

CSCI222 – System development

Assignment 1 – warehouse management tool

Version 0.9



|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID** | **Email** |
| Chan Xiaoru Kalista | 5986345 | xkchan002@mymail.sim.edu.sg |
| Tam He Sheng (Allen) | 5986485 | hstam001@mymail.sim.edu.sg |
| Teo Heong Hwee (Gavin) | 6098058 | hhteo005@mymail.sim.edu.sg |
| Jonas Chok Wen Jie | 6098459 | jwjchok001@mymail.sim.edu.sg |
| Kyaw Myo Aung, Johns | 6097868 | Myoak001@mymail.sim.edu.sg |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Effective Date | Summary of Change | Prepared by |
| 0.1 | 17/01/2018 | Initial Draft | Kalista Chan |
| 0.2 | 21/01/2018 | Updated:   * Functional Specs and Non-Functional Specs * Risk and Counter Measures * Format entire document * Added Roles and Responsibilities | Kalista Chan |
| 0.3 | 22/01/2018 | Updated:   * Business Case * Project Schedule v2 | Johns |
| 0.4 | 24/01/2018 | Updated:   * Business Case * Role Matrix | Johns |
| 0.5 | 28/01/2018 | Updated:   * Use Case Diagram * Detailed Use Case * Sequence Diagram | Teo Heong Hwee (Gavin) |
| 0.6 | 28/01/2018 | Updated:   * Class Diagram * Compile and reformatted entire report | Kalista Chan |
| 0.7 | 28/01/2018 | Updated:   * Compile and reformatted entire report | Teo Heong Hwee (Gavin) |
| 0.8 | 28/01/2018 | Updated:   * Data Dictionary | Allen He |
| 0.9 | 29/01/2018 | Updated:   * Functional Specs and added Class Diagram iteration 1 | Kalista Chan |

Table of Contents

[1 Introduction 3](#_Toc504989701)

[1.1 Purpose 3](#_Toc504989702)

[1.2 Business Case 3](#_Toc504989703)

[1.3 Definitions, Acronyms and Abbreviations 4](#_Toc504989704)

[2 Roles and Responsibilities 5](#_Toc504989705)

[2.1 Responsibility Matrix 5](#_Toc504989706)

[2.2 Role Allocation 6](#_Toc504989707)

[3 Implementation Plan 7](#_Toc504989708)

[3.1 Project Schedule 7](#_Toc504989709)

[4 Overview of System 8](#_Toc504989710)

[4.1 Software Requirements Specification 8](#_Toc504989711)

[4.1.1 Functional Specifications 8](#_Toc504989712)

[4.1.2 Non-Functional Specifications 12](#_Toc504989713)

[4.2 Use Case Diagram 13](#_Toc504989714)

[4.2.1 Iteration 1 13](#_Toc504989715)

[4.2.2 Iteration 2 14](#_Toc504989716)

[4.2.3 Final Version 15](#_Toc504989717)

[4.3 Use Case Detailed Description 16](#_Toc504989718)

[4.3.1 Detailed Use Case – Login 16](#_Toc504989719)

[4.3.2 Detailed Use Case – Create Summary Report 17](#_Toc504989720)

[4.3.3 Detailed Use Case – View Summary Report 17](#_Toc504989721)

[4.3.4 Detailed Use Case – Add Stock 18](#_Toc504989722)

[4.3.5 Detailed Use Case – Remove Stock 19](#_Toc504989723)

[4.3.6 Detailed Use Case – Edit Stock 20](#_Toc504989724)

[4.3.7 Detailed Use Case – Display Stock 21](#_Toc504989725)

[4.3.8 Detailed Use Case – Search Stock 22](#_Toc504989726)

[4.3.9 Detailed Use Case – Record Stock 23](#_Toc504989727)

[4.4 Sequence Diagram 24](#_Toc504989728)

[4.4.1 Sequence Diagram – Login 24](#_Toc504989729)

[4.4.2 Sequence Diagram – Create Summary Report 25](#_Toc504989730)

[4.4.3 Sequence Diagram – View Summary Report 26](#_Toc504989731)

[4.4.4 Sequence Diagram – Add Stock 27](#_Toc504989732)

[4.4.5 Sequence Diagram – Remove Stock 28](#_Toc504989733)

[4.4.6 Sequence Diagram – Edit Stock 29](#_Toc504989734)

[4.4.7 Sequence Diagram – Display Stock 30](#_Toc504989735)

[4.4.8 Sequence Diagram – Search Stock 31](#_Toc504989736)

[4.4.9 Sequence Diagram – Record Stock 32](#_Toc504989737)

[4.5 Class Diagram 33](#_Toc504989738)

[4.5.1 Iteration 1 33](#_Toc504989739)

[4.5.1 Iteration 2 34](#_Toc504989740)

[4.5.2 Final Version 35](#_Toc504989741)

[4.6 Data Dictionary 36](#_Toc504989742)

[5 Risk and Counter Measures 38](#_Toc504989743)

[6 Meeting Minutes 40](#_Toc504989744)

# Introduction

## Purpose

This document describes the business case, such as the system that we will be building, whom will it serve and the benefits that they system will bring. It will document the detailed for the whole project, from requirements until implementation, risks and counter measures. It will also include the software requirements specification, use cases as well as a domain model to provide a comprehensive description of the requirements for the proposed system.

## Business Case

The WMS enable the clients to prevent shortage, production delay and tracking inventory in real time basic. It’s the automated operation for incoming, outgoing and in-stock items for your business operation.

1. **Prevent Shortages**: It’s critical to understand the customer needs when you don’t have enough for supply. To avoid the costly delay in production and customer order fulfilment, WMS can help to improve those factors.
2. **Tracking inventory in real time**. The advantage of knowing how much inventory you have at all times is helpful as it allow you make smart decisions about when t make order products. Inventory management software will updates your records when you buy and sell products, which will be very useful to monitor leftover stocks.
3. **Optimize warehouse organization**. Speed is essential in having good customer service. Inventory management software makes your business move faster by improving your warehouse’s layout. You can group & categorized the products together and position popular products in ways that make them easier to access when orders come in.
4. **Improve customer service**: More information means better service. WMS help to reduce human error from inaccurate data. Using WMS can save the time & provide better quality service while it may be more satisfy the customers
5. **Quality Control**: Controlling the quality for your product is important. A good quality control may extend your business & it prevent the unnecessary cost that you may not aware due to using poor QC. WMS can help to monitor the incoming & outgoing date of stocks or you can even record down the product expire date in inventory.
6. **Space & storage management:** Knowing the amount of the products in your warehouse is very beneficial. It will help you to organize the space & storage. WMS will help you to plan your limited space by ordering the right amount of product which can fit into the space you have.

## Definitions, Acronyms and Abbreviations

**Acronyms and Abbreviations to be referenced from this table:**

|  |  |
| --- | --- |
| **Abbreviation/Acronyms** | **Definitions** |
| SRS | Software Requirement Specification |
| WMS | Warehouse Management System |
| QC | Quality Control |

