Software Requirements Specification

for

SmartShoppers System

Version 2.3 approved

Prepared by,

Amaan Vania, Areeba Abidi, Dong Jae Lee, Juan Leaniz, Khalid Othman, Mason D'Arcy

EECS4312 Project Group

November 14th 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team	2020-Nov-07	Initial Draft.	1.0
Team	2020-Nov-10	Modifications and Reorganization of text.	2.0
Team	2020-Nov-14	Review.	2.1
Juan Leaniz	2020-Nov-14	Review and formatting.	2.2
Team	2020-Nov-14	Final review.	2.3

1. Introduction

1.1 Purpose

The main purpose of this document is to provide an accurate and essential list of requirements necessary for the SmartShoppers system. The system specified within this document is an online shopping-aid that allows clients to view an aggregated list of stores close to their location, create a virtual shopping list and add items as they would in a physical setting. Compared to a traditional shopping system that simulates the shopping environment in a virtual setting, SmartShoppers will allow the customers to find products with the physical store of their choice more efficiently and with greater precision in terms of locating the item within the store. The system will allow the users to create a shopping list by searching for products, view the item's description and also see item suggestions based on the search history of other users for the specified store. Then the users will be presented with the list of items in the best sequence to find them starting at the front of the store to allow for improved routing. This system will require external services and applications in order to store the data for the virtual stores and the users. The user interface for this online system will be a graphical user interface (GUI) to clearly and effectively show the available functionalities.

1.2 Document Conventions

Within the document, the bolded word with numbering represents an important section that needs to be highlighted and sub sections are underlined with numbering or letters that appends the section numbering. Requirements in section 4 and 5 are bolded to bring attention to those reading the document.

Priorities in section 4 are as follows:

1 - Essential
2 - Important
3 - Desirable, but not critical

Other conventions such as acronyms used throughout the documents are listed in "Appendix A: Glossary" near the end of the document.

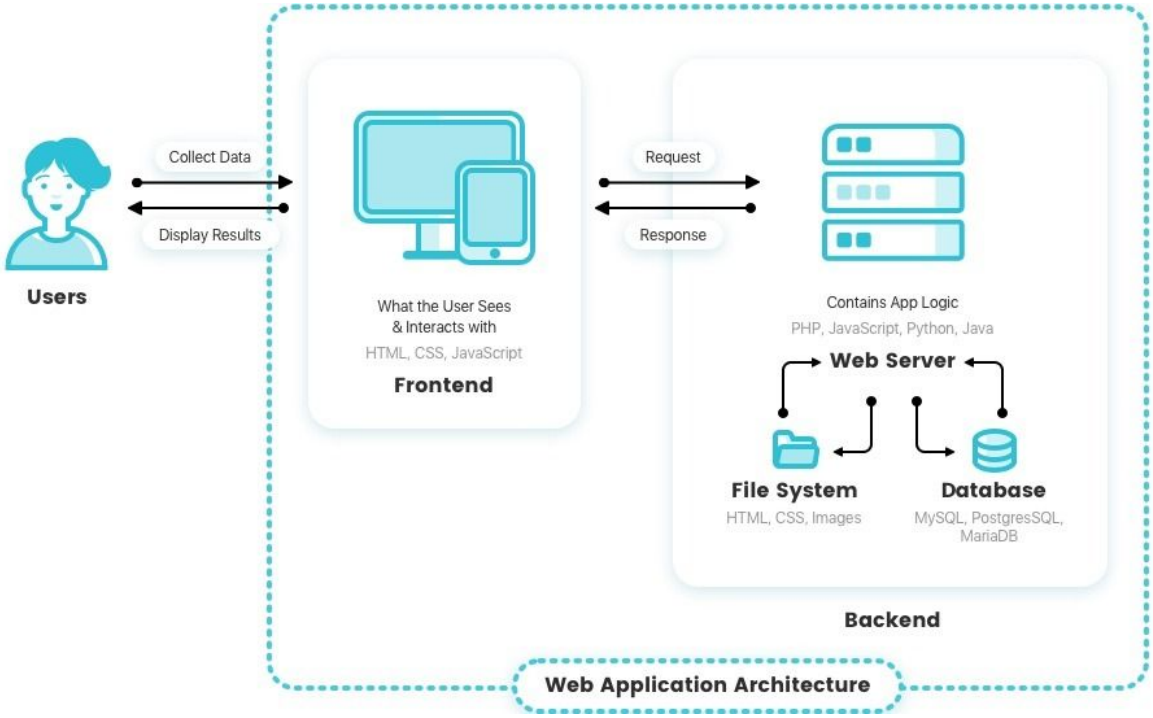
1.3 Intended Audience and Reading Suggestions

The primary audience of this document includes, but is not limited to: project managers, software architects and developers of the system, store owners and companies hosted on the system, and end users. This document may also interest project managers and developers of similar projects involving a virtual shopping component. The recommended order of reading is aligned with the order of this document and every group of audience will find it beneficial to read each section.

