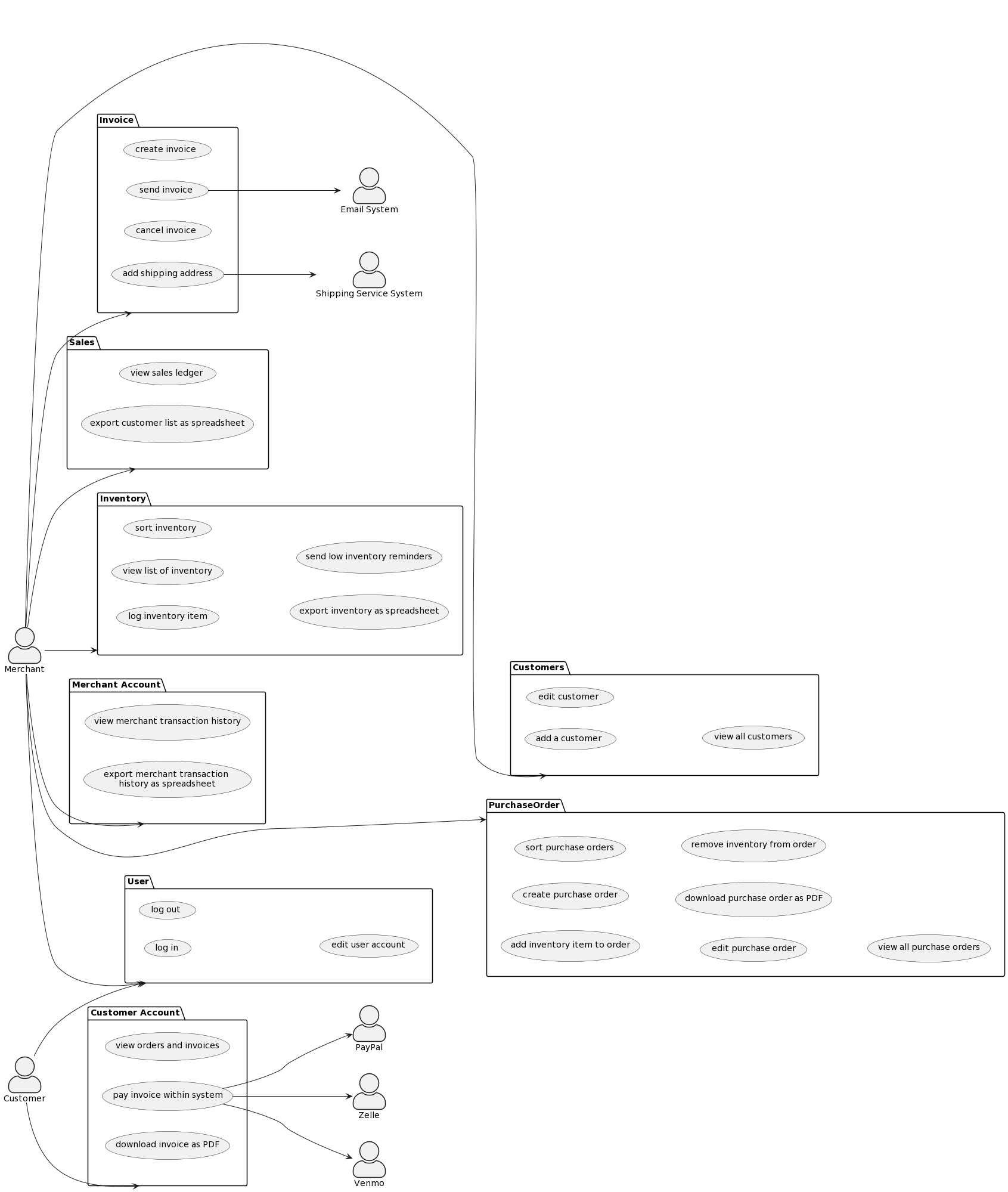
IMS (Inventory Management System)

Use-Case Specification

Version 0.3

# Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 10/19/22 | 0.1 | Added diagrams for all use cases.  Added Use Cases 1 and 2. | John Atkins |
| 10/20/22 | 0.1 | Added Use Cases 3 and 4 | Priya Rivera |
| 10/22/22 | 0.1 | Added Use Cases 5 and 6 | Lawrence Yang |
| 10/23/22 | 0.1 | Added Use Cases 7 and 8 | Connor Svelling-Pescatore |
| 10/23/22 | 0.2 | Updated use case diagram | Lawrence Yang |
| 10/23/22 | 0.3 | Further updated use case diagram with external actors. Added Use Case 9 and 10. Added page breaks between each Use Case Specification. Changed mentions of Client to Customer for consistency. | Sean Kilroy |
| 10/24/22 | 0.3 | Added brief descriptions to ToC | Priya Rivera |

