

# CPSC 350 – Final Project

## Drop Ship Inc.

William Dzubak (Project Manager), Avery Logue,  
Anh Tran, Joseph Griffith, and Caleb Mathew

## Table of Contents

Initiation Deliverables.....	4
Team Roster .....	4
Team Expectation Document.....	5
Project Initiation Memo .....	6
Phase 1 Deliverables .....	7
System Vision Document .....	7
Initial Subsystem Breakdown .....	8
Initial Financial Cost/Benefit Analysis .....	9
Project Risk and Feasibility Document.....	12
Project Approval Document .....	13
Phase 2 Deliverables .....	14
Project Environment Description.....	14
Work Breakdown Structure .....	16
Project GANTT Chart.....	17
Project Staff/Resource Allocation Plan .....	18
Project Evaluation and Monitoring Plan.....	18
Phase 3 Deliverables .....	19
Information Gathering Plan .....	19
Stakeholder Identification and Classification .....	35
Requirements Definition.....	37
Use Cases .....	39
Overview.....	39
Abbreviation Guide.....	39
Tabular Use Case Definitions – Client Information/Business Operations Subsystem .....	40
Graphical Use Case Diagram – Client Information/Business Operations Subsystem.....	41
Fully Developed Use Case Descriptions – Client Information/Business Operations Subsystem .....	42
Tabular Use Case Definitions – Warehouse Returns Subsystem .....	99
Graphical Use Case Diagram – Warehouse Returns Subsystem .....	99
Fully Developed Use Case Descriptions – Warehouse Returns Subsystem.....	100
Tabular Use Case Definitions – Warehouse Floor Subsystem .....	116
Graphical Use Case Diagram – Warehouse Floor Subsystem.....	117

Fully Developed Use Case Descriptions – Warehouse Floor Subsystem.....	118
Tabular Use Case Definitions – Warehouse Shipping Subsystem .....	148
Graphical Use Case Definitions – Warehouse Shipping Subsystem.....	148
Fully Developed Use Case Descriptions – Warehouse Shipping Subsystem.....	149
Domain Classes.....	164
Team Retrospective .....	167
Additional Deliverables (Appendices).....	169
Information Gathering Results.....	169
Relevant Communications Received .....	175
Relevant Documentation Received.....	175
External Information .....	176
Anything Else.....	176

# **Initiation Deliverables**

## **Team Roster**

William Dzubak	571-969-9763	william.dzubak.17@cnu.edu
Avery Logue	703-887-0248	avery.logue.17@cnu.edu
Anh Tran	757-903-6644	anh.tran.19@cnu.edu
Joseph Griffith	804-690-8370	joseph.griffith.19@cnu.edu
Caleb Mathew	571-292-7210	caleb.mathew.17@cnu.edu

# **Team Expectation Document**

Communication is the top priority of this group. It is expected that group members will reply to text messages pertaining to the project within a day, and that the Project Manager will be notified of any absences from meetings. In the event that one member has a problem with the workload or distribution of work, that member will talk to either the Project Manager or Professor Kreider.

Signed,

WD, AL, AT, JG, CM

# **Project Initiation Memo**

To: Pro. Christopher Kreider

From: Righteous Row - William Dzubak(Project manager), Avery Logue, Anh Tran, Joseph Griffith, and Caleb Mathew

Date: December 5, 2019

Subject: Drop Ship Inc: New information system

## **Introduction:**

The purpose of this project is to develop a new unified information system for Drop Ship Inc to handle their day to day business. The problem Drop Ship Inc having right now is they have three large and separate information systems that do not interact with each other, including the warehouse/inventory management system, the order management system, and the shipping management system. The new system needs to help Drop Ship Inc solve their problems by creating the connection among the warehouse/inventory management system, the order management system, and the shipping management system so they can handle their day to day business and repair for their future growing business as well.

## **Responsibilities:**

Project Manager – William Dzubak

- Develop Schedule
- Recruit and train team members
- Assign work to team members
- Monitor and control project deliverables
- Report project status
- Work directly with team members and client (Professor)
- Identify resources when needed

Project team members – Avery Logue, Anh Tran, Joseph Griffith, Caleb Mathew

- Follow project manager instructions
- Discuss any issues related to the project with team members and project manager as soon as possible

# **Phase 1 Deliverables**

## **System Vision Document**

### *Problem Description*

Drop Ship Inc, needs a new information system to better manage their day to day business activities. There are three main problems lie within communication between the different subsystems: the Warehouse System does not communicate with the Order System, meaning that employees are left guessing how many items they have left in inventory. Furthermore, the Shipping System does not communicate to the Warehouse System, which leaves the employees guessing again about what size boxes they need to ship each order. Additionally, the Customer Management System does not communicate with the Order System, which means that there is a large manual component to tracking customer information.

### *System Capabilities*

The main objective that this new system will fulfill is ensuring seamless interdepartmental communication. This will allow for the Warehouse System to effectively communicate with the Order System by interacting with it and notifying the warehouse of each order so that employees know how much of each item they have left in their inventory. Additionally, the system will incorporate an improved order management functionality, which allows managers to pull reports of past shipping, warehousing, packaging, and return orders. This will ensure that the client's information is accurately and automatically stored in a database. The new system will also communicate product name, weight, and dimensions between the Shipping and Warehouse Systems, so that employees will not have to guess what size box they need to pack the product in. It will also crack down on employee theft by requiring management approval before submitting a return or disposal worth more than \$50.

### *Business Benefits*

We anticipate that the new system will improve the speed at which new clients are brought in, along with helping to improve the speed at which the orders are processed and carried out. The shipping process itself will also be improved due to effective intersystem communication of product locations, weights, and dimensions.

# **Initial Subsystem Breakdown**

## *Initial Time and Cost Estimate*

<b>Subsystems</b>	<b>Functional Requirements</b>	<b>Iterations Required</b>	<b>Estimated Time</b>	<b>Estimated Cost</b>
Client Information/Business Operations System	19	6	23 Weeks	\$355,733
Warehouse Floor Information System	10	6	17 Weeks	\$262,933
Warehouse Shipping Information System	5	6	10 Weeks	\$154,667
Warehouse Returns Information System	5	6	10 Weeks	\$154,667
<b>Totals</b>	<b>39</b>	<b>24</b>	<b>60 Weeks</b>	<b>\$928,000</b>

# Initial Financial Cost/Benefit Analysis

## *Estimated Development Costs*

Expense Category	Amount
Wages & Benefits (1 PM, 6 Analysts)	\$734,000
Equipment	\$20,000
Facilities	\$54,000
Utilities	\$70,000
Travel	\$70,000
<b>Total Expenses</b>	<b>\$928,000</b>

### **Wages**

Given roughly 60 weeks from when the project was assigned until it will be completed before Black Friday of 2020, we estimate the total salary expense for the project to be \$650,000. Hiring six developers for 60 weeks at a weekly salary of \$1270 gives us \$76,200 per developer, meaning \$457,200 total. In addition to this, the project manager will have weekly and total salaries of \$1,420 and \$85,200, respectively, giving us a total wage cost (not including benefits) of \$542,400.

Benefits are estimated at \$11.38 per hour, per employee, giving us \$455.20 per employee, per week. This means \$27,312 in benefits per employee for the duration of the project, so \$191,148 in benefits total. Adding the wages to the benefits gives us a total cost of \$733,548. This figure is rounded up to \$734,000 in the table for simplicity's sake.

### **Equipment**

We will be purchasing computers and other necessary equipment for each member of the team at \$2,000 per employee and \$14,000 total. Additionally, we expect miscellaneous equipment costs to total around \$6,000, bringing the total equipment cost to \$20,000.

### **Facilities**

Office space in a large city costs around \$3 per square foot per month. Assuming an office of 1300 square feet, the cost of rent per month would be \$3900, giving us \$54,000 for 60 weeks of rent.

### **Utilities**

The utility cost is expected to be around \$70,000, which will cover all the programs needed for security, such as antivirus, file backups, and file backup.

## **Travel**

Since Drop Ship Inc. has hubs in ten major cities in the southern and Mid-Atlantic United States, the team will be traveling to each city to learn more about each warehouse and tour each facility. Factoring in airfare for the warehouses that are further away, car rental and gas money for the warehouses that are closer, hotel costs, and food costs for these trips, we estimate a total \$50,000 in travel expenses.

## *Estimated Annual Operating Costs*

<b>Benefit</b>	<b>Amount</b>
Increased Efficiency	\$400,000
Decrease in Loss and Theft	\$75,000
New Business	\$1,425,000
Expand Existing Business	\$600,000
<b>Total Benefits</b>	<b>\$2,500,000</b>

### **Increased Efficiency**

Currently, Drop Ship Inc. pays 80¢ for every dollar of underlying cost, meaning that they make a profit of 20¢ for every dollar earned. Assuming a profit of \$1,000,000, DSI would make \$200,000. With the benefits that the new system will provide, DSI will expect to make a profit of 40¢ for every dollar earned, doubling the total profit from \$200,000 to \$400,000.

### **Decrease in Loss to Theft/Damage**

Currently, Drop Ship Inc. experiences \$50,000 per year in losses due to stolen and damaged items. With the new system, we expect DSI to save \$75,000 per year by selling products that would have otherwise been stolen or damaged.

### **New Business**

With this new system, Drop Ship Inc. expects to enter three new markets, and gain five clients in each market, with each client bringing in \$95,000 in profits per year. These fifteen new clients will bring in \$1,425,000 per year.

### **Expand Existing Business**

Currently, the clients of Drop Ship Inc. split their business between DSI and three other clients, giving \$150,000 a year in profits to DSI. The new system will allow Drop Ship's clients to use DSI for all of their business needs, meaning that Drop Ship Inc. will bring in \$600,000 in additional profits.

## *Estimated Annual Benefits*

	Years			
	2019	2020	2021	2022
Estimated Cost	\$232,000	\$232,000	\$232,000	\$232,000
Leftover Cost	\$0	\$232,000	\$392,000	\$0
Estimated Benefits	\$0	\$72,000	\$1,261,500	\$1,261,500
	(\$232,000)	(\$392,000)	\$637,000	\$1,029,000

### **Summary**

We split the project payment throughout 4 years to spread out the cost for DSI. Since the project is planned to be finished before Black Friday 2020, there is no benefit for 2019. Since there are only 5 weeks left in the year where the system will first be used (2020), we multiplied the efficiency and decrease in loss benefit by 5/52. We then assumed that since the new system will be available by Black Friday, a busy time for customers, that 3 new clients will do business with DSI. For the 2 following years, we took the remaining benefits and split them. We anticipate that our system will reach all of the expected benefits.

# **Project Risk and Feasibility Document**

## *Potential Risks*

### **Organizational Risks and Feasibility**

- Employees may feel that they do not have the appropriate skills required to operate the new system
- The employees may fear the increased responsibility that comes with the adoption of a more efficient system
- The new system may cause fear of having more responsibilities than previously, such as getting a manager to sign off on returns
- Improvement in technology may cause employees to fear that their job may eventually become automated

### **Technological Risks and Feasibility**

- Clients may have trouble operating the new system once it is initiated

### **Resource Risks and Feasibility**

- The project manager should be aware of losing team members during the project due to other jobs
- Replacement team members may lack required skills to perform to their expectations

### **Schedule Risks and Feasibility**

- Availability of project members is unknown over a span of 60 weeks
- Estimates are not exact, and certain aspects of implementing the system may take longer than expected

## *Cost Benefit*

We estimate that if everything goes according to plan, DSI will break even in 2021. Our system will vastly improve the operations at DSI, reducing theft, increasing efficiency, and expanding business. Since the cost is split between 4 years, it is not a heavy burden on DSI and will improve profits gained from the new system.

## *Argument for Approval*

Drop Ship Inc. is a fairly successful company that has the potential to become great. However, their current information system is the primary facet of their business that is holding them back. The Righteous Row will help transform Drop Ship Inc. into a shipping powerhouse in the Upper South and Mid-Atlantic United States. Through improvements to the returns process, an increased efficiency in business processes, increased communication between departments, and the expansion of business, we guarantee that the adoption of this new system will cause DSI to optimize business operations.

# **Project Approval Document**

From the Office of CK Investor Services,  
100 University Pl, Newport News, VA, 23606  
November 1st, 2018

Hi Teams!

We wanted to thank you for your presentation of your initial project progress. Our organization has consulted, and after the additional information you provided us during the presentation, we have decided to formally approve your project. As investors, we look forward to hearing from you as you move on to the 3rd phase (analysis) of the project. Please let us know what we can do to help guide you through this project to successful completion!

Sincerely,  
Chris Kreider – CK Investor Services Representative

## Phase 2 Deliverables

### Project Environment Description

#### *Recording/Communication Plan*

##### **External Communication**

Skype allows for video calling with the client, which will be used on a weekly basis to communicate the work that was completed on the system in the past week. This weekly video call will also give both the development team and Drop Ship Inc. the chance to communicate any new information which may surface throughout the project. Furthermore, a monthly newsletter will be sent out to the workers of Drop Ship Inc. as the project progresses. This will allow the users to learn more about the new system before it releases, as well as come to management with any concerns.

##### **Internal Communication**

When communicating between our development team, we will use a Skype Business group for instant messaging. Instant messaging will be used for communicating the completion of daily business tasks, requests for help on certain facets of the project, or calls for meetings. Outside of the group, Skype will also be used for direct messaging between the project manager and the individual developers in the group if necessary.

#### *Processes and Procedures*

##### **Reporting and Documentation**

At the end of each workday, each employee will complete a log of what they accomplished that day. These logs will be shared and discussed with the team at the weekly meetings every Friday. As far as code documentation goes, standard JavaDoc will be used to document programs. Additionally, the pushes to Github should be documented with a brief title of what has changed in the code since it was pulled.

##### **Programming**

Pair programming will be used for the project. Upon approval of the project, each developer will be assigned a partner whom they will code with for the whole project.

##### **Testing**

Testing will be done at regular intervals (weekly basis) throughout the process. The project manager will assign a pair to work on the testing of the code.

##### **Deliverables**

Deliverables for the project will be transferred and collaborated on using Github, as it allows for the pushing and pulling of code quickly and easily.

# *Work Environment Description*

## **Personal Workstations**

Each developer will have their own desk in the office space, allowing enough room for their computer and two monitors. Additionally, since pair programming will be used, there will be room at these desks for another developer to have their own space and assist with programming.

## **Development Servers**

The office space will include 2 servers that will be used throughout the project. These servers will be maintained by the support staff.

## **Office Space**

The office space purchased will contain four offices, with the project manager having a private office and the remaining six developers sharing an office with their primary partner programmer. In addition to the offices, there will be a reception area, a conference room, a printing area, two bathrooms, and a kitchenette.

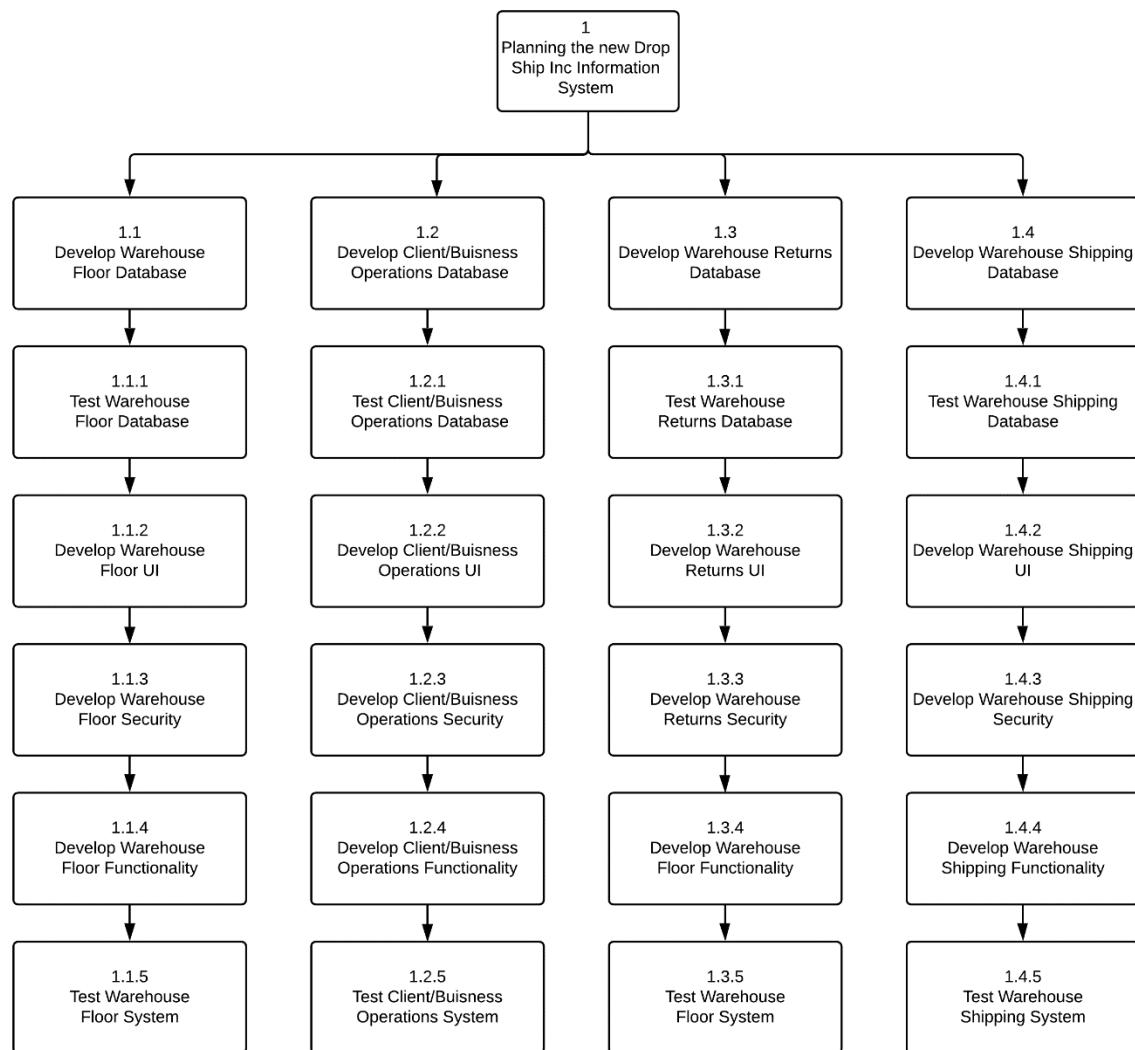
## **Support Staff**

We will have support staff ready to help whenever a piece of equipment or the servers break down. They will also provide upkeep and maintenance, and will be on call if something ever goes wrong.

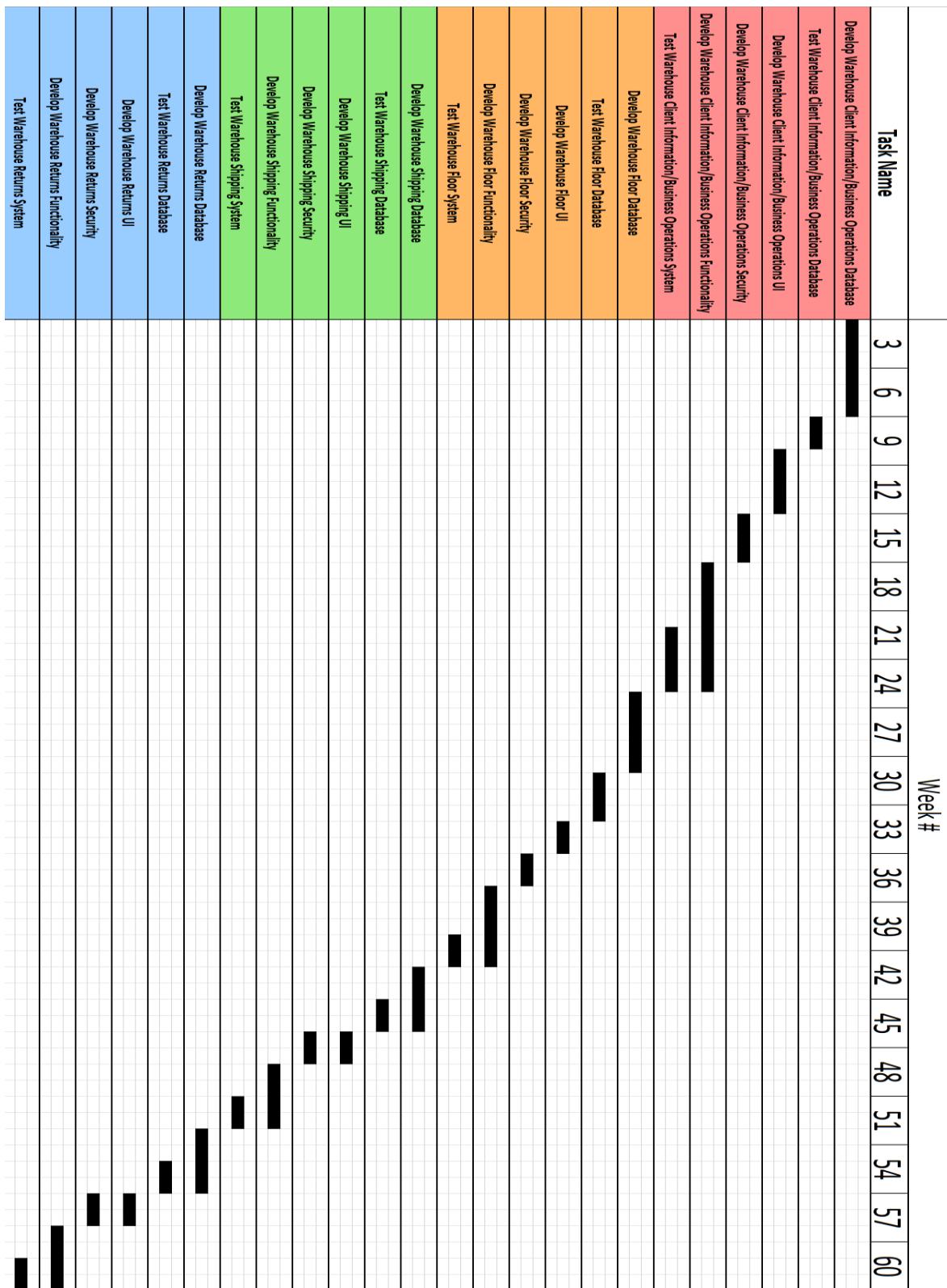
# Work Breakdown Structure

Work Breakdown Structure

Righteous Row



# Project GANTT Chart



# Project Staff/Resource Allocation Plan

Since we have a team of seven comprised of 1 project manager and 6 analysts, we decided that splitting the team up into 2 teams would be the optimal way to develop the system. Each team will have 1 team leader who reports to the project manager. The analysts will then be grouped up and pair program throughout the system development. This ensures that communication flows throughout the group and that work can get done more efficiently.

It is highly unlikely that equipment or additional staff will be ready by the time we start the project, so that time will be spent training the group to use the equipment, devising a detailed development plan, and conducting team building exercises. This will guarantee that our group is ready and trained when equipment and support staff arrive and that no time will be wasted waiting for their arrival.

# Project Evaluation and Monitoring Plan

## *Issue Tracking Plan*

Issue	Date Posted	Description	Priority	Impact	Reported by	Target date	Resolution Description	Status	Fix Date
1	10/22/2019	Example	Low	Milestone 1	Avery L	10/24/2019	Team met up and fixed the issue	Closed	10/24/2019

## **Description**

Our group will use this chart to record and store any issues we come across while developing the system. This way, we keep a detailed and thorough list of problems and can hone in on specific reoccurring issues. We hope to keep this updated throughout the entire process.

# Phase 3 Deliverables

## Information Gathering Plan

### *Company Description*

#### **Overview**

This document gives a brief description of Drop Ship Inc., and lists the problems that the company is having with their current information system. We used this document to guide us in understanding what was absolutely necessary for the new system to do.

#### **Document**

CPSC 350 Fall 2019 – Final Project – Drop Ship Inc. Company Profile

#### **Introduction**

Drop Ship Inc. (DSI) . is a distribution company that specializes in handling other manufactures products. Specifically, they provide a full series of warehousing, packaging, shipping and returns handling services for clients who do not want to handle it themselves. Drop Ship Inc. started as a small company in one of the partner's barns, and has since grown to over 10 warehouses in 10 major cities. DSI's tagline is “you provide the product, and we provide the service”, however, as their company has grown, their fragmented information system has resulted in significantly decreased service.

#### **Company Profile**

DSI currently has 10 small to medium warehouses. They are currently eying several expansion opportunities, including the potential acquisition of a distribution center. Each warehouse consists of 4 major areas which include warehousing/storage, picking/packing, shipping, and returns. DSI ships exclusively to the United States.

#### **Initial Problem Description**

Currently, DSI has 3 large and separate information systems that do not interact with each other including the warehouse/inventory management system, the order management system, and the shipping management system. As a result, a significant amount of manual interaction and coordination is needed between departments. Specifically, they mention the following problems

- **Warehouse System does not communicate with the order system** – If they have 100 widgets, and somebody orders 2, the warehouse system does not update to 98. As a result, they spend a large amount of time guessing how much inventory they have on.
- **Shipping System does not communicate with the warehouse system** – the warehouse system has information on the products such as product weight and size. As these systems do not communicate, the company ends up “guessing” what size box a shipment will need, and how much it will cost to ship it.

- **The Customer Management System does not communicate with the order system** – Customers are those who order products that DSI picks, packs and ships. The Customer management System tracks the customer address, as well as order and returns history. Currently, this is a mostly manual process, as once an order is filled, an employee needs to enter the information into this system. Due to the manual nature of the task, it is not uncommon for this step to be skipped resulting in low quality data.

DSI is looking for a team to develop a new unified information system to handle their day to day business, which as a result, should solve the above problems.

# *DSI – High Level Business Process Review*

## **Overview**

This document contained the results of an internal review in early 2016. It focused on some of the key business processes that DSI practiced. We used this information to gain insight to how certain parts of the system could be upgraded, and how we could keep certain parts of the business that work well.

## **Document**

Drop Ship Inc – High Level Business Process Review

NOTE: results of internal review early 2016. Focusing on key aspects of business, not all business processes covered.

## **Abbreviations**

AE – account executive

WFM – warehouse floor manager

WSM – warehouse shipping manager

WRM – warehouse returns manager

WFS – warehouse floor staff

WSS – warehouse shipping staff

WRS – warehouse returns staff

## **New Client Process**

1. AE creates new client information in system
2. AE adds products to the system
3. WFM adds items to inventory when products arrive
4. WFS places items in location within the warehouse

## **Add New Product**

1. AE adds products to the system
2. WFM adds items to inventory when products arrive
3. WFS places items in location within the warehouse

## **Add New Inventory**

1. AE confirms with client that new shipment is inbound
2. WFM adds items to inventory when product arrives
3. WFS places items in location within the warehouse

### **Ship New Order**

1. WSM initiates order fulfillment period
2. Multiple WSS are assigned to shipping stations
3. At each station, for each order a single member of the WSS
  - a. WSS identifies items to be packaged
  - b. WSS identifies target packing container
  - c. WSS locates and retrieves products to be shipped
  - d. WSS closes and seals the carton once packing is complete
  - e. WSS selects “packing complete” and shipping label is printed
  - f. WSS applies shipping label
  - g. WSS places completed package in shipping container

### **Process Return**

1. WRM identifies WRS that batch of returns has arrived
2. Each WRS works at a returns station where
  - a. WRS identifies order number from original container, packing slip, or address
    - i. if unable to obtain original order number, WRM overrides the need to provide this information
  - b. WRS opens container and checks contents against the original order
    - i. For each item returned, check if it is
      1. High value (>50\$)
        - a. Get approval from WRM
      2. Disposable – if so, place in designated disposal container
      3. Returnable – if so, check that it is in appropriate condition
    - c. WRS marks in the system that the item was returned
  3. At end of returns batch, WRM reviews all returns for that batch

### **Conclusion**

This report summarized several key operations and internal processes currently in place. There are some suggestions for improvement which will be addressed at a later meeting.

# *Expected Benefits*

## **Overview**

The expected benefits document gave us insight into how much money Drop Ship Inc. expects to make from an updated information system. We used this information to prepare the estimated annual benefit table.

## **Document**

### **Project Expected Benefits**

The client has identified the following expected benefits of the new DSI.

#### **Increase Efficiency**

Currently, for every dollar we earn, 80 cents is paid to cover the underlying costs. The biggest of these costs is human resources. The new system is designed to mirror other companies that have decreased their operating costs significantly. Assume we had \$1,000,000 in total money paid to us by client last year (which means we would only see about 200,000 as profit after paying all the bills!). Our expectation is that with the new system, we should reduce our costs to 60 cents for every dollar we earn. This will be done by increasing the efficiency, specifically, the time it takes from the time an order is opened, to the time it is in the bin ready for shipment. As a result, less staff will be able to ship more packaged, decreasing the costs.

#### **Decrease In Loss to Theft/Damage**

Currently, we lose about \$50,000 a year in damaged/stolen products. This loss occurs during the return process. When a product is returned, our staff will open the package, and determine whether it is too damaged (it is then destroyed). Otherwise, the product is returned to stock and shipped again. Our hope is that a better managed return process will prevent problems with this process. Each year after year one, we would expect to save \$75,000 in this area.

#### **New Business**

The new system should allow us to purchase new warehouses in new markets, and acquire new customers. It is our hope that we should be able to enter 3 new markets, and acquire 5 new customers per market. It is our hope that the average customer will bring in an additional \$95,000 a year in profits. This profit estimate assumes the increase in efficiency addressed above.

#### **Expand Existing Business**

Existing clients are prepared to transfer all of their business to us, which they currently split between 4 different providers. We currently receive \$150,000 a year in profits from this client. Assuming we can acquire all of their business, this would quadruple. This profit estimate assumes the increase in efficiency addressed above.

# *Fully Developed Use Case*

## **Overview**

This document included a blank shell of the fully developed use case. We used this shell as the basis for each fully developed use case description that we completed.

## **Document**

<b>Use Case Name</b>		
<b>Scenario</b>		
<b>Triggering Event</b>		
<b>Brief Description</b>		
<b>Actors</b>		
<b>Related Use Cases</b>		
<b>Stakeholders</b>		
<b>Preconditions</b>		
<b>Postconditions</b>		
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
<b>Exception Conditions</b>		

# *Interview Transcript*

## **Overview**

This is an interview transcript with a warehouse shipping manager. This helped us with finding use cases and non-functional requirements. It also helped solidify which role did what tasks, giving us information for our graphical use case model.

## **Document**

CPSC350 Final Project DSI

Interview Transcripts

## **Overview**

The following is the transcript of an interview with one of our key stakeholders, the *warehouse shipping manager*. This stakeholder is familiar with all of the activities performed by warehouse staff, including other managers.

### **Warehouse Shipping Related Questions**

**Question:** As the shipping manager, what roles/tasks do you oversee?

**Answer:** The shipping manager is the senior warehouse manager. The Shipping manager is familiar with all other roles/tasks performed in the warehouse. These fall into three departments: *shipping*, *floor* and *returns*.

**Question:** What are the primary ways in which the *shipping* staff will need to use the new system?

**Answer:** First and most importantly, they need to be able to ship an order. This is not necessarily a single task, but is used to describe a series of tasks in the order lifecycle. *Shipping staff* need the system to perform the following tasks as part of the “ship order” task

- Retrieve next order to be shipping
- Look up product information
- Print the packing slip
- Mark an order as packed
- Print the shipping label
- Mark order as ready for shipment

These tasks are repeated by shipping staff until all orders for a batch are complete.

**Question:** Is there anything specific you can do as the *shipping manager* that *shipping staff* cannot?

**Answer:** Yes, specifically, I initiate each order batch (at 8 am, 12 pm and 4pm). Once I do this, orders are then assigned to workstations in the warehouse where *shipping staff* then start the above process. Additionally, if there is an order that for some reason or another,

we do not have enough inventory to fill, I can mark an order as “insufficient inventory” so that it will go back into the batch to be processed once we have enough inventory.

### **Warehouse Floor Related Questions**

**Question:** Switching gears, I now have some questions about *warehouse floor staff*. Specifically, can you describe to me how *floor staff* would need to use a new system?

**Answer:** Sure! Floor staff are there to ensure that warehouse operations run smoothly. This generally falls into a few categories of uses including:

- Looking up product information
- Updating the on hand inventory for products
- Update inventory locations (As the result of a location transfer)

**Question:** Can you elaborate on “performing inventory transfers between locations”

**Answer:** Our warehouse needs to carefully track what products are at what location. If for some reason, a product (or batch of products) needs to be moved to a more appropriate location, this is performed by the floor staff. They specifically look up the existing product location, determine a new location, perform the move, and then update it in the system.

**Question:** Are there any tasks that the *warehouse floor manager* can perform that the general *floor staff* cannot?

**Answer:** Yes, on top of being able to do everything warehouse floor staff can, specifically, the *warehouse floor manager* is responsible for

- Assigning new inventory to a location in the warehouse
- Approving inventory transfers
- Approving inventory quantity updates

**Question:** Can you elaborate on approving inventory transfers and quantity updates, what does that mean?

**Answer:** Before inventory is moved to a new location by warehouse floor staff, a manager must approve it. This is for, among other reasons, safety reasons – we need to ensure that a given location is safe for a given product and the manager has the oversight to do that. In terms of approving inventory updates, any change in an inventories quantity must be approved by a manager. This is to, among other things, protect against theft. Warehouse Returns Related Questions

**Question:** Finally, with regards to warehouse operations, I have some questions about the returns department. Specifically, what tasks do general *returns staff* perform?

**Answer:** Processing returns is an important part of the service we provide our clients. Our returns staff perform this task, and rely heavily on the system to perform this task. A typical return would require the following steps in some order to be performed

- Look up order information
- Look up product information
- Mark order as returned
- Mark order as disposed

The process generally goes like this; a carrier (e.g. USPS) drops off all of the returns for a given day. Our staff then start processing the returns. Each returns staff member starts with one returned package, and looks up the order that it came from. From there, the items in that order are looked up, to determine if they need to be returned or destroyed. Finally, the staff member unpacks the item and performs the appropriate action.

**Question:** What tasks does the *returns manager* perform that *returns staff* cannot?

**Answer:** On top of being able to do everything *returns staff* can, The *returns manager* is in charge of oversight. The new system must enable our returns manager to perform the following

- Approve high dollar item returns
- Approve high dollar item disposal
- Monitor Returns Staff

**Question:** Can you elaborate more on “monitor returns staff”

**Answer:** Returns are one of the biggest areas where our clients can become unhappy. Being thorough in ensuring only valid returns are accepted maximize our clients sales profits. Specifically, we monitor our returns staff, to see 1.) how many returns they process per period (day, week and month) and 2.) how many high dollar items they process per period and 3.) how many items are disposed of each period. In this report, we are looking for anomalies of either underperforming staff, or staff who are disposing of too much product.

## Conclusion

**Interviewer:** Thank you for your time, we really appreciate your insight!

**Interviewee:** No problem, don't hesitate to let me know if you have any other questions or concerns!

## *Other Use Cases*

### **Overview**

Other use cases is an excel doc with users and their respective use cases. This document helped us with finding use cases and their users.

### **Document**

administrative account executive manager	everything account executive staff can do
administrative account executive manager	approve new client creation
administrative account executive manager	approve product deletion
administrative account executive staff	create new client
administrative account executive staff	review existing client
administrative account executive staff	update exiting client
administrative account executive staff	delete existing client
administrative account executive staff	create new products
administrative account executive staff	look up product information
administrative account executive staff	update product information
administrative account executive staff	delete existing products
administrative account executive staff	review existing order
administrative business operations manager	everything business operations staff can do
administrative business operations manager	approve new hires
administrative business operations manager	approve firing employees
administrative business operations manager	approve pay changes
administrative business operations staff	assign staff to shifts
administrative business operations staff	create new staff
administrative business operations staff	review existing staff information
administrative business operations staff	update existing staff information
administrative business operations staff	delete fired staff
Clients	create new order
Clients	cancel existing order
Clients	review existing orders
Clients	look up product information

# *Project Background*

## **Overview**

This document is a basic background on the project that includes the system capabilities and the business benefits. These provided us with useful information to write our system vision document.

## **Document**

### **Problem Description**

- Shipping Company that needs an information system to manage its business
- Currently, a combination of technology is used including an access database, email, and some custom written code that seems to keep breaking.
- Company is expected to grow rapidly and expand into new locations over the next few years

### **System Capabilities**

- Manage Client Information (e.g. product information which is provided by the client)
- Manage Inventory Information
- Manage Shipping Process
- Manage Returns Process
- Manage Employee relationship to activities

### **Business Benefits**

- Reduce time to acquire and onboard new clients
- Better manage inventory by clearly tracking quantity and location of all inventory
- Better manage shipping process by clearly understanding where every order is in the order lifecycle
- Better manage returns which will prevent theft and reduce costs
- Better manage employees performance at specific tasks

## *Sample Data*

### **Overview**

Sample data contained several excels table that has different objects with their respective fields. This helped with creating the tabular domain classes, the ERD, and the graphical domain class model.

## Document

### Employees

#### DSI Abbreviated Employee Role

Note: This is a template we use for hiring staff for new warehouses. Roles remain the same but staffing levels may vary

Employee Id	Employee Name	Whse	Department	Role	Level
123456	Chris, Captain	101	Administrative	Accounts Executive Business	Manager
123457	Chris, Captain	101	Administrative	Operations	Manager
123458	Chris, Captain	101	Administrative	Account Executive	Staff
123459	Chris, Captain	101	Administrative	Account Executive	Staff
123460	Chris, Captain	101	Administrative	Account Executive	Staff
123461	Chris, Captain	101	Administrative	Account Executive Business	Staff
123462	Chris, Captain	101	Administrative	Operations Business	Staff
123463	Chris, Captain	101	Administrative	Operations	Staff
123470	Chris, Captain	101	Warehouse	Floor	Manager
123471	Chris, Captain	101	Warehouse	Returns	Manager
123472	Chris, Captain	101	Warehouse	Shipping	Manager
123473	Chris, Captain	101	Warehouse	Returns	Staff
123474	Chris, Captain	101	Warehouse	Returns	Staff
123475	Chris, Captain	101	Warehouse	Returns	Staff
123476	Chris, Captain	101	Warehouse	Shipping	Staff
123477	Chris, Captain	101	Warehouse	Shipping	Staff
123478	Chris, Captain	101	Warehouse	Shipping	Staff
123479	Chris, Captain	101	Warehouse	Shipping	Staff
123480	Chris, Captain	101	Warehouse	Shipping	Staff
123481	Chris, Captain	101	Warehouse	Shipping	Staff
123482	Chris, Captain	101	Warehouse	Shipping	Staff
123483	Chris, Captain	101	Warehouse	Shipping	Staff
123484	Chris, Captain	101	Warehouse	Shipping	Staff
123485	Chris, Captain	101	Warehouse	Shipping	Staff
123486	Chris, Captain	101	Warehouse	floor	Staff
123487	Chris, Captain	101	Warehouse	floor	Staff
123488	Chris, Captain	101	Warehouse	floor	Staff
123489	Chris, Captain	101	Warehouse	floor	Staff
123490	Chris, Captain	101	Warehouse	floor	Staff

## Warehouses

DSI Warehouse Locations					
WHSE #	City	Address	Storage Locations	Shipping Stations	
101	Norfolk	redacted	614	8	
	Newport				
102	News	redacted	460	6	
103	Baltimore	redacted	768	10	
104	Charlotte	redacted	691	9	
105	Nashville	redacted	614	8	
106	Savannah	redacted	537	7	
107	Charleston	redacted	537	7	
108	Richmond	redacted	691	9	
109	Charlottesville	redacted	460	6	
110	Blacksburg	redacted	537	7	

## Clients

DSI Client Information				
client ID	name	address	Primary contact	
001	bobs widgets tims	redacted	name contact,	
002	thingamabobs	redacted	name contact,	
003	galindas gadgets	redacted	name	

## Products

product id	client id	product name	product description	product dimension	product weight
123456	001	widget	cool thingy!	2'x1'x1'	8oz
123457	002	thingamabob	look at this thing, how cool is it! even more widgitier than before,	1"x1"x1"	2 oz
123458	001	ultrawidget	who-da-thunk?	3'x2'x2'	2 lbs
123459	003	gadgetorous gadget	have you ever seen a better gadget? I think no!	6"x6"x2"	9oz

## Inventory

<b>product</b>			<b>on hand</b>	<b>available</b>
<b>id</b>	<b>whse</b>	<b>location</b>	<b>balance</b>	<b>balance</b>
123456	001	A-1-3	10	9
123456	002	B-7-2	10	10
123456	003	A-1-2	10	9
123456	004	D-8-3	10	9
123456	005	F-1-1	10	10
		C-11-2, C-11-		
123457	001	3	100	100
123458	009	X-8-3	105	20
123459	010	ZZ-2-2	10	0

## Customers

NOTE: Customer information is provided to us only when a new order is provided, we do not track customers that we have never had an order for

### customer

<b>id</b>	<b>Address</b>	<b>City</b>	<b>State</b>	<b>Zip</b>	<b>Phone Number</b>
		newport			
001	1000 university pl	new	VA	23606	75759488587
002	123 anywhere st	somewhere	US	12345	1234567890

## Orders

NOTE: orders track the big picture, it is the order items list that indicates what will get shipped  
total

<b>order</b>	<b>customer</b>	<b>line</b>	<b>total</b>	<b>total</b>	<b>packed</b>
<b>id</b>	<b>id</b>	<b>items</b>	<b>items</b>	<b>charge</b>	<b>date/time</b>
001	001	1	5	100	11/1/2017
002	001	5	25	250	11/1/2017

## Order Items

<b>order id</b>	<b>order seq</b>	<b>product id</b>	<b>quantity</b>
001	1	123456	5
002	1	123456	5
002	2	123457	15
002	3	123458	5

## **Shipping**

<b>order id</b>	<b>carrier</b>	<b>tracking number</b>
001	UPS	UPS- 123456778903 FX-123456-
002	FEDEX	DE789HI

# Stakeholder Identification and Classification

## List View

**Stakeholder Name:** *Administrative Accounts Executive Staff*

**Classification:** *Internal Operational*

This staff deals mostly with the client information, primarily by adding in new clients, deleting clients, updating old clients, and reviewing existing client information. Along with dealing with clients, the Administrative Accounts Executive Staff also handles product info. Additionally, they review existing product information, can create or delete product information, and can look up and update existing product information.

**Stakeholder Name:** *Administrative Accounts Executive Manager*

**Classification:** *Internal Executive*

In addition to approving new client creation and product deletion, the Administrative Accounts Executive Manager can do everything account executive staff can do.

**Stakeholder Name:** *Administrative Business Operations Staff*

**Classification:** *Internal Operational*

The Business Operations Staff ensure that the warehouses run smoothly by doing tasks such as scheduling employee shifts and managing the database of existing employees.

**Stakeholder Name:** *Administrative Business Operations Manager*

**Classification:** *Internal Executive*

In addition to being capable of doing everything that the Administrative Business Operations Staff can do, the ABO Manager is responsible for the approval of newly hired or fired employees. They are also responsible for approving employee pay changes.

**Stakeholder Name:** *Warehouse Floor Staff*

**Classification:** *Internal Operational*

The Warehouse Floor Staff are employed to ensure that all warehouse operations run smoothly. They do so by looking up product information and updating the on-hand inventory for products. Additionally, they perform inventory location changes and subsequently update the inventory database for items that need to move to a different location in the warehouse.

**Stakeholder Name:** *Warehouse Floor Manager*

**Classification:** *Internal Executive*

In addition to being capable of doing what the Warehouse Floor Staff can do, the Warehouse Floor Manager is responsible for assigning new inventory to a specific location in the warehouse. They also must approve inventory transfers, or when an item in the warehouse needs to change location - the Warehouse Floor Manager needs to approve this new location as being a safe and secure place for the item. Additionally, the WFM is responsible for updating changes in inventory quantity in order to prevent theft.

**Stakeholder Name:** *Warehouse Shipping Staff*

**Classification:** *Internal Operational*

The shipping staff are the employees who use the system to ship products. This task is comprised of several smaller tasks: retrieving the item(s) to be shipped, looking up the product information, printing the packing slip, marking an order as packed, printing the shipping label, and marking the order as ready for shipment.

**Stakeholder Name: *Warehouse Shipping Manager*****Classification: *Internal Executive***

The Warehouse Shipping Manager is the highest ranking warehouse employee who is familiar with all of the roles and tasks completed throughout the entire warehouse (shipping, returns, floor). In addition to being capable of doing everything that the Shipping staff does, the WSM initiates an order batch three times each day at 8:00 A.M., 12:00 P.M., and 4:00 P.M., in which orders are assigned to workstations manned by the shipping staff. Once a batch is initiated, the shipping staff begins the process of shipping the order. The WSM can also mark an order as “insufficient inventory”, meaning that it will be sent back to the batch to be shipped when the appropriate inventory is available.

**Stakeholder Name: *Warehouse Returns Staff*****Classification: *Internal Operational***

The Warehouse Returns Staff relies heavily on the system to perform their task. The process of returns staff is to look up order information and then look up product information. Based off information found, the product is to be marked as returned or disposed. The process is finished by a member unpacking returned product and performing appropriate action.

**Stakeholder Name: *Warehouse Returns Manager*****Classification: *Internal Executive***

The Warehouse Returns Manager is able to do everything returns staff can, while being in charge of oversight. The WRM must approve high dollar item returns/disposal and monitor returns staff (returns must be thorough to keep clients happy, therefore the WRM is expected to be careful and accurate on all decisions). The WRM monitors staff, specifically, to keep track of how many returns were processed (day, week, month) and account for all high dollar items processed per period/ how much is being disposed.

**Stakeholder Name: *Clients*****Classification: *External Executive***

Clients interact with the system the least when compared to any of the employees. They can create or cancel orders, and once an order is made, review that order. Clients also can look up product information.

