To Online Shop, or To Not Online Shop

DESIGN DOCUMENT

SDMay20-19
Goce Trajcevski

Amiah Gooding – Electrical Engineer
Matthew Martin – Report Manager/Scrum Master
Max Minard – Software Manager
Smruthi Sandhanam – Meeting Manager
Travis Stanger – Test Engineer
Yana Aleksandrova – Meeting Facilitator

sdmay20-19@iastate.edu
http://sdmay20-19.sd.ece.iastate.edu

Revised: October 6, 2019 Revision 1.0

Executive Summary

Development Standards & Practices Used

- Agile software development
- Commenting code
- Following all safety protocols

Summary of Requirements

- Generation of data: The state of items in a cupboard
- Signal generator: Trigger to update in data
- Geolocation: Showing stores nearby and deals
- Mobile application: Receives notifications
- A signal from interface that an item has been purchased

Applicable Courses from Iowa State University Curriculum

- Computer Science 227: Object-oriented Programming
- Computer Science 228: Introduction to Data Structures
- Computer Science 309: Software Development Practices
- Computer Science 311: Introduction to the Design/Analysis of Algorithms
- Computer Engineering 288: Embedded Systems I
- Electrical Engineering 230: Electronic Circuits and Systems
- Electrical Engineering 330: Integrated Electronics
- Electrical Engineering 333: Electronic Systems Design

New Skills/Knowledge acquired that was not taught in courses

- Amazon Web Services
- Additional new skills/knowledge to be determined

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

The To Online Shop or To Not Online Shop team would like to thank the Iowa State University College of Electrical and Computer Engineering for providing the student team a professional experience, resources, and consultation with experts. We would also like to thank Professor Goce Trajcevski for meeting with us weekly and guiding us through the development process of our product. The team appreciates the University's Electronics and Technology Group's (ETG) availability for providing hardware and server components for the project.

1.2 Problem and Project Statement

The goal of the project is to design a full-stack IoT-based solution that will help to strike a balance between online shopping and offline (in-store) shopping experiences for the consumers. The goal is to develop a project that will:

a. Monitor the status of the items in a shelf or cabinet
b. Generate a list of items "to buy" and prepare an online order
c. Detecting when a particular customer is in a proximity of an actual store (e.g., Target, Walmart, etc.)
d. Automatically send notifications that certain items from the "to buy" list that are scheduled for delivery, could actually be obtained in the near-by store (and with discount coupons).

There are two main components for this project. There is the home IoT hardware component that retrieves the data regarding the status of the items from the cabinet or pantry. The data received gets populated into a database with the itemized pantry contents. The other component is the mobile application which retrieves the items that need to be bought from the home device. The algorithms should determine if it is cheaper to buy an item online or in-store. Geolocation should also be used to find the best deal for a product.

Our solution is to use an RFID sensor or barcode scanner along with a microcontroller that can automatically monitor status of the items located in the pantry or cabinet. At midnight a signal will trigger the sensors such that status of the items will be stored using an AWS database. Our mobile application will analyze the data and then generate a list containing the items that need to be bought. We will write algorithms to determine if it's cheaper to buy items online or in-store. We will also use geolocation to figure out where the best deal for buying the products are.

1.3 Operational Environment

The sensors and microcontrollers used to monitor the status of the items will be stored in a dry pantry or cabinet area with a WiFi connection. The current device is a microcontroller-connected "Smart-Bin" that is expected to fit securely on the top of the shelf. All setup aside from the device will be done through mobile application.

1.4 Requirements

Constraints

- Budget of 200 dollars.
- Only viewing a few stores within a 5-mile radius.
- A user does not want to buy an item in person if it can be found for cheaper online.

Environmental Constraints

- All aspects of the project will be tested in a simulated environment.

1.5 Intended Users and Uses

The main user for the project is anyone who is looking to save money and shop more efficiently. The user will initially have to set up the hardware in their cabinet or pantry. Afterwards they will primarily interact with product using the mobile application. The user can monitor the status of items in their pantry or cabinet and can generate a list of items that they need to buy. The app will, furthermore, analyze the data and figure out where the best places to buy the list are and whether it is cheaper to buy online or at the store.

