Automating Inventory Management

Routing through Sensor Networks

Design Document

Team Number: sdmay19-29
Client: Jimmy Paul
Adviser: Dr. Goce Trajcevski

Team Members:
David Bis — Meeting Facilitator, Back-end Developer
Hanna Moser — Meeting Scribe, Front-end Developer
Adam Hauge — Report Manager, Computer Network Architect
Ben Gruman — Resource Acquisition, Hardware Architect
Sam Guenette — Public Relations, Back-end Developer
Noah Bix — Documentation Manager, Hardware Architect

Team email: sdmay19-29@iastate.edu
Team Website: http://sdmay19-29.sd.ece.iastate.edu/

Revised: October 12, 2018 / Version 1.0

Table of Contents

List of Figures

List of Tables

(no tables at this time)

List of Definitions

Master-slave System: A type of software design where multiple smaller *slave* components carry out work, communicate, and are controlled by a single *master* component.

Sensory Device - Definition

1 Introductory Material

1.1 Acknowledgment

The Inventory Automation Team would like to thank the Iowa State University Department of Electrical and Computer Engineering for providing this team with a professional experience, quality resources, and consultation with experts. The team appreciates the willingness of the Electronics and Technology Group's (ETG) to help provide hardware and server components for the project. Special thanks also go to Iowa State's Dr. Goce Trajcevski for weekly consultation with regards to dealing with technical issues and moving forward throughout the development process. Appreciation also goes towards our client, Crafty Inc., for the given project. Special thanks goes to CTO and co-founder Jimmy Paul for making time to meet with the development team to gain more information and feedback throughout the project's development.

1.2 Problem and Project Statement

Crafty Inc. is a warehouse company that delivers food to office pantries. Their current infrastructure is based on having a single employee at each of their client offices handling shipment orders and physically monitoring when the company's pantry needs to reorder certain products. Having such a middleman is an unnecessary expense for both Crafty and their clients, and is prone to human error. Furthermore, warehouse truck routing becomes severely inefficient and expensive from restocking at individual offices based on separate orders.

Our solution consists of developing a microcontroller device that automatically monitors the amount of items in an office pantry and places orders to the Crafty warehouse when the office has reached its minimum threshold value for certain inventory items. We will then develop software that processes the current orders from all office orders to optimize shipment routes. This solution will save a significant amount of money and time for Crafty and their clients.

1.3 Operational Environment

The microcontroller used to monitor office shipment levels is expected to be stored in a dry pantry area with a reliable power supply and WiFi connection. The current device consists of a microcontroller-connected "Smart-Bin" that is expected to fit on shelves. Any software setup with the device will be done through a web application. The web application will be available for Crafty employees stationed at office locations for setting up the physical device and in the warehouse for overseeing and handling shipment orders.

1.4 Intended Users and Intended Uses

There are three main users of our sensor network. Each user will interact with the sensor network through use of the web application component. The following are brief overviews of the three users:

1. *Pantry Employee*: Actor based in a given company office in charge of handling shipments sent from the warehouse. This user is in charge of setting up the microcontroller device and registering items for the device to monitor.

2. *Warehouse Employee*: Actor based in the company warehouse in charge of overseeing shipment orders and filling trucks with the proper items. This user will go into the application to view orders and routes. This user is expected to possess moderate technical experience.

3. *Warehouse Truck Driver*: Actor in charge of the physical delivery of items between the Crafty warehouse and client office pantries. This user will employ the online application to view their assigned shipping route for the day.

1.5 Assumptions and Limitations

Assumptions:

- Pantry sensor network has access to a reliable power source
- Pantry sensor network can connect to internet
- Pantry sensor network is not at risk of water damage
- Pantry Employee properly sets up sensor network and associates each monitoring device with the corresponding product being stored through the online application

Limitations:

- Accuracy of automatic orders is completely reliant on the accuracy of the sensors used in the sensor network
- Sensor network monitors product availability in bulk, not individually
- Sensor network can only make correct orders if the Crafty user properly establishes what type of products are being monitored

1.6 Expected End Product and Other Deliverables

The final product delivery will be split into a proof-of-concept, minimum viable product, and a final product deliverable. The proof-of-concept and minimum viable product (MVP) will be delivered by the end of the fall 2018 semester. The finalized product will be delivered at the end of the Spring 2019 semester. An overall design manual of the product and its architecture will also be provided for any future development.

1. Proof-Of-Concept (October 25th, 2018)
    - The Proof-of-Concept prototype will prove the products capabilities and potential to both monitor physical storage units and communicate that information to monitor and analyze. Users will be able to store a product in a single “Smart-Bin” which will send that information to an SQL Database. Users can log into a web application to register the device and monitor storage levels of the product in the

Smart-Bin

2. Minimum Viable Product (March 15th, 2018):
    - The Minimum Viable Product will be deployed for beta-testing. Multiple cheaper Smart-Bin devices will communicate with a single WiFi component that sends information to the database. The web application component will monitor information and build an appropriate and optimized shipment orders.

