

# **Software Requirements Specifications**

## **Inventory Optimization Suite**

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

There are page numbers on the table of contents, but there aren't page numbers on any of the pages 😞

## Contents

Revision History	6
1 Introduction	7
1.1 Purpose	7
1.2 Project Scope	7
1.3 Glossary of Terms	7
1.4 Overview	8
2 Overall Description	8
2.1 Product Perspective	8
2.2 Product Features	8
2.3 User Classes and Characteristics	9
2.3.1 System Administrator	9
2.3.2 Manager	9
2.3.3 Corporate User	9
2.3.4 Receiver	9
2.3.5 Stocking Employee	9
2.3.6 Customer	9
2.4 Operating Environment	9
2.5 Use Cases	10
2.5.1 Add or Subtract Stock	10
2.5.1.1 Description	10
2.5.1.2 Actors	10
2.5.1.3 Pre-Conditions	11
2.5.1.4 Main Flow	11
2.5.1.5 Post-Conditions	11
2.5.1.6 Alternative Flows	11
2.5.2 Change Stock Status	11
2.5.2.1 Description	11
2.5.2.2 Actors	11
2.5.2.3 Pre-Conditions	11
2.5.2.4 Main Flow	11
2.5.2.5 Post-Conditions	12
2.5.2.6 Alternative Flows	12
2.5.3 Login and Logout	12
2.5.3.1 Description	12

2.5.3.2 Actors	12
2.5.3.3 Pre-Conditions	12
2.5.3.4 Main Flow	12
2.5.3.5 Post-Conditions	12
2.5.3.6 Alternative Flows	12
2.5.4 View Stock Status	13
2.5.4.1 Description	13
2.5.4.2 Actors	13
2.5.4.3 Pre-Conditions	13
2.5.4.4 Main Flow	13
2.5.5 Add and Modify Users	13
2.5.5.1 Description	13
2.5.5.2 Actors	13
2.5.5.3 Pre-Conditions	13
2.5.5.4 Main Flow	13
2.5.5.5 Post-Conditions	14
2.5.5.6 Alternative Flows	14
2.5.6 Add New Shipments to Inventory	15
2.5.6.1 Description	15
2.5.6.2 Actors	15
2.5.6.3 Pre-Conditions	15
2.5.6.4 Main Flow	15
2.5.6.5 Post-Conditions	15
2.5.6.6 Alternative Flows	15
2.5.7 View Statistics	16
2.5.7.1 Description	16
2.5.7.2 Actors	16
2.5.7.3 Pre-Conditions	16
2.5.7.4 Main Flow	16
2.5.7.5 Post-Conidtions	16
2.5.7.6 Alternative Flows	16
2.6 Design and Implementation Constraints	16
2.7 Assumptions and Dependencies	17
2.7.1 Receiver Shipment Documentation -	17
2.7.2 Wholesome Eats Inc. Employee Administration Service	17
3 System Features	17
3.1 Product Configuration	17

3.1.1 Description and Priority	17
3.1.2 Stimulus / Response Sequences	17
3.1.3 Functional Requirements	17
3.2 Restocking	18
3.2.1 Description and Priority	18
3.2.2 Stimulus / Response Sequences	18
3.2.3 Functional Requirements	18
3.3 Receiving	18
3.3.1 Description and Priority	18
3.3.2 Stimulus / Response Sequences	18
3.3.3 Functional Requirements	19
3.4 System Notification Flags	19
3.4.1 Description and Priority	19
3.4.2 Stimulus/Response Sequences	19
3.4.3 Functional Requirements	19
3.5 Downtime	19
3.5.1 Description and Priority	19
3.5.2 Stimulus/Response Sequences	19
3.5.3 Functional Requirements	20
3.6 Statistics Tracking	20
3.6.1 Description and Priority	20
3.6.2 Stimulus / Response Sequences	20
3.6.3 Functional Requirements	20
3.7 Reports	20
3.7.1 Description and Priority	21
3.7.2 Stimulus/Response Sequences	21
3.7.3 Functional Requirements	21
3.8 Customer App	21
3.8.1 Description and Priority	21
3.8.2 Stimulus/Response Sequences	21
3.8.3 Functional Requirements	22
3.9 User Privileges	22
3.9.2 Stimulus/Response Sequences	22
3.9.3 Functional Requirements	22
3.10 System Administrative User Registration/Removal	23
3.10.1 Description and Priority	23
3.10.2 Stimulus/Response Sequences	23

