RAD V1.3

**Team name**

Team 3: Group B

**Team members**

Adam Guerin, Conor Brennan, Ryan Kelly, Mark Z. Mukiiza, William Mannix

**Date Created**

04/12/18

**Work Plan**

Table of Contents

[1. Problem Description and Current System Description 2](#_Toc531773650)

[2. Resourcing 3](#_Toc531773651)

[3.0 Requirements 3](#_Toc531773652)

[3.1 Use Case diagram 3](#_Toc531773653)

[3.2 Non functional requirements 3](#_Toc531773654)

[3.3 List of Actors 3](#_Toc531773655)

[**Floor Manager** 3](#_Toc531773656)

[**Operators** 3](#_Toc531773657)

[**Stock manager** 3](#_Toc531773658)

[3.4 List of use cases 3](#_Toc531773659)

[3.4.1 Floor Manager Use Cases 3](#_Toc531773660)

[3.4.2 Stock Manager Use Cases 3](#_Toc531773661)

[3.4.3 Operator Use Cases 3](#_Toc531773662)

[3.4.5 Common and admin use cases 3](#_Toc531773663)

[3.5 Use Case Documentation 3](#_Toc531773664)

[3.5.1 Floor Manager Use Cases 3](#_Toc531773665)

[3.5.2 Stock Manager Use Cases 3](#_Toc531773666)

# Problem Description and Current System Description

The team will be creating a desktop application that will handle all the jobs the customer requires to be done. This application will use a database to store and retrieve information. The application will also be multi-user friendly and will have a log in system for managers and operators.

The reason we are creating this application is because the customer’s current situation is that they are creating work orders and taking stock with pen and paper. Our system will make it easier to produce work-orders, keep track of stock and keep copies of work orders.

The manager receives an order from clients of a list of products needed to product. The manager writes down a list of components needed to produce assign’s the wok order to the operator.   
  
The operator uses the list of components to produce the different products needed. Once the products are produced, the operator sends the product to be quality assured by the manager.   
  
If an operator is running low or is out of a certain component, then application will notify the stock manager that this component needs to be ordered in to be replenished.

# Resourcing

We are doing just a windows form application in C#. The resources for our system are not that complicated.

**IDE:** Visual studio will be our choice for IDE because it has all the functionality that we need and also it is free and easy to access for anybody in our group.

