

Bullseye Sporting Goods
Inventory Management System Analysis

Prepared by

OOAD Team #007

Brady Walsh, Mike McDonald, Bailey Peters

April 11th, 2022

Contents

Introduction.....	3
Interview report	6
Current System	7
Use Case 1 Create Store Order.....	7
Create Store Order - State Chart Diagram	Error! Bookmark not defined.
Use Case 2 Prepare Store Order	12
Use Case 2 Prepare Store Order Activity Diagram	Error! Bookmark not defined.
Use Case 3 Deliver Store Order	15
Use Case 4 Create Reports	17
Executive Summary	Error! Bookmark not defined.

Introduction

The purpose of this document is to provide a detailed analysis of Bullseye Sporting Goods' current inventory management system, procedures, and practices. Team #7 was approached about developing a new system to replace the current system in use by the company. Our first step in the development of this new system is to understand how the current system functions in its present state, such is the intent of this document. This analysis will not provide any suggestions for improvement of the current system, nor make any recommendations regarding a new system implementation.

The information and data used in this analysis was obtained through several interviews with various members of the company, including Eduardo Concepcion (Regional Manager), Monica Munoz (Finance Manager), Chris Patstone (Warehouse Foreman), and Jose Perez (Store Manager). Our initial assessment was primarily conducted with the Saint John store and warehouse as a focal point for understanding how the current system functions, with relations to the other 7 locations in the Atlantic region. Group #7 would like to thank the members of Bullseye Sporting Goods for their contributions to this portion of the development process.

Executive Summary

This summary serves as a high-level overview of the documents in this Project. This summary will outline some issues discovered during the information gathering process of the analysis. During interviews with key members of staff the analysis team outlined the following logistical, administrative, and software issues.

During the interviews, stakeholders expressed a desire to implement a new inventory management system, to replace the current system in use by pre-existing storefront's, newly acquired locations, and the regional inventory warehouse. It was stated that the current system is not conducive to their needs on the ground-level and does not promote synchronicity between an expanded network of retail locations. The goal is to design, develop and implement a new system for BullsEye to handle the new influx of orders after the recent acquisition of seven new retail storefronts. The information contained within this document will focus solely on the analysis of the current system.

Our initial assessment involved mapping out the current system(s) in use between stores and the regional warehouse. Through our research, we were able to map out the communication between the parties involved with managing orders both sent and received by the regional warehouse, as well as how these orders are handled once they are sent or received by the storefront. We've developed a series of 12 use cases for the current system, including state chart diagrams, which demonstrate the varying degrees of cooperation required between the current systems and facilities in use by the company. Below is an overview of some current areas of opportunity within the current system, from logistics to software issues.

Logistical Issues

- During Interview #1 it was made clear through verbal communication that Dispatches are sent individually based on completion of each order, instead of a regulated schedule.
- During interview #1, Chris Patstone explained how emergency orders are handled on a case-by-case basis, with no uniform process for communicating these orders between the store and the warehouse. It was described that the lack of structured procedure ends up in redundant orders caused by lack of communication and ambiguity due to orders being completed by verbal communication in most instances.
- Currently, all orders are received and processed manually by the warehouse staff. Chris Patstone stated during interview #1 that this causes backorders not being communicated to stores, leading in an increase in shipping expenses when the stores inevitably place a new order to request the backordered items.

Administrative Issues

- During Interview #1 it was made clear through verbal communication that there is currently no standardized process for processing returns at the regional level.
- During interview #1, it was explained Chris Patstone that orders are still placed using print documentation, creating a risk of important payroll documents being discarded, causing delays in payment between BullsEye Sporting goods and third-party shipping companies involved with BullsEye.
- Currently, the warehouse manager is required to manually complete weekly summaries on warehouse activity in a text format, leading to productive time lost by the warehouse manager.

Software Issues

- During Interview #1 it was made clear through verbal communication that Store Managers don't have access to warehouse inventory data, a centralized inventory system, or a standardized system at the store level
- During interview #1, Eduardo Concepcion, Regional Manager, explained that currently, some stores are operating on an independent inventory management system, inherited from the previous owners of these locations. This leads miscommunication and a less synchronized network as not all uniquely identifiable product codes are uniform in their naming conventions.

Interview report

Speaking with the management of the company and stores, we were able to establish a broad baseline to work with. The systems used by the stores across the company are not universal, meaning stock levels cannot be seen by anyone outside the store, leading to confusion about stock levels and orders. Because of the lack of systems across the company, order placements are inconsistent, being placed on different days from other stores in the area, creating more costly deliveries than necessary. The use of “emergency orders” by the stores also creates more deliveries than necessary, as backorders are not communicated back to the stores.

After our interview with Chris Patstone, we gained an understanding of the existing system used by Bullseye’s warehouse. The system is standalone, much like the individual stores, and orders must be manually entered into it for processing. Because of the manual entry and no connected systems to the stores, store managers are unable to see the warehouse’s inventory when making orders, leading to backorders being made to complete the orders. However, the backorders are not communicated to the stores, leading the stores to think their orders were not completed, them placing another order to fill the stock they requested, and added stress for delivery drivers when stores get angry about the incomplete orders.

The miscommunication and lack of a connected system are causing most of the issues for Bullseye, creating unnecessary stress, confusion, and costs for the company. With a system that is synchronized across the entire company, stock levels can be accurately adjusted and compared, orders and backorders can be recorded and communicated to the stores, and unnecessary delays, such as counting orders before and after deliveries, can be avoided. A complete system will create a streamlined and transparent process for Bullseye to follow and maintain.

