



**Bullseye Case Study**  
**PROG1197i Individual Software Development  
Project (ISDP)**

**Specifications Manual**

**Winter 2023**

**Chris London & Aaron Mitchell**

*Last Update: Dec 31, 2022*

## Contents

<b>Introduction.....</b>	<b>3</b>
<b>Expectations .....</b>	<b>4</b>
<b>Sprint 1.....</b>	<b>6</b>
<b>GitHub Project Repository .....</b>	<b>6</b>
<b>Data Access Diagram .....</b>	<b>7</b>
<b>Data Dictionary.....</b>	<b>7</b>
<b>Security .....</b>	<b>8</b>
<b>User Permissions .....</b>	<b>9</b>
<b>Dashboard .....</b>	<b>10</b>
<b>Login.....</b>	<b>11</b>
<b>Logout .....</b>	<b>12</b>
<b>Add User .....</b>	<b>13</b>
<b>Edit User .....</b>	<b>14</b>
<b>Delete User .....</b>	<b>15</b>
<b>Sprint 2.....</b>	<b>16</b>
<b>Create Store Order .....</b>	<b>16</b>
<b>Create Emergency Order .....</b>	<b>17</b>
<b>Receive Store Order .....</b>	<b>19</b>
<b>Fulfil Store Order .....</b>	<b>20</b>
<b>Add Item to Backorder .....</b>	<b>21</b>
<b>View Store Order .....</b>	<b>22</b>
<b>View Location .....</b>	<b>23</b>
<b>Add Location.....</b>	<b>23</b>
<b>Edit Location .....</b>	<b>23</b>
<b>View Inventory .....</b>	<b>24</b>
<b>Edit Inventory .....</b>	<b>24</b>
<b>Move Inventory.....</b>	<b>25</b>
<b>Sprint 3.....</b>	<b>26</b>
<b>Sprint 4.....</b>	<b>26</b>



## Introduction

Welcome to the show! 😊

This is your introduction to the big tamale, the big kahuna, the end game – the capstone project! This capstone project is based on a case study in which you must build a fully functional software system from a combination of given set of specifications and your own creativity and knowledge/skills.

This course is an opportunity for students to gain practical experience in system development. Students apply systems design theory and computer programming skills to complete a small systems development project under the supervision of an instructor.

Student progress is closely monitored by the instructor(s) using intermittent technical reviews and manual submissions. Students program, test, document and deliver a realistic, small-scale system with components written in a variety of languages against an approved Relational Database Management System (RDMS). Emphasis is placed upon ***individual initiative, resourcefulness, and self-discipline*** to build and implement this project from start to finish.

This document contains the guidelines for your project. It is not all-inclusive, there is plenty of room for you to add your own functionality and creativity, but the overall expectation is that your final project will perform the required features as designed with the goal of providing the end users a properly working system.

To ensure we can assess - and you can demonstrate - the range of your abilities, ***you will create a combination of a desktop and web-based app***, using the **provided MySQL database (bullseyedb2023\_1.0)**. If this database script needs to be tweaked along the way, the version will be updated at the end and you will be notified (bullseyedb2023\_1.1, bullseyedb2023\_1.2, etc.).

You may **add new tables, triggers, stored procedures, etc.**, to the database to handle any functionality you may wish to add, but **you may not alter existing tables or data therein**.

If you add new tables/triggers/etc., you must create them using a separate .sql file to run ***after running the file provided***, as we will be running the provided file prior to all sprint reviews to ensure we have a valid database and my test data runs as expected

## Expectations

- Discipline, product ownership, professionalism, dedication to the task at hand
- Efficient coding practices
- Proper commenting
- All code and documentation will be stored in a private GitHub library, which will be shared ONLY with your instructors (via email invite)
- Weekly log in which you journal your progress
- Four (4) sprints with a series of features/requirements to be completed, as listed in this document
- **Assessment:**
  - Log/Journal      5%
  - Sprint 1          20%
  - Sprint 2          25%
  - Sprint 3          25%
  - Sprint 4          25%
- Formal Sprint Planning sessions will occur at the start of each sprint (attendance is mandatory)
- Weekly demonstrations of your progress will occur during scheduled class times
- Formal Sprint Reviews will occur at the end of each sprint (attendance is mandatory)
- Grades based on functionality based on given specifications. If you are unsure of what is meant in a specification, ASK. Getting the specification wrong because you did not ask is not an excuse
- No grades are given for inefficient code, commenting, poor UX / design. At this point in your career, these are expected and while you will not receive credit for doing it, you will be penalized for not following specs, not writing efficient code, not commenting, and not following good UX / design standards
- Every sprint is REQUIRED to successfully complete this course (i.e. you cannot decide not to submit a sprint and be successful)
- This is an **INDIVIDUAL** project. There will be **NO SHARING CODE**. You may only share “*concepts*” with your classmates in an effort to help them out of a jam.
  - (Example: Someone having trouble connecting to a DB with a data table, you could help by giving them advice like “I solved that by using this library \_\_\_\_\_” or “Did you remember to do this \_\_\_\_”.

