Software Requirements Specification for Grocery Spending Tracker: An application to help users save on grocery spending.

Team 1, JARS
Jason Nam
Allan Fang
Ryan Yeh
Sawyer Tang

December 4, 2023

Contents

Revision History

Date	Version	Notes
06/10/23	1.0	Initial version of the SRS

1 Purpose of the Project

1.1 User Business

With the world currently facing record high inflation, cost-of-living is at the highest it has ever been. This affects all daily necessities but is especially true for food and groceries. As a whole, all households are affected but there is particular financial strain on those with lower-incomes. As a result, interest in personal finance has grown and become more important in peoples' everyday lives. To assist these individuals, we are developing an application that can help users better understand their spending habits and make smarter financial decisions. This application will allow users to take photos of grocery receipts and track their overall spending, analyze spending trends, and receive suggestions on cheaper alternatives for purchased grocery items. Overall, we believe this application will help users stay more informed and reduce grocery spending in the long-term.

1.2 Goals of the Project

- The created application will help users save money on groceries over time.
- The application will provide accurate spending data and suggestions to end users.

2 Stakeholders

2.1 Client

Both the client and customer will be the users of the system. These primary stakeholders are made up of low-income households of Hamilton and university students from McMaster who want to save money on groceries. The stakeholders will make decisions that influence the system through surveys, studies, and focus groups.

2.2 Customer

Outlined in the above Client section.

2.3 Other Stakeholders

- Grocery stores in the area
- Household members (people who the groceries are being purchased for)
- Mobile application stores (iOS app store, Google Play Store)
- Any individuals who wants to save on groceries

2.4 Hands-On Users of the Project

Low income households and individuals who are budget constrained on their groceries. The archetypal customer will have access to a device with a camera and have access to the internet. This entails adults or young adults who are responsible for making decisions on grocery items. Customers want to reduce the cost of their grocery items whilst maximizing the convenience of their grocery shopping trip.

- Low or single income households in Hamilton
  - Shops for household groceries.
  - Moderate experience with navigation mobile applications.
  - 30-70 years of age.
  - Has vehicle or some means to travel.
- Students of McMaster University
  - Shops for individual groceries
  - Adept experience with navigation mobile applications.
  - 20-30 years of age.
  - Has no vehicle or limited means to travel.

2.5 Personas

- John Bertuzzi
  - Age: 22
  - Favorite food: Chicken and Pasta

  - Nursing Student at McMaster University
  - Living in Ainslie Wood East.
  - Athlete.
  - Missing teeth.

- Alice Woll
  - Age: 51
  - Favorite food: Grilled Cheese
  - Mother of 2, Spouse of William Woll.
  - Accountant for tech company.
  - Living in Westdale.
  - Likes to watch Hockey.

2.6 Priorities Assigned to Users

All users will have equal priority within the system.

2.7 User Participation

Everyday users will provide their shopping data to the community database. This will be of no additional time commitment than they are already required to use the system at a base level.

From an elicitation perspective, a sample of users will be responsible for providing usability, look, and feel requirements feedback. This will be done at various stages in development.

2.8 Maintenance Users and Service Technicians

N/A

3 Mandated Constraints

3.1 Solution Constraints

Mobile Device
Description: User will interact with the system through an app on their mo-

bile device (smartphone).
Rationale: Mobile devices are widely used and ubiquitous. While at a grocery store, users may need access to the system.
Fit Criterion: User interface will run entirely on a mobile device.

Internet Access
Description: User must have internet access.
Rationale: System servers and databases are used to store collective user data and process that data. User's devices are required to communicate with the servers and databases via internet access.
Fit Criterion: User interface will have internet access.

Device with Camera
Description: User must be equipped with a device that has a camera. This should be quite common as most smartphones have a camera.
Rationale: Users will be using the system while they are out on their grocery trips. Thus, the system needs to have a quick and easy method of data input. This also makes it easier for those who have their hands full or do not have the finger mobility to effectively type on their phone keyboards.
Fit Criterion: System must have access to a mobile camera.

3.2 Implementation Environment of the Current System

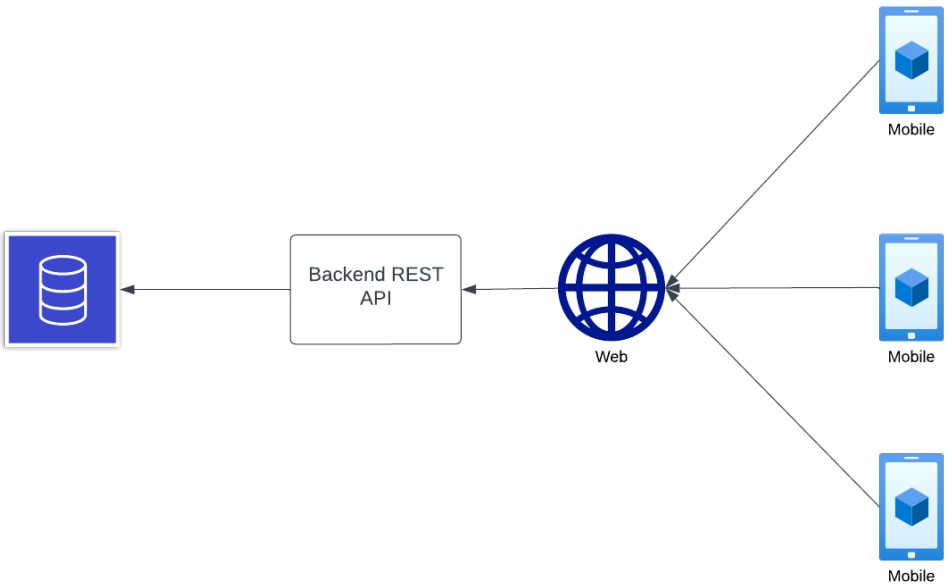

Figure 1: Implementation Environment

3.3 Partner or Collaborative Applications

N/A

3.4 Off-the-Shelf Software

The system requires the following off-the-shelf software:

- Tesseract OCR.
- Flutter for Front-end.
- NodeJS for Back-end.
- Express REST API.

- PostgreSQL as Relational Database.

3.5 Anticipated Workplace Environment

N/A

3.6 Schedule Constraints

The schedule for this project is in adherance with the milestones implemented by our course instructor. Found here.

3.7 Budget Constraints

The project budget is $750 paid for directly by the developers.

3.8 Enterprise Constraints

N/A

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Term	Definition
Consumer	A person who purchases goods and is a user of our solution
User	A term synonymous with consumer
System	A term referring to the entirety of the solution/application
UI	Acronym for user interface
UX	Acronym for user experience
MVVM	Model-View-ViewModel software design pattern to separate program logic and user interface controls

Table 1: Naming Conventions and Terminology

5 Relevant Facts And Assumptions

5.1 Relevant Facts

1. The cost of groceries in Hamilton and across Canada is steadily increasing, making it challenging for households to manage their budgets effectively.

