# IoT Remote Monitoring

## For Commercial Appliances

*Team 17*

**Clients:**
Taylor Greiner
Connor Jennings

**Advisor**:
Goce Trajcevski

**Members**:
Hongyi Bian,
John Fleiner,
Casey Gehling,
Thomas Stackhouse,
Ben Young,
Yuanbo Zheng

# Presentation Outline

- Project Plan
- Design
- Plans for Next Semester

# Project Plan

# Overview/Motivation

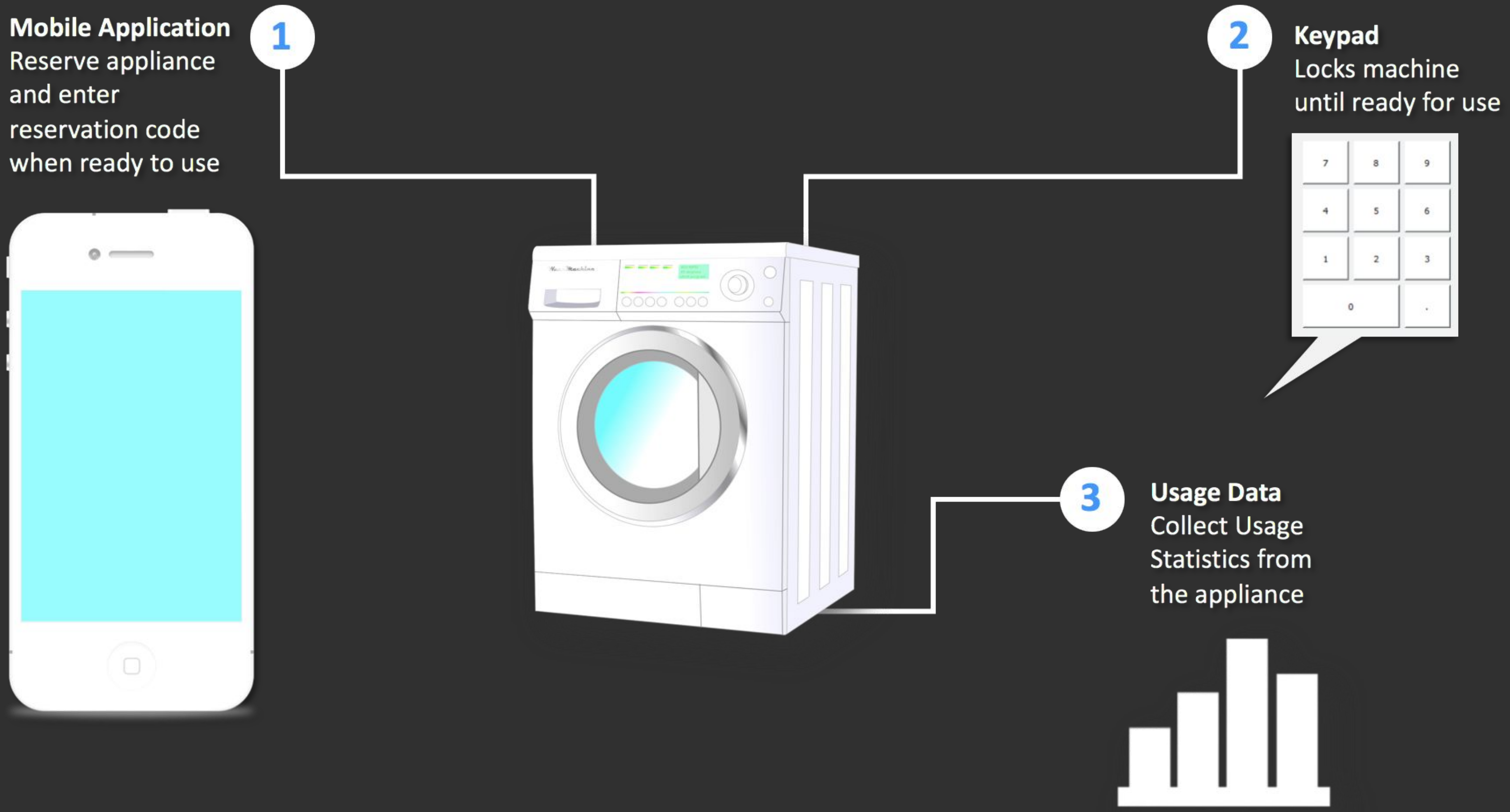
Laundromats are hard to manage and inefficient to use

- Customers
  - Have to go to a location to check if appliances are available
  - Additional waiting time
- Managers
  - Additional cost to manage remotely
  - Wear and tear estimations

# Problem Statement

Create a solution that uses the Internet of Things to allow commercial appliance reservation by users via a mobile interface.

