Software Requirements Specification

JAMR Kitchen IMS (JKI)

Javid Ditty, Andre Floyd, Mohamad Elzaatari, Ryan Len

1.0 About JAMR **2**
1.1 Company Details 2
1.2 Team Structure 2

2.0 Project Specifications **3**
2.1 Goals and Objectives 3
2.2 Statement of Scope 3
2.3 Software Context 3

3.0 Requirements **4**
3.1 Sign Up 4
3.2 Sign In 4
3.3 Set Password 5
3.4 Delete Account 5
3.5 Sign Out 6
3.6 Create Item 6
3.7 Delete Item 7
3.8 Create Store 7
3.9 Delete Store 8
3.10 Create Inventory 8
3.11 Set Item Quantity 9
3.12 Delete Inventory 9
3.13 Share Inventory 10
3.14 Set Inventory Share Permissions 11
3.15 Create Shopping List 11
3.16 Delete Shopping List 12
3.17 Share Shopping List 12
3.18 Set Shopping List Share Permissions 13
3.19 Create Alert 13
3.20 Initiate Alert 14
3.21 Delete Alert 15

4.0 Project Plan **16**

5.0 Appendix **18**
5.1 Supporting Technologies 18
5.2 References 18

1.0 About JAMR

1.1 Company Details

Company Name	JAMR Software Solutions
Company Address	18525 Northview Street, Dearborn, MI 48120
Phone Number	(313) 833-9505
Team Members	Javid Ditty, Andre Floyd, Mohamad Elzaatari, Ryan Len

1.2 Team Structure

As a result of having unknown and frequent changes in requirements, this project will proceed according to Scum, an agile software development model. Scrum teams have three roles, and each member in a Scrum team must adopt at least one role. Because the JAMR team will be a Scrum team for this project, the roles that each JAMR team member (see Section 1.1) will adopt are described in the table below. For more information about Scrum and its roles, see Section 3.0.

Scrum Master	Javid Ditty
Product Owner	Javid Ditty
Development Team	Javid Ditty
	Andre Floyd
	Mohamad Elzaatari
	Ryan Len

2.0 Project Specifications

2.1 Goals and Objectives

JAMR Kitchen IMS (JKI) is an inventory management system developed by JAMR Software Solutions. Its main goal is to help users track and restock foodstuffs in domestic, or small commercial, kitchens and pantries. Users will be able to create inventory databases to represent actual kitchens and entries in those databases to represent actual foodstuffs. Users will be able to view these representations and track the names, quantities, prices, purchase locations, expiration dates, storage locations, categories, and notes that they have given them. Using this information, users will be able to create digital shopping lists to help them plan to stock, or restock, their kitchens, updating their inventories afterward. Users will be able to share their inventories and shopping lists with other users, fostering communication and collaboration.

2.2 Statement of Scope

JKI is a web application hosted on a remote server that users can access using a web browser. Some user that navigates to JKI will be prompted to login or create an account with an email address and a password. Once the user has an account and has logged in, the user can:

- Logout, delete their account, and change their password.
- Create items that can be put in inventories and shopping lists.
- Create things that represent stores, including their prices for created items.
- Create inventories and view/edit/delete inventories with read/write/delete permissions.
- Create shopping lists and view/edit/delete shopping lists with read/write/delete permissions.
- Update their inventories as they purchase items on their shopping lists.
- Share their inventories and shopping lists with other users with certain access permissions.
- Set in-app and/or email restock alerts based on an item’s quantity and expiration date.

2.3 Software Context

JKI is designed to be used on mobile devices and laptops in kitchen, pantry, and market environments. Throughout the day, a user will be able to open the app and update their inventories and shopping lists as they use their kitchens and purchase groceries. Since users will both spontaneously and frequently open the app throughout the day, it must have low loading times and a simple UI that is easy to understand and navigate through. It will feel as easy as adding the information to a piece of paper.

The JAMR team was motivated to create JKI in response to organizational issues that each member encountered in their own kitchens. For example, some members:

- Bought items in bulk and forgot about their expiration dates.
- Forgot where they could buy an item that they’ve bought before.
- Forgot to purchase certain items while grocery shopping.
- Wanted to know the average price of an item before purchasing it.
- Forgot where they stored certain items in their kitchen.
- Were unsure about when to restock their kitchen.
- Were unsure about what they have stocked in their kitchen.

In short, the main motivation behind this project was the desire to solve problems that include, and relate, to these regularly occurring issues.

