Smart Laundry

DESIGN DOCUMENT

Team 21

Client & Advisor: Goce Trajcevski

Lily Mosman — *Backend Engineer*
Almedin Mulalic — *Backend Engineer*
Yazan Okasha — *Hardware Engineer*
Nathan Francque — *Hardware Engineer*
Jay Sanborn — *Mobile Development Engineer*
Grant Wanderscheid — *Team Lead and Mobile Development Engineer*

sdmay18-21@iastate.edu
https://sdmay18-21.sd.ece.iastate.edu/

Revised: 12/03/2017

Table of Contents

1 Introduction **3**

1.1 Project statement 3

1.2 Purpose 3

1.3 Goals 3

2 Deliverables **4**

3 Design **4**

3.1 System specifications 4

3.1.1 Non-functional 4

3.1.2 Functional 5

3.1.3 Standards 5

3.2 PROPOSED DESIGN/METHOD 6

3.3 DESIGN ANALYSIS 7

4 Testing/Development **7**

4.1 INTERFACE specifications 7

4.2 Hardware/software 8

4.2 Functional Testing Process 8

4.2.1 Sensor Test 8

4.2.2 Communication Test 8

4.2.3 Application Testing 9

4.2.4 Integrated System Test 9

4.3 Non-Functional Testing Process 9

4.3.1 Machine Learning 9

4.3.2 Live Store Map 9

4.3.3 UI For Breaking Down Statistics 10

4.4 Issues and Challenges 10

5 Design Testing/Implementation Results **11**

7 Concluding Remarks **11**

7 References **12**

8 Appendix **13**

List of Figures

Figure 1 **6**

Table 2 **8**

Table 1 **13**

1 Introduction

The following gives a basic summary of the project and our goals for it.

1.1 Project statement

The goal of this project is to bring clothing washers and dryers into the Internet of Things at an affordable cost of less than $30/machine. Our approach is to make a device that senses on/off states upon attachment to a broad range of laundry machines. The data gathered from the device will be accessible via an android mobile application. This enables owners to bring their machines into the Internet of Things and along with them, many more benefits (e.g. increased throughput, money savings, etc.). Ultimately, our solution will increase the efficiency for all machine owners - both residential and commercial. Users benefit from the data provided in the android app while machine owners benefit from easy commercial implementation and improved analytics.

1.2 Purpose

The goal of this project is to bring increased throughput to the owners and users of laundry machines. By allowing many existing or new laundry machines to be brought into the Internet of Things there are many benefits. On a commercial level, owners can enhance their outdated machines. By attaching our internet connected sensing unit to existing machines, customers can efficiently plan their machine access because they will know the availability in nearby stores.. On top of this, other commercial businesses, such as hotels and laundromats, can track when their machines are being run and how many resources they utilize[1]. This provides a valuable opportunity for big companies to further analyze the expenses and efficiency of their operation[2].

The smart device that we are creating allows anyone with internet access and a laundry machine to be able to track their laundry. This can be used to notify them when laundry is done as well as how much water and electricity is being used. This gives anyone the ability to become better informed about their efficiency, and at the same time improve it.

1.3 Goals

- Develop a machine state sensor that is easily attachable to laundry machines
- Have the developed device easily connect to the internet and send data to a server
- Store each sensor's data(e.g. machine on/off status, machine location, machine start time) in a database
- Have a mobile app that can pull data from the database and display it in a user friendly way that can easily be understood and used for the user to interpret the information
- Incorporate machine learning to be able to predict availability of laundry machines

2 Deliverables

- Device(s)
  - The delivered device(s) will be easily attachable to washers and dryers with the capability to sense a broad range of machines in order to tell if they are on or off. The sensing device will also be easily configured to use local wifi.
- Configured database
  - The delivered database will store various stores, machines, and users of the smart laundry app.
- Configured server
  - The delivered server will be able to gather information from nearby machine state sensing devices and then store it into the delivered database.
- Functioning android mobile app
  - The delivered mobile app will allow for users of it to easily access specific stores and their machine availability.