****

**Figure 1 - Use Case Diagram -** [**Plant UML link**](http://www.plantuml.com/plantuml/png/TLNDJXin4BxlKunoHu842bmgAjfAFLID5D53rSl8EoGM_bdPtar3LOa7QL-E9oasdiJhBBUTRpxFpyndxuqFw4AlLUCVfEdGeGOKmRfLs2a2t9AtcXf5zm621IVNcm2jT2I2jAP9LFXAJcpG140FEcEtlGzMaukOoDXijP5T9yqQLkG6AGXMEnz8poB9OoUPzqcZLAMJ2kSITqjK2UuA-0SfHGbzBD3lPBHDw13H3iK3hWdk_6jbFSBl1dfF0Zt1JDarI9FeTxVpoyfZ-y2kgy95hGpHSe12sDxiITVDdtt2BsOW4wpRJUHZZmoayzcpWZH8seAIFe2zFx0vIK7qreKZmk90U7xwIx-w4IaDldE4hTyGSUNdPL0oBIYxBSuuqjAqv7ocNnHzplPQ-50CG_EOkOxKasPgS6Wy9aN1HlgOvFdf7-G6JYSwc3pxaq_NRSLWATQxej9b7oDu-kPQSgEjODjMMudIjv6NEowL8HnX8EXok17lgjfEnU81NbTgcWvsTbhGMhiroc9RKIDp-V4pdoXbv4ZRWOh6xftLumJpMdMer1ibB-R5AFck-wCn8eZiunoBkjUQS3vL19Cyiox6TyjAGPoyt8fyTV6foqhOjRzSX40ZI5MC-MdLeESNxSLmvFVrbKfReI8UJT6NJx2YTasEmy-djCxDfqVWBMdF5-Kjv8sCRg70eoBoKW4Fi_1SIxc83dViXwqC6sd0y-kSo1SJskGJOr4kpbuArN1oylxmA6PxT2KpjjTMjlEoinL7ovuUyuV6HcGaVmXiuaxIT_hFn5qdYw8LwqzMDx8UHzOGhHioRQ_LVm00)

# Table of Contents

[**Revision History**](#_heading=h.5bq22ny2sjkb) **1**

[**Table of Contents**](#_heading=h.g1822zl61lov) **3**

[**1. Use Case Specification: Log Inventory Item**](#_heading=h.nm245u1v4pje) **7**

[1.1. Log Inventory Item](#_heading=h.gjdgxs) 7

[1.1.1.](#_heading=h.35nkun2) The Merchant logs an acquired item into the inventory database 7

[1.2. Flow of Events](#_heading=h.1ksv4uv) 7

[1.2.1. Basic Flow](#_heading=h.44sinio) 7

[1.2.2. Alternative Flows](#_heading=h.2jxsxqh) 8

[1.2.2.1. Item Already Exists](#_heading=h.z337ya) 8

[1.2.2.2. Size](#_heading=h.3j2qqm3) 8

[1.2.2.3. Price](#_heading=h.1y810tw) 8

[1.2.2.4. Images](#_heading=h.4i7ojhp) 8

[1.3. Pre-conditions](#_heading=h.2xcytpi) 8

[1.3.1. Authentication and Authorization](#_heading=h.1ci93xb) 8

[1.3.2. Database](#_heading=h.3whwml4) 8

[1.4. Post-conditions](#_heading=h.2bn6wsx) 8

[1.4.1. New item is visible](#_heading=h.qsh70q) 8

[1.4.2. New item is editable](#_heading=h.3as4poj) 8

[**2. Use Case Specification: Create a Purchase Order**](#_heading=h.rxaj0f6dngvu) **9**

[2.1. Create a Purchase Order](#_heading=h.1pxezwc) 9

[2.1.1.](#_heading=h.49x2ik5) The Merchant creates a Purchase Order 9

[2.2. Flow of Events](#_heading=h.2p2csry) 9

[2.2.1. Basic Flow](#_heading=h.147n2zr) 9

[2.2.2. Alternative Flows](#_heading=h.3o7alnk) 9

[2.2.2.1. Customer does not yet exist in database](#_heading=h.23ckvvd) 9

[2.2.2.2. Purchase Order has expired](#_heading=h.ihv636) 9

[2.3. Special Requirements](#_heading=h.1hmsyys) 10

[2.4. Pre-conditions](#_heading=h.41mghml) 10

[2.4.1. Inventory is on-hand in database: Inventory items must be ‘on-hand’ in the availability field to be added to a PO.](#_heading=h.2grqrue) 10