2. Overall Description

2.1 Product Perspective

SmartShoppers is stand-alone, Cloud based software as a service for ShoppersLand Inc. The objective of the software is to provide an enhanced experience for customers. To achieve this, the SmartShoppers provides a web application easily accessible to potential shoppers. The subsystems include: an interactive website hosted by a web-server, a back-end server, and a database. The web-server communicates to the back-end server through requests, and the back-end server returns with responses. The back-end server provides the general functionality of the application and is deployed on a cloud based server. It communicates directly with the database to store and retrieve key data.



Link: https://reinvently.com/wp-content/uploads/2019/08/scheme.jpg

2.2 Product Functions

The high level functionality includes:

- Customer users can sign up and register for an account
- Customer users can find all stores within 3km of a postal code
- Customer users can select a store as the their default store
- Customer users can create a virtual shopping list
- Having selected a store, customer users can search items by name to view their details including price and description

- Having selected a store, customer users can search items by category to view a list of all items in that category
- Having selected a store, customer users can search items by category to view a list of all items in that category
- Having selected a store and searched for an item or category, customer users can add an item and specify the quantity of that item to their list.
- Customer users can edit the quantity of an item in their list.
- Customer users can remove items from their list
- Given a specific item, the system can suggest a list of items based on the searches of other users who were interested in that specific item to the customer user. For example, users who searched for baseballs also searched for baseball bats.
- Given a customer user's shopping list, the system can generate the optimal order the customer should pick up the items at the respective store in the form of an ordered list.
- The application manager can search for a store in the system.
- The application manager user can add a store to the system.
- The application manager user can remove a store from the system.
- The application manager user can add a store manager account to a specific store.
- The application manager user can remove an existing store manager from a store.
- The application manager can update the details of a store including location and name.
- Store manager users can login.
- Store manager users can add categories to their respective store inventory.
- Store manager users can remove categories from store inventory.
- Store manager users can add new products to a category in their respective store.
- Store manager users can remove products from a category in their respective store.
- Store managers can update an item, including its name, price, description, and location.

2.3 User Classes and Characteristics

2.3.1 Users:

2.3.1.a Unregistered Users:

Unregistered users can be classified as users accessing the SmartShoppers application without a registered account. They are able to interact with the application but lack the ability to access key user functionality. They may choose store locations and search items. However, they are unable to create a shopping list. If they attempt to do so, they will be prompted by the application to register an account. To make full use of our application, unregistered users must sign up to make use of the functionality of our application.

2.3.1.b Registered Users:

Registered users can be classified as users who have signed up and are accessing the SmartShoppers application with a registered account. They have access to all the user based functionality of the application. This includes choosing a store

location, selecting and adding items to a shopping list and requesting an optimal sequence of shopping list items to pick up.

2.3.2 Managers:

2.3.2.a: Physical store manager

Physical store managers are regarded as managers of physical ShoppersLand Inc stores. Physical store managers are responsible for adding, editing and removing items for their corresponding store. As such, they are also responsible for updating the location of inventory that is moved within the store. Managers are also responsible for creating, editing and removing categories, as all items must be classified under some category.

2.3.3 System:

2.3.3.a: Application manager

Administrators are users which bear the responsibility of overlooking the SmartShoppers application. Some responsibilities include adding and removing store managers. The application managers are also able to add new stores, or remove existing stores.

2.4 Operating Environment

SmartShoppers is a cloud based application. The web-server is to be deployed on a cloud server. The database is also to be deployed via a cloud based database. For users, the application runs in a typical browser.

2.5 Design and Implementation Constraints

Accessibility:

The web application must be supported by popular browsers.

This includes:

- Windows: IE, Edge, Firefox, Chrome
- Mac: Safari, Firefox, Chrome
- Linux: Firefox, Chrome

Safety and Security:

The web-server should operate using HTTPS communication protocols. There must also exist Data Encryption (AES-256).

Interfaces:

The web-server and website must work in conjunction. They both must also work in conjunction with the database.

