EE/CprE/SE 491 Weekly Report 1

9/15/19 - 9/30/19

Group Number sdmay20-23

Project Title: Multi-Context Shopping Optimization

Client & Advisor: Goce Trajcevski, Ashfaq Khokhar

Team Members/Role: Max Garton, Arnoldo Montoya-Gamez, Ethan Shoemaker, Karla Montoya, Jesrik Gomez, Nate Wernimont (team member titles TBD)

Weekly Summary

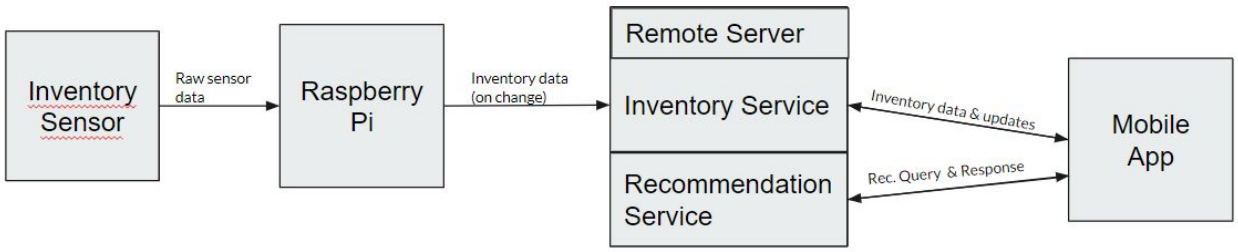
This week, the team researched the options for development tools, cloud solutions, sensor hardware, microcontrollers, and recommendation algorithms. We created a high level architecture diagram of the project, and began researching the different technologies and tools that can be used to create each component. After these decisions, the goal is to draft a design of each component and how they are connected in the overall project.

Past Week Accomplishments

Accomplishments during the week of 9/15-9/22:

- Obtained the initial project requirements from our client (entire team)
- Discussed and developed the high-level architecture of the project (Max & Arnoldo discussed, then Max created the diagram) attached below
- Set up team communication channels on slack (Arnoldo)

High Level Architecture



Accomplishments from this week (9/23 - 9/30):

- Researched initial hardware, algorithms, and technologies we will need to implement project deliverables (entire team - see individual contributions below)
- Updated the team website with team member bios (entire team)

Pending Issues

- Max Garton: N/A
- Arnoldo Montoya-Gamez: N/A
- Ethan Shoemaker: N/A
- Karla Montoya: N/A
- Jesrik Gomez: Our cloud computing platform will depend on the **FREE SERVICES** that satisfy our requirements.
- Nate Wernimont: N/A

Individual Contributions

Arnoldo:

- Brainstormed potential matching algorithms, and ways we can/could store our data.
    - Figured out we will need to process each nearby store's contents of our items (milk, butter, etc).
    - Then we will need to figure out a way to categorize each into different sections, meaning, if milk is one of our tracked items, we will categorize milk into skim, whole, and 2%. Then our users will have to specify what kind they want to keep track of.
    - Once we know what our user wants, and the stores that are nearby, if triggered, we'll be able to match items, which we will write our own algorithm for.
    - That will then send a notification to our user, suggesting they should buy some items, knowing that they are not home, there is a store nearby, and that store has the items the users need.

Ethan:

- Brainstormed possible sensors we can use for gathering data about the inventory a user has
    - Weight/pressure sensor - works for a variety of different products (my favored option)
    - IR/SONAR/Color sensor - works well for liquid containers but is difficult to attach to the containers in a non-intrusive way
    - Human Interface Device - the user has a panel or device that has either buttons or touch screen they can use to log the items they've acquired/used (this requires the user to log the data and makes the system less automated)
    - Wireless "repeater" - sensor enhancement that can transform any electrical signal (digital or analog) into a radio frequency that the receiver then tunes to
- Came up with some more ambitious ways to use the weight sensors to automatically register changes to the contents of a fridge possibly using AI or a database to predict which items have changed

Jesrik:

