**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](#_Toc550636)

[2. Resourcing 3](#_Toc550637)

[3.0 Requirements 4](#_Toc550638)

[3.1 Use Case diagram 4](#_Toc550639)

[3.2 Non functional requirements 4](#_Toc550640)

[3.3 List of Actors 5](#_Toc550641)

[**Floor Manager** 5](#_Toc550642)

[**Operators** 5](#_Toc550643)

[**Stock manager** 5](#_Toc550644)

[3.4 List of use cases 6](#_Toc550645)

[3.4.1 Floor Manager Use Cases 6](#_Toc550646)

[3.4.2 Stock Manager Use Cases 6](#_Toc550647)

[3.4.3 Operator Use Cases 6](#_Toc550648)

[3.4.5 Common and admin use cases 6](#_Toc550649)

[3.5 Use Case Documentation 7](#_Toc550650)

[3.5.1 Floor Manager Use Cases 7](#_Toc550651)

[3.5.2 Stock Manager Use Cases 20](#_Toc550652)

[3.5.3 Operator Use Cases 23](#_Toc550653)

[3.5.3 Common and Admin Use Cases 27](#_Toc550654)

# 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.

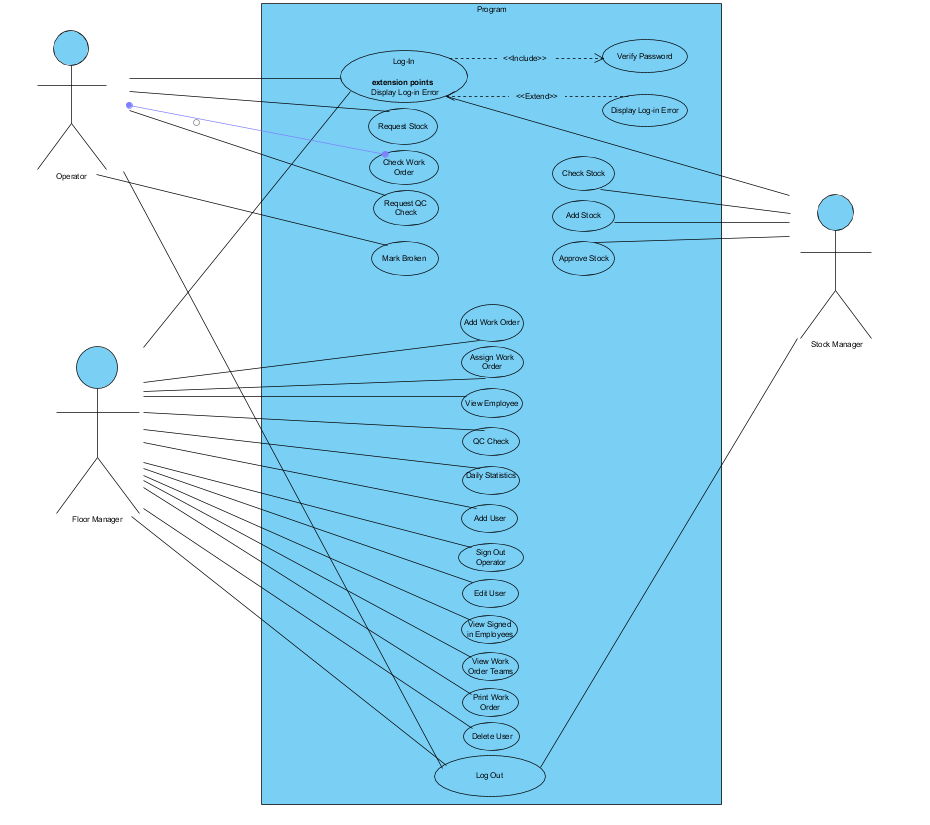
**SQL Database:** For our database we will use a combination of resources, first we will each have a local instance of a database running on xampp so that we won’t need to always be accessing a cloud database. For our cloud database we will use azure DevOps and a college instance of a database that can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

3.4.1.7 Add User **Adam**

3.4.1.8 Edit User **Adam**

3.4.1.9 View Signed-in Employees **Ryan**

3.4.1.10 View Work Order Teams **Ryan**

3.4.1.11 Print Work Order **Ryan**

3.4.1.12 Delete User **Adam**

### 3.4.2 Stock Manager Use Cases

3.4.2.1 Check Stock **Ryan**

3.4.2.2 Add Stock P1I1 **Conor**

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**Adam**

3.4.3.4 Mark Broken Stock P1I1 **Conor**

### 3.4.5 Common and admin use cases

3.4.5.1 Log In **Adam**

3.4.5.2 Log out P1I1 **Adam**

## 3.5 Use Case Documentation

### 3.5.1 Floor Manager Use Cases

**Use Case Name:** Add Work Order

**Participating Actor:** Floor Manager

**Entry Conditions:**

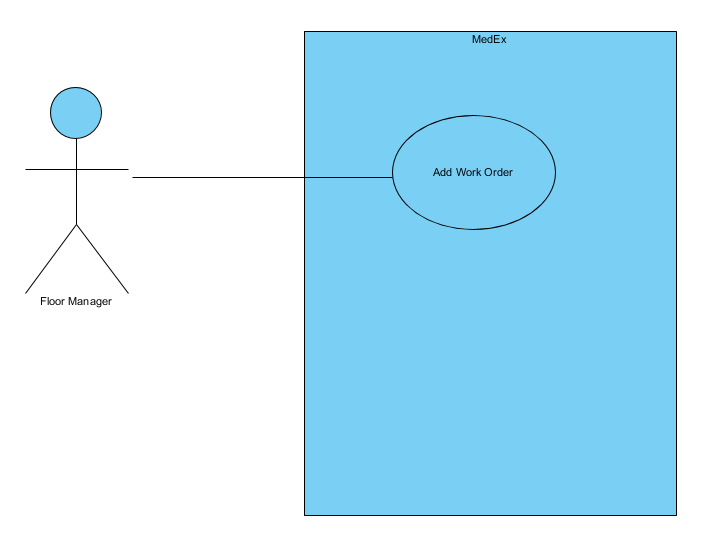
1. The floor manager has received a quote for a Product.
2. The Floor manager has successfully logged in and is displayed the Floor Manager menu.
3. The Add Work Order function has been invoked.

