IOWA STATE UNIVERSITY

EE/CprE/SE 491

Automating Inventory Management: Routing Through Sensor Networks

Team: sdmay19-29
Adviser: Goce Trajcevski
Client: Jimmy Paul (Crafty, LLC)

Team Members

David Bis, Adam Hauge, Noah Bix, Hanna Moser, Samuel Guenette, Ben Gruman

Outline

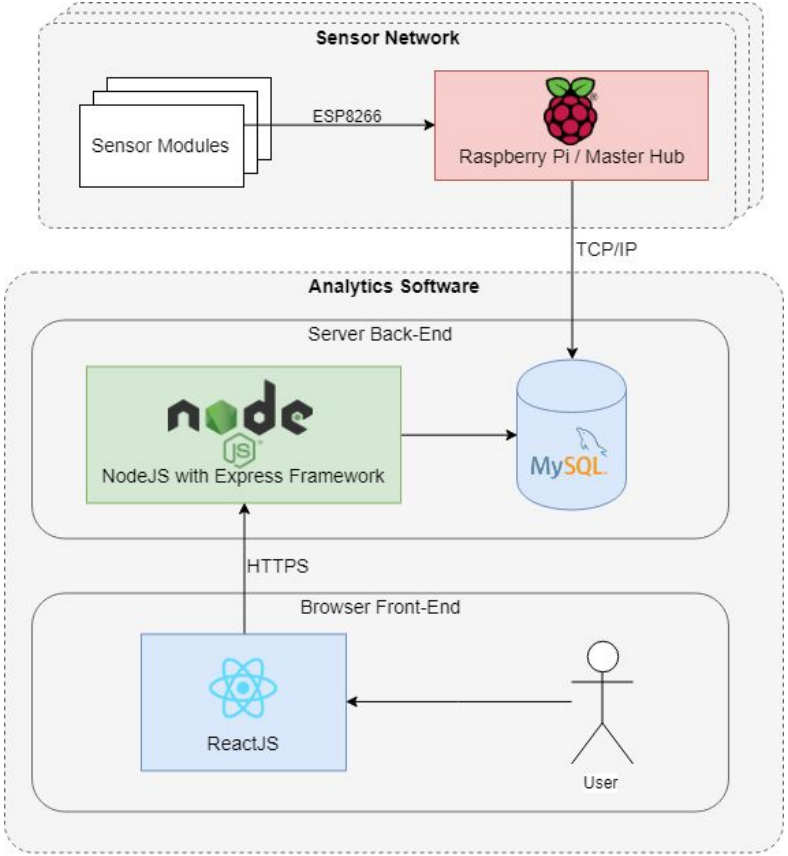
- Motivation and Solution
- Requirements
- Technical and Market Considerations
- Resource and Cost Estimations
- Design Decomposition
- Testing
- Current Status
- Future Plan

Motivation & Problem Statement

- **Crafty delivers food to office pantries**
  - Crafty employee stationed at each office
    - Keep track of inventory and reorder items
  - Trucks restock multiple offices based on individual orders
- **Issues**
  - Added expense for gas and labor
  - Human error
  - Time inefficiency
  - Route Inefficiency

Craft A Better Workplace

Block Diagram



Functional Requirements

Sensor Network

FR.1: Transmit product quantities individually

FR.2: Sensor arrays fit all stockrooms

Analytics Software

FR.3: Create optimal delivery routes

FR.4: Display & monitor the stockroom inventory data

Non-Functional Requirements

- **Scalability** - The sensor nodes should fit into multiple pantries
- **Data Integrity** - Sensors should be accurate and consistent over their lifetime
- **Availability** - Inventory should update at least once every weekday
- **Deployment** - New sensor devices should automatically detected by master
- **Usability** - User interface should be intuitive
- **Resilience** - Sensor network should be protected from typical motions of pantry inventory

Technical Considerations and Constraints

- **Master System** - Raspberry Pi vs. Arduino
- **Choice of Sensors** - Load cell, sonar, barcode scanner
- **Database** - MySQL vs. MongoDB

Market Survey

- **Impinj** - Automatic inventory management with RFID
- **Barcodes, Inc.** - Inventory management with barcode scanners
- **Route4Me** - Optimal route planning

Potential Risks & Mitigation

- **Sensor Degradation** - Sensors may not stay calibrated over time
  - Response Strategy - Calibrate the sensors on a regular basis
- **Communication** - Back-end cannot directly communicate with sensor networks
  - Response Strategy - Employ failsafe detection if sensor network does not communicate within expected time interval
- **Routing Inaccuracy** - Route time estimations are variable due to traffic conditions
  - Response Strategy - Use Google Maps API to account for traffic

