**Software Requirements Specification**

**for**

**WAREHOUSE MANAGEMENT TOOL**

**Prepared by :**

<TEAM C2>

<GRP MEMBER 1>

<GRP MEMBER 1>

<GRP MEMBER 1>

<GRP MEMBER 1>

<GRP MEMBER 1>

<GRP MEMBER 1>

**Table of Contents**

[1.0 INTRODUCTION 5](#_Toc505818495)

[1.1 VISION 5](#_Toc505818496)

[1.2 TARGET AUDIENCE 5](#_Toc505818497)

[1.3 PROJECT SCOPE 5](#_Toc505818498)

[2.0 OVERALL DESCRIPTION 6](#_Toc505818499)

[2.1 PRODUCT PERSPECTIVE 6](#_Toc505818500)

[2.2 PRODUCT FEATURES 6](#_Toc505818501)

[2.3 SYSTEM INTERFACES 6](#_Toc505818502)

[2.4 OPERATING ENVIRONMENT 6](#_Toc505818503)

[3.0 SOFTWARE SPECIFIC REQUIREMENTS 7](#_Toc505818504)

[3.1 DESCRIPTION 7](#_Toc505818505)

[3.2 SOFTWARE INTERFACES 8](#_Toc505818506)

[3.2.1 LOGIN FEATURES 8](#_Toc505818507)

[3.2.2 WM TOOL FEATURES 9](#_Toc505818508)

[4.0 PROJECT PLAN FOR INCEPTION, ELABORATION, AND CONSTRUCTION PHASE 13](#_Toc505818509)

[4.1 PROJECT SCHEDULE 14](#_Toc505818510)

[4.2 RISK ANALYSIS AND COUNTERMEASURES 15](#_Toc505818511)

[4.3 INCEPTION, ELABORATION, CONSTRUCTION PHASES 16](#_Toc505818512)

[INCEPTION PHASE 16](#_Toc505818513)

[ELABORATION PHASE 16](#_Toc505818514)

[CONSTRUCTION PHASE 16](#_Toc505818515)

[5.0 GROUP STRUCTURE, ROLES, AND DESIGN ARTEFACTS 18](#_Toc505818516)

[APPENDIX A 29](#_Toc505818517)

[FORMAL MEETING RECORD 29](#_Toc505818518)

[INCEPTION PHASE – ITERATION 1 30](#_Toc505818519)

[MEETING #1 – 2 JANUARY 2018 30](#_Toc505818520)

[Summary 30](#_Toc505818521)

[Agenda 30](#_Toc505818522)

[Report 30](#_Toc505818523)

[ELABORATION PHASE – ITERATION 1 31](#_Toc505818524)

[MEETING #2 – 2 JANUARY 2018 31](#_Toc505818525)

[Summary 31](#_Toc505818526)

[Agenda 31](#_Toc505818527)

[Report 31](#_Toc505818528)

[ELABORATION PHASE – ITERATION 2 32](#_Toc505818529)

[MEETING #3 – 2 JANUARY 2018 32](#_Toc505818530)

[Summary 32](#_Toc505818531)

[Agenda 32](#_Toc505818532)

[Report 32](#_Toc505818533)

[Appendix B 33](#_Toc505818534)

[WORK DIARIES 33](#_Toc505818535)

[TEAM MEMBER 1 : <INSERT NAME> 34](#_Toc505818536)

[TEAM MEMBER 2 : <INSERT NAME> 35](#_Toc505818537)

[TEAM MEMBER 3 : <INSERT NAME> 36](#_Toc505818538)

[TEAM MEMBER 4 : <INSERT NAME> 37](#_Toc505818539)

[TEAM MEMBER 5 : <INSERT NAME> 38](#_Toc505818540)

[TEAM MEMBER 6 : <INSERT NAME> 39](#_Toc505818541)

[Appendix C 40](#_Toc505818542)

[EVIDENCE OF USING VCS 40](#_Toc505818543)

[MEMBERS 41](#_Toc505818544)

[COMMITS 41](#_Toc505818545)

[PULL REQUESTS 42](#_Toc505818546)

## 1.0 INTRODUCTION

### 1.1 VISION

The aim of this project is to develop an inventory system where users are able to manage any incoming and outgoing stocks, as well as performing a query on specific stocks. This could help our target audiences to keep track of the stocks behaviours and it can also serve as a reminder if their stocks are running low.

### 1.2 TARGET AUDIENCE

Managers from warehouse / inventory-related industry.

### 1.3 PROJECT SCOPE

Warehouse Management Tool will serve to managers managing the warehouse inventory. Managers are able to interact with the system with a valid account, and it is provided with the basic features that all users would have access to, such as: modifying stocks amounts, searching for stocks, and viewing summary reports.

The system flow starts from stocks assignment and stocks modifications, to stock summary reviews.

In order to better manage the functions that the systems provide, the Warehouse Management tool is divided into several subsystems:

* Option to encrypt or not to encrypt data obtained from Warehouse Management Tool via the Security System
* Security System encrypt and decrypt manager’s password registered with the Warehouse Management Tool.
* Modification of Stocks System manages any adding, removing and editing of stocks information that are being modified by managers.
* Search System manages query by managers
* Report Summary System generates summary reports based on number of ingoing and outgoing stocks

## 2.0 OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

Warehouse Management (WM) Tool gives a better understanding of day-to-day operation with its summary information of how a warehouse is performing. Aside from this feature, managers can also modify the stocks in anyhow they deemed fit. The managers could also use this data for making critical management decisions.

