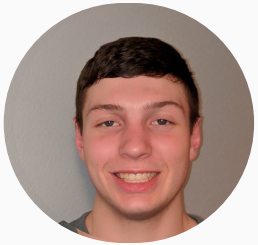
Multi-Context Shopping Optimization

sdmay20-23: Max Garton, Ethan Shoemaker, Arnoldo Montoya-Gamez, Karla Montoya, Nate Wernimont, Jesrik Gomez

Advisor: Goce Trajcevski

April 29th, 2020

Team Introduction



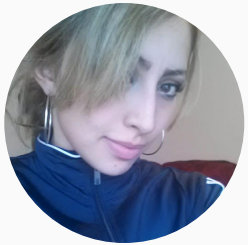
Max Garton
Computer Engineering
Meeting Lead

Ethan Shoemaker
Computer Engineering
Hardware Engineer

Jesrik Gomez
Software Engineering
Deployment Engineer

Nate Wernimont
Software Engineering
Backend Engineer

Arnoldo Montoya-Gamez
Computer Engineering
Frontend Engineer



Karla Montoya
Software Engineering
UI Engineer

Project Vision

Saving shoppers time and money by offering shopping recommendations

Market Survey

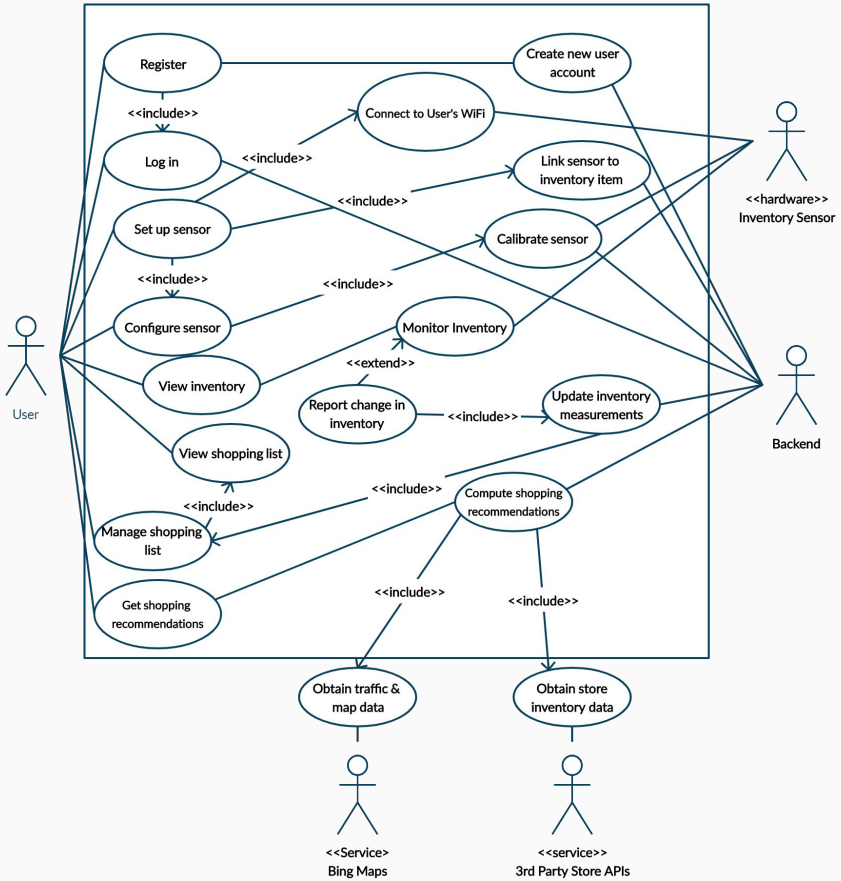
Existing Offerings

- Bottomless Coffee
  - Limited to coffee
  - One source of product
  - Online delivery
- Online shopping platforms
  - Multiple stores
  - Online and pickup
  - No automatic inventory detection

Our Project

- Generalized to most items or goods
- Any source of product (with online listings)
- In-store purchases using online information
- Automatic inventory detection

Use Case Diagram



Functional Requirements

- Access to interact with the backend system and services
- Automatic inventory measurement
- Automatic shopping list curation
- Shopping recommendations
- User notifications of sensor status changes