# Roles and Responsibilities

## Responsibility Matrix

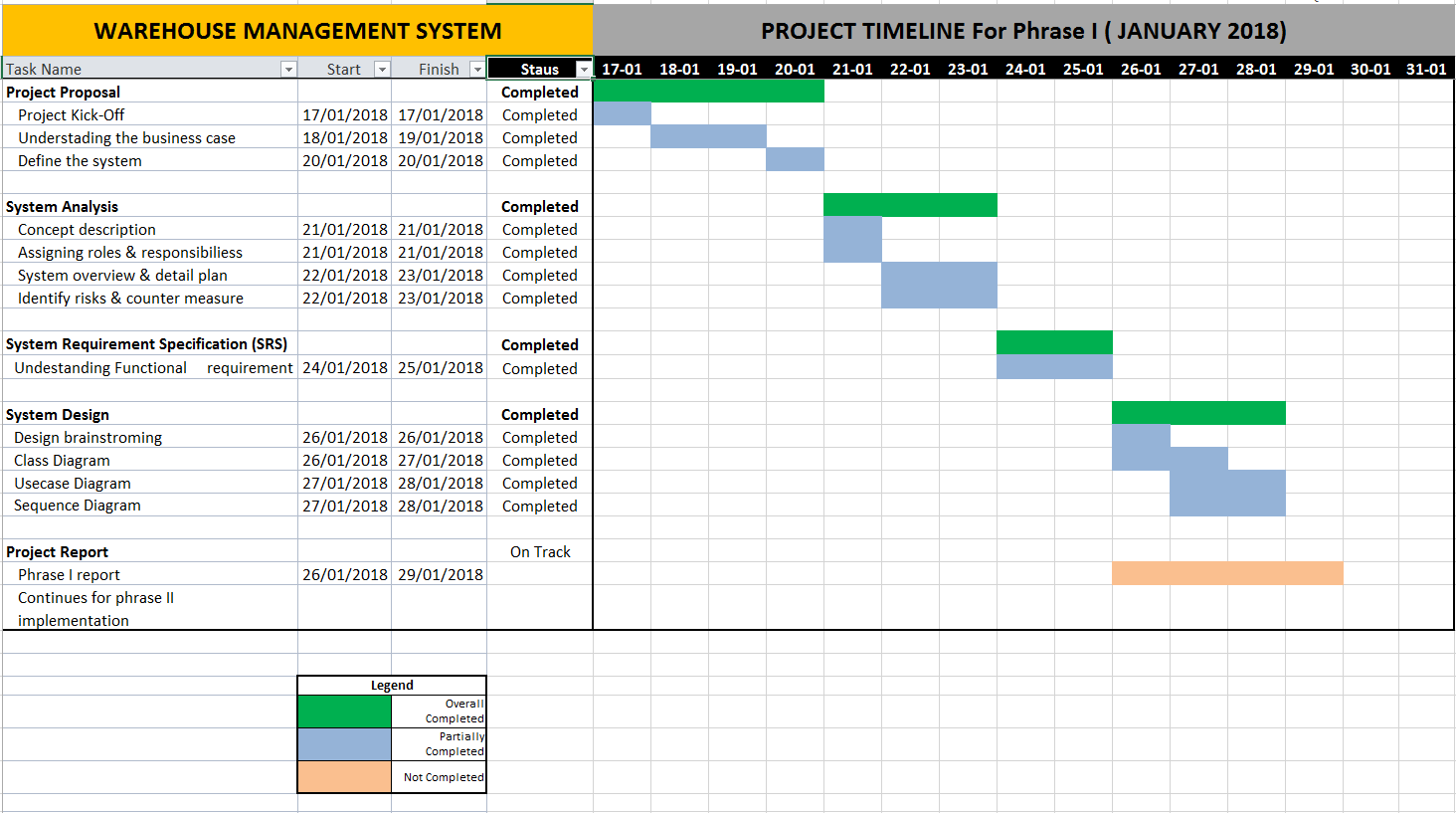
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Responsibilities | Name | | | | |
| Kalista | Allen | Gavin | Jonas | Johns |
| Business Case |  |  |  |  | **X** |
| Project Schedule |  |  |  |  | **X** |
| Class Diagram | **X** | **X** |  |  |  |
| Project Report | **X** | **X** | **X** | **X** | **X** |
| Software Requirement (SRS) | **X** |  |  |  |  |
| Data Dictionary |  | **X** |  |  |  |
| Use Case Diagram/Description |  |  | **X** | **X** |  |
| Sequence Diagram/Description |  |  | **X** | **X** |  |
| Risk Assessment | **X** |  |  |  |  |
| Overall Project Review | **X** | **X** | **X** | **X** | **X** |

## Role Allocation

| **Name | Title** | **Role** | **Responsibilities** |
| --- | --- | --- |
| Name | | The Project Manager is responsible for developing, in conjunction with the Project Sponsor, the project charter. The Project Manager ensures that the project is delivered on time, within budget, and to the required quality standards. | * Manage and lead the project team. * Manage the coordination of the partners and the working groups. * Develop and maintain a detailed project plan. |
| Name | Technical writer | Technical writer prepares the supporting documents & instruction manuals | * Provide manuals &   System reference documents |
| Name | System Analysis | System analysis evaluate system workflow & product specifications. | * Provide the overall workflow & concept |
| Name | System Designer | System designer design the system based on the business needs & functional requirement | * Design interfaces & functionality to include in system |
| Name | Developer | The developer is the one who implement the system according to system design & analysis workflow. | * System implementation * Coding& testing of the products |

# Implementation Plan

## Project Schedule



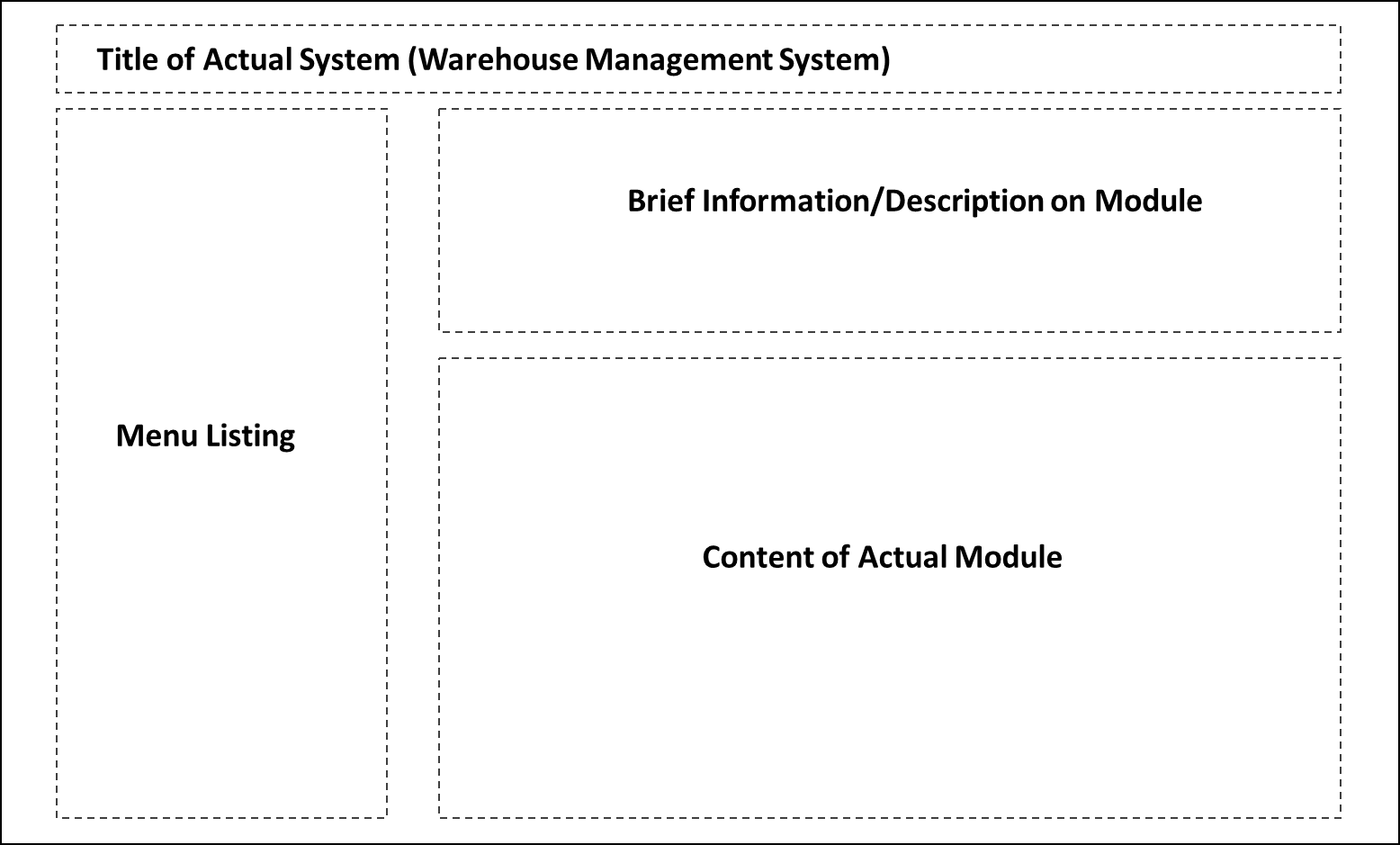
# Overview of System

## Software Requirements Specification

### Functional Specifications

1. **System Overall Design Layout**

The system will have an overall design layout as per below where each section will have their corresponding information displayed. There will be a menu list on the left of the system and a brief information of the module function on right top and the actual module on bottom right.



