

# Milestone 4: Warehouse Management System

---

## Requirements and Preliminary Design

Xingze Guo  
xig480

Rongli Han  
roh919

Xianming Luo  
xil430  
**10/27/2013**

Spencer Ondrusek  
spo798

Matt Triff  
mdt879

## 1. Summary Use Cases

**Use Case Name:** Statistics Reporting

**Level:** Summary

**Actors:** Manager

**Goal:** Provide a high level view of warehouse operations, to drive efficiency gaining decision making

**Activities:** Analyze staff efficiency for promotion and layoffs, view turnover speed of specific products to guide purchase order and discount sales decisions, gauge warehouse stock level

**Quality:** high quality, fast, accurate, and reliable

**Version:** September 28, 2013

**Use Case Name:** Stock Reporting

**Level:** Summary

**Actors:** Manager

**Goal:** Provide in depth view of information about warehouse stock

**Activities:** Monitor stock levels, view stock information for all products currently or previously contained in the warehouse

**Quality:** high quality, fast, accurate, and reliable

**Version:** September 28, 2013

**Use Case Name:** Routing Stocking Tasks

**Level:** Summary

**Actors:** Stock Handler

**Goal:** Increase efficiency of navigating the warehouse for stocking tasks

**Activities:** Putting away stock that has been received, gathering stock for an upcoming shipment, building pallets correctly

**Quality:** high quality, accurate, reliable, and effective

**Version:** September 28, 2013

**Use Case Name:** Receive Purchase Order

**Level:** Summary

**Actors:** Receiver, Stock Handler

**Goal:** Add new stock items to the system

**Activities:** Recording information on newly received stock, increasing warehouse stock level to correspond to received shipments, assign Stock Handler to store the product

**Quality:** high quality, accurate, high ease of use, reliable, flexible

**Version:** September 28, 2013

**Use Case Name:** Send Shipments

**Level:** Summary

**Actors:** Shipper, Stock Handler

**Goal:** Send the items ordered for shipment from the warehouse

**Activities:** Remove stock from the warehouse, track pallet contents, record progress of Stock Handlers in gathering items required for shipment.

**Quality:** high quality, accurate, high ease of use, reliable, flexible

**Version:** September 28, 2013

**Use Case Name:** Locate Stock

**Level:** Summary

**Actors:** Stock Handler, Manager, Shipper, Receiver

**Goal:** Find stock location within the warehouse

**Activities:** Find the location of the stock within the warehouse using the GUI to click on an overhead map to view the products in each bin, display location on stock reports, search using the main screen of the GUI by item number or name to highlight on the GUI the location of the stock. Display stock location when the shipper creates shipping tasks.

**Quality:** high quality, very accurate and reliable

**Version:** October 8, 2013

**Use Case Name:** Modify Inventory

**Level:** Summary

**Actors:** Receiver, Manager, Loss Prevention, Stock Handler

**Goal:** Add and correct stock information

**Activities:** Add new item information to the system, which would include name, description, item number, category, price, quantity, and location. Alter the quantity of a stock item, in the case of a lost, damaged, or stolen item. Change the location of the item in the warehouse when it has been moved. Remove an item from the system if it has been discontinued or will no longer be carried by the warehouse.

**Quality:** high quality, high ease of use, reliable

**Version:** October 8, 2013

**Use Case Name:** Order Stock

**Level:** Summary

**Actor:** Stock Handler, Manager, 3<sup>rd</sup> Party Product Suppliers

**Goal:** Manager orders additional products when stock levels are low

**Activities:** After viewing the Stock Level Report, the Manager will identify which products they will require more of for future shipments to customers. The Manager will create a new purchase order using the Management & Reporting System. They will then enter the item number and the quantity for the order, the system will then get the item name, size, and weight information and

attach it to the purchase order, which will be sent to the 3<sup>rd</sup> Party Product Suppliers.

**Quality:** high quality, high ease of use, reliable

**Version:** October 20, 2013

**Use Case Name:** Login

**Level:** Summary

**Actor:** Stock Handler, Manager, Shipper, Receiver

**Goal:** Allow users the appropriate level of access to the system

**Activities:** The GUI will prompt any user wishing to access the system for their username and password. Each user will be assigned their own username and password for the system. The system will only allow users access to the sections for their position. If a user attempts to access a section they are unauthorized for, they will be denied.

**Quality:** high quality, high ease of use, reliable

**Version:** October 20, 2013

**Use Case Name:** Manage Employees

**Level:** Summary

**Actor:** Manager

**Goal:** Provide HR functions to the manager, allow the addition of new employees, removal of old employees, and changing the position of an employee when they are promoted or change roles

**Activities:** Allows a manager to look up an employee, by name or by ID number. Once the manager has selected the employee, they can remove them from the system, or change their title to a different position. New employees can also be added through a form inputting their name and title.

**Quality:** high quality, high ease of use, reliable

**Version:** October 10, 2013

## 2. Fully-Dressed Use Cases

**Name:** Locate Stock

**Scope:** Stock Handler

**Level:** Subfunction

**Primary Actor:** Stock Handler

**Stakeholders and Interests:**

- Stock Handler, Manager, Shipper, Receiver: Find stock location within the warehouse

**Preconditions:**

- A Warehouse map has been coded into the GUI that is synonymous with the actual warehouse.
- Locations of stock are accurately recorded in the database.
- The warehouse's floor plan changes have been recorded in the database.
- The person knows the item ID for the item they are searching for.
- User has been logged in and authenticated by the System

**Main Success Scenario:**

1. From the main screen/map view of the warehouse, user selects the field "Locate Stock".
2. User enters in the item's product ID and clicks "OK".
3. The bins containing that item within the warehouse are highlighted on the map.
4. The user clicks on one of the highlighted bins, showing a detailed listing of the products in that bin, with the item being searched for highlighted.

**Extensions:**

- The item number they are looking for is incorrect. An error message is displayed to the user.
- Item does not currently have a location. The System sends an error message, notifying the user of this, and attaches all the information about the product currently in the database.

**Special Requirements:** None.

**Technology and Data Variations List:**

- User enters the numerical ID for the product
- User attempts to identify the product by name (close matches will be identified)

**Frequency of Occurrence:**

- New employees: Very often, due to new employees low level of familiarity with the warehouse
- Seasoned employees: Daily. Depending on the person and experience this system is likely to be used between a few times each day to a few times a week.

**Miscellaneous:**

- The user should also be able to browse item locations in a list format. They might vaguely remember an item and need a piece of information to jog their memory.