- Looked into different tools and technologies that will satisfy our initial project requirements for the web application, the mobile application, data storage and analytics, automation and notification services, scalability and modularization, and cloud computing platforms to host our services.
    - Web: NodeJS, PHP
    - Mobile: Android
    - Database:
        - Relational: mySQL, Oracle
        - Data Warehouse & Analytics: Microsoft Azure, IBM Cloud, AWS
    - Automation & Notifications: Jenkins, Cron
    - Modularization: Docker, Kubernetes

    - Cloud Computing: Microsoft Azure, IBM Cloud, Amazon Web Services

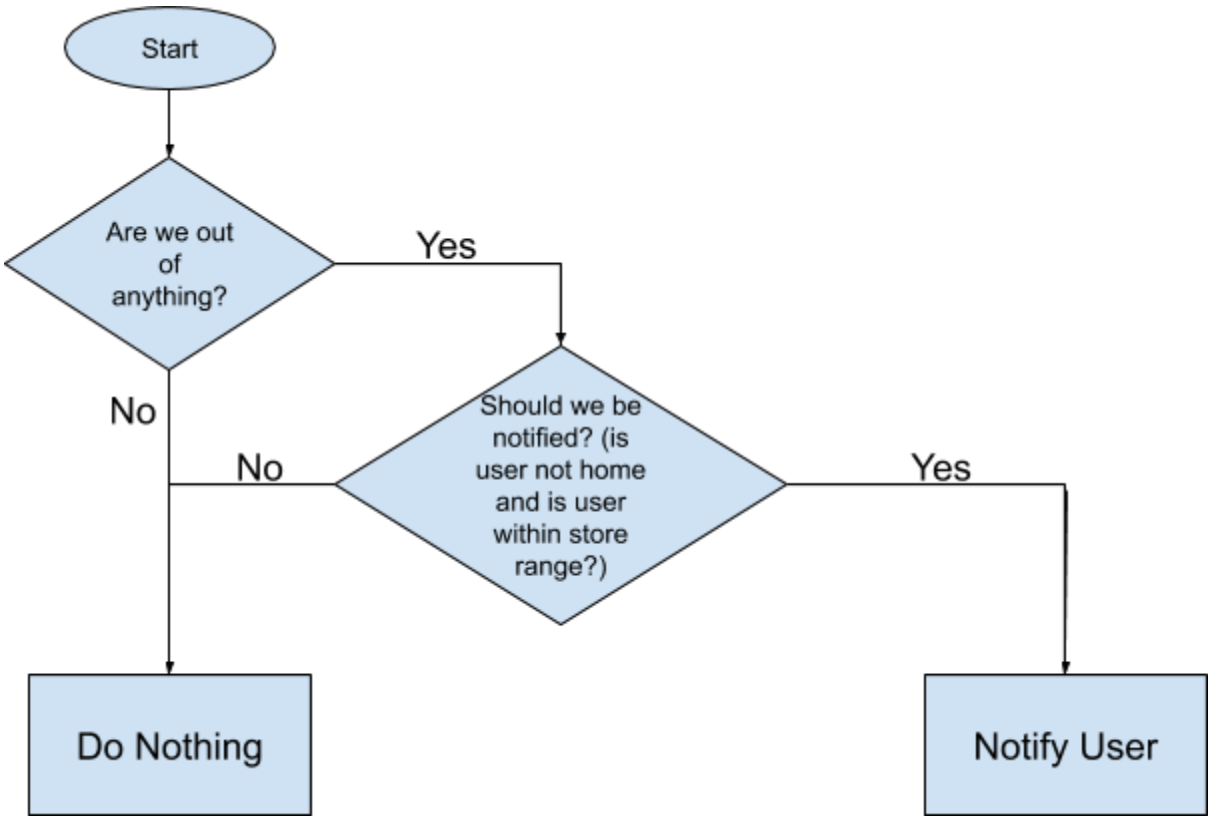
Karla:

- **Price Matching:**
  For the price matching, we have different options. Since the project is mostly about proving a concept, we can start by using the walmart Labs Product Lookup API, which can be found in the following link:
    - https://developer.walmartlabs.com/docs

- **Store Matching:**
  Another thing I researched this week was how we would find stores in range. In order to do it, we can actually use a tutorial created by the Google maps platform, which is found in the following link:
    - https://developers.google.com/maps/solutions/store-locator/simple-store-locator

- Also, Because we are developing using agile development, we will attempt to use these APIs, and if they don't work, we will either find others or create our own.