1. **Login**

This module allows the user to login and the main menu will display the options that they have the permissions to view. System will also validate the encrypted password and will lock the account if the user has made 3 unsuccessful attempts, this will result in the user not being able to login once the 3 attempts have been used up.

**Below depicts the processing logic for this function:**

1. User key in username and password.
2. System validate user is not locked and compare password.
3. If user is not locked then go to step 4 else show error message and end.
4. If password matches then save user session, redirect to main menu, and end.
5. If password does not match then add 1 try to user attempt and show error message and end.
6. **Main Menu**

This module will display a main menu with the following menu options and shows the low stock count of any stock falling under a configured amount.

* 1. Stock Category Management
  2. Stock Alert Setting Management
  3. Stock Listing Management
  4. Stock Inventory Management
  5. Stock Summary Report

System redirects to module page on user click action.

1. **Stock Category Management**

This module allows the user to manage the stock category of the system. The module will have a textbox to key in the category description and user will be able to select the parent category if any. If the parent category is not selected then the category will be set as a parent category.

* 1. Description
  2. Parent Category

Below depicts the processing logic for this function.

1. Load all category to parent category selector where ParentID = 0.
2. User key in category description and select parent category.
3. If parent category is not selected then set parent as 0.
4. If parent category is selected then set parent to selected value ParentID.
5. Save the entry when user click submit.
6. **Stock Alert Setting Management**

This module allows the user to enter threshold for each stock item. For each stock item id, there will only be 1 threshold. The system will alert the user that the particular stock in the warehouse is below the threshold on the main menu page. This setting can be manage under manage stock listing module under each of the stock item.

**Refer to section Functional Specifications - Part 6.**

1. **Stock Listing Management**

This module allows the user to manage the different stock item that the warehouse holds. User will have to create the stock first before any stock movement; stock in or stock out can be performed. User is able to Add/Edit the stock item, below states the fields for the stock item.

1. **Description**
2. **Category**
3. **Sub-Category**
4. **Unit Price**
5. **Threshold**
6. **Count [Default 0]**
7. **Created On**
8. **Created By**
9. **Modified On**
10. **Modified By**

**Below depicts the processing logic for this function.**

* 1. **Add Stock Item**

1. User enter all required field.
2. User click on submit button.
3. System save record on submit.
   1. **Edit Stock Item**
4. User select stock item to edit.
5. User update all record field.
6. User click on submit button.
7. System save record on submit.
   1. **Search Stock item**
8. User key in search criteria.
9. User click on submit button.
10. System retrieve all match records on submit.
11. **Stock Movement Management**

This module allows the user to enter stock movement into the system to keep track of current stock level real time.

* 1. **Stock Item**
  2. **Inbound/Outbound**
  3. **Count**

**Below depicts the processing logic for this function.**

* 1. **Inbound Stock**

1. User select stock item from the list.
2. User select inbound from the list.
3. User key in the stock movement count.
4. User click on submit button.
5. System save record on submit and update stock count for that stock item.
   1. **Outbound Stock**
      1. User select stock item from the list.
      2. User select outbound from the list.
      3. User key in the stock movement count.
      4. User click on submit button.
      5. System check user keyed in stock movement count is less than system stock count.
      6. System save record on submit and update stock count for that stock item.
6. **Stock Summary Report**

This module allows the user to generate reports based on the criteria selected. This allow user the flexibility to generate daily, weekly, monthly and yearly reports.

The report will have the below columns:

* 1. **Transaction Date**
  2. **Stock Item**
  3. **Inbound/Outbound**
  4. **Unit Price**
  5. **Total Price**

### Non-Functional Specifications

1. **Maintainability**

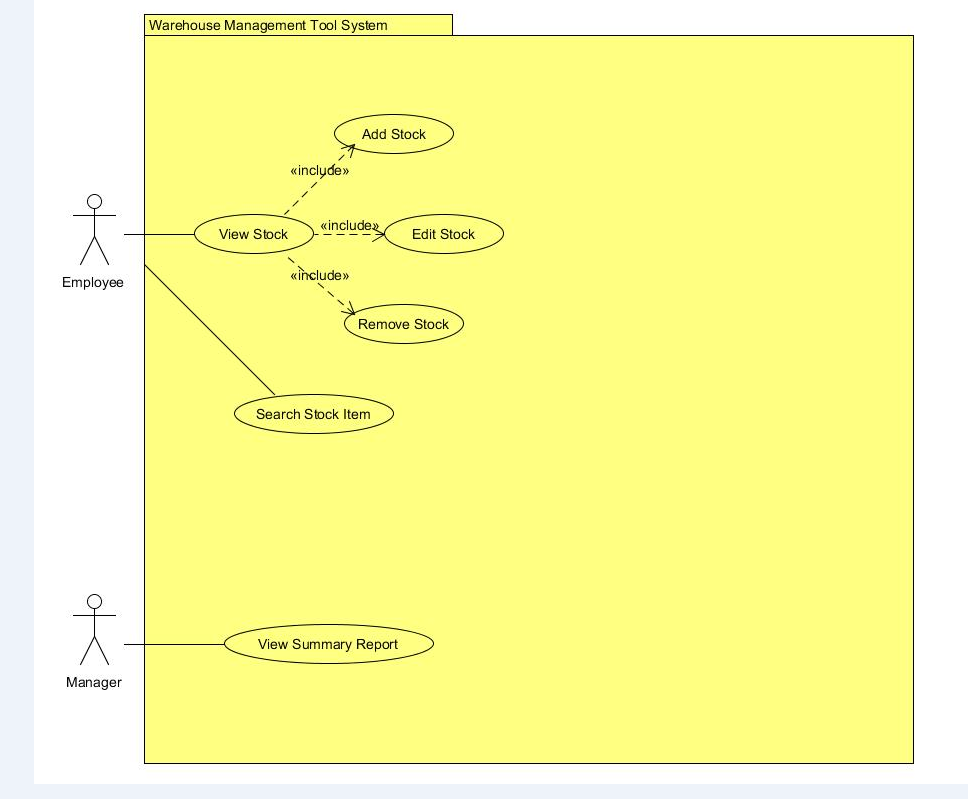
The system will be easy to maintain and will support scalability so that the system can grow and features can be added, functionalities to be further improved whenever necessary.

1. **Usability**

The system will be user-friendly and intuitive which will encourage user to use the system and it will save time for the user to figure out the system. System’s Usability needs to be a key component as it will reduce the number of change request from the user.