# Concept Diagram



**Mobile Application**
Reserve appliance and enter reservation code when ready to use

**Keypad**
Locks machine until ready for use

**Usage Data**
Collect Usage Statistics from the appliance

# Functional Requirements

- Reservation System
  - Remote users
  - Power control over appliance
- Online Transactions
  - 3rd-party software
- Data Tracking
  - Laundromat schedule

# Non-Functional Requirements

- Security
  - End to end security
  - Online Transactions
- Performance
  - User Experience
  - Battery Life
- Scalability
  - Clients Vision
- Usability
  - Two Native Mobile Applications

# Potential Risks and Mitigations

Raspberry Pi

- Waterproof
  - Waterproof Enclosure box
- Theftproof
  - Fasten to washing machine

Amount of Accesses to AWS Server

# Constraints & Considerations

Constraints

- Use of Battery for Mobile
- Network Connection in Laundry

Considerations

- Strengths of Our Design
  - Supporting both IOS and Android
  - App Battery-Friendly
- Potential Weaknesses of Our Design
  - Manually Connected the Microcontroller with Commercial Appliances (1:1 relationship)

# Market Survey

**IoT Pay-per-wash Industry: April 21, 2018:**

- Subscription based services (washing machine home installation; monthly charges)
- Companies such as UW Huismeester and Bundles implement the pay-per-wash business model

**Berendsen: Microsoft Azure and IoT Hotel Laundry Service: April 13, 2018:**

- IoT tracking of linen ID tags (1 million pieces throughout Europe daily)

**Samsung Electronics Laundry Innovation - WW6850N Washing Machine: January 7, 2018:**

- IoT compatible with Samsung’s SmartThings ecosystem

# Resource and Cost Estimate

| Items | Price |
|---|---|
| Amazon Web Service (IoT) | Free-Tier |
| Portable Wash Machine | $109.97 |
| Raspberry Pi 3 Model B | $35 |
| 16x2 LCD Display | $9.95 |
| 3x4 Matrix Keypad | $7.5 |
| 4 Channel Relay Module | $7 |
| Basic electronic kit (jump wires, resistors, etc.) | ~$15 |
| **Total:** | $184.42 |