3 Design

Table 1 in the appendix shows the various solutions brainstormed relating to the creation of the mechanism to be attached to washers and dryers. It lists the possible implementations regarding local server communication, mounting, power, and sensing.

3.1 System specifications

After discussing as a team we came to the conclusion that a machine attachable device would be most accepted if it did not require any alterations to be done to the machine. Different machines would require different alterations of various complexities. The easiest and least intrusive way for our device to be used would be to mount it onto the outside of the machine without having to alter anything within the machine. This lead to a team decision that the proposed design be one that does not require any physical alteration within the machine. The system specifications of our design determined which functional requirements are needed to build a minimal viable product(section 3.1.2). Everything else that we could think of that could further the overall product design and experience was classified as a non-functional requirement and can be seen in section 3.1.1.

3.1.1 Non-functional

- Machine learning (predictive scheduling)
- The system must track how many loads of laundry are completed
- Battery backup
- Live feed
- Live map
- Database GUI
- Extra Database tables
- Social media plugins

- Outage reporting
- The app is responsive
- Tracks water used
- Tracks electricity used
- Tracks detergent used
- UI for breaking down machine/location(s) statistics
- Secure communication between hardware and client

3.1.2 Functional

- There must be a Physical Device that
    - is mountable
    - has sensors on it
    - is able to connect to internet
    - is set up to be able to interface with a router/local server on site
- There must be a Database that
    - is a central database
    - stores sensor information
    - interfaces with the local server and the mobile app
- There must be a local server server that
    - interfaces with the database
    - receives data from local devices
    - sends information to the database
- There must be a mobile app that
    - is used by users of the laundry facilities
    - receives information from the database
    - sends post requests to the database
    - sends get requests to the database

3.1.3 Standards

Our team is following a couple standards when developing a solution to the stated problem we are solving. The device that is being made uses standard wifi(802.11 b/g/n) to communicate with a nearby server. This specific type of wifi is going to be used because the microcontroller module follows 802.11 b/g/n and it would be difficult to change it otherwise.

As the mobile application is developed, it will follow a MVC like coding architectural standard. The way the architecture is set up is to be a dependency injected model view controller. With this code structure it allows for "black box" development. It makes development go smoother and gives the application as a whole an easy to understand architecture. As an example of that, the database does not need to be implemented right away in order to develop a functioning app.