- The following will likely be the basic logic flowchart for decision making:



Max:

- Researched options for microcontrollers

    - Raspberry Pi (Full size R. pi 4 and the smaller R. pi zero W)
        - Pros: allows software flexibility, wifi built in, easy deployment of software onto the board, well-documented
        - Cons: higher power consumption, may be too flexible for our needs (overkill)
    - Arduino (Arduino Nano IOT)
        - Pros: cheap, simple, well-documented
        - Cons: requires C programs, may be tied to Arduino's IOT platform
    - Third party microcontrollers
        - Pros: cheap, flexible (many Linux options)
        - Cons: not as well documented or supported
- Developed a rough strategy for hardware prototyping
    - Use larger boards with pin headers to draft circuit/interface designs
    - After landing on solid prototype(s), implement them on the smaller, lower power boards (requires soldering, which is why we need to be confident in the design).
    - Once a solid, small prototype has been made, develop housing/case options for it
- Created high level architecture diagram (included above)
- Added deadlines and deliverables as issues in our Gitlab

Nate:

- Examined approaches for routing the user between various locations
    - Potential to use Google Directions API in order to find the actual driving duration between potential locations
    - MySQL supports spatial types and spatial queries
    - Traveling Salesperson problem
        - Using the agile approach, our algorithms can initially use the exact TSP algorithm in order to give the best results
        - As we start to handle more data, we can move to an approximate TSP algorithm in order to solve the problem in polynomial time, at the expense of potentially giving the users somewhat worse results
    - Routing users between locations
        - At first, we can use linear distances between stores and user location.
        - Then, we can move to computing the driving distances between various locations. In order to do this, it requires n! calls to Google Directions API, though. It would be beneficial to store the distances between locations in our table to avoid making these calls every time
    - In order to narrow down which locations to actually travel to (to satisfy all purchase constraints), we can make an integer linear programming algorithm.
        - Later on, we can relax the constraints in order to convert the algorithm from exponential to polynomial.

Name	Individual Contributions (Quick list of contributions. This should be short.)	Hours This Week	Cumulative Hours

Arnoldo	Created Lightning Talk Video, Researched Matching Algorithms.	7	10
Ethan	Researched on IoT strategies/tech	6	8
Jesrik	Researched development tools & technologies	5	9
Karla	Researched Matching Algorithms, Researched ways to store data	7	8
Max	Microcontroller research, high level arch. diagram, updated gitlab board	6	10
Nate	Researched Routing algorithms	8	11

Plans for Next Week

- Max Garton: Prototype a solution using microcontrollers and sensors, using data from some electronic sensor and sending data to a remote server. Add rough prototype to the design document and explain why decisions were made. Order microcontrollers for the team.
- Arnoldo Montoya-Gamez: Will be researching APIs that can be used for our Android App, and will find ways to find users locations.
- Ethan Shoemaker: Determine if refrigerators block wireless communication to the outside. Compile a spreadsheet/comparison of different sensor options and benefits to discuss with team; work towards making a decision on which sensor (s) to use. Ideate strategies for collecting data from sensors and the heuristic for storing it in the database (depends on which storage technique we're using). Write a 5 sentence elevator pitch of our project for a layman.
- Karla Montoya: Will create basic notification that can be sent to our mobile app, and will find ways to trigger it.
- Jesrik Gomez: Get feedback from the other team members about hardware requirements so we can start deciding which free cloud computing platform will work best.
- Nate Wernimont: Look at the SQL table design for our data.

Summary of Weekly Advisor Meeting
During our meeting with Goce on 9/26, we discussed some of the research that we've done to help decide which technologies to use for the project. Goce stressed the importance of being able to justify every decision and exploring all options. We need to be able to show that we made informed decisions rather than simply choosing something. We also clarified how we can best describe our project in layman's terms in a few sentences. The goal moving forward is to

work on an initial draft of our design document, including why and how every decision was made.