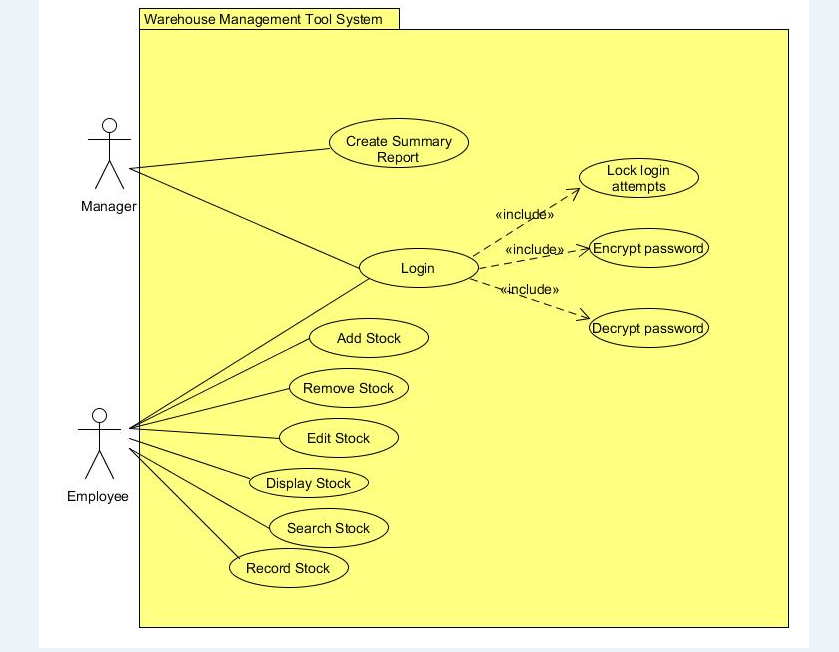
## Use Case Diagram

### Iteration 1



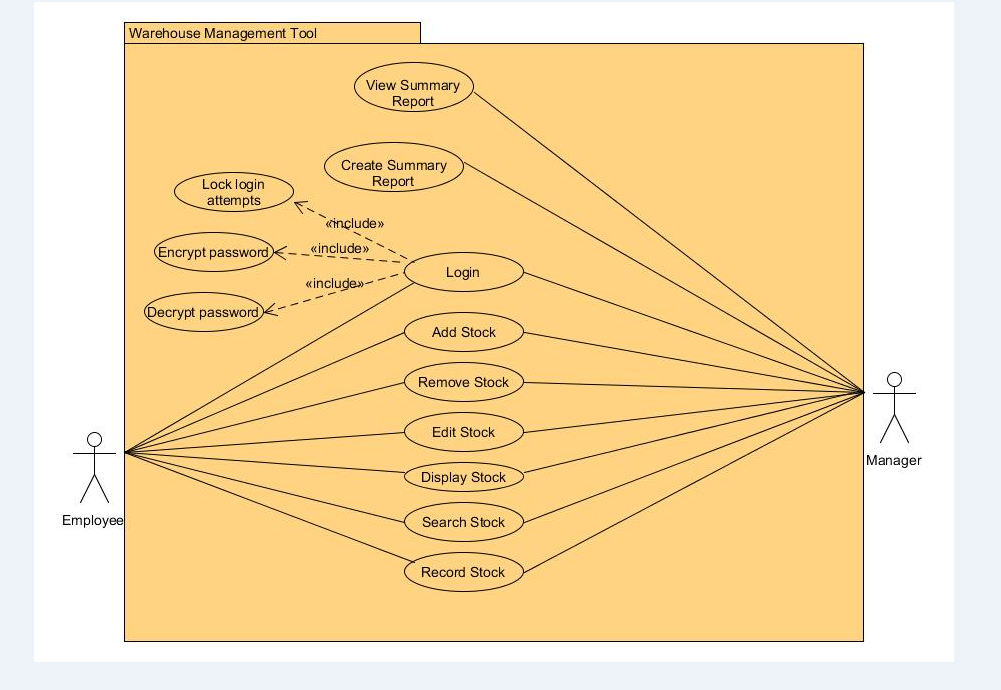
The Use Case Diagram above was first initially idea our group sketched out before our first group meeting for assignment 1.

### Iteration 2



The Use Case Diagram above was the modified version after we have discussed during our group meeting.

### Final Version



The Use Case Diagram above is the finalised version after various discussion and brainstorming session.

## Use Case Detailed Description

### Detailed Use Case – Login

|  |  |
| --- | --- |
| **Use Case Description** | This use case describe how the user login into the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must have a user account in the system  Hash algorithm to encrypt and decrypt password |
| **Flow** | **Main Flow**   1. System provides user with an input dialog box consisting of username and password field 2. User will input the account info into the field provided and press enter 3. System runs hash algorithm on the password to get the hash input 4. System checks if hash input matches user stored hashed password. 5. System will validate the user details 6. System will allow user access   **Alternate Flow (AF1) –Invalid password or username**   1. System inform user that the password or username entered is invalid   **Alternate Flow (AF2) –Three unsuccessful login attempts**   1. System will mark the record of the account as “locked” 2. System will not allow user to login into the system |
| **Post-conditions** | 1. System will open and display user functions |
| **Business Rules** | 1. Mandatory fields must be entered |

### Detailed Use Case – Create Summary Report

|  |  |
| --- | --- |
| **Use Case Description** | This use case creates summary report for Manager |
| **Primary Actor (s)** | Manager |
| **Pre-conditions** | Manager authorized access to access database |
| **Flow** | **Main Flow**   1. Manager gets access to database 2. Manager select information to generate report 3. Summary report generated |
| **Post-conditions** | 1. Report from database is generated |
| **Business Rules** | 1. Mandatory fields must be entered |

### Detailed Use Case – View Summary Report

|  |  |
| --- | --- |
| **Use Case Description** | This use case describe how manager view the summary report |
| **Primary Actor (s)** | Manager |
| **Pre-conditions** | Summary report exist in the database |
| **Flow** | **Main Flow**   1. User click on view report tab in system 2. System will display all the various past reports created 3. User click on the specific report 4. System will display the details of the specific report |
| **Post-conditions** | 1. System show report to the manager for viewing |
| **Business Rules** | 1. Mandatory fields must be entered |

### Detailed Use Case – Add Stock

|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager add stock into the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Add Stock” tab 3. Employee or Manager will key in the necessary information for the stock 4. System will store the information 5. System will display the added stock information 6. Use case ends   **Alternate Flow (AF1) – Invalid Entry**   1. Employee or Manager key in information with wrong format 2. System display “Invalid Value” 3. Use case ends |
| **Post-conditions** | 1. All the stocks added will be stored in the system |
| **Business Rules** | 1. Mandatory fields must be entered. |

### Detailed Use Case – Remove Stock

|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager remove stock from the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Remove Stock” tab 3. Employee or Manager will select the stock that he/she wish to remove 4. System will remove the stock 5. System will display “Selected stock have been removed” 6. Use case ends   **Alternate Flow (AF1) – Selection Exceeded**   1. Employee or Manager select more than 1 record 2. System display “You can only remove 1 stock at a time” 3. Use case ends |
| **Post-conditions** | 1. The selected stock will be removed from the system |
| **Business Rules** | 1. Mandatory fields must be entered. |

### Detailed Use Case – Edit Stock

|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager edit stock from the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Edit Stock” tab 3. Employee or Manager will select the stock that he/she wish to edit 4. Employee or Manager will edit the stock information 5. System will save the changes made 6. System will display the edited stock information 7. Use case ends   **Alternate Flow (AF1) – Selection Exceeded**   1. Employee or Manager select more than 1 stock 2. System display “You can only edit 1 stock at a time” 3. Use case ends |
| **Post-conditions** | 1. The system will save the edited stock information |
| **Business Rules** | 1. Mandatory fields must be entered. |

### Detailed Use Case – Display Stock

|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager display the stock/s from the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Display Stock” tab 3. Employee or Manager will select the stock that he/she wish to display 4. System will display the stock information selected 5. Use case ends   **Alternate Flow (AF1) – Selection Exceeded**   1. Employee or Manager select more than 1 stock 2. System display “You can only view 1 stock at a time” 3. Use case ends |
| **Post-conditions** | 1. The system will display the stock selected |
| **Business Rules** | 1. Mandatory fields must be entered. |