Non-Functional Requirements

- Performance
- Scalability
- Longevity of sensor devices

Ideal Design Considerations:

- User privacy
- Security

Technical Considerations

- Inventory sensor device: Arduino vs Raspberry Pi
- Cloud resources: Amazon Web Services vs Microsoft Azure vs IBM Bluemix
- Frontend application: Mobile application vs web application, iOS vs Android

Potential Risks & Mitigation

Security and Privacy

System Availability

Availability of Contextual Data

Scalability

Project Schedule

Task	January	February	March	April
Sensor Setup Process	MG			
Data Collection Process	ES			
Routing Service	JG			
Recommendation Service	NW			
Inventory and Shopping List View	AM			
Shopping Recommendation Screen	KM			
Sensor Enrollment Integration				
User Login Integration				
Low Inventory Integration				
Store Viability Integration				
Reiterate and Add Features				

Inventory Sensor Device

Backend Services

Mobile App

Integration Tasks

Project Kickoff Sep 16

Finalized Design Dec 6

MVP Mar 15

Final Product Apr 15

2019 | Sep | Oct | Nov | Dec | 2020 | Feb | Mar | Apr | 2020

Required Resources & Estimated Cost

Hardware: ~$250

- 2x Raspberry Pi 3+ A ($30 ea)
- 2x Raspberry Pi Zero W ($30 ea)
- 10x Weight sensors ($6 ea)
- Circuitry & enclosures ($70 ea)

