FMS

Name & Class ID : Murali Krishna Sai Chukka & 09

Name & Class ID : Rupesh Sai Ram Doddala & 10

Name & Class ID : Kavin Kumar Arumugam & 63

Name & Class ID : Rahul Reddy Yerva & 59

Name & Class ID : Tejaswi Ayyadapu & 03

**Table of Contents:**

1. Domain. 3
2. Project goal and objective. 3
   1. Motivation. 3
   2. Significance. 3
   3. Objective. 3
3. Architectural diagram. 4
4. Modules List. 5
   1. Login. 6
   2. User. 8
   3. User Group. 10
   4. Machine. 12
   5. Machine Group. 15
   6. Regular Maintenance. 17
   7. Area. 19
   8. Line. 21
   9. Reasons. 23
   10. Documents. 25
   11. Parts. 28
   12. Step group. 30
   13. Template. 34
   14. Settings. 37
   15. Import. 39
5. Mobile pages. 44
6. Source code and Video URLS. 48
7. References. 48

**Domain :**

## **MES (Manufacturing Execution System).**

## **Project Goal and Objectives :**

Motivation :

In many of the Manufacturing industries, a product is manufactured either by man power or by machinery or both sometimes. Once I had an opportunity to visit manufacturing unit in my country, Where the major part of the work is done by the equipment’s. There was a situation where any fault in the equipment, the maintenance is done manually. For Instance, if there was any failure in the equipment, the supervisor informs the maintenance team via phone/email. Then the maintenance team has a checklist which has steps for defect resolutions, which is time consuming process and un-trackable. This has triggered me the idea, which automates the whole process of maintenance system, which is called Factory Maintenance System(FMS).

**Significance :**

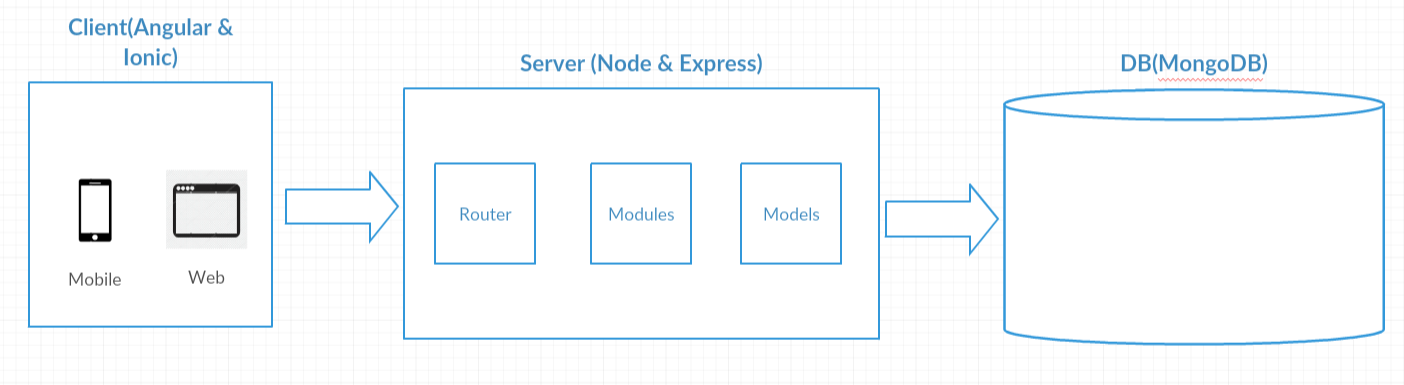
Importance of Factory Maintenance System:

1. Notify the defects or flaws in the machinery to the particular maintenance team , through a centralized ticketing system. Which reduces time consumption and man power.
2. Investigate the Root cause analysis of the failure and can derive solutions to avoid such failures in future.
3. Easy to guide the maintenance team.

**Objectives:**

To create a centralized maintenance system, which automates the maintenance process in a manufacturing unit and track their root cause.

**Architecture diagram:**



**Modules List:**

**Web Modules:**

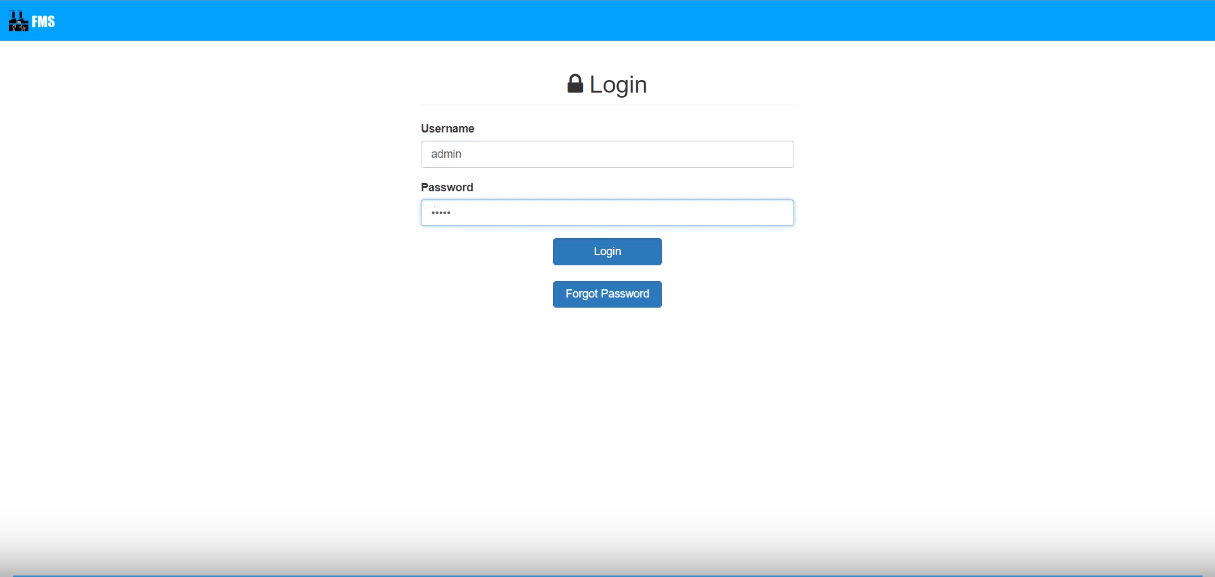
1. Login.
2. User.
3. User Group.
4. Machine.
5. Machine Group.
6. Regular Maintenance.
7. Area.
8. Line.
9. Reasons.
10. Documents.
11. Parts.
12. Step group.
13. Template.
14. Settings.
15. Import.

**Mobile Modules:**

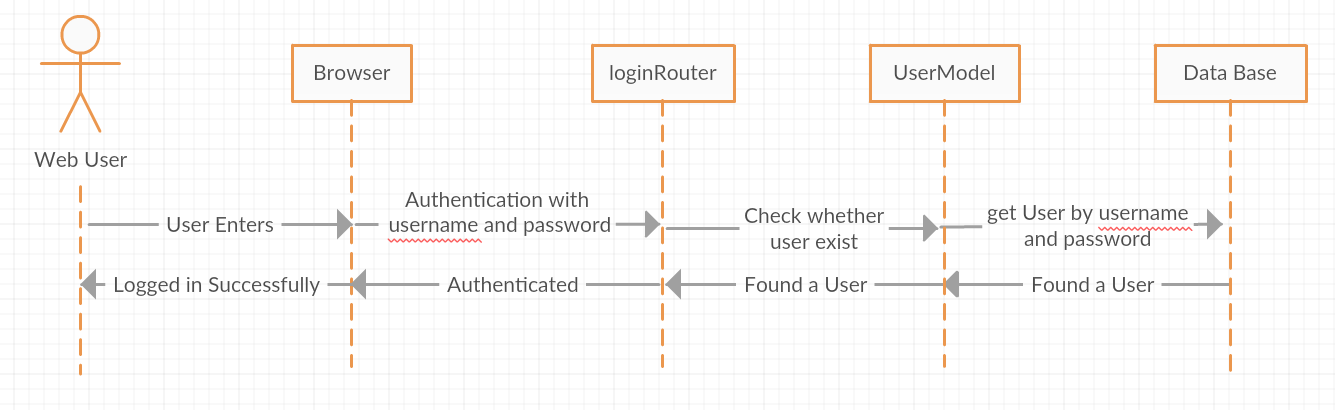
1. Login.
2. Alert.
3. Ticket.
4. Steps.

**Login:**

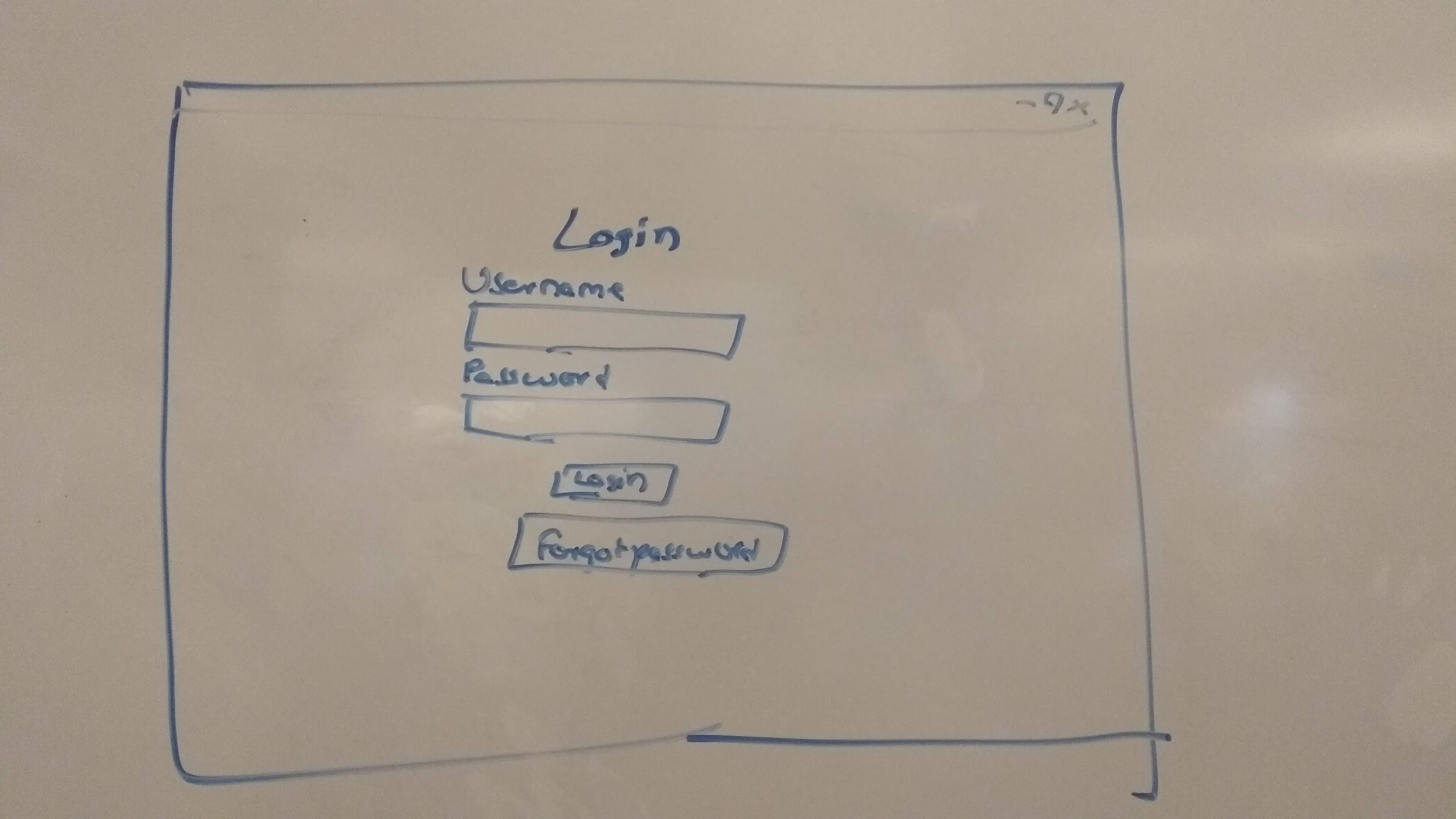
This is the login page. By default we are going to provide Admin user which will have all the permissions.

****

**Sequence diagram:**

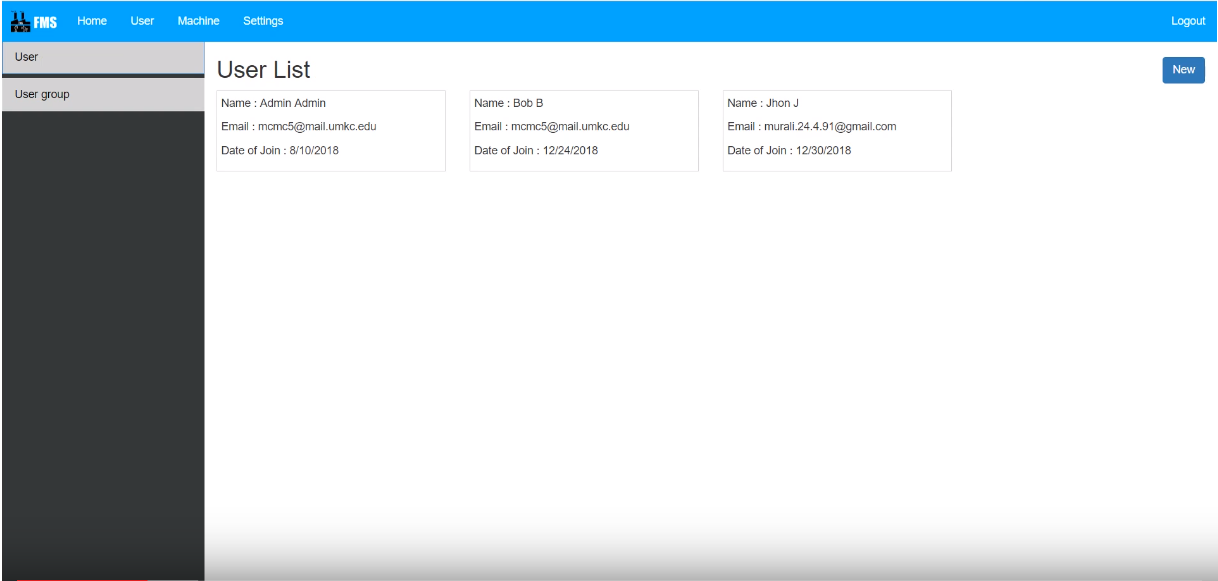


**Wireframe:**

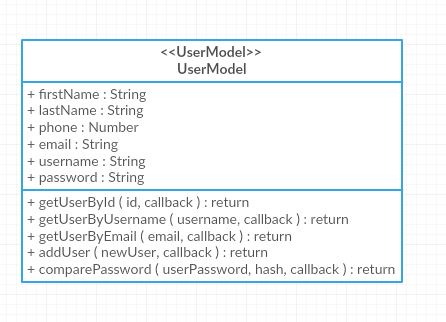


**User:**

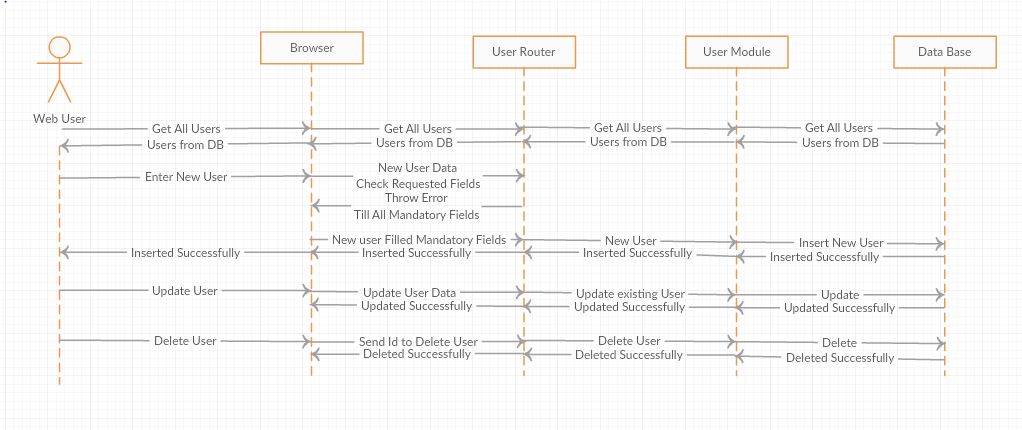
This is the user page where we are going to create user in the factory. This user is created by admin or the user group who is having access to this page.