3.0 Requirements

Name	Sign Up	ID	UC01
Description	Adds a new account to the system.		
Actors	Guest, Account DB		
Preconditions	1. Guest is accessing the sign-up page. 2. Guest does not have an account.		
Main Flow	1. Guest enters an email address and a password into the sign-up page. 2. Guest selects the submit button on the sign-up page. 3. System hashes the email address and password. 4. System cannot find an entry with the email hash in the Account DB. 5. System generates an account ID. 6. System adds an entry containing the account ID, email hash, and password hash to the Account DB. 7. System presents a confirmation message and the sign-in page.		
Postconditions	1. Guest is accessing the sign-in page. 2. Guest has an account.		
Exception Flows	4a. AccountAlreadyExists: System finds the email hash in the Account DB. 5a. System presents an AccountAlreadyExists message. 6a. Go to Step 1.		

Name	Sign In	ID	UC02
Description	Enables a guest to access an account.		
Actors	Guest, Account DB		
Preconditions	1. Guest is accessing the sign-in page. 2. Guest has an account.		
Main Flow	1. Guest enters an email address and a password into the sign-in page. 2. Guest selects the submit button on the sign-in page. 3. System hashes the email address and password. 4. System finds an entry with the email and password hash in the Account DB. 5. System presents a confirmation message and the home page.		
Postconditions	1. Guest is accessing the home page.		
Exception Flows	4a. AccountNotFound: System cannot find an entry with the email and password hash in the Account DB. 5a. System presents an AccountNotFound message. 6a. Go to Step 1.		

Name	Set Password	ID	UC03
Description	Sets an account's password to a new password.		
Actors	Member, Account DB		
Preconditions	1. Member is accessing the settings page.		
Main Flow	1. Member enters a password into the password field on the settings page. 2. Member selects the submit button on the settings page. 3. System presents a confirmation prompt. 4. Member accepts the confirmation prompt. 5. System hashes the member's email address and new password. 6. System finds an entry with the email hash in the Account DB. 7. System replaces the entry's current password hash with the new password hash in the Account DB. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the settings page. 2. Member's account has the new password.		
Exception Flows	4a. RejectPrompt: Member rejects the confirmation prompt. 5a. System clears the password field.		

Name	Delete Account	ID	UC04
Description	Removes an existing account from the system.		
Actors	Member, Account DB, Item DB, Inventory DB, Shopping List DB, Store DB, Alert DB		
Preconditions	1. Member is accessing the settings page. 2. Member has an account in Account DB.		
Main Flow	1. Member selects the delete account button. 2. System presents a confirmation prompt. 3. Member accepts the confirmation prompt. 4. System signs the member out of their account. 5. System finds entries with the account ID in the Account, Item, Inventory, Shopping List, Store, and Alert DBs. 6. System removes these entries from their corresponding databases.		
Postconditions	1. Member does not have an account.		
Exception Flows	3a. RejectPrompt: Member rejects the confirmation prompt.		

Name	Sign Out	ID	UC05
Description	Prevents a member from accessing an account.		
Actors	Member, Account DB		
Preconditions	None		
Main Flow	1. Member selects the sign out button. 2. System presents a confirmation prompt. 3. Member accepts the confirmation prompt. 4. System presents a confirmation message and the sign-in page.		
Postconditions	1. Member is accessing the sign-in prompt.		
Exception Flows	3a. RejectPrompt: Member rejects the confirmation prompt.		

Name	Create Item	ID	UC06
Description	Adds a new item to the system.		
Actors	Member, Item DB, Store DB		
Preconditions	1. Member is accessing the items page. 2. Entry with the member's account ID and provided item name cannot be found in the Item DB.		
Main Flow	1. Member selects the create new item button on the items page. 2. System presents an item creation prompt on the items page. 3. Member enters a name, a price, stores, categories, and notes into the prompt. 4. Member selects the submit button on the prompt. 5. System cannot find an entry with the member's account ID and the item name in the Item DB. 6. System adds an entry with the member's account ID and the information that the member provided to the Item DB. 7. System presents a confirmation message.		
Postconditions	1. Member is accessing the items page. 2. Entry with the information that the member provided is in the Item DB.		
Exception Flows	5a. ItemAlreadyExists: System finds an entry with the member's account ID and the item name in the Item DB. 6a. System replaces the existing information in the entry with the new information that the member provided. 7a. System presents a confirmation message.		

Name	Delete Item	ID	UC07
Description	Removes an existing item from the system.		
Actors	Member, Account DB, Item DB, Inventory DB, Shopping List DB, Store DB, Alert DB		
Preconditions	1. Member is accessing the items page. 2. An item exists in Items DB.		
Main Flow	1. Member selects an item from the item list on the items page. 2. Member selects the delete item button. 3. System presents a confirmation prompt. 4. Member accepts the prompt. 5. System finds an entry with the member's account ID and the item name in the Item DB. 6. System removes the entry from the Item DB and the associated entries in the Inventory, Shopping List, Store, and Alert DBs. 7. System removes the selected item from the item list on the items page. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the items page. 2. The item that the member selected is not in the Item DB.		
Exception Flows	4a. RejectPrompt: Member rejects the confirmation prompt.		