**Name:** Make Shipment

**Scope:** Shipper, Stock Handler

**Level:** User Goal

**Primary Actor:** Shipper

**Stakeholders and Interests:**

- Shipper: Needs to check the pending shipments, assign tasks to a Stock Handler and a shipping company
- Stock Handler: Requires a well organized list of items to be gathered
- Shipping Company: Wants to know the shipment specifications, such as name, weight, dimensions and destination
- Manager: Wants to know what shipments have been sent

**Preconditions:** Shipper is logged in and authenticated

**Success Guarantee:** The shipment task is assigned to a stock handler and a shipping company, the shipping record is saved.

**Main Success Scenario:**

1. Shipper opens the Shipment menu.
2. Shipper checks confirms destination, all required items are in the warehouse.
3. Shipper selects the “Update Shipment”.
4. Shipper inputs the tracking number and estimated delivery date.
5. Shipper assigns the shipment to a shipping company for pick-up and delivery.
6. Shipper assigns the shipment to a Stock Handler, to gather and transfer the products.
7. System saves the shipping record to the database.
8. System notifies the Shipper that the shipment was successfully updated and assigned
9. Shipper exits the Shipment menu.

**Extensions:**

- At any time, the Shipper can view the shipment specifications, but does not need to assign the shipment to a shipping company or Stock Handler.

**Special Requirements:**

Because the database updates concurrently from many different users, the list of shipment specifications needs be refreshed automatically every minute.

**Technology and Data Variations List:**

- Shipper chooses the requirement, Stock Handler and a shipping company by using a mouse to select from a drop-down menu of valid choices.
- Shipper inputs the tracking number and estimated delivery date by keyboard.
- The form of tracking number is sixteen digits.
- The form of delivery time is YYYY/MM/DD (e.g. 2013/10/25).
- Stock handlers and shipping companies are identified by name.

**Frequency of Occurrence:** Many times per day, depending on the rate of shipment requests.

**Miscellaneous:**

- Open Issue: What if the shipment needs to be canceled or changed?

**Name:** System Login

**Scope:** Warehouse Management System

**Level:** User Goal

**Primary Actor(s):** All Users of the System (Manager, Shipper, Receiver, Stock Handler)

**Stakeholders and Interests:**

- Manager: Login to update shipment and purchase orders, and monitor the status of the warehouse.
- Shipper: Login to update shipment specifications, assign shipment tasks.
- Receiver: Login to update products, input received purchase orders to the System, assign purchase order tasks.
- Stock Handler: Login to look up product location, receive and print daily tasks.

**Preconditions:**

- User has the application for the System launched.
- User has their account registered in the System.
- User enters a valid username and password that match the account registration information.
- User has been granted permission to access the system.

**Success Guarantee:**

- Username and password entered are correct.
- User can now access sections of the System to complete their tasks.

**Main Success Scenario:**

1. User enters a username that matches a record in the database.
2. User enters a password that matches the same record in the database.

**Extensions:**

- Users enter a wrong user name
- Users enter a wrong password
- Users enter a wrong user name and a wrong password
- The database does not record the information of users
- Users do not have valid permission to login the system

**Special Requirements:** None.

**Technology and Date Variations List:**

- Touch screen digital keyboard
- Physical keyboard

**Frequency of Occurrence:**

- Individual terminals: Once per day.
- Shared terminals: Many times per day as users switch to access their own functionality.

**Miscellaneous:**

- Need to prompt users to enter a new login when they attempt to access unauthorized sections of the system to prevent a full reboot when new functionality is needed

**Name:** Routing Stocking Tasks

**Scope:** Stock Handler

**Level:** Subfunction

**Primary Actor:** Stock Handler

**Stakeholders and Interests:**

- Manager: Desires Stock Handlers to be as efficient as possible, that they not attempt any actions that are impossible to complete (not enough space, stock not available for an order).
- Receiver: Have products put away as quickly as possible, ensure that all items are able to be stored in the warehouse.
- Shipper: Have products gathered quickly and efficiently, on a well formed pallet, with all the items required for the order.
- Stock Handler: Handle tasks as quickly and easily as possible, without errors or backtracking on the warehouse floor

**Preconditions:**

- Stock has been received by the Receiver, is sitting at the warehouse entrance
  - Stock is ready to be placed into bins in the warehouse for storage
- Order has been received by the Shipper
  - All stock required for the order is within the warehouse

**Success Guarantee:**

- Most efficient route has been located
- Items are ordered by their destination within the warehouse
- Items are ordered by their location within the warehouse for pickup

**Main Success Scenario(s):**

1. Receiver sends the purchase order manifest to this sub-function
  2. Items in the list are checked for a current location within the warehouse
  3. Items are assigned a bin to be placed in (with the old stock if it exists, otherwise a closest available location, where it will fit
  4. Order of the items is returned to another sub-function to create a printable plan for the Stock Handler
- 
1. Shipper sends the shipment manifest to this sub-function
  2. Items in the list are checked to confirm they are in the warehouse, and their current location(s) are recorded
  3. Route is planned that will take into account the weight of the items being gathered (heaviest first), and the location of the items throughout the warehouse.
  4. Order of the items is returned to another sub-function to create a printable plan for the Stock Handler



**Extensions:**

- Unable to store the items in the warehouse due to lack of space, return an error message to notify the user, do not return a list of items
- Unable to locate all the items required for the shipment, return an error message to notify the user, do not return a list of items

**Special Requirements:** Must not specify that the stock be stored in a bin that it does not fit.

**Technology and Data Variations List:**

- Input of a list of items to be stored
- Input of a list of items to be retrieved.
- Output is the sorted list of items that were entered, sorted in the order that they should be handled

**Frequency of Occurrence:**

- Multiple times per day, dependent on number of Stock Handler tasks.

**Miscellaneous:** None

**Use case name:** Manage Employees

**Scope:** Manager

**Level:** User Goal

**Primary Actor:** Manager

**Stakeholders and Interests:**

- Manager: Needs to add and removes employees from the system, record how fast they complete tasks, give employees a promotion or demotion
- Stock Handler, Shipper, Receiver: Report task completion speed.

**Preconditions:**

- Manager has been logged in and authenticated

**Success guarantee:**

- Staff information is recorded in the System
- Staff information accurately reflects the Manager's input

**Main Success Scenario:**

1. Manager accesses the Management menu.
2. Manager chooses to view a list of all employees, or searches for a specific employee by ID, name, or title.
3. Manager edits the employee's name, title, or speed information.
4. The updated information is stored in the database.
5. The Manager is notified that the changes have been saved.