****

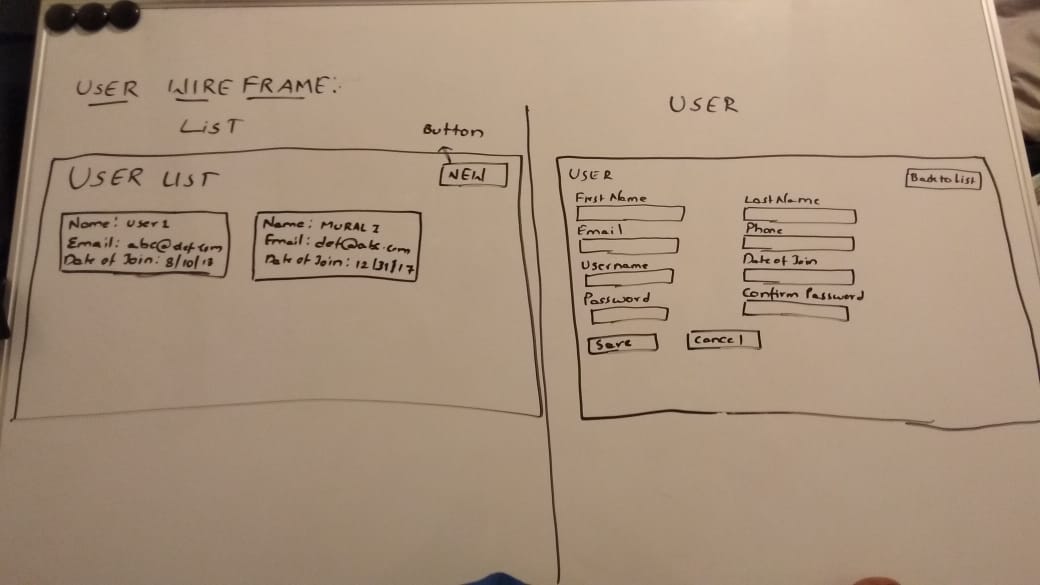
**Class diagram:**



**Sequence Diagram:**

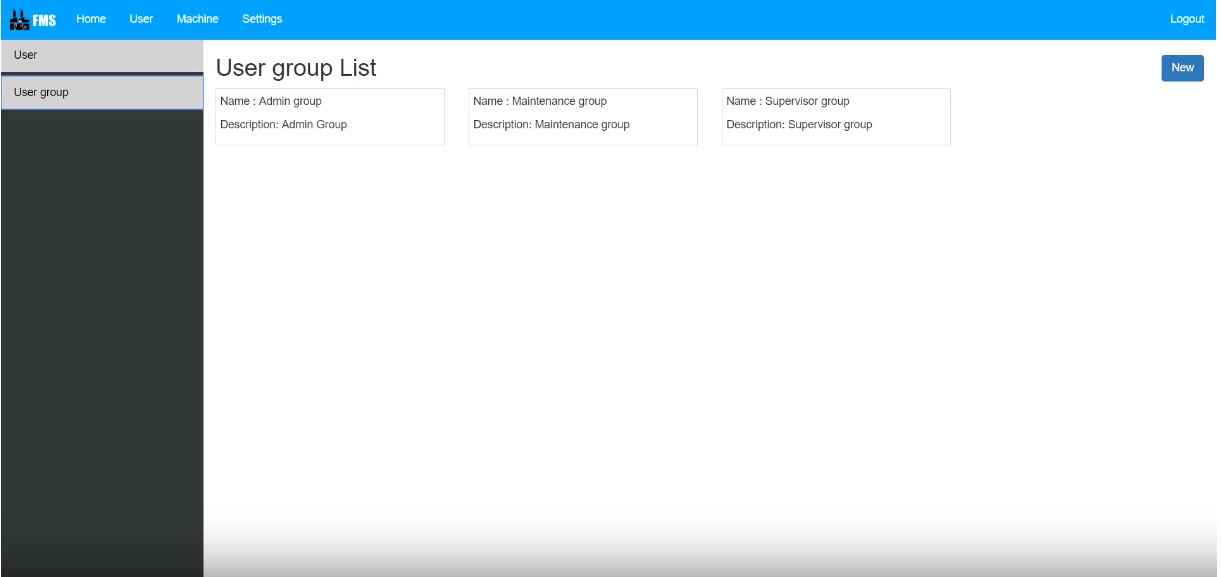


**Wireframe:**

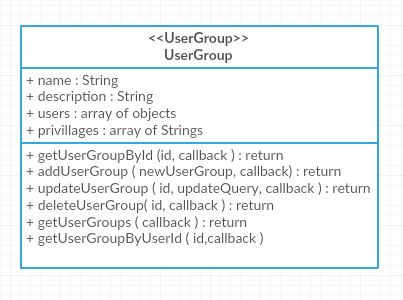


**User Group:**

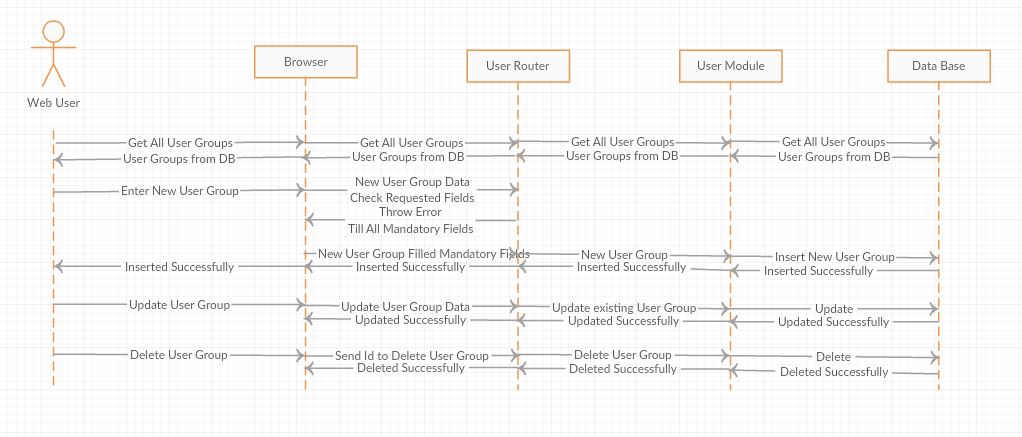
In this user group we can create different group and set privileges to each group. By default, Admin group will be given after deploying the application.

****

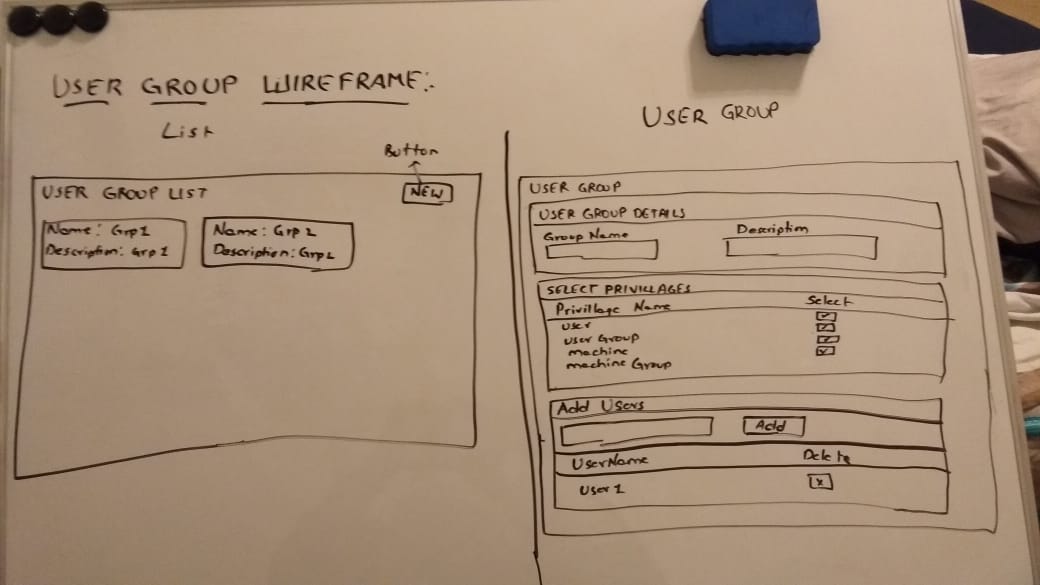
**Class diagram:**



**Sequence Diagram:**

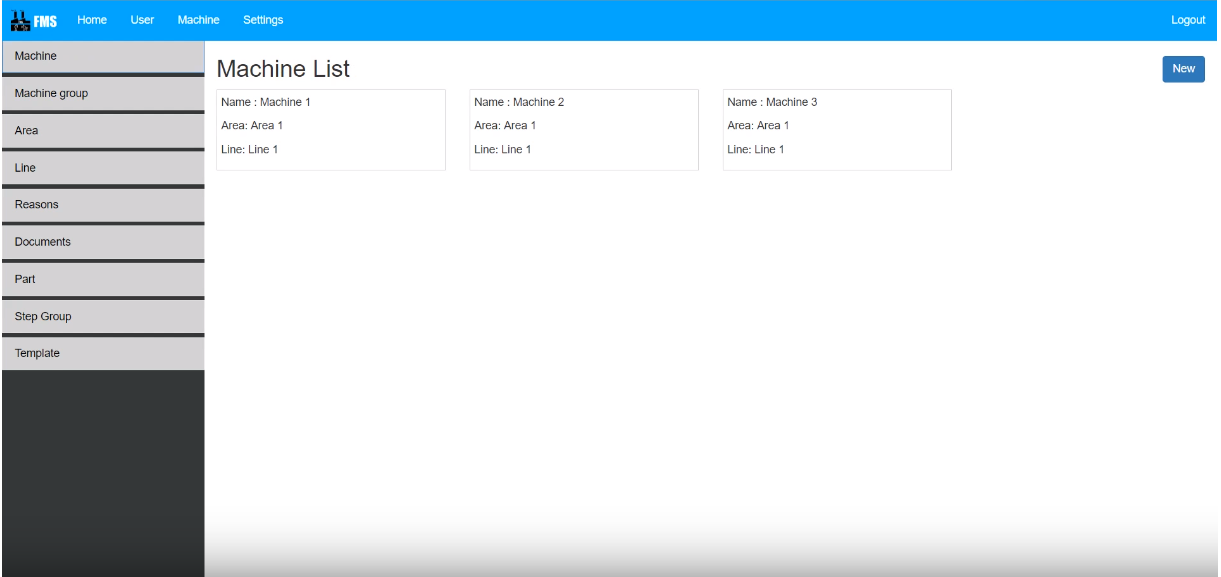


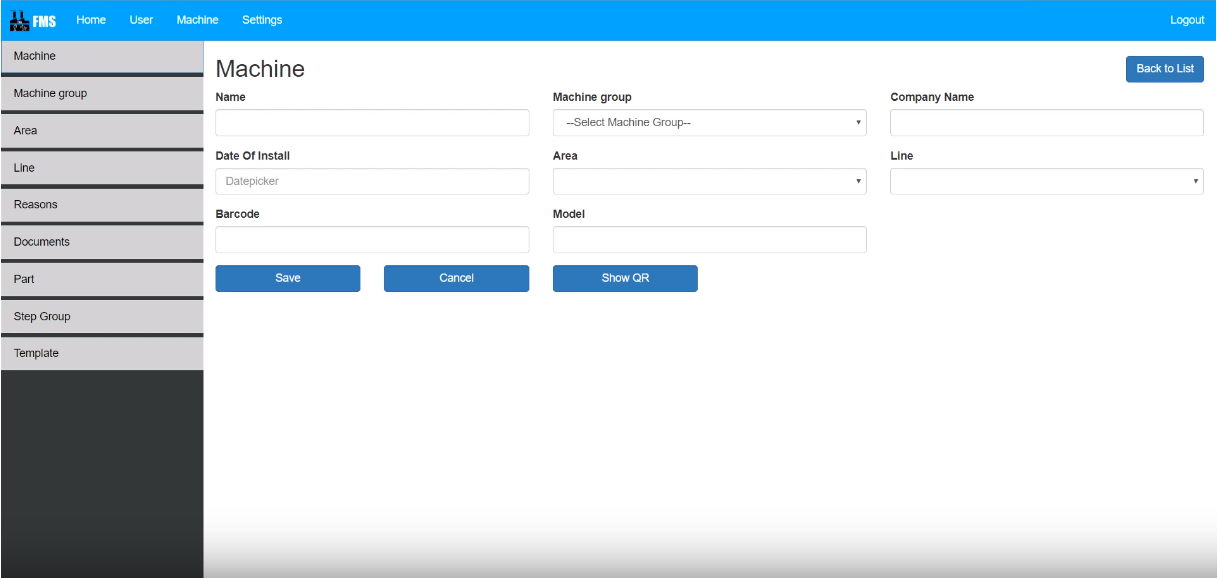
**Wireframe:**



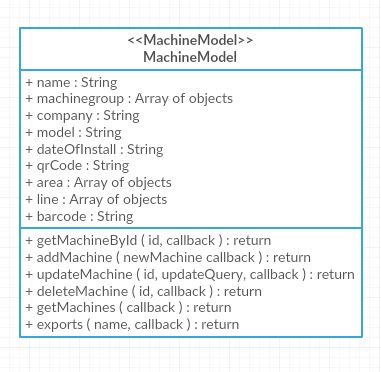
**Machine:**

In this page we can create the all machine present in the factory. We can create machine either manually or by importing via excel.

