Github: <https://github.com/amadoo157/IntermedSoftware.git>

Requirements Analysis document:

Ravyn:

* Allows you to input grocery list (specific products or individual items)
* Searches nearby stores for available coupons
* Cross-matches coupons with grocery list provided
* Can suggested recommended items based off of previous purchases
* Encryption security for shopper protection

Sid:

* Maps out a path for most efficient route
  + Keeps track of grocery stores around the current location (using the list+coupons)
  + Can recommend the best store to purchase from given total with most matched coupons
  + Can make a custom path based on requirements (time spent driving, total cost, proximity)
* Dietary Preferences (intolerances, allergies, diet type)

Amado:

* Price Tracking history
* Price Tracking reasoning
* Shopping Budget
* Budget History

Sneha:

* Id verification for 18+ purchases
* User verification through email
* List sharing
* Virtual assistant

Non-functional: Encryption security for member protection

Hiwot:

* Allows you to sufficiently scan a coupon
* Share coupons
* Save coupons
* Connects you to grocery store membership logins

Hiwot

**Use case 1:** Shopper   
**Stakeholders and interests:** Shopper wants best way to save money and fast shopping

**Pre conditions:** Shopper is above the age of 18 and authorized

**Success guarantee:**   
Connects you to the grocery store based on shoppers set needs and cross-matches coupons with the grocery list provided while shopping.

**Main success scenario (Basic Flow):**

1. Shopper Searches nearby stores for available coupons
2. Shopper gets notified with the best grocery store nearby (proximity and needs)
3. the shopper adds the selected coupons to their cart.
4. Keeps track of purchases and coupons whilst shopping
5. Shopper continues to clip coupons while shopping
6. Shopper purchases cart
7. Purchase is now in order history
8. Shopper is satisfied with order

**Extensions:**

* 1. Shopper Searches nearby stores for available coupons
* 2. Shopper gets notified with the best grocery store nearby (proximity and needs)
* 3. the shopper adds the selected coupons to their cart.
* 4. Keeps track of purchases and coupons whilst shopping
* 5. Shopper clips coupons while shopping and purchases cart
* Purchase is now in order history
* 6. At the checkout, the shopper presents the coupons, but they are informed that the store does not accept digital coupons or that the selected coupons are not valid at that particular store.
* 7. Shopper pays without coupons
* 8. Shopper gets frustrated and leaves store with no discounts applied

**Special requirements:**

* + system will try to have a 2-3 second response for 96% of all the user interactions.
  + well-defined recovery process in case of data loss or system failures.
  + Offer a virtual assistant powered by AI that can help users with questions and problems

**Technology and data variation list:**

* + Must be able to handle barcodes (QR code/ UPC code)
  + High network connectivity
  + Should support different payment methods

**Frequency of Occurrence:** constant

**Use case 2:** Saves coupon   
**Stakeholders and interests:** Shopper wants to be able to successfully save coupons for future purchases

**Pre conditions:** Shopper has an account registered

**Success guarantee:**   
Account is registered information is applied in account history (coupon history ) and save for later

**Main success scenario :**

1. User goes on website
2. New users click login regular users click login
3. User puts information in as stated
4. User logs in and continues to shop
5. User remembers that there has been a coupon saved that user wanted to use in the future
6. Shopper goes into coupon history to look at used coupons and notices some compounds that have been reinstated
7. Shopper doesn't have necessary funds for this purchase so shopper saves coupon for next time
8. User checks out and is satisfied with the purchase

**Extensions:**

1. Shopper logs int to account
2. User logs in and continues to shop
3. User remembers that there has been a coupon saved that user wanted to use in the future
4. Shopper goes into coupon history to look at used coupons and notices some coupons that have been reinstated
5. Shopper notices all saved coupons lead you to a dead link and cannot be added to cart
6. Shopper cannot save coupons unless it is used on site