2.6 User Documentation

The SmartShoppers application will contain a list of frequently asked questions (FAQ), along with their corresponding answers linked to the main dashboard of the application. There will exist sections on the website, where tutorials are provided. In these tutorials, the functionality of the application can be made apparent and easily replicated. Finally a contact form and help-line will be provided for those requiring assistance. These resources will be linked to the main dashboard of the application.

2.7 Assumptions and Dependencies

Assumptions:

- Assuming users are using reasonable browser versions.
- Managers will keep the products and stock up to date.
- Managers will add,edit, and remove inventory.
- Administrators will add and remove stores.
- Administrators will add and remove managers.
- System is not involved in the payment process of items.

Dependencies:

- Service availability will be heavily dependent on the cloud environment provider
- Database licensing dependencies

3. External Interface Requirements

This section of the SRS describes the external interface requirements for the SmartShoppers system. Requirements for user, hardware, software and communication interfaces are defined.

3.1 User Interfaces

This subsection of the SRS describes the user interface requirements.

3.1.1 Main website and Home

All of the system's web pages accessed by the users must be HTML compliant and a guest user must be presented with a landing page. The website must be organized into specific sections that are easily accessed. At minimum, the website shall contain the following sections: Home, Login/Sign-up, Store Locations, Products, Contact Us, Help. A search bar capable of searching for specific products must always be visible to the user.

3.1.2 Store location page

The store location page must allow the user to enter a postal code, city or province to obtain a list of nearby store locations and inform the user if an invalid input was entered.

3.1.3 Login/Sign-up page

The user must be presented with a username and password field and the option to log into the system or sign-up for a new account. The sign-up page shall request the user to enter all necessary information.

3.1.4 Products page

The user must be presented with the selected store's products. Products should employ an organization method by categories for easier access to specific sets of items. The user should have the ability to modify the way that items are presented on the page, such as lists or thumbnails with pictures of the products.

3.1.5 Contact Us page

The user must be presented with all the necessary information to contact customer help and the home store associated with their profile. This contact information may include phone numbers, e-mail addresses and/or physical addresses for the stores.

3.1.6 Help page

The user must be presented with different problem categories such as billing or technical issues. Each sub-section should provide relevant information and suggest a solution to typical problems a user may experience with the system. A Frequently Asked Questions (FAQ) section should be included in this page.

3.1.7 Web-based administrative interface

An authenticated user with administrative permissions must be presented with a page that allows them to modify the system as required.

3.2 Hardware Interfaces

All server-side components of the system shall execute on server-class computers. Client-side components such as a mobile application or web browser shall execute on personal-class computers or mobile devices. The payment processing is outsourced to a third party, therefore, the SmartShoppers system has no real interaction with external hardware such as Point-of-Sale (POS) devices.

3.3 Software Interfaces

The SmartShoppers system has three main types of software interfaces:

- programmatic interface: the system shall communicate with external software via an HTTP-based REST API.
- caching layer interface: the system shall retrieve commonly accessed data from an in-memory cache for more efficient data access.
- database interface: the system will exchange information by retrieving and storing data from/into a database.

3.3.1 Description of administrative interfaces

Administrative users shall interface with the system via a web-based UI, REST API or Database.

- Input: Create, Retrieve, Update, Delete (CRUD) data from the system
- Output: CRUD system data

3.3.2 Implementation constraints

The SmartShoppers system database shall not be hosted or installed on the same device that serves HTTP requests (i.e. the web server)

3.4 Communications Interfaces

3.4.1 Client-server communications

The communication architecture shall follow a client-server model. Communications between the client and the server must utilize the HTTP Secure (HTTPS) protocol for additional security and must be REST compliant. User session data must be stored server-side (stateful sessions).

3.4.2 Database communications

The communication between the system's web server components and database shall be secured using Transport Layer Security (TLS)

4. System features

This section outlines all the features and functional requirements of the SmartShoppers system. Each feature and functional requirement has a specified priority as defined in Section 1.2.

4.1 Account Signup

4.1.1 Description and Priority

A customer user creates an account. The user must enter correct information (unique e-mail, strong password). The user will be instantiated in the system.
Priority: 1

4.1.2 Stimulus/Response Sequences

1. Users access the front-end through a supported personal device.
2. System displays landing page.
3. System prompts the user with account creation information fields.
4. Users enter required data into fields.