## Security and Permissions

The Bullseye Inventory Management System (BIMS) has permission-based access. To access a particular aspect of the system, the user logged in must have the appropriate permission(s). Each feature/functionality will have the appropriate permission(s) listed with the other specifications. Most of these permissions are self-explanatory.

### System Permissions:

ACCEPTSTOREORDER  
ADDITEMTOBACKORDER  
ADDNEWPRODUCT  
ADDSITE  
ADDUSER  
CREATEBACKORDER  
CREATELOSS  
CREATEREPORT  
CREATESTOREORDER  
CREATESUPPLIERORDER  
DELETELOCATION  
DELETEUSER  
DELIVERY  
EDITINVENTORY  
EDITITEM  
EDITPRODUCT  
EDITSITE  
EDITUSER  
FULFILSTOREORDER  
MODIFYRECORD  
MOVEINVENTORY  
PREPARESTOREORDER  
PROCESSRETURN  
READUSER  
RECEIVESTOREORDER  
SETPERMISSION  
VIEWORDERS

# Sprint 1

Sprint 1 is designed to set up the basics and lay the groundwork for the entire system. The following requirements are included in Sprint 1:

<b>GitHub Project Repository</b> .....	6
<b>Data Access Diagram</b> .....	7
<b>Data Dictionary</b> .....	7
<b>Security</b> .....	8
<b>User Permissions</b> .....	9
<b>Dashboard</b> .....	10
<b>Login</b> .....	11
<b>Logout</b> .....	12
<b>Add User</b> .....	13
<b>Edit User</b> .....	14
<b>Delete User</b> .....	15

## GitHub Project Repository

1. Create a new **private** repository in your GitHub account (if you do not have a GitHub account or cannot remember the one you created in Dev Tools, create a new one and use that).
2. Name your new repository as follows:
  - a. **isd2023\_yourfirstnameyourlastname** (Example: **isd2023\_ChrisLondon**)
3. Invite your instructors to have access to your repository. You can do this inside your GitHub repository by going to Settings > Collaborators > Add by email

Invite your instructors by using the following addresses:

- a. [Chris.London@nbcc.ca](mailto:Chris.London@nbcc.ca)
  - b. [Aaron.Mitchell@nbcc.ca](mailto:Aaron.Mitchell@nbcc.ca)
4. This repository will house **ALL files associated with this project**, including this document and any document(s) you create, .sql files, source code, web pages, config files, notes, etc. This is your backup. “My hard drive crashed” or “My laptop died” are not excuses for losing your work. Keep your repository updated regularly. This is a work/life lesson that will serve you well in this industry.

## Data Access Diagram

Create a Data Access Diagram (ER Diagram) using the existing Database Schema. You can obtain this via the MySQL Workbench. Example of how to do this can be found here:

<https://medium.com/@tushar0618/how-to-create-er-diagram-of-a-database-in-mysql-workbench-209fbf63fd03>

Remember to upload this to your GitHub project repository.

## Data Dictionary

A data dictionary is a documented collection of data about the database and each table and field within, including names, definitions, default values, attributes, acceptable values. This is important so that you document each table so you understand the various fields, data types, sizes, etc.

**Example** data dictionary for a table called “employee”:

Table: Employee						
Field	Type	Format	Size	Description	Required?	Key?
empID	INT	999999	6+	Auto-generated unique identifier for each customer. First value starts at 100001	Y	PK
firstName	varchar(32)	Text	50	Customer First Name	Y	
lastName	varchar(32)	Text	50	Customer Last Name	Y	
startDate	date	YYYY-MM-DD		Date customer account was created. Default: Today	Y	
street	varchar(32)		50		N	
city	varchar(32)		50		N	
province	varchar(32)		2	Link to province table. Default: NB	Y	FK
postcode	varchar(32)	L9L9L9	6	Only store alphanumeric values. Display as L9L 9L9	N	
phone	varchar(32)	5065551212	9	Only store numeric values. Display as (506) 555-1212	N	
email	varchar(32)	x@x.ca	50		N	
...						