2. Grocery prices can vary based on store, location, and region.

3. People have diverse spending habits when it comes to groceries. Some prefer brand-name products, while others prefer generic brands for cost savings.

4. Shoppers have varying dietary preferences, including vegetarian, vegan, gluten-free, and other specialized diets.

5. The prices of certain fruits, vegetables, and seasonal items can fluctuate throughout the year, impacting the overall grocery bill.

6. Shoppers often seek discounts, promotions, and coupons to reduce their grocery expenses.

7. Many grocery stores offer loyalty programs or rewards cards that provide discounts and cashback incentives to frequent shoppers.

8. Some shoppers prefer to buy groceries in bulk to save money in the long run, while others may opt for smaller, more frequent purchases.

5.2 Business Rules

N/A

5.3 Assumptions

1. It is assumed that users will provide access to their personal spending data, location data, and historical shopping information voluntarily.

2. It is assumed that users will follow the recommendations and budgeting plans provided by the app.

3. It is assumed that users will have access to devices with optical sensors and visual display components for image capture and data presentation.

4. It is assumed that the app's performance will meet user expectations and not disrupt their daily lives.

5. It is assumed that following its recommendations and budgeting plans will result in tangible reductions in grocery spending for users.

6 The Scope of the Work

6.1 The Current Situation

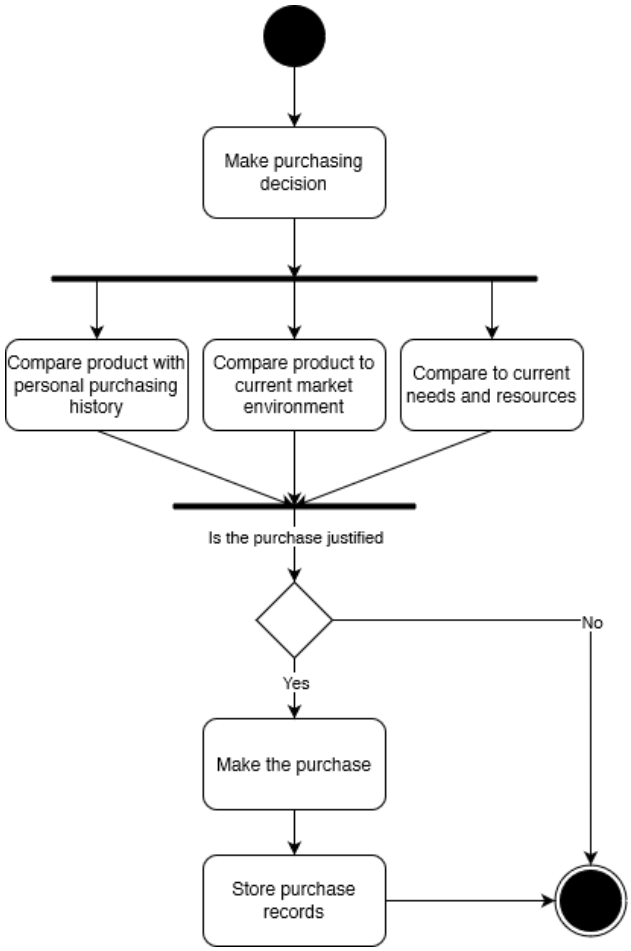

Figure 2: Current Situation Activity Diagram

6.2 The Context of the Work

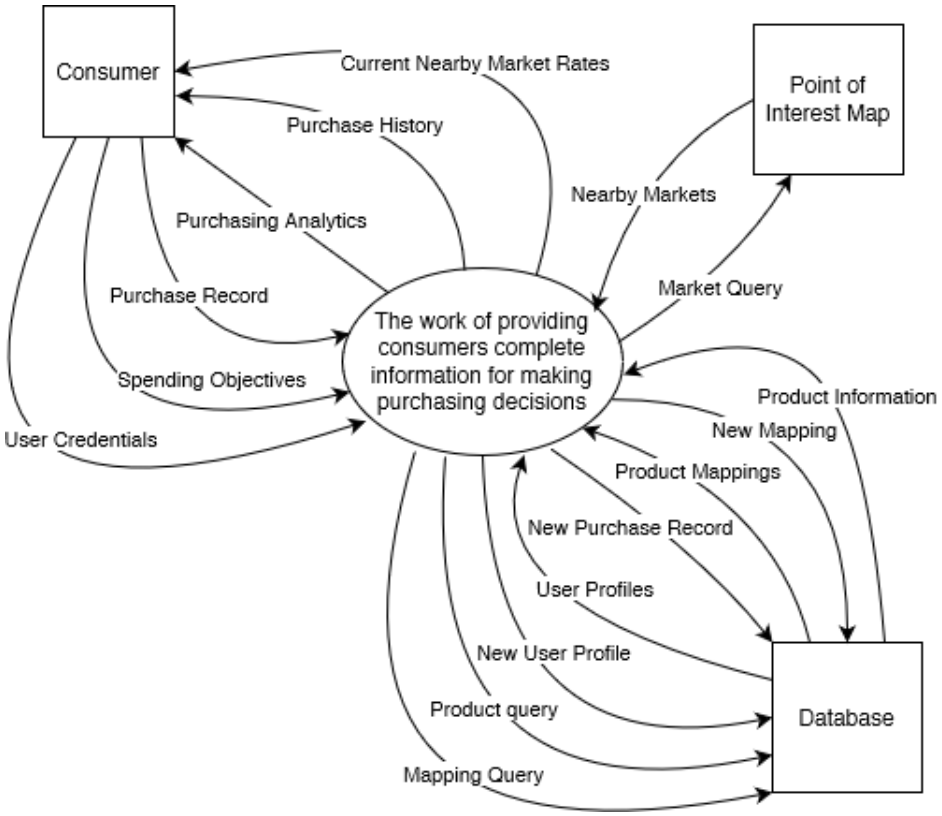

Figure 3: Work Context Model

6.3 Work Partitioning

Event Name	**Input/output**	**Summary of BUC**
User inputs user profile credentials	User Credentials (in)	Initialize user profile from received credentials
Consumer inputs spending objectives	Spending objectives (in)	Records the objectives and the owner

Consumer inputs purchase record	Purchase record (in)	Records the purchase and the owner of the purchase
User requests purchasing analytics	Purchasing analytics (out)	Report compiled and computed information from available data for the user profile
User requests personal purchase history	Purchase history (out)	Report compiled user purchasing history
User requests information on a product	Current nearby market rates (out)	Report market rates for a product that is accessible to the user
Compiling market information relevant to a user	Nearby markets (in)	Record information for accessible markets based on criteria
Request information on markets	Market query (out)	Send request for market information under a set of specified parameters
Database transmits product information	Product information (out)	Record information on product
New mapping received	Mapping Data (out)	Send new good/service mapping to database
Database transmits requested mapping data	Product mappings (in)	Record the product mappings
Send purchase record to database	New purchase record (out)	Send purchase record to database
Database transmits requested user profile data	User profiles (in)	Record the user profile data
Send user profile data	New user profile (out)	Send a user profile to database
Send product query to database	Product query (out)	Request relevant data based on product query from database