4.1.3 Functional Requirements

REQ-4.1.3.1: If the user visits the web portal of the front-end, the system displays fields for account creation including email and password. The front-end also displays information fields for logging in.
Priority: 1

REQ-4.1.3.2: If the user enters a non-alphanumeric string as a password into the account creation password field, then the software does not instantiate that user in the system and prompts them with an appropriate error message.
Priority: 1

REQ-4.1.3.3: If the user enters an alphanumeric string of characters less than eight characters long into the account creation password field, the software does not instantiate that user in the database and prompts them with an appropriate error message.
Priority: 1

REQ-4.1.3.4: If the user enters an invalid email into the account creation name field, the software does not instantiate that user in the database and prompts them with an error message.
Priority: 1

REQ-4.1.3.5: If the user enters a valid, unique email address and an alphanumeric password string of length eight or more into their respective fields, then the software instantiates that user in the system.
Priority: 1

REQ-4.1.3.6: If the software has instantiated the user in the database, the software prompts the user with an acknowledgement dialog, “Account created. Please login.”
Priority: 3

4.2. Customer User Login

4.2.1 Description and Priority

Users gain access to the system by entering their credentials.
Assumption: User has created an account.
Priority: 1

4.2.2 Stimulus/Response Sequences

1. User visits the front-end through a webportal.
2. Software displays the landing page and prompts the user to login.
3. User enters login information.
4. Software grants access and displays dashboard.

4.2.3 Functional Requirements

REQ-4.2.3.1: If the user visits the SmartShoppers webportal, the software prompts them with dialog fields for login including email and password fields. The software also displays account creation fields.
Priority: 1

REQ-4.2.3.2: If the user enters a valid alphanumeric password and email pair into their respective login fields, the software authenticates their login and loads the dashboard.
Priority: 1

REQ-4.2.3.3: If the user enters an invalid alphanumeric password and email pair into their respective login fields, the software does not authenticate the login and prompts the user with a dialog box and string, “Error.”
Priority: 2

4.3 Get and Set Shopping Location

4.3.1 Description and Priority

A user enters their location information and receives a list of relevant store locations.
Assumption: User has logged in.
Priority: 1

4.3.2 Stimulus/Response Sequences

1. Users enter their location information into respective information fields.
2. Software displays a list of relevant store locations to the user.
3. User selects their desired store location.
4. Users optionally set the current selection as their default store.

4.3.3 Functional Requirements

REQ-4.3.3.1: If the user enters a valid postal code, the system will output a list of proximate (3KM) shopping locations to that code.
Priority: 1

REQ-4.3.3.2: If the user enters a valid province and city, the system will output a list of shopping locations in that city.
Priority: 1

REQ-4.3.3.3: If the user selects a shopping location, the system will set it as their current shopping location.
Priority: 1

REQ-4.3.3.4: If the user selects their current shopping location as a default, the system will persist it as the default shopping location for future sessions.
Priority: 1

REQ-4.3.3.5: If the user enters an invalid postal code or province city combination, the system will prompt them with a message, “Error.”
Priority: 3

4.4 Create Virtual Shopping List

4.4.1 Description and Priority

The user can create a virtual shopping list by searching items offered by the selected store and adding them to a list. The user can search for an individual item or by category. The system can expose details about certain items and offer item suggestions to the user. Once a user completes their list, the system outputs it in a format that optimizes the path of the user through the selected store, and displays total price. The system will save the list, and overwrite a previous list.
Assumption: User has logged in. User has a store selected.
Priority: 1

4.4.2 Stimulus/Response Sequences

1. Users search for an item or category.
2. System outputs the item or item category list.
3. Users may view details of the item including price, quantity, and description.
4. User displays details.
5. Users may add an item to their list and specify the quantity.
6. System adds the item to their list with the specified quantity.
7. Users may remove an item from their list.
8. System updates the list by removing the item.
9. Users can optionally view suggested items based on their search.
10. System displays a list of suggested items.
11. The user chooses to complete and finish their list.
12. The system outputs the list as an optimal time-saving sequence for the user, in addition to the total cost.
13. The system stores the list for the customer (will overwrite a previous list)

4.4.3 Functional Requirements

REQ-4.4.3.1: If the user enters an existing item name in the store, the system will return the item with its details including descriptions, price and quantity.
Priority: 1

REQ-4.4.3.2: If the user searches for an item which does not exist in the store, the system will display the prompt “Sorry, that item is not available.”
Priority: 3

REQ-4.4.3.3: If the user selects an item returned by the system, the system will add that item to their virtual shopping list.
Priority: 1

REQ-4.4.3.4: If the user selects the option to view suggestions, the system will return a list of commonly searched together items by other users.
Priority: 2

REQ-4.4.3.4: If the user attempts to update the quantity of an existing item to a valid number, the system will update the list. A valid number is nonzero and positive.
Priority: 2

