

# **Software Requirements Specifications Inventory Optimization Suite**

**Adam Leung, James Barlow, Eric Buss, Lee Zeitz, Rafat Mahmud,  
Sean Burt, Sean Hoessmann, Torrey Randolph**

**Carrot Inc.**  
**January 30, 2017**

## **Contents**

<b>Revision History</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Purpose	5
1.2 Project Scope	5
1.3 Glossary of Terms	5
1.4 Overview	6
<b>2 Overall Description</b>	<b>6</b>
2.1 Product Perspective	6
2.2 Product Features	6
2.3 User Classes and Characteristics	7
2.3.1 System Administrator	7
2.3.2 Manager	7
2.3.3 Corporate	7
2.3.4 Receiver	7
2.3.5 Stocker	7
2.4 Operating Environment	7
2.5 Use Cases	8
2.5.1 Add or Subtract Stock	8
2.5.2 Change Stock Status	12
2.5.3 Add and Modify Users	22
2.5.4 Add New Shipments to Inventory	25
2.5.5 View Statistics	29
2.6 Design and Implementation Constraints	31
2.7 Assumptions and Dependencies	31
2.7.1 Receiver Shipment Documentation	31
2.7.2 Wholesome Eats Inc. Employee Administration Service	31
<b>3 System Features</b>	<b>32</b>
3.1 Restocking	32
3.2 Receiving	33
3.3 System Notification Flags	34
3.4 Downtime	34
3.5 Statistics Tracking	35
3.6 Reports	36

3.7 User Privileges	37
3.8 System Administrative User Registration/Removal	38
3.9 User Authentication	39
4 External Interface Requirements	40
4.1 User Interfaces	40
4.1.1 System View	40
4.1.2 Inventory View	40
4.1.3 Reciever View	40
4.1.4 Report View	41
4.2 Hardware Interfaces	41
4.3 Software Interfaces	41
4.3.1 Point of Sales System (POS)	41
4.3.2 Wholesome Eats Database	41
4.3.3 iOS	41
4.3.4 Windows 7 Compatible Web Browser	41
4.4 Communications Interfaces	41
4.5 Data Flow Diagrams	42
4.5.1 Level 0 Diagram	42
4.5.2 Level 1 Diagram	43
4.5.3 Level 2 Diagram - Part 1	44
4.5.4 Level 2 Diagram - Part 2	45
4.6 E/R Diagram	46
5 Other Non-Functional Requirements	47
5.1 Performance Requirements	47
5.2 Safety Requirements	47
5.3 Security Requirements	47
5.4 Software Quality Attributes	47
5.4.1 Usability	47
5.4.2 Availability	48
5.4.3 Correctness	48
5.4.4 Maintainability	48
6 References	49

## Revision History

Name	Date	Reason for Changes	Version
Carrot Inc.	2017-1-20	Drafting	0.1
Carrot Inc.	2017-1-26	Finalizing Content	0.2
Carrot Inc.	2017-1-30	Editing	1.0
Carrot Inc.	2017-2-07	Incorporating Client Input	1.1
Carrot Inc.	2017-2-22	Incorporating Client Input and Adding Diagrams	2.0

# 1 Introduction

## 1.1 Purpose

This document describes the software requirements specifications for the Inventory Optimization Suite (IOPS), version 2.0, as requested by Wholesome Eats Inc. The following pages detail all application features, characteristics, interfaces, and other functional and nonfunctional requirements of the IOPS. This document will be presented to Wholesome Eats Inc. and revised until agreements can be reached regarding the proposed system specifications.

## 1.2 Project Scope

Inventory Optimization Suite is a proprietary application targeted to store employees, managers and corporate officers of Wholesome Eats Inc., a multinational food distribution network. This application is targeted primarily towards mobile devices, which would replace the current physical laborious inventory management process employed at grocery stores owned by Wholesome Eats Inc. Store employees would use IOPS to manage store inventory, i.e., manage resources at shelf, keep accounts of total stock, handle rotten or damaged goods, etc., whereas managers would use IOPS to get an overview of current store inventory, observe inventory trends via statistical reports and visualizations, plan for future stock shipments, etc.

IOPS would allow Wholesome Eats Inc. to better use their staffing resources, increase stocking efficiency, track inventory history accurately, order new shipment with confidence, and optimize manager workflow. Overall, allowing for greater productivity and human resource management, leading to greater profits for Wholesome Eats Inc.

## 1.3 Glossary of Terms

<b>API</b>	Application Programming Interface
<b>Back</b>	The back of the store. Stores large quantities of goods to later be displayed
<b>CLIA</b>	Client Internal Application (i.e. Client App or Internal App)
<b>Front</b>	The store-front. Where customers may purchase inventory
<b>IOPS</b>	Inventory Optimization Suite
<b>JSON</b>	Javascript Object Notation
<b>POS</b>	Point of Sale System
<b>Floor Employee</b>	Stockers and Receivers

## 1.4 Overview

This requirements specification covers five topics: Overall Description, System Features, External Interface Requirements, followed by Non-functional Requirements.

The Overall Description section of the Inventory Optimization Suite includes the product perspective and features, user classes and characteristics, operating environments, design and implementation constraints, assumptions, and dependencies. The System Features section follows the Overall Description section and covers the main functions for the product. The External Interface Requirements section gives a detailed description of the external input and outputs of the system. Other Non-functional Requirements covers extra requirements that affect the developers.

# 2 Overall Description

## 2.1 Product Perspective

The Inventory Optimization Suite (IOPS) is intended to replace Wholesome Eats' current partially-physical inventory tracking system. IOPS will communicate with the currently existing inventory database at Wholesome Eats' to avoid the need for changes to the POS system. The system will be essentially distributed as a mobile friendly web application, targeted to Wholesome Eats' store employees, managers and corporate officials. Store employees will be interacting with the IOPS interface on iPads they are provided with, allowing them to use IOPS' core inventory management functionalities; in addition, managers and corporate will have further access to inventory metrics and reporting functionalities. System Admins will have most superior accessibility rights, enabling them to manage users within the system, in addition to all manager and store employee capabilities.

## 2.2 Product Features

The core product features of the Inventory Optimization Suite (IOPS) are as follows:

- System Administrative User Management
  - The capability for administrative users to add, remove, and modify other users of the system such as stockers, receivers, and managers.
- Stock and Inventory Management
  - The capability for users to add, remove, and alter stock-related and inventory-related information.
- Management and Corporate Reports and Analytics
  - The capability for Store Managers to view storewide data over selected time intervals. Reports include statistics on stock levels, sales figures, expired/rotten goods counts, etc.
- Smart Notification System
  - A system for sending floor employees notifications to inform them of low stock levels and upcoming product expiration.

## **2.3 User Classes and Characteristics**

### **2.3.1 System Administrator**

The IT department situated at the head office is assigned with the highest security clearance and is responsible for local maintenance of the system and modifying all system variables when necessary. System Administrators will have access to all application functionalities that Managers have (see section 2.3.2), and in addition, can add, remove, and modify users within the system and their privileges.

### **2.3.2 Manager**

Managers are responsible for individual storefronts, and have the highest security clearance within a given store. They have read and write access to overall stock counts, discrepancies, and only read access to storewide statistical reports. Managerial users are the most favoured users, as they must interact the most with the system, and expect the most from system functionality.

### **2.3.3 Corporate**

Corporate users need to evaluate a store's inventory or review the system's history. Corporate users have access to all application features that managers have access to, but are limited to read-only mode, and therefore have no permissions to edit store inventory or statistics.

### **2.3.4 Receiver**

Receivers bring shipped inventory into the store, input the amount of stock received, and report any discrepancies regarding the invoice. Receivers can modify the quantity of a product within the system, input the expiry date by entering the invoice number, and flag alerts for missing goods. Receivers will only be shown the minimal content required to complete their tasks. They are crucial to the success of the system and, therefore, important to satisfy.

### **2.3.5 Stocker**

Stockers move stock from the back of the store to the shelves. They also remove expired stock from the front to be thrown out. Stockers can input the amount of stock they move from the back to the front, and can log how much stock they throw out. Stockers will only have access to minimal current store inventory figures, and statistics that they require for aiding them with their jobs. Along with receivers, are crucial to the success of the system so they are important to satisfy.

## **2.4 Operating Environment**

IOPS will be deployed in the form of a web application, accessible to Wholesome Eats' employees via a web browser. The web application must be compatible with current popular web browsers and Windows 7, as used by managers on local workstations, as well as iPad tablets as used by stockers. IOPS will co-exist with currently-existing Wholesome Eats' inventory and store information database. For further information on browser specifications, please see section 4.

## 2.5 Use Cases

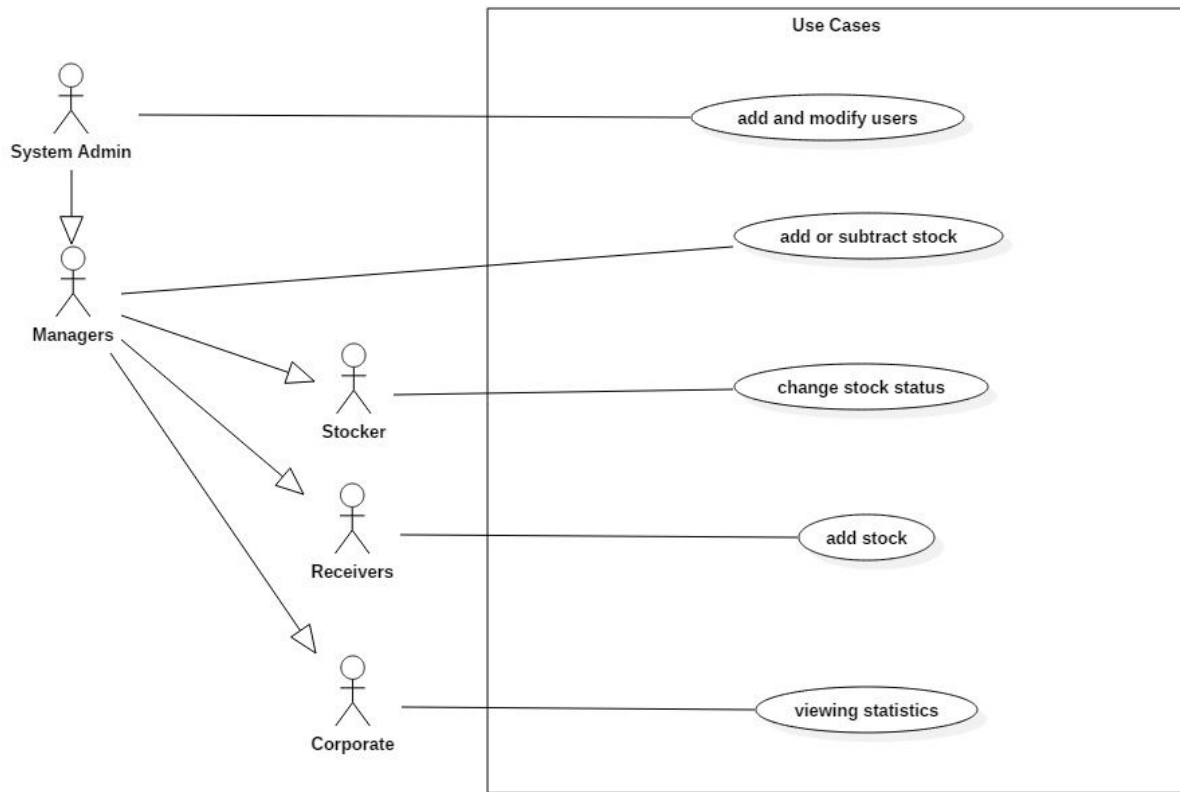


Figure 1: Use-Case Diagram for IOPS

Detailed descriptions of all actors present within Figure 2.5.1 can be found in section 2.3. System Administrators inherit directly or indirectly from Stocker, Manager, Receiver, and Corporate. Corporate users will only have a read-only view of manager analytics. All users must have completed a “login” use case in order to be authenticated and recognized as a user. They will then have access to the system as their specific user type.

A further decomposition of potential use cases is as follows:

### 2.5.1 Add or Subtract Stock

#### 2.5.1.1 Description

A manager discovers an error that was input during receivers adding new stock. Since receivers do not have permissions to modify stock, the manager logs into the IOPS manager account and fixes the receiver’s error.

#### 2.5.1.2 Actors

Store Manager, IOPS

#### 2.5.1.3 Pre-Conditions

An error has occurred in stock numbers and a manager is keeping track of their store’s condition. The user must have managerial privileges.



#### **2.5.1.4 Main Flow**

1. The manager begins their shift.
2. The manager is informed by one of their employees that receiving stock was entered incorrectly.
3. The manager accesses the IOPS website.
4. The manager navigates to a screen for managing the store's stock.
5. The manager searches for the product they were informed of.
6. The manager selects that product.
7. **<New Form>** The manager updates the product stock to reflect the correct amount.
8. **<Form Assessment>** The manager submits the update.
9. The system prompts the user to confirm the change.
10. The stock is now correct.
11. The use case ends.

#### **2.5.1.5 Post-Conditions**

The product's stock number now reflects the Manager's update.

#### **2.5.1.6 Alternative Flows**

- A. At **<Form Assessment>** if the user enters an invalid stock amount:
1. The system displays a message informing them that the amount they entered is invalid.
  2. Return to **<New Form>**

### 2.5.1.7 Main Flow Sequence Diagram

The following sequence diagram represents the main flow actions of the user and responses of the system.