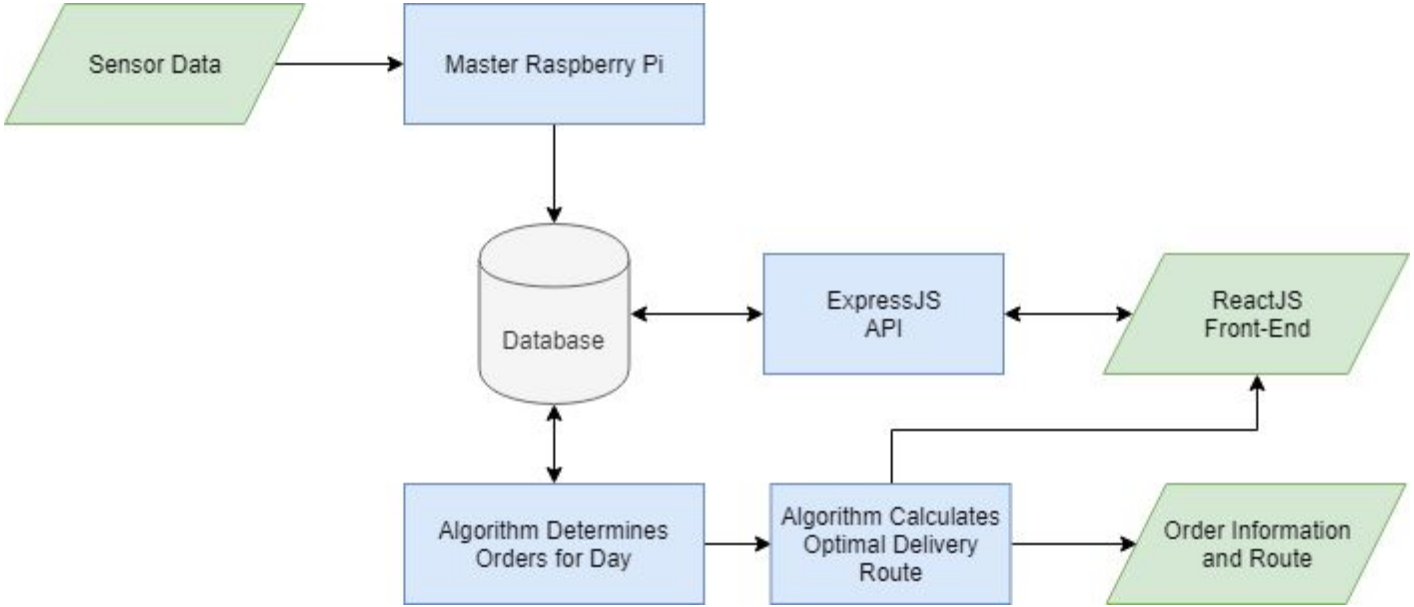
Resources and Cost Estimation

- **Sensor Network**
  - Master Microcontroller: $60.94
  - Sensor Array: $40.00
- **Analytics Software**
  - All used software is free of charge
- **Labor Resources**
  - Required for sensor network maintenance

Project Milestones & Schedule

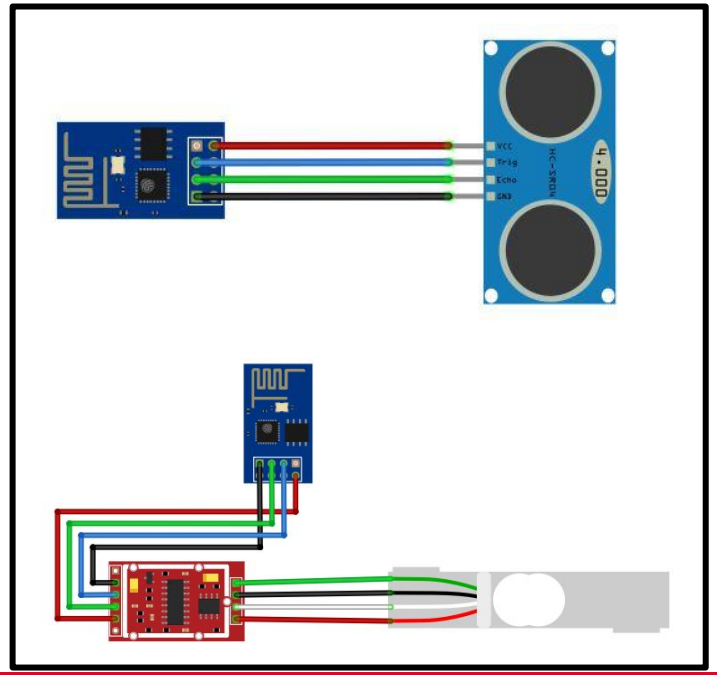
- **Proof-Of-Concept** (October 25th, 2018)
- **Minimum Viable Product** (March 15th, 2019)
- **Finalized Product** (April 15th, 2019)