Current System

Use Case Glossary

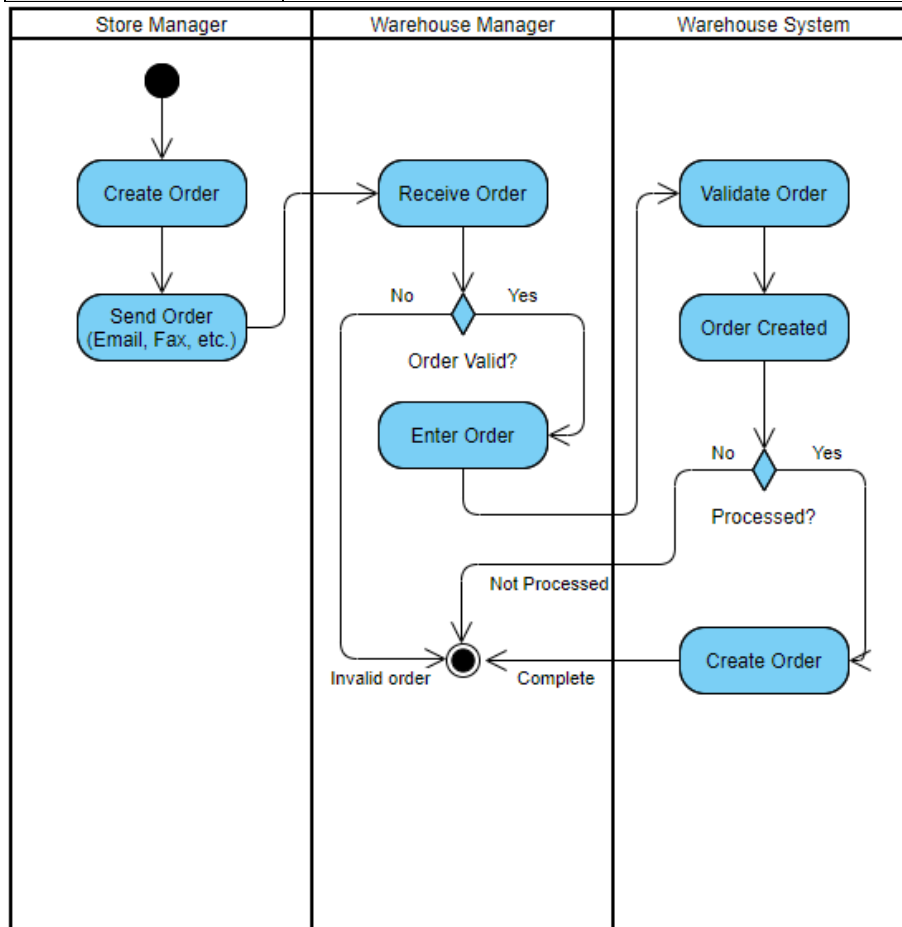
ID	Use Case Name	Primary Actor	Other Actors	Description
UC-001	Create Store Order	Store Manager	Warehouse Manager	The Store Manager places an order into the inventory management system which is then authenticated automatically and verified by the Warehouse Manager
UC-002	Prepare Store Order	Warehouse Manager	Store Manager	The Warehouse Manager uses the Manifest Produced by the Inventory Management System to prepare the order for delivery
UC-003	Deliver Store Order	Driver	Store Manager	The Driver delivers the specific goods requested by the store manager at the approved time
UC-004	Create Reports	Regional Manager	Store Manager	Store Manager spends time generating weekly reports
UC-005	Update Store Inventory	Store Manager	Warehouse Manager	The Store Manager manually enters information about the inventory of their store
UC-006	Verify Order	Driver	Warehouse Manager	Cross Reference Manifest and BOL before leaving DC
UC-007	Create Emergency Order	Store Manager	Warehouse Manager	Create stock order for store with maximum priority (all other orders on hold)
UC-008	Create Vendor order	Warehouse Manager	Vendor	Create order based on inventory requests aggregate for bulk
UC-009	Create Back Order	Warehouse Manager		When stock is not available, backlog order created, sequential priority (first come first serve)

UC-010	Return Stock	Store Manager		Store Manager Determines if stock can be resold, if not, stock is disposed of at the store level
UC-011	Add Backorder Stock to Next Order	Warehouse Manager	Store Manager	Store Managers can't see this data at this time, but back orders in the current state are added to weekly orders as they become available
UC-012	Order from Other Store	Store Manager	Other Store Manager	Through Phone Communication, one store manager requests a courier delivery of stock from another store. This process is not tracked in the IMS

Use Case 1 Create Store Order

Use Case ID	UC-001	Version			
Use Case Name	Create Store Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Medium		
Priority	High	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Store Manager				
Primary Business Actor	Store Manager				
Other Participating Actors	Warehouse Manager				
Other Interested Stakeholders	Regional Manager				
Description	The Process taken when a store manager requests more inventory				
Precondition(s)	The Store Manager creates an order, sent to the warehouse				
Trigger(s)	Order info arriving at Warehouse system				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Store Manager: Creates Order Warehouse Manager: Validate Order Warehouse Manager: Process Order	Order Verified: Valid Inventory order Verify Order: Valid Inventory order Info stored in system: Inventory order			
Alternate Flow	Actor Action, System/Result, Info required				
	Store Manager: submits Invalid order -- Order Not verified: Valid Inventory order Warehouse Manager: Verify order – Order cannot be verified Warehouse Manager: Process Order -- Order cannot be processed				
Conclusion	Store Order Created				
Post condition	Valid Inventory order, stored locally				
Business Rules	Must use existing system to Create Order.				