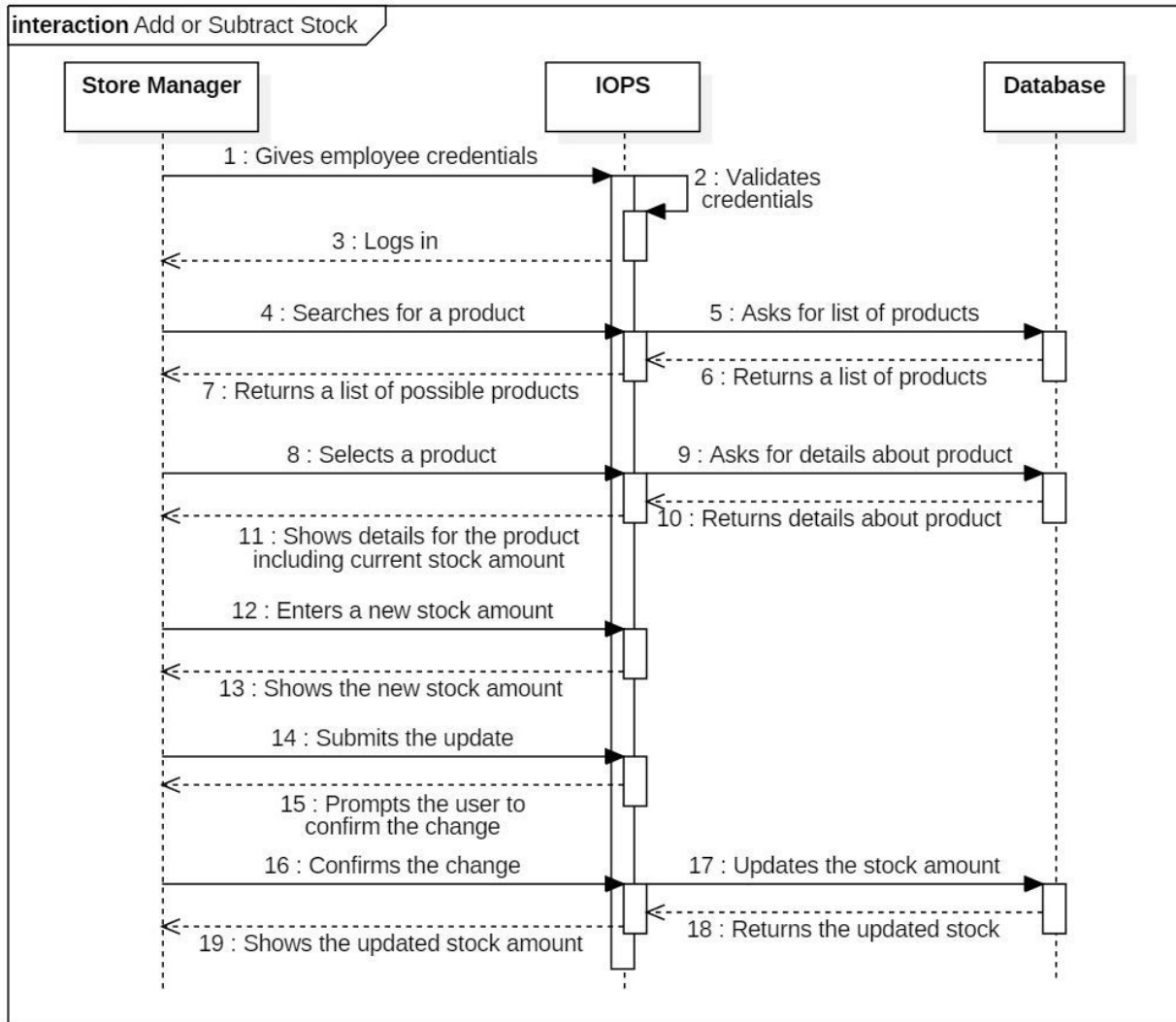


Figure 2: Sequence Diagram for Add or Subtract Stock

### 2.5.1.8 Main Flow UI Mockups

The add and subtract stock use case involves the use of three menus; login menu, main menu, and inventory menu. The first menu the user sees is the login menu, which is shown in the figure below. At this menu, the user adds their credentials and are then logged into the main menu.

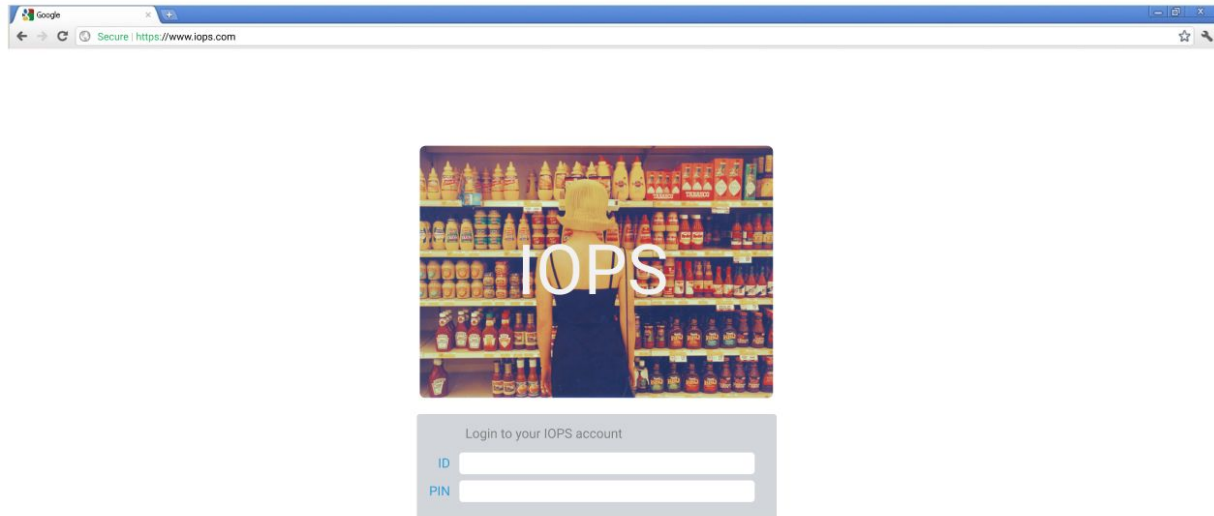


Figure 3: PC Login Menu

At the Main menu, the user is able to choose options such as inventory, reports, and system if the user has system administration privileges. For this use case, inventory is selected.

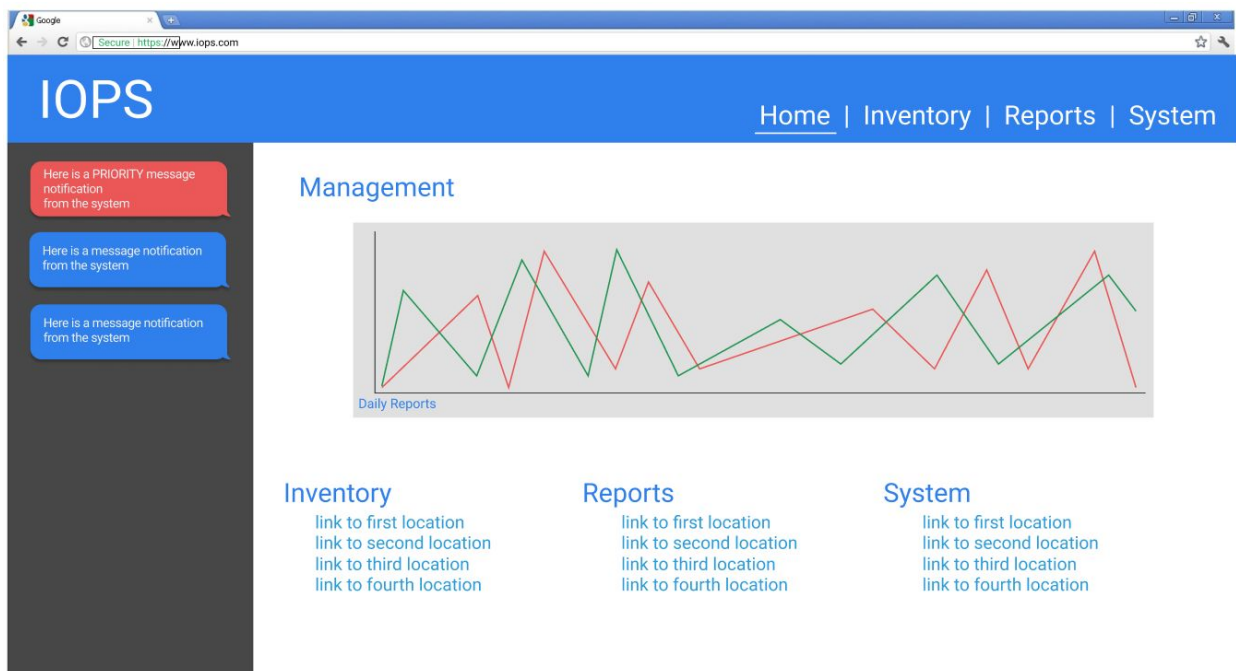


Figure 4: PC Main Menu

When the user enters the inventory menu, they are met with an inventory menu that

allows the user to search through the stores inventory and adjust stock levels. For this use case, the user adjusts the selected stock to fix a discrepancy and submits the change. Once this is done, a overlay similar to Figure 11 in section 2.5.2 Change Stock Status. This overlay shows the changes to be placed.

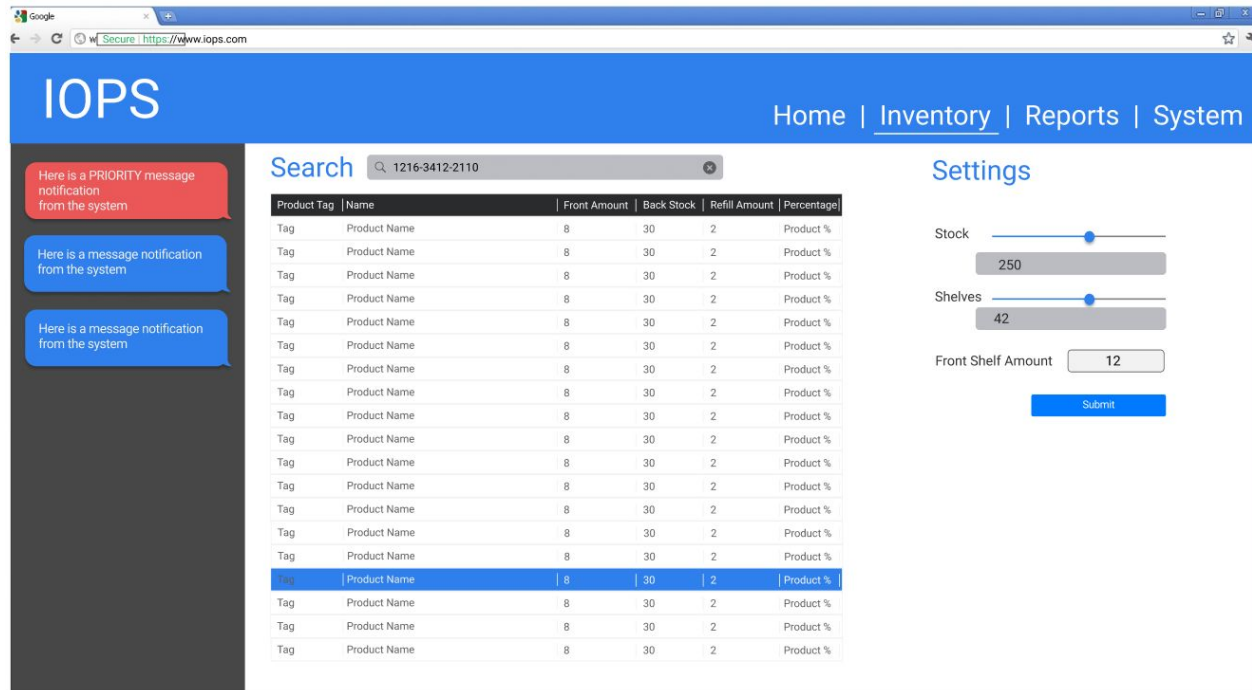


Figure 5: PC Inventory Menu

## 2.5.2 Change Stock Status

### 2.5.2.1 Description

A store employee is notified that some stock is low and wishes to change its status.

### 2.5.2.2 Actors

Stocker, IOPS

### 2.5.2.3 Pre-Conditions

This use case assumes that the employee has an account with stocker permissions, and assumes that the employee was already notified that the stock is low.

#### 2.5.2.4 Main Flow

1. The employee opens the Wholesome Eats web application.
2. The employee enters completed authentication using their employee ID and 4-digit PIN.
3. The system opens the stocker menu.
4. The employee uses the search to find the item.
5. **<Status Change>** The employee moves the desired number of items from the back to the front and submits the change.
6. The system prompts the user to confirm the change.
7. The user logs out.
8. **<End>** The use case ends.

#### 2.5.2.5 Post-Conditions

The product's front and back inventory numbers now reflect the employee's change.

#### 2.5.2.6 Alternative Flows

A. At **<Status Change>**, the user can instead flag the item if there is not the same number of items in the back as in the app:

1. The employee sees that the number in the system is different than what is in the back.
2. The employee selects the flagging option.
3. The employee updates the status of the product with the remaining product.
4. Return to **<End>**.

### 2.5.2.7 Main Flow Sequence Diagram

The following sequence diagram represents the main flow actions of the user and responses of the system.

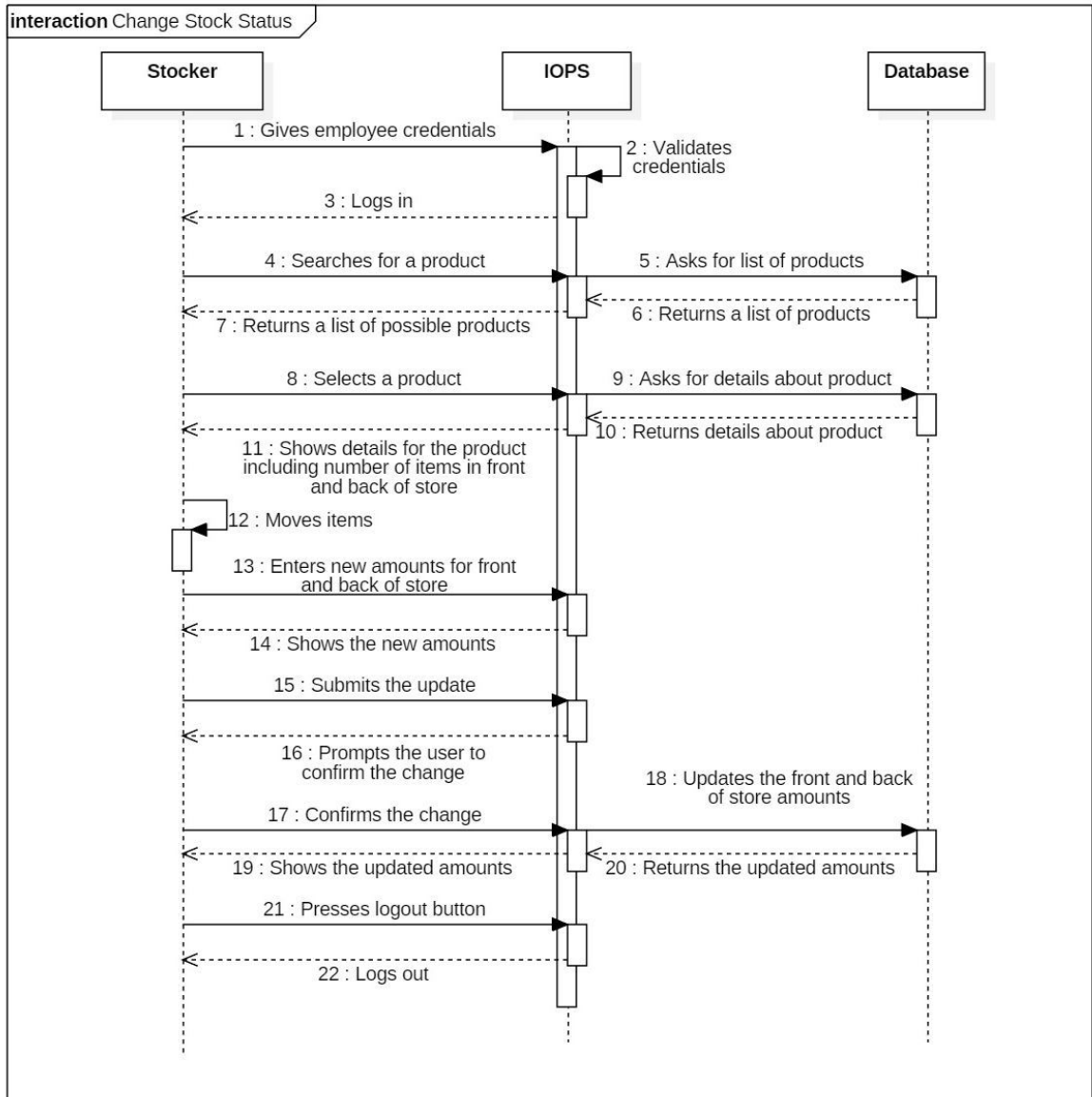
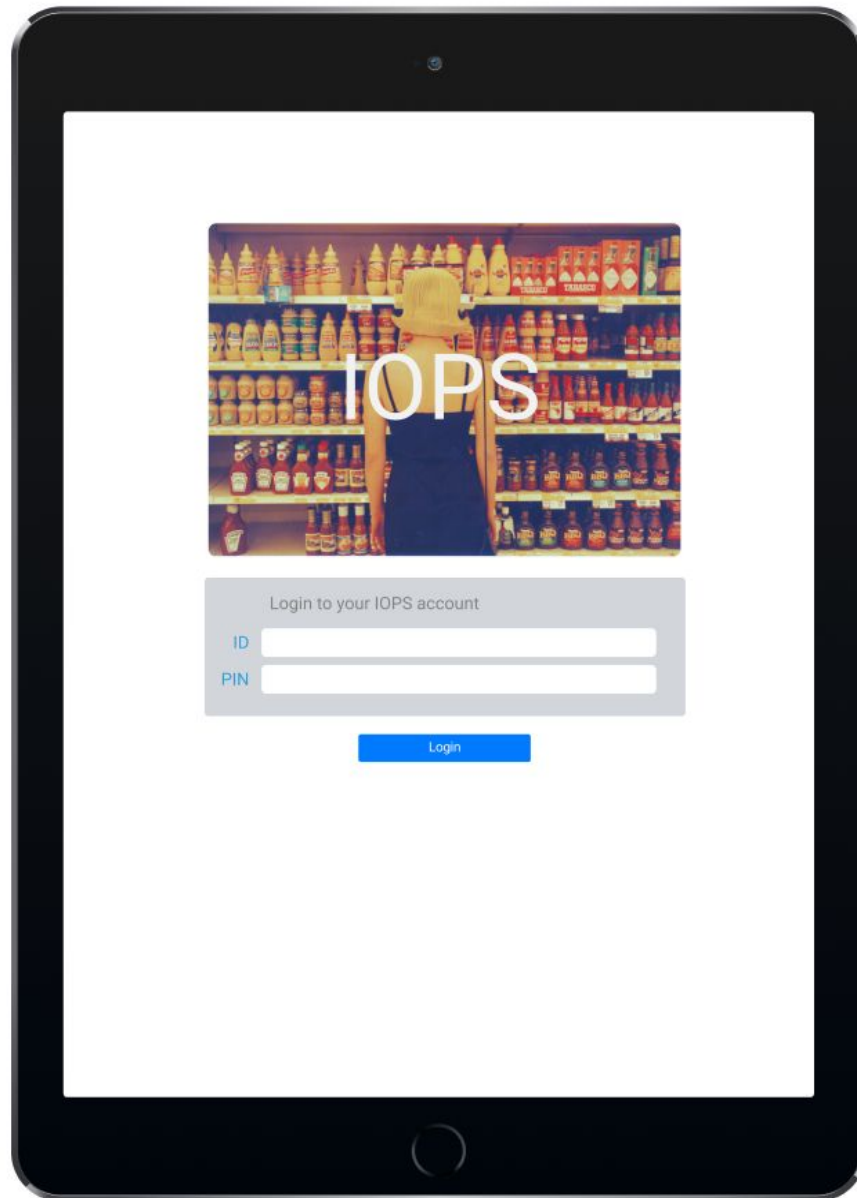