**Flow of Events:**

1. The Floor Manager clicks the ‘Add Work Order’ button.
2. The application displays a new window for the Floor Manager to add in the details of the work order.
3. The Work order number has been pre-set.
4. The Floor Manager Enters:
   1. Entry Date
   2. Print Date
   3. Quantity
   4. Description
   5. Signature
   6. Signature Date
   7. Employee Number
5. The Floor Manager Selects a Component too add to the work order from the list.
6. The Floor Manager clicks ‘**>’** button.
7. The application opens a new window with a textbox and a ‘Submit’ button.
8. The Floor Manager enters in the quantity of the component selected needed and clicks the ‘Submit’ button.
9. The system closes the Window and adds the component information into the list for the Work Order.
10. The floor manager may click the ‘**<**’ button which will remove the component from the work order.
11. The Floor Manager Clicks the submit button.
12. The application Adds the Work Order and the components linked to it to the system.
13. The System notifies the Floor manager if the Work order has been suseccfully added.
14. The Floor Manager clicks the ‘Exit’ button.

**Exit Conditions:**

1. The floor manager has clicked the ‘Exit’ button.
2. The ‘Add work order’ window closes and the user is displayed the Floor manager screen.



**Use Case Name:** QC check

**Participating Actor**: Floor Manager

**Entry Conditions:**

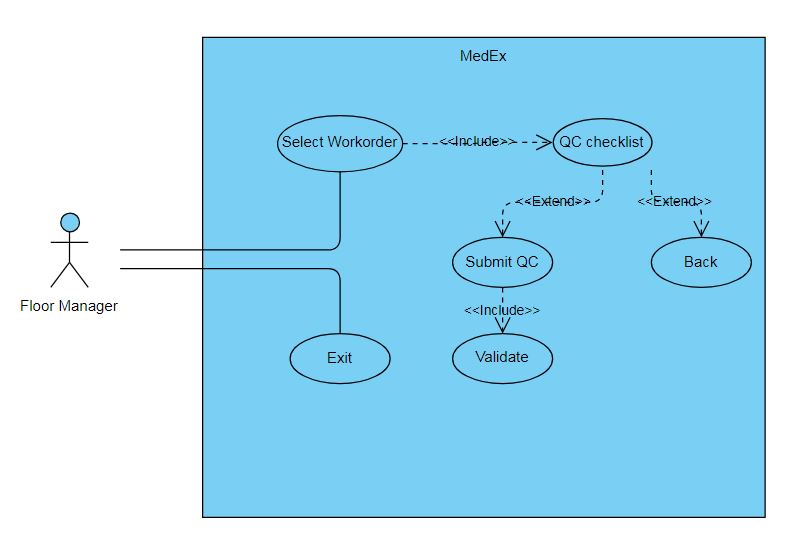
1. The Floor manager has logged in successfully and is displayed the Floor manager menu.
2. Work orders are flagged for a quality control check.
3. The QC check function is invoked.

**Flow of Events:**

1. The QC check form is displayed by the system with a list of work orders flagged for QC
2. The Floor manager selects a work order that they want to test for quality control
3. The system opens a checklist which the floor manager can check off boxes for whether or not the work order product passes the QC check tests.
4. The Floor manager submits the QC checklist.
5. The system takes the floor manager back to the list of work orders to be checked.

**Exit Conditions:**

1. The Floor Manager clicks the exit button and the form closes.



**Use Case Name:** Daily statistics

**Participating Actor**: Floor Manager

**Entry Conditions:**

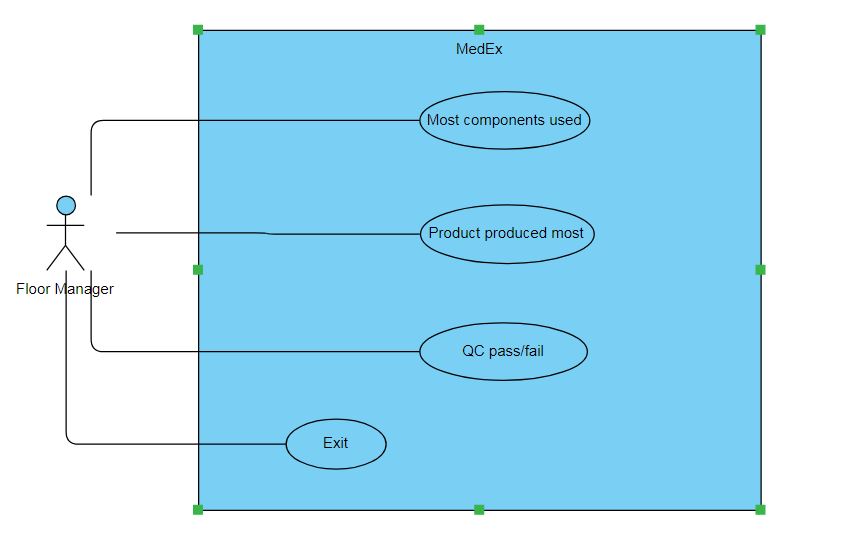
1. The Floor manager has logged in successfully and is displayed the Floor manager menu.
2. The Floor Manager wants to check the production of the employees and stock usage.
3. The Daily statistics function of the system has been invoked.

