Group 4 - Thao Nguyen, Hongjin Wu, Swetha Sarma , Amelia Mazer

Information System Analysis and Design - Spring 2022 Final Project

March 7th, 2022

StayFresh Produce Management System

A. PROJECT PROPOSAL

System Request - StayFresh Produce Management System
Project Sponsor: CEO of StayFresh - Yasser Dessouky VP of StayFresh - Sudakshina Kumar
Business Need - Problem Statement: Disruption from Coronavirus has been especially significant for the restaurant industry. Roughly 80,000 restaurants have temporarily or permanently closed since the start of the pandemic, according to estimates from the National Restaurant Association. It is more important than ever for restaurants to manage their inventory effectively. Inventory management helps restaurants keep the right amount of food and ingredients on hand so they have enough stock to serve all customers but also avoid spoilage and loss. Restaurants are more likely to find long-term success if they practice effective inventory management. StayFresh Produce Management System will allow midsize-to-large restaurants to manage their fresh produce inventory better and create a channel for restaurants to give back to their communities. What the system will do: ● Monitor fresh produce inventory level and pre-fill purchase order for low-stock items ● Monitor fresh produce's expiration dates and notify of items that are about to expired ● Connect the network of StayFresh restaurants with food banks and allows for food donation matching
Business Requirements: **A. Inventory Management** 1. The system should track the inventory level of all fresh produce items . 2. The system should alert when inventory of a fresh produce item falls to the reorder

point and pre-fills a purchase order for that item.

3. The system should allow for manual reconciliation of inventory.

B. Expiration Management

4. The system should track the expiration dates for all fresh produce items in inventory and alert when expiration dates are close by.

C. Donation Partnership Management

5. The system should allow the restaurants in StayFresh network to upload items up for donation, and allow food bank partners to register for those items

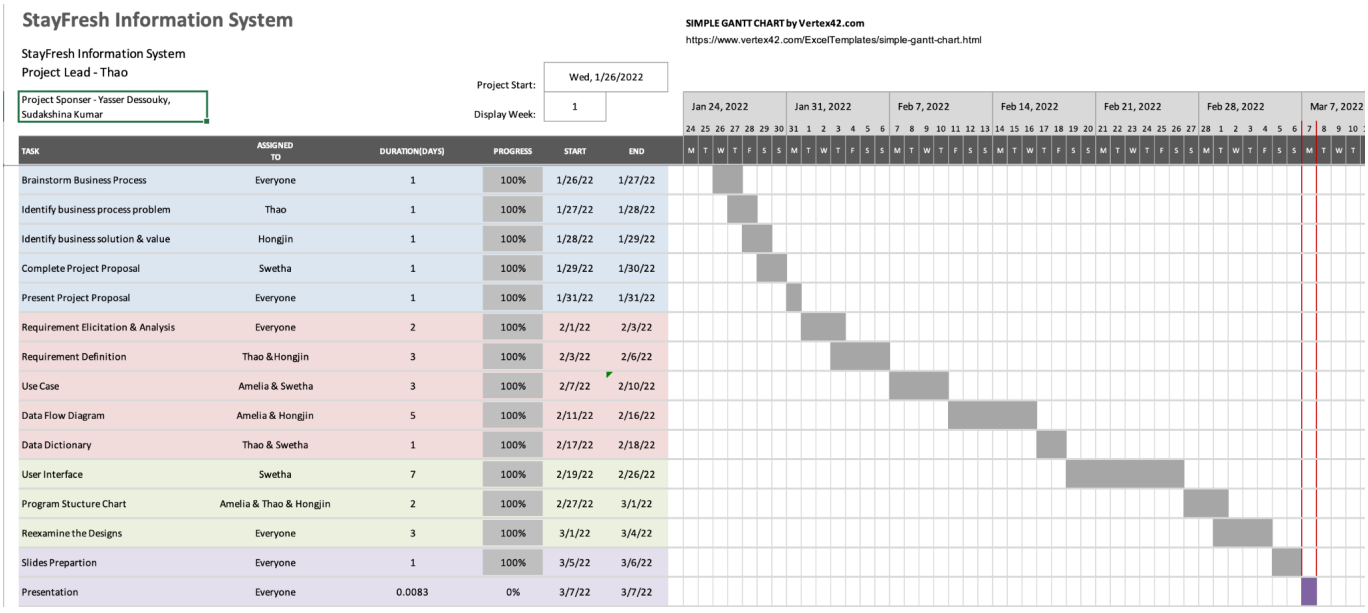
Business Value:

- Reduce food waste and spoilage by at least 50%
- Enhance restaurant's social image with donation partnership with food banks
- Enhance customer satisfaction with fresher produce

Special Issues or Constraints:

- Food inventory management is based upon the FIFO (first in first out) principles
- Expiry dates for fresh produce are estimated

B. GANTT CHART



StayFresh Information System

StayFresh Information System
Project Lead - Thao

Project Sponser - Yasser Dessouky, Sudakshina Kumar

Project Start: Wed, 1/26/2022
Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Jan 24, 2022 | Jan 31, 2022 | Feb 7, 2022 | Feb 14, 2022 | Feb 21, 2022 | Feb 28, 2022 | Mar 7, 2022

TASK	ASSIGNED TO	DURATION(DAYS)	PROGRESS	START	END
Brainstorm Business Process	Everyone	1	100%	1/26/22	1/27/22
Identify business process problem	Thao	1	100%	1/27/22	1/28/22
Identify business solution & value	Hongjin	1	100%	1/28/22	1/29/22
Complete Project Proposal	Swetha	1	100%	1/29/22	1/30/22
Present Project Proposal	Everyone	1	100%	1/31/22	1/31/22
Requirement Elicitation & Analysis	Everyone	2	100%	2/1/22	2/3/22
Requirement Definition	Thao &Hongjin	3	100%	2/3/22	2/6/22
Use Case	Amelia & Swetha	3	100%	2/7/22	2/10/22
Data Flow Diagram	Amelia & Hongjin	5	100%	2/11/22	2/16/22
Data Dictionary	Thao & Swetha	1	100%	2/17/22	2/18/22
User Interface	Swetha	7	100%	2/19/22	2/26/22
Program Stucture Chart	Amelia & Thao & Hongjin	2	100%	2/27/22	3/1/22
Reexamine the Designs	Everyone	3	100%	3/1/22	3/4/22
Slides Prepartion	Everyone	1	100%	3/5/22	3/6/22
Presentation	Everyone	0.0083	0%	3/7/22	3/7/22

C. PROJECT REPORT

I. Business Description

StayFresh is a computer technology company that provides software solutions and business applications to restaurants. We believe it's our responsibility to protect and preserve the environment for generations to come. That means that we provide solutions to effectively manage fresh produce inventory, such as low inventory notification, expiration notifications, donation partnership management, and much more. Through the development of our system, we expect to reduce food waste by 50%, donate 100% of surplus food, and recycle food scraps.

II. Scope of the System - how/why the system will be used in the environment

The system will be used in the restaurant management environment. This system's purpose is to mitigate food wastage, improve efficiency when dealing with fresh produce and reduce costs to help increase profits. System will be used for fresh produce inventory management, expiration management and donation management.

Scope of the System:

- The system will allow restaurant employees to scan and record wholesaler's delivery items into a general item database.
- System will produce its own barcodes, and allow the employee to record fresh produce information, including weight and location of produce into the fresh produce database.
- The system will calculate and verify expiration dates. Notify the restaurant employee a few days before the fresh produce is about to expire.
- System will handle fresh produce inventory and notify the inventory manager when inventory is low.
- System will integrate with the restaurant POS system and generate reports and statistics about fresh produce usage, inventory count, and expiration.
- System will allow restaurants and food banks to register and match food banks using a matching interface.
- System will store information about registered restaurants and food banks.

Out of Scope:

- Handle expiration and low inventory of items, other than fresh produce.
- System will only be used to support restaurants

III. System Analysis and Design Specification

1. Requirement Definition

Functional Requirements

Process Oriented Requirements

1. The system should track the inventory level of all fresh produce items

    1.1 The system should be able to import the product information from the wholesaler's barcode.

    1.2 The system should be able to produce a StayFresh barcode using the wholesaler's information such as product expiration date, name, quantity etc.

    1.3 The system should be able to log the location of products in the storing unit.

    1.4 The system should be able to receive the item ID and the amount taken out of inventory from user input from mobile applications and record the information in the inventory history database.

    1.5 The system's mobile application should be able to recognize text from images.

2. The system should alert when inventory of a fresh produce item falls to the reorder point and pre-fills a purchase order for that item.

    2.1 The system receives an alert when the inventory level of an item falls to its reorder point.

    2.2 The system creates and pre-fills a PO for that item using details from its last PO.

    2.3 The system sends the low-inventory alert and the pre-filled PO to the inventory employee

    2.4 Once the system receives the approved PO, it forwards the PO to the supplier to place an order for that item

3. System allows for manual reconciliation of inventory.

    3.1 The system allows manual change of inventory count.

3.2 The system allows inventory information to be updated as needed.

3.3 Manual inventory update requires approval from the inventory manager or a level above the person who initiates the change.

4. The system should track the expiration dates for all fresh produce items in inventory and alert when expiration dates are close by

    4.1 The system should be able to extract the expiration date for each specific product

    4.2 The system should notify the manager four days before the expiration date

    4.3 If the expiration date is unknown, the system should be able to extract the expiration date from the static table

    4.4 The system should be able to calculate the expiration date based off of the expired duration based on the static table

    4.5 The system should notify the manager on the day of expiration to throw out expired products.

5. The system should allow the restaurants in StayFresh network to upload items up for donation, and allow food bank partners to register for those items

    5.1 The system should allow external user (food banks/soup kitchens) to create an account

    5.2 The system should allow external users to select items for their order and pickup times.

    5.3 The system allows the employee to manually input the eligible items for donation.

    5.4 The system should be able to match the restaurant and the food bank/soup kitchen based on the items selected.

    5.5 The system notifies the external user when their order/request has been fulfilled and confirmed.

    5.6 The system should be able to relay the restaurant location, time for pickup and order details.

Information Oriented Requirements

1. The system must have a static table that contains the estimated expiration time for different types of fresh produce.

2. The system must have an item information data store that contains general item information from suppliers.

3. The system must have a fresh produce inventory data store that contains inventory information of the produce(name, expiration date etc)

4. The system must have a external user data store that contains external user information

5. The system must have a donation item data store that contains eligible item information for donation

Non-Functional Requirements

1. Operation
   - **1.1** The system should be compatible with the mobile application
   - **1.2** The system can run on any type of devices (PCs, tablets, mobiles, etc)
   - **1.3** The system will integrate with the existing inventory data stores.
   - **1.4** The system will integrate with the existing procurement system.

