Social Media in the Process Automation Industry

Distributed Software Development

Requirements Definition Version 0.2

In co-operation with:

ABB

Title:
Social Media in the Process Automation Industry
Course:
Distributed Software Development
Document:
Requirements Definition
Participants:
Robert Gustavsson
Dimitrios Kostopoulos
Ditmar Parmeza
Akhlaq Malik
Pierfrancesco Ranieri
Marta Milaković
Mario Milas
Tomislav Vresk
Supervisor:
Federico Ciccozzi
Date:
January 13, 2014

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
PR	Made final version Requirements copy, add comment, and fix some text part	26.12.2013	0.11
DK	Include two more requirements and fix some use case text	28.12.2013	0.12
DK	Update use case diagrams, refined sections	10.01.2014	0.13
MM	Overall review	13.01.2014	0.2

Table of Contents

1 Document Control

1.1 Document 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 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.

1.2 Abbreviations

Explanation	Abbreviations
Desktop user	DU
Mobile user	MU
Activity feed	AF

Table 1. *Abbreviations*

2 Introduction

2.1 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.

2.2 Project Scope

Currently the internal flow of information between the employees is not managed by any system, nor any automated way of retrieving data is available. 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 (called DUs, who can also be MUs) and the users without a fixed location (MUs), 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.

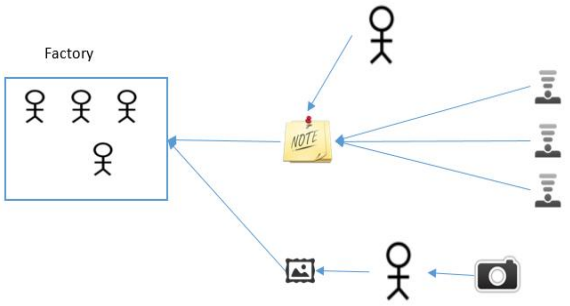

***Figure 1**: A basic model of how users and sensors will interact in the system. Sensors, MU, and DU will be able to send posts to the system, where all the users of a common environment will be able to see them. Furthermore the MU will be able to show pictures on the system.*

In the product's context feeds are considered as the information shared on the system. 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.

2.3 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 any automated way of retrieving data is available. There are two identified issues: leaks in information or delays. The former relates to information transferred from a category of employees called the Control Engineers. A 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, for 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 or nonfunctional 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.

Priority	BR Description	BR ID
very high	A social media application is needed that contains sensor and human user information.	BR.01
very high	A mobile application is needed, which is not limited only to outdoor purposes.	BR.02
very high	Both applications should be designed focusing on usability.	BR.03
very high	Both applications shall provide a way through which users shall post to the feed.	BR.04
very high	All information shall be available for the MU and DU.	BR.05
very high	An API should be provided having in mind portability for different type of mobile operating systems	BR.06

Table 2. *Business requirements prioritized as very high*

Priority	BR Description	BR ID
high	DU shall be informed about alarm failures by the previous shift.	BR.07
high	There shall be a flexible way of categorizing the feeds.	BR.08
high	Sensor shall be able to post alarms to the feed when they reach critical values.	BR.09
high	Sensor history data shall be presented graphically.	BR.10
high	Sensors shall be able to post current status to the feed.	BR.11
high	Sensors shall be able to notify, when their boundary limit for triggering an alarm, has been changed.	BR.12
high	Human users shall be able to trace the source of a post.	BR.13
high	MU shall be able to take a photo with the mobile application and post it to the feed	BR.14
high	DU shall be able to retrieve information about previous shift's unfinished tasks.	BR.15
high	The mobile application and the desktop application should provide different services.	BR.16
high	Posts published by a human user out of his office hours should be distinguished.	BR.17

Table 3. *Business requirements prioritized as high*

Priority	BR Description	BR ID
medium	Human user should be able to state his opinion for a post.	BR.18
medium	A predefined search filter should be available for the user.	BR.19
medium	There shall be an easy way of informing interested parties	BR.20

Table 4. *Business requirements prioritized as medium*

Priority	BR Description	BR ID
low	The application may support private messaging.	BR.21
low	Administrator privileges for creating accounts.	BR.22
low	There shall be an administrator for managing users.	BR.23

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. Then 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 or that it is connected to a nonfunctional requirement. The third column gives a brief description of each requirement and the last column states their priority in the development phase.