## Security

All employees using the system must have an active account and password (see employee table)

Password must be hidden on login form(s) and encrypted in DB (i.e. not stored in plain text) with AES 256 bit encryption

When viewed in the db, the password will be encrypted and NOT in plain text

## Audit Activity

Every action in the system creates a record of that activity in the Audit table, so you need to develop this aspect of the system now so you don't have to backtrack later.

Info to be tracked in each audit record includes:

- txnAuditID – unique, autonumbered ID
- txnID – id of the transaction being audited (if applicable)
- txnType – type of transaction (if applicable)
- siteID – site where activity originated
- employeeID – employee who initiated the activity
- status – status of the transaction (if applicable)
- date/time – current datetime when this record is created
- description of the activity – a clear text description of what is happening
  - Examples:
    - Add new employee record jsmith
    - Edit employee record jsmith

A trigger or an audit object that takes in the parameters above and writes them to the txnAudit table in the DB. However you do it, remember: ***it needs to be available for any transaction or activity that occurs in the system.*** Think of this as the logs, breadcrumbs required to trace activity, and a way to any potential issues or improper activity.

## User Permissions

Admin requires the ability to set and remove user permissions in the system.

Methods:

- Position-based assignment
  - Each new user created gets assigned the appropriate permissions required for the position they are assigned (see **Add User**, **Edit User**)
- Individual permissions
  - Each user can have additional permissions (beyond those assigned to their role)