Name	Create Store	ID	UC08
Description	Adds a new store to the system.		
Actors	Member, Store DB, Item DB		
Preconditions	1. Member is accessing the stores page. 2. Entry with the member's account ID and provided store name cannot be found in the store DB.		
Main Flow	1. Member selects the create new store button on the stores page. 2. System presents a store creation prompt on the stores page. 3. Member enters a name, location, items, and item prices. 4. Member selects the submit button on the prompt. 5. System cannot find an entry with the member's account ID and the store name in the Store DB. 6. System adds an entry with the member's account ID and the information that the member provided to the Store DB. 7. System presents a confirmation message.		
Postconditions	1. Member is accessing the stores page. 2. Entry with the information that the member provided is in the Store DB.		
Exception Flows	5a. StoreAlreadyExists: System finds an entry with the member's account ID and the store		

	name in the Store DB. 6a. System replaces the existing information in the entry with the new information that the member provided. 7a. System presents a confirmation message.

Name	Delete Store	ID	UC09
Description	Removes an existing store from the system.		
Actors	Member, Account DB, Inventory DB, Shopping List DB, Store DB		
Preconditions	1. Member is accessing the store page. 2. A store exists in Store DB.		
Main Flow	1. Member selects a store from the store list on the stores page. 2. Member selects the delete store button. 3. System presents a confirmation prompt. 4. Member accepts the prompt. 5. System finds an entry with the member's account ID and the store name in the Store DB. 6. System removes the entry from the Store DB and the associated items in the Inventory and Shopping List DBs. 7. System removes the selected store from the store list on the stores page. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the stores page. 2. The store that the member selected is not in the Store DB.		
Exception Flows	4a. RejectPrompt: Member rejects the confirmation prompt.		

Name	Create Inventory	ID	UC10
Description	Adds a new inventory to the system.		
Actors	Member, Inventory DB, Item DB, Store DB		
Preconditions	1. Member is accessing the inventories page. 2. Entry with the member's account ID and provided item name cannot be found in the Inventory DB.		
Main Flow	1. Member selects the create new inventory button on the inventories page. 2. System presents an inventory creation prompt on the inventories page. 3. Member enters a name, items, quantities, expiration dates, and storage locations. 4. Member selects the submit button on the prompt. 5. System cannot find an entry with the member's account ID and the inventory name in the Inventory DB. 6. System adds an entry with the member's account ID and the information that the		

	member provided to the Inventory DB. 7. System presents a confirmation message.
Postconditions	1. Member is accessing the inventories page. 2. Entry with the information that the member provided is in the Inventory DB.
Exception Flows	5a. InventoryAlreadyExists: System finds an entry with the member's account ID and the inventory name in the Store DB. 6a. System replaces the existing information in the entry with the new information that the member provided. 7a. System presents a confirmation message.

Name	Set Item Quantity	ID	UC11
Description	Set the quantity of an item in an inventory.		
Actors	Member, Inventory DB, Item DB		
Preconditions	1. Member is accessing the inventories page.		
Main Flow	1. Member selects an inventory from the inventory list on the inventories page. 2. System presents the items in the inventory in an item list. 3. Member selects an item in the item list. 4. Member selects the edit item button. 5. System presents an edit item prompt on the inventories page. 6. Member enters a new, valid item quantity into the item quantity field of the prompt. 7. Member selects the submit button. 8. System finds an entry with the member's account ID, inventory name, and item name in the Inventory DB. 9. System updates the quantity section of the entry with the information that the member provided. 10. System updates the item quantity information in the item list. 11. System presents a confirmation message.		
Postconditions	1. Member is accessing the inventories page. 2. The item that the member selected has the quantity that the member provided.		
Exception Flows	6a. InvalidQuantity: Member enters a new, invalid item quantity into the item quantity field of the prompt. 7a. System presents an invalid quantity warning message in the prompt. 8a. Continue from 6.		

Name	Delete Inventory	ID	UC12
Description	Removes an existing inventory from the system.		
Actors	Member, Inventory DB, Shopping List DB, Alert DB		

Preconditions	1. Member is accessing the inventories page. 2. An inventory exists in Inventory DB.
Main Flow	1. Member selects an inventory from the inventory list on the inventories page. 2. Member selects the delete inventory button. 3. System presents a confirmation prompt. 4. Member accepts the prompt. 5. System finds an entry with the member's account ID and the inventory name in the Inventory DB. 6. System removes the entry from the Inventory DB and its associated entries in Shopping List DB and Alert DB. 7. System removes the selected item from the inventory list on the inventories page. 8. System presents a confirmation message.
Postconditions	1. Member is accessing the inventories page. 2. The inventory that the member selected is not in the Inventory DB.
Exception Flows	4a. RejectPrompt: Member rejects the confirmation prompt.