[2.4.2. User is logged in with Merchant permissions](#_heading=h.nbago790pdp1) 10

[2.5. Post-conditions](#_heading=h.vx1227) 10

[2.5.1. Merchant can edit the Purchase Order](#_heading=h.3fwokq0) 10

[2.5.2. Merchant can generate an Invoice](#_heading=h.1v1yuxt) 10

[**3. Use Case Specification: Log In**](#_heading=h.gztcyipmqru2) **11**

[3.1. Use Case](#_heading=h.2u6wntf) 11

[3.1.1. The Merchant or Customer](#_heading=h.19c6y18) logs into the IMS 11

[3.2. Flow of Events](#_heading=h.3tbugp1) 11

[3.2.1. Basic Flow](#_heading=h.28h4qwu) 11

[3.3. Special Requirements](#_heading=h.2lwamvv) 11

[Usernames and passwords shall be encrypted.](#_heading=h.111kx3o) 11

[3.4. Pre-conditions](#_heading=h.3l18frh) 11

[3.4.1. The user must have an account set up with a working username and password.](#_heading=h.206ipza) 11

[3.5. Post-conditions](#_heading=h.4k668n3) 11

[3.5.1. If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.](#_heading=h.2zbgiuw) 11

[4. Use Case Specification: View list of inventory](#_heading=h.1egqt2p) **12**

[4.1. Use Case](#_heading=h.3ygebqi) 12

[4.1.1.](#_heading=h.2dlolyb) The merchant views all inventory 12

[4.2. Flow of Events](#_heading=h.sqyw64) 12

[4.2.1. Basic Flow](#_heading=h.3cqmetx) 12

[4.2.2. Alternative Flows](#_heading=h.1rvwp1q) 12

[4.3. Special Requirements](#_heading=h.3q5sasy) 12

[4.4. Pre-conditions](#_heading=h.kgcv8k) 12

[4.4.1. The user is logged in to the system with authorized access to the inventory database.](#_heading=h.ny4xe3819e5b) 12

[4.5. Post-conditions](#_heading=h.1jlao46) 12

[4.5.1. The system displays an accurate list of all the inventory.](#_heading=h.eoinojaxb0s) 12

[**5. Use Case Specification: View All Customers**](#_heading=h.ms335i3w14wa) **13**

[5.1. Use Case](#_heading=h.xvir7l) 13

[5.1.1.](#_heading=h.3hv69ve) The merchant wants to view a list of all their customers. 13

[5.2. Flow of Events](#_heading=h.1x0gk37) 13

[5.2.1. Basic Flow](#_heading=h.4h042r0) 13

[5.3. Special Requirements](#_heading=h.pkwqa1) 13

[5.3.1. Customer information may be subject to privacy and security regulations](#_heading=h.39kk8xu) 13

[5.3.2. The customer data sent from the system to the user should be sent encrypted over the network.](#_heading=h.bw8ui7u14htx) 13

[5.4. Pre-conditions](#_heading=h.1opuj5n) 13

[5.5. Post-conditions](#_heading=h.2nusc19) 13

[5.5.1. None](#_heading=h.1302m92) 13

[**6. Use Case Specification: Add a Customer**](#_heading=h.3chzd9kbx8r9) **14**

[6.1. Use Case](#_heading=h.2250f4o) 14

[6.1.1.](#_heading=h.haapch) The merchant adds a new customer to their customer list. 14

[6.2. Flow of Events](#_heading=h.319y80a) 14

[6.2.1. Basic Flow](#_heading=h.1gf8i83) 14

[6.2.2. Alternative Flows](#_heading=h.40ew0vw) 14

[6.2.2.1. Duplicate Customer Name](#_heading=h.2fk6b3p) 14

[6.3. Special Requirements](#_heading=h.1tuee74) 14

[6.3.1. Customer information may be subject to privacy and security regulations](#_heading=h.r6ix3ncbt11) 14

[6.3.2. Any form, response, and/or confirmation with customer information shall be encrypted.](#_heading=h.pb012twbngxt) 14

[6.4. Pre-conditions](#_heading=h.2szc72q) 14

[6.4.1. Merchant will be logged in.](#_heading=h.gt4fvryjw3a6) 14

[6.4.2. Only users authorized as Merchant will be able to add to the list of customers.](#_heading=h.54s5n6ah5j5h) 15