Cloud Resources: $0

Total Cost: ~$250

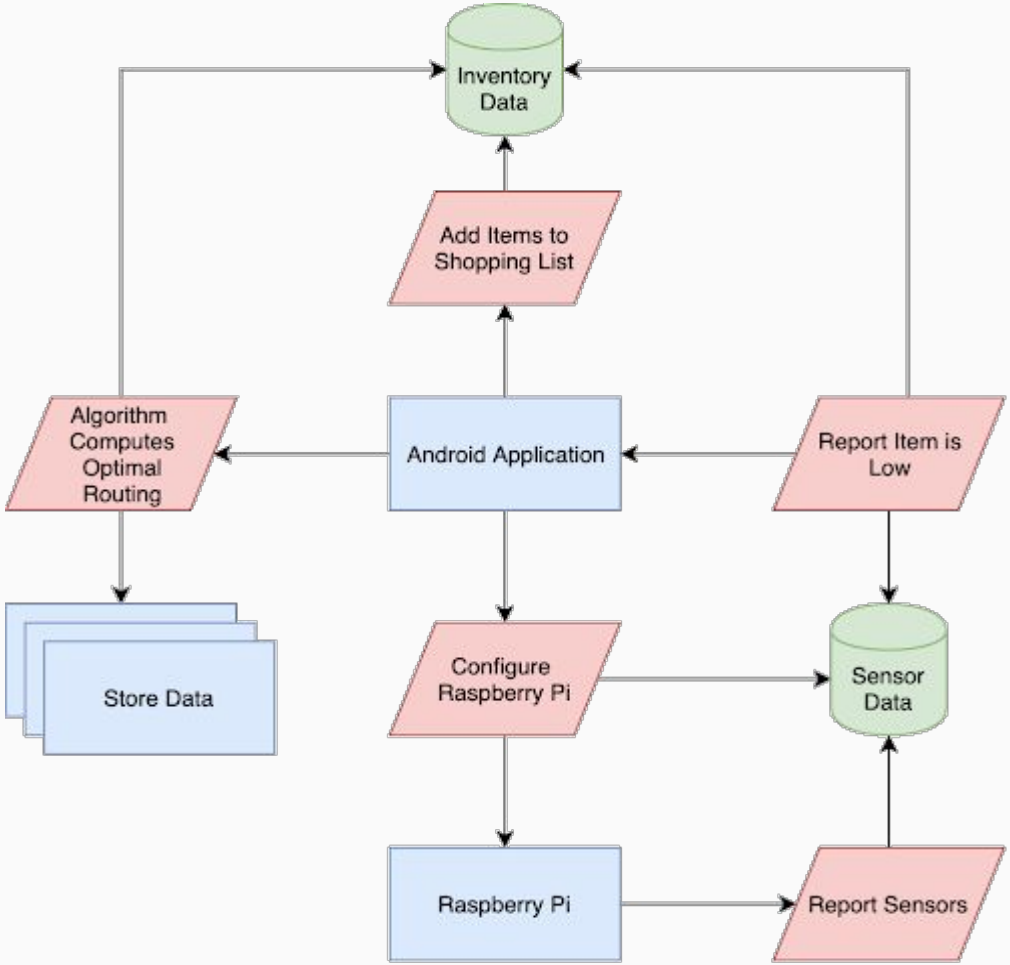
Functional Decomposition

Parallelograms are activities

Rectangles are objects

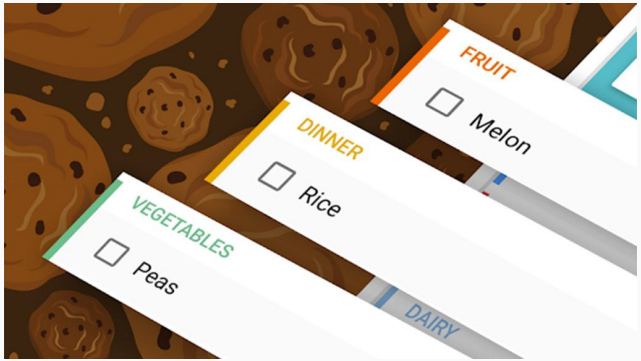
Containers are databases

Not pictured: users service



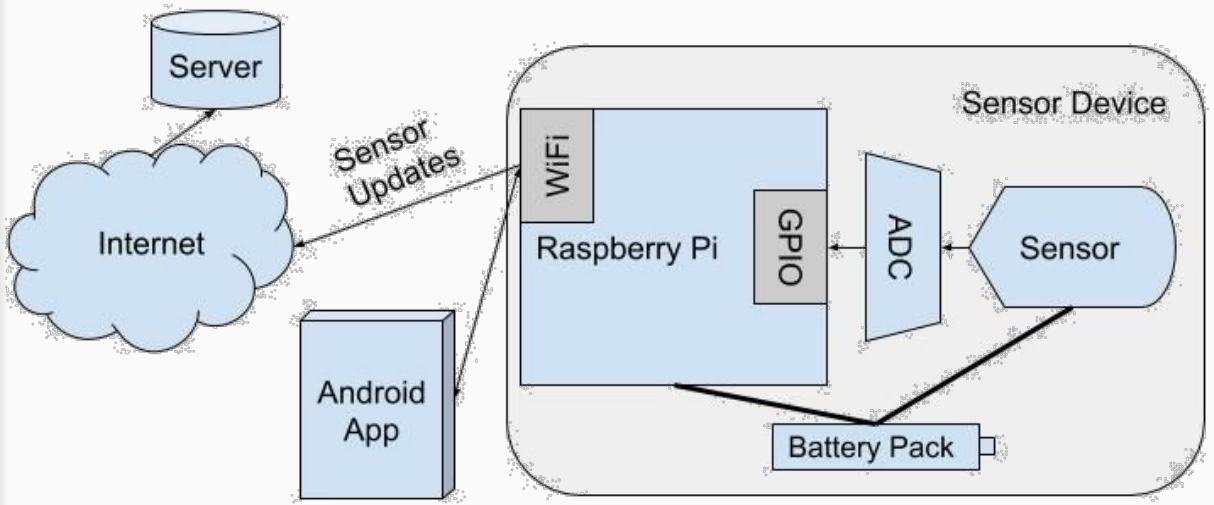
Major Features

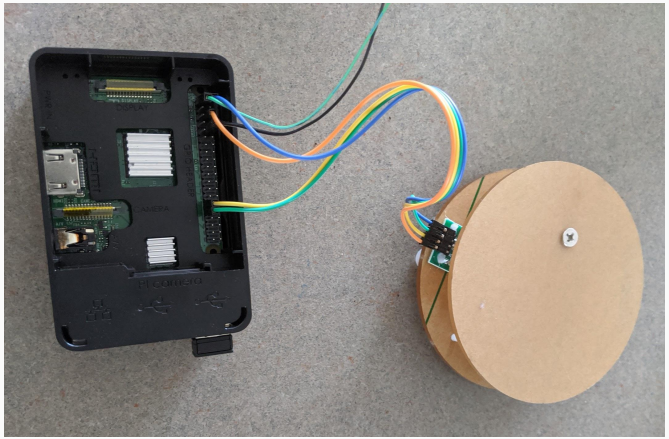
- Automatic inventory tracking
- Auto-curated shopping list
- Shopping optimization via cost and time routing