Figure 6: Sequence Diagram for Change Stock Status

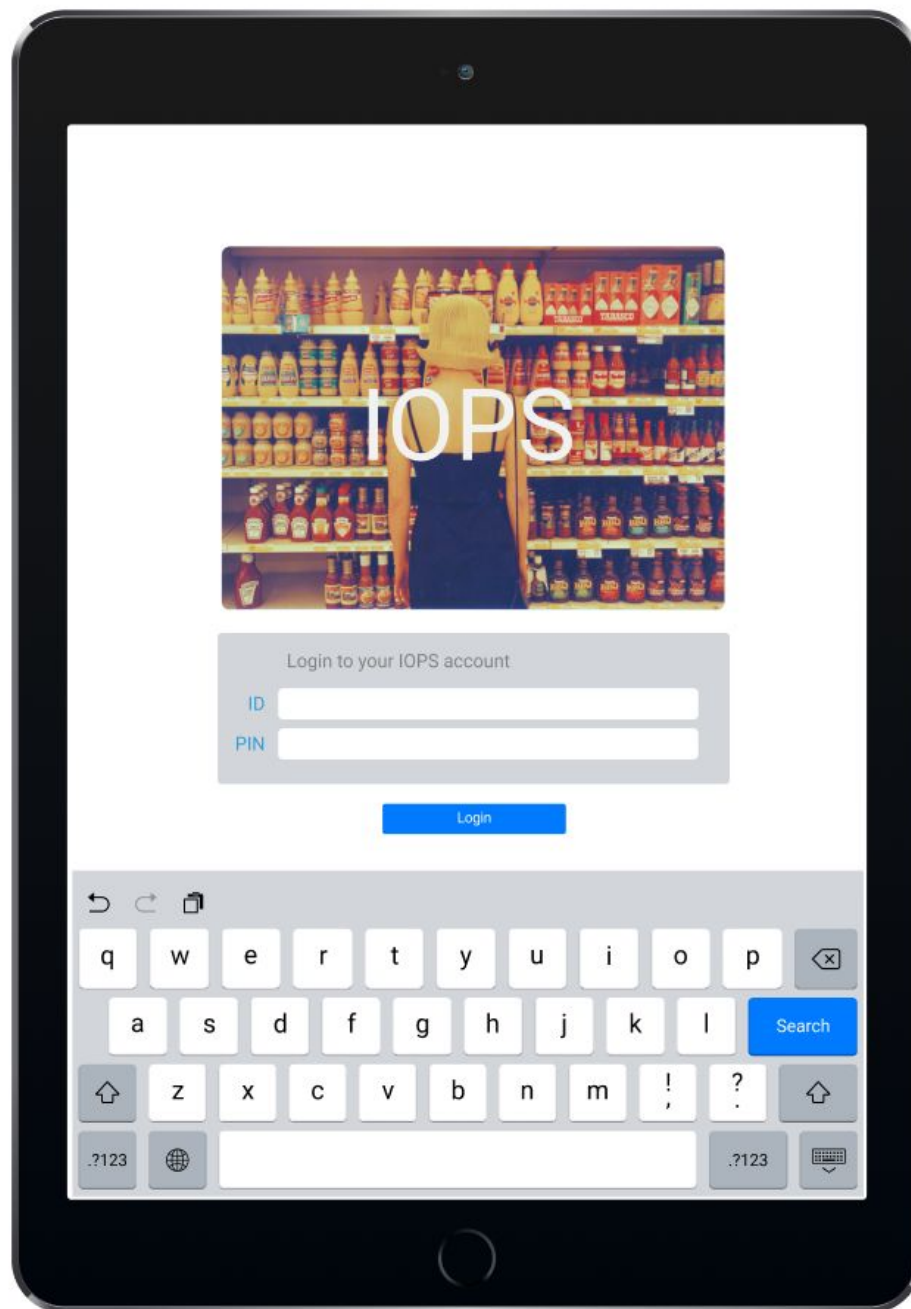
### 2.5.2.8 Main Flow UI Mockups

The UI involved includes the login and stocker menus. The first menu the user sees is the login menu. Within this menu, users are able to enter their employee ID and PIN to access the system.



*Figure 7: iPad Login*

The user inputs their credentials using the touch screen keyboard that appears once a text bar is pressed.



*Figure 8: iPad Login with keyboard*



The next menu that the user sees is the Stocker menu. This menu allows the user to search for a product and change the amount at the front, as well as flag a product for a discrepancy. The user presses the search bar and types in the product tag. The system outputs the product and products with similar tags and highlights the searched product.

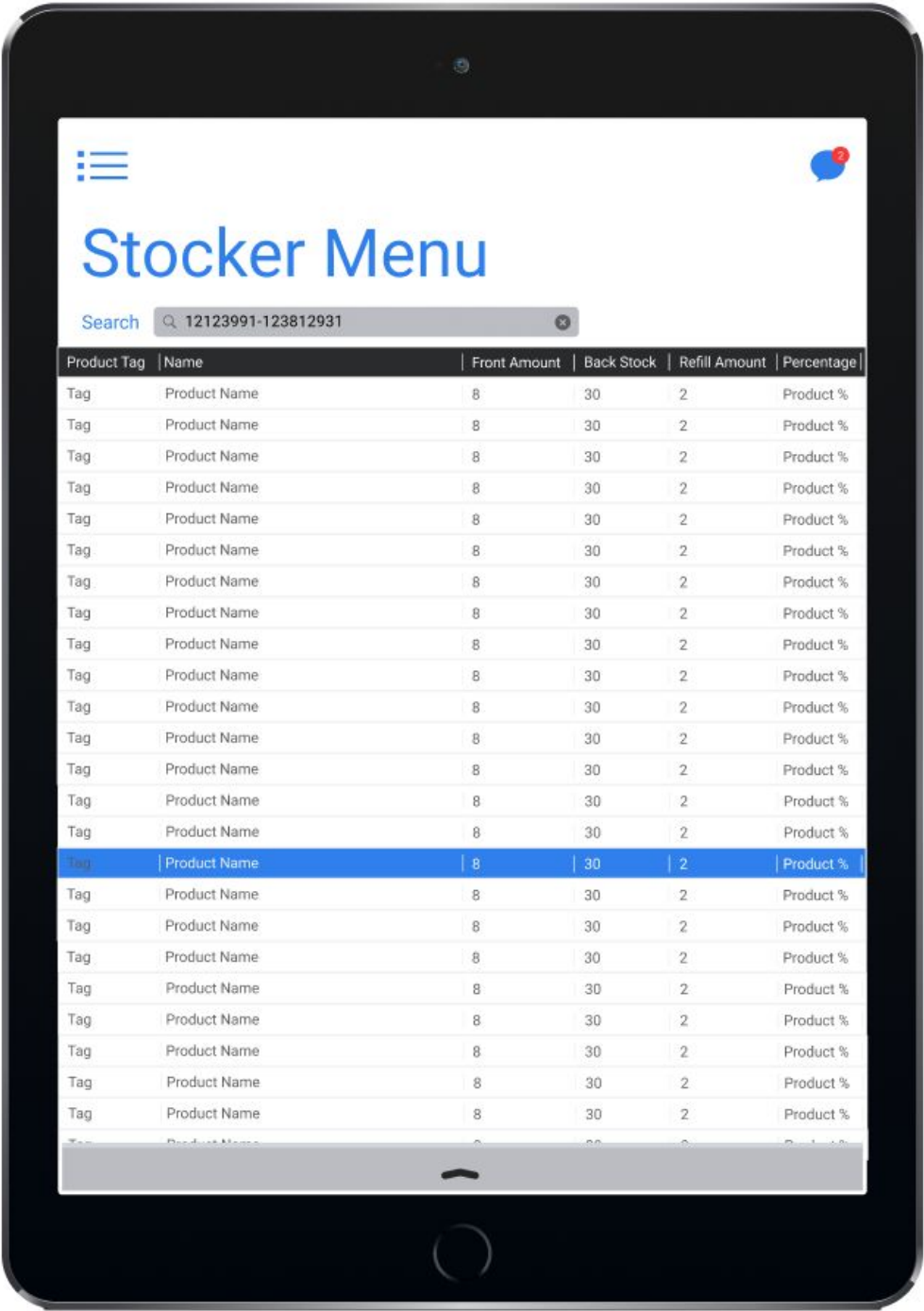




Figure 9: Stocker menu




# Stocker Menu

Search

Product Tag	Name	Front Amount	Back Stock	Refill Amount	Percentage
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %
Tag	Product Name	8	30	2	Product %

Front  
42 / 50



Stock  
250

Flag

Submit

17

The user then presses the submit button at the lower right corner of the menu. This action brings up a confirmation overlay that shows the front stock and remaining stock.

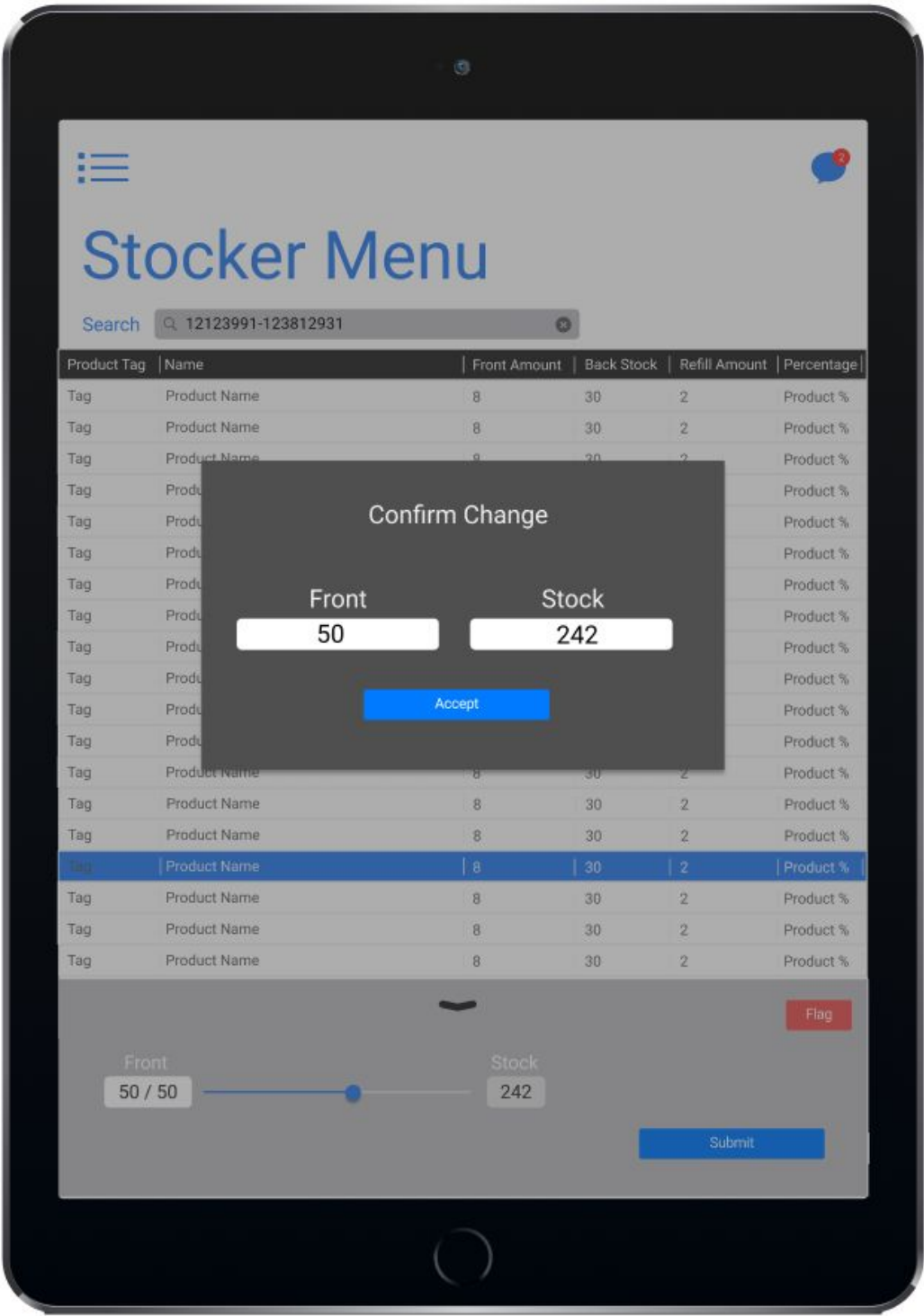


Figure 11: Stocker menu with confirmation overlay

If the front and back stock is correct, the user presses accept to confirm the changes.