**Special Requirement:**

- Employee information should be refreshed every minute, in the case that multiple managers are using the system.

**Technology and Data Variations List:**

- Employee name and speed can be entered with physical keyboard.

- Employee name and speed can be entered with a touch screen.
- Employee title will be selected from a drop down menu.

**Frequency of Occurrence:**

- Daily to enter employee speed information.
- Every few weeks or months for editing, adding or removing employee information.

**Issue:** None.

### 3. Supplementary Specification

**Reliability**

- Concurrency
  - The data shown to the user must be an up to date representation of the database.
- Performance
  - Users should be able to input and receive output data quickly.
  - The time cost of each operation should be less than 5 seconds.

**Supportability**

- Configurability
  - Any warehouse can utilize our system by only setting up their floor plan and warehouse location information.
  - Robust databases enable our system to support the world's largest warehouse or a small scale warehouse in the back of a shop.
- Adaptability
  - Due to the modular nature of our classes and the flexibility of Java we are able to easily create new specifications for different clients.
  - The warehouse program is able to work in a multi-tiered environment.
  - Multiple users can update the database at the same time without needing to worry about locks due to a single database updated using atomic actions.

**Usability**

- Human factors: The user of the warehouse system will be able to use a touchscreen monitor or keyboard and mouse to operate, thus,
  - Font should be clear to view, and the size of the text need be 12 to 14.
  - High contrast between background colours and foreground elements.
  - Consider warehouse system response time, database update speed and handling of the system in the case of a crash.
  - The warehouse system will be compatible with any operating system that can run the Java Virtual Machine.
  - Interactive map and menus are clean, simple, and easy to use.

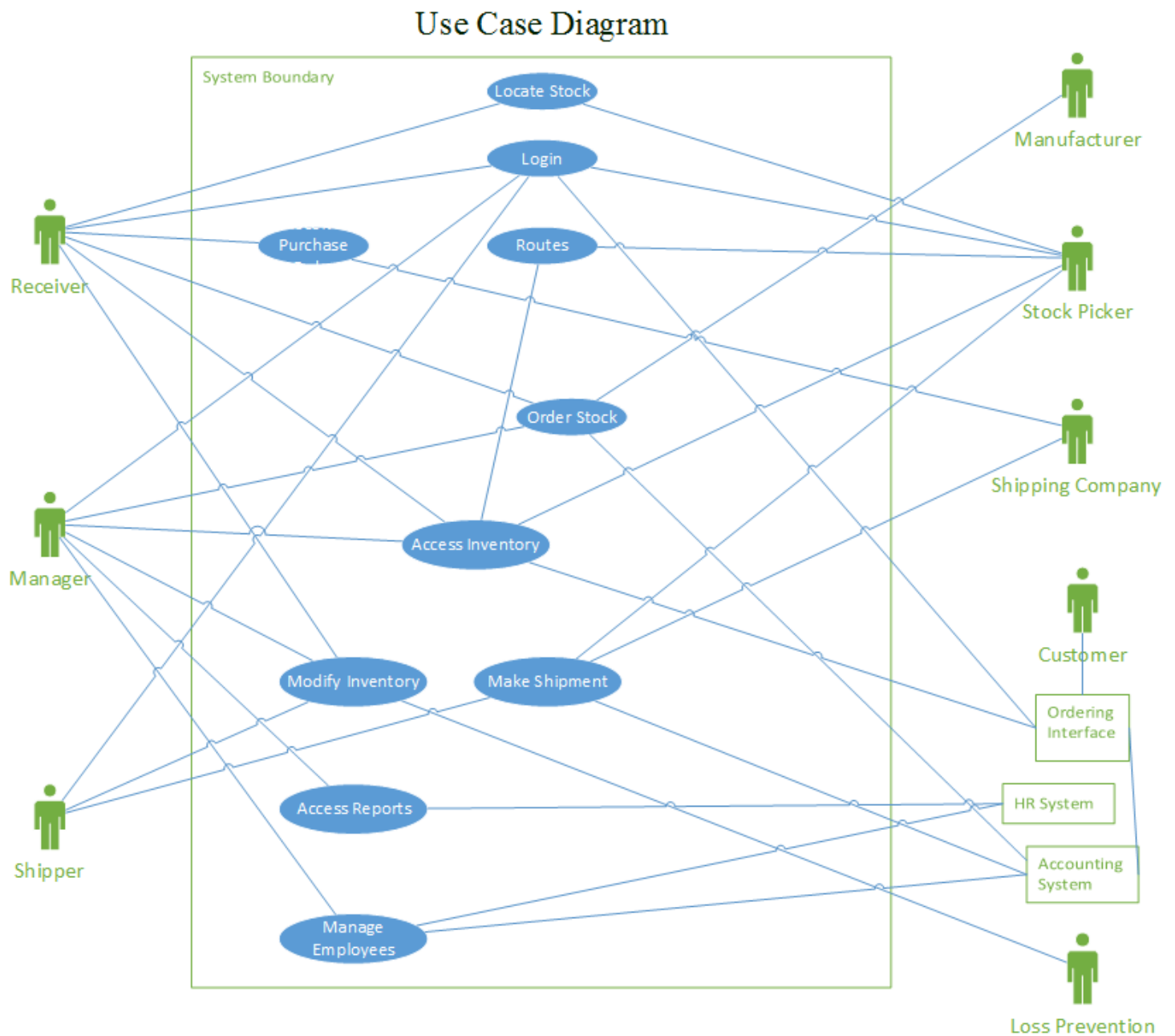
**Interfaces:**

- Noteworthy Hardware and Interfaces:
  - Touch screen monitor
  - Mouse
  - Keyboard
- Software Interfaces:
  - Graphical user interface
  - System-database interface
  - Be able to update/modify database through the warehouse system