Send mapping query to database	Mapping query (out)	Request relevant data based on mapping query from database

Table 2: Business Event List

6.4 Specifying a Business Use Case (BUC)

Title: Create a new profile
Trigger: User creates a new profile
Pre-condition: User has application running
Outcome:

1. User creates an profile identified by a username and a password
2. System creates new profile entry in database

Title: Log in to user profile
Trigger: User submits profile credentials
Pre-condition: Profile exists in database
Outcome:

1. System checks if profile credentials exists
2. If exists, system allows loads user profile
3. If not exists, systems shows error message

Title: Record user objectives
Trigger: User submits personal objectives
Pre-condition: User is logged into profile
Outcome:

1. System stores user budgeting objectives of budget with associated product or product groups

Title: Input purchase records
Trigger: User submits image or manual entry of purchase record
Pre-condition: User is logged into profile
Outcome:

1. System reads and translates records to Strings
2. System maps input records to database objects
3. If mapping does not exist, make guesses and prompt user for feedback
4. Save records to database

Title: Create new mapping feedback
Trigger: Product mapping is not found in database
Pre-condition: Product record has been submitted
Outcome:

1. Create mapping guesses using language model
2. Prompt user with guesses for feedback
3. User inputs feedback
4. Process and screen user feedback
5. If mapping is eligible, create new mapping entry in database

Title: Create user profile analytics
Trigger: User requests purchasing analytics
Pre-condition: User is logged into profile
Outcome:

1. Retrieve user data from database
2. Transform user profile data into graphics
3. Display graphics in application

Title: Retrieve profile purchase history
Trigger: User requests purchase history
Pre-condition: User is logged into profile
Outcome:

1. Retrieve user purchase history from database
2. Transform user purchase history into graphics
3. Display graphics in application

Title: Retrieve product information
Trigger: User requests product information
Pre-condition: User is logged into profile and has current location information set
Outcome:

1. Retrieve user purchase history from database relevant to the product
2. Retrieve markets accessible to the user from map service
3. Retrieve product information belonging to the accessible markets from database
4. Transform data into graphics
5. Display graphics in application

Title: Set location information
Trigger: User sets location information
Pre-condition: User is logged into profile and location information is available
Outcome:

1. User sets initial location
2. User sets accessibility radius
3. Information is stored locally on device

7 Business Data Model and Data Dictionary

7.1 Business Data Model

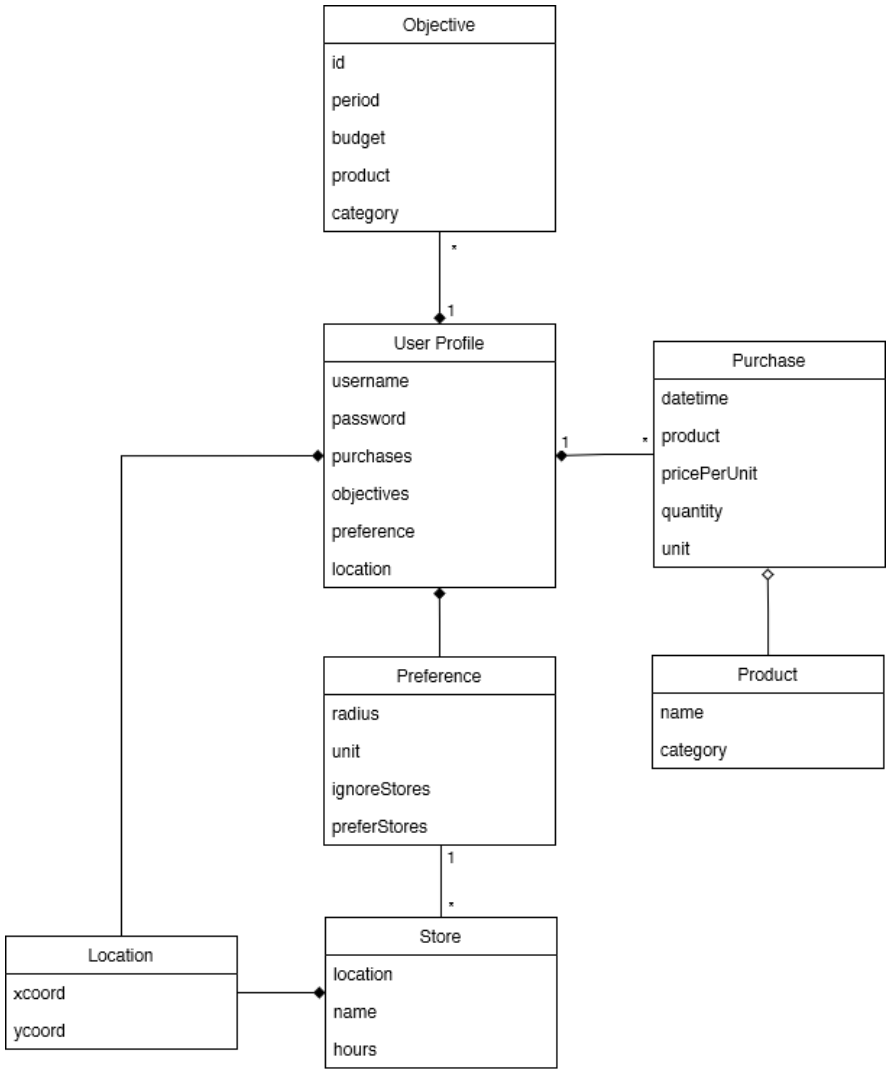

Figure 4: Class Diagram

7.2 Data Dictionary

Name	Content	Type
User Profile	Username + Password + Purchases + Objectives + User Preference + User Location	Class
Objective	Objective Id + Period + Budget + Target Product + Target Category	Class
Purchase	Datetime + Purchased Product + Price Per Unit + Quantity + Unit	Class
Product	Product Name + Category	Class
Preference	Accessible Radius + Ignore Stores + Prefer Stores + Prefer Units	Class
Store	Store Location + Store Name	Class
Location	X-Coordinate + Y-Coordinate	Class
Username	*Unique String*	Attribute
Password	*String*	Attribute
Purchases	*List of Purchase class*	Attribute
Objectives	*List of Objective class*	Attribute
User Preference	*Preference class*	Attribute
User Location	*Location class*	Attribute
Objective Id	*String*	Attribute
Period	*String: day/week/month/year*	Attribute
Budget	*Currency amount measured by Double to 2 decimals*	Attribute
Target Product	*String product name*	Attribute
Target Category	*String category name*	Attribute
Datetime	*YYYY-MM-DD HH:MM:SS*	Attribute
Purchased Product	*String Product name*	Attribute
Price Per Unit	*Int price per unit of product*	Attribute
Unit	*String*	Attribute
Quantity	*Double*	Attribute
Product Name	*String*	Attribute
Category	*String*	Attribute