3. Finalized Product (April 15th)
    - The Finalized Product is ready to integrate for use with the client's existing infrastructure. The smart-bin device is expected accurately monitor multiple products. The user interface will also be provide easy setup, handling, monitoring, and registering of smart bin devices. The database is connected to a Cloud Management System to analyze data and learn to better optimize shipping routes and customer orders.

2 Specifications and Analysis

2.1 Proposed Design

The solution design will be broken into two primary components: the sensor network and the analytics software. The sensor network will help satisfy any functional and non-functional requirements regarding inventory monitoring. The analytics will satisfy any requirement regarding route planning and optimization using the data from the sensor network.

The sensor network will consist of a master-slave system with the master component being a Raspberry Pi and the slave components being individual sensory devices. A *sensory device* will employ a variety of sensors in order to to monitor inventory levels of a single product. The client is suggested to use multiple sensory devices to monitor different products stored in their pantry. Once the Pi receives the data from the sensory devices, it will send that data directly to the MySQL database on the back-end. Multiple sensors were selected to minimize potential error in the inventory calculations. Two sensors that the team intends to implement are weight sensors and barcode scanners. These sensory devices will communicate with the Raspberry Pi with an ESP8266 wireless chip.

The analytics software will be made with a three-tier web application. The tiers include a ReactJS front-end, NodeJS back-end with an Express framework, and a MySQL database. The front-end will be used for data visualization and sensor network configuration. The back-end will be used to act as both an API and a data processing unit.

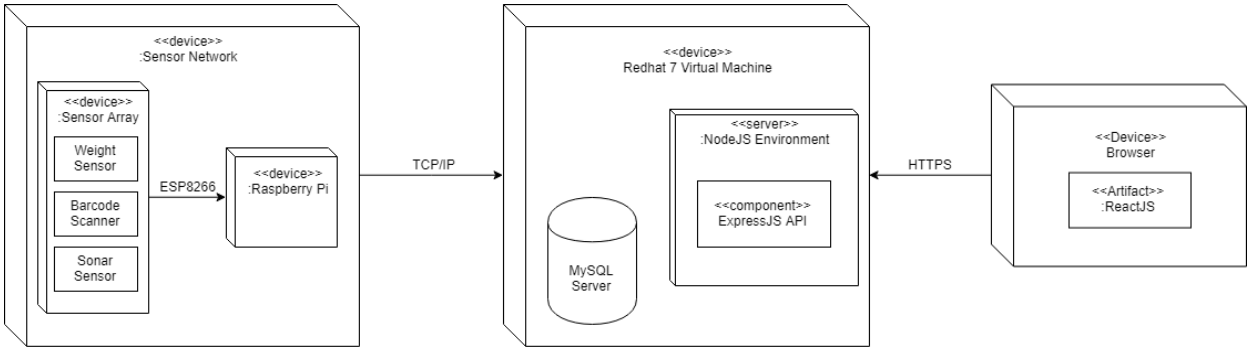

Figure 1: Deployment Diagram of Solution Components

2.2 Design Analysis

There are two primary component systems for continued development on the project. The analytics system is completely dependent on the continuous stream of accurate data sent from the sensor network. This could become a problem if there is a loss of connection anywhere during the transmission process. The analytics component is also completely reliant on the sensors and hardware being implemented. Any issues in in sensor network directly affects the analytics component's ability to monitor and place orders, as well as being able to calculate an optimal route for delivery.

An alternative considered for the mastercomputer for the sensor network was an Arduino microcontroller, as opposed to the Raspberry Pi. While both options provide sufficient features to satisfy the requirements, the Raspberry Pi was selected due to its ease of use for development purposes. One advantage that the Arduino has over the Raspberry Pi is its on-board Analog-to-Digital Converter (ADC), which the Raspberry Pi lacks. Since the project relies on gathering data from a variety of sensors, an ADC is crucial for making sense of the data. However, the Raspberry Pi can be equipped with a separate chip for the ADC, a cheap solution that provides results comparable to those of an Arduino. Another important difference is the Raspberry Pi's on-board operating system (OS). A non-functional requirement for the project was scalability. Since Crafty services multiple pantries, it is important that the sensor network is modularly configurable. Since the OS is on-board the Raspberry Pi, it enables the user to easily replace and alter sensor array configurations. The OS enables changes at run-time through a program written for the device. One last difference is that the Raspberry Pi has an on-board wireless capability, whereas the Arduino does not. Since the master computer must make calls to upload data to the database, the feature is convenient. Likewise, it could be solved on the Arduino with a chip similar to that of the ADC.