***Tabular View***

	Operational	Executive
Internal	Accounts Executive Staff Business Operations Staff Warehouse Shipping Staff Warehouse Floor Staff Warehouse Returns Staff	Accounts Executive Manager Business Operations Manager Warehouse Shipping Manager Warehouse Floor Manager Warehouse Returns Manager
External		Clients

# Requirements Definition

## *Functional Requirements*

### **Client Information/Business Operations Subsystem**

1. Create New Client
2. Approve New Client Creation
3. Review Existing Client
4. Update Existing Client
5. Delete Existing Client
6. Approve Client Deletion
7. Assign Staff to Shifts
8. Create New Staff
9. Approve New Hires
10. Review Existing Staff Information
11. Update Existing Staff Information
12. Delete Fired Staff
13. Approve Firing Employees
14. Request Pay Change
15. Approve Pay Changes
16. Look Up Order Information
17. Create New Order
18. Review Existing Order
19. Cancel Existing Order

### **Warehouse Returns Subsystem**

20. Mark Items as Returned
21. Approve High-Dollar Item Returns
22. Mark Items as Disposed
23. Approve High-Dollar Item Disposal
24. Generate Returns Report

### **Warehouse Floor Subsystem**

25. Create New Product
26. Look Up Product Information
27. Update Product Information
28. Delete Existing Product
29. Approve Product Deletion
30. Update On-Hand Product Quantity
31. Approve Product Quantity Update
32. Update Product Location
33. Approve Product Location Change
34. Assign New Product to Location

### **Warehouse Shipping Subsystem**

35. Initiate Order Batch
36. Identify Target Packing Container(s)
37. Mark Order as Packed
38. Mark Order as Ready for Shipment
39. Mark Order as Insufficient Inventory

## *Nonfunctional Requirements*

### **Usability Requirements**

- In addition to English, the new system will support both Spanish and French
- The new system will have a search function that gives the user the ability to search for tasks that the system can perform
- Has help function page to give user a list of helpful functions to navigate the system
- The new system will use the use the DSI color scheme

### **Reliability Requirements**

- The system will have a backup database in the event of a database failure or malfunction
- The system will be up and running during 99% of business hours
- The System will take downtime to update as needed during less busy times or overnight

### **Performance Requirements**

- The system will support a maximum of 750 connections at 125ms

### **Security Requirements**

- Each employee will have user ID and password
- Each client will also have client ID and password
- Data on server is encrypted and secured

# Use Cases

## *Overview*

For each subsystem, there is a Graphical Use Case Model, depicting the functional requirements and the actors who complete each activity. For each functional requirement in each subsystem, there is a Fully Developed Use Case Description, User Story with Acceptance Criteria, and the accompanying Activity Diagram and System Sequence Diagram.

## *Abbreviation Guide*

The following abbreviations will be used throughout the Use Cases for brevity's sake:

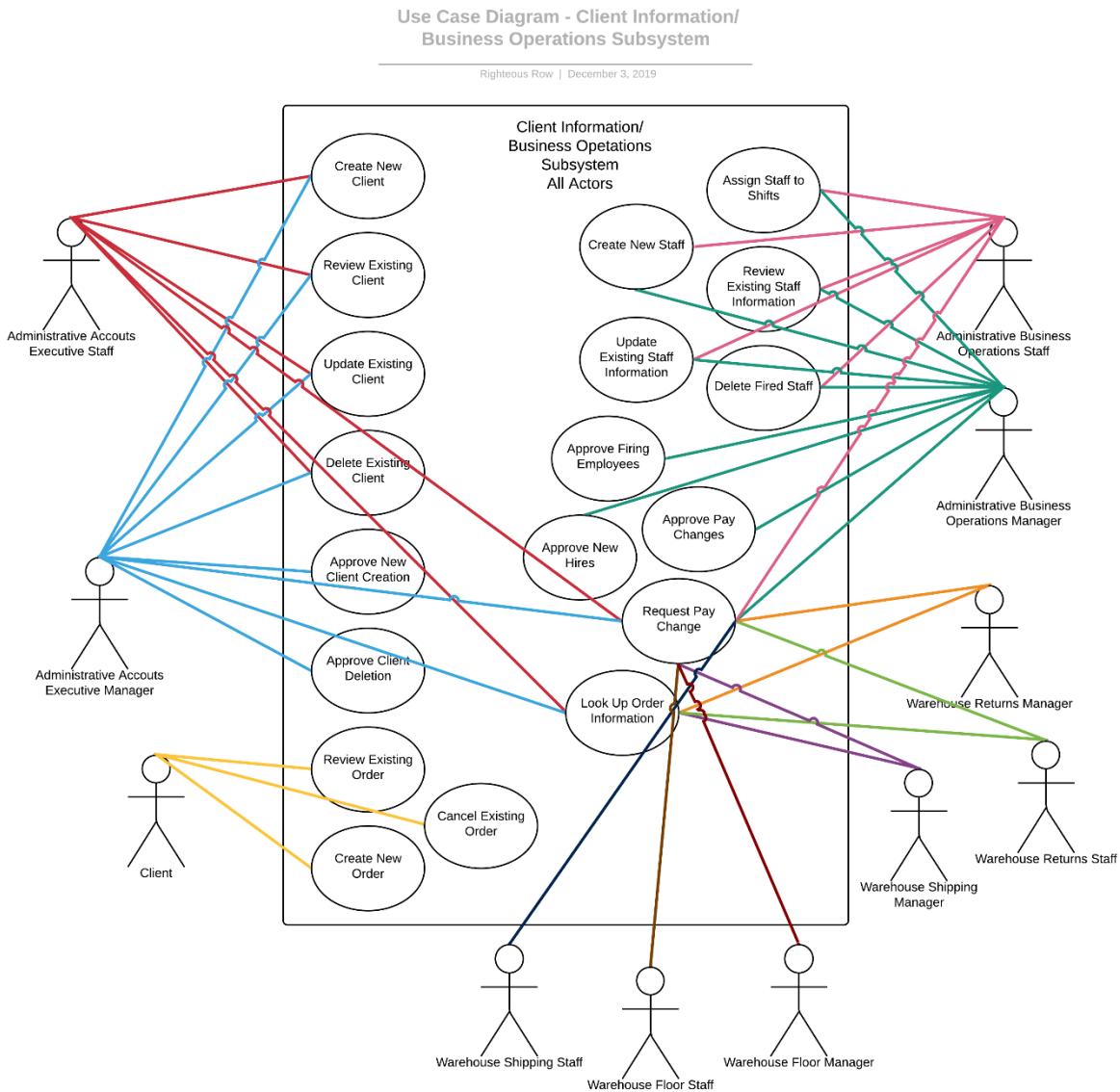
AAES	Administrative Accounts Executive Staff
AAEM	Administrative Accounts Executive Manager
ABOS	Administrative Business Operations Staff
ABOM	Administrative Business Operations Manager
WSS	Warehouse Shipping Staff
WSM	Warehouse Shipping Manager
WFS	Warehouse Floor Staff
WFM	Warehouse Floor Manager
WRS	Warehouse Returns Staff
WRM	Warehouse Returns Manager

The "Client" stakeholder will not be abbreviated

## *Tabular Use Case Definitions – Client Information/Business Operations Subsystem*

Client Information/Business Operations Subsystem		
Use Case Name	Actors	Brief Description
Create New Client	AAES, AAEM	A new client's information is added to the database
Approve New Client Creation	AAEM	A new client's information is reviewed and approved/denied
Review Existing Client	AAES, AAEM	An existing client's information is displayed
Update Existing Client	AAES, AAEM	An existing client's information is updated
Delete Existing Client	AAES, AAEM	An existing client is deleted from the database
Approve Client Deletion	AAEM	The deletion of an existing client is reviewed and approved/denied
Assign Staff to Shifts	ABOS, ABOM	Warehouse and executive employees are scheduled to shifts for two week periods
Create New Staff	ABOS, ABOM	A new employee's information is added to the database
Approve New Hires	ABOM	A new employee's information is reviewed and approved/denied
Review Existing Staff Information	ABOS, ABOM	An existing employee's information is displayed
Update Existing Staff Information	ABOS, ABOM	An existing employee's information is updated
Delete Fired Staff	ABOS, ABOM	An existing employee is deleted from the database
Approve Firing Employees	ABOM	The deletion of an existing employee is reviewed and approved/denied
Request Pay Change	AAES, AAEM, ABOS, ABOM, WFS, WFM, WSS, WSM, WRS, WRM	An employee requests a change in their pay
Approve Pay Changes	ABOM	A requested pay change is reviewed and approved/denied
Look Up Order Information	AAES, AAEM, WRS, WRM, WSM	An employee views an order by entering its Order ID in the system, and is shown the information of the order
Create New Order	C	A client places an order for DSI to ship their products from one of DSI's warehouses
Review Existing Order	C	A client views an active/previous order of theirs, and is shown its information
Cancel Existing Order	C	The client uses the system to cancel an order that they have previously placed with DSI

# Graphical Use Case Diagram – Client Information/Business Operations Subsystem



# *Fully Developed Use Case Descriptions with User Stories, Activity Diagrams, and System Sequence Diagrams – Client Information/Business Operations Subsystem*

## Use Case #1: Create New Client

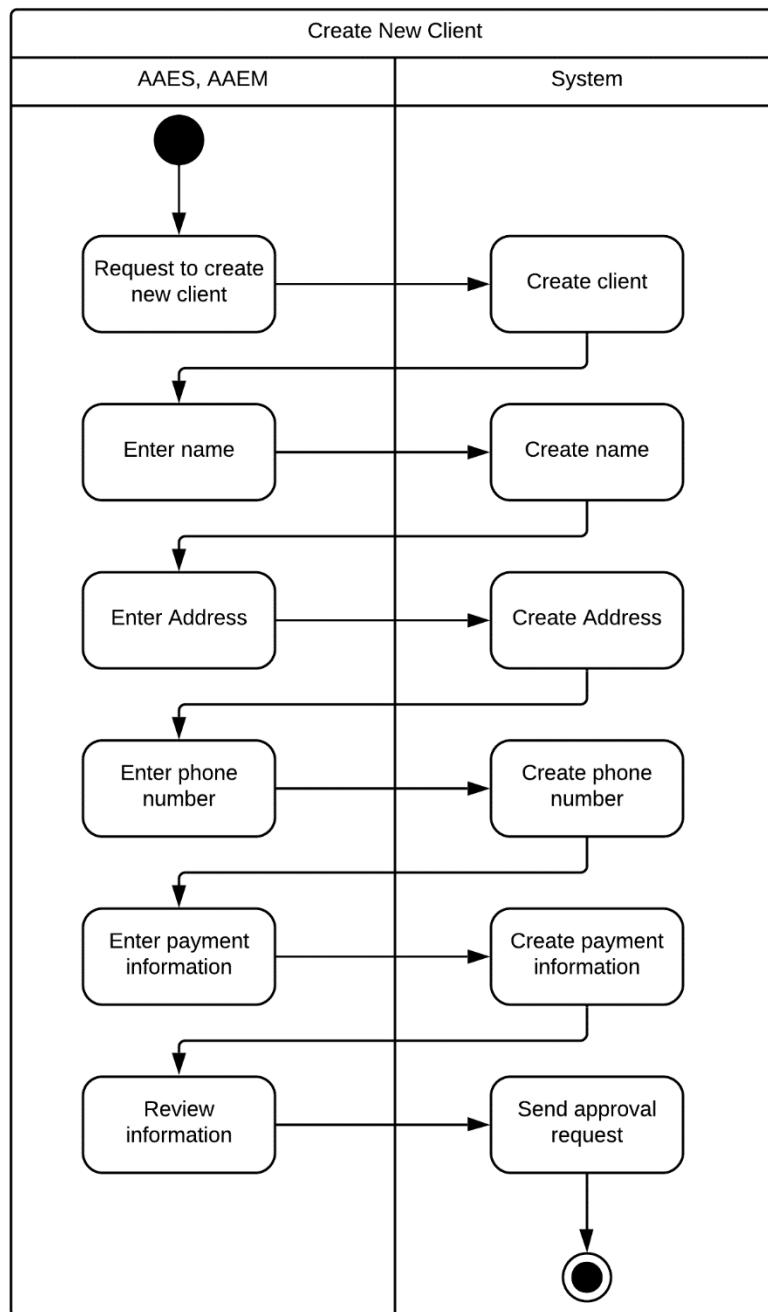
<b>Use Case Name</b>	Create New Client															
<b>Scenario</b>	Add new client into the system															
<b>Triggering Event</b>	A new customer wants to conduct business with Drop Ship Inc.															
<b>Brief Description</b>	A new client's information is added to the database															
<b>Actors</b>	AAES, AAEM															
<b>Related Use Cases</b>	Approve New Client Creation															
<b>Stakeholders</b>	Clients															
<b>Preconditions</b>	The AAES/AAEM has the client information that they wish to enter into the database															
<b>Postconditions</b>	A request to create a new client is sent to the AAEM															
<b>Flow of Activities</b>	<table border="1"> <thead> <tr> <th>User</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to add new client to the database</td> <td>1.1 System creates a new entry in the client database 1.2 System prompts for client name</td> </tr> <tr> <td>2. User inputs customer name</td> <td>2.1 System creates name 2.2 System prompts for client address</td> </tr> <tr> <td>3. User inputs customer address</td> <td>3.1 System creates address 3.2 System prompts for client phone number</td> </tr> <tr> <td>4. User inputs customer phone number</td> <td>4.1 System creates phone number 4.2 System prompts for client payment information</td> </tr> <tr> <td>5. User inputs customer payment information</td> <td>5.1 System creates payment information 5.2 System prompts user to review all information to complete database entry</td> </tr> <tr> <td>6. User reviews information and finishes database entry</td> <td>6.1 System sends an approval request to the AAEM</td> </tr> </tbody> </table>	User	System	1. User indicates desire to add new client to the database	1.1 System creates a new entry in the client database 1.2 System prompts for client name	2. User inputs customer name	2.1 System creates name 2.2 System prompts for client address	3. User inputs customer address	3.1 System creates address 3.2 System prompts for client phone number	4. User inputs customer phone number	4.1 System creates phone number 4.2 System prompts for client payment information	5. User inputs customer payment information	5.1 System creates payment information 5.2 System prompts user to review all information to complete database entry	6. User reviews information and finishes database entry	6.1 System sends an approval request to the AAEM	
User	System															
1. User indicates desire to add new client to the database	1.1 System creates a new entry in the client database 1.2 System prompts for client name															
2. User inputs customer name	2.1 System creates name 2.2 System prompts for client address															
3. User inputs customer address	3.1 System creates address 3.2 System prompts for client phone number															
4. User inputs customer phone number	4.1 System creates phone number 4.2 System prompts for client payment information															
5. User inputs customer payment information	5.1 System creates payment information 5.2 System prompts user to review all information to complete database entry															
6. User reviews information and finishes database entry	6.1 System sends an approval request to the AAEM															
<b>Exception Conditions</b>	3.1 Address is invalid 4.1 Phone number is invalid 5.1 Payment information is invalid															

As an AAES or AAEM, I want to **create a new client** so that **DSI can do business with them**

- Obtain all client information
- Finish client entry in database

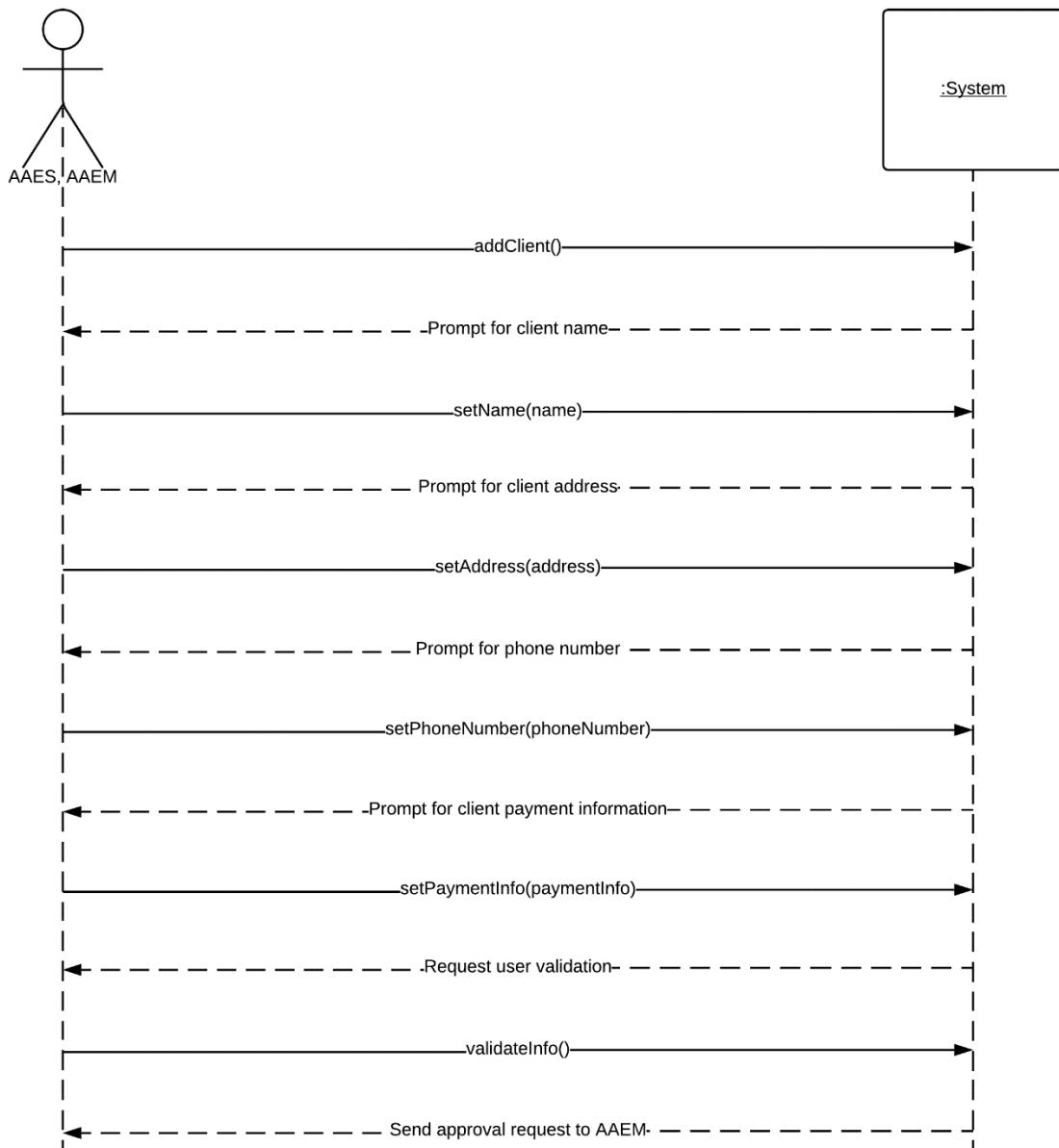
## Create New Client

Righteous Row



## Create New Client

Righteous Row



## Use Case #2: Approve New Client Creation

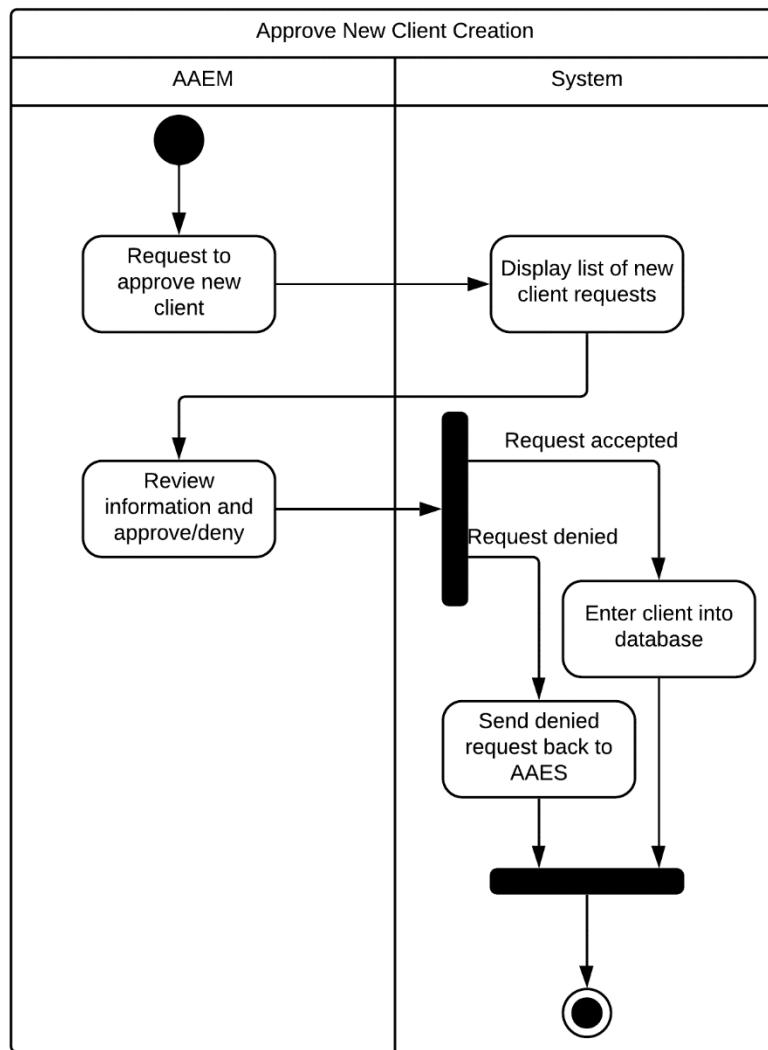
<b>Use Case Name</b>	Approve New Client Creation							
<b>Scenario</b>	The creation of a new client is approved							
<b>Triggering Event</b>	A new client is created in the database and must be approved by the AAEM							
<b>Brief Description</b>	A new client's information is reviewed and approved/denied							
<b>Actors</b>	AAEM							
<b>Related Use Cases</b>	Create New Client							
<b>Stakeholders</b>	Clients, AAES							
<b>Preconditions</b>	A new client has been created and sent to the AAEM for approval							
<b>Postconditions</b>	If approved, The new client is entered into the database If denied, the AAES is notified							
<b>Flow of Activities</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;"><b>User</b></th> <th style="text-align: center;"><b>System</b></th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to approve any new clients</td> <td>1.1 System displays list of new clients that are pending manager approval 1.2 System prompts user to review and approve/deny the creation of each new client</td> </tr> <tr> <td>2. User reviews the information of each new client, and marks them as either approved or denied</td> <td>2.1 System completes entry of approved new clients into the database 2.2 System sends denied requests back to the AAES</td> </tr> </tbody> </table>	<b>User</b>	<b>System</b>	1. User indicates desire to approve any new clients	1.1 System displays list of new clients that are pending manager approval 1.2 System prompts user to review and approve/deny the creation of each new client	2. User reviews the information of each new client, and marks them as either approved or denied	2.1 System completes entry of approved new clients into the database 2.2 System sends denied requests back to the AAES	
<b>User</b>	<b>System</b>							
1. User indicates desire to approve any new clients	1.1 System displays list of new clients that are pending manager approval 1.2 System prompts user to review and approve/deny the creation of each new client							
2. User reviews the information of each new client, and marks them as either approved or denied	2.1 System completes entry of approved new clients into the database 2.2 System sends denied requests back to the AAES							
<b>Exception Conditions</b>	1.1 There are no new clients pending manager approval							

As a **AAEM** I want to **approve new client creation** so that **clients can be saved in our system**

- Client creation request has been approved
- The new client exists within the database

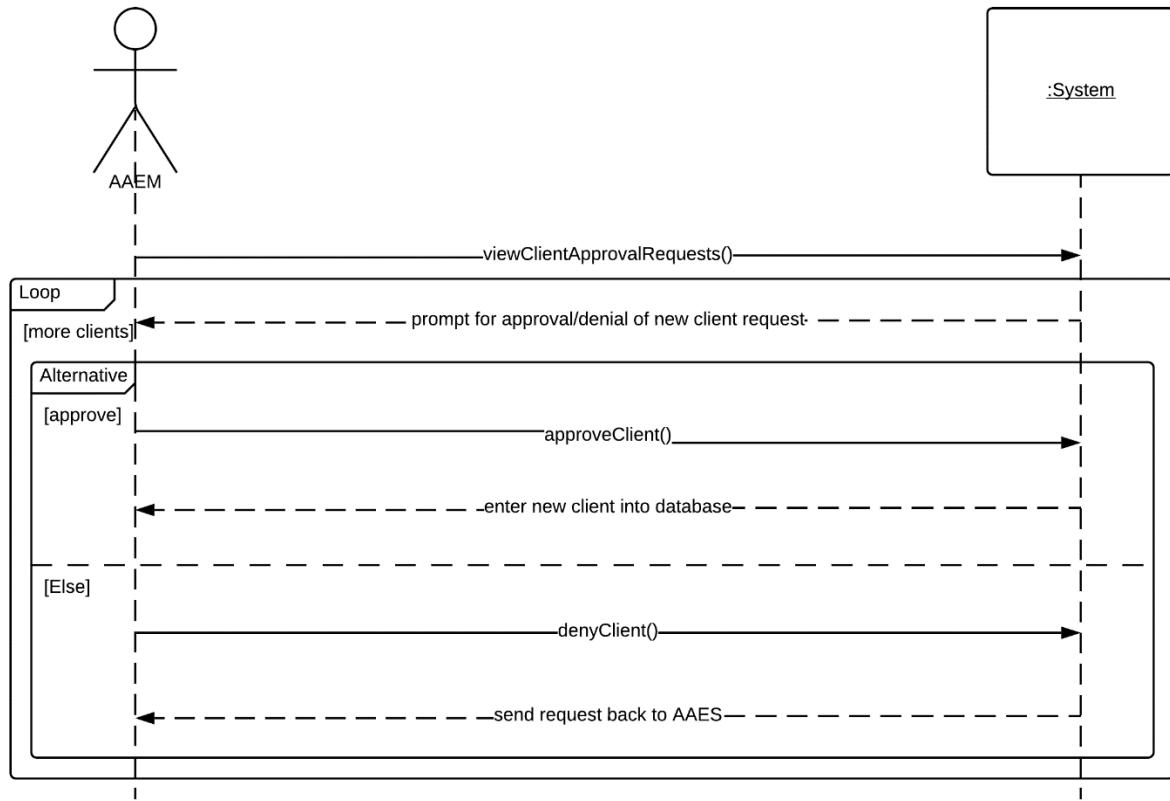
## Approve New Client Creation

Righteous Row



## Approve New Client Creation

Righteous Row



## Use Case #3: Review Existing Client

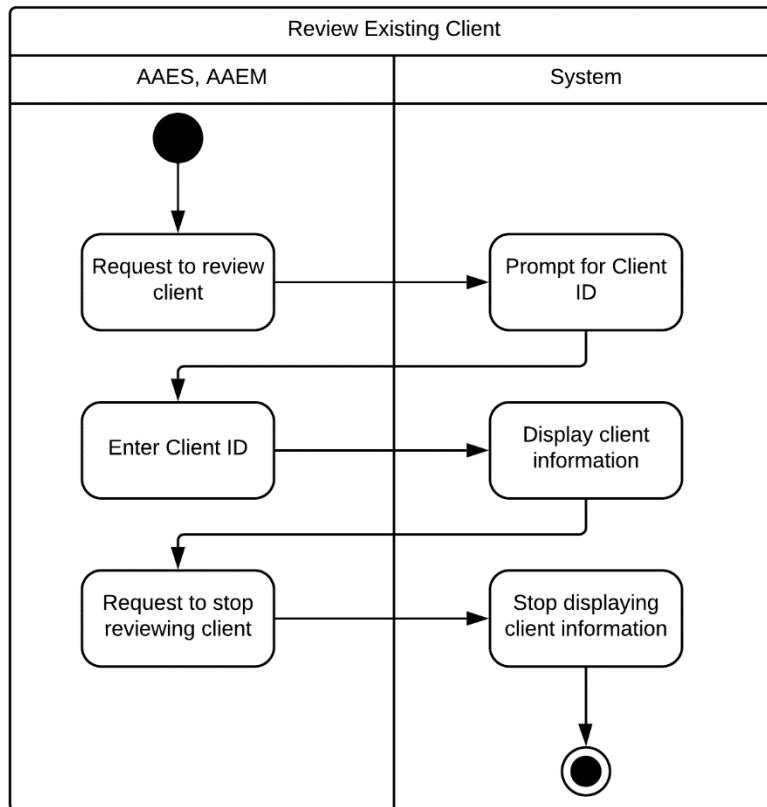
<b>Use Case Name</b>	Review Existing Client	
<b>Scenario</b>	Review an existing client's information	
<b>Triggering Event</b>	An employee wishes to view a client's information for a business purpose	
<b>Brief Description</b>	An existing client's information is displayed	
<b>Actors</b>	AAES, AAEM	
<b>Related Use Cases</b>	Create New Client, Update Existing Client	
<b>Stakeholders</b>	Clients	
<b>Preconditions</b>	The client exists within the database	
<b>Postconditions</b>	The client's information is left unchanged	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to review a client's information	1.1 System prompts user to enter the Client ID for the client who's information they wish to view
	2. User inputs the Client ID	2.1 System displays the existing client's information
	3. User indicates desire to stop reviewing the client's information	3.1 System stops displaying the client's information
<b>Exception Conditions</b>	2.1 Client ID is invalid	

As an **AAES or AAEM**, I want to **review an existing client's information** so that **I can carry out business tasks involving their information**

- The client's information is displayed
- The client's information is left unchanged

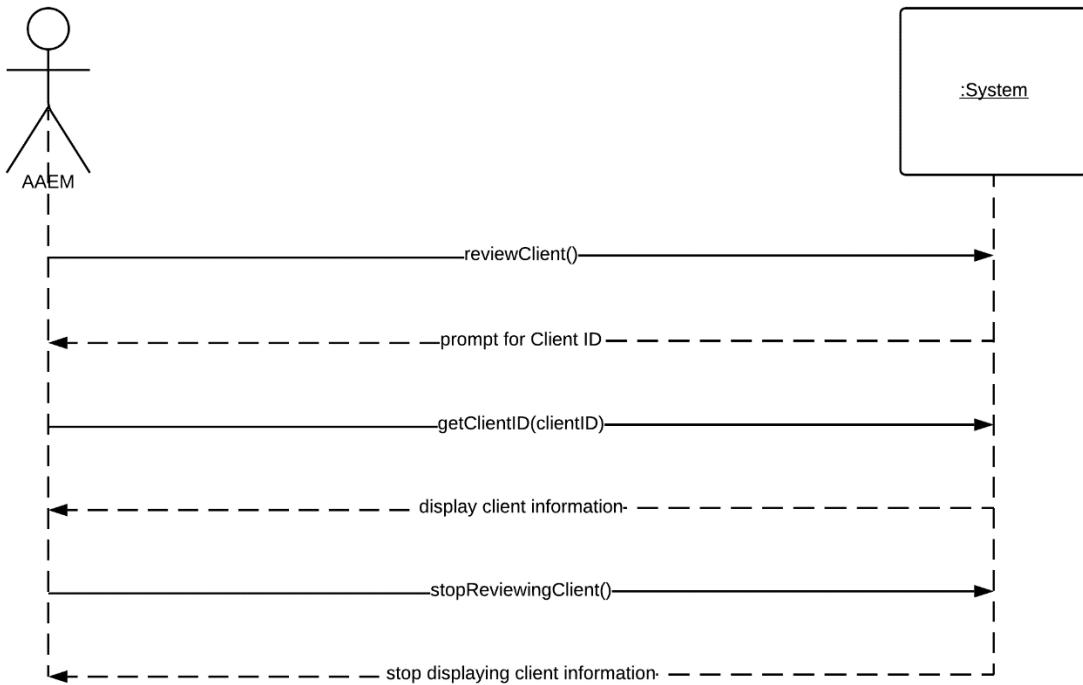
## Review Existing Client

Righteous Row



## Review Existing Client

Righteous Row



## Use Case #4: Update Existing Client

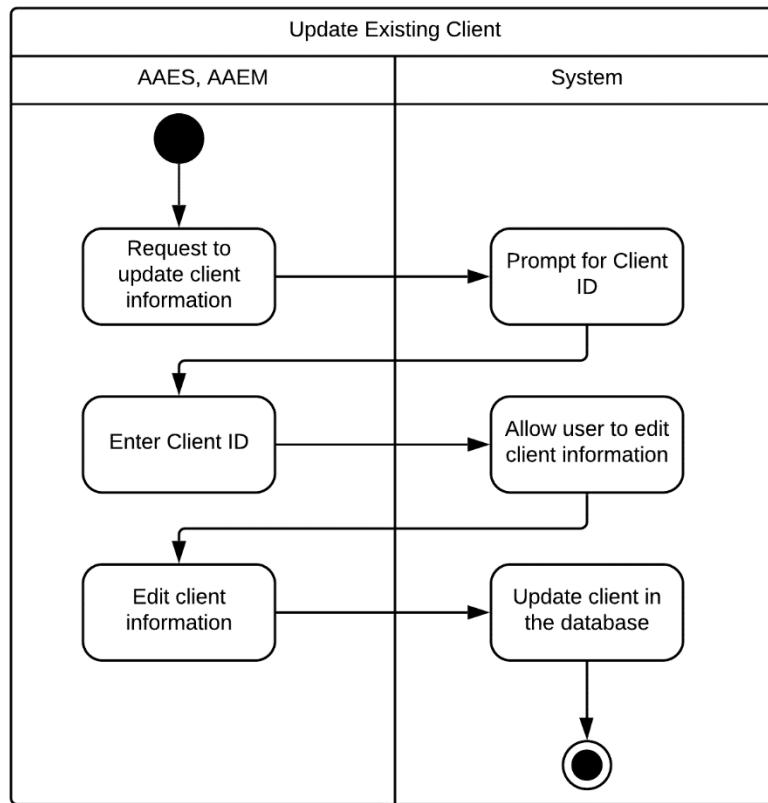
<b>Use Case Name</b>	Update Existing Client	
<b>Scenario</b>	Update an existing client's information	
<b>Triggering Event</b>	A client informs an employee that part of their information has changed	
<b>Brief Description</b>	An existing client's information is updated	
<b>Actors</b>	AAES, AAEM	
<b>Related Use Cases</b>	Create New Client, Review Existing Client	
<b>Stakeholders</b>	Client, AAES, AAEM, WSS, WSM	
<b>Preconditions</b>	The client exists within the system	
<b>Postconditions</b>	The client's information is changed according to the client's request	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to update an existing client's information	1.1 System prompts user to enter the Client ID for the client who's information they wish to update
	2. User inputs Client ID	2.1 System displays clients information and allows user to edit it
	3. User changes information and indicates desire to save changes	3.1 System prompts user to review their changes
	4. User reviews the changes and submits them	4.1 System updates the client's information in the database
<b>Exception Conditions</b>	2.1 Client ID is invalid	

As an **AAES or AAEM**, I want to **update an existing client's information** so that **DSI has their most recent address, contact information, etc.**

- The client's information is changed in the system

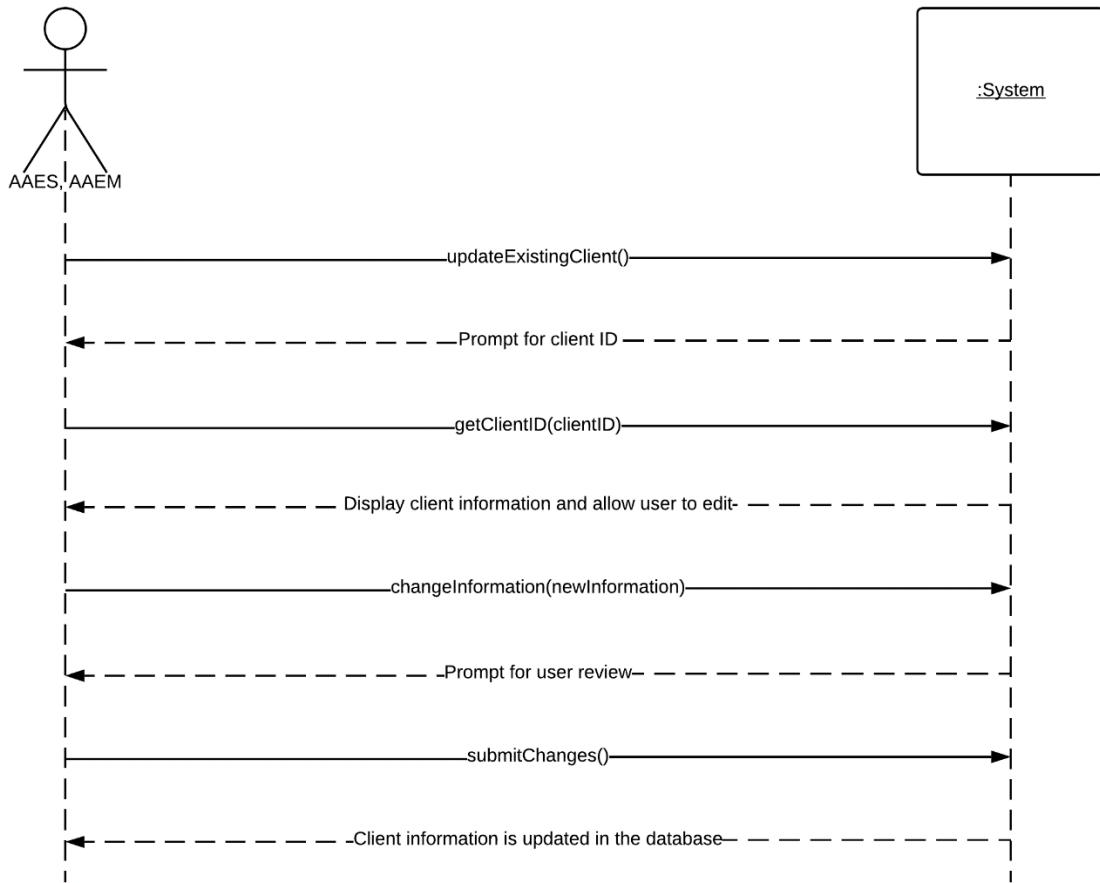
## Update Existing Client

Righteous Row



## Update Existing Client

Righteous Row



## Use Case #5: Delete Existing Client

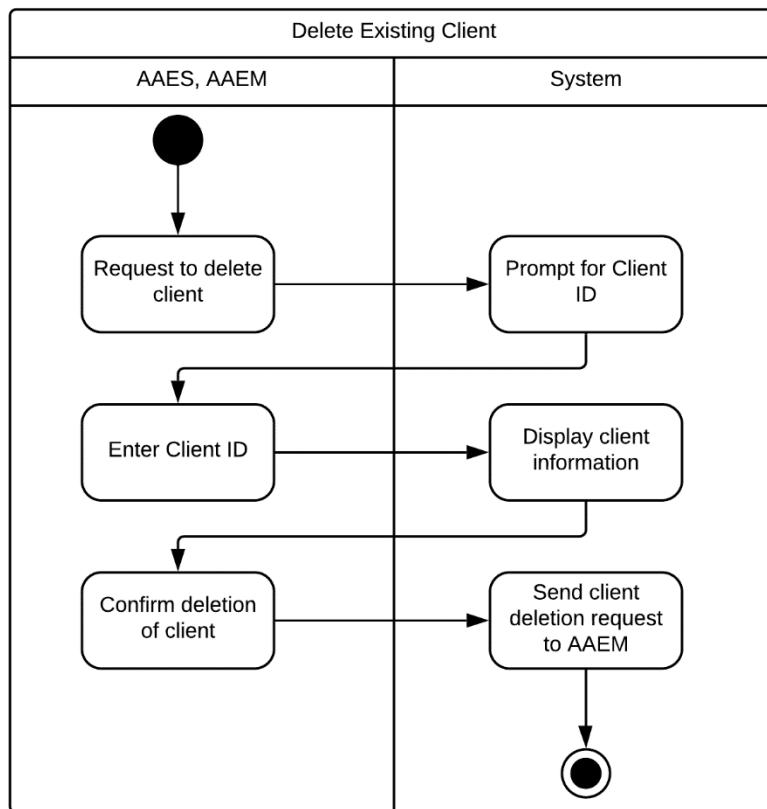
<b>Use Case Name</b>	Delete Existing Client	
<b>Scenario</b>	Delete an existing client from the database	
<b>Triggering Event</b>	A client no longer wishes to do business with DSI	
<b>Brief Description</b>	An existing client is deleted from the database	
<b>Actors</b>	AAES, AAEM	
<b>Related Use Cases</b>	Approve Client Deletion	
<b>Stakeholders</b>	Client	
<b>Preconditions</b>	The client exists within the database	
<b>Postconditions</b>	The request for the client to be deleted is sent to the AAEM	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to delete a client from the database	1.1 System prompts the user to enter the Client ID of the client to be deleted
	2. User inputs the Client ID of the client they wish to delete from the database	2.1 System pulls up the clients information and asks the user to review and confirm that they wish to delete the client from the database
	3. User confirms that they wish to delete the client from the database	3.1 System sends a request for the client to be deleted to the AAEM
<b>Exception Conditions</b>	2.1 Client ID is invalid	

As an **AAES or AAEM**, I want to **deleting an existing client** so that **DSI's records contain only present clients**

- A request for approval of the deletion is sent to the AAEM

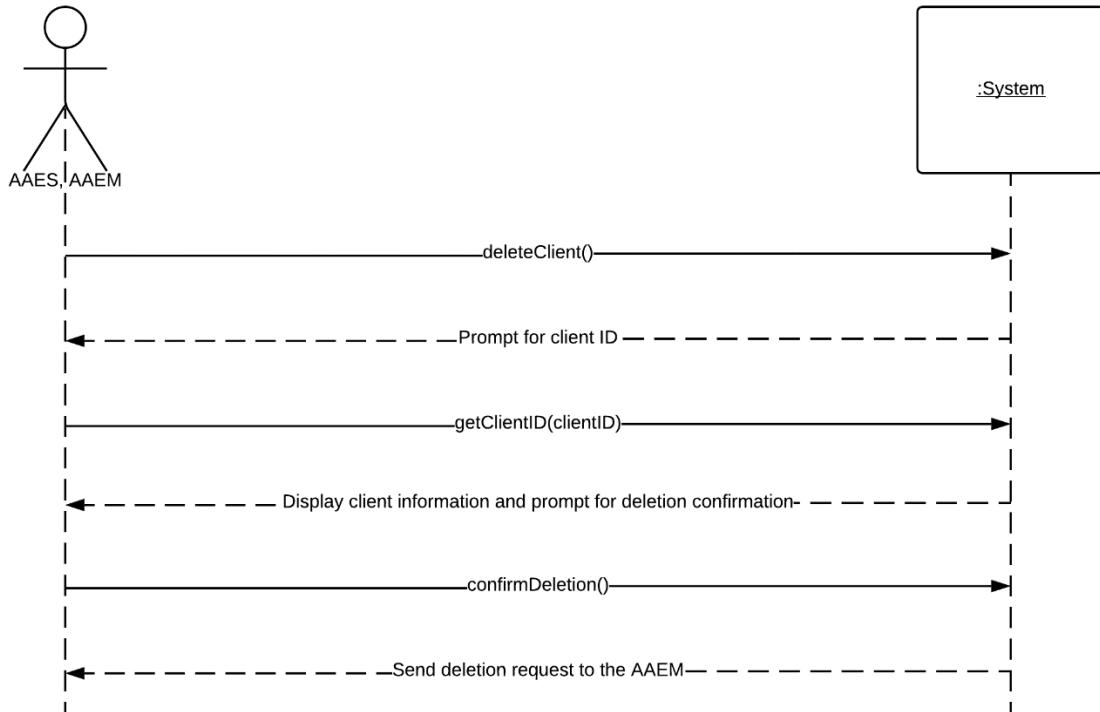
## Delete Existing Client

Righteous Row



## Delete Existing Client

Righteous Row



## Use Case #6: Approve Client Deletion

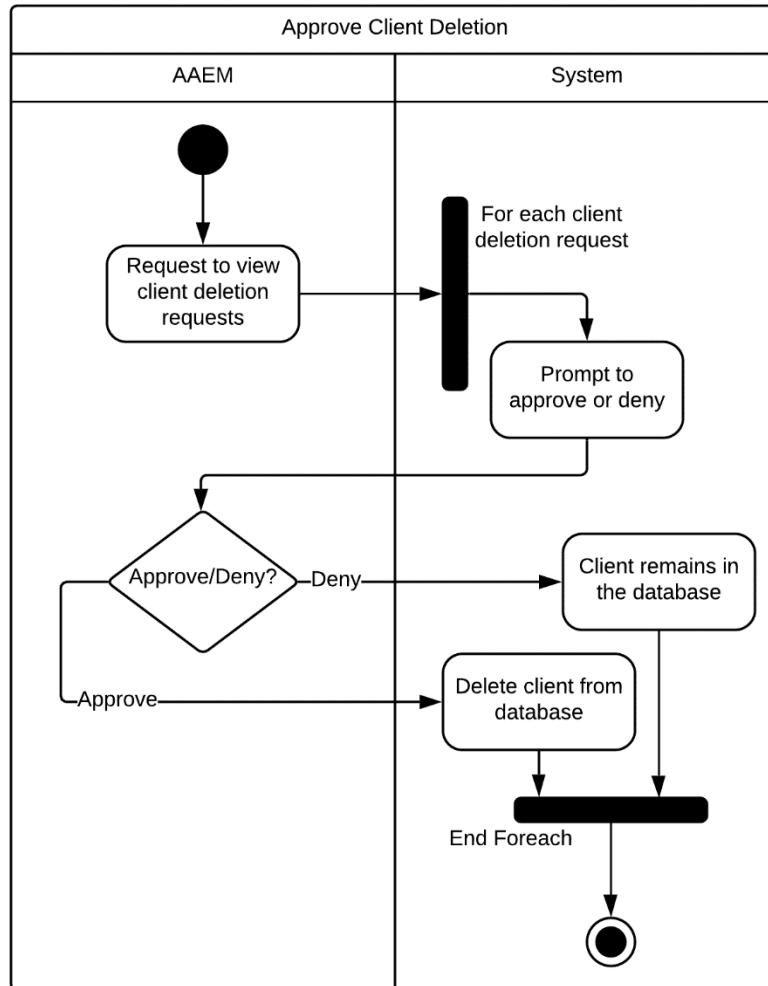
<b>Use Case Name</b>	Approve Client Deletion									
<b>Scenario</b>	Approve the request for a client to be deleted from the database									
<b>Triggering Event</b>	A request to delete a client from the database is sent to the AAEM									
<b>Brief Description</b>	The deletion of an existing client is reviewed and approved/denied									
<b>Actors</b>	AAEM									
<b>Related Use Cases</b>	Delete Existing Client									
<b>Stakeholders</b>	Client, AAES									
<b>Preconditions</b>	A request to delete a client from the database has been sent to the AAEM									
<b>Postconditions</b>	If the request is approved, the user is deleted from the database If the request is denied, the user remains in the database									
<b>Flow of Activities</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;"><b>User</b></th> <th style="text-align: center;"><b>System</b></th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to respond to client deletion requests</td> <td>1.1 System displays list of new client deletion requests that are pending manager approval 1.2 System prompts user to review and approve/deny the deletion of each client</td> </tr> <tr> <td>2. User reviews the information of each request, and marks them as either approved or denied</td> <td>2.1 System displays list of requests that the user has chosen to approve and/or deny, and prompts the user to review their choices</td> </tr> <tr> <td>3. User reviews and submits choices</td> <td>3.1 System completes deletion of clients from database</td> </tr> </tbody> </table>	<b>User</b>	<b>System</b>	1. User indicates desire to respond to client deletion requests	1.1 System displays list of new client deletion requests that are pending manager approval 1.2 System prompts user to review and approve/deny the deletion of each client	2. User reviews the information of each request, and marks them as either approved or denied	2.1 System displays list of requests that the user has chosen to approve and/or deny, and prompts the user to review their choices	3. User reviews and submits choices	3.1 System completes deletion of clients from database	
<b>User</b>	<b>System</b>									
1. User indicates desire to respond to client deletion requests	1.1 System displays list of new client deletion requests that are pending manager approval 1.2 System prompts user to review and approve/deny the deletion of each client									
2. User reviews the information of each request, and marks them as either approved or denied	2.1 System displays list of requests that the user has chosen to approve and/or deny, and prompts the user to review their choices									
3. User reviews and submits choices	3.1 System completes deletion of clients from database									
<b>Exception Conditions</b>	1.1 There are no active client deletion requests									

