Report-1 Overview

The main focus of Report-1 is to generate the system specification of your software based solution to the customer's state of requirements. This report shall be submitted in three steps:

1. CSR

Problem Statement

The working environment of a warehouse or a large store is complex and arduous. Many products exist in a large warehouse, such as that of Amazon or Walmart, and it can be difficult to locate and ship items in an efficient and timely manner. We would like to request a software that is: easily accessible, easy to use, and responsive. We require a product that will allow us to reduce the amount of time and effort it takes to perform our jobs adequately. Not only do we require a application to perform these tasks but also make it user friendly.

We would like to see a software that can give us a competitive edge and be cheap to use and maintain. This application must be easily accessible on all devices. We would also like to be able to access this application through a web browser. This would make it easier for us to gain access to this application over various ways so we would not be dependant of specific hardware. Especially for a smaller business like us, we can may not necessarily be able to provide all employees specific hardware for this software we want. That is why I would like to have as many options as possible for the application to run on.

This application must be well constructed and easy to interpret at first glance. This would include buttons and layout being thoughtfully designed so we can clearly understand where all the features are. We request a software that is friendly for newer employees so we can minimize training and specialization for this tool. The interface must be simple and smart, buttons being easy to click and items easy to search. The interface must also have a graphical reaction to each action that is done so we know what is interactable. The interface should also contain a map of the warehouse or store so employees/customers know exactly where to look when requesting an

items location. The map should be based on some sort of grid so it is easy for us to interpret. We do not want anything overly intricate, just something that provides a basic scale to represent our warehouse's shelves. We would also like some sort of key for the map so we can easily understand what each part of the map can represent. We would also like to request a way for employees to comment about the map so managers will be able to fix any problem that employees can run into. These problems could include items being stocked in the wrong area, damaged goods, broken tools, warnings, or notes for the department. These comments should have features such as being anonymous comments and who can see these comments. That way some employees can leave comments for themselves and for other employees or managerial staff. We would like an application with a decisive interface that will make it easy for our employees to understand the application and the features it can provide.

The application must have all the basic functions that most warehouse management software should provide. These include an item catalog, item search, and item tracking. We would also like to request a possible way to search items and be able to pull information like quantity, location and serial number. The search feature of the application must include every item that is housed in the warehouse and the quantity of the item. There must be a feature that removes the item when it has been taken out of the warehouse, and a feature that can allow the user to add more stock to the item list. The application should also be able to warn employees when an item is low in stock. This warning should be customizable based on a numerical amount of that item. For example we could receive a warning about item A when we have 5 of it left in the warehouse and get another warning for item B when we have 20 of it left in the warehouse. A comprehensive list of items must be easy to access and amend. After we search for an object we can have the application map the requested item on the interface and provide not only a written but a visual location of the item. This feature must not only be able to map out single items but be able to map out multiple items. A map of the warehouse must be easy to access after a search of an item(s) is done, and must give the quickest route to the item(s). With all these features not only should the application be an effective tool, but should be easily accessible for employees. In short we require an application that can manage items in our warehouse, map out items for employees to find, and be easy to understand.

Glossary of Terms

Database- a structured set of data held in a computer, especially one that is accessible in various ways.

Browser - a program with a graphical user interface for displaying HTML files, used to navigate the World Wide Web.

Application- computer software designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user.

Interface- a device or program for connecting two items of hardware or software so that they can be operated jointly or communicate with each other.

GUI - is a user interface that includes graphical elements, such as windows, icons and buttons.

2. Sys. Requirements

Functional Requirements

Label	Weight	Description
1. REQ-Search	1	The ability to search for and find items and their quantity
2. REQ-Map	2	A map and simple directions to where the item is located
3. REQ-List	3	A comprehensive list of items and the quantity
4. REQ-Add	4	The ability to add items and their quantity to the stock
5. REQ-Remove	5	The ability to remove items from stock and auto subtract quantity

Nonfunctional Requirements

6. The application must be able to handle a large amount of items to be searched through.
7. The search function must be quick and efficient with no errors, likely utilizing binary search.
8. The application must have a button to quickly return to the initial page to search for further items.
9. The database and server holding the list of items and the source of the pages must not be reachable from any of the application pages.
10. Every page must have the same aesthetically pleasing look with simple colors and easily readable text from any device.
11. The application must correctly list the item, its quantity, and where the item is located without any error.
12. The application must have a consistently reliable and easy to read interface between multiple platforms such as computers, phones, tablets, etc.