# Project Milestones and Schedule - Semester 1

**Milestones**

*Mobile App*

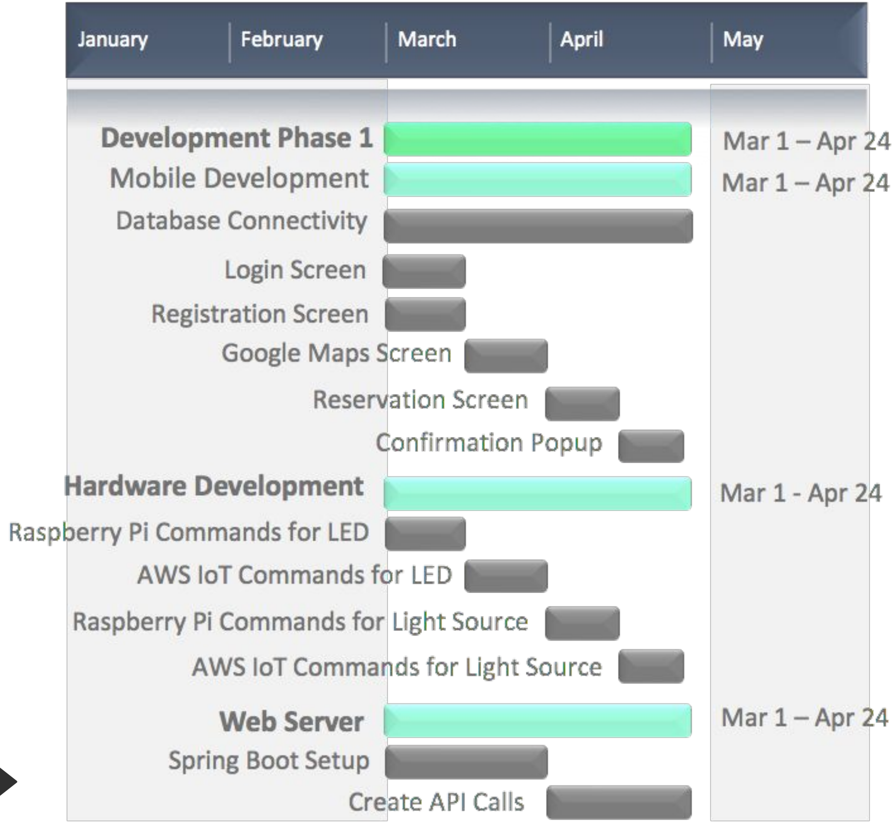
- Android and iOS

*Backend*

- Spring Boot and API requests

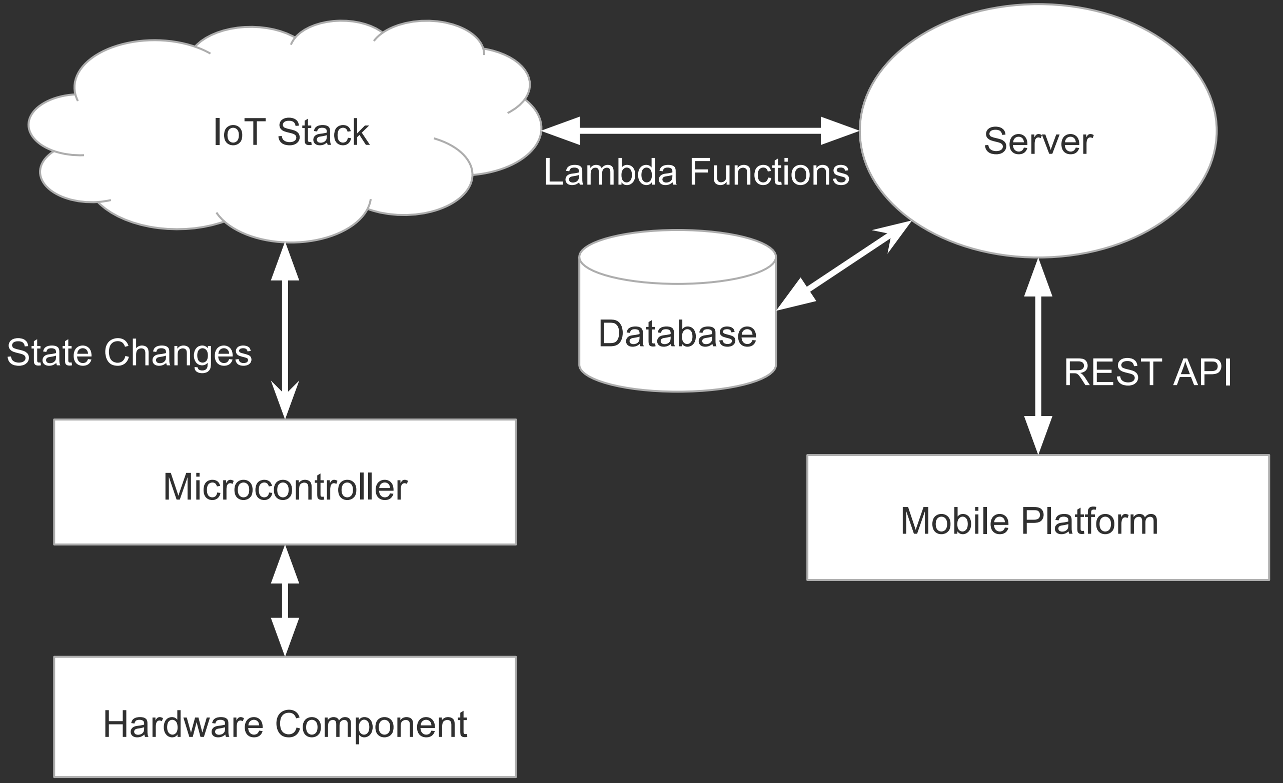
Hardware

- AWS IoT Commands (Power)
- Raspberry Pi



# Design

# Architecture Diagram



# Functional Decomposition

Mobile

- Android and iOS Native Applications

Backend

- Java Spring Boot server, AWS, database

Hardware

