RAD V1.1

**Team name**

**Team members**

Adam Guerin, Conor Brennan, Ryan Kelly, William Mannix

**Date Created**

9/11/18

**Work Plan**

Table of Contents

[1. Problem Description and Current System Description 2](#_Toc530498802)

[2. Resourcing 3](#_Toc530498803)

[3.0 Requirements 3](#_Toc530498804)

[3.1 Use Case diagram 3](#_Toc530498805)

[3.2 Non functional requirements 4](#_Toc530498806)

[3.3 List of Actors 4](#_Toc530498807)

[3.4 List of use cases 5](#_Toc530498808)

[3.4.1 Log in 5](#_Toc530498809)

[3.4.1.1 Verify password 5](#_Toc530498810)

[3.4.1.2 Log in error 5](#_Toc530498811)

# 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.

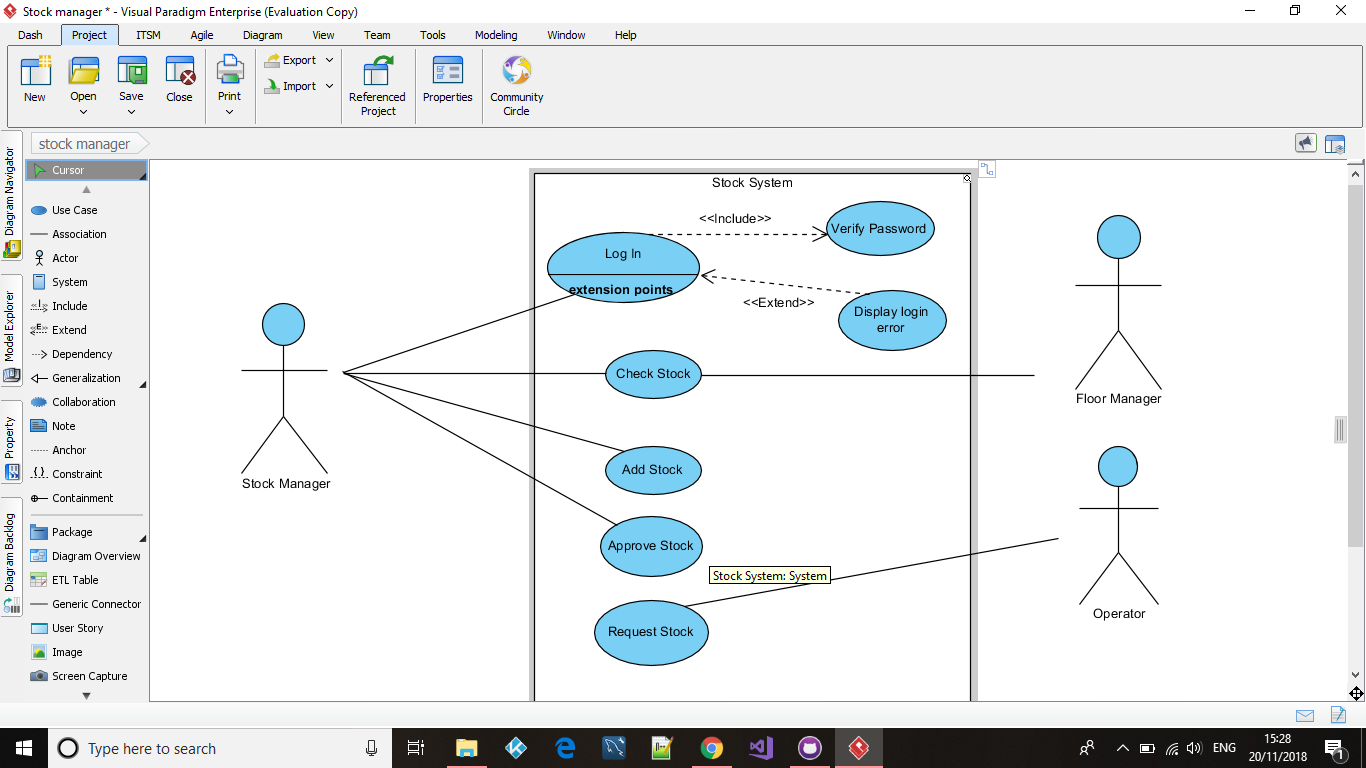
**SQL Database:** For our database we will use a combination of resources, first we will each have a local instance of a database running on xampp so that we won’t need to always be accessing a cloud database. For our cloud database we will use azure DevOps and a college instance of a database that can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

**Operational Manager**

The operational 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.

# List of use cases

# Log in

# Verify password

# Log in error

* + 1. Manage stock
       1. Request stock
       2. Add stock
       3. Approve stock
    2. View orders
       1. Request stock
       2. Complete order
    3. Common and admin use cases

Common use case  
The operator will only be given common privileges which include being able to view the product that is to be made and view the list of components needed to produce the product. The Operator will also be able to request new components to the stock manager if stock is running low.

Admin use case  
The floor manager and the stock manager will be given admin privileges. This will include being able to order new stock and quality approve products.

# 