On Screen Appearance Requirements

13. Easily clickable buttons with easy to read text
14. Easy to use search function that quickly finds all items
15. Interface that reacts to actions to show that the application is responsive

Stakeholders

The groups most interested in this software should be any smaller business that is looking for any organizational tool to help them start up warehouse management. Other groups that should be interested in this device could be retail stores, warehouses, and any organization in need of item management. So more specific examples could include a used-book store or any general online store that ships products from warehouses. The users of this device should include business managers, warehouse workers, business owners, their employees and their customers.

Actors and Goals

We plan on splitting the users into four roles. The four groups are Top-Manager, Manager, Employee and Customer. Each group will have an individual goal and level based access to certain portions of the application. The Top-Manager can create and delete any user accounts for the application. They can override any other user's actions and have full-access administration rights. They also have all the abilities that any lower tier user would have. Managers have the ability to edit the store layout, map out each individual item in the store. Managers can also add or remove employees from the software. They can also add products and respond to emergency request. Employees have the ability to map out each individual product and find the best algorithm to pick up each item. They also are able to send request to the manager to be approved. These request can be things like items being added or removed or change of location for an item. Each higher role manages not only the warehouse but the users below them so they can have more of a controlled environment when using the application. The Employee role is independent from the three previous roles as they have the ability to order items and send request to the warehouse describing what items they would like to order. All of these roles have the ability to use this application on a mobile device such as a smartphone or they can use the application on a desktop computer as long as they have a compatible browser for the device.

Use Cases

3A. **Casual Description**

For Adding and Removing Items this use case is responds when the customer interacts with ordering items. The customer has the ability to add and remove items from there shopping queue. This use case is similar to most online shopping as it gives the user the ability to order items directly from the warehouse. The managers can also use this process when adding or removing items from the warehouse. Collecting Items is a use case that happens when a customer adds or removes items from a list. The list is a collection of items that relate to the

items in the warehouse. This list can then be processed and sent to the warehouse so the warehouse workers can process the order. For Manager's Layout the management level user has the ability to map out the warehouse in a scale map. The manager-level user can then have the ability to edit the map and designate where each individual item it located with the store. Updating information can then be done when the user logs into the app and changes their personal information. This can be down when the user send a change request that relates to information such as address, ordering information, or other various personal information.

Use Case 1 - Adding / Removing Item	
Initiating Actor	Customer
Actors Goal	To add or remove items from their shopping list from their handheld device.
Pre-Conditions	Actor has logged into the app and the search GUI is showing
Post-Conditions	Once the user has selected all the items they wish to purchase there will be their list and a subtotal at the bottom of the page.
Flow Of Events	1) User logs into app. 2) User selects the menu tab. 3) User selects the search tab. 4) User searches for items and adds them to the list (repeat this step for as many items as the user wants). 5) If the user wishes to remove an item they can search it and press remove from list. 6) User presses the finished button and a finalized list with its subtotal populates the page.

Use Case 2 - Collecting Items	
Initiating Actor	Customer
Actors Goals	To follow set path and retrieve items on list
Pre-Conditions	User has added items to a list
Post-Conditions	User will have acquired desired items on their list
Flow Of Events	1) Once user has created their list, they press the go button on the bottom 2) The map will tell the user where to go from the entrance to their first item. 3) Once user has retrieved their item, they will press the “next item” button. 4) The map will then show the user how to get to the next item from their last items location. 5) User will repeat step four until all desired items have been acquired.

Use Case 3 - Managers Layout	
Initiating Actor	Manager
Actors Goals	Create and populate a map, or change existing map.
Pre-Conditions	Manager has logged in as a manager
Post-Conditions	The store map will be complete with shelf locations and items that populate the shelves
Flow Of Events	1) Manager drags and drops shelf locations from a toolbox displayed in the GUI 2) If there was a previously saved store layout the page will be populated with the appropriate layout and the manager can continue to execute the remaining steps 3) Once the manager has set up the store layout, they may add items onto a shelf selecting that shelf. They may also add extra information during this step, such as price and quantity. 4) After the manager has completely set up the store, they will click save and the information about the layout will be sent to a database.

