Group5 legacywms
Report3

Glossary Of Terms:

- **WMS:** Warehouse Management System
- **BOL:** Bill of lading, a document that shows the contents of the delivery.
- **Inbound**: The department that will take care of unloading, receiving and putting away the product.
- **Outbound**: The department that will take care of picking, packing and shipping the product.
- **Picking:** Term used during outbound process when pulling inventory that is ready to be shipped out.
- **Replenish:** The process of re-stocking inventory from the reserve location to the active location.
- **Active location:** The location where the outbound department pulls the inventory.
- **Reserve location:** Inventory that does not fit in the active location is placed here.
- **User:** The associate working on the floor.
- **Query:** Extracts requested data and formats it in a certain way.
- **Pick ticket:** A pick ticket is a list used for gathering items to be shipped from a store or warehouse.

Github URL:
https://github.com/wmsrNull/wmsr.git

Customer Problem Statement:

Customer Problem Statement #1

The inbound process is one of the most important steps of managing a warehouse efficiently. If a mistake is made during this step, it will eventually lead to serious issues. It is important for the warehouse management system (WMS) and the associate on-the-floor, to both work quickly and accurately. Typically, the inbound-receiving process starts with a semi-truck or bulk-van backing up into the dock. Since the product tends to be shipped by pallet, a forklift is usually necessary to unload the shipment on to the receiving area. Once the product is verified through a bill of lading (BOL) receipt, the plastic wrap comes off the pallets, and the product is ready to be received.

How does the associate know where to store the product once it is checked-in/scanned? Will the associate have to go aisle-by-aisle and try to find the appropriate rack where the product group will be? The associate would waste an extensive amount of time by walking every aisle with their pallet jack and equipment. Obviously, this process will be incredibly inefficient. How important is time management in a warehouse setting? We believe that our WMS will in fact help the associate speed up the product-put-away process. This will save your company countless hours of labor for unnecessary tasks daily. Below is a simplified list of the receiving process using the WMS.

- The user will enter or scan the product ID from the case in the system as well as enter other necessary attributes, such as quantity.
- Once the user inputs the data, the WMS will look at the product's dimensional weight (dims), then it will look through the list of available storage locations.
- The WMS then will query a list of available locations where the user can store the product.
- Once the product is put away, the WMS will save where the product was placed for future allocation.

With the aid of our system, your company will be saved from massive expenses. It will also ensure accuracy and effectiveness as next time a product is needed for outbound, the WMS will know the location of the product and managing it not be difficult.

Customer Problem Statement #2

When dealing with pharmaceutical products, it is important to stay compliant at all times. It is important because, at one point or another, these healthcare products will come in contact with a real patient. That being said, in the healthcare world, there are numerous scientific studies happening every day on regular drugs and treatments. After a study, it is common for a scientist to conclude that a specific drug may need to be recalled due to it being harmful or inefficient. What will happen if a drug gets recalled and it is already in a storage location within your warehouse? How will your employees know where the recalled products are stored? What will the consequences be if the system allows you to ship a recalled product? What if a product reaches its expiration date? Your employees will waste unnecessary time walking the floor looking for a product if the system does not specify where the product location is. This will result in wasted money and warehouse effectiveness on your end. Another issue which may arise is if a recalled or expired product gets accidentally delivered to a patient, then you may come across a lawsuit. Surely there must be a way to protect your company with the help of our WMS.

- Using our WMS you will have the ability to update product entries and add tags such as "Recalled" or "Expired".
- A search will easily show your employees a list of products with tags to which the WMS will use to specify the location of the product.

- A lock in the system will be added to these products automatically, that way it is impossible for the recalled or expired inventory to be shipped out accidently.

Our WMS will help secure your company's assets by assuring that harmful products are not shipped out by adding a lock. Also, our system will help save time and labor by decreasing the amount of time it takes to collect this type of inventory.

Customer Problem Statement #3

In a typical warehouse setting, there may be an active location and a storage location. Generally, all of the outbound products are picked from the active location. The products are picked from this location because it is closer to the outbound area. The storage location is only used to replenish the active location when its inventory gets too low. Let's say an order comes in and takes up most of a specific product "A" within the active location. Now, there is very little of product "A" in the active location. How will your associates know to re-stock that area with product "A" from the storage location? What will happen if another order of "A" comes in and it is not re-stocked yet? There can be a lot of time wasted by excessively walking around the warehouse if your employees have to go from location to location to get what they need. We believe our WMS will help your company solve those issues.

- Once a type of product gets low in the active location, our WMS will notify employees from the reserve side to replenish the product.
- Our system will then specify which product is low and where it is located.

This process will ultimately save your company costs. It will also decrease the amount of time a product as to which gets picked for outbound as the inventory will always be in its intended location.

System Requirements:

- Functional Requirements:

REQ-x	Priority Weight for the Requirement	A Brief Requirement Description
Use Case 1	3	Employee shouldn't be able to edit data, only see it.
Use Case 2	1	System needs to make sure the user has signed in with authentication and allow the user to sign out.
Use Case 3	2	Administrator is the only one who should be able to edit/remove/add data.
Use Case 4	4	User should be able to enter the product ID when receiving.
Use Case 5	7	WMS will generate a suggested location list for every product received.
Use Case 6	6	The User will see an error message if attempts to ship out a product with a lock.

Use Case 7	10	System will have both an Inbound module and an Outbound module
Use Case 8	9	The WMS is programmed to update and restock automatically once the supplies are running low by printing out a list and showing it to the employee.
Use Case 9	5	WMS will know the current status of each product ID received (for example: “In Inventory” or “Shipped”)
Use Case 10	8	WMS can generate a report displaying all orders successfully shipped.

- Non-Functional Requirements:

The non functional requirement	Metric used to measure non functional requirement
The interface should be easy for the users to operate.	Take 1 day to learn
The database search queries should not take more than 5 seconds.	seconds
The system and all the data should all be backed up.	monthly
The system should be able to handle multiple users at once.	30 Users
The system should be operable on multiple different types of devices.	Handheld, Desktop, Mobile
A system update will be outside of company operating hours and should not take more than 3 hours to complete.	hours

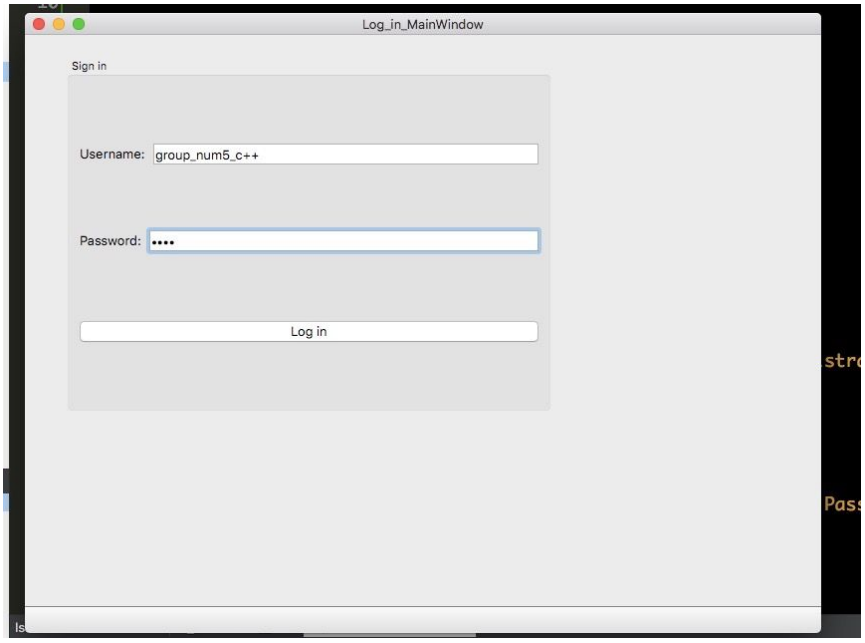
- Constraints:

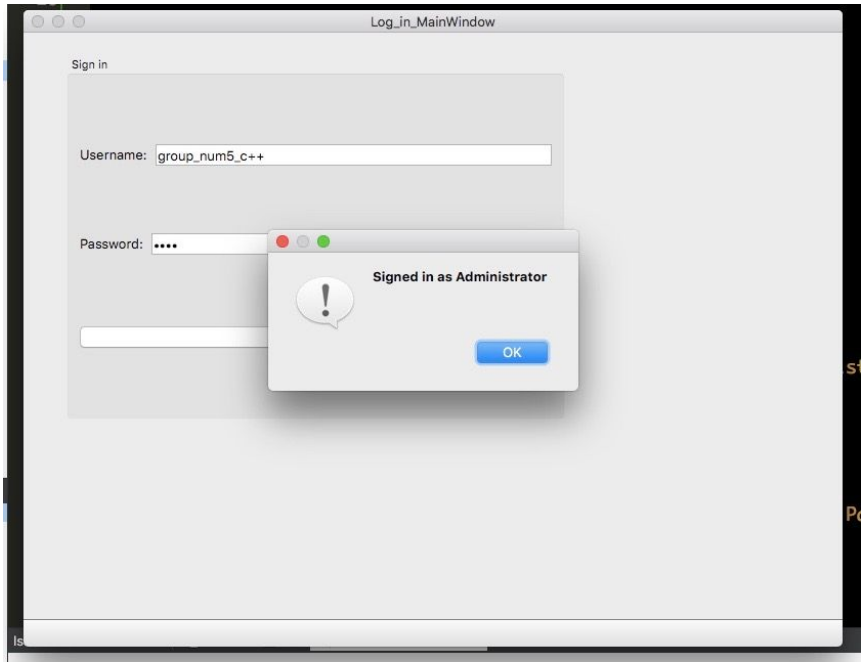
Since we only have a limited staff of 4 people working for Legendary Services we will set our WMS tentative go-live to December 2019.