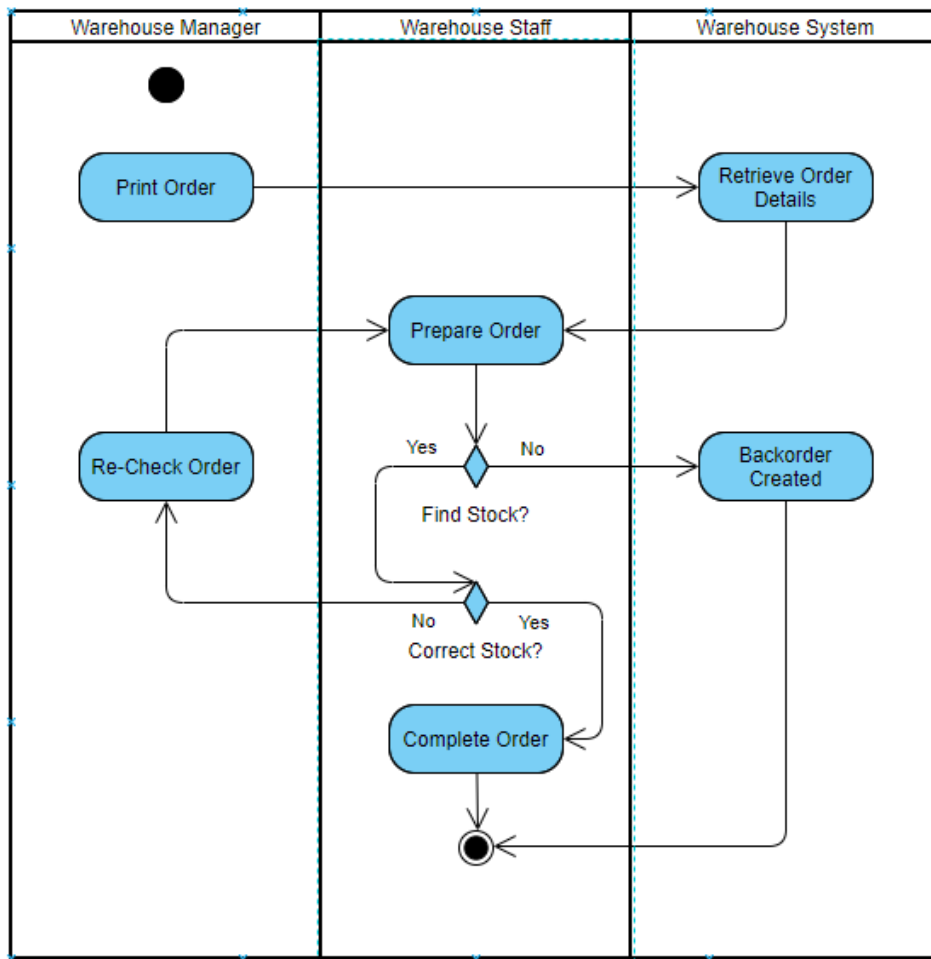
Implementation Constraints and Specifications	Both store and Warehouse are operational and have communications
Assumptions	The Store and Warehouse Managers both understand how to make and verify an order
Notes / Open Issues	



Use Case 2 Prepare Store Order

Use Case ID	UC-002	Version			
Use Case Name	Prepare Store Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low		
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Warehouse Manager/Staff				
Primary Business Actor	Warehouse Manager				
Other Participating Actors	Warehouse Staff, Store Manager				
Other Interested Stakeholders	Regional Manager, Delivery Company				
Description	The Process of preparing the order for delivery at the Warehouse				
Precondition(s)	Available stock, Ability to move product, Valid Emergency or Weekly Store Order				
Trigger(s)	Valid Store Order in system				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Warehouse Manager: creates docs Warehouse Staff: Prepare Order Warehouse Manager: check order	BOL, Manifest created: Valid Order Prepared Order: Manifest Checked order: Manifest			
Alternate Flow	Actor Action, System/Result, Info required				
	Warehouse Staff: Cannot find stock -- Order placed on backorder Warehouse Manager: incorrect stock – Order rechecked				
Conclusion	Delivery Driver picks up delivery				
Post condition	Delivery is ready to be loaded, and is the correct assortment				
Business Rules	Deliveries must be checked by Store Manager				

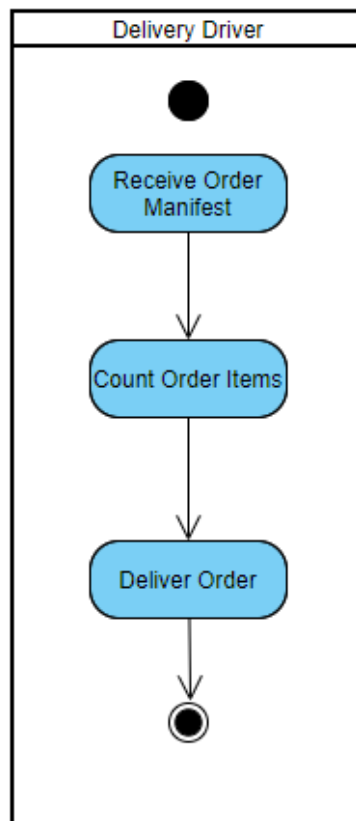
Implementation Constraints and Specifications	Warehouse Manager and Staff require correct documentation (Manifest) to verify order.
Assumptions	Warehouse Manager and Warehouse Staff understand the system in place
Notes / Open Issues	Lacking Digital verification, the Warehouse Manager needs to check each order manually, costing the company time and resources.



Use Case 3 Deliver Store Order

Use Case ID	UC-003	Version	
Use Case Name	Deliver Store Order	Date	2022/03/28
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low
Priority	High	<div>Use Case Type:</div> <div><input type="checkbox"/> Business Requirement</div> <div><input checked="" type="checkbox"/> System Analysis</div>	
Source	Delivery Driver		
Primary Business Actor	Warehouse Manager		
Other Participating Actors	Store Manager, 3 rd Party dispatch		
Other Interested Stakeholders	Regional Manager, Store Manger		
Description	A valid Sore Order is delivered to the predetermined store (on schedule)		
Precondition(s)	The Driver has required information (ie. Addresses, timeline, documentation)		
Trigger(s)	Driver is given dispatch to deliver order		
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required	
	Delivery Driver: Checks Order Delivery Driver: Delivers Order	Checked Order: stock and Manifest Delivered Order: Time and Address	
Alternate Flow	Actor Action, System/Result, Info required		
	Store Orders only Use Case Flow depends on valid orders and pre-determined timeline, there is no alternate path as verification is done before the driver arrives.		
Conclusion	Store Receives Order		
Post condition	Order is Correct and on time		