Use Case 4 - User Updates information	
Initiating Actor	Customer
Actors Goals	Update accounts details.
Pre-Conditions	User has made an account, and has logged in.
Post-Conditions	Users details will be changed to what they desire.
Flow Of Events	1) User selects the drop down menu button. 2) User selects gear box at the bottom of the menu drop down. 3) Now a page will be populated with the current information of the customer. 4) User can now update information such as birthday. 5) Once user has made the appropriate changes, they can click save and the updated information. 6) Updated information is sent to database.

3C. **Traceability Matrix**

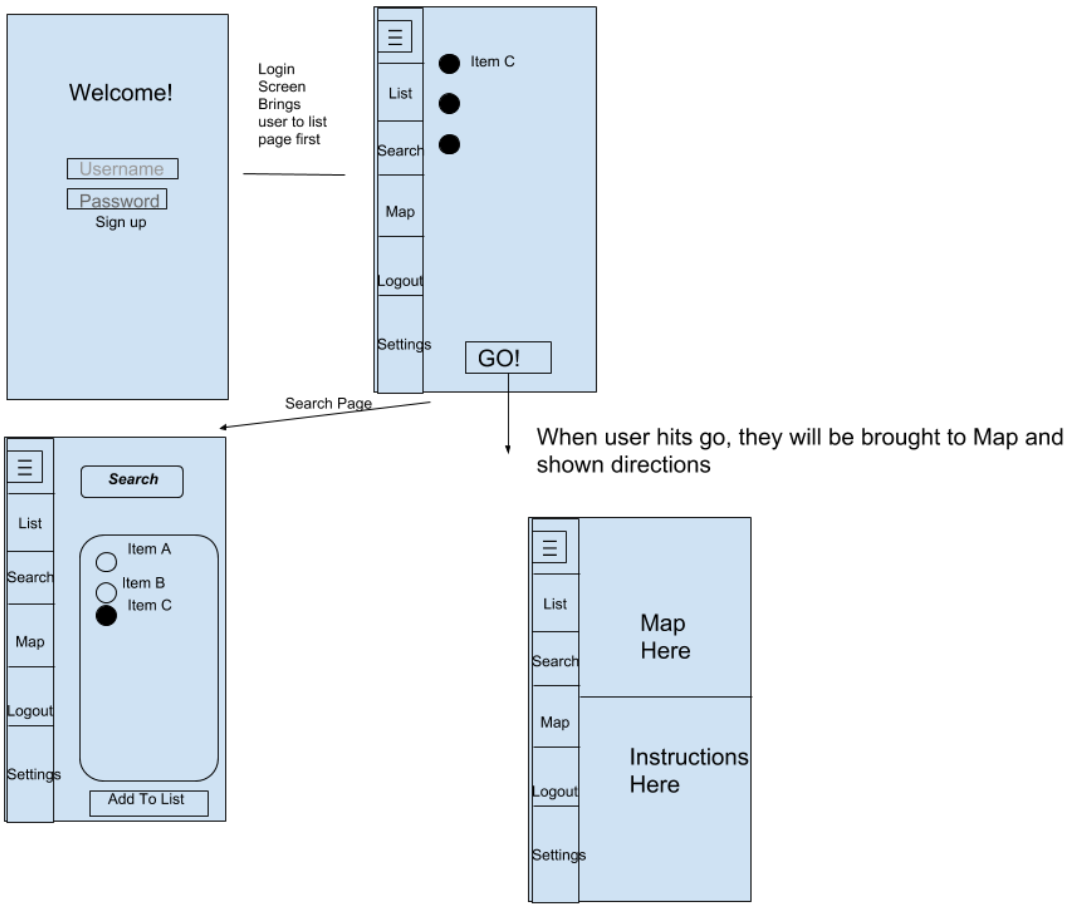
Our most important Use Case will be Adding and Removing Items. We have decided this portion is the foundation of our application and should be the first Use Case to be constructed and demonstrated. This is the most important use case as it will be the first use case that should not only be one of the primary functions of our application but also be the basis for communication among different users, the database and the application. This Use Case maps into the system requirements as well because this use case will demonstrate how the Actor's actions

here use our add, search, and, remove functions. This Use case is also needed to map our system requirement for our list function as well. This Use Case will also demonstrate the application's ability to be demonstrated on mobile and regular website.