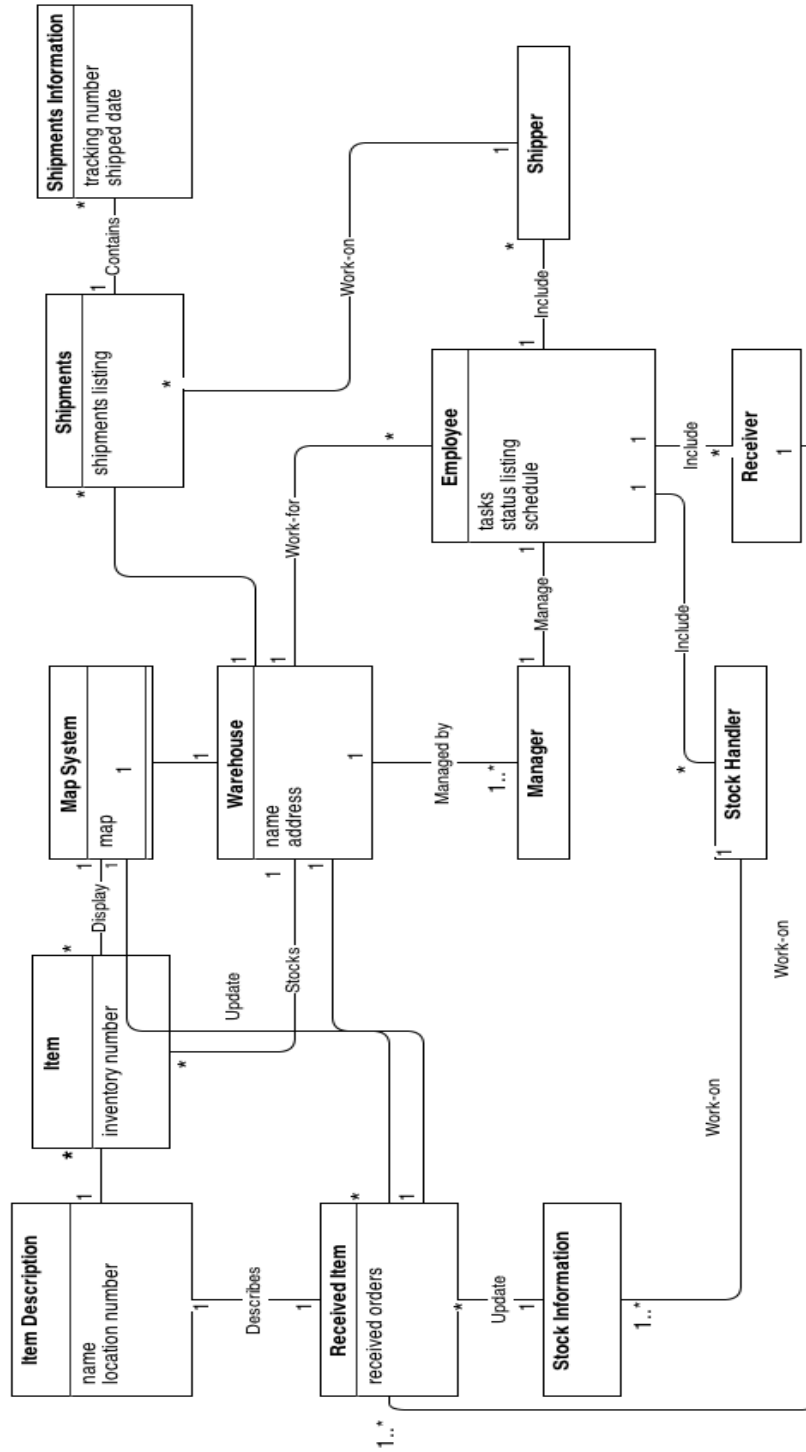
**Legal Issues:**

- Utilization of open source components such as the Neo4j database is allowed as it is a academic/non-commercial project
- Some products that may be stored in the warehouse have storage restrictions due to safety (i.e. flammable and explosive products should not be stored near to each other)

## 4. Use Case Diagram



## 5. Domain Model



## **6. Glossary**

Forklift – A machine with a “fork” on the front, can be used to lift heavy pallets high into the air

Pallet – A wooden platform for placing products on, commonly used in a warehouse to bundle products for shipment and moving the products using a forklift

Purchase Order – A pallet containing products that have been ordered by the warehouse

Shipment – A pallet that will be sent via truck to its destination

Shipper – An employee who plans and prepares shipments to be sent to the purchaser

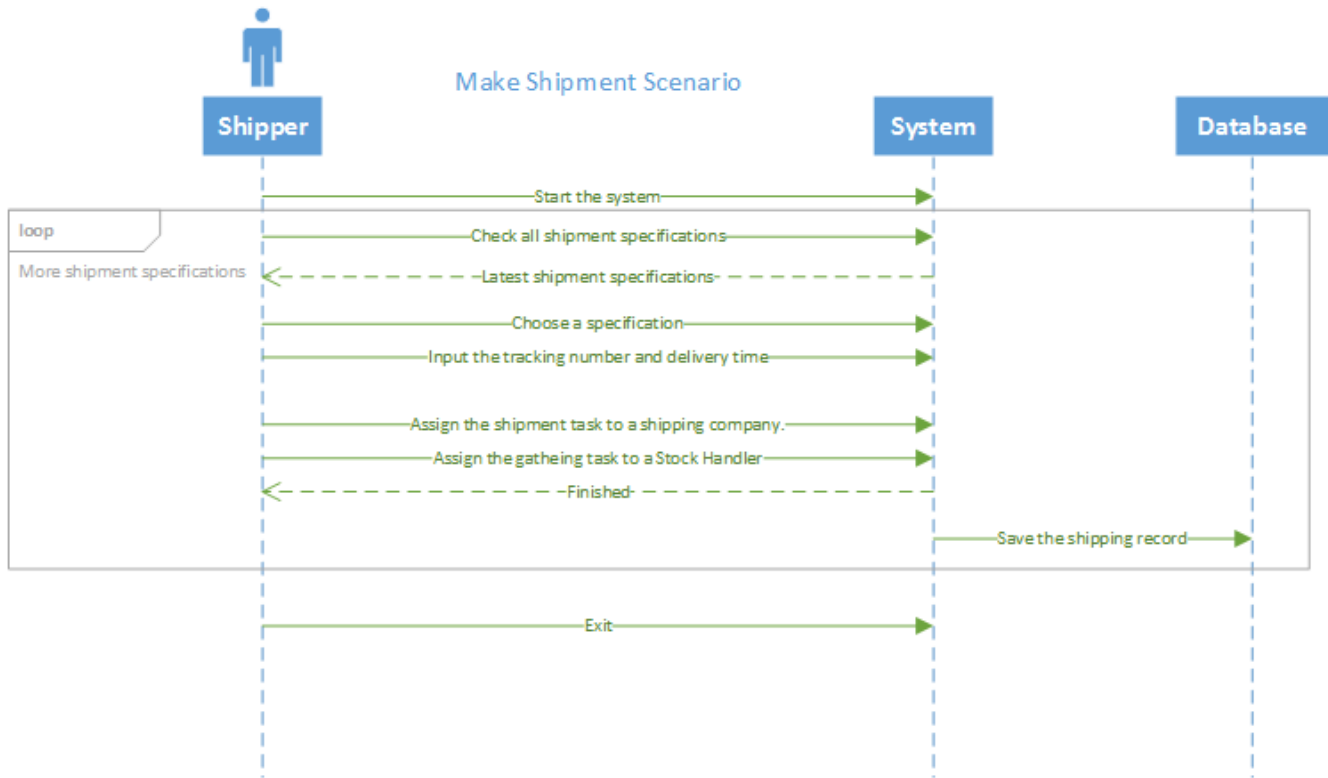
Stock – A product or item that can be bought by customers