3.10.3 Functional Requirements	23
3.11 User Authentication	23
3.11.1 Description and Priority	23
3.11.2 Stimulus / Response Sequences	23
3.11.3 Functional Requirements	24
4 External Interface Requirements	24
4.1 User Interfaces	24
4.1.1 System View	24
4.1.2 Inventory View	24
4.1.3 Reciever View	24
4.1.3 Report View	24
4.1.4 Customer Application	25
4.2 Hardware Interfaces	25
4.3 Software Interfaces	25
4.3.1 Point of Sales System (POS)	25
4.3.2 Wholesome Eats Database	25
4.3.3 iOS	25
4.3.4 Windows 7 Compatible Web Browser	25
4.4 Communications Interfaces	26
5 Other Non-Functional Requirements	26
5.1 Performance Requirements	26
5.2 Safety Requirements	26
5.3 Security Requirements	26
5.4 Software Quality Attributes	27
5.4.1 Usability	27
5.4.2 Availability	27
5.4.3 Correctness	27
5.4.4 Maintainability	27
5.4.5 Testability	27
6 References	27
Appendix A: Issues List	29

# Revision History

Name	Date	Reason for Changes	Version
Initial Draft	2017/1/20	Drafting	0.1
Initial Draft 2.0	2017/1/26	Finalizing Content	0.2
Final Draft	2017/1/30	Editing	1.0

# 1 Introduction

## 1.1 Purpose

This document describes the software requirements specifications for the Inventory Optimization Suite (IOPS), version 1.0, as requested by Wholesome Eats Inc. The following pages detail all application features, characteristics, interfaces, and other functional and nonfunctional requirements of the IOPS system. 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 ***The system is primarily for mobile devices, but a non trivial part of the system will still be used in a web app interface***

Inventory Optimization Suite is a proprietary application targeted to store employees, managers, corporate officers, and customers 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. A client facing version of IOPS would also be available for customers of Wholesome Eats Inc., displaying store promotions and current inventory in real time.

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>CUA</b>	Customer Application
<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

## 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

**Please clarify if the administrator interface encompasses the manager interface (and that it is indeed supposed to be a WEB application; the iPad is for regular employee use)**

The Inventory Optimization Suite (IOPS) is intended to replace Wholesome Eats' current partially-physical inventory tracking system. IOPS will communicate with the existing inventory database to avoid the need for changes to the POS system. The system will have two different interfaces, both part of a core web application, targeted to Wholesome Eats' store employees and their customers. Store employees will be interacting with the administrator interface on iPads they are provided with, allowing them to use IOPS' core inventory management functionalities, while customers will be interacting with the public interface through their native mobile web browsers, allowing them to view a real time inventory status overview. The administrator interface of IOPS will have the ability to read and update the existing inventory database, while the customer interface will only have read permissions, allowing its users to view what products the store carries and whether or not they are in stock.

### 2.2 Product Features

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

- System Administrative User Management
  - The system will allow super users to add, remove and modify administrative users such as stockers, receivers, managers, etc. into the system.
- Inventory Management
  - The system will allow administrative users to add, remove, and alter stock-related information and allow for complete store inventory management through its administrative web app interface.
- Customer Stock Overview
  - The system will allow for a separate read-only real-time stock overview of current store inventory to customers of Wholesome Eats Inc.
- Management and Corporate Reports and Analytics
  - The system will provide store managers with appropriate visualizations and analytic measurements of store metrics over a chosen time period i.e. stock levels, sales levels, expired and rotten goods levels, etc.
- Smart Notification System
  - The system will aid store floor employees by means of device notifications in times of low stock levels and product expiration to facilitate store restocking procedures.

**Employees must also be able to update the stock levels as they place items on the shelves - otherwise the system will never be up to date**



## 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 the system when necessary. They can add, remove, and modify users in the system and their privileges. As administrators must have full control over the system but will likely interact with little of it during normal operation, administrators only need to be somewhat favoured.



### 2.3.2 Manager

**Fine, as long as limited to their own store (we're a big chain, we have some of the greatest assets in the world). AND only READ access to reports, not write**

Managers run a specific storefront, and have the highest security clearance in a given store. They have read and write access to overall stock count, discrepancies, and 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. Therefore, an intuitive, yet dense, user interface and an easy to navigate application are important features to implement.

