sddec18-17: IoT Remote Monitoring Mobile App for Commercial Appliances

Week 5 Report
February 28 - March 10

Team Members

John Fleiner — *Mobile Application Development Lead*
Ben Young — *iOS Development Lead*
Thomas Stackhouse — *Backend Lead*
Hongyi Bian — *Hardware Test*
Yuanbo Zheng — *Meeting Facilitator*
Casey Gehling — *Meeting Scribe / Backend developer*

Client

Greiner Jennings Holdings
Taylor Greiner
Connor Jennings

Faculty Adviser

Goce Trajcevski

Summary of Progress this Report

Since the previous status report, our team has seen significant progress in all three areas of development: front-end, backend, and hardware. The android team made several additions to the previously developed UI screens. The location activity (screen) now includes toggle buttons on each of the washing machine rows in the expandable list menu so that several appliances may be selected at once. Once ready to reserve, a user is taken to a reservation page where they may select a month, date and time that are available for reservation. Near the end of the week, the mobile team analyzed the current layout of the application in hopes of finding a more optimal solution that would simplify the amount of required screens to perform a reservation. The backend team worked on implementing gradle profiles. Spring Boot profiles allow for different configurations to be loaded into Spring Boot. Spring Boot functionality was then extended to include API endpoints for basic services such as users, appliances, and laundromat locations. Each service returns a JSON object that may be parsed by the mobile team. The hardware team setup a raspberry pi microcontroller in a private network environment. To create a mockup of how to turn a washing machine controller on/off, the team used a set of LEDs and resistors on the breadboard to test GPIO functionality through a Python script. Once the LED could be turned on/off, AWS IoT was used to register the raspberry pi.. A python/bash script on the raspberry pi then received commands from AWS IoT to turn the LED on/off.

Pending Issues

Our prior status report listed several pending issues including: obtaining a washing machine controller and obtaining/storing a full-sized washing machine.

Washing Machine Controller
We have yet to receive a washing machine controller, however our clients have provided us with an update. We will be receiving a full-sized washing machine and a portable washing machine. The portable washing machine will replace the need for obtaining a washing machine controller.

Full-Sized Washing Machine

The full-sized washing machine that will be obtained is a Fisher & Paykel GWL 15 Washing Machine. Our current pending issue with the full-sized washing machine is whether or not the machine will be placed/stored on campus. If the machine is to be stored on campus, our team will need to receive clearance to do so. If the machine is to be accessed remotely, there may be an issue in regards to interfacing our raspberry pi with the appliance.

Portable Washing Machine

The portable washing machine that will be obtained is a Best Choice Products SKY2767 Portable Washer. Our current pending issue with the portable washing machine is that is has not been acquired by us. We will need to meet with the client to pick up the portable washer so that the hardware team can connect their raspberry pi to turn the machine on/off.

Plans for Upcoming Reporting Period

Backend

Spring Security integration will start being set up and researched more, as before having the system integrated with actual washing machine data that data should be protected. Looking into the setup of the backend on an AWS EC2 instance will also be looked at, as the mobile development team will want somewhere external to start testing their API calls on physical mobile devices.

Front-End

For the upcoming reporting period, the mobile development team will works towards making significant UI changes to the current state of the application. Currently, the method for reserving an appliance goes several screens deep and we believe that it can be simplified. To do so, our team will be implementing Google Maps to search for laundromat locations based on GPS location. Second, rather than having several screens to iterate through selection choices, a single form screen will be used to query our database for all data that meet the users' request.

Hardware

The hardware team plans on receiving the portable washing machine so that the raspberry pi can interface with the washing machine on/off switch.

Summary of Weekly Adviser Meeting

To help prepare for the both the second version of the design documentation and the panel presentation, our advisor asked that we provide an analysis of the decisions outlined in the design document. For example, our team selected AWS IoT as our cloud service solution without providing an analysis comparing AWS IoT with Bluemix, MS Azure, Google Firebase, etc. A comparison amongst the various cloud service would provide a pros and cons chart that can be used to validate our decisions when presenting to a panel. Similarly, the mobile platform is using the native environment and programing languages for Android and iOS rather than using a cross-platform framework/development tool. It is required that we provide a detailed explanation as to why one decision is made over another.

Our advisor also asked that we provide a demo plan that clarifies what will and will not be presented during the panel presentation. The demo plan is to be shared with our clients so that the ideal deliverables may be confirmed.