3.2 PROPOSED DESIGN/METHOD

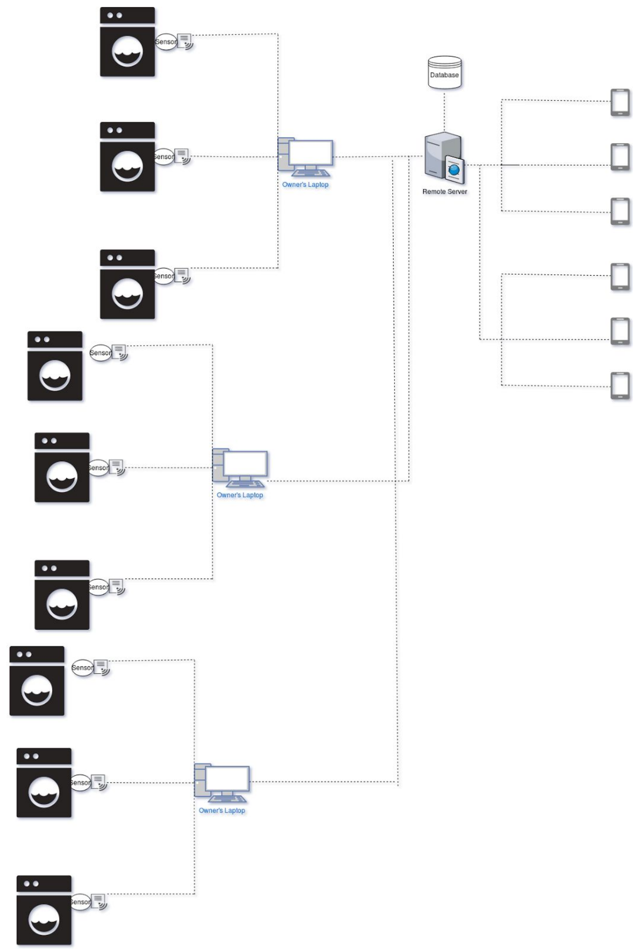

FIGURE 1.: An architectural diagram of our complete system

As shown above, the proposed design has a machine state sensing device that is attached to the outside of the a machine. This device will be developed so that it can reliably measure the vibrations of the machine it's mounted on in order to know if it is in use or not. Our sensing mechanism will then be able to communicate to a local server within close proximity of the laundry machine. This will allow for the state of machines to be sent to that local server to analyze the data. When the local server runs the data through an algorithm that we develop, it will be able to determine if specific machines are running or not. The results will then be sent and stored into a central database containing all of the machines and stores using a smart laundry sensor. This database allows for the mobile app to query it and relay store/machine information to a user pertaining to a user specified store and machine.

To effectively sense the machines on and off state we decided to go with a induction sensor. The sensor is then onboard the microcontroller that is wifi enabled to communicate with the local server. We also went with the design decision to have a microcontroller that needs to be plugged in at all times rather than having some sort of battery to it.

3.3 DESIGN ANALYSIS

Strengths of proposed solution:

- Non intrusive to laundry machines
- Easy to add our solution to most machines
- Machine specific wiring research is not needed from us or machine owner
- Cannot run out of power when plugged into an outlet at all times
- Cordless/clutter-free communication between sensing unit and local server
- Able to provide real time data of laundry machines to users

Weaknesses of proposed solution:

- No 100% guarantee for state of machine
- Needs an available plugin in close proximity
- Machine state sensor has the ability to be stolen since it is on the machine exterior

Observations from testing the proposed solution:

Our group has developed a solution that uses a fast vibration switch to detect and collect data on the subtle movements of the machine it's attached to. It then sends the collected data to a client side server for processing. The server performs data analysis to determine the machine's state and once the state is determined, then it sends a post request to our database. Our mobile app monitors the database for changes and then displays the updated information to the user.

During our first observations of the system as a whole, we see that the vibration sensor is not able to reliably sense the machine state as much as we expected. This aligns with one of our expected weaknesses but we did not predict that it would be as unreliable as it has shown to be. Despite this weakness, many of our predicted strengths are proving themselves. The cost of implementation still remains low, the design remains chordless, and the data collected can be seen in the mobile app real time.

4 Testing/Development

Testing for our project is broken down into phases. In the following sections, the major aspects of testing will be explained.

4.1 INTERFACE SPECIFICATIONS

In our design there are multiple points where our different encorporated devices will need to be able to communicate with each other. The machine state sensing device needs to be able to send machines' status updates to a local server, this will be done through wifi. Communication between the local server and central database will then happen through the use of a Firebase library that allows for storage of information into the database. The mobile app will also communicate in the same way, using a Firebase library that allows for the storage and retrieval of database information.

4.2 Hardware/software

The hardware we are using in our testing phase is various laundry machines. The washing and drying machine models can be seen in table 1 below. By testing our machine state sensing device on various machines, we can adapt a sensing algorithm that can work on a broad range of machine models.

Model	Machine Type
xxxxx	Washer
xxxxx	Dryer

Table 2 *The table above shows that various washers and dryer models that were used during the sensor's testing phases. This testing has yet to be done at this point.*

4.2 Functional Testing Process

The following functional testing processes test the functional ability for the developed solution to properly function when tested individually and when tested as a whole.

4.2.1 Sensor Test

Testing of the sensor's capabilities included the following:

1. Successful reading of on/off state of machine when it is running or not running on various machine models
2. Being able to not have the sensor switch the on/off state of the machine's status when it is being unloaded or loaded
3. Not allowing another machine's vibrations that is in contact with subject machine to trigger on status

4.2.2 Communication Test

When testing the communication we will examine the connections made between all stages of the integrated system separately as follows:

- **Sensing unit** -> **Local server**
    - Confirm that machine on/off state can be transmitted from the sensing unit to the local server through wifi connectivity
- **Local server** -> **Central database**
    - Confirm data can be sent from the local server to correct location in central database
- **Central database** <-> **Mobile app**
    - Confirm that data can be received and sent to and from the central database and mobile application

4.2.3 Application Testing

Testing the application on its own consists of navigating throughout the application on multiple devices and making sure it is responsive to what the user is pressing. For our application to be considered "responsive" we are defining it to have no noticeable delay between pressing something and a resulting action being carried out.