### 2.3.3 Corporate User

Corporate users are users that may need to evaluate a store's inventory or review the system's history. They may review all data, similarly to managers, but are limited to read-only mode. Corporate are favoured like managerial users. However, since corporate users cannot modify information, an easily understandable interface is the base concern.

### 2.3.4 Receiver

**Stocking and receiving employees are critical to the success of the system. They are one of the most important users groups to satisfy. They of a higher priority than customers and corporate users, and potentially even managers. Receivers and stocking employees are critical.**

Receivers bring shipped inventory into the store. They input the amount of stock received, and report any discrepancies regarding the invoice. Receivers may modify the quantity of a product in the system, input the expiry date by scanning invoice barcodes, and flag alerts for missing goods. Receivers will interact with very little of the system, and are therefore not very important to satisfy. This little interaction suggests that the interface should display as little as necessary, and be efficient to use so receivers can minimize system use.

### 2.3.5 Stocking Employee

Stocking employees move stock from the back of the store to the shelves. They also remove expired stock from the front to be thrown out. Stocking employees may input how much stock they are moving from the back to the front, and log how much stock they are throwing out. Stocking employees interact with a similar amount of the system as receivers, and are similarly not very important to satisfy. They will require a similarly simplified interface as receivers.

### 2.3.6 Customer

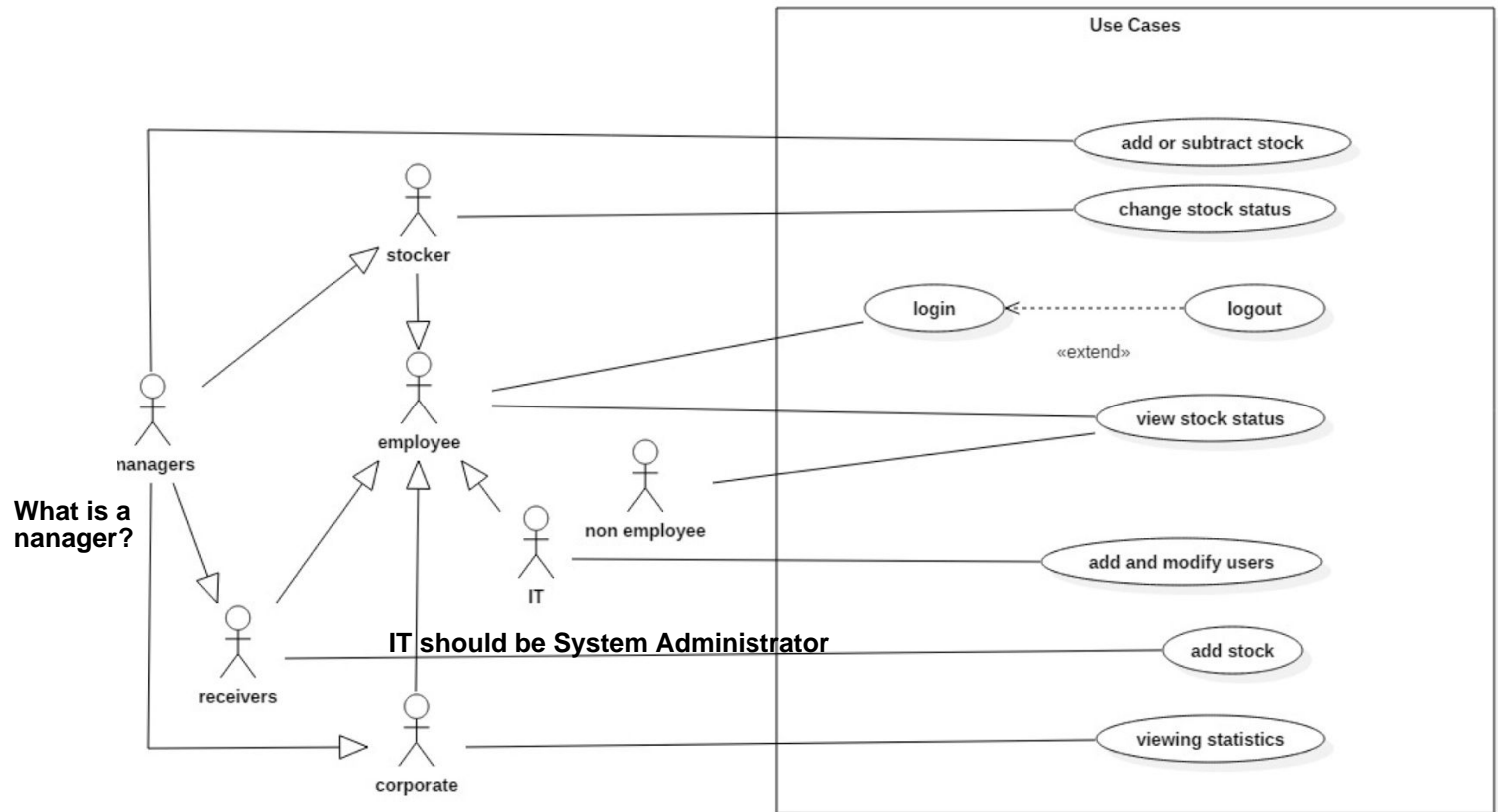
Customers may wish to know what the cost of a product is, as well as the quantity left in a specific store. Customers have read-only access to this information. As customers may only observe small portions of data and have no method of directly modifying the system, customers are the least important users to satisfy.

