**Shopping List Documentation**

Requirements Engineering

Requirements Gathering

\*\*\*\*\*\*\*\*Desiree’s notes\*\*\*\*\*\*\*\*\*\*\*

Requirements Analysis

\*\*\*\*\*Refine and modify the requirements gathered\*\*\*\*

Requirements specification

Actor

An actor is an agent external to the system, that interacts with it.

The actors for the Grocery Shopping List are as follows:

1. Grocery Shopper

2. Grocery Seller

Concepts

A concept is an agent inside the Grocery Shopping List system, that gives it

its functionality.

The concepts for the Grocery Shopping List are as follows:

1. Bookkeeper

2. Clerk

3. Diary

4. Catalogue

5. Product

6. Book Shelf

User Stories

A user story highlights a use case for the Grocery Shopping List system.

The user stories for the Grocery Shopping List system are as follows:

1. Shopper completes registration and submit it, to create an account.

2. Shopper completes login and submit it, to create shopping list.

3. Shopper requests diary, to view previous shopping list.

4. Shopper requests catalogue, to view products available for purchase.

5. Shopper requests current shopping list, to view products I have selected to buy.

6. Shopper creates shopping list, to remember products to buy.

7. Shopper writes product to shopping list, to add product to shopping list.

8. Shopper erases products from shopping list, to remove product from shopping list.

9. Shopper deletes shopping list, to discard current shopping list.

10. Shopper checks out shopping list, to register shopping list in diary.

11. Shopper views current shopping list total.

User Acceptance Test

A user acceptance test is a process of verifying that a user story has been met.

The user acceptance tests for the Grocery Shopping List system are as follows

1. Given that I am a shopper and I have no "Grocery Shopping List" account, when I complete a registration, I can submit it to create a "Grocery Shopping List" account.

2. Given that I am a shopper and I have a "Grocery Shopping List" account, when I login, I can create a shopping list.

3. Given that I am a shopper and I am logged into the "Grocery shopping List" system, when I request my diary, I can view previous shopping lists.

4. Given that I am a shopper and I am logged into the "Grocery shopping List" system, when I request the catalogue, I can view products currently available for purchase.

5. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when I request my current shopping list, I can view products I have selected to buy.

6. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, I can create a shopping list.

7. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when I add a product to my current shopping list, I can remember to buy it.

8. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when I have bought a product, I can remove the product from my current shopping list.

9. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when I delete my current shopping list

10. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when check out my current shopping list it is added to my diary

11. Given that I am a shopper and I am logged into the "Grocery Shopping List" system, when create a shopping list, I can view the sum total of the products I have added to the list.

Architectural styles

Client-server

\*The grocery shopping list implements the Multi-tier client-server architectural style which consists of the presentation tier, application tier and the Data tier.

\*The shopper will interact with the software using a website by browsing through the catalogue , all user inputs will be collected by the presentation tier and process them to the application tier

\*On the application tier, the shopping list website make use of Node.js as servers run time and make use of the express to facilitate the communication to with the database

\*The software will make use of the mongoes from the mongoDB on the Data tier level to connect the application tier express to the database.

\*Due restricted communication the shoppers user interface and the serves via an API, the data tier is separated from the clients and thus allow much tight security

\*This separation allows software developers to test the system at ease

\*This system division can be difficult to design and implement due to having many layers

Representational state transfer

\*The shopping list architecture is restful and therefore ,it has certain constraints associated with it

\*The server and client evolves separately ,that is modification made to the client interface may not affect the server structure

\*Each shopping list session is not retained by the server

\*Proxy servers can be added in the system, and this will allow to show the most frequently purchased product by each users at the beginning of the catalogue.

\*shoppers have no knowledge regarding the intermediate connections made between the website and servers involved

\*This architecture make use of standard interfaces for the request and response ,that is the shoppers access multiple resources on the website using server-provided links,(example ,clicking on the product and going to the url that will give the product description )

Architectural view of the shopping list website

\*\*\*\*Check the Shopping list architectural view.io\*\*\*\*\*