As an **AEEM**, I want to **approve client deletion requests** so that **clients can be removed from the database**

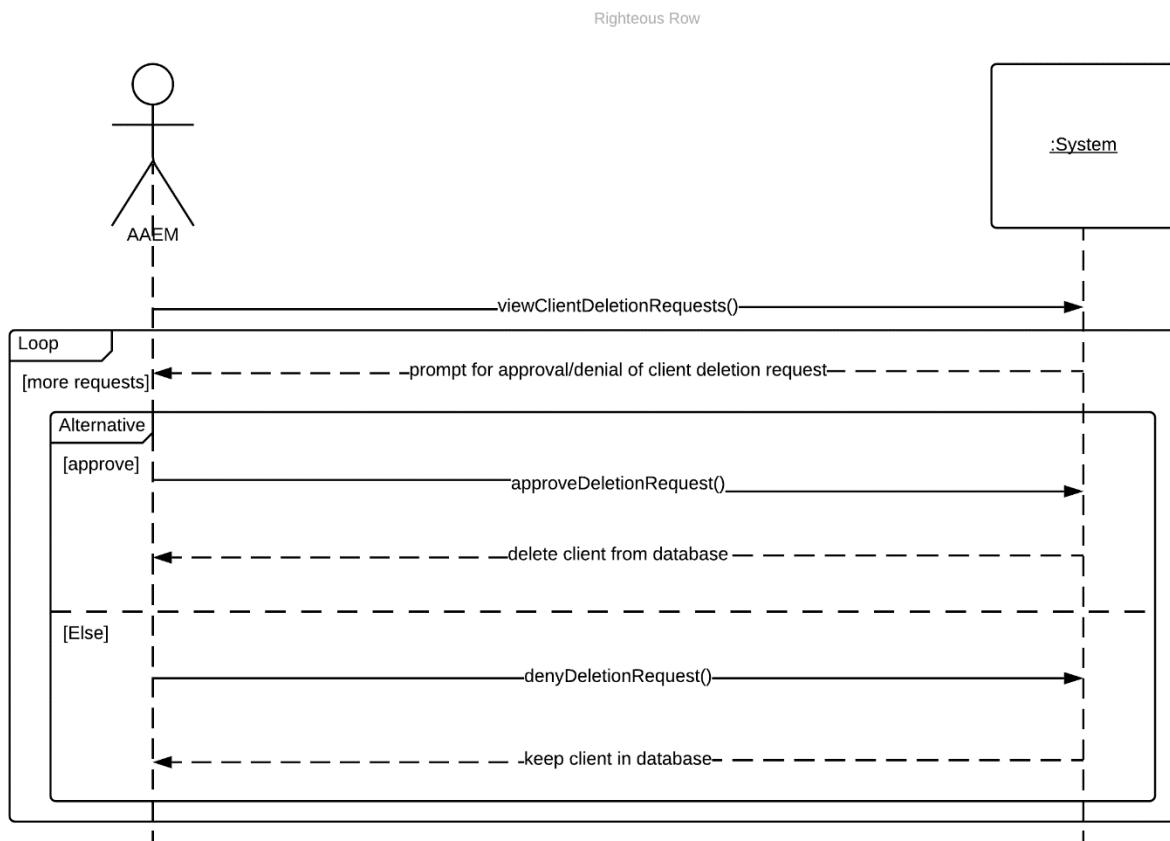
- Client deletion request is sent and approved by the AAEM
- The client's information is deleted from the database

## Approve Client Deletion

Righteous Row



## Approve Client Deletion



## Use Case #7: Assign Staff to Shifts

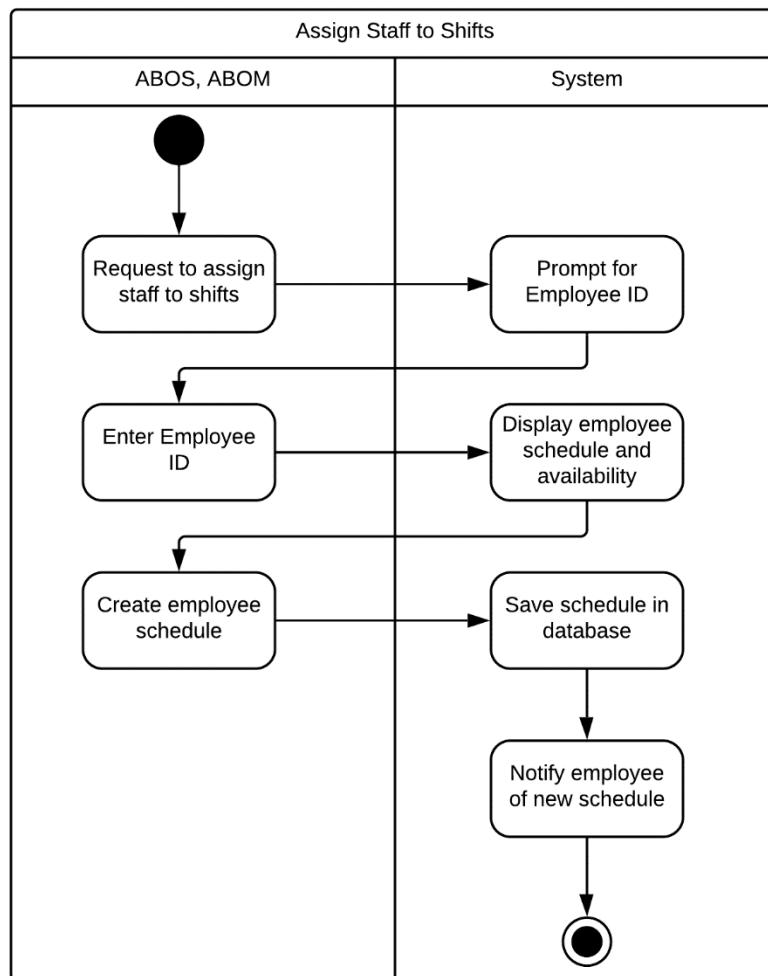
<b>Use Case Name</b>	Assign Staff to Shifts	
<b>Scenario</b>	Assign employees of all departments to shifts	
<b>Triggering Event</b>	Employee schedules must be made for the upcoming two weeks	
<b>Brief Description</b>	Warehouse and executive employees are scheduled to shifts for two week periods	
<b>Actors</b>	ABOS, ABOM	
<b>Related Use Cases</b>	Review Existing Staff Information	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	The employees exist within the system	
<b>Postconditions</b>	The employees can view their schedules for the next two weeks	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to assign staff to shifts  2. User inputs the Employee ID  3. User creates the schedule and indicates desire to submit it when finished  4. User reviews the schedule and submits it	1.1 System prompts user to input the Employee ID for the staff member they wish to assign shifts to  2.1 System displays the next two week period in which the employee will work, along with their availability and requested time off 2.2 System allows user to create the employee schedule for the two week period  3.1 System prompts user to review the schedule  4.1 System saves the schedule 4.2 System notifies the employee of their available schedule
<b>Exception Conditions</b>	2.1 Employee ID is invalid	

As an **ABOS or ABOM**, I want to **assign staff to shifts** so that **they have their work schedule**

- Full-time employees are scheduled for 40 hours per week
- Part-time employees are scheduled for no more than 25 hours per week
- Employees receive their schedules via the Client Information/Business Operations Subsystem

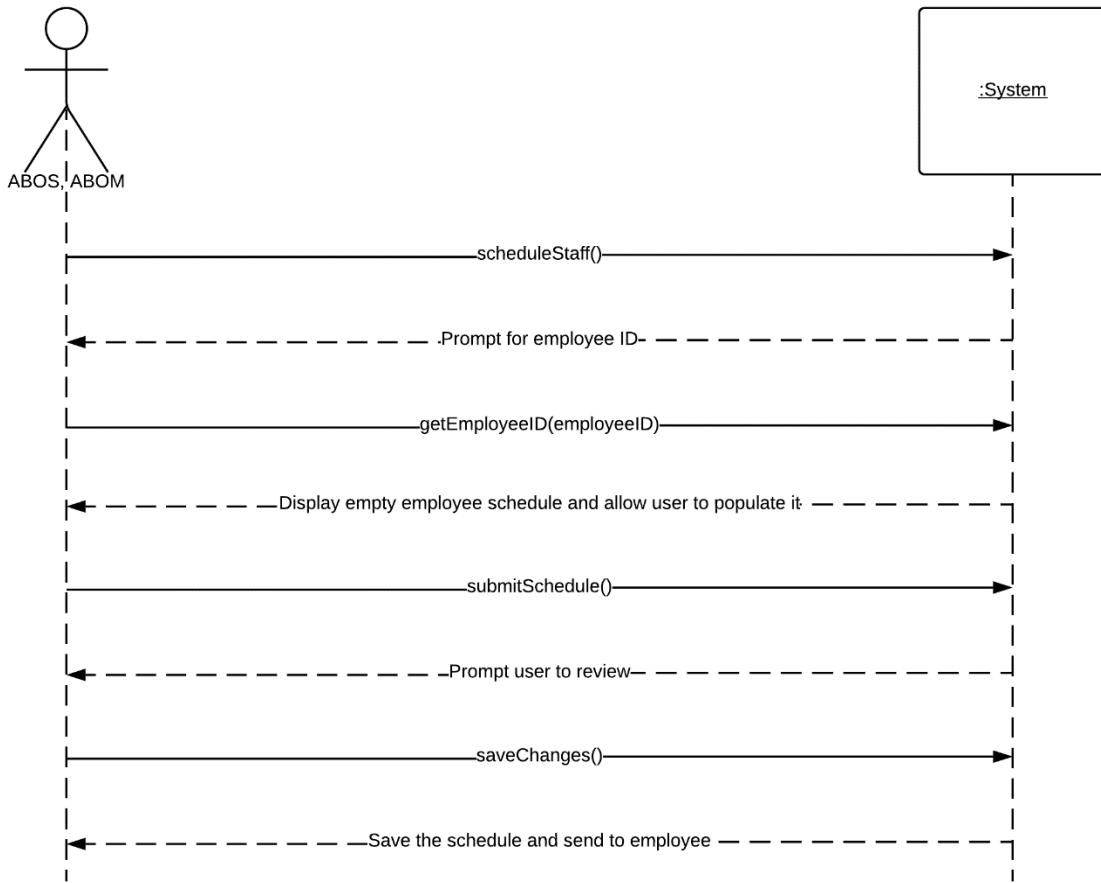
## Assign Staff to Shifts

Righteous Row



## Assign Staff to Shifts

Righteous Row



## Use Case #8: Create New Staff

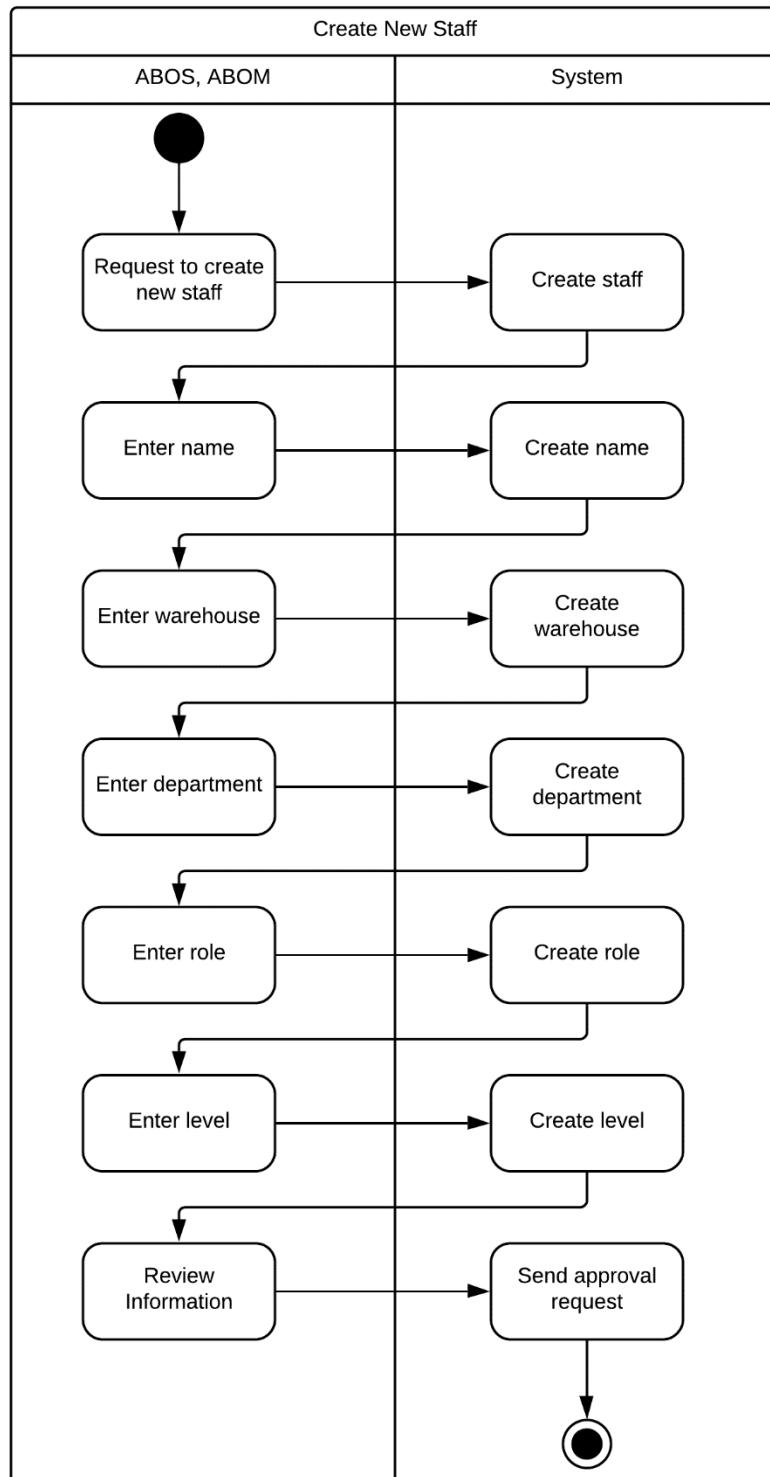
<b>Use Case Name</b>	Create New Staff	
<b>Scenario</b>	Create a new employee in the database	
<b>Triggering Event</b>	A new staff member is hired	
<b>Brief Description</b>	A new employee's information is added to the database	
<b>Actors</b>	ABOS, ABOM	
<b>Related Use Cases</b>	Approve New Hires	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	The employee has been hired by DSI	
<b>Postconditions</b>	A request to enter the new employee into the database is sent to the ABOM	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to create a new staff member in the database	1.1 System creates new staff member and generates an Employee ID 1.2 System prompts user to input the name of the employee
	2. User inputs employee's name	2.1 System adds employee name to the database entry 2.2 System prompts user to select the warehouse at which they work
	3. User inputs employee's warehouse	3.1 System adds warehouse to the database entry 3.2 System prompts user to select the department in which the new employee has been hired
	4. User inputs employee's department	4.1 System adds department to the database entry 4.2 System prompts user to select the role for which the new employee has been hired
	5. User inputs employee's role	5.1 System adds role to the database entry 5.2 System prompts user to select the level at which the employee has been hired
	6. User inputs employee's level	6.1 System adds employee level to the database entry 6.2 System prompts user to review the information of the new employee
	7. User reviews and submits new staff information	7.1 System sends a request to approve the new staff to the ABOM
<b>Exception Conditions</b>	3.1 Warehouse is invalid 4.1 Department is invalid 5.1 Role is invalid 6.1 Level is invalid	

As an **ABOS or ABOM**, I want to **create new staff** so that **DSI can hire new employees**

- All staff information is recorded
- A request for approval of the creation is sent to the ABOM

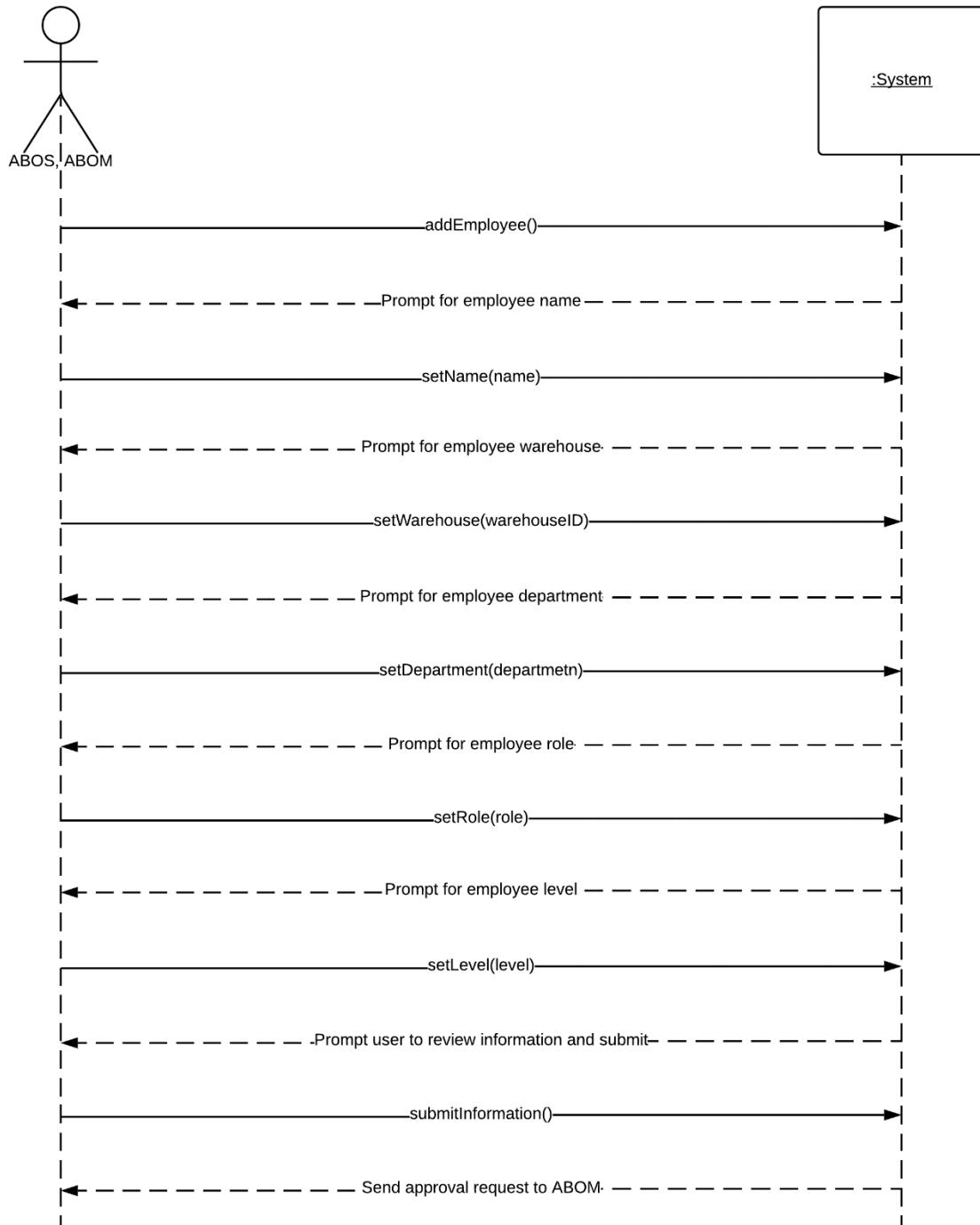
## Create New Staff

Righteous Row



## Create New Staff

Righteous Row



## Use Case #9: Approve New Hires

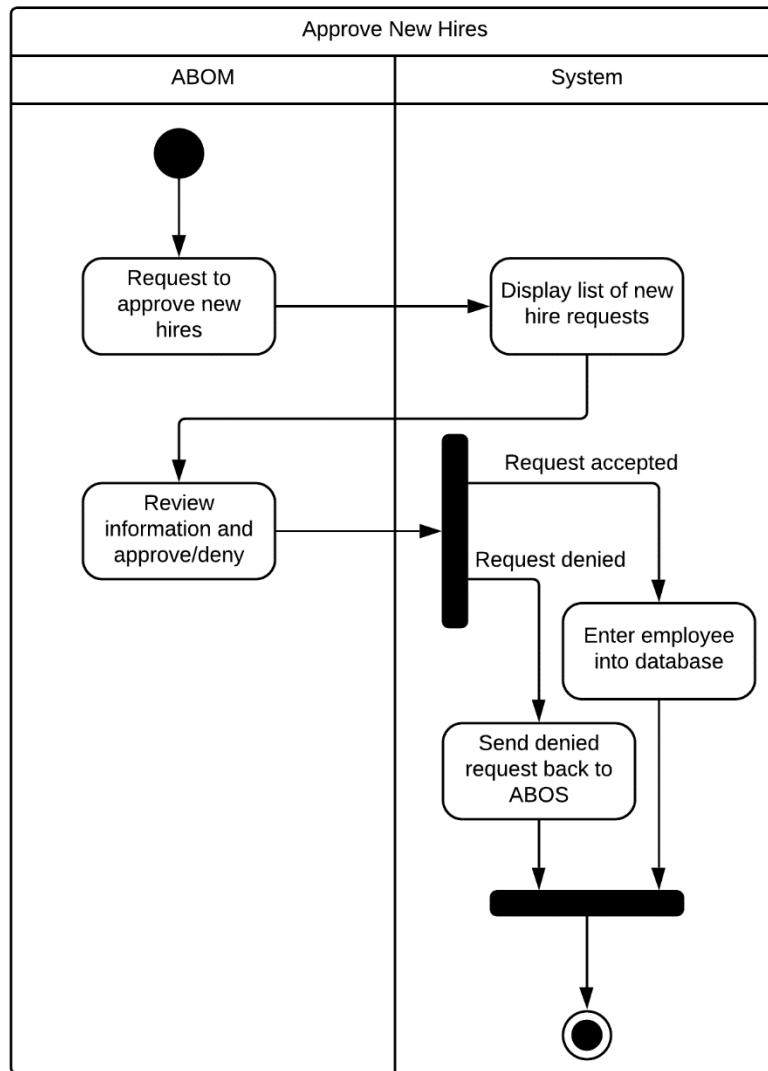
<b>Use Case Name</b>	Approve New Hires	
<b>Scenario</b>	The creation of a new employee is approved	
<b>Triggering Event</b>	A new employee is created in the database and must be approved by the ABOM	
<b>Brief Description</b>	A new employee's information is reviewed and approved/denied	
<b>Actors</b>	ABOM	
<b>Related Use Cases</b>	Create New Staff	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	A new employee has been created and sent to the ABOM for approval	
<b>Postconditions</b>	If approved, The new employee is entered into the database If denied, the ABOS is notified	
<b>Flow of Activities</b>	<b>User</b> 1. User indicates desire to approve any new employees  2. User reviews the information of each new employee, and marks them as either approved or denied  3. User reviews and submits choices	<b>System</b> 1.1 System displays list of new employees that are pending manager approval 1.2 System prompts user to review and approve/deny the creation of each new employee  2.1 System displays list of the new employees that the user has chosen to approve and/or deny, and prompts the user to review their choices  3.1 System completes entry of approved new employees into the database 3.2 System sends denied requests back to the ABOS
<b>Exception Conditions</b>	1.1 There are no new employees pending manager approval	

As an **ABOM**, I want to **approve new hires** so that **new employees are created within the system**

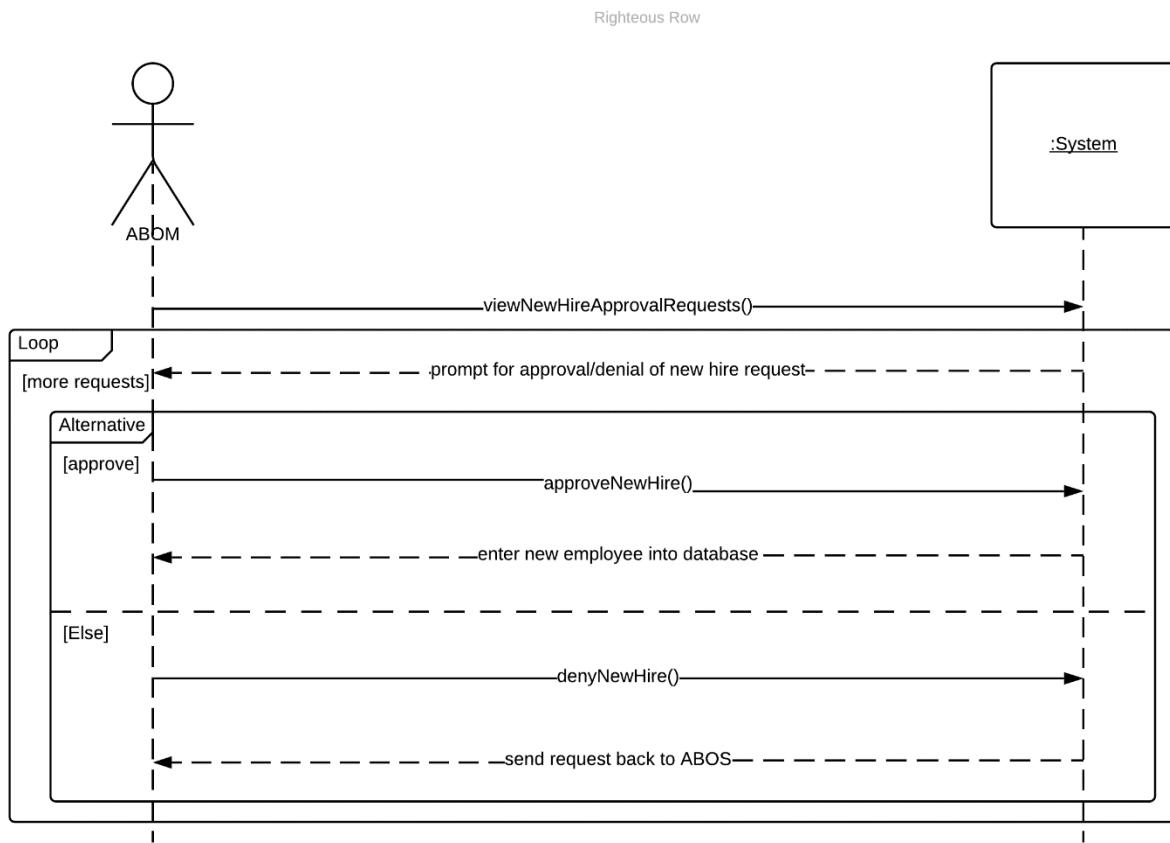
- Employee creation request is sent and approved by the ABOM
- The new employee's information is saved in the database

## Approve New Hires

Righteous Row



## Approve New Hires



## Use Case #10: Review Existing Staff Information

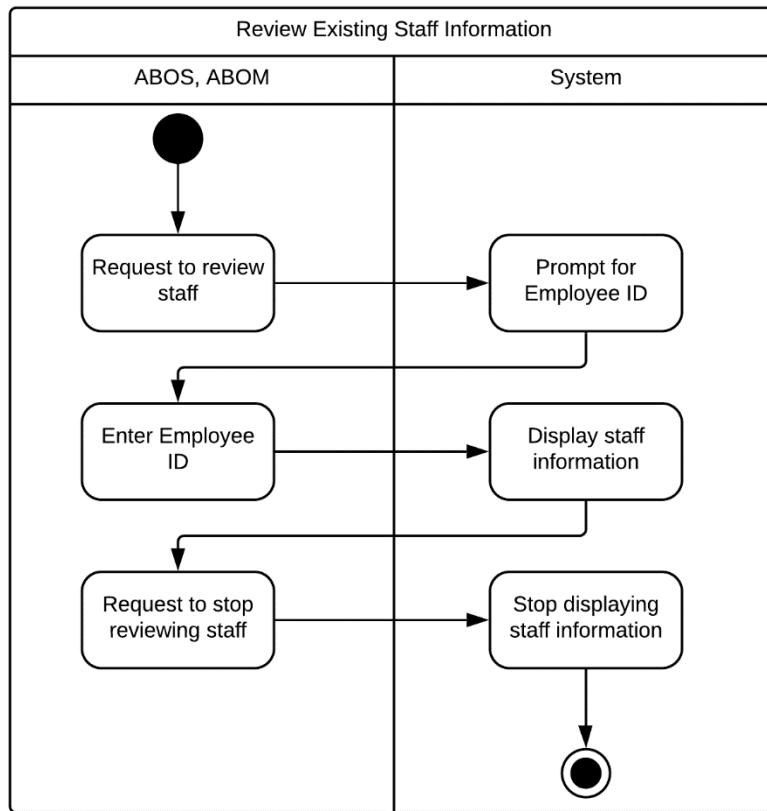
<b>Use Case Name</b>	Review Existing Staff Information	
<b>Scenario</b>	Review an existing employee's information	
<b>Triggering Event</b>	An employee wishes to view another employee's information for a business purpose	
<b>Brief Description</b>	An existing employee's information is displayed	
<b>Actors</b>	ABOS, ABOM	
<b>Related Use Cases</b>	Update Existing Staff Information	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	The employee exists within the database	
<b>Postconditions</b>	The employee's information is left unchanged	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to review a employee's information	1.1 System prompts user to enter the Employee ID for the employee who's information they wish to view
	2. User inputs the Employee ID	2.1 System displays the existing employee's information
	3. User indicates desire to stop reviewing the employee's information	3.1 System stops displaying the employee's information
<b>Exception Conditions</b>	2.1 Employee ID is invalid	

As an **ABOS or ABOM**, I want to **review an existing employee's information** so that **I can carry out business processes requiring their information**

- The employee's information is displayed
- The employee's information is left unchanged

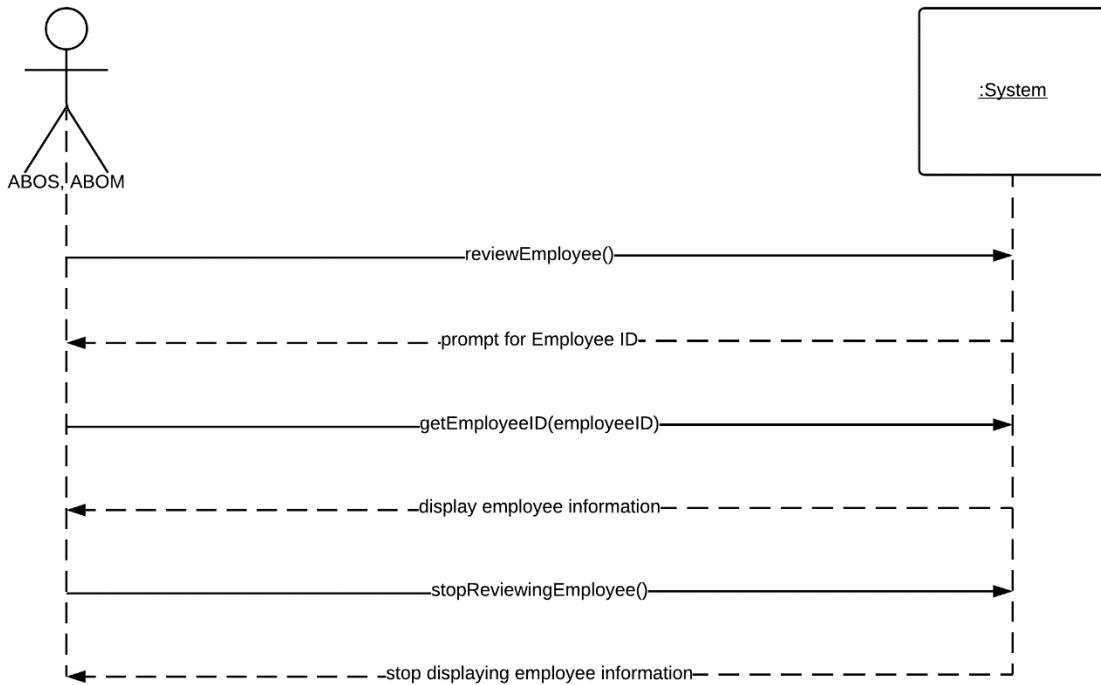
## Review Existing Staff Information

Righteous Row



## Review Existing Staff Information

Righteous Row



## Use Case #11: Update Existing Staff Information

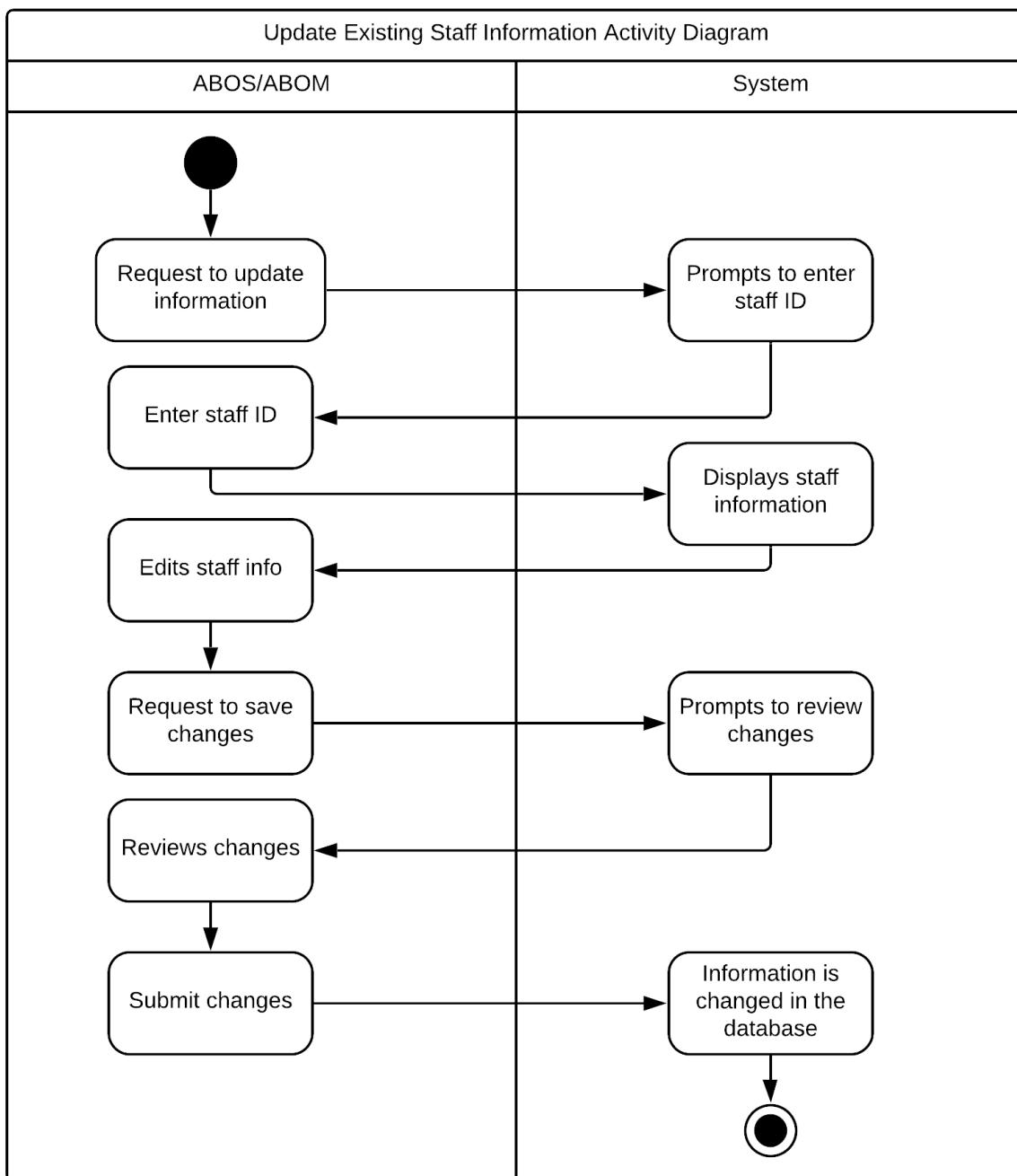
<b>Use Case Name</b>	Update Existing Staff Information	
<b>Scenario</b>	Staff information is updated in the system	
<b>Triggering Event</b>	An existing employee has information that needs to be updated	
<b>Brief Description</b>	An existing employee's information is updated	
<b>Actors</b>	ABOS, ABOM	
<b>Related Use Cases</b>	Review Existing Staff Information	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	The ABOM or ABOS has information to update existing staff information The staff member exists within the system	
<b>Postconditions</b>	The staff information is updated	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to update existing staff information	1.1 System prompts the user to enter the staff ID
	2. User inputs the Staff ID	2.1 System displays staff information and allows the user to edit it
	3. User changes the information and indicates desire to save changes	3.1 System prompts the user to review the changes
	4. User reviews the changes and submits them	4.1 System updates the staff information in the database
<b>Exception Conditions</b>	2.1 Staff ID is invalid	

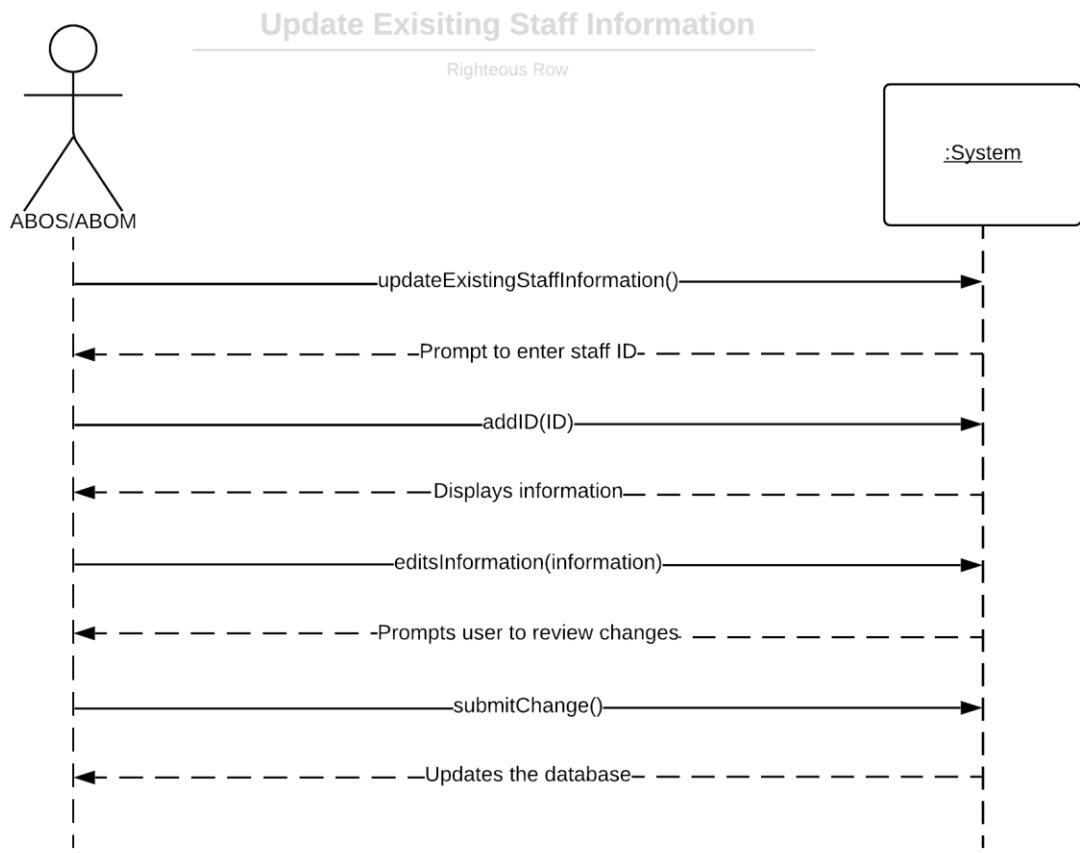
As an **ABOS** or **ABOM**, I want to **update an existing employee's information** so that **DSI has their most recent address, contact information, etc.**

- The employee's information is changed in the system

## Update Existing Staff Information

Righteous Row





## Use Case #12: Delete Fired Staff

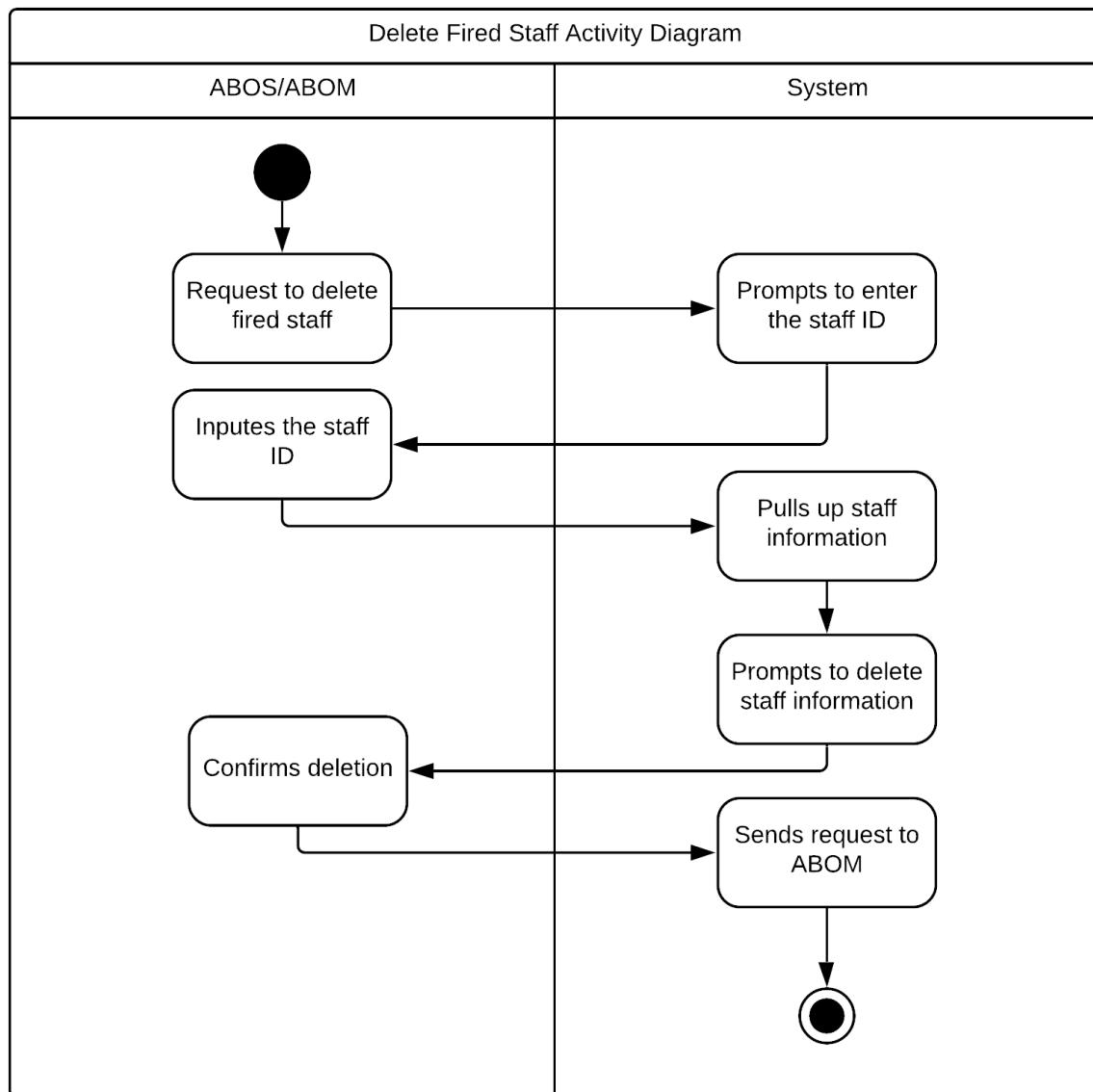
<b>Use Case Name</b>	Delete Fired Staff	
<b>Scenario</b>	Delete a fired staff member from the database	
<b>Triggering Event</b>	A staff member has been fired	
<b>Brief Description</b>	An existing employee is deleted from the database	
<b>Actors</b>	ABOS, ABOM	
<b>Related Use Cases</b>	Approve Firing Employees	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	The staff member exists within the database	
<b>Postconditions</b>	The request for the staff to be deleted is sent to the ABOM	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to delete a fired staff member 2. User inputs the ID of the fired staff member 3. User confirms that they wish to delete the fired staff member's information	1.1 System prompts the user to enter the ID of the fired staff member 2.1 System displays the fired staff member's information and asks the user to confirm that they wish to delete the fired staff member's information 3.1 System sends a request for the fired staff member's information to get deleted to the ABOM
<b>Exception Conditions</b>	2.1 Employee ID is invalid	

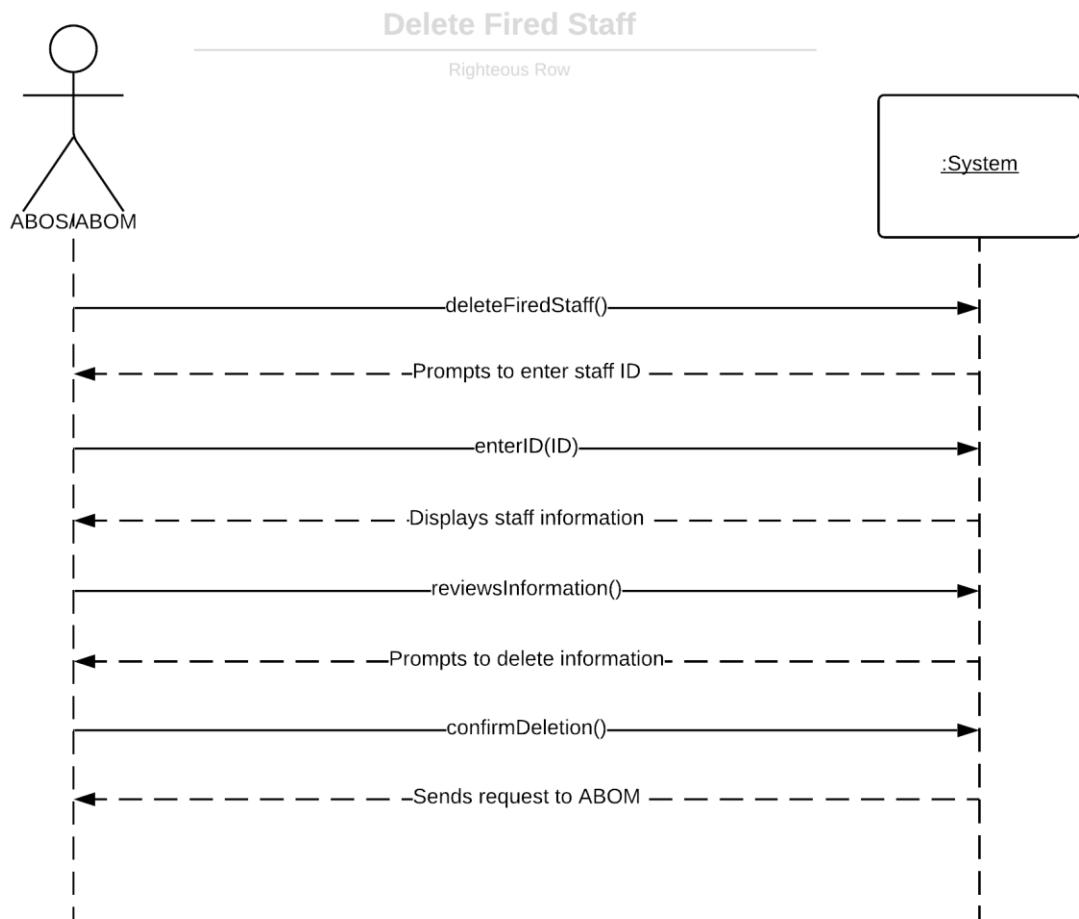
As an **ABOS or ABOM**, I want to **deleting a fired employee** so that **DSI's records contain only present employees**