- Raspberry Pi microcontroller, commercial appliances, digital display, keypad

# HW/SW/Technology Platforms Used

## Mobile Development

Android IDE: Android Studio

iOS IDE: xcode

Android Language: Java

iOS Language: Swift

## Back-End

Spring Boot

AWS IoT

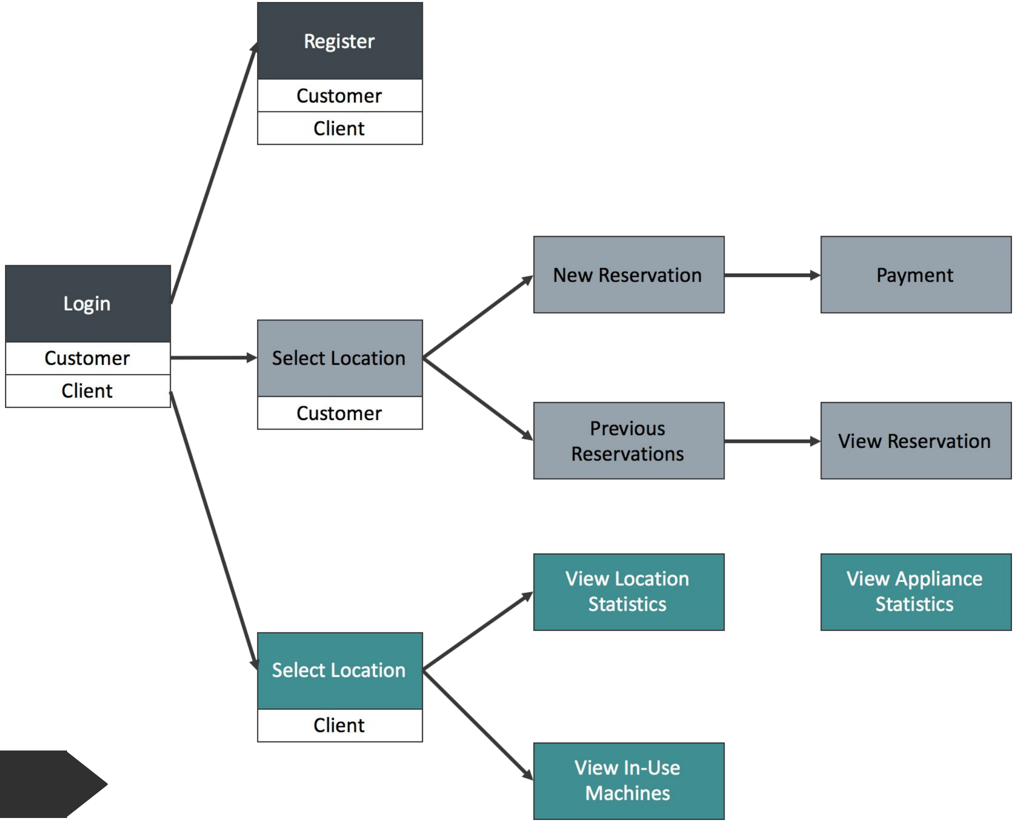
## Hardware

Single-Board Computer:
Raspberry Pi 3 Model B

# Detailed Design - Mobile Application

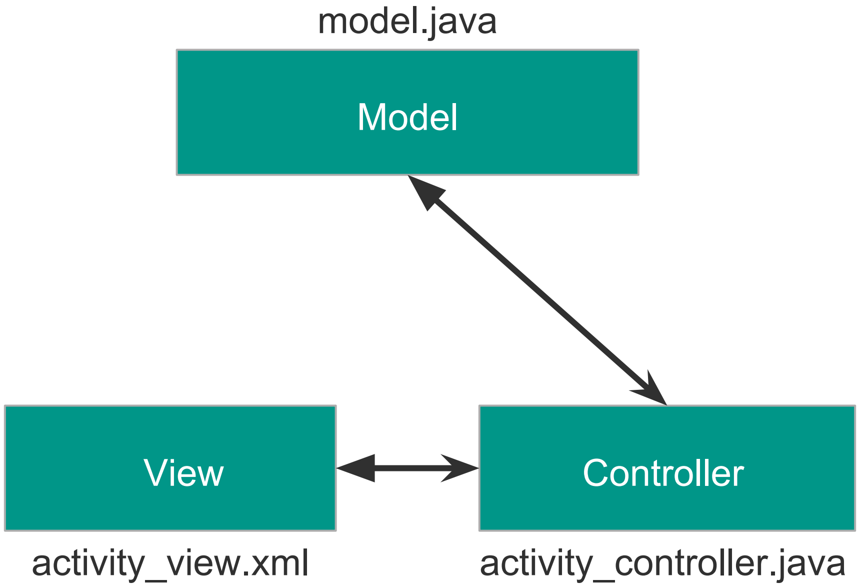
- Screen Sketch Diagram
  - Similar pattern to the *Uber* Model