## 2.4 Operating Environment

Inventory Optimization Suite will connect to and access data from the existing Point of Sales system that currently controls a database which contains data on the stock and store information. The IOPS connection to the POS must be both read and write in order to update information and stock at the request of managers and employees. IOPS will be in the form of a web app, accessible via browser. The web app

must be compatible with current popular web browsers and Windows 7, as used by managers, as well as iPad tablets as used by stockers. The customer facing web app must have a read-only interface that can display stock for customers using modern web browsers, mainly Safari, Firefox, Chrome, Opera and Internet Explorer.

## 2.5 Use Cases



**Figure 2.5.1: Use-Case Diagram for IOPS**

Descriptions of all actors present within Figure 2.5.1 can be found in section 2.3. All of Stocker, Manager, Receiver, Corporate and IT inherit directly or indirectly from store employees; with the exception of non-employees (described as customers within section 2.3), who can only view stock status within the customer-specific app interface. Corporate also has a read-only view of the manager analytics, whereas IT can only manage users within the application.

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

**IT should have permissions to do whatever they want. They likely won't ever have to, but we don't want that functionality to be locked down. They should be able to fix errors, if a manager makes a mistake for example.**

Store manager, IOPS system

#### 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 and the stock is now correct.
9. 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>**

### 2.5.2 Change Stock Status

#### 2.5.2.1 Description

A store employee is notified that some stock is low. The employee logs into their IOPS account and sees that there is product in storage. After bringing it out to the front, he changes the status of the product to indicate it has been moved from the back to the front.

#### 2.5.2.2 Actors

Stocker, IOPS system

#### 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 fingerprint authorization.
3. The system opens the main 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.

**We don't want a finger  
print, we want a simple  
PIN**

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>.

### 2.5.3 Login and Logout

#### 2.5.3.1 Description

This use case describes how a user logs in and out of the IOPS system.

#### 2.5.3.2 Actors

Store employee, IOPS system

#### 2.5.3.3 Pre-Conditions

The store employee must have an account that is authorized to log into the system. Either the employee's login credentials or fingerprint must be stored in the system's database.

#### 2.5.3.4 Main Flow

1. The user opens the Wholesome Eats web app.
2. The user selects the Login button.
3. A page is loaded displaying Username and Password fields.
4. <Login>The user enters their login credentials and presses Login.
5. <Main Menu>The web app displays the main interface of the user depending on their account type.
6. The user can use the IOPS system depending on their user privileges.
7. The user completes their tasks.
8. <Logout>The user selects the Logout button, which has replaced the Login button from Step 2.
9. The user is logged out, returning to the home page of the web app.
10. The use case ends.

#### 2.5.3.5 Post-Conditions

The user is capable of logging back into the system.

#### 2.5.3.6 Alternative Flows

A. At <Login>, the user can instead select a fingerprint scanner option:

1. The user uses the Touch ID fingerprint scanner.
2. The credentials associated with the fingerprint are entered and the user logs in.
3. Return to <Main Menu>.

**No fingerprints, just use a login**

- B. At <Logout>, the user instead closes the browser:
1. The user closes the browser without logging out.
  2. The system automatically logs the user out.
  3. Use case ends.

## 2.5.4 View Stock Status

### 2.5.4.1 Description

**Please generalize use case to include all items, not just turkeys**

This use case describes how a customer checks the availability of turkeys at Wholesome Eats.

