**CS3354 Software Engineering**

**Final Project Deliverable 2**

Group 8

Grab and Go Groceries

Payton Harmon

Isaac Blackwood

Jonathan Schroeter

Luis Pensado

Caleb Munson

Cameron Cook

Brian Doupnik

**Table of Contents**

[Project Deliverable 1 Content](#_gpplf4b8yhv1) 3

[**PROJECT DESCRIPTION:**](#_jvxmudxpowth) **3**

[**ADDRESSING PROPOSAL FEEDBACK:**](#_94zte24cov2i) **3**

[**REPOSITORY**](#_xcflsposju19) **4**

[**TASK DELEGATION:**](#_wqwojs9xg17) **4**

[**SOFTWARE PROCESS MODEL:**](#_pkfgkoi2fa9g) **5**

[**FUNCTIONAL REQUIREMENTS:**](#_1x8pkw6yz4w2) **5**

[**NON-FUNCTIONAL REQUIREMENTS:**](#_7eoiuun77qj) **5**

[**DIAGRAMS:**](#_180o1kkylqws) **8**

[Use Case Diagram](#_vs2g4tuby5g3) 8

[Login Sequence Diagram](#_fzrk2zwkiwk4) 9

[Search for an Item Sequence Diagram](#_v354h7q09wss) 10

[Select Store Sequence Diagram](#_3q56md2gi03h) 11

[Add Item to Shopping List Sequence Diagram](#_tdjchx2g7hyi) 12

[View Shopping List Sequence Diagram](#_b35t5n7l2k33) 13

[Receive Grocery Store Layout Sequence Diagram](#_j74ruwov56r) 14

[Class Diagram](#_jyi0nyhbwv80) 14

[Architectural Design](#_6fbz5084q5n3) 15

[Project Deliverable 2 Content](#_f2jzb9oudggz) 16

[**PROJECT SCHEDULING:**](#_bvnw87pfdygn) **16**

[**COST, EFFORT, AND PRICING ESTIMATION:**](#_e0hu8960lyh9) **17**

[**COST OF PERSONNEL:**](#_m2gv6bd67ell) **19**

[**COST OF HARDWARE:**](#_fv95ac62f8fi) **20**

[**COST OF SOFTWARE:**](#_irbrl5jlp2hh) **21**

[**TESTING:**](#_owcb2m9yh2tj) **22**

[**COMPARISONS:**](#_jgkp8v5q0ojv) **25**

[**CONCLUSION:**](#_7b7g9ychkcgm) **26**

[**PRESENTATION LINK:**](#_qxtxlazcne8w) **26**

[**References**](#_3reqakjn63pi) **27**

### Project Deliverable 1 Content

# **PROJECT DESCRIPTION:**

We will be designing an app that takes from different grocery stores databases, to enable customer searches for items over multiple stores. The customer should select a store then search for an item. The app then finds where the item is in the store and outputs the location to the user. The app will give the location to the user in the form of a map of the store with a guide to the item.

# ADDRESSING PROPOSAL FEEDBACK:

It was suggested that we provide a comparison to other similar services. To address these areas, in the final deliverable we plan on doing thorough research on potential competing services and applications. This will enable us to provide a comparison between their model and ours, and emphasize how our product differs from theirs.

# REPOSITORY

<https://github.com/jonathanschroeter/3354-Grab-and-Go-Groceries>

# TASK DELEGATION:

* Addressing to proposal feedback - Payton September 29th
* Software process model and why - Group September 29th
* Functional Requirements - Caleb, Jonathan, Cameron, Payton September 29th
* Non-Functional Requirements - October 6th
  + Operational - Luis
  + Development - Isaac
  + Accounting - Cameron
  + Safety/Security - Brian
  + Ethical - Cameron
  + Regulatory - Payton
* Case Diagram - Jonathan, Caleb October 8th
* Sequence Diagram- October 13th
  + Jonathan - Log in
  + Caleb - Search for Item
  + Caleb - Select Store
  + Cameron - Add Item to Cart
  + Payton - View Shopping List
  + Luis - Receive Grocery Store Layout
* Class Diagram - Brian October 13th
* Architectural Design - Payton (October 13th)
* Project Scheduling - Luis, Brian (October 27)
  + Cost, Effort, and Pricing - Tentative (Payton) (November 3)
  + Cost of Hardware - Caleb (October 29)
  + Cost of Software - Cameron (October 29)
  + Cost of Personnel - Jonathan (October 29)
* Test plan/Code (Look up items/store) - Isaac (November 3)
* Comparison with similar designs (Walmart shopping, Google maps, Instacart)- Brian (Instacart), Jonathan (Walmart Shopping), Isaac (Google Maps) (November 3)
* Conclusion - Team (November 5)
* References - Payton(November 5)

# SOFTWARE PROCESS MODEL:

Prototyping model: Prototyping model is best for our development model because our project is modular in nature and works well with incrementally prototyping and adding new modules with each prototype. Furthermore, due to the application’s increased utility during covid, we would like to have fast development with a minimum viable product quickly available.

# FUNCTIONAL REQUIREMENTS:

* The users and stores should be able to create, edit and delete their account.
* The user should be able to select item(s) from a grocery store and have its locations displayed within an accurate map of the store's layout.
* Each store must be individually identifiable within the application based on its address and distance from the user.
* The user should be able to intuitively search for any item offered by the store and have its price, stock and location displayed.
* The user should be able to send their grocery list to a specified store and get pricing, stock and location information displayed for that specific store.
* Stores must be able to set and update item locations in their layout.

# 

# NON-FUNCTIONAL REQUIREMENTS:

Security Requirements:

* Each User’s data should be stored in an object on a central database with user data as members. There will be create account, edit account and delete account services each provided by a server.
* The password should be 12 characters long and have uppercase, lowercase, a number, and a special character.
* The central database for account information needs to be securely maintained so user account and GPS information are accessible to malicious third parties.

Usability Requirements:

* The application should be able to adapt to different screen resolutions and include colorblind assist.
* Users of all ages should be able to use the application intuitively with no more than 2 “tips” required.
* Adding items to the grocery list should be intuitive for users to understand without having to go to our help page for assistance.

Space Requirements:

* The app should be under half a gigabyte on both IOS and Android devices.
* Each individual company should manage their own catalogue of items.

Performance Requirements:

* Our servers should be able to withstand a large number (determined by marketing department projections) of concurrent users without having any slowdown.
* When the user makes a query, our internal latency should be less than 5 seconds.