#### 2.2 PRODUCT FEATURES

Warehouse Management tool has several features for users to navigate:

* Security System is able to encrypt and decrypt manager’s password registered with the Warehouse Management Tool.
* Modification of Stocks System manages any adding, removing and editing of stocks information that are being modified by managers.
* Search System manages query by managers
* Report Summary System generates summary reports based on number of ingoing and outgoing stocks

#### 2.3 SYSTEM INTERFACES

Our Warehouse Management Tool is developed as a desktop application that runs through localhost to provide stocks modifications related services.

Users can log in into the system from the PC . Our system provides basic services such as login, and search, as well as stocks modifications related services, like adding stocks, removing stocks, modifying stocks, and generating summary reports.

#### 2.4 OPERATING ENVIRONMENT

Windows 10

Mac OS

Ubuntu

## 3.0 SOFTWARE SPECIFIC REQUIREMENTS

#### 3.1 DESCRIPTION

The main purpose of WM Tool is to allow users to add/remove/modify stocks with the ability to search through all the stock records and to generate summary reports.

WM Tool provides basic functionality accessible through the following basic functions:

* Add stock : Adds each stock to the records.
* Remove stock : Remove stock form the records.
* Modify stock : Modify the details from the records.
* Search stock : Search stock from the records.:
* View summary reports : Daily,Weekly and Monthly reports of inventory movements and total sales are recorded.

All functionalities included based on the business needs of a WM Tool, to allow more convenient way of viewing stocks inventory in a glance through summary reports.

#### 3.2 SOFTWARE INTERFACES

##### 

|  |
| --- |
| **3.2.1.1 Login Function** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**   * User will login using their registered username and password. |

|  |
| --- |
| **3.2.1.2 Password Encryption** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**   * System will first encrypt the password and see if it matches the password stored at Managers.txt * If it matches, managers will be directed to WM tool menu page. * Else, it will prompt him to enter user’s login details again. |

|  |
| --- |
| **3.2.2.1 WM TOOL INTERFACE** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**   * WM Tool will display a series of functions :   ***Add new stock***  ***Remove stock***  ***Edit stock item***  ***Search stock item***  ***Daily stock summary report***  ***Weekly stock summary report***  ***Monthly stock summary report***  ***Yearly stock summary report***  ***Quit*** |

|  |
| --- |
| **3.2.1.2 Add Stock** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘addStock’ feature is a key component in the warehouse inventory management system, the implementation of this feature would allow for the addition of unique stock by the Manager  Feature is of High Priority as it is required to allow Managers to manipulate the warehouse inventory.  Allow user to add a new stock item to the record. The user will be prompt for the following:   * Stock ID * Stock name |

|  |
| --- |
| **3.2.1.3 Remove Stock** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘removeStock’ feature is a key component in the warehouse inventory management system, the implementation of this feature would allow Managers to remove obsolete stocks  Allow user to remove existing stock item from the record. The user will be prompt for the following:   * Stock ID * Stock name |

|  |
| --- |
| **3.2.1.4 Edit Stock** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘editStock’ feature is a key component in the warehouse inventory management system, the implementation of this feature would allow for the editing of residual stock efficiently.  Allow user to modify existing stock item from the record. The user will be prompt for the following:   * Stock ID * Stock name   User will then be redirected to another submenu that allows user to modify their desired category. |

|  |
| --- |
| **3.2.1.5 Search Stock** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘searchStock’ feature is a secondary component in the warehouse inventory management system, the implementation of this feature would allow users to view relevant and specific stock information  Allow user to search and sort existing stock item from the record. The user will be prompt for the following:   * Stock ID * Stock name * Stock Category |

|  |
| --- |
| **3.2.1.6 Stock Summary Report** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘displaySummary’ feature is a key component in the warehouse inventory management system, the implementation of this feature would allow Managers to view and make critical day-to-day managerial decisions  Allow user to view the summary report from the record. The user will be asked for the following options:   * A) Daily * B) Weekly * C) Monthly * D) Yearly |

|  |
| --- |
| **3.2.1.7 Option to Encrypt** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘login Security’ feature is a key component in the warehouse inventory management system, the implementation of this feature prevent unauthorized access and help maintain data integrity  Allows user to choose to encrypt or not to encrypt data before exit |

|  |
| --- |
| **3.2.1.8 alertStockLow** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘alertStockLow’ feature is a primary component in the warehouse inventory management system, the implementation of this feature helps to prevent stock from being depleted and disrupting operations  Feature is of High Priority as it prevents disruption to the warehouse’s operations by allowing the Manager to replenish stock before it depletes  – System checks to ensure that stock quantity is above the stipulated threshold  – System informs user in session if stock reaches threshold limit |

|  |
| --- |
| **3.2.1.9 setThreshold** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  The ‘setThreshold’ feature is a primary component in the warehouse inventory management system, the implementation of this feature enables manager to set the tolerance which enables the program to be versatile in helping managers make day-to-day decisions  Feature is of High Priority as it prevents disruption to the warehouse’s operations by allowing the Manager to set the tolerance before the notification is called  Manager selects the ‘setThreshold’ option is the main menu  – System request Manager enter a new threshold limit  – System performs data validation on new threshold limit  – System updates the existing threshold value with the new limit if successful |