**Flow of Events:**

1. The system displays a form with three statistic options which the floor manager can choose from.
2. The Floor Manager selects an option
3. A chart opens displaying the statistics of the particular option.

**Exit Conditions:**

1. The Floor Manager presses the exit button and the form closes.



**Use Case Name:** Assign Work Order

**Participating Actor:** Floor Manager

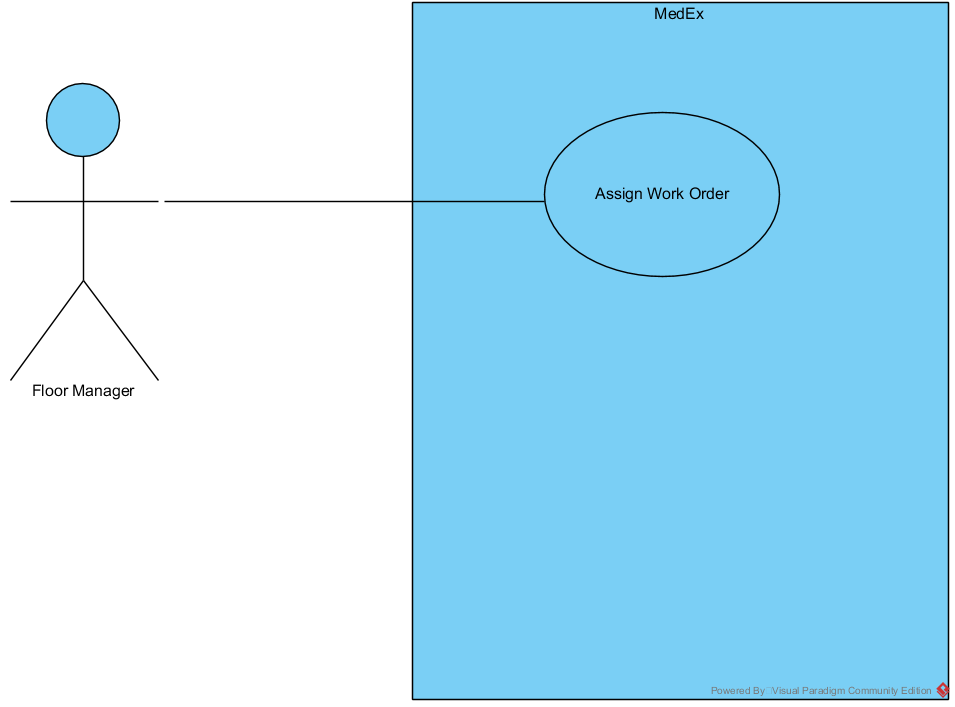
**Entry Conditions:**

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 Floor Manager clicks the ‘Assign Work Order’ button.
2. The system is accessed and responds by displaying the assign work order window.
3. The Floor Manager is presented with a list of available Operators and All Work Orders.
4. The Floor Manager can then select the name of the Operator they want to assign the work order to and then the work order they want assigned to them.
5. The Floor Manager can either click ‘Exit’ to close the window or click ‘Assign’ to assign the work order.
6. If the Floor Manager clicked ‘Assign’ the system responds by displaying a window stating that an Operator has been assigned and requests the Floor Manager to click ‘Ok’.
7. The Floor Manager clicks ‘Ok’.

**Exit Conditions:**

1. The Floor Manager clicks ‘Exit’.
2. Floor Manager has assigned a work order to an Operator.

**Use Case Name:** View Employees

**Participating Actor:** Floor Manager

**Entry Conditions:**

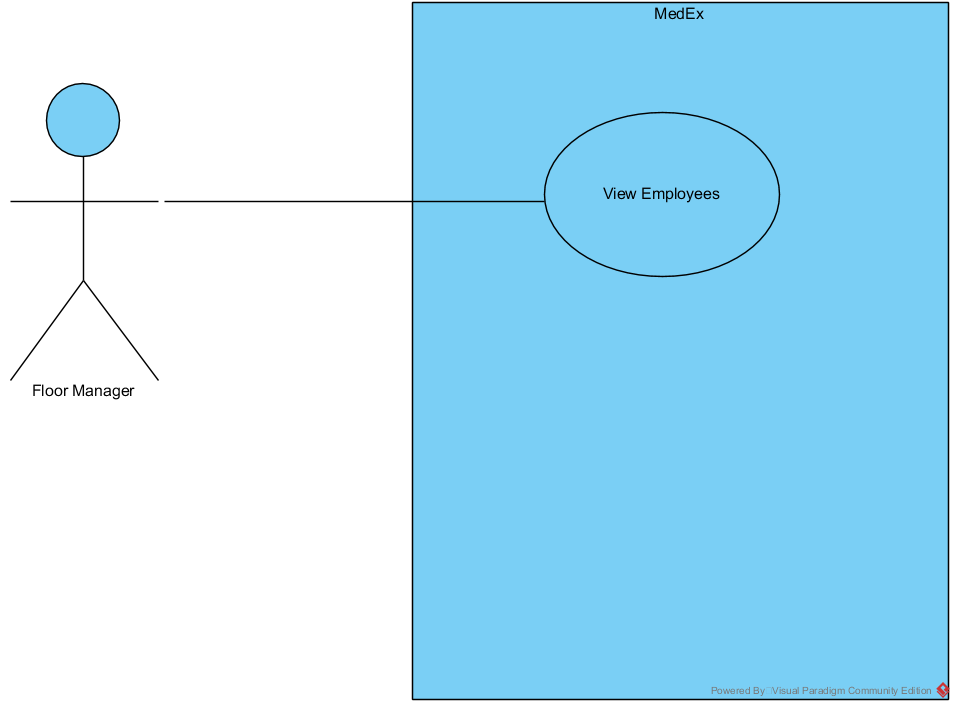
1. The Floor Manager needs to view a list of all the MedEx Employees.
2. The View Employee’s function of the system has been invoked.