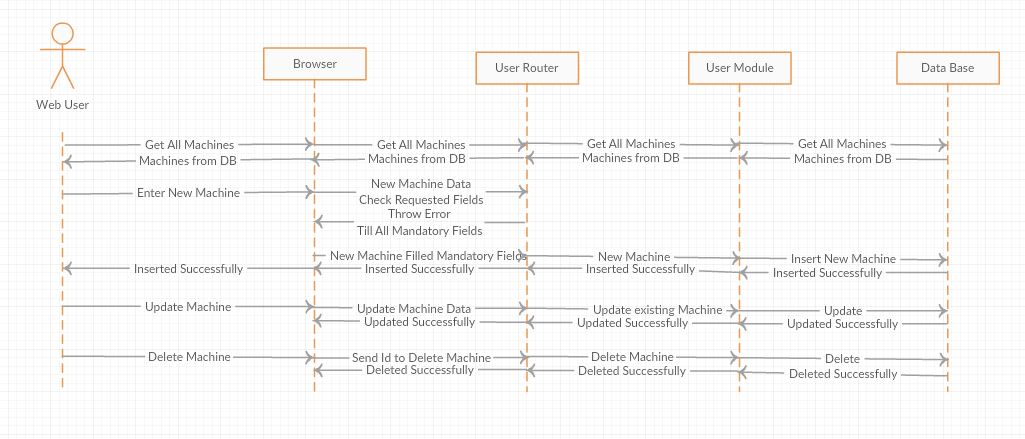
****

****

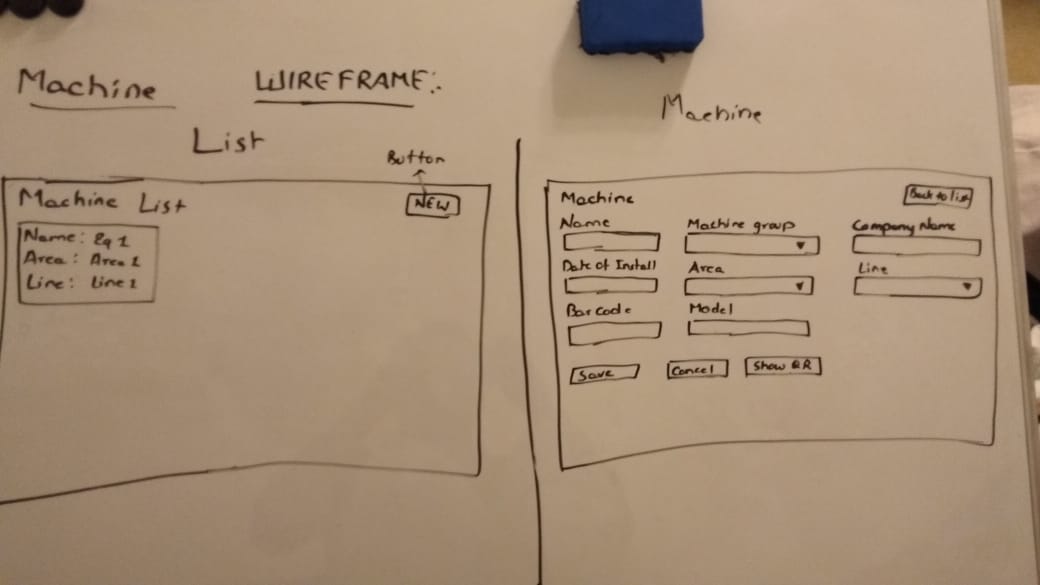
**Class diagram:**



**Sequence diagram:**

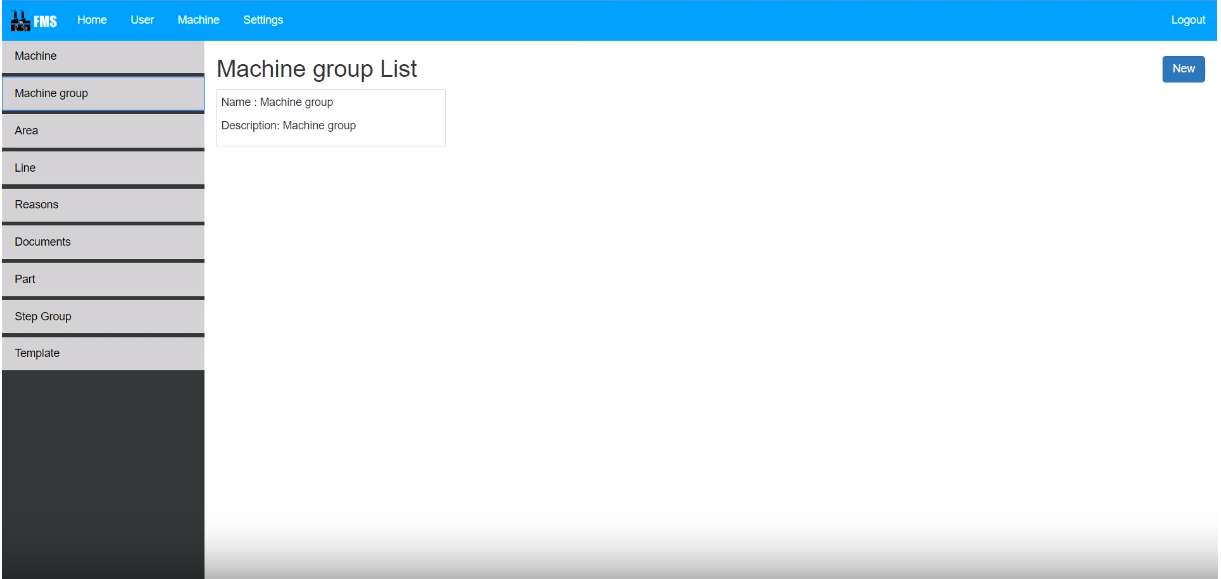


**Wireframe:**

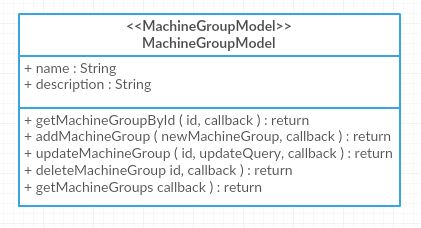


**Machine Group:**

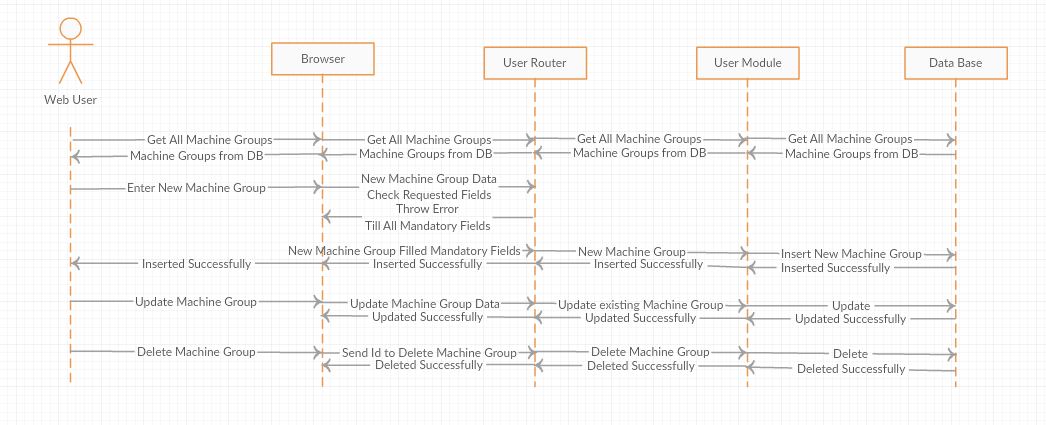
We can group the machines by its category. We can add this machine group inside the machine page.

****

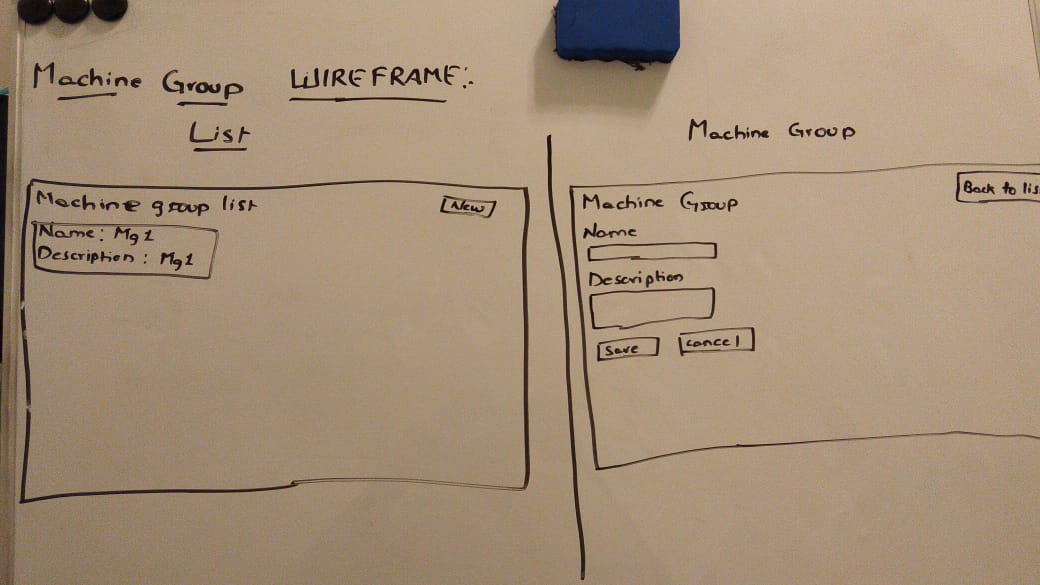
**Class diagram:**



**Sequence diagram:**

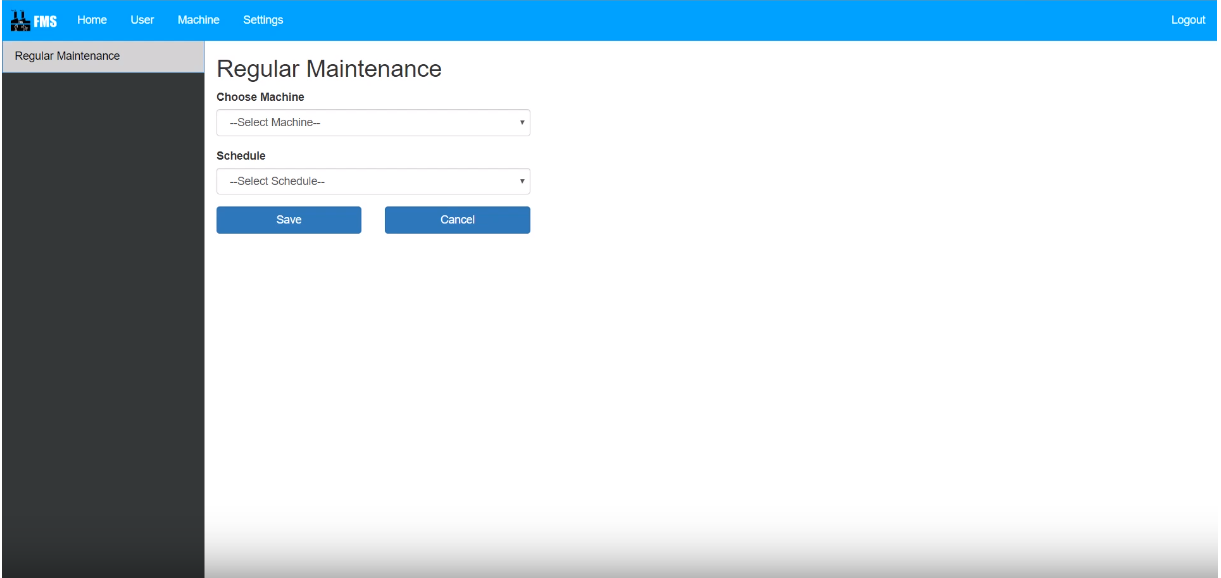


**Wireframe:**

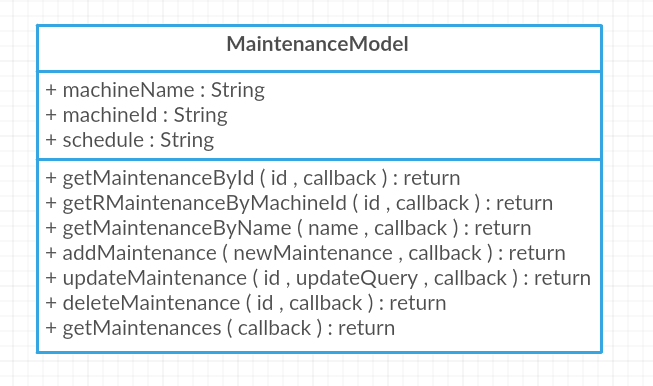


**Regular Maintenance:**

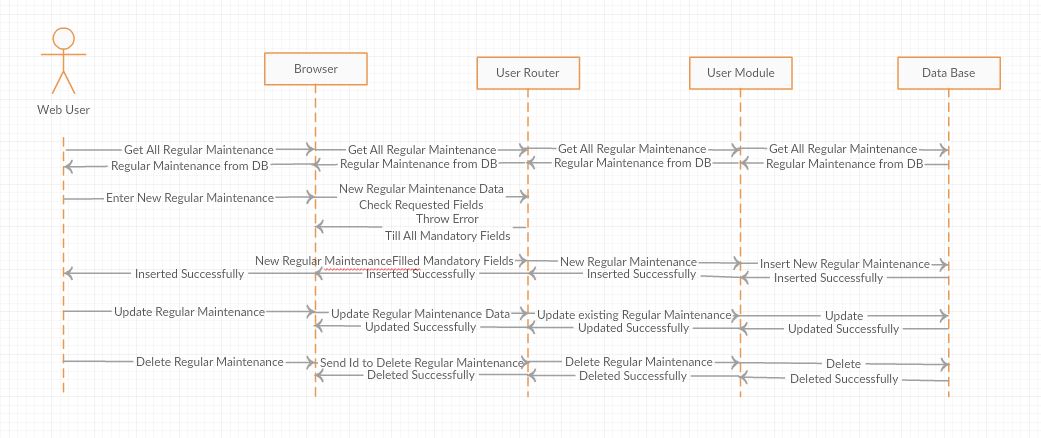
Every machine we need to do maintenance regularly otherwise it will fail during the production which will affect the profit. In this page we can schedule regular maintenance which will create maintenance ticket automatically.