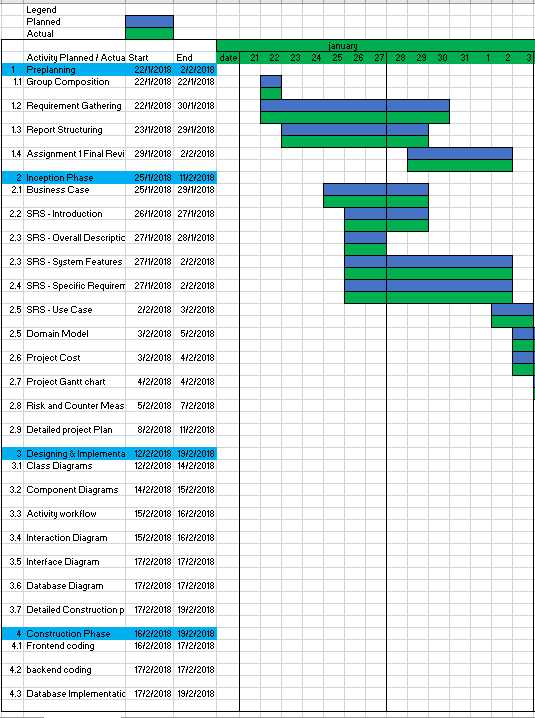
|  |
| --- |
| **3.2.1.10 Quit** |
| Requirement Type: **Functional** |
| PRIORITY: **High** |
| **DESCRIPTION:**  Allow user to quit the whole system. |

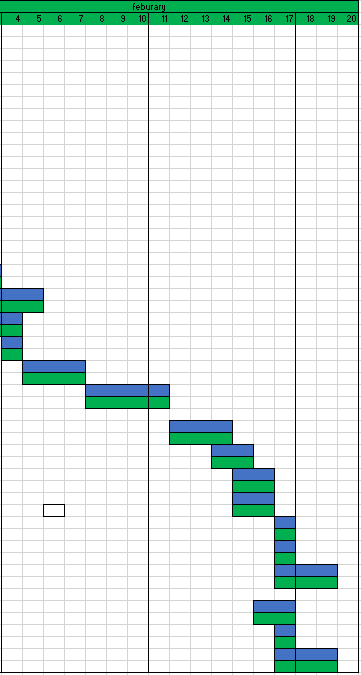
**3.3 CONSTRAINTS**

The system should strictly be implemented using c++.

# 4.0 PROJECT PLAN FOR INCEPTION, ELABORATION, AND CONSTRUCTION PHASE

## 4.1 PROJECT SCHEDULE





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Impact Type** | **Risk Seriousness (%)** | **Likelihood of Occurrence (%)** | **WBS**  **(affected work / task )** | **Risk Description** |
| 1 | Delay in productivity | 90% | 100% | \* | Schedule Inconsistency and Miscommunication |
| 2 | Delay in productivity | 60% | 50% | \* | Library support implementation |
| 3 | Code corruption | 70% | 5% | \* | Files got corrupted by accident |
| 4 | Manpower | 90% | 10% | \* | Team member left the group |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Risk Description** | **Proposed Mgmt Plan** | **(Possible) Reduction in Risk Seriousness (%)** |
| 1 | Schedule Inconsistency and Miscommunication | Plan 1 : Meet every week to discuss current status and to clear any doubts/thoughts on project. | -50% |
| Plan 2 : Do a meet up and have a major revamp to current goals/plans. | -25% |
| 2 | Library support implementation | Plan 1 : Needs to dedicate more time to research on it’s own Library Support for programming software. | -20% |
| 3 | Files got corrupted by accident | Plan 1: Do backups constantly to avoid corruption. In the event if any files got corrupted, we still have version that’s isn’t too much off. | -10% |
| 4 | Team member left the group | Plan 1: Delegate more jobs to cover the loss of the member | -30% |

## 4.2 RISK ANALYSIS AND COUNTERMEASURES

## 4.3 INCEPTION, ELABORATION, CONSTRUCTION PHASES

## INCEPTION PHASE

This phase is the planning phase, where business case is established and scope of the project is determined through the Software Requirement Specification(SRS). A detailed plan for the whole project is also produced in order to determine the tasks required to achieve the desired goal of the project.

The outcome of the inception phase includes:

* A Complete Business Case
* Initial phase of diagrams: Use case, Class Diagram, Component Diagram, Activity diagram.
* A Complete Risks and Countermeasures
* A Complete Project Plan of Inception, Elaboration, and Construction Phase

## ELABORATION PHASE

In this phase, we will revise our current plans and will also revise our existing diagrams to cater for construction phase.

The diagrams for elaboration phase should include:

* Class Diagram
* Activity Diagram
* Use Case Diagram
* Component Diagram

## CONSTRUCTION PHASE

This is the implementation phase where majority of the coding is being implemented to the system according to requirements from inception phase and designs from the elaboration phase.

The outcome of the construction phase will be a complete executable program.