**Actor(s):** ADMIN

**Permission(s):** SETPERMISSION

**Sample:** None

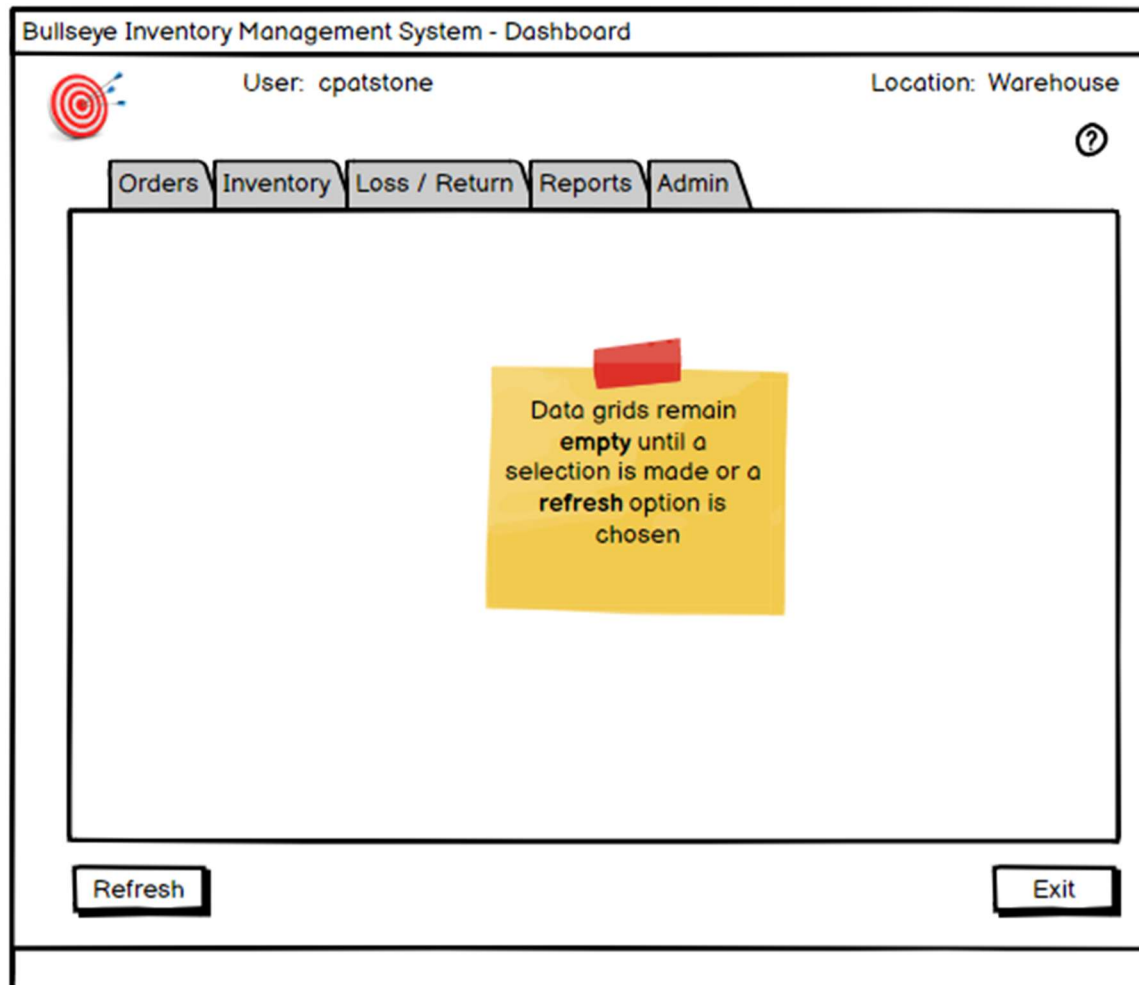
## Dashboard

A dashboard that provides access to all aspects of the system, based on who is logged in.

**Actor(s):** ALL

**Permission(s):** ALL\* (access to various areas/functionality is determined by permissions)

**Sample:**



## Login

Every Bullseye user must have:

- A valid Username and password (first initial, last name - i.e. jperez, as described in the DB)
- Ability to recover username and/or password via email (we will mimic this with a form like the one below)
- permissions to perform specific actions
- Password encrypted in DB (i.e. not in plain text) – AES 256 bit
- Password rules: minimum 8 characters, start with a letter, contains at least 1 capital letter and 1 special character

A user who logs in with the correct username and password will be presented with the dashboard and have access to the appropriate functionality.

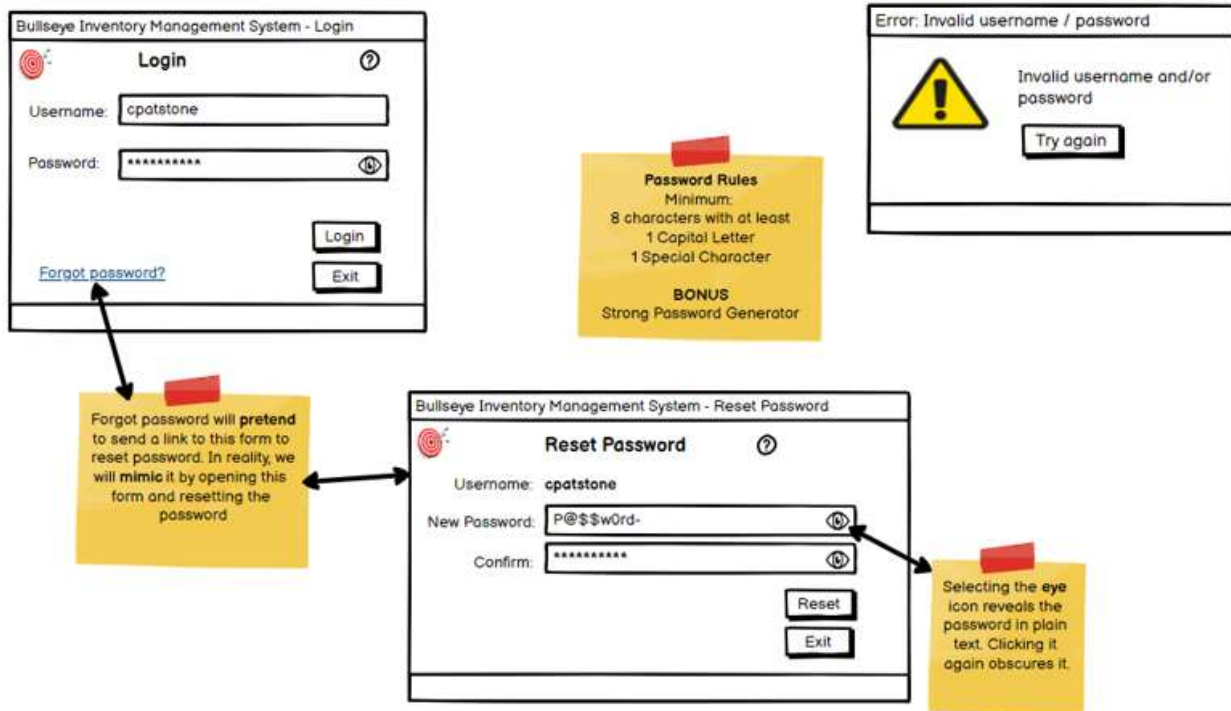
Any attempt to log in with an account that is inactive should fail (with appropriate warning)