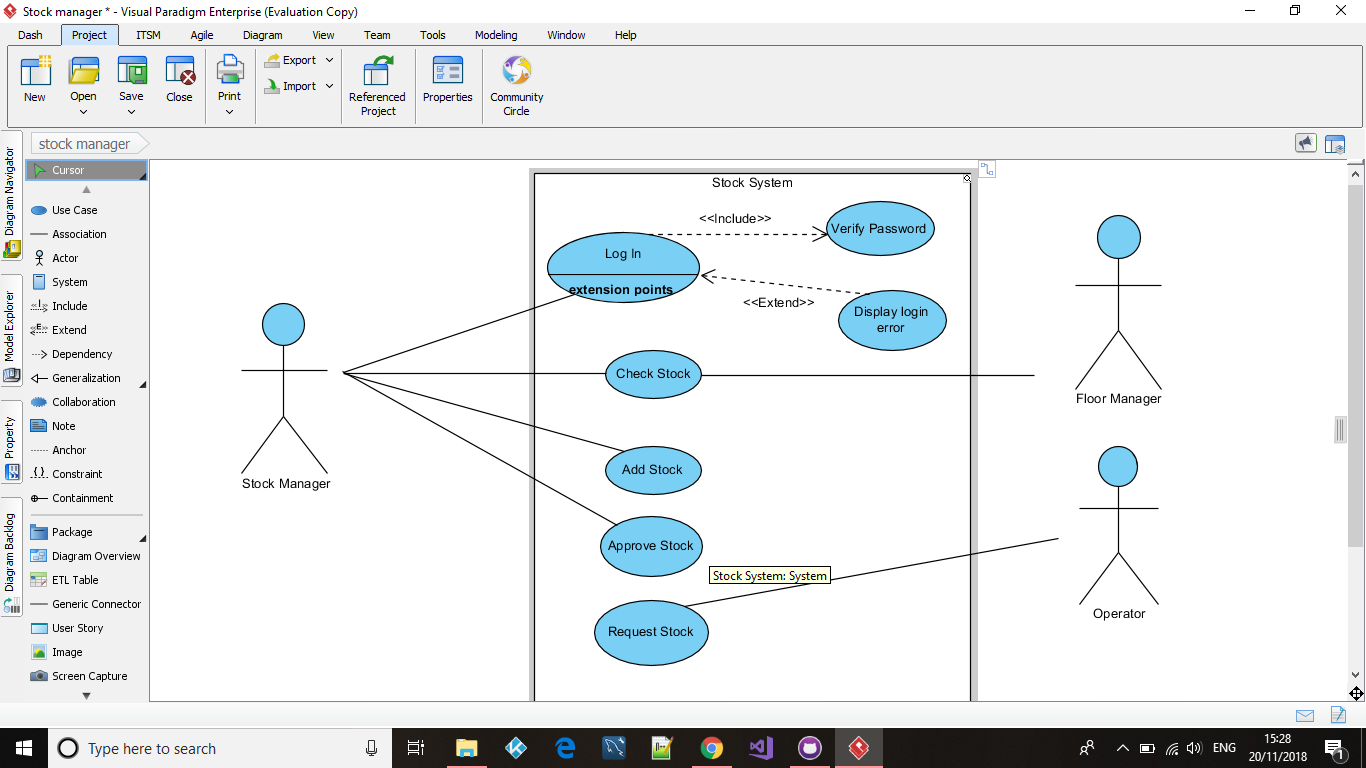
**SQL Database:** For our database we will use a combination of resources, first we will each have a local instance of a database running on xampp so that we won’t need to always be accessing a cloud database. For our cloud database we will use azure DevOps and a college instance of a database that can only be accessed inside the college.

**Source Control:** Our source control will be azures DevOps GitHub backend and using GitHub desktop to allow us to maintain solid source control.

**Xampp:** This will be used to manage a local instance of our application and our SQL database, this will only be for local productivity and will not be accessed from anywhere also only our own machine.

# Requirements

## Use Case diagram



## Non functional requirements

## List of Actors

### **Floor Manager**

The Floor manager oversees planning and execution of the work orders within the application. He/she will be able to monitor and analyse the current system and be able to keep track of stock and finished products. He/she should also have access to day to day statistics of how the product is doing and have a rough estimate of when the product is completed. When the product is completed the manager must do a check of the product to make sure that the product is up to the standards set out by the company.

### **Operators**

The operators will have the option to look at the current Work orders and decide what they will need to work on. They will also need to request new stock which the manager can then approve. The head operator of a work table then does a quality check of their progress and approves their progress of the product.

### **Stock manager**

The Stock manager is responsible for keeping track of how much stock is in the warehouse and is responsible for ordering in new stock if it is running low. The stock manager is notified on the system if components are out of stock or running low on stock.