### Detailed Use Case – Search Stock

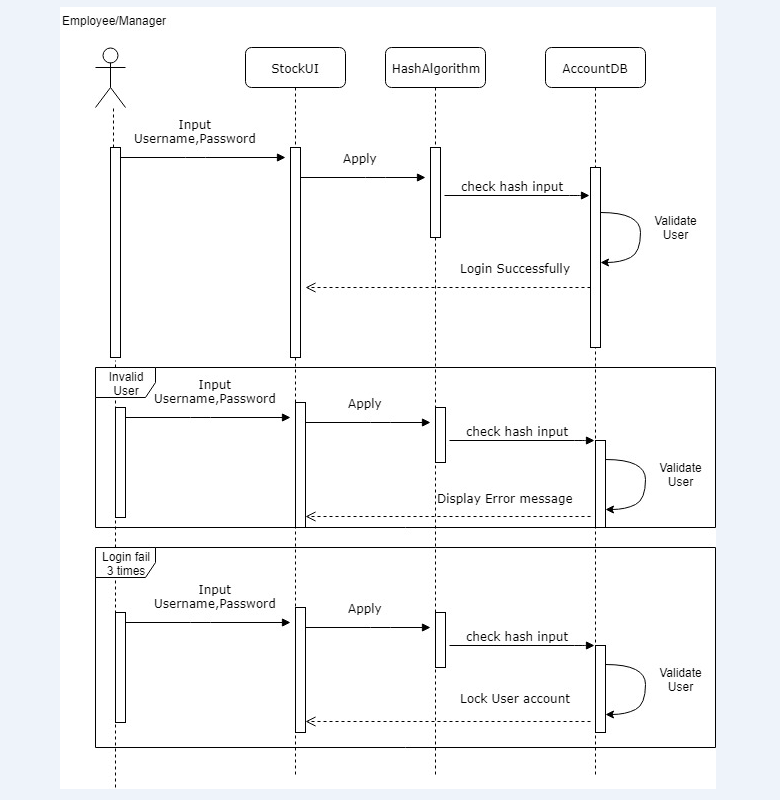
|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager search the stock from the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Search Stock” tab 3. Employee or Manager will select a particular category or sub-category 4. Employee or Manager will key in the stock name or keyword 5. System will display the stock related to the keyword being searched 6. Employee or Manager will select the stock that he/she wanted 7. Use case ends   **Alternate Flow (AF1) – Stock not found**   1. Employee or Manager key in the stock name on the search bar 2. System unable to locate the stock 3. System display “No available stock found” 4. Use case ends |
| **Post-conditions** | 1. The system will display a list of stock being searched based on the keyword according to the price range and quantity in ascending or descending order |
| **Business Rules** | 1. Mandatory fields must be entered. |

### Detailed Use Case – Record Stock

|  |  |
| --- | --- |
| **Use Case Description** | This use case describes how an Employee and Manager record the stock into the system |
| **Primary Actor (s)** | Employee, Manager |
| **Pre-conditions** | User must first login to the system |
| **Flow** | **Main Flow :**   1. Employee or Manager will login to the system with his/her username and password 2. Employee or Manager will click on the “Record Stock” tab 3. Employee or Manager will select “incoming stock” or “outgoing stock” 4. Employee or Manager will key in the stock’s detail 5. System will save the detail 6. Use case ends   **Alternate Flow (AF1) – Invalid Entry**   1. Employee or Manager key in the stock’s detail 2. System detect that the information is in wrong format 3. System display “invalid entry” 4. Use case ends |
| **Post-conditions** | 1. The system will save the detail keyed in by Employee or Manager and store them according to their respective category |
| **Business Rules** | 1. Mandatory fields must be entered. |

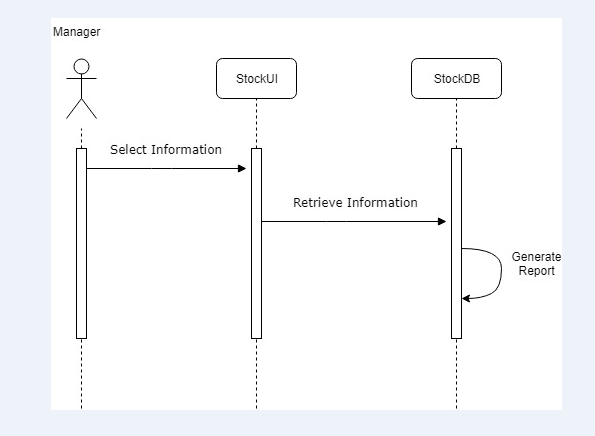
## Sequence Diagram

### Sequence Diagram – Login



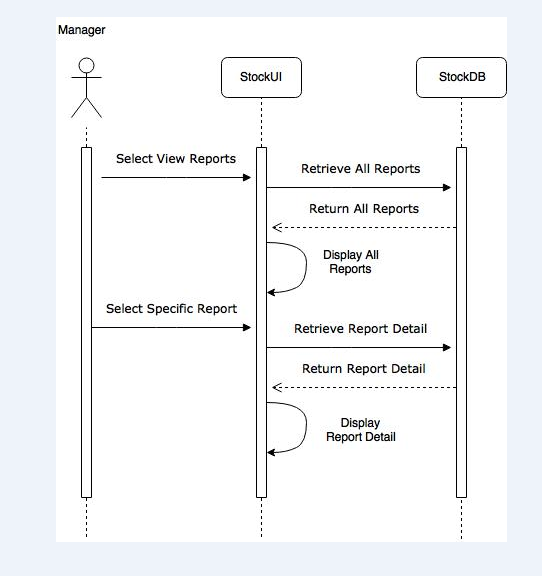
The Sequence Diagram above show the process of how the user will login to the system.

### Sequence Diagram – Create Summary Report



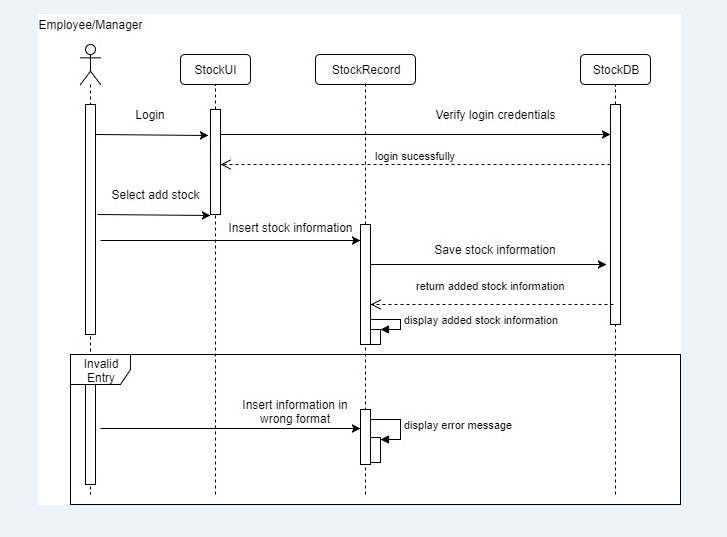
The Sequence Diagram above shows how the Manager create a summary report from the system.

### Sequence Diagram – View Summary Report



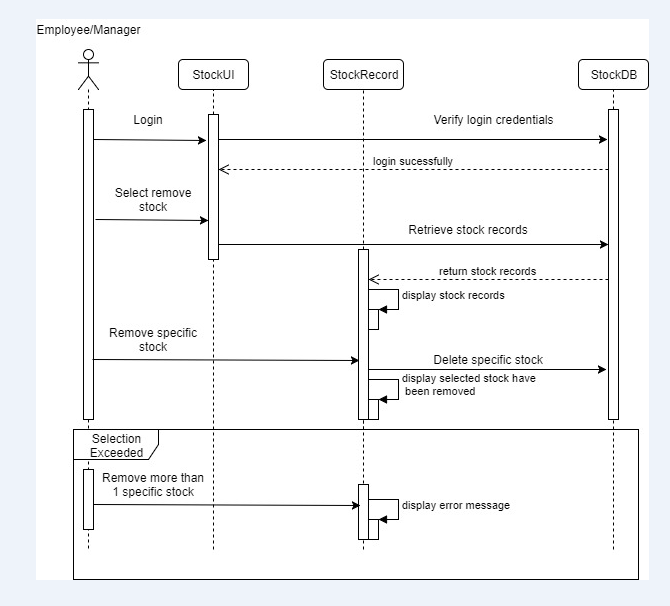
The Sequence Diagram describes how the Manager view a summary report from the system.