2. Performance
   - **2.1** The system should be able to accommodate 30-40 users concurrently
   - **2.2** The expiration date static table should be updated monthly
   - **2.3** The system should be available for use 24 hours per day, 365 days per year

3. Security
   - **3.1** User access controls and permissions can only be changed by the system's administrator.
   - **3.2** The system must include all available safeguards to protect from viruses and malwares (eg. worms, Trojan horses, ransomwares, etc)
   - **3.3** The website must include a 256-bit SSL security certificate to properly secure data transferred between the user's browser and the website.

4. Cultural and Political
   - **4.1** Sensitive information will be protected under the Data Protection Act
   - **4.2** System must have the option to be in languages other than English
   - **4.3** System should have ADA compliance and web accessibility.

4.4 System should be able to accommodate imperial and metric system

2. Use Case

Business Requirement 1 The system should track the inventory level of all fresh produce items

Use Case Name: Check In Fresh Produce Items	**ID: UC - 001**	**Priority:** High
Brief Description: This use case describes how the kitchen personnel record new arrived fresh produce items.		
Actor: Kitchen Personnel		
Trigger: Fresh produce purchase order is delivered to the restaurant **Type** ☑External Temporal		
Preconditions: 1. Kitchen personnel are authenticated. 2. Fresh produce inventory data store is available and on-line. 3. General Item information data store is available and on-line. 4. The estimated expiration time table is available to access. 5. PO datastore is available and on-line		
Normal Course	**Information for Steps**	
1. Kitchen personnel scans the wholesaler's barcode on the delivered package.	← Produce Item Barcode	
2. System checks for item information	← Item Information	
3. System creates StayFresh barcode according to the item information	→ StayFresh code	
4. Kitchen personnel break the package of produce items to storable smaller boxes and create StayFresh labels for each box.	← StayFresh code	
5. Kitchen personnel specifies the ID of the location where the boxes of items will be stored in a storing unit.	← Location ID	
6. Systems stores and updates the fresh produce inventory information.	→ Inventory Information	
Alternative Course(s)		

A1: No expiration data in the item information (branch at step 6) 1. System checks for an estimated expiration time in the estimated expiration time table 2. System calculates an expiration date according to the above time and the date of delivery for the item.	← Estimated Expiration Time (static expiration datastore) ← Delivery Date (PO data store) → Estimated Expiration Date

Post conditions:
1. Fresh produce inventory data store is updated.

Exceptions:
E1: Item information no found (occurs at step 2)
1. The system displays message that bar code didn't recognize and asks the personnel to scan the barcode again
2. Return to Normal Course (step 3) if information found, otherwise continue to the next step
3. The system stores the wholesaler's bar code with default information
4. The system sends alert message to inventory manager
5. Return to Normal Course (step 3) with default information

Summary:

Inputs	Source	Outputs	Destination
Produce Item Barcode	Kitchen Personnel	StayFresh Code	Fresh Produce inventory datastore
Item Information	General Item Information Data Store	Inventory Information	Fresh Produce Inventory Data Store
StayFresh code	Fresh Produce Inventory Data Store	Estimated Expiration Date	Fresh Produce Inventory Data Store
Location ID	Kitchen Personnel		
Estimated Expiration Time	Estimated Expiration Time Table		
Delivery Date	PO Datastore		

Business Requirement 1 : The system should track the inventory level of all fresh produce items

Use Case Name: Check Out Fresh Produce Items	**ID: UC - 002**	**Priority:** High
Brief Description: This use case describes how the kitchen personnel check out fresh produce times from the storing unit		

Actor: Kitchen Personnel

Trigger: Kitchen personnel needs to take out fresh produce from the storing unit
Type ☑ External ☐ Temporal

Preconditions:
1. Kitchen personnel is authenticated
2. Fresh produce inventory data store is available and on-line
3. Item information data store is available and on-line
4. Inventory history data store is available and on-line

Normal Course	**Information for Steps**
1. Kitchen personnel scans StayFresh barcode on the box of the item being removed	← StayFresh code ← Item Information
2. Kitchen personnel weights the removal items on StayFresh Scale and scans to read the weight	← Removal Item Weight
3. Kitchen personnel confirms the weight read	← Removal Item Weight Confirmation
4. System logs the removal history	→ Produce Usage Information
5. System updates inventory	→ Updated Inventory Information
Alternative Course(s) N/A	

Post conditions:
1. Removal information is logged
2. Fresh produce inventory data store is updated

Exceptions:
N/A

Summary:

Inputs	**Source**	**Outputs**	**Destination**
StayFresh code	Kitchen personnel	Produce Usage Information	Fresh Produce Usage History Data Store
Item Information	Fresh Produce Inventory Data Store	Updated Inventory Information	Fresh Produce Inventory Data Store
Removal Item Weight	Kitchen personnel		
Removal Item Weight Confirmation	Kitchen personnel		

Business Requirement 2: The system should alert when inventory of a fresh produce item falls to the reorder point and pre-fills a purchase order for that item.

<table>
<tr><td colspan="2">Use Case Name: Alert Low Inventory</td><td>ID: UC -003</td><td>Priority: High</td></tr>
<tr><td colspan="4">Brief Description: This use case describes how the system alerts the inventory manager that an item’s stock has fallen to the reorder point.</td></tr>
<tr><td colspan="4">Actor: System</td></tr>
<tr><td colspan="4">Trigger: Item’s stock is fallen to the reorder point
Type: ☑ External □ Temporal</td></tr>
<tr><td colspan="4">Preconditions:
1. The reorder point of the item was calculated and recorded in the inventory data store
2. Restaurant personnel is authenticated
3. Fresh produce inventory data store is available and on-line
4. Item information data store is available and on-line</td></tr>
<tr><td colspan="2">Normal Course</td><td colspan="2">Information for Steps</td></tr>
<tr><td colspan="2">1. The system creates a new PO and pre-fills the form with details, including supplier information, item quantity and unit cost from the last PO of that item</td><td colspan="2">← Item’s last PO</td></tr>
<tr><td colspan="2">2. The system sends the low-stock notification with the pre-filled PO to the inventory employee</td><td colspan="2">→ Low stock notification
→ System-generated PO</td></tr>
<tr><td colspan="2">3. The inventory employee reviews the system-generated PO and makes adjustments if needed,</td><td colspan="2">← PO revision</td></tr>
<tr><td colspan="2">4. The system sends the employee-generated PO to manager for approval</td><td colspan="2">→ Employee-generated PO</td></tr>
<tr><td colspan="2">5. The inventory manager approves the PO</td><td colspan="2">← PO approval</td></tr>
<tr><td colspan="2">6. The system forwards the approved PO to the vendor specified inside the order</td><td colspan="2">→ Manager-generated PO</td></tr>
<tr><td colspan="2">Alternative Course(s)

A1: The employee-generated PO is rejected by the inventory manager (branch at step 4)</td><td colspan="2">← PO rejection</td></tr>
<tr><td colspan="2">1. The system notifies the inventory employee of the rejection</td><td colspan="2">→ PO rejection notification</td></tr>
<tr><td colspan="2">2. The inventory manager makes adjustments, and approves the PO</td><td colspan="2">← PO revision
← PO approval</td></tr>
</table>

3. Return to normal course (step 5)	
Post conditions: 1. The PO for the low-stock item is sent to the supplier	
Exceptions: N/A	
Summary:	

Inputs	**Source**	**Outputs**	**Destination**
Item's last PO	Purchase order data store	Low stock notification	Inventory employee
PO revision	Inventory employee	System-generated PO	Inventory employee
PO approval	Inventory manager	Employee-generated PO	Inventory manager
PO rejection	Purchase order data store	Manager-generated PO	Supplier
PO revision	Inventory manager	PO rejection notification	Inventory employee

Business Requirement 3: System allows for manual reconciliation of inventory.

Use Case Name: Manual Update of Inventory Information	**ID: UC - 004**	**Priority:** High
Brief Description: This use case describes how the user updates inventory information manually in the system		
Actor: Inventory employee		
Trigger: A need to manually adjust inventory arises **Type:** ☑ External □ Temporal		
Preconditions: 1. The user is authenticated and logged in.		

Normal Course	**Information for Steps**
1. The user searches for the inventory item he/she wants to update	← Searched keyword
2. System displays searched results and the user chooses the item he/she wants to update	→ Searched results ← Item selection
3. The user chooses the reason for the manual change from a list of reasons or adds his/her own.	← Reason for manual inventory change
4. The user provides the new inventory count for that item	← New inventory count
	→ Inventory information → New inventory count

5. The system forwards inventory information, with the updated count and the reason for manual change to the inventory manager for approval 6. The inventory manager approves the update 7. The system records the change in the inventory data store	→ Reason for manual inventory change ← Approval for the update → New inventory count
Alternative Course(s) N/A	
Post conditions: 1. Inventory data store is updated with new information.	
Exceptions: E1. Upper management rejects the change 1. The system notifies the employee of the rejection 2. The system closes the session	

Summary:

Inputs	**Source**	**Outputs**	**Destination**
Searched keyword	Inventory Employee	Searched results	Inventory Employee
Item selection	Inventory Employee	Inventory information	Inventory Manager
Reason for manual inventory change	Inventory Employee	New inventory count	Inventory Manager
New inventory count	Inventory Employee	Reason for manual inventory change	Inventory Manager
Approval for update	Inventory Manager	New inventory count	Fresh Produce Inventory Data Store

Business Requirement 4: The system should track the expiration dates for all fresh produce items in inventory and alert when expiration dates are close by

Use Case Name: Track and Alert Expiration Date	**ID: UC -005**	**Priority:** High
Brief Description: This use case is about the tracking of the expiration dates for each of the fresh produce that StayFresh stores.		
Actor: System		
Trigger: When the item details are collected and item is stored **Type** External ☑Temporal		
Preconditions:		

The system has extracted the expiration date for the specified product from the item database	
Normal Course	**Information for Step**
1. Every night, the system will check the current date with the items expiration date to see if they need to send out notifications to the managers	← Expiration Date ← Current Date (Fresh Produce Inventory Data Store)
2. If the date is prior to four days to the expiration date the manager will be notified that the item will expire in four days	← Expiration Date ← Current Date → Notification for the Manager four days left
3. On the day of expiration, if the item is still present the inventory manager will be notified to compost the expired items	← Current Date ← Expiration Date → Notification for the Manager to compost the items
4. When the item is expired the item status is changed to "Expired".	← Item Status (Fresh Produce Inventory Data Store) → Updated Item Status (Fresh Produce Inventory Data Store)
Alternative Course(s): N/A	
Post conditions: 1. The end result should be that the item database contains the expiration date for the item and the manager is notified when the date comes closer or it is the day of expiration.	
Exceptions: N/A	

Summary:

Inputs	Source	Outputs	Destination
Expiration Date	Fresh Produce Inventory Data Store	Notification for the Manager four days left	Manager
Current Date	Fresh Produce Inventory Data Store	Notification for the manager to compost items	Manager
Item Status	Fresh Produce Inventory Data Store	Updated Item Status	Fresh Produce Inventory Data Store

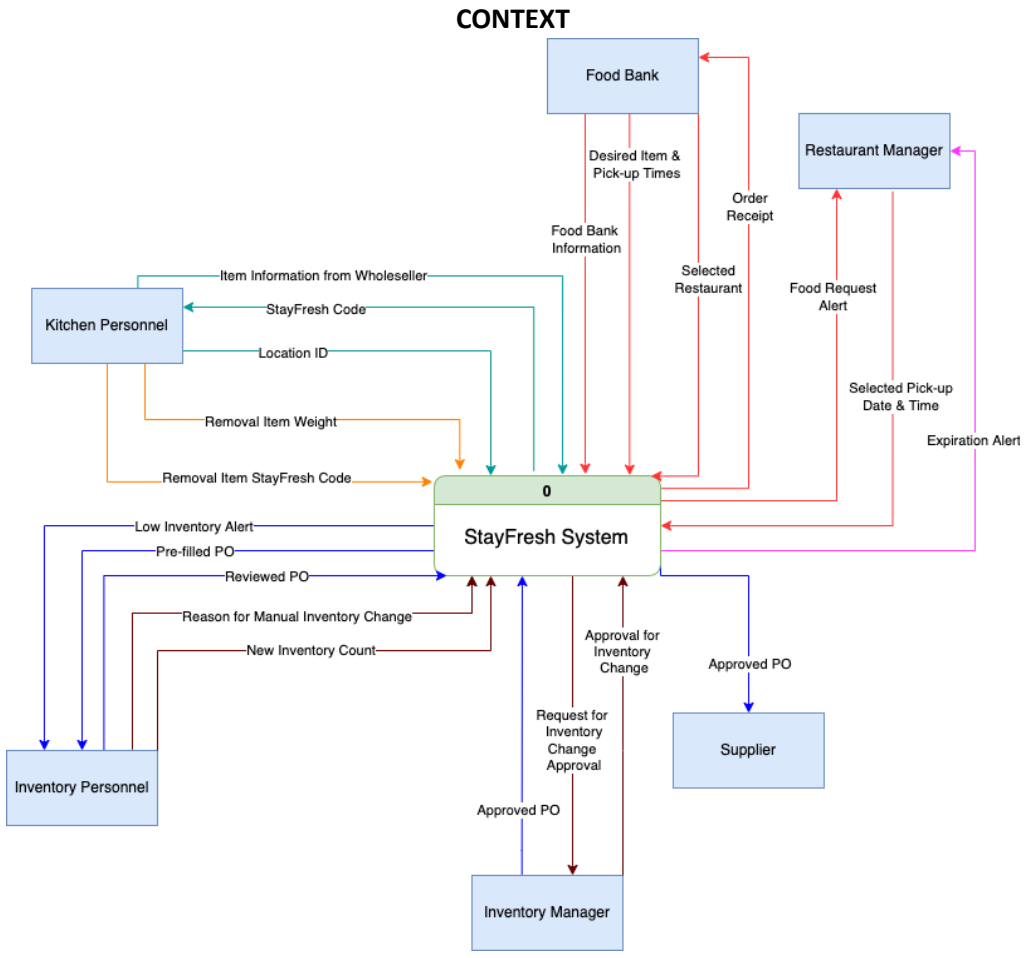
Business Requirement 5: The system should allow the restaurants in StayFresh network to upload items up for donation, and allow food bank partners to register for those items

Use Case Name: Match Food Banks to Restaurants	**ID: UC -006**	**Priority: Low**

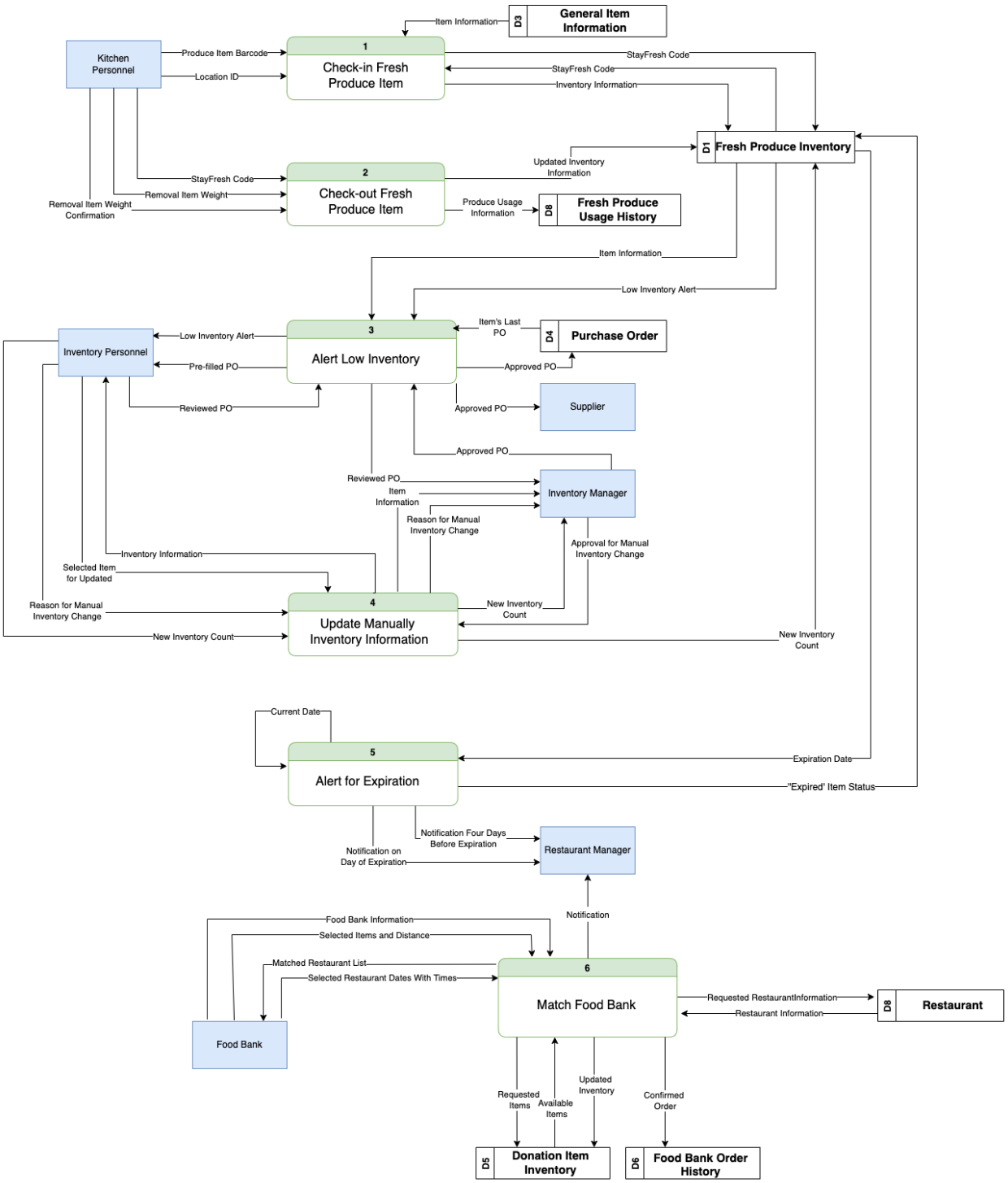
Brief Description: This use case describes how a food bank gets matched to a restaurant that has specific items that the food bank wants.	
Actor: Food Bank	
Trigger: Food Bank needs to stock up on items **Type** ☑ External Temporal	
Preconditions: 1. All restaurants using the StayFresh software is registered in the system 2. Employee from restaurant correctly inputs the list of donatable items into the Donation Item Inventory	
Normal Course	**Information for Steps**
1. The food bank creates an account on the user interface	←Food Bank Information
2. The food bank then selects the items that they are interested in, as well as a maximum distance	←Selected Items ←Maximum Distance
3. System then processes the food banks requests and searches for restaurants within that distance	→Processed Request
4. System outputs restaurants with match percentage(percentage of items actually match with the items inputted), distance away from food bank, available items and missing items as well	→Matched restaurants
5. The food bank then chooses the restaurants they are interested in and selects a pickup-date and time restaurant gives	←Selected Restaurants ←Selected date and time
6. The system sends a notification to the restaurant about a matched food bank	→Restaurant Notification
7. System then sends a receipt to the food bank with the restaurant location, items that are being picked up, and pick-up time and date.	→Processed Receipt →Confirmed Order →Updated Donation Item Data
Alternative Course(s) 1. Food Bank is already a registered customer and logs in instead	← Login Information
Post conditions: 1. Restaurant gets items ready for food bank pick-up 2. Food Bank goes to restaurant and picks up the items on the selected date and time	
Exceptions: N/A	
Summary: **Inputs** **Source** **Outputs** **Destination**	

Food Bank Information	Food Bank	Processed Request	Donation Item Inventory Data Store
Selected Items	Food Bank	Matched Restaurants	Food Bank
Maximum Distance	Food Bank	Restaurant Notification	Restaurant Manager
Selected Restaurant	Food Bank	Processed Receipt	Food Bank
Selected Dates and Times	Food Bank	Confirmed Order	Food Bank Order History Data Store
		Updated Donation Item Data	Donation Item Inventory Data Store