- A request for approval of the deletion is sent to the ABOM

## Delete Fired Staff

Righteous Row





## Use Case #13: Approve Firing Employees

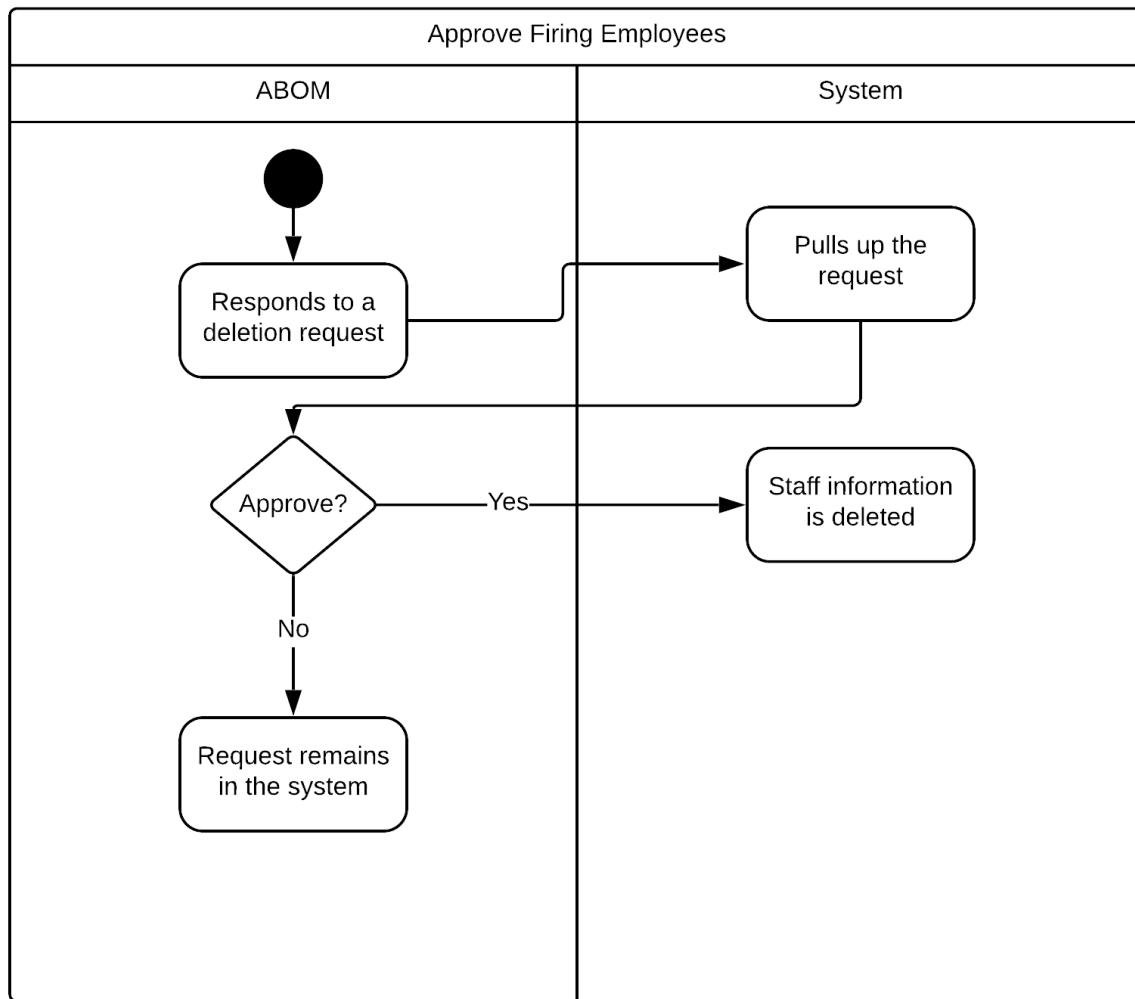
<b>Use Case Name</b>	Approve Firing Employees	
<b>Scenario</b>	Approve the request for a fired staff's information to be deleted from the database	
<b>Triggering Event</b>	A request to delete a fired staff's information from the database has been sent to the ABOM	
<b>Brief Description</b>	The deletion of an existing employee is reviewed and approved/denied	
<b>Actors</b>	ABOM	
<b>Related Use Cases</b>	Delete Fired Staff	
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM	
<b>Preconditions</b>	A request to delete a fired staff's information from the database has been sent to the ABOM	
<b>Postconditions</b>	If the request is approved, the fired staff member's information is deleted from the database If the request is denied, the fired staff member's information stays in the database	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to respond to a fired staff's deletion request  2. User approves or denies the request	1.1 System prompts the user to either approve or deny the request  2.1 If the user approves the request, the fired staff's information is deleted from the database 2.2 If the user denies the request, the fired staff's information remains in the database
<b>Exception Conditions</b>	1.1 There are no active requests to fire employees	

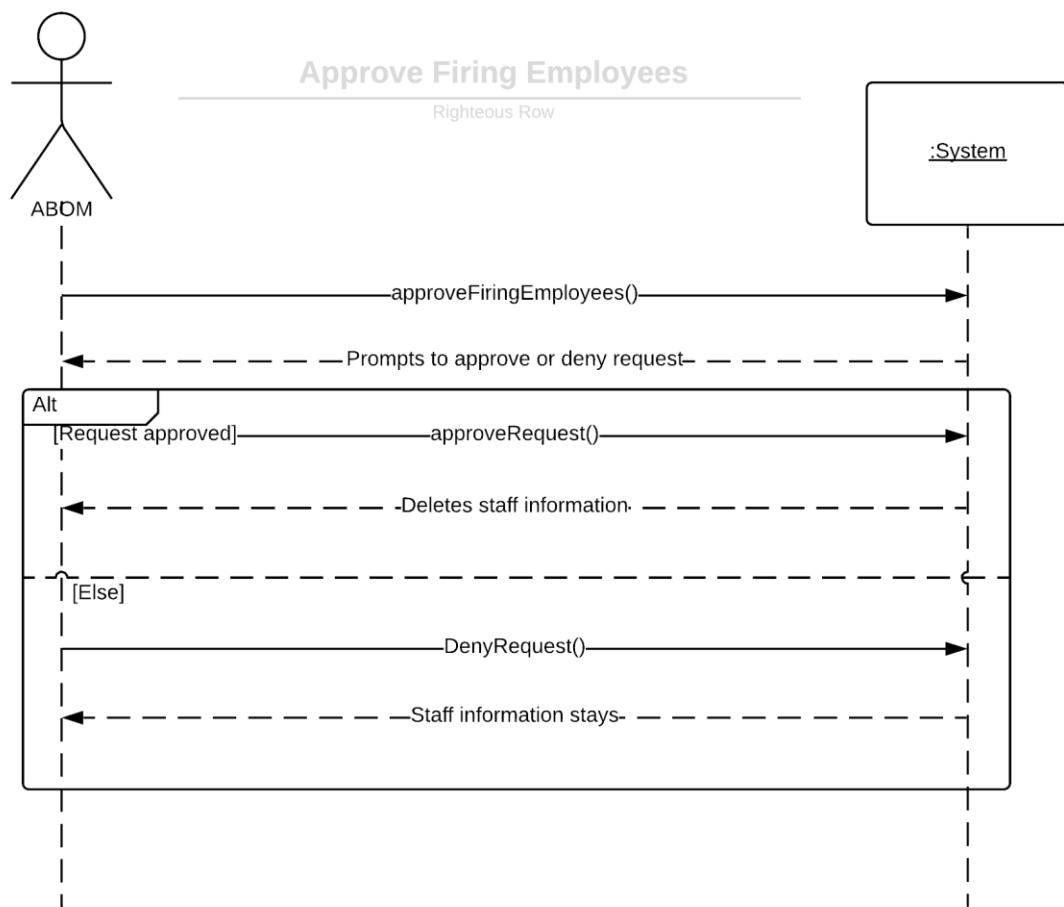
As an **ABOM**, I want to **approve employee deletion requests** so that **fired employees can be removed from the database**

- Employee deletion request is sent and approved by the ABOM
- The employee's information is deleted from the database

## Approve Firing Employees

Righteous Row





## Use Case #14: Request Pay Change

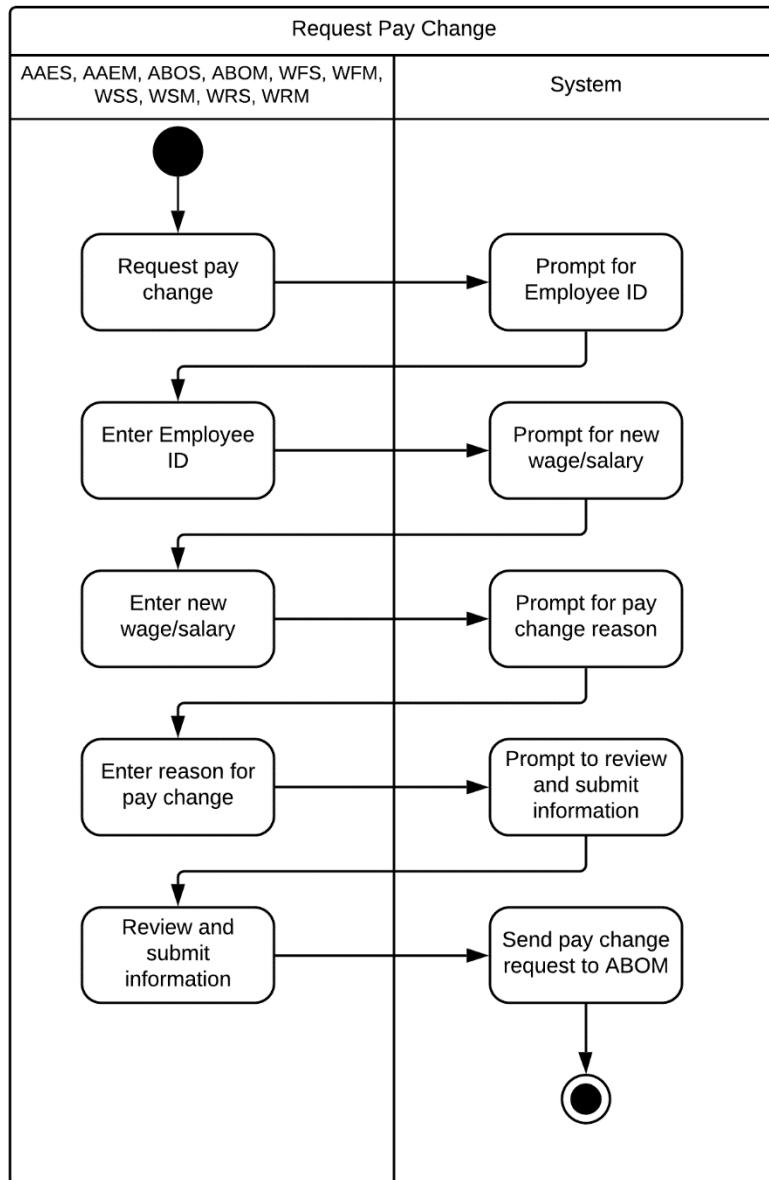
<b>Use Case Name</b>	Request Pay Change													
<b>Scenario</b>	A pay change is requested by an employee													
<b>Triggering Event</b>	An employee desires a change in their pay													
<b>Brief Description</b>	An employee requests a change in their pay													
<b>Actors</b>	AAES, AAEM, ABOS, ABOM, WFS, WFM, WSS, WSM, WRS, WRM													
<b>Related Use Cases</b>	Approve Pay Changes													
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WFS, WFM, WSS, WSM, WRS, WRM													
<b>Preconditions</b>	The employee exists within the system													
<b>Postconditions</b>	A pay change request is sent to the ABOM													
<b>Flow of Activities</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>User</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to request a pay change</td> <td>1.1 System prompts the user for their employee ID</td> </tr> <tr> <td>2. User inputs their employee ID</td> <td>2.1 System prompts user to input their desired new wage/salary</td> </tr> <tr> <td>3. User inputs their desired pay raise</td> <td>3.1 System prompts user to input reason for pay change</td> </tr> <tr> <td>4. User inputs a reason for their pay raise</td> <td>4.1 System prompts user to review and submit their information</td> </tr> <tr> <td>5. User reviews and submits their information</td> <td>5.1 System sends pay change request to ABOM</td> </tr> </tbody> </table>	User	System	1. User indicates desire to request a pay change	1.1 System prompts the user for their employee ID	2. User inputs their employee ID	2.1 System prompts user to input their desired new wage/salary	3. User inputs their desired pay raise	3.1 System prompts user to input reason for pay change	4. User inputs a reason for their pay raise	4.1 System prompts user to review and submit their information	5. User reviews and submits their information	5.1 System sends pay change request to ABOM	
User	System													
1. User indicates desire to request a pay change	1.1 System prompts the user for their employee ID													
2. User inputs their employee ID	2.1 System prompts user to input their desired new wage/salary													
3. User inputs their desired pay raise	3.1 System prompts user to input reason for pay change													
4. User inputs a reason for their pay raise	4.1 System prompts user to review and submit their information													
5. User reviews and submits their information	5.1 System sends pay change request to ABOM													
<b>Exception Conditions</b>	2.1 Employee ID is invalid 3.1 New wage/salary is more than 10% than old wage/salary 4.1 User doesn't enter description													

As an **AAES, AAEM, ABOS, ABOM, WFS, WFM, WSS, WSM, WRS, or WRM**, I want to **request a pay change** so that **my wage or salary is increased**

- A request for approval of the pay change is sent to the ABOM

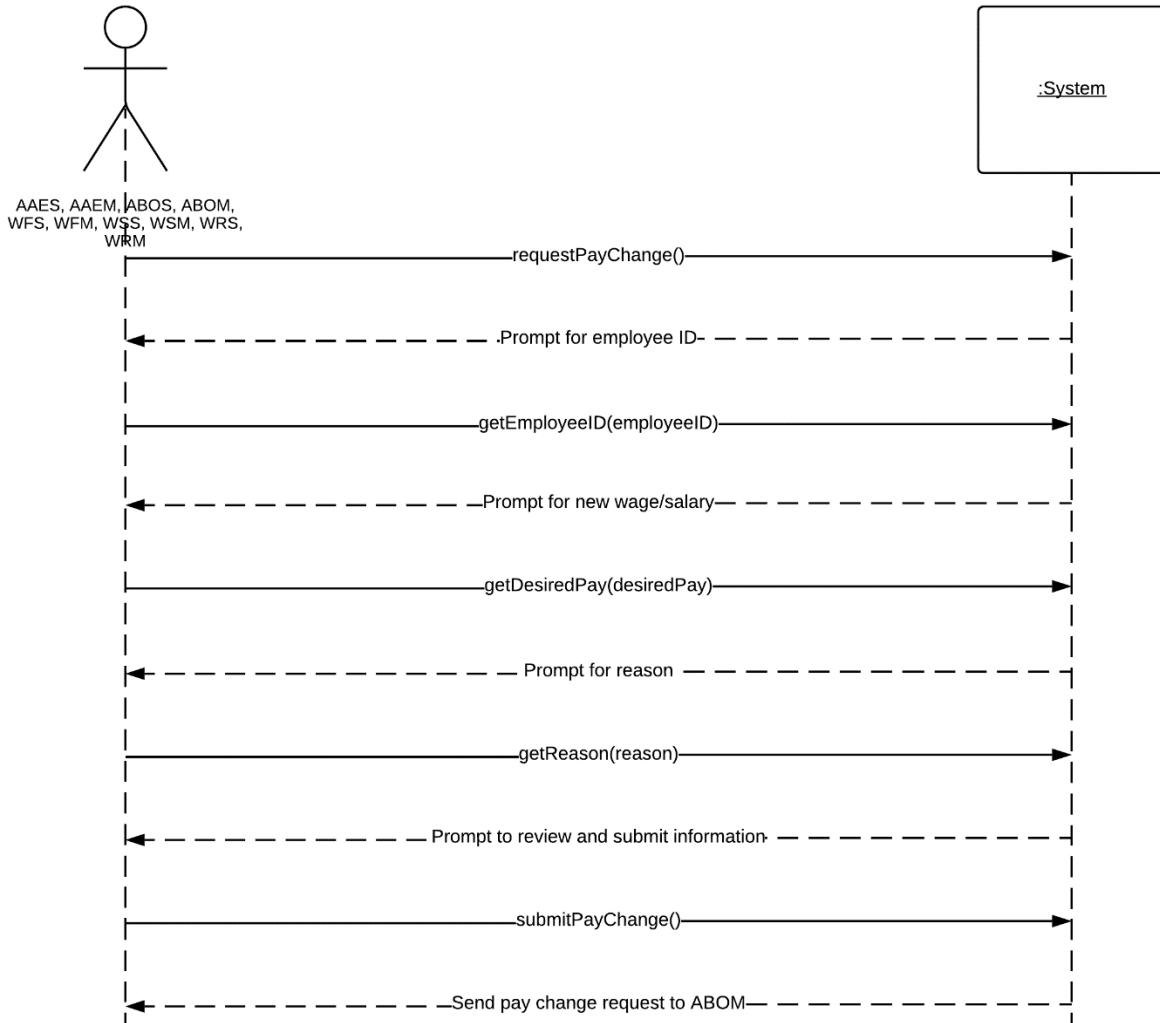
## Request Pay Change

Righteous Row



## Request Pay Change

Righteous Row



## Use Case #15: Approve Pay Changes

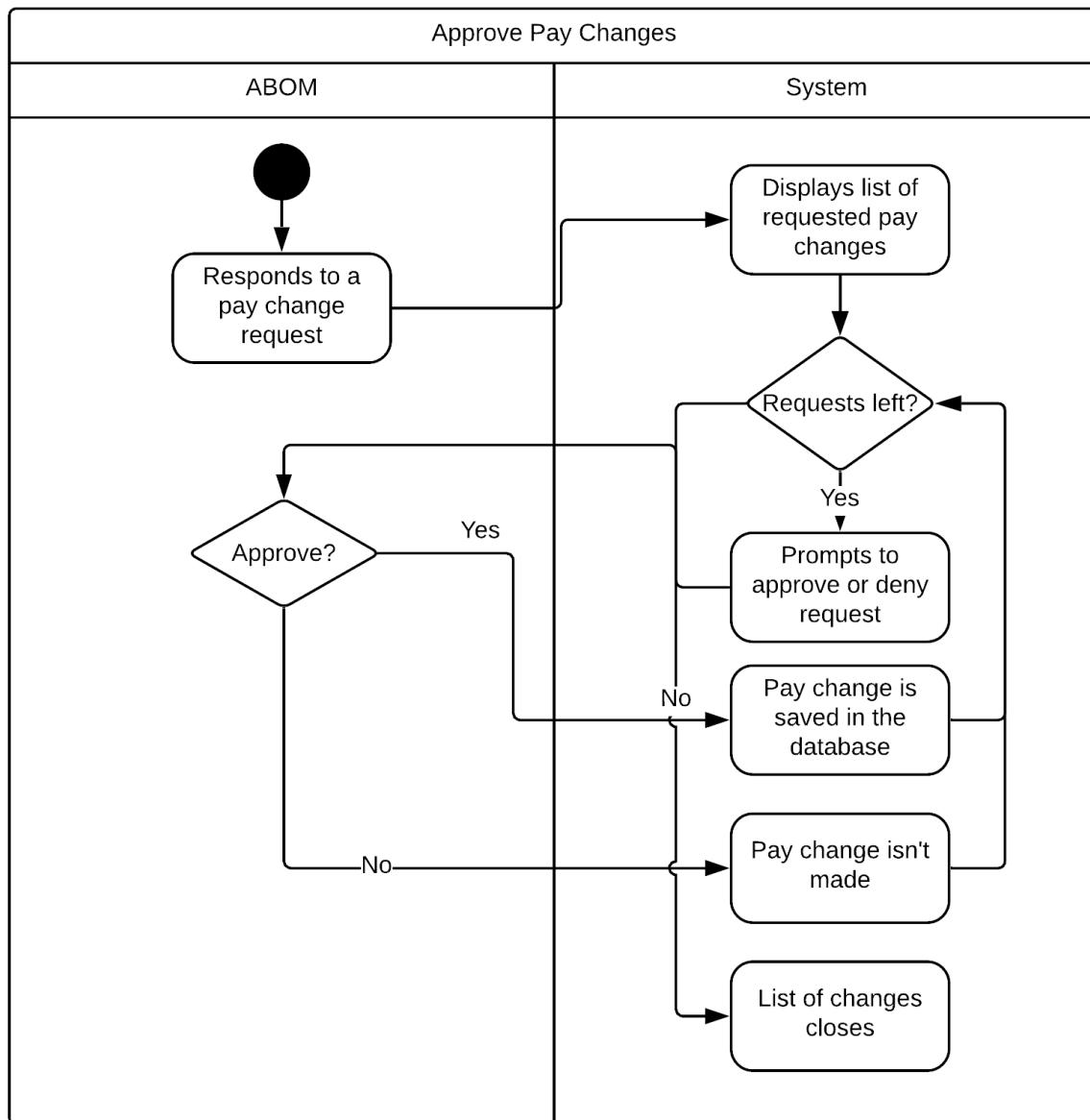
<b>Use Case Name</b>	Approve Pay Changes							
<b>Scenario</b>	Changes in pay rates are approved by the ABOM							
<b>Triggering Event</b>	A pay change request is sent to the ABOM							
<b>Brief Description</b>	A requested pay change is reviewed and approved/denied							
<b>Actors</b>	ABOM							
<b>Related Use Cases</b>	Request Pay Change							
<b>Stakeholders</b>	AAES, AAEM, ABOS, ABOM, WSS, WSM, WFS, WFM, WRS, WRM							
<b>Preconditions</b>	A pay change request has been made and sent to the ABOM for approval							
<b>Postconditions</b>	If approved, the pay change is made and put into the database If denied, the pay change is not made							
<b>Flow of Activities</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;"><b>User</b></th> <th style="text-align: center;"><b>System</b></th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to approve a pay change request</td> <td>1.1 System displays a list of requested pay changes 1.2 System prompts the user to approve or deny the pay changes</td> </tr> <tr> <td>2.1 User approves or denies the request</td> <td>2.1 If the user approves the request, the pay change is saved in the database 2.2 If the user denies the request, the pay remains the same</td> </tr> </tbody> </table>	<b>User</b>	<b>System</b>	1. User indicates desire to approve a pay change request	1.1 System displays a list of requested pay changes 1.2 System prompts the user to approve or deny the pay changes	2.1 User approves or denies the request	2.1 If the user approves the request, the pay change is saved in the database 2.2 If the user denies the request, the pay remains the same	
<b>User</b>	<b>System</b>							
1. User indicates desire to approve a pay change request	1.1 System displays a list of requested pay changes 1.2 System prompts the user to approve or deny the pay changes							
2.1 User approves or denies the request	2.1 If the user approves the request, the pay change is saved in the database 2.2 If the user denies the request, the pay remains the same							
<b>Exception Conditions</b>	1.1 There are no available pay changes to review							

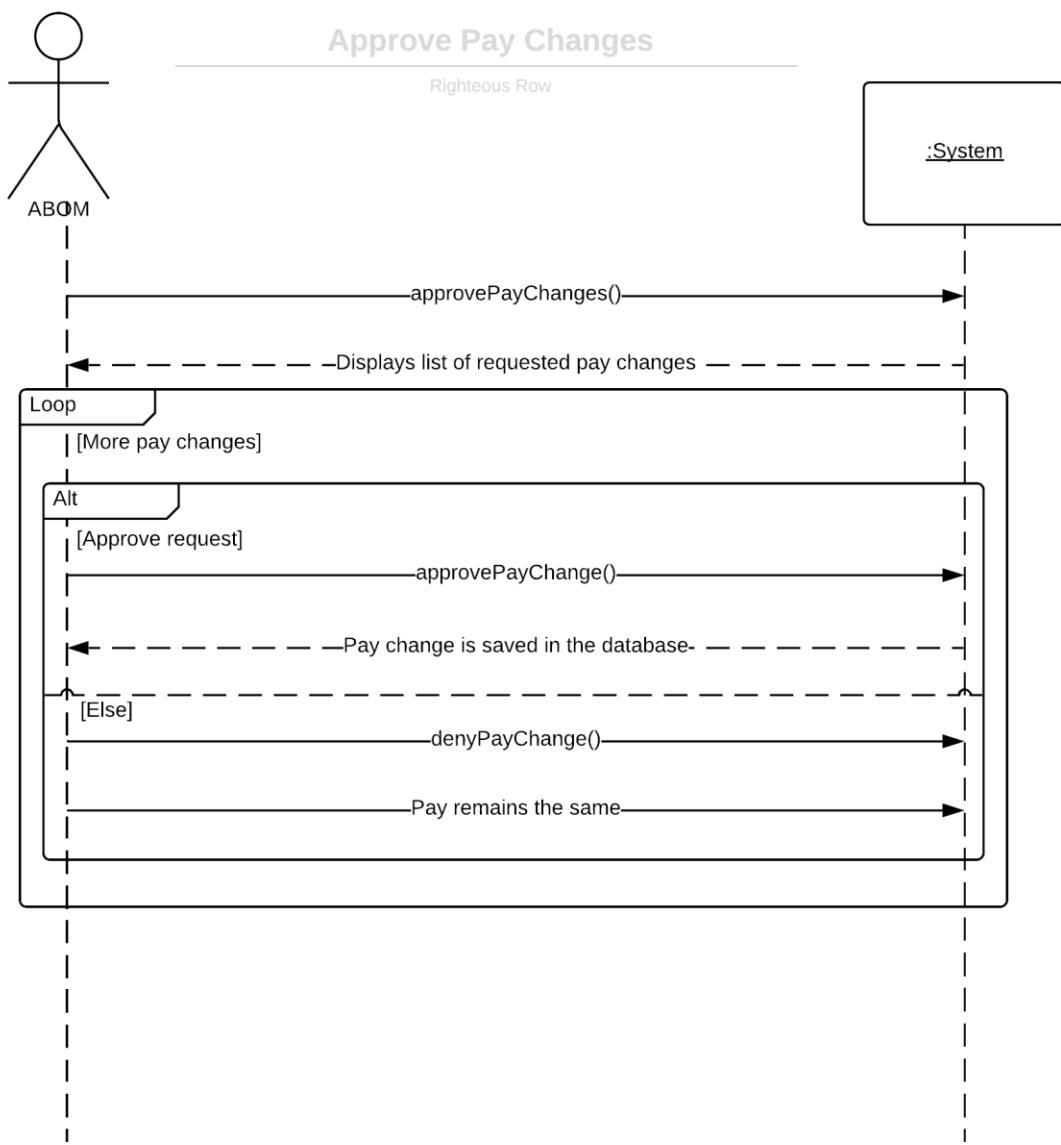
As an **ABOM**, I want to **approve pay change requests** so that **employees who request a pay change can have their wage or salary changed**

- Pay change request is sent and approved or denied by the ABOM
- The employee's wage or salary is changed in the database if the request is accepted
- The employee's wage or salary remains the same in the database if the request is denied

## Approve Pay Changes

Righteous Row





## Use Case #16: Look Up Order Information

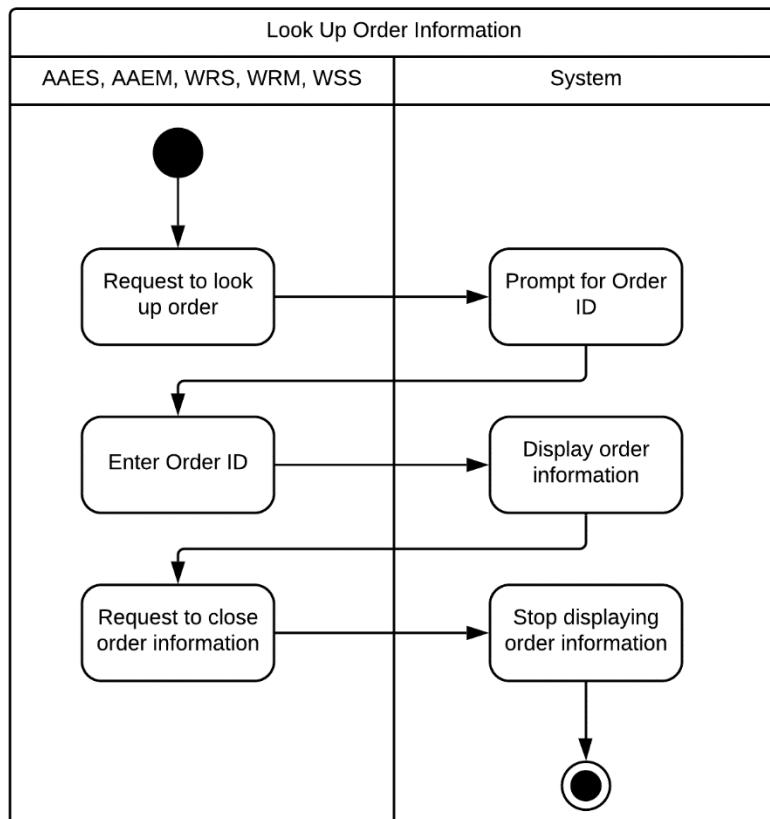
<b>Use Case Name</b>	Look Up Order Information	
<b>Scenario</b>	Look up the information of an order that a client has placed	
<b>Triggering Event</b>	An employee desires to look up an order's information	
<b>Brief Description</b>	An employee views an order by entering its Order ID in the system, and is shown the information of the order	
<b>Actors</b>	AAES, AAEM, WRS, WRM, WSM	
<b>Related Use Cases</b>	Review Existing Client	
<b>Stakeholders</b>	Clients	
<b>Preconditions</b>	The order that the user desires to look up in the system exists	
<b>Postconditions</b>	The order that the user looks up remains unchanged	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to look up an order in the database	1.1 System prompts user to enter the Order ID for the order they wish to look up
	2. User inputs the Order ID	2.1 System displays the order's information
	3. User indicates desire to close order information	3.1 System stops displaying order information
<b>Exception Conditions</b>	2.1 Order ID is invalid	

As an **AAES, AAEM, WRS, WRM, or WSM**, I want to **look up the information of an existing order** so that **I can carry out business tasks involving the order's information**

- The order's information is displayed
- The order's information is left unchanged

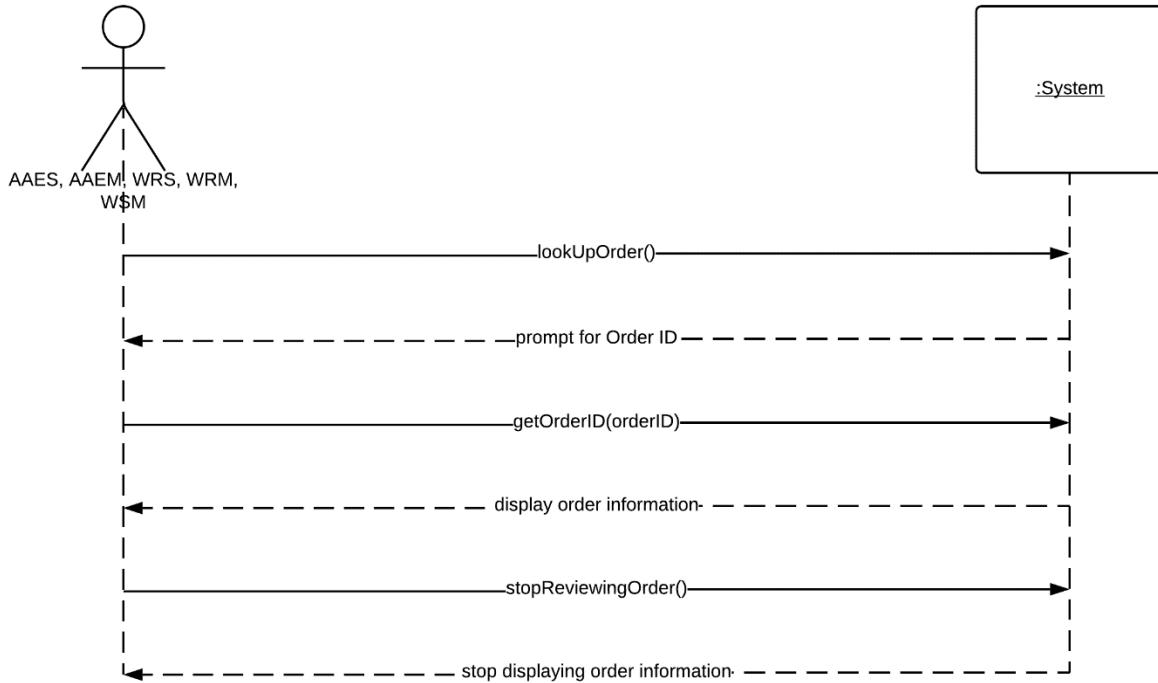
## Look Up Order Information

Righteous Row



## Look Up Order Information

Righteous Row



## Use Case #17: Create New Order

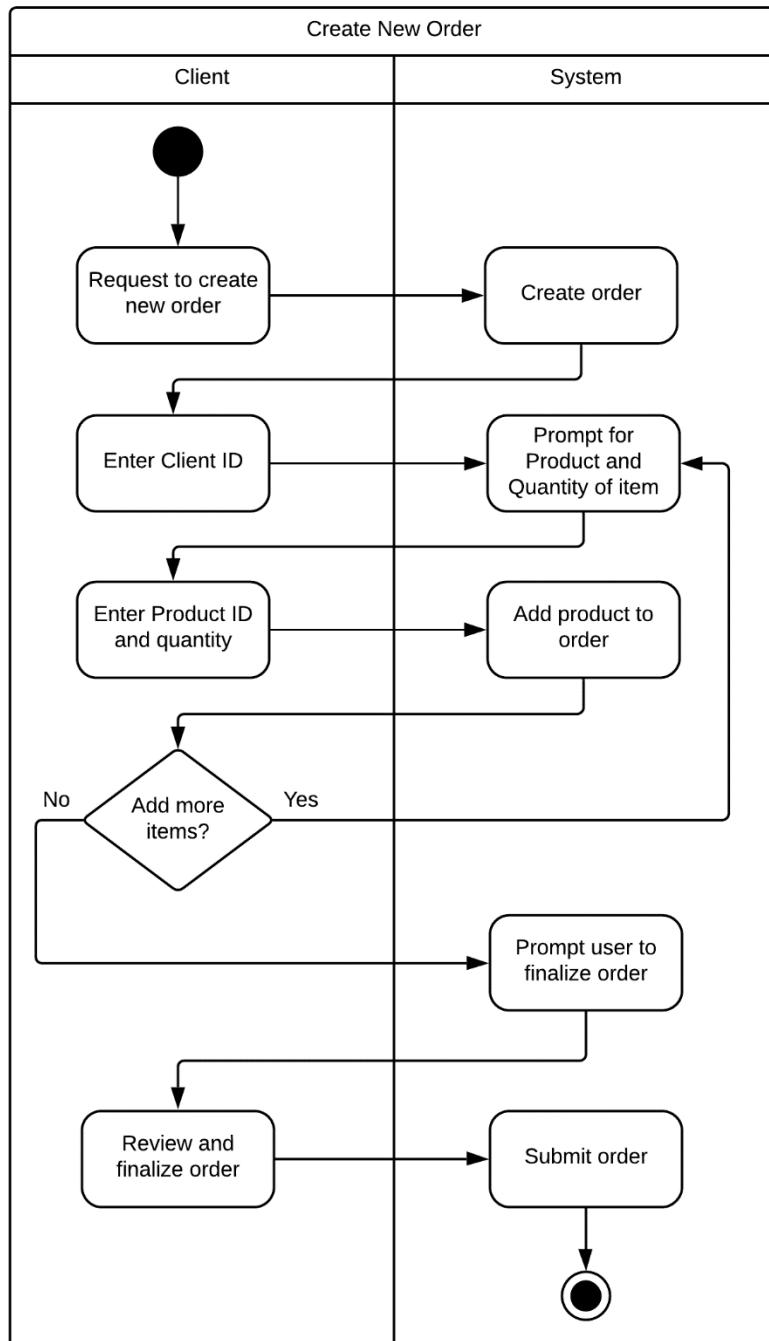
<b>Use Case Name</b>	Create New Order	
<b>Scenario</b>	Create a new client order	
<b>Triggering Event</b>	A client desires to place an order to ship a product from a DSI warehouse	
<b>Brief Description</b>	A client places an order for DSI to ship their products from one of DSI's warehouses	
<b>Actors</b>	Clients	
<b>Related Use Cases</b>	Review Existing Order, Cancel Existing Order	
<b>Stakeholders</b>	Warehouse Floor Manager, Warehouse Floor Staff, Warehouse Shipping Staff, Warehouse Shipping Manager, Clients	
<b>Preconditions</b>	The client is in the client database	
<b>Postconditions</b>	The order is placed into our system The shipping department is notified of the order	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. Client indicates desire to have items shipped from a DSI warehouse	1.1 System creates a new order and assigns it an Order ID 1.2 System prompts for user to enter Client ID
	2. User enters their Client ID	2.1 System verifies Client ID and assigns it to the order 2.2 System prompts user to input Product ID and quantity of the first item to be shipped
	3. User inputs the Product ID and quantity of the product to be shipped	3.1 System adds the product to the order 3.2 System asks the user if they wish to ship any other items
	4. Indicate desire to ship more items or finalize order	4.1 If client desires to add more items to order, return to step 2.2 4.2 If client desires to finalize order, prompt user to review order
	5. User reviews order and indicates desire to finalize it	5.1 System completes the entry of the order into the database 5.2 System prints packing slip in the Warehouse Shipping Department
<b>Exception Conditions</b>	2.1 Client ID is invalid 3.1 Quantity input by the user is more than what the warehouse has on hand	

As a **Client**, I want to **create a new order** so that **my products can be shipped from DSI's warehouse**

- System obtains all order information
- Shipping information is sent to the shipping department

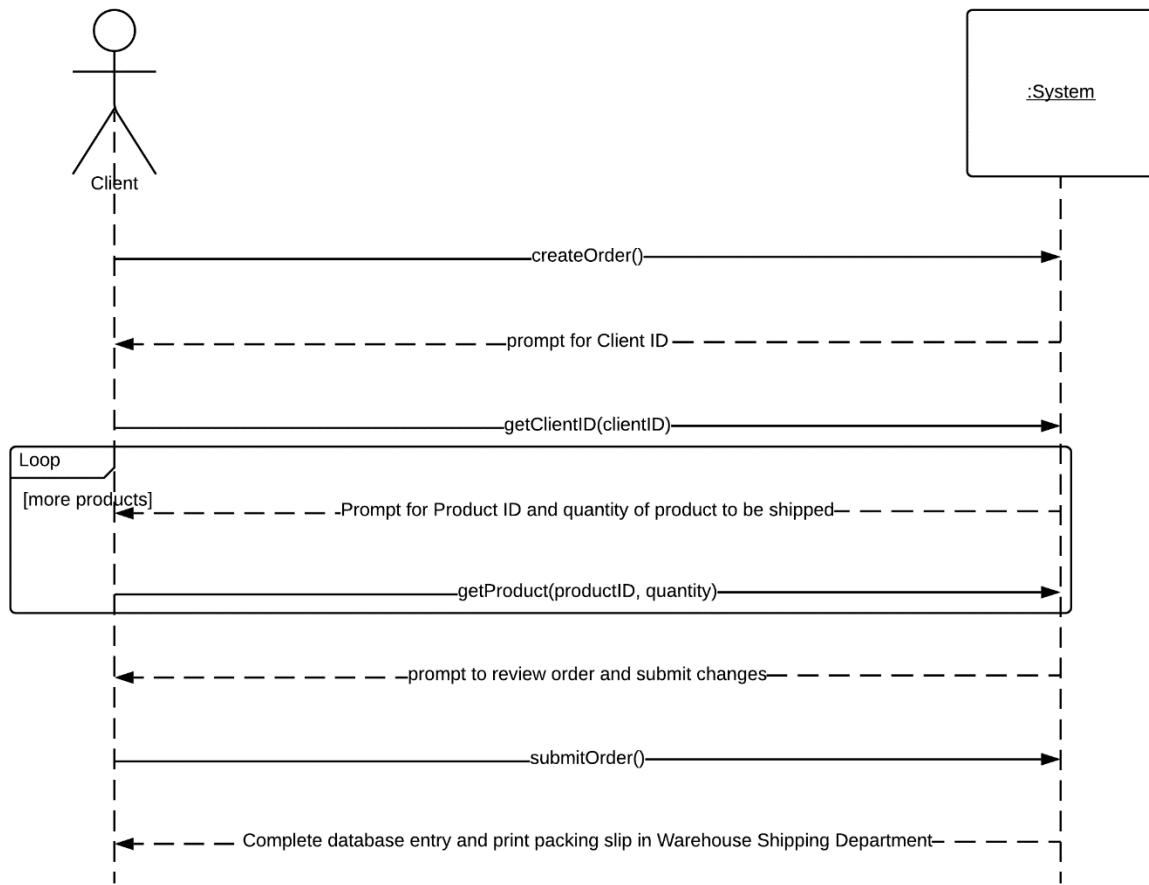
## Create New Order

Righteous Row



## Create New Order

Righteous Row



## Use Case #18: Review Existing Order

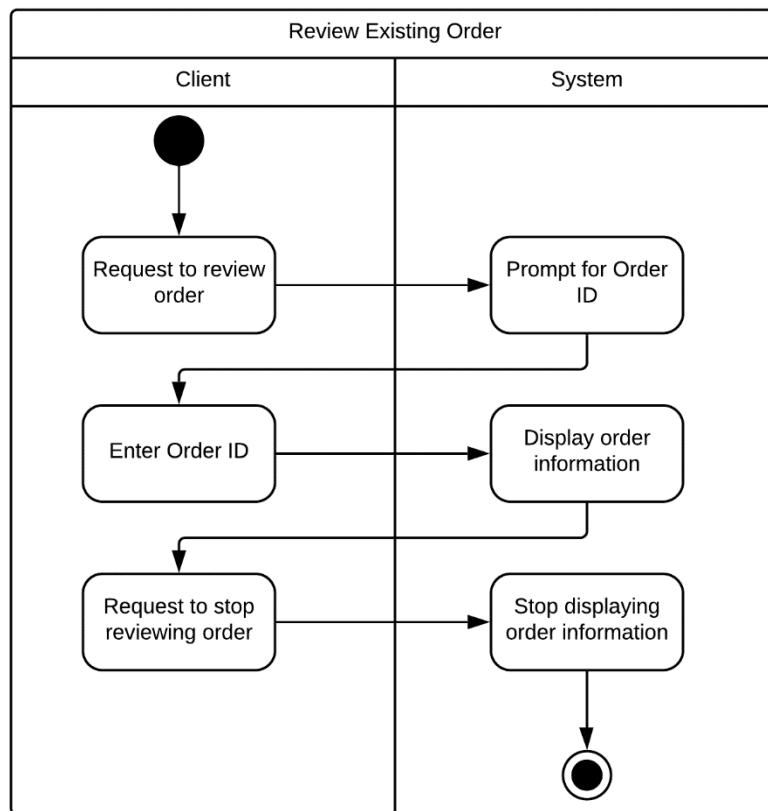
<b>Use Case Name</b>	Review Existing Order	
<b>Scenario</b>	Review an existing order	
<b>Triggering Event</b>	A client desires to view an order of theirs	
<b>Brief Description</b>	A client views an active/previous order of theirs, and is shown its information	
<b>Actors</b>	Clients	
<b>Related Use Cases</b>	Create New Order, Cancel Existing Order	
<b>Stakeholders</b>	Clients	
<b>Preconditions</b>	The order that the client wishes to review exists within the system	
<b>Postconditions</b>	The order remains unchanged	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to review an order of theirs	1.1 System prompts user to enter the Order ID for the order they wish to review
	2. User inputs the Order ID	2.1 System displays the order information
	3. User indicates desire to close the order information	3.1 System stops displaying the order information
<b>Exception Conditions</b>	2.1 Order ID is invalid	

As a **Client**, I want to **look up the information of an existing order of mine** so that I can carry out business tasks involving the order's information

- The order's information is displayed
- The order's information is left unchanged