Dependability Requirements:

* Servers should maintain 99.9% uptime.
* Bugs should not be mentioned in more than 10% of user reviews
* Servers should only be down on Sundays from 2 am through 4 am CST for server maintenance.

Environmental Requirements:

* Users should be able to access the applications on both android and IOS platforms.

Accounting Requirements:

* The application operational cost should be cost less than what we make in revenue from partnered stores, advertisement, and data collection.
* Storage payments should be covered by companies that we are storing data for.
* We should use profits to advertise the product on a local level and to develop the application further

Ethical Requirements:

* We shouldn’t store or sell data that doesn’t directly help the customer with an improved shopping experience
* Customers can see the data we are storing from their device usage through a transparency program
* Our application shouldn’t hinder the experience of other customers

Safety/Security Requirements:

* To comply with data security laws across the United States, personal information should be salted and hashed before being saved in the database, and encrypted client side.
* When a user deletes their account, proper protocols must be used to ensure that none of their personal information remains on our servers
* Present users with a comprehensive agreement clarifying that we will not sell their data for money and consenting to their information being stored on our servers.
* Users shall receive a notification of profile changes via preferred communication method on record when profile information is modified.

Development Requirements:

* The application will be developed in Java, because it is low cost and portable. Must be ported to swift for IOS

Operational Requirements:

* Users of the application shall identify themselves using their provided username and password.
* Operations will be limited to the continental United States and the partnered stores.
* The user must be able to access the internet to use the application.

Regulatory Requirements:

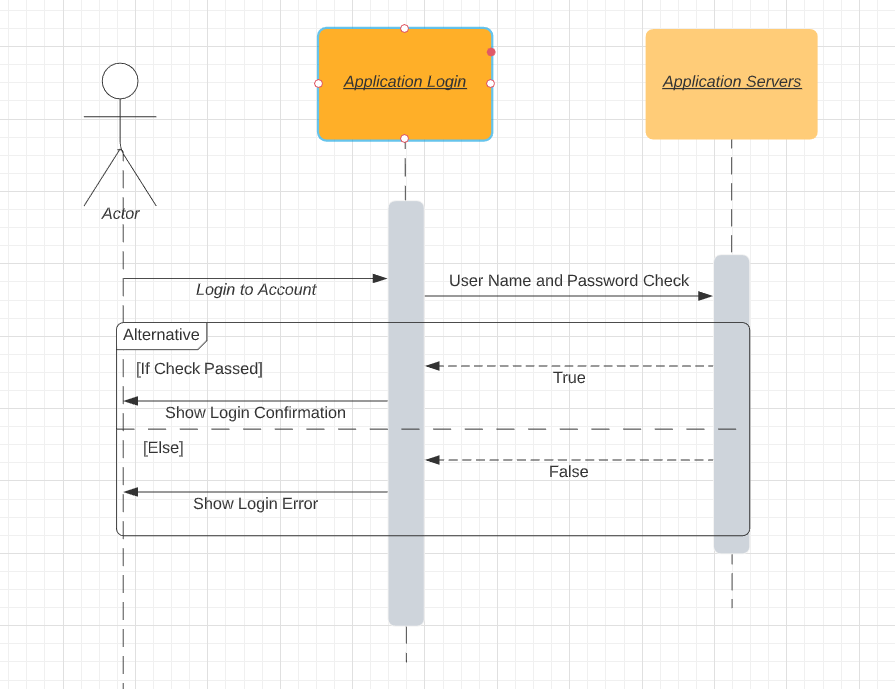
* The privacy policy must be available for the user to access.
* The user must be notified and asked to accept use of cookies due to GDPR regulations.

# 

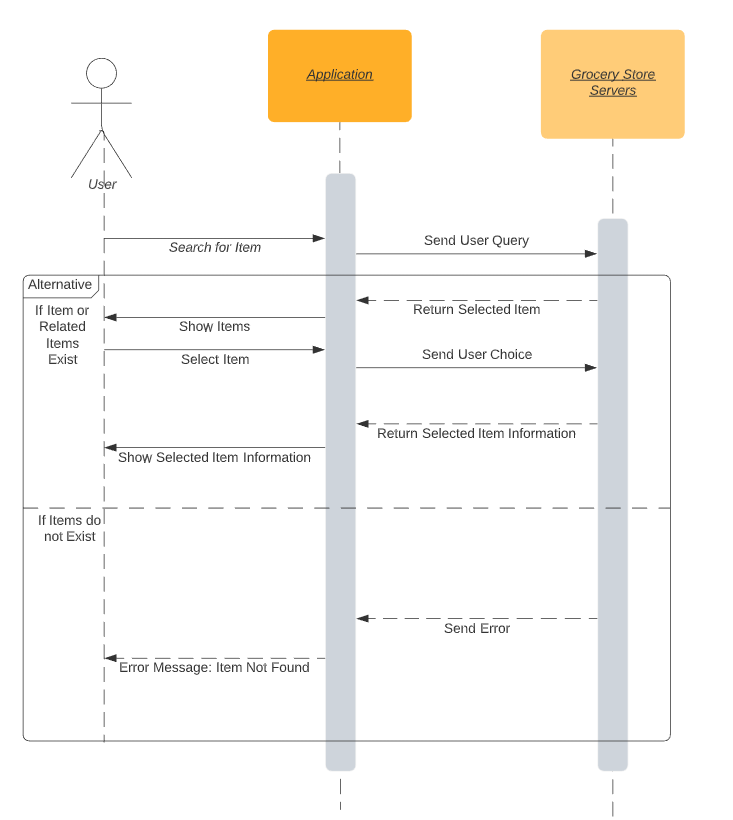
# **DIAGRAMS**:

## Use Case Diagram

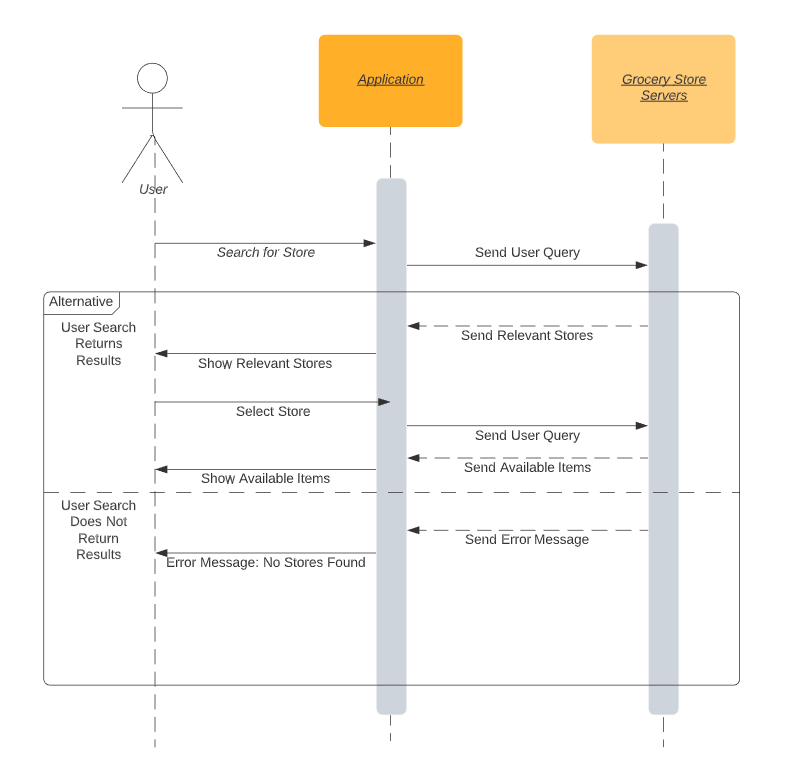
## Login Sequence Diagram



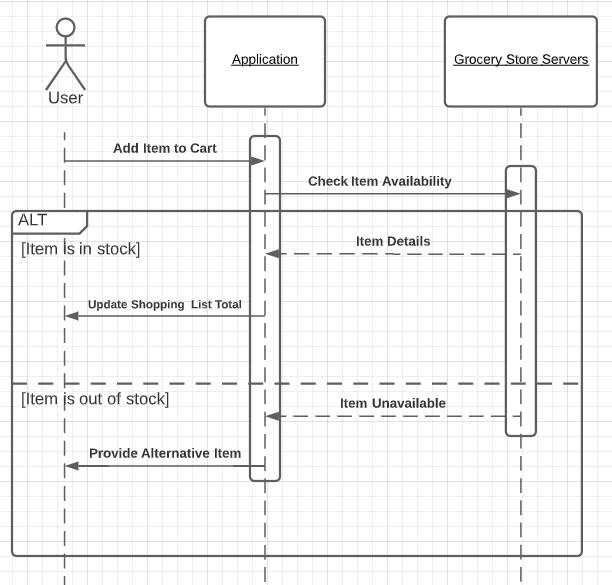
## Search for an Item Sequence Diagram



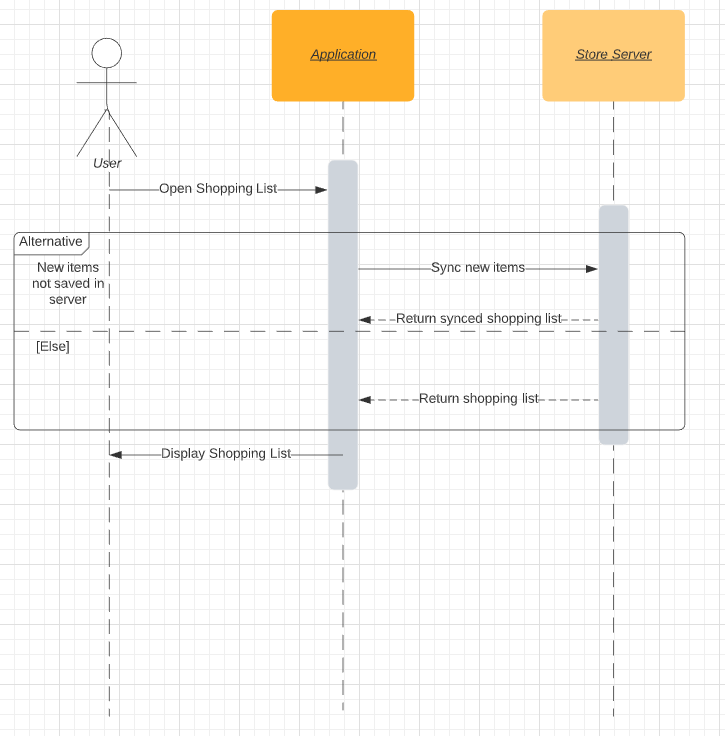
## Select Store Sequence Diagram



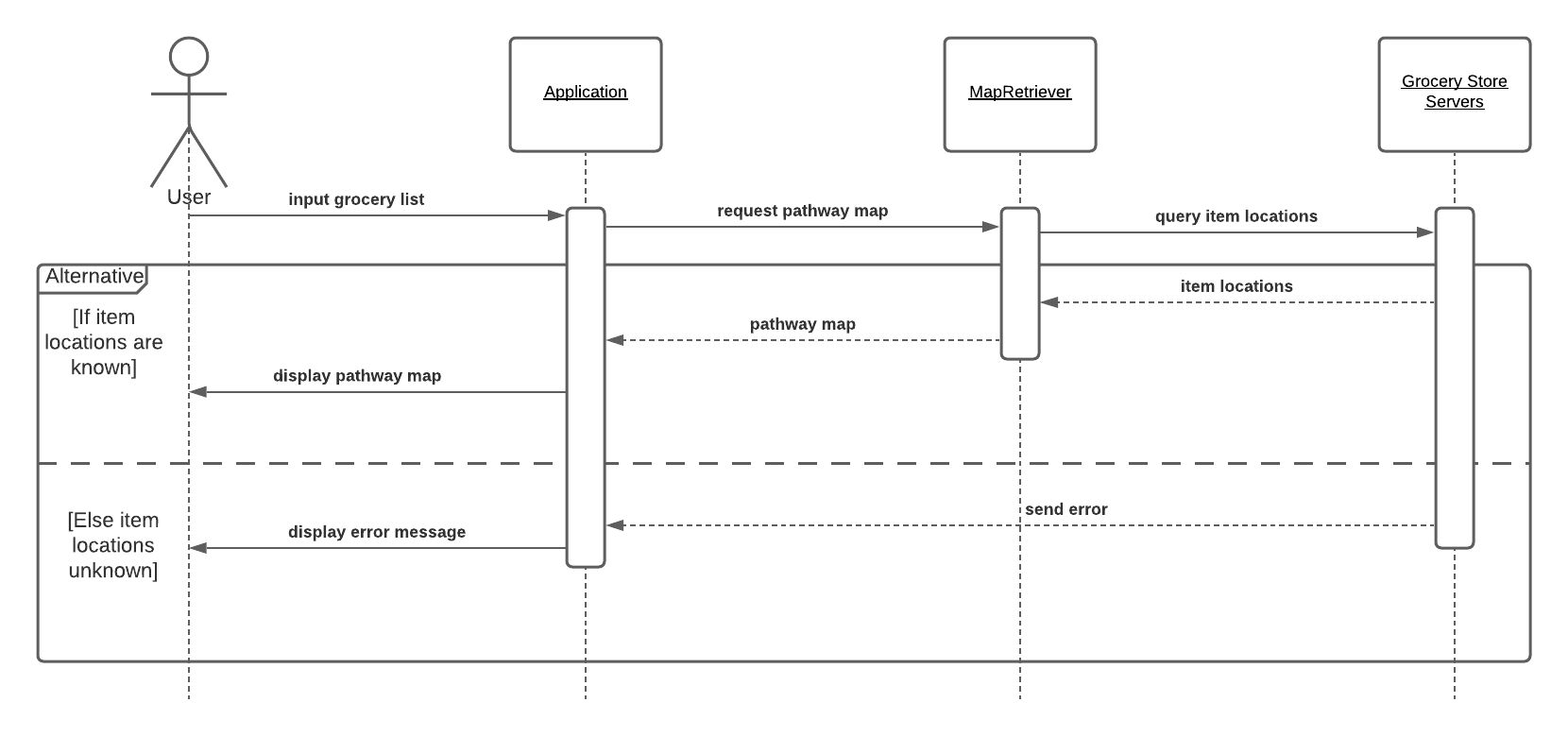
## Add Item to Shopping List Sequence Diagram