REQ-4.4.3.4: If the user attempts to remove an existing in the list, the system will update the list.
Priority: 2

REQ-4.4.3.5: When the user is finished adding items and completes the list, the system will output and save the list in an easy-to-read format. The order of the output list will be functionally determined by the shortest path the user will need to take through the store to collect all the items, and be accompanied by the total price of all items in the list.
Priority: 1

REQ-4.4.3.6: If the user tries to complete an empty list, the system will prompt the user with the message, “Please add at least one item to the list.”
Priority: 3

4.5 Get Shopping List

4.5.1 Description and Priority

A user accesses their most recently created shopping list.
Assumption: User has logged in.
Priority: 1

4.5.2 Stimulus/Response Sequences

1. Users access their most current list.

2. Software displays the list.

4.5.3 Functional Requirements

REQ-4.5.3.1: If the user has an existing shopping list, the system will retrieve it and display it to the user.
Priority: 1

REQ-4.5.3.2: If the user has no existing list, the system will display an error, “Error: Please create a shopping list first.
Priority: 1

4.6 Application Manager Login

4.6.1 Description and Priority

An application manager logs in to the system.
Priority: 1

4.6.2 Stimulus/Response Sequences

1. Application managers enter their credentials.
2. System grants them access.

4.6.3 Functional Requirements

REQ-4.6.3.1: If the application manager enters the correct login credentials, the system will grant them access.
Priority: 1

REQ-4.6.3.2: If the application manager enters incorrect login credentials (name or password), the system prompts them with an appropriate error message.
Priority: 1

4.7 Add Store

4.7.1 Description and Priority

An application manager adds a store to the system.
Assumptions: The application manager is logged in.
Priority: 1

4.7.2 Stimulus/Response Sequences

1. Application manager enters details about a store including its name and location details.
2. System creates a record of the store.

4.7.3 Functional Requirements

REQ-4.7.3.1: If the application manager enters a unique store name and valid location data, the store will be instantiated in the system.
Priority: 1

REQ-4.7.3.2: If the application manager enters a non-unique store name or invalid location data, the store will not be instantiated in the system.
Priority: 1

REQ-4.7.3.3: If the store is not instantiated in the system, then an error prompt will be shown to the application manager.
Priority: 1

4.8 Remove Store

4.8.1 Description and Priority

An application manager removes a store from the system.
Assumptions: The application manager is logged in.
Priority: 1

4.8.2 Stimulus/Response Sequences

1. application manager requests a store be removed by name.
2. System deletes all information pertaining to the store.

4.8.3 Functional Requirements

REQ-4.8.3.1: If the application manager enters a valid store name, the system will remove the store, including all associated categories, items, and manager accounts.
Priority: 1

REQ-4.8.3.2: If the application manager enters an invalid store name data, the system will prompt the administrator with an appropriate error message.
Priority: 1

4.9 Add Store Manager

4.9.1 Description and Priority

An application manager adds a store manager user to the system.
Assumptions: The application manager is logged in.
Priority: 1

4.9.2 Stimulus/Response Sequences

1. An application manager searches for an existing store by name.
2. The system returns the store.
3. The application manager enters new manager credentials: ID and password.
4. The system instantiates the manager user for that store.

4.9.3 Functional Requirements

REQ-4.9.3.1: If the application manager enters a store name that exists in the system, the system shall return the correct store.
Priority: 1

REQ-4.9.3.2: If the application manager enters a non-existing store name, the system shall return a message, “That store does not exist.”
Priority: 1

REQ-4.9.3.3: If the application manager enters valid manager credentials (alphanumeric password at least 8 characters long) and a unique ID, the manager will be instantiated by the system with respect to the store.
Priority: 1

REQ-4.9.3.4: If the manager is not instantiated into the system, an appropriate error prompt will be displayed to the application manager.
Priority: 1

4.10 Remove Store Manager

4.10.1 Description and Priority

An application manager removes a store manager user from a store in the system.
Assumptions: The application manager is logged in.
Priority: 1

4.10.2 Stimulus/Response Sequences

1. An application manager searches for an existing store by name.
2. The system returns the store.
3. The application manager enters the manager ID.
4. The system removes the manager account from the store.