## 3.4 List of use cases

### 3.4.1 Floor Manager Use Cases

3.4.1.1 Add Work Order P1I1 **Ryan**

3.4.1.2 Assign Work Order P1I1 **Mark**

3.4.1.3 View Employees P1I1 **Mark**

3.4.1.4 QC Check P1I1 **Will**

3.4.1.5 Daily Statistics **Will**

3.4.1.6 Sign-Out Operator P1I1 **Conor**

3.4.1.7 Write Report

3.4.1.8 Add User

3.4.1.9 Edit User

3.4.1.10 Delete User

### 3.4.2 Stock Manager Use Cases

3.4.2.1 Check Stock **Done**

3.4.2.2 Add Stock P1I1 **Ryan, Conor, Adam, Mark, Will**

3.4.2.3 Approve Stock P1I1 **Conor**

### 3.4.3 Operator Use Cases

3.4.3.1 Request Stock P1I1 **Ryan**

3.4.3.2 Check Work Order P1I1 **Ryan**

3.4.3.3 Request QC Check P1I1 **Done**

3.4.3.4 Mark as Finished P1I1 **Done**

3.4.3.5 Mark Broken Stock P1I1 **Conor**

### 3.4.5 Common and admin use cases

3.4.5.1 Log In **Done**

3.4.5.2 Log out P1I1 **Done**

## 3.5 Use Case Documentation

### 3.5.1 Floor Manager Use Cases

#### **Use Case Name:** Add Work Order

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. A new work order has been identified and its details need to be added to the system.
2. The add work order function of the system has been invoked.

**Flow of Events:**

1. A new work order is present and examined by the Floor Manager.
2. The system is accessed and responds by displaying a new window for the Floor Manager to enter the work order details and confirm if its correct.
3. The Floor Manager checks that the work order is correct and clicks ‘Confirm’ to submit the information.
4. The system responds by displaying a window with all the details of the work order and the Floor Manager to closes it.
5. The window is closed.

**Exit Conditions:**

1. The Floor Manager clicks Confirm to add the new work order to the system.
2. The Floor Manager clicks Cancel and the new work order isn’t added to the system.

#### **Use Case Name:** Assign Work Order

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to assign a work order to an Operator.
2. The assign work order function of the system has been invoked.

**Flow of Events:**

1. The system is accessed and responds by displaying the assign work order window.
2. The Floor Manager is prompted to select the name of the Operator they want to assign the work order to and then the work order they want assigned to them.
3. The system responds by displaying a window with the name of the operator and the work order they’ve been assigned and requests the Floor Manager to click ‘Ok’.
4. The Floor Manager clicks ‘Ok’.

**Exit Conditions:**

1. The Floor Manager has assigned a work order to an Operator.

#### **Use Case Name:** View Employees

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to view a list of Employees/Operators.
2. The view employee’s function of the system has been invoked.

**Flow of Events:**

1. The system is accessed and responds by displaying the Employee/Operator window.
2. The window displays a list of Employees/Operators and whether they are logged in or not.

**Exit Conditions:**

1. The Floor Manager has viewed the list of Employees/Operators.

#### **Use Case Name:** Quality Control

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to sign off on the stages of the Operators assembly.
2. The quality control function of the system has been invoked.

**Flow of Events:**

1. The system displays a notification for the Floor Manager to sign off on a stage of the Operators assembly line.
2. The Floor Manager clicks the notification.
3. The system responds by displaying a window with the notifications information. (the stage that the Operator needs signing off on) and requests the Floor Manager to click ‘Confirm’ or ‘Stop’.
4. The Floor Manager clicks ‘Confirm’.
5. The system responds by sending a notification to the Operator informing them to proceed onto the next stage of Assembly.

**Exit Conditions:**

1. The Floor Manager clicks Confirm and the Operator can proceed.
2. The Floor Manager clicks Stop and the Operator has to stop the assembly at the current stage.

#### **Use Case Name:** Daily Statistics

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The.
2. The.

**Flow of Events:**

1. The.
2. The.

**Exit Conditions:**

1. The.

#### **Use Case Name:** Sign-Out Operator

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to sign an Operator out for the day.
2. The sign-out operator function of the system has been invoked.

**Flow of Events:**

1. The system is accessed and responds by displaying the sign-out Operator window.
2. The window displays a list of Operators that are currently logged in an on the assembly line.
3. The Floor Manager clicks on the Operator he wants to sign out for the day.
4. The system responds by asking if the Floor Manager is sure they want to sign out the Operator.
5. The Floor Manager clicks ‘Confirm’
6. The Operator is signed out.

**Exit Conditions:**

1. The Floor Manager clicks ‘Confirm’ and an Operator is signed out.

#### **Use Case Name:** Write Report

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The.
2. The.