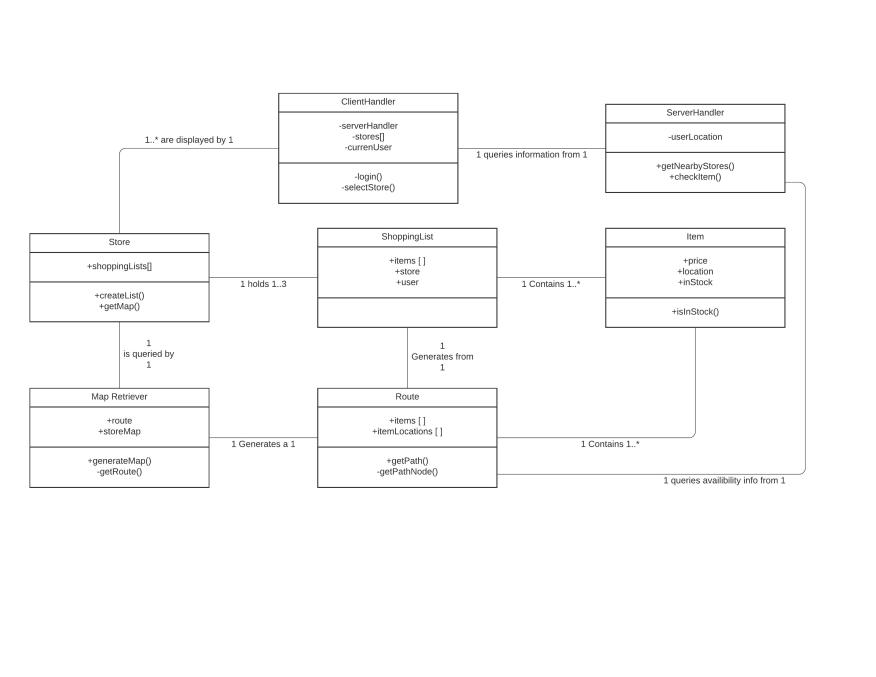
## View Shopping List Sequence Diagram



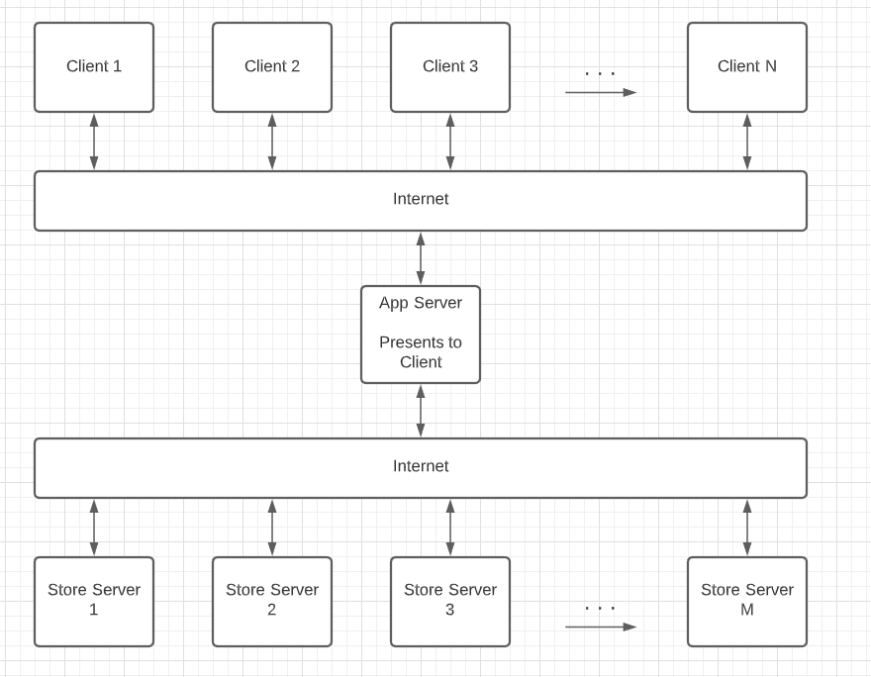
## Receive Grocery Store Layout Sequence Diagram