Name	Share Inventory	ID	UC13
Description	Share an existing inventory with another member.		
Actors	Member, Inventory DB, Account DB		
Preconditions	1. Member is accessing the inventories page.		
Main Flow	1. Member selects an inventory that they are the owner of from the inventory list on the inventories page. 2. Member selects the share button. 3. System presents a share inventory prompt on the inventory page. 4. Member enters the username of another member into the prompt. 5. Member selects the submit button. 6. System finds an entry in the Account DB that contains the username. 7. System finds an entry in the Inventory DB that contains the inventory name. 8. System cannot find the username in the shared access section of the entry. 9. System adds the username to the shared access section of the entry. 10. System presents a confirmation prompt.		
Postconditions	1. Member is accessing the inventories page. 2. The other member has access to the selected inventory.		
Exception Flows	6a. AccountNotFound: System cannot find an entry in the Account DB that contains the username. 7a. System presents an AccountNotFound message in the share inventory prompt. 8a. Continue from 4. 8b. AlreadyShared: System finds the username in the shared access section of the entry. 9b. System presents a confirmation prompt.		

Name	Set Inventory Share Permissions	ID	UC14
Description	Sets the read/write/delete share permissions for an inventory.		
Actors	Member, Inventory DB, Account DB		
Preconditions	1. Member is accessing the inventories page.		
Main Flow	1. Member selects an inventory from the inventory list on the inventories page. 2. Member selects the set share permissions button. 3. System presents a prompt with a dropdown menu on the inventories page that contains read/write/delete permission options. 4. Member selects a permission option in the dropdown menu. 5. Member selects the submit button. 6. System finds an entry with the member's account ID and the inventory name in the Inventory DB. 7. System updates the share permissions section of the entry with the submitted permissions. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the inventories page. 2. The inventory that the member selected has updated permissions.		
Exception Flows	N/A		

Name	Create Shopping List	ID	UC15
Description	Creates a new shopping list.		
Actors	Member, Shopping List DB, Item DB, Store DB		
Preconditions	1. Member is accessing the shopping list page. 2. Entry with the member's account ID and provided shopping list name cannot be found in the Shopping List DB.		
Main Flow	1. Member selects the create new shopping list button on the shopping list page. 2. System presents a shopping list creation prompt on the shopping list page. 3. Member enters a name and items into the prompt, including their prices, their quantities, and the stores they plan to purchase the items from. 4. Member selects the submit button on the prompt. 5. System cannot find an entry with the member's account ID and the shopping list name in the Shopping List DB. 6. System adds an entry with the member's account ID and the information that the member provided to the Shopping List DB. 7. System presents a confirmation message.		
Postconditions	1. Member is accessing the shopping list page.		

	2. Entry with the information that the member provided is in the Shopping List DB.
Exception Flows	5a. ShoppingListAlreadyExists: System finds an entry with the member's account ID and the shopping list name in the Shopping List DB. 6a. System replaces the existing information in the entry with the new information that the member provided. 7a. System presents a confirmation message.

Name	Delete Shopping List	ID	UC16
Description	Removes an existing shopping list from the system.		
Actors	Member, Shopping List DB, Item DB, Store DB		
Preconditions	1. Member is accessing the shopping list page. 2. A shopping list exists in Shopping List DB.		
Main Flow	1. Member selects a shopping list from the lists of shopping lists on the shopping lists page. 2. Member selects the delete shopping list button. 3. System presents a confirmation prompt. 4. Member accepts the prompt. 5. System finds an entry with the member's account ID and the shopping list name in the Shopping List DB. 6. System removes the entry from the Shopping List DB. 7. System removes the selected shopping list from the list of shopping lists on the shopping list page.		
Postconditions	1. Member is accessing the shopping list page. 2. The shopping list that the member selected is not in the Shopping List DB.		
Exception Flows	4a. RejectPrompt: Member rejects the confirmation prompt.		

Name	Share Shopping List	ID	UC17
Description	Share an existing shopping list with another member.		
Actors	Member, Shopping List DB, Account DB		
Preconditions	1. Member is accessing the shopping lists page.		
Main Flow	1. Member selects a shopping list that they are the owner of from the list of shopping lists on the shopping lists page. 2. Member selects the share button. 3. System presents a share shopping list prompt on the shopping lists page. 4. Member enters the username of another member into the prompt. 5. Member selects the submit button.		

	6. System finds an entry in the Account DB that contains the username. 7. System finds an entry in the Shopping List DB that contains the shopping list name. 8. System cannot find the username in the shared access section of the entry. 9. System adds the username to the shared access section of the entry. 10. System presents a confirmation prompt.
Postconditions	1. Member is accessing the shopping lists page. 2. The other member has access to the selected shopping list.
Exception Flows	6a. AccountNotFound: System cannot find an entry in the Account DB that contains the username. 7a. System presents an AccountNotFound message in the share shopping list prompt. 8a. Continue from 4. 8b. AlreadyShared: System finds the username in the shared access section of the entry. 9b. System presents a confirmation prompt.

