Social Media in the Process Automation Industry

Distributed Software Development

Requirements Definition Document

Version 0.1

In co-operation with:

ABB

Title:
Social Media in the Process Automation Industry

Product name:
ABBConnect

Course:
Distributed Software Development

Document:

Requirements Definition Document

Participants:
Robert Gustavsson
Dimitrios Kostopoulos
Ditmar Parmeza
Akhlaq Malik
Pierfrancesco Ranieri
Marta Milaković
Mario Milas
Tomislav Vresk

Supervisor:
Federico Ciccozzi

Date:
November 6th, 2013

Revision History

Initials	Action	Date	Version
DK, TV, PR, AM	Initial Draft	22.10.2013.	0.01
PR	Review of requirements (general feeds, human feeds, and sensor feeds)	22.10.2013	0.01
DK	Sections Project Scope and Current solutions added	23.10.2013	0.02
DK	Functional & Non-functional refined	24.10.2013	0.03
DK	Use case diagrams added	25.10.2013	0.04
PR, MMi	Copy and finish use cases for Sensor, proofreading	25.10.2013	0.04
AM	Quality characteristics and sequence diagrams added	26.10.2013	0.06
PR	General requirements refined, completed UC10, SUC1, and SUC2	29.10.2013	0.07
AM,DK	Fixed the document skeleton	29.10.2013	0.07
DK	Business requirements added	29.10.2013	0.07
AM	Functional requirements table added	29.10.2013	0.07
RG	Fixed Simulator use-case and description	29.10.2013	0.07
DK	Fixed use case diagrams	30.10.2013	0.08
DK	Improved section 3.2 and 4	01.11.2013	0.09
DK,AM,PR	Completed initial version	06.11.2013	0.1

Contents

1 Introduction

1.1 Purpose of this document

This document aims to give a detailed description on how the applications that will be implemented are designed. This is done by using high-level, component and more detailed class charts. The charts will follow the UML-standards and will give a clear, structured overview of the system with the purpose of making the implementation more effective in the implementation phase. Furthermore, graphical user interfaces will be available for the front-end team.

1.2 Intended Audience

This document is intended mainly for the development team but also for the customer and supervisor in order to ensure that they are that all the requirements have been listed and taken note of. Team members will use this document as a reference while designing the application.

1.3 Scope

This document lists and provides a detailed description of requirements that will be useful in the software implementation process, which will take place from 04/11 till 26/12. Requirements are categorized and prioritized by the requirement team. Moreover, this document includes use-case diagrams and sequence diagrams used to describe them. The purpose of these diagrams is to give a clear overview of the interaction of users and other actors with the system. Additionally, they can be used for verifying all the specified functions are implemented in accordance to the requirements.

1.4 Abbreviations

In this section, we provide short explanations for each of the abbreviations that have been used and mentioned in this document in order to make sure that every abbreviation used here will be properly understood by everyone who will read the document.

ABB	(Asea Brown Boveri) A multinational corporation with focus on robotics, power technology and automation
AF	Activity feed
DU	Desktop user
FER	Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
GUI	Graphical User Interface
MDH	Mälardalen University, Västerås, Sweden
MU	Mobile user
PM	Project Manager
UI	User Interface
UML	Unified Modeling Language

Table 1: Abbreviations

2 Background and objectives

2.1 Customer

The customers are Aneta Vulgarakis and Jonas Bronmark from ABB.

Website: www.abb.com

Contact: Aneta.Vulgarakis@se.abb.com

2.2 Supervisor

Federico Ciccozzi, PhD - Mälardalen University

Contact: Federico.Ciccozzi@mdh.se

2.3 Project vision

The project vision implies building a desktop and a device application that would practically have the features of social media for communication within the ABB staff. The purpose of building this application is to facilitate the communication between the small factory employees as much as possible - when it comes to stating different technical or professional issues in the company, reporting them to other colleagues, and trying to provide and agree on reasonable solutions. For a more detailed project vision, go to the project plan and vision document.

2.4 Purpose & Benefits

ABB is one of the largest engineering companies in the world and specifically the branch accommodated in Västerås is cooperating with Mälardalen University. The customer, Aneta Vulgarakis and Jonas Bronmark, have set the improvement of information flow between factory employees as the main goal. In fact, they want to investigate if the introduction of a social media application in the daily communication process could be beneficial. The goal of the product is for employees to be able to share knowledge quickly and simple.

2.5 Project Scope

The expected outcome is a product able to control the flow of information and feasible for instant sharing. Hence a communication channel is needed that transfers collected data to all the interested groups of employees. This product should be used by both the users in the control rooms (DU and can be mobile users) and the other users without a fixed location (mobile users), with specific focus on their needs. Since their needs and tasks are quite different, it is essential that two separate applications are implemented. One of them will focus on the DU's activities as a web application, and one on the MU's activities as a mobile application.

The main concept is presented below in Figure 1, which shows a basic model of interaction. The system handles all information gathered from sensors as values and critical states (alarms), and forwards them as text to the whole factory group. The same can be achieved by the DU. Lastly, the product should help the MU to send his pictures, posts, comments, and distribute them in the factory.