META REPORT

# 5.0 GROUP STRUCTURE, ROLES, AND DESIGN ARTEFACTS

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Number : C2** | | | |
|  | **Student Number** | **Name** | **Email Address** |
| 1 |  | (Team Leader) | Tel :    Email : |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |  |

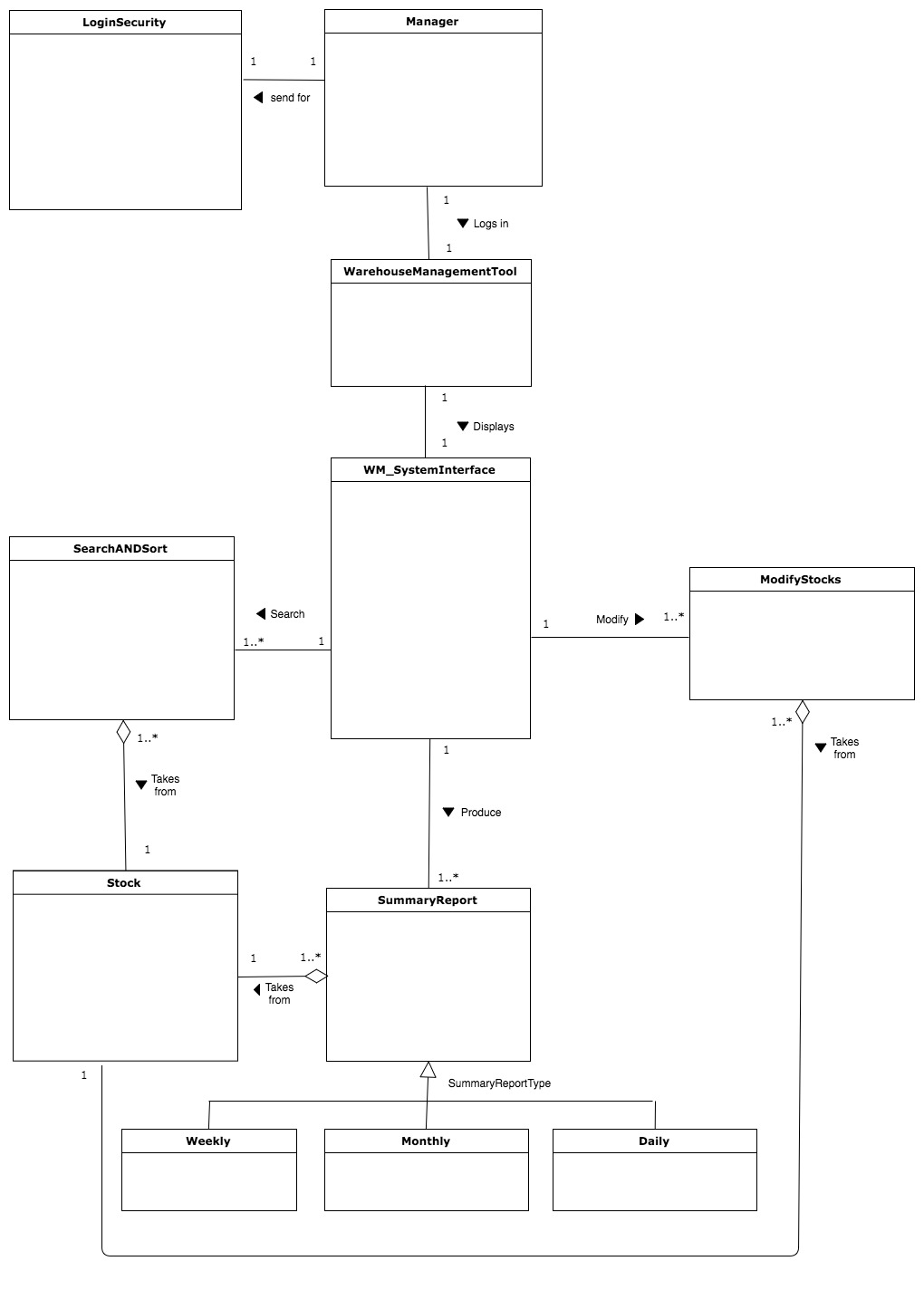
**5.1 GROUP STRUCTURE**

**5.2 ROLES AND REPSONSIBLILTIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Number : C2** | | | |
|  | **Student Name** | **Role** | **Artefacts** |
| 1 | Minton | Manager | … |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