Priority	FR Description	BR ID	FR ID
very high	Every user shall be able to post notes to the system.	BR.04	FR.01
very high	As a human user, you shall be able to upload media files to the system.	BR.04	FR.02
very high	The system should characterize every post/action with the time and date it was published.		FR.03
very high	The system shall provide static and dynamic ways of filtering feeds.	BR.08 BR.15	FR.04
very high	A human user shall be able observe the AF of a sensor/human.	BR.05 BR.08	FR.05
very high	The system shall include user information in every post/action.	BR.13	FR.06
high	Every human user shall be able to log in to both applications by providing username and password.		FR.07

high	Every user shall have his own AF.		FR.08
high	A warning should be posted whenever the value of a sensor is critical or its boundary value was changed.	BR.09	FR.09
high	The previous posts of a user should be displayed whenever his profile page is viewed.		FR.10
medium	The system shall provide a way of connecting a human user with a post.		FR.11
medium	A human user shall be able to view the general information about another user or a sensor.	BR.05	FR.12
medium	A human user should be able to comment to a note.	BR.18	FR.13
medium	A human user should be able to save a filter for the feeds.	BR.19	FR.14
medium	The system shall include location information in every post.	BR.13	FR.15
medium	Posts published outside of working hours of a user shall be distinguished with less importance.		FR.16

Table 6. *Common functional requirements for web and mobile applications*

Priority	FR Description	BR ID	FR ID
high	The system shall provide a way for the human user to get summarized sensor data.	BR.10 BR.07	FR.17
high	The summarized sensor data shall be presented in a graphical way.	BR.10	FR.18
medium	DU shall view the current status of selected sensors through widgets.		FR.19
medium	The user should be able to update his profile picture.		FR.20
low	The location information shall be presented with an indoor map picture.	BR.13	FR.21
low	The most recent user's activity shall be described graphically.		FR.22

Table 7. *Functional requirements specific for web application*

Priority	FR Description	BR ID	FR ID
very high	As a mobile engineer, the user shall be able to take a picture with his phone.	BR.14	FR.23
low	As a mobile engineer, the user may post his current location.		FR.24

Table 8. *Functional requirements specific for mobile application*

Priority	FR Description	BR ID	FR ID
high	Simulated control system shall be able to change the boundary value of a sensor.	BR.12	FR.25
high	Simulated sensors shall be able to post their current status.	BR.11	FR.26
medium	Simulated control system shall be able to set the time interval for storing sensor values.		FR.27

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 kind 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 new criteria 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 View Profile page

When the system redirects to the profile page of a user, the most valuable information is all his previous posts. More account specific profile information will be given as FR.12 describes.

FR.11 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.12 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.13 Commenting

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

FR.14 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.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.

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 Update profile image

When a DU visits his profile page, he shall be able to update his profile image or avatar. This image is the one that characterizes the user in some of his actions (included in post or comment).

FR.21 Location visual representation

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.22 Graphical activity representation

When a profile page of a user is viewed, three different graphs shall be presented. The first should show the number of posts per date for the respective user, while the second one will be more detailed in the time period of the last week. The last graph shall present the total created post grouped by their post type (priority).

FR.23 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.24 Post location

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

FR.25 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.

Even though one mobile platform is enough to meet the customer's needs, the architecture of the system should be created having in mind deployment in more mobile operating systems.