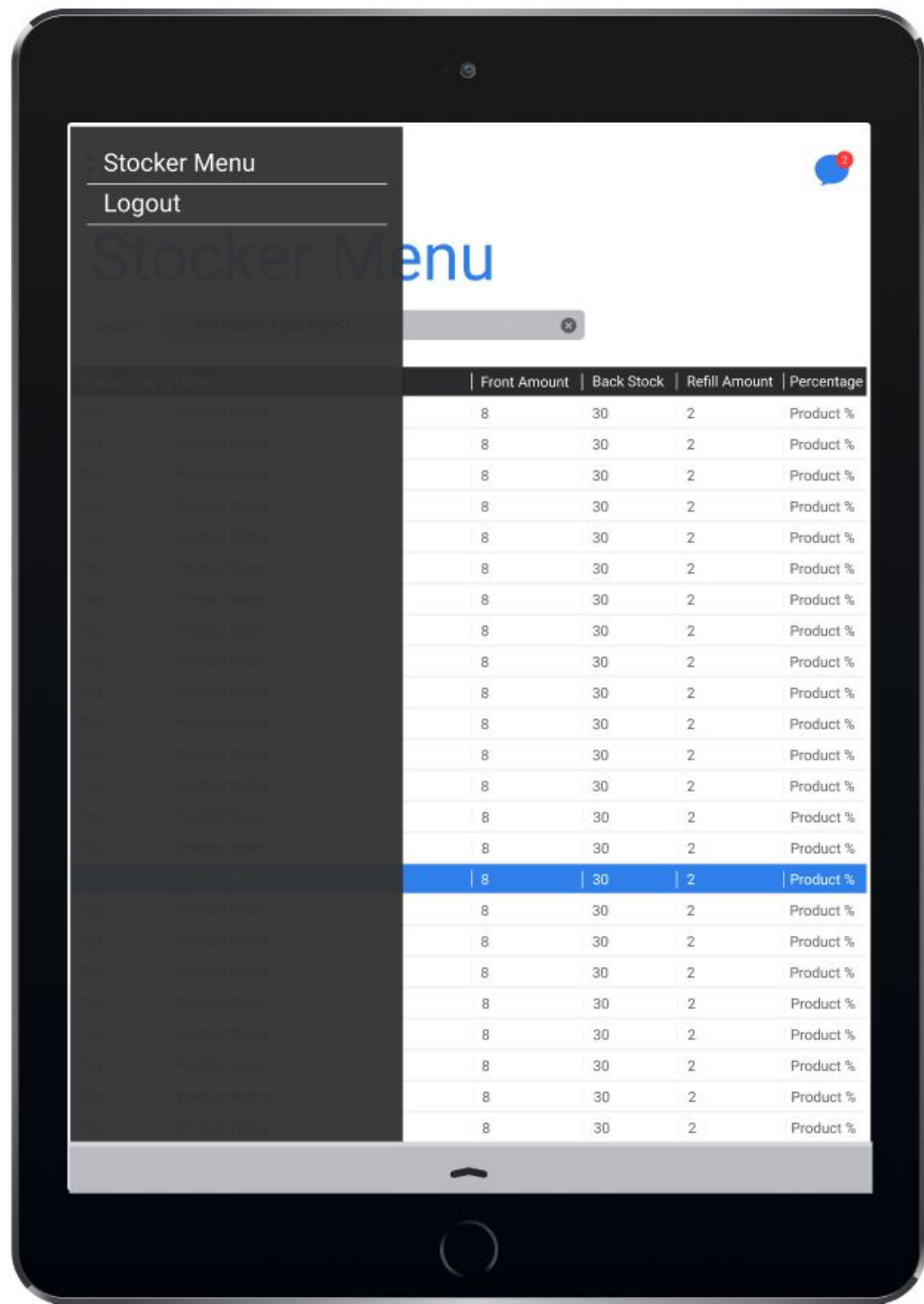


Figure 12: Stocker menu with left toggle overlay

Once these actions are complete, the user logs out using the left toggle, which is opened

by pressing the toggle button located at the top left of the screen.

Alternatively, a user might receive a notification through the right toggle overlay seen below. Within this overlay, the user is able to see priority notifications in red, and regular notifications in blue. Upon receiving this notification, a user can continue to make changes to the stock status.

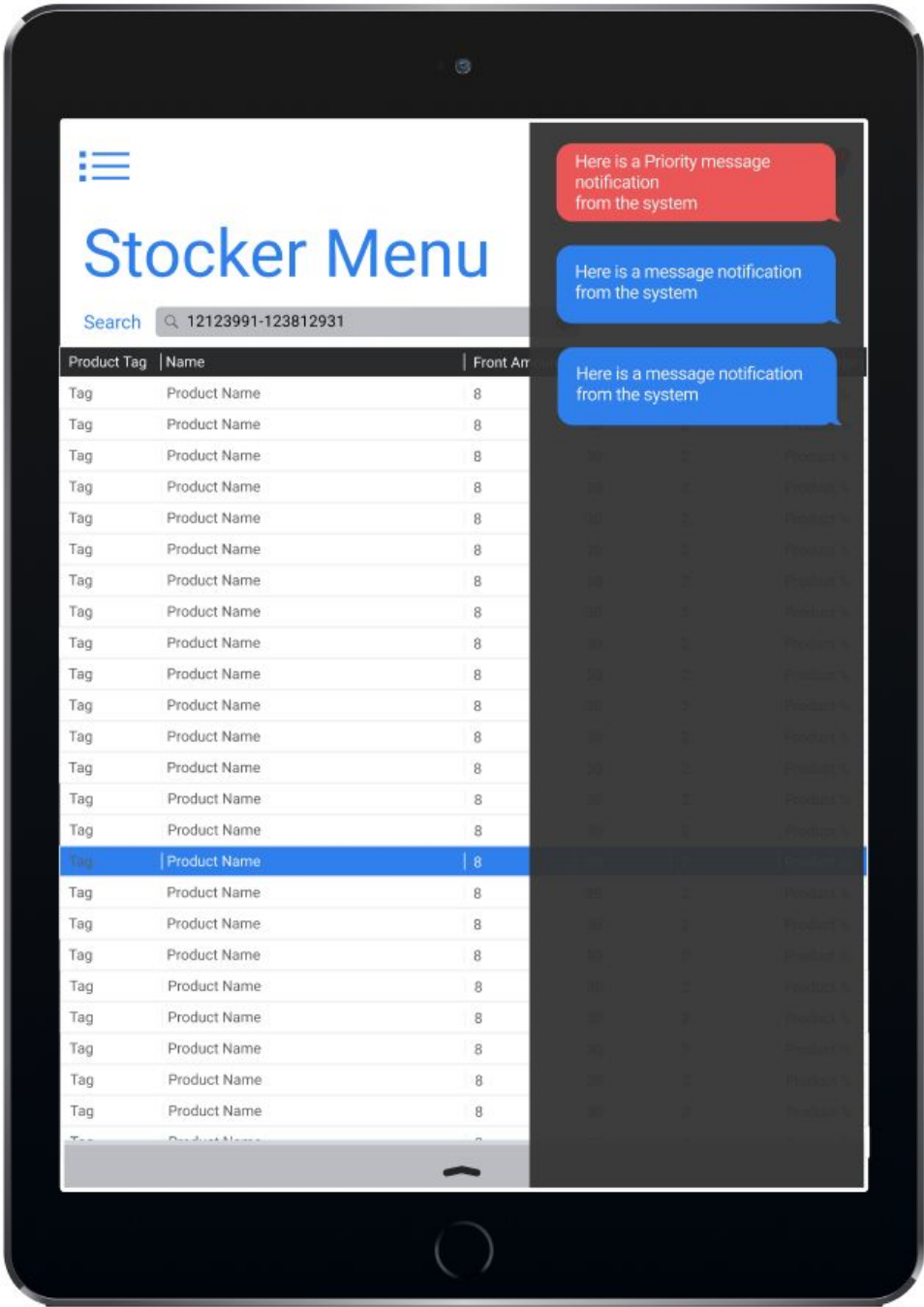


Figure 13: Stocker menu with right notification overlay

## 2.5.3 Add and Modify Users

### 2.5.3.1 Description

This use case describes how users may be added or modified in the IOPS.

### 2.5.3.2 Actors

System Admin, IOPS, Store Manager

### 2.5.3.3 Pre-Conditions

The System Admin must have been added to the system with System Admin privileges

### 2.5.3.4 Main Flow

1. The use case begins when the Store Manager contacts a System Admin.
2. The System Admin accesses the IOPS website.
3. **<Main Menu>** The system displays various System Admin controls.
4. **<Add New User>** The System Admin moves to the Add New User section.
5. The system displays a form to be filled with user information. See Figure 14 in Appendix B.
6. **<New User Form>** The System Admin fills in the form with the new user's full name, date of birth, user type, hiring date, and employee ID, and selects Confirm.
7. **<New User Form Assessment>** The system assesses if the form is complete.
8. The system generates a temporary PIN and stores the user information in the system database.
9. The system removes the form and displays the temporary PIN.
10. The System Admin contacts the Store Manager, and informs them of the generated PIN.
11. The System Admin selects Return to Main Menu
12. **<Returned to Main Menu>** The system displays various System Admin controls.
13. The use case ends.

### 2.5.3.5 Post-Conditions

If Main Flow or Alternative Flow A or B, the new user may now authenticate into the IOPS.

If Alternative Flow D or E, the user has been modified.

### 2.5.3.6 Alternative Flows

A. At **<New User Form Assessment>**, if the form is incomplete or invalid, the system will:

1. Display a message stating "The Form is Incomplete".
2. Display the form to be filled with user information, indicating what is incomplete or invalid.
3. Return to **<New User Form>**.

B. At **<Website Access>**, if the System Admin is not already authenticated, then:

1. The System Admin enters their credentials.

2. The IOPS authenticates the credentials.
3. Return to <**Main Menu**>.

C. At <**New User Form**>, if the System Admin selects Cancel instead of Confirm, then:

1. The system ignores the input from the System Admin.
2. Return to <**Returned to Main Menu**>.

D. At <**Add New User**>, if the user the use case is about is being modified, then:

1. The System Admin selects Modify Existing User.
2. The system requests which user the System Admin would like to modify.
3. The System Admin selects the user.
4. The system displays a form with the selected user's attributes.
5. <**Modification Form**> The System Admin modifies the existing fields and selects Confirm.
6. <**Modification Form Assessment**> The system assesses if the form is complete.
7. Return to <**Returned to Main Menu**>.

E. At <**Modification Form Assessment**> in Alternative Flow D, if the form is incomplete or invalid, the system will:

1. Display a message stating "The Form is Incomplete".
2. Display the form with user attributes, indicating what is incomplete or invalid.
3. Return to <**Modification Form**>.

F. At <**Modification Form**> in Alternative Flow D, if the System Admin selects Cancel instead of Confirm, then:

1. The system ignores the input from the System Admin.
2. Return to <**Returned to Main Menu**>.

### 2.5.2.7 Main Flow Sequence Diagram

The following sequence diagram represents the main flow actions of the user and responses of the system.

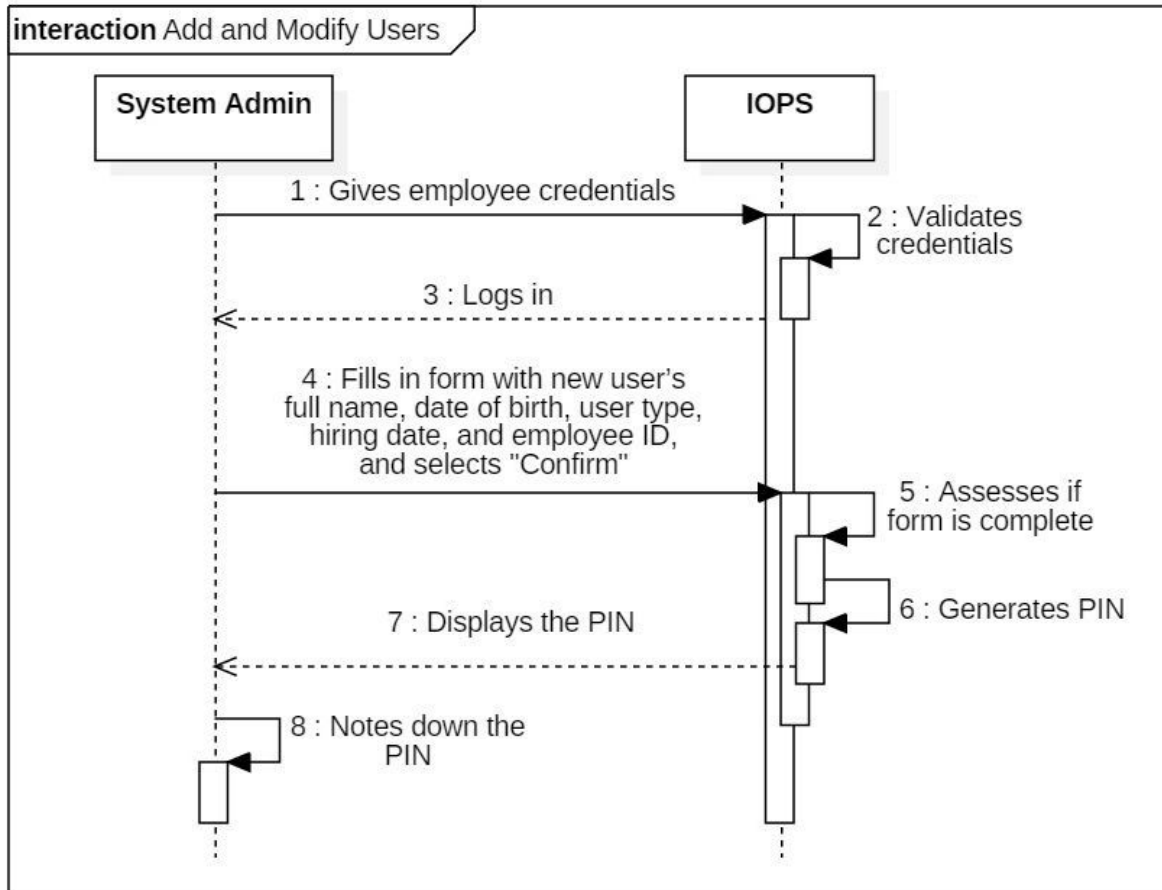


Figure 14: Sequence Diagram for Add and Modify Users



### 2.5.3.8 Main Flow UI Mockups

The UI involved with this use case is the systems page of the web application. Here a System Admin can select users and modify their settings using the left half of the screen. The other half of the screen allows a System Admin to add a new user to IOPS.