Functional Decomposition



Detailed Design 1 - Sensor Network

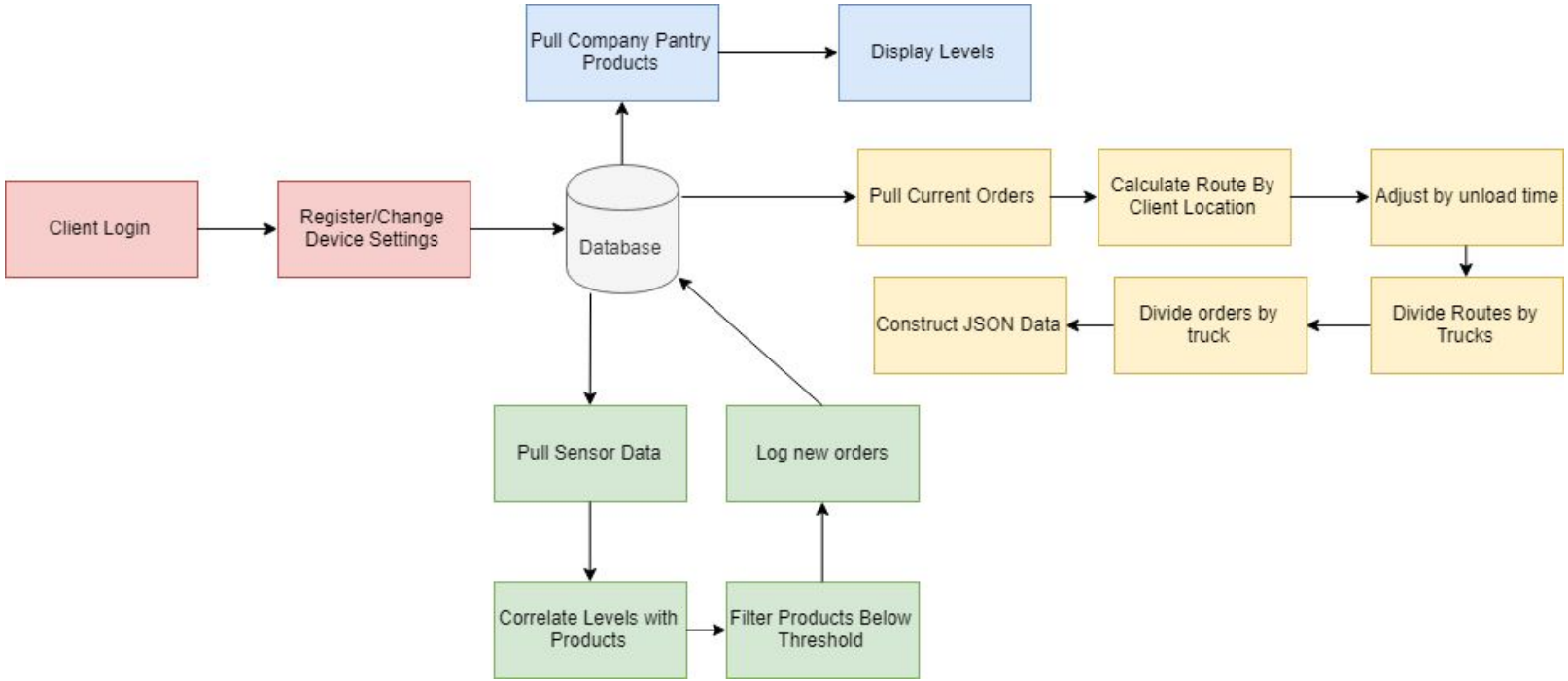
Sensor Module Components



Raspberry Pi

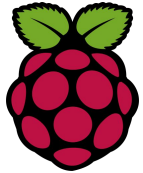
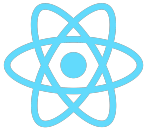
Database

