**My Pantry**

**Project Requirement Capture**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change/Comments** |
| 0.1 | 2/17 | Everyone | Came up with list of functional requirements, non-functional requirements, and scenarios |
| 0.2 | 2/17 | Rachel | Made document on Google docs for this document, and made a basic outline |
| 0.3 | 2/18 | Rachel | Filled out the current situation and statement of work |
| 0.4 | 2/21 | Yang | Added scenario for adding an entry, and the respective use case model for it. Filled out project schedule |
| 0.5 | 2/21 | Yang | Reorganized functional requirements list |
| 1.0 | 2/21 | Rachel | Added two scenarios and their respective use case models |
| 1.1 | 2/22 | Rachel | Added two written use cases |
| 1.2 | 2/23 | Zach | Added use case scenario, use case model, and two written use cases |
| 1.3 | 2/24 | Rachel | Edited the statement of work |
| 1.4 | 2/24 | Yang | Added two use cases and edit functional requirement |
| 1.5 | 2/24 | Nate | Added use case scenario, use case model. Edited use case scenario. |
| 2.0 | 2/24 | Rachel | Made PowerPoint Presentation, and submitted document |

**Table of Contents**

Project Requirements Capture Deliverables/Schedule………..……….... 4

Current Situation………...……………………………………………………4

Statement of Work.…………………………………………………………...4

Functional Requirements..…………………………………………………...5

Nonfunctional Requirements..………………………………………………. 6

Scenarios…….……………………………………………………………....6-8

Use Cases - Written………………………………………………………...8-9

Use Cases - Models……………………………………………………...10-14

Project Schedule....…………...……………………………………………...14

**Project Requirements Capture Deliverables/Schedule**

|  |  |
| --- | --- |
| **Description** | **Complete Date** |
| Requirement Capture Document | 2/24 |
| Powerpoint Presentation | 2/24 |
| Team Review | 2/24 |
| Assign Presenter | 2/24 |
| Upload Document/Presentation to Repository | 2/24 |
| Project Requirement Review/Presentation | 2/24 |

**Deliverables**

**Problem Description**

**Current Situation**

Essentially, we are trying to solve the consumer issue of keeping track of inventory and shopping lists. Consumers often forget what items they already have available in their pantry. This results in them running out of things and not having anymore immediately available, or buying too much of an item because they could not remember if they already had the item or not. Our website will solve this issue by allowing a user to log their current inventory into the website, where they can easily see what they have available and how much they have available. The website will also include a shopping list feature that allows the user to add things they need to purchase when they go to the store. All of this information is stored in a database.

**Statement of Work**

This project involves the development of a pantry application. The project will require a team of 4 developers. The time allotted to complete the project is approximately 4 months (January 2016 - April 2016). Therefore, as of now, the team has 8 person months left to complete the project. The team will deliver a completed and tested application by April 20, 2016.

**Functional Requirements**

**Inventory List**

**---- Display the inventory**

-The website should be able to display all the entries in the inventory list.

-The website should be able to display the inventory by selected category.

-The website should be able to display the inventory by the order of

(Bought date, expire date, quantity, alphabet, etc.).

-The website should be able to provide a print friendly (text only) list of the

inventory.

**---- Search the inventory**

-Users should be able to search the inventory using the name of the item.

-Users should be able to search the inventory with item UPC/SKU number.

**---- Edit entry**

-Users should be able to manipulate the quantity of item.

-Users should be able to delete entries in the inventory.

-Users should be able to change bought date/ expire date.

-Users should be able to add item to shopping list when consumed.

-Users should be able to save the changes after editing entries..

**---- Add entry**

-Users should be able to add new entries by UPC/SKU number.

-Users should be able to add new entries by enter the name and image

URL.

-Users should be able to add new entries by the history of inventory.

**Shopping List**

**---- Display**

-The website should be able to display the shopping list by order of

(entered date, alphabet, etc.)

-The website should be able to display a print friendly list.

**----** **Add Entry**

-Users should be able to keep shopping note (unstructured).

-Users should be able to add items to shopping list by UPC/SKU.

-Users should be able to add items to shopping according to the history.

**---- Edit Entry**

-Users should be able to edit the shopping note.

-Users should be able to move to items to the inventory.

-Users should be able to delete items

**Database**

-Information from entries should be properly stored in the database

-The database should be able to store information from the mobile team

**Non-Functional Requirements**

-Must be able to run on different browsers (Chrome, Mozilla, Opera, Internet

Explorer)

-Easy to use/User friendly

-Information must be properly added to the database

-Easy to test/maintain

**Scenarios**

1. **Search for something**

Joe and his family are looking to buy some fruits. Joe and his family decide they want to buy some bananas, apples, and oranges. However, because bananas and oranges don’t usually come in different types, Joe needs to decide what kind of apples he wants to get. So he gets on the web version of MyPantry to search for apples. Joe types in the word “apple” and the results show him a timestamp of which kind of apples his family last bought. Also listed are the types of apple available, such as Granny Smith, Gala, Red Delicious, and Honeycrisp apples. From here, Joe selects the type of apples he wants and is able to plan his trip to the store to go and obtain said apples.