3D. **Fully Dressed Description**

Since we only have four Use Cases we thought we should give spend most of our time describing the most important Use Case - Adding and Removing Items. This function's Actor is supposed to be the customer but its function can also be implemented by employees or managers as well. The goals of the this particular Use Case is, "To add or remove items from their shopping list from their handheld device." This means the customer has the ability to add and remove items from their shopping queue when selecting items that are located within the warehouse. This Use Case will not only add and remove items but when the customer finalized the order the request will then be sent to the corresponding warehouse so the employees of the warehouse house can process the order. This means the employees will also have to add and remove items from the warehouse in order to keep track of the numbers of items in the warehouse. The preconditions of this Use Case are that the application's GUI has to be showing once the Actor has logged in. Once the Actor has finished the process of the Use Case the postcondition will then purchase the items and there will be their list and a subtotal at the bottom of the page. The information will then be sent to the warehouse for processing the order. This Use Case is needed before all others as it maps out the necessary system requirements and the list function cannot be done unless we have the add function first.

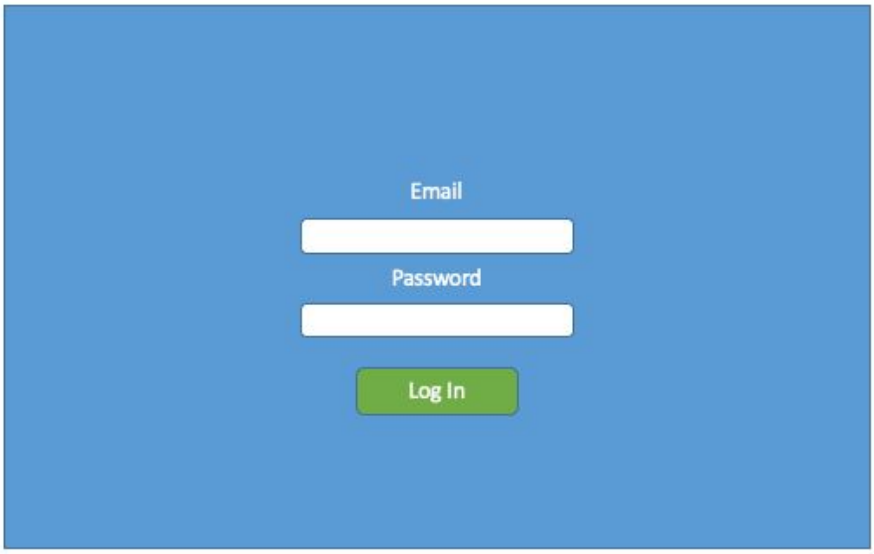
4A. **Preliminary Design**



When a user logs in, they will immediately be brought to their list page. From there, the user should navigate to the search page to allow them to add items to their list. From there, they would hit “Go” which brings them to the map page giving them directions to their items throughout the store in the most efficient manner.

This setup allows the user to change their list as they shop if they want simply by hitting the menu button and going to their list.