[6.4.3. Database exists with customer information associated with the Merchant.](#_heading=h.ktazizdovk0s) 15

[6.5. Post-conditions](#_heading=h.3s49zyc) 15

[6.5.1. A new unique Customer Id is created for the customer.](#_heading=h.279ka65) 15

[6.5.2. The list of customers now includes the newly created customer and their information.](#_heading=h.ncfy673qmfi3) 15

[**7. Use Case Specification: View all Purchase Orders**](#_heading=h.z18844bmickm) **16**

[7.1. Use Case](#_heading=h.36ei31r) 16

[7.1.1.](#_heading=h.1ljsd9k) Merchant wants to view a list of all their purchase orders. 16

[7.2. Flow of Events](#_heading=h.45jfvxd) 16

[7.2.1. Basic Flow](#_heading=h.2koq656) 16

[7.2.2. Alternative Flows](#_heading=h.zu0gcz) 16

[7.3. Special Requirements](#_heading=h.2y3w247) 16

[7.4. Pre-conditions](#_heading=h.3x8tuzt) 16

[7.5. Post-conditions](#_heading=h.rjefff) 16

[**8. Use Case Specification: Sort Purchase Orders**](#_heading=h.6mi7rljarsvn) **17**

[8.1. Use Case](#_heading=h.4anzqyu) 17

[8.1.1.](#_heading=h.2pta16n) The user can filter and sort their purchase orders. 17

[8.2. Flow of Events](#_heading=h.14ykbeg) 17

[8.2.1. Basic Flow](#_heading=h.3oy7u29) 17

[8.2.2. Alternative Flows](#_heading=h.243i4a2) 17

[8.3. Special Requirements](#_heading=h.42ddq1a) 17

[8.4. Pre-conditions](#_heading=h.wnyagw) 17

[8.5. Post-conditions](#_heading=h.1vsw3ci) 17

[**9. Use Case Specification: Create invoice**](#_heading=h.8xfj87d20gz1) **17**

[9.1. Use Case](#_heading=h.1a346fx) 18

[9.1.1.](#_heading=h.3u2rp3q) The Merchant creates an Invoice to bill the Customer 18

[9.2. Flow of Events](#_heading=h.2981zbj) 18

[9.2.1. Basic Flow](#_heading=h.odc9jc) 18

[9.2.2. Alternative Flows](#_heading=h.38czs75) 18

[9.2.2.1. System returns no Purchase Orders.](#_heading=h.1nia2ey) 18

[9.2.2.2. The Merchant needs to adjust the price of the item in Invoice.](#_heading=h.hspolrqjah5o) 18

[9.3. Special Requirements](#_heading=h.11si5id) 18

[9.4. Pre-conditions](#_heading=h.20xfydz) 19

[9.5. Post-conditions](#_heading=h.302dr9l) 19

[**10. Use Case Specification: Download Invoice as PDF**](#_heading=h.h8dmto8f4ay2) **20**

[10.1. Use Case](#_heading=h.2eclud0) 20

[10.1.1.](#_heading=h.thw4kt) The Customer downloads a copy of their invoice as a PDF 20

[10.2. Flow of Events](#_heading=h.3dhjn8m) 20

[10.2.1. Basic Flow](#_heading=h.1smtxgf) 20

[10.2.2. Alternative Flows](#_heading=h.4cmhg48) 20

[10.2.2.1. The system returns no invoices.](#_heading=h.2rrrqc1) 20

[10.2.2.2. The file download fails.](#_heading=h.16x20ju) 20

[10.3. Special Requirements](#_heading=h.261ztfg) 20

[10.4. Pre-conditions](#_heading=h.356xmb2) 20

[10.5. Post-conditions](#_heading=h.44bvf6o) 20

# 

# Use Case Specification: Log Inventory Item

## Log Inventory Item

### Brief Description

The Merchant logs an acquired item into the inventory database, completing all necessary fields for tracking the inventory item.

## Flow of Events

### Basic Flow

The Merchant accesses an interactive item in the system to begin to log a new item into the inventory database.

The system will display a form containing all possible fields necessary for inputting a new inventory item, including:

**Vendor or Brand name**:

* *Examples*: Gucci, Givenchy, Chanel, Yurman.
* *Interaction*: UI will provide a selectable list of unique values in the database, or Merchant can enter a new one.

**Type of item:**

* *Examples*: shoes, handbag, necklace.
* *Interaction*: UI will provide a selectable list of unique values in the database, or Merchant can enter a new one.