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

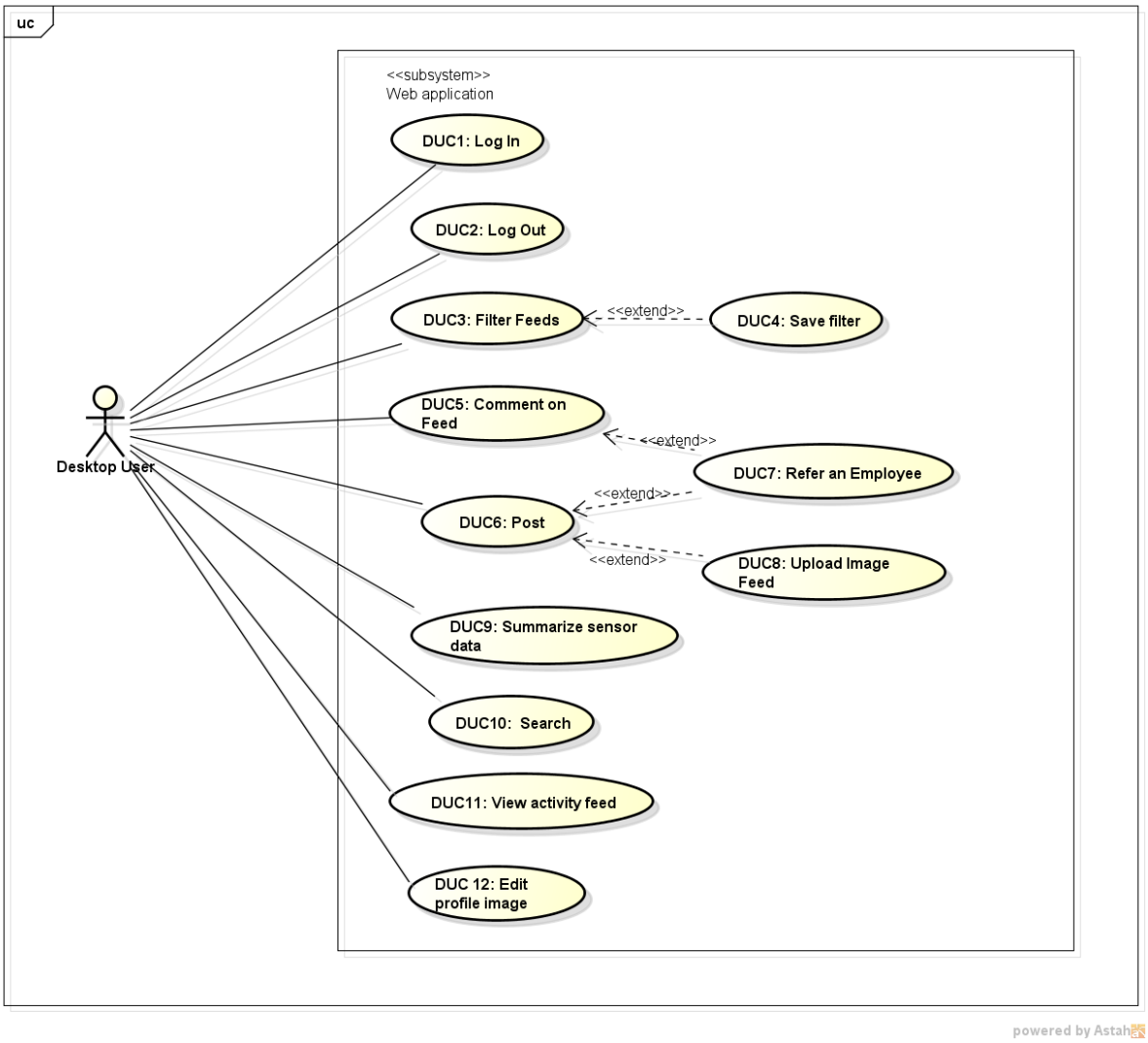

Figure 2: *Use case diagram of desktop user*

Use case name:	**DUC1: Log in**
Participating Actors:	Desktop Engineer
Preconditions:	
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.
Postconditions:	The user is redirected to the main feed page.
Special Requirements:	