**Special requirements:** enter in coupon to save and use before expiration date

**Technology and data variation list:**

* + High network connectivity
  + Best interface

**Frequency of Occurrence:** once every two weeks whenever shopper is frequently shopping

**Use case 2:** Sharing coupon   
**Stakeholders and interests:** Shopper wants to be able to successfully sharing coupons for future purchases

**Pre conditions:** Shopper has an account and is a frequent shopper

**Success guarantee:**   
Account is registered information is private and authentication is required as well as password requirements

**Main success scenario :**

1. User goes on website
2. New users click login regular users click login/ User puts information in as stated
3. User logs in and continues to shop
4. User wants to send coupon to friend
5. You can easily copy and paste the link provided where it says “share coupon”
6. User sends it to friend and the text consists of a greeting link with coupon to log in or sign up

**Extensions:**

1. User goes on website
2. New users click login regular users click login/ User puts information in as stated
3. User logs in and continues to shop
4. User wants to send coupon to friend
5. You can easily copy and paste the link provided where it says “share coupon”
6. User sends it to friend and the text consists of a greeting link with coupon to log in or sign up
7. Friend logs in but link is inactive
8. All links if shared are inactive and shopper is dissatisfied

**Special requirements:** allow concerns to copy and paste the shared link

**Technology and data variation list:**

The app must quick responses and be able to track calculations -Ensure secure data storage and encryption of budget information.

**Frequency of Occurrence:** once every two weeks whenever shopper is frequently shopping

**Use case 3:** Connects you to grocery store membership logins  
**Stakeholders and interests:** Shopper wants to save time and efficient personnel

**Pre conditions:** shopper is a member and has an account with specified location

**Success guarantee:**   
Shipper can use in store membership account to access access services and maybe preferences

**Main success scenario :**

1. Shopper logs in to accounts
2. Shopper inputs deserted locations
3. Shopper allows for the in store membership linage
4. Shopper inputs information like member ID
5. System logs you in and the shopper can began to shop
6. User then applies any necessary coupons and proceeds to checkout
7. Grocery stores gets notified and prepares order

**Extensions:**

1. Shopper logs in to accounts
2. Shopper inputs deserted locations
3. Shopper allows for the in store membership linage
4. Shopper inputs information like member ID
5. System encounters an issues linking member to the right account
6. Shopper is now not allowed to have an efficient checkout process and has to go in store

**Special requirements:** System should have a quick response time and linking process. System should also be reliable. Assistant should be able to help with any issues

**Technology and data variation list:**

* + High availability
  + Real time communication
  + Geo location

**Frequency of Occurrence:** constant

* + Occasionally depending on users shop rate

Non functional:

GUI

* System has to have active handling, active link handling and data storage. By this we are ensuring shoppers data and information is securely stored and users can easily access. This makes sure links for saved coupons can be accessed and shared and account logins can be secured. System should also properly display information and user friendly application for better performance and response time

Amado

**Use Case 1: Price tracking history**

**Primary Actor:** Shopper

**Stakeholders and Interests:** The shopper wants to track the price history of products in order to make informed purchasing selections.

**Preconditions:**The shopper has a registered active account for the application and is logged in to that account.

**Success Guarantee (Postconditions):**The shopper can examine the past 18-24 month prices on the precise product they are looking for, as well as price fluctuations and comparisons.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the app and goes to the price tracking feature.
2. The application provides a search button to search for specific products.
3. They enter in said product, after that the application provides the avg price of the product in the last six months of the different store around the area.
4. The user picks a store like and are able to see the prices of the 6 month , 12 month , or the 18 -24 months.
5. They choose a specific spread and are able to see the fluctuation and the price day they open the app back till 24 months.

**Extensions (or Alternative Flows):**

2a) They forget the product

* If it’s something that they shop for on a weekly to daily basis, the application will provide a recommendation in the search bar.
* If not, they can either log out and come back when they remember at a later time.

**Special Requirements:**

* The search recommendation for the shopping search bar retains a history of up to one month.