**Style of item (optional):**

* *Examples*: Purple Label, Dionysus, etc.
* *Interaction*: UI will provide a selectable list of unique values in the database, or Merchant can enter a new one.

**SKU/Lot Number:**

* *Interaction*: UI will provide a free-text field to enter the vendor’s SKU and/or Lot number for identification purposes.

**Description:**

* *Examples*: color, pattern, material, etc.
* *Interaction*: UI will provide a free-text field.

**Size:**

* *Examples*: small, medium, large, 20-inch, 22-inch, etc.
* *Interaction*: UI will provide a selectable list of unique values in the database, or Merchant can enter a new one.

**Quantity:**

* *Interaction*: UI will provide a free-text field that accepts only numbers.

**Purchase Price, Retail Price, Sale Price:**

* *Interaction*: UI will provide a free-text field that accepts a dollar amount.

**Images:**

* *Interaction*: UI will provide ability to upload one or more images associated with the product.

**Availability:**

* *Interaction*: UI will provide the following selectable options: ordered, on-hand, reserved, sold

Merchant will Submit the item. System will validate all of the provided data. If there are problems, System will provide clear error messages and how they can be fixed.

Upon successful submission, the System will save the data, and present the new item in a read-only view.

### Alternative Flows

#### Item Already Exists

In some cases, inventory for the same item may already exist in the database. If so, the user should be able to select the item and add new details. Described in the next sections are variations and how each should be handled.

#### Size

All fields may be identical, except for different sizes of the same product. In this case, Merchant will increase only the QUANTITY for that SIZE, and the System will log the new inventory appropriately.

#### Price

All fields may be identical, except for PURCHASE PRICE, RETAIL PRICE, SALE PRICE of the same product. In this case, Merchant can create a copy of the item, and enter new PRICE details. PRICE for existing items will remain unchanged.

#### Images

Images will apply to all items that share VENDOR, TYPE, and STYLE, SKU. Merchant can further tag IMAGES related only to DESCRIPTION or SIZE.

## Pre-conditions

### Authentication and Authorization

Only users authorized as Merchant will be able to log an inventory item.

### Database

A database exists containing all of the required tables and relations for this use case.

## Post-conditions

### New item is visible

Once an item is logged, the Merchant will be able to see the item among a list of inventory items.

### New item is editable

Once an item is logged, the Merchant will be able to edit the data in any field of the existing record.

# Use Case Specification: Create a Purchase Order

## Create a Purchase Order

### Brief Description

Merchant creates a Purchase Order that integrates Customer information with inventory items, prices, shipping method, and total price.

## Flow of Events

### Basic Flow

The Merchant selects an option in the UI to create a new Purchase Order.

The System will display an interactive form in which the Merchant can add the required data to complete the Purchase Order. This includes:

**PO Number:**

* *Interaction:* System will generate a unique PO number that will be displayed on the Purchase Order

**Customer Information**:

* *Data:* Customer id, name, company, address, phone, email
* *Interaction*: Merchant will have the option to select an existing Customer, which when selected will populate all the data customer data fields in the view.

**Inventory items:**

* *Data:* vendor, type, style, sku, size, quantity, sale price, discount (optional)
* *Interaction*: Merchant can add any ‘on-hand’ item into the Purchase Order. Once added, the availability of the item changes to ‘reserved’

**Shipping:**

* *Data:* shipping address (name, company, address), shipping vendor, shipping speed, package weight, package size
* *Interaction*: Merchant can select from a list of shipping addresses associated with the selected Customer, or he can add a New Shipping address. Merchant can select from a list of shipping vendors and shipping speeds. Shipping costs will be calculated and included on the Purchase Order.

**Payment Terms:**

* *Data:* due date, invoice number
* *Interaction*: Merchant enters a due date. An Invoice number is automatically created and associated with the Purchase Order.

When the PO is finalized, Merchant will have options for remitting it to the Customer, which may include PDF, email, or a Customer view within the system. These will be described in separate use-cases.

### Alternative Flows

#### Customer does not yet exist in database

The Merchant can select an option to add a New Customer, which will open a New Customer form. Once completed, the Customer will be added to the Purchase Order.

#### Purchase Order has expired

Once the date has passed, the Merchant will be given an option to cancel the Purchase Order. If canceled, the inventory ‘availability’ status will revert to ‘on-hand’.

## Special Requirements

Initially the Shipping options may be entered manually. In the final system, the UI should interface with Shipping Company APIs to automatically calculate shipping costs, as well as to schedule pickups and provide tracking.