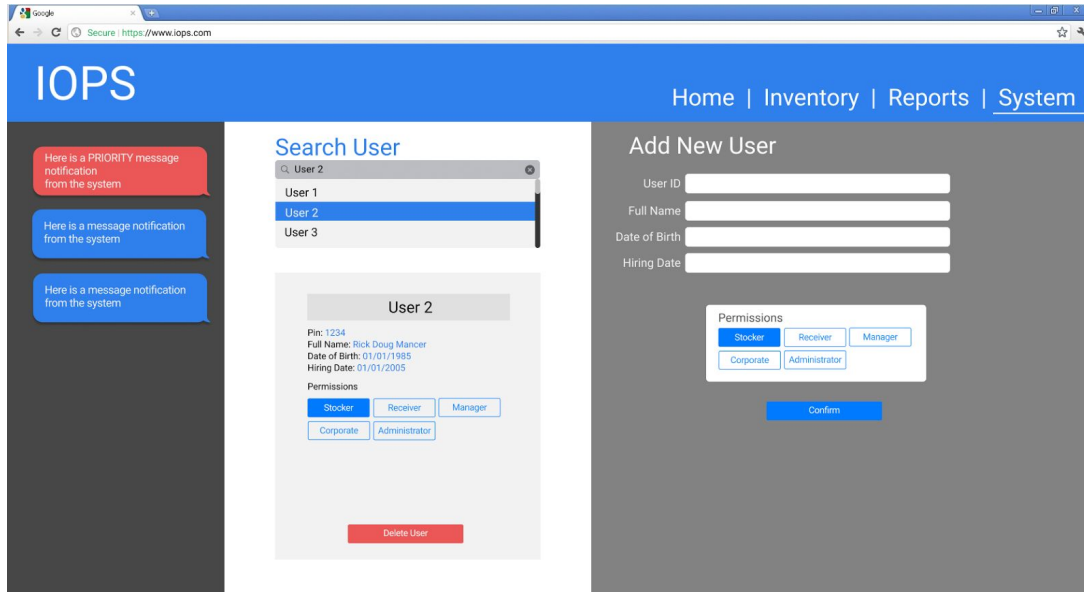


Figure 15: PC System menu

## 2.5.4 Add New Shipments to Inventory

### 2.5.4.1 Description

This use case describes how a receiver employee adds new shipment stock to the database through the IOPS.

### 2.5.4.2 Actors

Store receivers, IOPS, Database.

### 2.5.4.3 Pre-Conditions

It is assumed that the store receivers are employees of the Wholesome Eats store and has an account with store receiver permissions.

### 2.5.4.4 Main Flow

1. This use case begins when a new shipment of stock arrives at the store.
2. The receiver checks the new stock for damages before signing off that the shipment contains the correct items.
3. The receiver opens the IOPS application.
4. **<Login Credentials>** The receiver starts their authentication through an employee ID and 4-digit PIN.
5. **<Receiver Menu>** The system brings the user to the receiver menu of the site.
6. **<Add Stock>** Using the invoice provided by the shipping company, the receiver enters the unique shipment identifier.

7. **<Input Check>** The system looks up the corresponding product data based on the unique identifier and shows it to the user.
8. The system prompts the receiver to finalize the new stock input.
9. The database is updated.

#### 2.5.4.5 Post-Conditions

New stock has been successfully added to the database through the IOPS.

#### 2.5.4.6 Alternative Flows

A. At **<Login Credentials>**, if the user's PIN is invalid, the system will:

1. Display a message stating "Invalid PIN".
2. Display the form to be filled with the user's login and PIN again.
3. Return to **<Receiver Menu>** once credentials are valid.

B. At **<Input Check>**, If the system does not recognise a unique identifier, the system will:

1. Display a message stating "Invalid invoice Identifier" and display the identifier.
2. Return to **<Add Stock>**.

#### 2.5.4.7 Main Flow Sequence Diagram

The following sequence diagram represents the main flow actions of the user and responses of the system.

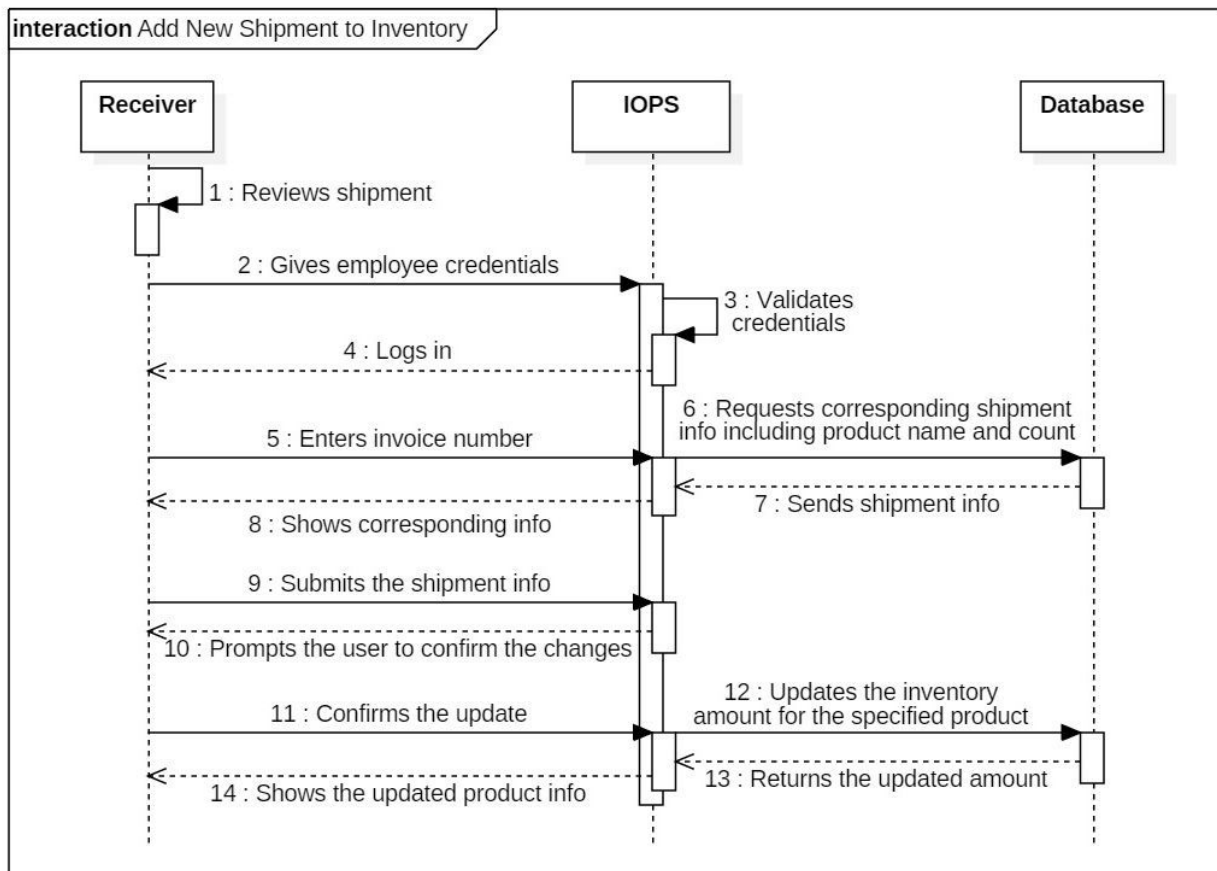


Figure 16: Sequence Diagram for Add New Shipment to Inventory

#### 2.5.4.8 Main Flow UI Mockups

The UI involved with this use case is the receiver menu. To add a shipment a receiver enters the unique shipment identifier in the invoice field. This will show the receiver the new stock. Once the shipment has been entered the receiver presses the confirm button. This will add the shipment.

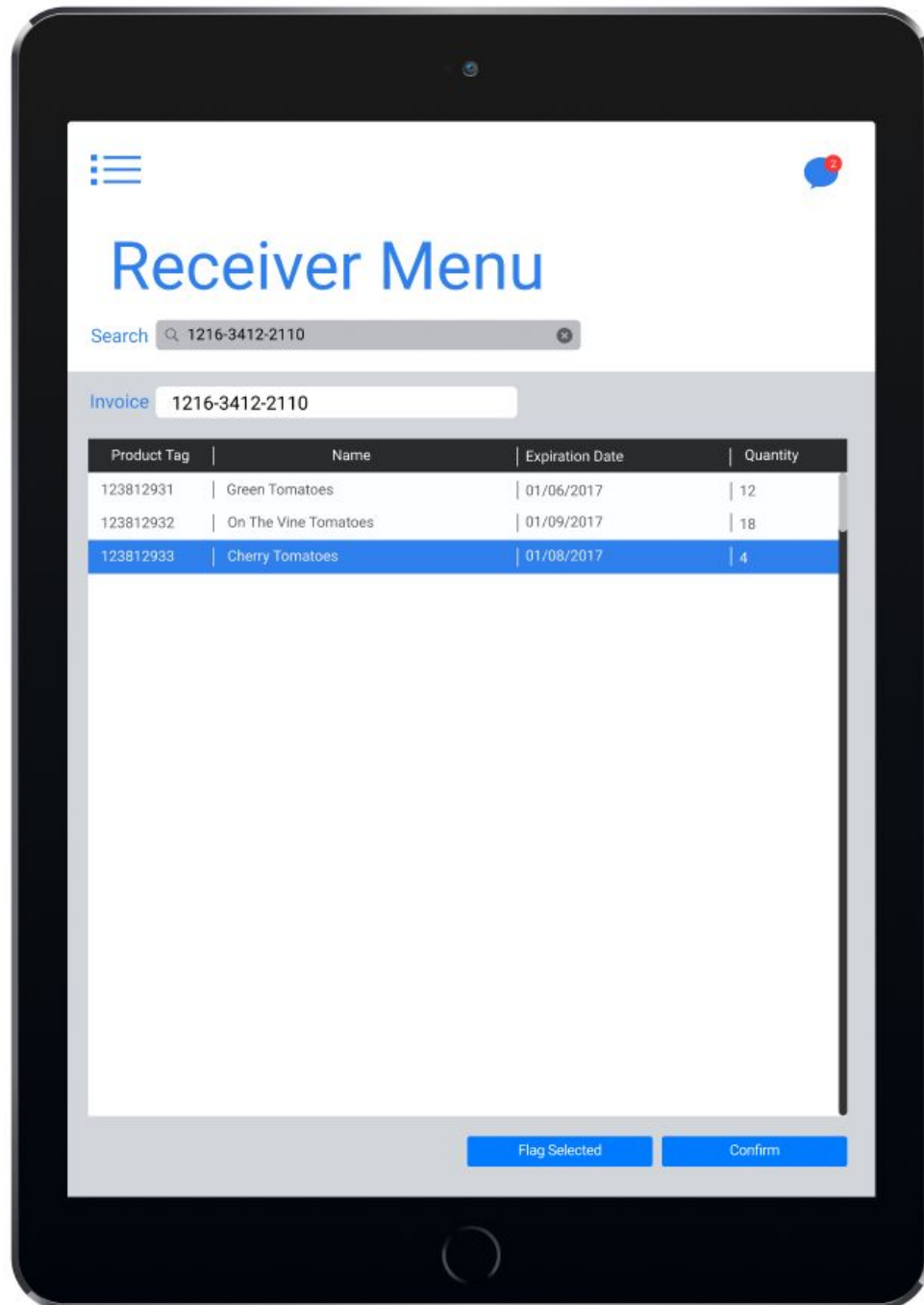


Figure 17: iPad receiver menu

If the received shipment should have flags related to it the receiver can press the flag selected button to bring up the following menu. This will allow the receiver to indicate any issues with the incoming stock.



Figure 18: iPad receiver menu

## 2.5.5 View Statistics

### 2.5.5.1 Description

The store management department wants to see how the store compared to the previous year. The night manager opens the IOPS application and views the day's analytics.

### 2.5.5.2 Actors

Store Manager, Corporate User, IOPS, Database

### 2.5.5.3 Pre-Conditions

Stock and sales information must have been collected over the course of the day. The user must have managerial privileges.

### 2.5.5.4 Main Flow

1. The manager begins their shift.
2. The manager accesses the IOPS website.
3. <New User Form>The night manager navigates to reports.
4. <New User Form>The night manager views the statistics for the day's stocks and sales.
5. The use case ends.

### 2.5.5.5 Post-Conditions

The night manager is now aware of the day's statistics and informed of any major discrepancies.

### 2.5.5.6 Alternative Flows

- A. At <New User Form> if there is no data for the selected time:
1. The system displays a message informing there is no data for the selected time.
  2. Return to <New User Form>
- B. This process can be performed by corporate users rather than by managers.

### 2.5.5.7 Main Flow Sequence Diagram

The following sequence diagram represents the main flow actions of the user and responses of the system.

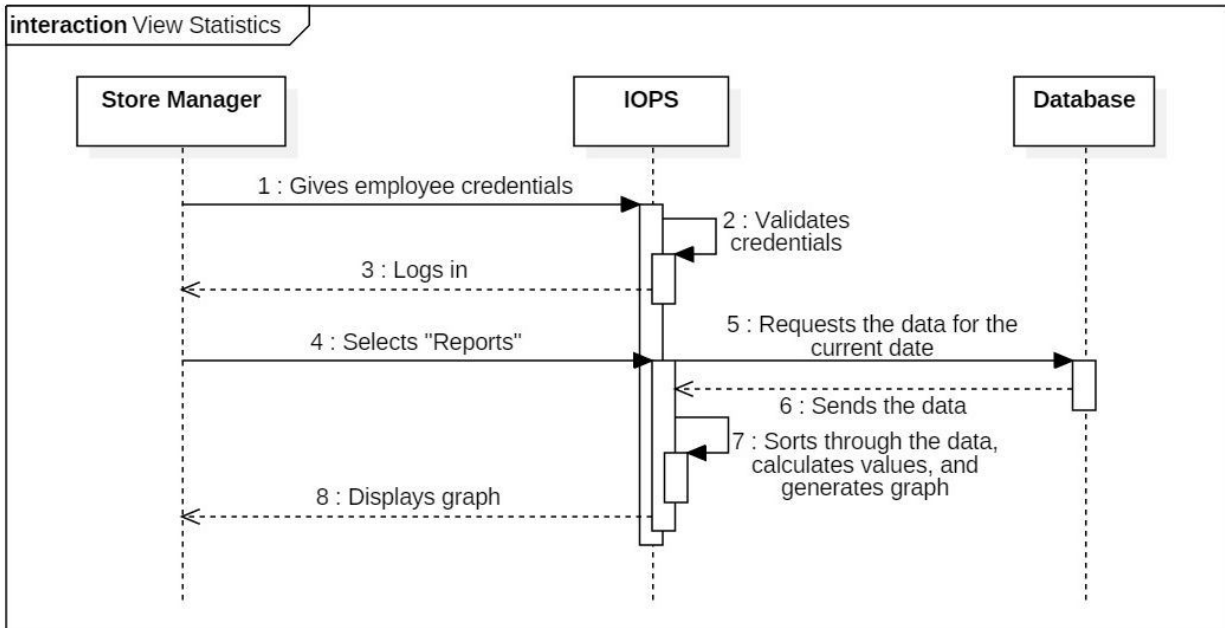


Figure 19: Sequence Diagram for View Statistics

### 2.5.5.8 Main Flow UI Mockups

The UI involved with this use case is the reports menu of the web application. Here, the user can add reports to be compared. Adding reports is done by pressing the “+” button.