- Dual-UI System
  - Customers and admins vs drivers and riders

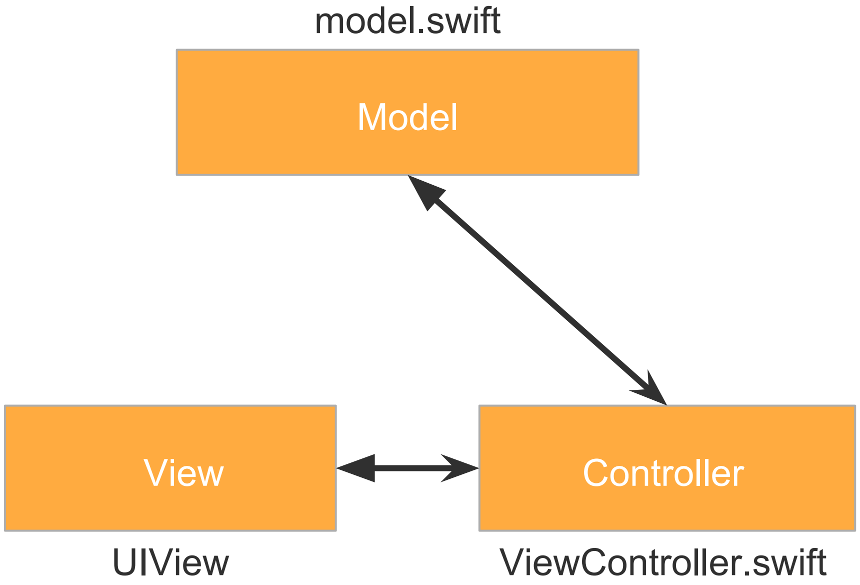


# Detailed Design - Mobile Application

Android

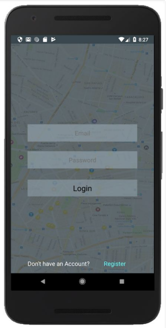


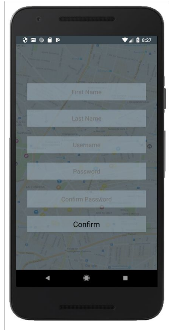
iOS

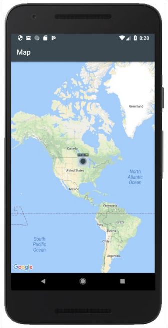


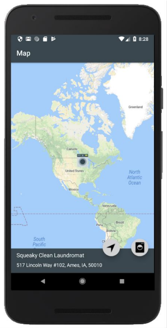
# UI

## Android

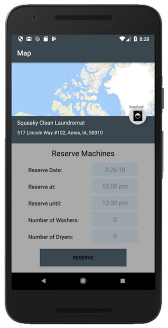








Squeaky Clean Laundromat
517 Lincoln Way #102, Ames, IA, 50010

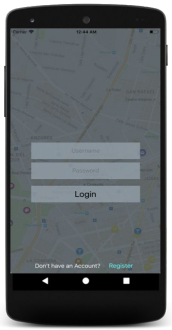


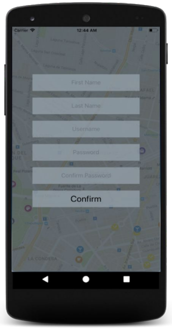
Squeaky Clean Laundromat
517 Lincoln Way #102, Ames, IA, 50010

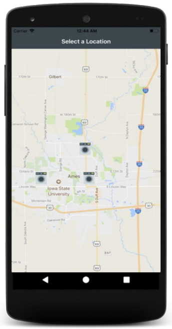
Reserve Machines

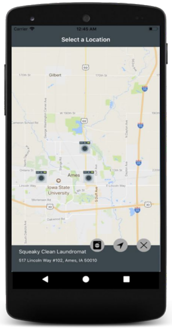
Reserve Date: 3-26-18
Reserve at: 12:00 pm
Reserve until: 12:30 pm
Number of Washers: 0
Number of Dryers: 0