****

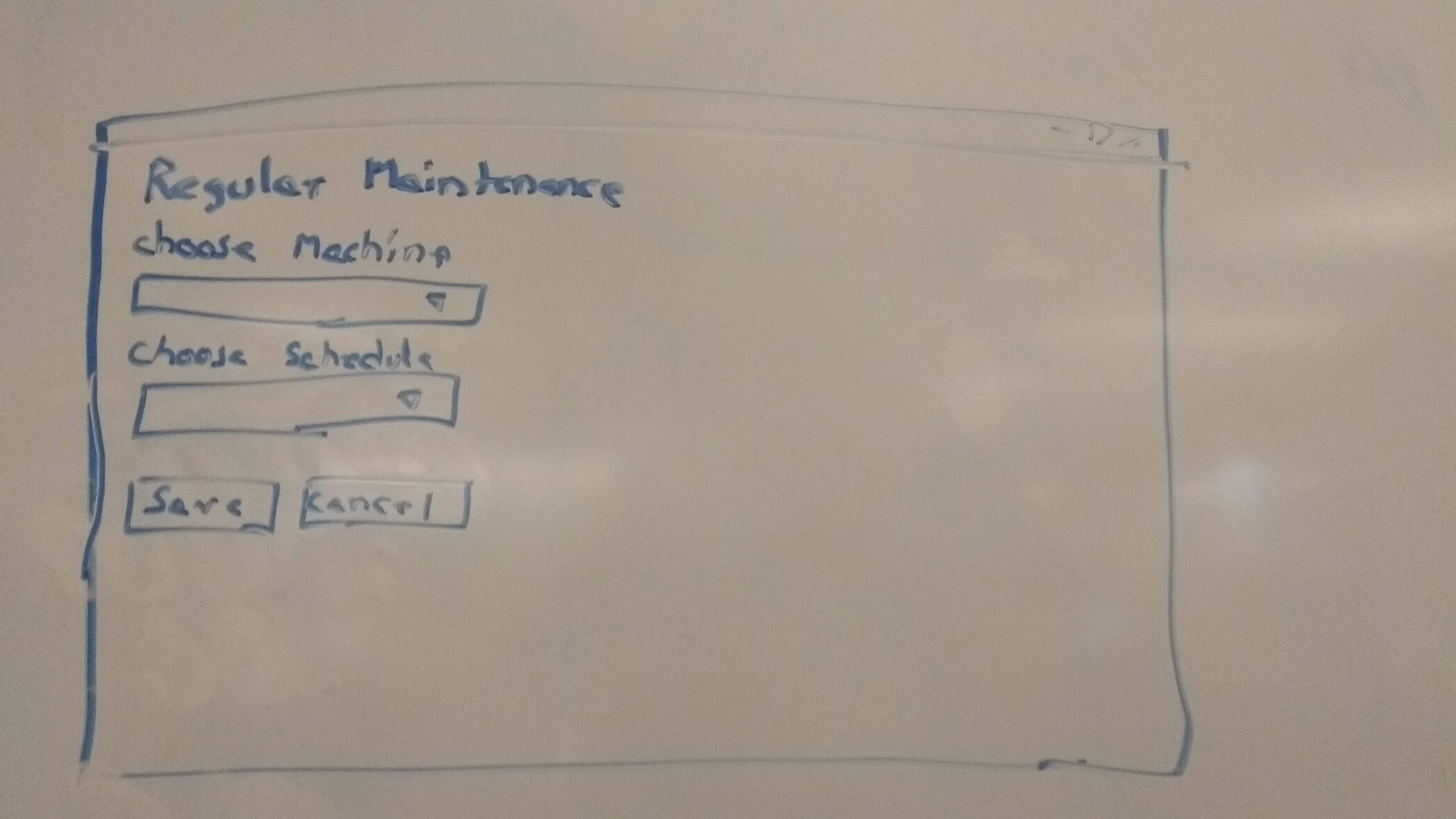
**Class diagram:**



**Sequence diagram:**

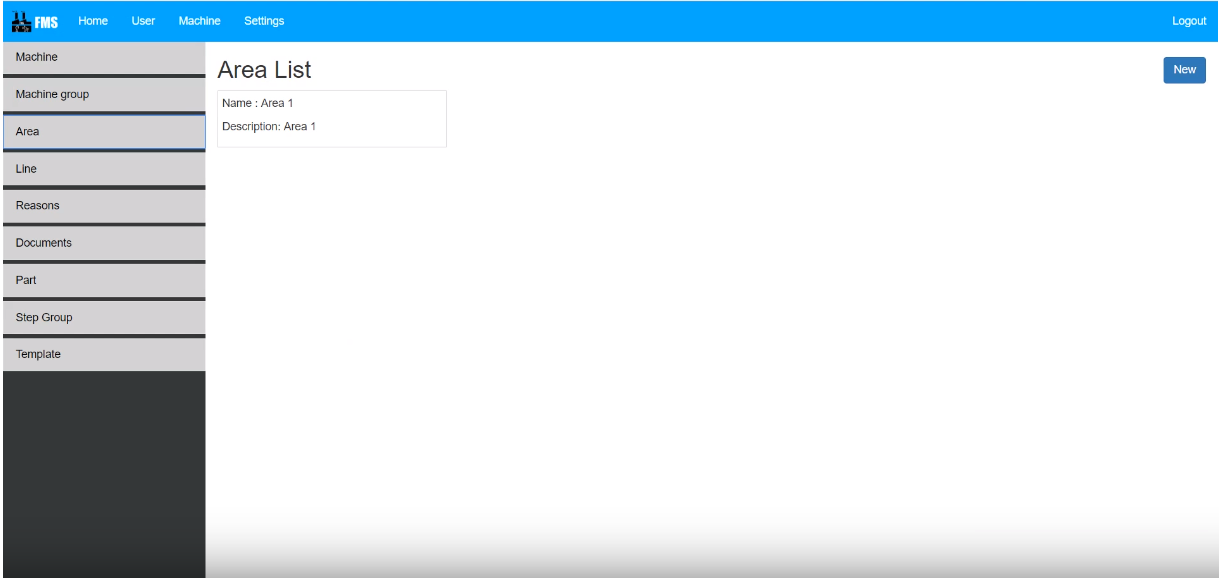


**Wireframe:**

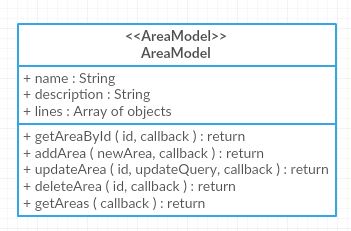


**Area:**

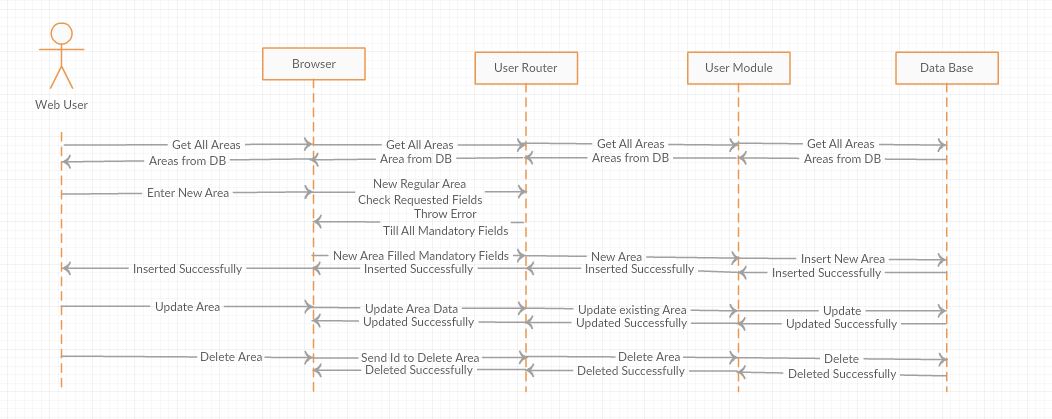
The factory will be divided into areas. Every area will contain lines. In the line machine will be place. In this page we can create the areas in the factory.

****

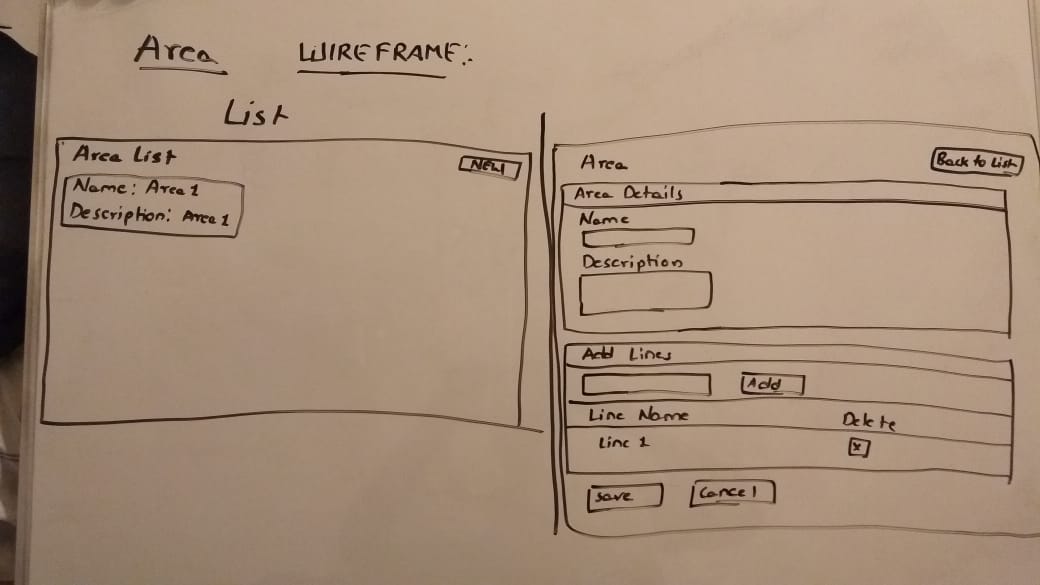
**Class diagram:**



**Sequence diagram:**

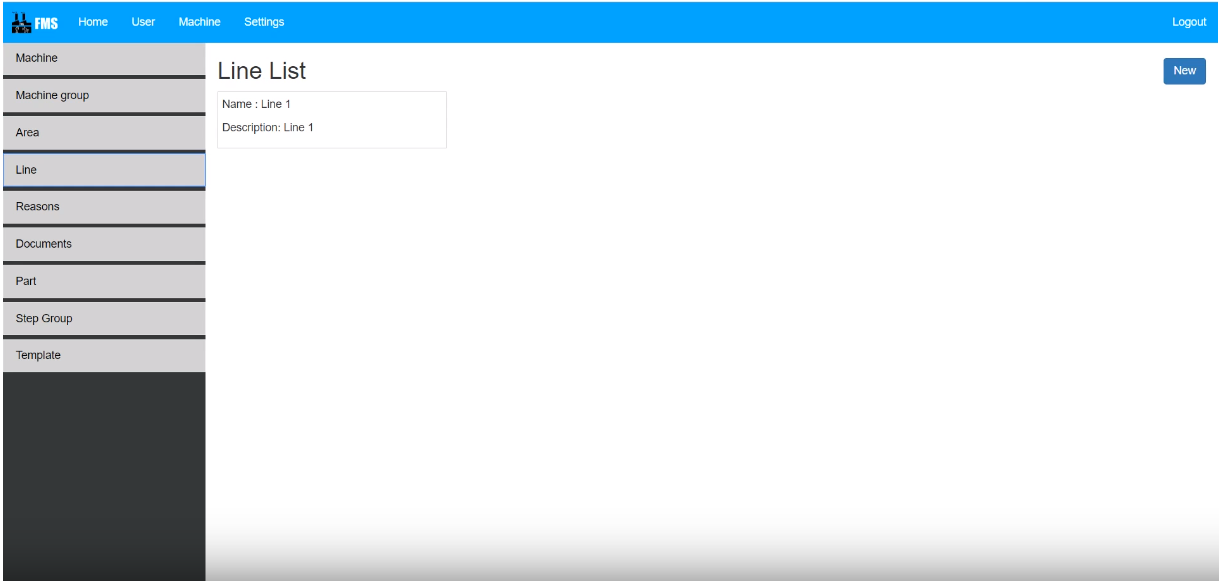


**Wireframe:**

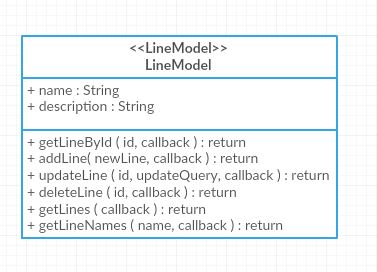


**Line:**

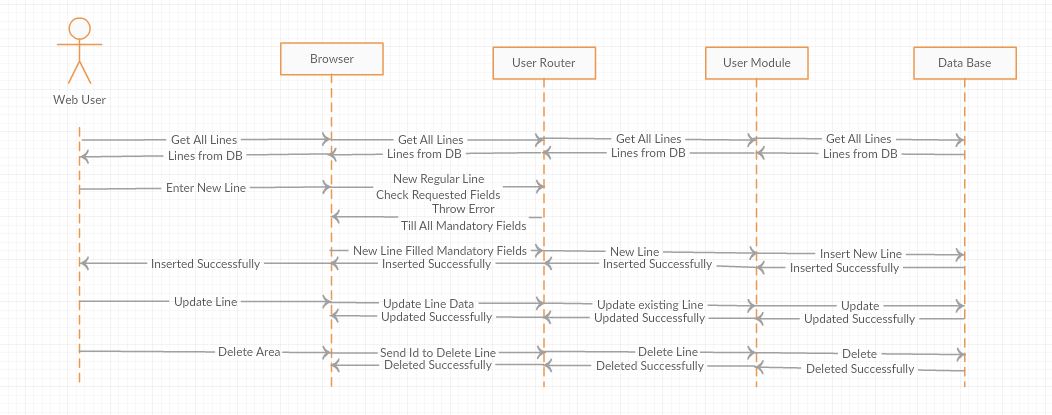
In this page we can create line in the factory. In the machine creation this lines are mentioned so that the maintenance guy will understand where the machine exactly placed.