### Sequence Diagram – Add Stock



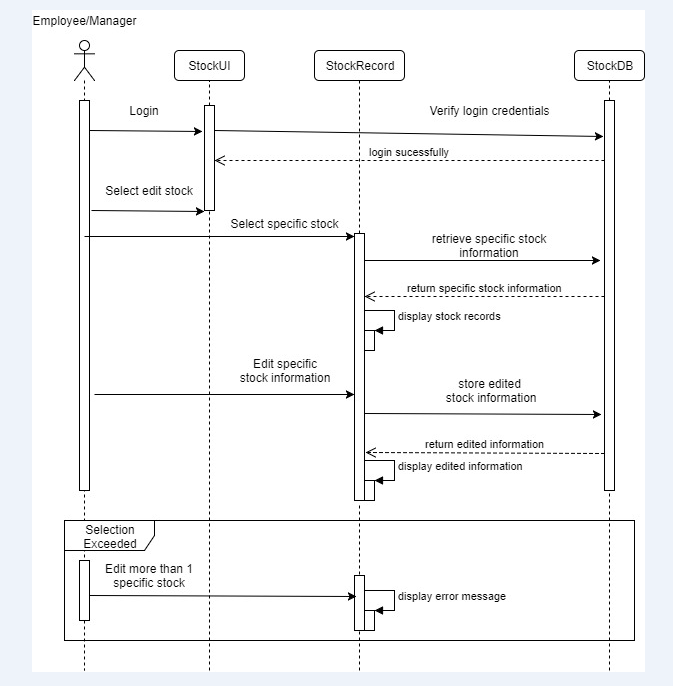
The Sequence Diagram above show how the Manager and Employee add a stock record into the system.

### Sequence Diagram – Remove Stock



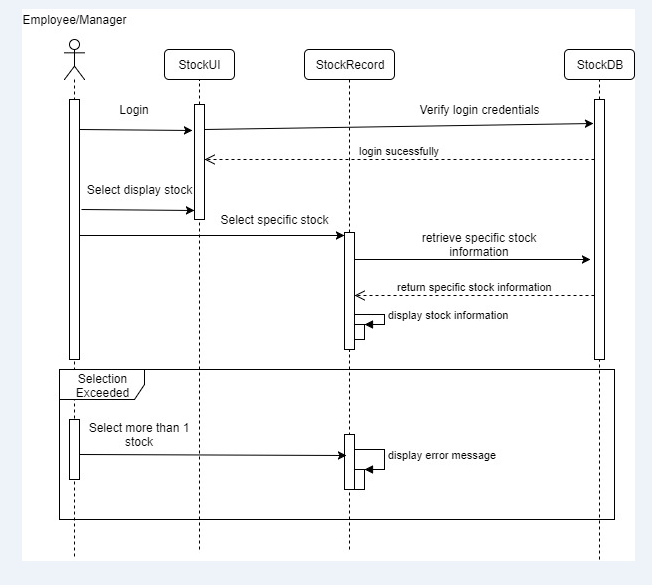
The Sequence Diagram above show how the Manager and Employee remove a stock record from the system.

### Sequence Diagram – Edit Stock



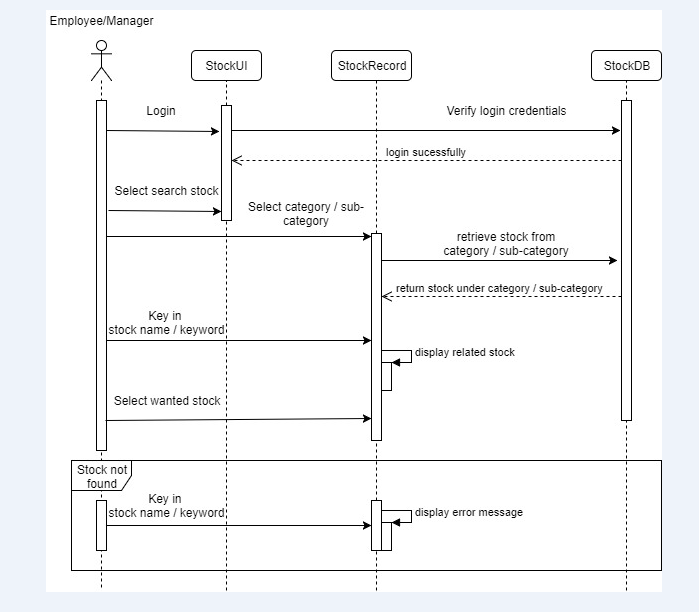
The Sequence Diagram above show how the Manager and Employee edit a stock record from the system.

### Sequence Diagram – Display Stock



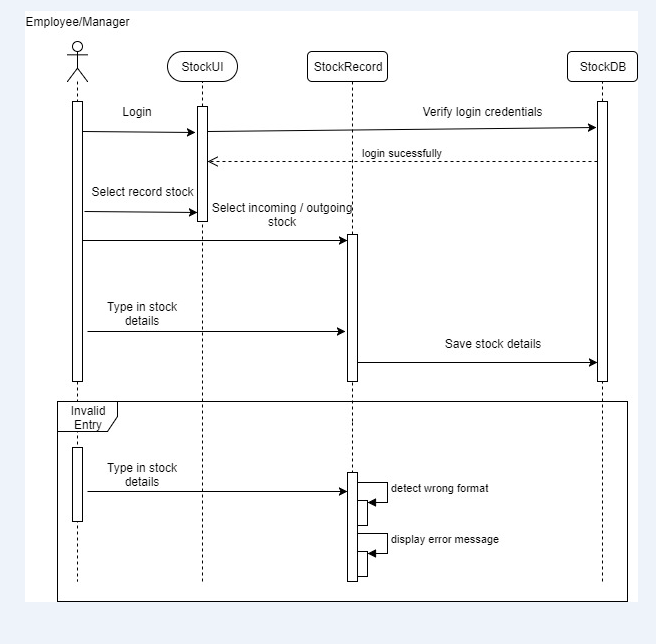
The Sequence Diagram above show how the Manager and Employee navigate through the system to display stock record.

### Sequence Diagram – Search Stock



The Sequence Diagram above shows how the Manager and Employee search for a particular stock record from the system.