### 2.5.4.2 Actors

Customer, IOPS system

### 2.5.4.3 Pre-Conditions

The user must have internet access.

### 2.5.4.4 Main Flow

**The customer app and the employee app are entirely different entities (we want two apps). The employees have an app, and customers have an app. The employee app is much higher priority than the customer app.**

1. The user opens the Wholesome Eats web app.
2. The system displays options to view the website as an employee or non-employee.
3. The user selects “Non-Employee”.
4. The system displays a search bar.
5. The user types “turkey” into the search bar.
6. The system displays a list of related products.
7. The user selects the preferred item.
8. The system displays details regarding the product’s current stock levels and freshness.
9. The user looks at the information and closes the application.
10. The use case ends.

## 2.5.5 Add and Modify Users

### 2.5.5.1 Description

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

### 2.5.5.2 Actors

IT employee, IOPS system, Store manager

### 2.5.5.3 Pre-Conditions

The IT employee must have been added to the system with IT employee privileges

### 2.5.5.4 Main Flow

1. The use case begins when the store manager contacts an IT employee.
2. The IT employee accesses the IOPS website.
3. <Main Menu> The system displays various IT controls.
4. <Add New User> The IT employee selects Add New User.
5. The system displays a form to be filled with user information.
6. <New User Form> The IT employee fills in the form with the new user’s full

**Employee ID must be entered as well**

**Their username should  
just be their employee ID**

name, date of birth, user type, and hiring date, and selects Confirm.

7. **<New User Form Assessment>** The system assesses if the form is complete.
8. The system uses the information to generate a unique username and temporary password and stores the user information in the system database.
9. The system removes the form and displays the username and temporary password.
10. The IT employee contacts the store manager, and informs them of the generated username and password.
11. The IT employee selects Return to Main Menu
12. **<Returned to Main Menu>** The system displays various IT controls.
13. The use case ends.

#### **2.5.5.5 Post-Conditions**

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

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

#### **2.5.5.6 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 IT employee is not already authenticated, then:

1. The IT employee enters their credentials.
2. The IOPS system authenticates the credentials.
3. Return to **<Main Menu>**.

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

1. The system ignores the input from the IT employee.
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 IT employee selects Modify Existing User.
2. The system requests which use the IT employee would like to modify.
3. The IT employee selects the user.
4. The system displays a form with the selected user’s attributes.
5. **<Modification Form>** The IT employee 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 IT employee selects Cancel instead of Confirm, then:

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

## 2.5.6 Add New Shipments to Inventory

### 2.5.6.1 Description

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

### 2.5.6.2 Actors

Store receivers, IOPS system, Database.

### 2.5.6.3 Pre-Conditions

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

### 2.5.6.4 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 a fingerprint scan.
5. <**Main Menu**> The system brings the user to the main menu of the site.
6. The receiver navigates to the new shipment page.
7. <**Add Stock**> Using the invoice provided by the shipping company, the receiver enters the unique identifier and product count.
8. The system prompts the receiver to finalize the new stock input.
9. <**Input Check**> The system looks up the corresponding product data based on the unique identifiers.
10. The database is updated.

### 2.5.6.5 Post-Conditions

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

### 2.5.6.6 Alternative Flows

A. At <**Login Credentials**>, if the fingerprint scan fails the system will:

1. Display a message stating “Invalid Scan”.
2. Display the form to be filled with the user’s login and password.
3. Return to <**Main Menu**>.

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

will:

1. Display a message stating “Invalid Identifier” and display the identifier.
2. Return to <Add Stock>.

## **2.5.7 View Statistics**

### **2.5.7.1 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.

### **2.5.7.2 Actors**

Store manager, Corporate User

### **2.5.7.3 Pre-Conditions**

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

### **2.5.7.4 Main Flow**

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

### **2.5.7.5 Post-Conditions**

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

### **2.5.7.6 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.

## **2.6 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 system

Additionally, in the event that a store loses power or the database crashes, any changes to the inventory database must be stored in an offline cache.



## 2.7 Assumptions and Dependencies

### 2.7.1 Receiver Shipment Documentation -

Currently, it is assumed that all shipment documentation as received by receivers include sufficient information to elicit all required fields by IOPS, such as:

- 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. If not, additional measures need to be implemented in order to meet requirements as provided within the RFP.