Identifying and other sensitive information such as address, phone, e-mail, and customer id shall be encrypted.

## Pre-conditions

### Inventory is on-hand in database: Inventory items must be ‘on-hand’ in the availability field to be added to a PO.

### User is logged in with Merchant permissions

## Post-conditions

### Merchant can edit the Purchase Order

The Merchant can select a Purchase Order from a list of existing POs and edit it.

### Merchant can generate an Invoice

Once the PO is accepted by the Customer, Merchant can generate an Invoice, which will be covered in a separate use-case.

# Use Case Specification: Log In

## Use Case

### Log In

This use case describes how a Merchant or Customer logs into the IMS.

## Flow of Events

### Basic Flow

This use case starts when the actor wishes to log into the Inventory Management System.

* The system requests that the actor enter his/her name and password.
* The actor enters his/her name and password.
* The system validates the entered name and password and logs the actor into the system.

*3.2.1.1.Exception Flow* Invalid name and/or password

* If the actor enters an invalid name and/or password in the Basic Flow, the system displays an error message. The actor can either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends.
* If the actor attempts [X] number of incorrect entries, the system is locked for [Y] amount of time.

## Special Requirements

### Usernames and passwords shall be encrypted.

## Pre-conditions

### The user must have an account set up with a working username and password.

## Post-conditions

### If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.

# Use Case Specification: View list of inventory

## Use Case

### Brief Description

This use case describes how a merchant views the list of all inventory in the IMS.

## Flow of Events

### Basic Flow

The actor wishes to view their product inventory.

* .
* The actor selects the function “View All Inventory”.
* The actor selects “Check Stock”
* The system displays the list of product’s stock arranged by product name.
* The actor inserts the name of a product in a search box.
* The system displays the searched product stock.

### Alternative Flows

None

## Special Requirements

* + 1. The UI should limit the results to X results per page.

## Pre-conditions

## The user is logged in to the system with authorized access to the inventory database.

### 

## Post-conditions

## The system displays an accurate list of all the inventory.

# Use Case Specification: View All Customers

## Use Case

### Brief Description

Merchant wants to view a list of all their customers. The main actor is the merchant, and they are interacting with the IMS.

## Flow of Events

### Basic Flow

* Merchant selects the UI option to “View Customers”.
* System responds by displaying a table of all the customers. The data in the table, initially sorted by Customer Id, is:
  + Customer Id
  + Customer name
  + Company name
  + Address
  + Phone
  + E-mail
* Merchant may select an option to sort the table.
* The system responds with a prompt to select which data field and whether ascending or descending order.
* The merchant selects the data field, and either ascending or descending order..
* The system then displays the table sorted by desired options.
  + 1. *No Customer*

If there is no customer for the merchant, instead of displaying a table, the system will display a message indicating that there are no customers to display, and show an UI option to “Add a Customer”. There is no sorting option available.

## Special Requirements

### Customer information may be subject to privacy and security regulations

### The customer data sent from the system to the user should be sent encrypted over the network.

## Pre-conditions

* + 1. Merchant shall be logged in.
    2. Only users authorized as Merchant shall be able to view the list of customers.
    3. Database exists with customer information associated with the Merchant.

## Post-conditions

### None

# Use Case Specification: Add a Customer

## Use Case

### Brief Description

The merchant adds a new customer to their customer list. The actor is the merchant, and they are interacting with the IMS.

## Flow of Events

### Basic Flow

* Merchant selects the UI option to “Add a Customer”.
* Systems responds with a form with these fields:
  + Customer name
  + Company name (optional)
  + Address
  + Phone
  + E-mail
* Merchant fills in the fields, and clicks an UI element to submit the form.
* The system responds with a confirmation screen, which redisplays the submitted customer data, a form field for a message to send to the customer to be later sent in a welcome e-mail. This form field may be empty. There are UI options for the user to confirm, to edit, and for the user to cancel.
* Merchant sees the confirmation screen, fills the welcome message as desired, and either clicks the UI to confirm, to edit, or to cancel.
* If the merchant clicks to confirm, the system replies with a confirmation that the new customer has been added to the database, and if the e-mail field is not empty, sends a welcome message via E-mail System to the customer with the Customer Name, Customer Id, Merchant’s name, Merchant’s e-mail address, and previously entered welcome message if not empty.
* If the Merchant clicks to edit, the system redisplays the earlier form, but prefilled with the existing information.
* If the Merchant clicks to cancel, the system displays the default UI screen.