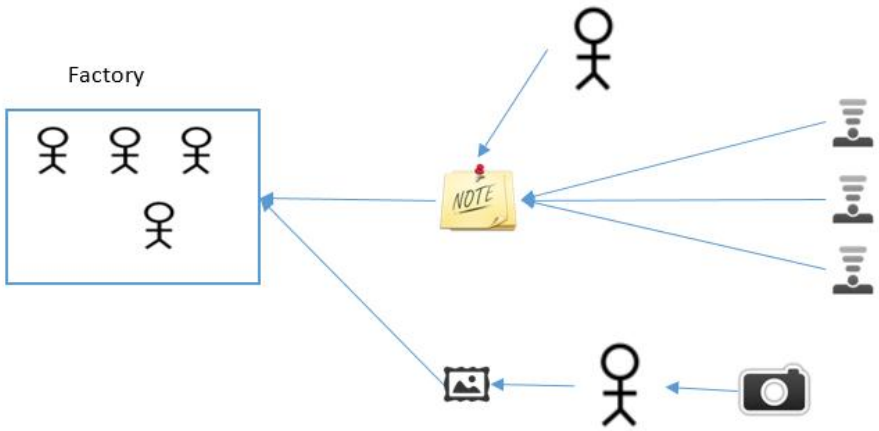

Figure 1: A basic model of how users and sensors will interact in the system. Sensors, MU, and DU will be able to send posts to the system, where all the users of a common environment will be able to see them. Furthermore the MU will be able to show pictures on the system.

The information shared on the system is called feeds. When a user publishes some type of information, the feed is updated with a result of the action. Every result of a user's action is stored and called an AF of the user.

2.6 Constraints

Taking into consideration the high level description of the product functions, two different type of constraints should be taken into account: technical constraints, which affect the outcome of functions due to architectural limitations, and constraints introduced by the customer.

Technical constraint:

Both applications will not be able to update the AF or other information in the network without an Internet connection.

Customer constraint:

It has been specified that two versions of the application are required, one desktop and one mobile. Moreover, the team responsible for the development of the system, will not be able to access real sensors and a simulation of their interaction is required.

3 Business Requirements

3.1 Description of Current Business State

Currently the flow of information is not managed by any system, nor is any automated way of retrieving data available. There are two identified issues: leaks in information or delays. The former relates to information transferred from a category of employee called the Control Engineer. The Control Engineer is located inside the control room and he is the main responsible for the status of processes and tasks. He needs to inform his fellow co-workers, who are arriving on the next shift, on any incomplete tasks or notify them for an out of the ordinary equipment (sensor) behavior. The preferred way of passing information from one shift to another had been through sticky notes. However, some of those sticky notes could never reach the recipient, and a leak in information was introduced.

The other important concept was noticed in a normal checking procedure done by another group of employees who are working outdoors. Those employees are considered as inspectors too. They are responsible for checking the equipment located in different places and their task is to inform about damaged equipment by taking its picture. Later on the engineers need to document and store those failures, but the followed procedure adds complexity and limits the spread and time of arrival of information. In more detail, the engineers need to find a computer, connect the device that took the picture, send it as an email to some co-workers, and store it locally for future use. This means that a group of people will remain uninformed in case of a failure, or maybe the file will not be found after a while.

3.2 Specific Business Requirements

This section's purpose is to describe and summarize the functionality that the customer believes and evaluates as needed for the delivered product. Most of those identified business level requirements can later on be translated as functional behavior. In order to categorize the stakeholder's needs and identify their importance, we prioritize them. Four priority levels will be used to group the requirements: very high, high, medium and low. Very high implies requirements whose presence is crucial in fulfillment of the purpose of the project. A requirement is considered of high priority when it is a critical requirement necessary for the next release. Medium priority requirements imply non-critical system actions that are fairly desired by the stakeholders. Lastly, low priority requirements imply a functional addition, which would be qualitatively preferable to have at some point and if the human resources permit it.

BR ID	BR Description	Priority
BR.01	A social media application is needed that contains sensor and human user information.	very high
BR.02	A mobile application is needed, which is not limited only to outdoor purposes.	very high
BR.03	Both applications should be designed focusing on usability.	very high
BR.04	Both applications shall provide a way through which users shall post to the feed.	very high
BR.05	All information shall be available for the MU and DU.	very high

Table 2: Business requirements prioritized as very high

BR ID	BR Description	Priority
BR.06	DU shall be informed about alarm failures by the previous shift.	high
BR.07	There shall be a flexible way of categorizing the feeds.	high
BR.08	Sensor shall be able to post alarms to the feed when they reach critical values.	high
BR.09	Sensor history data shall be presented graphically.	high
BR.10	Sensors shall be able to post current status to the feed.	high
BR.11	Sensors shall be able to notify, when their boundary limit for triggering an alarm, has been changed.	high
BR.12	Human users shall be able to trace the source of a post.	high
BR.13	MU shall be able to take a photo with the mobile application and post it to the feed	high
BR.14	DU shall be able to retrieve information about previous shift's unfinished tasks.	high
BR.15	The mobile application and the desktop application should provide different services.	high
BR.16	Posts published by a human user out of his office hours should be distinguished.	high