**5.3 DESIGN ARTEFACTS**

**Design Artefacts – Domain Model**



**5.3 DESIGN ARTEFACTS**

**Design Artefacts - Use Cases (Iteration 1)**

**<USE CASE>**

**<USE CASE TEXTUAL DESCRIPTION>**

**Design Artefacts - Use Cases (Iteration 2)**

**<USE CASE>**

**<USE CASE TEXTUAL DESCRIPTION>**

**Design Artefacts - Activity Workflows (Iteration 1)**

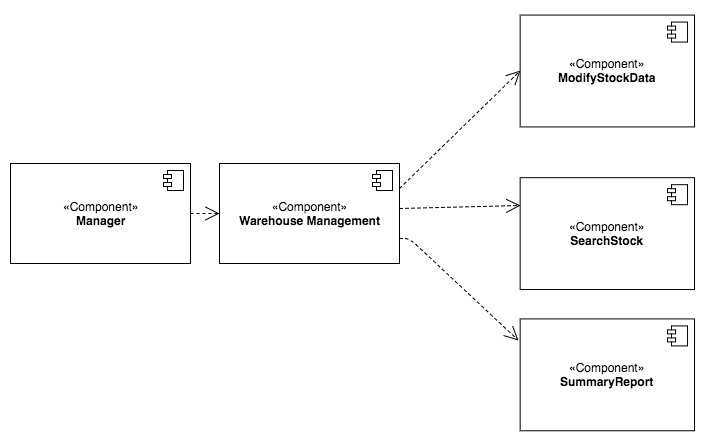
**<ACTIVIY WORK FLOW>**

**Design Artefacts - Activity Workflows (Iteration 2)**

**<ACTIVIY WORK FLOW>**

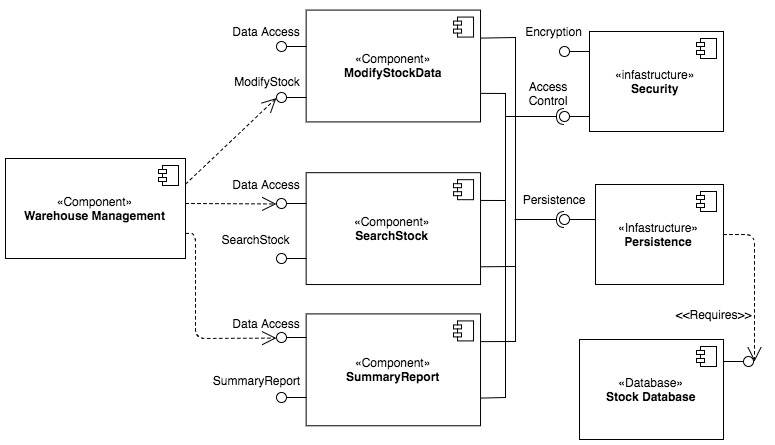
**Design Artefacts – Component Diagrams**

**(Inception phase: Iteration 1)**

****

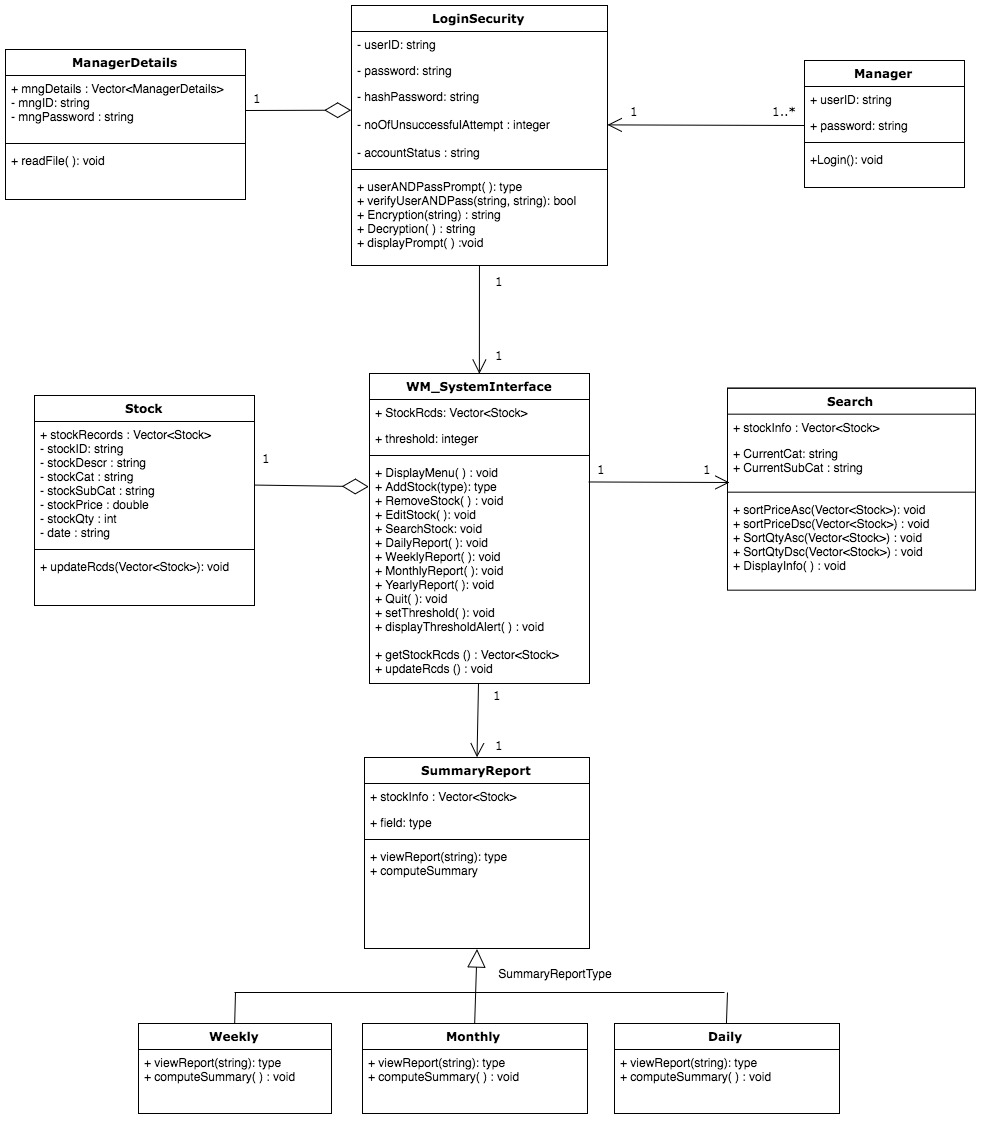
**Design Artefacts - Component Diagrams**