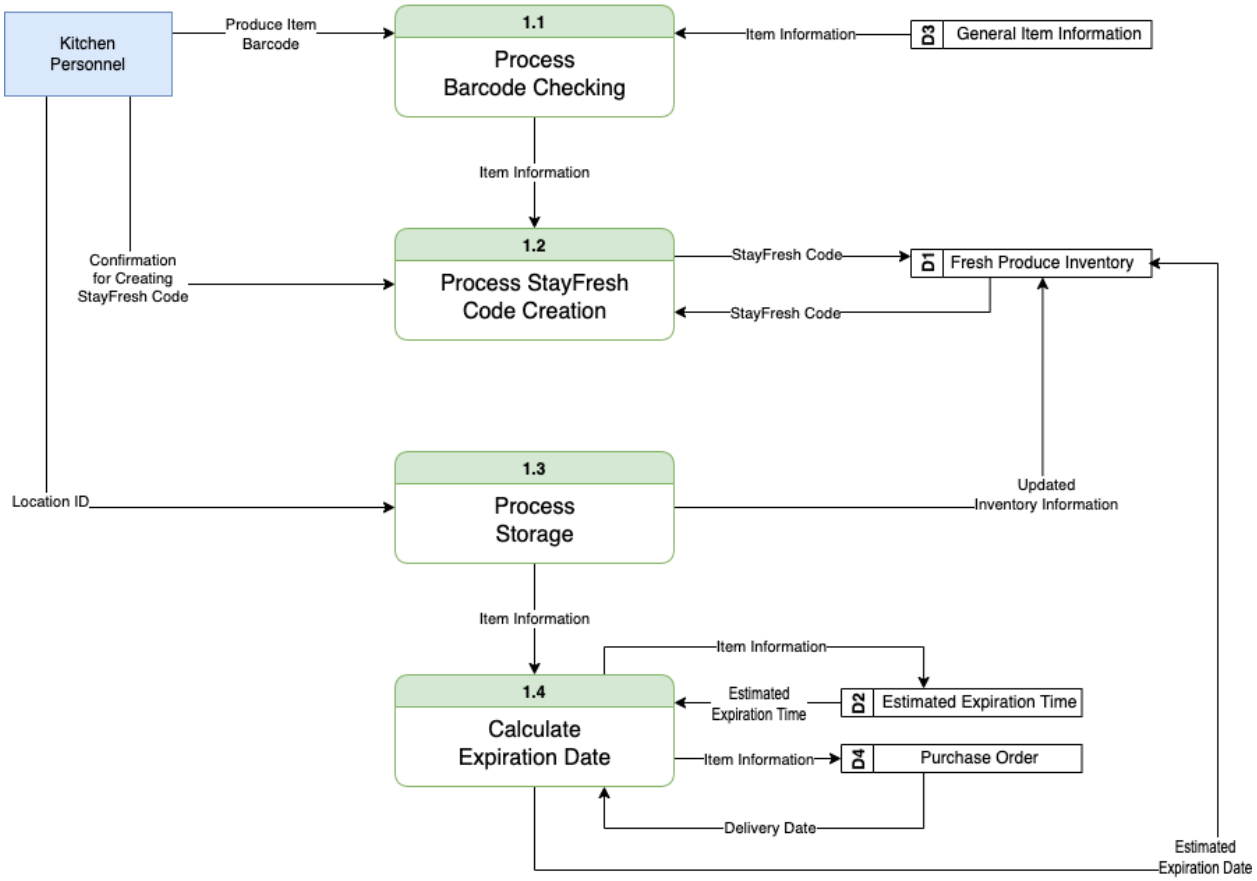
3. Data Flow Diagrams with Data Dictionary



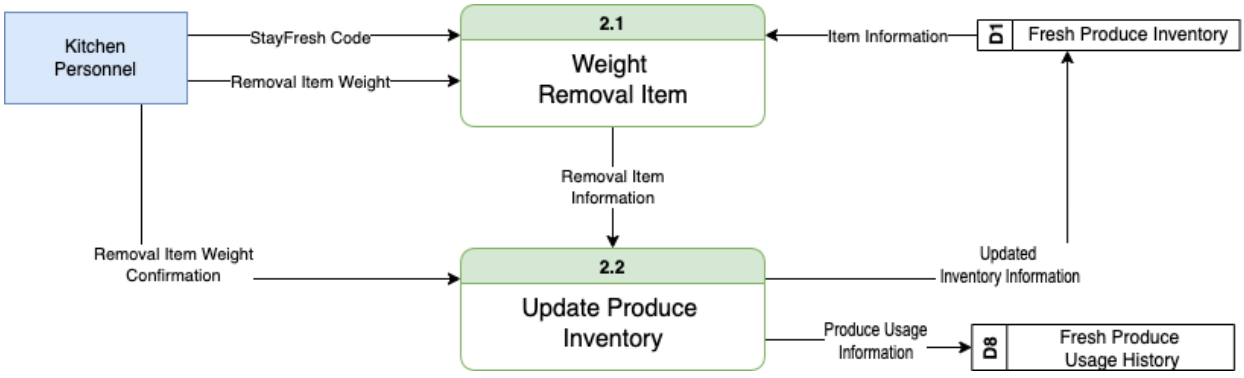
LEVEL 0


D3 General Item Information
Item Information
Kitchen Personnel
Produce Item Barcode
Location ID
1
Check-in Fresh Produce Item
StayFresh Code
StayFresh Code
Inventory Information
D1 Fresh Produce Inventory
Updated Inventory Information
StayFresh Code
Removal Item Weight
Removal Item Weight Confirmation
2
Check-out Fresh Produce Item
Produce Usage Information
D8 Fresh Produce Usage History
Item Information
Low Inventory Alert
Inventory Personnel
Low Inventory Alert
Pre-filled PO
Reviewed PO
3
Alert Low Inventory
Item's Last PO
D4 Purchase Order
Approved PO
Approved PO
Supplier
Approved PO
Reviewed PO
Item Information
Reason for Manual Inventory Change
Inventory Manager
Approval for Manual Inventory Change
Inventory Information
Selected Item for Updated
Reason for Manual Inventory Change
New Inventory Count
4
Update Manually Inventory Information
New Inventory Count
New Inventory Count
Current Date
5
Alert for Expiration
Expiration Date
"Expired' Item Status
Notification Four Days Before Expiration
Notification on Day of Expiration
Restaurant Manager
Notification
Food Bank Information
Selected Items and Distance
Matched Restaurant List
Selected Restaurant Dates With Times
6
Match Food Bank
Requested RestaurantInformation
Restaurant Information
D8 Restaurant
Food Bank
Requested Items
Available Items
Updated Inventory
Confirmed Order
D5 Donation Item Inventory
D6 Food Bank Order History