### 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. If this is not the case, IOPS administrative user account credentials will only be modifiable through Wholesome Eats Inc. IT services.

## 3 System Features

### 3.1 Product Configuration

#### 3.1.1 Description and Priority

REMOVE TEXT

All internal users may modify stored values in the system to record inventory changes depending on their user privileges. A stocking employee may modify the record of items stored in the front, receivers may log shipments of goods and when the goods expire, and managers may modify all values, such as price or erroneous values. The POS system will also modify values to record customer sales. This is a high priority feature.

#### 3.1.2 Stimulus / Response Sequences

What are the actual stimulus/response points?

A shipment of inventory is received. A receiving employee logs the additional stock into the system, along with the expiry date by filling out a form. If a discrepancy in the shipment is noticed by the employee, they may optionally fill out a portion of the form noting the problem.

#### 3.1.3 Functional Requirements

REQ	Description
1.01	The POS must be able to interact with the system.
1.02	The system must be capable of denying invalid data.

## 3.2 Restocking

### 3.2.1 Description and Priority

Provide notifications for 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.

### 3.2.2 Stimulus / Response Sequences

A notification will display on the user'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 user 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 store front and back stock counts in the database.

Which user  
gets  
notified?



### 3.2.3 Functional Requirements

REQ	Description
2.01	The system must contain an interface that will display restocking notifications.
2.02	The system must contain an interface where administrative users can enter the product id and quantity of what they have removed from the storefront, the product number and quantity of what they have moved from the stockroom to the storefront, and the reason(s) for doing so.
2.03	The system must be able to calculate when products need to be taken off the shelf due based on their expiration dates.
2.04	The system must be able to calculate when products need to be moved from the stockroom to the storefront, based on stock counts and sales.

## 3.3 Receiving

### 3.3.1 Description and Priority

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

### 3.3.2 Stimulus / Response Sequences

The user will have a screen where they can enter a unique code 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.

### 3.3.3 Functional Requirements

REQ	Description
3.01	The system must have an interface for shipment receivers to enter the unique identifier from shipment invoices.
3.02	The system must update inventory data to include new shipment data, including products, quantities, and expiration dates for non-perishables.

## 3.4 System Notification Flags

### 3.4.1 Description and Priority

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

### 3.4.2 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.

### 3.4.3 Functional Requirements

REQ	Description
4.01	The system must notify the correct manager for the store and not all users registered as managers.

## 3.5 Downtime

### 3.5.1 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 manager. When a connection is reestablished, the cache will update the system database with the changes that had 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.

### 3.5.2 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.

**IT should set the cache timer, not the managers**

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.

### 3.5.3 Functional Requirements

REQ	Description
5.01	The handheld cached data stored must fit within 16 GB at a time

## 3.6 Statistics Tracking

### 3.6.1 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.

### 3.6.2 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.

### 3.6.3 Functional Requirements

REQ	Description
6.01	The system must allow for product's shipment amounts and expiry dates to be entered into the database.
6.02	The system must allow for the number of damaged/spoiled items to be entered into the database.

## 3.7 Reports

### 3.7.1 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.

### 3.7.2 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.

### 3.7.3 Functional Requirements

REQ	Description
7.01	The system must only allow select members (managers and corporate users) to view reports.
7.02	The system must provide historical reports for as far back as database records are available.

## 3.8 Customer App

### 3.8.1 Description and Priority

The system will provide a customer facing application. This interface should provide customers with information regarding the stock of specific goods at their local Wholesome Eats. The customer application should provide fuzzy values for stock, as in specific stock numbers are not necessarily provided, instead customers will be warned if a product is low. The customer application is a lower priority.

### 3.8.2 Stimulus/Response Sequences

The customer app will be accessed by customers over the internet. They will specify their local Wholesome Eats. They will then be able to use the app to query the current stock of the grocery store. This query will make contact the system's database to provide up-to-date information. Data retrieved from the database is described in fuzzy language and presented to the user.

### 3.8.3 Functional Requirements

REQ	Description
8.01	The customer application must be available on customer devices.
8.02	The customer application must provide stock information for all Wholesome Eats locations.
8.03	The customer application must specify stock amounts in fuzzy values.
8.04	The customer application does not allow access to another other parts of the system.
8.05	The customer application must be available during peak shopping times.