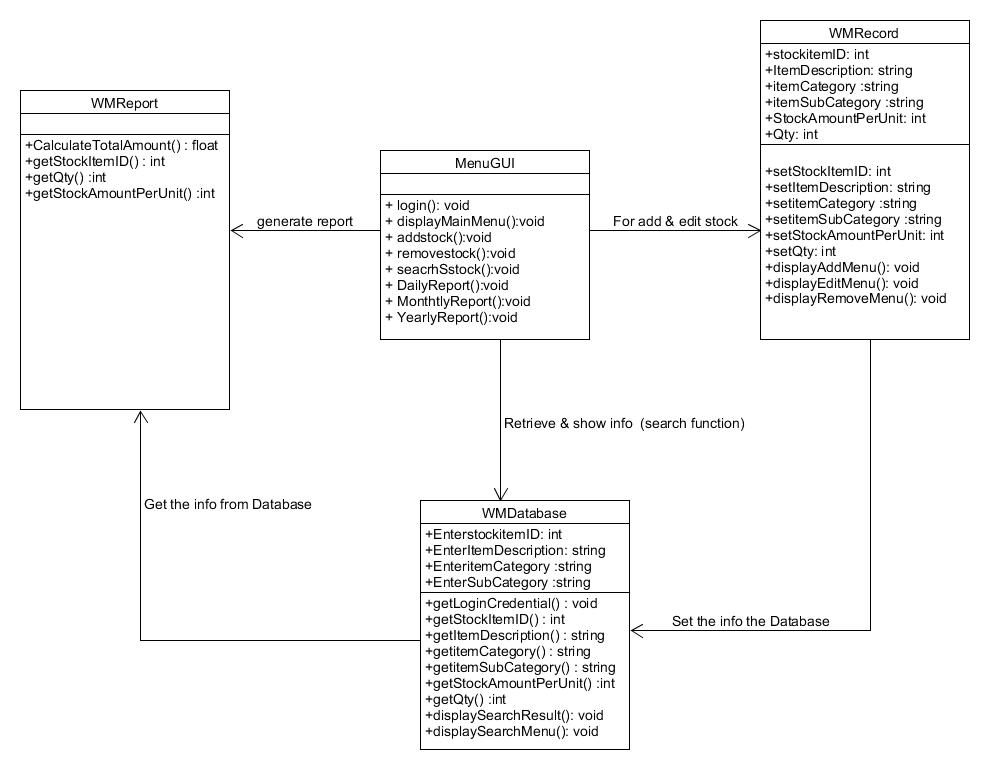
### Sequence Diagram – Record Stock



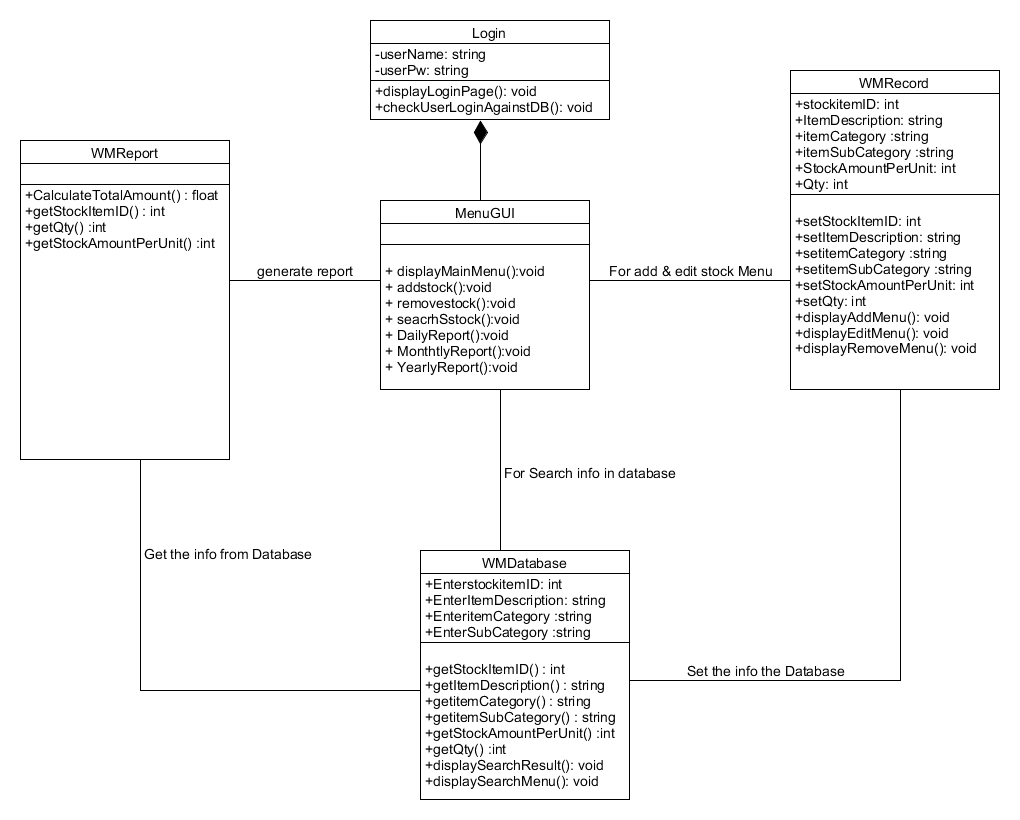
The Sequence Diagram above show how the Manager and Employee store a particular stock record into the system.

## Class Diagram

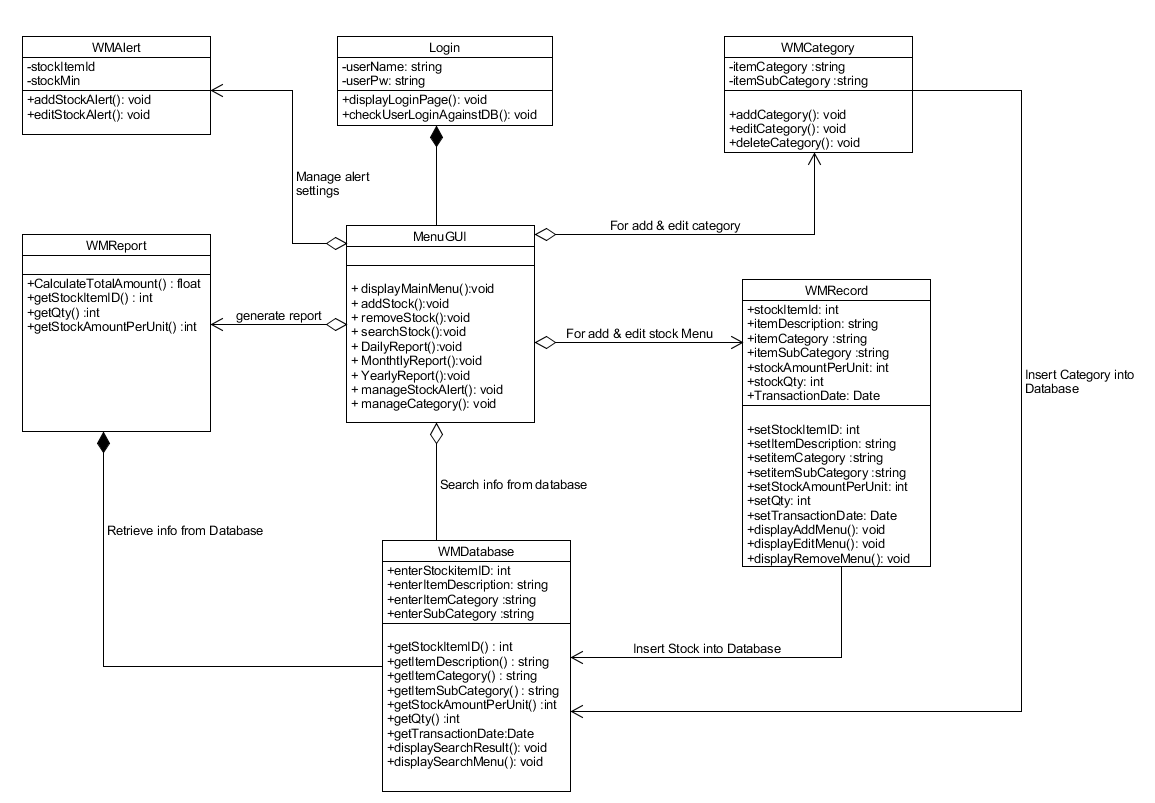
### Iteration 1



### Iteration 2



### Final Version



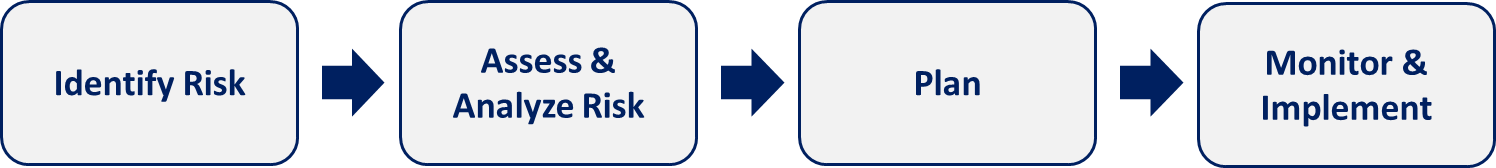
## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **WMRecord** |  |  |  |  |
| **Field Name** | **Data Type** | **Field size** | **Description** | **example** |
| Stock Item ID | int | 7 | Unique identifier for Stock item ID | 456 |
| Item Description | String | 256 | Description for the item | Samsung Phone XYZ |
| Item Category | String | 256 | Description of category of the item | Electronics |
| Item Sub-category | String | 10 | Description of Sub category of the item | Mobile Phone |
| Amount Per Unit | Int | 5 | Amount of quantity for each stock item | 1200 |
| Qty | Int | 5 | Quatity of the stock | 100 |
| Transacted date | Date | 8 | Transaction Date for the item | 12-12-2017 |
|  |  |  |  |  |
| **WMDatabase** |  |  |  |  |
| **Field Name** | **Data Type** | **Field size** | **Description** | **example** |
| Stock Item ID | int | 7 | Unique identifier for Stock item ID | 456 |
| Item Description | String | 256 | Description for the item | Samsung Phone XYZ |
| Item Category | String | 256 | Description of category of the item | Electronics |
| Item Sub-category | String | 10 | Description of Sub category of the item | Mobile Phone |
|  |  |  |  |  |
| **WMAlert** |  |  |  |  |
| **Field Name** | **Data Type** | **Field size** | **Description** | **example** |
| Stock Item ID | int | 7 | Unique identifier for Stock item ID | 456 |
| Stock Min | int | 7 | For inserting Min value for stock alert | 40 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Login** |  |  |  |  |
| **Field Name** | **Data Type** | **Field size** | **Description** | **example** |
| UserName | String | 50 | User Name for login | admin123 |
| Password | String | 50 | Password for login | [sim@connect123](mailto:sim@connect123) |
|  |  |  |  |  |
| **WMCategory** |  |  |  |  |
| **Field Name** | **Data Type** | **Field size** | **Description** | **example** |
| Item Category | String | 256 | Description of category of the item | Electronics |
| Item Sub-category | String | 10 | Description of Sub category of the item | Mobile Phone |

