS	64	Shuangquan and Madison	Create iOS application - all aspects to fulfil requirements and create a user friendly experience.
Test Hardware	32	David and Kevin	Ensure that hardware data is accurate and proper. Ensure it is usable by application.

Test APIs	32	Sean and Donald	Make sure the API part works
Test iOS	32	Shuangquan and Madison	Make sure the iOS Part works
Integration Testing	16	Everyone	Make sure the integration between the items works
Final Testing	128	Everyone	Make sure it works and fulfils the project requirements

4.3.1 Personal Effort Table

4.4 Other Resource Requirements

Aside from assistance in purchasing the hardware and accounts needed to complete the project, the only resources that will be required is guidance and suggestions from the project's clients and advisor.

4.5 Financial Requirements

Part	Description	Quantity	Unit Price	Total
Arduino	Mega 2560	1	$13.99	$13.99
Sprayer	Field King	1	$154.95	$154.95
Flow Meter	1/2" Flow Meter	1	$9.95	$9.95
Compass Sensor	Compass	1	$8.39	$8.39
GPS Sensor	Neo 6M	1	$12.99	$12.99
Bluetooth Sensor	DSD TECH	1	$7.99	$7.99
GPS Sensor	DIY	1	$15.99	$15.99
			Total:	$224.25

4.5.1 Bill of Materials/Financial Requirements

5.0 Testing and Implementation

This project will require unit testing for the application, integrity testing for the backend and API components, and user-study testing for the hardware/sensors to ensure that the requirements are fulfilled. Each requirement will require specific testing. The non-functional requirements will need user-study testing.Testing hardware for accuracy is another example of user-study testing. The individual components that will need to be tested are the application, specific pages, functions, and use cases, the backend, map features, database interactions, hardware sensors, hardware accuracy, hardware reliability, and non-functional requirements. These will be outlined in specific tests and traced back to each requirement, to ensure that each is verified by test(s).

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

5.1.1 For software (iOS)

1. XCTest framework - overall tests
2. XCTestCase - unit tests and performance tests.
3. XCUIElementQuery - user interface tests.

5.1.2 For Hardware

No interfacing needed for hardware testing. This will be completed by testing sensors with known values and comparing the reported values. User-study tests will be used for these requirements. The nature of these tests will be easily verifiable when using known data, such as amounts of liquid sprayed, location, direction sensors are facing, etc.

5.2 Hardware and Software

– Indicate any hardware and/or software used in the testing phase
– Provide brief, simple introductions for each to explain the usefulness of each

5.2.1 For software (iOS)

We are using Xcode/XCTest to do all the testing which is iOS related.
Use the XCTest framework to write unit tests for the projects that integrate seamlessly with Xcode's testing workflow. Tests assert that certain conditions are satisfied during code

execution, and record test failures (with optional messages) if those conditions are not satisfied. Tests can also measure the performance of blocks of code to check for performance regressions, and can interact with an application's UI to validate user interaction flows.

5.2.1 For Hardware

To test hardware, we will use known values, such as known volumes of liquids for the flow sensor, known directions (cardinal directions primarily) for the compass, and GPS coordinates (longitude and latitude) and mapping software to compare GPS reported values. This will allow us to determine the accuracy and functionality of the hardware.

5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

ID	Type of Tests	Description
1	Unit Test	Tests login button can log user in
2	Unit Test	Tests sign up function can register a new user
3	Unit Test	Tests table view can display correct data
4	Unit Test	Tests cached data can be stored in Core Data
5	Unit Test	Tests the data can be displayed properly in map view
6	Integration Test	Tests the data can be received from the hardware device
7	Integration Test	Tests the data can be rendered in map view properly
8	Integration Test	Tests data can be sent out to cloud database
9	System Test	Test iOS App can works with all hardware devices

10	System Test	Tests the system complies all regulatory and legal requirements
11	Acceptance Test	Tests iOS App meets all requirements

5.3.1 iOS Tests Table

ID	Type of Tests	Description
1	User Study	After spraying a known amount of liquid, verify accuracy of Flow Sensor is within accuracy requirement
2	User Study	After facing a known direction, verify the compass reports the appropriate direction
3	User Study	After spraying at a known location, verify that the GPS reports a location within the accuracy requirement
4	Acceptance Test	Verify all hardware requirements are met.

5.3.2 Hardware Tests Table

5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

ID	Type of Tests	Description
1	Performance Test	Tests app launch speed is within 3 seconds
2	Performance/Usability/Compatibility Test	Tests UI is fluent with large amount data been displayed

3	Usability/Compatibility Test	Tests app runs on all iOS devices
4	Usability/Compatibility Test	Tests the UI of the all runs all iOS model
5	Security Test	Data can only be viewed by the owner

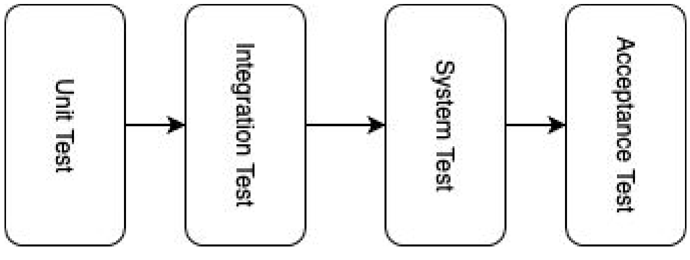
5.4.1 iOS Non-Functional Tests Table

ID	Type of Tests	Description
1	Acceptance Test	Verify that non-functional requirements are met (weight, portability, etc.)
2	Reliability Test	Verify that hardware sensors function as expected to adhere to usability requirements (ease of use).
3	Usability Test	Verify that data is transferable to application when in use (connection is strong enough)
4	Usability Test	Verify hardware works in required weather conditions

5.4.1 Hardware Non-Functional Tests Table

5.5 Process

– Explain how each method indicated in Section 2 was tested
– Flow diagram of the process if applicable (should be for most projects)



5.5.1 Flow Diagram

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place
-Modeling and Simulation: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.
-List the implementation Issues and Challenges.

Currently, the initial operations of the various components has been verified and tested informally. The hardware components have been tested for accuracy individually, and an integrated program has been developed. The backend and map service has been developed and tested with mock data.
(to be expanded upon)

6.0 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

Currently, each component of the project is under development. Individually, each area, including the iOS application, hardware/sensors and backend and APIs have made progress towards their goals. The next step will be finalizing where each stands, and begin to integrate each of these together, to create the final product. The goals include creating a functioning application that receives data from the hardware, which then uses the backend and APIs to display the data in a meaningful and efficient way to the end user. This will give the end user smart data about a spraying application. The data will include GPS locations of the spray, direction of the spray at each location, data about the spray (what chemical, when it was sprayed, etc.), and the flow rate, or how much was sprayed, during the application.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

"XCTest," *XCTest | Apple Developer Documentation*. [Online]. Available: https://developer.apple.com/documentation/xctest. [Accessed: 18-Nov-2019].

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.
If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.