Table 3: Business requirements prioritized as high

BR ID	BR Description	Priority
BR.17	Human user should be able to state his opinion for a post.	medium
BR.18	A predefined search filter should be available for the user.	medium
BR.19	There shall be an easy way of informing interested parties	medium

Table 4: Business requirements prioritized as medium

BR ID	BR Description	Priority
BR.20	The application may support private messaging.	low
BR.21	Administrator privileges for creating accounts.	low
BR.22	There shall be an administrator for managing users.	low

Table 5: Business requirements prioritized as low

4 Functional Requirements

Functional requirements are categorized in relation to their working context. Since there will be three applications in the whole project (web application, mobile application and control system simulator), there should be three tables, one per application. However, since most of the requirements are similar for the web and mobile version, one more table was included to summarize common features. Thus, Table 6 displays common functional requirements for both web and mobile applications. Table 7 and Table 8 are related to web and mobile application specific requirements respectively. The last table, Table 9, describes requirements specified for the simulator system.

All the tables follow a similar pattern. Hence, the first column contains the functional requirement ID for traceability purposes in the whole development process. Column two includes business requirements IDs, which were already defined in the previous section. The goal of that column is to highlight the relations between those two types of requirements and ensure that all the business needs are covered. It is worth mentioning that whenever an ID is missing from the second column, it is assumed that the requirement was introduced by the project team. The third column gives a brief description of each requirement and the last column states their priority in the development phase.

FR ID	BR ID	FR Description	Priority
FR.01	BR.04	Every user shall be able to post notes to the system.	very high
FR.02	BR.04	As a human user, you shall be able to upload media files to the system.	very high
FR.03		The system should characterize every post/action with the time and date it was published.	very high
FR.04	BR.07 BR.14	The system shall provide static and dynamic ways of filtering feeds.	very high
FR.05	BR.05 BR.07	A human user shall be able observe the AF of a sensor/human.	very high
FR.06	BR.12	The system shall include user information in every post/action.	very high
FR.07		Every human user shall be able to log in to both applications by providing username and password.	high
FR.08		Every user shall have his own AF.	high
FR.09	BR.08	A warning should be posted whenever the value of a sensor is critical or its boundary value was changed.	high
FR.10		The system shall provide a way of connecting a human user with a post.	medium
FR.11	BR.05	A human user shall be able to view the general information about another user or a sensor.	medium
FR.12	BR.17	A human user should be able to comment to a note.	medium
FR.13	BR.18	A human user should be able to save a filter for the feeds.	medium
FR.14	BR.12	The system shall include location information in every post.	medium
FR.15	BR.12	The location information shall be presented with an indoor map picture.	medium
FR.16		Posts published outside of working hours of a user shall be distinguished with less importance.	medium

Table 6: Common functional requirements for web and mobile applications

FR ID	BR ID	FR Description	Priority
FR.17	BR.09 BR.06	The system shall provide a way for the human user to get summarized sensor data.	high
FR.18	BR.09	The summarized sensor data shall be presented in a graphical way.	high
FR.19		DU shall view the current status of selected sensors through widgets.	medium

Table 7: Functional requirements specific for web application

FR ID	BR ID	FR Description	Priority
FR.20	BR.13	As a mobile engineer, the user shall be able to take a picture with his phone.	very high
FR.21		As a mobile engineer, the user may post his current location.	low

Table 8: Functional requirements specific for mobile application

FR ID	BR ID	FR Description	Priority
FR.22	BR.11	Simulated control system shall be able to change the boundary value of a sensor.	high
FR.23	BR.10	Simulated sensors shall be able to post their current status.	high

Table 9: Functional requirements specific for control system application

FR.01/FR.02/FR.03/FR.06 Publish

The core functionality of this system is to post notes, specifically for the upcoming working shift. Whenever a user publishes information, he has to specify its importance (normal or high). Since the posts should be traceable, they should include information about its owner. Those information can be a user name, the shift (schedule time under the post was published) and the place (control room/ coordinates) where the post took place. Time is also an important information and the application should include it automatically to every post.

The simplest content of a post is the text representation, and it can be more complex with media file support. The highest priority is to be able to add text notes and pictures. All other kinds of media like video and audio messages are secondary goals.

FR.04/FR.05 Filter feeds

A user should be able to decide which feeds he wants to view on the main page. Even though the system will have a default priority to display all the feeds, the user should be able to specify a new criterion that allows him to filter the information. The criteria can be given in a static way, like predefined lists that filter posts by priority or by user's role (sensor, human user). The system should also support dynamic ways of filtering, like selecting a user's name and viewing his AF.

FR.07 Log in

Human users should be provided with an option that lets them insert their given credentials. It should be the main page showed to them, before accessing any other functionality.

FR.08 AF

All activities of a user should be posted on his AF. This AF is a continuous feed of actions that were taken by the user. Whenever a user interacts with an action to another user's AF, then the latter should be notified.