**(Elaboration phase: Iteration 1)**

****

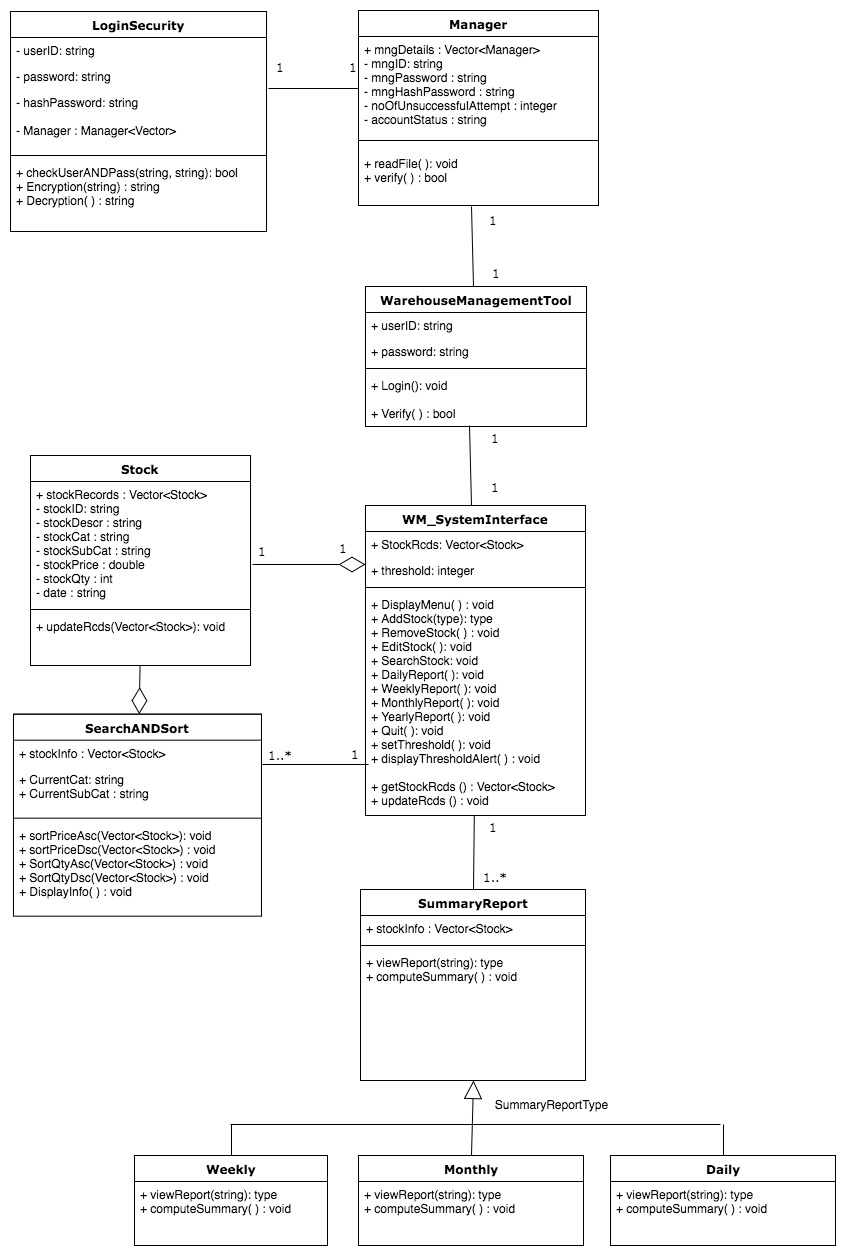
**Design Artefacts – Class Diagrams**

**(Inception phase: Iteration 1)**

****

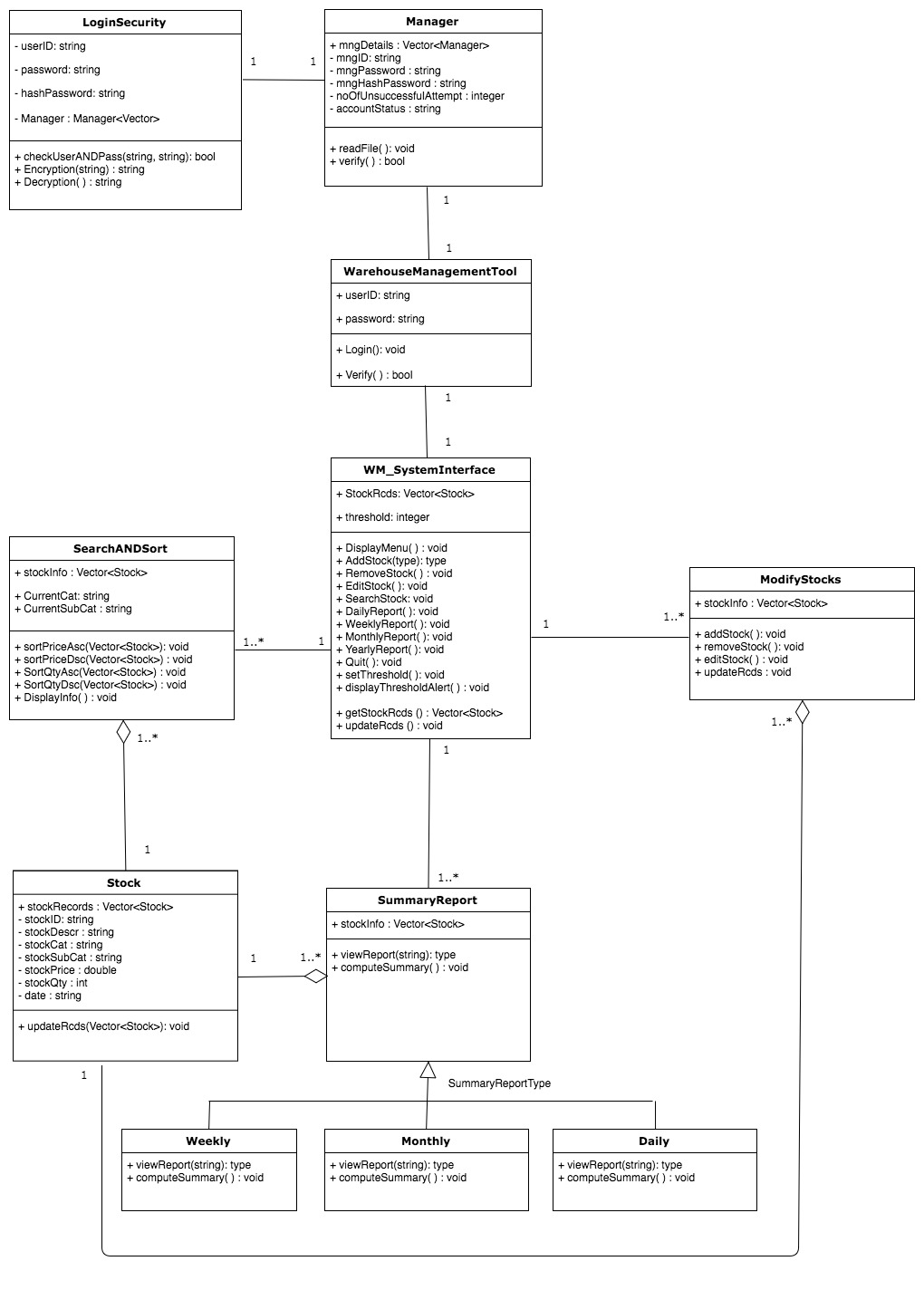
**Design Artefacts - Class Diagrams**

**(Elaboration phase: Iteration 1)**



**Design Artefacts - Class Diagrams**

**(Elaboration phase: Iteration 2)**

****

**Design Artefacts - State Diagrams (Iteration 1)**

**<State Diagrams>**

# APPENDIX A

# FORMAL MEETING RECORD

## INCEPTION PHASE – ITERATION 1

### MEETING #1 – 22 JANUARY 2018

### Summary

This is our group’s first meeting where we discuss the project structure, goals & assigned our individual roles.

We started off the project by planning our deadlines accordingly using the Gantt Chart.

As we have been familiarised with the various diagrams used through our previous lab exercise, we decided to start on the Use Case Diagram first as it will be used as the pillar of the project to ensure consistency throughout the other diagrams that we will be designing. In order to be efficient, Cyrus, Kai Xiong & Denise worked on finalising iteration 1 of the Use Case Diagram, where Sihui, Ken & Minton started on iteration 1 of the class diagram.

We concluded the meeting by discussing our goals for our 2nd meeting. Which is to finish up iteration 1 of all the diagrams and we will discuss further if any changes will be needed.

### Agenda

* + Planning project timelines
  + Delegate roles and tasks