**Flow of Events:**

1. The Floor Manager clicks the ‘View Employees’ button.
2. The system is accessed and responds by displaying the View Employee window.
3. The Window displays a list of Employees and their details such as Employee Number, Position, Full Name and Contact Number.
4. The Floor Manager clicks ‘Exit’.

**Exit Conditions:**

1. The Floor Manager has viewed the list of all the Employees and clicks ‘Exit’.

****

**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 must 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:**

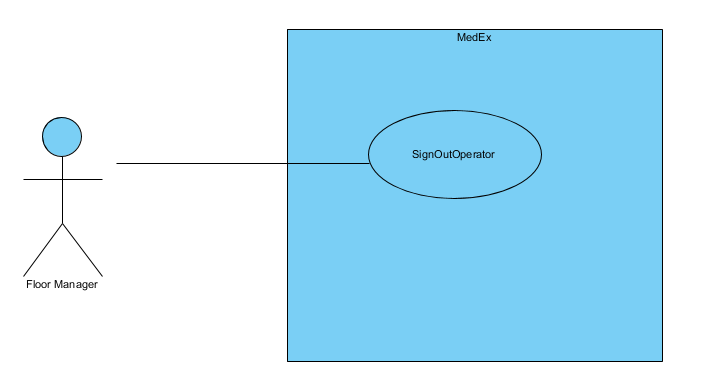
1. An employee has asked to leave work.
2. The Floor manager wants to sign out the operator on what work order they are on.

**Flow of Events:**

1. The Floor manager presses the “sign out operator” button on screen.
2. The application open a new window displaying the list of work orders with the associated employee.
3. The Floor manager clicks on the work order the employee leaving is associated with and then clicks “sign out”:
4. The employee is taking out of the list.

**Exit Conditions:**

1. The Floor manager has clicked ‘Exit’ button.
2. The ‘Sign out operator’ window closes, and the user is displayed the Floor Manager menu screen.



**Use Case Name:** View Signed-in Employees

**Participating Actor:** Floor Manager

**Entry Conditions:**

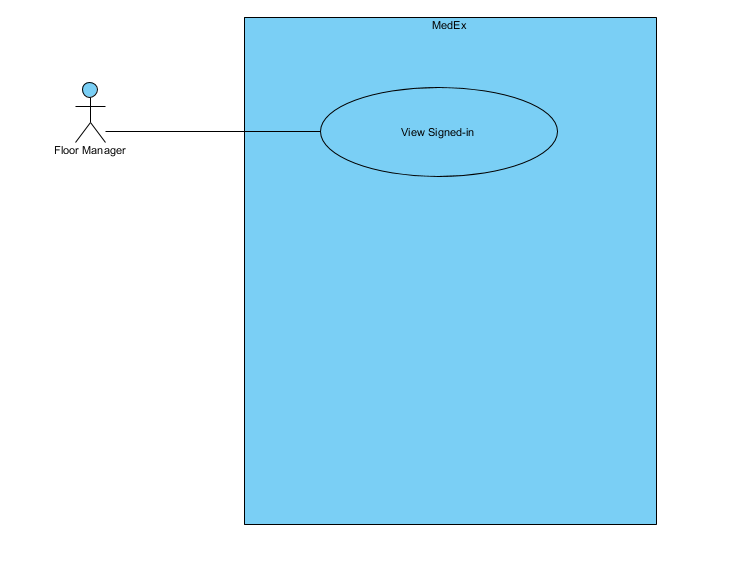
1. The Floor Manager has successfully logged in and is displayed the operator menu.
2. The ‘View Signed-in Employees Function has been invoked.

**Flow of Events:**

1. The floor Manager clicks the ‘View Signed-in Employees’ button.
2. The Application responds by displaying a new window which displays the ‘View SignedIn’ window.
3. The floor manager is displayed a list of Employees that are logged into the system.
4. The list displays the Employees name, position and Work order assigned.
5. The Floor manager clicks the ‘Exit’ Button.

**Exit Conditions:**

1. The Floor Manager has clicked the ‘Exit’ button.
2. The ‘View SignedIn’ window closes, and the user is displayed the Floor Manager Screen.



**Use Case Name:** View Work Order Teams

**Participating Actor:** Floor Manager

**Entry Conditions:**

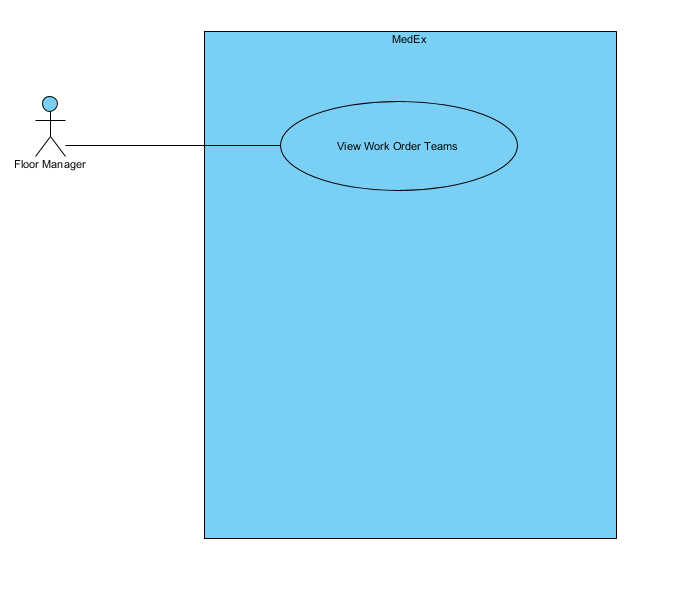
1. The Floor Manager has successfully logged in and is displayed the Floor Manager menu.
2. The ‘View Work Order Teams’ Function has been invoked.