1.6 Assumptions and Limitations

Assumptions:

- Pantry or cabinet monitoring devices have a power source
- Pantry or cabinet monitoring device can connect to internet
- Pantry or cabinet monitoring device is not at risk of water damage
- The customer properly sets up hardware

Limitations:

- Accuracy of automatic orders is completely based on the accuracy of the sensors to detect the status of an item (Ex. "Empty or full")
- The cost to produce the end product shall not exceed two hundred dollars
- The product will only be able to view the stores within a five-mile radius
- The product will not select stores which total prices exceed user's budget

1.7 Expected End Product and Deliverables

The final product will be divided into deliverables for the first semester and the second semester. The final product deliverable will be at the end of second semester.

Semester 1:

1. **Formalize the scope of the project and identify main components (September 20th, 2019)**
    - In this deliverable we are meeting with the customer and are coming up with our project requirements and are finalizing the components for our product. This is the initial planning phase.
2. **Complete the survey of relevant literature and "off the shelf" products (October 10th, 2019)**
    - In this deliverable we are deciding what products we are implementing in our project. We are to purchase the sensors and microcontrollers that we will be implementing into the final product. Figuring out how we will be developing the mobile application is also important.

3. **Decide upon use-case scenarios and main approaches to be investigated as "candidate-solutions" (November 1th, 2019)**
    - In this deliverable we are deciding the use cases and approaches that we will be implementing in the product.
4. **Converge on the design, and provide implementation(s) of some basic functionality (November 30th, 2019)**
    - Complete the initial design for the product and start some basic functionality to demo to the panel. This should include work on some core features to show initial development.

Semester 2:

1. **Revisit the design decisions (January 30th, 2019)**
    - Continue from the development of the previous features and work on assigned tasks. Demo completed features to client and make necessary improvements
2. **Define and start with the individual components testing (February 28th)**
    - Not defined yet
3. **Complete the initial integration, define and start with integration-testing for plausible use-cases (March 30th, 2019)**
    - Not defined yet
4. **Complete "dry-runs" of the demo and finalize the deliverable-version (April 30th, 2019)**
    - Not defined yet

2. Specifications and Analysis

2.1 Proposed Design

The solution design will have two main components: the Home IOT and Application side. The sensor on the IOT side will satisfy functional and non-functional requirements for the monitoring of grocery inventory in the kitchen cupboard. The application analytics will satisfy requirements of geolocation and optimization using data from the sensor network.

The sensory device will employ sensors to monitor inventory levels of single product. The client will have multiple devices in order to monitor different cupboard. The sensor will receive data and send directly to the database on the back-end. We intend to implement a RFID sensor that will communicate with AWS and our IOS application.

The software will be an IOS application. We will use swift to develop our application that will interact with our database. The front-end will use geolocation to send a notification when an individual is close to a grocery store and notify them of cupboard inventory. We will use grocery store APIs and be able to compare to online shopping prices.

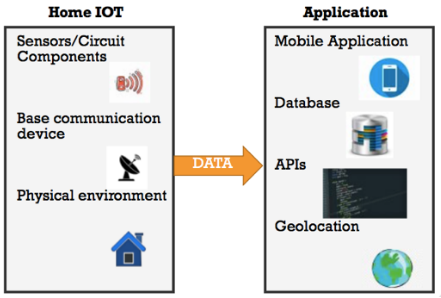

Figure 1: Deployment Diagram of Proposed Design

2.2 Design Analysis

There are two primary component systems for continued development on the project. The Home IOT system is dependent on the stream of data from the sensor network. We have to ensure that there is continuous transmission through the process. The application is reliant on the sensors and hardware being implemented and functioning. Issues in the network directly affects components on the application side ability to geolocate as well as compare prices, and make decision whether to online shop or not.

We are considering the route of exploring and RFID sensor and using an Arduino Microcontroller. There were several options to consider we chose the RFID as a dependable form of sensing items, and the Arduino for its analog-to-digital converted as well as its built in Wi-Fi/Bluetooth capabilities. Since we will be gathering a variety of data the Arduino seemed like the right choice for the implementation.