Detailed Design 2 - Analytics Software



Hardware/Software/Tech Platforms

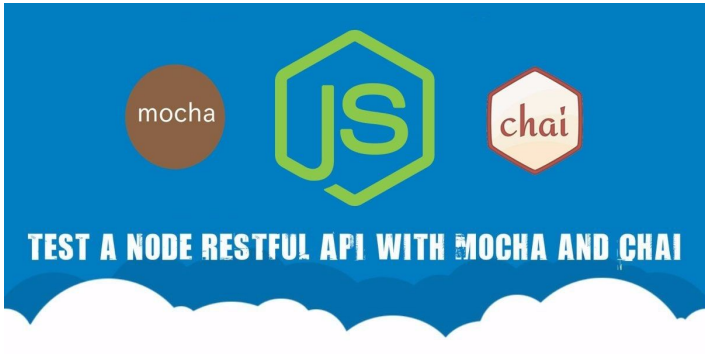
- **Web Component**
  - ReactJS
  - NodeJS Backend Processing
    - Sequelize API to Query Database
  - MySQL Database
  - Redhat 7 Server
- **Monitoring Device**
  - Raspberry Pi 3 (Master Component)
  - ESP8266 (Slave Component)
  - Python
- **Sensors**
  - Weight sensor
  - HC-SR04 Ultrasonic distance sensor
  - Barcode scanner



Functional Test Plan

- **Software Testing**
  - Mocha & Chai unit testing
    - Mock data & expected outputs
  - Real World Testing
- **Hardware Testing**
  - Python script testing
  - Verify established connections with sensor arrays
    - CSV Log
- **Integration Testing**

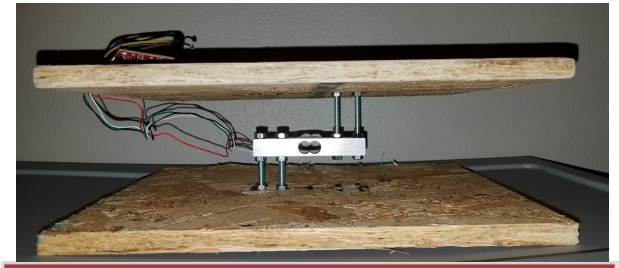


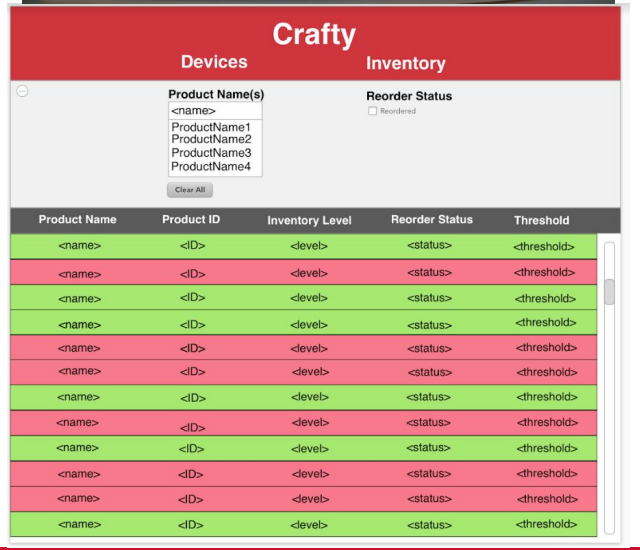
Non-Functional Test Plan

- **Scalability** - Connect *k* sensors to network and verify displayed on front-end
- **Data Integrity** - Track data on Raspberry Pi to compare to database value
- **Availability** - Reorder EOD and verify new order/delivery route next morning
- **Deployment** - Setup network in desired location and verify functionality
- **Usability** - Track automatic and manual tracking/reorder time and compare
- **Resilience** - Check accuracy in sensor reading every month

Prototype Implementations: Current Status

- Weight Sensor Implementation
- Barcode Scanner Implementation
- Master-Slave Network Setup
- Database Setup
- Back-End processing setup
- Front-End Screen Sketches





Current Project Status with Respect to Milestones

Task Responsibility/Individual Contributions

David Bis — Back-end Developer, Meeting Facilitator

Sam Guenette — Back-end Developer, Public Relations

Hanna Moser — Front-end Developer, Meeting Scribe

Adam Hauge — Computer Network Architect, Report Manager

Ben Gruman — Hardware Architect, Resource Acquisition

Noah Bix — Hardware Architect, Documentation Manager

Plan for Next Semester

- Incorporate multiple sensors
- Implement multiple sensor arrays
- Construct orders based on live data
- Configure route optimization
- Front-End Implementation
- Validation - Does it save time/money?

Questions