**Flow of Events:**

1. The Floor Manager clicks the ‘View Work Order Teams’ button.
2. The application responds by displaying a new window which displays the ‘Work Order Teams’ Window.
3. The Floor Manager is displayed a list Box with the list of Work Order Numbers.
4. The Floor Manager selects a Work Order.
5. The Employee ID and name of the Employee That is working on that work order is Displayed in the list View box.
6. The Floor Manager Clicks the ‘Exit’ button.

**Exit Conditions:**

1. The Floor Manager has clicked the ‘Exit’ button.
2. The ‘Work Order Teams’ window closes and the user is displayed the Floor Manager Screen.



**Use Case Name:** Add New User

**Participating Actor**: Floor Manager

**Entry Conditions:**

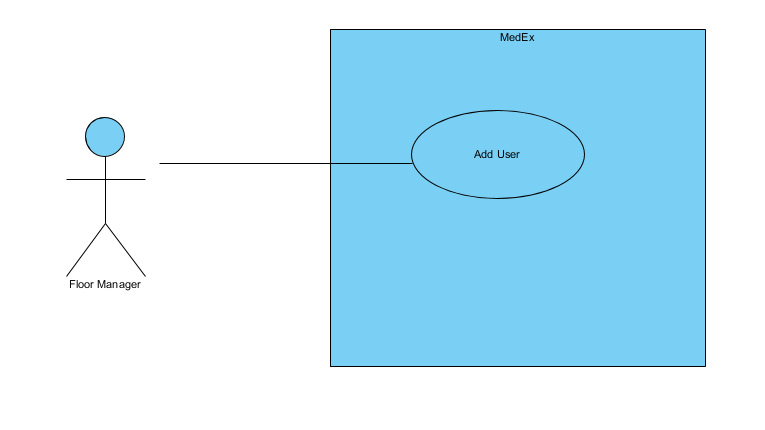
1. The Floor Manager has a new employee to add to the system.
2. The Add New User function has been invoked.

**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:** Delete User

**Participating Actor**: Floor Manager

**Entry Conditions:**

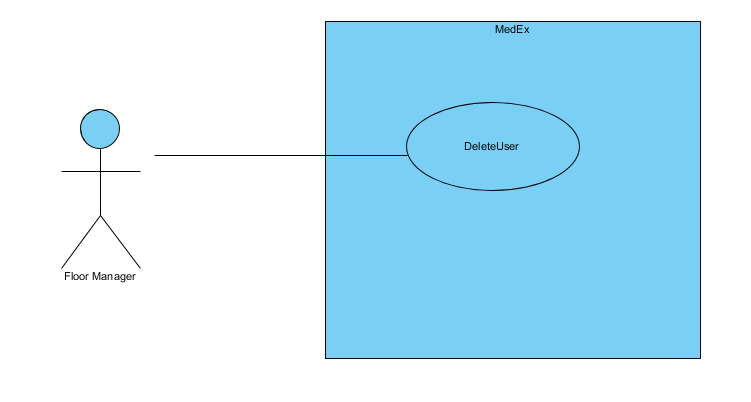
1. The Floor Manager needs to delete a user from the system.
2. The Delete User Function has been invoked.

**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.



**Use Case Name:** Edit User

**Participating Actor**: Floor Manager

**Entry Conditions:**

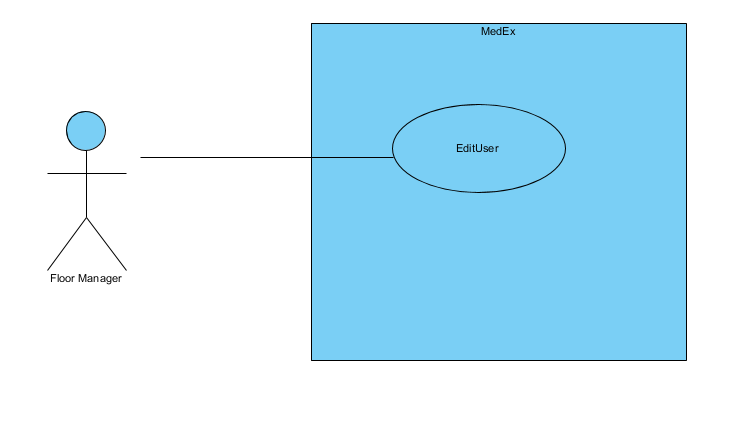
1. The Floor Manager needs to change a value for an employee.
2. The Edit User function has been invoked.

**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:** Print Work Order

**Participating Actor:** Floor Manager

**Entry Conditions:**

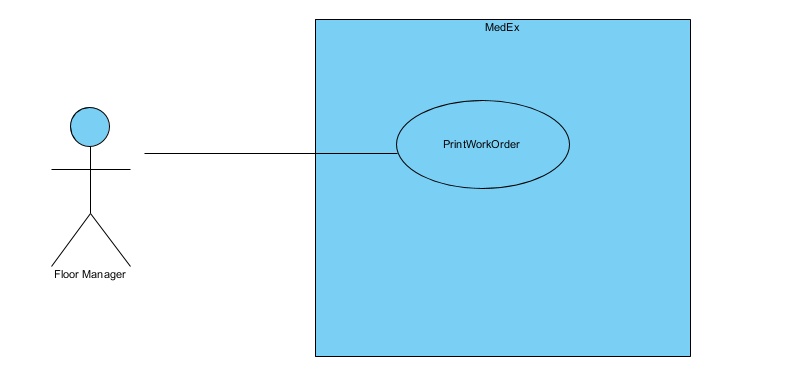
1. The Floor Manager has successfully logged in and is displayed the Floor Manager menu.
2. The ‘Print Work Order’ Function has been invoked.

