Project Plan
Commercial Laundry Dashboard

The Capstone Experience
Team Whirlpool

Mike Babiarz
Nico Vergara
Domenic Belcastro
Ryan Feldman
Yuqi Xiao

Department of Computer Science and Engineering
Michigan State University
Spring 2017
Functional Specifications

• Provide Whirlpool and their Trade Partner with a platform to maintain and monitor their appliances
• Displays alerts, errors and statuses of the machine
• Unifying appliance data into a single network
Design Specifications

• Filter Based on:
  Trade Partner > Location > Appliance
• Add Graphic Visualizations
• Create Multiple Dashboard Views
• Interactive Map Capability
### Screen Mockup: Web App (Data)

#### Commercial Laundry Dashboard

<table>
<thead>
<tr>
<th>Admin</th>
<th>Trade Partner</th>
<th>Location</th>
<th>Appliance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,465 Total Machines</td>
<td>1,125 Machines Online</td>
<td>340 Machines Offline</td>
</tr>
<tr>
<td>Home</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cedar Village</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **View Map**
Screen Mockup: Web App (Map)
Screen Mockup: Mobile App
Technical Specifications

- Two (2) Ubuntu LTS servers
  - One will host Backend Application/API
  - Second will host Postgres Database
- Frontend Application
- Backend Application/API
- Postgres Database
System Architecture

Call Whirlpool API to update information on Commercial Laundry Dashboard App

API returns JSON response that is processed by Commercial Laundry Dashboard App

Database returns data that is processed by the Commercial Laundry App

Backend App/API

API returns data for Mobile & Web App, which is displayed using D3.js

Mobile & Web App sends request to API

Commercial Laundry Dashboard reads/writes data to database

Mobile & Web App

= Google App Engine
System Components

• Hardware Platforms
  ▪ Google App Engine Server

• Software Platforms / Technologies
  ▪ Google App Engine
  ▪ Flask
  ▪ Cordova Ionic
  ▪ D3.js
Testing

• Test Driven Development (TDD) will be a fundamental process while developing applications.
• Features must be tested before merging into master and production.
• Python unit tests library will be used for testing backend application.
• Frontend application will be tested using Jasmine and Karma.
Risks

• Database Usage/API Calls
  ▪ Extensive amount contained in Whirlpool’s API
  ▪ Building our own API to filter only the necessary information

• Managing Dashboard Configurations
  ▪ User configuration model and configuration process isn’t finalized due to complexity and edge cases.
  ▪ Creating a basic configuration and expanding off that

• Converting the Design to Whirlpool’s Design
  ▪ Not sure when we will receive the frontend designs
  ▪ Send Whirlpool our mockups for constant feedback

• Push Notifications
  ▪ Not sure how Ionic handles push notifications
  ▪ Researching Ionic libraries and php solutions
Questions?