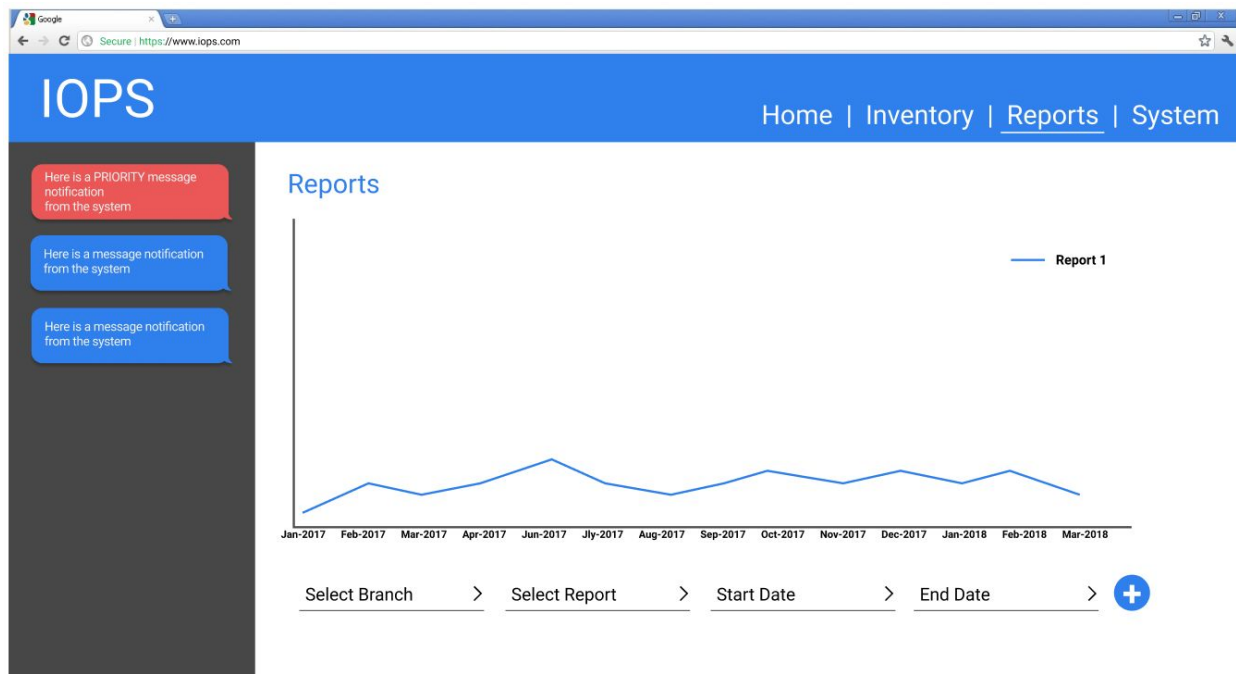


Figure 20: iPad reports menu

This allows for multiple reports to be compared, which allows for different reports such as sales and losses. For this use case, only one report was generated for the day.

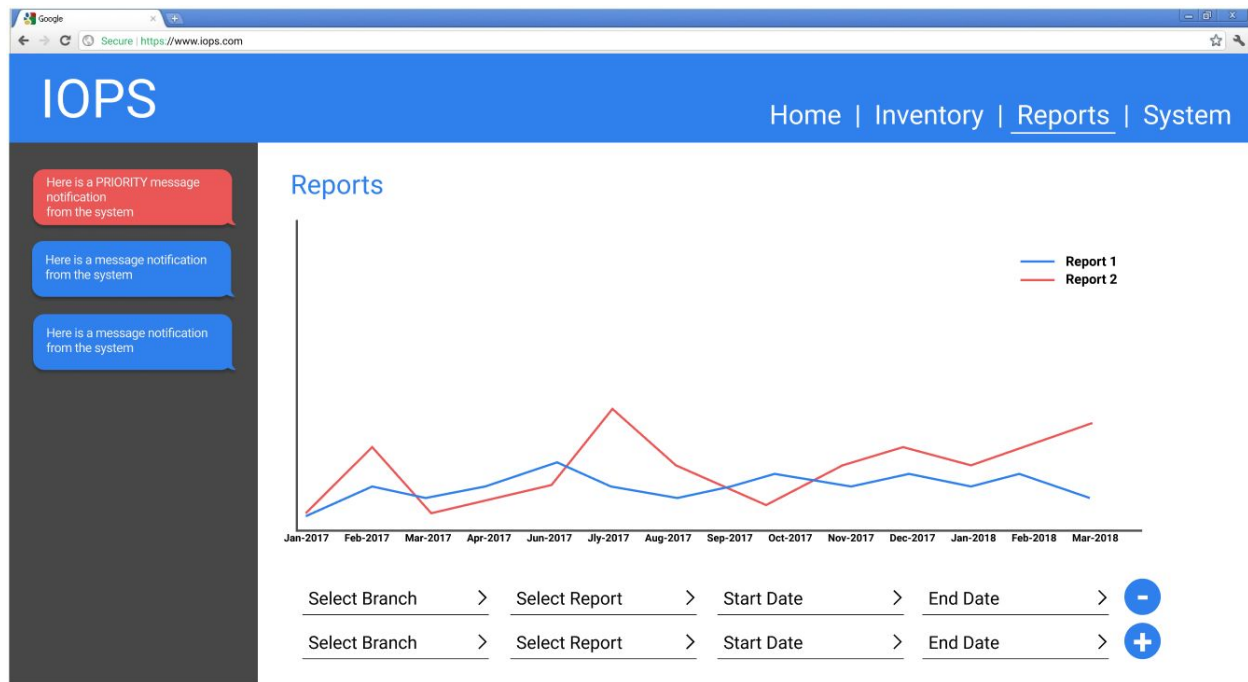


Figure 21: iPad reports menu with second report

## 2.6 Design and Implementation Constraints

The design and implementation constraints given by Wholesome Eats Inc. are as followed:

- The IOPS must use or extend Wholesome Eats Inc.'s current database
- The IOPS must be compatible with the iOS for store floor devices, and Windows 7 for manager workstations.

## 2.7 Assumptions and Dependencies

### 2.7.1 Receiver Shipment Documentation

It is assumed that the following shipment information is available in a Wholesome Eats database and can be retrieved by entering the corresponding unique identifier:

- Shipment Invoice Number
- Shipment Content Name/ID
- Shipment Content Weight/Amount
- Shipment Content Expiration dates

It is also assumed that some process currently exists within Wholesome Eats Inc. grocery stores to handle the expiration dates of goods part of the produce section.

### 2.7.2 Wholesome Eats Inc. Employee Administration Service

Since Wholesome Eats Inc. is a global phenomenon in terms of food distribution, it is assumed that some form of employee account administration or email service exists, which will facilitate

initial creation of store employee accounts. In addition, all IOPS user account credentials will only be modifiable by the System Admins of Wholesome Eats Inc.

## 3 System Features

### 3.1 Restocking

#### 3.1.1 Description and Priority

Provide notifications indicating items that need restocking, based on calculations done on stored inventory and expiration data. This feature is designed to decrease the time required to keep shelves stocked, which is one of the main objectives of the client, making it a high priority.

#### 3.1.2 Stimulus / Response Sequences

A notification will display on the employee's screen, alerting them of a product that needs to be restocked, either because it is running low, or because it has expired and needs to be disposed of. The employee will be able to dismiss the alert by filling out a form specifying what they have replaced and why. The system will then update the storefront and back stock counts in the database to represent the inputted information.

#### 3.1.3 Functional Requirements

REQ	Description
1.01	The system must contain an interface that will display restocking notifications.
1.02	<p>The system must contain an interface where users can enter the product ID and quantity of what they have removed from the storefront, the product ID and quantity of what they have moved from the stockroom to the storefront, and the reason(s) for doing so.</p> <p><b>Rationale:</b> Stores need to maintain individual counts on inventory in the back, and front of store, and overall inventory count. This allows for creating precise reporting features and aiding the smart notification system.</p> <p><b>Test Scenario:</b> Load IOPS with stocker security permissions. Edit quantity moved from store back to storefront. Commit changes and verify currently updated values are accurate.</p>
1.03	<p>The system must be able to calculate when products need to be taken off the shelf based on their expiration dates. The system must notify users of this deadline.</p> <p><b>Rationale:</b> Stocker's should receive notifications when store front stock is running low, thus appropriate stock count state needs to be stored to implement notifications.</p> <p><b>Test Scenario:</b> Reduce storefront stock below to configured threshold. Expect to see notifications of low stock, and item info on stocker account.</p>
1.04	The system must be able to calculate when products need to be moved from the



	<p>stockroom to the storefront, based on stock counts and sales. The system must notify users when items need to be restocked.</p> <p><b>Rationale:</b> Stocker's should receive notifications when store back stock is running low, thus appropriate stock count state needs to be stored to implement notifications.</p> <p><b>Test Scenario:</b> Reduce store back stock below to configured threshold. Expect to see notifications of low stock, and item info on stocker account.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 3.2 Receiving

### 3.2.1 Description and Priority

Provide an interface for adding incoming shipment data to the store inventory database. This is a high priority feature, as it is required to keep an accurate inventory count.

### 3.2.2 Stimulus / Response Sequences

The user will have a screen where they can manually enter the unique identifier from the shipping invoice. The system will retrieve the shipment information corresponding to the invoice identifier from the database and update the store inventory to reflect the new product.

### 3.2.3 Functional Requirements

REQ	Description
2.01	<p>The system must have an interface for shipment receivers to enter a unique identifier from shipment invoices.</p> <p><b>Rationale:</b> Receivers need to add new shipments within the system, to account for newly purchased stock.</p> <p><b>Test Scenario:</b> Use a receiver account to add mock purchase invoice into the system. Switch to stocker account and check for newly added invoice within the inventory.</p>
2.02	The system must update inventory data to include new shipment data, including products, quantities, and expiration dates for non-perishables.
2.03	The system must save invoice form data to the database.
2.04	<p>The system must analyze input data, ensuring the inputs are complete (every requested input field is provided) and within the allowed parameters.</p> <p><b>Rationale:</b> Improper or incomplete data in the database may cause the system to fail when using that data.</p> <p><b>Test Scenario:</b> Insert invalid or incomplete data into system. Expect message</p>

	warning user of invalid or incomplete data. The data should not have been added to the database.
--	--------------------------------------------------------------------------------------------------

### 3.3 System Notification Flags

#### 3.3.1 Description and Priority

The system sends a notification to managers when a flag is raised.

#### 3.3.2 Stimulus / Response Sequences

When there is a discrepancy between what was ordered versus what arrived at the store, the receivers raises a flag in the system. The system will then forward this flag information to the corresponding Store Manager.

#### 3.3.3 Functional Requirements

REQ	Description
3.01	<p>The system must notify the correct manager for the store and not all users registered as managers.</p> <p><b>Rationale:</b> Any given store can have multiple managers all in different departments. Only the managers concerned with a given notification should be notified. Also, managers from other stores are not concerned with the shipment details of a different store.</p> <p><b>Test Scenario:</b> Prompt a system flag by entering an invoice number that does not match an order. Monitor which user accounts receive an alert notifying them of the discrepancy.</p>

### 3.4 Downtime

#### 3.4.1 Description and Priority

If the global system is inaccessible to the store, the store will record all database changes to an in-store caching device. This caching device will record all database changes over a period of time set by the IT department. When a connection is reestablished, the cache will update the system database with the changes that occurred during downtime. If both the system and local device are inaccessible, the user changes made during this time must be stored on the employee iPad and applied when the device is available again. When multiple conflicting changes cannot be automatically resolved, the system will notify and prompt the manager for resolution. This is a high priority feature.

#### 3.4.2 Stimulus / Response Sequences

Internet service to the store is unavailable and system communication falls back to the local device. Handheld devices now send database changes to the local device. Internet service is

restored and the caching device updates the system database with the new information.

The system and device are unavailable due to power failure while a shipment of goods arrives. The receivers perform normal activities and the system will store the employee input data on the employee's handheld devices until a connection to the database can be reestablished. If the power failure lasts long enough that the employee device cache is filled, the user will be notified that the system is temporarily unavailable.

### 3.4.3 Functional Requirements

REQ	Description
4.01	<p>The handheld cached data stored must fit within 10 GB at a time.</p> <p><b>Rationale:</b> Recent store inventory data must exist locally to still support application functionality in case of internet/power failures.</p> <p><b>Test Scenario:</b> The device will be disconnected from the server and the data cached on the device will be used to view current inventory status.</p>

## 3.5 Statistics Tracking

### 3.5.1 Description and Priority

The system will collect information from Wholesome Eats Inc. data stores and its proprietary database on the following data points:

- Inventory levels
- Expiration dates
- Shipment errors
- Number of products currently on shelf
- Number of spoiled products

This feature will provide the basis for most other system features, making this high priority.

### 3.5.2 Stimulus / Response Sequences

Inventory levels, expiration dates, and any shipment errors will be entered into the system by receivers when a shipment is obtained. The number of a product on the shelf at any given time is calculated from information provided by the POS system and store employees. When a shelf is restocked to capacity, the store employee will indicate this to the system. When a sale is made, the POS system decreases the inventory count in the database. IOPS then updates the amount of that product still on the shelf. Lastly, store employees will be able to indicate to the system items that have expired or been damaged when necessary.

### 3.5.3 Functional Requirements

REQ	Description
-----	-------------

<b>5.01</b>	<p>The system must allow for product's shipment amounts and expiry dates to be entered into the database.</p> <p><b>Rationale:</b> Manager reports need to contain monthly figures of expired goods, shipment amount, to better adjust purchase of future stocks.</p> <p><b>Test Scenario:</b> Receiver adds in mock shipment amounts and expired goods into the system. Newly added figures for expired stock and new shipment amount are reflected into the weekly manager report created by IOPS.</p>
<b>5.02</b>	<p>The system must allow for the number of damaged/spoiled items to be entered into the database.</p>

## 3.6 Reports

### 3.6.1 Description and Priority

The system provides historical and current details on stock levels and sales to Wholesome Eats managers and corporate users. Reports are significant in their role of evaluating other features of this system. They provide access to the stocks database in a human readable manner. Additionally it will help in evaluating future solutions for Wholesome Eats. This makes basic reports a high priority. Complex reporting and fully tailored user interfaces for different individuals are a low priority.

### 3.6.2 Stimulus / Response Sequences

Reports are produced by the system when requested by a manager or corporate users within Wholesome Eats. Data for the reports is drawn directly from the database. Select personnel will access the web application and choose a type of report for a specific time period. Some reports will require a range of dates to be selected.

### 3.6.3 Functional Requirements

REQ	Description
<b>6.01</b>	<p>The system must only allow select members (managers and corporate users) to view reports.</p> <p><b>Rationale:</b> Only managerial employees of the business require instant access to store statistics and reports. Therefore, store floor employees such as stockers and receivers do not need permission to view reports.</p> <p><b>Test Scenario:</b> Log into IOPS with a mock managerial account (manager, corporate, system admin) and try and view reports and statistics. The system should successfully display charts according to given parameters. Log into IOPS from a mock store floor employee account. The account should have no access to viewing reports or statistics.</p>

<b>6.02</b>	The system must provide historical reports for as far back as database records are available.
-------------	-----------------------------------------------------------------------------------------------

## 3.7 User Privileges

### 3.7.1 Description and Priority

Users will have different interfaces and privileges depending on their user class. This feature is required in order to provide access to the system for users while having security of individual logins.

### 3.7.2 Stimulus / Response Sequences

User privileges will depend on their user class which is defined by their account type. Their interface will be selected after they log in, with different features and access depending on their privileges.