****

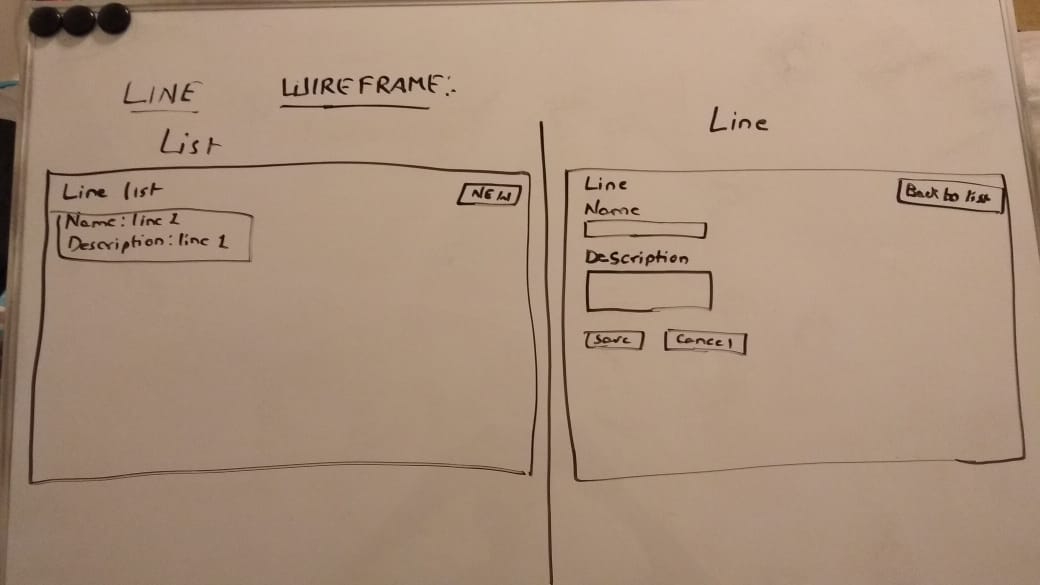
**Class diagram:**



**Sequence diagram:**

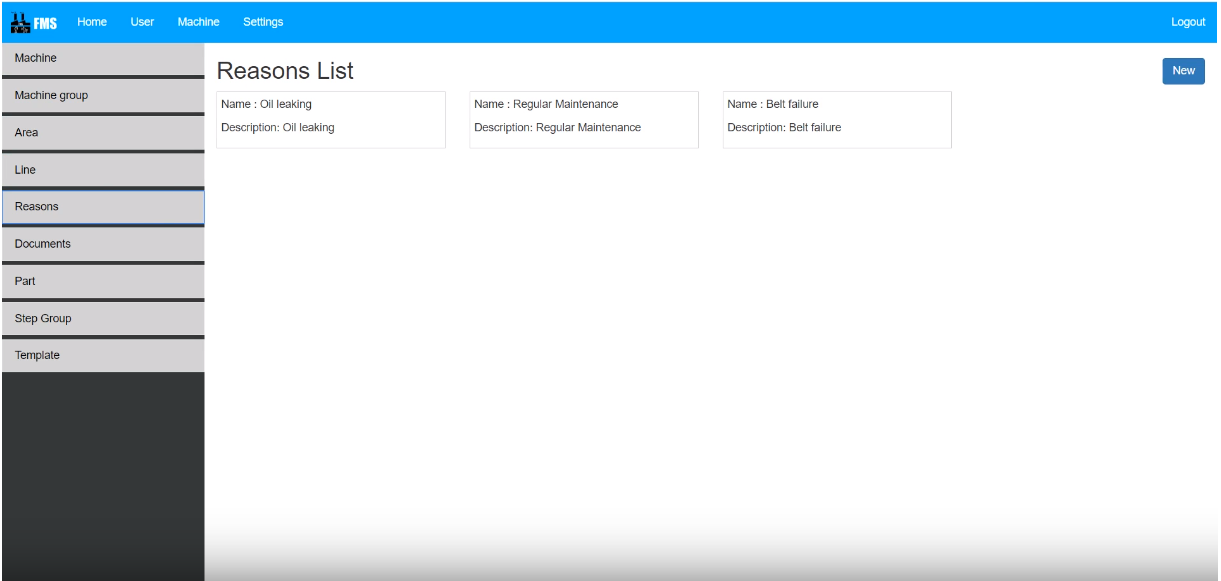


**Wireframe:**

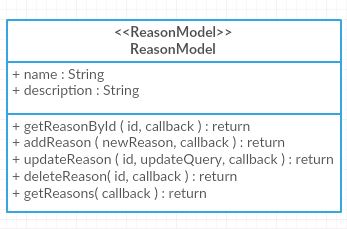


**Reasons:**

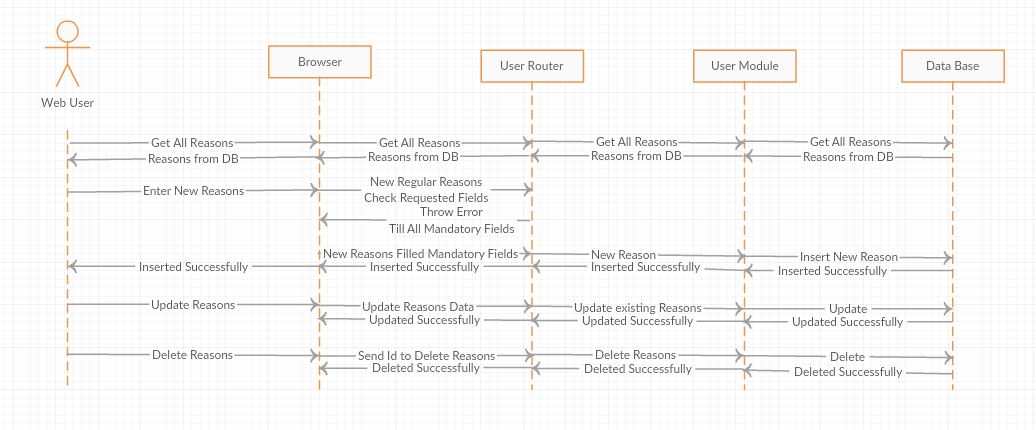
Machine will fail because of reason. In this page we can create the reason for machine failure.

****

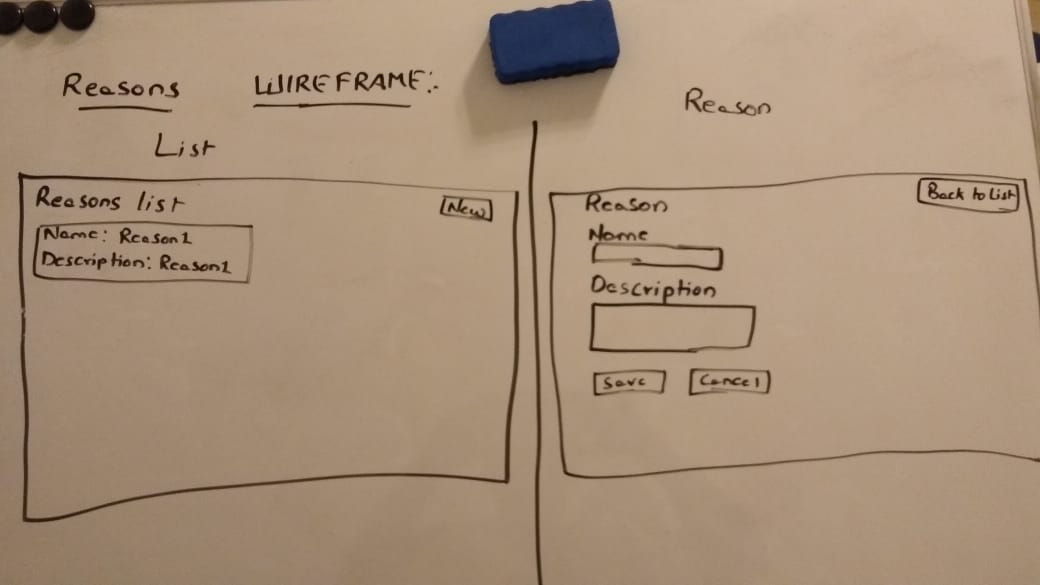
**Class diagram:**



**Sequence diagram:**



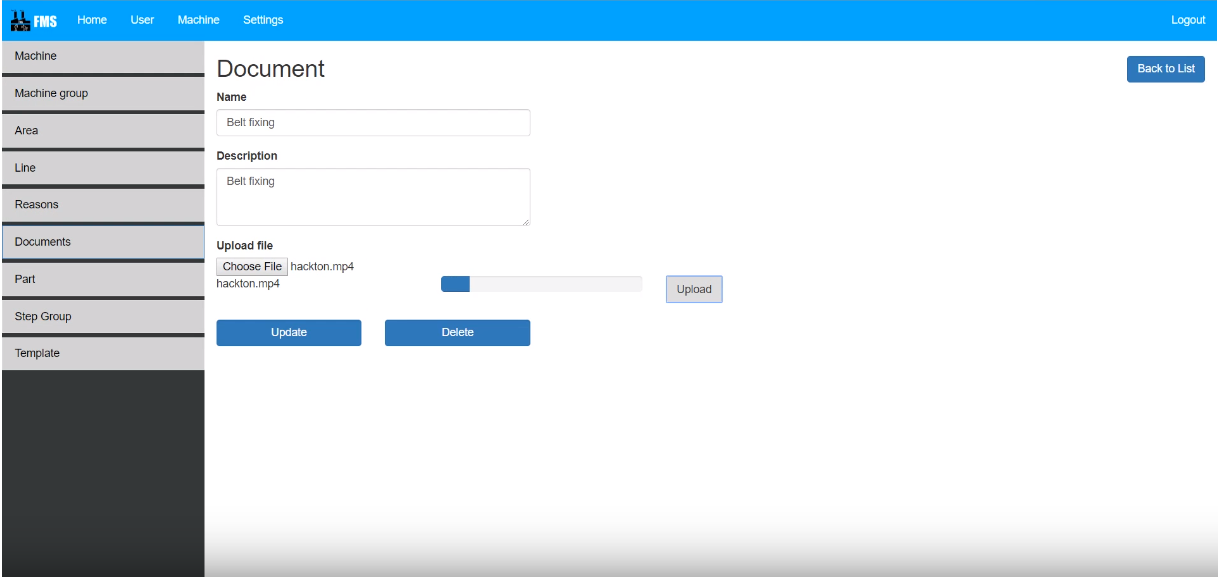
**Wireframe:**



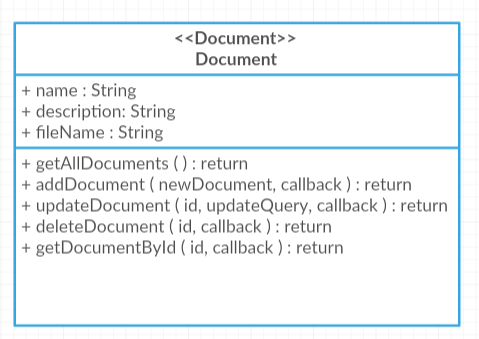
**Documents:**

During the maintenance, the maintenance people need some references for referring. In this page we can create the documents and we can refer them in each step of maintenance.

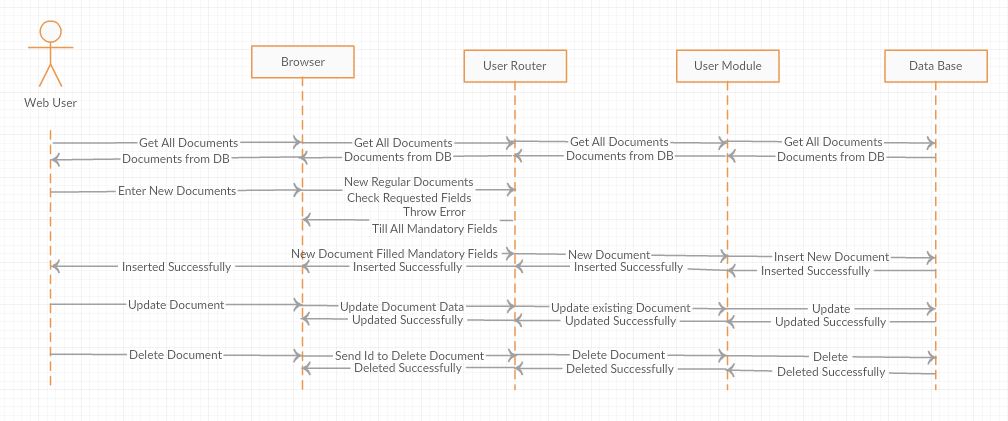
****

****

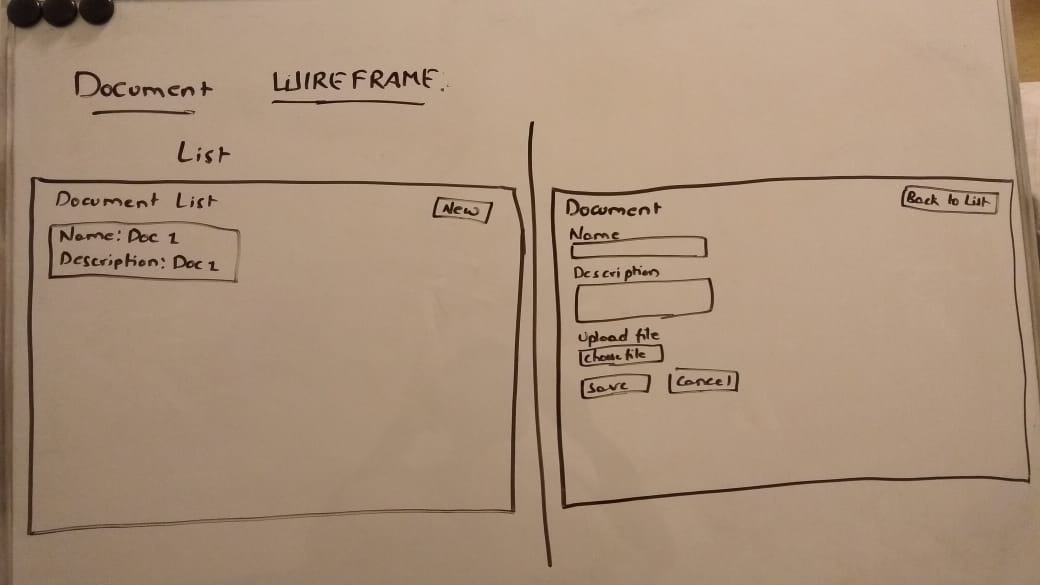
**Class diagram:**



**Sequence diagram:**

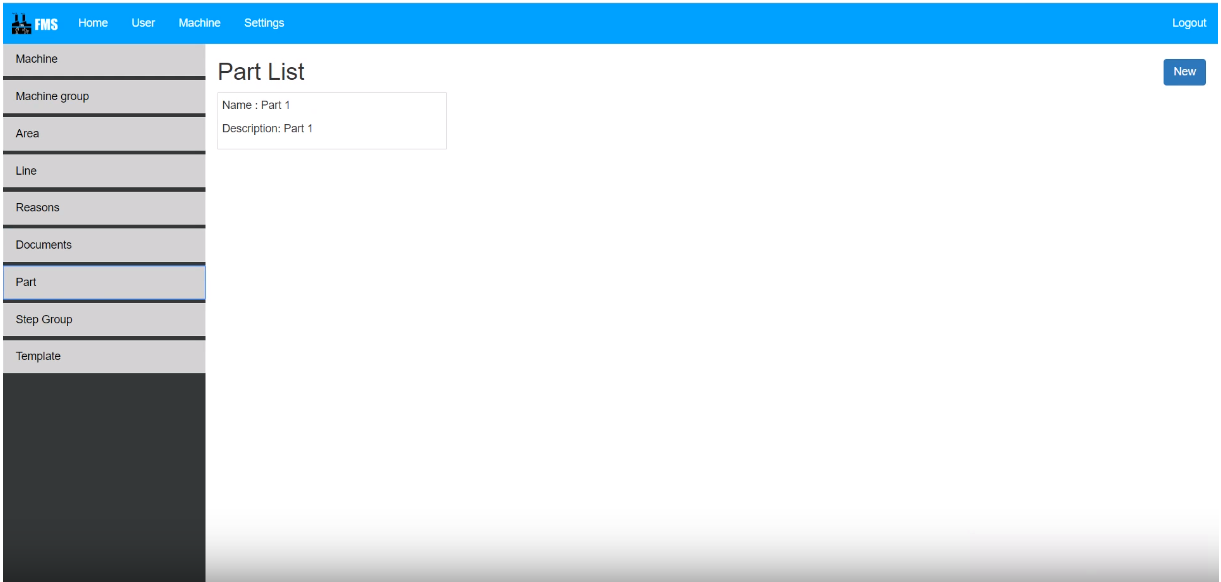


**Wireframe:**

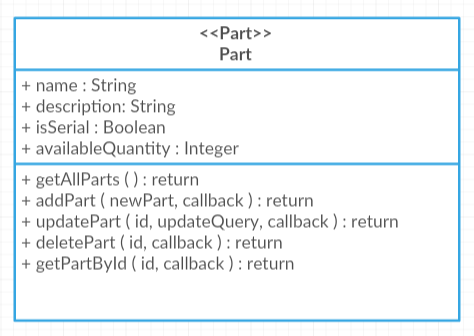


**Parts:**

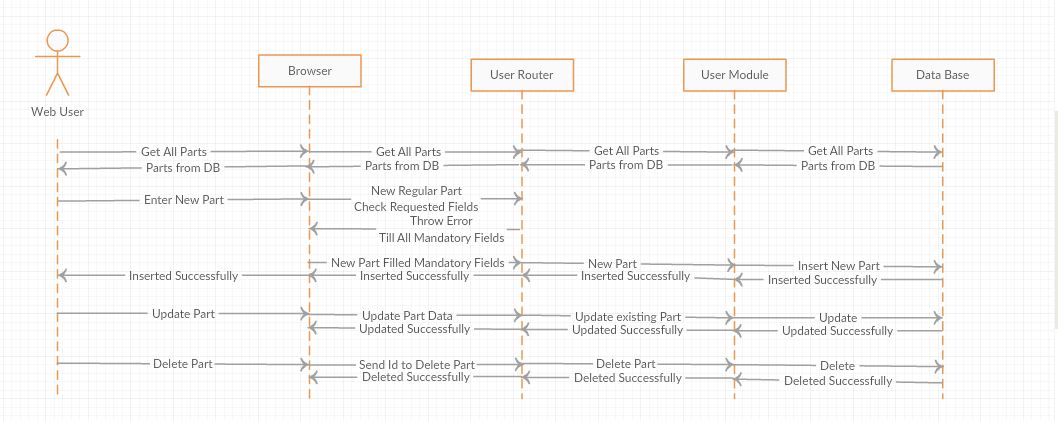
Every time we need some part to do maintenance. In this page we can create the parts and refer them in each step of maintenace.