**Technology and Data Variations List:**

* Search Functionality: The app can use different search methods for efficient product lookup.
* Graphing Tools: Employ tools for clear price trend visualization.

**Frequency of Occurrence:**

* This depends on the user since they will probably check this function out when prices are higher or when they are doing monthly shopping, which is why we predict they use it monthly.

**Use Case 2: Price tracking history**

**Primary Actor:** Shopper

**Stakeholders and Interests:** The Shopper wants to track price history and understand the reasons behind price fluctuations.

**Preconditions:**The shopper has a registered active account for the application and is logged in to that account.

**Success Guarantee (Postconditions):**The shopper can examine the past 18-24 month prices on a product and be able to get information on why the product is either increasing or decreased in price whether that reason be inflation, low supply,too much supply or others.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the app and goes to the price tracking feature.
2. The application provides a search button to search for specific products.
3. They enter in said product, after that the application provides the avg price of the product in the last six months of the different store around the area.
4. The user picks a store they like and can see the prices of the last 6 months.
5. They choose a specific spread and are able to see the fluctuation and the price day they opened then they click the button that gives more info which will show lines that are either red for increase or green for decrease of prices .
6. The person can click the line that appears based on the product and market and get the reason that the product increased or decreased.

**Extensions (or Alternative Flows):**

2a) They entered the said product and couldn’t find the product.

* They can still click for more information, which will reveal the explanation why the product was created..
* One of the reasons is if the product is discontinued, which means the company no longer sells it, and the information will disclose when it was discontinued and why.
* Another possibility is that the product is not in the database, in which case the extra information would be "sorry for the bad news, but this product is not in the database." This would only happen with products that aren't sold through the app or in local retailers.

**Special Requirements**

* The get more information function only works for information that can be found in the price tracking functionality.

**Technology and Data Variations List:**

* Colorized graphing tools: this provides the price trend visualization and color of it provides the whether the product increases or decreases in price.

**Frequency of Occurrence:**

* Our prediction is that it will be approximately every month or so, depending on the shoppers and how they perceive the pricing of products.

**Use Case 3: Shopping budget**

**Primary Actor**: Shopper

**Stakeholders and Interests:**Shoppers want to set and manage a shopping budget to control spending.

**Preconditions:** The Shopper has an active and registered account in the application.

**Success Guarantee (Postconditions):**The Shopper can successfully create, manage, and track their shopping budget within the application.

**Main Success Scenario (Basic Flow):**

1. The shopper opens the app and goes to the shop budget function .
2. The shopper starts to input their upper bound of their monthly budget in the list .
3. Additionally they also input what they will be shopping for .
4. After that the user goes to shop later on they end up shopping everything on the list and are on budget.

**Extension:**

4a) There budget exceeds the limit

* The shopper starts to buy things that weren’t on the list which lead them to exceed the budget, the application alerts them and offers options to adjust the cart or continue with the excess spending.

**Special Requirements:**

* The budget can be changed by the user anytime they want if they believe it to be too low for the time line that they adjust it to or they can even change the timeline .

**Technology and Data Variations List:**

* The app must handle real-time calculations of the remaining budget balance.
* Ensure secure data storage and encryption of budget information.

**Frequency of Occurrence:**

* The use case occurs as often as the Shopper sets or manages their shopping budget which is mostly monthly

**Use Case 4:Budget history**

**Primary Actor:** Shopper

**Stakeholders and Interests:** Shoppers want to see their budget in the past inorder to see how they can improve their shopping.

**Preconditions:**The Shopper must have a budget per month in past in order for the function

**Success Guarantee (Postconditions):** The shopper can successfully review their past budget and be able to see which months they exceeded and which months they didn’t.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the app and goes to the shop budget function .
2. Shopper then navigates to the "Budget History" feature.
3. The application displays a list of their past month budgets and tells them they have either exceeded or went under the budget.
4. From there the user can see there past budget and is able to plan out how they want to change

**Special Requirements:**

3a. Shopper can‘t find there budget history,

* This happens when they haven’t shopped through the application or have shopped through the application but didn’t create a budget, which will then lead them to the application to create a budget.