**Flow of Events:**

1. The.
2. The.

**Exit Conditions:**

1. The.

**Use Case Name:** Add New User

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager has a new employee to add to the system.
2. The Add New User function has been invoked.

**Flow of Events:**

1. The system opens up a new form showing input boxes for all of the information needed to make a new user.
2. The Floor Manager inputs all the correct data and chooses which user type the user will be, after all the data is entered the Floor Manager clicks the “Add New User” button.
3. The system then displays a message box to show if the new user has been created or not.
4. The Floor Manager then clicks “Ok”.
5. The Add new User form stays open for if the Floor Manager would like to add another user to the system.

**Exit Conditions:**

1. The Floor Manager clicks exit and the new user has been added to the system.
2. The new user can now log onto the system and will have the correct form displayed for their user type.

**Use Case Name:** Edit User

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to change a value for an employee.
2. The Edit User function has been invoked.

**Flow of Events:**

1. The System responds by opening a form with a text boxes for all the data that can be changed.
2. The Floor Manager enters all the users correct data and whichever field needs to be changed can be done while the data is being entered.
3. The Floor Manager then clicks the edit user button.
4. The System responds by displaying a text box with a message saying whether the user’s data has been changed.

**Exit Conditions:**

1. The Floor Manager clicks “Ok” on the message box and now the user’s information has been edited.
2. The Floor Manager now clicks “Exit” on this form to go back to the menu for the Floor Manager.

**Use Case Name:** Delete User

**Participating Actor**: Floor Manager

**Entry Conditions:**

1. The Floor Manager needs to delete a user from the system.
2. The Delete User Function has been invoked.

**Flow of Events:**

1. The System responds by opening a form with a text box for the user id of the user that will be deleted.
2. The Floor Manager enters the user id for the user and clicks the delete user button.
3. The System responds by showing a message box displaying whether or not the user has been deleted or not.

**Exit Conditions:**

1. The Floor Manager clicks “Ok” on the message box and now the user is deleted from the system.
2. The Floor Manager clicks “Close” on the form to bring them back to the Floor Manager menu.

### 3.5.2 Stock Manager Use Cases

#### **Use Case Name:** Add Stock

**Participating Actor**: Stock Manager

**Entry Conditions:**

1. The Stock Manager has acquired new stock that they must now update the quantity in the database with.
2. The add Stock function has been invoked.

**Flow of Events:**

1. The stock manager has received new stock.
2. The stock Manager press the “add stock” button the system.
3. The list of components has been displayed on screen with an empty box where the number of stock received can be added to.
4. The stock manger highlights the component and enters the amount of stock received into the box and presses the “add stock” button.
5. The record updates and a message is displayed saying “stock updated”.

**Exit Conditions:**

1. The stock manager pressed “ok” on the message.
2. The stock manger presses the back button to go back to the main window.

### Operator Use Cases

### Common Use Cases

**Use Case Name:** Log in

**Participating Actor**: All Actors

**Entry Conditions:**

1. The actor had opened up the program for the first time (in this session).
2. The log in function of the program has been invoked.

**Flow of Events:**

1. The system opens a page with the MedEx logo and 3 form inputs, username, password and a submit button.
2. The actor responds by entering their correct data into the fields and clicking “Log In”.
3. The system responds by checking the actor’s details against a list of user credentials stored on the database.

**Exit Conditions:**

1. The System has let the actor through and now has opened up the desired form with the correct list of use cases for the user.

**Use Case Name:** Log out

**Participating Actor**: All Actors

**Entry Conditions:**

1. The actor is done their work and clicks the log out button on their screen to log out for the day.
2. The Log out function of the system has been invoked.

**Flow of Events:**

1. The System responds by changing the state of the user in the database to logged out and closing the form of the user.

**Exit Conditions:**

1. The actor is now logged out and the system invoked the “Log In” function of the program.