Accessible Radius	*Integer*	Attribute
Ignore Stores	*List of Stores to ignore*	Attribute
Prefer Stores	*List of Stores to prioritize*	Attribute
Prefer Units	*String: metric/imperial*	Attribute
Store Location	*Location class*	Attribute
Store Name	*String*	Attribute
X Coordinate	*Double*	Attribute
Y Coordinate	*Double*	Attribute

Table 3: Business Event List

8 The Scope of the Product

8.1 Product Boundary

Figure 5: Use Case Diagram

8.2 Product Use Case Table

PUC No	PUC Name	Actor/s	Input/Output
1	Register for an account	User, System	User account data (in), Created user account (out)
2	Sign into application	User, System	User account data (in), Sign in authentication (out)
3	Enter spending goals	User	User spending preferences (in)
4	Enter location preferences	User	User location preferences (in)
5	Enter budgeting preferences	User	User budgeting preferences (in)
6	Enter receipt data	User	Photo of receipt (in)
7	Record user inputs	System	Inputs stored in database (out)
8	Process receipt	System	Receipt data stored in database (out)
9	Map receipt data to database	System	Individual product data properly stored in database (out)
10	Display analyzed spending data	System	Visualized spending data (out)
11	Provide budgeting suggestion	System	Budgeting suggestion (out)
12	Suggest cheaper grocery alternatives	System	Item, price, and location of cheaper alternative (out)

Table 4: Product Use Case Table

8.3 Individual Product Use Cases (PUC's)

Title: Register for an account
Trigger: User wants to create an account for the application
Pre-condition: User has necessary information requested by the system
Outcome:

1. User enters their email and password when prompted and submits the data.
2. System validates the data.
3. If valid, system stores user data in the database and creates the account.
4. If invalid, system prompts user with an error.

Title: Sign into application
Trigger: User wants to sign into their account
Pre-condition: User has an account for the application
Outcome:

1. User enters their email and password when prompted and submits the data.
2. System looks for any accounts with matching credentials.
3. If a match is found, system authentication is successful and user is granted application access.
4. If no match is found, system authentication fails and user is prompted with an error.

Title: Enter spending goals
Trigger: User wants to enter their spending goals for the application
Pre-condition: User is signed into an account
Outcome:

1. User enters information regarding their spending objectives and outcomes for application.
2. System records the entered spending goals into the database.

Title: Enter location preferences
Trigger: User wants to enter their location preferences for the application to consider
Pre-condition: User is signed into an account
Outcome:

1. User enters information regarding their current location and preferred area of coverage for the app.
2. User selects stores they prefer and stores they want to ignore.
3. System records the entered location preferences into the database.

Title: Enter budgeting preferences
Trigger: User wants to enter their budgeting preferences for the application to consider
Pre-condition: User is signed into an account
Outcome:

1. User enters price range and budgeting preferences for the application to consider.
2. System records the entered budgeting preferences into the database.

Title: Enter receipt data
Trigger: User wants to enter a receipt to add new spending data
Pre-condition: User is signed into an account and has a receipt to use
Outcome:

1. User takes a photo of the receipt.
2. System processes data from the receipt *(see Individual Product Use Case for "Process receipt")* and stores data in the database *(see Individual Product Use Case for "Map receipt data to database")*.

Title: Process receipt
Trigger: User enters a photo of a receipt to be analyzed by the application
Pre-condition: The receipt photo has already been taken and submitted to the application
Outcome:

1. System searches the receipt looking for purchase date, time, and location information.
2. System searches for products purchased and their prices.
3. Products are mapped *(see Individual Product Use Case for "Map receipt data to database")* and purchase data is recorded in the database.

Title: Map receipt data to database
Trigger: User enters a photo of a receipt to be analyzed by the application
Pre-condition: The receipt has already been processed by the system
Outcome:

1. System goes through each found product on the receipt and checks if it exists in the database.
2. If an entry in the database exists for the product, the date, time, location, and purchase price are recorded in that entry.
3. If no entry for the product exists in the database, an entry is created and the date, time, location, and purchase price are recorded.

Title: Display analyzed spending data
Trigger: User wants to check their spending data
Pre-condition: The user is logged in and has spending data already entered into the application
Outcome:

1. The system checks the user's purchases in the database.
2. The system displays a visual representation of the user's purchase history and data is provided on what was purchased and when.

Title: Provide budgeting suggestion
Trigger: User exceeds budgeting preferences submitted
Pre-condition: The user is logged in and has entered their budgeting preferences into the application
Outcome:

1. The system analyzes purchase history and compares it to the user's budgeting preferences

2. The system suggests items to stop purchasing and/or cheaper grocery alternatives if they exist *(see Individual Product Use Case for "Suggest cheaper grocery alternatives").*

Title: Suggest cheaper grocery alternatives
Trigger: A recently purchased item by the user is purchased at a cheaper price
Pre-condition: A database entry for the recently purchased item and spending data exist
Outcome:

1. The system analyzes recent purchase history of the item and purchase location from different users.

2. If the location is within the current user's location preferences, the item, price and location are prompted to the user.

3. If the location is not within the current user's location preferences, no prompt is created.

9 Functional Requirements

9.1 Functional Requirements

Requirement ID: FR1
Use Case: 1
Description: The system must minimize user effort and reduce the likelihood of user abandonment during registration by allowing easy access and controls to user registration.
Rationale: A registration interface is crucial for reducing the entry barrier

for new users, enhancing user satisfaction, and decreasing the likelihood of registration abandonment.
Fit Criterion: The interface should allow users to enter their email and password and complete the registration in no more than three steps, with each step clearly delineated.

Requirement ID: FR2
Use Case: 2
Description: The system must validate user credentials (email and password) when a user attempts to sign in.
Rationale: Secure and reliable authentication is essential for protecting user accounts and sensitive data from unauthorized access and complying with data protection regulations.
Fit Criterion: Users with valid credentials must be granted access to the application, while users with invalid credentials must be denied access and receive a clear, non-ambiguous error message regarding the specific authentication failure within 3 seconds of their attempt.

Requirement ID: FR3
Use Case: 6, 7, 8, 9
Description: The system must support the scanning and recognition of receipt data from photo images. The system must accurately recognize, extract, and parse text and numerical data from various receipt formats and layouts.
Rationale: Enabling users to upload photo images of receipts and converting these into digital spending data significantly enhances user convenience by eliminating manually entering data.
Fit Criterion: The system must successfully recognize and extract data from uploaded receipt photos with an accuracy rate over 90% under typical lighting conditions.