## iOS

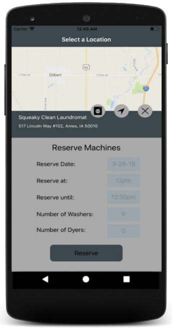








Squeaky Clean Laundromat
517 Lincoln Way #102, Ames, IA 50010

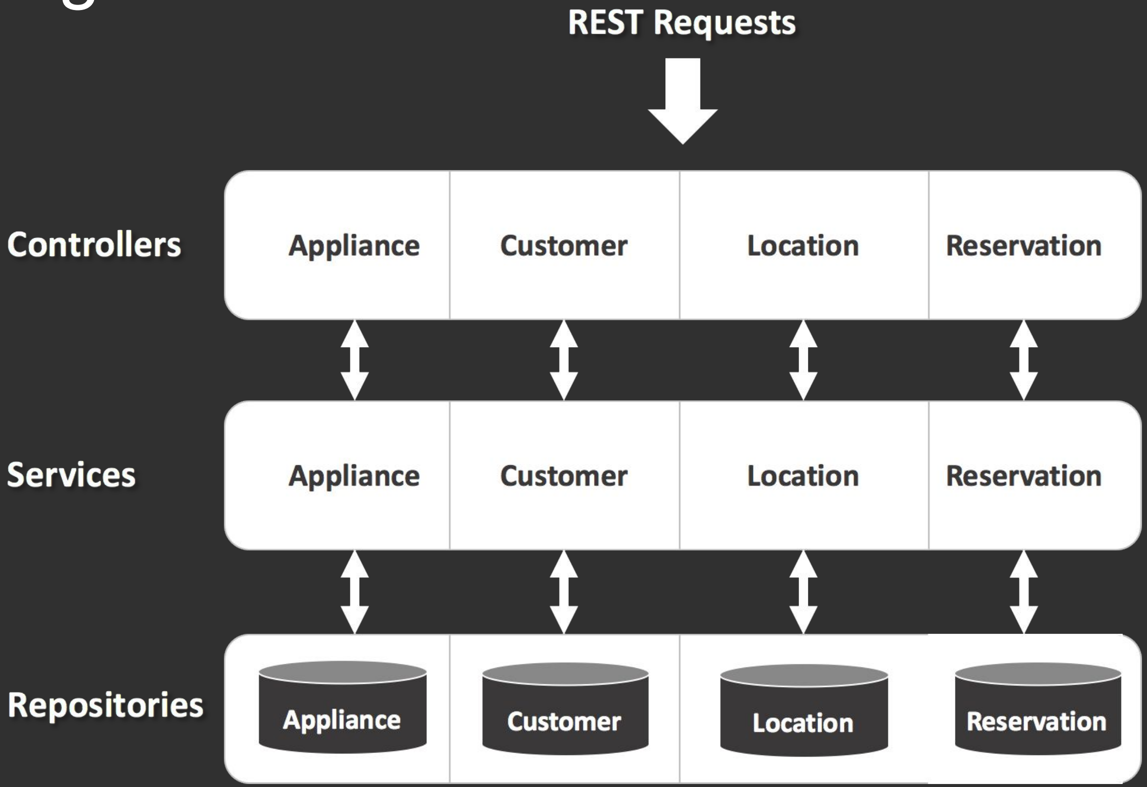


Squeaky Clean Laundromat
517 Lincoln Way #102, Ames, IA 50010

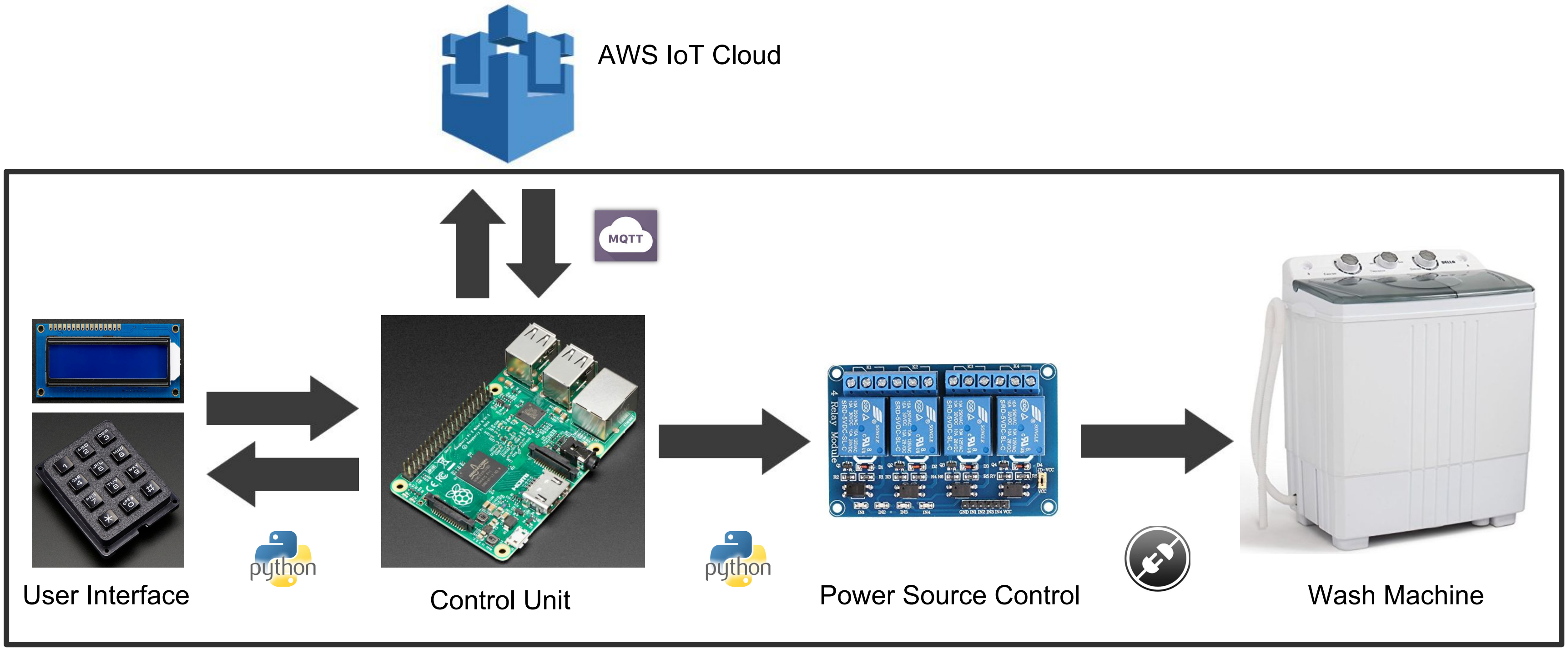
Reserve Machines

Reserve Date: 3-26-18
Reserve at: 12pm
Reserve until: 12:30pm
Number of Washers: 0
Number of Dyers: 0