Table 10. *User case 1, Login in*

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

Table 11. *Desktop use case 2, Log out*

Use case name:	**DUC3: Filter feeds**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User chooses to filter human feeds by priority (work, vacation or sticky note) 2. System gets the request. 3. System updates the feeds.
Alternative flows:	1a. A time filter may be chosen. 1. User chooses to filter the feeds and receive feeds from a specific time period. 2. System shows a pop up message showing one textbox for the starting time and one textbox for the ending time. 3. User selects the text boxes and a calendar appears. 4. User selects a starting date and an ending date. 5. User clicks on button “Save changes”. 6. System updates the feeds. 7. User closes the pop-up window. 1b. User chooses to view only the feeds of previous shift. 1. User selects the last shift option. 2. System retrieves the feeds from last shift. 3. System shows the feeds.
Postconditions:	The notification page updates its content.
Special Requirements:	

Table 12. *Desktop use case 3, Filter feeds*

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:	
Postconditions:	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 or device that the user uses.

Table 13. *Desktop use case 4, Save filter*

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 selects the textbox below the post. 3. System increases the textbox size. 4. User types his comment. 5. User presses the "Post" button. 6. System validates the text of the comment. 7. System adds the comment to the post.
Alternative flows:	5a User tries to post a comment without inserting text. 1. System shows an error message. 3. System returns the textbox to its normal state.
Postconditions:	A comment is added below the post.
Special Requirements:	

Table 14. *Desktop use case 5, Comment on Feed*

Use case name:	**DUC6: Post**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. The user clicks on the “Publish” dropdown button. 2. User selects the option “Note”. 3. User selects a note type. 4. User types the text message. 5. User presses the “Post new note” button. 6. System stores the post. 7. System updates the feeds.
Alternative flows:	2a User selects to attach a media file too. 1. Upload a media file. 5a User selects to tag an employee. 1. Tag an employee. 2. Return to 4.
Postconditions:	A post is created both on feeds page and on the AF page of the employee.
Special Requirements:	

Table 15. *Desktop use case 6, Post*

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 dropdown list with all the usernames. 3. User selects one or more users. 4. System includes the users in the tag textbox. 5. Employee is tagged.
Alternative flows:	3a. User writes manually the tagged users. 1. User types a user's details (username, firstname or lastname). 2. Search. 4a User is not included in the database. 1. An alert informs of wrong name provided.
Postconditions:	An employee is added in a post as tagged.
Special Requirements:	

Table 16. *Desktop use case 7, Refer an employee*

Use case name:	**DUC8: Upload a media file**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User selects the option "Picture". 2. User selects a note type. 3. User types the text message. 4. User presses the "Attach File" option. 5. System request for file location. 6. User selects the file. 7. System checks the file extension 8. User presses the "Post new picture note" button. 9. System uploads the media file. 10. System stores the post. 11. System updates the feeds.
Alternative flows:	3a User selects to upload an invalid file. 1. System shows a message to inform the user. 2. System returns to 2.
Postconditions:	A post with image is created both on feeds page and on the AF page of

	the employee.
Special Requirements:	

Table 17. *Desktop use case 8, Upload a media file*

Use case name:	**DUC9: Summarize sensor data**
Participating Actors:	Desktop Engineer
Preconditions:	Log in, View Sensor profile page
Basic Flow:	1. User selects a date period. 2. System receives the historical data of the sensor. 3. System graphically presents the sensor values.
Alternative flows:	
Postconditions:	A graph of historical sensor values is added in the profile page of the sensor.
Special Requirements:	

Table 18. *Desktop use case 9, Summarize sensor data*

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 returns an empty list.
Postconditions:	The result area is updated with the results.
Special Requirements:	

Table 19. *Desktop use case 10, Search*

Use case name:	**DUC11: View activity feed**
Participating Actors:	Desktop Engineer
Preconditions:	Log in
Basic Flow:	1. User clicks on another user's name. 2. Systems redirects to his profile page. 3. System shows all activities (posts, comments, tags) in a time sequence.
Alternative flows:	
Postconditions:	
Special Requirements:	