Requirement ID: FR4
Use Case: 4, 7
Description: The system must offer an interface that allows users to specify and update their current physical location and set a desired radius for area coverage.
Rationale: Enabling users to input and adjust their location and area of

coverage is vital for providing customized and relevant information, services, or offers specific to their locale.
Fit Criterion: Users must be able to set and update their location and preferred area of coverage, with changes reflected in the system within 3 seconds.

Requirement ID: FR5
Use Case: 3, 7
Description: The system must provide structured input fields where users can input and modify their grocery objectives and desired outcomes.
Rationale: Enabling users to set and alter their spending goals directly in the application allows improved grocery planning and encourages healthier grocery spending habits.
Fit Criterion: Users must be able to change their spending goals with the updated information displayed in less than a second after input.

Requirement ID: FR6
Use Case: 5, 7
Description: The system must offer a feature where users can specify their preferred price ranges for goods and set comprehensive budgeting parameters.
Rationale: Providing users with the ability to define their price ranges and budgeting preferences is necessary for personalized shopping experience to match user constraints.
Fit Criterion: The system must reflect user's preferences in filtering and recommending products with an accuracy rate of at least 90%.

Requirement ID: FR7
Use Case: 8, 9
Description: The system must process each item listed on a receipt by identifying each item details. It must update the user's spending log in the database.
Rationale: Accurately capturing detailed purchase information from receipts is important for informing users with a comprehensive overview of their grocery spending habits.
Fit Criterion: The system should accurately extract and parse grocery item details from receipts, achieving a data accuracy of at least 90%.

Requirement ID: FR8
Use Case: 11, 12
Description: The system must analyze the user's purchase history and compare it to the budgeting preferences the user has submitted. It must suggest actions the user can take to meet their budgeting goals.
Rationale: This functionality allows users to make informed decisions and encourages them towards making well informed grocery purchases.
Fit Criterion: Suggestions should be considered relevant and useful in user feedback surveys, with at least 70% positive response rate.

Requirement ID: FR9
Use Case: 12
Description: The system must actively monitor prices of items that a user has purchased within the last 10 days. When it detects that an item is available at a lower price, it should promptly notify the user.
Rationale: Notifying users about lower price alternatives on grocery items they recently purchased enlightens users about potential cost saving alternatives and emphasizes the usefulness of the application.
Fit Criterion: Notifications on lower price alternatives must be delivered to the user within 24 hours of detection.

Requirement ID: FR10
Use Case: 10
Description: The system must fetch and display a visual representation of the user's grocery purchase history, including details on items purchased and their respective timestamps.
Rationale: Providing a visual representation of grocery purchase history converts unprocessed data into a comprehensive and easily understandable format for users.
Fit Criterion: The system should present the user's purchase history data with over 95% accuracy.

Requirement ID: FR11
Use Case: 6, 7, 8, 9
Description: The system must allow users to verify and edit grocery items that were not accurately scanned and recognized by the system.
Rationale: It is essential for the system to ensure data accuracy and com-

pleteness of tracking of user grocery purchases. Although scanning and recognizing image text using OCR have become advanced, it is impossible to ensure 100% accuracy of results all the time. Therefore, allowing users to verify and update items will enhance reliability of data.
Fit Criterion: The system should reflect changes made by users on the purchase items log in less than 3 seconds. Also, user feedback on ease of edit functions should be positive, with more than 80% of users rating it easy.

	PUC1	PUC2	PUC3	PUC4	PUC5	PUC6	PUC7	PUC8	PUC9	PUC10	PUC11	PUC12
FR1	X											
FR2		X										
FR3						X	X	X	X			
FR4				X			X					
FR5			X				X					
FR6					X		X					
FR7								X	X			
FR8											X	X
FR9												X
FR10										X		

Figure 6: Traceability Matrix with Functional Requirements and Product Use Cases

9.2 Added Section - Formal Specification

In regards to identifying functions for project:

- **User must provide data to application.**
  For all users 'a' who are regular users:

$$\forall a \in \text{users} \quad (a.\text{role} = \text{“user”} \Rightarrow a.\text{inputDataToApplication}() \Rightarrow \text{true})$$

- **All data must be stored in database.**
  For all data 'd' generated or used by the application:

$$\forall d \quad (d.\text{storeInDatabase}() \Rightarrow \text{true})$$

- **All user data must be encrypted in database.**
  For all user data 'ud' stored in the database:

$$\forall ud \quad (ud.\text{encryptInDatabase}() \Rightarrow \text{true})$$

10 Look and Feel Requirements

In this section, any references to an *E#* refers to elicitation notes in Appendix B. *E#* corresponds to the label for the relevant question.

10.1 Appearance Requirements

- **AR1**: The app should have a consistent colour scheme throughout as requested in *E1*. This would be validated through an 80% acceptance rate from stakeholder surveys.
- **AR2**: App icons should convey the related feature to improve intuitiveness as requested in *E1*. This would be validated with an 85% success rate when asking stakeholders to pair features with icon during testing.
- **AR3**: The user interface should avoid clutter and features should only take up 85% at most of available screen space as requested in *E1*.

10.2 Style Requirements

- **SR1**: App features and options on different pages should be in consistent locations for intuitiveness as requested in *E1*. This would be validated through an 80% acceptance rate from stakeholder surveys.
- **SR2**: Features and options should be grouped based on relation to each other to improve intuitiveness with an 80% acceptance rate for application organization from stakeholder surveys.
- **SR3**: Indicators and messages should be present to convey information such as for loading or errors as requested in *E3*.

11 Usability and Humanity Requirements

In this section, any references to an *E#* refers to elicitation notes in Appendix B. *E#* corresponds to the label for the relevant question.

11.1 Ease of Use Requirements

- **EUR1**: The app should save user data and state when closed in order to minimize user input.
- **EUR2**: The app should allow users to undo or cancel changes made.
- **EUR3**: Navigation options should always be available regardless of the application page as requested in *E6*.

11.2 Personalization and Internationalization Requirements

- **PIR1**: English will be supported.
- **PIR2**: The app should have dark and light mode colour schemes as requested in *E7*.
- **PIR3**: The app should allow users to enable/disable external notifications as requested in *E7*.

11.3 Learning Requirements

- **LR1**: The app should have an FAQ or Help page to assist users as requested in *E5*.
- **LR2**: The app should have a tutorial to assist users in learning the available features as requested in *E5*.
- **LR3**: Users should be able to understand core functionality of the app within 15 minutes as requested in *E4*.

11.4 Understandability and Politeness Requirements

- **UPR1**: The app should use common language found in the English dictionary.
- **UPR2**: The app should use inoffensive language based on a 90% acceptance rate from stakeholder surveys.