All of our data is sent to a central database and accessed in our application. For the back-end component we will be using AWS to monitor database information and make decision based of specifications and prior information. The user of our application will interact with a registration to register the sensory device and have input for the products being monitored as well as the conditions for when the product need to be ordered. The front end is going to be built with Swift for an IOS application, because of the team familiarity with the framework.

We are using RFID (Radio frequency identification) since the sensor can pick up when an item passes through the cupboard door rather than having to physically scan an item. It also has the capability to detect several items at once and has a good range of detection. We debated using a barcode scanner but this would mean each item has to be scanned individually, which would result in a cost of time and having items go one by one. RFID is the option we will be pursuing staying within our budget and being able to have an ease of sensing items.

2.3 Development Process

We will be using the Agile developmental method. This way we are allowed to iterate often and get feedback from our stakeholder often as well. It is the framework that fits best as we are meeting with our client each week to discuss development. Through these frequent meetings we will be able to discuss sprints and next steps in the development process. If there are steps that do not meet our requirements we will be able to go back and iterate on certain parts rather than starting over. Since we have a very software focused project once initial hardware is setup this will flow rather nicely, and the same steps can be applied across systems on our project.

2.4 Design Plan

Our design plan includes the 5 pillars design thinking, empathize, define, ideate, prototype and test. In empathize we met with our client to see what our stakeholder expected of our final project. In define we established the problem and constraints, and determined requirements and goals our project. Ideate is where we brainstorm solutions, analyze time and costs, assign roles, and make a plan for going forward. For the prototyping section of our design process we split our development into hardware and software. This is to develop the two separate parts separate test them on their own and then integrate them back together for final testing. As this illustrates our overall design process we have the ability to go back and iterate to fix or improve designs.

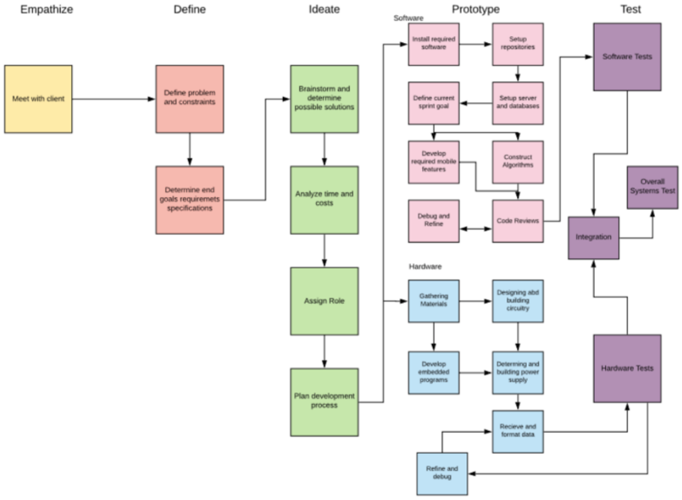
Empathize
Define
Ideate
Prototype
Test
Software
Meet with client
Define problem and constraints
Determine end goals requiremets specifications
Brainstorm and determine possible solutions
Analyze time and costs
Assign Role
Plan development process
Install required software
Setup repositories
Define current sprint goal
Setup server and databases
Develop required mobile features
Construct Algorithms
Debug and Refine
Code Reviews
Hardware
Gathering Materials
Designing abd building circuitry
Develop embedded programs
Determing and building power supply
Recieve and format data
Refine and debug
Software Tests
Integration
Overall Systems Test
Hardware Tests

Figure 2: Design Diagram

3. Statement of Work

3.1 Previous Work And Literature

Automatic inventory management is not a new concept and has been integrated in the past. Companies such as Rolls Royce are implementing sensors to track their inventories locations. In addition to them many other companies have implement RF id tags to track what is moving from room to room or warehouse to warehouse.