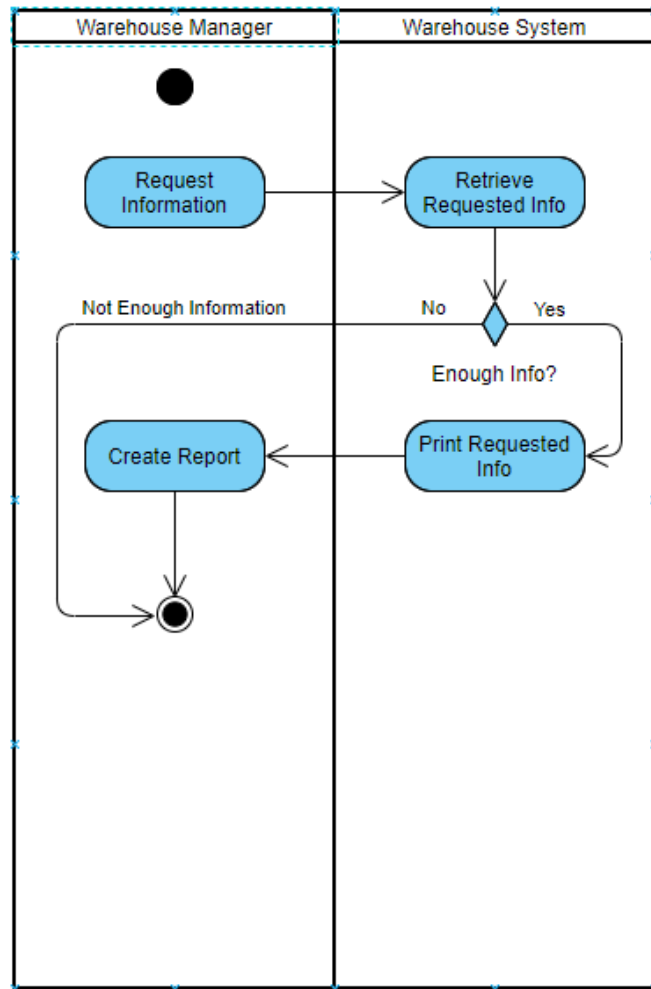
Business Rules	Driver must arrive and depart both locations at pre-determined time set by 3 rd party dispatch
Implementation Constraints and Specifications	Weekly Orders can only be delivered on that store's specified delivery date
Assumptions	The Delivery Driver can deliver orders and understands the task at hand.
Notes / Open Issues	How Many deliveries can one driver do in a day?



Use Case 4 Create Reports

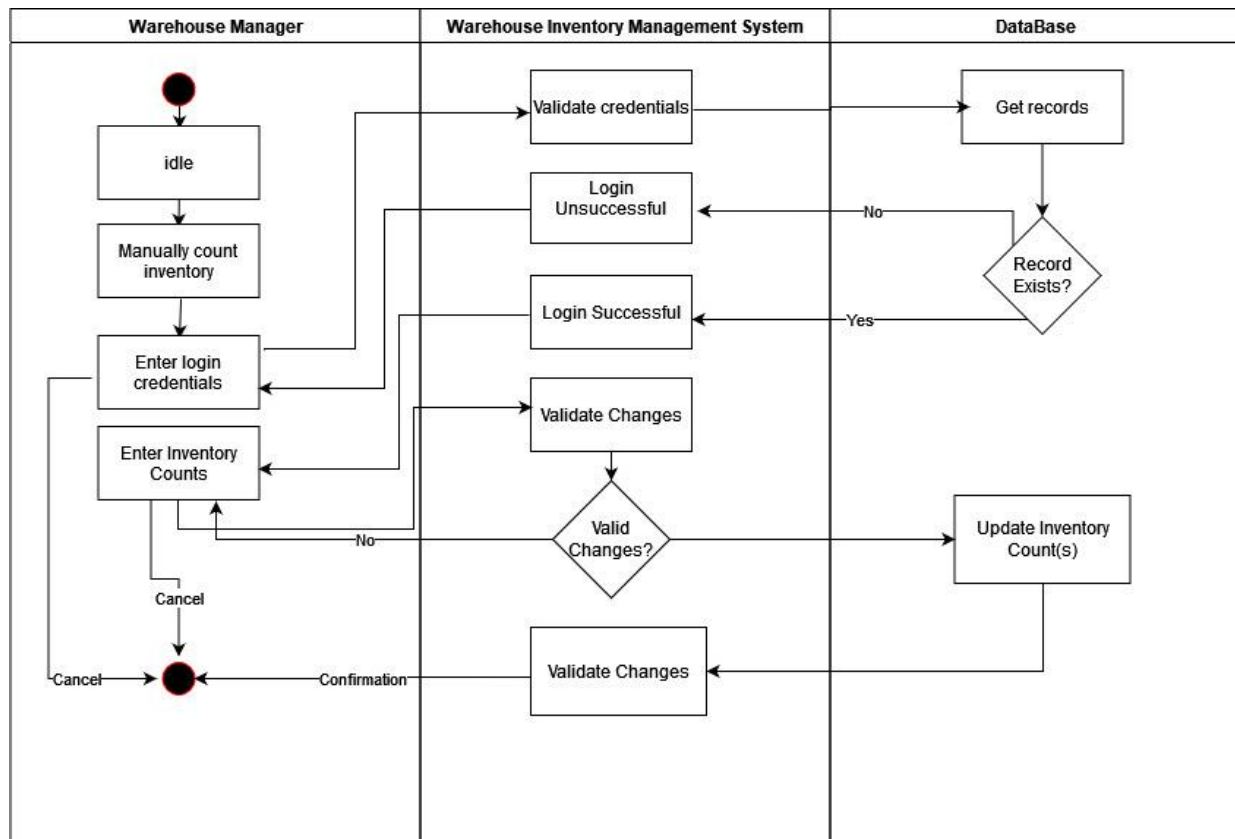
Use Case ID	UC-004	Version	
Use Case Name	Create Reports	Date	2022/03/28
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Medium
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis	
Source	Warehouse Manager		
Primary Business Actor	Regional Manager		
Other Participating Actors	N/A		
Other Interested Stakeholders	Store Manager		
Description	Weekly, the Warehouse Manager Submits reports on all activity done by the Warehouse that week, this is done manually		
Precondition(s)	Warehouse Manager has access to Text Editor		
Trigger(s)	7 days since last report		
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required	
	Warehouse Manager: Makes Report	Report Made: Warehouse Info	
Alternate Flow	Actor Action, System/Result, Info required		
	Warehouse Manager: Fails to Make Report -- Report not Made: Lack of Info		
Conclusion	Report Made and Submitted to Regional Manager		
Post condition	A complete and legible report with all the necessary info		