### Report

**Urgent Items**

There were no urgent items to report in this meeting as this is the first meeting we held.

**Non-urgent Items and Other Reports**

There were no non-urgent items or other reports to report in this meeting as this is the first meeting we held.

## ELABORATION PHASE – ITERATION 1

### MEETING #2 – 25 JANUARY 2018

### Summary

In our 2nd meeting, we spent our time ensuring the consistency of all our diagrams that we have done. In order to make sure that happens, we sat down and went through each diagram one by one, voicing out our opinions and amending according to what we feel is right. We have also agreed to creating an extra diagram (state diagram) for this project which Minton will be handling.

We seek our lecturer’s advice on each of our diagrams and amended accordingly. After which, we distributed the tasks for each one of us to be done by the next meeting.

### Agenda

* + Consultation on diagrams feedback
  + Decision on extra diagram

### Report

**Urgent Items**

* State Diagram
* Detailed Use Case
* Compiling everything into our report

**Non-urgent Items and Other Reports**

--

## ELABORATION PHASE – ITERATION 2

### MEETING #3 – 9 February 2018

### Summary

* <SUMMARY HERE>

### Agenda

* + <AGENDA HERE>

### Report

**Urgent Items**

There were no urgent items to report in this meeting as this is the first meeting we held.

**Non-urgent Items and Other Reports**

There were no non-urgent items or other reports to report in this meeting as this is the first meeting we held.