Any attempt to log in with a username/password combination that is not accurate should fail (with appropriate warning)

**Actor(s):** ALL

**Permission(s):** None \* (all active users accounts can log in)

**Sample:**



## Logout

Every active Bullseye user can logout if they are currently logged in.

Logout methods:

1. Manual (select 'Exit' button from dashboard)
2. Automatic (after a period of inactivity, a time value which can be modified, default is 15 minutes). You can use either a config file or a database table to set this value.

**Actor(s):** ALL

**Permission(s):** None \* (all active users accounts can log out)

**Sample:** None

## Add User

Admin requires the ability to add a user to the system.

This will normally be triggered by a new user joining the company.

New user accounts are set “active” by default (the system sets their “active” status to TRUE or 1 in employee table).

**Actor(s):** ADMIN

**Permission(s):** ADDUSER

**Sample:**

The screenshot shows the 'Bullseye Inventory Management System - Add New Employee' form. The form is titled 'User: Admin' and 'Location: Admin'. It contains the following fields and controls:

- Employee ID:** A text box containing '1016'. An annotation points to it stating: 'Employee ID set automatically by system, not editable'.
- Username:** A text box containing 'flintstone'. An annotation points to it stating: 'Username and email defaults to first initial and lastname. If same already exists, add a 01 after, then 02, etc.'.
- Password:** A text box containing '\*\*\*\*\*' with a password icon. An annotation points to it stating: 'Password Rules Minimum: 8 characters with at least 1 Capital Letter 1 Special Character BONUS Strong Password Generator'.
- First Name:** A text box containing 'Fred'.
- Last Name:** A text box containing 'Flintstone'.
- Email:** A text box containing 'flintstone@bullseye.ca'.
- Position:** A dropdown menu with 'Warehouse Employee' selected. An annotation points to it stating: 'New Employees are Active by default'.
- Location:** A dropdown menu with 'Warehouse' selected.
- Active:** A checkbox labeled 'Active' which is checked.
- Buttons:** 'Save' and 'Exit' buttons.

Below the form, there are two dropdown menus:

- Store Manager:** A dropdown menu with the following options: Warehouse Manager, Warehouse Employee, Regional Manager, Financial Manager, Trucking / Delivery, Admin.
- Warehouse:** A dropdown menu with the following options: Corporate, Saint John Retail, Sussex Retail, Moncton Retail, Dieppe Retail, Oromocto Retail, Fredericton Retail, Miramichi Retail.

## Edit User

Admin requires the ability to edit a user's data in the system.

This will normally be triggered by a user changing jobs, getting married, etc.

**Actor(s):** ADMIN

**Permission(s):** EDITUSER

**Sample:**

**Bullseye Inventory Management System - Modify Employee**

User: **Admin** Location: **Admin**

Employee ID:  → Employee ID set automatically by system, not editable

Username:

Password:

First Name:

Last Name:

Email:

Position:

Location:

**Position Dropdown:**

- Store Manager
- Warehouse Manager
- Warehouse Employee
- Regional Manager
- Financial Manager
- Trucking / Delivery
- Admin

**Location Dropdown:**

- Warehouse
- Corporate
- Saint John Retail
- Sussex Retail
- Moncton Retail
- Dieppe Retail
- Oromocto Retail
- Fredericton Retail
- Miramichi Retail

## Delete User

Admin requires the ability to remove a user from the system.

This will normally be triggered by a user leaving the company.

Any user “removed” is NOT deleted, the system simply sets their “active” status to FALSE (0 in employee table).