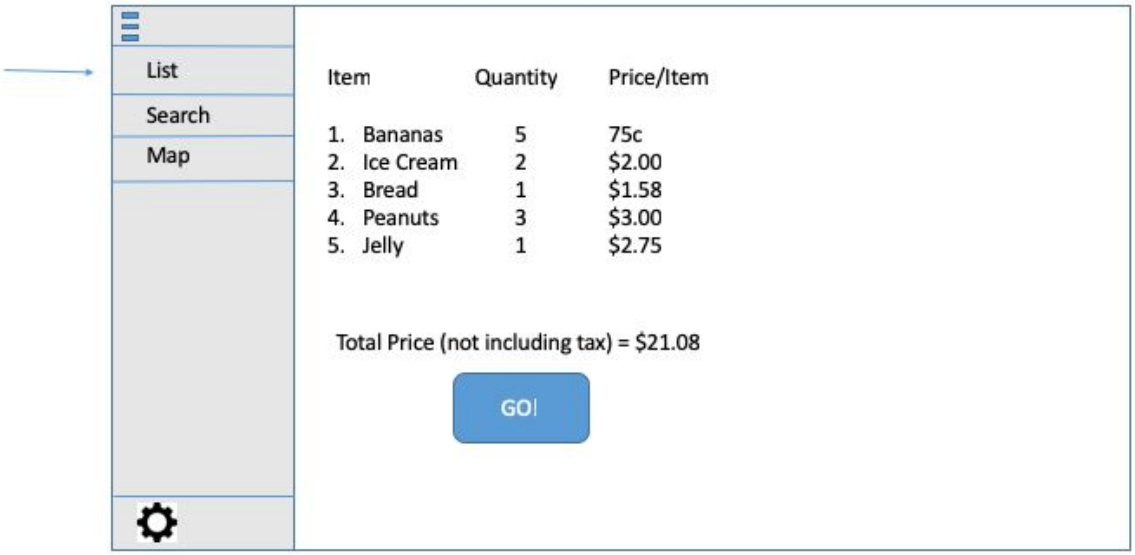
Index Page



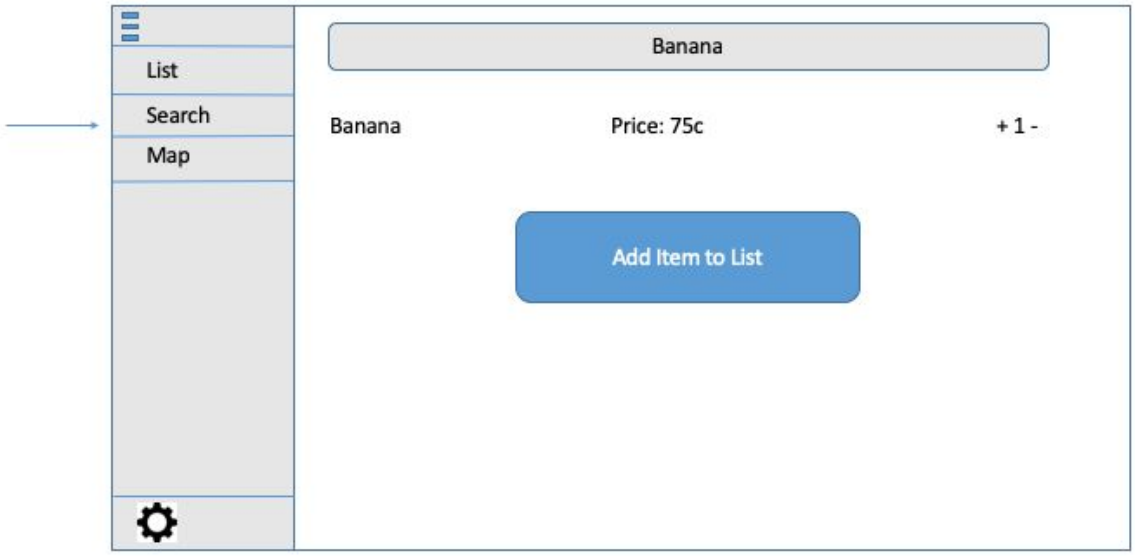
Home Page

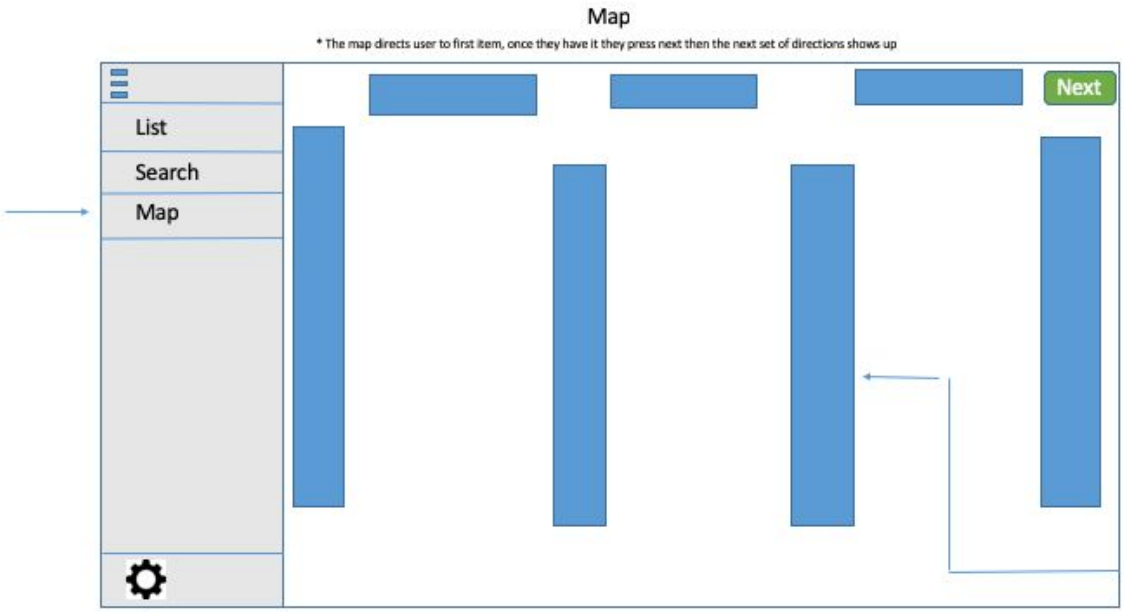
List Page

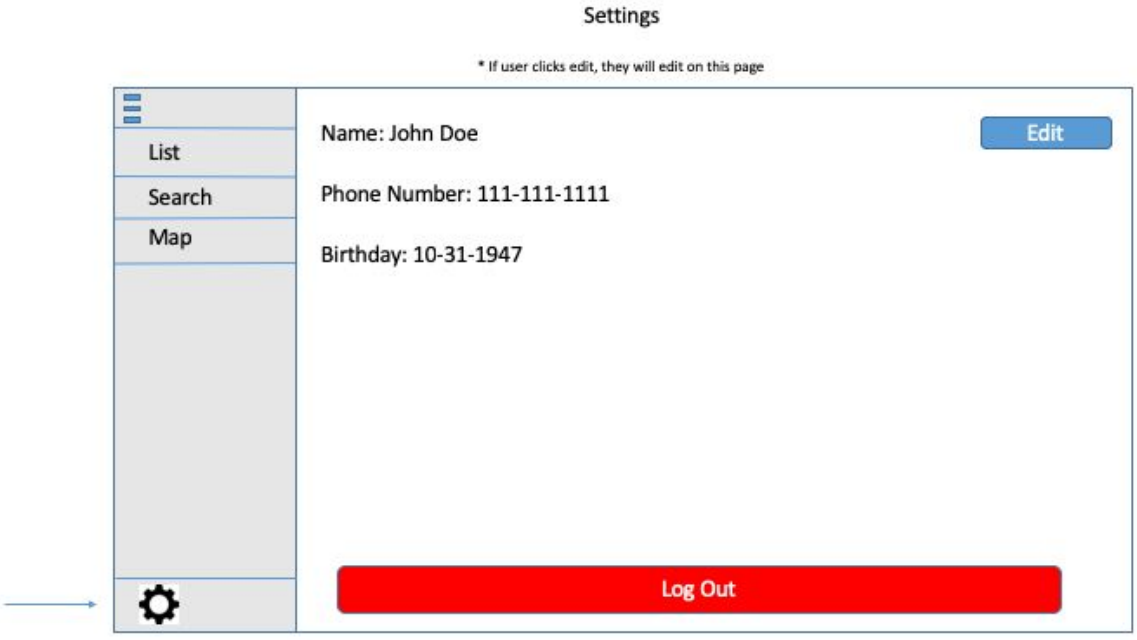
* This is an example, if there are no items the page will be blank. Go! Button takes user to map



Search Page







4B. User Effort Estimation

For a typical user experience the amount of clicks is minimal. A user will login with their login information (Clerical). From there they will hit login (nav) and be brought to their list. From then they should select search (nav) and then enter in the items they want in the search bar

(clerical). In order to add an item to their list, they select the bullet point that they want to add, then hit add to list (2 navigational clicks, n+2 when adding multiple items). Once completed, they click back to their list (1 navigational, and then hit go another navigational). In order to get the application functioning from login to hitting go, the user will enter clerical information twice at login and however many times corresponding to the amount of items in their list. As far as navigational clicks, the user will click roughly 4-5 times but this does depend on the amount of items in their list.

The other time a user will have to enter clerical data in when they sign up. This is all very basic information name, address, age, and gender.