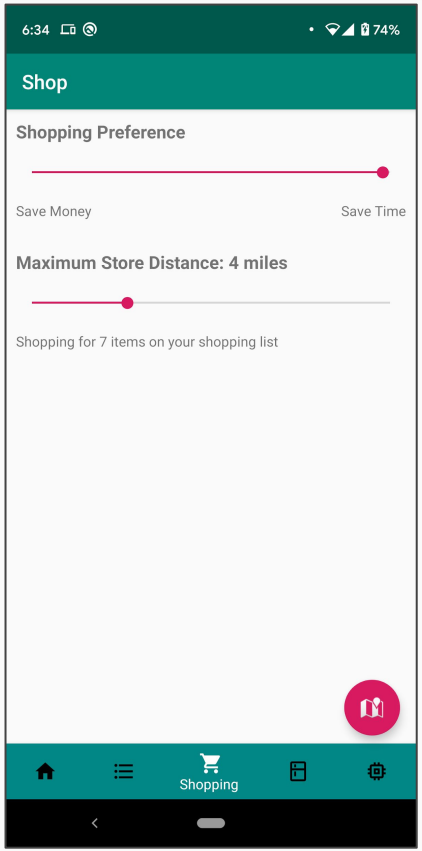
Inventory Sensor Device Design

- Measures inventory by weight
  - Checks at regular intervals
  - Sends updates to remote inventory service
  - Calibrated empty, low and full weights during setup
- Configured to connect to user's WiFi network