**Technology and Data Variations List:**

* Retrieve and Display Data: the application ensures to be efficient with showing the past month budgets and their status (exceeded or under).

**Frequency of Occurrence:**This feature would probably would be based on the shopper but our prediction is that it be used based on the the 6 month span

**Nonfunctional:**

Scalability

* **Adding Features with Ease:**We want to make it simple to incorporate new features into our program . This is a significant component for us because we want to ensure that our program can handle additional additions. By structuring our software in this way, we will be able to build it more frequently and do beta testing on particular features, which is the most essential approach to collect user feedback on our app.
* **Accommodating More Users:**The second concern is ensuring that our app can support additional users as it gains popularity. This is yet another crucial component of our app because we want everyone to use it, and in order to achieve our desired level of growth, we need to ensure that both the speed of the app doesn't decrease as more users use it and that the security of the app remains strong at all times.

Ravyn:

**USE CASE 1: Budgeting and Restricted Travel**

**Primary Actor:** shopper

**Stakeholders and interests:** shopper has a budget and does not have access to a car/transportation to their local store

**Preconditions:** The shopper has access to an IOT device to use the app, there are standard username/password recovery features for accessing an account, the shopper has a valid address where groceries can be delivered if necessary

**Success guarantee:** The shoppers grocery list is stored under recent purchases, the recommended algorithm adjusts according to most recent/frequent purchases, the shopper earns points on successful use of coupons, the shopper has their groceries delivered to the address specified in their account information

**Main success scenario:**

1. Shopper opens app and fills cart with groceries
2. These groceries can come from either recommendations, previous purchases or redirects using coupons presented to the shopper
3. The shopper can pick groceries and then have the app scan through each item to determine the best deals from a given store, the app will then recommend purchasing groceries through the store with the most savings
4. Shopper is given many payment options to complete transaction or they can use the stored payment method in their account information
5. Purchase is validated and groceries are delivered to specified address in shoppers account information, the groceries are then stored under previous purchases

**Extensions:**

4a. shopper is unable to complete purchase due to lack of funds or cannot access digital payment methods

1. The shopper can either save their grocery list to purchase and deliver at a later time and come back when digital purchasing methods are in order, or they can opt to pay with cash when the delivery is made
2. If the shopper still cannot pay with cash the groceries are taken back to the store until the shopper comes to pay for and collect them

**Special requirements:**

If the shopper still cannot pay with cash the groceries are taken back to the store and held for a maximum of 3 days or until the shopper comes to pay for and collect them, else they are returned back to inventory

**Technology and data variation list:**

2. Item identifier codes are grouped according to type of item with sub numbers (ie food -> dairy, grain, produce)

4. Payment options can either be stored and saved to shopper’s account or manually entered into app at time of checkout

**Frequency of occurrence:** This case will occur when a shopper does not have a form of transportation to the store or has a set budget/ not enough money

**USE CASE 2: Use personalized coupons**

**Primary Actor:** Shopper

**Stakeholders and Interests:** The primary interest of the shopper is to save as much money as possible while purchasing groceries using recommended coupons.

**Preconditions:** shopper lives near a variety of stores with separate coupon deals

**Success Guarantee:** The shopper will have a list of coupons that they can choose to add the items for in their cart, then these coupons will be applied depending on if the user decides to purchase from that certain store or not

**Main Success Scenario:**

1. The shopper opens the app and can navigate to the recommended coupons tab to view coupons they can add the items for to their list
2. They can fill their cart using recommended coupons generated specifically for them. They simply click to accept the coupons, and the items are added to their cart.
3. Depending on what store the user decides to purchase their groceries from it will apply those recommended coupons the user selected to their total price

**Extensions:**

3a. Coupon redemption failed

1. System returns an error stating that the coupon redemption failed and why (expired coupon, item out of stock etc)
2. The system can recommend another coupon with a similar product to the one the shopper originally wanted to redeem at all the local stores to the shopper