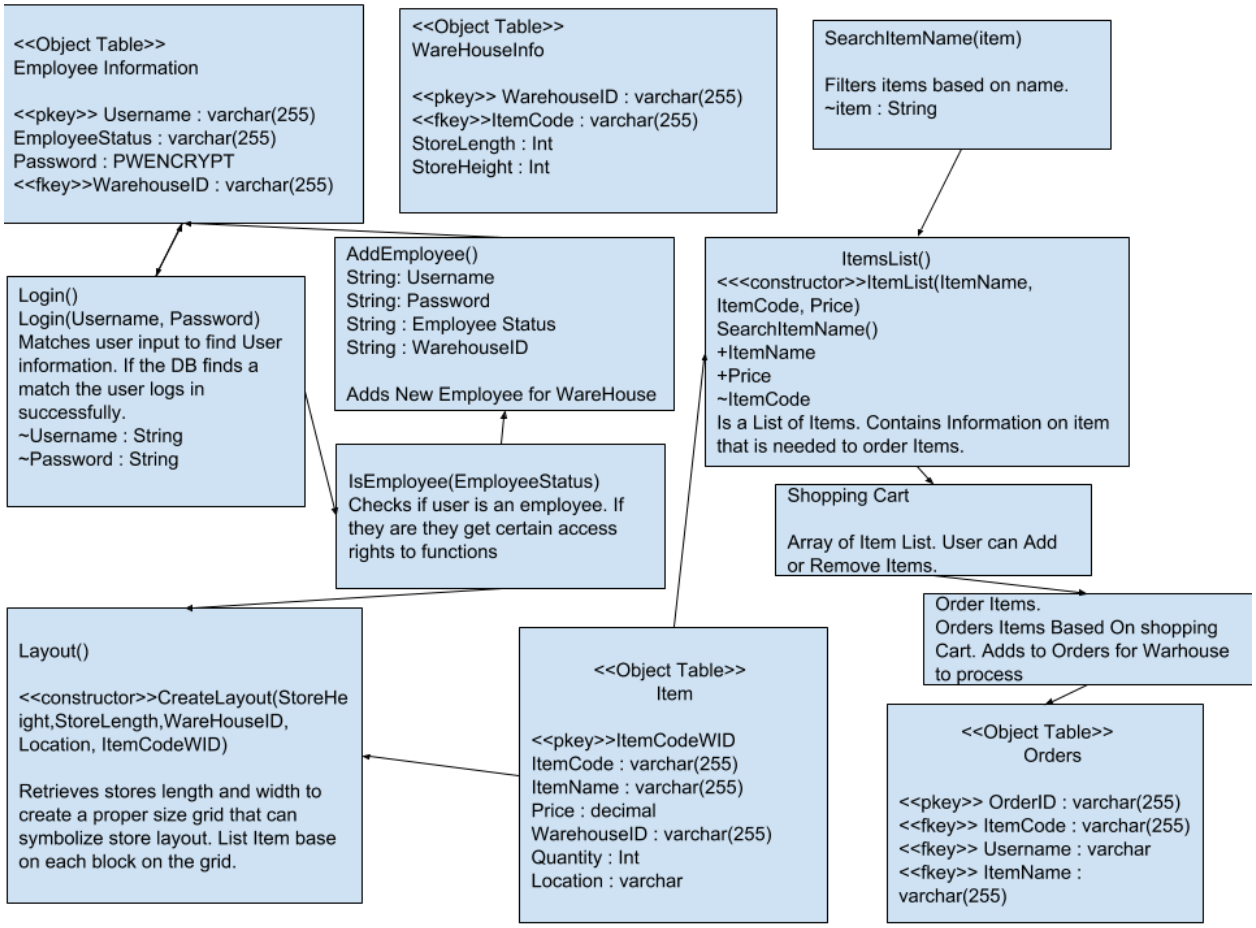
References

- "GUI." *Tech Terms*, techterms.com/definition/gui.
- "Web Browser." *Tech Terms*, techterms.com/definition/web_browser.
- "Application." *Tech Terms*, techterms.com/definition/application.
- "Interface." *Tech Terms*, techterms.com/definition/interface.
- "Database." *Tech Terms*, techterms.com/definition/database.