Table 20. *Desktop use case 11, View activity feed*

Use case name:	**DUC12: Edit profile image**
Participating Actors:	Desktop Engineer
Preconditions:	Log in, DU views his profile page
Basic Flow:	1. User clicks the update avatar button. 2. Systems redirects to his profile page. 3. System shows all activities (posts, comments, tags) in a time sequence. 4. System request for file location. 5. User selects the file. 6. System checks the file extension. 7. System uploads the media file. 8. System updates the profile image.
Alternative flows:	5a User selects to upload an invalid file. 1. System shows a message to inform the user. 2. System returns to 2.
Postconditions:	The profile image is updated.
Special Requirements:	

Table 21. *Desktop use case 12, Edit profile image*

6.3 Use case model Mobile User

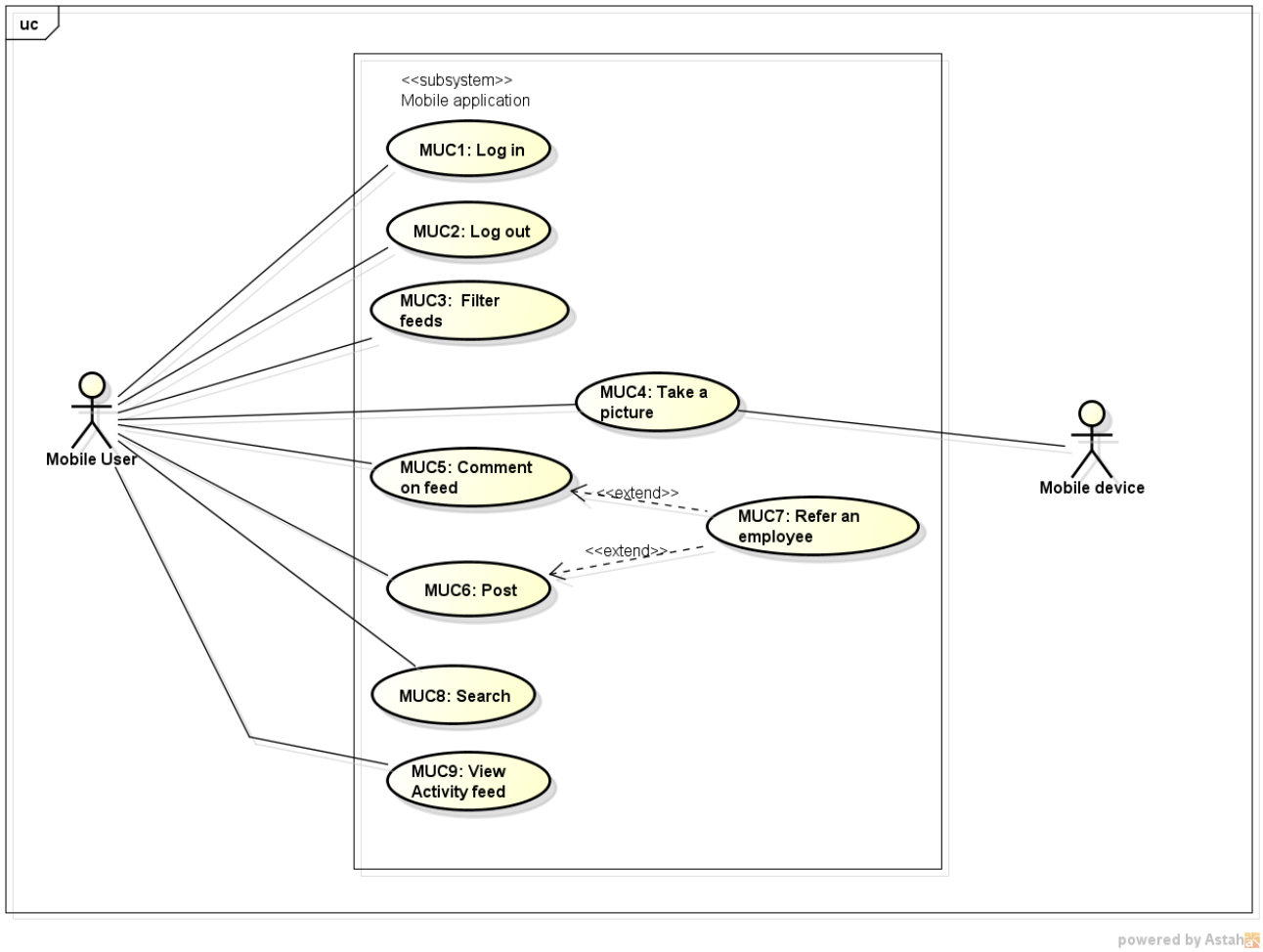

Figure 3: *Use case diagram of mobile user*

Use case name:	**MUC1: Log in**
Participating Actors:	MU
Preconditions:	Application is running
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.
Postconditions:	The mobile engineer is authenticated.
Special Requirements:	