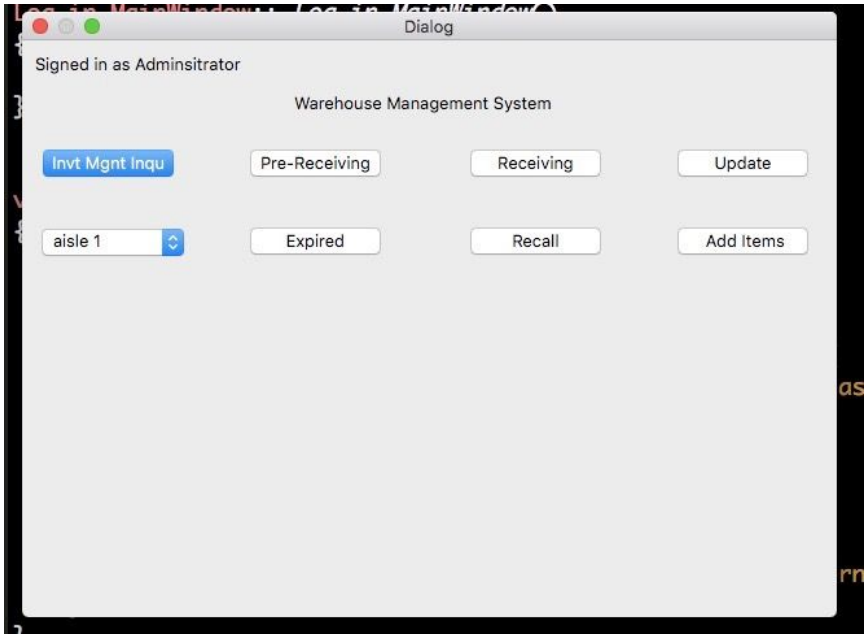
- User-interface requirements:
    - The log-in page is the first page that is shown on the user interface.
    - From the administrator’s perspective, once he/she logs in, they are the only one who can make/override any changes as needed.
    - Employee should be able to login and see what’s available in the system/aisle.

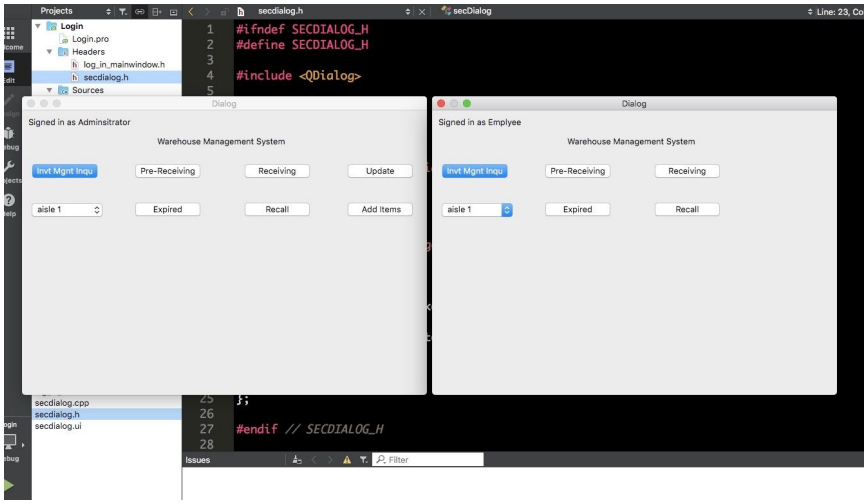
- The last picture shows the differences between the administrator Vs employee log-in page.

Images of User Interface:









Functional Requirements Specification:
/ / / / / / / / / / / /

Use Case 1:

- **Actor:** Employee
- **Goal:** Employee is not allowed to edit data, only see it.
- **Use Case:** viewData
- **Entry Condition:** The employee clicks the “view data” button to request an update.
- **Flow of Events:**
    - The system shows the employee the data.
- **Exit Condition:** The employee isn’t given a button to “edit data”, and is given the button to “view data”.

/ / / / / / / / / / / /

Use Case 2:

- **Actor:** Employee, Admin

- **Goal:** The goal is to allow the user to sign in, using their own credentials, and allow the ability to sign out.
- **Use Case:** loginLogout
- **Entry Condition:** The user hits the “login” button, which opens a large window called Login.
- **Flow of Events:**
    - A small box with the labels “Username” and “Password” will prompt.
    - The user will enter their username and password.
    - The system will check if the user’s username and password are correct then authorize the login.
    - The user will have successfully signed it and home page of WMS will appear.
    - On the right hand corner, there will be a button called “Logout”.
    - Once the user hits the “Logout” button a “Successfully logged out.” prompt will populate.
- **Exit Condition:** The user will have successfully logged in and then logged out.

/ / / / / / / / / / / /

Use Case 3:

- **Actor:** Admin
- **Goal:** Administrator is the only one who should be able to edit/remove/add data.
- **Use Case:** editData
- **Entry Condition:** The admin clicks “edit” button to request an update.
- **Flow of Events:**
    - The system checks if the user has permission to edit data.
    - The system sees that the user is an admin, so the system allows access to edit data.
- **Exit Condition:** permission is granted to edit/update data.

/ / / / / / / / / / / /

Use Case 4:

- **Actor:** Employee
- **Goal:** The User should be able to enter the product ID when receiving.
- **Use Case:** enterProductID
- **Entry Condition:** The user will click “Enter product ID”.
- **Flow of Events:**
    - The system will prompt the user to enter the product ID.
    - The user will enter the product ID.
    - The user will click “submit”.
- **Exit Condition:** The product ID is then stored.

/ / / / / / / / / / / /

Use Case 5:

- **Actor:** Employee
- **Goal:** The WMS will generate a suggested location for every product received.
- **Use Case:** suggestedLocation
- **Entry Condition:** a product is marked as received in the database.
- **Flow of Events:**
    - The system looks where similar products, to the received product, are stored.
    - The system checks if there is still availability for storage in that location.
    - The system checks how much of that product fits in that location.
    - The system tells the associate to store the product in that location.
    - If there is not enough room there, the system checks other locations on the floor for storage space.
    - If there isn't any more space on the floor, the system tells the associate to store the product in a reserve location.

- **Exit Condition:** The system records the products location and stores that information into the database.

/ / / / / / / / / / / /

Use Case 6:

- **Actor:** Employee
- **Goal:** The WMS will generate an error message if the user attempts to ship out a product with a lock (e.g. an expired product).
- **Use Case:** lockPrompt
- **Entry Condition:** The user generates a pick ticket for an order.
- **Flow of Events:**
    - The system will display a list of the ordered goods to collect.
    - The employee will go to the product locations, then scan each item that is picked.
    - The system verifies each item that is scanned to check if it is on the pick ticket and it is free of locks.
    - The user attempts to scan an expired product (with a lock).
    - The system will send back an error message to the user indicating that the product is expired and cannot be processed.
- **Exit Condition**: The employee is denied access to send out locked product.

/ / / / / / / / / / / /

Use Case 7:

- **Actor:** Employee
- **Goal:** The goal is for the employee to see if they are working within the inbound or outbound environment.
- **Use Case:** inboundOutboundMod
- **Entry Condition:** The user logs in to the WMS and is currently at the home page.
- **Flow of Events:**
    - The user will click one of the buttons labeled “Inbound” or “Outbound”.
    - If the user hits “Inbound” a new menu will appear with additional options.
    - If the user hits “Outbound” a new, different menu will appear with additional options.
    - The top of the page will always display the title of the current module the user is in.

Exit Condition: The title of the page displays the users current module.

/ / / / / / / / / / / /

Use Case 8:

- **Actor:** Employee
- **Goal:** The WMS provide a list of locations that need to be restocked.
- **Use Case:** restockList
- **Entry Condition:** The user will hit the “replenish” button on the GUI.
- **Flow of Events:**
    - The GUI will activate a trigger to scan the storage location quantities on the products.
    - The system will check to see if the storage location is below its minimum quantity threshold.
    - If the actual quantity on the storage location is lower than the accepted minimum, then the system will pull a query with all locations that need a replenishment.
    - The user will see the query that indicates which locations need to be restocked.
    - The user hits print list on the GUI.
- **Exit Condition:** The user receives the list of locations that need replenishment.
-

/ / / / / / / / / / / /

Use Case 9:

- **Actor:** Employee
- **Goal:** The User has the ability to search the current status of each product ID received.
- **Use Case:** productStatus
- **Entry Condition:** The user hits the “product ID” master button on the GUI.
- **Flow of Events:**
    - The system will automatically generate all of the product IDs on the database.
    - There will be a field on the GUI that allows the user to search by product ID.
    - The user will hit the “search” button and input the product ID (e.g. 123456).
    - The WMS will query the results and only give back the attributes for 123456.
    - The system will display the status for “123456”, current status as “In inventory”.
- **Exit Condition:** The user has received the status of a custom product ID requested.

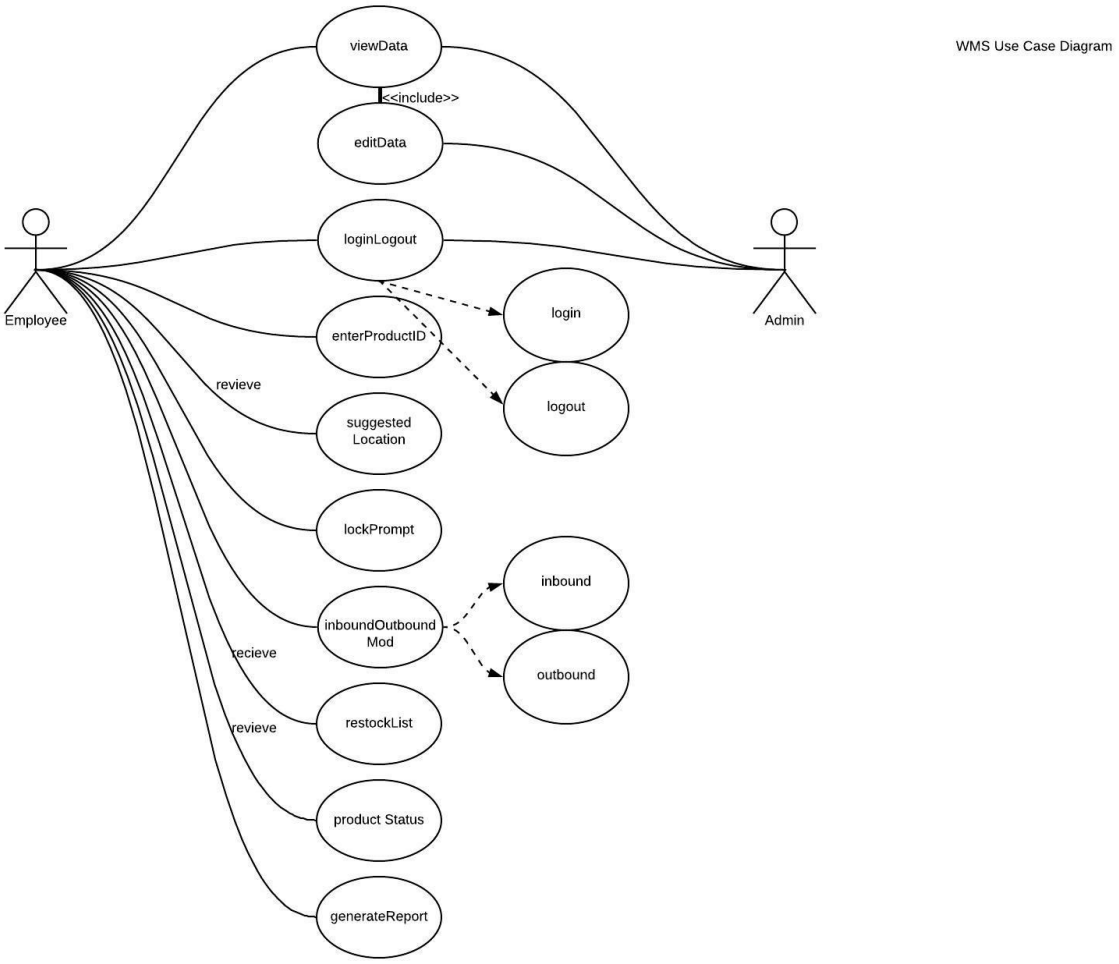
/ / / / / / / / / / / /

Use Case 10:

- **Actor:** Employee
- **Goal:** WMS can generate a report displaying all orders successfully shipped.
- **Use Case:** generateReport
- **Entry Condition:** The user will click on the “generate report” button.
- **Flow of Events:**
    - The system will ask the user to enter the date.
    - The user will enter the date for the report.
- **Exit Condition:** The report is received for the user to view.