### Alternative Flows

#### Duplicate Customer Name

* After the Merchant fills in the form with the new customer information, and clicks submit.
* The system sees that the Customer name duplicates a name already in the system. On the confirmation screen, it has a notification that there is a duplicate customer name, and highlights any data field that duplicates an existing customer. The UI element to confirm the customer data and proceed will not work if all the customer data duplicates an existing customer’s data.

## Special Requirements

### Customer information may be subject to privacy and security regulations

### Any form, response, and/or confirmation with customer information shall be encrypted.

## Pre-conditions

### Merchant will be logged in.

### Only users authorized as Merchant will be able to add to the list of customers.

### Database exists with customer information associated with the Merchant.

## Post-conditions

### A new unique Customer Id is created for the customer.

### The list of customers now includes the newly created customer and their information.

# Use Case Specification: View all Purchase Orders

## Use Case

### Brief Description

Merchant wants to view a list of all their purchase orders. The system will display an interactable summary table of all their purchase orders with the ability to click on a row for a detailed view.

## Flow of Events

### Basic Flow

* Merchant selects the UI option to “View Customers”.
* The system responds by displaying a table that shows key information of individual PO’s
  + Each row contains
    - Purchase order id
    - Date and time of generation
    - Customer full name concatenated from their first and last
    - Scrollable textbox of order notes
    - Line item with highest extended total
    - Total $ order amount
    - Fulfillment status
    - Detail view button
* The user can select a row’s detail button to navigate to that record’s detail page.

### Alternative Flows

* + - 1. No Purchase Orders Associated with Merchant

If the merchant has no purchase orders associated with their user ID, then the table will not display any records and instead will display a message with a link to create a PO. Otherwise, they can navigate away from the page.

## Special Requirements

* + 1. Only records associated with the merchant’s ID or access level will be viewable.
    2. The table will only load X records at a time to ensure smooth operation.

## Pre-conditions

* + 1. The user must be authenticated as a Merchant.
    2. In order to display records, the merchant must enter customers and purchase orders.

## Post-conditions

* + 1. A view is presented with a list of valid purchase orders.
    2. No purchase orders are listed.

# Use Case Specification: Sort Purchase Orders

## Use Case

### Brief Description

The user can use an assortment of radio buttons, dropdowns, and textboxes to filter and sort their purchase orders in the “View all Purchase Order” page.

## Flow of Events

### Basic Flow

* Purchase order list is initially sorted by date of generation in descending order.
  + Sort Options - Select column header to sort, icons indicate desc, or asc order.
    - Purchase order ID - Ascending/Descending
    - Purchase order date - Ascending/Descending
    - Customer name - Ascending/Descending
    - Total order amount - Ascending/Descending
  + Filter Options - In table header, user selects dropdown and specifies value to filter. Filter will remain applied until removed to allow multi-filtering.
    - Purchase order ID - Text
    - Purchase order date - Calendar picker/date-range picker
    - Customer name - Searchable dropdown of customers
    - Total order amount - Two textboxes with comparison operator to allow user to search for amounts between, greater than, less than, or equal to an amount.
    - Order status - Dropdown menu of available order statuses such as “Ordered” and ‘Shipped”
* Table will respond to sort options and reload results accordingly.
  + Sorting results will not result in an API call
  + Filtering results will make an API call in order to load only the results the user is searching for.

### Alternative Flows

* + - 1. No records are found when filtering records.

If there are no records in the database that fit the filters specified by the user, the table will display a message that no records match the search parameters. The user will have the ability to modify their search parameters in order to find results.

## Special Requirements

* + 1. The table will always indicate to the user when it is loading results.

## Pre-conditions

* + 1. The user must be authenticated as a merchant.
    2. The user must be navigated to the “View all Purchase Orders” page.
    3. There must be records in the database in order to sort and filter them.

## Post-conditions

* + 1. The user will have a curated list of purchase orders that fits their specified parameters.
    2. The user can click on individual purchase order records to modify their data.

# Use Case Specification: Create invoice

## Use Case

### Brief Description

The Merchant creates an Invoice to bill the Customer for their corresponding Purchase Order. The system will pull in data from the Purchase Order, calculate Sales Tax due, and pull in other static information such as where and how the Customer can pay the Invoice.