****

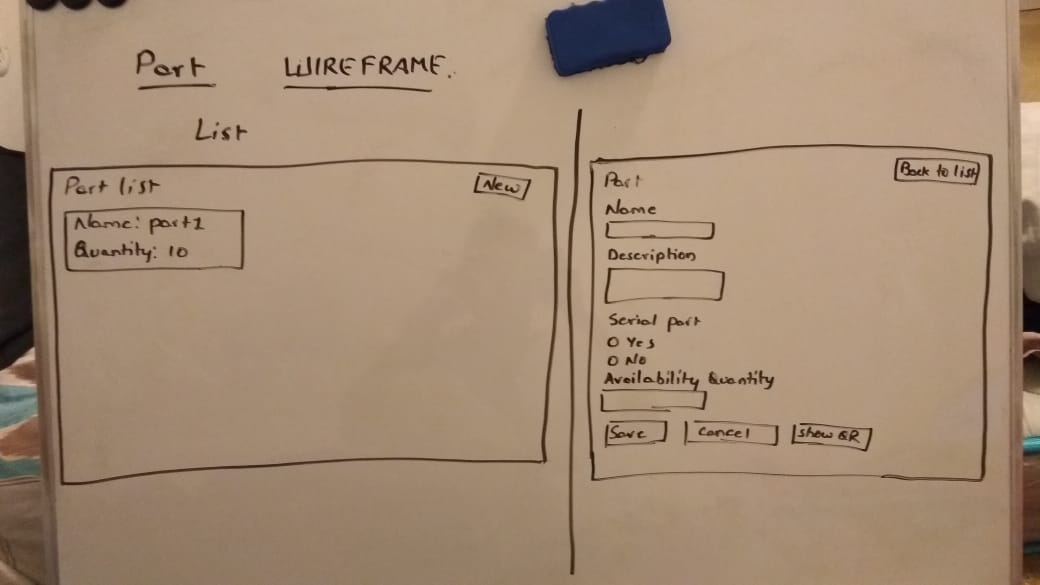
**Class diagram:**



**Sequence diagram:**

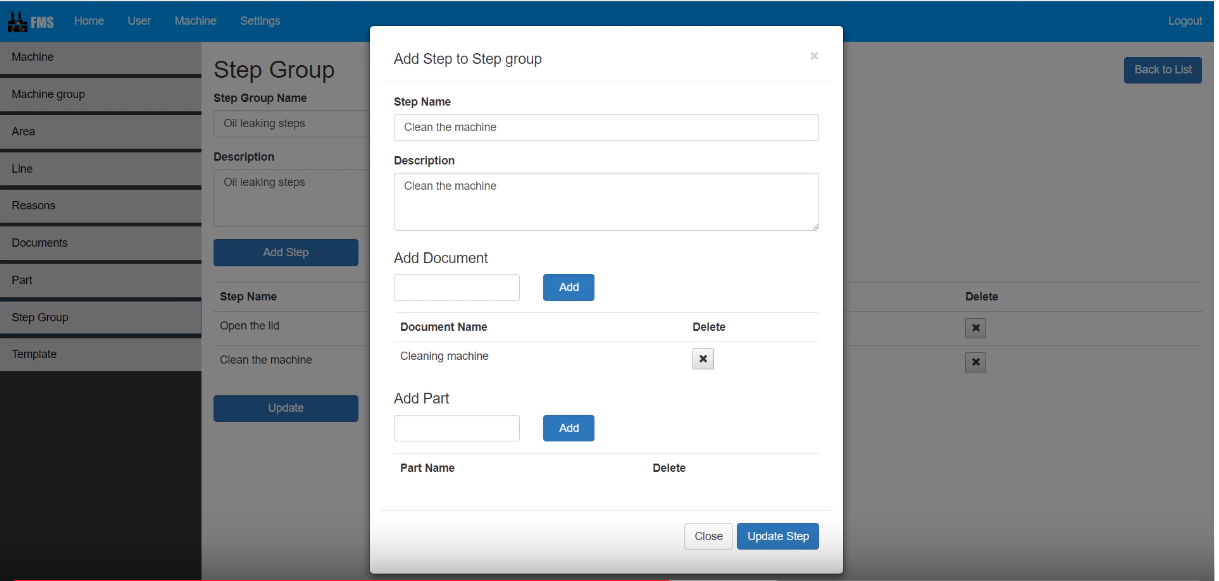
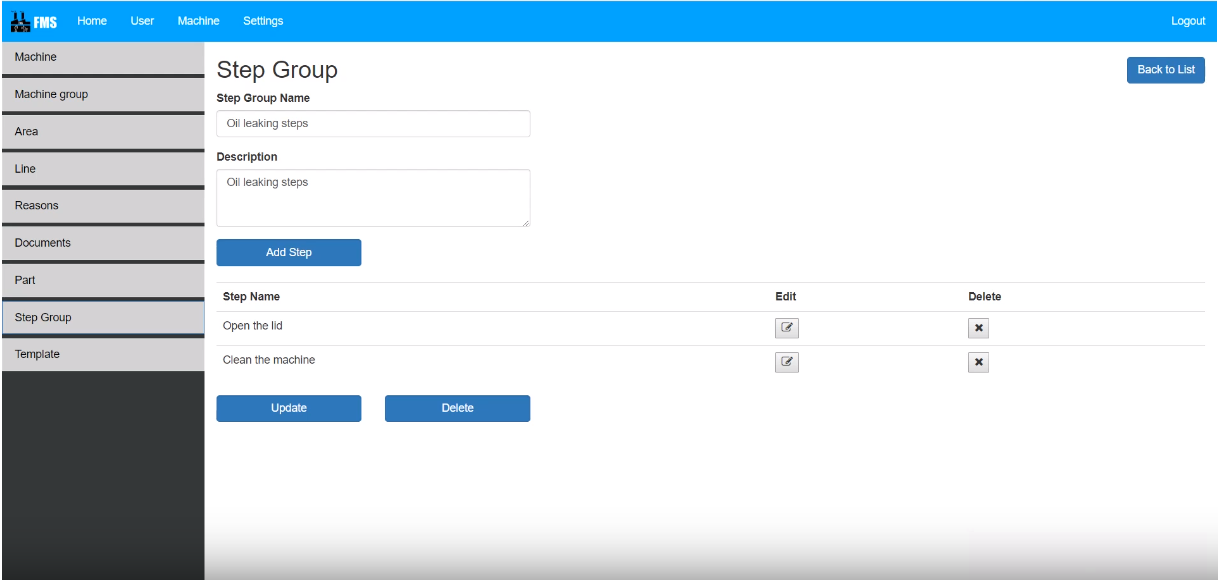


**Wireframe:**

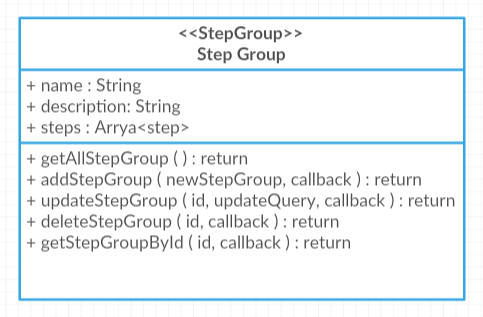


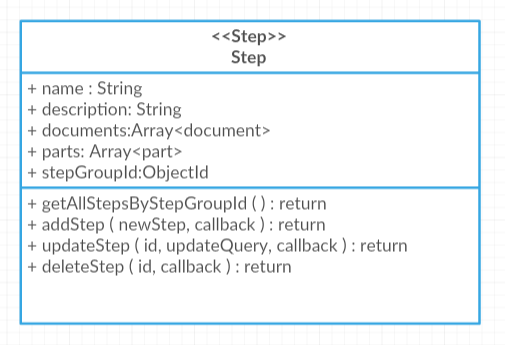
**Step Group:**

Every maintenance contains different steps to perform. Here we can configure the steps and we can also configure the documents and parts in each step.