Table 22. *Mobile user case 1, Login in*

Use case name:	**MUC2: Log out**
Participating Actors:	MU
Preconditions:	Log in
Basic Flow:	1. User presses the more options button. 2. User presses the "Log Out" action. 3. The system redirects the user to the login page.
Alternative flows:	
Postconditions:	The MU has signed out and the login page is shown.
Special Requirements:	

Table 23. *Mobile user case 2, Log out*

Use case name:	**MUC3: Filter feeds**
Participating Actors:	MU
Preconditions:	Log in
Basic Flow:	1. User taps the Filters tab. 2. User chooses to filter the feeds by type (Huma, Sensor). 4. User views the feed page. 5. System filters the feeds and show the relevant ones.
Alternative flows:	2a. The user choose a saved filter 1. User selects the name of a saved filter. 2. Return to 4. 2b. The user choose feeds from last shift. 1. User taps the button “Previous Shift”. 2. Return to 4.
Postconditions:	The system shows to the user the feeds filtered.
Special Requirements:	

Table 24. *Mobile user case 3, Filter feeds*

Use case name:	**MUC4: Take a picture**
Participating Actors:	MU, mobile device
Preconditions:	Log in
Basic Flow:	1. The MU selects the camera icon. 2. System initiates the camera. 3. User takes a picture. 4. User accepts the picture. 5. System stores the picture. 6. Image is shown on post screen.
Alternative flows:	2a. Some other process is using the camera. 1. The system should inform the user with a message. 4a. User does not accept the picture. 1. Return at 2.
Postconditions:	A picture is taken and added to the feed.
Special Requirements:	The system should be able to use the camera.

Table 25. *Mobile user case 4, Take a picture*

Use case name:	**MUC5: Comment on feed**
Participating Actors:	MU
Preconditions:	Log in
Basic Flow:	1. The MU chooses a feed. 2. The system loads feed information. 3. The MU taps the comment tab. 4. The MU selects to add a comment in a textfield. 5. The MU types the comment. 6. System validates the comment. 7. The system stores the comment.
Alternative flows:	
Postconditions:	A comment by the MU should be shown on comments screen of the respective feed. The number of comments of the respective feed in the feeds screen should be updated.
Special Requirements:	

Table 26. *Mobile user case 5, Comment on feed*

Use case name:	**MUC6: 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 the post content. 4. The MU publishes the post. 5. The system saves the post. 6. System updates the feeds.
Alternative flows:	3a. User includes also a picture. 1.Take a picture. 2. Refer an employee. 3. Return to 4. 3a.2. User does not refer an employee. 3. Return to 4 basic flow. 3b. User refers to other employees. 1. Refer an employee. 2. Return to 4.
Postconditions:	The post is shown on the feed screen.
Special Requirements:	

Table 27. *Mobile user case 6, Post*

Use case name:	**MUC7: Refer an employee**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. User taps the tag option button. 2. System shows the tag screen. 3. Systems loads all the users of the system. 4. MU selects users. 5. System add users as tagged.
Alternative flows:	4a. User types names of specific users 1.MU selects the search textbox 2. Search. 3. Return at 4.
Postconditions:	The tagged users should be shown on the tags tab of the chosen feed.
Special Requirements:	It is possible to refer only workers in the same work environment

Table 28. *Mobile user case 7, Refer an employee*

Use case name:	**MUC9: Search**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. The MU type the details (username, last name of first name for humans) of another user or a sensor. 2. The system searches the typed user or sensor. 3. System updates a list with the returned result.
Alternative flows:	
Postconditions:	All the results for the search are shown to the user
Special Requirements:	