**Special Requirements:** produce specific error messages relating to coupon redemption failures and produce results for recommended coupons in 3 seconds or less

**Technology and Data Variation List:** there will be a specific recommended coupons tab to separate those from the general shopping part of the app

**Frequency of Occurrence:** This case will occur whenever a shopper wants to make use of the recommended coupons section of the app

**USE CASE 3: Finding best store based deals**

**Primary Actor:** Shopper

**Stakeholders and Interests:** The shopper wants to find the best store to purchase groceries from, taking into consideration location and deals

**Preconditions:** the shopper has an account with our app and can travel to a handful of nearby stores

**Success Guarantee:** The shopper will have a list of all the nearby stores and their coupons stacked against the shopper’s grocery list, with total savings from each store to determine where the shopper can get the best deal on their list of groceries

**Main Success Scenario:**

1. The shopper opens the app and beings inputting a grocery list filled with specific products or individual items (ie add dove body wash, campbells soup etc)
2. The app locates nearby stores depending on the saved address in the shopper’s account information and cross references the shopper’s list with the available coupons from each store
3. The app will display all the savings from each store against the shopper’s list to determine where is the most efficient place to purchase the list of groceries from

**Extensions:**

1a. An item a user wants is not available in a store

1. The app will locate all the nearby stores with that item and notify the shopper that however many of the items on their list is not available at a certain store

2a. One or more stores are closed

1. The app will notify the shopper that one or more stores are past operating hours and will ask if the shopper wants to continue shopping or save their grocery list and be notified at a specific time according to the app when all the stores are open to be able to accurately cross reference.

**Special Requirements:** produce specific error messages when items or not available or shoppers are outside of store hours

**Technology and Data Variation List:** item identifier is entered by keyboard, items marked as shopper groceries are stored in a list in the database

**Frequency of Occurrence:** This is a generic case where a shopper needs to buy groceries

**USE CASE 4: Utilizing Specific Recommended Items and Purchase History**

**Primary Actor:** Shopper

**Stakeholders and Interests:** Can suggested recommended items based off of previous purchases and purchase history

**Preconditions:** the user has made previous purchases using the app and has populated the purchase history database

**Success Guarantee:** The shopper should see a list of recommended items while shopping either in a sidebar or at the bottom of the application screen. The shopper should also be able to access a purchase history page where they can review and rebuy previously purchased groceries

**Main Success Scenario:**

1. The user logs into the application after having made previous purchases
2. The app will now be populated with personalized recommendations based off of previous purchases as well as a purchase history to make it easier to shop and create a grocery list
3. This will speed up the time it takes for the shopper to make their purchases and successfully get their groceries

**Extensions:**

3a. Any of the items that were purchased even with coupons that need to be refunded or returned

1. If any item from the shopper’s grocery list is damaged or needs to be returned for any reason all refund processes will be handled in the app no later than 3 weeks after purchasing the item, with special return instructions provided depending on the item

**Special Requirements:** secure authentication for account usage and items with certain restrictions (listed in later use cases) will be handled at the start of list making process

**Technology and Data Variation List:** item identifier is entered by keyboard, items marked as shopper groceries are stored in a list in the database

**Frequency of Occurrence:** This is a standard occurring event that will account for over 85% of all use cases

**Nonfunctional Requirement:**

* **Encryption security for shopper protection:** When a shopper makes an account on the app they will need to link a valid email address to confirm identity not only when creating the account but also intermittently when the app decides to do random authentication checks. Also, with identifying personal information such as I.D cards being stored in the system databases for age restricted items there will be end to end encryption to ensure all personal information is stored securely. Other personal information such as addresses for delivery and location based services as well as phone numbers will be kept secure under account information. If a shopper wishes to make any changes to this information they will need to follow certain steps with their email address to receive a link to a secure portal to update any information.