## Flow of Events

### Basic Flow

* The Merchant clicks a button in the UI to Create an Invoice.
* The System generates a table of all open Purchase Orders. The table includes Purchase Order Number and Customer Name to simplify the process of identifying a Purchase Order.
* The Merchant selects a Purchase Order from the table. The Merchant can use the search bar to find Purchase Order.
* The System generates a visual preview of the Invoice with the following:
  + Data from the Purchase Order:
    - Unique Invoice Number and Payment due date
    - Customer Information: Customer ID, name, company, address, phone number, email address
  + Sales Tax calculated for the Invoice based on the address listed in the Customer Information.
  + For each unique product, the following Product Information is included: price, discount, size, quantity, item name
  + Static information defined for how to pay:
    - URL link to pay invoice within the system.
    - Send a check or money order to the Merchant business address as listed in the Merchant User profile.
  + A checkbox to indicate the Purchase Order Agreement has been accepted by the Customer and the Purchase Order has been fulfilled.
* The Merchant reviews the Invoice preview and checks the Purchase Order Agreement and Fulfillment box.
* The system makes the Submit button clickable.
* The Merchant hits the Submit button.
* The system updates the Purchase Order record as agreed and fulfilled. The system stores the generated Invoice to be used for creating PDFs for download as well as making it visible to the Customer in the View Orders and Invoices section as described in other Use Cases.

### Alternative Flows

#### System returns no Purchase Orders.

If the system returns no open Purchase Orders when generating the table or if no results are returned when the Merchant attempts to search by Purchase Order number or Customer Name, then a message indicating that “No open purchase orders exist.” is rendered inside the table graphic. A link to create a new Purchase Order will be rendered alongside this message.

#### The Merchant needs to adjust the price of the item in Invoice.

The Merchant can edit the discount and/or price fields to change discount amounts or account for price adjustments before hitting the Submit button.

## Special Requirements

* + 1. The system must have business logic to calculate Sales Tax, per the law, for the jurisdiction that the Customer has indicated in their address in the Customer Information on the Purchase Order.
    2. The system must be able to reliably generate the list of all open Purchase Orders.
    3. The system must adhere to privacy standards by not constructing an Invoice for the wrong Customer.

## Pre-conditions

* + 1. The system must have a valid Purchase Order containing the Invoice Number matching the respective Invoice. *See the Create Purchase Order Use Case for requirements of a valid Purchase Order.*
    2. The system must not have an existing Invoice record for the Invoice Number located in the Purchase Order.
    3. The system state must have the User logged into their account with Merchant permissions.

## Post-conditions

* + 1. The system will have a completed Invoice associated with a record in the database that has a unique Invoice Number.
    2. The system will update the Invoice record status to indicate that the Invoice is ready for delivery via the methods described in other Use Cases.
    3. The system will allow for Use Cases that require a completed Invoice such as sending an email with the Invoice and downloading the Invoice as a PDF.

# Use Case Specification: Download Invoice as PDF

## Use Case

### Brief Description

The Customer downloads a copy of their invoice as a PDF from within the UI to keep a local record of their purchases. The system must take the Invoice record and generate a PDF file from it, then make it available for download by the Customer.

## Flow of Events

### Basic Flow

* The Customer navigates to the View Orders and Invoices section in the UI.
* The system generates a list of all orders and invoices associated with the Customer.
* The Customer finds the relevant Invoice and clicks the Download as PDF button.
  + The Customer should be able to use a search bar to filter by Invoice number, Created Date, or Due Date.
  + The Customer should be able to filter the list by Invoice number, Created Date, and Due Date.
* The system looks up the Invoice using the associated Invoice number in the record the Customer selected.
* The system takes the unprocessed Invoice record and formats it into a human readable format. If the Merchant has added custom branding and logos for use in their correspondence with a Customer, then the system should add this now.
* The system pushes the file to the Customer’s device for download.

### Alternative Flows

#### The system returns no invoices.

If the system returns no Invoices when generating the table or if no results are returned when the Customer attempts to search by Invoice number, Created Date, or Due Date, then a message indicating that “No Invoices exist.” is rendered inside the table graphic.

#### The file download fails.

If the system fails to push the Invoice PDF to the Customer, then an alert message will prompt the Customer to try the download again.

## Special Requirements

* + 1. The system must obfuscate sensitive information in the Invoice, such as full credit card numbers, for security and privacy reasons.
    2. The system must allow for an acceptable data transfer rate once the Customer begins downloading the PDF.
    3. The system state must have the User logged into their account with Customer permissions.

## Pre-conditions

* + 1. The system must have an agreed upon and fulfilled Purchase Order.
    2. The system must have an Invoice record for the relevant Invoice.

## Post-conditions

* + 1. The system will make a database record to track that the Customer has downloaded the PDF for this Invoice.