**Flow of Events:**

1. The Floor Manager clicks the ‘Print Work Order’ button.
2. The application responds by displaying a new window which displays the ‘Print Work Order’ Window.
3. The Floor manager is displayed a window with a list of work orders.
4. The Floor manager selects a work order from the list and clicks print.
5. The window closes and a new window opens with the details of the Work order they have selected.
6. The Floor manager clicks ‘print’.
7. The window closes and the program takes a screen capture of the work order.
8. The program opens the screen capture in photo viewer for the manager to print.
9. The floor manager clicks the ‘Exit’ button.

**Exit Conditions:**

1. The Floor Manager has clicked the ‘Exit’ button.
2. The ‘Print Work Order’ window closes and the user is displayed the Floor Manager Screen.



### 3.5.2 Stock Manager Use Cases

**Use Case Name:** Add Stock

**Participating Actor:** Stock Manager

**Entry Conditions:**

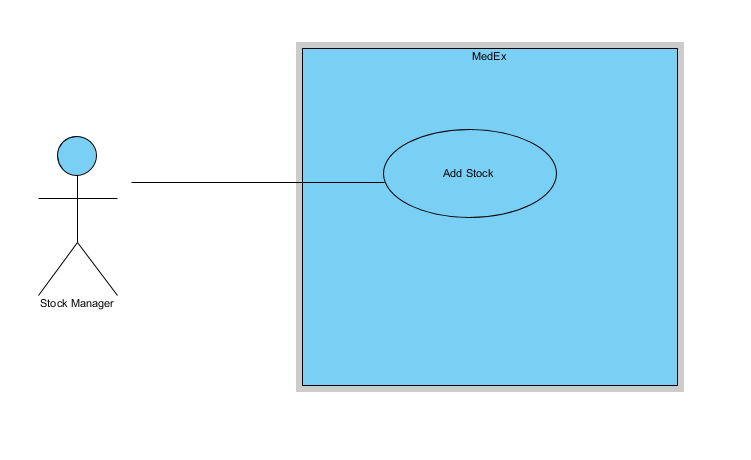
1. The Stock Manager has logged in successfully and is displayed the Stock Manager menu.
2. The Add Stock function has been invoked.

**Flow of Events:**

1. The Stock Manager presses the ‘Add Stock button.
2. The application opens a new window displaying the text boxes for the name of the component and the quantity display with a randomly generate number for the batch number.
3. The Stock Manager can click on the text boxes and enter the correct data.
4. The Stock Manager then clicks “Enter”.

**Exit Conditions:**

1. The Stock Manager has clicked ‘Exit’ button.
2. The ‘Add Stock’ window closes, and the user is displayed the Stock Manager menu screen.



**Use Case Name:** Check Stock

**Participating Actor:** Stock Manager

**Entry Conditions:**

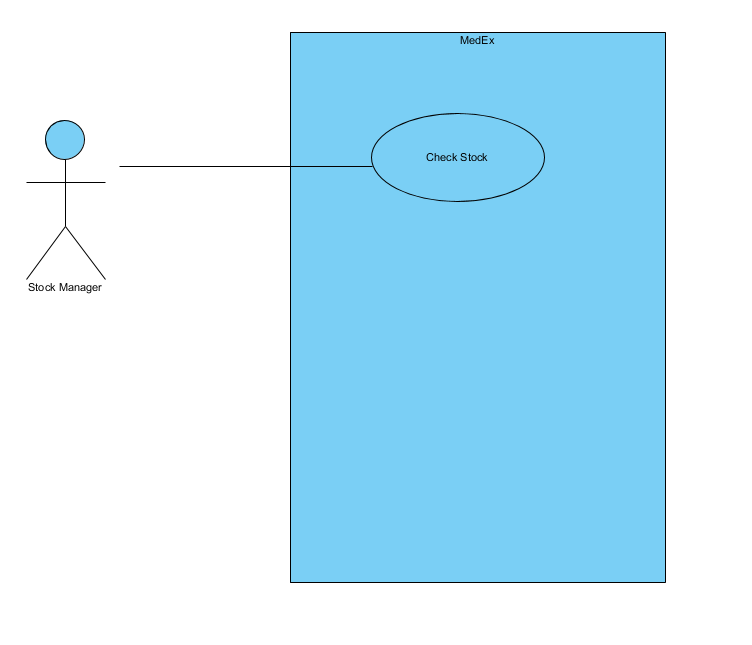
1. The Stock manager has logged in successfully and is displayed the stock manager menu.
2. The Check Stock function has been invoked.

**Flow of Events:**

1. The Stock manager presses the ‘Check Stock’ button.
2. The application open a new window displaying the stock in the system and a button called ‘refresh’.
3. The Stock manager can either:
   1. Clicks the ‘refresh button’ which re-calls the data from the database and re-displays it to the screen.
   2. Clicks the ‘exit’ button.

**Exit Conditions:**

1. The stock manager has clicked ‘Exit’ button.
2. The ‘Check Stock’ window closes and the user is displayed the Stock Manager menu screen.