Sneha:

**Use Case 1: List Sharing**

Primary Actor: Shopper

Stakeholders and Interests:

- Shopper: Wants to create and share shopping lists with other users

- Recipient User: Receives and views shared shopping lists

Preconditions:

- The shopper is logged into their account

- The shopper has created a shopping list that they want to share

Success Guarantee:

- The shopper successfully shares a shopping list with another user, and the recipient can view it

Main Success Scenario:

1. The shopper navigates to their shopping list

2. The shopper selects the option to share the list

3. The shopper provides the email address or username of the recipient user

4. The system sends an invitation to the recipient user

5. The recipient user receives the invitation and accepts it

6. The system grants the recipient user access to the shared shopping list

Extensions:

- 5a. If the recipient user declines the invitation, sharing is not enabled

- 6a. If the shopper revokes sharing access, the recipient user loses access to the list

- 6b. If the recipient user removes a list, the list remains with the shopper only

Special Requirements:

- Access control and permissions management for shared lists through security

- Notification system for sending and receiving sharing invitations

Technology and Data Variation List:

- User account management and authentication

- Notification and email services for sending invitations

Frequency of Occurrence:

- Whenever a shopper wants to share a shopping list with another user

**Use Case 2: User Verification Through Email**

Primary Actor: Shopper

Stakeholders and Interests:

- Shopper: Wants to create a account securely

- System Admin: Needs to ensure user account security

Preconditions:

- The shopper is wants to register for a new account

Success Guarantee:

- If the shopper's email is successfully verified it allows them to create an account

Main Success Scenario:

1. The shopper provides their email address during the registration process

2. The system sends a verification link to the provided email address

3. The shopper clicks the verification link

4. The system verifies the email address

5. If the email is verified, the shopper can proceed with creating their account

Extensions:

- 5a. If the email verification fails, the system informs the shopper on the account creating page and restricts account creation

Special Requirements:

- Sending email verification links

- Verification link expiration and reset procedures

Technology and Data Variation List:

- Email sending and verification mechanisms

- Secure storage of email addresses

Frequency of Occurrence:

- Whenever a new shopper registers for an account

**Use Case 3: Virtual Assistant to Help with Purchases**

Primary Actor: Shopper

Stakeholders and Interests:

- Shopper: Wants assistance in finding and using coupons efficiently

- Virtual Assistant Provider: A third-party service providing the functionality

Preconditions:

- The shopper has enabled the virtual assistant feature

Success Guarantee:

- The virtual assistant successfully assists the shopper with their queries

Main Success Scenario:

1. The shopper activates the virtual assistant within the application

2. The virtual assistant asks the shopper about their shopping preferences

3. The virtual assistant provides personalized coupon recommendations

4. The shopper can choose to use or discard the recommendations

Extensions:

- None

Special Requirements:

- Privacy concerns with AI through security

Technology and Data Variation List:

- Integration with a virtual assistant API or service

Frequency of Occurrence:

- Occurs whenever the shopper uses the virtual assistant feature

**Use Case 4: ID Verification for 18+ and 21+ Purchases**

Primary Actor: Shopper

Stakeholders and Interests:

- Shopper: Wants to purchase age-restricted products

- Store Staff: Needs to verify the shopper's age to ensure legalities

Preconditions:

- The shopper has selected an age-restricted product for purchase

Success Guarantee:

- The shopper's age is successfully verified with the ID, allowing them to purchase the products

Main Success Scenario:

1. The shopper selects an age-restricted product

2. The system prompts the shopper for age verification

3. The shopper provides their government ID with a picture

4. The system verifies the picture and shopper's age

5. If the shopper is of legal age, the purchase is allowed

Extensions:

- 5a. If the age verification fails, the system informs the shopper and denies the purchase

Special Requirements:

- Shopper needs to take picture of ID for verification process (camera feature)

Technology and Data Variation List:

- Age verification APIs or methods

- Store and manage user birthdate data securely

- Identity verification services if necessary

Frequency of Occurrence:

- Occurs whenever a shopper attempts to purchase an age-restricted product

**Non-functional requirement:** Backup and Recovery

This plays a critical role in ensuring the resilience and continuity of our software. It involves implementing a robust system for routinely backing up essential data, including user profiles, shopping lists, coupon information, and purchase histories. This safeguard helps protect against data loss caused by unexpected events like system failures, data corruption, or other unforeseen issues. Furthermore, the recovery component of this requirement ensures that, in the event of a system failure or data loss, we can quickly restore the software to a previous state, minimizing downtime and data loss for our users. This not only enhances the reliability of our platform but also provides users with the peace of mind that their information and preferences are secure and can be recovered in case of unforeseen circumstances.

Sid:

**Use Case:** Dietary Preferences Filter for Grocery Shopping

**Primary Actor:** Shopper

**Stakeholders and Interests:**

* Shopper: Wants to quickly sort meals to see which ones match their dietary needs.

**Preconditions:**

* Shopper has a registered account and is logged in.

**Success Guarantee (Postconditions):**

* The shopper can filter and view groceries that match their dietary needs..

**Main Success Scenario (Basic Flow):**

1. Shopper opens the grocery shopping app.
2. They navigate to the product search or browse section.
3. The application has a "Dietary Preferences" filter.
4. Shopper selects the "Dietary Preferences" filter.
5. The application displays a list of frequently used preferences.
6. Shopper selects one or more dietary preferences from the list.
7. The application updates the product list based on the dietary filters.
8. Shopper can refine their search by selecting additional filters.
9. Shopper can add products to their shopping cart for mapping.

**Extensions (or Alternative Flows):**

1. Shopper changes dietary preferences:
   1. If the shopper's dietary choices change, they can return to the filter and make changes.
2. No products found for selected preferences:
   1. If there are no products available, the application should inform the shopper and provide alternatives.

**Special Requirements:**

* An interface that is easy to use should be provided by the dietary preferences filter.
* Permit customers to store their dietary choices.
* Update the database of items to reflect dietary information.

**Technology and Data Variations List:**

* To make sure the filtering is done correctly, use the data and labels.
* Reduce the filter time by putting search algorithms into practice.

**Frequency of Occurrence:**

* Customers can choose dietary filters everytime they make a shopping list.
* Shoppers will most likely utilize the filter once in a while to match their shopping list with what they need.

**Use Case:** Grocery Store Locator Feature

**Primary Actor:** Shopper

**Stakeholders and Interests:**

* Shopper: Wants to easily find and keep track of nearby grocery stores.

**Preconditions:**

1. Shopper has a registered account and is logged in
2. Access to the shopper's location, whether it is done automatically or manually.

**Success Guarantee (Postconditions):**

* The shopper can keep track of grocery stores.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the grocery shopping app.
2. They navigate to the "Store Locator" or "Find Stores" section.
3. Display the shopper’s current location and allow them to search for other stores nearby.
4. Shoppers either allow the app to access their location(automatically or manually).
5. The application then shows a list of stores within a specified radius.
6. Shoppers can view details about each store, including its name, address, opening hours, contact information, and ratings/reviews if available.
7. The shopper can view details about the store(name, address, hours, contact info, ratings, avg cost).ss
8. The shopper can favorite stores by moving their items up the list.
9. The application allows shoppers to filter the types of stores they see.
10. Shoppers can then get directions to the store.

**Extensions (or Alternative Flows):**

1. No stores found in the specified radius:
   1. If no grocery stores are found within the specified radius, the application should suggest expanding the search radius or changing their location.
2. Shopper updates their location:
   1. If the shopper moves to a new location, they can update their location, and the application will refresh the list of nearby stores accordingly.

**Special Requirements:**

* The store locator feature should provide a user-friendly map interface and an easy way to filter and sort nearby stores.
* Shoppers should be able to save their favorite stores for quick access in the future..

**Technology and Data Variations List:**

* Use GPS or location services to find the shopper's location.
* Implement a database of grocery store information.