### 3.7.3 Functional Requirements

REQ	Description
<b>7.01</b>	<p>The System Administrators must have permissions to modify the privileges and account types of other users.</p> <p><b>Rationale:</b> Wholesome Eats administration needs edit employee permissions into the system depending on new hires or changes in employee status.</p> <p><b>Test Scenario:</b> Change specified employee permission for a stocker to a receiver. Login from the specified employee account. Confirm account now has receiver access and functionality instead of stocker accessibility.</p>
<b>7.02</b>	<p>The System Administrators must have permissions to create new users.</p> <p><b>Rationale:</b> Wholesome Eats administration needs to add employees into the system depending for new hires.</p> <p><b>Test Scenario:</b> Create new employee account for IOPS. Confirm account now has correct employee type privileges and functionality.</p>
<b>7.03</b>	<p>The System Administrators must have access the store stock and POS system.</p> <p><b>Rationale:</b> Wholesome Eats administration needs to reset, configure store inventory count and status.</p> <p><b>Test Scenario:</b> Attempt to successfully update store back and front stock levels using a system administrator account. Expect to see appropriate changes in new store inventory levels.</p>

<b>7.04</b>	The Manager department must have permissions to change stock.
<b>7.05</b>	The Manager department must have permissions to view all store stocks.
<b>7.06</b>	<p>The Corporate accounts must have permissions to view all the statistics and information of sales and stock changes.</p> <p><b>Rationale:</b> Wholesome Eats employees from corporate needs to view time based store performance or stock levels of its individual Wholesome Eats stores in different cities.</p> <p><b>Test Scenario:</b> Attempt to view monthly inventory, damaged/expired stock, sales reports from corporate account. Expect to see appropriate reports for appropriate stores.</p>
<b>7.07</b>	Employees in the stocking department must have permissions to change the status of stock locations in the store (front or back).
<b>7.08</b>	<p>Employees in the stocking department must have permissions to flag stock that is missing or damaged.</p> <p><b>Rationale:</b> Store employees need to account for damaged, missing, or expired stock, in an effort maintain an accurate log of current store inventory. This information can be further used by managers to micro-manage particular sections of the store.</p> <p><b>Test Scenario:</b> Use a stocker account to attempt to flag damaged and or missing goods from IOPS. Expect to see updated inventory from stocker menu.</p>
<b>7.09</b>	Employees in the receiving department must have permissions to add new stock that was shipped.
<b>7.10</b>	Employees in the receiving department must have permissions to flag damaged goods.

### 3.8 System Administrative User Registration/Removal

#### 3.8.1 Description and Priority

The administrative interface of IOPS has a multitude of users including stockers, receivers, managers, etc. that need to be added or removed from the system as required by Wholesome Eats Inc. prior to them being able to use the system. This feature is of high priority and is mandatory to complete. Initial System Admin accounts will need to be set up by the IOPS development team.

#### 3.8.2 Stimulus / Response Sequences

Upon logging into the IOPS Admin accounts, users will have the option to create new users, remove and/or modify existing users, and view system administrative user base. Both creating

and removing users will have associated forms to fill, before confirming the action. New users will be assigned credentials, and will be prompted to change them upon first login. Any future changes to employee credentials can be made by contacting the IT department.

### 3.8.3 Functional Requirements

REQ	Description
8.01	<p>IOPS must allow the addition of users with various privileges into the system.</p> <p><b>Rationale:</b> New hires will need accounts to be able to access the system. New hires can include stock employees, cashiers, managers, etc., all of which require different permissions.</p> <p><b>Test Scenario:</b> Create two accounts, one with manager privileges and one with receiver privileges. Try and view reports as the manager, and try and enter an invoice number as the receiver. Both should be successful. Try and view reports as the receiver. This should not be successful.</p>
8.02	<p>IOPS must allow the removal and modification of users from currently existing system.</p>

## 3.9 User Authentication

### 3.9.1 Description and Priority

All employees need to be able to login to the system in order to access features available as per their user privileges.

### 3.9.2 Stimulus / Response Sequences

An employee arrives at work and needs to access the system to modify some stock status. The employee enters their credentials and waits for the system to verify. If the credentials are valid, the system allows the user into the system. Otherwise, the system rejects their authentication.

### 3.9.3 Functional Requirements

REQ	Description
9.01	<p>The system must correctly verify employee credentials and allow them access to the system.</p> <p><b>Rationale:</b> IOPS should only allow store employees to access particular store information.</p> <p><b>Test Scenario:</b> Attempt to log into the system with correct credentials for store employee. Expect to see employee type specific interface.</p>

<b>9.02</b>	The system must reject authentication if login credentials are not valid.
<b>9.03</b>	The system must allow users to log out.
<b>9.04</b>	<p>The system must automatically log user out after five minutes of inactivity.</p> <p><b>Rationale:</b> If an employee iPad is left unattended while the employee helping a customer or chasing a thief, the system should not be accessible to an unauthorized user.</p> <p><b>Test Scenario:</b> Log into the system. Leave system idle for 5 minutes. Try and access system and observe whether authentication is required first.</p>

## 4 External Interface Requirements

### 4.1 User Interfaces

The systems provides a set of different interfaces depending on the user:

- System views for administrators,
- Inventory views for stockers and managers,
- Receivers views for receivers and stockers,
- Report views for managers and corporate users.

Users who have access to multiple views should be able to easily navigate between them.

#### 4.1.1 System View

This view provides administrative tools to support the web application. This includes user registration and removal, any database and caching controls, as well as manual access to system flags. Because this view is meant for a small set of technical employees it should be designed with functionality in mind.

#### 4.1.2 Inventory View

The inventory view provides information to stockers on the current situation in their store. This includes sending notifications when restocking is needed as well as the removal of past due goods. In addition to notifications this interface provides stockers inputs to modify stock amounts. Additionally, managers should be able to configure product-specific information from this interface. This includes modifying properties such as the expiration dates of products and designating shelf space.

#### 4.1.3 Reciever View

The receiver view should allow users to enter a unique ID to specify an incoming shipment. This will update the other sections of the application to reflect the new stock.



#### **4.1.4 Report View**

The report view is a static section of the application meant to allow managers and corporate users to review data collected by the system. This interface should allow users to select different report types, as well as parameters for generating reports, such as selecting a range of dates that a report should cover.

### **4.2 Hardware Interfaces**

The application will be primarily operated within web browsers, having to support devices running two particular operating systems, Apple iOS for store iPads, and Windows 7 for managers workstations. All inter-device communication will take place by means of the internet, therefore, having to support the required device and internet protocols to interact with device network adapters.

### **4.3 Software Interfaces**

#### **4.3.1 Point of Sales System (POS)**

The POS system is an existing Wholesome Eats system that reads sold items and sends the sold data to a database. This system utilizes a scanner that inputs a single item that is sold, and outputs the information to their existing database. The scanner can read checked items, referenced with existing stock items and is sent to the Wholesome Eats product database. The sent data is in the form of a JSON file, and is delivered within 100ms. This data transfer is sent to the database with a HTTP POST method.

#### **4.3.2 Wholesome Eats Database**

The Wholesome Eats Database is an external database that currently exists at Wholesome Eats.

This MySQL relational database system stores inventory information, which includes prices, stocks, and product. Requests and commands are made to this database through HTTP methods, which are executed within 200ms. Data sent to and from this database utilize the JSON format.

#### **4.3.3 iOS**

iOS is a mobile operating system designed by Apple inc. that runs on all Apple mobile devices. This system will allow users to access the customer product that allows users to view Wholesome Eats' store stock.

#### **4.3.4 Windows 7 Compatible Web Browser**

A Microsoft Windows 7 compatible browser is used since the client company utilizes the Windows 7 operating system for their business computers. Employees of the client company, Wholesome Eats, will use the browser to access this product. The input from the client through the browser is sent to Wholesome Eats server, while the output is displayed on the client's screen.

### **4.4 Communications Interfaces**

As IOPS will be a browser based application, it will need to support all popular web browsers at the point

of formulating this document including:

- Google Chrome (Version 55.0)
- Internet Explorer 8
- Edge
- Mozilla Firefox 50.0
- Opera

IOPS browser-side modules will also have to communicate with productions servers, which should be done strictly over HTTPS protocol. Therefore, all IOPS user interaction needing to retrieve information from Wholesome Eats. Inc. databases will be encrypted. All notifications to store employees and managers will exist within the system. Therefore, no additional requirements of e-mail communication need to be established. However, addition and removal of registered store employees within IOPS would require e-mail communication to configure user credentials.

## 4.5 Data Flow Diagrams

### 4.5.1 Level 0 Diagram

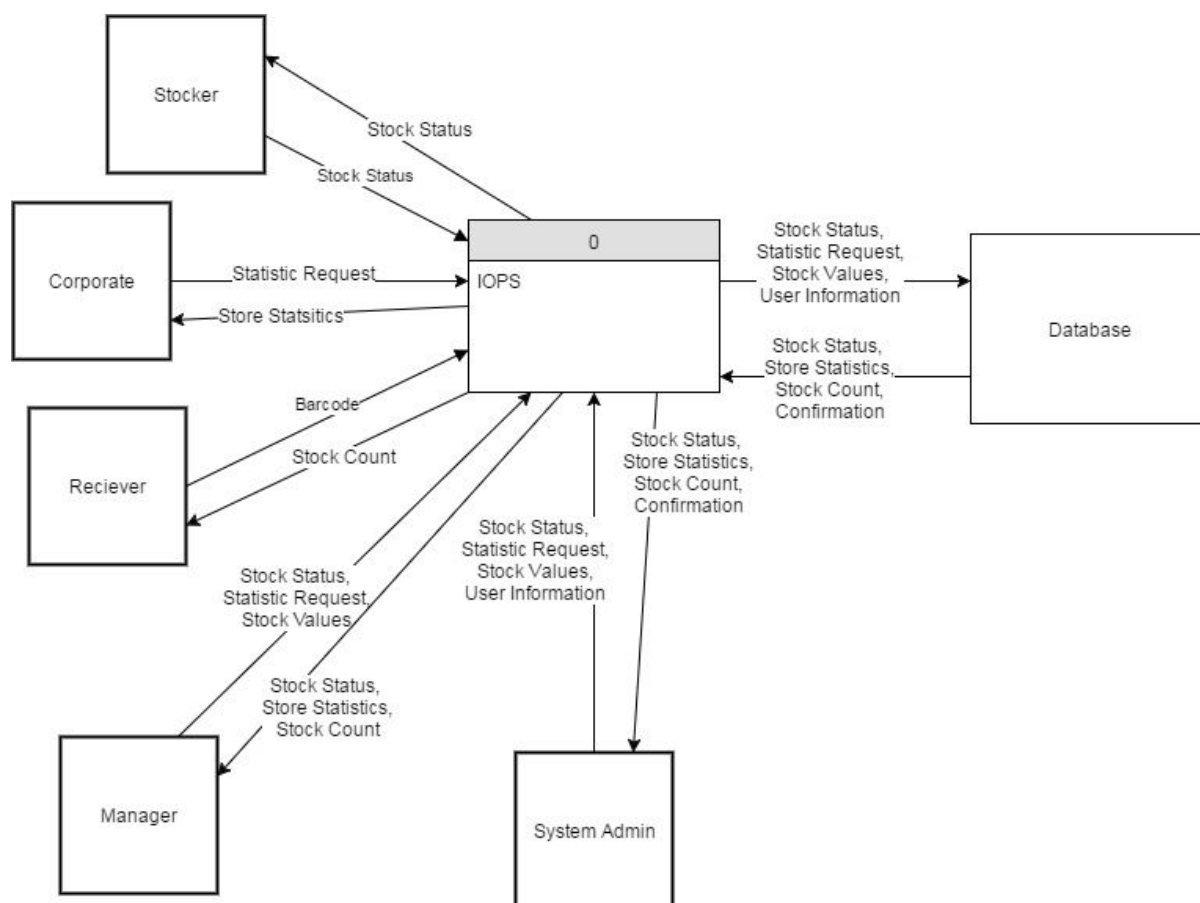


Figure 7: Level 0 Data Flow Diagram for the IOPS

A high level data flow diagram for the IOPS. The only external entities are the employees of Wholesome Eats as well as their database which stores product information.

#### 4.5.2 Level 1 Diagram

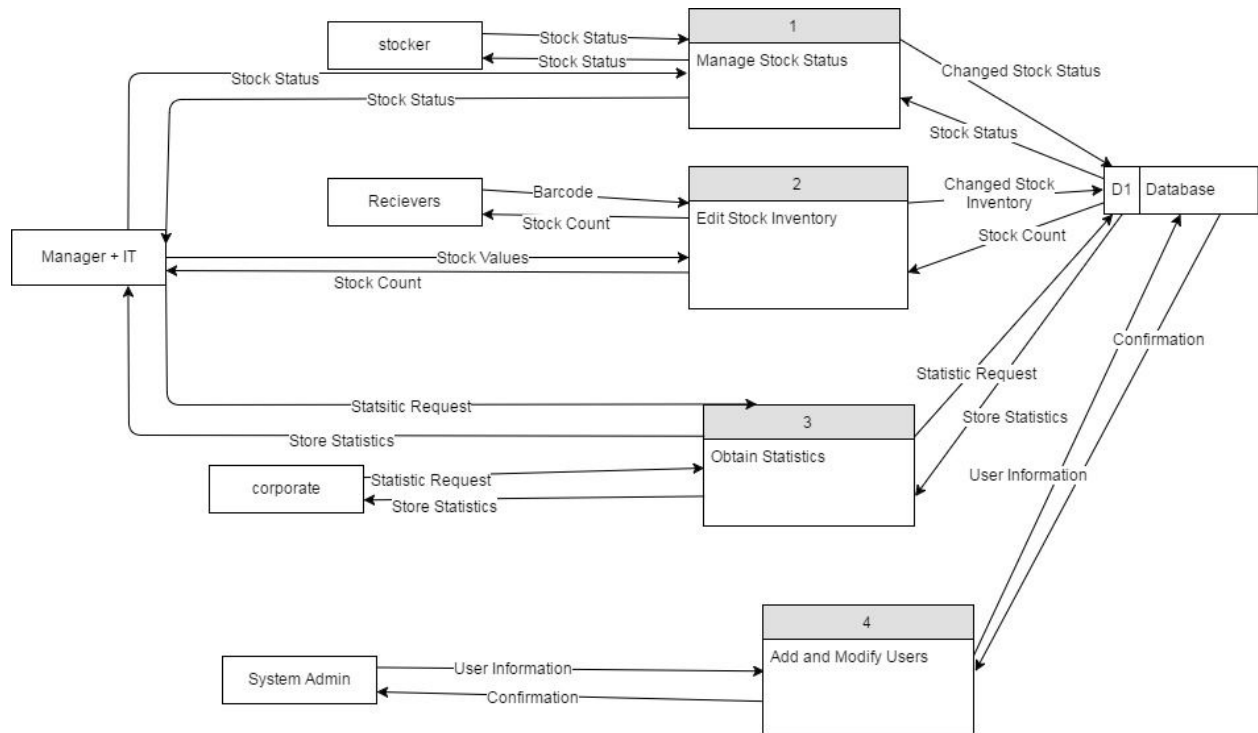


Figure 8: Level 1 Data Flow Diagram for the IOPS

The main functions of IOPS are the ability to manage stock status, edit stock inventory, obtain store statistics, and to add and modify user accounts and privileges.