11.5 Accessibility Requirements

- **ACR1**: The app should allow users to change text size as requested in *E7*.
- **ACR2**: The app should have multiple colour schemes to help those with visual impairment as requested in *E7*.

12 Performance Requirements

In this section, any references to an *E#* refers to elicitation notes in Appendix B. *E#* corresponds to the label for the relevant question.

12.1 Speed and Latency Requirements

- **SLR1**: The app should take on average 1 second to complete backend tasks and at worst should never exceed 12 seconds as elicited in *E8*.
- **SLR2**: The navigation of the app should be smooth, taking no more than 1 second to render and handle page changes as elicited in *E8*.

12.2 Safety-Critical Requirements

- N/A

12.3 Precision or Accuracy Requirements

- **PAR1**: The app should be able to correctly read user inputs with over 90% accuracy as elicited in *E10*.
- **PAR2**: The app should return spending data to the user with correctness over 95% of the intended value as elicited in *E10*.

12.4 Robustness or Fault-Tolerance Requirements

- **RFR1**: The app should have proper error-handling against improper user input to account for experimentation as elicited in *E6*.

12.5 Capacity Requirements

- **CR1**: The app should take up no more than 1GB of space as elicited in *E11*.

12.6 Scalability or Extensibility Requirements

- N/A

12.7 Longevity Requirements

- N/A

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

- **EPER1** The application is expected to be compatible with smart mobile devices running Android.
- **EPER2** The application is expected to be compatible with smart mobile devices running IOS.
- **EPER3** The application is expected to be compatible with tablet devices running iPadOS or Android.

13.2 Wider Environment Requirements

- **WER1**: The application is designed to operate on devices equipped with camera peripherals.
- **WER2**: The application is designed to operate on devices that have access to the internet.

13.3 Requirements for Interfacing with Adjacent Systems

- **IASR1**: The application shall work with iOS 16.
- **IASR2**: The application shall work with Android 13.

13.4 Productization Requirements

- N/A

13.5 Release Requirements

- **RER1**: The application shall undergo thorough testing and quality control procedures prior to each application releases and updates. The project will aim for 80% code coverage.
- **RER2**: The application shall identify and resolve defects or issues prior to application releases and updates.
- **RER3**: The application will be subject to a Final Demonstration from March 18 to March 29 of 2024.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

- **MTR1**: All back-end maintenance and upgrades should be completed between the hours of 12:00am and 5:00am.
- **MTR2**: Front-end maintenance should not be coupled with backend maintenance.
- **MTR3**: Maintenance should be able to be completed by the non-original developers.

14.2 Supportability Requirements

- **SPR1**: Users should be able to operate and navigate the system without any additional manual or instruction.
- **SPR2**: There will be a function built into the system for users to report usage issues or bugs to the developers.

14.3 Adaptability Requirements

- **APR1**: System must operate on IOS and Android Devices.
- **APR2**: System is intended to be used when the user is out of their home.

15 Security Requirements

15.1 Access Requirements

- **ACR1**: The application shall require acceptance of terms and services to allow access to application's database at all times.

15.2 Integrity Requirements

- **INR1**: The application shall ensure data integrity by using encryption mechanisms for preventing unauthorized modifications.
- **INR2**: The application shall be protected from actors with harmful intentions.

15.3 Privacy Requirements

- **PRR1**: The application shall maintain proper data anonymization and sanitation techniques.
- **PRR2**: The application shall require explicit user consent to collect, process, and share user data.

15.4 Audit Requirements

- **AUR1**: The application shall maintain audit logs that capture user activities and system events which can be reviewed by authorized system administrators.

15.5 Immunity Requirements

- N/A

16 Cultural Requirements

In this section, any references to an *E#* refers to elicitation notes in Appendix B. *E#* corresponds to the label for the relevant question.

16.1 Cultural Requirements

- **CUR1:** The app should allow users to use the metric system, imperial system, or a combination of the two (Canadian) as requested in *E12* to accommodate a diverse user base and regional preferences.
- **CUR2:** The app should have full functionality for uncommon purchase items as requested in *E12* to ensure the software is usable in the real-life dynamic market environment
- **CUR3:** The app should differentiate different currencies used (specifically USD and CAD) as requested in *E12* to accommodate regional preferences.

17 Compliance Requirements

17.1 Legal Requirements

- **LR1:** The app should comply with the app distribution platform guidelines: Google Play Store for Android, App Store for iOS
- **LR2:** The app should comply with local privacy laws (PIPEDA for Canada) [1]

17.2 Standards Compliance Requirements

N/A

18 Open Issues

There are currently no open issues for this project.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

There exists some existing products that address the problem we are also addressing.

flashfood matches users with food that is nearing their expiration and are marked down to prevent waste and provide cheaper prices.

ibotta helps users track their spending and offer coupons to those wanting to save money.

19.2 Reusable Components

- **node-tesseract-ocr**: for reading labels or receipts.
- **PostgreSQL**: for storing relational data.
- **flutter**: framework for mobile app development.

19.3 Products That Can Be Copied

There are no known products that can legally be copied to form a solution for this problem.

20 New Problems

20.1 Effects on the Current Environment

- The introduction of the new system for analyzing user spending may require integration with existing systems or databases to ensure a seamless flow of data. Changes to data connectors and interfaces may be necessary as a result, which could have an impact on the current environment.
- Depending on the number of users and the volume of data processed, there may be an increased network load on the current environment. Network performance may be impacted, necessitating capacity scaling.

20.2 Effects on the Installed Systems

- The new system should be compatible with the operating systems, browsers, and hardware configurations used by existing users. Compatibility problems could cause system outages or user access problems.
- Integration with existing security systems is important to ensure that user data remains secure. Data breaches or unauthorised access could result from any security integration flaws.

20.3 Potential User Problems

- Users may experience difficulties adjusting to the new system, particularly if it significantly alters the user interface or process. To deal with this problem, training and support materials should be made accessible.
- Users might encounter problems during the migration of their existing spending data to the new system. Ensuring a smooth transition and data integrity is essential.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

- The anticipated implementation environment might not entirely satisfy the new product's hardware or software requirements. This could result in system instability or reduced performance.

20.5 Follow-Up Problems

- After the initial implementation, updates, bug fixes, and maintenance will be required. If issues are not addressed promptly, users may grow dissatisfied.

21 Tasks

21.1 Project Planning

Deliverable	Deadline
Hazard Analysis Revision 0	October 20
V&V Plan Revision 0	November 3
Proof of Concept Demonstration	November 13-24

Table 5: Development Phase 1

Deliverable	Deadline
Design Document Revision 0	January 17
Revision 0 Demonstration	February 5-February 16

Table 6: Development Phase 2

Deliverable	Deadline
V&V Report Revision 0	March 6
Final Demonstration (Revision 1)	March 16-March 29
Final Documentation (Revision 1)	April 4

Table 7: Development Phase 3

21.2 Planning of the Development Phases

Development is being split into three phases based on an overarching goal for each: the Proof of Concept build, the Revision 0 build, and the Revision 1 build.