FR.09 Sensor notification

Each sensor has predefined lower and upper boundary values. Whenever those values are changed by the simulator system, a warning should be posted and the widgets in the web application, related to current sensor data, should be updated. It is also very important for the system to notify (alarm) when a current value of a sensor violates the boundaries.

FR.10 Refer an employee

A way shall be provided for every human user to connect another user to a post by providing his name. The reason for associating an employee to a post is because he has interest in the content of the post.

FR.11 View Profile

A user should be able to view profile information of another user. Each profile is open for the whole company. Depending on the user's nature (sensor/human user), different information is shown. For human user a name, an email and a phone number should be displayed, while for a sensor a name and location are sufficient.

FR.12 Commenting

A human user will comment on posts. That way a simplest way of asynchronous communication can be supported.

FR.13 Save filters

A user should be able to save the information of a filter that he has set on a device. This setting could be interchangeable depending on the user needs. These saved filters are related with the user account, and they are available for the user on every machine he uses.

FR.14/FR.15 Location information

In order to improve traceability of posts and easing of dynamic search, location information should be included in every post. This information can be a working room name for a DU or a place added by the MU. For locations inside a working place an image representation should be available that displays the relevant position of the rooms on a map. The map will display rectangles as rooms and a dot emphasizing the current room.

FR.16 Non-important posts

There is a possibility that a user will post some type of information which is not related to a work activity to the feeds. This case depends on the time when a post is submitted. Thus, if a human user publishes outside his working hours, then the post is automatically prioritized as low priority.

FR.17/ FR.18 Summarized feed

A user can see all notes taken by his coworkers in the previous shift. The system should be able to somehow highlight problems that occurred several times and problems that were critical. An option will be available to the user in order to summarize especially sensor problems or sensor activity through graphs. The DU can select a period of time and the system shall return a graph representation of a sensor's data during that period.

FR.19 View sensor data

A DU can also observe the current data and status of a sensor in the plant outside the office. The representation of this data should be somehow aggregated and shown in a widget per sensor.

FR.20 Take a picture

A MU should be able to take a picture with his mobile device and upload it to the server. The picture is taken by using the camera of the mobile device and the file shall be included as post content without expecting the user to upload it.

FR.21 Post location

When the MU publishes to feed, the application also includes his geological coordinates.

FR.22 Sensor boundary values

For testing purposes, the simulated environment shall give the option of changing the boundary values of a sensor. This way, warnings of that type can be posted.

5 Non-Functional Requirements

Security requirements

- Each user will have access to the feeds of his working environment.
- A user of a working environment should be able to see all the members of the same group.

Performance requirements

- Time of selecting data from database must be less than 2.5 seconds.
- Web application should work on all common web browsers (versions older than IE9 excluded)

Governmental requirements

On the 24th of October 1998 the Swedish Personal Data Act (1998:204) came into force. Hence, the application should not be violating this revised from 1973 act and should protect all personal data according to this governmental constraint.

Portability

The system shall be portable enough to fulfill the following criteria:

- One Browser (Google Chrome) optimized to ensure the desktop usage.
- Window Phone application as a mobile platform.

Usability

The main focus of the project is to provide a simple, intuitive and easy to use user interface. These mentioned goals are similar for the desktop and also for the handheld device. Usability will be measured in two different ways, in manner of time needed by a user to achieve a goal and in training time. In more detail:

- The user should be able to find the desired action in less than 10 clicks.
- The training period of a customer that will let him be familiar with the system is set to two days.

Color selection

The working environment of the engineers (dark or bright) should be taken into account, when the colors of the GUI are decided.

6 Use Case Models

6.1 Actors

There are three identified actors that use the system. Two of them are recognized as human users, and one of them as a simulated sensor system. Every participant of the system will act inside a specific context/environment, and different functionality should be available to him. The types of defined users are explained in more detail below.

Desktop user: Since this person is familiar with computers, a good level of knowledge in technology is expected. However, the time limitations and the demands of the user require a user-friendly interface. Filtering of information should be done in a quick way, and sensor information is most important for this user.

Mobile user: His job demands the use of a mobile device (either a tablet or a smart-phone) that makes the process easier. The size of the screen of these portable devices is limited and there should be a limit in the amount of displayed features. Mobile users are supposed to have a basic experience and knowledge of using these devices. Therefore, the user interface should be as simple as possible and allow fast navigation.

Simulator: Sensors, as mentioned, are also responsible for spreading their values and notifying users. They are placed in different places in the factory to continuously measure the state, temperature and other properties of a machine. In the prototype version that will be committed, all activities are generated by the system. This way the stakeholders will get an idea on how the data can be presented to the users. So, sensors will be simulated with custom made tools that are responsible for sending sensor data to the database. However, these simulator actions will be guided by an external user (admin), who is considered as a supporting actor.