However the portion of our project that checks the prices is not as commonly known. But it is well known that there are quite few price checking applications available for people to find saving. These application are mainly built for specific items or stores though. The prices evaluations that need to be set up for our project need to be for multiple stores and items. Additionally they will be geographically based for which store is the cheapest.

3.2 Technology Considerations

For sensor tracking there are several different variation available on the market. However, for our project we were able to reduce sensor option to two different kinds Bluetooth or RF ids.

Bluetooth

- Positive: Easy integration and quite inexpensive
- Negative: Can be prone to interference with other devices

RF ID

- Positive: Will never have to worry about battery life and accuracy is very high
- Negative: Tend to radiate a lot of energy to get tag signal

3.3 Task Decomposition

Creating tasks into small actionable items helps a project reach the end goal. This project can be separated into the following tasks:

1. Creation of environments
    a. An environment for the sensors to begin collecting data.
    b. AWS system is set up with EC2 and database instances.
2. Have the sensors begin to send data from the cabinet into the database being hosted on AWS.
3. Application pulls in information from the database on what item a user needs to purchase.
4. Application calculates the cheapest option for purchasing the item, whether in store or online.
5. Application notifies the user on the action that needs to take place.

3.4 Possible Risks and Risk Management

Risks:

- Accuracy of automatic orders is completely reliant on the accuracy of the sensors used in the sensor network
- Can only make correct orders if the sensors properly establish a connection with the application which will then run an algorithm to tell which is the cheapest option
- Unfamiliarity with Technology

Risk Management

- Frequent testing of sensor network
- Ensure to test pre-made database for good price differences
- Securing outside sources that are familiar with the technology and being sure to compile information and placing in one drive for other to use

3.5 Project Proposed Milestones and Evaluation Criteria

- Proof-of-Concept

    - The phone application component should be set-up and ready for user to log in and view data being collected by the storage monitoring device.
- Minimum Viable Product
    - The monitoring devices should be properly calibrated to measure what is being stored. The phone application component should be able to monitor storage levels of multiple pantries in multiple homes and be able to produce a shipping orders and an optimized route based on those storage levels.
- Beta-Testing
    - The device and application will be constantly tested with real products over a period to time to find bugs, issues, and areas to improve.
- Integration
    - The phone application is modified so it is ready with the pantry's database
- Finalize Product
    - The finalize product will have all bugs fixed while the phone application's monitoring and price checking system is optimized. The entire project set up to handle multiple smart-bins from multiple pantries.

3.6 Project Tracking Procedures

We will be posting a weekly status report to track our progress made during the week. We will also be documenting our meeting notes after every meeting to record our thoughts and concerns that don't make it onto the weekly report. Having separate meeting notes will also serve as a way to validate what is discussed and decided during meetings.

Gitlab will also be heavily used to keep track of weekly progress. This will be an easily accessible way to know where we are on more specific tasks. It will also be a way to look back on more specific tasks that are generalized on the weekly report.

3.7 Expected Results and Validation

Validation 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 product count by comparing the calculated total with the true value. Validation will occur with a variety of products at various quantities to ensure our measurement algorithms return accurate data on what the product is. 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.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

Task #	Milestone/Activity	Start Date	End Date
1	Formalize Scope/ Identify main components	9/6/19	9/20/19
2	Research	10/1/19	10/10/19

3	Use-case scenarios and main approaches	10/7/19	11/1/19
4	Converge on design/ Basic functionality	11/1/19	11/30/19
5	Revisit design decisions	1/13/20	1/30/20
6	Begin Component Testing	1/30/20	2/28/20
7	Complete initial integration w/ testing	2/28/20	3/30/20
8	Complete “dry-runs” of demo	3/30/20	4/30/20

Task #	Sep-19	Oct-19	Nov-19	Dec-19	Jan-20	Feb-20	Mar-20	Apr-20
1	X							
2		X						
3		X						
4			X					
5					X			
6						X		
7							X	
8								X

As you can see from the tables above, we have separated out our project into 8 main components that span the time allotted for the project. From the initial scope to the final demos, the table gives an overview of what is to be expected and by when each part should be completed. It also accounts for winter break, which explains the gap in the month of December. We feel like these steps highlight the main tasks that will be tackled on a monthly basis, and it gives plenty of time for the actual construction of the solution. We will follow an agile-like approach to the software development, and make sure to constantly test the hardware and software separately before integrating and testing the final, complete system. We have even given a month after the final integration to test, demo, and possible expand upon our solution.