**Actor(s):** ADMIN

**Permission(s):** DELETEUSER

**Sample:**

Bullseye Inventory Management System - User Management

User: Admin Location: Admin

### User Management

Employees Permissions

ID	Username	Password	First Name	Last Name	Email	Active	Position	Site
1	admin	*****	Admin	Admin	admin@bullseye.ca	1	9999	Corporate
1000	econcepcion	*****	Eduardo	Concepcion	econcepcion@bullseye.ca	1	1	Corporate
1001	mmunoz	*****	Monica	Munoz	mmunoz@bullseye.ca	1	2	Corporate
1002	jperez	*****	Jose	Perez	jperez@bullseye.ca	1	3	Saint John
1003	cpatstone	*****	Chris	Patstone	cpatstone@bullseye.ca	1	4	Warehouse

Refresh Edit Add New Remove Exit

Remove sets an employee Inactive in DB

Confirm Remove User?

Confirm you wish to remove user from system?

User: cpatstone  
Location: warehouse

Confirm Cancel



## Sprint 2

### Create Store Order

Create a new weekly order for a store (site).

**Actor(s):** Store Manager, Asst Store Manager, Warehouse Manager

**Features/Notes:**

- A new order can be created ONLY IF no order is currently open and active
- Only 1 NEW order of type ORDER or EMERGENCY can be active for each site (unlimited number in other statuses)
- When time to submit existing store order, should be options for both automatic (time-based) and manual option to submit

NEW Store Manager has created a new store order

SUBMITTED Store Manager has submitted the store order to the warehouse

Process:

- Store Order items should be automatically added based on reorderThreshold. Any item quantity currently below the reorderThreshold for this site should be added automatically – by case size - enough to put it over the reorderThreshold
- Store Manager has option to edit quantity of any item(s) being ordered before submitting, add any other items, remove any items, etc.( by case size)
- Can only add items which are ACTIVE = 1

siteIDTo (where the order will be delivered, i.e. the store)

siteIDFrom (should default to warehouse. Included in case of store-to-store transfers, but that feature has not been implemented yet)

shipDate (should default to be the next regular delivery date for the siteIDTo location, then may be updated to be a different shipDate)

quantity Quantities must be incremented/decremented for items by the listed case size

**txnType:** ORDER

**txnStatus:** NEW or SUBMITTED

**Permission(s):** CREATESTOREORDER

Bullseye Inventory Management System - Process Store Order

User: jperez Location: Saint John

Orders Inventory Loss/Return Reports **Store Order**

Saint John (102 items) Submitted

ID	Description	Qty	Threshold	Ordered
10000	Jetspeed FT1 Senior Ice Hockey Skates	2	6	4
10001	Super Tacks AS1 Senior Ice Hockey Skates	3	5	2
10002	Supreme 2S Pro Senior Ice Hockey Skates	2	8	6
10003	Nexus 2N Senior Ice Hockey Skates	5	9	4
10004	Jetspeed FT390 Senior Ice Hockey Skates	6	4	2
10005	Supreme 2S Senior Ice Hockey Skates	2	5	3
10006	Vapor X900 Senior Ice Hockey Skates	3	7	4
10007	Tacks 9080 Senior Ice Hockey Skates	2	14	12
10008	Supreme 2S Pro Junior Ice Hockey Skates	1	5	

Refresh Search  Add Remove Submit

Should default to Regular

Dynamic Search

## Create Emergency Order

Create a new emergency order for a store (site).

\*\*\* Can (should) likely be couple with **Create Store Order** form

**Actor(s):** Store Manager, Asst Store Manager, Warehouse Manager

### Features/Notes:

- A new emergency order can be created ONLY IF no other emergency order is currently open and active for this location/site
- Only 1 NEW order of type EMERGENCY can be active for each site (unlimited number in other statuses)
- Can only add items which are ACTIVE = 1

NEW Store Manager has created a new store order

SUBMITTED Store Manager has submitted the emergency order to the warehouse

Process:

- Store Manager can add up to a maximum of 5 separate line items

## STUDENT EDITION

To (where the order will be delivered)

siteIDFrom (should default to warehouse. Included in case of store-to-store transfers, but that feature has not been implemented yet)

shipDate (should default to be the next regular delivery date for the siteIDTo location, then may be updated to be a different shipDate)

quantity Quantities must be incremented/decremented for items by the listed case size