6.2 Use case model: Desktop User

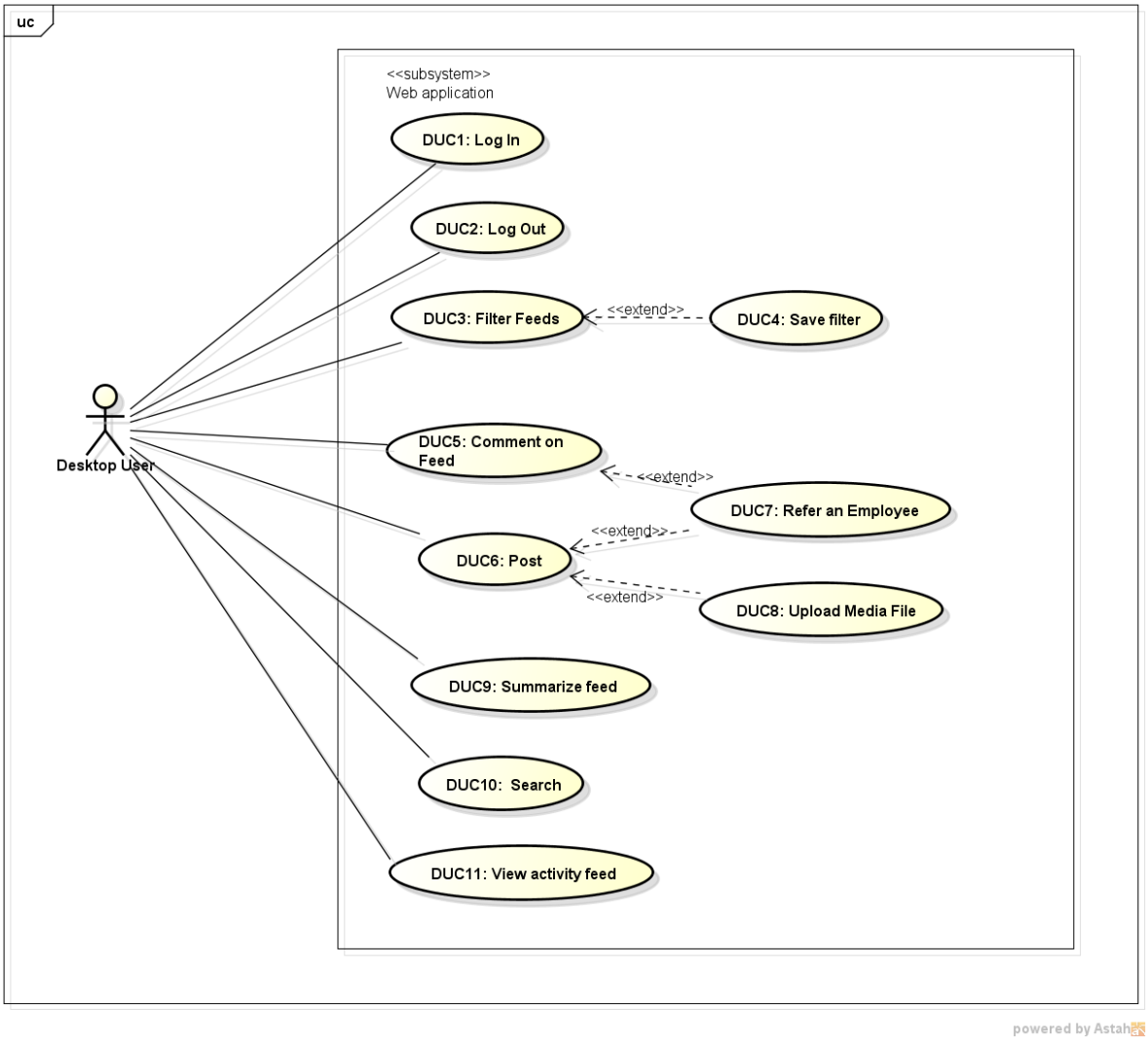

Diagram 1: Use case diagram of desktop user

Use case name:	**DUC1: Log in**
Participating Actors:	Desktop Engineer
Preconditions:	Log out
Basic Flow:	1. The user selects the username and password textboxes and fills in his credentials. 2. The system gets the request and checks the database. 3. User is found on the database. 4. Authentication succeeded.
Alternative flows:	3a User is not included in the database. 1. An alert informs for wrong credentials.

Post conditions:	The user is redirected to his profile page.
Special Requirements:	

Use case name:	**DUC2: Log out**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. The user clicks the log out button. 2. The system redirects the user to the public page.
Alternative flows:	
Post conditions:	The user views the public page.
Special Requirements:	

Use case name:	**DUC3: Filter feeds**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User chooses to filter the feeds by their nature (sensor/human/mixed) 2. System gets the request. 3. System updates the feeds.
Alternative flows:	1a. A different filter may be chosen regarding the importance of a feed. 1. User chooses to filter the feeds by their priority. 2. Return at 2.
Post conditions:	The notification page updates its content.
Special Requirements:	

Use case name:	**DUC4: Save filter**
Participating Actors:	Desktop Engineer
Preconditions:	Log in, the user has set a filter
Basic Flow:	1. User chooses to save the filter 2. The system gets the request 3. The system saves the filter for the user

Alternative flows:	1. User changes the setting of a saved filter 2. The system gets the request 3. The system saves the changes of the existing filter
Post conditions:	The filter should be shown with the other saved filters
Special Requirements:	The filter is related to the user's account, and will be available on every machine that the user uses.