**Use Case Name:** Approve Stock

**Participating Actor:** Stock Manager

**Entry Conditions:**

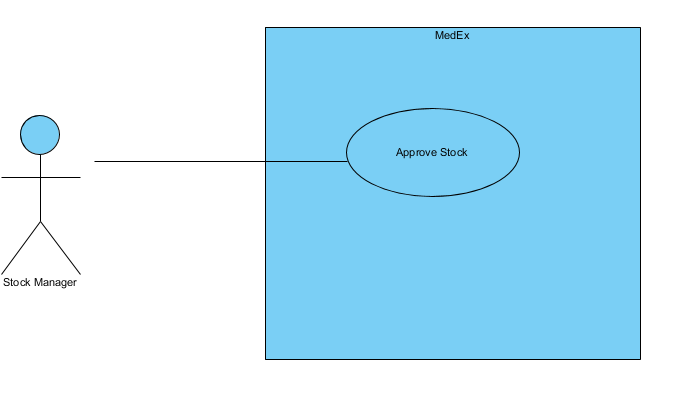
1. The Stock Manager has logged in successfully and is displayed the Stock Manager menu.
2. The Approve Stock function has been invoked.

**Flow of Events:**

1. The Stock Manager presses the ‘Approve Stock button.
2. The application opens a new window displaying a list box of the requests are displayed.
3. The Stock Manager can click on one of the requests and the next list box is displayed with the components to that request.
4. The Stock Manager then clicks on a component name and the quantity of this component is displayed in a text box where the Stock Manager can choose to change the quantity requested or not.
5. The Stock Manager Clicks ‘Approve All’ button.

**Exit Conditions:**

1. The Stock Manager has clicked ‘Exit’ button.
2. The ‘Approve Stock’ window closes, and the user is displayed the Stock Manager menu screen.



### 3.5.3 Operator Use Cases

**Use Case Name:** Request Stock

**Participating Actor:** Operator

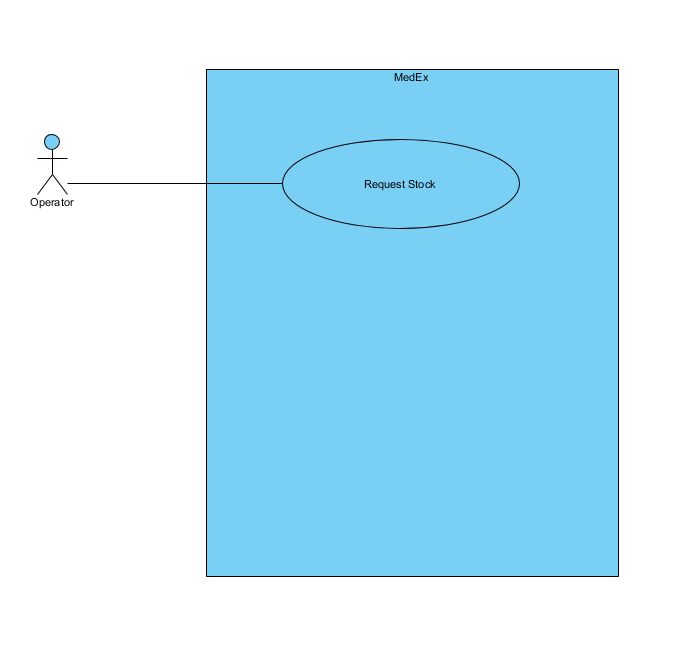
**Entry Conditions:**

1. The Operator has been assigned a Work Order
2. The Operator has successfully logged in and is displayed the operator menu.
3. The Request Stock Function has been invoked

**Flow of Events:**

1. The operator selects ‘Request Stock’ button.
2. The application responds by displaying a new window which displays the Request Stock window
3. The Operator is Displayed a list Box with the component names linked to their work order.
4. The Operator selects a component name in the list Box and the Quantity needed box is Filled.
5. The Operator enters the Quantity needed for their shift.
6. The Operator clicks Submit Request.
7. The Component Name and Amount is added to The list View box.
8. The Operator may repeat Steps 4 – 6.
9. The Operator selects Submit.
10. The data is sent to the Database.
11. A message displays saying the request is submitted.
12. The Operator Clicks the ‘Exit’ button.

**Exit Conditions:**

1. The Operator has clicked the ‘Exit’ button.
2. The ‘Request Stock’ window closes and the user is displayed the Operator Screen

**Use Case Name:** Check Work Order

**Participating Actor:** Operator

**Entry Conditions:**

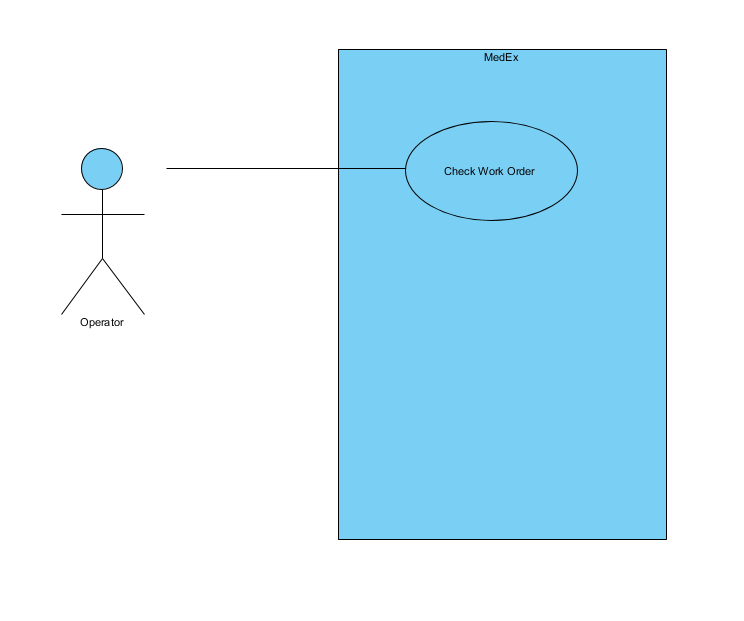
1. The Operator has been assigned to a Work Order.
2. The Operator has successfully logged in and is displayed the operator menu.
3. The Check Work Order Function has been invoked.