Business Rules	Reports must be submitted weekly
Implementation Constraints and Specifications	Currently Done Manually
Assumptions	Warehouse Manager knows how to use Word
Notes / Open Issues	This Process can be automated with proper investment



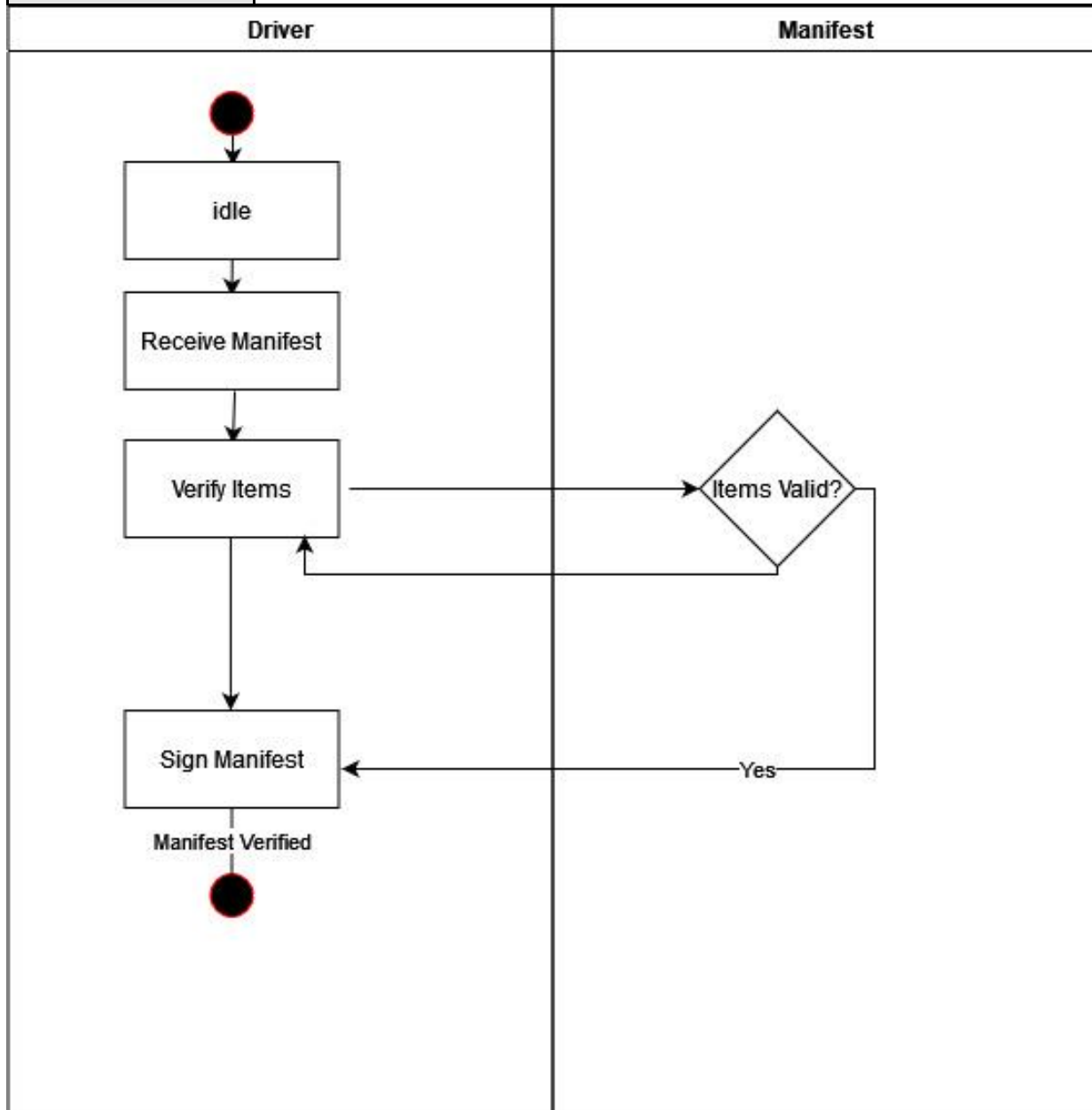
Use Case ID	UC-005	Version	
Use Case Name	Update Store Inventory	Date	2022/03/28
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	High
Priority	Medium	Use Case Type:	
Source	Store Manager		

Primary Business Actor	Store Manager	<input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis
Other Participating Actors	Warehouse Manager	
Other Interested Stakeholders	Regional Manager	
Description	Store Manager counts and updates their current inventory	
Precondition(s)	Store has Inventory	
Trigger(s)	Store Manager deems it needed	
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required
	Store Manager: Counts Inventory Store Manager: Updates Inventory	Inventory Counted: Inventory Data Inventory Updated: Inventory Count
Alternate Flow	Actor Action, System/Result, Info required	
	Store Manager: Forgets how to count – Inventory not Counted	
Conclusion	Updated Inventory Data	
Post condition	Inventory Management System (Updated)	
Business Rules	Updates performed on a “as needed” basis	
Implementation Constraints and Specifications	Access to inventory management system required Valid User must enter data	
Assumptions	Store Manager knows how to count, Store Manager understands how to update info in system	
Notes / Open Issues	Can be done automatically	