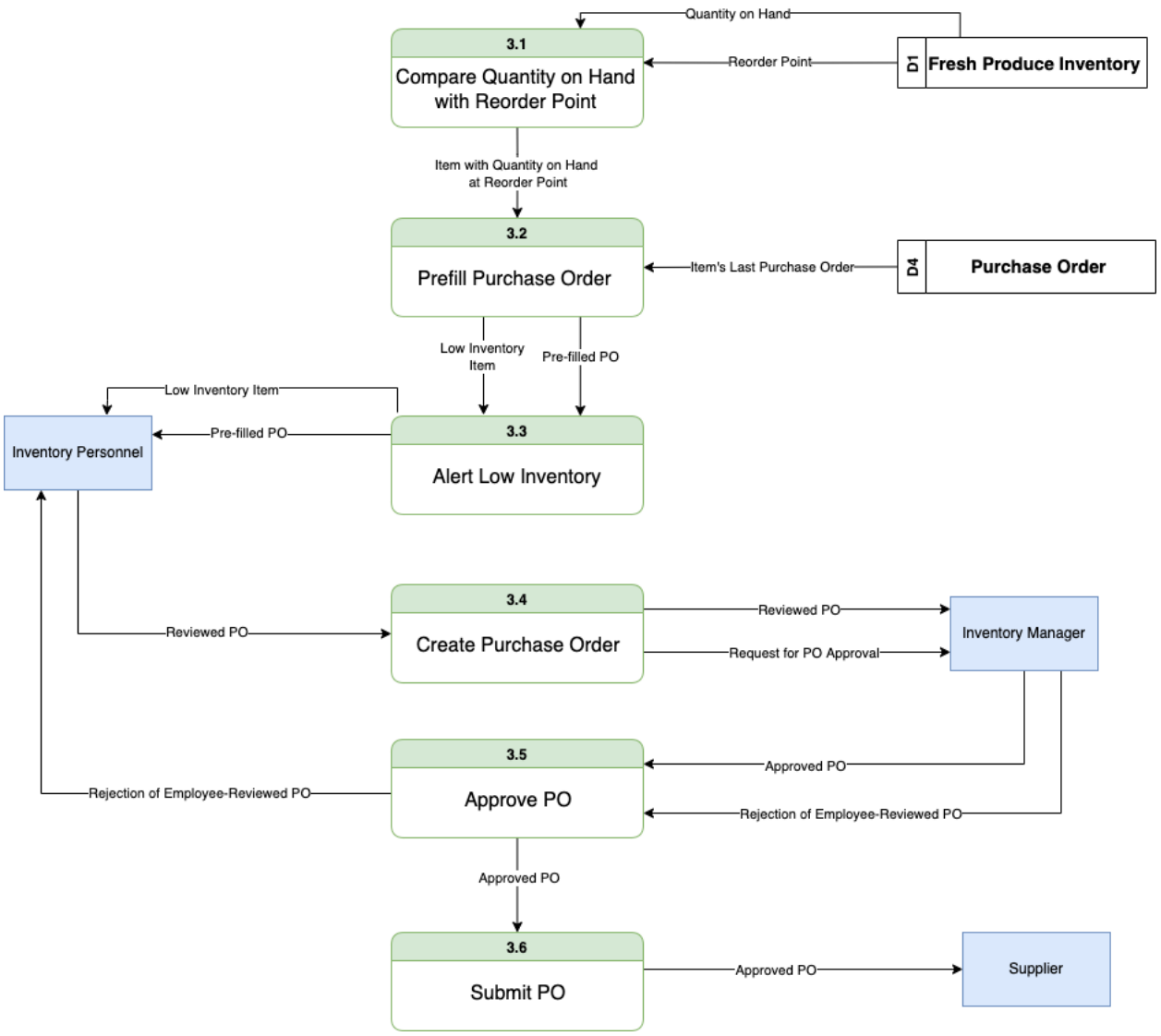
LEVEL 1 - Check in Fresh Produce



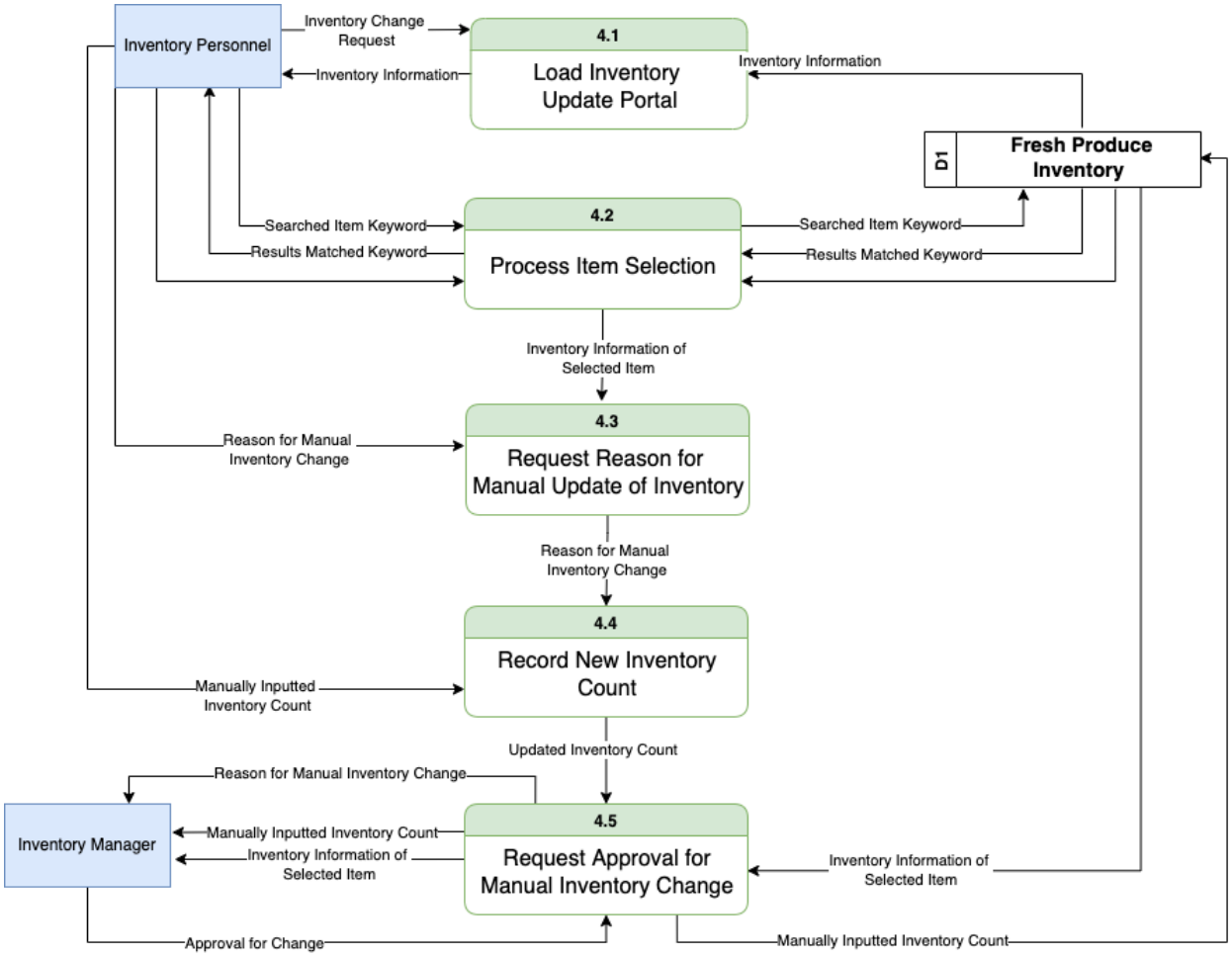
Level 1 - Check out Fresh Produce



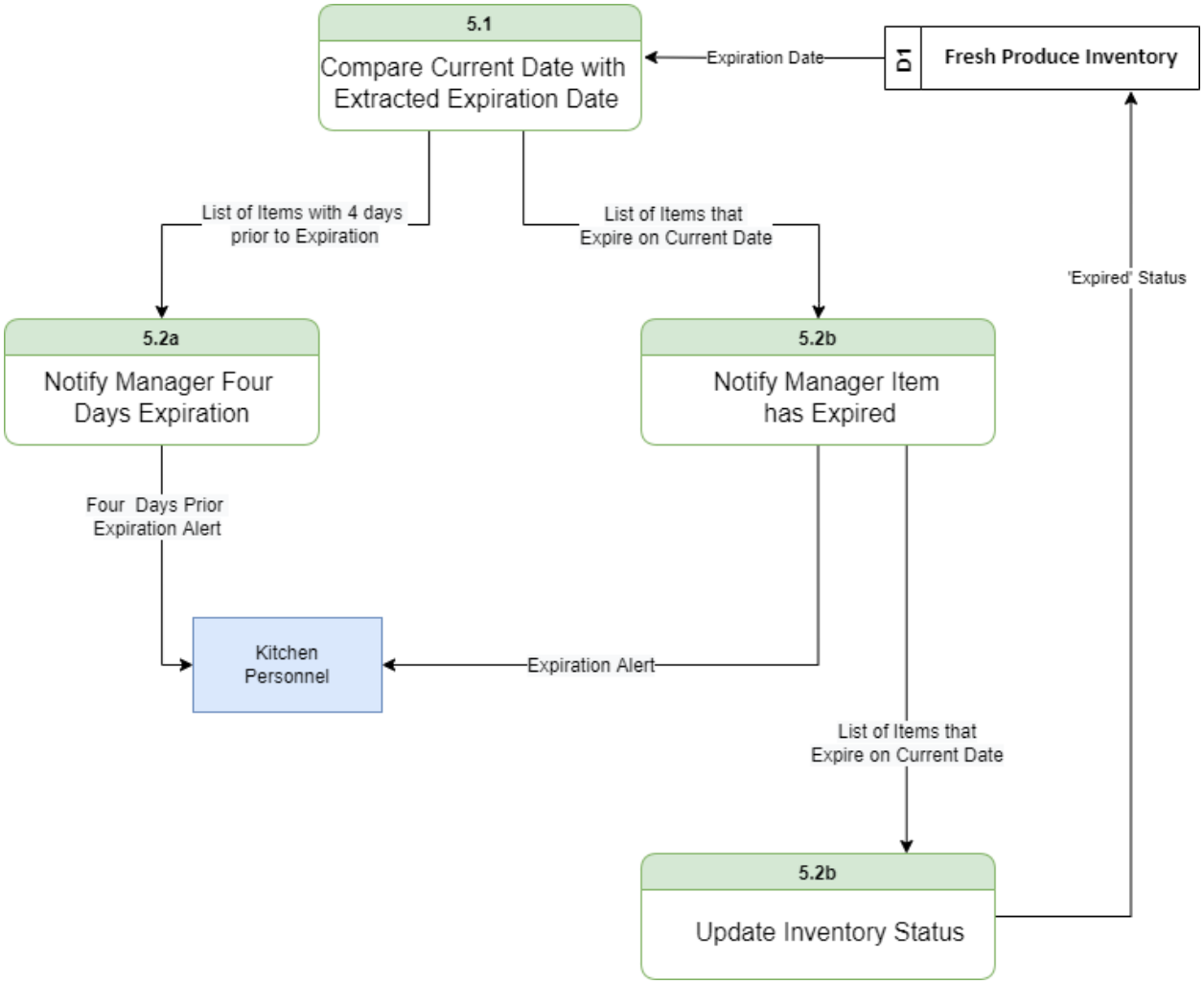
Level 1 - Alert Low Inventory



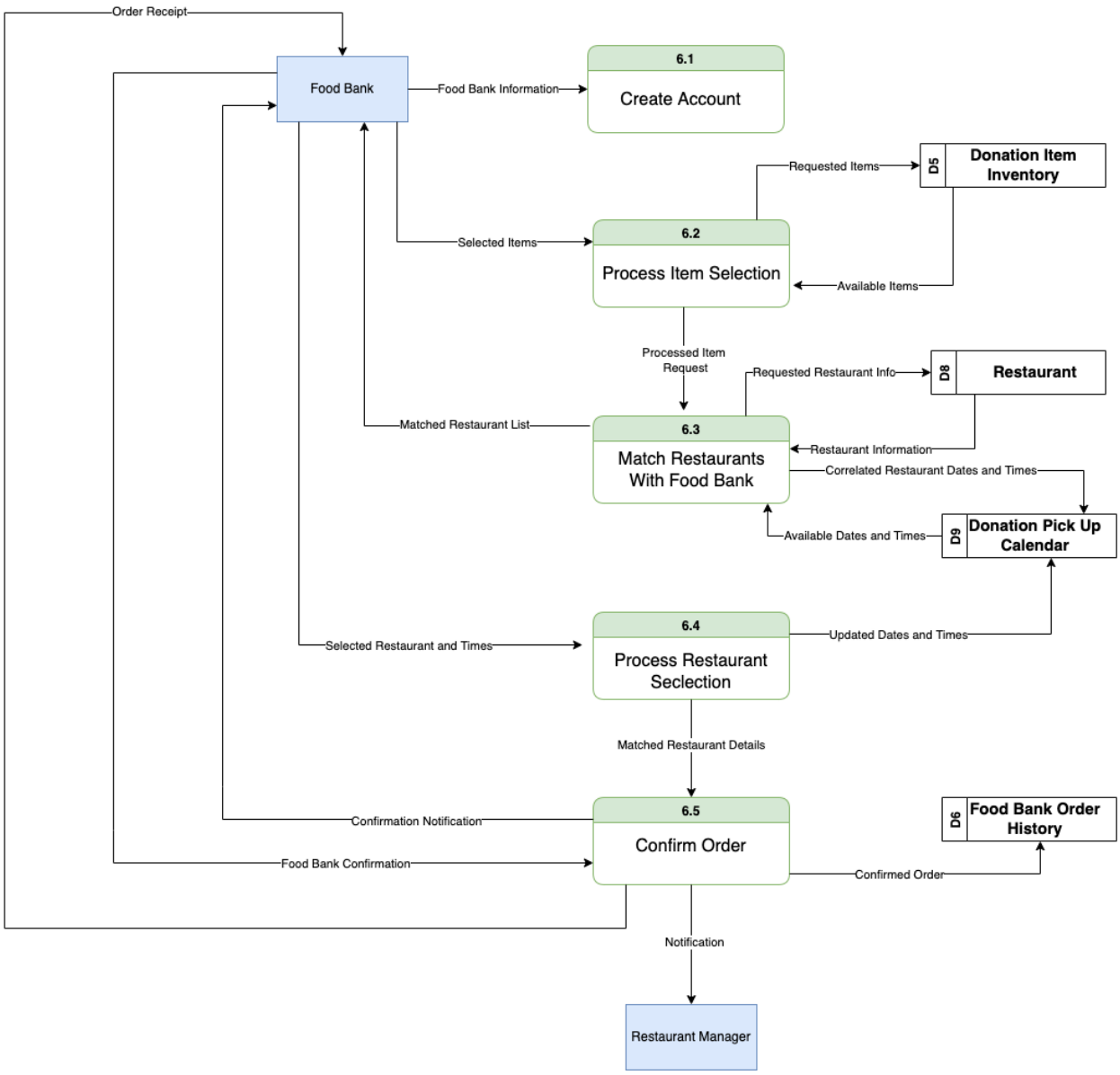
Level 1 - Manually Update Inventory



Level 1 - Alert Expired Produce



Level 1 - Match Food Bank



ENTITY DATA DICTIONARY

Restaurant Employee

Attribute	Data Type	Domain	Default Value
Employee ID	NUMBER	{00001-99,999}	Required
Name	TEXT	TEXT(150)	REQUIRED
Date of birth	DATE	{01/01/1900-12/31/9999}	NULL
Phone number	TEXT	TEXT(20)	NULL
Department	TEXT	TEXT(20)	NULL
Hire date	DATE	{01/01/1900-12/31/9999}	NULL
Manager	Boolean	{True, False}	FALSE