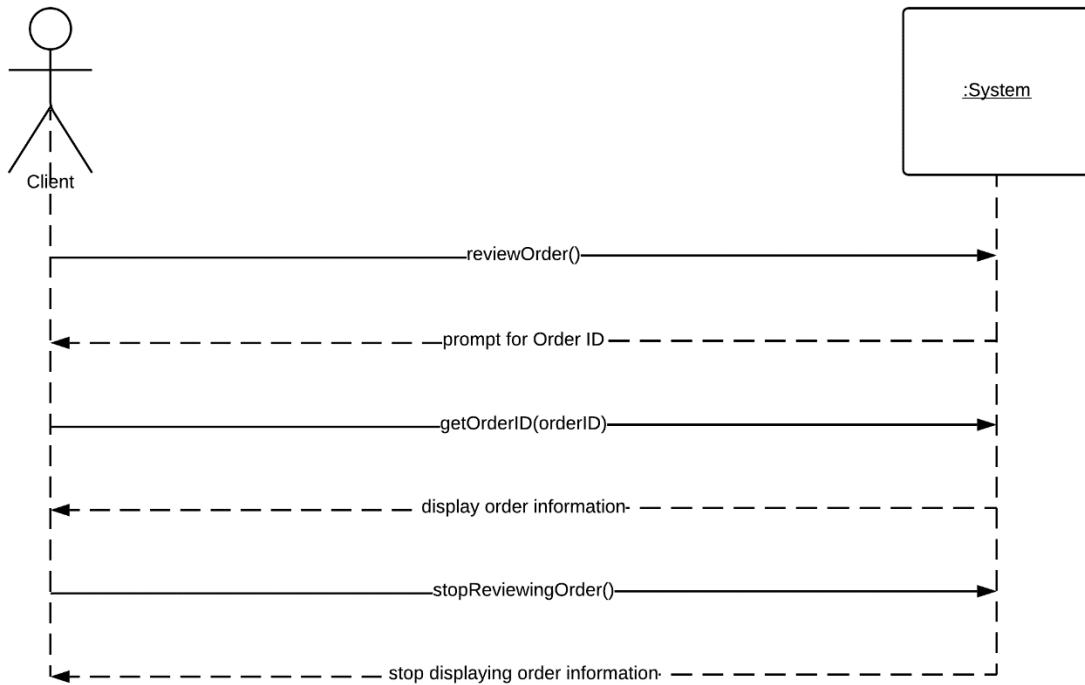
## Review Existing Order

Righteous Row



## Review Existing Order

Righteous Row



## Use Case #19: Cancel Existing Order

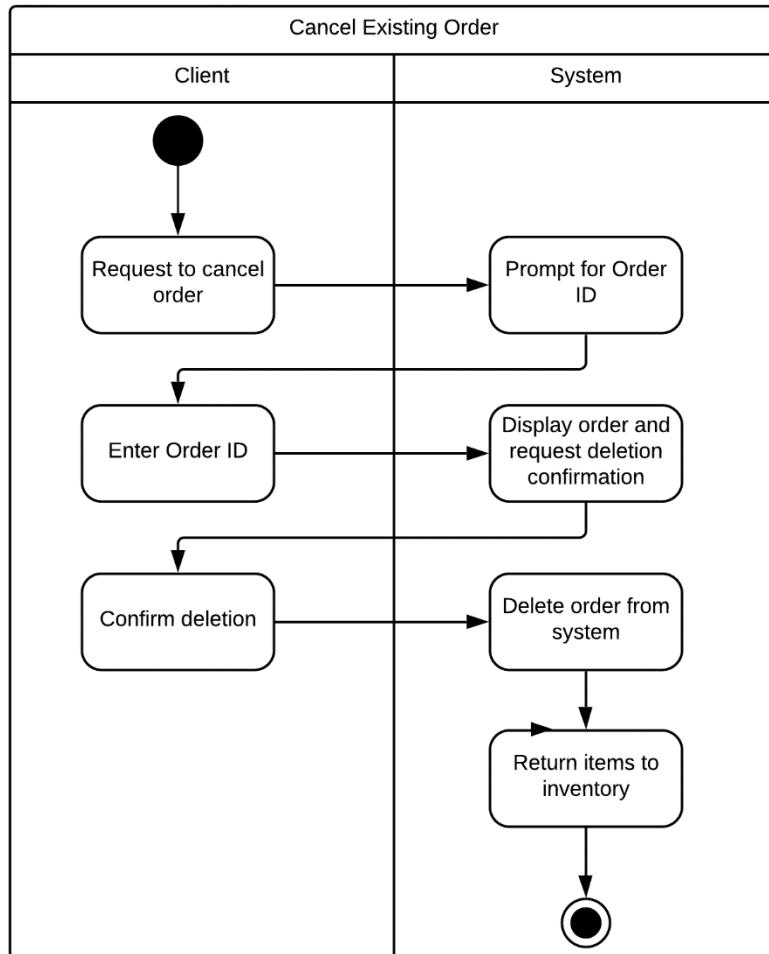
<b>Use Case Name</b>	Cancel Existing Order	
<b>Scenario</b>	Cancel a client's existing order	
<b>Triggering Event</b>	A client desires to cancel an order that they have previously placed with DSI	
<b>Brief Description</b>	The client uses the system to cancel an order that they have previously placed with DSI	
<b>Actors</b>	Clients	
<b>Related Use Cases</b>	Create New Order, Review Existing Order	
<b>Stakeholders</b>	Warehouse Shipping Staff, Warehouse Shipping Manager	
<b>Preconditions</b>	The order to be cancelled must exist within the system The order to be cancelled must not have already been shipped from a DSI warehouse	
<b>Postconditions</b>	The order is deleted from the system The items in the order are returned to inventory	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to cancel an order  2. User inputs the Order ID of the order they wish to cancel  3. User confirms that they wish to cancel the order	1.1 System prompts user to enter Order ID of the order to be cancelled  2.1 System pulls up the order information, and prompts user to confirm if it is indeed the order they wish to cancel  3.1 The order is deleted from the system, and the products from the order are returned to inventory
<b>Exception Conditions</b>	2.1 Order ID is invalid	

As a **Client**, I want to **cancel an existing order of mine** so that **I do not get sent items that I no longer want**

- The order's information is deleted from the database
- The items in the order are returned to inventory

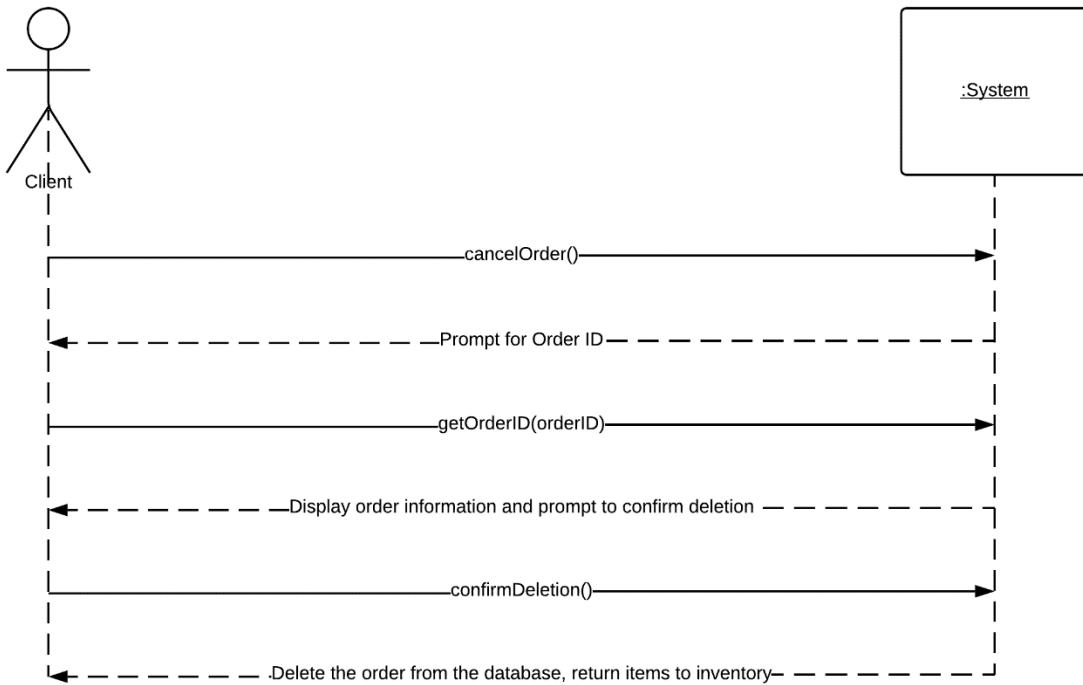
## Cancel Existing Order

Righteous Row



## Cancel Existing Order

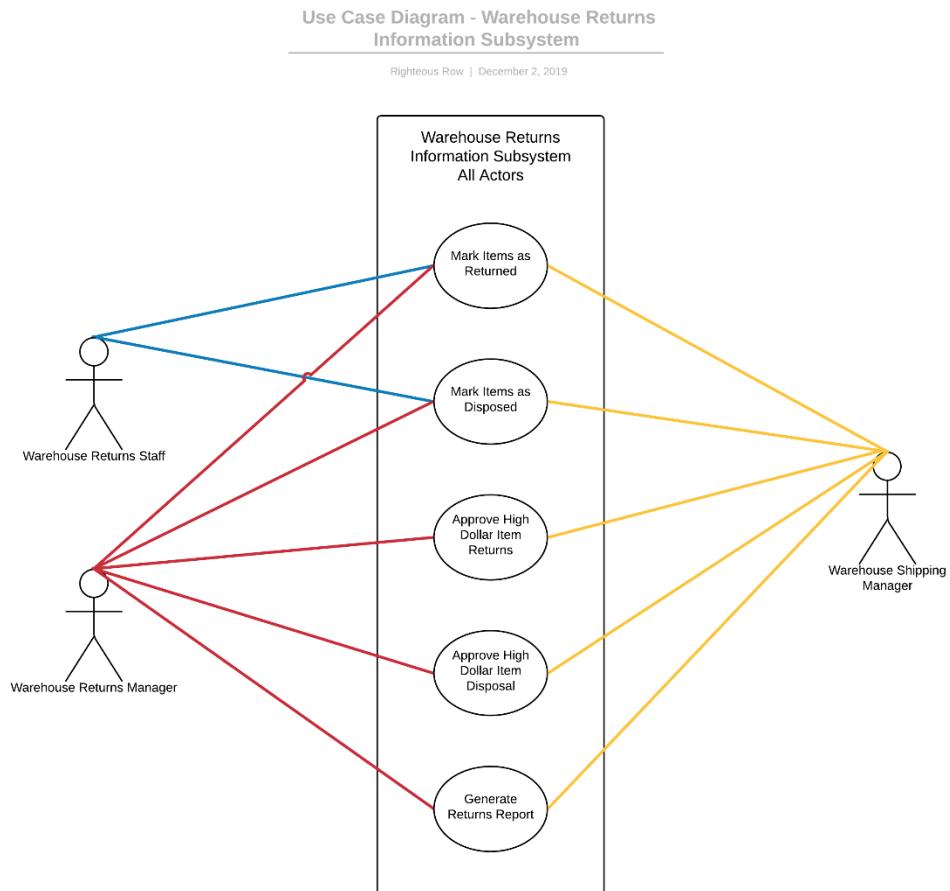
Righteous Row



## *Tabular Use Case Definitions – Warehouse Returns Subsystem*

Warehouse Returns Subsystem		
Use Case Name	Actors	Brief Description
Mark Items as Returned	WRS, WRM, WSM	A client sends an order that DSI previously shipped back to the warehouse, and the employee marks it as “returned” and places it back in the inventory
Approve High-Dollar Item Returns	WRM, WSM	The request to return a high-dollar item is reviewed and approved/denied
Mark Items as Disposed	WRS, WRM, WSM	A client sends an order that DSI previously shipped back to the warehouse, and the employee marks it as “disposed” and discards it
Approve High-Dollar Item Disposal	WRM, WSM	The request to dispose of a high-dollar item is reviewed and approved/denied
Generate Returns Report	WRM, WSM	Generates a report to see how many returns the WRS processes each period, how many high-dollar returns they process per period, and how many items are disposed of each period

## *Graphical Use Case Diagram – Warehouse Returns Subsystem*



# *Fully Developed Use Case Descriptions with User Stories, Activity Diagrams, and System Sequence Diagrams – Warehouse Returns Subsystem*

## Use Case #20: Mark Items as Returned

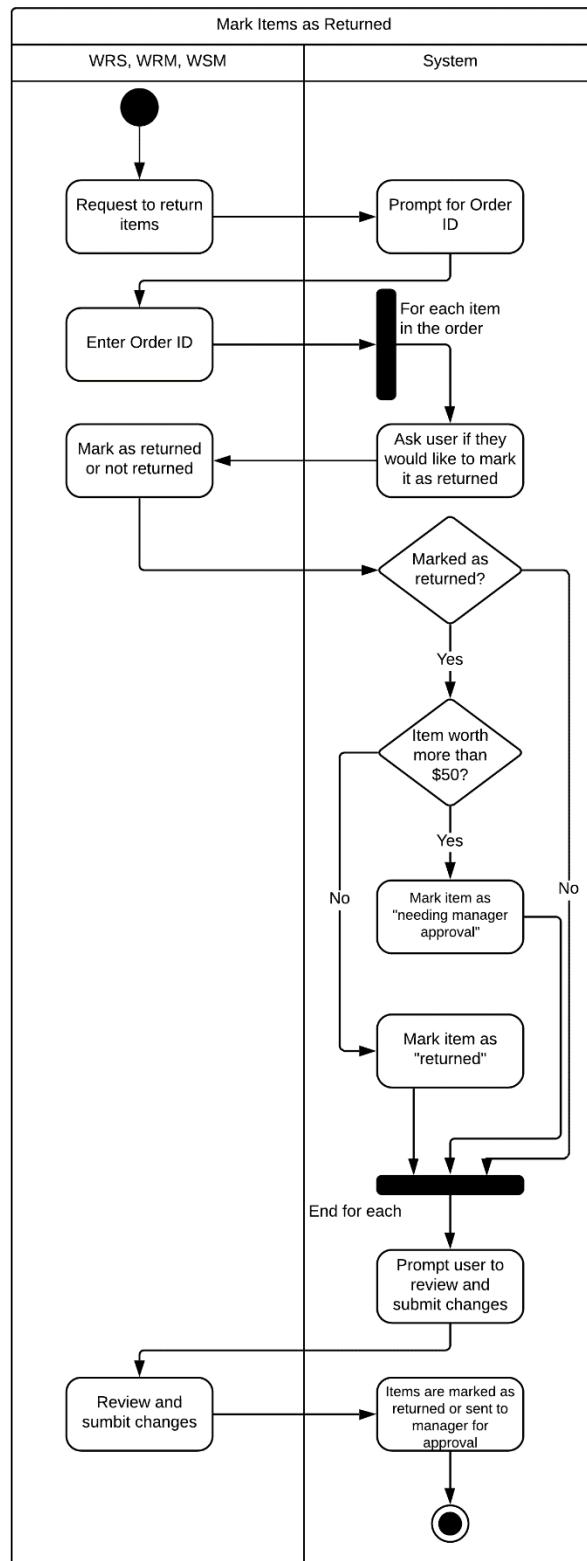
<b>Use Case Name</b>	Mark Items as Returned	
<b>Scenario</b>	Marks items returned by the client as “returned”	
<b>Triggering Event</b>	Client returns an item that can be returned to inventory	
<b>Brief Description</b>	A client sends items of an order that DSI previously shipped back to the warehouse, and the employee marks it as “returned” and places it back in the inventory	
<b>Actors</b>	WRS, WRM, WSM	
<b>Related Use Cases</b>	Approve High-Dollar Item Returns	
<b>Stakeholders</b>	Clients	
<b>Preconditions</b>	The items to be returned are part of an order that exists within the database The items to be returned have been previously shipped through DSI	
<b>Postconditions</b>	The items are marked as “returned” in the database The items are returned to inventory The client is refunded for the cost of the items Any high-dollar returns are sent to the WRM for approval	
<b>Flow of Activities</b>	<b>User</b> 1. User indicates desire to return items  2. User inputs Order ID  3. For each item in the order, indicates if they would like to return it or not  4. User reviews and accepts/declines the changes to the database	<b>System</b> 1.1 System prompts user to enter Order ID for the order from which items will be returned  2.1 For each item in the order, system asks the user if they would like to mark it as returned  3.1 If the user wishes to return the item, it is marked as returned in the database 3.2 If the user wishes to return the item and it is worth more than \$50, it is marked as needing manager approval 3.3 If the user does not wish to return the item, it is not marked as returned 3.4 While there are still items left in the order, return to step 2.1. Otherwise, continue to step 3.5. 3.5 Once the user has responded to the last item in the order, system displays which items have been marked as returned, needing manager approval, and not returned, and prompts the user to review and approve the changes  4.1 System marks items as returned in the database 4.2 System sends items needing manager approval to the WRM
<b>Exception Conditions</b>	2.1 Order ID is invalid	

As a **WRS** **WRM**, or **WSM**, I want to **mark an item as returned** so that **items returned by clients (that are able to be resold) are accounted for within the system**

- If the item is worth less than \$50, the status of item is set to returned, and the item is returned to inventory
- If the item is worth more than \$50, a request to approve the return is sent to the WRM

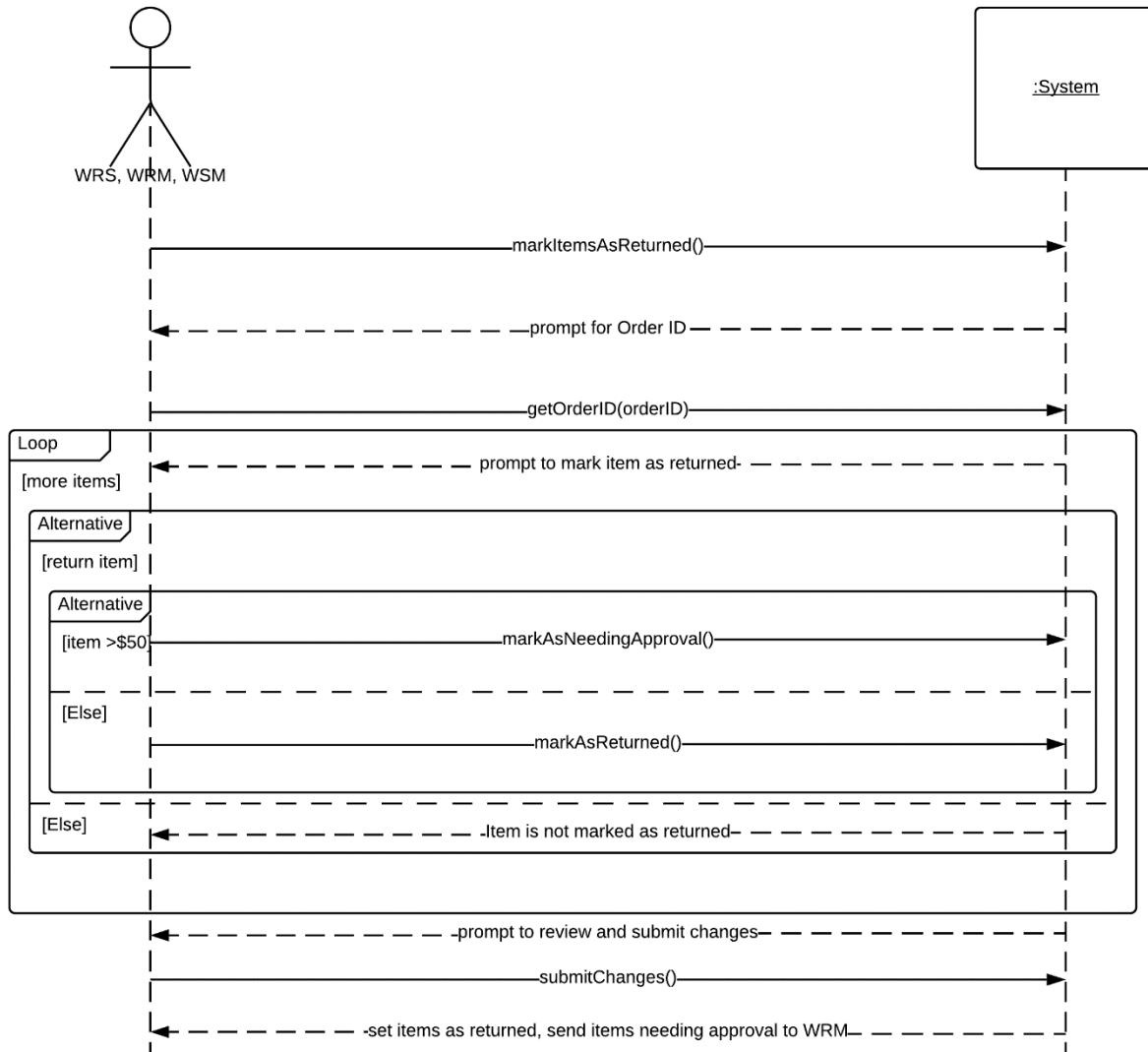
## Mark Items as Returned

Righteous Row



## Mark Items As Returned

Righteous Row



## Use Case #21: Approve High-Dollar Item Returns

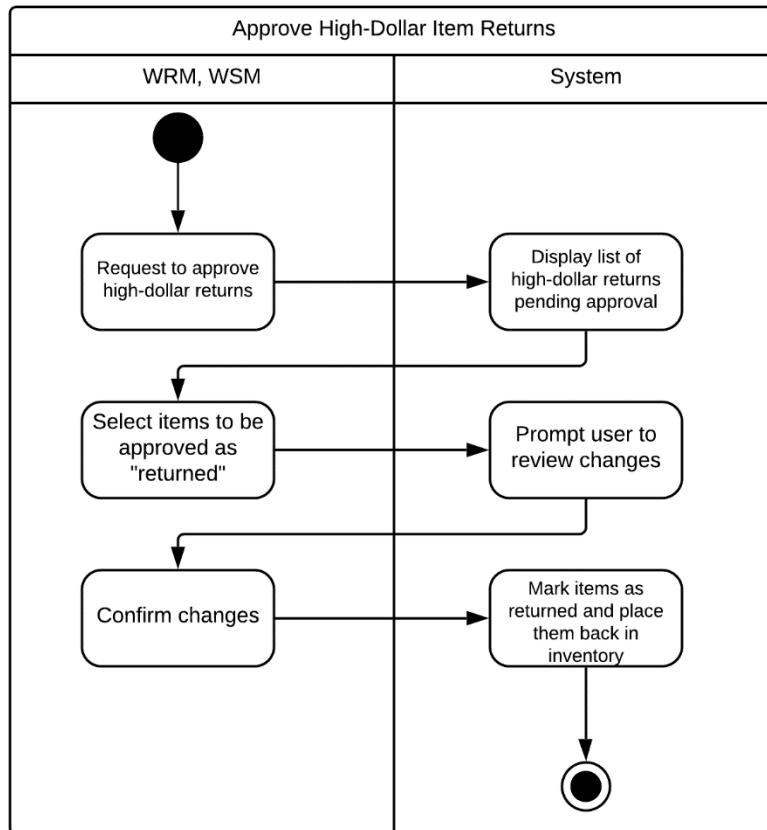
<b>Use Case Name</b>	Approve High-Dollar Item Returns	
<b>Scenario</b>	Mark items worth more than \$50 as “returned”	
<b>Triggering Event</b>	Warehouse returns staff processes a batch of returns that contains high-dollar items	
<b>Brief Description</b>	The request to return a high-dollar item is reviewed and approved/denied	
<b>Actors</b>	WRM, WSM	
<b>Related Use Cases</b>	Mark Items as Returned	
<b>Stakeholders</b>	Clients, WRS	
<b>Preconditions</b>	WRS finishes processing a batch of returns and wishes to return an item worth more than \$50	
<b>Postconditions</b>	The items are marked as “returned” in the database The items are returned to inventory The client is refunded for the cost of the items	
<b>Flow of Activities</b>	<b>User</b> 1. User indicates desire to approve high-dollar item returns  2. User selects the items they wish to mark as returned  3. User confirms that they wish to return selected high-dollar items	<b>System</b> 1.1 System displays list of high-dollar items that are pending manager approval to be returned 1.2 System prompts user to approve or deny the high-dollar returns  2.1 System prompts user to review and approve the list of selected high-dollar items  3.1 System marks items as returned, and they are moved back into inventory
<b>Exception Conditions</b>	1.1 There are no active high-dollar return requests	

As an **WRM or WSM**, I want to **approve return requests of high-dollar items** so that **expensive items are accounted for if they are returned**

- High-dollar item return request is sent and approved by the WRM
- The item is marked as returned in the database

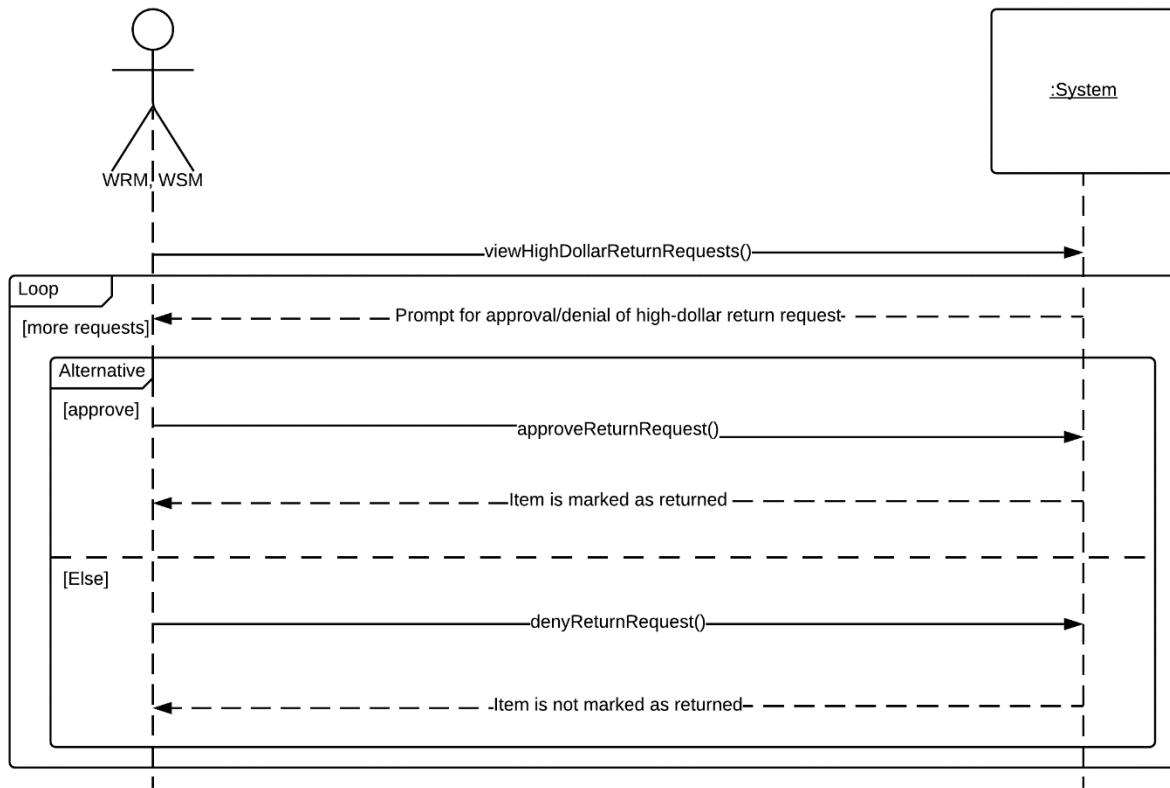
## Approve High-Dollar Item Returns

Righteous Row



## Approve High-Dollar Item Returns

Righteous Row



## Use Case #22: Mark Items as Disposed

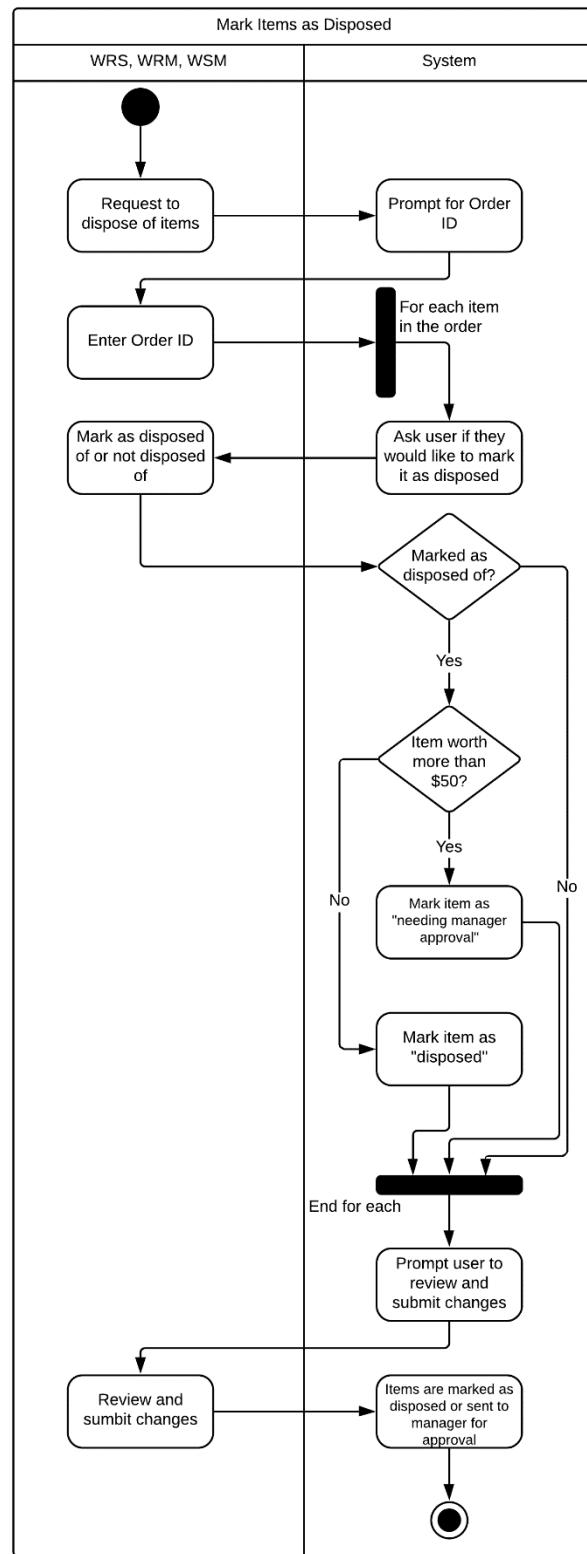
<b>Use Case Name</b>	Mark Items as Disposed											
<b>Scenario</b>	Marks items returned by the client as “disposed”											
<b>Triggering Event</b>	Client returns an item that can not be returned to inventory and must be disposed											
<b>Brief Description</b>	A client sends an order that DSI previously shipped back to the warehouse, and the employee marks it as “disposed” and discards it											
<b>Actors</b>	WRS, WRM, WSM											
<b>Related Use Cases</b>	Approve High-Dollar Item Disposal											
<b>Stakeholders</b>	Clients											
<b>Preconditions</b>	The items to be disposed are part of an order that exists within the database The items to be disposed have been previously shipped through DSI											
<b>Postconditions</b>	The items are marked as “disposed” in the database The items are discarded The client is refunded for the cost of the items Any high-dollar disposals are sent to the WRM for approval											
<b>Flow of Activities</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 2px;">User</th> <th style="text-align: center; padding: 2px;">System</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">1. User indicates desire to dispose of items</td> <td style="padding: 2px;">1.1 System prompts user to enter Order ID for the order from which items will be disposed of</td> </tr> <tr> <td style="padding: 2px;">2. User inputs Order ID</td> <td style="padding: 2px;">2.1 For each item in the order, system asks the user if they would like to mark it as disposed</td> </tr> <tr> <td style="padding: 2px;">3. For each item in the order, indicates if they would like to dispose of it or not</td> <td style="padding: 2px;">3.1 If the user wishes to dispose of the item, it is marked as disposed in the database 3.2 If the user wishes to dispose the item and it is worth more than \$50, it is marked as needing manager approval 3.3 If the user does not wish to dispose of the item, it is not marked as disposed 3.4 While there are still items left in the order, return to step 2.1. Otherwise, continue to step 3.5. 3.5 Once the user has responded to the last item in the order, system displays which items have been marked as disposed, needing manager approval, and not disposed, and prompts the user to review and approve the changes</td> </tr> <tr> <td style="padding: 2px;">4. User reviews and accepts/declines the changes to the database</td> <td style="padding: 2px;">4.1 System marks items as disposed in the database 4.2 System sends items needing manager approval to the WRM</td> </tr> </tbody> </table>	User	System	1. User indicates desire to dispose of items	1.1 System prompts user to enter Order ID for the order from which items will be disposed of	2. User inputs Order ID	2.1 For each item in the order, system asks the user if they would like to mark it as disposed	3. For each item in the order, indicates if they would like to dispose of it or not	3.1 If the user wishes to dispose of the item, it is marked as disposed in the database 3.2 If the user wishes to dispose the item and it is worth more than \$50, it is marked as needing manager approval 3.3 If the user does not wish to dispose of the item, it is not marked as disposed 3.4 While there are still items left in the order, return to step 2.1. Otherwise, continue to step 3.5. 3.5 Once the user has responded to the last item in the order, system displays which items have been marked as disposed, needing manager approval, and not disposed, and prompts the user to review and approve the changes	4. User reviews and accepts/declines the changes to the database	4.1 System marks items as disposed in the database 4.2 System sends items needing manager approval to the WRM	
User	System											
1. User indicates desire to dispose of items	1.1 System prompts user to enter Order ID for the order from which items will be disposed of											
2. User inputs Order ID	2.1 For each item in the order, system asks the user if they would like to mark it as disposed											
3. For each item in the order, indicates if they would like to dispose of it or not	3.1 If the user wishes to dispose of the item, it is marked as disposed in the database 3.2 If the user wishes to dispose the item and it is worth more than \$50, it is marked as needing manager approval 3.3 If the user does not wish to dispose of the item, it is not marked as disposed 3.4 While there are still items left in the order, return to step 2.1. Otherwise, continue to step 3.5. 3.5 Once the user has responded to the last item in the order, system displays which items have been marked as disposed, needing manager approval, and not disposed, and prompts the user to review and approve the changes											
4. User reviews and accepts/declines the changes to the database	4.1 System marks items as disposed in the database 4.2 System sends items needing manager approval to the WRM											
<b>Exception Conditions</b>	2.1 Order ID is invalid											

As a **WRS WRM, or WSM**, I want to **mark an item as disposed** so that **items returned by clients (that are unable to be resold) are accounted for within the system**

- If the item is worth less than \$50, the status of item is set to disposed, and the item is disposed of
- If the item is worth more than \$50, a request to approve the disposal is sent to the WRM

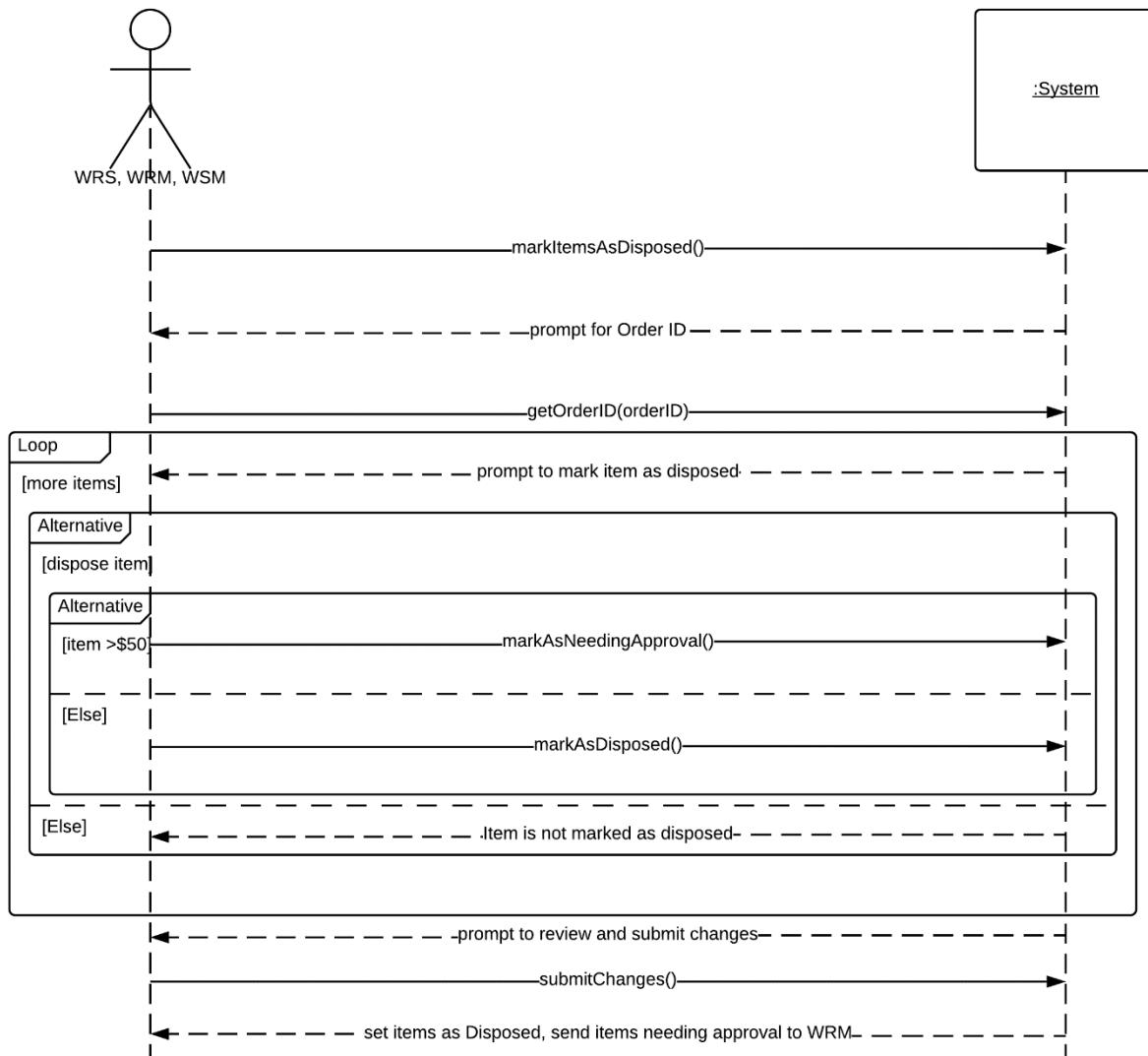
## Mark Items as Disposed

Righteous Row



## Mark Items As Disposed

Righteous Row



## Use Case #23: Approve High-Dollar Item Disposal

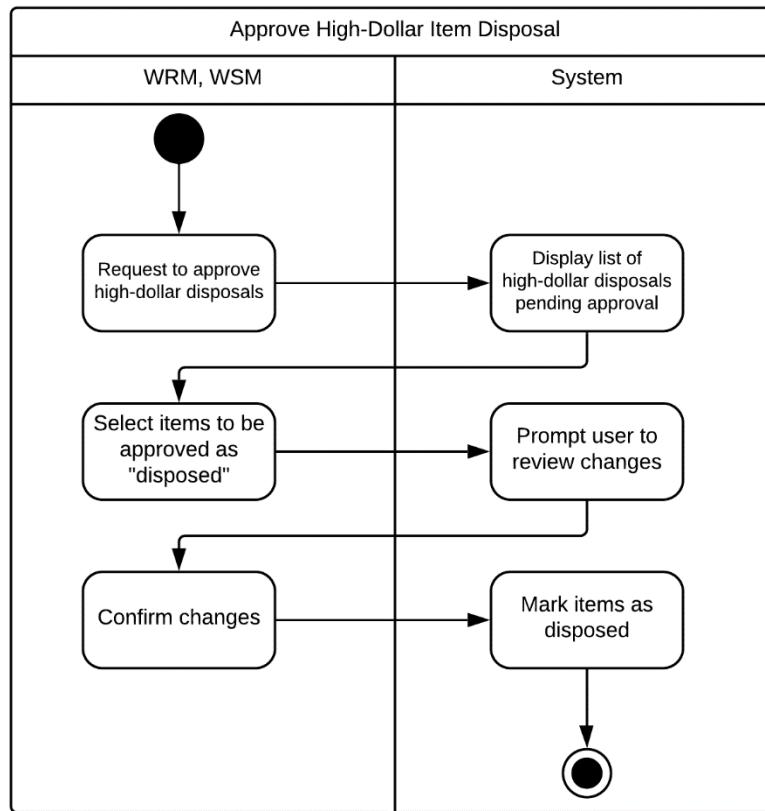
<b>Use Case Name</b>	Approve High-Dollar Item Disposal	
<b>Scenario</b>	Mark items worth more than \$50 as “disposed”	
<b>Triggering Event</b>	Warehouse returns staff processes a batch of items to be disposed that contains high-dollar items	
<b>Brief Description</b>	The request to dispose of a high-dollar item is reviewed and approved/denied	
<b>Actors</b>	WRM, WSM	
<b>Related Use Cases</b>	Mark Items as Disposed	
<b>Stakeholders</b>	Clients, WRS	
<b>Preconditions</b>	WRS finishes processing a batch of disposals and wishes to dispose of an item worth more than \$50	
<b>Postconditions</b>	The items are marked as “disposed” in the database The items are disposed of The client is refunded for the cost of the items	
<b>Flow of Activities</b>	<b>User</b> 1. User indicates desire to approve high-dollar item disposals  2. User selects the items they wish to mark as disposed of  3. User confirms that they wish to dispose of selected high-dollar items	<b>System</b> 1.1 System displays list of high-dollar items that are pending manager approval to be disposed of 1.2 System prompts user to approve or deny the high-dollar disposals  2.1 System prompts user to review and approve the list of selected high-dollar items  3.1 System marks items as disposed of, and they are disposed of
<b>Exception Conditions</b>	1.1 There are no active high-dollar disposal requests	

As an **WRM or WSM**, I want to **approve disposal requests of high-dollar items** so that **expensive items are accounted for if they disposed of**

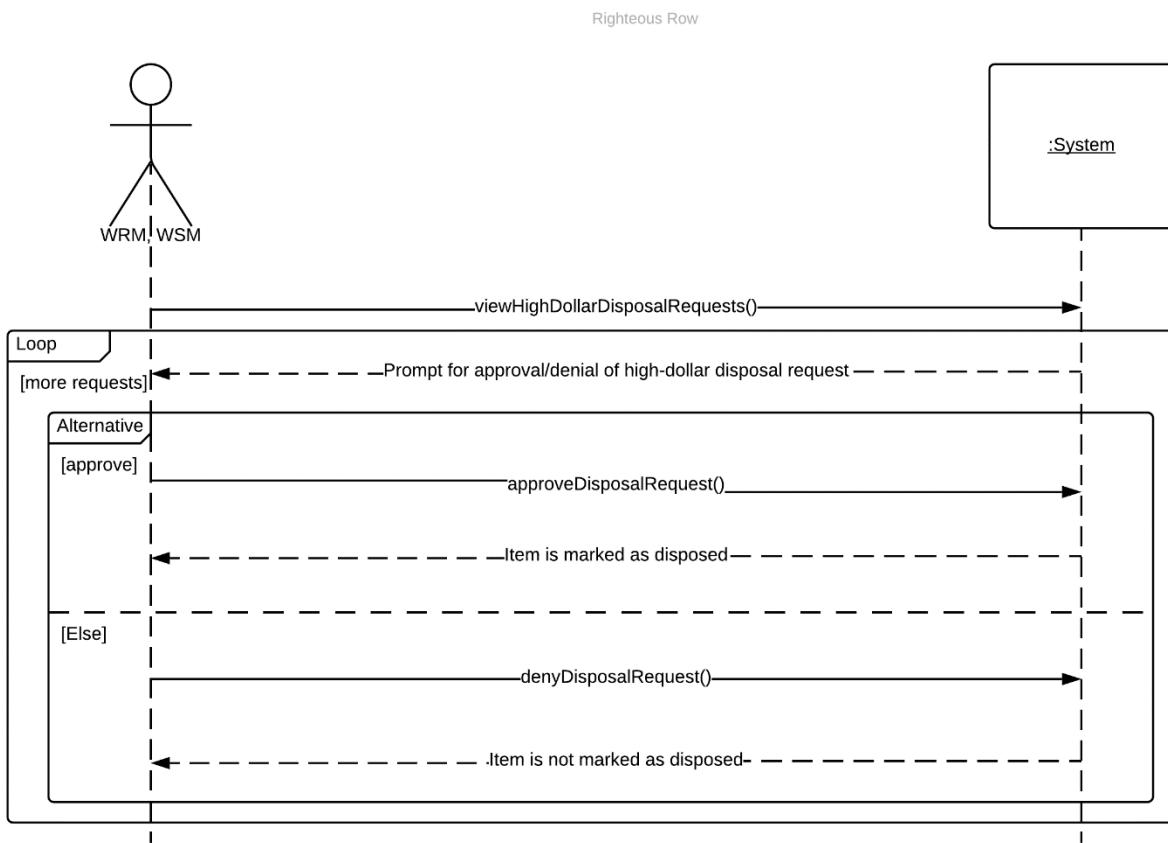
- High-dollar item disposal request is sent and approved by the WRM
- The item is marked as disposed in the database

## Approve High-Dollar Item Disposal

Righteous Row



## Approve High-Dollar Item Disposal



## Use Case #24: Generate Returns Report

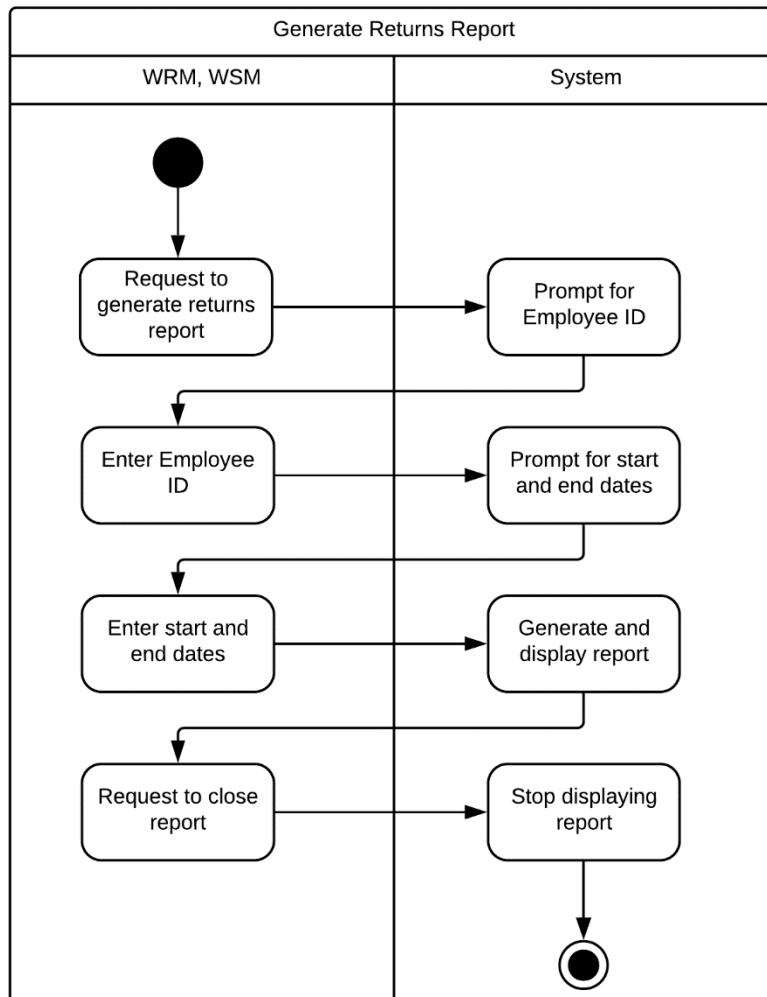
<b>Use Case Name</b>	Generate Returns Report	
<b>Scenario</b>	Generate a report containing information about an employee's returns and disposals	
<b>Triggering Event</b>	WRM desires to monitor the returns staff	
<b>Brief Description</b>	Generates a report to see how many returns the WRS processes each period, how many high-dollar returns they process per period, and how many items are disposed of each period	
<b>Actors</b>	WRM, WSM	
<b>Related Use Cases</b>	Mark Items as Returned, Mark Items as Disposed, Approve High-Dollar Item Returns, Approve High-Dollar Item Disposal	
<b>Stakeholders</b>	ABOS, ABOM, WRS, Clients	
<b>Preconditions</b>	The employee which the report will be generated for must exist within the system	
<b>Postconditions</b>	The employee information remains unchanged in the system	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to generate a returns report for a returns employee  2. User inputs the Employee ID  3. User inputs the desired dates  4. User indicates desire to close the returns report	1.1 System prompts user to enter the Employee ID of the employee for whom the report will be generated  2.1 System prompts user to enter starting and ending dates for the information in the report  3.1 System displays an employee profile containing how many returns the employee has processed in that time frame, how many high-dollar returns they have processed in that period, and how many items they have disposed of in that period  4.1 System stops displaying the returns report
<b>Exception Conditions</b>	2.1 Employee ID is invalid 3.1 Start date or end date is invalid	