Stock Handler – An employee, who puts stock items inside the warehouse when they have arrived for storage, and gathers stock items for shipment when they have been purchased

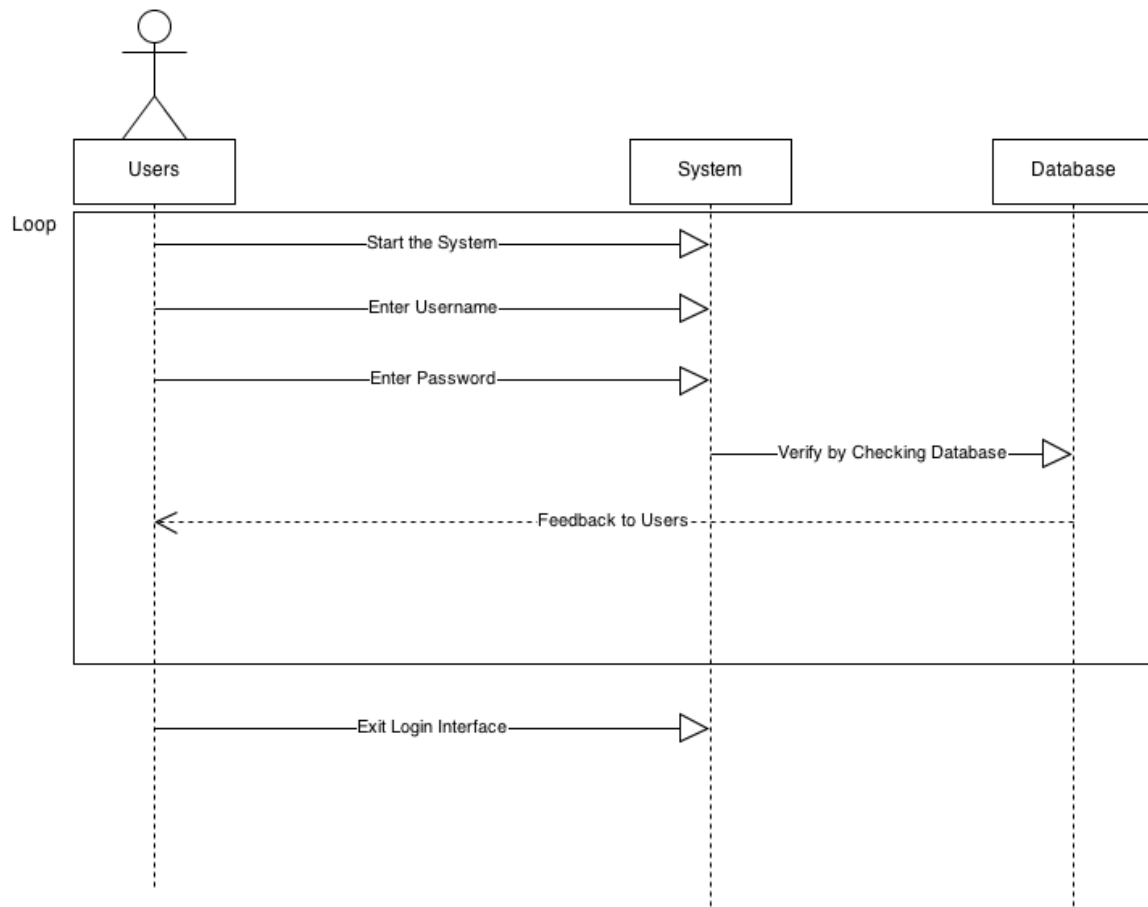
Receiver – An employee who records and tracks all incoming shipments to the warehouse

## 7. System Sequence Diagram

Make Shipment System Sequence Diagram:

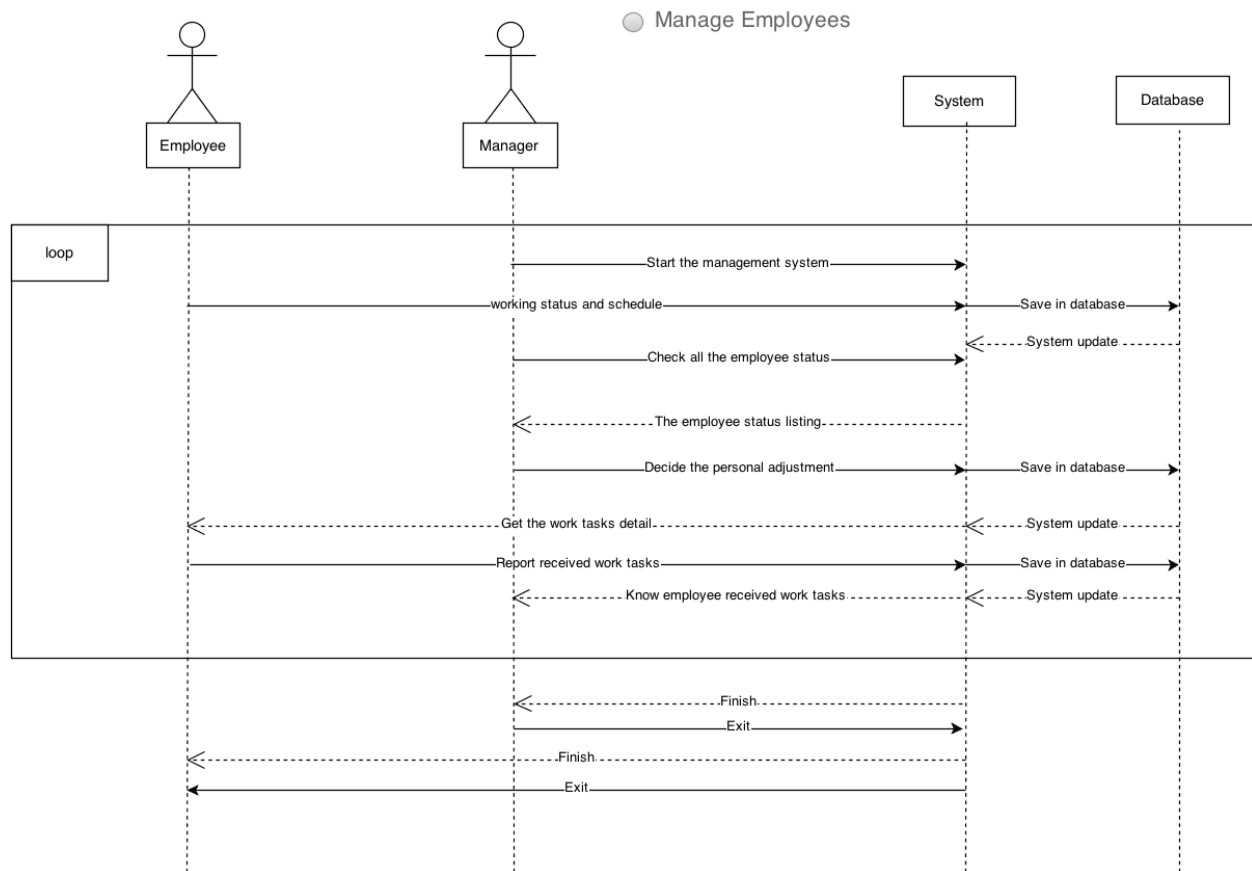


## Login System Sequence Diagram:

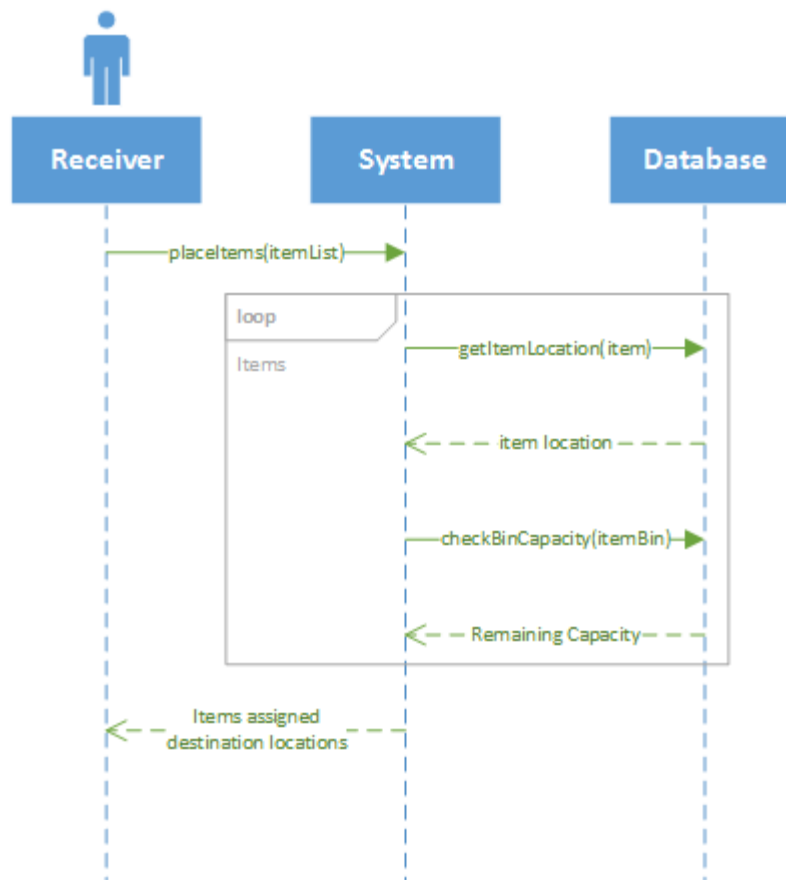




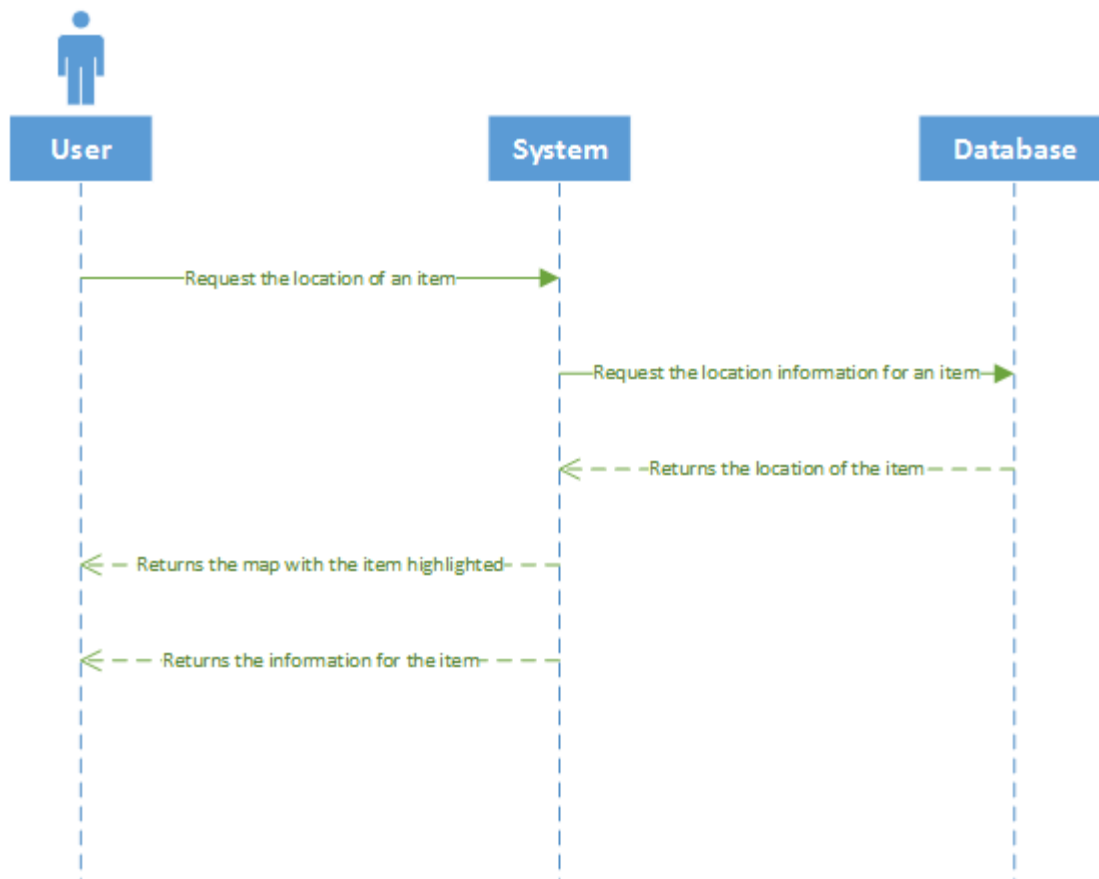
## Manage Employees System Sequence Diagram:



## Routing Stocking Tasks System Sequence Diagram:



### Locate Stock System Sequence Diagram:



## 8. Operation Contracts

**Operation:** checkShipRequire()

**Cross References:** uses cases: NIL

**Preconditions:** NIL

**Postconditions:**

\*All shipment requirements are listed in user interface.

**Operation:** assignTask()

**Cross References:** uses cases: check Shipment Requirements

**Preconditions:** At least one Stock Handler is available.

At least one shipping company is available.

**Postconditions:**

- All shipment requirements are listed in user interface.
- A shipment requirement S is chosen.
- Tracking number is assigned by user input.
- Delivery time is assigned by user input.
- S is distributed to a shipping company
- The gathering task is distributed to a Stock Handler.
- The shipping information is recorded.

**Operation:** onStartSystem()

**Cross References:** use cases: NIL

**Precondition:**

- System hardware works correctly
- User has launched the applications

**Postconditions:**

- System starts successfully
- System loads all data, loads GUI, and connects to databases.
- System is ready to login.

**Operation:** verifyUserInfo()

**Cross References:** use cases: Login

**Precondition:**

- Users already have been registered with the system
- Users already have the valid username entered
- Users already have the valid password entered

**Postconditions:**

- Users successfully login into the system if the username and password is correct. Else user is given feedback to ask them to double check their input.

**Operation:** checkEmployeeStatus()

**Cross References:** uses cases: NIL

**Precondition:**

- Employees have been registered in the System.
- Manager is logged onto the system.

**Postconditions:**

- Full report of all employees, title, ID, and task speed status.

**Operation:** assignPersonalAdjustment()

**Cross Reference:** use cases: Manage Employees

**Precondition:**

- Employees have been registered in the System.
- Manager is logged onto the system.

**Postcondition:**

- Employee title information has been modified and saved to the database.
- Employees that have quit or been fired are removed from the system.
- Employee task speed has been updated.

**Operations:** checkBinCapacity(itemBin: binID)

**Cross References:** use cases: Routing Stocking Tasks

**Preconditions:**

- itemBin instance b exists
- b exists within the set of bin ID's

**Postconditions:**

- ItemList instance iL was created
- Item instances i[] were created
- i associated with item instance j from database
- Float instance sum was created
- sum became i[].size

**Operations:** getItemLocation(item: productID)

**Cross References:** use cases: Routing Stocking Tasks

**Preconditions:**

- Item instance i exists

**Postconditions:**

- Integer instance n was created
- n was associated with Bin ID instance b from database

**Operations:** generatePallets (orderList[][]: item)

**Cross References:** use cases: Generate List

**Preconditions:**

- The information in the database is sorted into pallets
- The information is in a doubly linked list.

**Postconditions:**

- The list of items is generated in a printable list that can be displayed.

**Operations:** printList(orderList[][]: item)

**Cross References:** use cases: Generate List

**Preconditions:**

- The information in the database is sorted into pallets
- The information is in a doubly linked list.

**Postconditions:**

- The list is printed with all relevant information.

## 9. Implementation

### 9.1 Running Prototype

We have included running prototype with our submission for this milestone, to run the toy prototype, extract the included file “group02-runningPrototype.tar” and navigate to the extracted “dist” directory.

The prototype can be started by either double-clicking on the ‘Warehouse.jar’ file, or by using the command line to run ‘java -jar “Warehouse.jar”’.

Currently database output data is simulated, as such, no additional set up is required.

### 9.2 Implementation of Primary Success Scenarios

For this milestone, we have chosen to implement the primary success scenario for the use case Make Shipment:

#### **Make Shipment**

- This functionality can be accessed using “Shipping” button on the main interface
- Select a shipment from the list and click “Ship”
- From the top list, select a Stock Handler
- From the bottom list, select a Shipping Company
- Enter a Tracking Number
- Click “Confirm” to complete the task
- Or, at any time click “Cancel” to select another shipment

### 9.3 Skeleton Implementation

A skeleton implementation has been completed for all portions of the project. The .java files can be inspected by opening up the included “group02-runningPrototype.tar” file. Within each of the packages are the .java files that contain our skeleton implementation. The packages for each portion of the project are as follows:

database – Package containing database connection classes and interfaces for each type of user

manager – Package containing classes to manage employees and create reports

receiver – Package containing classes to perform receiving tasks

shipper – Package containing classes to perform shipping tasks

stockHandler – Package containing classes to perform stock handling tasks

warehouse – Package containing initial application start up file

warehouseui – Package containing GUI classes

### 9.4 User Manual/ReadMe File

Please find the ReadMe.txt file contained within the included file “group02-runningPrototype.tar”. The file summarizes the information included in the 9.1-9.3 sections above.

## 10. Project Plan

### 10.1 Completed Tasks

Date	Who	# of Hours	Task
9/16/2013	All	1.5	Meeting to decide on project and tasks
9/17/2013	Matt Triff	0.25	Set Up GitHub Repo, added everyone
9/18/2013	Xingze Guo	1.5	Design and write early proposal of function Shipper and Receiver
9/18/2013	Rongli Han	1.5	Write management part for early project proposal
9/18/2013	Spencer Ondrusek	1.5	Design and write early proposal of function Stock Picker
9/19/2013	Matt Triff	1.5	Combined and edited Milestone 1: Early Project Definition
9/24/2013	Spencer Ondrusek	1.5	Design the GUI for the stock picker
9/24/2013	Xingze Guo	1.5	Design the GUI for Shipper and Receiver
9/24/2013	All	1.5	Discussed the Project and divided sections again
9/24/2013	Rongli Han	1.5	Design the basic GUI for management and reporting
9/26/2013	Spencer Ondrusek	1.5	Milestone 3 part 1.4