/ / / / / / / / / / / /

Use Case Diagram:



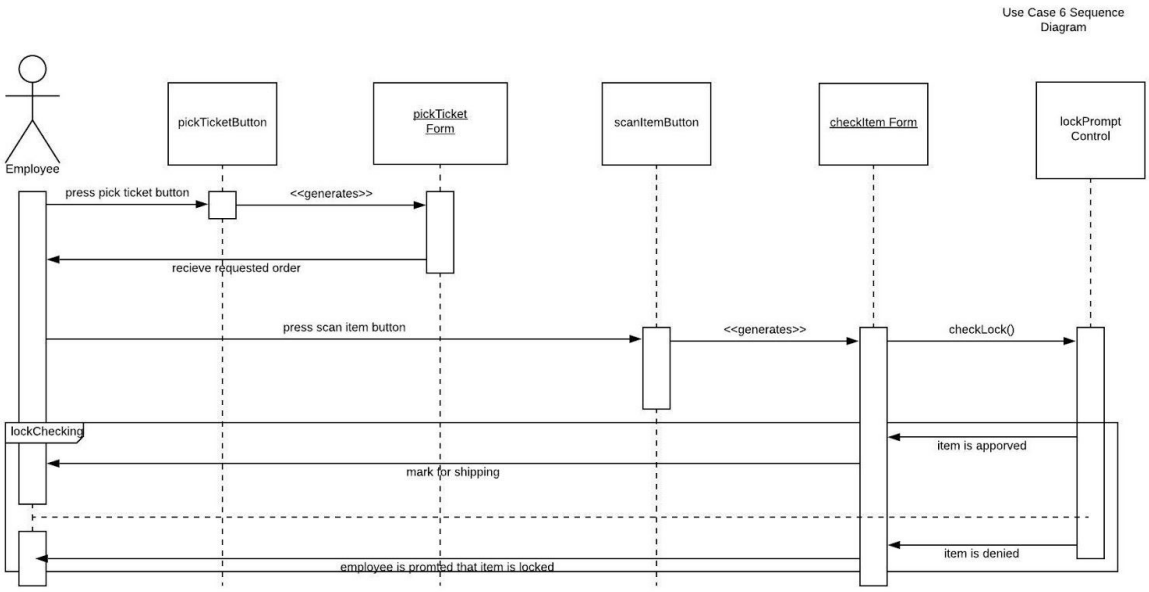
Traceability Matrix:

	Use Case 1	Use Case 2	Use Case 3	Use Case 4	Use Case 5	Use Case 6	Use Case 7	Use Case 8	Use Case 9	Use Case 10
Req Case 1	✓		✓							
Req Case 2		✓								
Req Case 3	✓		✓							
Req Case 4				✓						
Req Case 5					✓			✓		

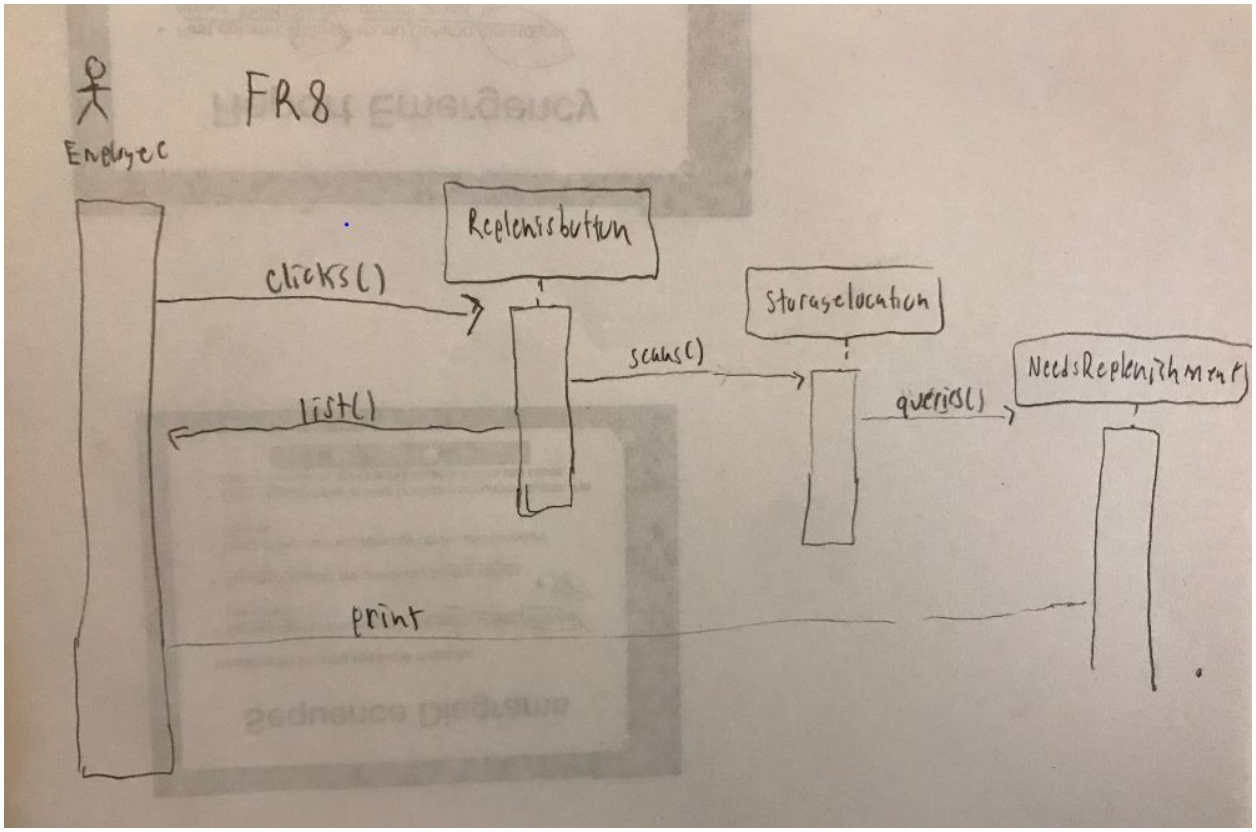
Req Case 6						✓				
Req Case 7							✓			
Req Case 8					✓			✓		
Req Case 9									✓	
Req Case 10										✓

Sequence Diagrams:

Sequence Diagram For Use Case 6:



Sequence Diagram for Use Case 8:



Legendary Services WMS
User Experience and Interface Specifications

User Interface Specifications:

1.0 Log In

1.1 The user will see a login button on the screen when system starts up

1.2 The user will click on the login button and a message box will appear with

1.3. User inputs their own username and password

1.4 If credentials are approved WMS homepage will appear

2.0 Log Out

2.1 Once the user is ready to log out he must click on “logout” button on the upper right hand corner

2.2 The WMS will exit from the page last viewed

2.3 A message prompt that says “Successfully Logged out” will appear

3.0 Receive an inbounded product

3.1 The user will inspect the new case that just inbounded and look for the printed productID.

3.2 The user will enter the productID into the field “Receive new item”

3.3 Once the productID is entered, there will be additional fields that the user can fill in. For example the expiration date and quantity.

3.4 Once the user is done entering the information for the item, he will hit the “Save” button on top.

3.5 The WMS will now save the entry into the database and can be later searched by productID

4.0 Add a lock

4.0 Add a lock

4.1 If the admin wants to add a lock to the recalled item then the user must have the productID already.

4.2 Under ProductID Master module, there will be a button called “Lock”

4.3 The GUI will populate a string box that tells the user “Enter productID”

4.4 The admin will enter the productID.

4.5 WMS will return a query of the productID and all of its attributes.

4.6 A radial button will appear next to the row with attributes. The user must hit the radial button and click on the “Lock Item” button.

4.7 WMS will prompt a box that states “Reason for lock?” and a character box that allows the admin to write recalled.

4.8 The UI will return a “lock successful”prompt if process is complete.

5.0 List all products in warehouse

5.0 List all products in warehouse

5.1 The user will have to go to the item master module.

5.2 Once in the item master module, there will be a button called “list”

5.3 The WMS will return a query of all the items that are registered in the database.

6.0 Generate report of shipped orders

6.0 Generate Report of Shipped Orders

6.1 The user will have to go to the outbound module first.

6.2 Under outbound there will be a button called “Reports”.

6.3 That button will be a dropdown button with a list of all reports that can be generated.

6.4 The user will have to hit the “Shipped orders”options

6.5 The user will then have to specify a date. There will be a “shipped date from” and “shipped date to”

6.6 If the user wants to only see the orders shipped today (e.g 9/26/19) then he should enter the same date for both entries.

6.7 The WMS will populate a query that shows all successfully shipped orders/items that are no longer in the warehouse.

7.0 List the status of inventory

7.1 The user will go to the item master module.

7.2 Under item master, there will be a button called “Filter by Status”

7.3 A drop down menu will generate, an example of status will be “In inventory” or “Not putaway” “Awaiting Pickup”.

7.4 The user will click on one of the status options and WMS will generate a query of only those items with the attached status.

8.0 View list with replenishments needed

8.1 If a user wants to replenish some locations from storage then it must go to the “Replenish” button under the inventory module.