**txnType:** EMERGENCY

**txnStatus:** NEW or SUBMITTED

**Permission(s):** CREATESTOREORDER

Bullseye Inventory Management System - Process Store Order

User: jperez Location: Saint John

☐ Regular ☒ Emergency

Orders Inventory Loss/Return Reports **Emergency Order**

Saint John (102 items) Submitted

ID	Description	Qty	Threshold	Ordered
10000	Jetspeed FT1 Senior Ice Hockey Skates	2	6	4
10001	Super Tacks AS1 Senior Ice Hockey Skates	3	5	2
10002	Supreme 2S Pro Senior Ice Hockey Skates	2	8	6
10003	Nexus 2N Senior Ice Hockey Skates	5	9	4
10004	Jetspeed FT390 Senior Ice Hockey Skates	6	4	2
...				3
				4
				12

Refresh Search FT390 Add Remove Submit

Dynamic Search

## Receive Store Order

**Actor(s):** Warehouse Manager

**Description:** Warehouse Manager receives an order from store site

**Features/Notes:**

**txnType:** ORDER, EMERGENCY

**txnStatus:** SUBMITTED -> PROCESSING

- SUBMITTED after order has been submitted by site and before it has been sent to warehouse workers for assembly
- PROCESSING while order is being prepared by warehouse staff

**Permission(s):** RECEIVESTOREORDER

Bullseye Inventory Management System - Process Store Order

User: cpatstone Location: Warehouse

Orders Inventory Loss/Return Reports **Process Store Order**

Saint John (102 items) Submitted

ID	Description	Qty	Avail
10000	Jetspeed FT1 Senior Ice Hockey Skates	2	6
10001	Super Tacks AS1 Senior Ice Hockey Skates	3	5
10002	Supreme 2S Pro Senior Ice Hockey Skates	2	8
10003	Nexus 2N Senior Ice Hockey Skates	5	9
10004	Jetspeed FT390 Senior Ice Hockey Skates	6	4
10005	Supreme 2S Senior Ice Hockey Skates	2	5
10006	Vapor X900 Senior Ice Hockey Skates	3	7
10007	Tacks 9080 Senior Ice Hockey Skates	2	14
10008	Supreme 2S Pro Junior Ice Hockey Skates	1	5

Refresh Exit

Highlight items with less available stock than quantity ordered

## Fulfil Store Order

**Actor(s):** Warehouse Manager, Warehouse Workers

**Description:** Complete order for a store (site).

**Features/Notes:**

This is the process of warehouse workers manually checking off items

Orders should be fulfilled by making the warehouse employee check off each item as it is gathered for the order until it is complete, at which time the txnStatus = ASSEMBLED

Items added to the order should be removed from the warehouse inventory and added to the warehouse bay inventory (siteID=<ordering\_siteId>, itemLocation="2" (Warehouse Bay)

**txnType:** ORDER, EMERGENCY

**txnStatus:**

- PROCESSING while order is being prepared by warehouse workers
- ASSEMBLED once order is assembled at warehouse and ready for pickup

**Permission(s):** FULFILLSTOREORDER

## Add Item to Backorder

**Actor(s):** Warehouse Manager, System

**Description:** Add an ordered item to a backorder

**Features/Notes:**

- If no backorder exists, create when first item added for a site
- If an existing backorder already exists for this store, add items to that backorder instead of creating a new backorder
- Default shipDate is the NEXT standard delivery day for that site
- shipDate can also be modified manually by Warehouse Manager
- quantity must be incremented/decremented by case size

**txntype** = BACKORDER

**Permission(s):** CREATEBACKORDER, VIEWORDERS

## View Store Order

**Actor(s):** Warehouse Manager, ANY Store Manager, ANY Asst Store Manager

**Description:** View all outstanding store orders

### Features/Notes:

- View all outstanding store orders (those with a txnStatus NOT "CLOSED")
- Default to show orders only from user's site
- Each Store should be able to see ALL their own orders,
- Warehouse Manager should be able to see ALL orders
- Default view should be to show only orders with a txnStatus active order (i.e. txnStatus NOT CLOSED or CANCELLED), but with capability to filter to include closed or cancelled orders

**Permission(s):** VIEWORDERS

Bullseye Inventory Management System - View Store Orders

User: cpotstone Location: Warehouse

### View Store Orders

Submitted ?