The timeline also allows for an adequate amount of time for preparation before we start creating a solution. It accounts for research, brainstorming, and analysis of requirements and constraints before constructing a prototype. This will ensure enough thought is put into what is expected as a final result and what steps can be taken to accomplish this. Overall, this timeline distributes a sufficient amount of time to each component of our project and, if followed correctly, will allow our team to finish our project while meeting quality expectations and time constraints.

4.2 Feasibility Assessment

The expected result of the project will be a proof of concept IOT solution for a specific area in the house, with hopes of potentially expanding to other parts of the home as well. As for a realistic projection, our team believes we will be able to successfully construct a proof of concept, especially since it will be narrowed down to a single location in the home. Any foreseen challenges will primary reside in tackling new technologies that our teammates might not have had experience with prior to the project. One example of this might be using Amazon Web Services to set up our server and database. It will require more research to use this technology since none of our team members have experience with it.

4.3 Personnel Effort Requirements

Task	Description	Approx. hours
Set up AWS server and database	Research and set up the AWS server and backend services, as well as database design.	40+ After further research is done, a better estimate can be given on time.
Develop front end user interface	Design and construct the iOS mobile application that the user will interact with.	100 This task will take up a lot of time since it handles the majority of the mobile application that needs to be built.
Algorithms and data analysis	Perform any required analytics and algorithm formulation to effectively compare the prices between online and local stores and whether the user's distance away from the stores might affect this decision.	50 This task will require a decent amount of time since the algorithms we produce will influence the choices made by the user. Therefore, constant analysis of these algorithms will be needed.
Construct circuits and integrate hardware components.	Effectively plan out and construct the overall hardware involved with the project.	80 This job will require a lot of work from the hardware team since it must be mapped out and calculated perfectly.
Embedded programming	Program the Arduino to correctly transfer the data collected from the sensors to the AWS server.	80 This task can be tedious and require some research to pursue, so it has been given a decent amount of time.
Testing and Integration	Consistently test the software and hardware separately before integrating them into one system. Perform subsequent tests and demos on overall system.	150 We made sure to leave a lot of time for testing to ensure that the quality of our solution is the best it can be. This helps alleviate the possibility of errors and future confusion after integration has been completed.

This table represents the main tasks involved in the construction of our IOT solution. Each task is given a description and estimated time to complete it, as well as an explanation behind these approximations. As of now, these are just estimates of how much time each task may take and they are subject to change during the process of development throughout the year.

4.4 Other Resource Requirements

The specific materials needed to complete the project are:

- Arduino Uno
- ESP8266 Wi-Fi module
- RFID or barcode scanner
- Wires and circuit components
- Power supply
- AWS server and database
- Makeshift cabinet and items for test simulation

4.5 Financial Requirements

The financial resources provided to us is 200 dollars, mainly intended for hardware components. After researching specific products, we have listed the prices of any components that we may need to purchase below:

Arduino:	$19-$25
ESP8266:	$7-$10
RFID or barcode scanner:	$50-$100

After a quick look at these prices, we can assure that we will meet the $200 limit that has been given to us.

5. Testing and Implementation

5.1 Interface Specifications

In any project, comprehensive testing is a major component to having a successful outcome. It will be crucial to test the sensors we use in our IoT network. With an agile operating style, we will consistently be testing our sensors to ensure accurate data is being pulled into the database. To guarantee the sensors work for an increased number of items, tests will be performed on various quantities and varieties of items. The data pulled by the sensors will then be compared to the database to ensure the mobile application is able to run the right algorithms and notify the customer.