5.Domain Analysis

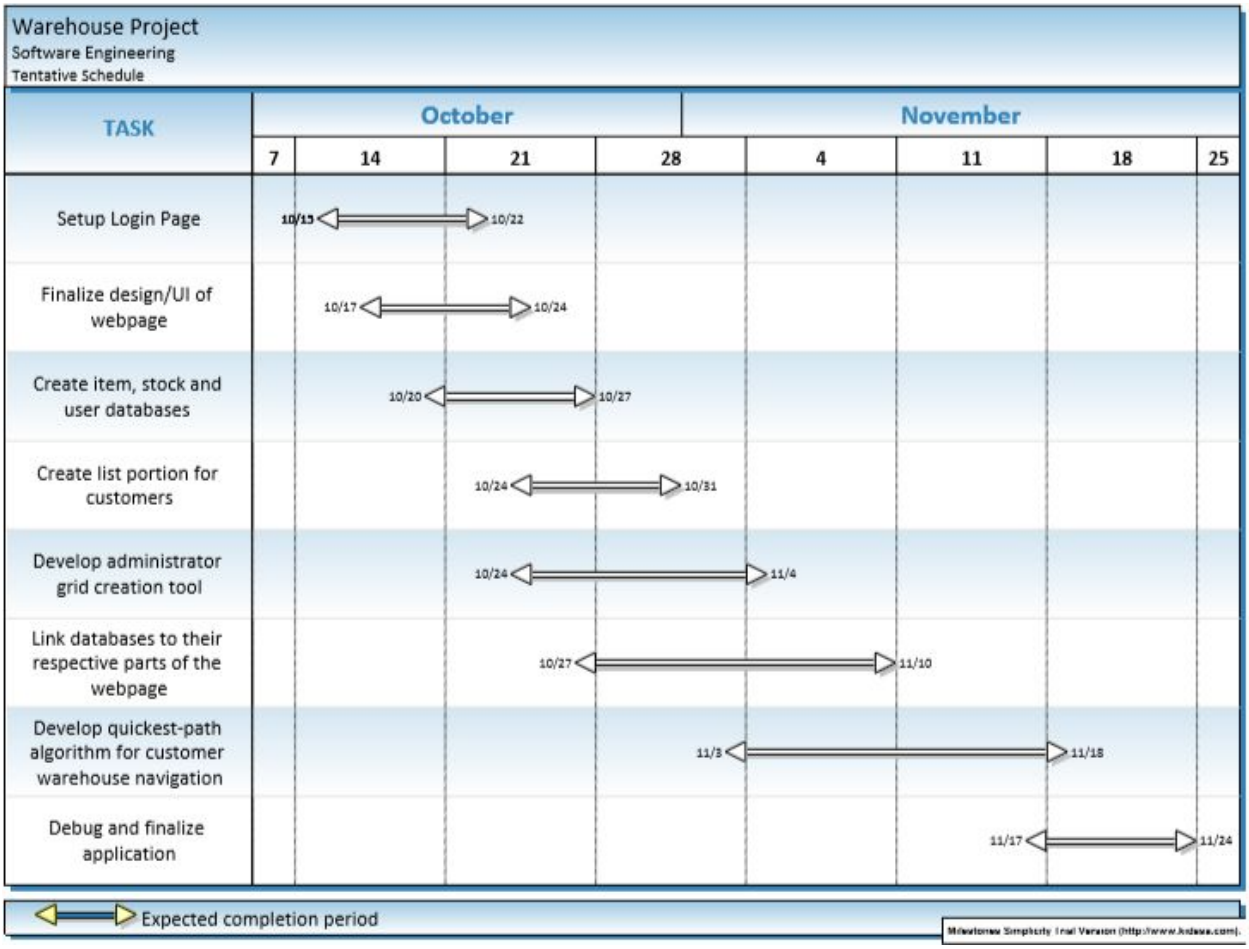
5A Domain Model

UML Diagram



6. Plan of Work

By the first demonstration we plan to have basic functionality finished. The UI should be usable, though not completely finished, and some of the basic tasks like logging in and creating the list should be usable. After the first demo, the rest of the functionality will be worked on. The UI should be completed and polished while the map and other functionality should be usable. The database will be integrated in this time to give functionality to the list and the map. Much of the UI work will consist of making sure all of the components work together smoothly. By the end of the semester all functionality will be available and work smoothly with all other components of the project.


Warehouse Project
Software Engineering
Tentative Schedule
TASK
October
November
7
14
21
28
4
11
18
25
Setup Login Page
10/13
10/22
Finalize design/UI of webpage
10/17
10/24
Create item, stock and user databases
10/20
10/27
Create list portion for customers
10/24
10/31
Develop administrator grid creation tool
10/24
11/4
Link databases to their respective parts of the webpage
10/27
11/10
Develop quickest-path algorithm for customer warehouse navigation
11/3
11/18
Debug and finalize application
11/17
11/24
Expected completion period