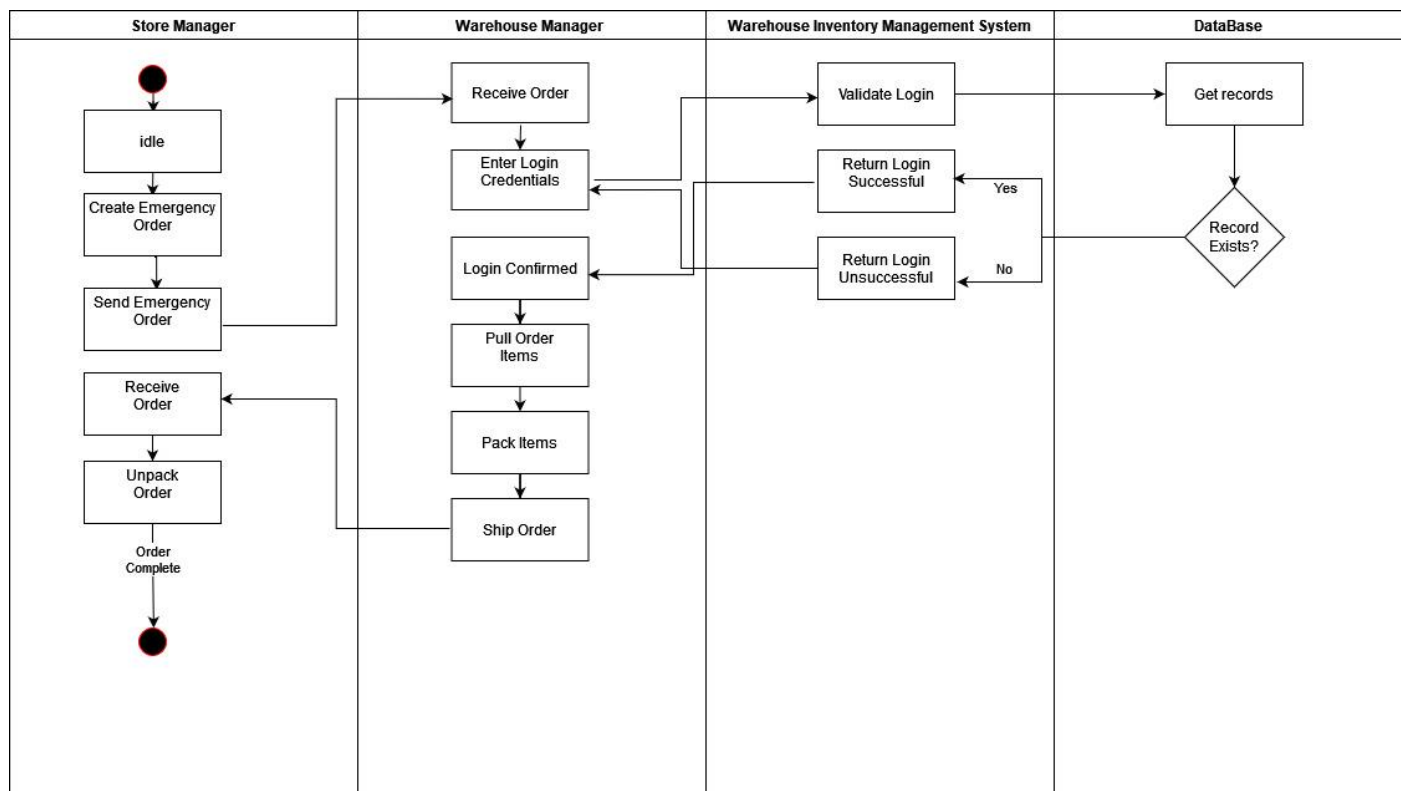
Use Case ID	UC-006	Version			
Use Case Name	Verify Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low		
Priority	Low	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Deliver Driver				
Primary Business Actor	Delivery Driver				
Other Participating Actors	Warehouse Manager, Store Manager				
Other Interested Stakeholders	Regional Manager				
Description	The Delivery Driver Acquires Manifest and assures that it is accurate based on the order that he/she is picking up				
Precondition(s)	Manifest Made, Order Prepared, Dispatch Created				
Trigger(s)	Dispatch Created				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Driver: Verifies Order	Order Verified: Manifest			
Alternate Flow	Actor Action, System/Result, Info required				
	Driver: Cannot Verify Order -- Manifest Required (not available)				
Conclusion	Verified Order				
Post condition	An order that is verified				

Business Rules	Drivers must verify order before loading
Implementation Constraints and Specifications	Time made available to the driver as well as dock space to check order
Assumptions	Driver understands how to check order
Notes / Open Issues	Drivers don't always check the order before loading.