Use case name:	**DUC5: Comment on Feed**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. The user spots a post that he is interested in. 2. User types his comment in the textbox below the post. 3. User presses the "Comment" option. 4. System validates the text of the comment. 5. System adds the comment to the post.
Alternative flows:	3a User presses uses other combination of button to commit. 1. User presses the Enter button. 2. System validates the text of the comment. 3. System adds the comment to the post.
Post conditions:	A comment is added below the post.
Special Requirements:	

Use case name:	**DUC6: Post**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. The user selects a textbox to add text. 2. User types the note. 3. User presses the "Post" action. 4. System stores the post. 5. System updates the feeds.
Alternative flows:	2a User selects to attach a media file too. 1. Upload a media file. 2. File is attached. 3. System returns to 4 (basic flow). 2b User selects to tag an employee. 1. Tag an employee. 2. Return to 4.

Post conditions:	A post is created both on feeds page and on the AF page of the employee.
Special Requirements:	

Use case name:	**DUC7: Refer an employee**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. The user selects the "Tag" option. 2. System shows a window and asks for an employee name. 3. User types a name. 4. System checks the database for the user with the corresponding name. 5. Employee is tagged.
Alternative flows:	4a User is not included in the database. 1. An alert informs of wrong name provided. 2. System hides the tag window.
Post conditions:	An employee is added in a post as tagged.
Special Requirements:	

Use case name:	**DUC8: Upload a media file**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User presses the "Attach File" option. 2. System request for file location. 3. User selects the file. 4. System checks the file extension 5. System uploads the media file.
Alternative flows:	3a User selects to upload an invalid file. 1. System shows a message to inform the user. 2. System returns to 2.
Post conditions:	A media file is uploaded to the AF of the user.
Special Requirements:	

Use case name:	**DUC9: Summarize feed**
Participating	Desktop Engineer

Actors:	
Preconditions:	Log in, Edit Profile (which shift)
Basic Flow:	1. User clicks on previous shift 2. User gets a summarized feed of the previous shift 3. User can scroll through the feed and can open also the whole activity of the previous shift
Alternative flows:	2a. There was no previous shift 1. Return to all available shifts. 2b. User has not specified his shift in his profile 1. Return to his profile page in the edit mode with the message that he should add his shift.
Post conditions:	
Special Requirements:	

Use case name:	**DUC10: Search**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User selects the “Search” textbox. 2. User types name of a sensor/employee. 3. System checks on the database. 4. System returns the results.
Alternative flows:	3a. Name was not found in the database. 1. System notifies the user that no users were found.
Post conditions:	The result area is updated with the results.
Special Requirements:	

Use case name:	**DUC11: View feeds**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User clicks on another user’s AF. 2. System shows all activities (media, notes) in a sequence.
Alternative flows:	
Post conditions:	

Special Requirements:	