All data is sent to a central database only accessed by the web component. The back-end component monitors database information and places orders based on thresholds and and prior specification provided by a Crafty user. The user interacts with either a registration form to register sensory devices and input the product being monitored along with conditions for reordering. The front-end is built with ReactJS, which was chosen due to the team's familiarity

with the framework over other frameworks such as Angular. The order sent will be a set of JSON which is sent to the Crafty's existing architecture to process. The data being sent is not only the order or products, but also an ideal shipping route based on what multiple Crafty clients ordered.

In terms of the physical sensors, we originally chose Radio Frequency Identification (RFID) sensors. These sensors would be able to detect when an item passes through the pantry doorway without having to physically scan the item. This sensor can also detect multiple items at once and has a large range of detection. This idea was eventually replaced with a barcode scanner due to RFID being too expensive, since the RFID detector is quite costly, and each product box will need to have an RFID chip attached, which causes the cost to grow with scale. The downside of a barcode scanner is that the user will have to physically scan each item. Also, as everyone who has been to the grocery store would know, these scanners occasionally take multiple attempts to identify the barcode. These factors will cause the barcode scanner to take a significantly larger amount of time and labor. However, since our pantries are typically a small size, the amount of time to scan each item should be negligible.

3 Testing and Implementation

3.1 Interface Specifications

Testing will act as a major component in building our sensor network. We will consistently test our sensor network to ensure that all of our sensors record an accurate count of product by comparing the calculated total sent to the database with the true value. Testing will occur with a large variety of products at various quantities to ensure our measurement algorithms return accurate data. We will be sure to compare all data stored in the database with the measured data to check that the information given to our users is as accurate as possible.

3.2 Hardware and Software

The sensors used to measure inventory will be tested as network hardware components. The barcode scanner will be keeping track of the quantity of boxes in the pantry. When the quantity of each product is less than the desired amount of inventory, an order will be placed at the Crafty warehouse. Software testing will be applied to ensure a working initial connection for communication with the Raspberry Pi. However, the majority of testing for this item is to find the most efficient way to use this barcode scanner. Whether it be scanning items as they enter and exit pantry, or scanning all items at the end of the work day.

The weight scale is our second sensor that requires testing. Multiple weight scales will be placed in the client's pantry to keep track of the amount of inventory of each item. When the weight falls below a preset threshold, an order will be placed to restock that product. Sufficient calibration and adjustments for different products will have to be integrated into the scale. Once

one scale is fully functional, multiple scales will then have to be tested and integrated as well. All further testing for the scale monitor will be tested using the web application, verifying the proper information is being stored on the MySQL database.

Postman, a GUI software used to simulate REST API calls, will be used to test the API. Postman provides various features that help automate the test process, such as preparing test suites for calling the different endpoints, as well as feeding test data into the system.

3.3 Process

All sensors are being tested by comparing the quantity measured with the true quantity of different products.. This will ensure that the combination of sensors being used are sending accurate data visible to our users.

The analytics software will face unit and user testing to assure accuracy. This involves the use of mock data and simulations. The application will also be tested by actual users in order to gather feedback and observe areas to refine and improve.

The primary features will be implemented independently of the other features. Testing features that require the products of other features will be supplied with dummy data to simulate the entire functioning system in order to isolate specific features. This will be beneficial when testing the analytics software, which as mentioned in section 3.2, requires inventory data to be sent from the sensor network in order to be successful. Each feature will have an automated testing suite associated with it, so that it can be tested as an isolated feature, and retested after itself and other features have been implemented into the system in order to prevent regression errors.

3.4 Results

1. Sensor Monitoring Testing
   (Component has not yet been tested)
2. Order Creation Testing
   (Component has not yet been tested)
3. Route Optimization Tests
   (Component has not yet been tested)

4 Closing Material

4.1 Conclusion

At this point in the project, the team has completed gathering requirements by learning the needs and expectations of our client through various team meetings and communicating with Crafty. Furthermore, the team has been making great progress working together to design a

solution to our given problem. The team is in the beginning stages of development and is prototyping a solution. Once all required materials are acquired, we will be able to demonstrate a solution to the client as quickly as possible in order to improve the product. The team has split into two groups to work on software and hardware for this project. The software group will focus on creating and organizing the database and placing online orders from the Crafty warehouse. The hardware team will work with sensors and microcontrollers to create a sensor network that gathers all inventory data.

4.2 References

Intellectual Reference

1. https://www.impinj.com/solutions/healthcare/inventory-management/
2. http://wind.cs.purdue.edu/doc/adhoc.html

Pricing Reference Websites:

1. https://www.digikey.com/product-detail/en/SEN-13329/1568-1852-ND/7393715/?itemSeq=272691527
2. https://www.digikey.com/product-detail/en/SEN-13879/1568-1436-ND/6202732/?itemSeq=272691544
3. https://www.adafruit.com/product/1083

4.3 Appendices

Raspberry Pi Model 3b Datasheet
https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_2p0.pdf