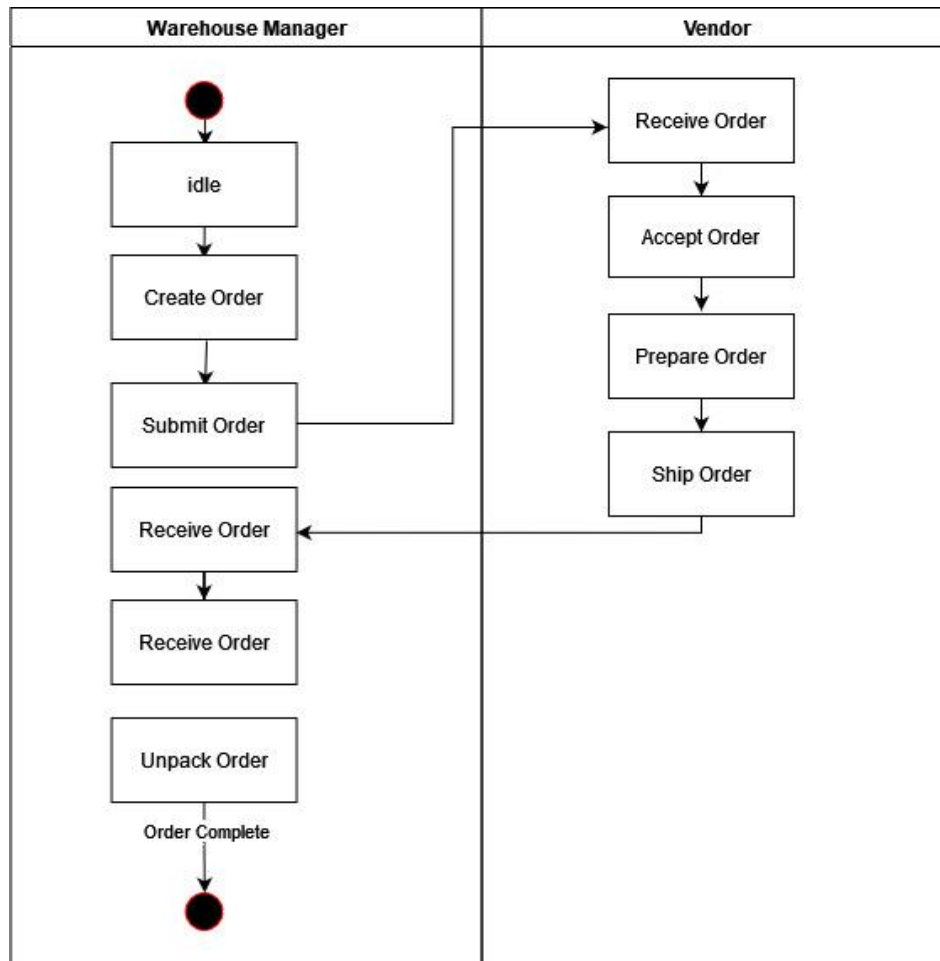
Use Case ID	UC-007	Version			
Use Case Name	Create Emergency Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Medium		
Priority	High	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Store Manager				
Primary Business Actor	Store Manager				
Other Participating Actors	Warehouse Manager				
Other Interested Stakeholders	Regional Manager				
Description	An order is created without a specified delivery date and given rush priority				
Precondition(s)	Store does not have any available stock for specific items, warehouse has item in stock				
Trigger(s)	Item is out of stock at store level				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Store Manager: Creates Emergency Order Warehouse Manager: Sends Order	Emergency Order made: Valid Order Emergency Order sent: In Stock			
Alternate Flow	Actor Action, System/Result, Info required				
	Store Manager: Creates Order -- Item not in Warehouse Stock: Not in Stock				
Conclusion	Emergency order made				
Post condition	A valid order with rush priority				

Business Rules	Store Manager is the only one that can make emergency orders on behalf of store
Implementation Constraints and Specifications	Deliveries made during business hours, Emergency orders with Rush Priority are processed before all other orders
Assumptions	The Store Needs a product that is not in stock, The Warehouse has product in stock
Notes / Open Issues	More Constraints should be put on emergency orders



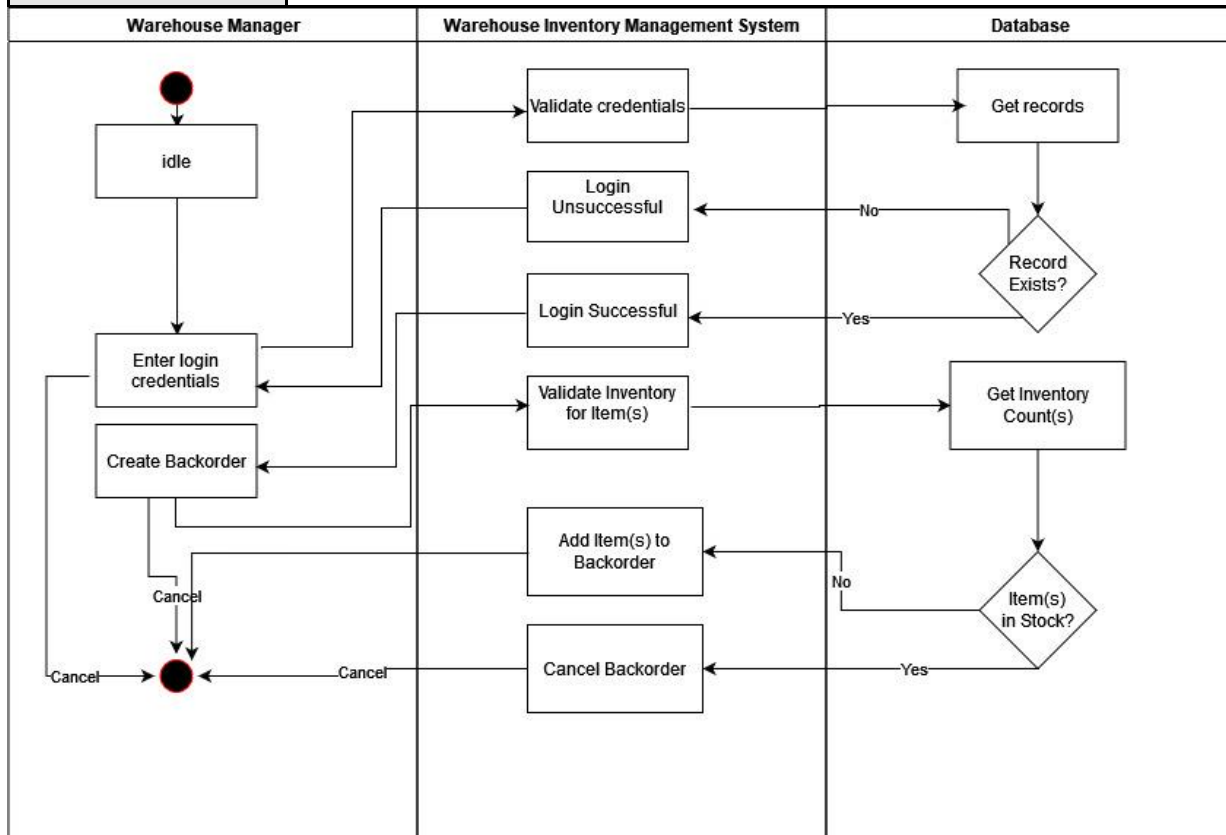
Use Case ID	UC-008	Version			
Use Case Name	Create Vendor Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Medium		
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Warehouse Manager				
Primary Business Actor	Warehouse Manager				
Other Participating Actors	Vendor				
Other Interested Stakeholders	Store Manager, Regional Manager				
Description	An order is made for the warehouse to receive a product from a vendor				
Precondition(s)	Valid Vendor, Valid order				
Trigger(s)	Warehouse runs out of stock of specific items				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Warehouse Manager: Creates order	Order Created: item Out of Stock			
Alternate Flow	Actor Action, System/Result, Info required				
	N/A				
Conclusion	Vendor Order Created				
Post condition	Valid Order				
Business Rules	Reasonable discretion encouraged on behalf of Store Manager				