8.2 A prompt will show up stating “Print Replenishment Slip”

8.3 The user will hit the Print button and WMS will generate a query with all locations that are below the minimum quantity.

8.4 The list is now printed

9.0 Print pick ticket

9.1 The user will go to the outbound module.

9.2 The user must hit the “Pick Orders” button

9.3 A query will be made with orders that are ready to be shipped out.

9.4 The user will then hit the “View orders” button

9.5 Once the orders are on the screen, the user must check how many lines/items the user wants to pick

9.6 The user must then hit the “Print pick ticket” button

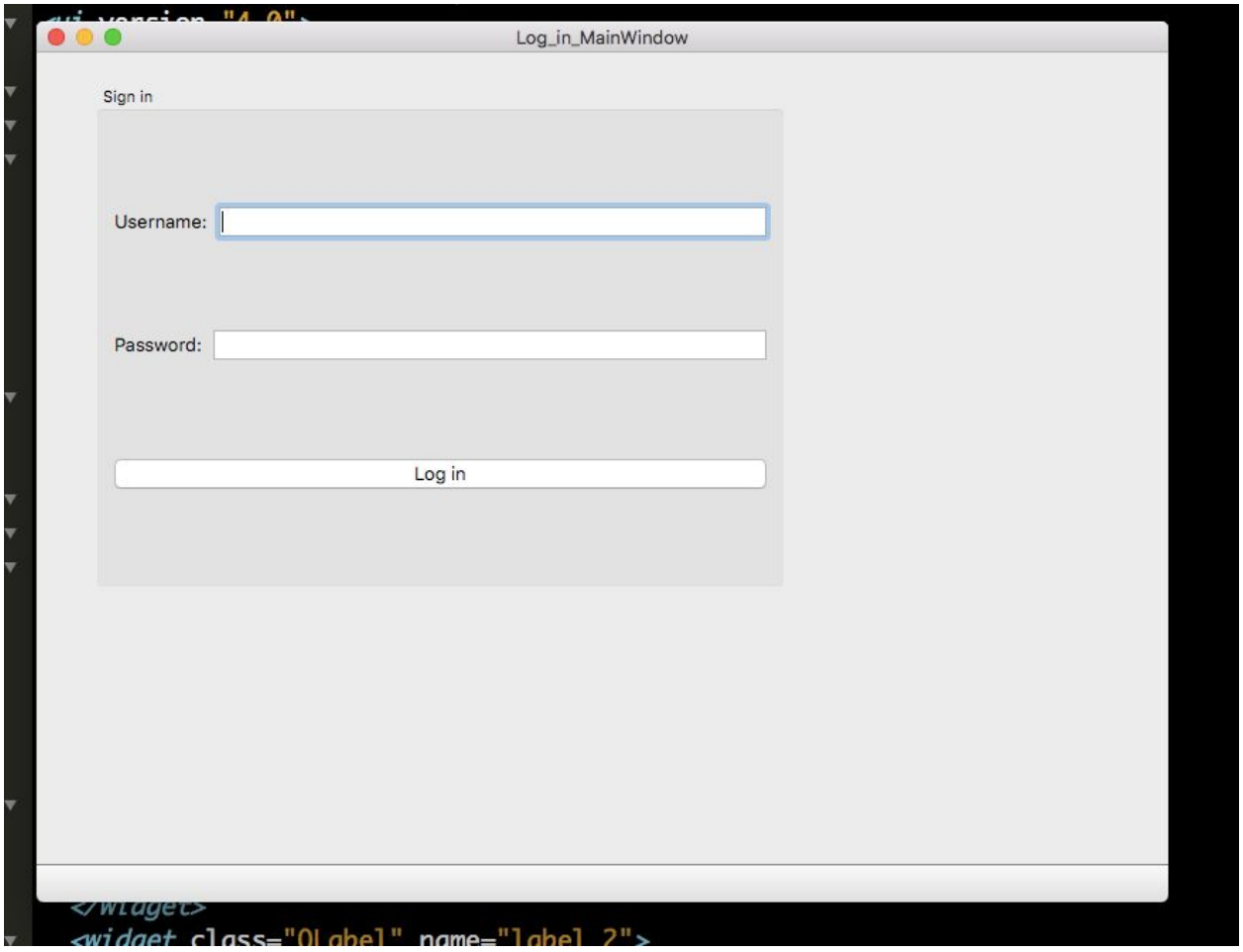
User Interface

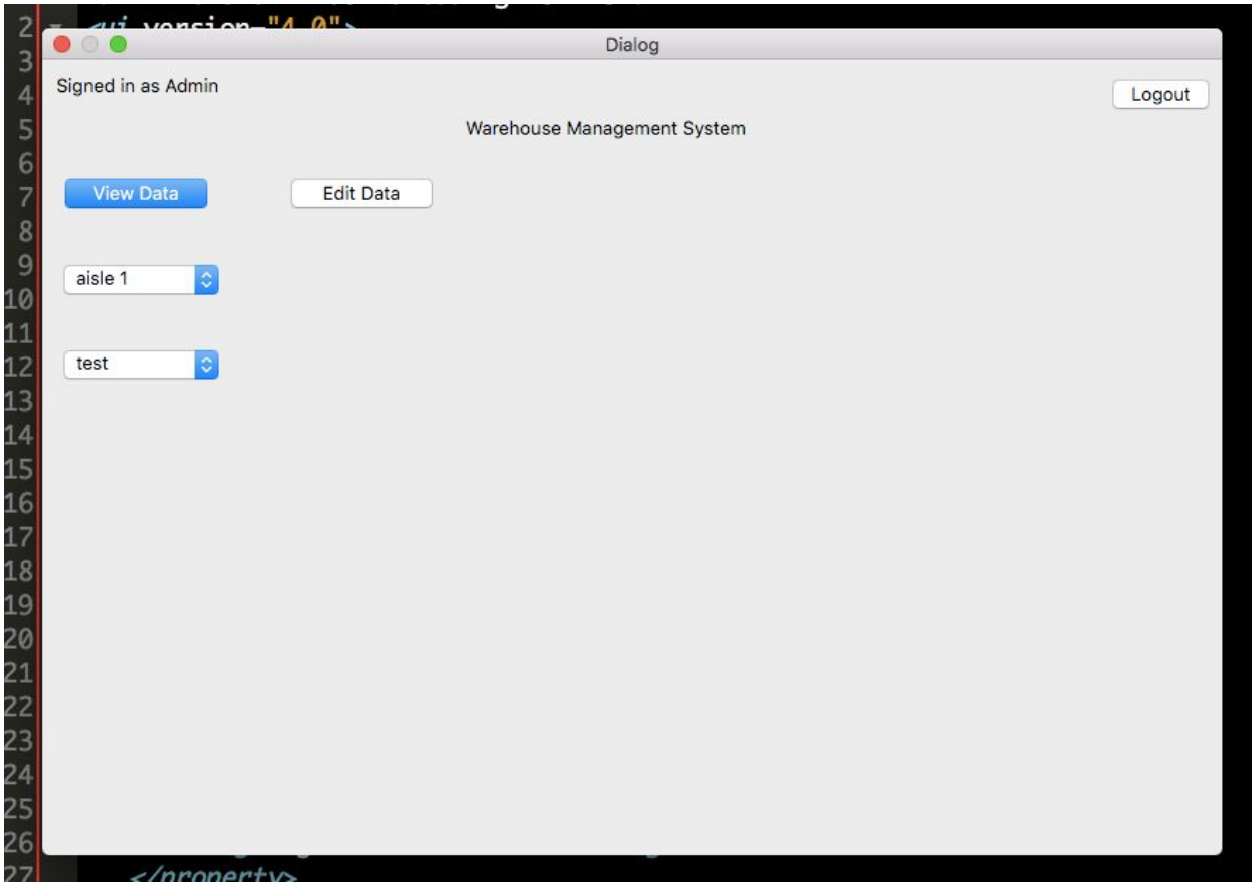
- Employee and Admin are using the login button to sign in, once they have logged in, they will see things differently. Employee can view data, enter product Id, look at suggested location, show the restock list and generate the reports.

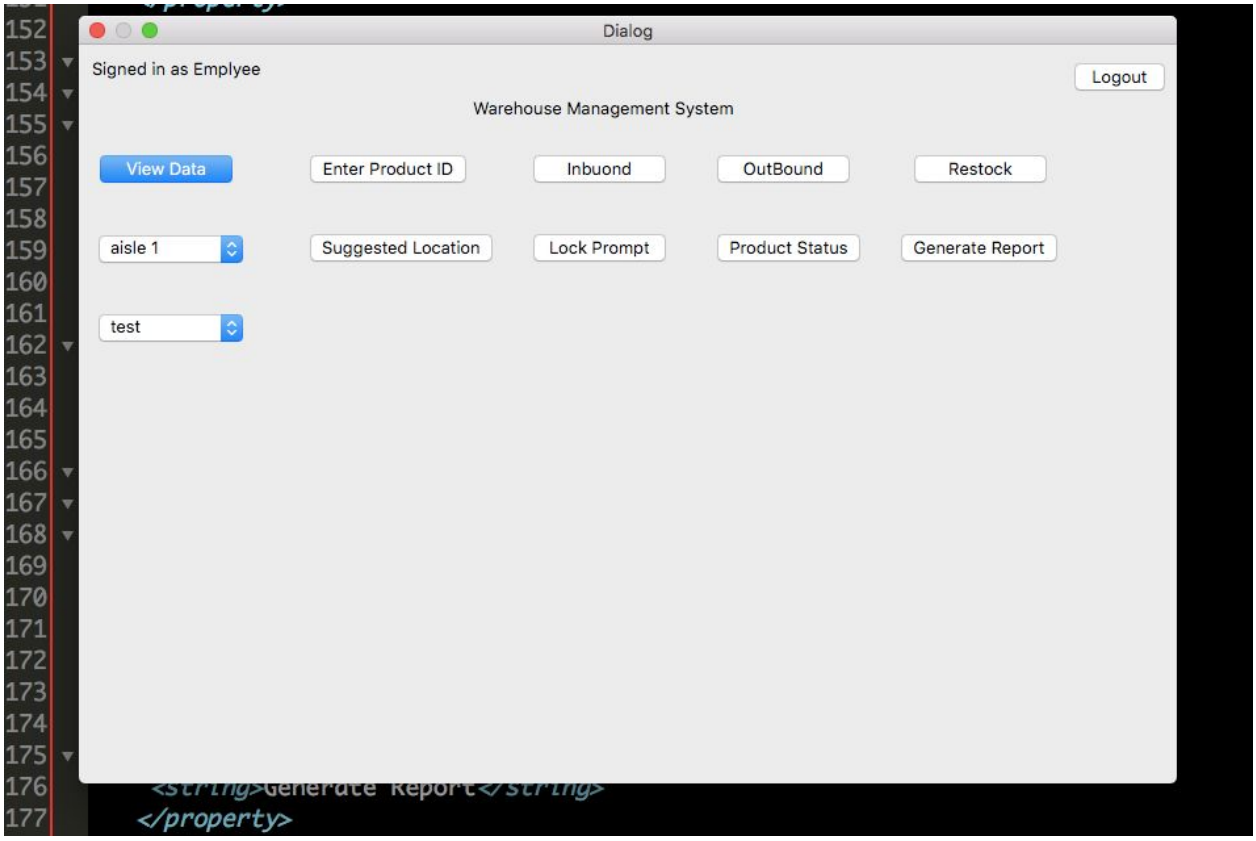
- On the other hand administrator can view as well as edit any data in the WMS as needed.
- Both the employee and the administrator are allowed save files after they are done with their tasks.