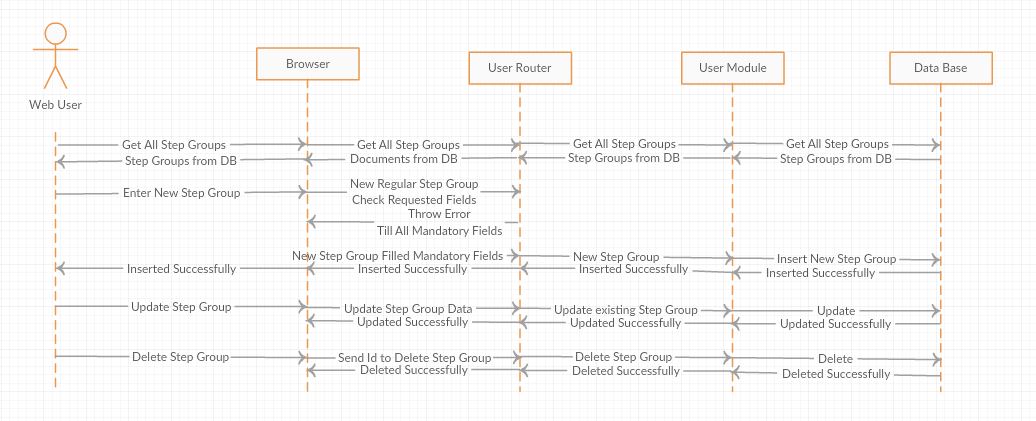
****

**Class diagram:**

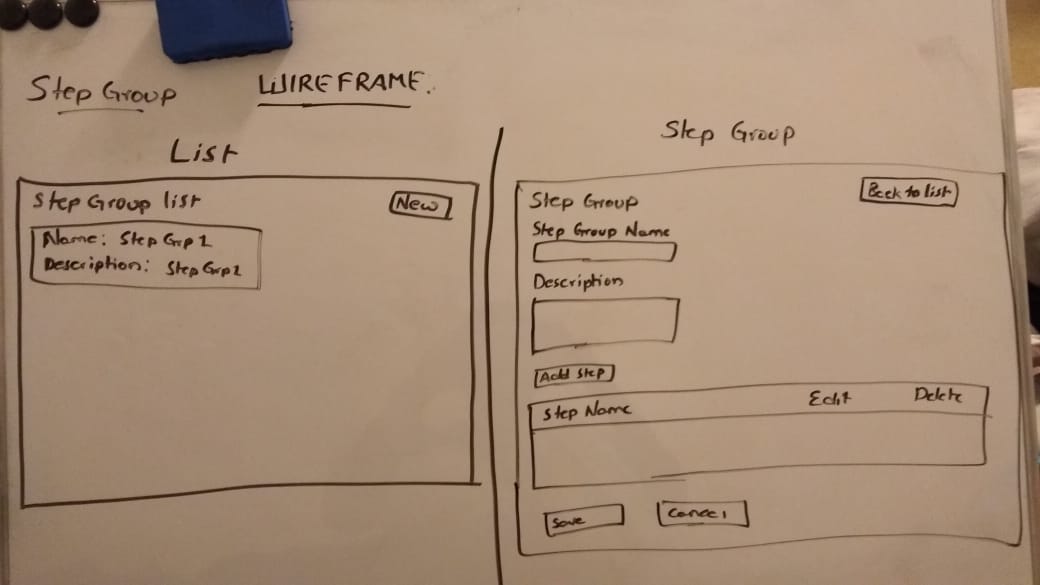


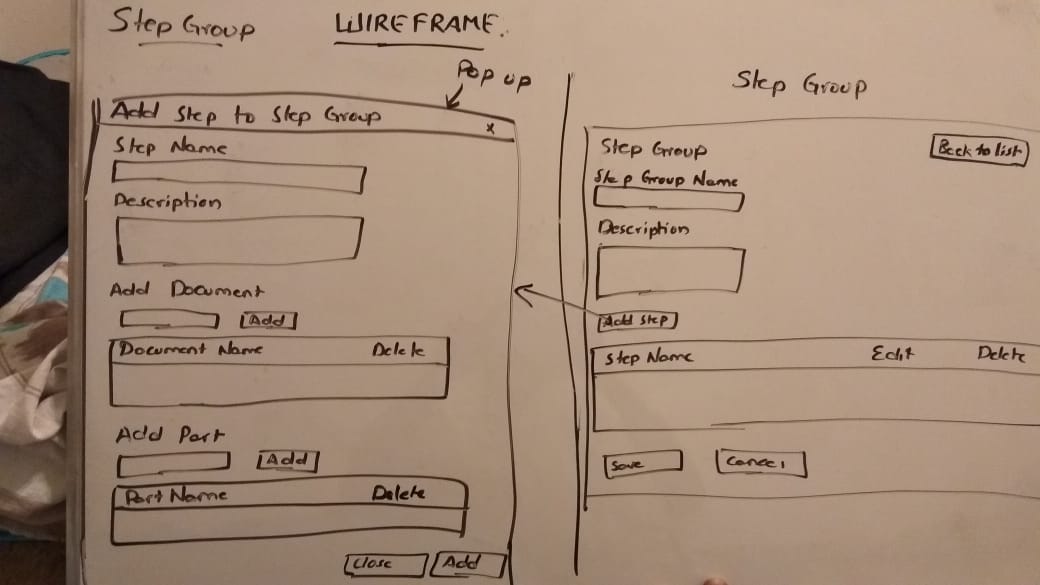


**Sequence diagram:**



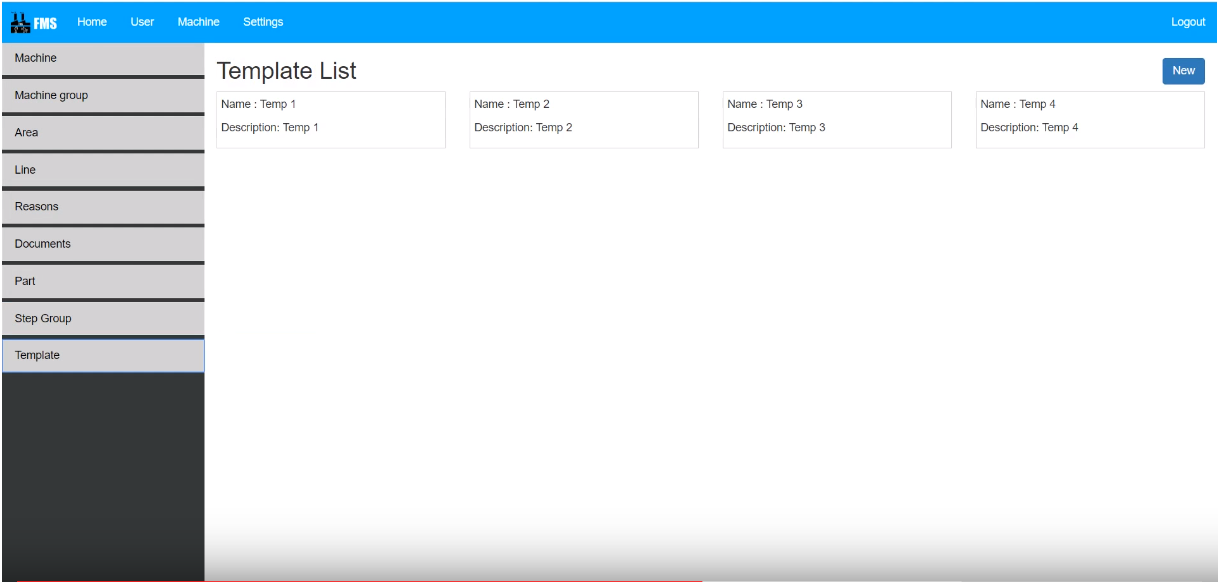
**Wireframe:**

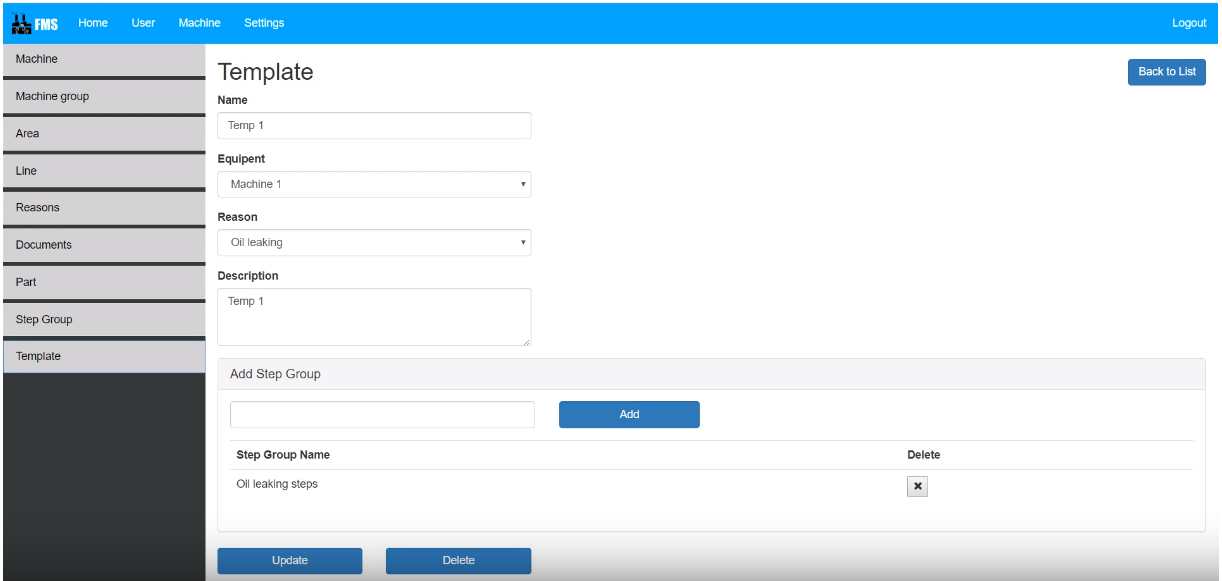




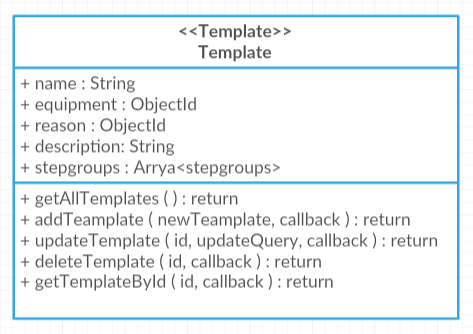
**Template:**

Once the ticket is created the system need to fetch the steps to perform maintenance. In this template page we will configure the steps for machine and that reason.

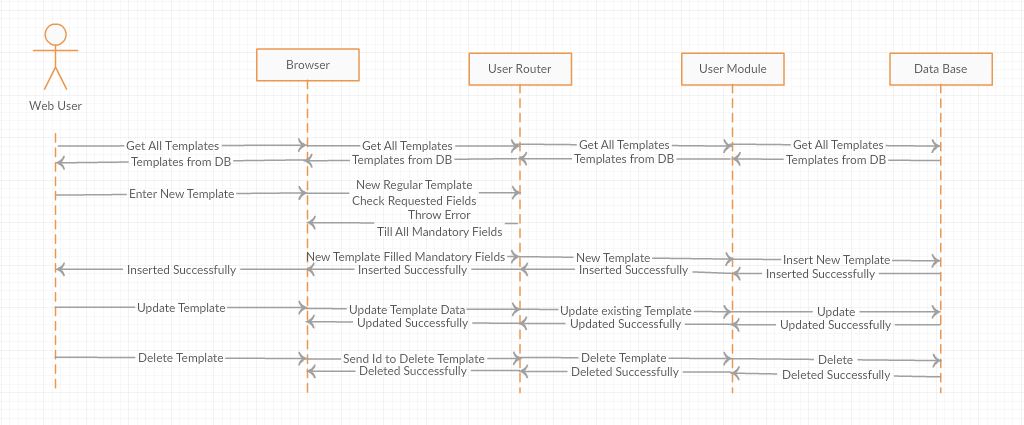
****

****

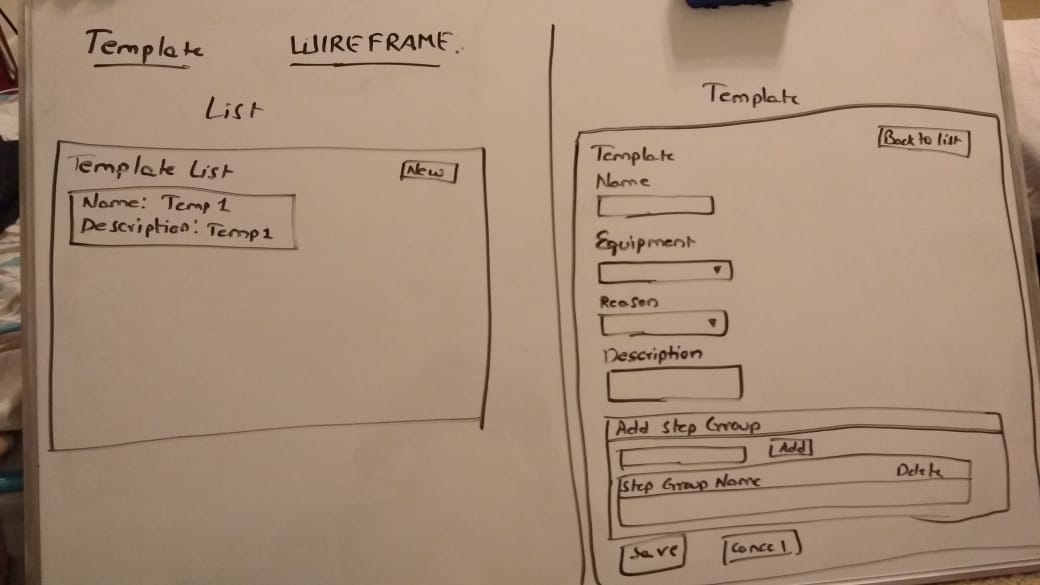
**Class diagram:**



**Sequence diagram:**

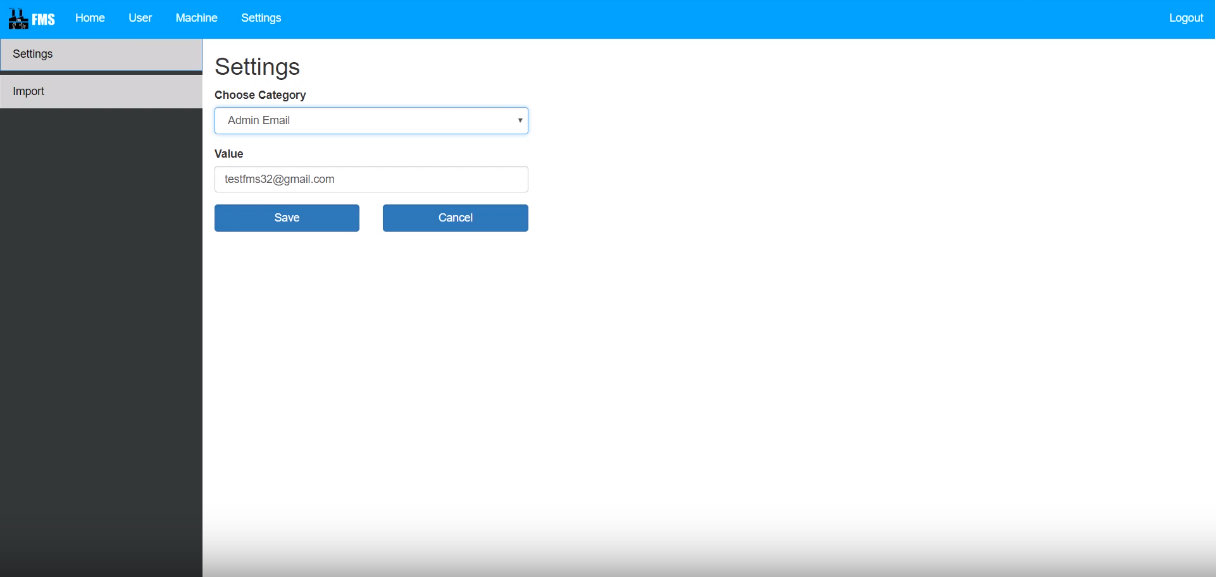


**Wireframe:**

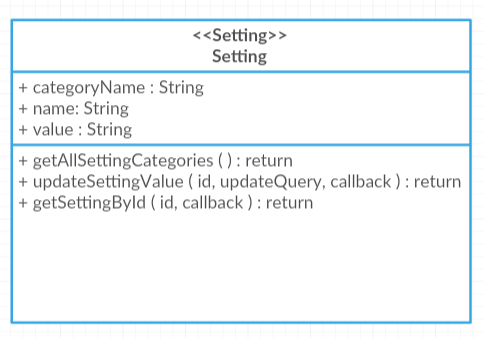


**Settings:**

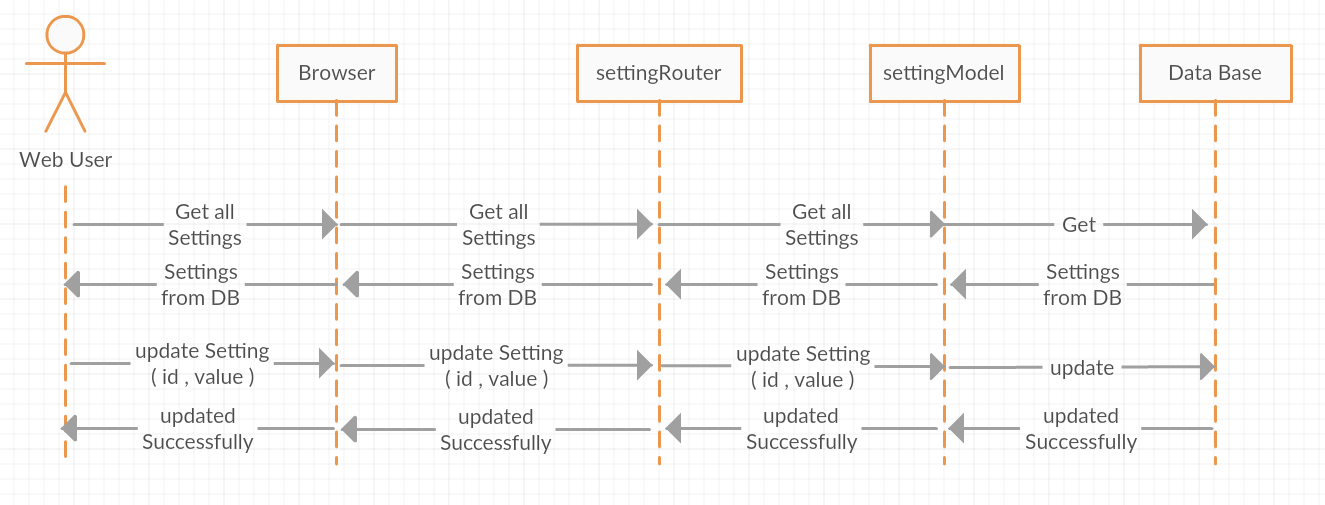
In this settings page we can configure the application settings.