Supplier

Attribute	Data Type	Domain	Default Value
Supplier ID	NUMBER	{0001-9,999}	REQUIRED
Name	TEXT	TEXT(150)	NULL
Address	TEXT	TEXT(500)	NULL
Phone Number	TEXT	TEXT(20)	NULL
Email	TEXT	TEXT(255)	NULL

Food Bank

Attribute	Data Type	Domain	Default Value
Food Bank ID	NUMBER	{0001-9,999}	REQUIRED
Name	TEXT	TEXT(255)	REQUIRED
Employee Contact Name	TEXT	TEXT(255)	REQUIRED
Email	TEXT	TEXT(255)	REQUIRED
Contact Number	TEXT	TEXT(20)	NULL

DATA STORE DATA DICTIONARY

D1 Fresh Produce Inventory

Attribute	Data Type	Domain	Default Value
StayFresh Code	NUMBER	{1,000,000 - 9,999,999}	REQUIRED
Item Name	TEXT	TEXT(255)	NULL
Quantity	NUMBER	{0-100000}	0
Expiration Date	DATE	{01/01/1900 - 12/31/9999}	REQUIRED
Last Update	DATETIME	{01/01/1900 00:00:00 -12/31/9999 23:59:59}	NULL

D2 Estimated Expiration Time Table

Attribute	Data Type	Domain	Default Value
Entry ID	NUMBER	{0-999,999}	REQUIRED
Item Name	TEXT	TEXT(255)	NULL
Estimated Days Before Expired	NUMBER	{0-9999}	0

D3 General Item Information

Attribute	Data Type	Domain	Default Value
Item ID	NUMBER	{0000001-9,999,999}	REQUIRED
Item Name	TEXT	TEXT(255)	NULL
Expiration Date	DATE	{01/01/1900 - 12/31/9999}	NULL
Supplier ID	NUMBER	{0001-1000}	REQUIRED

D4 Purchase Order

Attribute	Data Type	Domain	Default Value
Purchase Order ID	NUMBER	{000001-999999}	REQUIRED
Supplier ID	NUMBER	{0001-9,999}	REQUIRED

Created By	NUMBER	{00001-99,999}	REQUIRED
Created At	DATETIME	{01/01/1900 00:00:00 -12/31/9999 23:59:59}	Today's Datetime
Approved By	NUMBER	{00001-99,999}	REQUIRED
Approved At	DATETIME	{01/01/1900 00:00:00 -12/31/9999 23:59:59}	Today's Datetime
Product ID	NUMBER	{0000001-9,999,999}	REQUIRED
Quantity	NUMBER	{0-999,999}	REQUIRED
Total	NUMBER	{0-99,999,999.00}	REQUIRED

D5 Donation Item Inventory

Attribute	**Data Type**	**Domain**	**Default Value**
Entry ID	NUMBER	{00001-99,999}	REQUIRED
Restaurant ID	NUMBER	{0001-9,999}	REQUIRED
Item ID	NUMBER	{0001-9,999}	REQUIRED
Item Name	TEXT	TEXT(255)	REQUIRED
Quantity	NUMBER	{0.00-99,999.00}	1.00

D6 Food Bank Order History

Attribute	**Data Type**	**Domain**	**Default Value**
Order ID	NUMBER	{00001-99,999}	REQUIRED
Food Bank ID	NUMBER	{0001-9,999}	REQUIRED
Pick-up Datetime	DATETIME	{01/01/1900 00:00:00 -12/31/9999 23:59:59}	REQUIRED
Item Name	TEXT	TEXT(255)	REQUIRED
Quantity	NUMBER	{0.00-10000.00}	0

D7 Fresh Produce Usage History

Attribute	Data Type	Domain	Default Value
Entry ID	NUMBER	{0001-9,999}	REQUIRED
StayFresh Code	NUMBER	{1,000,000 - 9,999,999}	REQUIRED
Quantity	NUMBER	{0.00-10,000.00}	0
Check-out Datetime	DATETIME	{01/01/1900 00:00:00 -12/31/9999 23:59:59}	REQUIRED

D8 Restaurant

Attribute	Data Type	Domain	Default Value
Restaurant ID	NUMBER	{0001-9,999}	REQUIRED
Restaurant Name	TEXT	TEXT(255)	REQUIRED
Address	TEXT	TEXT(255)	REQUIRED
Phone Number	TEXT	TEXT(255)	REQUIRED
Email	TEXT	TEXT(255)	REQUIRED

D9 Donation Pickup Calendar

Attribute	Data Type	Domain	Default Value
Item ID	NUMBER	{0001-9,999}	REQUIRED
Restaurant ID	NUMBER	{0001-9,999}	REQUIRED
Date	DATE	{01/01/1900 - 12/31/9999}	REQUIRED
Time	TIME	HHMM	REQUIRED

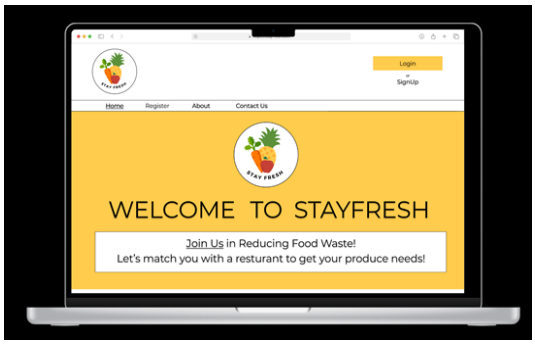
4. User Interface Design and Output Reports

Interface 1: Food Bank Order and Account Interface (Website)

Process 1: Creating an Account (for Food Bank Partners)

Step 1: Home Page and Create an Account either by clicking on Register or SignUp.

Fig 1 – Home Page



Step 2: When User Clicks on Register or SignUp, the registration page shows up and they type in their information.

Fig 2 – Registration Page for an Account with StayFresh

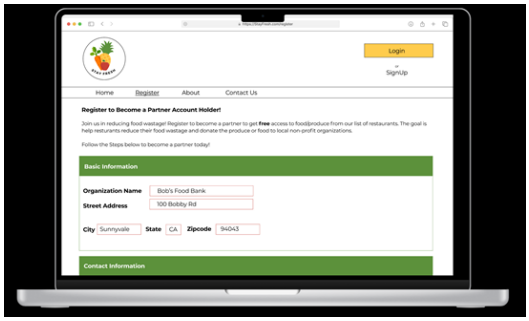


Fig 3 – Registration Page for an Account with StayFresh

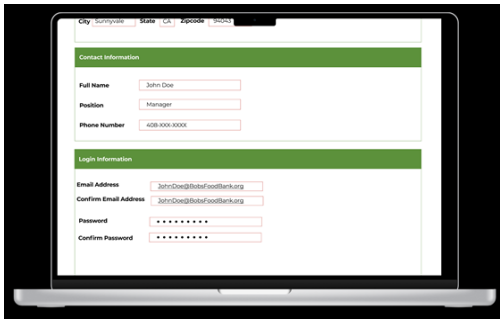
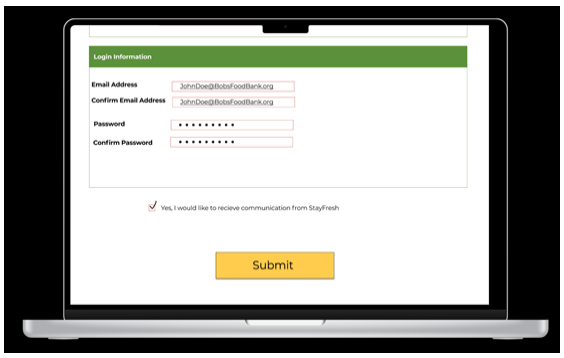
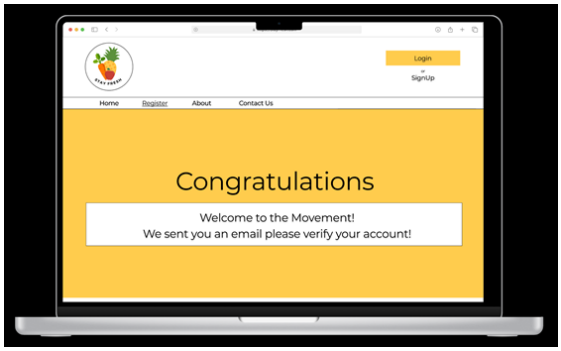


Fig 4 – Registration Page for an Account with StayFresh



Step 3: Once User has clicked Submit, there is a confirmation page along with further instructions on verifying the account.

Fig 5 – Confirmation Page



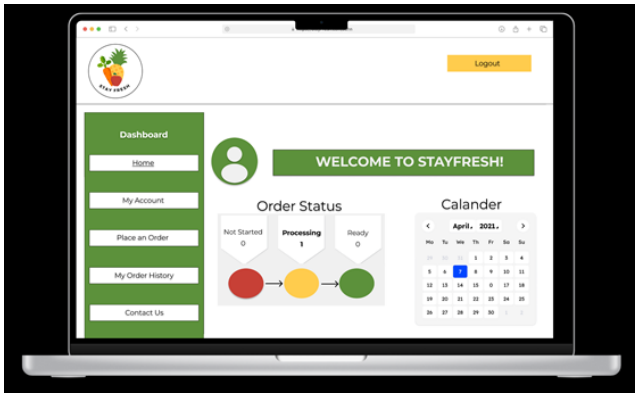
Process 2: Place an Order for Food Bank Partners (use case Matching)

Step 1: Once the user has created an account, go to the login page

Fig 6 – Food Bank Login Page

Step 2: Once the user has logged in, they enter their dashboard. To place an order click the 'Place an Order button'

Fig 7– Food Bank User Dashboard



Step 3: Enter the wanted items and the desired distance the user wants to travel to pick up the items.

Fig 8 – Place an Order with your number preferred of Items

Step 4: The system generates a list of potential restaurants with their match percentage and a list of available and missing items. It also ranks based on the highest percentage within the distance they entered. Select the desired date and pick up time.