- Finally they can either log themselves out after everything is finished, or in the event that they forget to do so, the system will log them out after a certain amount of time.

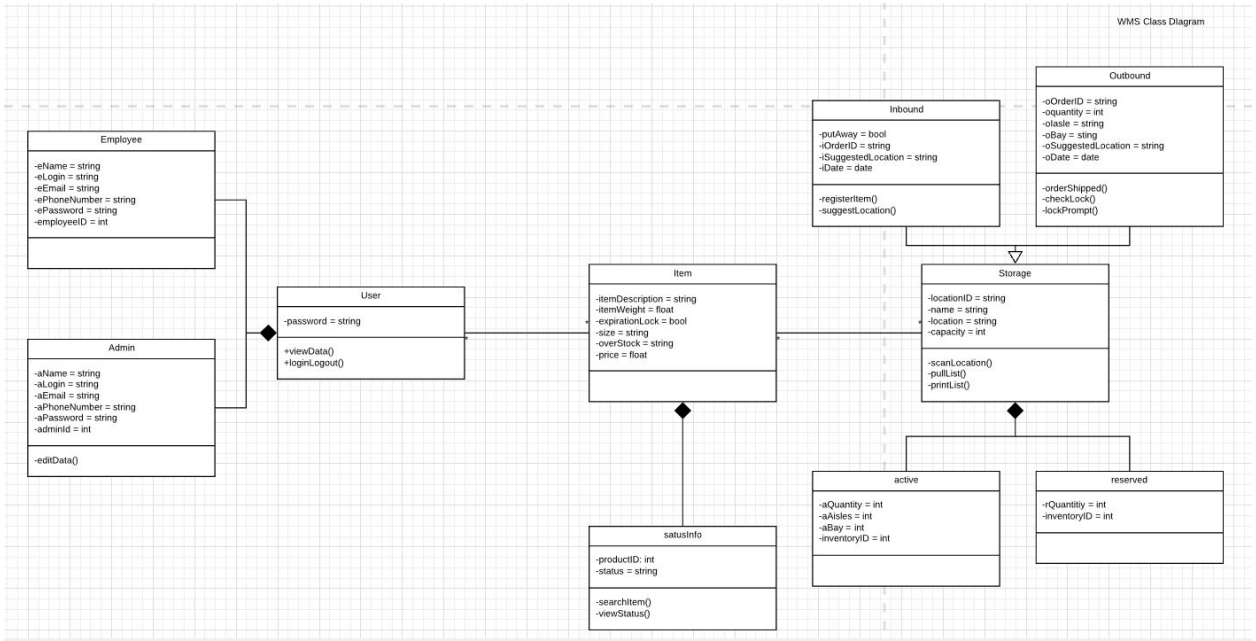
(I’m altering/adjusting the UI accordingly based on our project as it goes)




Dialog
Signed in as Admin
Logout
Warehouse Management System
View Data
Edit Data
aisle 1
test



Domain Analysis:
Analysis Objective Model: Class Diagram



Project Size Estimation Based off of Use Case Points:

Unadjusted Actor Weight

Actor type	Description of how to recognize the actor type	Weight
Simple	The actor is an employee that will be viewing data.	0.5
Average	The actor is the admin that will be editing and viewing data.	1.0
Complex	The actor is a scanner that will be used to input data into the database	3.0

UAW = 6

- Actors: Employee, Admin and Scanner

Weight per actor: Employee .5, Admin 1.0, Scanner 3.0

Unadjusted Use Case Weight

Use case category	Description of how to recognize the use-case category	Weight
Simple	Logging in to the GUI. Process takes no more than 2 steps	0.5
Average	Logging out of the GUI. Process takes 1 step.	0.5
Complex	User registers the itemID in WMS. This process only has one participating actor and should take no more than 4 steps.	2.5
Complex	After the user registers the itemID the WMS will suggest a putaway location. Process will take less than 6 steps.	3.0
Average	User receives the lock prompt after encountering an expired product. Process will take 3 steps. Only one participating actor.	2.0
Simple	Switching between inbound and outbound modules. This process will take only 2 steps.	1.0
Average	The status of the item is generated. Process will only take 4 steps. Only one actor.	1.5
Average	WMS will generate a report of the orders shipped. This process will take less than 3 steps. One participating actor.	1.0

UUCW= 80

UUCP=86

Technical Complexity Factors:

Factor	Description	Weight	Perceived Impact	Calculated Factor
T1	The Interface should be easy for the users to operate.	0.5	1.0	0.5
T2	The database search queries should not take more than 5 seconds.	1.0	1.5	1.5
T3	The system and all the data should all be backed up.	3.0	4.0	12.0
T4	The system should be allowed to handle multiple different users at once.	2.0	3.0	6.0
T5	The system should be operable on multiple different types of devices	1.0	3.0	3.0
T6	A system update will be outside of company operating hours and should not take more than 3 hours to complete.	2.0	3.0	6.0

Technical Complexity Factor Total: 29.0
TCF = 0.89

Environmental Complexity Factors:

Factor	Descrition	Weight	Perceived Impact	Calculated Factor
E1	Team members having a busy schedule	1.5	3	4.5
E2	Beginner familiarity with UML based development	1.5	2.0	3.0
E3	Knowledge with Graphical User Interfaces	2.0	2.5	5
E4	Knowledge with databases and application	1.0	2.0	2
E5	Highly motivated team, but some members are too busy with other projects sometimes	1	2.5	2.5
E6	Understanding how to write report as well as implement code at the same time	2.0	3.5	7.0
E7	No set leader	1.0	2.5	2.5

Environmental Factor Total: 26.5
ECF = 0.605

Total Number of Use Case Points:

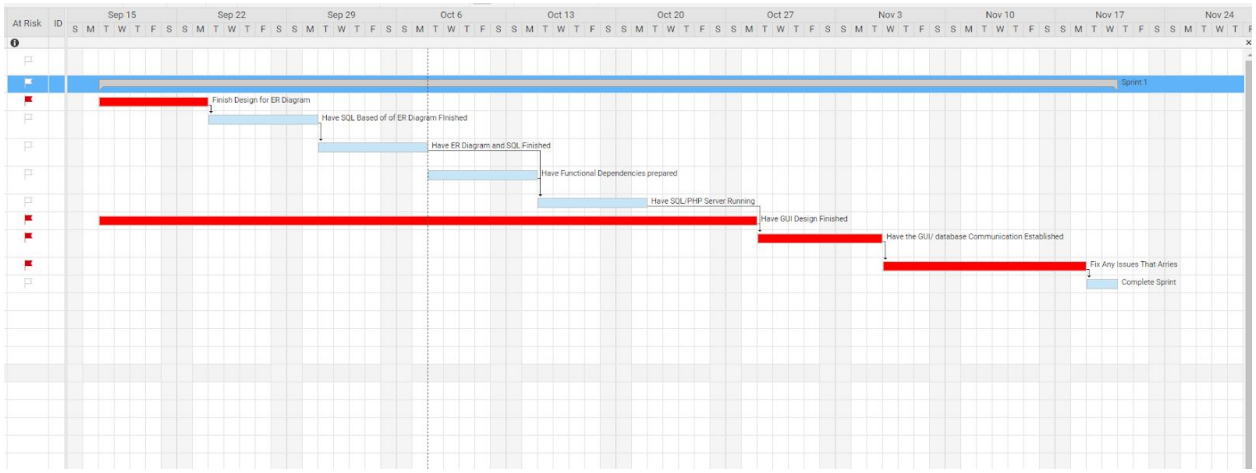
UCP = UUCP x TCF xECF

UCP = 46.3

Plan of work: Gantt Chart

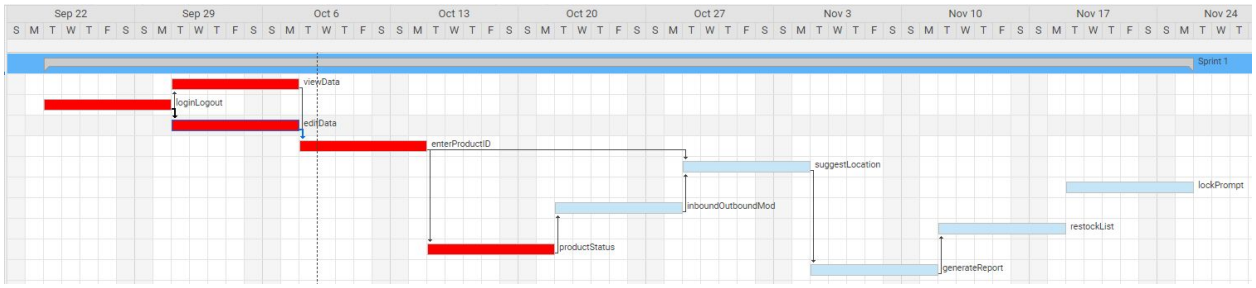
Gantt Chart For System Design:

At Risk	ID	Task Name	Start Date	End Date	Duration	Predecessors
		For help with Gantt Charts in Smartsheet, click to check out the help article.				
		Sprint 1	**09/17/19**	**11/20/19**	**47d**	
	1	Finish Design for ER Diagram	09/17/19	09/23/19	5d	
	2	Have SQL Based of of ER Diagram FInished	09/24/19	09/30/19	5d	3
	3	Have ER Diagram and SQL Finished	10/01/19	10/07/19	5d	4, #REF
	4	Have Functional Dependencies prepared	10/08/19	10/14/19	5d	
	5	Have SQL/PHP Server Running	10/15/19	10/21/19	5d	5, 6
	6	Have GUI Design Finished	09/17/19	10/28/19	30d	
	7	Have the GUI/ database Communication Established	10/29/19	11/05/19	6d	7, 8
	8	Fix Any Issues That Arries	11/06/19	11/18/19	9d	9
	9	Complete Sprint	11/19/19	11/20/19	2d	10



Gantt Chart for Use Case Implementation:

At Risk	ID	Task Name	Start Date	End Date	Duration	Predecessors
		Sprint 1	**09/24/19**	**11/25/19**	**45d**	
	Use Case 1	viewData	10/01/19	10/07/19	5d	3
	Use Case 2	loginLogout	09/24/19	09/30/19	5d	
	Use Case 3	editData	10/01/19	10/07/19	5d	3
	Use Case 4	enterProductID	10/08/19	10/14/19	5d	4, 2
	Use Case 5	suggestLocation	10/29/19	11/04/19	5d	8, 5
	Use Case 6	lockPrompt	11/19/19	11/25/19	5d	
	Use Case 7	inboundOutboundMod	10/22/19	10/28/19	5d	10
	Use Case 8	restockList	11/12/19	11/18/19	5d	11
	Use Case 9	productStatus	10/15/19	10/21/19	5d	5
	Use Case 10	generateReport	11/05/19	11/11/19	5d	6



The ownership of this project belongs to Alejandro Vasquez, Antonio Mendoza , Andres Dominguez, Sela Khoun.
Our break down of work done is included on the individual work breakdown section.