#### 4.5.3 Level 2 Diagram - Part 1

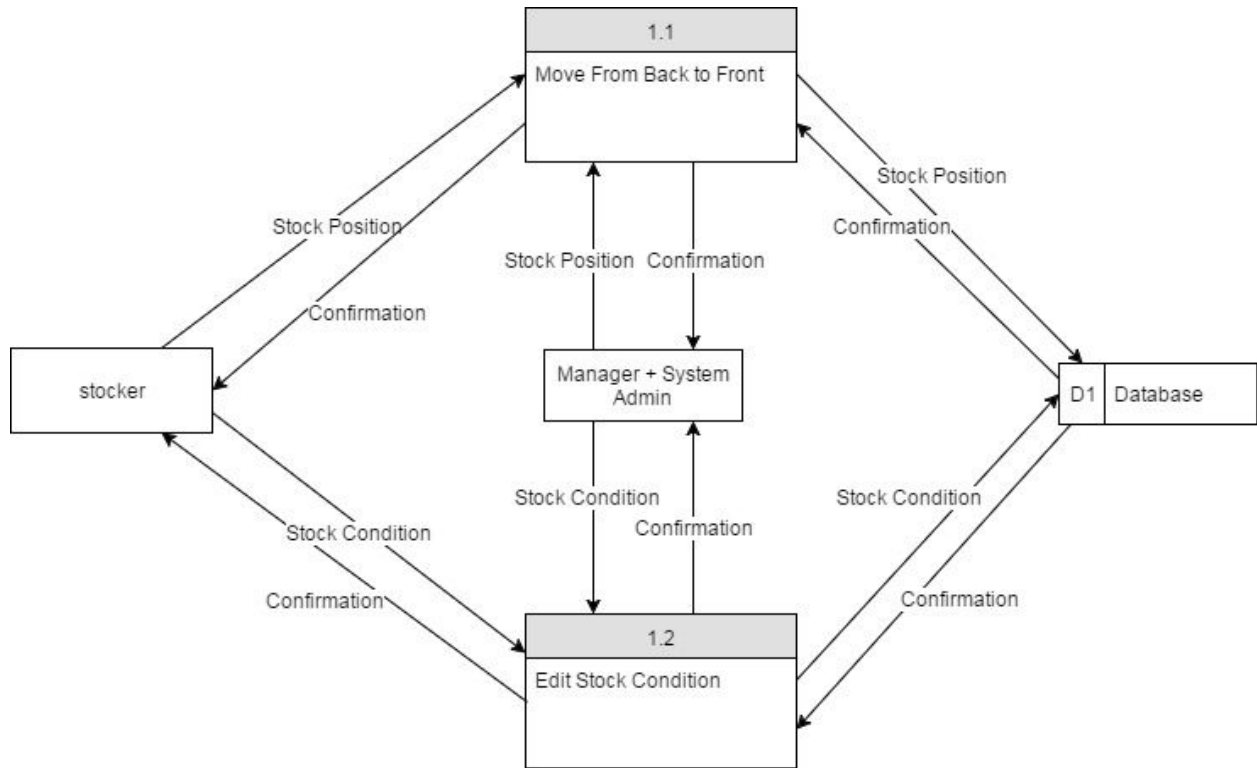


Figure 9: Level 2 Data Flow Diagram Expanding on Process 1, Manage Stock Status.

Looking closer at process 1, manage stock status, this process gives the ability to move the position of stock from the back of the store to the front. Additionally, it gives the ability to mark stock as expired.

#### 4.5.4 Level 2 Diagram - Part 2

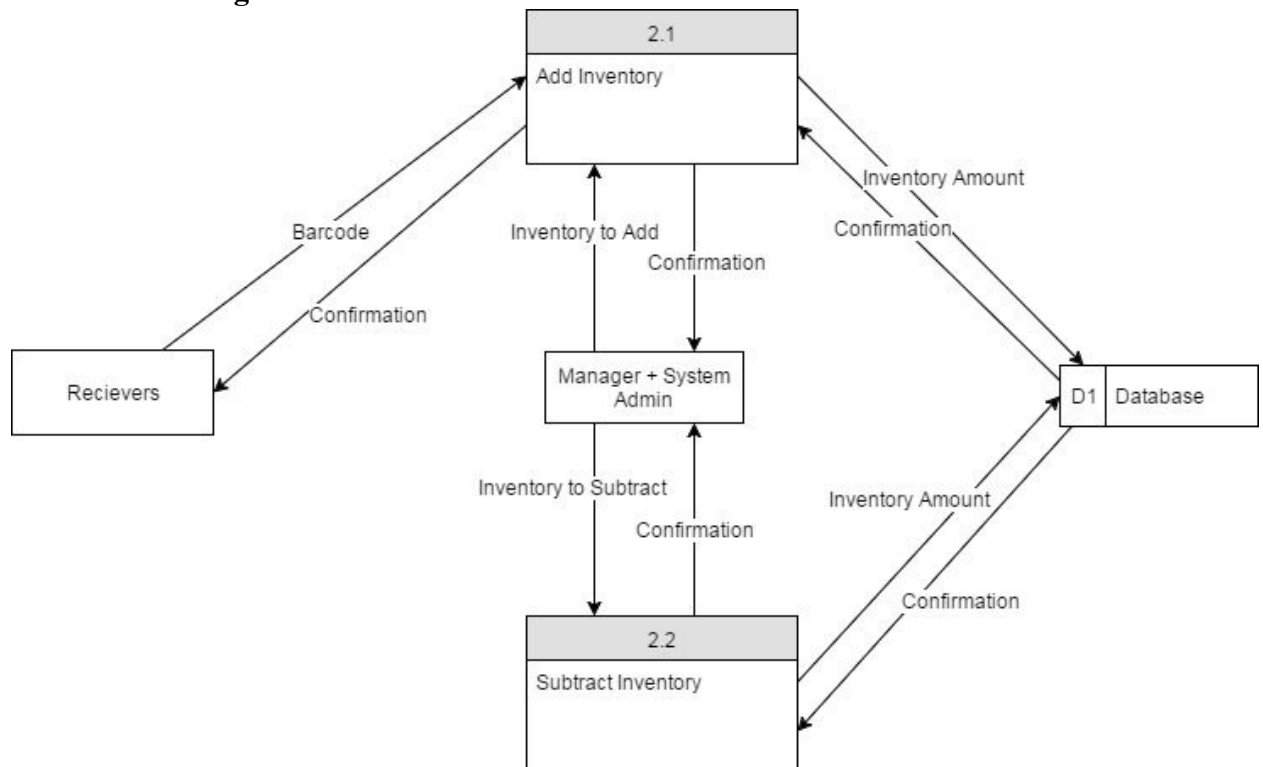


Figure 10: Level 2 Data Flow Diagram Expanding on Process 2, Edit Stock Inventory

Looking deeper into process 2, edit stock inventory, this process gives the ability to add to the total amount of inventory as well as subtract from the total inventory.

#### 4.5.5 Level 2 Diagram - Part 3

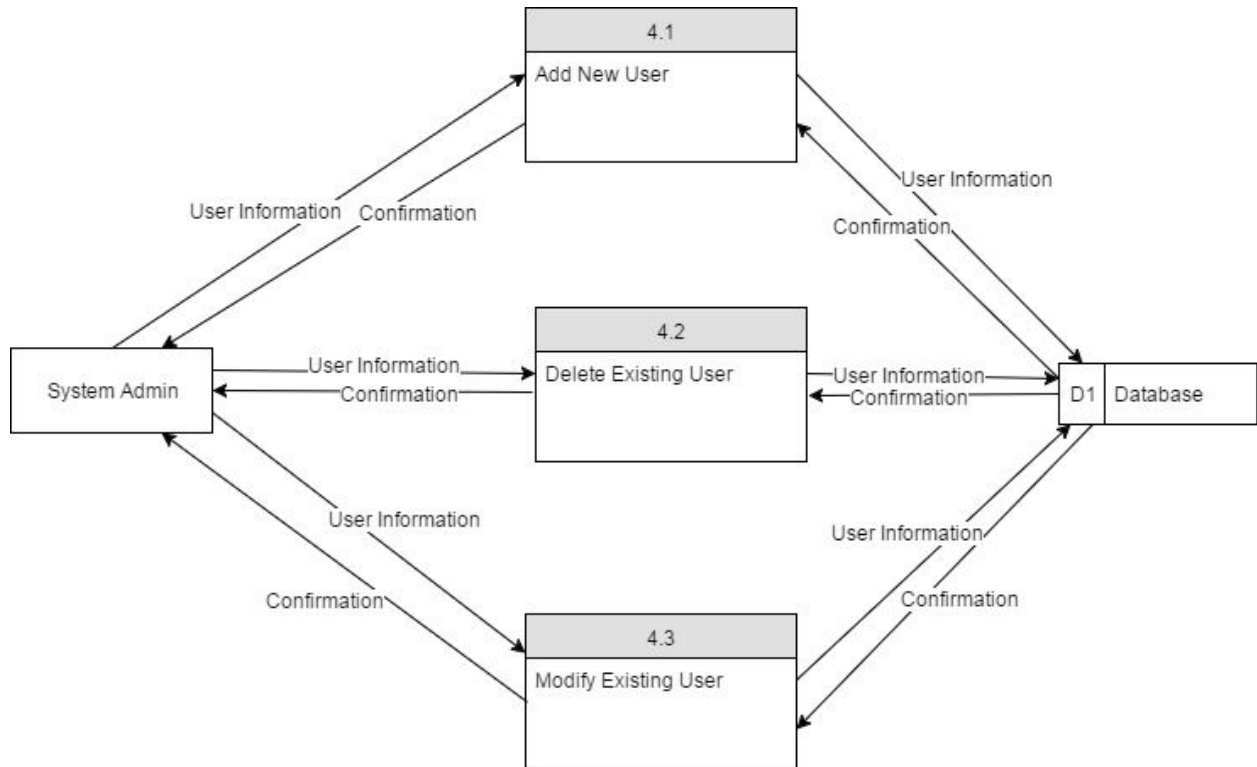


Figure 11: Level 2 Data Flow Diagram Expanding on Process 4, Add or Modify Users

A closer look at process 4, add or modify users, this process gives the ability to add a new user into the system. It also gives the ability to delete or modify information of existing users.

#### 4.6 E/R Diagram

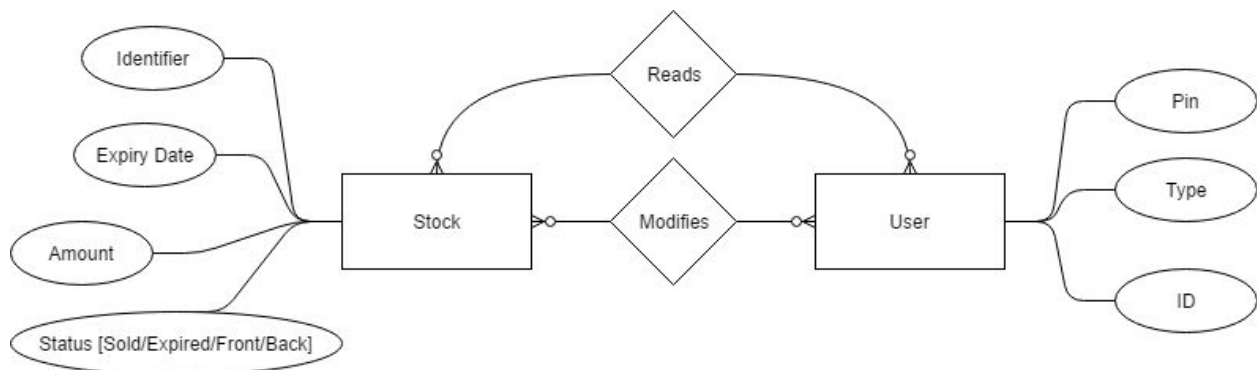


Figure 12: Entity-Relationship Diagram

Each users has their own PIN, type and ID and each Stock has its own Identifier, Expiry Date, Amount and Status. Many users can modify and read many stock.

## **5 Other Non-Functional Requirements**

### **5.1 Performance Requirements**

The IOPS application UI should be responsive and adhere to standard web application UI interaction response times of 0.1s. All network calls, and server side API response times for initial bytes of data should be limited to 400ms, provided a minimum average global internet speed of 5.6Mbps [5]. This will accommodate for additional device rendering delays, and provide for smooth user interactions. IOPS users should also experience smooth transitions when alternating between various application pages, experiencing transition times of less than 300ms.

### **5.2 Safety Requirements**

The notification system of the IOPS application should provide subtle, effective, but not severely distracting notifications to operating users. If using push notifications for mobile devices, the system should not occupy more than 10% of available screen space. IOPS should strictly enforce reminders for removal of out of date goods and produce, to ensure Wholesome Eats Inc. customers are not sold expired items. IOPS should also provide warnings when asking stockers to move overly heavy goods from store back to front.

On application boot, IOPS must provide safety messages warning users to be wary of surroundings when using the application.

### **5.3 Security Requirements**

All users of IOPS must initially need to authenticate with the application by means of a two-factor authentication mechanism. All user credentials must be safely encrypted using currently accepted cryptographic standards when storing them within the database. All inter-device IOPS communication must be carried out using HTTPS protocols, after obtaining a valid SSL certificate.

If idle, users must be timed out and have to re-authenticate every shift. All iPads are proprietary to each employee while in store premises, and must therefore utilise their 4-digit authentication PIN upon getting locked out of their own devices. All internal store devices should only be able to access the application while within store premises, by the usage of static local device IP's. All internal development and communication API endpoints must be private and take into account of common vulnerability threats (i.e. SQL injections, etc) and be carefully sanitized. All application form inputs must also be carefully sanitized and take into account common vulnerability threats (i.e. cross site scripting, etc).

### **5.4 Software Quality Attributes**

The following quality attributes are vital to the associated stakeholders of IOPS:

#### **5.4.1 Usability**

- Initial user testing will be done prior to deployment on a single Wholesome Eats store, and employee ratings should be greater than 80% overall for final deployment.

#### **5.4.2 Availability**

- IOPS functionality for clients of Wholesome Eats Inc. should not face downtimes of more than 24hrs every two quarters.
- IOPS backend production servers for employee web client interaction should be available 20 hrs/day, covering all of the daily working hours, leaving the rest for regular maintenance and upgrades if need be.

#### **5.4.3 Correctness**

- All information shown to the associated stakeholders (i.e. employees, managers) should consist of the most recent database figures (data from up to 20 seconds ago), and should not contain outdated inventory figures.

#### **5.4.4 Maintainability**

- IOPS code development structure should be modular and follow object-oriented development practices as specified in *Object-Oriented Design Heuristics* [4] in order to support future expansion of the platform.
- IOPS development pipelines must adapt continuous integration and delivery in order to support hotfixes, high priority bugs, and new business requirements.



## 6 References

- [1] Carrot Inc. (2017, Jan). *Elicitation Notes*. [Online]. Available: <https://web.uvic.ca/~rafatm/elicitationNotes.pdf> [Jan 19, 2017].
- [2] IEEE (1998) *IEEE Recommended Practice for Software Requirements Specifications*. The Institute of Electrical and Electronics Engineers Inc., pp. 10-20. [Jan 20, 2017].
- [3] Wholesome Eats Inc. (2017, Jan). *Inventory Optimization Suite Request for Proposal*. Wholesome Eats Inc. [Online]. Available: <https://web.uvic.ca/~rafatm/5-1.RFP.pdf> [Jan 17, 2017].
- [4] [1]A. Riel, *Object-oriented design heuristics*, 1st ed. [Place of publication not identified]: Addison-Wesley, 2011. Addison-Wesley, 2011.
- [5] Internet Speeds By Country [Online]. Available: <https://www.fastmetrics.com/internet-connection-speed-by-country.php#top-10-worldwide>