4.10.3 Functional Requirements

REQ-4.10.3.1: If the application manager enters a store name that exists in the system, the system shall return the correct store.
Priority: 1

REQ-4.10.3.2: If the application manager enters a non-existing store name, the system shall return a message, “That store does not exist.”
Priority: 1

REQ-4.10.3.3: If the application manager enters a valid manager ID, and the manager will be removed from the system with respect to the store.
Priority: 1

REQ-4.10.3.4: If the application manager entered an incorrect manager ID, the system will prompt the manager with an appropriate error message.
Priority: 2

4.11 View Store Details

4.11.1 Description and Priority

An application manager searches for an existing store.
Priority: 1

4.11.2 Stimulus/Response Sequences

1. An application manager searches a store by name in a search field.
2. System retrieves the store and displays its details including name, location, and store managers bound to it, and a list of actions the application manager can take including add manager, remove manager, or remove store.

4.11.3 Functional Requirements

REQ-4.11.3.1: If the application manager enters the name of an existing store, the system will return that stores’ details including name, location, and store managers bound to it.
Priority: 1

REQ-4.11.3.2: If the application manager enters the name of a store that does note exist in the system, the system will prompt the application manager with an error message.
Priority: 2

4.12 Update Store Details

4.12.1 Description and Priority

An application manager updates the details of an existing store including name and/or location.
Priority: 2

4.12.2 Stimulus/Response Sequences

1. An application manager searches a store by name in a search field.
2. System retrieves the store and displays its details including name, location, and store managers bound to it, and a list of actions the application manager can take including add manager, remove manager, remove store and update details.
3. The application managers selects the update details action.
4. The system prompts the application manager for details to update.
5. The application manager enters details to update and submits them.
6. The system updates the store with the respective details.

4.12.3 Functional Requirements

REQ-4.12.3.1: If the application manager enters valid details to update, the system will update the store correctly with those details.
Priority: 1

REQ-4.12.3.2: If the application manager enters an updated name of the store which already exists in the system, or the updated location is already occupied by another store, the system will prompt the application manager with an error message.
Priority: 2

4.13 Store Manager Login

4.13.1 Description and Priority

A store manager logs into a specific store.
Priority: 1

4.13.2 Stimulus/Response Sequences

1. Store manager visits the front-end through a webportal.
3. Software displays the landing page and prompts the manager to login.
4. Store manager credentials including store name, login ID, and password.
5. Software grants access and displays dashboard.

4.13.3 Functional Requirements

REQ-4.13.3.1: If the manager enters the correct credentials including store name, manager ID, and password, the system will grant access to that store and associated manager features.
Priority: 1

REQ-4.13.3.2: If the manager enters incorrect credentials, the system will display an appropriate error message.
Priority: 2

4.14 Add Category

4.14.1 Description and Priority

A store manager adds a new category of products with a description.
Assumptions: The store manager is logged in.
Priority: 1

4.14.2 Stimulus/Response Sequences

1. The store manager adds a new category, entering its name and a description.
2. The system instantiates the new category.

4.14.3 Functional Requirements

REQ-4.14.3.1: If the manager enters a unique new category name and a description, the system instantiates the item.
Priority: 1

REQ-4.14.3.2: If the category is not instantiated, the system prompts the manager with an appropriate error message.
Priority: 2

4.15 Add Item to category

4.15.1 Description and Priority

A store manager adds a new product, specifying its category and location in the store, and its distance to all other items in the store.
Assumptions: The store manager is logged in.
Priority: 1

4.15.2 Stimulus/Response Sequences

1. The store manager adds an item, entering its price, description, and location.
2. The system instantiates the new item.

4.15.3 Functional Requirements

REQ-4.15.3.1: If the manager enters a unique new product name, a description, and location that is not occupied in the store, the system instantiates the item.
Priority: 1

REQ-4.15.3.2: If the item is not instantiated, the system prompts the manager with an appropriate error message.

Priority: 2

4.16 Update category

4.16.1 Description and Priority

A store manager specifies an existing category, and updates its name or description attribute.
Assumptions: The store manager is logged in to the correct store.
Priority: 1

4.16.2 Stimulus/Response Sequences

1. The store manager enters the category name, attribute to update, and the value of the attribute to update.
2. The system mutates the existing category with the passed value.

4.16.3 Functional Requirements

REQ-4.16.3.1: If the manager enters an existing category name, and passes a valid attribute name and value, the system will mutate the respective category.
Priority: 1

REQ-4.16.3.2: If the enters a non-existing category name or attribute name, the system will display an appropriate error message.
Priority: 2

4.17 Remove category

4.17.1 Description and Priority

A store manager removes an existing category at a store.
Assumptions: The store manager is logged in to the correct store.
Priority: 1

4.17.2 Stimulus/Response Sequences

1. The store manager enters the category name.
2. The system removes the category and all corresponding items.

4.17.3 Functional Requirements

REQ-4.17.3.1: If the manager enters an existing category name, the system will remove the respective category and all existing items within it.
Priority: 1

REQ-4.17.3.2: If the manager enters a non-existing category name, the system will display an appropriate error message.