# Risk and Counter Measures

The following image below will explain how we are going to mitigate the risks in our project:



| **Risk** | **Likelihood** | **Impact** | **Mitigation Strategy** | **Resolution** |
| --- | --- | --- | --- | --- |
| Insufficient knowledge on  Sub-versioning | High | High | 1. Every individual in the team to research and share their proposed sub-versioning system 2. Use free sub-versioning system online to explore | * 1. We experiment with the different sub-versioning tools and have decided to use GitHub, as it is a free tool online. |
| Different understanding of class diagram concept | High | High | 1. Discuss and talk to lecturer to clear our doubts 2. Finalize our final correct understanding of domain model | 1. We have talked and discussed amongst each other after researching and come to a common consensus that the class diagram is how the system will work in terms of coding. 2. Built our class diagram based on that knowledge |
| Tight timeline | Medium | High | 1. Accommodate and find time during the weekends to meet up and discuss | 1. We have resolved this by meeting up during the weekends and after class to complete this assignment. We also used other forms of communication (i.e. WhatsApp) to catch up on our individual progress. |

| **Risk** | **Likelihood** | **Impact** | **Mitigation Strategy** | **Resolution** |
| --- | --- | --- | --- | --- |
| Insufficient conceptual knowledge of Sequence diagram | Medium | High | 1. Research and read up on sequence diagram examples online 2. Talk to our lecturers to confirm the understanding 3. Perform re-iteration of our sequence diagram and check with each other if they understand the sequence diagram drawn | 1. We have first came up with the first cut our sequence diagram and got our lecturers advice before we continue re-iterating. 2. Discussed together and made changes based on the teammates input and suggestions. |

# Meeting Minutes

**Meeting Minutes #1**

|  |  |  |
| --- | --- | --- |
|  | CSCI222- Systems Development  Assignment 1 | |
| Meeting Agenda & Minutes of Meeting | Page 40 of 43 |

| Meeting: | Assignment 1 – Meeting #1 | | |
| --- | --- | --- | --- |
| Date of Meeting:  (DD-MMM-YYYY) | 17-JAN-2018 | Time: | 7:15pm – 10:00pm |
| Location: | Level 5 seating area | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting Objective** | | | |
| * Allocation of Roles and Responsibilities * Understanding the requirements for the Warehouse Management System * Come out with Project timeline/Schedule | | | |
| **Attendees** | | | |
| Name | Student Number | E-mail | |
| Kalista Chan | 5986345 | xkchan002@mymail.sim.edu.sg | |
| Gavin Teo Heong Hwee | 6098058 | hhteo005@mymail.sim.edu.sg | |
| Allen Tam He Sheng | 5986485 | hstam001@mymail.sim.edu.sg | |
| Jonas Chok | 6098459 | jwjchok001@mymail.sim.edu.sg | |
| Kyaw Myo Aung , Johns | 6097868 | Myoak001@mymail.sim.edu.sg | |
|  |  |  | |
| **Meeting Agenda** | | | |
| **Topic** | | | **Owner** |
| Allocation of Roles and Responsibilities   * Documenting in report | | | Allen |
| Project Timeline/Schedule | | | Gavin |
| Brainstorm and detailing the requirements | | | Kalista |
| **Pre-work/Preparation (documents/handouts to bring, reading material, etc.)** | | | |
| **Description** | | | **Prepared by** |
| NIL | | | NIL |

| **Meeting Minutes** | | |
| --- | --- | --- |
| **Topic** | **Owner** | **Type** |
| **Allocation of Role & Responsibility**  Allen asked the team for their various strengths and weaknesses. The team agreed to have these roles for our project-:   * Class Diagram – Allen & Kalista * Data Dictionary – Allen * Risk & Counter Measures – Kalista * Functional & Non-Functional Requirements – Kalista * Use Case Diagram & Detailed Use Case – Heong Hwee & Jonas * Sequence Diagram – Heong Hwee & Jonas * Business Case & Project Schedule – Johns | Allen | Action |
| We discussed about our understanding of the sub-versioning and our different understanding of class diagrams and sequence diagram. After discussion, we realised that we needed to do more research in this 3 aspects, so as to complete the different diagrams.  Items to be discussed the next meeting:   * Research to be done by each team mates * Sharing of the different sub-versioning tools available   Share and explain understanding of class and sequence diagram and how to apply it for our assignment | All | Action |
| We analysed the requirements together and came up with a draft version of a Use Case Diagram and a Class Diagram based on our understanding of the requirements. | Info | Info |
| Our team have discussed and have came up with a list of questions below to clarify with Mr Liaw.   * Risk encounters: plan or coding? * Do we need sequence diagram with use case * Is individual work diary required. * Do we need to use database system. * If we do not use database system do we need a entity diagram. * Ask about versioning of the system. | All | Action |

**Meeting Minutes #2**

|  |  |  |
| --- | --- | --- |
|  | CSCI222- Systems Development  Assignment 1 | |
| Meeting Agenda & Minutes of Meeting | Page 42 of 43 |

| Meeting: | Assignment 1 – Meeting #2 | | |
| --- | --- | --- | --- |
| Date of Meeting:  (DD-MMM-YYYY) | 21-JAN-2018 | Time: | 09:00am – 11:00am |
| Location: | Block B Level 2 seating area | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting Objective** | | | |
| * Use Case Diagram * Sequence Diagram * Business Case and Project Plan * Domain Model and Data Dictionary * Analysis and Design plan | | | |
| **Attendees** | | | |
| Name | Student Number | E-mail | |
| Kalista Chan | 5986345 | xkchan002@mymail.sim.edu.sg | |
| Gavin Teo Heong Hwee | 6098058 | hhteo005@mymail.sim.edu.sg | |
| Allen Tam He Sheng | 5986485 | hstam001@mymail.sim.edu.sg | |
| Jonas Chok | 6098459 | jwjchok001@mymail.sim.edu.sg | |
| Kyaw Myo Aung , Johns | 6097868 | Myoak001@mymail.sim.edu.sg | |
| **Meeting Agenda** | | | |
| **Topic** | | | **Owner** |
| Use Case Diagram | | | All |
| Sequence Diagram | | | Heong Hwee and Jonas |
| Detailed Use Case | | | Heong Hwee and Jonas |
| Project Detailed Plan | | | John |
| Domain Model | | | Allen and Kalista |
| Data Dictionary | | | Allen and Kalista |
| **Pre-work/Preparation (documents/handouts to bring, reading material, etc.)** | | | |
| **Description** | | | **Prepared by** |
| NIL | | | NIL |

| **Meeting Minutes** | | |
| --- | --- | --- |
| **Topic** | **Owner** | **Type** |
| **Meeting Agenda**   * Allen have allocate different role and assignment for each individual team member * To finalise the flow of the system   **Business Case & Project Plan**   * John to write down out business case along with project plan   **Finalise Use Case Diagram**   * Heong Hwee and Jonas to design the Use Case Diagram * Proceed to Sequence Diagram and Detailed Use Case   **Finalise Class Diagram**   * Allen and Kalista to discuss and design for class diagram   **Meeting Minutes**   * Heong Hwee to write out the meeting minutes for team 2nd meeting | All  John  Heong Hwee & Jonas  Allen & Kalista  Heong Hwee |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Review of Previous Meeting Actions** | | | | | |
| **#** | **Action** | **Status** | **Owner** | **Due Date** | **Updates** |
| 1 | Understanding of Class & Sequence Diagram & how to apply to assignment 1 | Completed | All | 21/01/18 |  |
| 2 | Sharing of sub-versioning tools | In-Progress | All |  |  |

Sub versoning testing