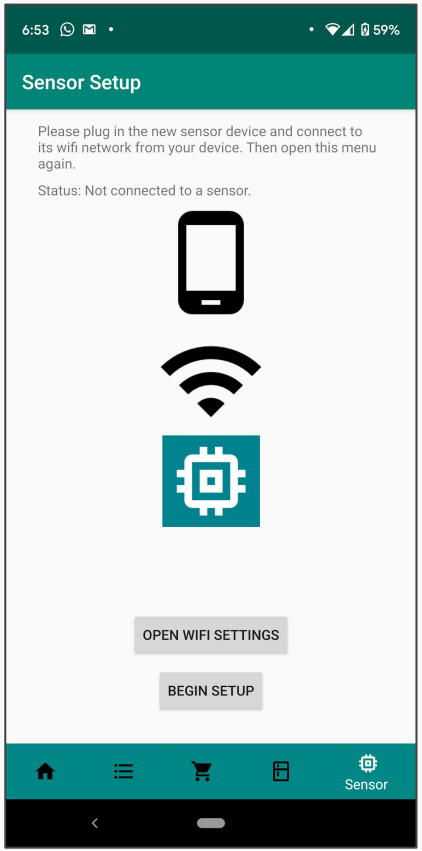




Android Application Design

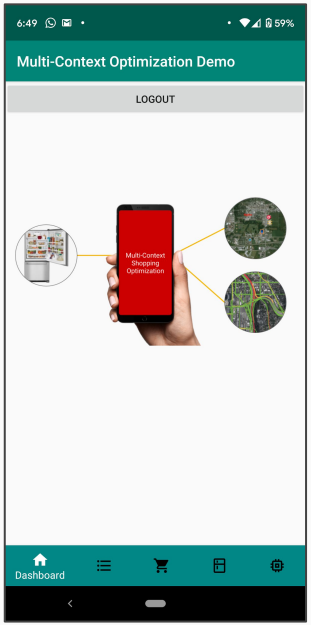


Shop

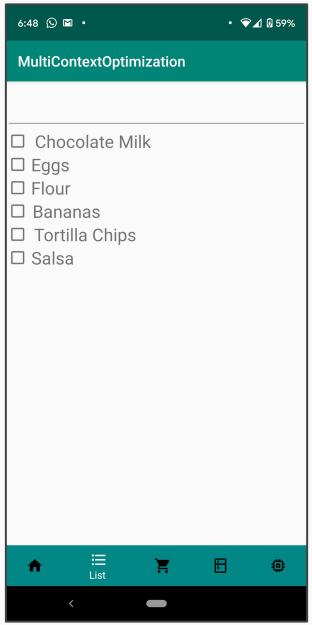


Setup

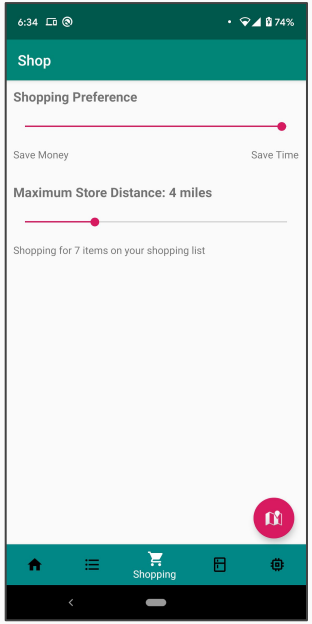
User Interface Design



Navigation

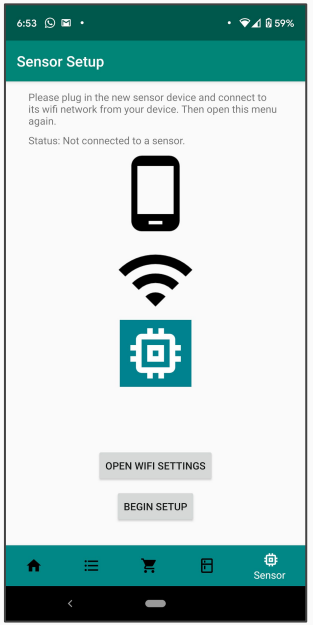


Shopping List



Shop

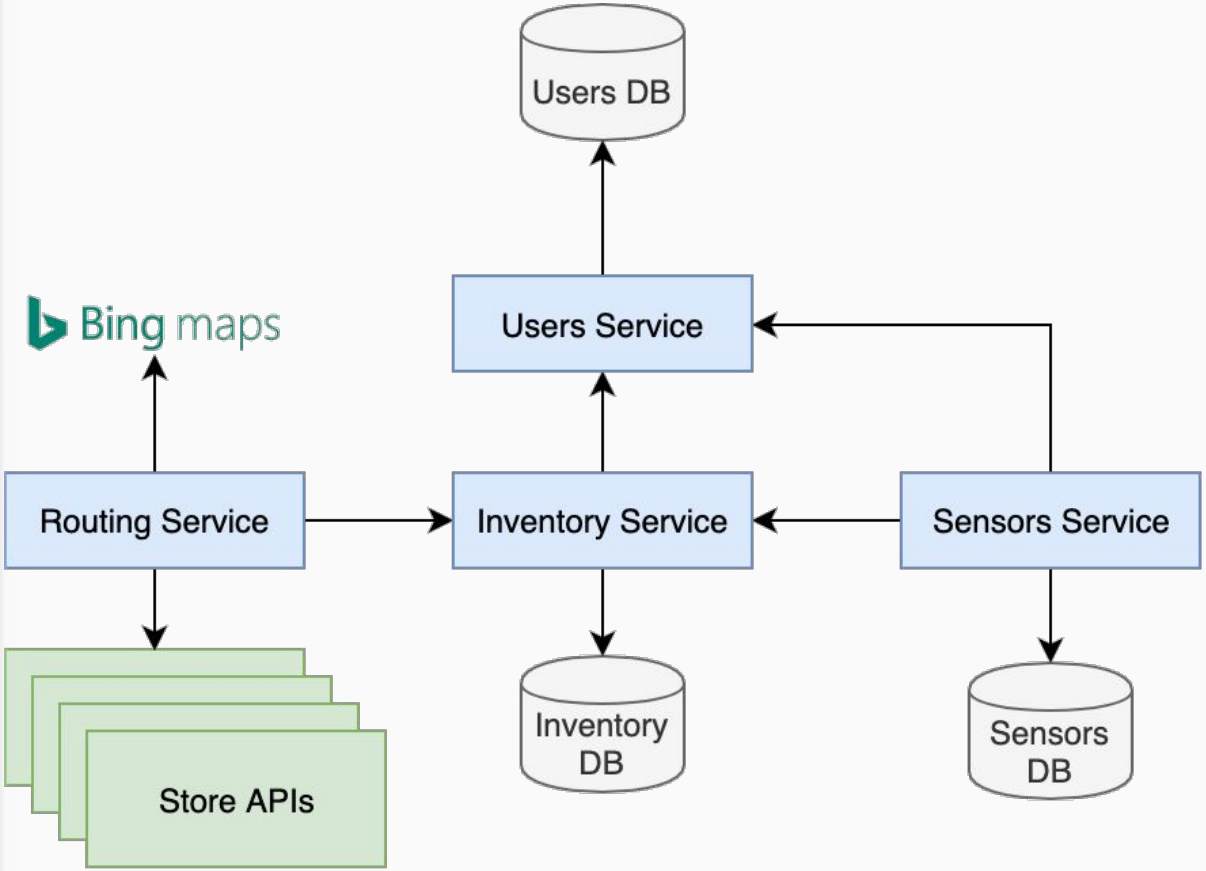
Sensor Status



Setup

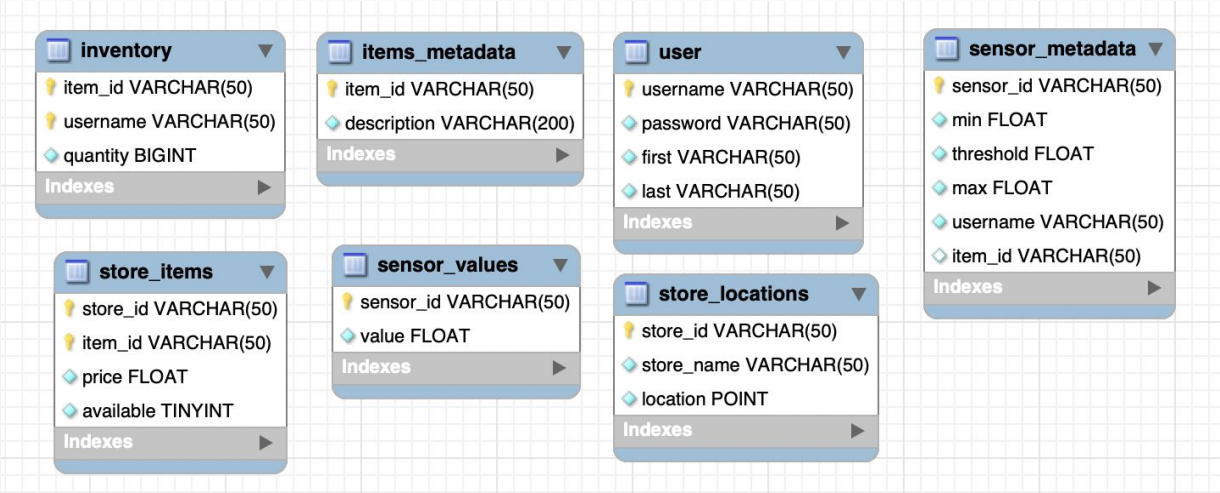
Backend Services Design

- Users Service
  - Handle user account details
- Sensors Service
  - Receive & process inventory measurements
- Inventory Service
  - Track inventory
- Routing Service
  - Route user to stores given shopping needs



Data Design

- Simple tabular structure
- User has:
  - Inventory
    - Quantity needed for each item
  - Sensors
    - Min and max weight
    - Current value
- Stores are unrelated entities used in recommendation computation
  - Items in stock
  - Price of each item
  - Location



System Design and Technical Considerations - Software and Hardware Platforms

Web Server

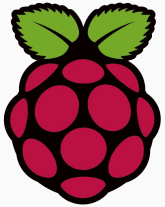
- Microsoft Azure over AWS and IBM
- Go
- MySQL

Mobile Platform over Web Platform

- Android over iOS
- Java

Inventory Sensor Platform

- Raspberry Pi over Arduino
- Python



Functional Testing

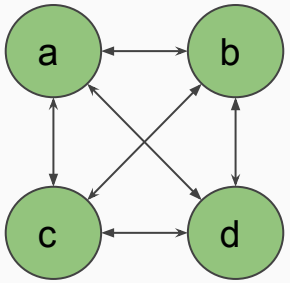
- Create an account and sign in
- Install and set up an inventory sensor device
- View inventory status
- Viewing and modifying the shopping list
- Automatic inventory tracking
- Receiving shopping recommendations