Submitted

Submitted

Received  
In Progress  
Ready  
In Transit  
Delivered  
ALL

Location	Status	Items	Weight (kg)	Delivery Date
Saint John	Submitted	102	468	2023/01/09
Moncton	Submitted	114	503	2023/01/10
Dieppe	Submitted	86	370	2023/01/10
Fredericton	Submitted	95	421	2023/01/11
Oromocto	Submitted	86	345	2023/01/11
Sussex	Submitted	65	265	2023/01/12
Miramichi	Submitted	78	380	2023/01/13
...				

Refresh Exit

Next order(s) due are highlighted

Refill (refresh) data grid based on combo box selection

## View Location

**Actor(s):** ALL

**Description:** View all company sites (stores, warehouses, corporate offices, etc.)

**Features/Notes:** All Users (including any customers online): Read only

**Permission(s):** None required

## Add Location

**Actor(s):** Admin, Warehouse Manager

**Description:** Add a new site (store, warehouse, corporate office, etc.)

**Features/Notes:** Admin can add new sites as the company grows

**Permission(s):** ADDSITE

## Edit Location

**Actor(s):** Admin

**Description:** Edit info about an existing site (store, warehouse, corporate office, etc.)

**Features/Notes:**

**Permission(s):** EDITSITE



## View Inventory

**Actor(s):** ALL

**Description:** View all company inventory (stores, warehouses, corporate offices, etc.)

**Features/Notes:** All Users (including any customers online): Read only

**Permission(s):** None required

**Bullseye Inventory Management System - Inventory**

User: cpatstone Location: Warehouse

Search:

Orders Inventory Loss/Return Reports Warehouse

ID	Description	Quantity	SKU
10000	Jetspeed FT1 Senior Ice Hockey Skates	5	60000
10001	Super Tacks AS1 Senior Ice Hockey Skates	5	60001
10002	Supreme 2S Pro Senior Ice Hockey Skates	2	60002
10003	Nexus 2N Senior Ice Hockey Skates	6	60003
10004	Jetspeed FT390 Senior Ice Hockey Skates	4	60004
10005	Supreme 2S Senior Ice Hockey Skates	3	60005
10006	Vapor X900 Senior Ice Hockey Skates	5	60006
...			

Refresh Exit

**Warehouse**

- Dieppe
- Fredericton
- Miramichi
- Moncton
- Oromocto
- Saint John
- Sussex
- ALL

**Live Search**  
(updates as you type, displays **all** when no search criteria exists)

**Location defaults** to the location the user logged in is assigned to. Chris Patstone is assigned to the warehouse, so warehouse is default, but he can select other locations, or ALL

## Edit Inventory

**Actor(s):** Admin, Store Manager, Warehouse Manager

**Description:** STORE MANAGER or WAREHOUSE MANAGER can modify **ONLY** the **reorderThreshold** for inventory at their particular site only

**Features/Notes:** Store Managers or Warehouse Manager can only change the **reorderThreshold** for their own location - i.e. Store Manager of Moncton retail store can modify the **reorderThreshold** for any inventory item in the Moncton store only

Warehouse Manager can set the item ACTIVE status

**Permission(s):** EDITINVENTORY

## Move Inventory

**Actor(s):** All

**Description:** Used whenever inventory changes “place”

**Features/Notes:**

Change location of inventory (i.e. warehouse-to-order, storage to shelf, shelf-to-shelf, etc.)

Examples:

- Inventory is placed into an order (removed from the Warehouse inventory and associated with an order, siteID=<warehouse\_bay>, itemLocation = <orderid>)
- Inventory is loaded on a truck for delivery (removed from warehouse\_bay onto truck inventory, siteID=<truck>, itemLocation = <orderid>)
- Inventory is received by a store (siteID=<ordering\_site>, itemLocation = “STOREROOM”)
- Inventory is moved within a store/warehouse (example: from a storeroom or shelf/aisle to another specific shelf or aisle, siteID=<site>, itemLocation = <new shelf in store>)

**Included as part of the following Use Cases:** Prepare Store Order, Fulfil Store Order, Pickup Store Order, Transport Store Order, Deliver Store Order, Accept Store Order

**Permission(s):** MOVEINVENTORY

## **Sprint 3**

Pickup Store Order

Transport Store Order

Deliver Store Order

Place Online Order

Prepare Online Order

Receive Online Order

Accept Store Order

Modify Record \*

## **Sprint 4**

Create Loss

Process Return

Add New Product

Edit Product

Create Supplier Order

Create Reports

Create Message

Receive Message

## Transaction Status State Chart