Name	Set Shopping List Share Permissions	ID	UC18
Description	Sets the read/write/delete share permissions for a shopping list.		
Actors	Member, Shopping List DB, Account DB		
Preconditions	1. Member is accessing the shopping lists page.		
Main Flow	1. Member selects a shopping list from the list of shopping lists on the shopping lists page. 2. Member selects the set share permissions button. 3. System presents a prompt with a dropdown menu on the shopping lists page that contains the read/write/delete permission option. 4. Member selects a permission option in the dropdown menu. 5. Member selects the submit button. 6. System finds an entry with the member's account ID and the shopping list name in the Shopping List DB. 7. System updates the share permissions section of the entry with the submitted permissions. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the shopping list page. 2. The shopping list that the member selected has updated permissions.		
Exception Flows	N/A		

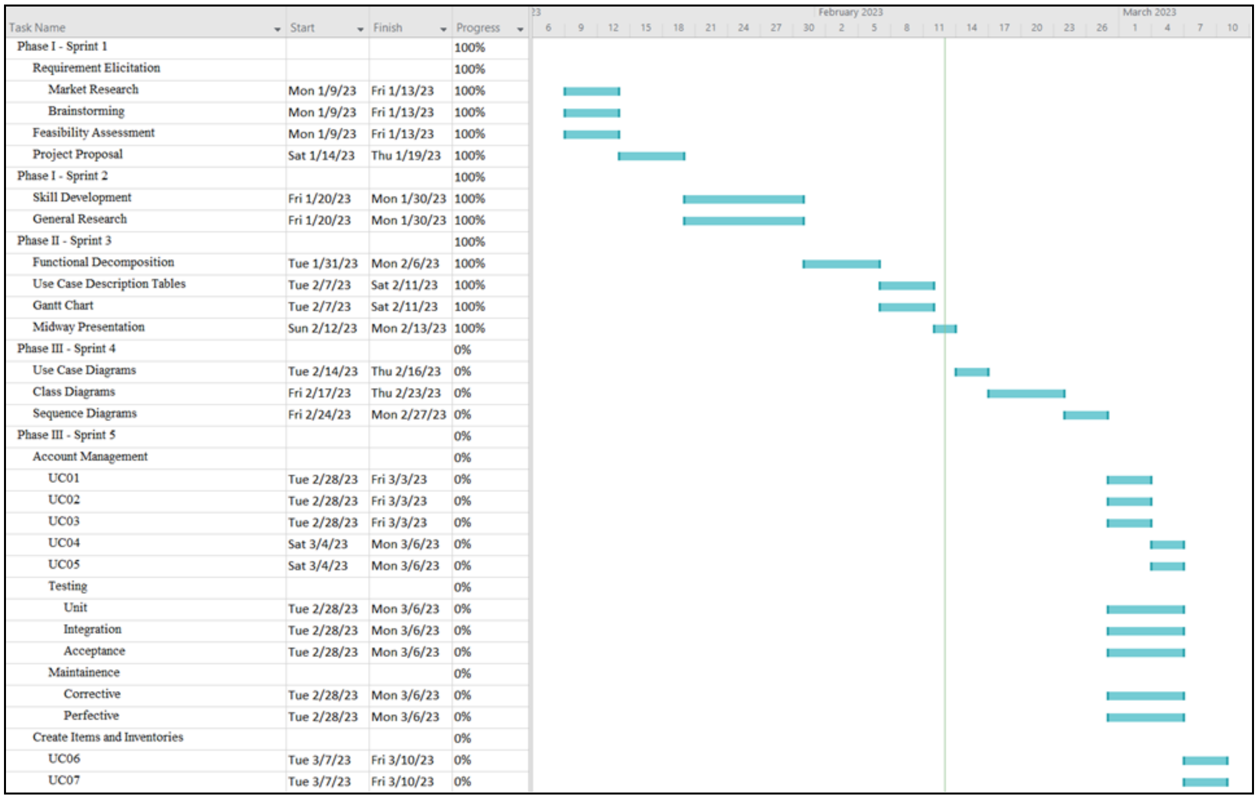
Name	Create Alert	ID	UC19
Description	Adds a new alert to the system.		
Actors	Member, Inventory DB, Alert DB		

Preconditions	1. Member is accessing the inventories page.
Main Flow	1. Member selects an inventory from the inventories list on the inventories page. 2. Member selects the create an alert button. 3. System presents an alert creation prompt on the inventories page. 4. Member enters a name, an inventory item, and a quantity threshold and/or expiration date into the prompt. 5. Member selects the submit button on the prompt. 6. System cannot find an entry with the member's account ID and the alert name in the Alert DB. 7. System adds an entry with the member's account ID and the information that the member provided to the Alert DB. 8. System presents a confirmation message.
Postconditions	1. Member is accessing the inventories page. 2. Alert that the member specified is in the Alert DB.
Exception Flows	6a. AlertAlreadyExists: System finds an entry with the member's account ID and the inventory name in the Alert DB. 7a. System replaces the existing information in the entry with the new information that the member provided. 8a. System presents a confirmation prompt.