To inform Goce of our current development progress, the mobile team provided a demo of the mobile application prototype. It is planned that our clients will receive a similar demo during the next client meeting.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
John Fleiner	This week, I continued to make development progress on the Android application. Location Activity (Screen): As a brief review, the location activity is a screen that displays a list of laundromat locations. Each location can be expanded to view the selected location-based washing machines. This week, I updated the expanded rows that list the washing machines. Each washing machine row now has a status indicator that represents whether the machine is available or reserved, similar to the online/offline status used by facebook. I also added a toggle button to each row so that multiple washing machines may be selected. Reservation Activity (Screen): The reservation activity provides a relatively clean and simplistic way for users to select a month, day, and time to reserve a set of washing machine appliances. The Month/day is represented by a horizontal list, similar to the active friends horizontal sliding list used by facebook. The month/day can be swiped left or right. The times can be viewed in a standard list that lays below the horizontal list. Once a month, day, and time have been selected, the customer may proceed to the payment/checkout screen.	15	66
Ben Young	During this week of development I made chances to the location view and made decisions on how the reservation should be taken by the user. Location View (Screen): I changed the design of the location screen to include the reservation system on the same page to be able to decrease the amount of views the user has to go through to reserve a machine. At first I displayed the amount of free washers and dryers on the location page but decided against this because of further discussion. I made this decision because John and I came the conclusion that the customer doesn't really care how many machines are free a a location they only care about if the number of machines they want to reserve are	8	42

	available at that location.		
Thomas Stackhouse	There were two major things that I worked on during the time of this status report. The first one was defining the design document portion which described the back end solution and the different portions contained in it. I also worked on fixing an error in the database accessing of several of the new objects added in the previous status section, and then implemented “gradle profiles” similar to spring profiles for easier deployment. During the database accessing error fix, I helped a team member spot an error with how the database accessing was set up, as well as helped streamline the service and repository methods by pointing out repository methods that did not need to be created due to pre-defined functions that the team member did not know about that are automatically created with the repository interface we are using. Finally, the largest thing that I worked on was implementing gradle profiles. Spring Boot profiles are used to load different configurations in Spring Boot, and allow switching things like database connections, mocked external calls vs actually calling external sources, etc. by using commands to define profiles on the application startup. However, there is not a simple solution such as Maven build profiles. Therefore, I engineered a solution that added custom Gradle profiles to the back end based on a compilation of several reference materials that I found. Currently, there is a local profile and a remote profile, and these are implemented using ‘-Plocal’ and ‘-Premote’ respectively. If no profile is defined at startup, the default profile is local. During the next development cycle, I plan on looking into setting up Spring Security and looking into deploying the solution onto an EC2 instance inside of AWS.	17	44
Hongyi Bian	This week our hardware team have made a significant progress in terms of knowing how to control a local LED from AWS IoT cloud. We first set the raspberry pi up in a private network environment (will eventually get this	10	33

	part ready during demo period). Then tested a couple of GPIO pins through local Python scripts. This involved with having some electronic components on breadboard in order to test the GPIO functionalities. For simplicity, we used a set of LEDs and resistors in the end to test our remote control functionality. The last job was to set up a “thing” on the AWS IoT cloud, this includes establish a device registration, protocol selection, certificates generation and fundamental connection. So that in the end of this week, we were able to run a simply test Python/bash script on the raspberry pi to receive commands from the cloud, turn on/off the LEDs accordingly. There still exist high voltage control as well as network stablibilities issues. So our goal is to further look into those problems in next development week (week 6).		
Yuanbo Zheng	During this week, our hardware team can control a LED from our breadboard through the AWS cloud. This means we can control our device remotely which is a very important progress. We first reset the network settings on the Raspberry pi then the Raspberry pi can connect with a private network. After that, we find out which GPIO ports of the Raspberry pi could send out the PWM signal. After we connected the Raspberry pi with our breadboard (There were already a LED light and some resistors on that), we can turn on/off the lights through the Raspberry pi by using the implemented Python program. The last thing which is the most crucial part is that we need to turn on/off the light through AWS IoT test panel. We will publish some “commands” from the AWS cloud. When our Raspberry pi received the expected “commands”, it will turn on/off the LED light.	10	37
Casey Gehling	During this period, I extended functionality to include API endpoints for basic services such as user, appliance, and location. These services are able to pull from the remote database (depending on the option given when starting up the spring boot server) and return results in JSON format to the caller.	9	37

	Currently, you are able to pull all of the different types of records, as well as create a new one. I ran into a couple of bugs, and it set me back a little, but Thomas was able to help me resolve the issues during a weekend meeting. I also added functionality to allow a user to register and login to our application.		