As an **WRM or WSM**, I want to **generate a returns report** so that **I can monitor the returns staff to see various information about their returns and disposals**

- The returns report is generated and displayed

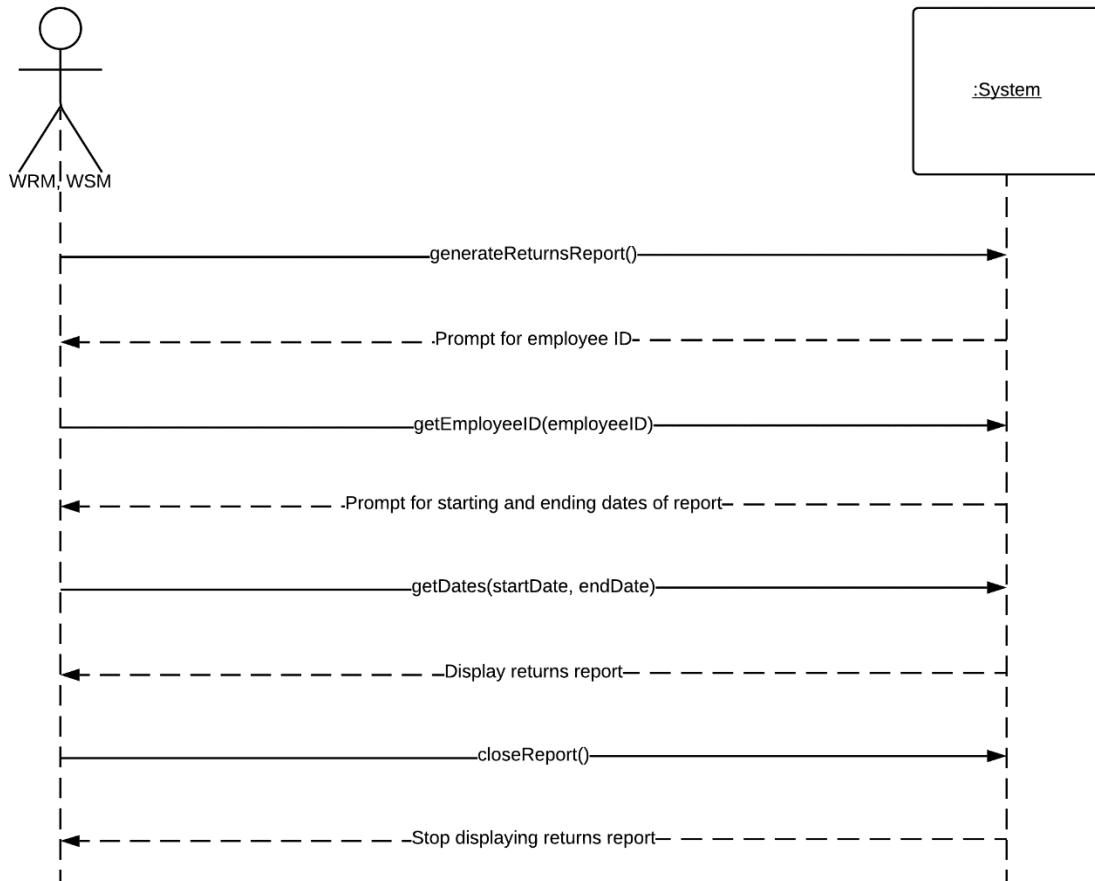
## Generate Returns Report

Righteous Row



## Generate Returns Report

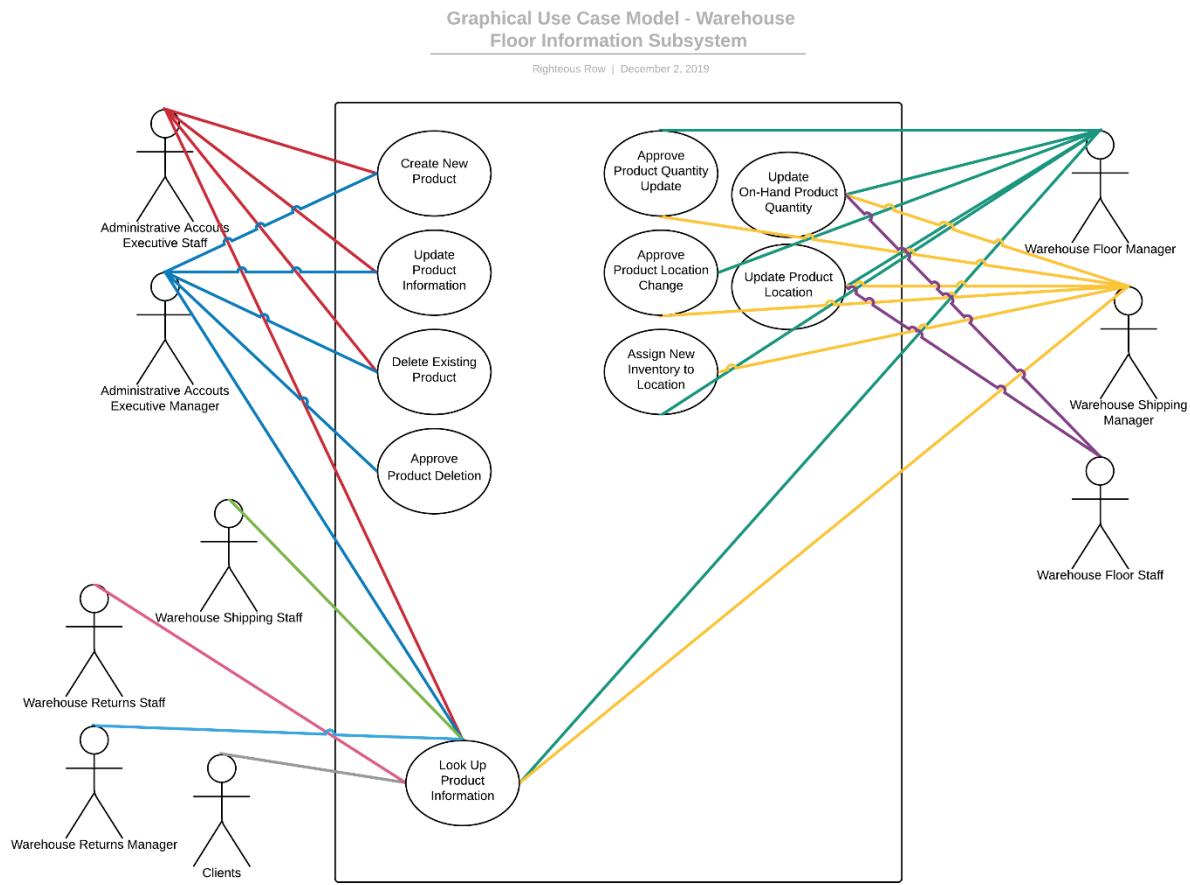
Righteous Row



## *Tabular Use Case Definitions – Warehouse Floor Subsystem*

Warehouse Floor Information System		
Use Case Name	Actors	Brief Description
Create New Product	AAES, AAEM	A new product is created and stored in the database
Look Up Product Information	AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM, C	The information of an existing product in the database is displayed
Update Product Information	AAES, AAEM	The information of an existing product is updated
Delete Existing Product	AAES, AAEM	An existing product is deleted from the database
Approve Product Deletion	AAEM	The deletion of a product from the database is reviewed and approved/denied
Update On-Hand Product Quantity	WFS, WFM, WSM	The quantity of a product is updated and changed in the database
Approve Product Quantity Update	WFM, WSM	The quantity update of a product is reviewed and approved/denied
Update Product Location	WFS, WFM, WSM	The location of a product in the warehouse is changed in the database
Approve Product Location Change	WFM, WSM	The location change of a product is reviewed and approved/denied
Assign New Product to Location	WFM, WSM	A new product in the inventory is assigned to a location in the warehouse

# Graphical Use Case Diagram – Warehouse Floor Subsystem



# *Fully Developed Use Case Descriptions with User Stories, Activity Diagrams, and System Sequence Diagrams – Warehouse Floor Subsystem*

## Use Case #25: Create New Product

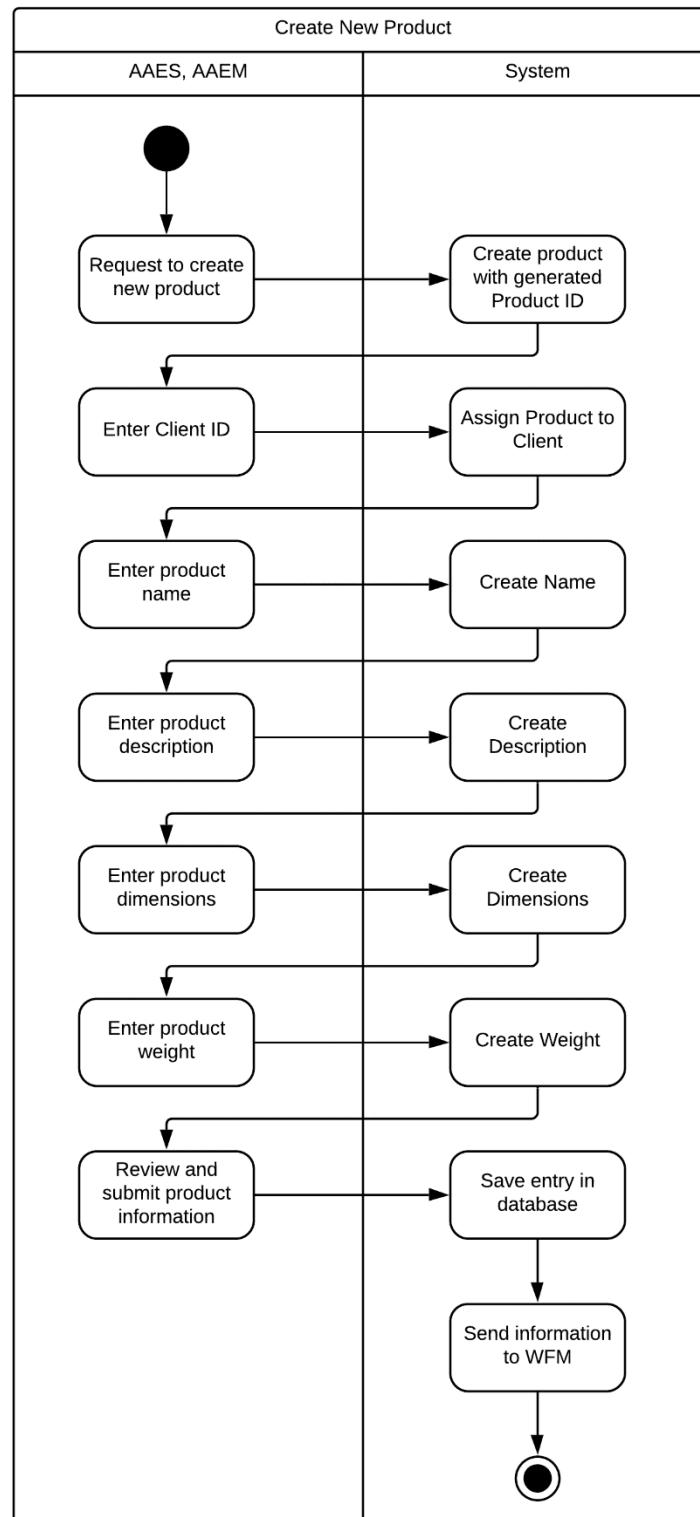
<b>Use Case Name</b>	Create New Product																	
<b>Scenario</b>	Create a new product in the database																	
<b>Triggering Event</b>	DSI receives a new product that has not been stored in any of their warehouses before																	
<b>Brief Description</b>	A new product is created and stored in the database																	
<b>Actors</b>	AAES, AAEM																	
<b>Related Use Cases</b>	Assign New Product to Location																	
<b>Stakeholders</b>	AAES, AAEM, WSS, WSM, WFS, WFM, WRS, WRM, Clients																	
<b>Preconditions</b>	The client who stores the new product in the warehouse exists within the database																	
<b>Postconditions</b>	The new product is entered within the system																	
<b>Flow of Activities</b>	<table border="1"> <thead> <tr> <th>User</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to create a new product in the system</td> <td>1.1 System creates new entry in the product database and generates a Product ID 1.2 System prompts user to input the Client ID for the client who is storing the product with DSI</td> </tr> <tr> <td>2. User inputs the Client ID</td> <td>2.1 System prompts user to input the product name</td> </tr> <tr> <td>3. User inputs the product name</td> <td>3.1 System prompts the user to input a product description</td> </tr> <tr> <td>4. User inputs a product description</td> <td>4.1 System prompts the user to input the product dimensions</td> </tr> <tr> <td>5. User inputs product dimensions</td> <td>5.1 System prompts user to input the product weight</td> </tr> <tr> <td>6. User inputs product weight</td> <td>6.1 System prompts user to review the information of the new product</td> </tr> <tr> <td>7. User reviews and submits new product information</td> <td>7.1 System saves database entry for the new product 7.2 System sends information to WFM to create a spot in inventory</td> </tr> </tbody> </table>	User	System	1. User indicates desire to create a new product in the system	1.1 System creates new entry in the product database and generates a Product ID 1.2 System prompts user to input the Client ID for the client who is storing the product with DSI	2. User inputs the Client ID	2.1 System prompts user to input the product name	3. User inputs the product name	3.1 System prompts the user to input a product description	4. User inputs a product description	4.1 System prompts the user to input the product dimensions	5. User inputs product dimensions	5.1 System prompts user to input the product weight	6. User inputs product weight	6.1 System prompts user to review the information of the new product	7. User reviews and submits new product information	7.1 System saves database entry for the new product 7.2 System sends information to WFM to create a spot in inventory	
User	System																	
1. User indicates desire to create a new product in the system	1.1 System creates new entry in the product database and generates a Product ID 1.2 System prompts user to input the Client ID for the client who is storing the product with DSI																	
2. User inputs the Client ID	2.1 System prompts user to input the product name																	
3. User inputs the product name	3.1 System prompts the user to input a product description																	
4. User inputs a product description	4.1 System prompts the user to input the product dimensions																	
5. User inputs product dimensions	5.1 System prompts user to input the product weight																	
6. User inputs product weight	6.1 System prompts user to review the information of the new product																	
7. User reviews and submits new product information	7.1 System saves database entry for the new product 7.2 System sends information to WFM to create a spot in inventory																	
<b>Exception Conditions</b>	2.1 Client ID is invalid																	

As an **AAES or AAEM**, I want to **create new products** so that **clients' items are stored within the database**

- The products are created within the database
- A request to create a storage spot in the warehouse is sent to the WFM

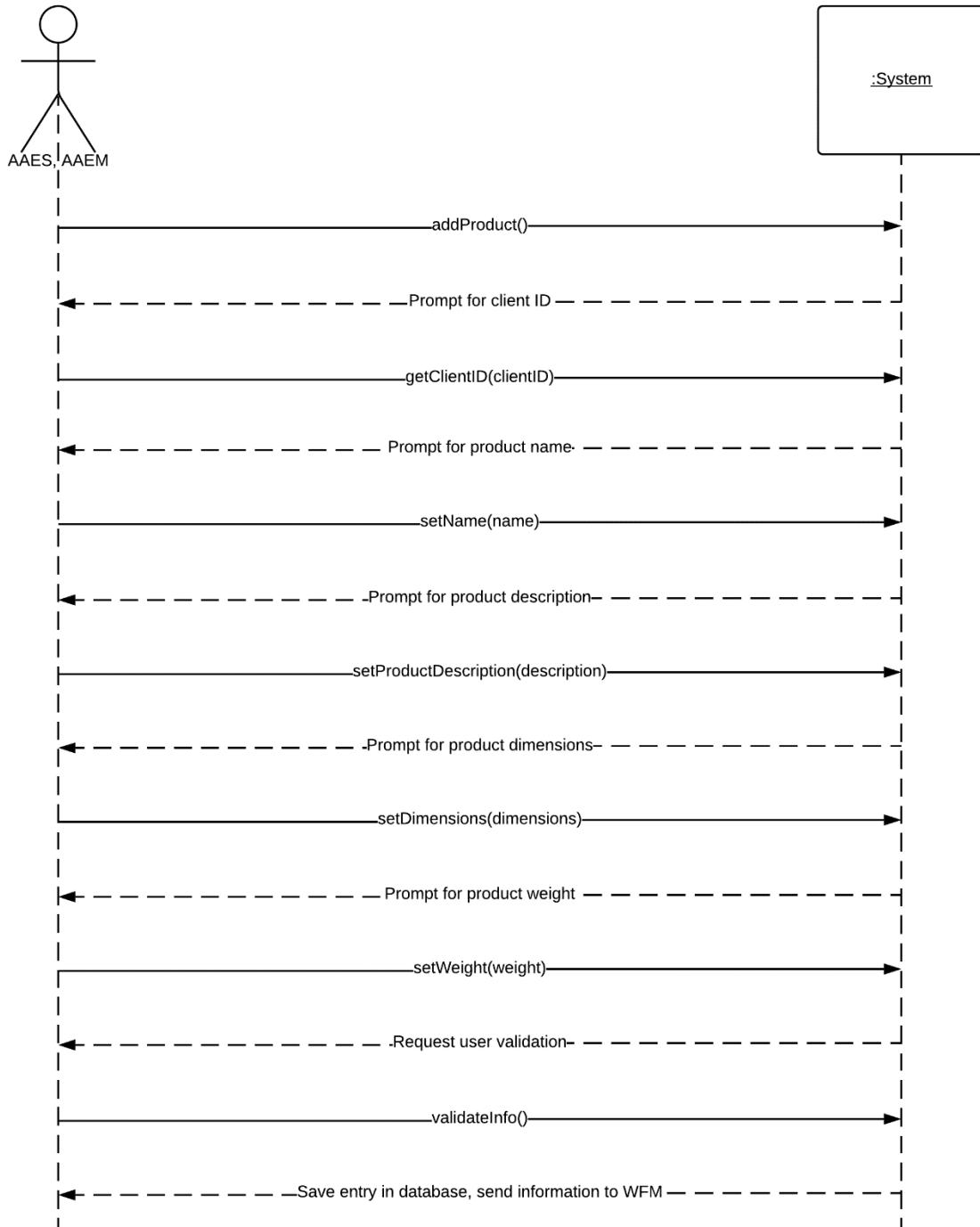
## Create New Product

Righteous Row



## Create New Product

Righteous Row



## Use Case #26: Look Up Product Information

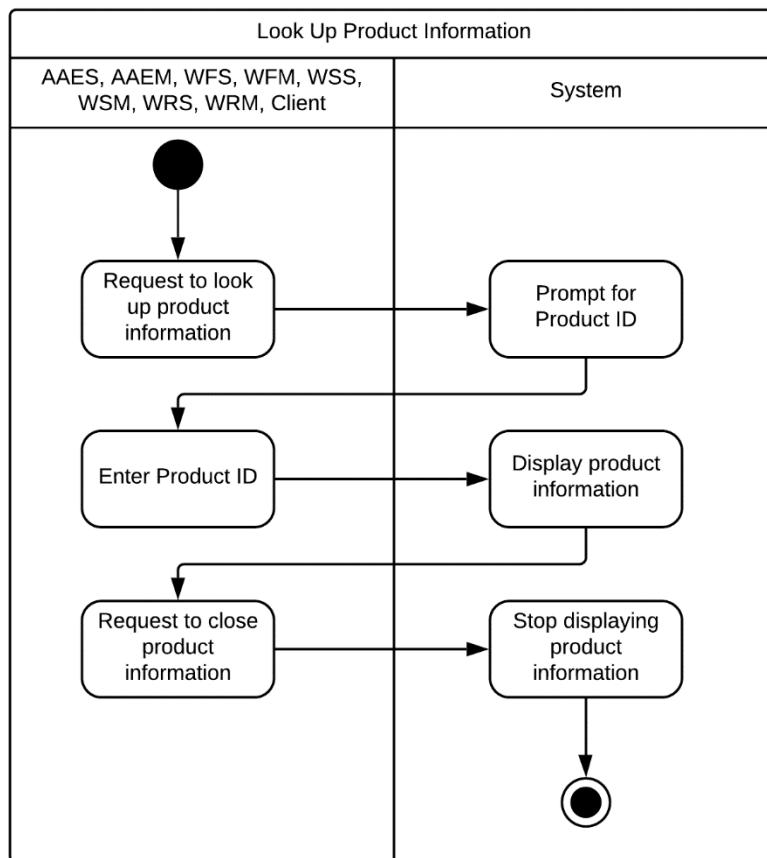
<b>Use Case Name</b>	Look Up Product Information	
<b>Scenario</b>	Look up a product's information in the database	
<b>Triggering Event</b>	An employee or client desires to look up a product's information	
<b>Brief Description</b>	The information of an existing product in the database is displayed	
<b>Actors</b>	AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM, Clients	
<b>Related Use Cases</b>	Update Product Information	
<b>Stakeholders</b>	AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM, Clients	
<b>Preconditions</b>	The product to be looked up exists within the system	
<b>Postconditions</b>	The product's information is left unchanged	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to look up a product's information  2. User enters the Product ID  3. User indicates to desire to close product information	1.1 System prompts user to enter the Product ID of the product to be looked up  2.1 System displays the product's information  3.1 System stops displaying the product's information
<b>Exception Conditions</b>	2.1 Product ID is invalid	

As an **AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM, or Client**, I want to **look up the information of an existing product** so that **I can carry out business tasks involving the product's information**

- The product's information is displayed
- The product's information is left unchanged

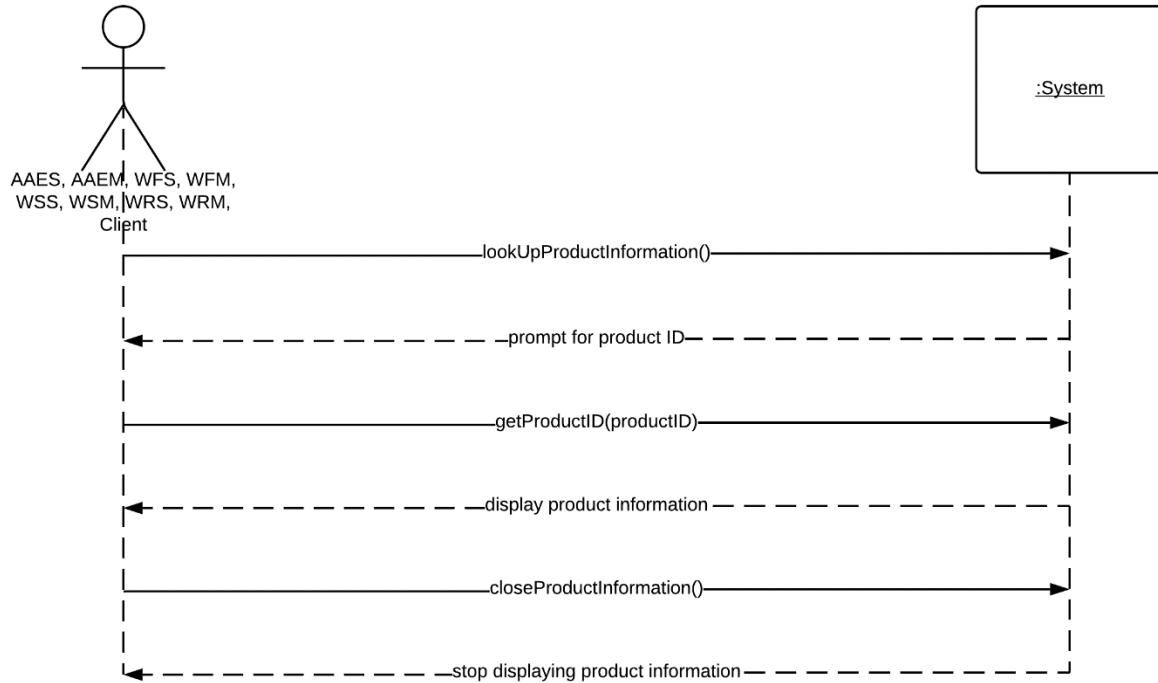
## Look Up Product Information

Righteous Row



## Look Up Product Information

Righteous Row



## Use Case #27: Update Product Information

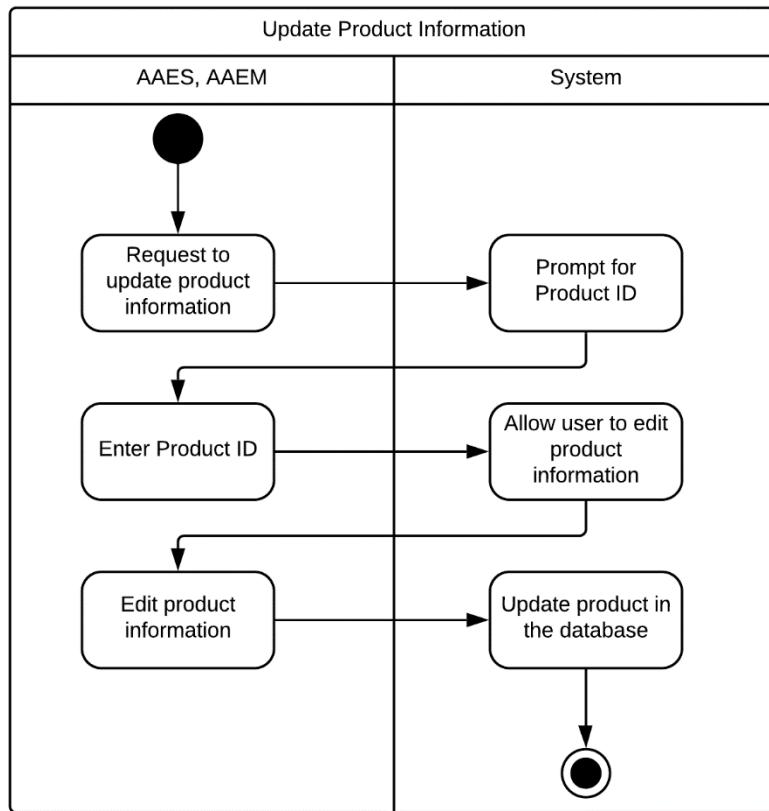
<b>Use Case Name</b>	Update Product Information	
<b>Scenario</b>	Update an existing product's information	
<b>Triggering Event</b>	A product's information has changed and must be updated in the database	
<b>Brief Description</b>	The information of an existing product is updated	
<b>Actors</b>	AAES, AAEM	
<b>Related Use Cases</b>	Look Up Product Information	
<b>Stakeholders</b>	Client, WFS, WFM, WSM	
<b>Preconditions</b>	The product exists within the system	
<b>Postconditions</b>	The product's information is changed according to the user's request	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to update an existing product's information  2. User inputs Product ID  3. User changes information and indicates desire to save changes  4. User reviews the changes and submits them	1.1 System prompts user to enter the Product ID for the product who's information they wish to update  2.1 System displays the product's information and allows user to edit it  3.1 System prompts user to review their changes  4.1 System updates the product's information in the database
<b>Exception Conditions</b>	2.1 Product ID is invalid	

As an **AAES or AAEM**, I want to **update an existing product's information** so that **DSI has its most recent description, weight, dimensions, etc.**

- The product's information is changed in the system

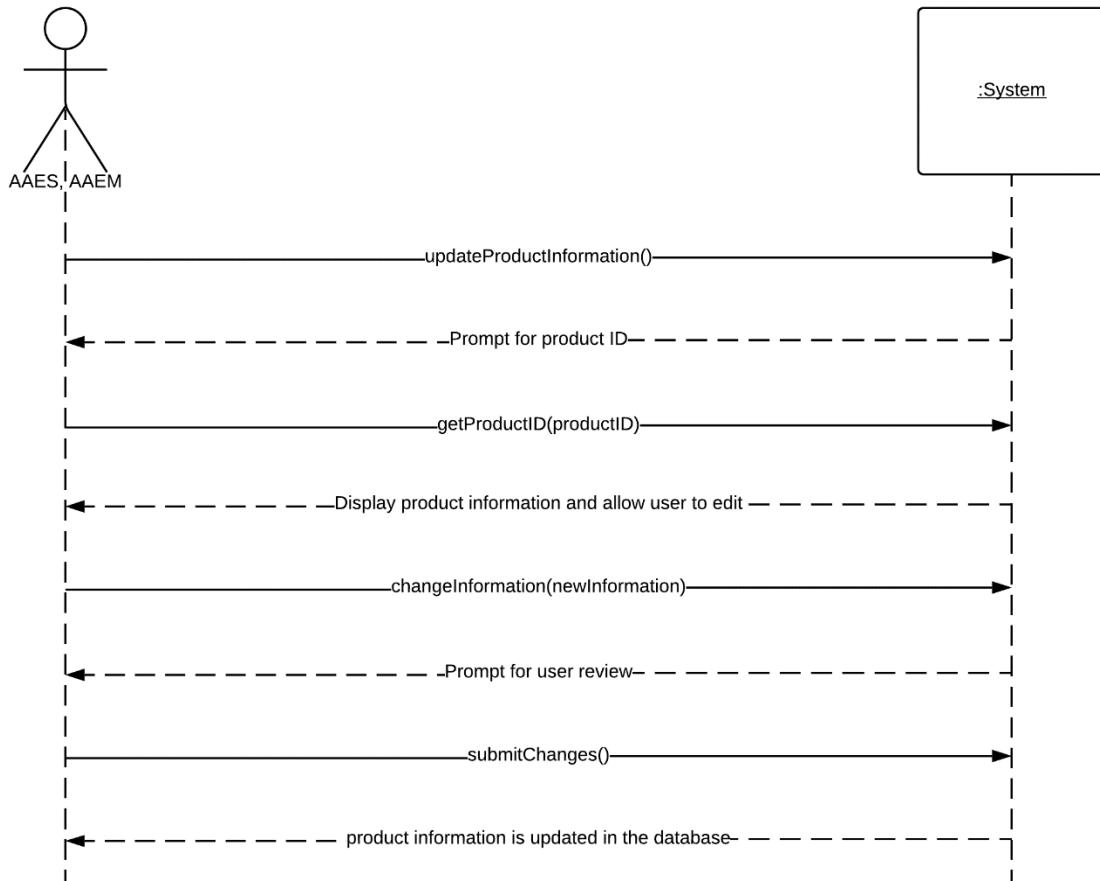
## Update Product Information

Righteous Row



## Update Product Information

Righteous Row



## Use Case #28: Delete Existing Product

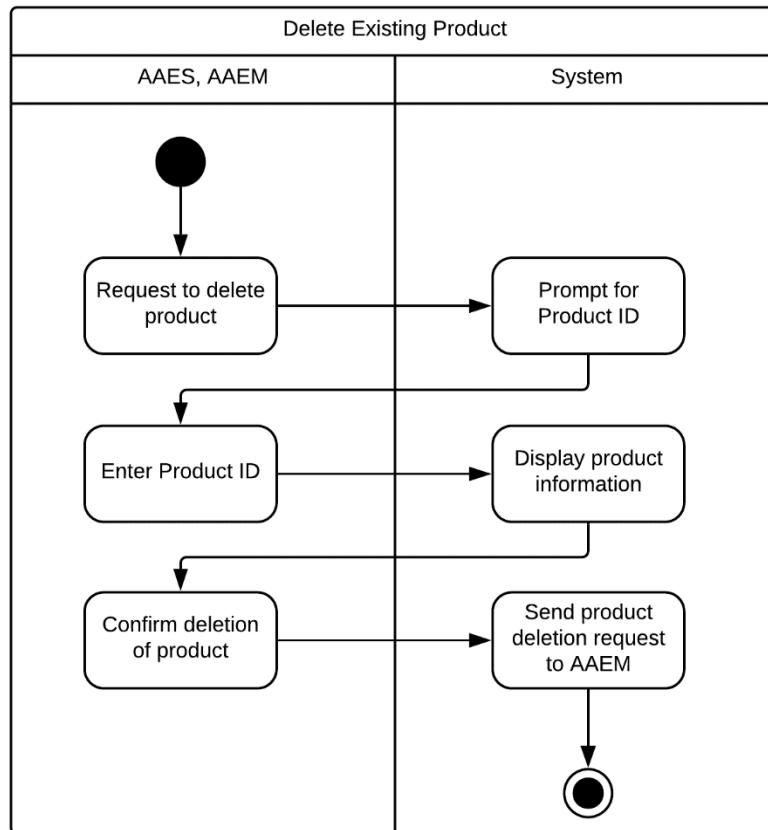
<b>Use Case Name</b>	Delete Existing Product	
<b>Scenario</b>	Delete an existing product from the database	
<b>Triggering Event</b>	A client decides to stop storing a certain product at a DSI warehouse	
<b>Brief Description</b>	An existing product is deleted from the database	
<b>Actors</b>	AAES, AAEM	
<b>Related Use Cases</b>	Approve Product Deletion	
<b>Stakeholders</b>	Clients, WFS, WFM, WSS, WSM	
<b>Preconditions</b>	The product exists within the system	
<b>Postconditions</b>	The request for the product to be deleted is sent to the AAEM	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to delete a product from the database  2. User inputs the Product ID of the product they wish to delete from the database  3. User confirms that they wish to delete the product from the database	1.1 System prompts the user to enter the Product ID of the client to be deleted  2.1 System pulls up the product's information and asks the user to review and confirm that they wish to delete the product from the database  3.1 System sends a request for the product to be deleted to the AAEM
<b>Exception Conditions</b>	2.1 Product ID is invalid	

As an **AAES or AAEM**, I want to **deleting an existing product** so that **DSI's records contain only products that are currently being shipped from DSI**

- A request for approval of the deletion is sent to the AAEM

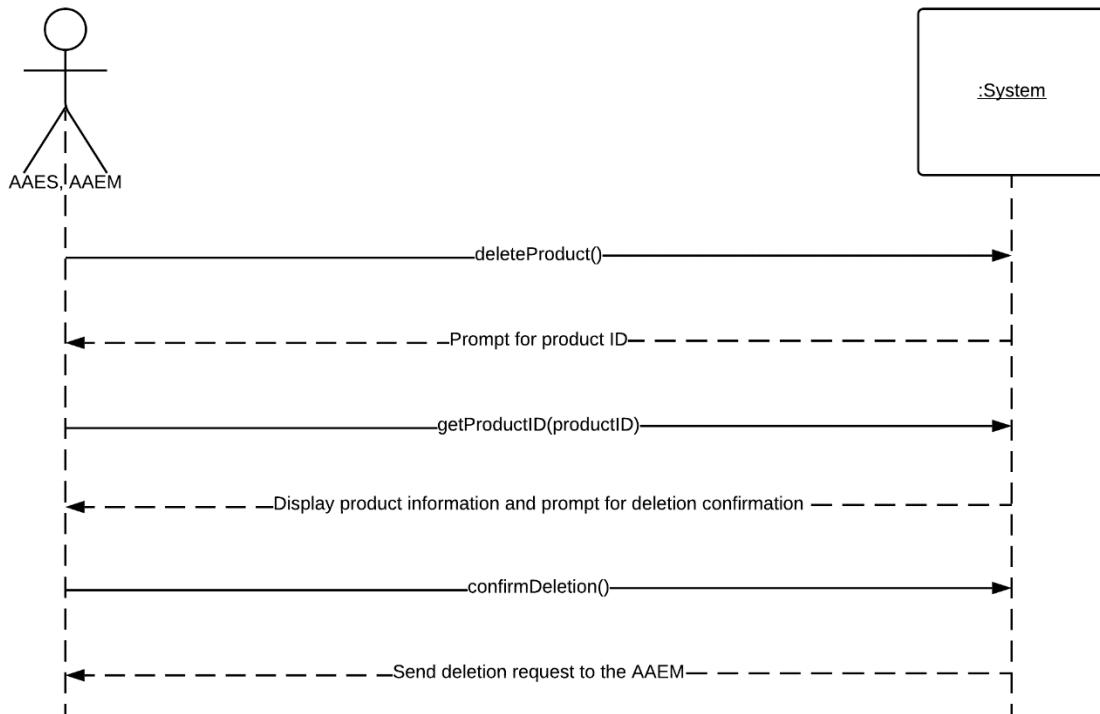
## Delete Existing Product

Righteous Row



## Delete Existing Product

Righteous Row



## Use Case #29: Approve Product Deletion

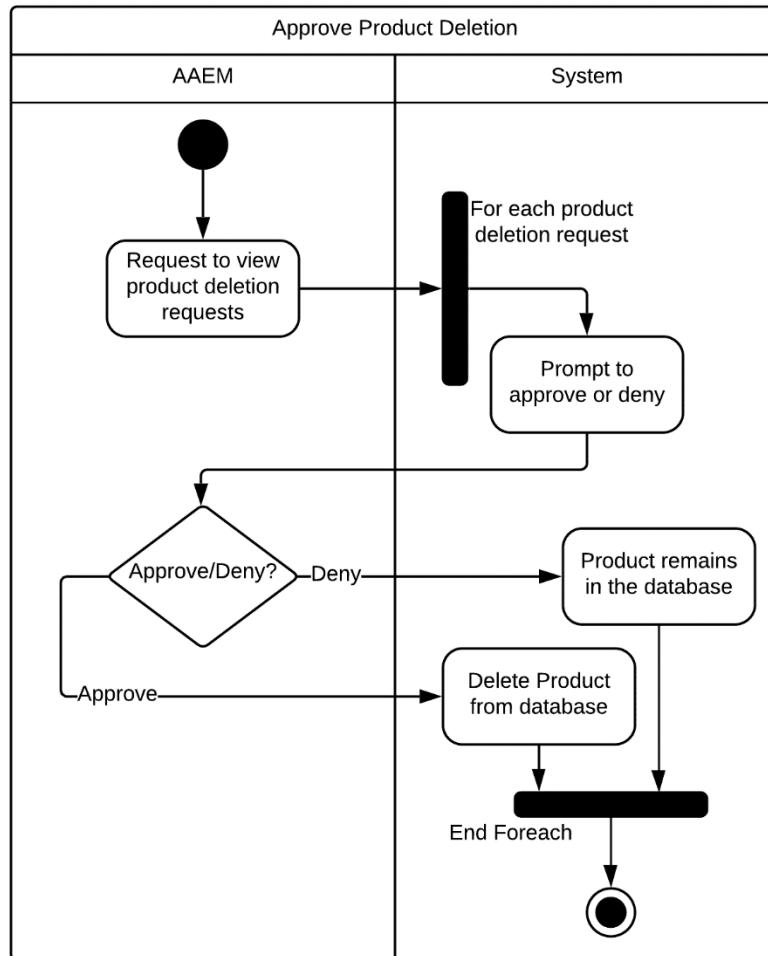
<b>Use Case Name</b>	Approve Product Deletion	
<b>Scenario</b>	Approve the request for a product to be deleted from the database	
<b>Triggering Event</b>	A request to delete a product from the database is sent to the AAEM	
<b>Brief Description</b>	The deletion of a product from the database is reviewed and approved/denied	
<b>Actors</b>	AAEM	
<b>Related Use Cases</b>	Delete Existing Product	
<b>Stakeholders</b>	Clients, WFS, WFM, WSS, WSM	
<b>Preconditions</b>	A request to delete a product from the database has been sent to the AAEM	
<b>Postconditions</b>	If the request is approved, the product is deleted from the database If the request is denied, the product remains in the database	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to approve product deletion requests	1.1 System displays list of product deletion requests that are pending manager approval 1.2 System prompts user to approve or deny the product deletion request
	2. User selects the requests to be approved	2.1 System prompts user to review and approve the list of selected product deletion requests
	3. User confirms that they wish to delete selected products	3.1 System deletes the selected products in the database
<b>Exception Conditions</b>	1.1 There are no active product deletion requests	

As an **AAEM**, I want to **approve product deletion requests** so that **products that are no longer being shipped by DSI can be removed from the database**

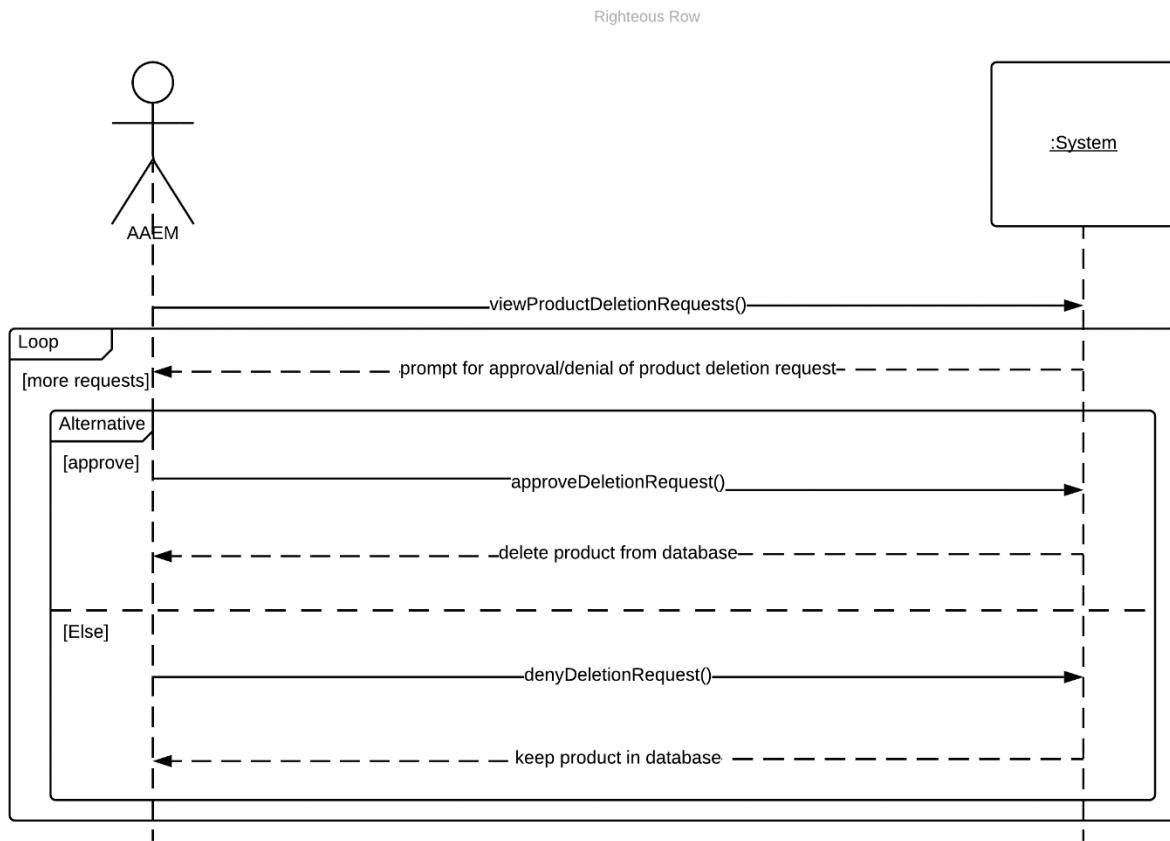
- Product deletion request is sent and approved by the AAEM
- The product's information is deleted from the database

## Approve Product Deletion

Righteous Row



## Approve Product Deletion



## Use Case #30: Update On-Hand Product Quantity

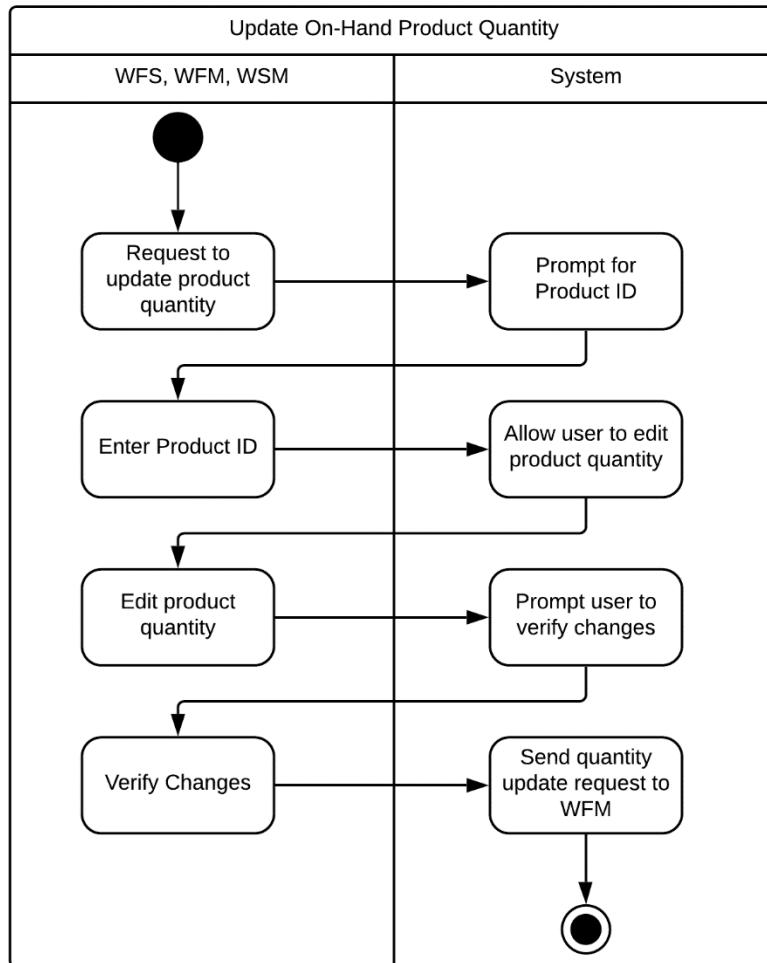
<b>Use Case Name</b>	Update On-Hand Product Quantity	
<b>Scenario</b>	New products are created/existing products are deleted, and the product quantity must be updated to new changes	
<b>Triggering Event</b>	WFM updates product quantity when necessary	
<b>Brief Description</b>	The quantity of a product is updated and changed in the database	
<b>Actors</b>	WFS, WFM, WSM	
<b>Related Use Cases</b>	Approve Product Quantity Update	
<b>Stakeholders</b>	WFS, WFM, WSM, AAES, AAEM	
<b>Preconditions</b>	Product exists within the database	
<b>Postconditions</b>	The quantity of the product is updated in inventory	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to update the quantity of a product	1.1 System prompts user for the product ID of the product to be updated
	2. User enters product ID	2.1 System displays product information and allows quantity to be edited
	3. User changes and submits quantity update	3.1 System prompts user to verify their changes
	4. User verifies their changes	4.1 System sends a quantity update request to the WFM
<b>Exception Conditions</b>	2.1 Product ID is invalid	

As a **WFS, WFM, or WSM**, I want to **update the quantity of items in the database** so that **the quantity displayed in the system accurately reflects the quantity of the product DSI has on hand**