6.3 Use case model: Mobile User

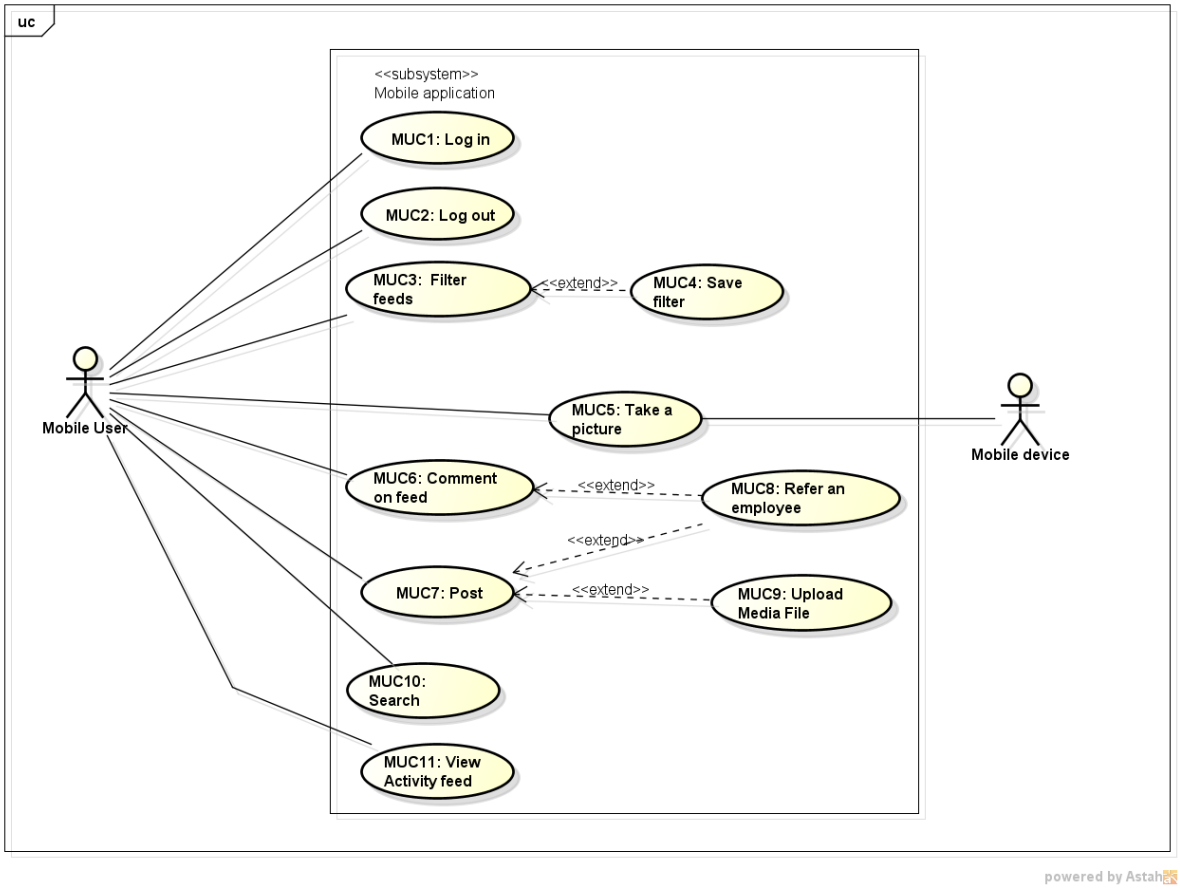

Diagram 2: Use case diagram of mobile user

Use case name:	**MUC1: Log in**
Participating Actors:	MU
Preconditions:	The user should not be logged in yet
Basic Flow:	1. User views the login screen. 2. User taps the text fields with his/her username and password. 3. System checks the credentials. 4. System shows the feeds screen.
Alternative flows:	User is not included in the database, or taps wrong credentials. 1. An alert informs the user for the wrong credentials.
Post conditions:	The mobile engineer is authenticated.
Special Requirements:	

Use case name:	**MUC2: Log out**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. User presses the “Log Out” action. 2. The system redirects the user to the login page.
Alternative flows:	
Post conditions:	The mobile engineer has signed out and the login page is shown.
Special Requirements:	

Use case name:	**MUC3: Filter feeds**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. User taps the setting button. 2. User chooses to filter the feeds by user or priority.
Alternative flows:	The user choose an existing filter
Post conditions:	The system show to the user the feeds filtered.
Special Requirements:	

Use case name:	**MUC4: Save filter**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. User sets a filter. 2. User chooses to save the current filter setting. 2. System stores the filter setting for the user.
Alternative flows:	
Post conditions:	The current filter is available on the saved filter.
Special Requirements:	

Use case name:	**MUC5: Take a picture**
Participating Actors:	MU, mobile device
Preconditions:	MU logged in
Basic Flow:	1. The MU selects the camera icon. 2. The MU selects the option to take a picture. 3. System initiates the camera. 4. User takes a picture. 5. System stores the picture. 6. Upload of media file.
Alternative flows:	If some other process is using the camera, the system should inform the user with a message. If the user want to upload a photo that is stored on the mobile device: 1. The MU selects the camera icon. 2. The MU selects the option to upload a stored photo. 3. The MU selects a photo stored on the mobile device. 4. The system uploads the media file.
Post conditions:	A picture is taken and added to the feed.
Special Requirements:	The system should be able to use the camera, and to access to the stored files of the mobile device

Use case name:	**MUC6: Comment on feed**
Participating Actors:	MU
Preconditions:	The MU should be authenticated.
Basic Flow:	1. The MU chooses a feed. 2. The MU taps the settings button. 3. The system shows the options. 4. The MU taps the comment option. 6. The MU selects to add a comment. 7. The system stores the comment.
Alternative flows:	
Post conditions:	A comment by the MU should be shown on the chosen feed
Special Requirements:	

Use case name:	**MUC7: Post**
Participating Actors:	MU

Preconditions:	The user should be authenticated.
Basic Flow:	1. The MU selects the button to add posts. 2. A text box pops up for writing a post. 3. The MU write and submit the post. 4. The system saves the post. 5. System updates the feeds.
Alternative flows:	
Post conditions:	The post is shown on the main page
Special Requirements:	

Use case name:	**MUC8: Refer an employee**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. User chooses an existing feed or a new feed. 2. User taps the option button. 3. System shows the options. 4. User taps the comment option. 5. System shows the comments for the feed or the comment. 6. User selects to add the comment. 7. System stores the comment.
Alternative flows:	
Post conditions:	A comment by the MU should be shown on the chosen feed for the comments, and on the main page for new posts.
Special Requirements:	It is possible to refer only workers in the same work environment

Use case name:	**MUC9: Upload media file**
Participating Actors:	MU
Preconditions:	The user should be authenticated, the user is writing a post.
Basic Flow:	1. The MU taps the option button. 2. The system shows the options. 3. The MU taps the upload media file option. 4. The MU selects a media file. 5. The system adds the selected media file.
Alternative flows:	

Post conditions:	When the post is submitted, the media file is added to the post
Special Requirements:	