****

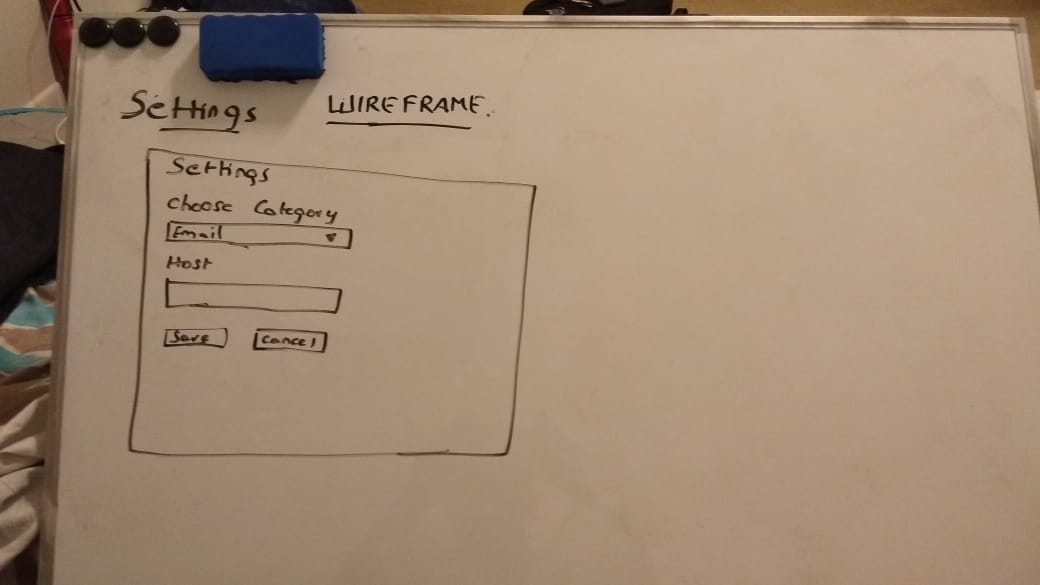
**Class diagram:**



**Sequence diagram:**

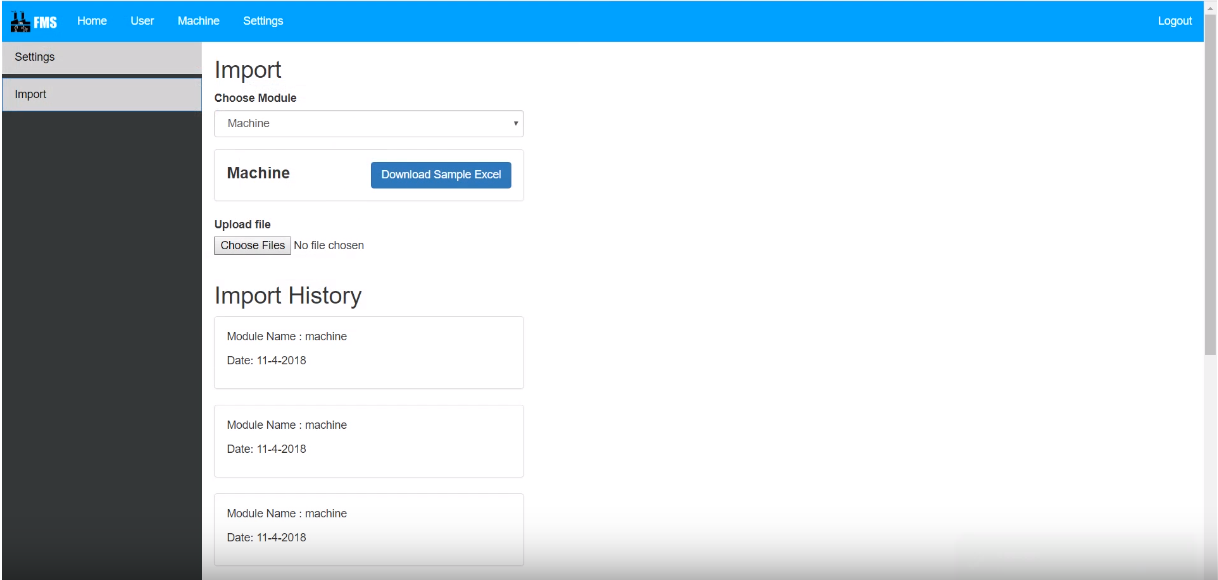


**Wireframe:**



**Import:**

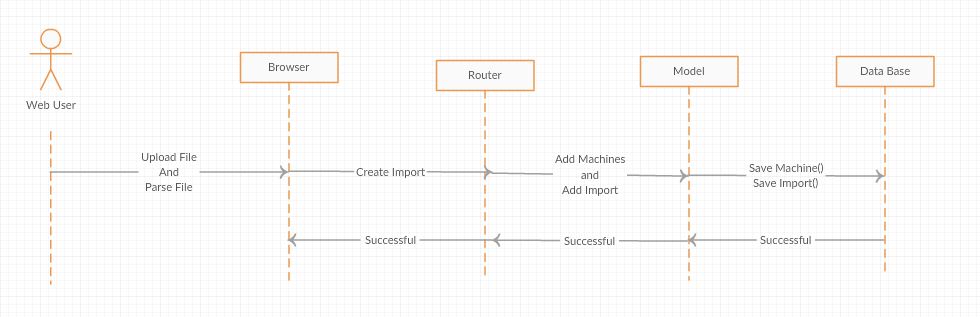
In this import page we can import machine for excel documents.

****

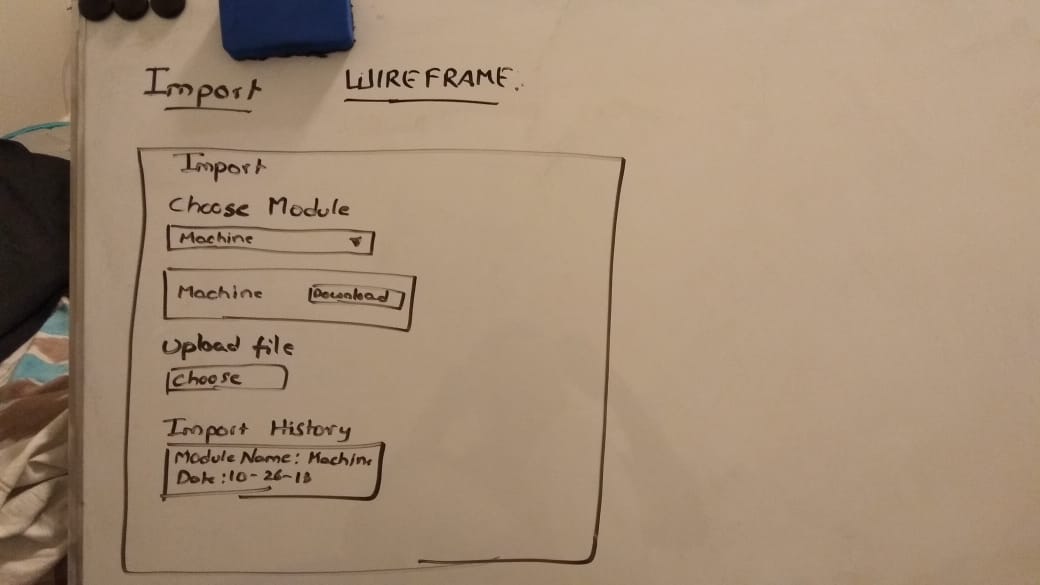
**Class diagram:**



**Sequence diagram:**

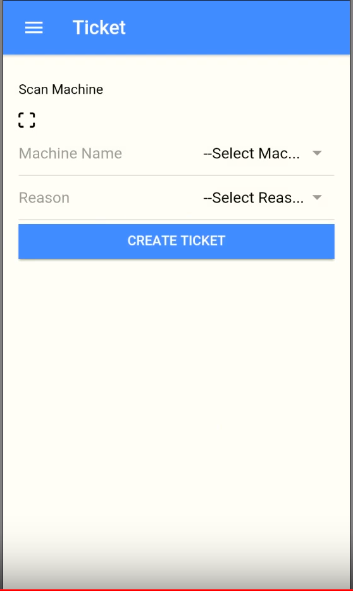
****

**Wireframe:**

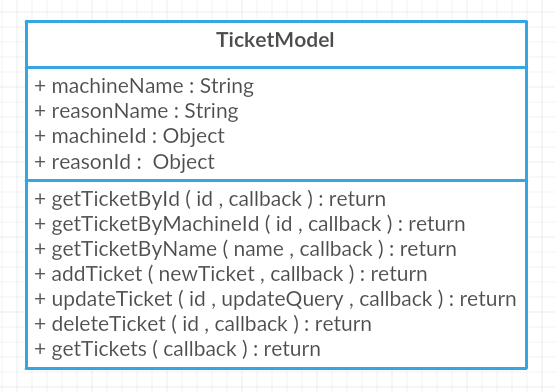


**Ticket:**

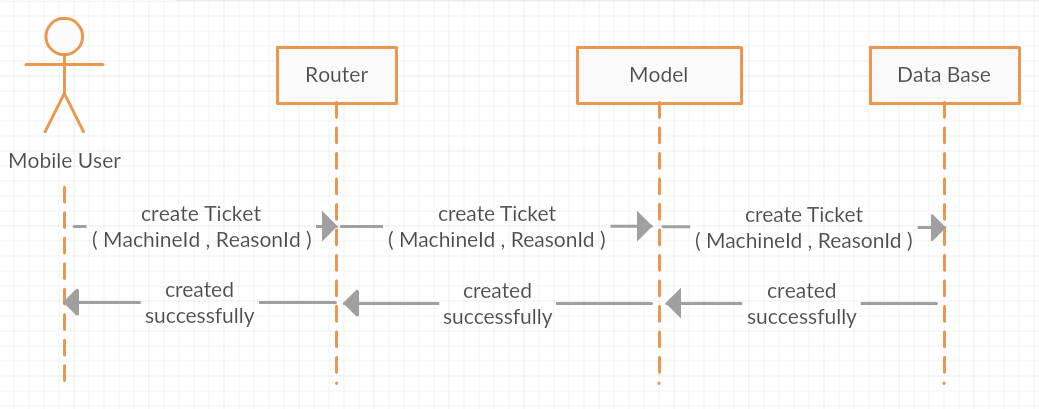
In this ticket page we can create ticket when the machine is failed during production.

****

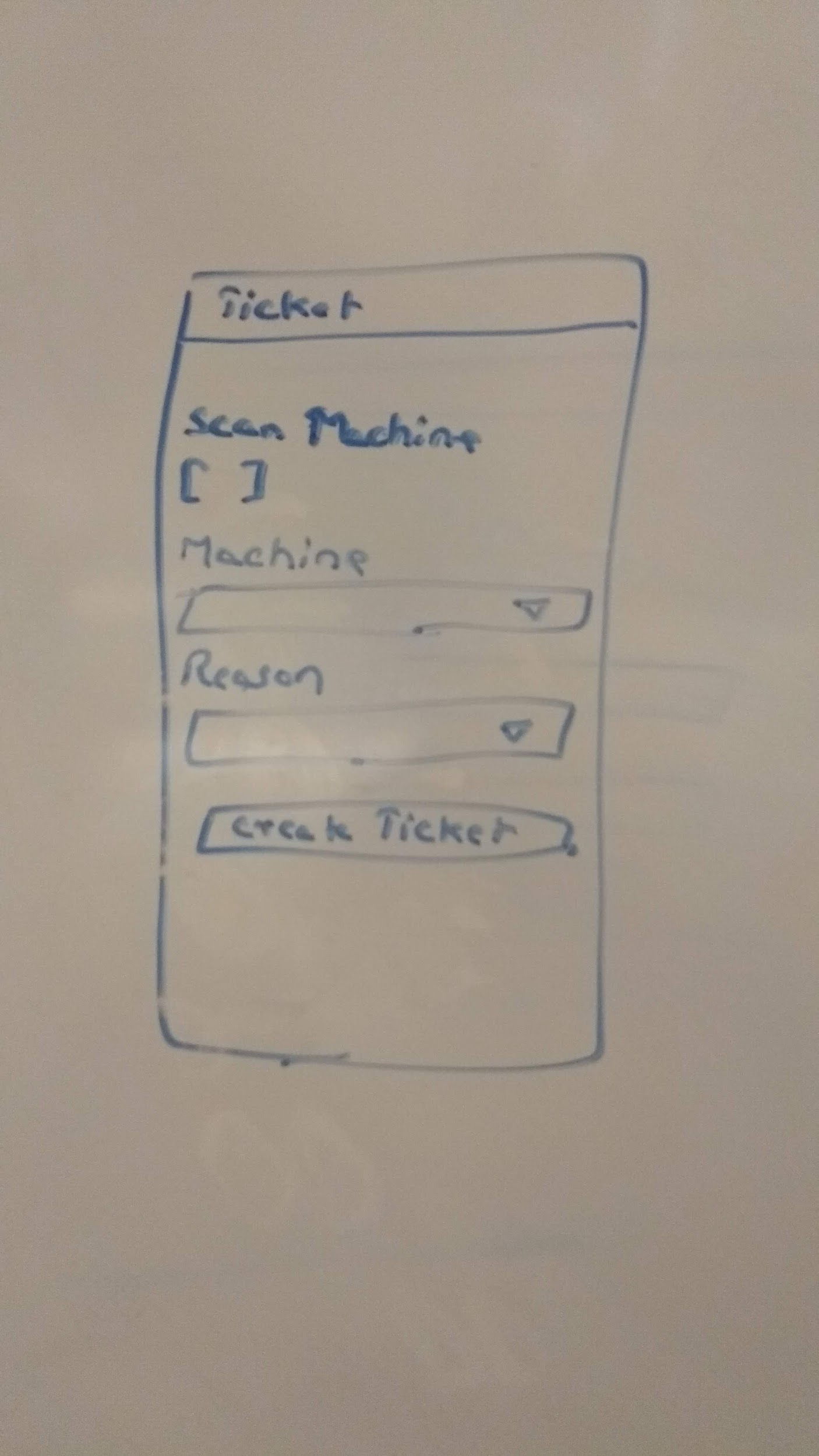
**Class diagram:**



**Sequence diagram:**

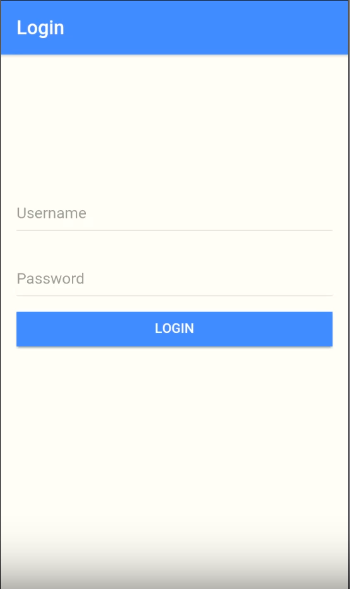


**Wireframe:**

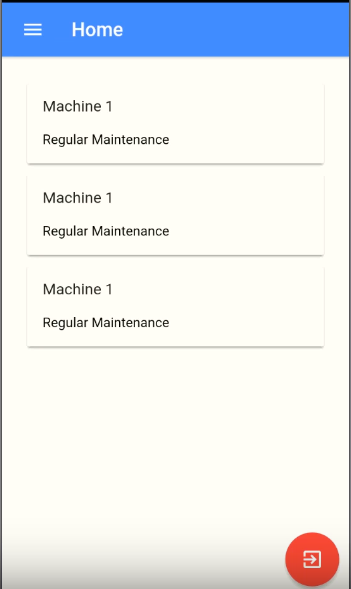
****

**Mobile Pages:**

**Login:**

****

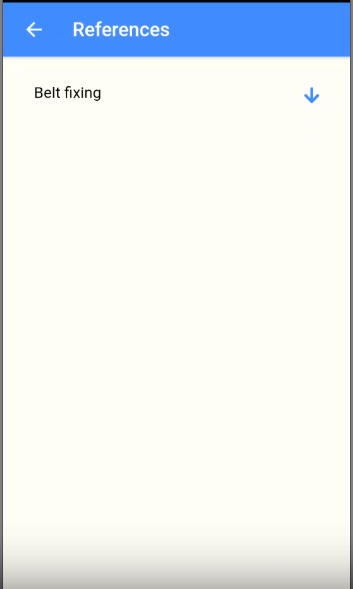
**Home**

****

**Steps:**

****

**References:**

****

**Source code:**

[**https://github.com/chkrish9/CS5551\_Team\_4\_Project**](https://github.com/chkrish9/CS5551_Team_4_Project)

**Video Url:**

**<https://www.youtube.com/watch?v=Mw6W-P02KGg>**

**References :**

<https://www.npmjs.com/package/ngx-qrcode2>

<https://getbootstrap.com/docs/3.3/components/>

<https://angular.io/docs>

<https://www.youtube.com/watch?v=DQ9pZ2NKXRo&index=2&list=PLillGF-RfqbZMNtaOXJQiDebNXjVapWPZ>

<https://www.npmjs.com/package/ng2-file-upload>

<https://www.npmjs.com/package/xlsx>

<https://www.npmjs.com/package/multer>

<https://socket.io/>

[https://www.npmjs.com](https://www.npmjs.com/)