1. **Editing an entry**

Emily just come back from grocery shopping. She bought a case of beer and a

case of Coke for the party tonight. So she add 6 to the beer and 12 to the Coke

in her inventory. She feels a little bit tired so she decide to enjoy a Coke right

now. Emily opens 1 can of Coke and take away 1 Coke from the inventory. Then

she goes to check the fridge to see if the vegetable snack is still fresh.

Unfortunately, it smells bad. So Emily throw it away and delete the vegetable

snack in the inventory list and add the vegetable snack to the shopping list.

3. **Adding/removing an entry**

John wants to make s'mores. He sees that he already has graham crackers and marshmallows. However, he notices that he doesn’t have any chocolate. He goes to the store and buys the chocolate. He doesn’t want to forget how much chocolate he has, so he decides to add it to his inventory on MyPantry after returning from the store. He starts by filling out the name. Then he adjusts the quantity to the amount he bought. Lastly he chooses the correct category for it and then hits the submit button. Now the chocolate is added to his MyPantry inventory in the database.

4. **Creating a New Shopping List**

# Señor Guapo has run out of ingredients for peanut butter and jelly sandwiches, and he also needs more apples. He wants to make sure that he doesn’t forget to buy them at the store, so he pulls up the My Pantry website and makes a new shopping list. Señor Guapo makes peanut butter and jelly quite often, so he decides that he will buy two jars of peanut butter. He adds a new item, enters peanut butter, and then increases the quantity to two. Then he saves it to his new shopping list. Next, Señor Guapo adds another new item, enters bread, and leaves the quantity at the default of one, and saves the item. Last, Señor Guapo adds his final item and enters apples. He decides he will get seven apples, so he enters the number 7 for quantity and saves the item. After Señor Guapo adds apples to the list he views her list to make sure everything was entered properly. Then he realizes that instead of typing in 7 for apples as he had intended, he entered 70. So he edits the item, changes the number to 7 and saves it. Señor Guapo now has his shopping list ready for the next time he goes to the store.

5. **Searching/Editing Shopping List**

Señor Guapo loves peanut butter and jelly so much that he decides to make five of them for lunch. He uses up all the ingredients he had at home, and couldn’t remember if he had already added them to his shopping list on My Pantry. Señor Guapo pulls up the My Pantry website and views his shopping list. Then he searches for each of the ingredients (Bread, jelly and peanut butter). While looking at his list, he realizes he still has apples on her shopping list from the previous week, which he already purchased, so he removes apples from the list. He also notices that he had 10 jars of jelly and 10 jars of peanut butter on his list. Realizing that he may have been a little too excited about peanut butter and jelly sandwiches at the time, he decides that he really does not need that many jars of each. So, he edits the items and reduces the quantity to 5 jars each saves his edits.

**Use Cases - Written**

**Use Case Name:** Display all the inventory (default order: date entered)

Participant Actors:

User (Emily in the Scenarios)

Flow of the Events:

1. User browser the inventory

2. The database pull all the information to the webpage.

**Use Case Name:** Display the inventory by category

Participant Actors:

User (Emily in the Scenarios)

Flow of the Events:

1. User browser the inventory

2. User wants to show the items in a particular or some of the categories.

3. The database pull the information that user required to the webpage.

**Use Case Name:** Display the inventory by order of

Participant Actors:

User (Emily in the Scenarios)

Flow of the Events:

1. User browser the inventory

2. User wants to show the items by the order of (Bought date, expire date, quantity, etc.)

3. The database pull the information that user required to the webpage.

**Use Case Name:** Creates New entry in List

Participant Actors:

User (John)

Flow of Events:

1. User fills out item name

2. User fills out quantity

3. User selects category

4. User hits submit button

**Use Case Name:** Removes old entry in List

Participant Actors:

User (John)

Flow of Events:

1. User searches for an item already in database

2. information is displayed if found

3. Item is removed from database

**Use Case Name:** Create New List

Participant Actors:

User (Señor Guapo)

Flow of Events:

1. User clicks button to create a new shopping list

2. User types in a name for the new shopping list

3. User hits the create button

**Use Case Name:** Edit an Entry

Participant Actors:

User (Señor Guapo)

Flow of Events:

1.User searches for the list or item they would like to view

2.User finds the entry they would like to edit

3. User presses the edit button

4. User makes changes to entry (item name, quantity etc.)

5. User saves edits

**Use Case Name:** Search

Participant Actors:

User (Joe)

Flow of Events:

1. User types in search term into the search bar.

2. User presses search

3. Program searches the database to see if item is in the

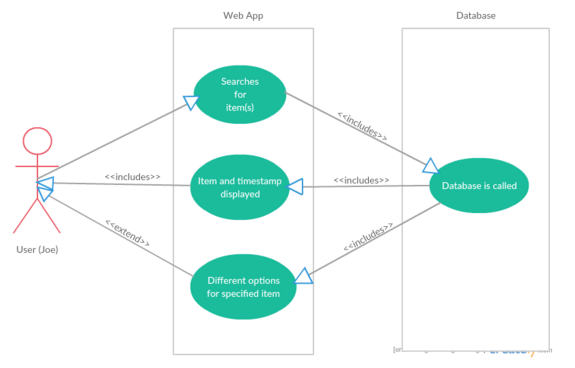
inventory/shopping list

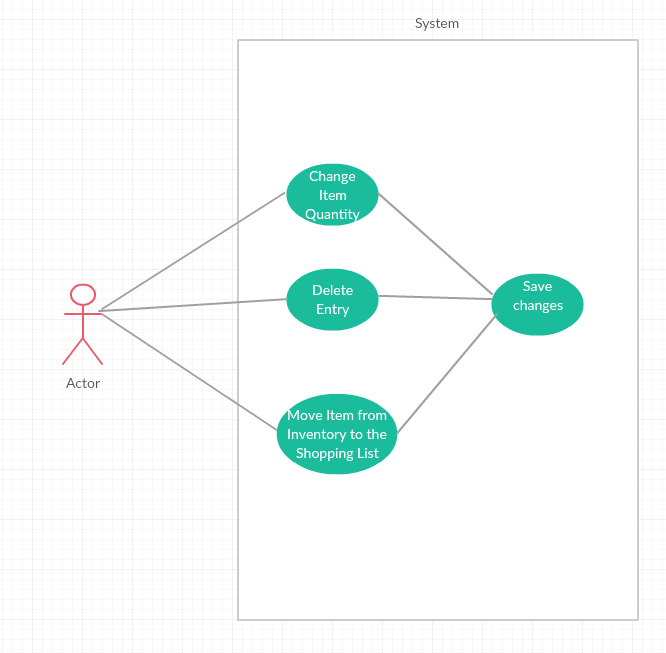
4. Item will be displayed on the screen if it exists. Otherwise user will

receive message that item is not in inventory/shopping list.

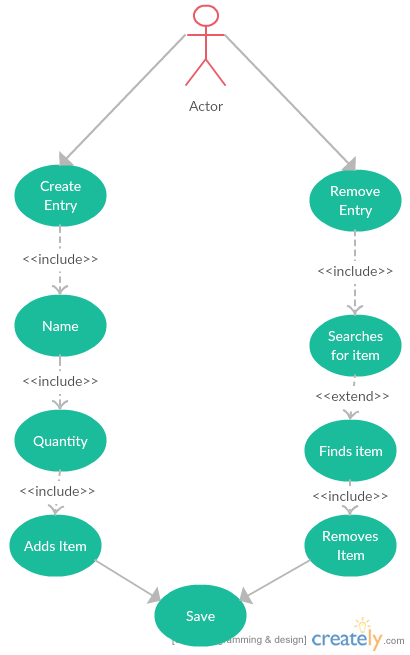
**Use Cases - Models**

1. Searching for Something

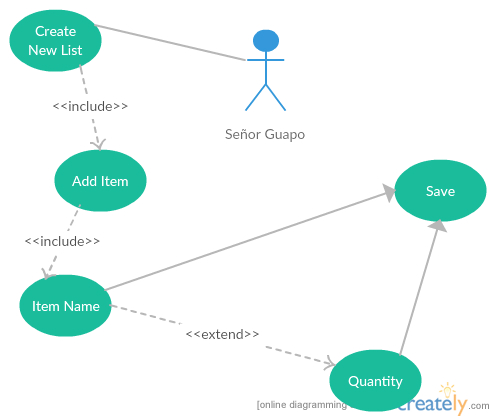


2. Editing an Entry 

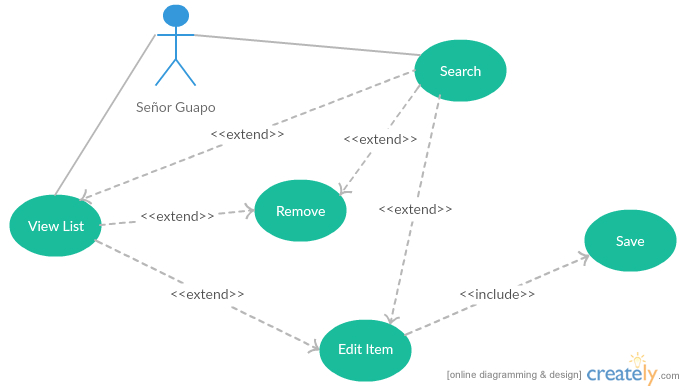
3. Adding/Removing an Entry



4. Creating a Shopping List



5. Searching/Editing Shopping List



**Project Schedule**

### Software Development Plan completed

### Requirement Capture 02/11/2016 - 02/25/2016

### Implementation 02/25/2016 - 04/13/2016

### Testing 04/13/2016 - 04/20/2016

### System Demo 04/20/2016