4.2.4 Integrated System Test

An entire working system will not be tested until all individual parts outlined above are first tested and confirmed to be fully functioning. The integrated system testing will happen in three stages.

1. First Stage - Using only one machine state sensing mechanism registered to one store
    a. For the first testing stage to be considered successful, the on/off status of the machine must be able to be seen by a user of the mobile app within a minute of it turning on or off. This should be reflected in both the "store map" page and the "machines" page.
2. Second Stage - Using 3 machine state sensing mechanisms registered to one store
    a. For the second testing stage to be considered successful, the on/off status of multiple machines must be able to be seen by a user of the mobile app within a minute of it turning on or off. This should be reflected in both the "store map" page and the "machines" page.
3. Third Stage - Using 6 machine state sensing mechanisms registered to 2 stores
    a. For the second testing stage to be considered successful, the on/off status of multiple machines from multiple stores must be able to be seen by a user of the mobile app within a minute of it turning on or off at the correct store location. This should be reflected in both the "store map" page and the "machines" page for the user selected store.

4.3 Non-Functional Testing Process

We have omitted some non-functional requirements from our testing process because we will not be able to complete all of them within the allotted time frame. The testing procedure we have included are for the non-functional requirements we intend to complete.

4.3.1 Machine Learning

The constructed machine learning model should be able to predict occupancy time frames of the washers/dryers with an accuracy of 80%. This would ensure that the user does not waste their time waiting for inaccurate data reducing their overall experience.

4.3.2 Live Store Map

When testing the live store map, it will be evaluated based on the ease of use for the user and the amount of helpful information shown. The testing of the store map should reflect the associated machine states accurately within five minutes to be considered a live store map. The ease of use for the user and the helpful information shown will be evaluated as a pair, as they go together. Users should be able to dissect the displayed information easily in order to obtain the information they are looking for. The information displayed on the store map should be machine numbers, locations, and their state. It should be easily seen that certain information corresponds to certain machines as to not confuse users.

4.3.3 UI For Breaking Down Statistics

The statistics UI would be a data rich display that would allow the user to view comprehensive information about their laundromat and/or individual washer/dryer. This display would be tested on the user's perceived ease of use and the ability to present relevant data such as machine state, average predicted availability, and daily usage metrics. If the UI is able to provide these criteria for the user on an hourly basis, the test would be considered successful.

4.4 Issues and Challenges

- Sensors
  - Finding the right sensor is quite difficult, we need a low cost sensor that is able to reliably sense data. So far, we have encountered numerous issues with the "Fast Vibration Sensor Switch". The sensor requires movement focused at the sensor head to detect vibrations. This itself is a large problem as the sensor has to be sitting directly on the machine and cannot be moved. We have also found that the range of values we obtain when sensing is too large to be accurately analyzed. The troubles with the vibration sensor have lead to increased sensor research and the ordering of a non invasive current sensor. If the current sensor is able to obtain accurate readings, this will be a great fix for the problems we have faced.
- DHCP changing client side server IPs
  - When the client side server runs, the IP of the computer is configured to the hardware. The issue is most routers run on DHCP which will dynamically allocate IPs to a computer. When that occurs, the hardware will post data to a non-existing IP and all the necessary data will be lost. Currently, we do not have a solution and we are researching possible ideas.
- Interactive Store Map
  - We have faced multiple challenges with implementing an interactive store map. One challenge has been figuring out the best way to implement it. First we thought about implementing it within the android application. After doing research and talking to Professor Ponpandi about options for implementing it, we decided this approach was too time consuming and would have many places for error. The approach we decided to go with is one that implements the interactive store map on a webpage. We found that this could be done with relative ease and the web page would be presented as part of a webview within the mobile app.
- Database
  - We originally intended to use a SQL database in tandem with our mobile application. This was a challenge for our team because of how much data we were going to need to store within it, the scalability of it, and learning how to actually configure the database. After struggling to implements the database in this way we stumbled upon Google Firebase and decided to implement it instead. Firebase allows for easy database management and real-time updates within our mobile application which made the switch an easy decision. Changed data can easily be

monitored within the mobile application using the imported Firebase package while also allowing for the easy expansion of the database as we inevitably add more machine, stores, and users to it.