**Flow of Events:**

1. The Operator clicks the ‘Check Work Order’ button.
2. The application responds by displaying a new window which displays the work order details.
3. The Operator Clicks the ‘Exit’ button.

**Exit Conditions:**

1. The Operator has clicked the ‘Exit’ button.
2. The ‘Check Work Order’ window closes and the user is displayed the Operator Screen



**Use Case Name:** Mark Stock as Broken

**Participating Actor:** Operator

**Entry Conditions:**

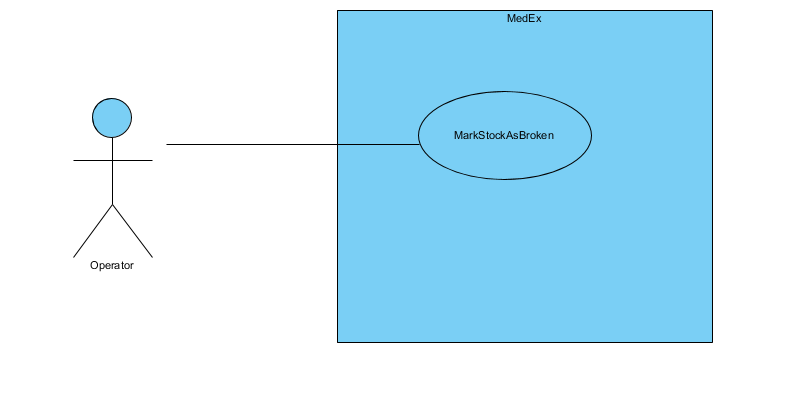
1. The Operator has logged in successfully and is displayed the Operator menu.
2. The Mark Stock as Broken function has been invoked.

**Flow of Events:**

1. The Operator presses the ‘Mark Stock as Broken’ button.
2. The application opens a new window displaying the components associated with the work order the operator is on.
3. The Operator can click on the name of the part that has broken and type in to the text box how many parts have broken.
4. The operator then clicks “Enter”.

**Exit Conditions:**

1. The Operator has clicked ‘Exit’ button.
2. The ‘Mark Stock as Broken’ window closes, and the user is displayed the Operator menu screen.



**Use Case Name:** Request QC check

**Participating Actor**: Operator

**Entry Conditions:**

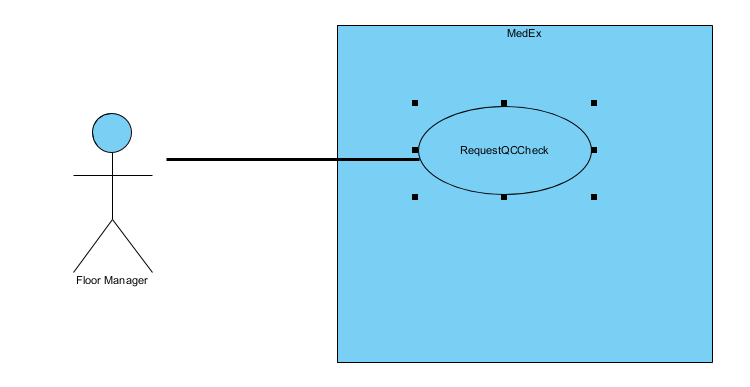
1. The Operator needs to have the product he just created checked by the manager.
2. The Request QC Check function of the system has been invoked.

**Flow of Events:**

1. The System opens a form of all the work orders that have been completed but have not been checked yet.
2. The Operator chooses the work order he/she just completed and clicks the Request QC Check button.
3. The system responds by displaying a message box saying a QC check has been quested for this work order.

**Exit Conditions:**

1. The Work order has now been marked to be checked by the Floor Manager and the Floor manager will get a notification that he/she will need to perform a QC check on this workorder.
2. The Operator clicks close on the form and continues on with his/her work.



### Common and Admin Use Cases

**Use Case Name:** Log in

**Participating Actor**: All Actors

**Entry Conditions:**

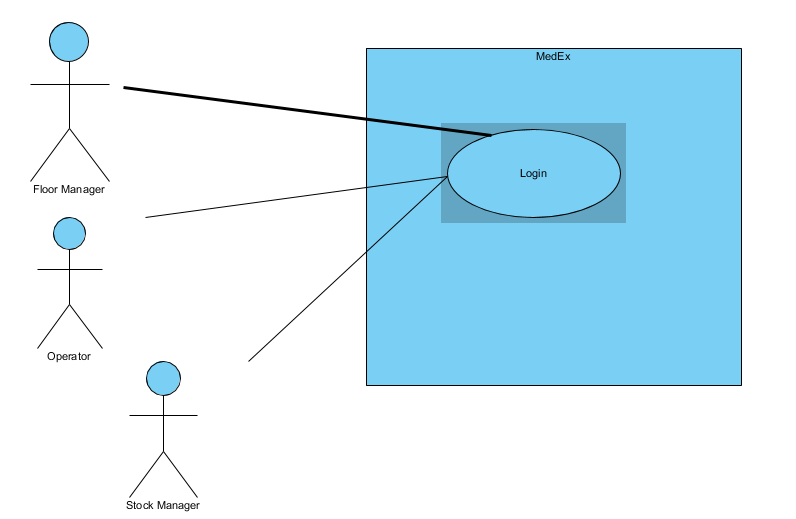
1. The actor had opened up the program for the first time (in this session).
2. The log in function of the program has been invoked.

**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.