References:

References Used:

LucidChart. “UML Class Diagram Tutorial.” *YouTube*, 10 October. 2019. https://www.youtube.com/watch?v=UI6lqHOVHic.

QT. “Qt Tutorials For Beginners 10 - Simple Login app using QT.” *Youtube,* 18 April. 2016 https://www.youtube.com/watch?v=6_elY8O20I8

QT. “C++ Qt 15 - QPushButton.” *Youtube*, 20 January, 2011 https://www.youtube.com/watch?v=dOovH6TvaY4

QT.”C++ Qt 26 - QMessageBox.” *Youtube,* 29 January, 2011 https://www.youtube.com/watch?v=zwCjcZD7GlU

QT.”Qt Tutorial : C++ Notepad App.” *Youtube*, 22 January 2018 https://www.youtube.com/watch?v=I96uPDifZ1w

QT.”QT C++ GUI Tutorial 5- How to open a new window from a pushbutton in Qt.” 27 July, 2013 https://www.youtube.com/watch?v=tP70B-pdTH

Detail of problem statement:

Goal is to provide the customer with a software tool that will help their business be more effective on the warehouse floor. The WMS will provide security, performance and availability. The operation system that is used in some warehouses of today can be outdated. That means the productivity

level can be improved. There are warehouses of today that do most transactions on pen and paper. For example, after a shipment of goods is received the staff will lose sight of the item location. In many cases it is up to the employee to decide where the inventory will be stored. That is okay but a system is necessary to save the location number of where the product was placed. It is ineffective to solely rely on memory. If an item is later needed for shipment the employee must use memory to locate the product. This method is not feasible thousands of products are in the inventory. What will actually happen is the employee will spend unnecessary time walking around the warehouse looking for the product. There are other systems in the market that provide location registration of a product. Our WMS will have that feature but it will also recommend an empty location after receiving a product. When a product is received the WMS will scan locations and recommend the storage location by checking QTY. This will save time by improving the put away speed.

The customer is dealing with healthcare products therefore the products will eventually expire. Problems may arise if there is no system in place that is keeping track of dates. Usually warehouses are cycle counted manually and that is when dates are checked for expiration. Cycle counting can take a long time when there are thousands of products in the warehouse. It is common for a product to expire before it has been counted yet. This happens because there is just no enough time and labor for this method to be efficient. To add that means expired product can be shipped out if employees are not careful. Our WMS will keep track of dates from the beginning. As soon as an item is received it will have a date attached to it. In the future when the product expires an automatic lock will be applied to the item. This feature will prevent the item from being shipped out. Our system will save labor as cycle counts will no longer be needed regularly. Also WMS will minimize expired goods returns.

Most supply chain warehouses have two different locations where the product is stored. A location is used to store the overstock and the other location stores the product that is frequently used. This system can be effective is the right system is used. The problem is when there is no database that keeps track of the min and max of each location. If there is no system in place, then an employee must visually see which locations are low in stock. Our system will check the min and max of the active location. That way whenever it runs below the minimum accepted threshold the WMS will notify an employee to restock the location. This will assure that the stock is always up to the level needed for maximum effectiveness. The reserve location will be used to replenish the active location.

Acceptance Test Cases:

<table>
<tr><td colspan="2">Test Case Identifier: TC1
Use Case Tested: viewData
Pass/Fail Criteria: The test passes if the employee is allowed to view data.
InputData: view data button</td></tr>
<tr><td>Test Procedure:</td><td>Expected Result:</td></tr>
<tr><td>Step 1: click on the view data button in employee mode.

Step 2: click on the edit data button in employee mode.</td><td>Step 1:The employee is allowed to view data.

Step 2: The employee is denied permission and an error message is addressed to the user.</td></tr>
</table>

Test Case Identifier: TC2 **Use Case Tested:** Login **Pass/Fail Criteria:** The test passes if the employee and administrator can log in by entering the correct username and password. **InputData:** Admin: edit data button. Employee: view data button	
Test Procedure:	**Expected Result:**
Step 1: Employee types in his/her username and password **Step 2:** Employee clicks on view data **button** **Step 3:** Employee clicks on edit data **Step 4:** Admin clicks on edit data	**Step 1:** the system allows for log in once the username and password is verified Step 2: Employee is allowed to see data Step 3: Employee is denied to edit data. Step 4: Admin is allowed to edit data.

Test Case Identifier: TC3 **Use Case Tested:** editData **Pass/Fail Criteria:** The test passes if the admin is allowed to edit data, the employee is denied permission to edit data, and the admin and the employee are allowed to view data. **InputData:** edit data button, view data button	
Test Procedure:	**Expected Result:**
Step 1: click on the view data button in employee mode. **Step 2:** click on the edit data button in employee mode. **Step 3:** click on the view data button in admin mode. **Step 4:** click on the edit data button in admin mode.	**Step 1:**The employee is allowed to view data. **Step 2:** The employee is denied permission and an error message is addressed to the user. **Step 3:**The admin is allowed to view data. **Step 4:** The admin is allowed to edit data.

Test Case Identifier: TC4 **Use Case Tested:** enterProductID **Pass/Fail Criteria:** The test passes if the user is allowed to enter the product ID. **InputData:** Enter Product ID button	
Test Procedure:	**Expected Result:**
Step 1: The user will click on the Enter Product ID button.	**Step 1:** The system will prompt the user to enter the product ID.

Step 2: The user will enter the product ID and click on the Submit button.	**Step 2:** The product ID will be stored in the system.

Test Case Identifier: TC5 **Use Case Tested:** suggestedLocation **Pass/Fail Criteria:** **InputData:** scanner	
Test Procedure:	**Expected Result:**
Step 1: Scan a product where similar products on the floor have space next to them. **Step 2:** Scan a product the warehouse is in low supply of. **Step 3:** Scan a product the warehouse is in maximum supply of.	**Step 1:** The system will tell the employee to store the product near other similar product. **Step 2:** The system will tell the employee to replenish a low supply location. **Step 3:** The system will tell the employee to store the item in a reserved location off of the floor.

Test Case Identifier: TC6 **Use Case Tested:** lockPrompt **Pass/Fail Criteria:** The system fails if it allows an expired product to be processed by not populating the warning prompt. **InputData:** Scanner, Generate ticket button	
Test Procedure:	**Expected Result:**
Step 1: Hits the generate pick ticket button in the outbound module. **Step 2:** Scans a product that is expired. **Step 3: Attempts to close an order with an expired product.**	**Step 1:** The system will generate a list of items that will need to be picked. **Step 2:** The system will generate a prompt “Warning: Expired inventory” **Step 3:** The system will not process the order until the item with a lock is removed from list.

Test Case Identifier: TC7 **Use Case Tested:** inBound/outBound **Pass/Fail Criteria:** The test passes when the inbound goes into the designated storage/location and the outbound is pulled from the storage/location. **InputData:** show location button	
Test Procedure:	**Expected Result:**

Step 1: When inbound show location button is clicked. **Step 2:** when outbound show location button is clicked.	**Step 1:** The system will show the location to which the incoming items to be stored at. **Step 2:** The system will show the location to which the outgoing items to be pulled from.

Test Case Identifier: TC8
Use Case Tested: restockList
Pass/Fail Criteria: The test passes if WMS provides the employee a list of locations that need to be replenished.
InputData: replenish button, view list button, print list button

Test Procedure:	**Expected Result:**
Step 1: Click on replenish button. **Step 2:** Click on view list. **Step 3:** Click the print list button.	**Step 1:** System will display the replenish page. **Step 2:** System will display a query of locations that are low in quantity. **Step 3:** System will print the list in a readable format.