# Detailed Design - Backend



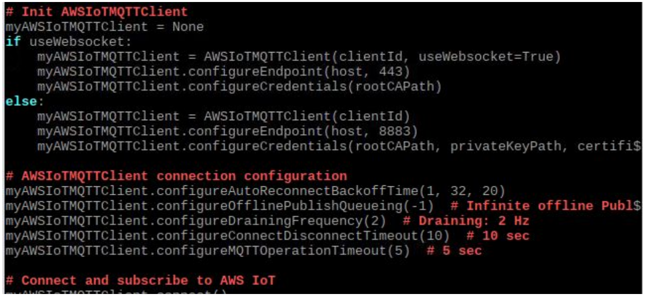
# Detailed Design - Hardware



# Detailed Design - Hardware

## Script

### Python

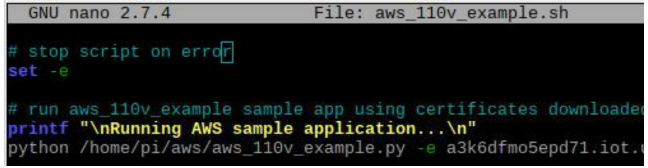


```
# Init AWSIoTMQTTClient
myAWSIoTMQTTClient = None
if useWebsocket:
    myAWSIoTMQTTClient = AWSIoTMQTTClient(clientId, useWebsocket=True)
    myAWSIoTMQTTClient.configureEndpoint(host, 443)
    myAWSIoTMQTTClient.configureCredentials(rootCAPath)
else:
    myAWSIoTMQTTClient = AWSIoTMQTTClient(clientId)
    myAWSIoTMQTTClient.configureEndpoint(host, 8883)
    myAWSIoTMQTTClient.configureCredentials(rootCAPath, privateKeyPath, certifi$

# AWSIoTMQTTClient connection configuration
myAWSIoTMQTTClient.configureAutoReconnectBackoffTime(1, 32, 20)
myAWSIoTMQTTClient.configureOfflinePublishQueueing(-1)  # Infinite offline Publ$
myAWSIoTMQTTClient.configureDrainingFrequency(2)  # Draining: 2 Hz
myAWSIoTMQTTClient.configureConnectDisconnectTimeout(10)  # 10 sec
myAWSIoTMQTTClient.configureMQTTOperationTimeout(5)  # 5 sec

# Connect and subscribe to AWS IoT
```

### Bash

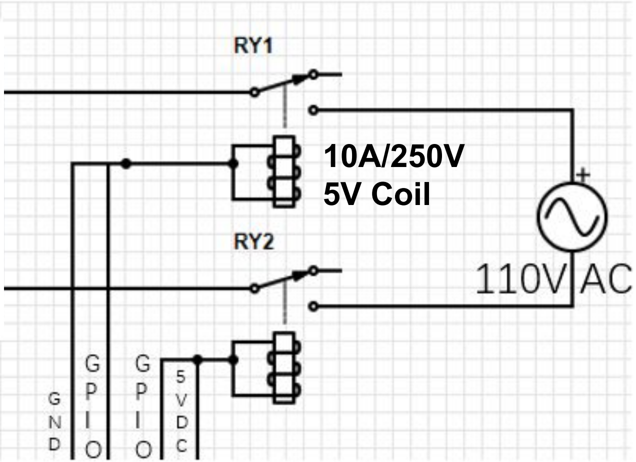


```
  GNU nano 2.7.4                 File: aws_110v_example.sh

# stop script on error
set -e

# run aws_110v_example sample app using certificates downloaded
printf "\nRunning AWS sample application...\n"
python /home/pi/aws/aws_110v_example.py -e a3k6dfmo5epd71.iot.
```

## Circuit



**Washer Power: 260W**
**Spiner Power: 140W**
**Voltage: 110V/60Hz**
**Current: 1.2A ~ 2.2A**

# Plan For Next Semester

# Plan For Next Semester - Mobile

- Client - Side Mobile Application
  - Appliance Usage Data
  - Android and iOS Charts Library
- Testing
  - Android Profiler → Memory, CPU, Network
  - Debug Analyzer → CPU, Memory, Network
  - Android junit → Correctness
  - iOS unit → Correctness
  - Usability Survey
- Security
  - Payment Transaction Service → Braintree or Stripe SDK