Fig 9 – Chosen restaurant with Matching Percentage and Delivery and Pick Up Time

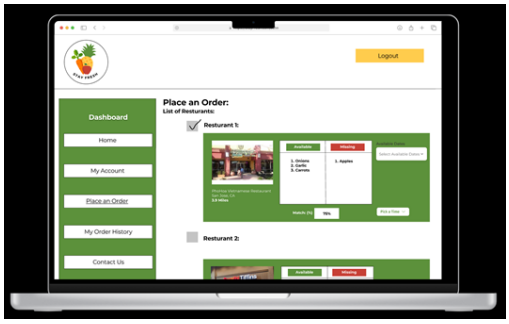
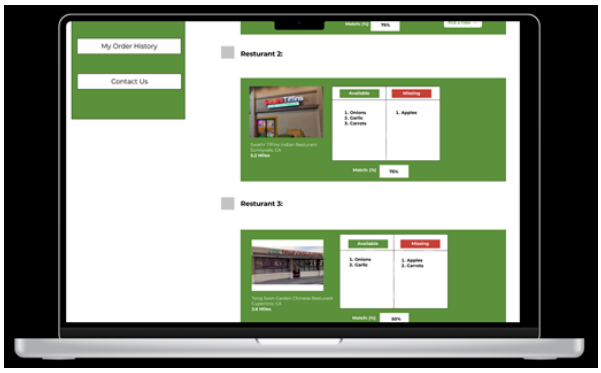
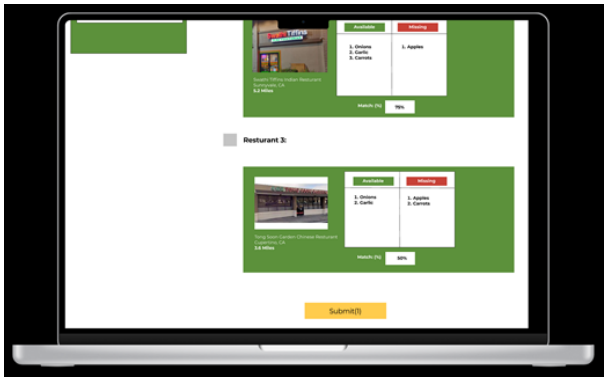


Fig 10 – Chosen restaurant with Matching Percentage and Delivery and Pick Up Time



Step 5: Click the submit button to place the order

Fig 11 – Chosen restaurant with Matching Percentage and Delivery and Pick Up Time



Step 6: Once the order is placed, a confirmation page with the selected pick up time and date appears.

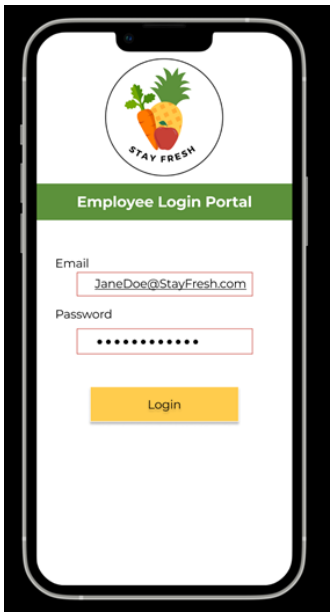
Fig 12 – Confirmation Order Placed

Interface 2: Employee Scan Items Interface (App)

Process 3: Check Out Items from Inventory (for Employees)

Step 1: Login to the Employee Portal

Fig 13 – Employee Login Portal



Step 2: Click on either Check In or Check Out (in this case Check Out Used Case)

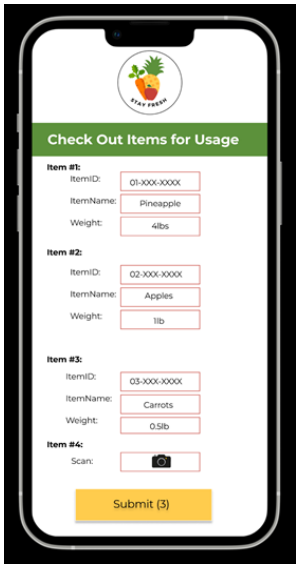
Fig 14 – Check In/ Check Out Interface

Step 3: Scan the barcode to get the information needed

Fig 15- Scan the Item Barcode to Get the information

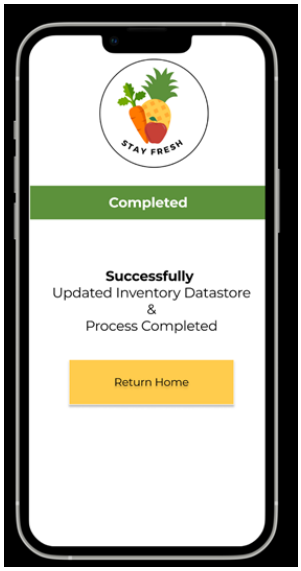
Step 4: Once finished scanning click submit to update the inventories.

Fig 16- Sample Output from Scanning Barcodes



Step 5: Confirmation of Updated Datastore

Fig 17- Confirmation Page of Update

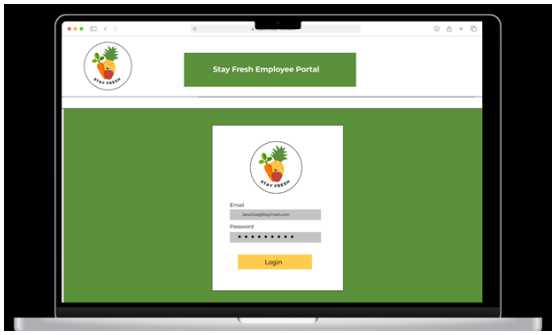


Interface 3: Employee Inventory/Expiration Management Internal System (Website)

Process 4: Employee Website Management Interface

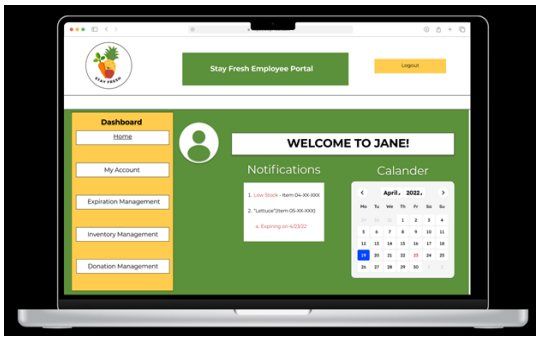
Step 1: Employee/Manager logins to the Employee Portal for Inventory/Expiration/Donation Management.

Fig 18- Login Portal for Employee



Step 2: Employee will have a list of Notifications in List form and calendar form