## Class Diagram

****

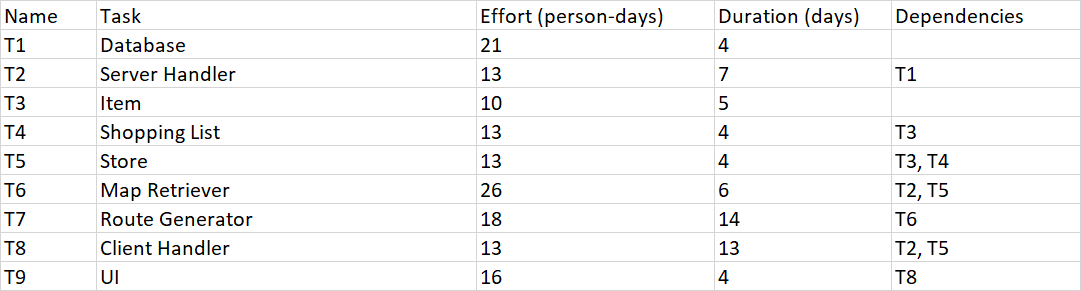
## Architectural Design



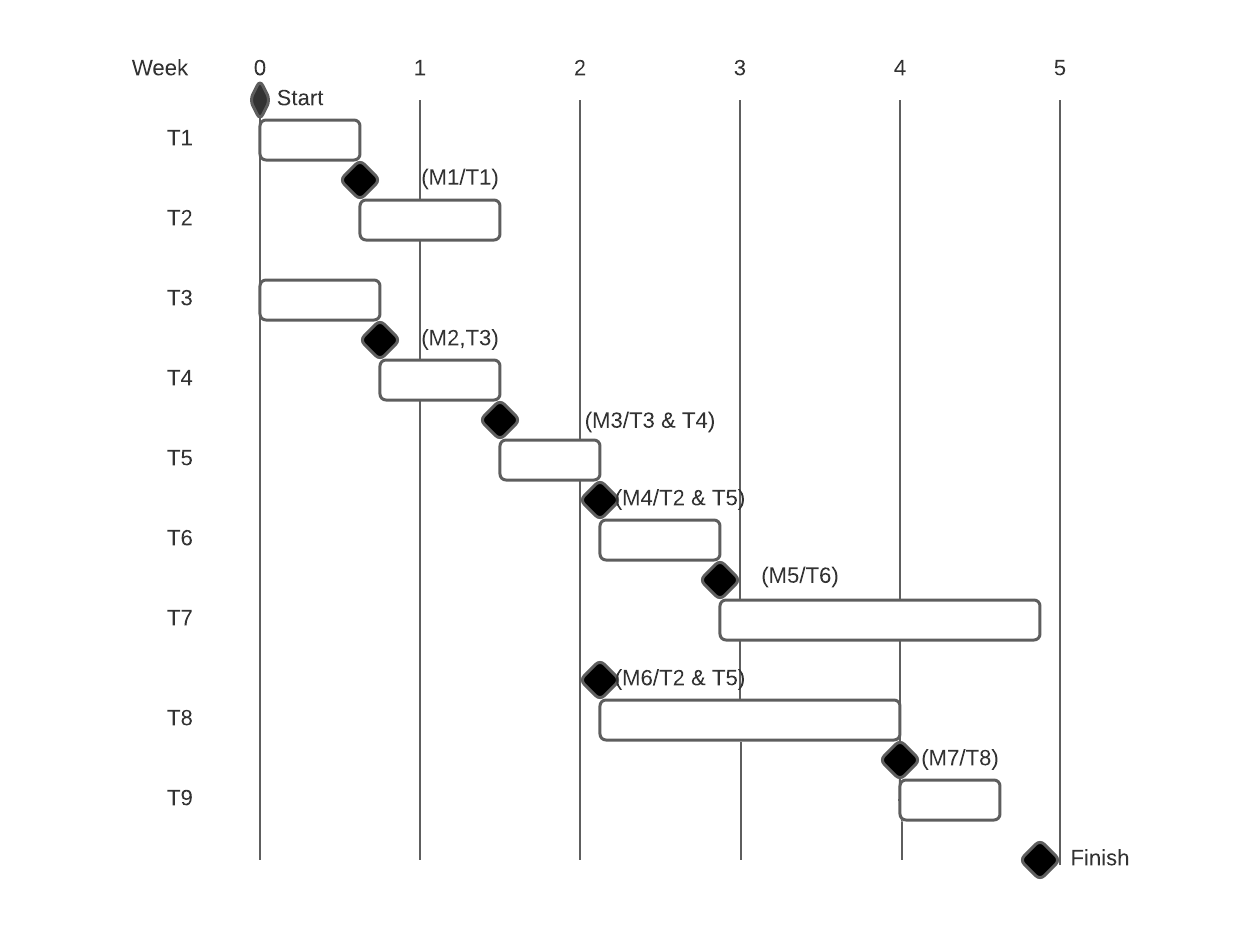
### Project Deliverable 2 Content

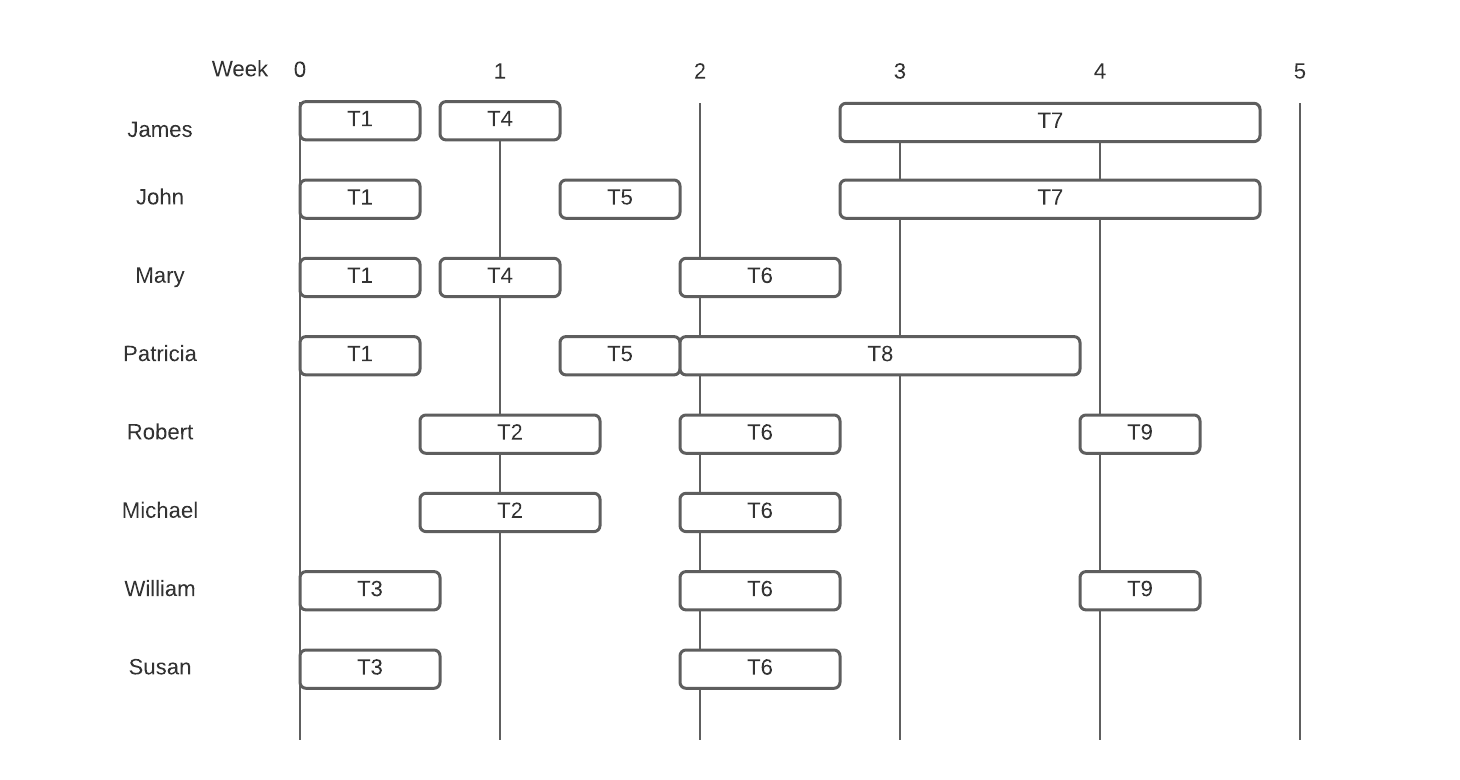
# PROJECT SCHEDULING:

Employees will work 8 hours a day and 7 days a week.

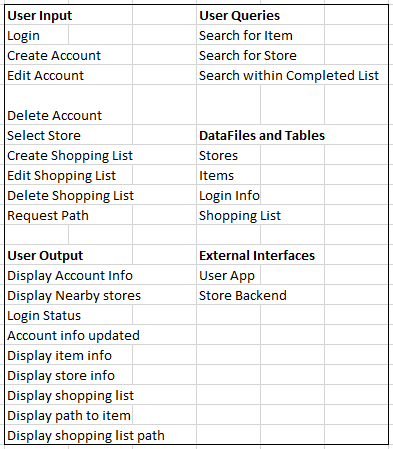
**Activity Dependency Chart**  


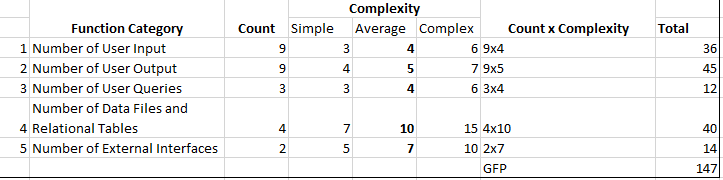
**Activity Bar Chart**



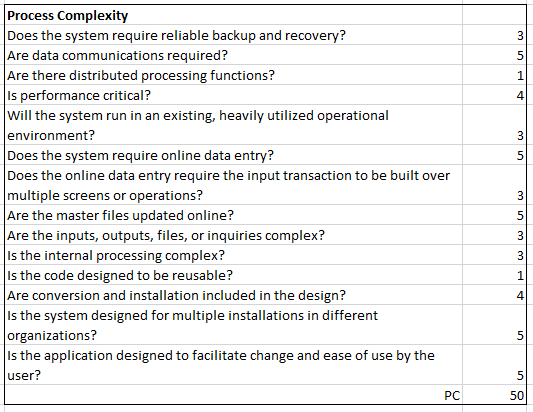
**Staff Allocation Chart**

# COST, EFFORT, AND PRICING ESTIMATION:

For our estimation we decided to use function point estimation, and assume that the average developer could complete 10 function points per week. The counts were determined as follows:

We determined that they all have average complexity and got the following chart to calculate the Gross Function Point:

Next we calculated the Process Complexity by assigning a value between 0-5 based on the importance of each questions:



With the PC, we can now calculate the process complexity adjustment (PCA) using the following equation:

PCA = 0.65 + (0.01 \* PC)

Substituting for our PC we get:

1.15 = 0.65 + (0.01 \* 50)

Next we’ll calculate the function point (FP) by taking GFP \* PCA:

147 \* 1.15 = 169

Now we’ll calculate the expected time of work assuming each worker can do 10 function point per week:

169 / 10 = 16.9 weeks.

And then if we have 8 workers the total time would be approximately:

16.9 / 8 = **2.1125 weeks.**

This time reflects approximate time *assuming no scheduling issues*.

# COST OF PERSONNEL:

(Using Lines of Code Method)

FUNCTIONS: LINES OF CODE:

UI/UX 300

ClientHandler 250

Database Management 400

Store 250

Map Retriever 500

ShoppingList 250

Route 350

Item 200

ServerHandler 250

2,750

Using the estimate of 537.5 LOC/pm from [1].

Cost of outsourcing developers from India $35 an hour, with them working around 40 hours a week, so 5,600 a month as seen in [2].

Cost per line of code is

5600/537.5 = $10 per line of code

10 \* 2,750 = $27,500 total estimated project cost

2,750/537.5= 5.1 persons-month

with 8 developers the ETA will be .6375 months (assuming everything can be done concurrently).

We need to find the cost per month for 8 developers

5,600\*8= 44,800

Since our project is only going to last .6375 months or .7 months 44,800 \* .7 = $31,360, will be the total cost estimation for the project personnel.

This LOC estimate does not take into account parts of the project that can not be done in parallel, so the ETA will be longer than this.

If we take into account project scheduling, the FP method, and the LOC method we can expect a timeline between 2-6 weeks with a minimum price of $22,400 and a maximum price of $67,200.

We would need to have some technicians for IT/technical support for customers, and one full time developer to fix any issues that arise.

For the developer, we would want them to be from the United States so they will be able to fix things on the fly during business hours. The average salary for a software developer based on the U.S Bureau of Labor is $107,510 per year according to [3].

IT would be outsourced, and based on North Star Inc, it would cost between $1000-$2500 per month. So as a worst case scenario, we will go with $2,500 per month as provided by [4].

So our development costs are (as a worst case scenario) $67,200 for the cost of personnel for development, $107,510 per year for a software developer, and $30,000 per year for IT/technical support. Therefore, the cost of personnel for our first year will be $204,710.

# COST OF HARDWARE:

We will need to utilize a server for our program.