- A request for approval of the product quantity update is sent to the WFM

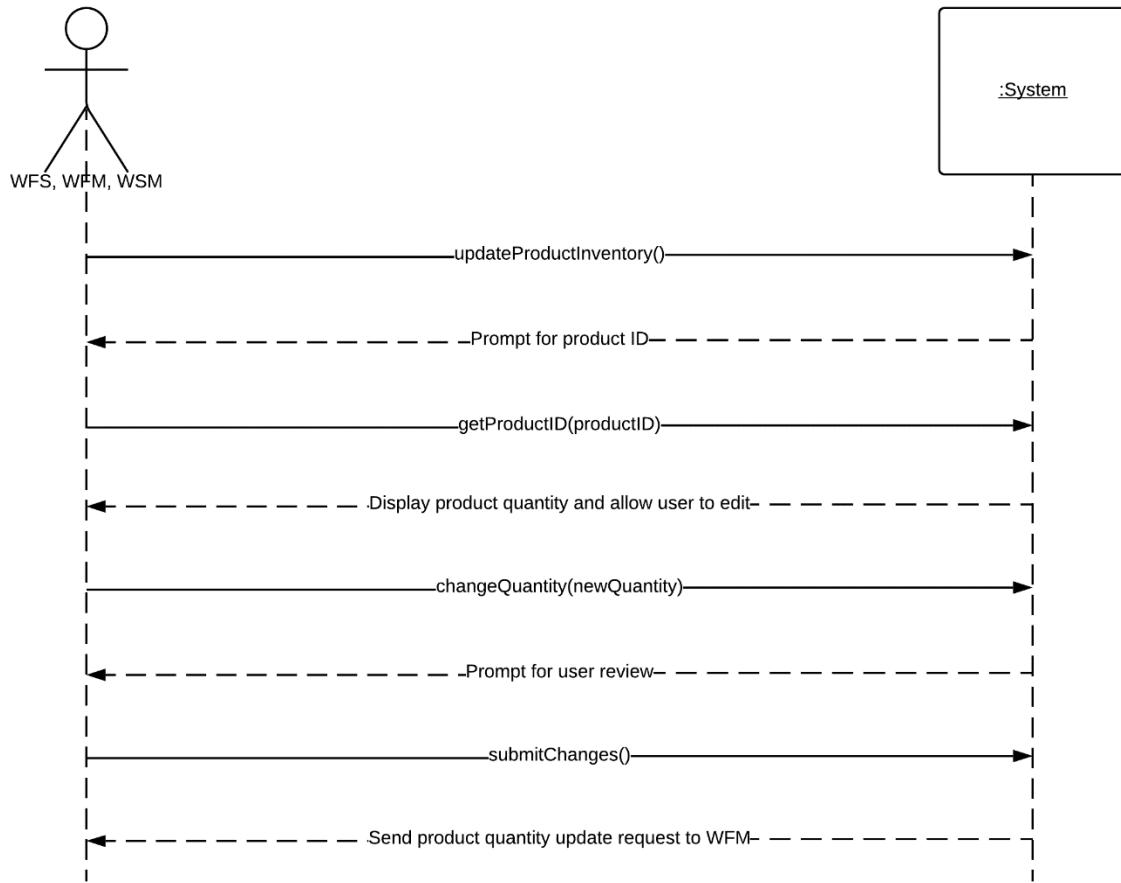
## Update On-Hand Product Quantity

Righteous Row



## Update On-Hand Product Quantity

Righteous Row



## Use Case #31: Approve Product Quantity Update

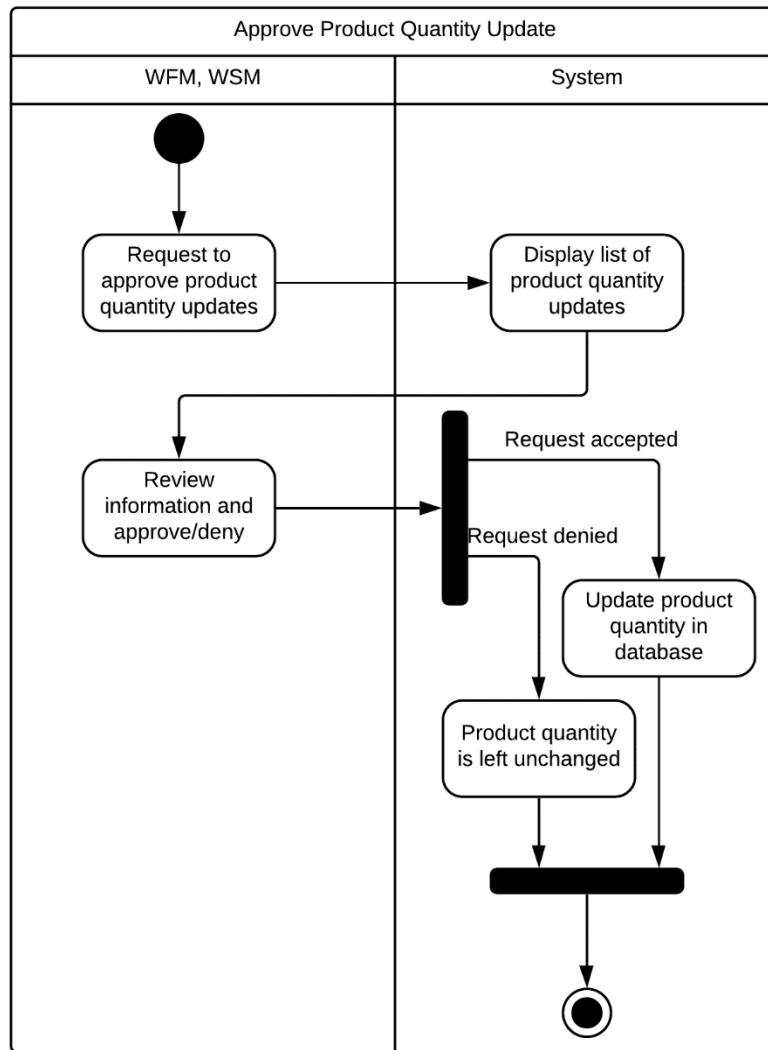
<b>Use Case Name</b>	Approve Product Quantity Update	
<b>Scenario</b>	Product quantity update request is approved	
<b>Triggering Event</b>	A product quantity update request is sent to the WFM	
<b>Brief Description</b>	The quantity update request of a product is reviewed and approved/denied	
<b>Actors</b>	WFM, WSM	
<b>Related Use Cases</b>	Update On-Hand Product Quantity	
<b>Stakeholders</b>	WFS, WFM, WSM, AAES, AAEM	
<b>Preconditions</b>	A product quantity update request has been sent to the WFM	
<b>Postconditions</b>	Product quantity is updated in inventory	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to approve product quantity update requests  2. User selects the requests to be confirmed  3. User confirms that they wish to update selected product quantities	1.1 System displays list of product quantity updates that are pending manager approval to be updated 1.2 System prompts user to approve or deny the product quantity update  2.1 System prompts user to review and approve the list of selected product quantity updates  3.1 System updates the quantities in the database
<b>Exception Conditions</b>	1.1 There are no active product quantity update requests	

As a **WFM or WSM**, I want to **approve product quantity update requests** so that **product quantities are changed within the system**

- Product quantity update request is sent and approved by the WFM
- The product's quantity is updated in the database

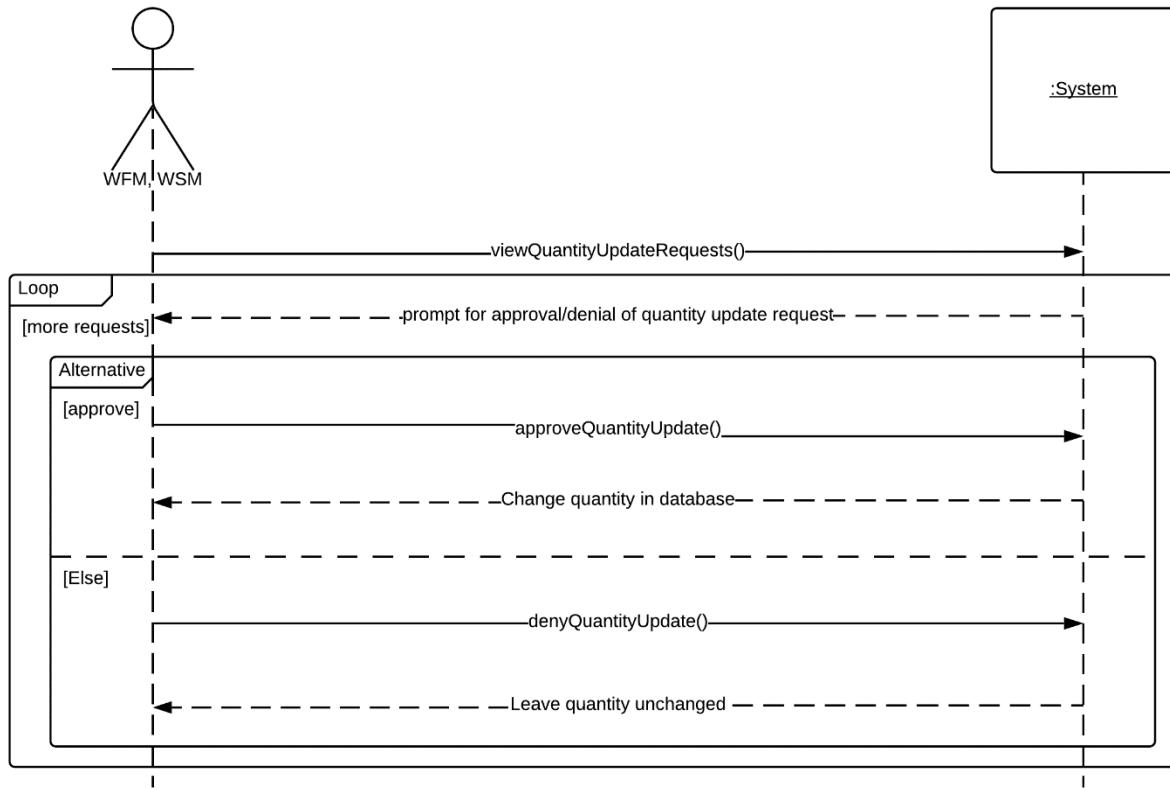
## Approve Product Quantity Update

Righteous Row



## Approve Product Quantity Update

Righteous Row



## Use Case #32: Update Product Location

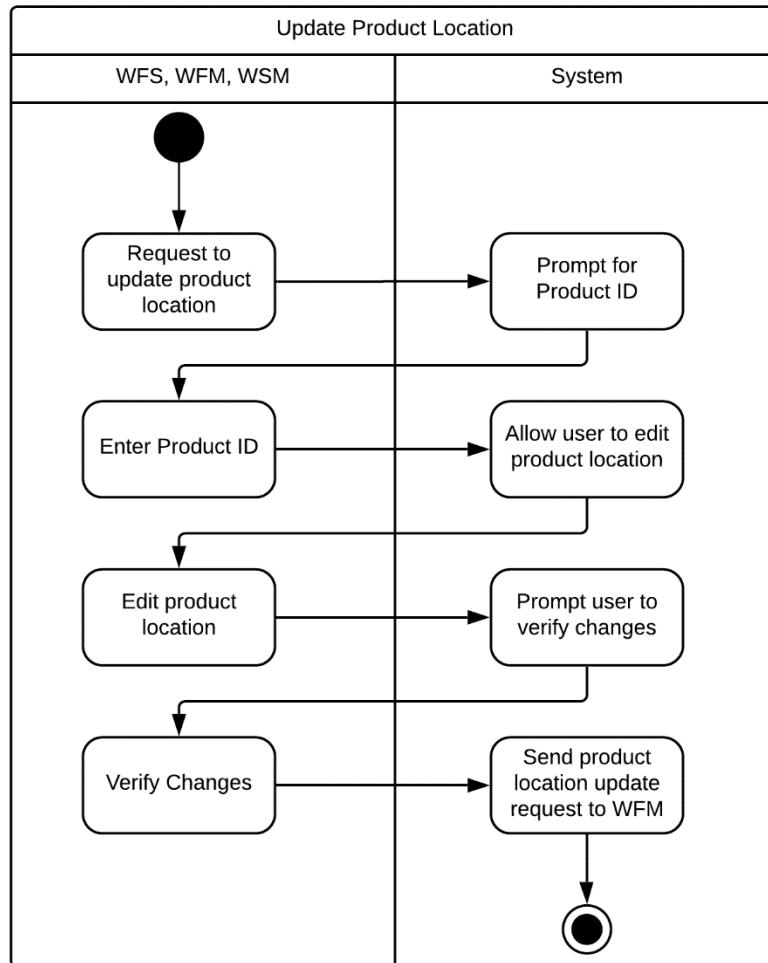
<b>Use Case Name</b>	Update Product Location	
<b>Scenario</b>	Update the location of an existing product	
<b>Triggering Event</b>	The location of an existing product is changed and must be updated	
<b>Brief Description</b>	The location of a product in the warehouse is changed in the database	
<b>Actors</b>	WFS, WFM, WSM	
<b>Related Use Cases</b>	Approve Product Location Change	
<b>Stakeholders</b>	AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM	
<b>Preconditions</b>	The product exists within the system	
<b>Postconditions</b>	The location of the product is changed in the database	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to update the location of a product	1.1 System prompts user to input product ID of the product
	2. User enters product ID	2.1 System displays the product's location and allows user to edit it
	3. User edits the product's location	3.1 System prompts user to review their changes
	4. User reviews and submits their changes	4.1 System sends a product location update request to the WFM
<b>Exception Conditions</b>	2.1 Product ID is invalid	

As a **WFS, WFM, or WSM**, I want to **update the location of a product in the database** so that **the accurate warehouse location of the item is displayed**

- A request for approval of the product location update is sent to the WFM

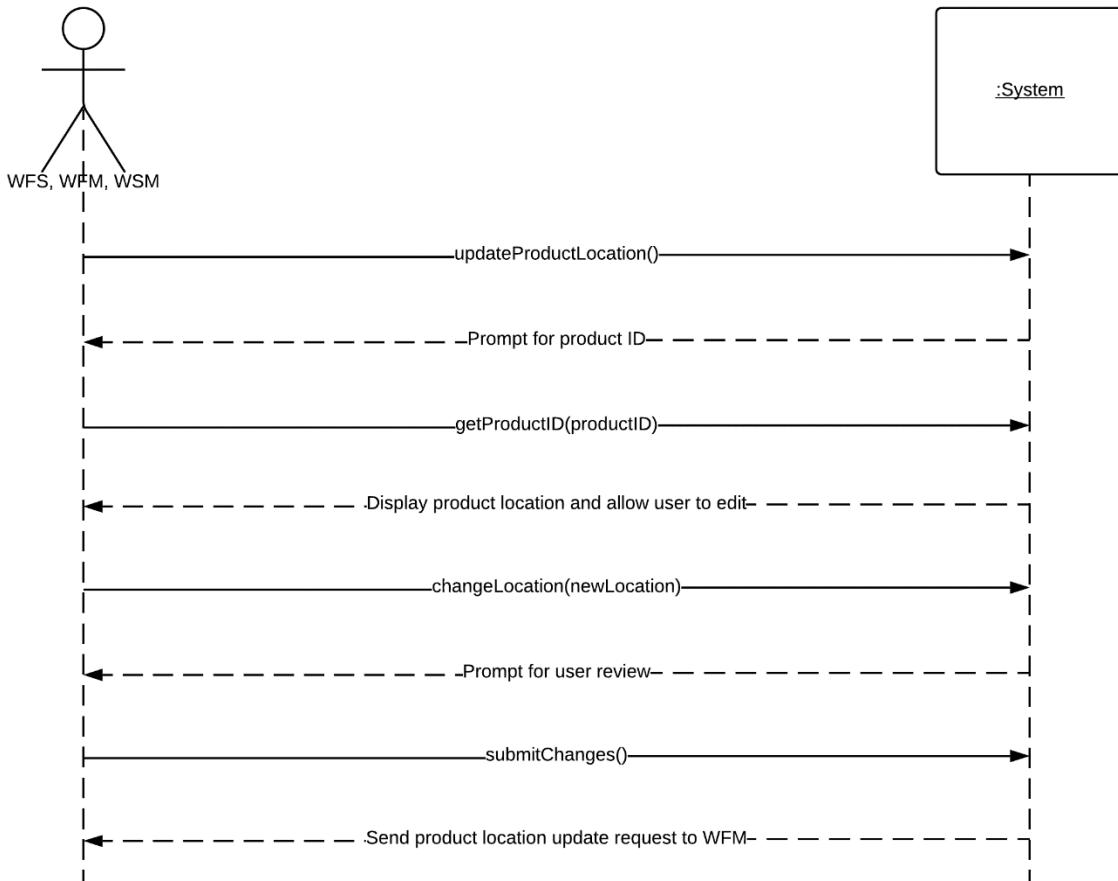
## Update Product Location

Righteous Row



## Update Product Location

Righteous Row



## Use Case #33: Approve Product Location Change

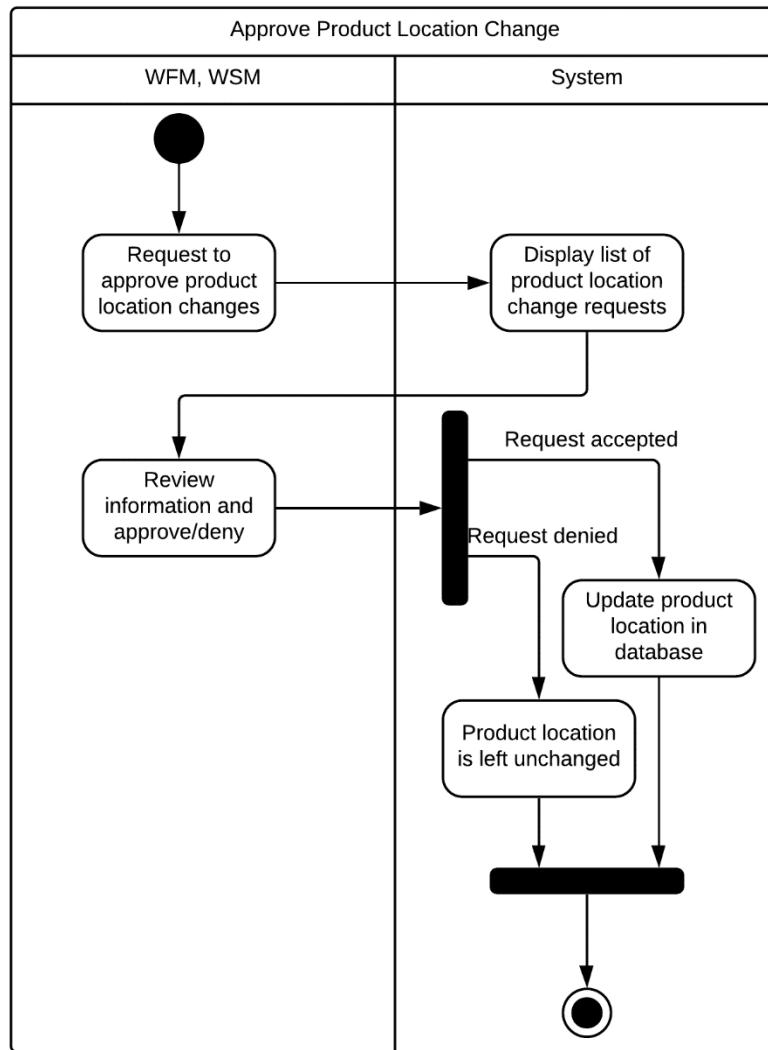
<b>Use Case Name</b>	Approve Product Location Change	
<b>Scenario</b>	Product location change request is approved	
<b>Triggering Event</b>	A product location change request is sent to the WFM	
<b>Brief Description</b>	The location change of a product is reviewed and approved/denied	
<b>Actors</b>	WFM, WSM	
<b>Related Use Cases</b>	Update Product Location	
<b>Stakeholders</b>	AAES, AAEM, WFS, WFM, WSS, WSM, WRS, WRM	
<b>Preconditions</b>	A product location change request has been sent to the WFM	
<b>Postconditions</b>	The product's location in the database is changed	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire to approve any product location change requests  2. User reviews the location change of each request, and marks them as either approved or denied  3. User reviews and submits choices	1.1 System displays list of product location changes pending manager approval 1.2 System prompts user to review and approve/deny the creation of each request  2.1 System prompts user to review and submit their changes  3.1 System updates the product location in the database
<b>Exception Conditions</b>	1.1 There are no active product location change requests	

As a **WFM or WSM**, I want to **approve product location change requests** so that **product locations are changed within the system**

- Product quantity update request is sent and approved by the WFM
- The product's quantity is updated in the database

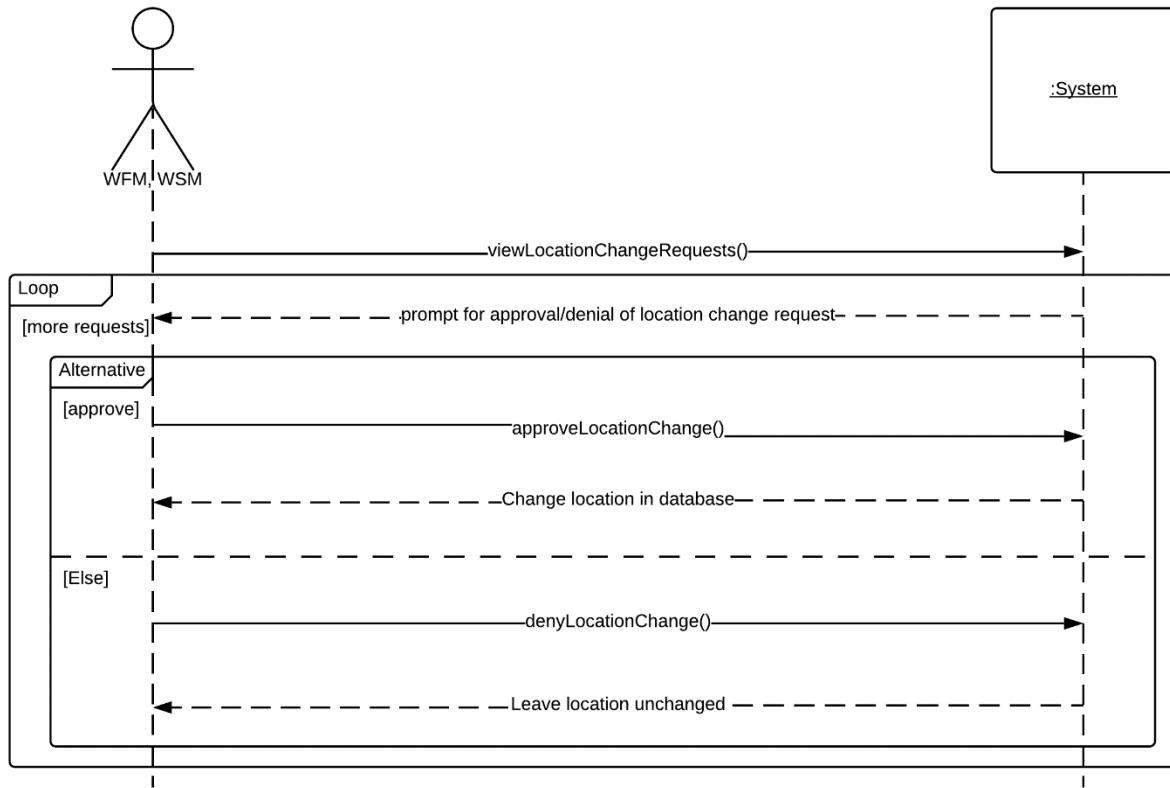
## Approve Product Location Change

Righteous Row



## Approve Product Location Change

Righteous Row



## Use Case #34: Assign New Product to Location

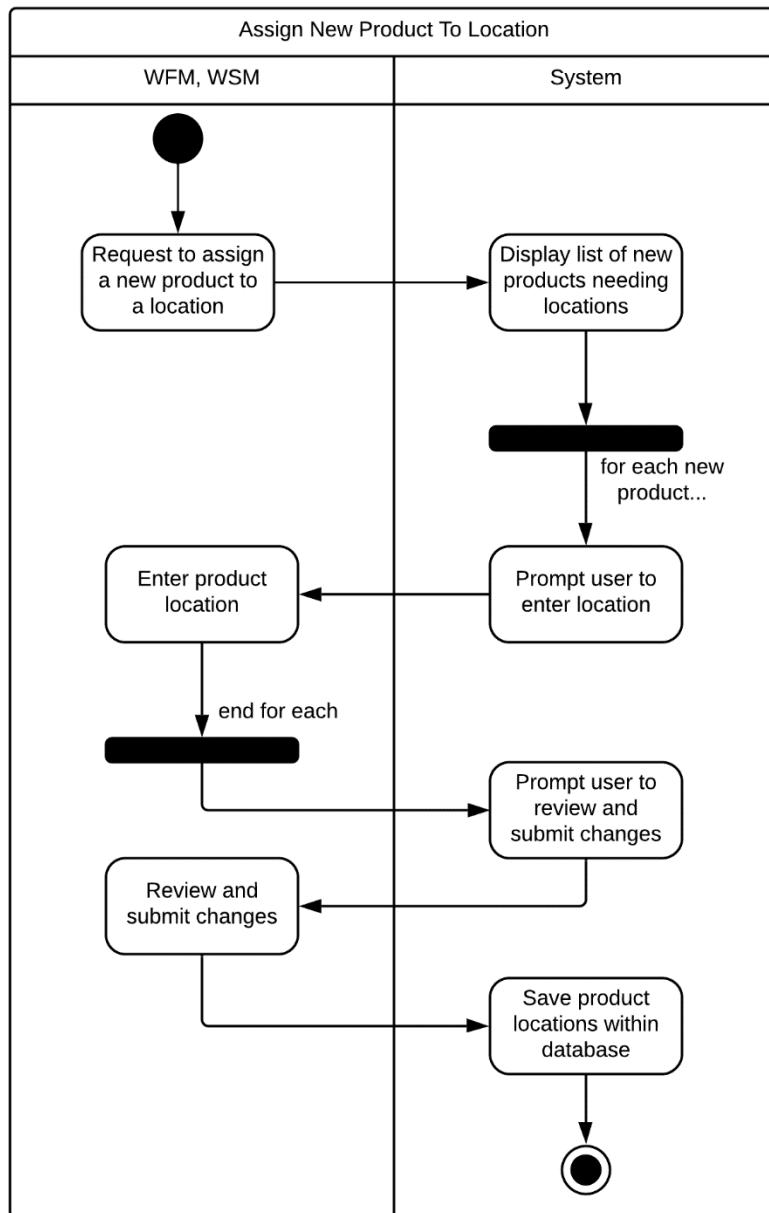
<b>Use Case Name</b>	Assign New Product to Location	
<b>Scenario</b>	A product that is new to DSI is assigned to a new location in the warehouse	
<b>Triggering Event</b>	A product not in the database arrives at DSI to be stored	
<b>Brief Description</b>	A new product in the inventory is assigned to a location in the warehouse	
<b>Actors</b>	WFM, WSM	
<b>Related Use Cases</b>	Create New Product	
<b>Stakeholders</b>	Client, WFS, WFM, WSS, WSM, WRS, WRM	
<b>Preconditions</b>	The product does not exist within the DSI database	
<b>Postconditions</b>	The product is created and assigned a location within a warehouse	
<b>Flow of Activities</b>	User	System
	1. User indicates desire to assign a new product a location within the warehouse	1.1 System displays list of new products needing locations 1.2 For each new product, system prompts user to enter location
	2. User enters a location for each product	2.1 While there are still products, return to step 1.2 2.2 System displays list of new products and their locations, and prompts user to review and submit their changes
	3. User reviews and submits their changes	3.1 Product locations are saved within the database
<b>Exception Conditions</b>	1.1 There are no active new products that need to be assigned to locations	

As a **WFM or WSM**, I want to **approve product quantity update requests** so that **product quantities are changed within the system**

- Product quantity update request is sent and approved by the WFM
- The product's quantity is updated in the database

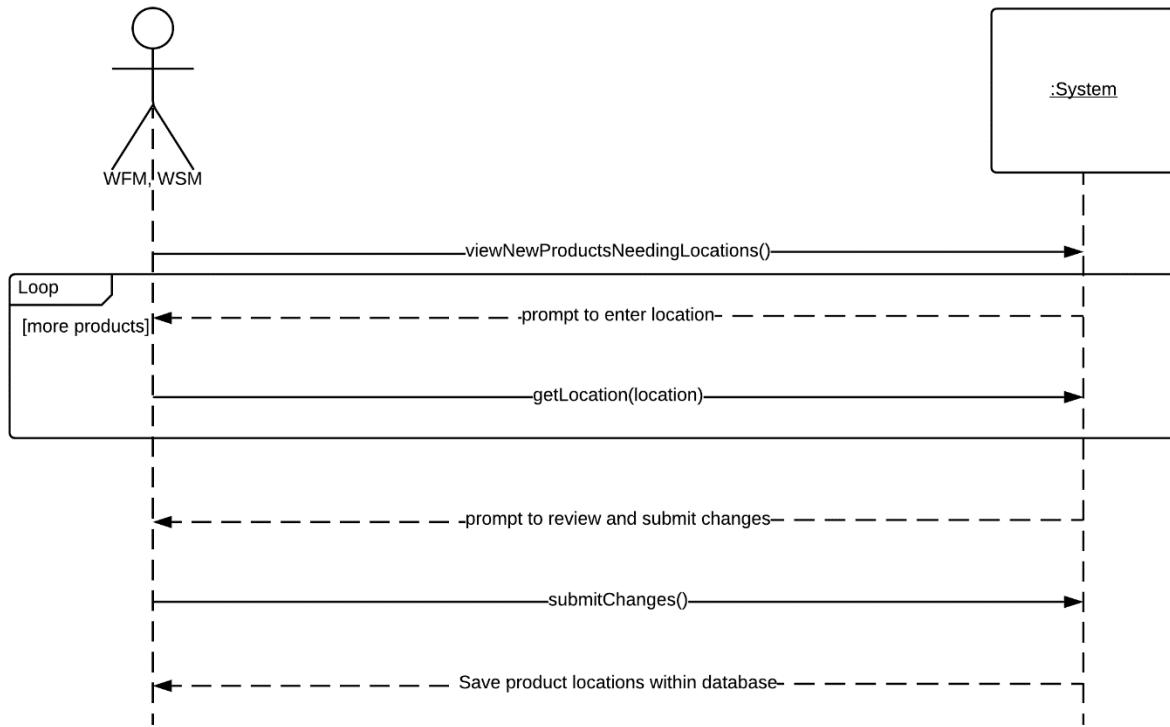
## Assign New Product To Location

Righteous Row



## Assign New Product To Location

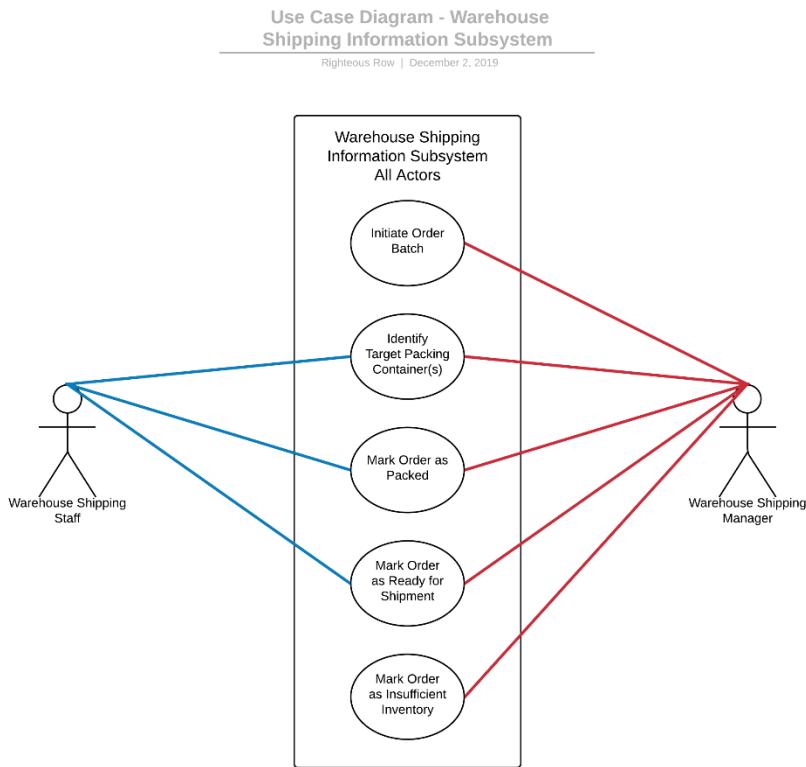
Righteous Row



## *Tabular Use Case Definitions – Warehouse Shipping Subsystem*

Warehouse Shipping Information Subsystem		
Use Case Name	Actors	Brief Description
Initiate Order Batch	WSM	The order batch is initiated, and staff are assigned to workstations
Identify Target Packing Container(s)	WSS, WSM	Given the items in an order, the system recommends a specific size of box for it to be packaged in
Mark Order as Packed	WSS, WSM	An order is marked as “packed” in the system
Mark Order as Ready for Shipment	WSS, WSM	An order is marked as “ready for shipment” in the system
Mark Order as Insufficient Inventory	WSM	An order is marked as “insufficient inventory” in the system if there is not enough of a product in the order to complete it

## *Graphical Use Case Definitions – Warehouse Shipping Subsystem*



# *Fully Developed Use Case Descriptions with User Stories, Activity Diagrams, and System Sequence Diagrams – Warehouse Shipping Subsystem*

## Use Case #35: Initiate Order Batch

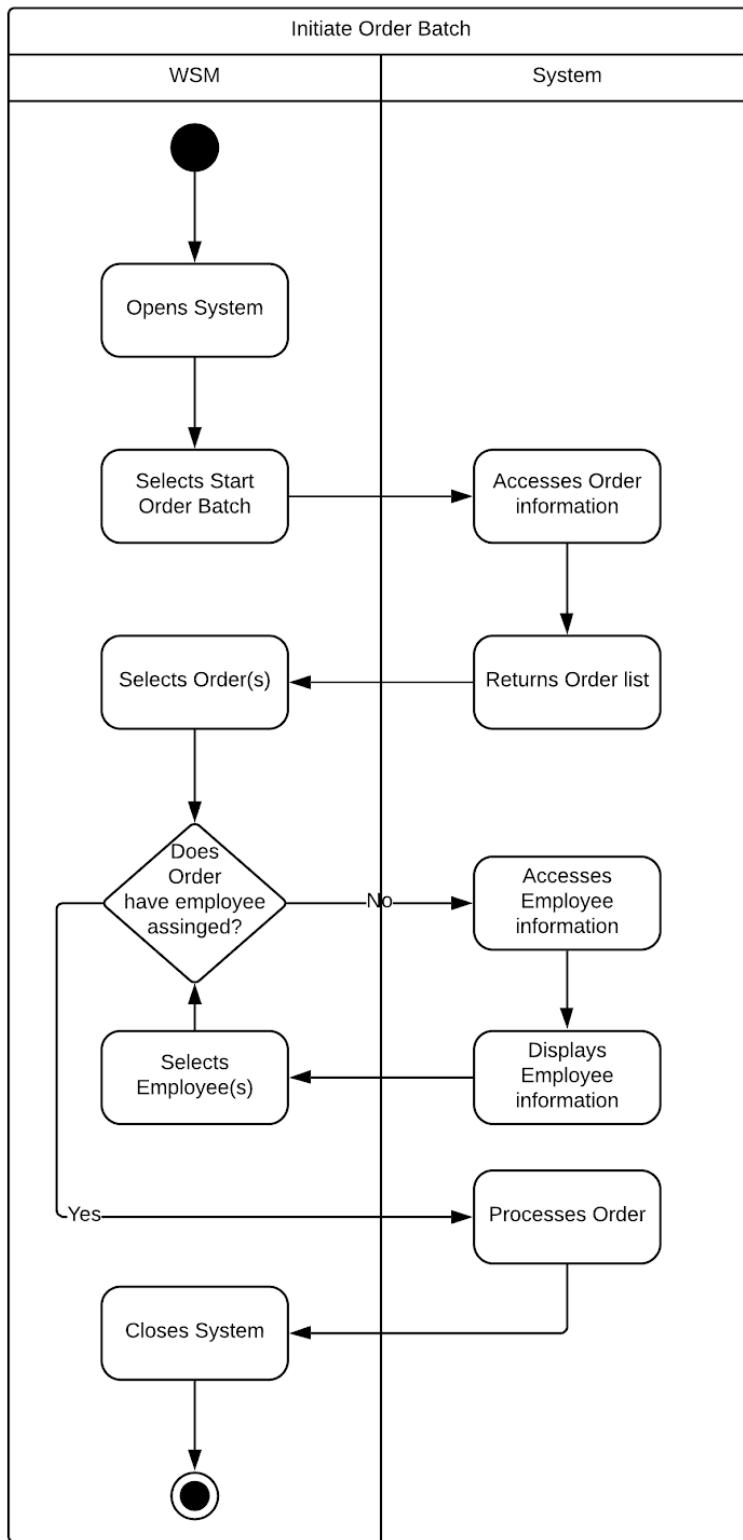
<b>Use Case Name</b>	Initiate Order Batch											
<b>Scenario</b>	The WSM initiates a group of orders to be processed											
<b>Triggering Event</b>	Two or more orders are received											
<b>Brief Description</b>	The order batch is initiated, and staff are assigned to workstations											
<b>Actors</b>	WSM											
<b>Related Use Cases</b>	Create New Order, Cancel Existing Order											
<b>Stakeholders</b>	WSM, WSS, Client											
<b>Preconditions</b>	Two or more completed order requests are made and validated											
<b>Postconditions</b>	Workers are assigned to workstations. Order are properly sequenced.											
<b>Flow of Activities</b>	<table border="1"> <thead> <tr> <th>User</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. User indicates desire to initiate order batch</td> <td>1.1 System look up orders 1.2 System returns order list</td> </tr> <tr> <td>2. User selects order(s) from list</td> <td>2.1 System displays order batch information 2.2 System displays employee selection list</td> </tr> <tr> <td>3. User selects warehouse employee(s) from the list</td> <td>3.1 System initiates user order</td> </tr> <tr> <td>4. User repeats action 3 for each order</td> <td>4.1 System processes batch</td> </tr> </tbody> </table>	User	System	1. User indicates desire to initiate order batch	1.1 System look up orders 1.2 System returns order list	2. User selects order(s) from list	2.1 System displays order batch information 2.2 System displays employee selection list	3. User selects warehouse employee(s) from the list	3.1 System initiates user order	4. User repeats action 3 for each order	4.1 System processes batch	
User	System											
1. User indicates desire to initiate order batch	1.1 System look up orders 1.2 System returns order list											
2. User selects order(s) from list	2.1 System displays order batch information 2.2 System displays employee selection list											
3. User selects warehouse employee(s) from the list	3.1 System initiates user order											
4. User repeats action 3 for each order	4.1 System processes batch											
<b>Exception Conditions</b>	1.2 There are no active orders 2.1 Client cancels existing order											

As a **WSM**, I want to **initiate an order batch** so that **the shipping process can begin**

- Orders are successfully assigned to shipping staff

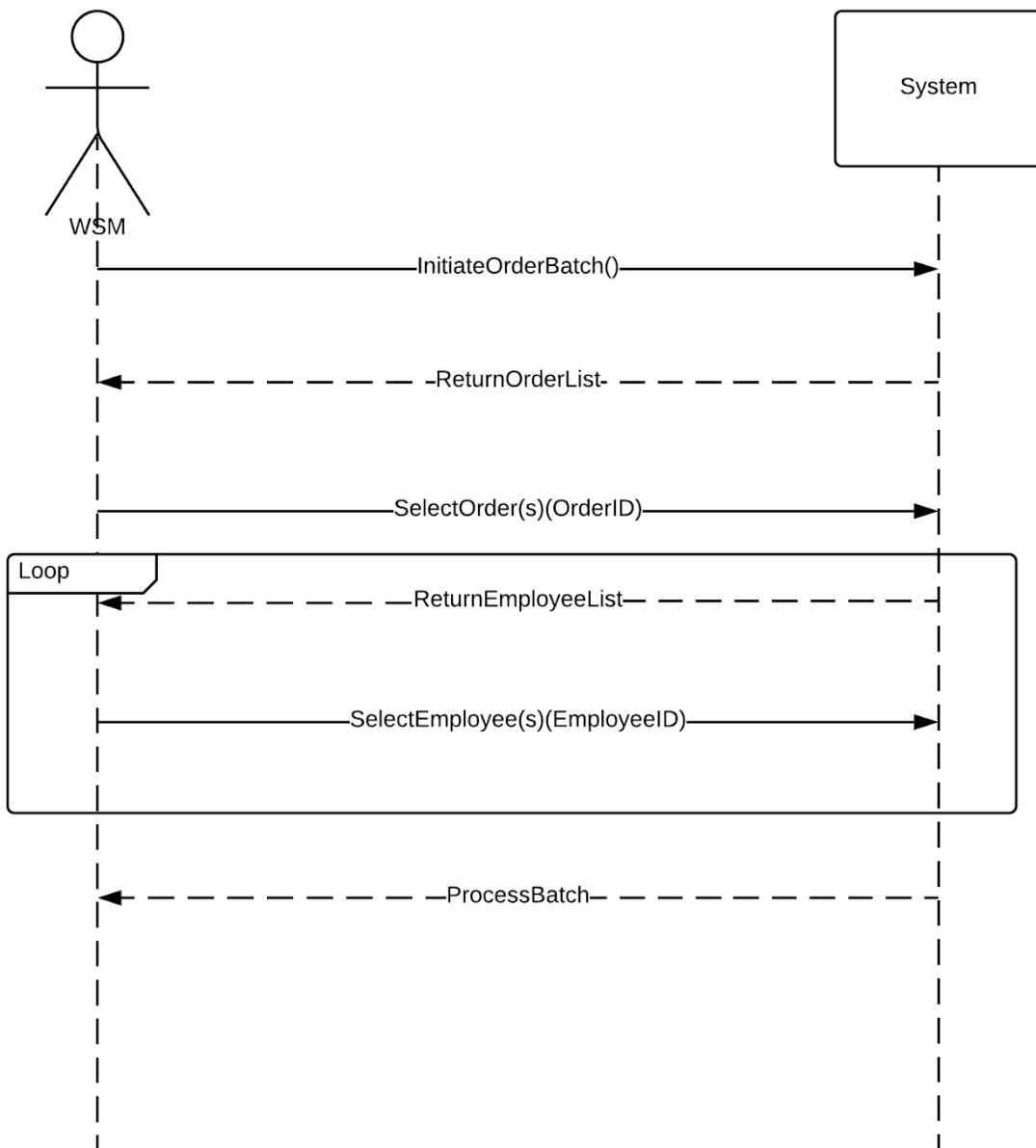
## Initiate Order Batch

Righteous Row



## Initiate Order Batch

Righteous Row



## Use Case #36: Identify Target Packing Container(s)

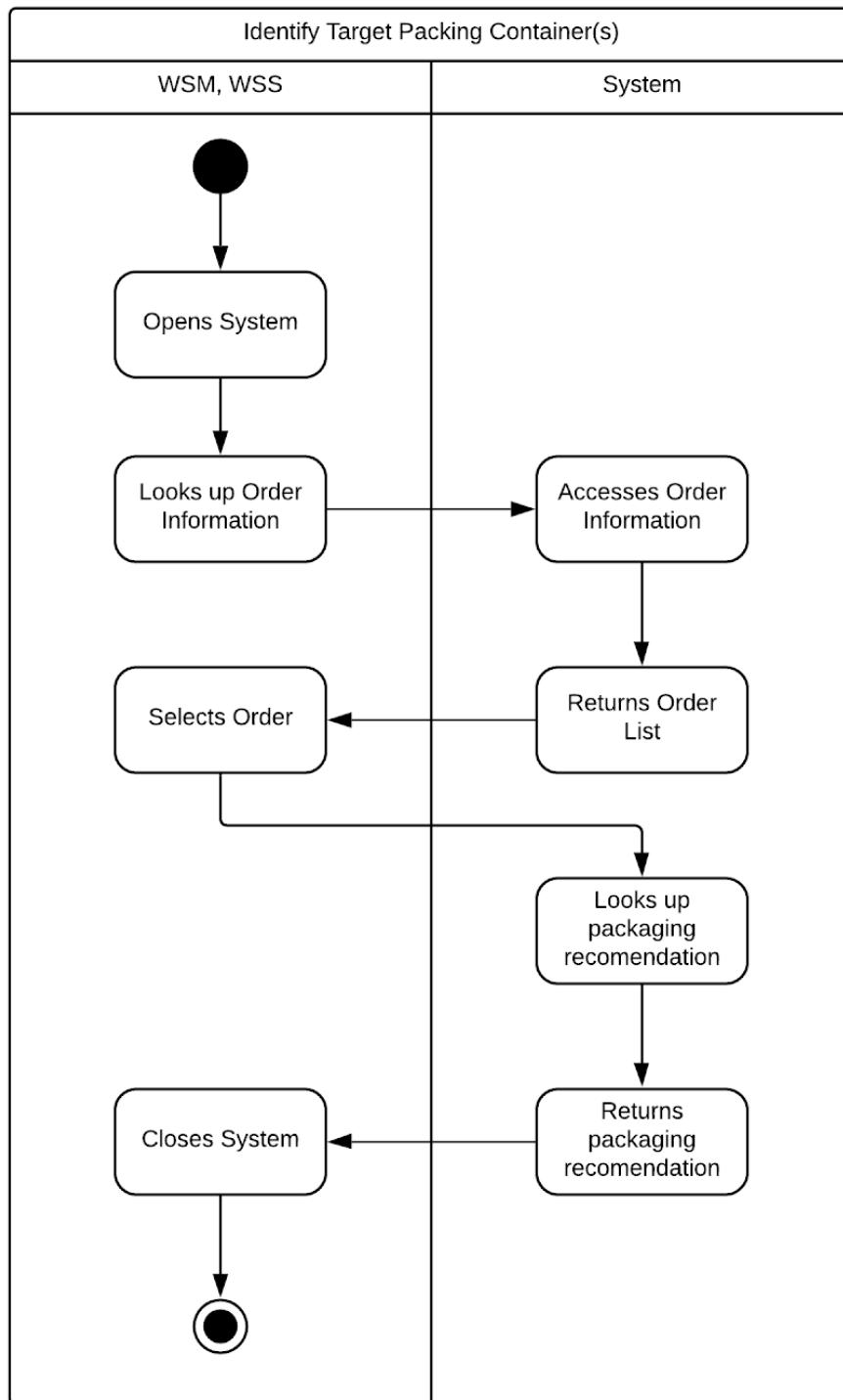
<b>Use Case Name</b>	Identify Target Packing Container(s)	
<b>Scenario</b>	System recommends box sizes	
<b>Triggering Event</b>	An order is initiated and not yet packed	
<b>Brief Description</b>	Given the items in an order, the system recommends a specific size of box for it to be packaged in	
<b>Actors</b>	WSS, WSM	
<b>Related Use Cases</b>	Initiate Order Batch, Look Up Order Information	
<b>Stakeholders</b>	WSS, WSM, Client	
<b>Preconditions</b>	An Order is initiated	
<b>Postconditions</b>	A box size is recommended based on the products in the order	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User indicates desire for packing information	1.1 System looks up order information 1.2 System returns order list
	2. User selects order	2.1 System matches order against algorithm 2.2 System returns packaging recommendation list
<b>Exception Conditions</b>	1.2 There are no active orders	