5 Design Testing/Implementation Results

During the testing of our implementation we have had many successes. The developed mobile app has been able to work very well in displaying appropriate information to the user in a helpful manner. It has a very natural flow within the app which creates a good user experience. Developing the database for use with our system has presented the challenge of scalability and learning how to implement it effectively. Luckily we were able to solve this problem relatively quickly by discovering and using a google firebase (also leading to much success with mobile app with data querying). The firebase implementation of a database has also facilitated in the overall success of communication we have been able to see between the local server that we developed and the database. This has allowed for data collected by the local server to be sent with some simple database post requests.

The local server we implemented gave us some troubles in the earlier stages. Initially, it was designed to use C#. However, this did not allow external IPs to connect to the server. This proved to be a large problem as we need the hardware with an attached sensor to send a post request to the application. This lead to a switching of approach to a node JS server. This language was not as familiar for some of our team members and development was slow, but it allowed for external IPs to connect to the application. The sensing unit that we have developed thus far has been able to reliably communicate post its data to the local server for further analysis despite the sensor reading being inaccurate.

The sensor is not accurate enough and gives a large range of numbers when active. We cannot analyze and interpret this type of data due to the uncertainty that propagates because of this. To overcome this, we have moved towards different sensors.We have learned that we should have kept exploring different sensors at the early stages. We also learned that we needed a more robust testing mechanism for the sensor as we would have caught the issue earlier. Currently, we are exploring an induction sensor to detect current flowing to the machine. The sensors we are exploring are still non-invasive and would be placed over the power cord of the machine. As we move forward we are also closely observing the ability of the sensor to reliable communicate the machines state with the local server.

7 Concluding Remarks

The goal of this project is to bring more efficiency to the owners and users of laundry machines. By allowing any existing or new laundry machine to be brought into the Internet of Things, their processes can be more closely monitored by anyone who owns or uses the machine. In order to accomplish this goal we will develop a working machine sensing device, expandable database, and mobile application. The way in which each of these elements are implemented is done so that the functional requirements set in place are not compromised and the quality of the non functional requirements are optimized.

7 References

List any references used in the document. These are an essential part of your review so far.

- NodeJS Docs
  - https://nodejs.org/en/docs/
- ExpressJS Docs (framework for NodeJS)
  - http://expressjs.com/en/api.html
- Firebase Real Time Database
  - https://firebase.google.com/docs/database/android/read-and-write
- Android Native Interactive Store Map
  - https://blahti.wordpress.com/2012/06/26/images-with-clickable-areas/
  - http://catchthecows.com/2011/08/09/ImageMap_for_Android/
  - https://forums.xamarin.com/discussion/3196/how-to-get-image-mapping-coordinates
- Arduino WiFi Library Docs
  - https://www.arduino.cc/en/Reference/WiFi
- Arduino HttpClient Library Docs
  - https://www.arduino.cc/en/Tutorial/HttpClient
- Python Serial Library Docs
  - https://pythonhosted.org/pyserial/
- Python Argparse Library Docs
  - https://docs.python.org/2/library/argparse.html
- Interactive Web Based Store Map
  - https://html.com/images/how-to-make-an-image-map/
- Android Clean Architecture
  - https://medium.com/@dmilicic/a-detailed-guide-on-developing-android-apps-using-the-clean-architecture-pattern-d38d71e94029

8 Appendix

Communication with local server	Power	Mounting	Sensing
• Bluetooth • Wifi • Ethernet	• Plugged in • AA or similar batteries • Rechargeable battery pack	• Suction cup • Screw onto machine • Glue • Double sided tape like adhesive	• Vibrations • Magnets • Integrated wiring • Sound • Accelerometer

Table 1 .: Above is the table of brainstormed ideas for possible solution combinations