## 3.9 User Privileges

### 3.9.1 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.

### 3.9.2 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.

### 3.9.3 Functional Requirements

REQ	Description
9.01	The IT department must have permissions to modify the privileges and account types of other users.
9.02	The IT department must have permissions to create new users.
9.03	The IT department must not have access the store stock and POS system.
9.04	The Manager department must have permissions to change stock.
9.05	The Manager department must have permissions to view all store stocks.
9.06	The Corporate accounts must have permission to view all the statistics and information of sales and stock changes.

IT should  
have access  
to everything

<b>9.07</b>	Employees in the stocking department must have permissions to change the status of stock locations in the store (front or back).
<b>9.08</b>	Employees in the stocking department must have permissions to flag stock that is missing or damaged.
<b>9.09</b>	Employees in the receiving department must have permissions to add new stock that was shipped
<b>9.10</b>	Employees in the receiving department must have permissions flag damaged goods.
<b>9.11</b>	Customer accounts must have permission to view stock that is available.

### 3.10 System Administrative User Registration/Removal

#### 3.10.1 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 IT accounts will need to be set up by IOPS development team.

#### 3.10.2 Stimulus/Response Sequences

**Not all users have email, just higher ups. If no email, their password is reset when they call IT**

Upon logging into the IOPS IT 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, but the option to alter credentials will be sent via email to company email.

#### 3.10.3 Functional Requirements

<b>REQ</b>	<b>Description</b>
<b>10.01</b>	IOPS must allow the addition of users with various privileges into the system.
<b>10.02</b>	IOPS must allow the removal and modification of users from currently existing system.

### 3.11 User Authentication

#### 3.11.1 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.

#### 3.11.2 Stimulus / Response Sequences

An employee gets to 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 should allow the user into the system, otherwise, reject authentication.

### 3.11.3 Functional Requirements

REQ	Description
11.01	The system must correctly verify employee credentials and allow them access to the system.
11.02	The system must reject authentication if login credentials are not valid.
11.03	The system must allow users to log out.

## 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,
- Customer application.

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 in specific sections 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.



### **4.1.3 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.

### **4.1.4 Customer Application**

The customer application is meant to provide summary stock information to the public. Customers should be able to select their local Wholesome Eats and review current stock information.

## **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. The application should also interface with all popular smart mobile devices to meet Wholesome Eats Inc's customer facing requirements. 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 barcode scanner reads in barcodes as binary data, which is then 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 safely encrypted. All notifications to store employees and managers would be by means of native device push notifications, 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. Internal company email accounts are required to be used for initial account configuration and future modifications.

## 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 standard internet connectivity. 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

volume levels  
arent important  
in this  
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. If using device speakers, IOPS should limit volume levels to 25% of maximum capability. 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

Only single  
factor  
authentication.

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.

Fingerprints  
are not  
allowed for  
login

If idle, users must be timed out and have to re-authenticate every shift. All Ipad are proprietary to each

Make sure logout is  
permitted when iPad is  
locked when another  
employee needs to login.

Not every employee gets an iPad. Store shares a limited amount of iPads

**employee interface should only be accessible from the internal network. ie (the store)**

employee while in store premises, and must therefore implement fingerprint recognition 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**

- IOPS application-user interaction must meet the requirements of all stakeholders by being intuitive, easy to localize and globalize (i.e. language support, etc), and resulting in a good overall end user experience.
- The reporting feature for the manager class of users in particular should be easy to use for non-technical users, and sufficient documentation should be provided explaining regular use-cases.

### **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 20hrs/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, customers, managers) should be up to date with most recent database figures, and should not provide particularly customers, with outdated inventory figures.

### **5.4.4 Maintainability**

- IOPS code development structure should be modular and follow object oriented development best practices in order to support future expansion of the platform.
- IOPS development pipelines must adapt deployment methodologies such as continuous integration and delivery, in order to support hotfixes, high priority bugs, or new business requirements.

### **5.4.5 Testability**

- IOPS code development structure and architecture should allow for intensive unit, integration, and end-to-end automation testing schemes, in order to best replicate application usage and discover bugs before deployment to production

## 6 References

[1] Carrot Inc. (2017, Jan). *Elicitation Notes*. [Online]. Available: <https://web.uvic.ca/~rafatm/elicitatationNotes.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].

## Appendix A: Issues List

- How much time should pass before a user is automatically logged out
  - For employees
  - For customers
- How many Wholesome Eats stores are to be supported by the system

**many stores should be  
supported by the system,**

**5 minutes must pass before a  
user is logged out**