The first development phase has a focus on initial documentation and implementation. Rather than building a full app at the phase, development will involve familiarization with the technology and constructing core features to prove the concept is feasible. This phase will end with the Proof of Concept Demonstration where the feasibility will be proven to stakeholders.

The second phase of development will focus on design of the application and constructing the Revision 0 build of the application. With the initial functionality from the first development phase as a starting point, the actual app structure will be built around it. This will include more UI/UX development as well as additional features that are needed for the Revision 0 version. This phase ends with the Revision 0 Demonstration where stakeholders can see the first iteration of the full application and provide feedback.

The third phase of development is focused on reviewing Revision 0 and iterating on feedback from the Revision 0 Demonstration. Here, documentation will be reviewed and finalized based on any changes made during development. Furthermore, no new large-scale features will be implemented. Instead, development will be restricted to improving the Revision 0 version and implementing feedback from stakeholders. This phase will end with the Revision 1 Demonstration and submission of the finalized Revision 1 Documentation.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

N/A

22.2 Data That Has to be Modified or Translated for the New System

N/A

23 Costs

The team currently has no intention of spending money and aims to utilize free software and technology for this project.

24 User Documentation and Training

24.1 User Documentation Requirements

- The application shall give detailed instructions on how to use the application in user manuals or guides. Registration, sign-in, data entry, budgeting, spending analysis, and any other key features should all be included in this documentation.
- The application shall offer a FAQ section that answers to frequent user issues and inquiries. This can enable users to quickly find answers to their problems without the need for outside assistance.
- The application shall offer a troubleshooting guide that details common problems and how to fix them. This should include technical difficulties, login issues, and data sync issues.
- The application shall inform users on how to stay up to speed with software updates and new features. Include release notes that describe the additions and modifications done in each update.

24.2 Training Requirements

- The application shall provide on-boarding content that new users can access when they initially register for the application, such as tutorial videos or guidelines. These resources should to make it easier for users to learn the basic features.
- The application shall clearly communicate the available user support channels, such as a customer support email, chat, or phone number. Users should be made aware of where to look for assistance when they run into problems.

- The application shall provide a system that will allow users to comment on the documentation and training materials. The training procedure and documentation quality can be continually improved with the help of this feedback.

25 Waiting Room

At the moment, there are no requirements or features being excluded from the initial version.

26 Ideas for Solution

Some initial ideas for the implementation that the team came up with are as follows:

- Application will utilize *node-rsa* to handle encryption and security for user data.
- Front-end will use MVVM architecture.
- Natural language processing will be used to assist mapping from receipts to the database.

Appendix A — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. **What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.**

   For this application, many skills will be required in order to ensure this project's success. This includes a combination of project specific knowledge as well as a few less technical skills. Together, these skills will ensure the smooth development of our application but will also likely prove useful in the future as we move forward with our careers.

   (a) Working knowledge of *Flutter* development and front-end design
   (b) Knowledge working with *Node.js* backends and REST APIs
   (c) Skills related to the use of the *Tesseract-OCR* engine
   (d) Working knowledge of *PostgreSQL*
   (e) Experience working with Natural Language Processing or Machine Learning
   (f) Writing proper software documentation
   (g) Familiarity and experience with software testing/coverage
   (h) Presentation and communication skills for elicitation and demos

2. **For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?**

For the above skills, there are many approaches that could be taken to learning and developing them. In terms of skills *(a)* to *(e)*, these would be categorized as development oriented skills. These would involve knowledge that could directly apply to the technology being employed for this capstone project. Since these are each popular technologies in their own right, many resources could be accessed to learn about them. Watching *YouTube* videos and tutorials would be one resource that could be used to strengthen and learn these skills. *YouTube* tutorials often provide more practical applications for the skill in question and the site has content that covers a wide variety of topics. Additionally, official documentation could be read in order to see how the original developers intend the technology to be used. This would provide a more in-depth understanding of each aspect of the technology. There are also plenty of third-party online articles and forums that could be referenced when learning these types of skills. One last approach that could be taken is referencing open-source projects using these technologies to see how they are used in real development.

Skills *(f)* and *(g)* are technical but less directly involved with the coding portion of development. Despite this, there are still plenty of resources that could be used to learn and familiarize ourselves with them. One place to look could be previous coursework, in particular, *SFWRENG 3RA3* and *SFWRENG 3S03* notes could be used to reference documentation and software testing respectively. Furthermore, resources for these topics also exist on *YouTube* which could be leveraged in addition to third-party articles. One last resource could also be to leverage group work and collaboration. This would provide active engagement as well as opportunity for feedback to improve these skills.

Lastly, skill *(h)* is a general skill that is important for all engineers to be comfortable with. One approach to mastering this skill could be practical applications, including but not limited to holding mock presentations or focus groups with the team. Practicing presentations and communication is one of the easiest ways to help develop this skill. Moreover, there are many seminars and courses online that help teach strong communication practices such as ones on *Coursera* or *LinkedIn Learning*.

With these in consideration, each team member has listed their chosen skills, their chosen approaches from above, and why they chose to

pursue these skills below.

Ryan Yeh
Chosen Skills: (a), (f), (g), (h)
At the beginning of this project, I was assigned as the lead for documentation and testing. As such, I would like to take the opportunity to strengthen my understanding of these skills in order to meet my team's expectations. Outside of this project though, I believe these skills will be very helpful regardless of what type of developer I end up becoming. Additionally, I wanted to strengthen my understanding and skills in front-end development. UI design is a software field that I have an interest in but lack sufficient experience currently. Therefore, I would love to take this opportunity to learn more about it and strengthen my understanding of it. Lastly, I chose communication skills as something to focus on due to my belief that it is something I could improve upon. Communication and presentation skills will always be universally needed and putting an emphasis on it now would greatly help me in the future.

My approach for developing my documentation and software testing skills will be to refer to previous year course notes. This will help me to reinforce the material I have already learned and refresh myself on topics I may have forgotten. I also plan to read third-party articles as a supplement to learn more about certain topics that may not have been covered in as much detail in class. For *Flutter* and front-end design, I plan to watch *YouTube* tutorials in order to get more a hands-on experience with the technology. I also plan to reference documentation and articles in order to further my learning beyond the videos watched. Finally, I will leverage my group in order to strengthen my communication skills. I will try and place myself into positions that force me to practice communication such as leading meetings, as well as practicing presentations with the rest of my team.