Functional Testing

- Install and set up an inventory sensor device
  a. Power on a sensor for the first time
  b. Open the sensor setup screen
  c. Follow setup instructions on screen
- Acceptance criteria:
  a. The sensor device is connected to the user's wifi network
  b. The inventory service relates the new sensor to the user

Integration Testing

- Install and set up an inventory sensor device
  a. Sensor device registers with sensor service and has an assigned sensor ID
  b. Sensor service associates the sensor with the user



Testing Goals

- Non-Functional Testing
  - Scalability
  - Performance
  - Security
  - Usability

Engineering Standards and Design Practices

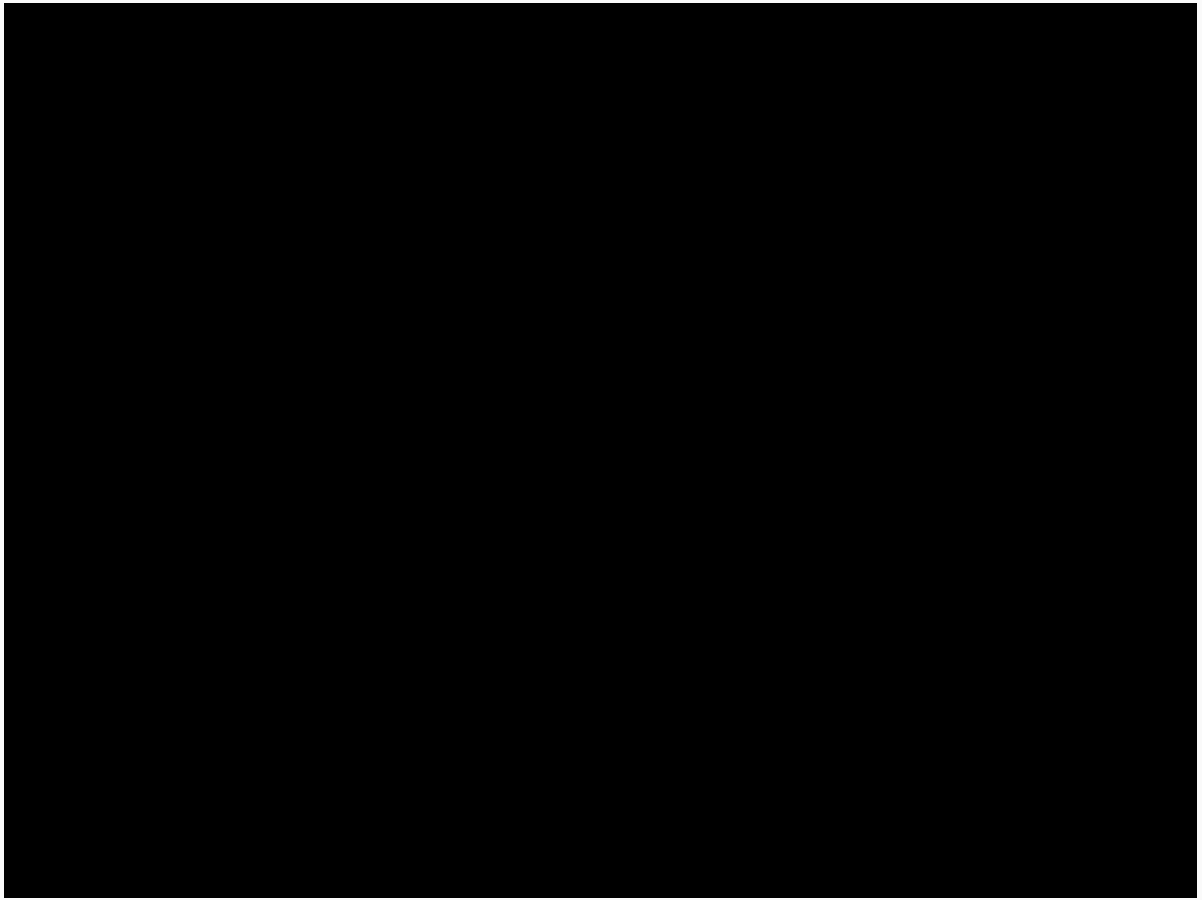
Technology Standards

- HTTPS
- RPC
- IEEE 802.11 Wifi
- IP & TCP

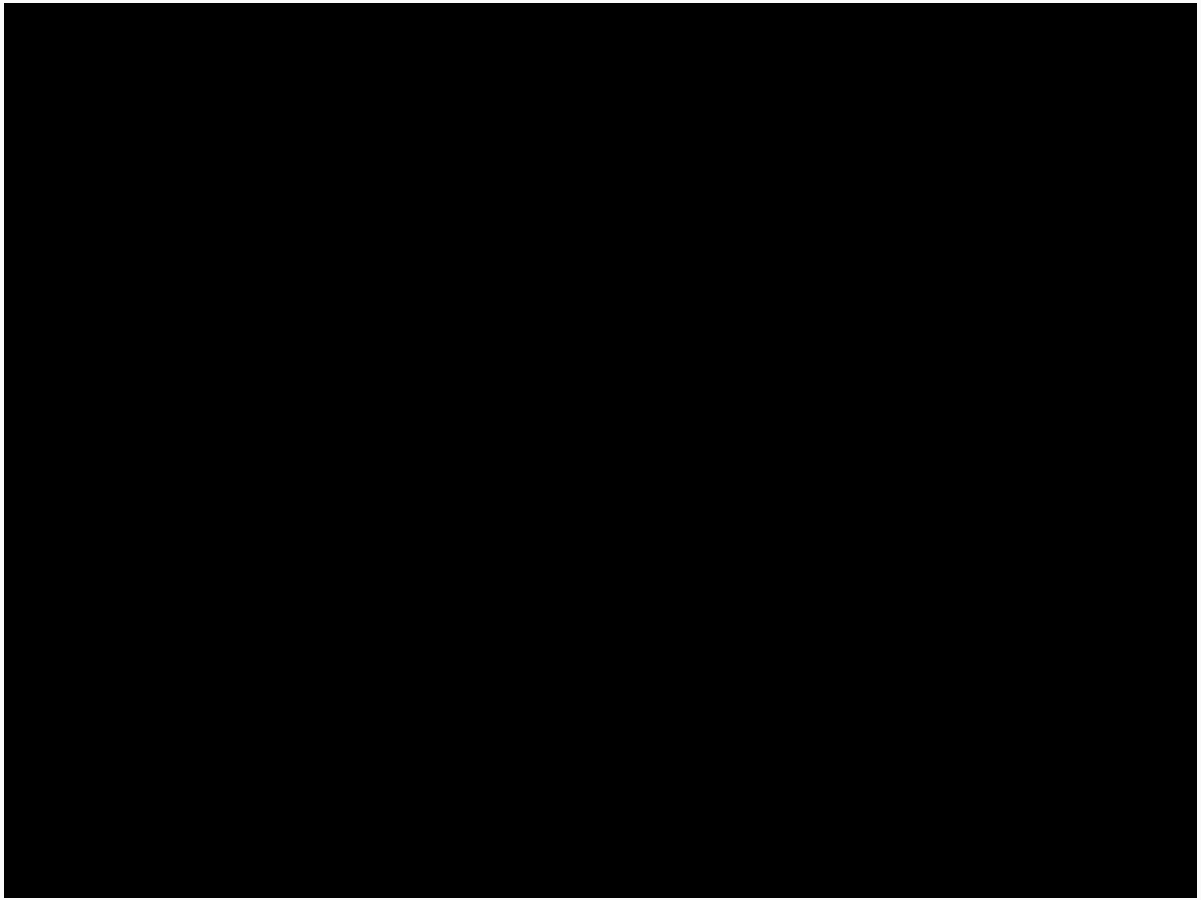
Design Practices

- Iterative prototyping
- Microservice architecture
- Modular design
- Object oriented programming

Mobile Application Demo



Sensor Setup Demo



Challenges & Takeaways

Challenges

- Low tier of cloud computing platform
- Busy schedules of team members
- Difficulty working remotely with hardware

Takeaways

- Collaboration on large tasks accelerates completion
- Work earlier instead of harder
- Integration planning from the start

Questions?

Image Sources

- Bing Maps
- Clipart Wiki
- Walmart
- Target
- Aldi
- Hy-Vee