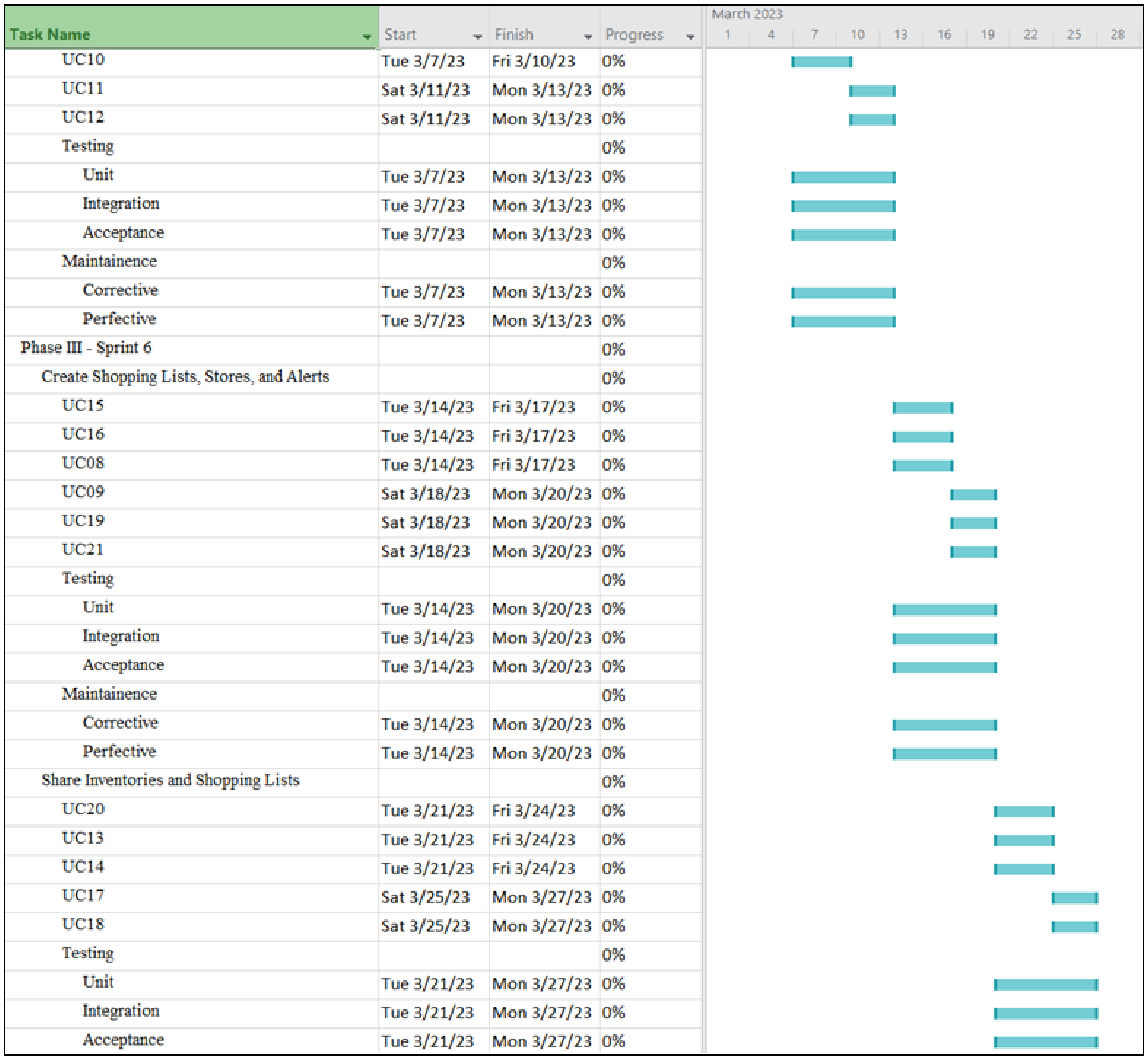
Name	Initiate Alert	ID	UC20
Description	Initiates, or triggers, an existing alert.		
Actors	Member, Inventory DB, Alert DB		
Preconditions	1. Item has at least one alert attached to it. 2. Item quantity is less than the quantity threshold and/or the current date is past the expiration date.		
Main Flow	1. System compares an item's quantity and/or the date to its alerts' quantity thresholds and/or expiration dates using the Inventory DB. 2. System finds that an item's quantity and/or the date is less than its alerts' quantity thresholds and/or past its alerts' expiration dates using the Inventory DB. 3. System finds the alerts' entries in the Alert DB. 4. System marks the entries as triggered in the Alert DB. 5. Member accesses the inventories page. 6. System finds the triggered alert entries for the member in the Alert DB. 7. System presents the triggered alerts on the inventories page.		
Postconditions	1. Alerts whose trigger conditions are met are initiated. 2. Member observes initiated alerts on the inventories page.		
Exception Flows	N/A		