**Frequency of Occurrence:**

* Shoppers will use this feature whenever they are at a new location.

**Use Case:** Find Cheapest Store for Grocery Shopping Feature

**Primary Actor:** Shopper

**Stakeholders and Interests:**

* Shopper: Wants to find the store where they can purchase all their groceries at the lowest total cost.

**Preconditions:**

* Shopper has a registered account and is logged in.
* Access to the shopper's location, whether it is done automatically or manually.
* A shopping list with the items.

**Success Guarantee (Postconditions):**

* The shopper knows the store with the lowest total cost for the items in their list.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the grocery shopping app.
2. They navigate to the "Find Cheapest Store” section.
3. The application prompts the shopper to enter their location.
4. Shoppers enter their location or allow the app to access their location.
5. The application then shows a map with the total price of the shopping list for each location.
6. Shoppers can sort the list of stores by cost or distance.
7. Shoppers can select a store from the list to view the individual prices of each item on their shopping list.

**Extensions (or Alternative Flows):**

1. No stores found in the specified location:
   1. If there are no grocery stores in the specified location, the application should notify the user.
2. Shopper updates their location:
   1. If the shopper moves, they can update their location which will refresh the grocery list.

**Special Requirements:**

* The feature should provide an interface for creating and managing a shopping list.
* The application should have a database of grocery prices for each store.

**Technology and Data Variations List:**

* Use GPS or location services to accurately determine the shopper's location.
* Maintain a database of grocery item prices for different stores and regularly update this data.

**Frequency of Occurrence:**

* Shoppers may use the "Find Cheapest Store" feature whenever they plan their grocery shopping, potentially multiple times a week.

**Use Case:** Map Custom Route for Grocery Shopping to Save Money

**Primary Actor:** Shopper

**Stakeholders and Interests:**

* **Shopper:** Wants an efficient route to buy groceries at the lowest cost.

**Preconditions:**

* Shopper has a registered account and is logged in..
* Access to the shopper's location, whether it is done automatically or manually.

**Success Guarantee (Postconditions):**

* The user can get a customized route that maximizes their savings over the shortest route.

**Main Success Scenario (Basic Flow):**

1. Shopper opens the grocery shopping app.
2. They access the "Map Custom Route" or "Optimize Shopping" feature.
3. The application prompts the shopper to enter or select their location for accurate routing.
4. Shoppers enter their location or allow the app to access their location.
5. The application displays a map with the shopper's location and nearby grocery stores.
6. Shoppers can view the details about each store.
7. Shoppers then can filter stores based on distance.
8. The app then finds the optimal route to travel so that they can purchase items at the lowest cost.
9. The app then displays the route in order of maximum savings.
10. Shoppers can follow the mapped route and mark items as purchased as they visit each store.

**Extensions (or Alternative Flows):**

1. No stores found in the specified location:
   1. The app should alert the user and recommend changing or broadening the search parameters.
2. Shopper updates their location:
   1. Customers can use the program to update their location if they move.

**Special Requirements:**

* An easy-to-use interface for seeing the best path should be provided via this feature.
* The application should regularly update and maintain a database of grocery item pricings and coupons for each store.
* Think about putting in place a route optimization algorithm that considers the shopper's current location, the stores they intend to visit in sequence, and costs.

**Technology and Data Variations List:**

* To locate the consumer precisely, use location services or GPS.
* Keep track of the pricing of groceries at various retailers in a database and update it frequently.

**Frequency of Occurrence:**

* When planning a supermarket shop, shoppers can utilize the "Map Custom Route" tool to optimize their route and save money. This can happen several times a week.

**Non functional requirement:** Response Time

* The reaction time requirement specifies the desired system performance in terms of interaction speed. It says that for 96% of user interactions, the application should aim for a response time of two to three seconds. This metric deals with how rapidly the system should react to customer inquiries. Delivering a smooth and responsive user experience depends on achieving this reaction time requirement.