# Appendix B

# WORK DIARIES

## TEAM MEMBER 1 : <INSERT NAME>

**INCEPTION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 2**

**< INSERT WORK DIARIES >**

## TEAM MEMBER 2 : Leaw Si Hui

**INCEPTION PHASE – ITERATION 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Inception phase: Iteration 1** | | | |
| **Task to Complete** | **Planned Schedule** | **Actual Work Time** | **Completed** |
| Component Diagram | 7 days | 1 day | Complete |
| Class Diagram | 7 days | 1 day | Complete |
| **Report on Defects and Countermeasures** | | | |
| **Defects** | | **Countermeasures** | |
| -- | | -- | |

**ELABORATION PHASE – ITERATION 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaboration phase: Iteration 1** | | | |
| **Task to Complete** | **Planned Schedule** | **Actual Work Time** | **Completed** |
| Component Diagram Iteration 2 | 7 days | 3 days | Complete |
| Class Diagram Iteration 2 | 7 days | 3 days | Complete |
| Final Report Format | 1 Day | 1 Day | Complete |
| **Report on Defects and Countermeasures** | | | |
| **Defects** | | **Countermeasures** | |
| -- | | -- | |

**ELABORATION PHASE – ITERATION 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Elaboration phase: Iteration 2** | | | |
| **Task to Complete** | **Planned Schedule** | **Actual Work Time** | **Completed** |
| Using version control | 1 day | 1 day | Complete |
| **Report on Defects and Countermeasures** | | | |
| **Defects** | | **Countermeasures** | |
| -- | | -- | |

## TEAM MEMBER 3 : <INSERT NAME>

**INCEPTION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 2**

**< INSERT WORK DIARIES >**

## TEAM MEMBER 4 : <INSERT NAME>

**INCEPTION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 2**

**< INSERT WORK DIARIES >**

## TEAM MEMBER 5 : <INSERT NAME>

**INCEPTION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 2**

**< INSERT WORK DIARIES >**

## TEAM MEMBER 6 : <INSERT NAME>

**INCEPTION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 1**

**< INSERT WORK DIARIES >**

**ELABORATION PHASE – ITERATION 2**

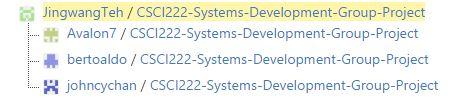
**< INSERT WORK DIARIES >**

# Appendix C

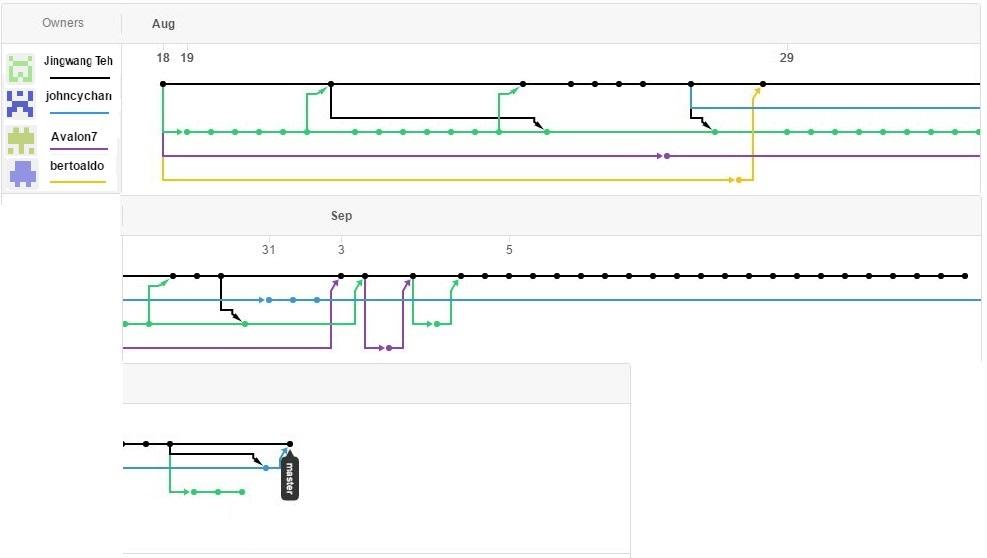
# EVIDENCE OF USING VCS

**(VERSION CONTROL SOFTWARE)**

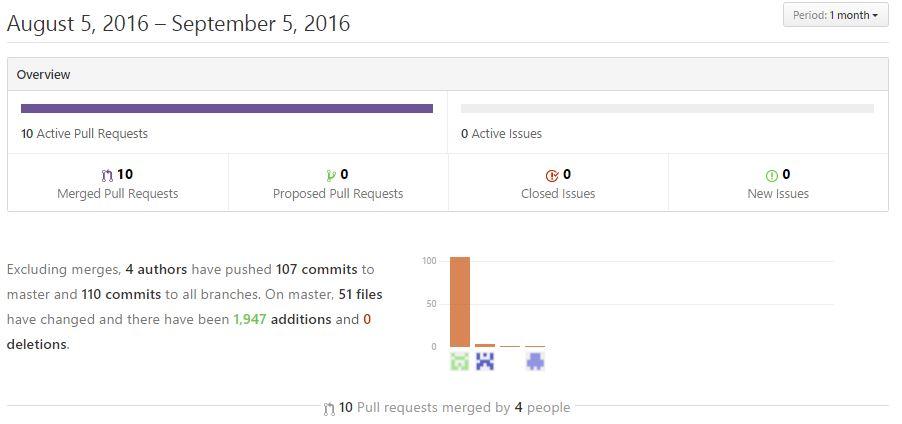
## MEMBERS



## COMMITS



## PULL REQUESTS



...