Priority: 2

4.18 Remove Item

4.18.1 Description and Priority

A store manager removes an existing product from the store.
Assumptions: The store manager is logged in.
Priority: 1

4.18.2 Stimulus/Response Sequences

1. The store manager removes the item by name.
2. The system removes the item from the system with respect to the store.
3. The system updates the items location to be empty.

4.18.3 Functional Requirements

REQ-4.18.3.1: If the manager attempts to remove an existing item by its valid name, the system will remove the item and its location will be updated to empty.
Priority: 1

REQ-4.18.3.2: If the manager searches for an invalid item, the system will return a message, “Invalid item name.”

4.19 Update Item details

4.19.1 Description and Priority

A store manager updates information about an item in their store.
Assumptions: The store manager is logged in.
Priority: 1

4.19.2 Stimulus/Response Sequences

1. The store manager searches for the item.
2. The system returns the item.
3. The store manager specifies the attribute to update (price, description)
4. The system updates the items attribute.

4.19.3 Functional Requirements

REQ-4.19.3.1: If the manager searches for a valid item, the system will return the item.
Priority: 1

REQ-4.19.3.2: If the manager enters valid updated information for the item attribute (non-empty description, positive value price), the system will update the item information.
Priority: 2

REQ-4.19.3.3: If the manager searches for an invalid item, the system will return a message, “Invalid item name.”

4.20 Move Item

4.20.1 Description and Priority

A store manager moves the location of an item in their store to an empty location, or swaps it with an existing item.
Assumptions: The store manager is logged in.
Priority: 1

4.20.2 Stimulus/Response Sequences

1. The store manager searches for the item.
2. The system returns the item.
3. The store manager specifies the location to move the item.
4. The system updates the items location, and swaps the item with another item if that location is occupied.

4.20.3 Functional Requirements

REQ-4.20.3.1: If the manager searches for a valid item to move, the system will return the item.
Priority: 1

REQ-4.20.3.2: If the manager specifies a valid location, the specified item will be moved to that location. If there exists an item already at the new location, the two items exchange locations.
Priority: 2

REQ-4.20.3.3: If the manager searches for an invalid item, the system will return a message, “Invalid item name.”

REQ-4.20.3.4: If the manager enters an invalid location, the system will return a message, “Invalid location.” Invalid locations are all locations outside the bounds of the store.
Priority: 2

5. Non-functional Requirements

This section describes the non-functional requirements for the SmartShoppers system.

5.1 Performance Requirements

General Performance Requirements:

I. All pages of the application must take a maximum of 10 seconds to load on the DSL broadband connection.
II. Application must be able to support 50 concurrent users. Without degrading the performance of the application.
III. Unless a specified time for maintenance is scheduled, the application is expected to strive for 99.9% uptime. The maintenance timing will be relayed to the user in advance.
IV. Application must notify the user if there are network connectivity issues.

5.1.1 Account Signup

REQ-5.1.1.1: When the user visits the SmartShoppers registration webportal, it must take less than 2 seconds to load the page with all the required information and fields.
REQ-5.1.1.2: Upon providing the incorrect information (non-uniques email, non strong password, account already exists) the user must be notified within 8 seconds.
REQ-5.1.1.3: Users must be notified if there was an issue in the registration process.
REQ-5.1.1.3: The system must instantiate a new account in under 8 seconds.

5.1.2 User login

REQ-5.1.2.1: If the user visits the SmartShoppers webportal, it must take less than 2 seconds to load the page with all the required information and fields.
REQ-5.1.2.2: Upon providing the incorrect information (non-uniques email, non-strong password, account does not exist) the user must be notified within 8 seconds.
REQ-5.1.2.3:All login input boxes are of the type password.

5.1.3 Get and Set Shopping Location

REQ-5.1.3.1: The response time for interacting with the database is less than 8 seconds.
REQ-5.1.3.2: If the user enters an invalid postal code or province city combination, the system will prompt them with a message, “Error”, within 8 seconds.

5.1.4 Create Virtual Shopping List

REQ-5.1.4.1: All notification from the system must be updated within 8 seconds of submission. (invalid item name, user searches for an item which does not exist).
REQ-5.1.4.2: System should be able to handle user requests within 10 seconds.