9/26/2013	Spencer Ondrusek	1.5	Milestone 3 part 1.7
9/26/2013	Xingze Guo	2	Milestone 3 part 1.3, 1.6 for Shipper and Receiver and 1.7
9/27/2013	Matt Triff	2	Milestone 3, parts 1.1, 1.5
9/27/2013	Rongli Han	2	Milestone 3 part 1.2(Business Case), 1.4and 1.8 for management and reporting
9/27/2013	Xianming Luo (Sam)	2	Design the GUI for the prototype
9/28/2013	Matt Triff	5	Prepared Milestone 3 document to be submitted
9/28/2013	Xianming Luo (Sam)	3	Implement the GUI toy prototype
9/30/2013	Matt Triff	2	Completing final edits and submission of Milestone 3 document
10/4/2013	All	1	Prepare for Presentation
10/7/2013	Matt Triff	0.5	Update Tasks List for Milestone 4
10/8/2013	Matt Triff	4	Created database schema, created all tables in database, completed 2 use case summaries, set up branches and GitHub repository
10/11/2013	Rongli Han	1	Completed Summary Use Case for order stock, login and manage employee, Milestone4
10/11/2013	Rongli Han	1.5	Completed Fully Dressed Use Case for manage employee, Milestone 4
10/11/2013	Rongli Han	1	Completed Sequence Diagrams of manage employee, Milestone 4
10/11/2013	Xingze Guo	1.5	Completed fully dressed use case of make shipment
10/11/2013	Xingze Guo	0.5	Completed Operation Contracts of Make Shipment
10/11/2013	Xingze Guo	1	Completed Use Case Diagram
10/11/2013	Xingze Guo	0.5	Completed Sequence Diagrams of Make Shipment
10/16/2013	Matt Triff	1.3	Completed fully dressed use case, tallied status of milestone4 tasks and emailed team
10/17/2013	Rongli Han	0.5	Work on Operation Contracts for checkEmployeeStatus, Milestone 4
10/18/2013	Matt Triff	3	Completed skeleton for database package
10/18/2013	Spencer Ondrusek	2	completed fully dressed use case, operations and SSD
10/20/2013	Spencer Ondrusek	3	worked on implementing the print function and skeleton, communication on revising GUI
10/21/2013	All	0.5	Meeting to discuss status of Milestone 4
10/24/2013	Rongli Han	3	Created Skeleton for manage employee and



			manage product, Milestone 4
10/24/2013	Xingze Guo	2	Design skeleton for Shipper
10/24/2013	Xingze Guo	1	Design skeleton for Receiver
10/26/2013	Rongli Han	0.5	Work on Supplementary Specificaton Usability, Milestone 4
10/26/2013	Xingze Guo	3	Working on code of use case: Make shipment
10/26/2013	Rongli Han	0.5	Created Skeleton for Report, Milestone 4
10/26/2013	Spencer Ondrusek	0.5	Supplementary case - Supportability
10/26/2013	Xianming Luo (Sam)	2	Worked on Domain Model Diagram

## 10.2 Upcoming Tasks

Who	Task	Estimated Hours	Due Date
Matt	Implement JDBC Connection	1	11/04/2013
Matt	Fill MySQL database with dummy data	2	11/04/2013
Matt	Implement and setup Neo4j database	3	11/08/2013
Spencer/Matt	Fully implement Routing Stocking Tasks	6	11/22/2013
Han	Fully implement Manage Employees	8	11/22/2013
Sam	Fully implement GUI map	15	11/25/2013
Han/Xingze	Fully implement order stock and create shipments	8	11/15/2013
Spencer/Sam	Fully implement Locate Stock	10	11/29/2013
Xingze	Fully implement receive purchase order	10	11/8/2013
Matt/Sam	Fully implement login	5	11/15/2013
Han	Fully implement modify stock	2	11/08/2013
Han/Matt/Spencer	Fully implement employee and product reports	7	11/25/2013

## 11. Meetings

Date/Time	Topic	Members Present	Summary
9/16/2013 4pm-5pm	Decide on Project, Assign Work Units	All (xig480, roh919, xil30, spo798, mdt879)	<ul style="list-style-type: none"> <li>Decided on Warehouse Management System</li> <li>Decided on Everyone's Tasks <ul style="list-style-type: none"> <li>Han - Management System</li> <li>Xingze - Shipper/Receiver</li> <li>Sam - GUI</li> <li>Spencer - Stock Picker</li> <li>Matt - Database/Back end</li> </ul> </li> </ul> Decided to use GitHub for version control
9/24/2013 5pm-5:45pm	Decide on GUI for Toy Prototype Assign Tasks for Milestone 3	All (xig480, roh919, xil30, spo798, mdt879)	<ul style="list-style-type: none"> <li>Decided on general GUI for toy prototype</li> <li>Decided who will do what for Milestone 3 <ul style="list-style-type: none"> <li>Han - 1.2</li> <li>Xingze - 1.3</li> <li>Spencer - 1.4, 1.7</li> <li>Sam - 1.11</li> <li>Matt - 1.1, 1.5</li> <li>Everyone - 1.6, 1.8</li> </ul> </li> </ul>
10/7/2013 10:30am-11:20am	Divide Tasks for Milestone 4	All (xig480, roh919, xil30, spo798, mdt879)	<ul style="list-style-type: none"> <li>Read and divided tasks for Milestone 4 <ul style="list-style-type: none"> <li>Everyone: 1 Fully Dressed, 1 SSD, 2 OC, 1 Skeleton</li> <li>Han: 3 Summaries</li> <li>Matt: 2 Summaries, Setup up DB and Ant</li> <li>Spencer: 1 Full Implementation</li> <li>Xingze: Use case diagram</li> <li>Sam: Domain model</li> </ul> </li> </ul>
10/22/2013 4:50pm-5:20pm	Status Update on Milestone 4	All (xig480, roh919, xil30, spo798, mdt879)	<ul style="list-style-type: none"> <li>Reviewed status of everyone's tasks</li> <li>Assigned each person 1 Summary Specification</li> </ul>
10/25/2013 10:30am-11:20am	Clarify Remaining Tasks for Milestone 4	Xingze, Sam, Spencer, Matt (xig480, xil30, spo798, mdt879)	<ul style="list-style-type: none"> <li>Discussed any remaining</li> <li>Clarified ETA for completion</li> <li>Discussed GUI changes required for fully implemented use cases</li> <li>Clarified Milestone requirements</li> </ul>