Implementation Constraints and Specifications	Vendor Must have item in stock, Vendor must accept an order from the Warehouse Manager, Warehouse must be Out of Stock of specified item
Assumptions	Warehouse Manager can determine if something is out of stock, The Warehouse Manager can contact the vendor
Notes / Open Issues	This Process can be automated



Use Case ID	UC-009	Version			
Use Case Name	Create Back Order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Medium		
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Warehouse Manager				
Primary Business Actor	Warehouse Manager				
Other Participating Actors	N/A				
Other Interested Stakeholders	Regional Manager				
Description	When an item is out of stock but part of a store order, the warehouse manager creates a backorder of that item				
Precondition(s)	Item is out of stock at warehouse level				
Trigger(s)	Store Level order is made while item is out of stock at warehouse level				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Warehouse Manager: Creates Backorder	Backorder Created: info on Out-of-Stock Items			
Alternate Flow	Actor Action, System/Result, Info required				
	Warehouse Manager: does not Create Backorder -- Item already in stock				
Conclusion	Backorder Created				
Post condition	Specified items are out of stock at warehouse level				
Business Rules	Only the Warehouse Manager can create back orders				

Implementation Constraints and Specifications	Back orders can only be created if the stock is not available at both the store and warehouse level
Assumptions	Warehouse Manager capable of determining the availability of products in the warehouse
Notes / Open Issues	This Process can be automated



Use Case ID	UC-0010	Version			
Use Case Name	Return Stock	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low		
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Store Manager				
Primary Business Actor	Store Manager				
Other Participating Actors	N/A				
Other Interested Stakeholders	N/A				
Description	Store Manager assesses value and determines outcome for return process				
Precondition(s)	Store Manager receives return from customer				
Trigger(s)	Customer Requests Return				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Store Manager: Returns Product	Product on shelf: Manager Deemed product fit for resale Product Disposed of: Manager deemed product not fir for sale			
Alternate Flow	Actor Action, System/Result, Info required				
	N/A				
Conclusion	Refund given at point of sale; warehouse does not see returned product.				
Post condition	Valid Return requested be customer				
Business Rules	All returns are deemed valid by the Store Manager at their discretion				

Implementation Constraints and Specifications	No Constraints beyond Store Manager discretion
Assumptions	Store Manager has good discretion and knows how to refund a customer
Notes / Open Issues	Should be done at warehouse level and logged in the correct database

Use Case ID	UC-0011	Version			
Use Case Name	Add Back Order, to next order	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low		
Priority	Medium	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Warehouse Manger				
Primary Business Actor	Warehouse Manager				
Other Participating Actors	Store Manager				
Other Interested Stakeholders	Regional Manager				
Description	The Warehouse Manager adds stock from back order to next weekly order for store				
Precondition(s)	Store has weekly order and back order on file				
Trigger(s)	Back order is on file when weekly order is triggered				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Warehouse Manager: adds stock	Stock Added: backorder			
Alternate Flow	Actor Action, System/Result, Info required				
	Warehouse manager: does not add stock -- Stock Not Added: no backorder				
Conclusion	Warehouse Manager has added the stock onto the existing order				
Post condition	1 complete order ready for delivery				
Business Rules	Only the Warehouse Manager can add back orders to regular orders				

Implementation Constraints and Specifications	Only done when both backorder and regular order on file.
Assumptions	Backorder is valid and will not overload the Delivery Truck
Notes / Open Issues	N/A

Use Case ID	UC-0012	Version			
Use Case Name	Order From Another Store	Date	2022/03/28		
Author	Brady Walsh, Bailey Peters, Mike McDonald	Complexity	Low		
Priority	Low	Use Case Type: <input type="checkbox"/> Business Requirement <input checked="" type="checkbox"/> System Analysis			
Source	Store Manager #1				
Primary Business Actor	Store Manager #2				
Other Participating Actors	Store Manger #2				
Other Interested Stakeholders	Regional Manager				
Description	A Store Manager calls another store manager to request stock from their inventory				
Precondition(s)	Store Manager #2 has the item(s) Store Manager #1 needs				
Trigger(s)	Store Manger #1 needs an Item that's out of stock				
Basic Flow (Typical Course of Events)	Actor Action,	System/Result, Info required			
	Store Manager #1: calls SM#2 Store Manager #2: Sends SM#1 product	Call made: correct Phone Number Product Sent Via Carrier: valid stock and info.			
Alternate Flow	Actor Action, System/Result, Info required				
	Store Manager #2: does not have stock/resources to send stock – No Order Sent				
Conclusion	Store Manager #1 asks Store Manager #2 for item(s)				
Post condition	Items sent via courier to Store Manger #1's location				
Business Rules	N/A				

Implementation Constraints and Specifications	N/A
Assumptions	Both Store Managers have each other's phone number
Notes / Open Issues	This is a bad way to manage store inventory

State Diagram (Ordering Process)