Use case name:	**MUC10: Search**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. The MU type the name of another user or a sensor 2. The MU taps the search button 3. The system searches the typed user or sensor
Alternative flows:	If the search found nothing, the system show a message to the user
Post conditions:	All the results for the search are shown to the user
Special Requirements:	

Use case name:	**MUC11: View activity feed**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. The MU taps the AF button 2. The system retrieves the AF of the selected user
Alternative flows:	
Post conditions:	All the feeds posted by the MU are shown.
Special Requirements:	

6.4 Use case model: Control System

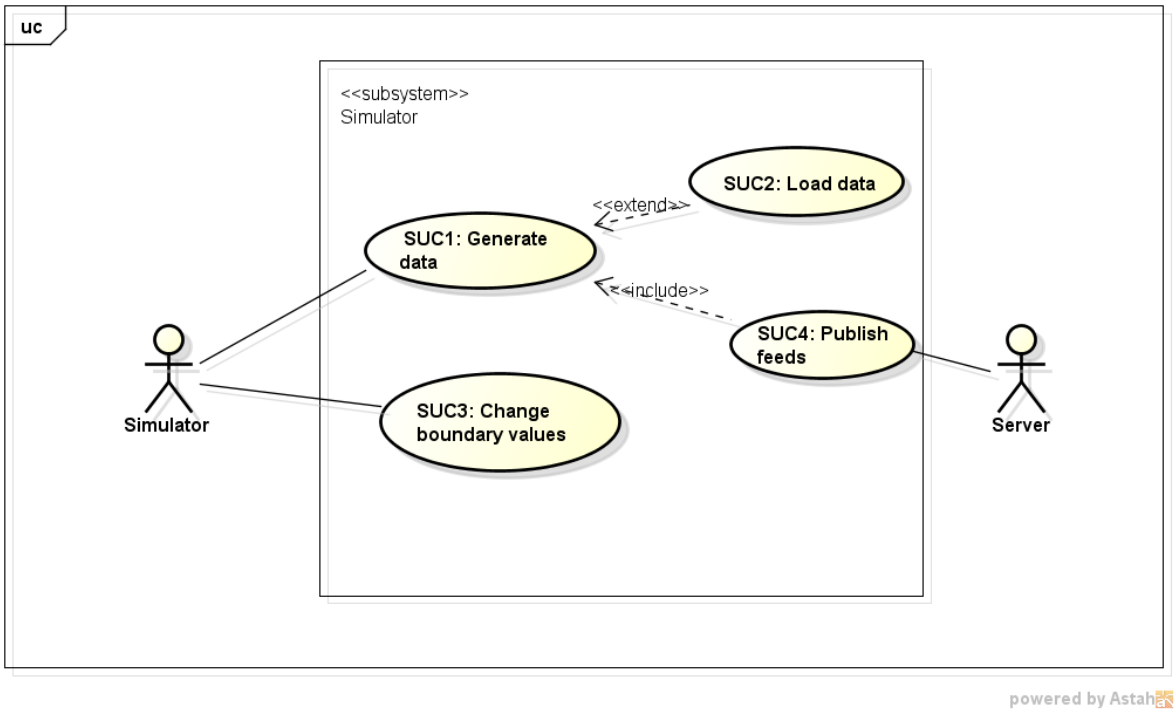

Diagram 3: Use case diagram of simulator

Use case name:	**SUC1: Generate data**
Participating Actors:	Simulator
Preconditions:	Simulator user has defined the boundaries of the sensor and has also specified the wanted time of posting
Basic flow:	1. Simulator user clicks “Generate Data” 2. The simulator randomly picks 100 values that are outside and inside the sensor boundaries. 3. The simulator sets a higher priority to the values outside of the boundaries 4. The simulator saves the values internally
Alternative flows:	
Post conditions:	
Special Requirements:	

Use case name:	**SUC2: Load Data**
Participating Actors:	Simulator
Preconditions:	The simulator user has defined the sensor boundaries
Basic Flow:	1. The simulator user clicks “Load Data”. 2. The simulator provides an open dialog. 3. The simulator user chooses the wanted file. 4. The simulator loads the file. 5. The simulator adds higher priority to the values that are outside of the boundaries. 6. The simulator stores the data internally.
Alternative flows:	4a The user has not specified a high or low boundary. 1. A message box informs the user. 2. User types a value. 3. Return to step 1.
Post conditions:	
Special Requirements:	

Use case name:	**SUC3: Change Boundary Values**
Participating Actors:	Simulator
Preconditions:	
Basic Flow:	1. The simulator user enters the lower and upper bound of the sensor in the text boxes
Alternative flows:	
Post conditions:	
Special Requirements:	

Use case name:	**SUC4: Publish feeds**
Participating Actors:	Simulator
Preconditions:	The simulator user has chosen to load or generate sensor values
Basic Flow:	1. The simulator user clicks “Publish Values” 2. The simulator goes through all saved sensors 3. The simulator sends the values for all sensors to the database with the time difference of 1 second.

Alternative flows:	
Post conditions:	The generated values are stored in the database
Special Requirements:	

7 List of Tables