As a **WSS or WSM**, I want to **identify target packing containers** so that **the orders that I am shipping are placed in appropriate sized boxes**

- User receives a recommendation of what box size to use for the items in a specific order

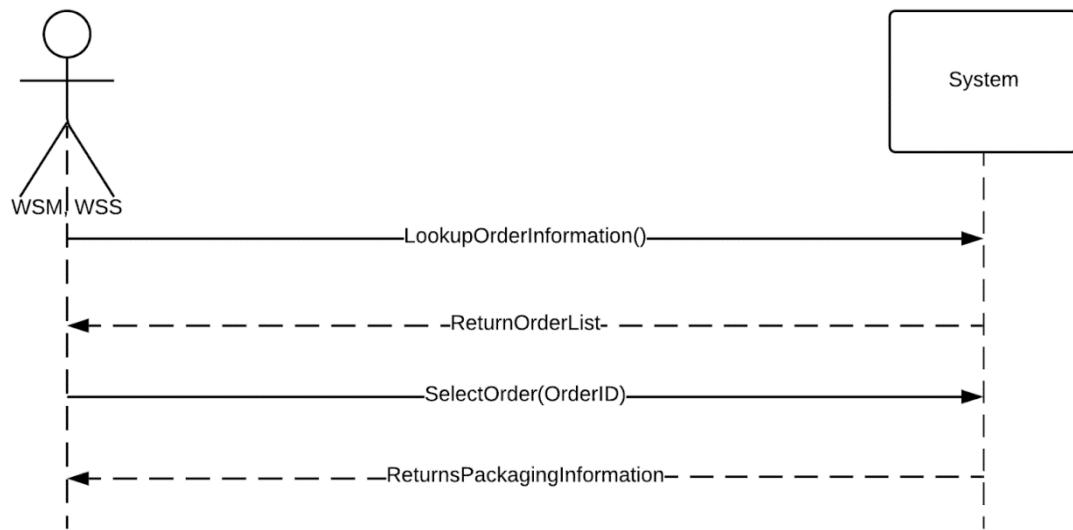
## Identify Target Packing Container(s)

Righteous Row



## Identify Target Packing Container(s)

Righteous Row



## Use Case #37: Mark Order As Packed

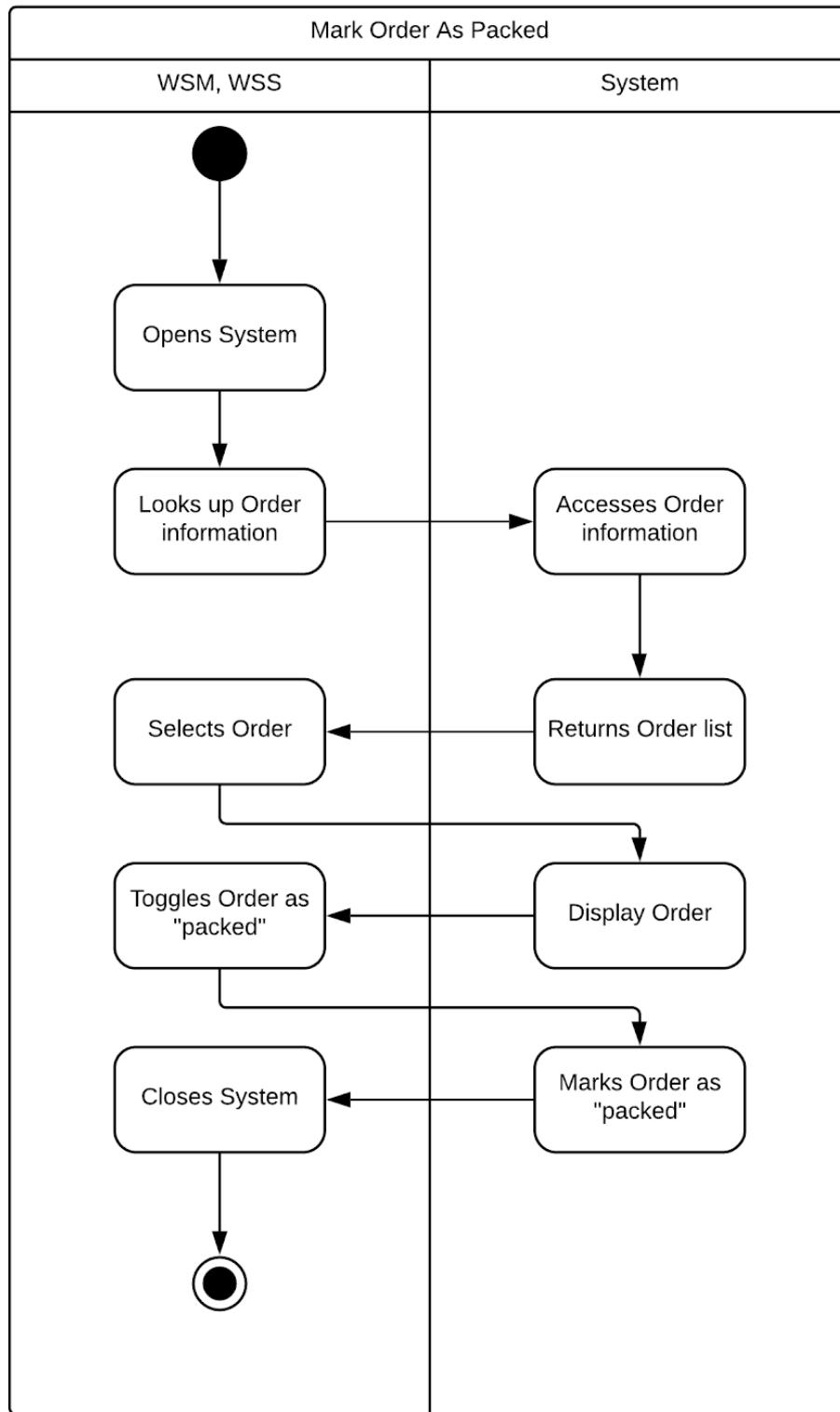
<b>Use Case Name</b>	Mark Order As Packed	
<b>Scenario</b>	An employee marks an order as “packed”	
<b>Triggering Event</b>	A Warehouse Shipping Staff/Manager finishes packing an order.	
<b>Brief Description</b>	An order is marked as “packed” in the system	
<b>Actors</b>	WSS, WSM	
<b>Related Use Cases</b>	Mark Order as Ready for Shipment, Look Up Order Information	
<b>Stakeholders</b>	WSS, WSM, Client	
<b>Preconditions</b>	The order to be packed exists within the system The order is not yet packed	
<b>Postconditions</b>	The order is marked as “packed” The order’s shipping label is printed	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User queries opens order.	1.1 System looks up order information 1.2 System displays list of orders.
	2. User selects order to be marked as “packed”	2.1 System displays order.
	3. User toggles “unpacked” to “packed”	3.1 System marks order information as “packed” 3.2 System prints shipping label
<b>Exception Conditions</b>	1.2 There are no active orders	

As a **WSS or WSM**, I want to **mark an order as packed** so that **the order can be prepared for shipment**

- Order is marked as packed in the database
- A shipping label is printed

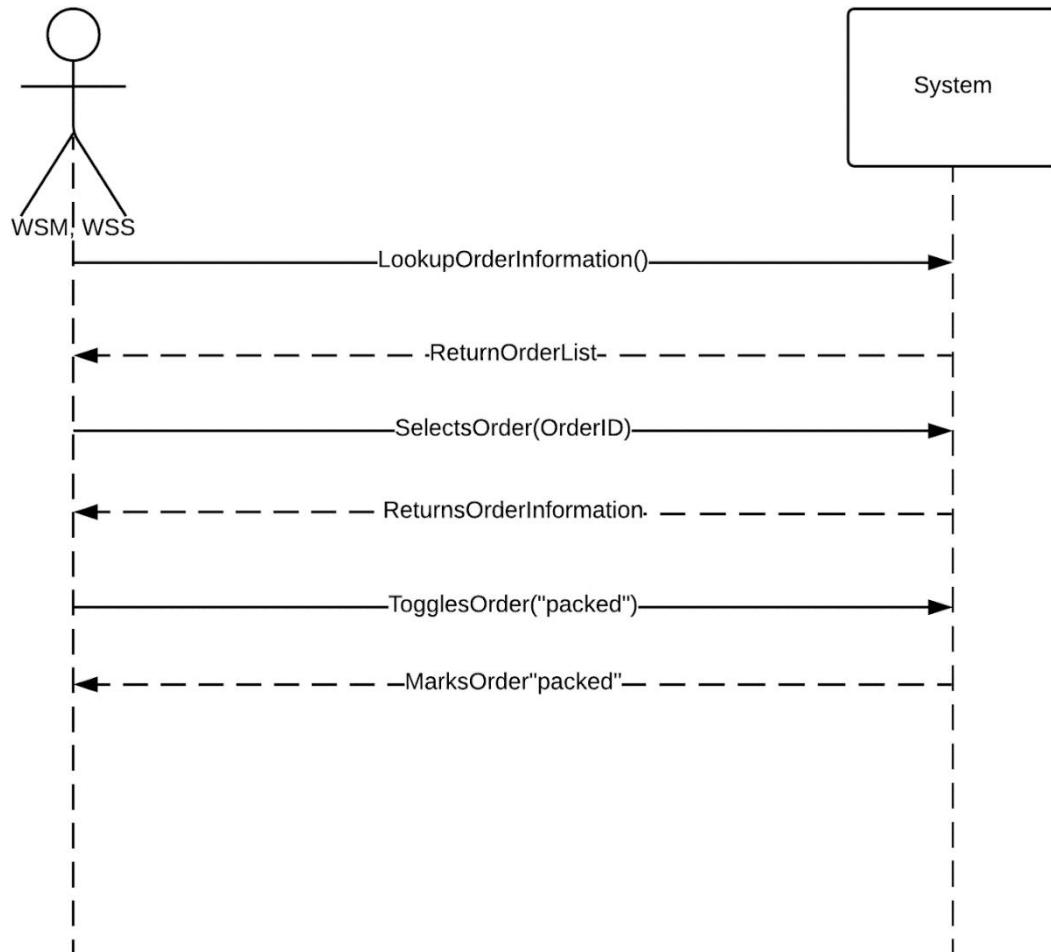
## Mark Order as Packed

Righteous Row



## Mark Order as Packed

Righteous Row



## Use Case #38: Mark Order As Ready For Shipment

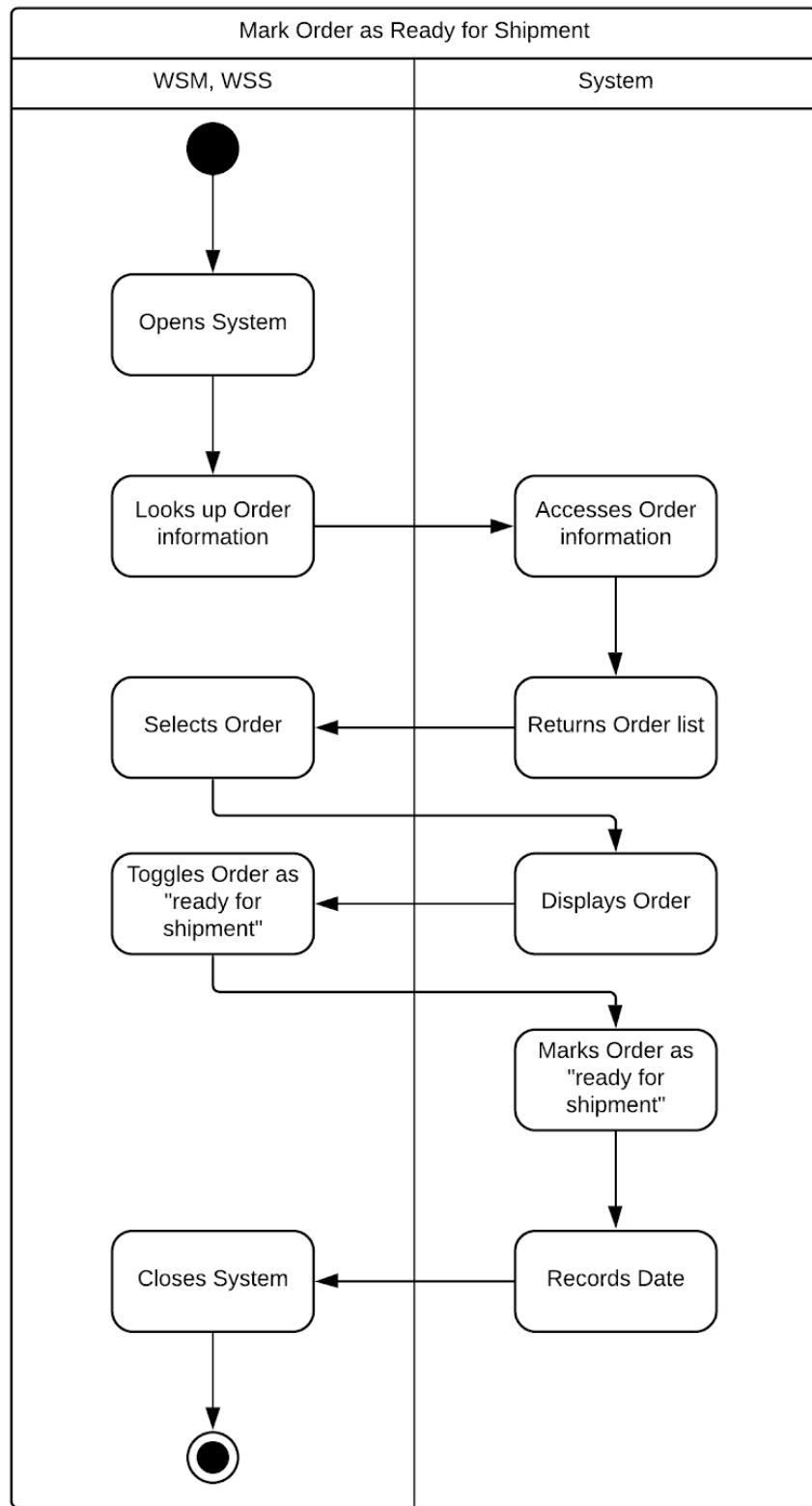
<b>Use Case Name</b>	Mark Order As Ready for Shipment	
<b>Scenario</b>	An employee marks an order as “ready for shipment”	
<b>Triggering Event</b>	The shipping label for an order is printed, and the order status needs to be changed	
<b>Brief Description</b>	An order is marked as “ready for shipment” in the system	
<b>Actors</b>	WSS, WSM	
<b>Related Use Cases</b>	Mark Order As Packed, Look Up Product Information	
<b>Stakeholders</b>	WSS, WSM, Client	
<b>Preconditions</b>	The order is already marked as packed The shipping label has been printed for the order	
<b>Postconditions</b>	The order status is changed from “packed” to “ready for shipment”	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User queries open orders	1.1 System looks up order information 1.2 System displays list of orders
	2. User selects order to be marked “ready for shipment”	2.1 System displays order
	3. User toggles “not ready for shipment” to “ready for shipment”	3.1 System marks order information as “ready for shipment” 3.2 System records date order was marked as “ready for shipment”
<b>Exception Conditions</b>	1.1 There are no active orders	

As a **WSS or WSM**, I want to **mark an order as ready for shipment** so that **it can be shipped to the client**

- Order is marked as ready for shipment in the database
- Order is placed in a shipping container

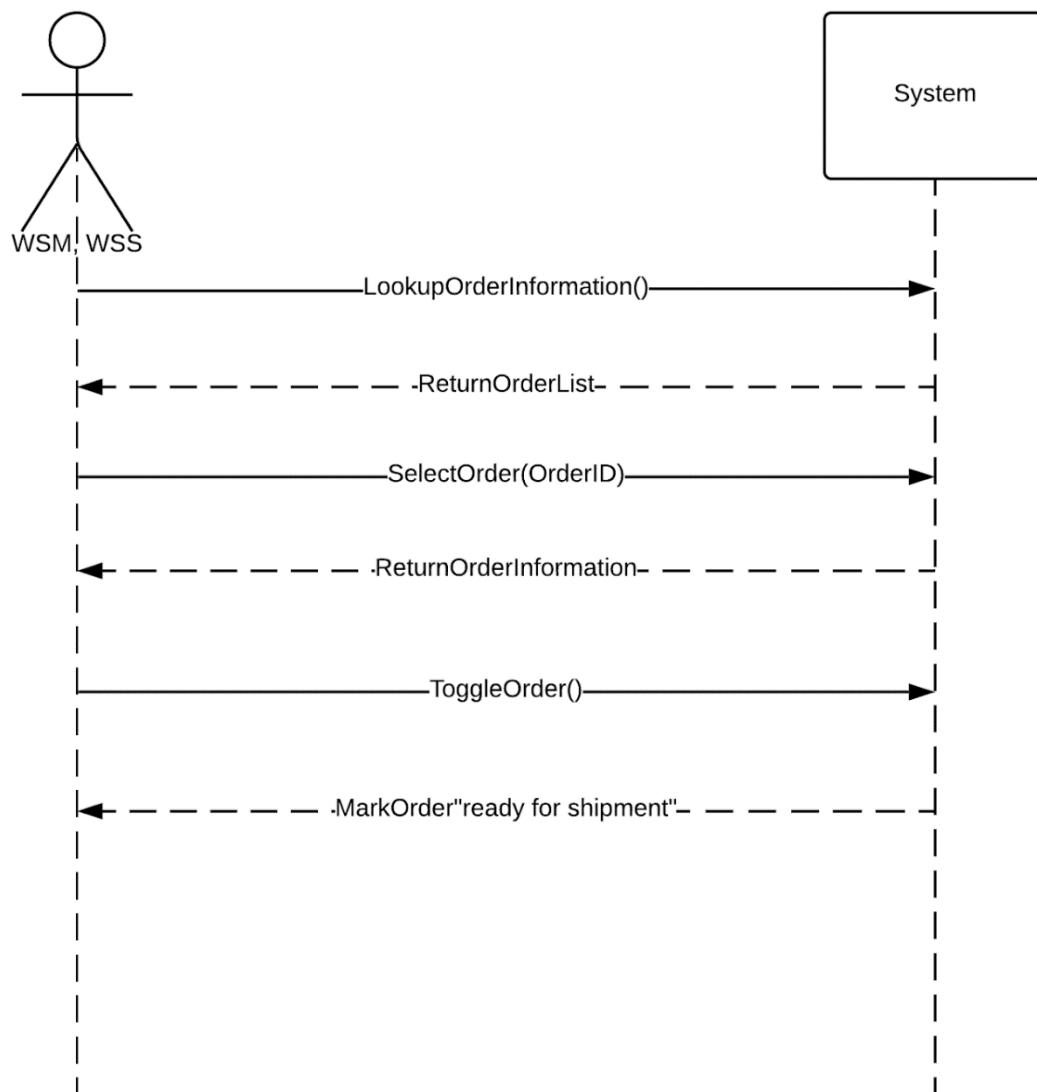
## Mark Order as Ready for Shipment

Righteous Row



## Mark Order as Ready for Shipment

Righteous Row



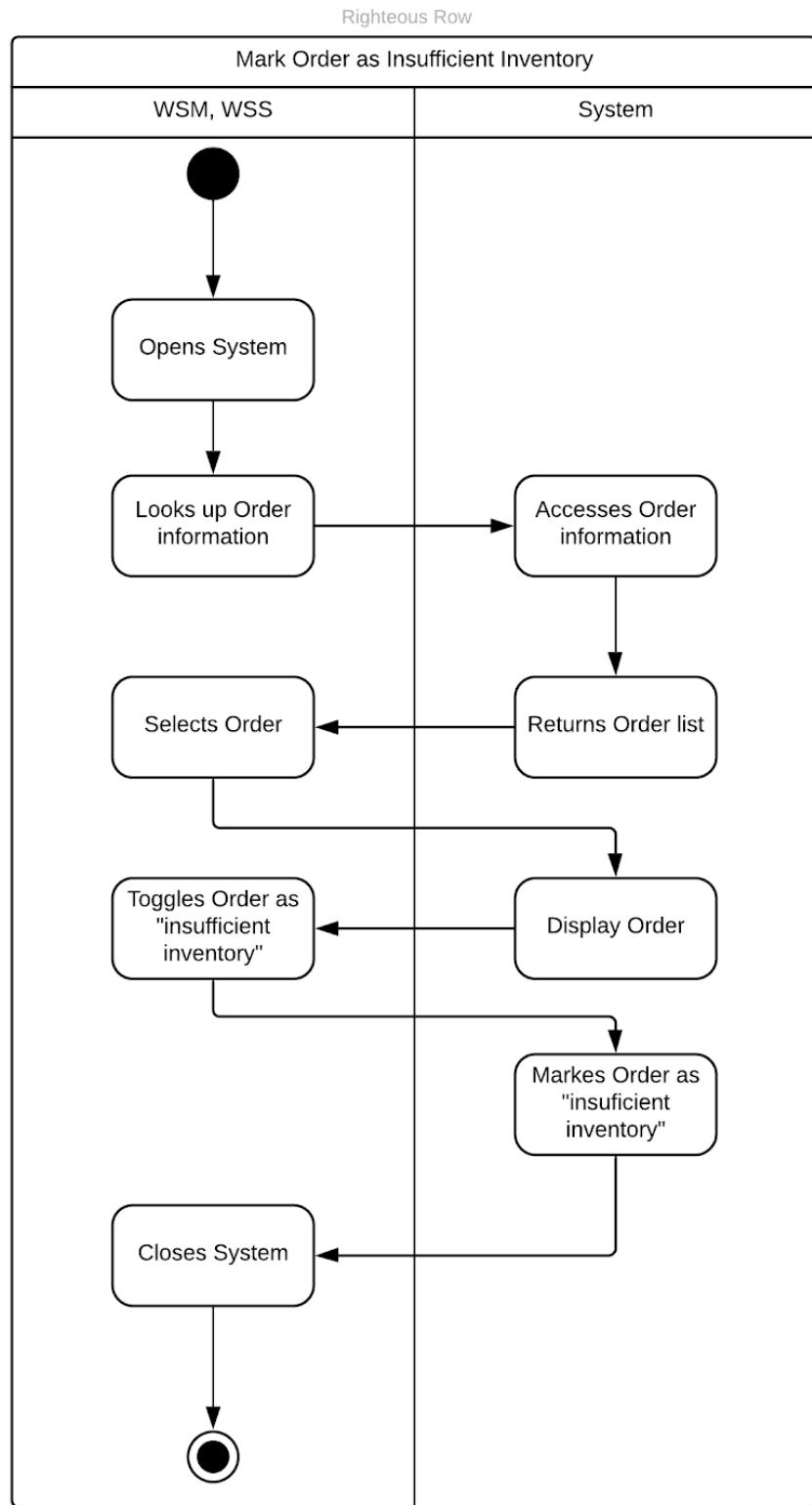
## Use Case #39: Mark Order as Insufficient Inventory

<b>Use Case Name</b>	Mark Order as Insufficient Inventory	
<b>Scenario</b>	An order is marked as “insufficient inventory” in the system.	
<b>Triggering Event</b>	A client places an order that contains a product that the warehouse does not have enough of to fill the order	
<b>Brief Description</b>	An order is marked as “insufficient inventory” in the system if there is not enough of a product in the order to complete it	
<b>Actors</b>	WSM	
<b>Related Use Cases</b>	Initiate Order Batch, Look Up Order Information	
<b>Stakeholders</b>	WSS, WSM, Client	
<b>Preconditions</b>	An order for which there is insufficient inventory is placed and exists within the system	
<b>Postconditions</b>	The order is marked as “insufficient inventory”	
<b>Flow of Activities</b>	<b>User</b>	<b>System</b>
	1. User queries opens order  2. User selects order to be marked as “insufficient inventory”  3. User toggles “sufficient inventory” to “insufficient inventory”	1.1 System looks up order information 1.2 System displays list of orders  2.1 System displays order  3.1 System marks order as “insufficient inventory”
<b>Exception Conditions</b>	1.1 There are no active orders	

As a **WSM**, I want to **mark an order as insufficient inventory** so that **the client is alerted that DSI does not have enough of that product to ship to them**

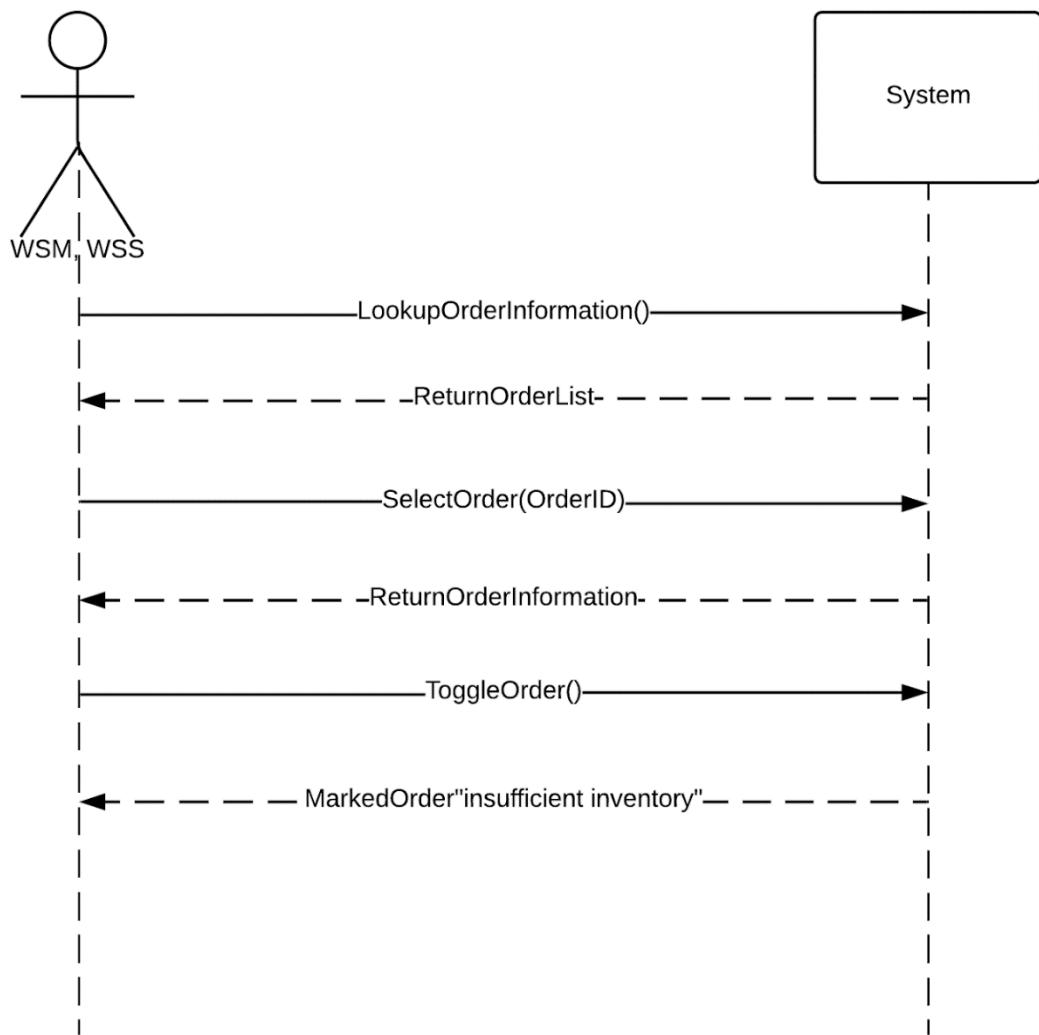
- Order is marked as insufficient inventory in the database
- Customer is alerted that the order could not be completed

## Mark Order as Insufficient Inventory



## Mark Order as Insufficient Inventory

Righteous Row



# Domain Classes

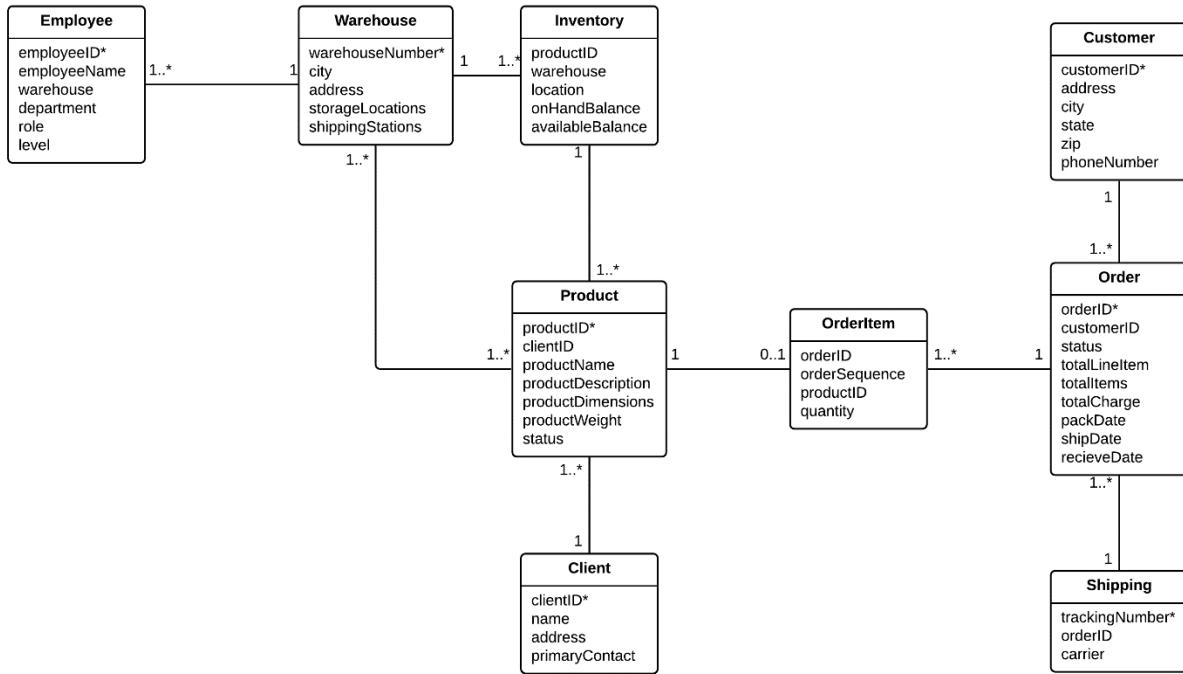
## *Tabular Domain Classes*

Object	Fields – Primary Key Denoted by Asterisk (*)
Employee	employeeID*, employeeName, warehouse, department, role, level
Warehouse	warehouseNumber*, city, address, storageLocations, shippingStations
Client	clientID*, name, address, primaryContact
Product	productID*, clientID, productName, productDescription, productDimensions, productWeight, status
Customer	customerID*, address, city, state, zip, phoneNumber
Order	orderID*, customerID, status, totalLineItem, totalItems, totalCharge, packDate, shipDate, recieveDate
OrderItem	orderSequence, orderID, productID, quantity
Shipping	trackingNumber*, orderID, carrier
Inventory	productID, warehouse, location, onHandBalance, availableBalance

# Graphical Domain Class Model

Graphical Domain Class Model

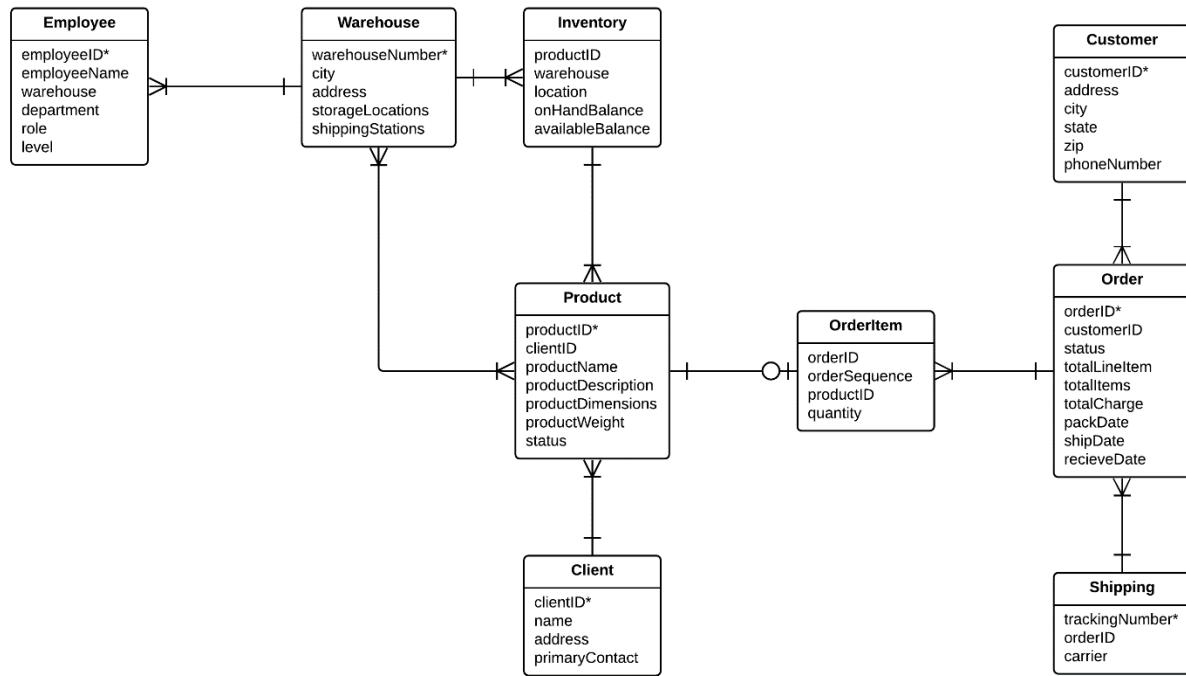
Righteous Row



# Entity Relationship Diagram

Entity Relationship Diagram

Righteous Row



# Concluding Deliverables

## Team Retrospective

### *William Dzubak*

As the project manager of the Righteous Row, I have worked more on a school project than I ever have before. The sheer amount of information to manage was a lot, but I think I did a fairly good job of doing so. One challenge we did experience was finding time to effectively meet and collaborate outside of class, as three of the five team members live off-campus. Luckily, we were able to find a time that worked for us, and had a lot of free time during class to conduct project work towards the end of the semester. Overall, this project was definitely a challenge, but it was very achievable. The main challenge came in setting aside time to thoughtfully and carefully plan everything out and organize what needed to be done. I believe that the Righteous Row did an superb job in completing this project.

### *Joseph Griffith*

As a team member of the Righteous Row, for me this project was a bit difficult. The hardest part about the project was the amount of information we received at the beginning. All of the information gathered in the beginning of the project made it very hard to determine what was a use case and what wasn't. The use cases seemed like a daunting task to complete but after I did one of them the rest were easier to understand. The biggest roadblock with the project was scheduling group meetings. Everyone had different schedules and the times where we did meet were about once a week. Despite that the project manager William was really good about scheduling meetings taking into account everyone's schedule, contacting us beforehand, and reserving rooms in the library for us to meet in. Even though a lot of the material in this project seemed insurmountable I'm glad I had the experience of working on this project and taking this class.

### *Avery Logue*

As a team member of the Righteous Row, I thought this project was a decent challenge. There were easy parts, such as all of the graphs and charts, and there were more challenging parts, such as the cost/benefit analysis. Overall, I thought it was doable and the time given to complete the project was reasonable. Most of my troubles came in the form of procrastination. Should I have managed my time better? Yes. One can only speculate the joy of completing everything in a timely manner. I thought that all my teammates pulled their own weight and tried their best. I especially commend our project manager, William, for going above and beyond to make sure this project was perfect. I enjoyed this experience and the skills I learned. Hopefully I can put them to use in the future.

## *Caleb Mathew*

As a team member of the Righteous Row, I found this project to be challenging but highly rewarding. Before beginning the project, the workload seemed very intimidating, but once I started working on it, the requirements became much more attainable. My group did a great job with communicating and coordinating team meet ups with such busy schedules. Our workload was well organized, and each group member did their part. I thought the most challenging portion was the use cases and their respective diagrams. I did not have trouble with the requirements, but the time spent on each use case and diagram to ensure accuracy and proper formatting was the difficult part. However, working on these use cases closely helped me develop a much stronger understanding of certain topics that I failed to comprehend prior to this project (activity diagrams, system sequence diagrams, flow of activities, stakeholders, etc.). Overall, I enjoyed this project and believe our group did an excellent job with it.

## *Anh Tran*

As a team member of Righteous Row, I think this project is a big challenge. It took a lot of time to finish. Every time I look back to the works I have done, there is always something that I need to edit. One of the most challenges is team working time. It was not that easy to have teamwork time while all my team members have different classes schedule and some of us not living on campus. My two team members have to spent time working on the project during the weekend to catch up with the due date. Using many different applications to complete the project, we ran into format issue which was a big time consumption to edit. New applications and knowledge such as lucid chart, sequence diagram, activity diagram are another time consumption while our team working on this project. We did not have that much time to work together on the project, every time team member have concern about something the question go back and forth several times until the question is answered. Study room need to schedule ahead for team meeting which may cause time wasting as well.

# **Additional Deliverables (Appendices)**

## **Information Gathering Results**

### *Interview Transcript – Chris Kreider – October 9<sup>th</sup>, 2019*

#### **Overview**

Below is the notes taken from an interview with Chris Kreider, CK Investor Services Representative, on October 9<sup>th</sup>, 2019. This initial interview allowed us to understand basic knowledge of the system, and helped us formulate the functional requirements for it.

#### **Interview Notes**

##### **Open Questions**

- What works well in each existing information system?
  - Warehouse does not have much going well
  - Not much to it - they have products, but don't know where they are
  - Shipping system works well with post office - don't have to worry about postage labels
  - Order Management works with the API - MAINTAIN THIS
- What improvements do you want to see within each system?
  - Warehouse system needs to interact with order system (know inventory)
    - immediate communication
  - Shipping system needs to communicate and know about the inventory/potential problems (shortages)
  - Order management needs to pull reports on past orders
    - Find out who ordered what in order to market more effectively
- How would you order those improvements from most important to least important?
  - Warehouse
  - Shipping
  - Customer Management
- Who will be using each system and for how long do they use it per day?
  - Two broad categories
    - Warehouse
      - Warehouse floor
      - Shipping staff
      - Return staff
    - Management
      - Executive business ops user
      - Make sure people show up each day
- What system is the most used and least?
  - Most: Shipping
  - In-between: Warehouse
  - Least: Order Management
- What are some unique functions of each system?
  - Warehouse has a returns system, but it has problems with theft
    - Returns “defective” products

- Needs a better process for doing so (maybe prove its defective)
- Bin packing system
  - System knows which box to use based on the size and amount of products
- Order management system uses the API
  - Status of orders, etc.

### Closed Questions

- What is your budget and time frame for the project?
  - Best case: done by Black Friday of 2020 with functionality prior to being on sale so that we can preemptively advertise
  - Be profitable within 3 years
- How much experience is required to operate each system?
  - DSI will handle all in-house training as long as systems function properly

# *Developed Survey – DSI Employees – October 10<sup>th</sup>, 2019*

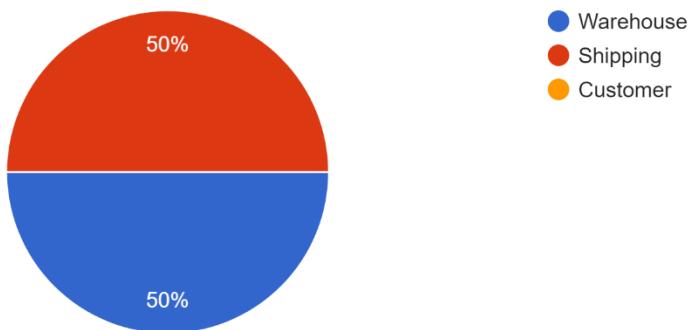
## **Overview**

This survey was issued to all DSI employees on October 10<sup>th</sup>, 2019. This survey proved to be rather unsuccessful, as a typo and the requirement of a short answer question frustrated some employees. Nonetheless, we gathered some valuable information in questions 3 and 4 about user thoughts on how friendly the system is, and how much they use the system to communicate with other employees throughout the course of their day.

## **Survey Results**

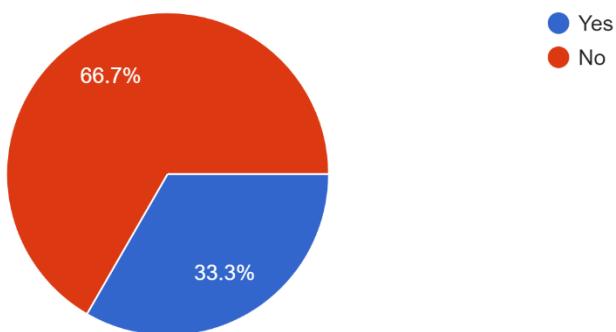
### In what department do you work?

6 responses



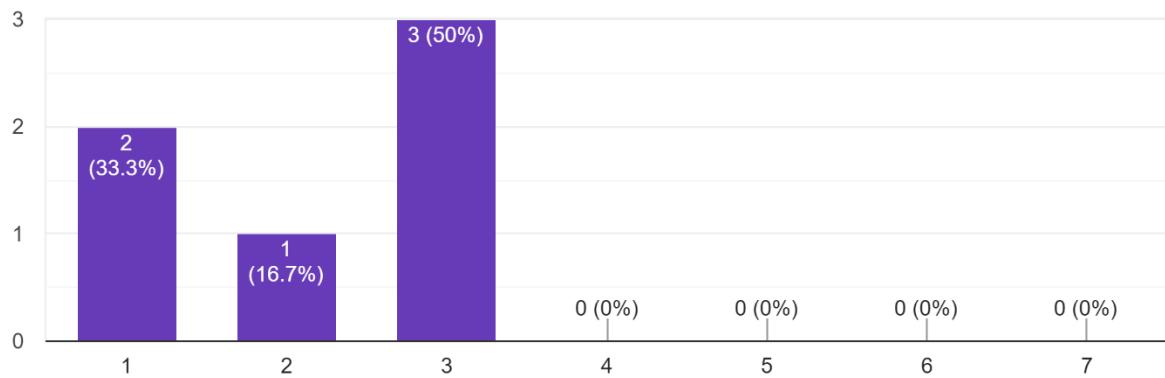
### Are you a manager in your department?

6 responses



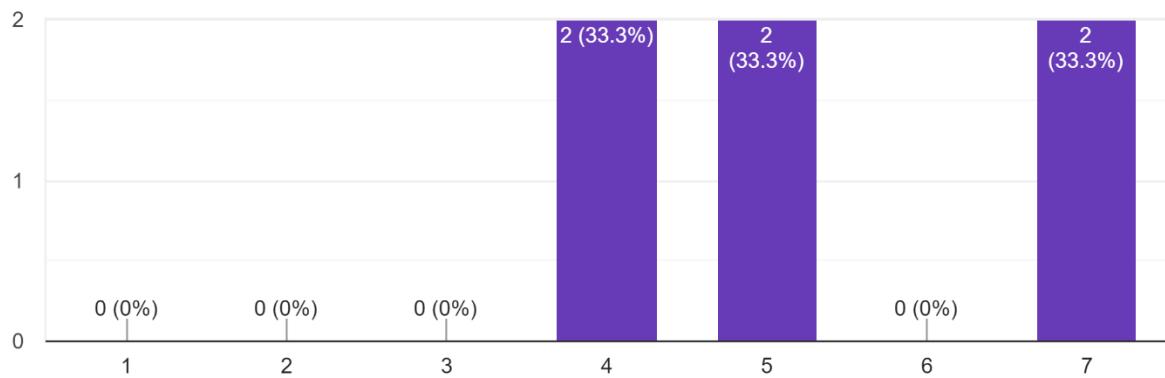
## How user-friendly is the system that you work with?

6 responses



## How often do you communicate with other employees to complete business tasks?

6 responses



**If you could improve one thing about the system you work with, what would it be?**

- Why is this required, this is dumb!
- Better user interface
- Snack machines baby!
- Easier to do manager related tasks
- Don't wanna answer this, don't have time
- Quicker to load

## *External Resource – Systems Analysis and Design Textbook*

### **Overview**

The class textbook was consulted frequently throughout the course of the project to ensure that documents, diagrams, and other business processes were being done correctly.

### **Citation**

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016) *Systems Analysis and Design in a Changing World*. Boston, Massachusetts: Cengage Learning.

## *Internal Documentation References – Lecture Slides*

### **Overview**

Throughout the course of the project, the lecture slides posted on CNU Scholar were also consulted frequently to once again ensure that documents, diagrams, and other business processes were being done correctly.

### **Link**

[https://scholar.cnu.edu/webapps/blackboard/content/listContent.jsp?course\\_id=\\_788427\\_1&content\\_id=\\_964765\\_1&mode=reset](https://scholar.cnu.edu/webapps/blackboard/content/listContent.jsp?course_id=_788427_1&content_id=_964765_1&mode=reset)

# **Relevant Communications Received**

## *Email From Chris Kreider – Project Extension*

### **Overview**

The following is an email received from Chris Kreider detailing the extension of the project's due date by two days

### **Email**

From: Christopher Kreider [chris.kreider@cnu.edu](mailto:chris.kreider@cnu.edu)

12/2/2019 9:23 AM

Hi Teams,

I have decided to give an extension on the projects due date due to its closeness to the thanksgiving break.

All project deliverables (notebook in class, pdf to scholar, etc) are now due, Thursday, December 5th!

We will use class this Tuesday to answer any last minute questions and have a team work day.

--Professor Kreider

# **Relevant Documentation Received**

## *Email From Chris Kreider*

### **Overview**

This email is from Chris Kreider in response to a question asked by Project Manager William Dzubak. His question is as follows:

“I’m finishing up organizing our use cases by subsystem in a tabular form, so I’m putting the use case on the left of the table, along with the actors on the right side. In the interview transcript that was provided to us, the Warehouse Shipping Manager (WSM) is said to be the senior manager, being familiar with all of the tasks of the warehouse (floor, shipping, and returns). That being said, would the WSM be listed as an actor for each of the use cases involving the floor, shipping, and returns staff/managers? My gut tells me yes, but I just want to double check and be sure. Thank you!”

## Email

From: Christopher Kreider [cchriskreider@cnu.edu](mailto:cchriskreider@cnu.edu)

11/20/2019 8:35 AM

I would go ahead and do it that way.

Thanks,

**Christopher Kreider**  
**IS/Cybersecurity Instructor**  
**Christopher Newport University**  
**Luter Hall 331**  
**1 Avenue of the Arts**  
**Newport News, VA 23606**  
**Voice: 757-594-8857**

## External Information

### *Other Sources*

#### **Overview**

Other sources were used in the making of this project to find information behind the rough estimates, such as the cost of laptops and the cost of rent for office space. Google was used to find these results.

## Anything Else