5.1.5 Add Store

REQ-5.1.5.1: All notification from the system must be updated within 8 seconds of submission. (non-unique store name or invalid location data)
REQ-5.1.5.2: The process of adding a store must not take more than 30s.
REQ-5.1.5.3: If the application restarts the unsaved data will not be retriable.

5.1.6 Remove Store

REQ-5.1.6.1: All notification from the system must be updated within seconds of submission. (invalid store name data)
REQ-5.1.6.2: If the system was shut down in between the removal process. The removal process will be paused.

5.1.7 Add Store Manager

REQ-5.1.7.1: All notification from the system must be updated within 8 seconds of submission. (non-existing store name, the system shall return a message, “That store does not exist.”, manager is not instantiated into the system, an appropriate error prompt will be displayed to the application manager)

5.1.8 Remove Store Manager

REQ-5.1.8.1: All notification from the system must be updated within 8 seconds of submission. (non-existing store name, the system shall return a message, “That store does not exist.”, incorrect manager ID, the system will prompt the manager with an appropriate error message)

5.1.9 Add Item

REQ-5.1.9.1: All notification from the system must be updated within 8 seconds of submission. (If the item is not instantiated, the system prompts the manager with an appropriate error message)

5.1.10 Remove Item

REQ-5.1.10.1: All notification from the system must be updated within 8 seconds of submission. (invalid item, the system will return a message)

5.1.11 Update Item details

REQ-5.1.11.1: All notification from the system must be updated within 8 seconds of submission. (invalid item, the system will return a message)

5.1.12 Move Item

REQ-5.1.12.1: All notification from the system must be updated within 8 seconds of submission. (invalid item, the system will return a message, “Invalid item name.”, invalid location)

5.2 Safety Requirements

REQ-5.2.1: For the protection of the used the minimum age limit on the application is 16 years.
REQ-5.2.2: Without registering for the service user must have access to the functionality of the application.
REQ-5.3.3: There are different types of users. each of the type of users only have accessibility to their set of functionalities.

5.3 Security Requirement

Confidentiality:
REQ-5.3.1: The location of the user is protected.
REQ-5.3.2: All users are required to login via email and passwords.
REQ-5.3.3: All user ids are unique.
REQ-5.3.4: All users are required to login via email and passwords.

Integrity:
REQ-5.3.5: All users must be authenticated via their email.

Availability:
REQ-5.3.6: All information received from the user is protected, no one but the application manager, and the individual user has access to it.

5.4 Software Quality Attributes

Reliability:
REQ-5.4.1: The system must be able to direct uses to the correct store.
REQ-5.4.2:The system must be able to arrange the most optimal path for the user based on their shopping list.

Correctness:
REQ-5.4.3:The system must follow the requirements outlined in this SRS document.

Adequacy:
REQ-5.4.4:The system will not ask for unnecessary information from the user (acceptable personal information required is address and email).
REQ-5.4.5: Most of the inputs from the user will not need a specific restricted data type (eg: no input field must be restricted to int, with an exception of the number part of the street address).
REQ-5.4.6: All outputs must be clear and well structured, in a form that is easy to interpret. If there was an error in an event, the user should be notified.

Learnability:
REQ-5.4.7: User interface must be easy to learn and adapt to. Use pre-existing symbols and layouts. (eg: shopping list button has a image of shopping list)
REQ-5.4.8: Present information as close to reality as possible. (eg: adding to the shopping list happens after the object has been selected).
REQ-5.4.9: Add a short tutorial after the user has logged in for the first time.

Robustness:
REQ-5.4.10: If the user accidentally disconnects from the network, the application must save the last working image of the software.

Maintainability (Readability & Portability):

Readability:
REQ-5.4.12: The code for the application must include clear and specific comments.

Portability:
REQ-5.4.14: The code for the application must include documentation that is easy to follow.
REQ-5.4.15: Create an application that is adaptable on multiple devices, and environment.

5.5 Business Rules

Non-registered User:

I. Can look to see if an item exists in the store but will not have access to a cart

Registered User:

I. Can look to see if an item exists in the store
II. User has access to a cart, so they are able to find the optimal path to the items within the cart

Appendix A: Glossary

This section contains a glossary of terms used throughout the document.

AES	Advanced Encryption Standard
API	Application Programming Interface
FAQ	Frequently Asked Questions
HTML	Hypertext Market Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
POS	Point-of-Sale
REST	Representational State Transfer
SRS	Software Requirement Specification
TLS	Transport Layer Security
UI	Graphical User Interface