Test Case Identifier: TC9
Use Case Tested: productStatus
Pass/Fail Criteria: The test passes if the user enters the productID and the status is provided by the system.
InputData: Product Status Button, Text Box

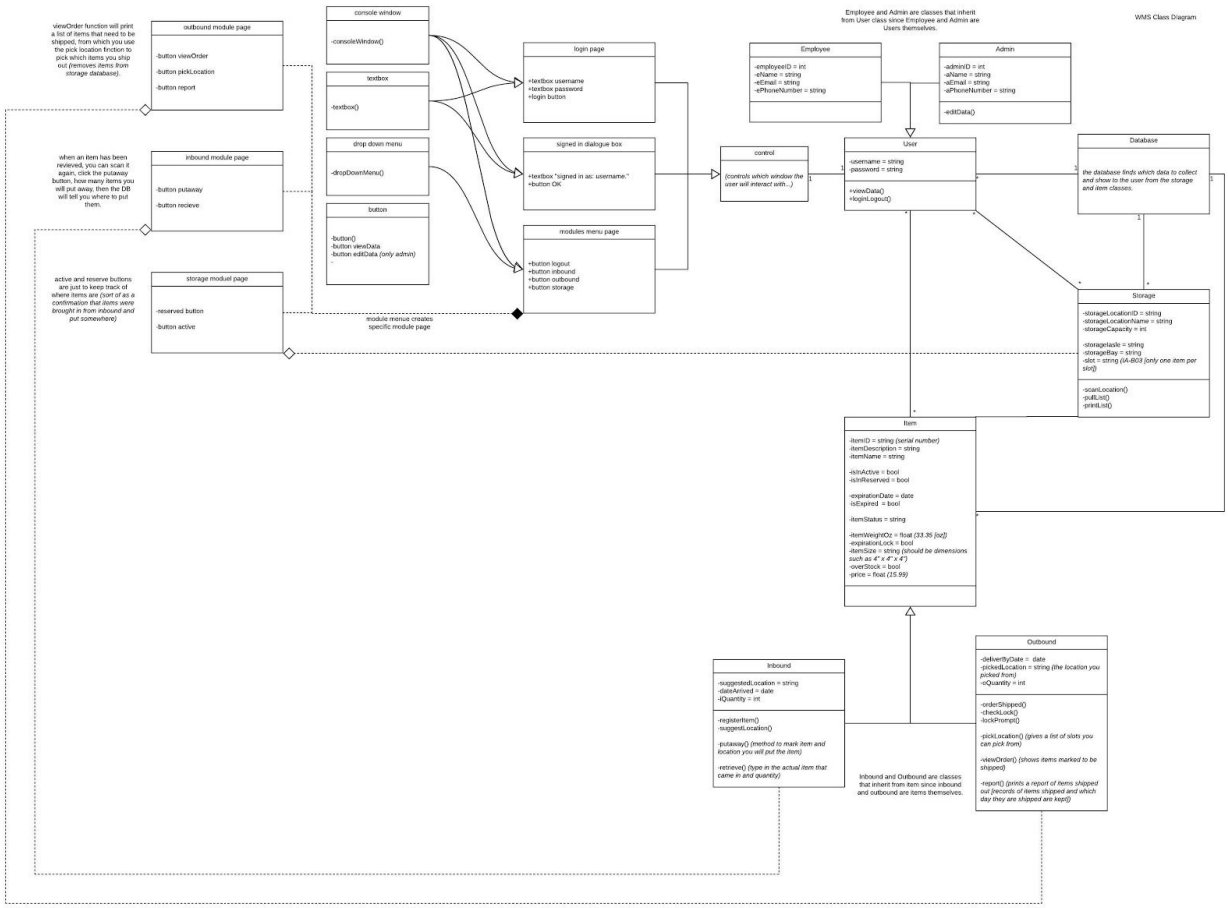
Test Procedure:	**Expected Result:**
Step 1: Clicks Product Status button. **Step 2:** User enters the productID **Step 3:** User clicks on the view status button.	**Step 1:** The system will open a page and inform the user to enter productID. **Step 2:** The system validates that the productID entered is valid. **Step 3:** The system will display the status for the productID.

Test Case Identifier: TC10
Use Case Tested: generateReport
Pass/Fail Criteria: The test passes if WMS generates a report displaying successfully shipped orders.

InputData: Generate Report Button	
Test Procedure:	**Expected Result:**
Step 1: User clicks the Generate Report button. **Step 2:** The user enters the date.	**Step 1:** The system will open a page asking the user to enter the date. **Step 2:** The system displays the report for the user to view.

Class Diagram and Interface Specifications:

Class Diagram:



Data types and Operation Signature:

Class name = Admin
This is a class that is tied to the user Admin

Attributes

- adminID: Unique ID of an admin
- aName: Name of the admin
- aEmail: Email that belongs to the admin
- aPhoneNumber: phone number that belongs to the admin

Functions

- editData(): This function will provide access to edit data

Class name = Employee
This is a class that will list attributes of the employee

Attributes

- employeeID: Unique ID of the employee
- eName: Name of the employee
- eEmail: Email that belongs to the employee
- ePhoneNumber: phone number that belongs to the employee

Class name = User
This class holds information that is inherited by the employee and admin

Attributes

- username: Unique username a user will use to log in
- password: Password that a user will need to log in

Functions

- viewData(): This function will provide the users access to view data in the GUI
- loginLogout(): This function will take care of the functionality of logging in and logging out

---

Class name = Item
the item table will show all descriptive traits of a certain product

Attributes

- itemID: each item received will have its unique item ID
- itemDescription: a string that will provide a short description of the item
- ItemName: the name of the item
- isInActive: Boolean data type will display a Y or N if the item is stored in the active table

- isInReserve: Boolean data type will display a ‘Y’ or ‘N’ depending if the item is stored in the reserve table
- expirationDate: the date the item expires
- isExpired: display Boolean depending if the product has expired
- itemStatus: shows the current stage of the item e.g “In Receiving”
- itemWeightOz: Weight of the item in ounces
- expirationLock: displays if the item has a lock due to being expired
- itemSize: will list the dimension of the item, format will be same as 4’ x 4’ x 4’
- overStock: will list Y or N if the item is in overstock, if it does not fit in active
- price: price of the item as a float

Class name = Outbound
This class will display all of the functions and attributes associated with the process of preparing an item to be shipped out (Exit the warehouse).

Attributes

- deliverByDate: suggested deadline of an item, item expected to be delivered by this date
- pickedLocation: the name of the location the user has picked from
- oQuantity: how many items the user has pulled

Functions

- orderShipped(): This function will pull a query of all the items that have being successfully shipped out
- checkLock(): Before an item is shipped out, this function will make sure that the items are free of locks
- lockPrompt(): This function will prompt a warning message if it identifies an item with a current lock
- pickLocation(): This function will get an order and display a list of slots that the user can pick from
- viewOrder(): This function will display in list view the orders that are ready to be picked

---

Class name = Inbound
This class will take care of the steps of receiving and registering a product. (Incoming to the warehouse)

Attributes

- suggestedLocation: the name of the location where the product will be stored next
- dateArrived: The date the item was checked in
- iQuantity: Indicates the quantity of an item being received

Functions

- registerItem(): This function will make sure to save the new Item entry into the database
- suggestLocation(): Depending on the space a newly registered item takes, it will suggest where the item should be stored.

Class name = Storage
This class will list the attributes and functions related to where the item is stored. (Where the item lives in the warehouse)

Attributes

- storageLocationID: unique ID with the specific name of the item's storage location. For example Isle 2 Bay 04 Slot 1
- storageLocationName: The category title of the storage location
- storageCapacity: Number that lists what is the maximum that location can hold
- storageAisle: Isle number of where the item is stored
- storageBay: Bay string of where the item is stored
- slot: Only one item fits in a slot, this will display the number of slot where the item is located.

Functions

- scanLocation(): This function will scan the storage locations for availability. It will be looking at quantities.
- pullList(): This function will pull a list and display the fields in the storage table.

printList(): This function will display and print a list of what is stored in the storage table, * May not need this function

Class name = Admin
This is a class that is tied to the user Admin

Attributes

- adminID: Unique ID of an admin
- aName: Name of the admin
- aEmail: Email that belongs to the admin
- aPhoneNumber: phone number that belongs to the admin

Class name = inbound module page
This is a class that triggers inbound related actions after button is pressed in UI

Actions

- button putaway: After this button is pressed a user is ready to putaway received items
- button receive: After this button is pressed a user is ready to receive items

Class name = Outbound Module
This is a class that triggers outbound related actions after button is pressed in UI

Actions

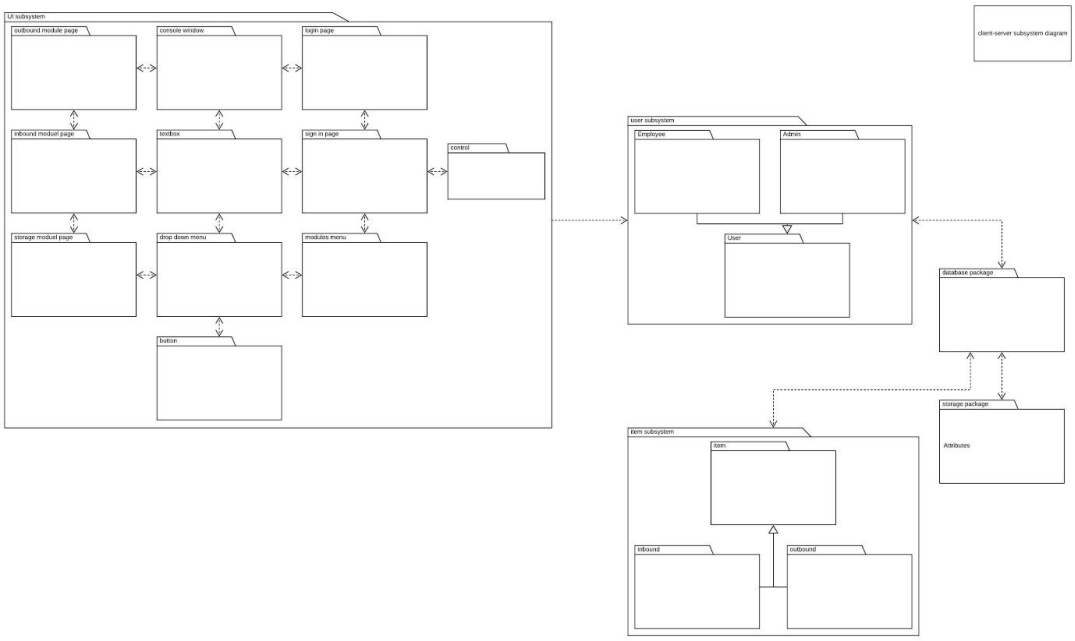
- button viewOrder: This button is a trigger to populate a list of orders that need to be shipped out
- button pickLocation: Once this button is hit the user is ready to pick items for shipment
- button report: This button is a trigger to generate a report of items already shipped

System Architecture and System Design:

Architectural Design:

We are using the client-server. There will be one server and a couple clients. The clients will communicate with the server by sending requests to get what they need. The Ui subsystem goes to the user subsystem. The user subsystem and the item subsystem go to the database package. The database package goes to the storage package.

Identify Subsystems:



This is our UML package diagram for our subsystems based off of the server-client architecture model.

Mapping Subsystems to hardware:

Hardware Type 1: Computer with Monitor, what the employee will use to view UI (Client)

Subsystems: User Subsystem, UI Subsystem, Storage package and Item Subsystem

---

Hardware Type 2: Scanner / Portable Device, device that will register the item (Client)

Subsystems: User Subsystem and item Subsystem

Hardware Type 3: Database Server (Server)

Subsystems: Database Subsystem, Storage package and Item Subsystem

---

Persistent Data Storage:

- Knowing the amount of products we have in stock such as inbound, and outbound is important to us, so we wanted to keep track of all of our data, including employees' information, and inventory.
- To save some space and resource, any inventory that has left the warehouse or has been sold for more than 12 months will be removed from the database.
- We will store our data inside the database for faster access.
- To access the data, we will be using MySQL.
- Depending on the constrain of our resource and how much data we will have, we might be able to store our data in flat file instead.