Sawyer Tang
Chosen Skills: (a), (b), (d), (h)
For this project, I was assigned to backend development. In following this, I am choosing to improve my abilities in *NodeJS* and *SQL*. I have used some of these technologies in brevity over my past internship positions but I am looking to deepen my experience with them further

with the hopes of bringing this experience to future personal projects and jobs. In addition to wanting to improve my backend skills, I also want to dive into front end. More specifically, mobile development in *Flutter*. This is something that I have always had an interest in learning and this project seems to me like a perfect opportunity. Finally, I believe that being able to effectively communicate your ideas is an invaluable skill and it is something that I am always looking to improve for myself and career.

My approach to gaining these frontend and backend skills will be to consume public documentation, online forums, and *YouTube* video tutorials. For communication skills, I practice them every day during meetings or when working with team members. I also look to improve these skills during our team presentations and demos.

Jason Nam

Chosen Skills: (a), (c), (e), (g)

During the early stages of project team formation and idea generation, I was tasked with the job of undertaking feasibility studies related to implementing OCR for the team's project solution. At that point in time, I had extensive experience working with Machine Learning for Natural Language Processing. However, I had never had the opportunity to learn proper Machine Learning models related to OCR. I am eager to learn the workings behind OCR and how we can implement it in our project. I am also eager to further enhance my ML for NLP skills. Similar to my previous point, I also had the opportunity to experience full-stack development involving web development. However, I never had a chance to learn Mobile App development, especially front-end app development. I am confident I can utilize my skills from web development to transfer over to app development and improve from there on. Finally, I want a tangible skill I can acquire during the duration of this project. I believe that skills related to software testing and coverage will be very valuable later on in life.

My approach to acquiring OCR and Machine Learning Recognition skills will involve extensively reading over the user manual for *Tesseract OCR* package. Fortunately, there are many sources of information related to Tesseract OCR online in many different platforms, including *YouTube*, *Tesseract Github Repository*, and official user manuals. For

app development using *Flutter*, I want to start from the fundamentals. First, I will endeavor to learn *Dart* programming language, which is the programming language behind *Flutter*. Then, I will refer to many open source *Flutter* development projects to familiarize myself with official *Flutter* development. For software testing and coverage related skills, I will refer to previous course 3S03 - Software Testing course to re-acclimatize myself on the topic.

Allan Fang
Chosen Skills: (a), (c), (f), (h)
I will be the primary developer for the frontend and OCR portions of this project. For these skills (*a* and *c*), I will pursue both the approach of following tutorials and the approach of reading documentation. From past experiences, I have found that there is a knowledge aspect and execution aspect to skills similar to the aforementioned. While documentation is usually very comprehensive, I feel that it is necessary to supplement the learning process with additional material. Following or just watching a tutorial of these skills in action are usually good ways to fill in the knowledge gaps.

For skills *f* and *h* that all groups members will be responsible for, I will be pursuing the approach of practice and feedback (which also includes referring to past course work). I believe that for these skills, practical application is most important. Actively engaging in these skills, along with consistent feedback and improvement, will allow me and my group mates to develop a deeper understanding of these skills in the personal application context. Furthermore, feedback from multiple sources can introduce broader perspectives previously unknown and will result in immediate improvements.

Added Section - Appendix B — Stakeholder Requirements Elicitation Notes

- **E1:** Consider your favourite or most used mobile apps, what elements of the UI stand out? What in your opinion makes an appealing user interface?
  - Minimize options to create a cleaner UI.

  - Utilization of easily accessible menu bars (like Instagram) to improve navigation UX.
  - Consistent and cohesive colour theme throughout.
  - Use of icons that clearly convey what the option is meant for.
  - Put options in consistent locations to minimize hand movements (easy to tap interface).

- **E2:** When you think of intuitive applications, what about the application makes the UI or UX intuitive?
  - Many apps borrow UI layouts from each other, helps with familiarity in terms of navigation.
  - Always accessible navigation options.
  - Proper organizing of features (unrelated features should be separated on the UI).
  - Swipe navigation and minimal long presses.
  - Saving state when closed to minimize user input on future uses.

- **E3:** How important do you consider application feedback and what feedback would you expect from an application? (i.e. loading indicators)
  - High importance to communicate errors and loading to users.
  - Should always convey when issues happen or something is being done.
  - When fetching information or other longer duration tasks, loading bar should be utilized.
  - Utilization of messages or pop ups if resources are unavailable or something goes wrong.

- **E4:** When learning a new app, how long would you consider a reasonable amount of time to consider an app easy-to-use?
  - Average acceptable time to consider an app intuitive would be to be able to figure out basic functionality within 15 minutes.

- **E5:** What features would you want to help with the learning experience of a new app?
  - The use of a tutorial on first launch to take users through the UI and different features.
  - The use of tooltips for certain options and features.
  - A button that users can press to access a Help menu or FAQ.
- **E6:** How do you usually go about learning a new application?
  - All participants said they tend to brute force applications in order to learn (i.e. clicking around and seeing how the application responds).
  - As an addition to this brute force approach, participants explained how useful an undo feature would be as well as always accessible navigation.
- **E7:** If you were using an application, what personalizations/accessibility settings would you consider a must?
  - Dark/Light mode.
  - Ability to change text sizes.
  - Multiple language support.
  - Allowing users to enable or disable notifications if used.
  - Alternate color palettes to help with those with visual impairment like color blindness.
- **E8:** Consider an app you use often, generally, how long would you say you wait for operations to complete? How long would you be willing to wait for an application before closing the app?
  - Ideally, regular operations should take a second or less.
  - At most, participants would wait an average of 12 seconds before closing the app or thinking something went wrong.
- **E9:** What kind of phone do you use? (Apple/Android/Other)
  - All participants used Apple devices.

- **E10:** On a scale of 1-10, for an app of this nature, how important would you consider accuracy of the system and data? How important would you consider responsiveness on the same scale?
  - All participants considered accuracy of the features and data to be the most important (10) due to the financial aspect of the app.
  - Responsiveness was considered important (8) in order to ensure a positive user experience. If the app does not feel good to use, most participants said they would find another app instead.
- **E11:** How much do you consider the size of an application you install? How big of an app would you consider reasonable?
  - Most participants said they did not pay much attention to the size of applications when installing them.
  - Participants said they would consider anything over 1GB unreasonable for an app of this nature.
- **E12:** For an app of this nature, what features or points of interest come to mind based on your identity as a Canadian?
  - Participant stated imperial and metric units are hard to navigate as Canada uses a unique mix. them.
  - Participant stated that culturally unique grocery items (such as Chinese vegetables) need to be included.
  - Participant stated that the app should consider USD vs CAD as purchases from the United States are common when living in Canada

Added Section - Appendix C — Bibliography

[1] Office of the Privacy Commissioner of Canada, "Seizing opportunity: Good privacy practices for developing mobile apps," Office of the Privacy Commissioner of Canada, https://www.priv.gc.ca/en/privacy-topics/technology/mobile-and-digital-devices/mobile-apps/gd_app_201210/ (accessed Oct. 4, 2023).