Name	Delete Alert	ID	UC21
Description	Removes an existing alert from the system.		
Actors	Member, Inventory DB, Alert DB		
Preconditions	1. Member is accessing the inventories page. 2. An alert exists in the Alerts DB.		
Main Flow	1. Member selects an inventory from the inventories list on the inventories page. 2. Member selects the edit alert button. 3. System presents an edit alert prompt on the inventories page. 4. Member selects an alert from the alerts list on the edit alert prompt. 5. Member selects the delete alert button. 6. System finds an entry with the member’s account ID and the alert name in the Alert DB. 7. System removes the entry from the Alert DB. 8. System presents a confirmation message.		
Postconditions	1. Member is accessing the inventories page. 2. The alert that the member selected is not in the Alert DB.		
Exception Flows	N/A		

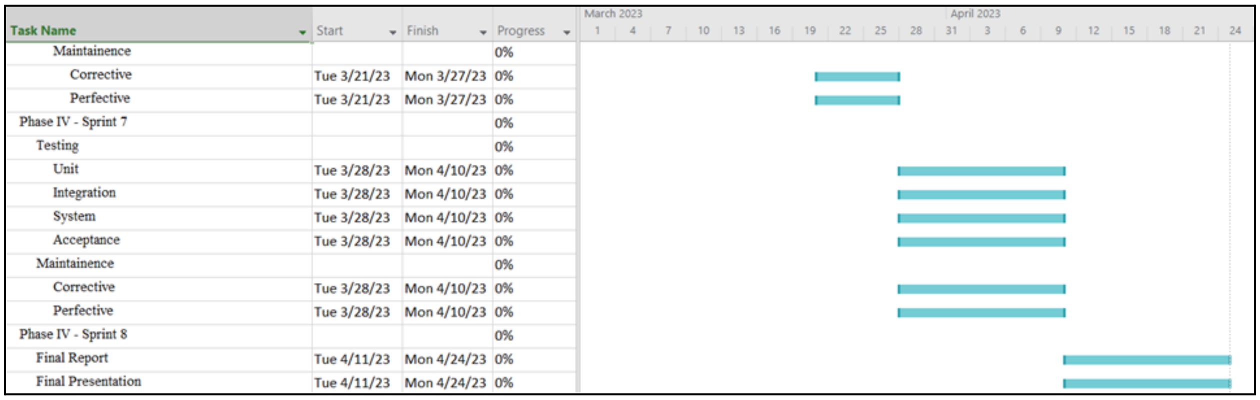
4.0 Project Plan



Task Name	Start	Finish	Progress
Phase I - Sprint 1			100%
Requirement Elicitation			100%
Market Research	Mon 1/9/23	Fri 1/13/23	100%
Brainstorming	Mon 1/9/23	Fri 1/13/23	100%
Feasibility Assessment	Mon 1/9/23	Fri 1/13/23	100%
Project Proposal	Sat 1/14/23	Thu 1/19/23	100%
Phase I - Sprint 2			100%
Skill Development	Fri 1/20/23	Mon 1/30/23	100%
General Research	Fri 1/20/23	Mon 1/30/23	100%
Phase II - Sprint 3			100%
Functional Decomposition	Tue 1/31/23	Mon 2/6/23	100%
Use Case Description Tables	Tue 2/7/23	Sat 2/11/23	100%
Gantt Chart	Tue 2/7/23	Sat 2/11/23	100%
Midway Presentation	Sun 2/12/23	Mon 2/13/23	100%
Phase III - Sprint 4			0%
Use Case Diagrams	Tue 2/14/23	Thu 2/16/23	0%
Class Diagrams	Fri 2/17/23	Thu 2/23/23	0%
Sequence Diagrams	Fri 2/24/23	Mon 2/27/23	0%
Phase III - Sprint 5			0%
Account Management			0%
UC01	Tue 2/28/23	Fri 3/3/23	0%
UC02	Tue 2/28/23	Fri 3/3/23	0%
UC03	Tue 2/28/23	Fri 3/3/23	0%
UC04	Sat 3/4/23	Mon 3/6/23	0%
UC05	Sat 3/4/23	Mon 3/6/23	0%
Testing			0%
Unit	Tue 2/28/23	Mon 3/6/23	0%
Integration	Tue 2/28/23	Mon 3/6/23	0%
Acceptance	Tue 2/28/23	Mon 3/6/23	0%
Maintainence			0%
Corrective	Tue 2/28/23	Mon 3/6/23	0%
Perfective	Tue 2/28/23	Mon 3/6/23	0%
Create Items and Inventories			0%
UC06	Tue 3/7/23	Fri 3/10/23	0%
UC07	Tue 3/7/23	Fri 3/10/23	0%