Network Protocol:

- The network protocol that will be used will be the transmission control protocol, and internet protocol. It will use an internet connection to connect the devices being used.
- Hypertext Transfer Protocol(HTTP) will be used to establish the connection.
- HTTP allows for the use of mobile devices for the warehouse management system.
- Using HTTP should allow for the different devices to use the same network, such as the mobile devices.

Hardware Requirements:

Server Requirements: *(support up to 100 terminals)*

- CPU: Intel Xeon processor 3 GHz or greater (4+ cores).
- RAM: 16 GB.
- 1 Gbps or faster network interface.
- 100 GB (for Operating System, Application, Logs, etc.).
- 1 TB or more (for Database).
- 500 GB (for case files).
- Access to the internet to host.

Computer requirements:

- RAM: 1 GB
- LCD Color Display (Monitor Resolution 1024 × 768 or higher).
- OS: Windows 7 or above.
- Keyboards/mice for data entries.
- Internet access (to connect to server).
- Portability (laptop).
- Battery life lasting at least 8 ½ hours (to cover one of the staff's full shift) or higher.

Scanner Requirements:

- WiFi Radio Module (works up to 100m from WiFi access point).
- Micro USB Recharging.
- Battery lasting at least 8 ½ hours or higher.
- Integrated Barcode Scanner (1D / 2D barcodes supported).
- LCD Display (Screen Resolution 640 × 480 or higher).
- Shock Resistance.
- Dust and Splash Resistance.

Algorithms and Data Structures:

Algorithms:

We are not using a mathematical model for our program.

Data Structures:

- Hash Tables: we will use hash tables to store our numeric and character data.
- Arrays: we will use arrays to handle variable values quicker.
- Linked lists: to organize and display the values entered by the user.

User Interface Design and Implementation:

We decided to keep our user interface the same as our prototype. the reason we were not making any changes is because we want our WMS to be as user friendly as possible for our customers yet effective at the same time. We don't want to confuse user by adding too many buttons or too less. All the buttons that appear on the user interface should be clear to the user. Users should be able to look at each button and determine what the button is supposed to do, whether the user needs to log in, looks for inventory or what needs to be restocked etc…

We want to keep our UI mainly simple to use, so we will not be focusing on the aesthetics yet. The reason why is because we do not want to confuse the employees using the WMS. A Lot of the actions will be repetitive and would like our system to be fast, effective and not use up many resources just running the UI alone.

In the future, when we update our software, one of our possible options is that we might be adding more buttons or search box for keywords on our interface as we think if it is necessary. We will also be adding more colors to the background of pages, this will be done to help the user differentiate in which module they are currently in.

Plan for Testing Non-Functional Requirements:

Nonfunctional requirement	How we will test the requirement	Expected Result
The interface should be easy for the users to operate.	We will get someone inexperienced with the ui to try it out and see how comfortably they work with it.	Volunteer will complete one UI use case in less than 30 seconds.
The database search queries should not take more than 5 seconds.	We will run common search queries and see how long they take and see if any queries need revising.	Queries will return in < 5 seconds.
The system and all the data should all be backed up.	We will check the database backup manager and see that it is backed up.	Back up system will have the same data as primary within 30 seconds of a primary save.
The systems should be able to handle multiple users at once.	We will log on to the system and see if any errors or lag occurs.	System will handle at least 3 users at one time.
The system should be operable on multiple different devices.	We will try to log onto the system with different types of devices and	Simultaneously update data from database tool and UI within 1

	see how the system handles it.	minute.
A system update will be outside of company operating hours and should not take more than 3 hours to complete.	We will perform an update and check the log to see how long it will take.	Successfully update system in < than 3 hours.

QT source codes and comments:

“widget.h” section:

I included some pre-difined QT’s functions like, QSqlTableModel is to have QT show mysql database in the table-like in the userinterface. QMessageBox is to show a prompt the user what message that they are about see or performce once a certain button is clicked. QSqlError is for QT to detect if the coneection from QT to mysql database is working or not such as whether if we enter the correct password or the access name. QSqlDriver is get QT driver to recognize mysql database tables. (unfortunately as spoken in class, all of the driver versions of QT ars NOT working at the moment).

Widget: Widget(QWidget *parent) and ui set up section:
I used “QMYSQL” for QT to connect to mysql (not working due to "dirver not loaded")
setDatabaseName("WMS”) this is what our mysql database connection server name is called.
Port 3306 is what listed on our database server part of the connections as well.
setPassword(“xxxx”); using password to grant access from QT to mysql database server.

if (!mDatabase.open())… QMessagebox is used here to let the user know if they have any error in connecting the mysql database server.

new QSqlTableModel(this);
setTable(“employee”); // is used to show all of the employee table information in our mysql database server that we have created so far.

Widget::~Widget()… Last section of source code:
mModel->insertRow(mModel->rowCount()); // this line is used to insert employee information to the employee table.

mModel->removeRow(ui->tableView->currentIndex().row()); // this line is used to delete employee's information from the table.

mModel->select(); // this line is used to update the employee's information to the table (after insert into the table).

(all of these codes seems to be running just find, but as I mentioned earlier that I couldn't get QT drivers, yet I was so close of getting everything to work.) I will submit these code and stick with it for now and would like to spend more time working on QT driver hoping to get to finally run).

QT "Tutorial Qt Creator - MySQL/MariaDB (SQL en C++)" *Youtube, 20 January, 2015* https://www.youtube.com/watch?v=yxy0yvZnX1Y&t=1025s

SQL Data Definition Language

SQL Statements used for creating tables:

```
CREATE TABLE `employee` (
  `employeeID` varchar(45) NOT NULL,
  `eName` varchar(45) NOT NULL,
  `eMail` varchar(45) NOT NULL,
  PRIMARY KEY (`employeeID`)

CREATE TABLE `customer` (
  `customerID` varchar(45) NOT NULL,
  `cName` varchar(45) NOT NULL,
  `cAdress` varchar(45) NOT NULL,
  `cCity` varchar(45) NOT NULL,
  PRIMARY KEY (`customerID`)

CREATE TABLE `admin` (
  `adminID` varchar(45) NOT NULL,
  `aName` varchar(45) NOT NULL,
  `aMail` varchar(45) NOT NULL,
  PRIMARY KEY (`adminID`)

CREATE TABLE `inbound` (
  `itemID` varchar(45) NOT NULL,
  `dateArrived` date NOT NULL,
  `iQuantity` int(11) NOT NULL,
```

```
  `suggestedLocation` varchar(50) NOT NULL,
  FOREIGN KEY (`itemID`)

CREATE TABLE `item` (
  `itemID` varchar(45) NOT NULL,
  `itemDescription` varchar(45) NOT NULL,
  `expirationDate` date NOT NULL,
  `itemStatus` varchar(45) NOT NULL,
  `itemSize` varchar(45) NOT NULL,
  `isExpired` varchar(45) NOT NULL,
  `isInActive` varchar(45) NOT NULL,
  PRIMARY KEY (`itemID`)

CREATE TABLE `outbound` (
  `orderNumber` varchar(45) NOT NULL,
  `itemID` varchar(45) NOT NULL,
  `pickedLocation` varchar(45) NOT NULL,
  `oQuantity` varchar(45) NOT NULL,
  `deliverByDate` date NOT NULL,
  `customerID` varchar(45) NOT NULL,
 PRIMARY KEY (`orderNumber`)
FOREIGN KEY (‘customerID’)

CREATE TABLE `storage` (
  `storageLocationID` varchar(50) NOT NULL,
  `storageLocationName` varchar(45) NOT NULL,
  `itemName` varchar(45) NOT NULL,
  `stockQuantity` varchar(45) NOT NULL,
  PRIMARY KEY (`storageLocationID`)
```

Sample insert statement:

```
INSERT INTO `legacywms`.`item` (`itemID`, `itemDescription`, `expirationDate`, `itemStatus`, `itemSize`, `isExpired`, `isInActive`) VALUES ('773935', 'NEFAZODONE HCL 200MG TAB 60/BO', '2019-11-10', 'In Storage', '21', 'FALSE', 'true');

INSERT INTO `legacywms`.`storage` (`storageLocationID`, `storageLocationName`, `itemName`, `stockQuantity`) VALUES ('KACC41DD06', 'Reserve', 'OFLOXACIN 0.3% OTIC DROP 10ML', '9');
```

Sample edit statement:

```
UPDATE `legacywms`.`storage` SET `stockQuantity`='7' WHERE `storageLocationID`='KACC42AA01';
```

1. **History of work, current status and future work**:

    a) We came across a problem with the Qt IDE we were using, where it would not allow support to our database. Due to this reason, we had to scrap our semester's worth of development in our GUI and find another IDE to develop our GUI. We asked other developers for advice, with more experience than our team, and we were referred to using the Visual Studios 2019 IDE. Now, we are working our way to catch back up. At our current state, the database still has not connected to the GUI and we only have a simple GUI since we are still adjusting to the differences between the Qt IDE and the Visual Studios IDE as well as trying to cope with other exhaustive projects that require just as much attention. Our client/leader requested a report on the issue by tuesday (11/26/2019) where we hope to have a functional GUI that communicates and allows queries to the database. Due to our major setback, we are behind our timeline, but we hope to catch on back up to it with in a week with optimism. Although we are behind not all work was lost in vain. We have a solid idea on how the application should look like once we get it running. It's only a matter of coding and building the right queries to fetch the data.

    b) **Key accomplishments**:

We were able to get a functioning database and a functioning GUI so far. We are now back on track to making progress in our project.

    c) **Possible future directions**:

As we mentioned during our presentation, we won't be able to get QT to connect to MySQL database due to DRIVER failure, therefore we would have to migrate our project to a different program. One of the programs that we are using to create the user interface and to connect to MySQL database is Visual Studio. We are still experiencing with it, if this works out, we will use it for our final project, otherwise we would have to move our project to a flat file or some other programs.

With another sprint, we could optimize our queries to be as most cost efficient as well as deliver exceptional results. We could continue on to "pretty" up our software by adding graphics, and color coding to allow easier reading on the user. With more time we would have liked to include more text box prompts that would interact with the user.

2. **References**:

We used a variety of videos on youtube to help us better understand how to connect the database to the GUI. One of these youtubers is programmingknowledge. Most of the Database development was done by our development team due to our prior experience in work with development of database structure. Mostly the only thing we needed outside sources for was the development of the GUI.

William was the other developer that helped our development team get back on track with our goals and timeline.