In the short term, it would be better to rent a server rather than own a server since we would not have any upfront costs. Also, by renting a server, we would not have to worry about maintenance costs on the server.

We would likely want to have a mid-size server, with the possibility of considering upgrading to a large server eventually if needed.

The costs for a medium server is $140/month (Given we use servermania’s pricing seen here [5])

Our initial test of the server would be 1 year, so the total would be 12 \* $140 or $1680.

After the year is up, we could consider whether it would be worth to rent a different server or even choose to own a server (and maintain it)

Since we are outsourcing the coding and development, we would not have to provide any equipment or computers for employees.

# COST OF SOFTWARE:

**Development Environment:**

Because we are outsourcing our development and coding, we will not need to pay the licensing or royalties that are commonly involved with commercial development. These costs are found in the IDEs that are used and, on some occasions, the cost to use a programming language at a commercial level.

**API Access:**

To work with some stores, we might require access to their API Keys. One store we looked at was Walmart, and they offer API access through their Open API program. This key was offered through a closed beta program as seen at [6]. API’s usually use a monthly cost program, ranging from $25 - $500 depending on the complexity and number of allowed requests. We estimate that access to Walmart’s API could cost $150 / Month. Another store, Target, offers a $50 / Month API that allows Unlimited requests and is gotten at [7].

The Google Maps API is free for iOS and Android deployment with unlimited requests, however Javascript API access is $7 / 1000 requests / month from [8]. Because we are targeting a mobile deployment, those are the primary costs. However, if we chose to utilize the Google Maps functions in the back end, that would be an additional charge.

**Deployment:**

Software costs that come from deployment include the software solutions we are using in the back end. Since we are using our own server, we need to license the server software and tools that we would utilize. A Windows Server license costs $550 from [9] and the rest of the back-end software can be open source and free.

**Distribution:**

There are generally fees involved with distributing commercial software. These are costs that we would need to cover. According to [10], there is an annual fee of $99 to publish as a developer on the iOS App Store as well as a 30% cut of revenue that Apple receives. There is not an equivalent annual fee on the Android Google Play Store, but similar to Apple, they take a portion of the revenue equal to 30%.

**Costs Breakdown:**

API Access Target $ 50 / Month

Walmart $ 150 / Month

Google Maps Backend $ 7 / 1000 Requests (Estimated $70 / Month)

**Total $ 270 / Month**

Deployment Software

Windows Server $ 550

Distribution

iOS Developer License $ 99 / Annually

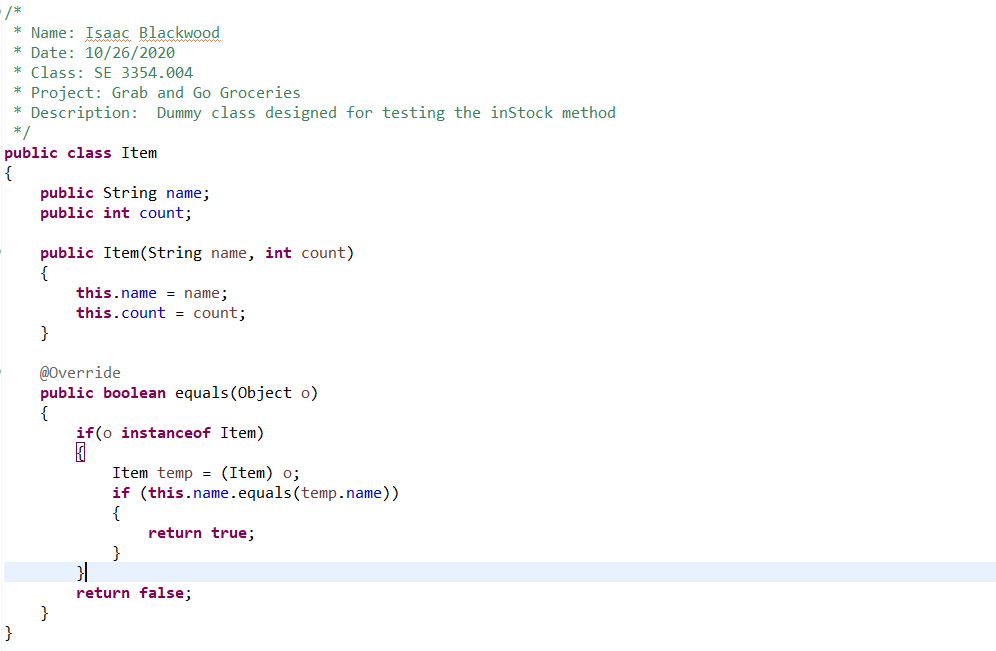
iOS / Android Publishing Costs 30% of Revenue

# TESTING:

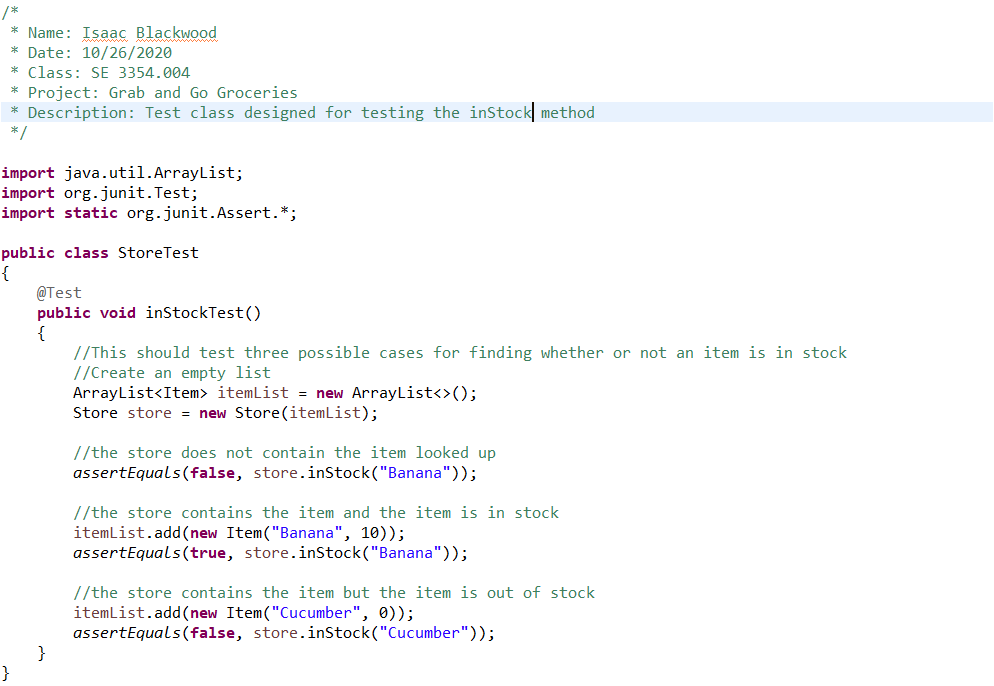
We used Black-Box Unit Testing to test the boolean inStock(String itemName) method. The method should return true if and only if the store contains the item, and the item is in stock. Equivalence partitioning yields the following test cases:

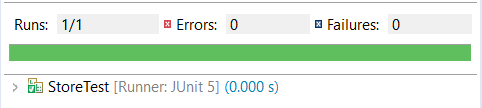
* Item does not exist
* Item exists but stock count is 0
* Item exists and stock count is greater than 0

**Code** (Check if Item is in Stock)



**Test Code**



**Results**

The code successfully passed all three test cases. The inStock function returned false for an item that did not exist. The inStock function returned true for an item that existed and was in stock. The inStock function returned false for an item that existed and was not in stock.

# COMPARISONS:

**Google Maps:**

Google Maps is a generic map application which uses GPS and data about locations to guide users to select locations with high accuracy according to [11]. Our application performs a similar task, providing users with directions to nearby stores and products in stores based on store location data. In fact, the location services in our application may be powered by Google Maps or a similar map application. However, while similar in function, our application is significantly different from other mapping applications because its purpose is unique, in that it directs users to specific items within a store. Additionally, while Google maps has traditionally integrated data from a variety of different sources, including crowdsourcing according to [11], we will be getting data directly from the stores we service. Finally, we also provide features such as item pricing, shopping list editing. These features are what sets our app apart from generic GPS map applications.

**Walmart Shopping:**

The Walmart app, Walmart Shopping & Grocery [12], is similar to our application in some ways, but in other ways it is very different. The Walmart shopping application also deals with grocery store purchases and allows customers to search for stores and items, but only within the Walmart database [12]. It is for Walmart stores, while our application would incorporate multiple grocery store chains. The Walmart app also is for customers purchasing items and picking it up, our application is just searching for items and showing where they are in the store. We have nothing to do with the customer purchasing the items. Our idea is similar to Walmart, but it has some differences that make it unique.

**Instacart**:

A prime competitor to our service could be Instacart [13]. However, Grab and Go Groceries and Instacart fulfill two different needs. Instacart is useful for those who know exactly what they want, and are capable of paying a little extra for that food to be provided to them in a convenient manner according to [13]. Grab and Go Groceries would not have an added fee per grocery order, and still allows the user to do their shopping themselves. This means that our application allows individuals to still deviate from their intended purchases at the last minute, if they so desire.

In effect, Instacart revolutionizes shopping, and is a product that changes the very core of the shopping experience [13]. Grab and Go Groceries, on the other hand, simply enhances the existing shopping experience by removing the struggle of trying to find products in an unfamiliar store and not knowing store stock.

# CONCLUSION:

Our team was very adamant on the “Grab and Go Groceries” idea because it was something that we all felt like we’d use ourselves. We wanted to create something that we saw value in creating, instead of something that had already been done before. Our team had some different ideas about where the project should end up going at the beginning, but eventually we all settled on one idea and this idea became the foundation of Grab and Go Groceries.

Our plan ended up changing along the way after our team discussed how feasible some of our ideas were. Originally in our plan for our application, we wanted to incorporate a voting system for users to vote on accuracy of information inside stores. Our team realized this feature could be abused, could be problematic, and could cause confusion for customers and stores alike. Therefore, our team decided to remove this idea in favor of stores only having the ability to change item location. While we did not stick to all the original ideas that were set out when we first created our project, we believe that the ideas we went with provide the best future for Grab and Go Groceries.

# PRESENTATION LINK:

https://web.microsoftstream.com/video/39b2494c-3e35-418e-a6a0-ee2c9e87629d

# References

[1] Mahal, D., 2014. The Programmer Productivity Paradox. [online] dzone.com. Available

at: <https://dzone.com/articles/programmer-productivity>.

[2] Rumyantseva, S., 2020. Average Hourly Rates For Offshore Development Services:

Software Development Costs Guide. [online] Qubit Labs. Available at: <https://qubit-labs.com/average-hourly-rates-offshore-development-services-software-development-costs-guide/>.

[3] Bureau of Labor Statistics. 2020. Occupational Outlook Handbook: Software

Developers. [online] Available at: <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>.

[4] NorthStarInc. n.d. What Is The Cost Of IT Support For Small Business?. [online]

Available at: <http://www.nssit.com/what-is-the-cost-of-it-support-for-small-business/> [Accessed 5 November 2020].

[5] Lahn, M., 2019. What's The Cost Of A Server For Small Business - Servermania.

[online] Servermania.com. Available at: <https://www.servermania.com/kb/articles/how-much-does-a-server-cost-for-a-small-business/> [Accessed 5 November 2020].

[6] Agarwal, A., 2015. Walmart Open API Developer Blog. [online]

Developer.walmartlabs.com. Available at: <<https://developer.walmartlabs.com/blog>>.

[7] RapidAPI. 2019. Feeditem-Target. [online] Available at:

<https://rapidapi.com/rpvicknair/api/feeditem-target/pricing> [Accessed 5 November 2020].

[8] Google Cloud. n.d. Pricing & Plans | Google Maps Platform | Google Cloud. [online]

Available at: <https://cloud.google.com/maps-platform/pricing/> [Accessed 5 November 2020].

[9] Microsoft.com. n.d. Windows Server 2019. [online] Available at:

<https://www.microsoft.com/en-us/windows-server/> [Accessed 5 November 2020].

[10] Developer.apple.com. 2020. Apple Developer Program Membership Fee Waivers.

[online] Available at: <https://developer.apple.com/support/membership-fee-waiver/> [Accessed 5 November 2020].

[11] Nightingale, R., 2017. How Does Google Maps Work?. [online] MakeUseOf. Available

at: <https://www.makeuseof.com/tag/technology-explained-google-maps-work/> [Accessed 5 November 2020].

[12] 2020. Walmart Mobile App. Walmart.

[13] Instacart. n.d. Instacart. [online] Available at: <https://www.instacart.com/> [Accessed 4

November 2020].