Task Name	Start	Finish	Progress
UC10	Tue 3/7/23	Fri 3/10/23	0%
UC11	Sat 3/11/23	Mon 3/13/23	0%
UC12	Sat 3/11/23	Mon 3/13/23	0%
Testing			0%
Unit	Tue 3/7/23	Mon 3/13/23	0%
Integration	Tue 3/7/23	Mon 3/13/23	0%
Acceptance	Tue 3/7/23	Mon 3/13/23	0%
Maintainence			0%
Corrective	Tue 3/7/23	Mon 3/13/23	0%
Perfective	Tue 3/7/23	Mon 3/13/23	0%
Phase III - Sprint 6			0%
Create Shopping Lists, Stores, and Alerts			0%
UC15	Tue 3/14/23	Fri 3/17/23	0%
UC16	Tue 3/14/23	Fri 3/17/23	0%
UC08	Tue 3/14/23	Fri 3/17/23	0%
UC09	Sat 3/18/23	Mon 3/20/23	0%
UC19	Sat 3/18/23	Mon 3/20/23	0%
UC21	Sat 3/18/23	Mon 3/20/23	0%
Testing			0%
Unit	Tue 3/14/23	Mon 3/20/23	0%
Integration	Tue 3/14/23	Mon 3/20/23	0%
Acceptance	Tue 3/14/23	Mon 3/20/23	0%
Maintainence			0%
Corrective	Tue 3/14/23	Mon 3/20/23	0%
Perfective	Tue 3/14/23	Mon 3/20/23	0%
Share Inventories and Shopping Lists			0%
UC20	Tue 3/21/23	Fri 3/24/23	0%
UC13	Tue 3/21/23	Fri 3/24/23	0%
UC14	Tue 3/21/23	Fri 3/24/23	0%
UC17	Sat 3/25/23	Mon 3/27/23	0%
UC18	Sat 3/25/23	Mon 3/27/23	0%
Testing			0%
Unit	Tue 3/21/23	Mon 3/27/23	0%
Integration	Tue 3/21/23	Mon 3/27/23	0%
Acceptance	Tue 3/21/23	Mon 3/27/23	0%



Task Name	Start	Finish	Progress
Maintainence			0%
Corrective	Tue 3/21/23	Mon 3/27/23	0%
Perfective	Tue 3/21/23	Mon 3/27/23	0%
Phase IV - Sprint 7			0%
Testing			0%
Unit	Tue 3/28/23	Mon 4/10/23	0%
Integration	Tue 3/28/23	Mon 4/10/23	0%
System	Tue 3/28/23	Mon 4/10/23	0%
Acceptance	Tue 3/28/23	Mon 4/10/23	0%
Maintainence			0%
Corrective	Tue 3/28/23	Mon 4/10/23	0%
Perfective	Tue 3/28/23	Mon 4/10/23	0%
Phase IV - Sprint 8			0%
Final Report	Tue 4/11/23	Mon 4/24/23	0%
Final Presentation	Tue 4/11/23	Mon 4/24/23	0%

5.0 Appendix

5.1 Supporting Technologies

Name	Category	Usage	References
HTML	Prog. Language	Structure web content	3
CSS	Prog. Language	Style web content	3
JS	Prog. Language	Make web content dynamic	3, 4
SQL	Prog. Language	Read and edit the project databases	3, 4
NodeJS	Web Server	Serve web content to clients	4, 5
MySQL	DBMS	Manage the project databases	4, 6
VSCode	IDE	Write the project code	7
GitHub	VCS	Manage the code repository	8
Zoom	Comms	Facilitate remote team meetings	9
WhatsApp	Comms	Facilitate asynchronous team communication	10

5.2 References

1	Scrum Information	Pressman, Roger S. Software Engineering: A Practitioner's Approach 9th ed. MCGRAW-HILL COMPANIES, 2005
2	Proposal Format Inspiration	https://templatelab.com/project-proposal-templates/
3	HTML, CSS, JS, SQL Tutorials	https://www.w3schools.com/default.asp
4	NodeJS-MySQL Tutorial	https://www.w3schools.com/nodejs/nodejs_mysql.asp
5	NodeJS Homepage	https://nodejs.org/en/
6	MySQL Homepage	https://www.mysql.com/
7	VSCode Homepage	https://code.visualstudio.com/
8	GitHub Homepage	https://github.com/
9	Zoom Homepage	https://zoom.us/
10	WhatsApp Homepage	https://www.whatsapp.com/