# Plan For Next Semester - Backend

- Testing
    - Integration
    - Unit (~80-90% coverage)
- Security
    - Spring security
- Data Analytics
- Functionality Enhancements
- Performance Optimization
- Documentation
    - API
    - Javadoc comments

# Plan For Next Semester - Hardware

- Keypad & LCD Screen
    - Implementation
- Integration with Portable Washing Machine
    - Exterior Setup
    - Network Environment Setup
- Testing
    - Circuit Correctness Testing
    - AWS IoT Communication Testing
    - User Interface Testing
- Security
    - Raspberry Pi SSH Security
    - MQTT Channel Security

# Responsibilities and Contributions of Members

**Mobile Team - Responsible for the Android and iOS App Development**

- John Fleiner
- Ben Young

**Backend Team - Responsible for Spring Boot and AWS IoT**

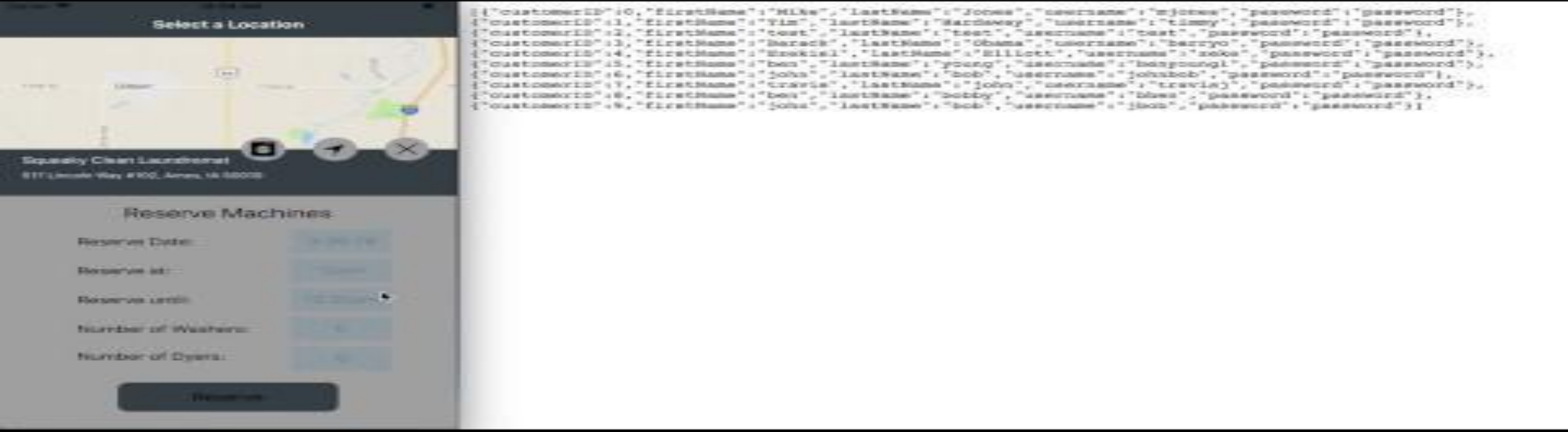
- Thomas Stackhouse
- Casey Gehling

**Hardware Team - Responsible for Raspberry Pi and Washer Components**

- Hongyi Bian
- Yuanbo Zheng

# Prototype Demo - Hardware

# Prototype Demo- Front-End & Backend



```
[{"customerID":0,"firstName":"Mike","lastName":"Jones","username":"mjones","password":"password"},
{"customerID":1,"firstName":"Tim","lastName":"Hardaway","username":"timmy","password":"password"},
{"customerID":2,"firstName":"test","lastName":"test","username":"test","password":"password"},
{"customerID":3,"firstName":"Barack","lastName":"Obama","username":"barryo","password":"password"},
{"customerID":4,"firstName":"Ezekiel","lastName":"Elliott","username":"zeke","password":"password"},
{"customerID":5,"firstName":"ben","lastName":"young","username":"benyoung1","password":"password"},
{"customerID":6,"firstName":"john","lastName":"bob","username":"johnbob","password":"password"},
{"customerID":7,"firstName":"travis","lastName":"john","username":"travisj","password":"password"},
{"customerID":8,"firstName":"ben","lastName":"bobby","username":"bbee","password":"password"},
{"customerID":9,"firstName":"john","lastName":"bob","username":"jbob","password":"password"}]
```

# Questions?