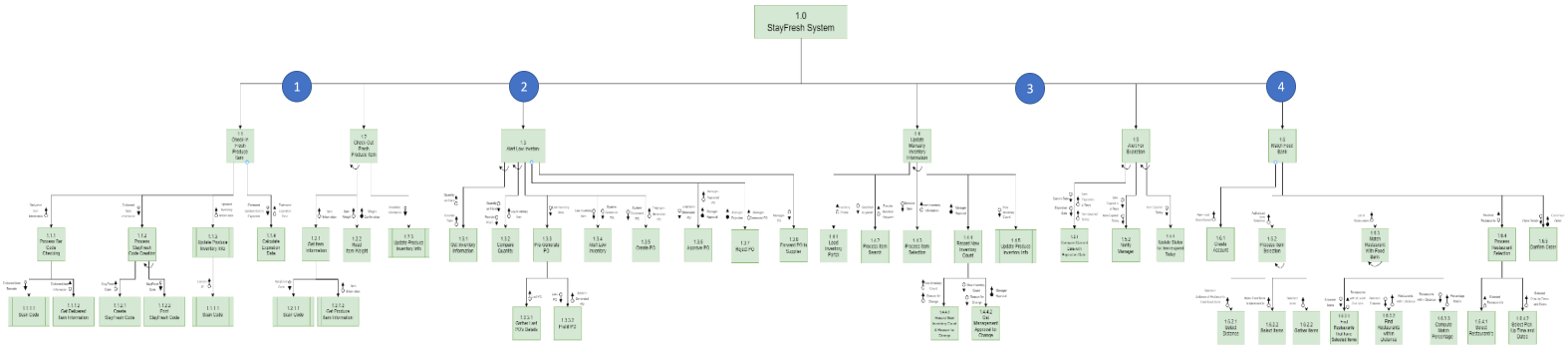
Fig 19- Employee Dashboard

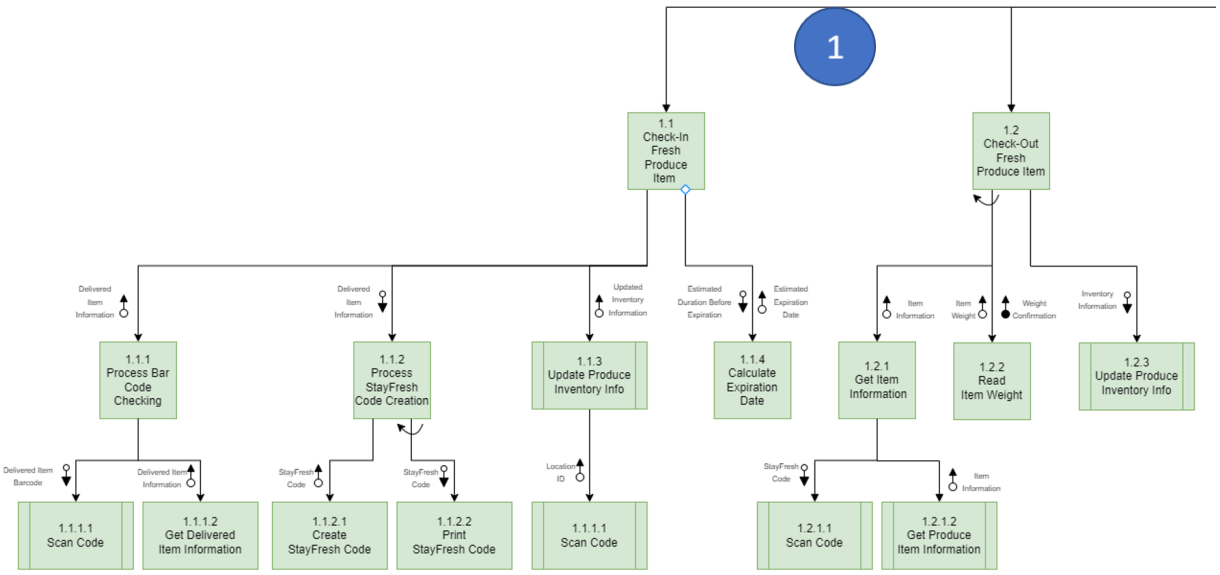


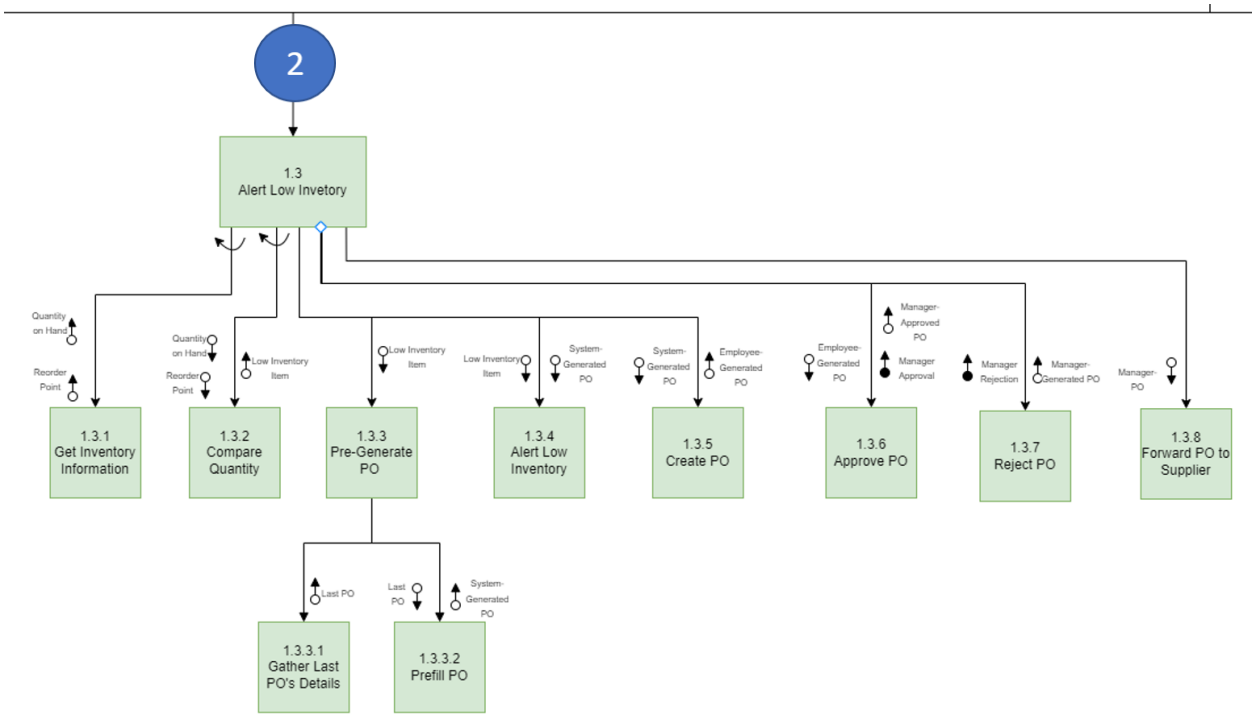
Step 3: Employee can click on the date in red to see what items are expiring either soon or are expired.

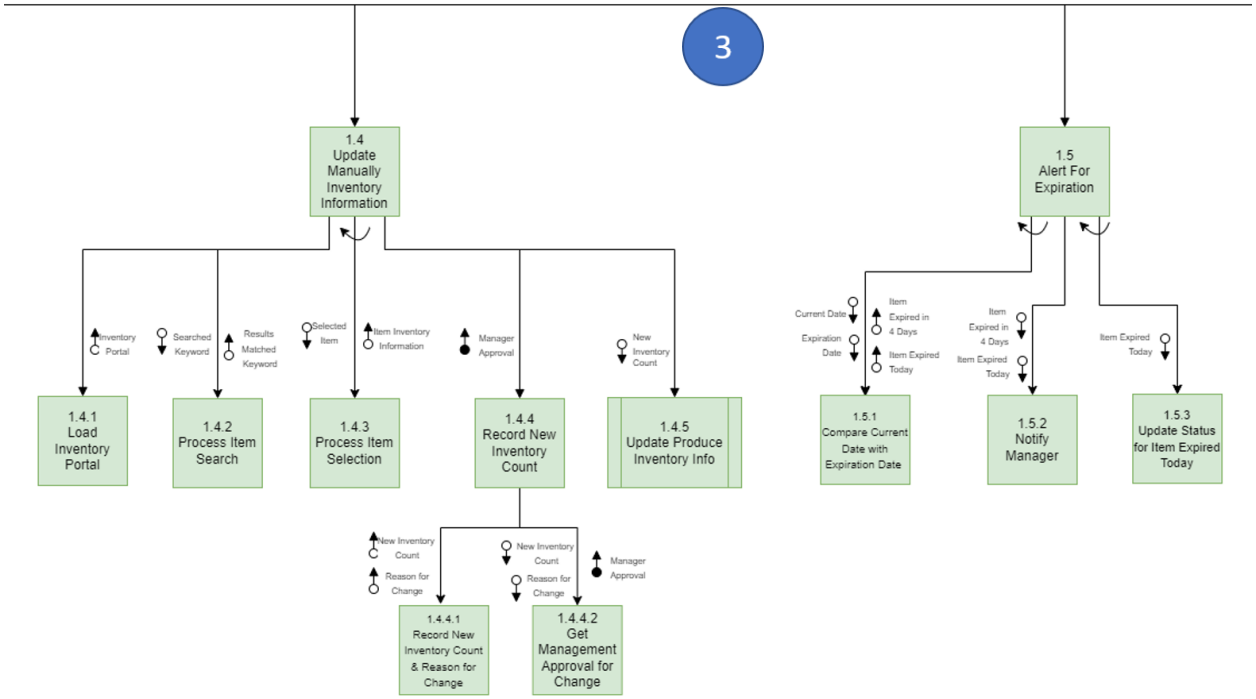
Fig 19- Employee Dashboard Date Expired Item Output

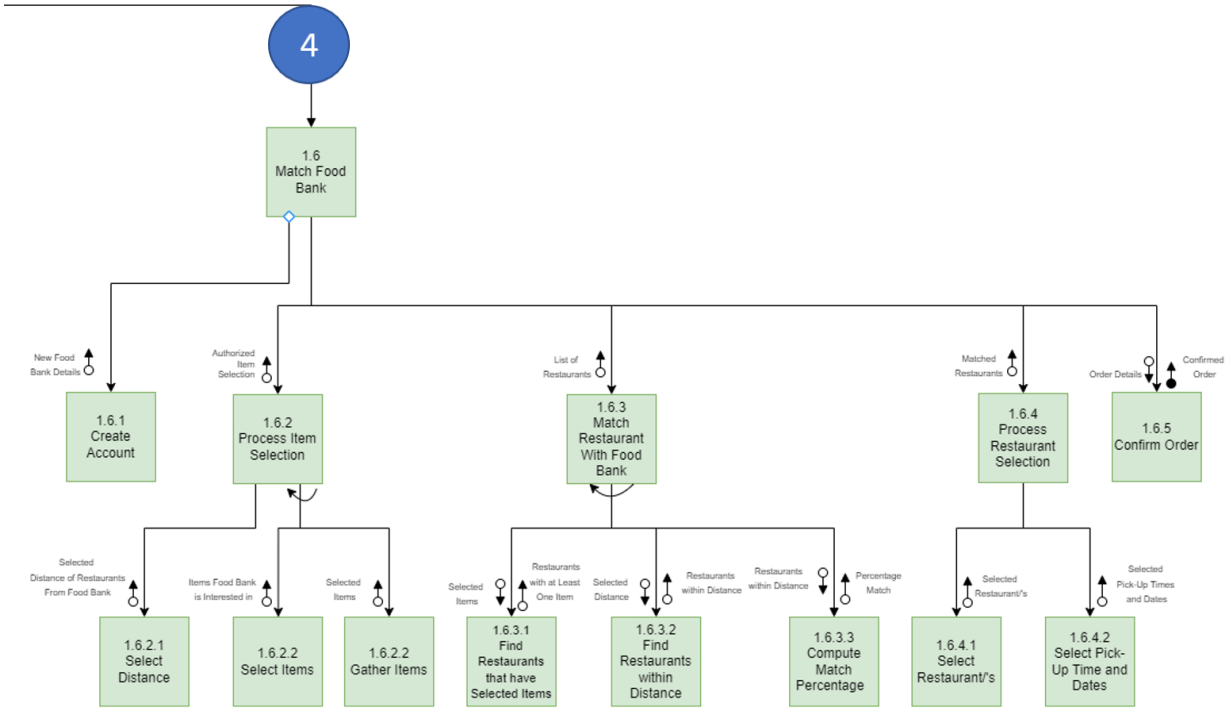
5. Program Structure Chart






2
1.3
Alert Low Invetory
Quantity on Hand
Reorder Point
Quantity on Hand
Reorder Point
Low Inventory Item
Low Inventory Item
Low Inventory Item
System-Generated PO
System-Generated PO
Employee-Generated PO
Employee-Generated PO
Manager-Approved PO
Manager Approval
Manager Rejection
Manager-Generated PO
Manager-PO
1.3.1
Get Inventory Information
1.3.2
Compare Quantity
1.3.3
Pre-Generate PO
1.3.4
Alert Low Inventory
1.3.5
Create PO
1.3.6
Approve PO
1.3.7
Reject PO
1.3.8
Forward PO to Supplier
Last PO
Last PO
System-Generated PO
1.3.3.1
Gather Last PO's Details
1.3.3.2
Prefill PO


3
1.4
Update Manually Inventory Information
Inventory Portal
Searched Keyword
Results Matched Keyword
Selected Item
Item Inventory Information
Manager Approval
New Inventory Count
1.4.1
Load Inventory Portal
1.4.2
Process Item Search
1.4.3
Process Item Selection
1.4.4
Record New Inventory Count
1.4.5
Update Produce Inventory Info
New Inventory Count
Reason for Change
New Inventory Count
Reason for Change
Manager Approval
1.4.4.1
Record New Inventory Count & Reason for Change
1.4.4.2
Get Management Approval for Change
1.5
Alert For Expiration
Current Date
Expiration Date
Item Expired in 4 Days
Item Expired Today
Item Expired in 4 Days
Item Expired Today
Item Expired Today
1.5.1
Compare Current Date with Expiration Date
1.5.2
Notify Manager
1.5.3
Update Status for Item Expired Today



6. Future Scope

While the first version of the StayFresh system focuses mainly on the management of fresh produce, StayFresh will continue putting efforts into reducing food waste and creating a sustainable food system by supporting, but not limited to the following functionalities in the future.

1. To enhance the fresh produce management system by using food sensors to precisely monitor fresh produce states. Also by including machine learning algorithms to predict more accurately the expiry date for each inventory product, based on past data, food usage rules and storage condition of each restaurant, and to create recommended dish lists according to the fresh produce inventory state.

2. To expand the expiration management system to include all food products.

3. To upgrade the food donation system allowing restaurants to select donatable food directly from their data stores, instead of adding each donation item manually.

7. Conclusion

One of StayFresh's core missions is to improve sustainability in the restaurant industry. Our solution, StayFresh Produce Management System, allows restaurants to effectively manage their fresh produce inventory lifecycle and create a channel to redistribute excess food. StayFresh's solution helps restaurants control food cost, therefore increasing profit margin, while also practicing food sustainability. We expect our solution to reduce a restaurant's food waste by 50%, and allow it to donate 100% of surplus food.

The company will continue to improve this product. In the near future, the clients can expect to see new cutting-edge features added to create an even more seamless and effective solution.