5.2 Hardware and software

The first of the sensors we will be using is a RFID sensor. This sensor will be tested extensively for its accuracy. The RFID scanner will keep track of the quantity of items an individual has in their cabinet. When the RFID sensor indicates an item is no longer in the cabinet it should alert the application that the user is out of that given item. Testing this sensor will be done using software practices as a connection from the RFID sensor to the Arduino will need to be working to have good communication between the devices. Most of the testing with the RFID sensor will be finding the most efficient way to use the sensor to detect if an item is in the cabinet or not. When an item is detected as no longer being in the cabinet, the database being hosted on AWS will be updated for this item. The testing for this will be done from the web application to verify the proper tables have been updated. This testing is very important for this project since knowing exactly what is in the cabinet is key for knowing when notify a user to buy an item.

The testing of the software will be completed in a variety of way. First, the algorithms and code will be tested through automated unit tests. This will ensure the functionality of the code is always stable. Additionally, the back-end API calls will be tested through a software called Postman. Postman is a program that provides a GUI for testing REST API calls, which are being used in this project. This will help with the automation of feeding data into the database.

5.3 Functional Testing

Throughout the project many tests will be completed to ensure the functionality is performing as expected. In the application itself, unit tests will be created and ran for each feature. These can be created directly within XCode as we are programming the mobile application in swift. After the unit tests are complete, integration testing will be completed where the various features will then be tested as a group. To follow up this testing, the functionalities of the application will be tested from front to end. This will be done to ensure there is a direct flow from items being detected in the cabinet to individuals receiving alerts to either go buy the item or that it has been bought online for them. Lastly, we will test the acceptance of the application and make sure the system is in compliance with the requirements set forth by the customer.

5.4 Non-Functional Testing

Testing the non-functional features is important for the user experience. As more individuals begin to use the application, parts of the program can possibly not work as well as expected. This is where the non-functional testing comes into play. With our application being hosted on AWS and using its various services, it makes testing for these cases easier. There are many platforms that can be used to test these areas. For example, we can use BlazeMeter to simulate additional traffic on our application. This will help us explore the scenario of the number of users using the application increasing dramatically. We will be using services like BlazeMeter to test the availability, scalability, and reliability of our application.

5.5 Process

The main process for testing the sensors will be completed by comparing the data collected by the sensors and stored in the database to the actual quantity of items in the cabinet. This testing will ensure the data being sent to the server with customer visibility is accurate.

The algorithms will face unit tests and user tests to confirm the expected output of the application. This will involve using the pre-populated table as mentioned in section 5.2. Additionally, users will test the application to test the given features, give feedback, and report any bugs found in the program.

The testing of the primary features included in the application will be tested separately from the sensors. To test these features which rely on data from the sensors, a pre-populated database will be used to simulate different scenarios. This type of testing will be helpful when trying to have granular testing of the features. Each feature laid out for this part of the project will have automated tests created so that the implementation can be continuously tested as new features get applied. This setup will ensure regression errors are prevented at a high rate.

5.6 Results

1. Sensor data retrieval testing
   a. No tests completed yet
2. Item purchase testing
   a. No tests completed yet
3. In-person purchase route optimization testing
   a. No tests completed yet

6. Closing Material

6.1 Conclusion

Thus far, our team has learned about all of the expectations of our client and gathered the necessary requirements to discuss at our meetings we have conducted so far this semester. Additionally, our team had done a good job at working with each other in order to come up with a solution for our problem. We are currently in the early development stages and still ironing out the final details before we begin prototyping. We have split into two main subgroups, one with two people and the other with four people, in order to organize the work for this project. The two-person subgroup will primarily work on the Home IOT, including sensors, circuit components, the base communication device, and the physical environment. The four-person subgroup will work primarily on the mobile application, geolocation, database, and all necessary APIs.

6.2 References

Pricing Reference Websites

https://store.arduino.cc/usa/arduino-uno-rev3

https://www.sparkfun.com/products/13678

6.3 Appendices

Arduino Datasheet

https://www.arduino.cc/en/uploads/Tutorial/595datasheet.pdf

ESP8266 Documentation

https://buildmedia.readthedocs.org/media/pdf/arduino-esp8266/docs_to_readthedocs/arduino-esp8266.pdf