Table 29. *Mobile user case 8, Search*

Use case name:	**MUC10: View activity feed**
Participating Actors:	MU
Preconditions:	The user should be authenticated.
Basic Flow:	1. The MU has selected a user. 2. System loads his personal details. 3. MU navigates to the activity tab. 2. The system retrieves the AF of the selected user.
Alternative flows:	1a. MU has searched for a user. 1. Search. 2. Return to 1.
Postconditions:	All the actions (comments, tags, posts) completed by the MU are shown.
Special Requirements:	

Table 30. *Mobile user case 9, View activity feed*

6.4 Use Case Model Control System

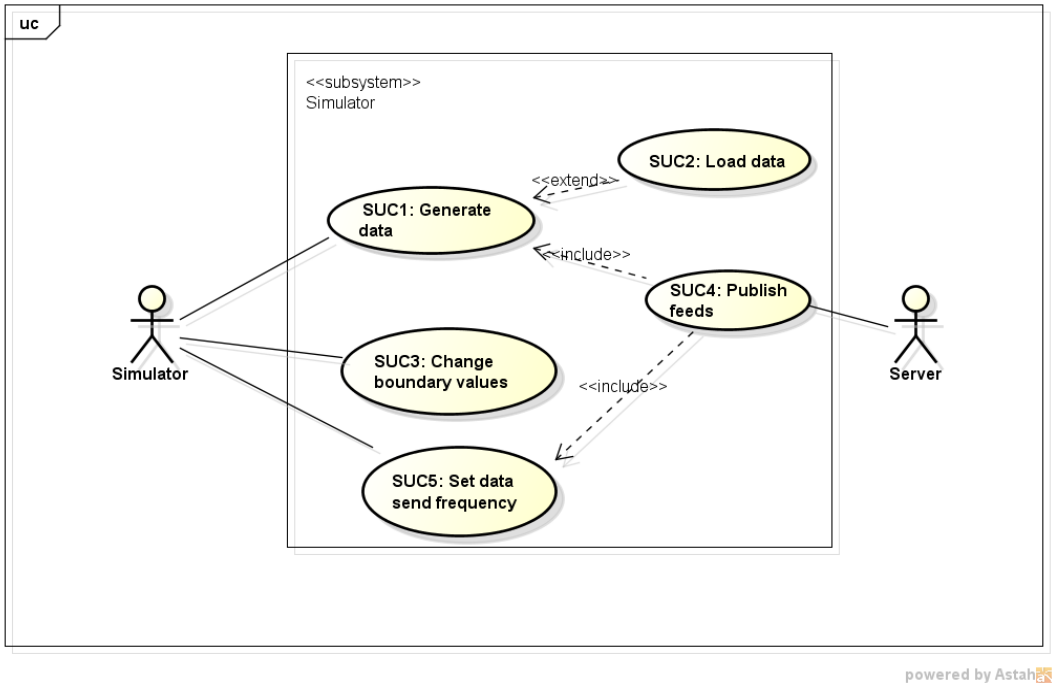

***Figure 4**: Use case diagram of simulator*

Use case name:	**SUC1: Generate data**
Participating Actors:	Simulator
Preconditions:	Simulator user has defined the value boundaries of the sensor.
Basic flow:	1. Simulator user clicks “Generate Data” 2. The simulator randomly pick 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:	1a The user has not specified a high or low boundary. 1. A message box informs the user. 2. User fills in the missing boundary values. 3. Return to step 1.
Postconditions:	
Special Requirements:	

Table 31. *Sensor user case 1, Generate data*

Use case name:	**SUC2: Load Data**
Participating Actors:	Simulator
Preconditions:	Simulator user has defined the value boundaries of the sensor.
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 fills in the missing boundary values. 3. Return to step 1.

Postconditions:	
Special Requirements:	

Table 32. *Sensor user case 2, Load Data*

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:	
Postconditions:	The text box values are updated.
Special Requirements:	

Table 33. *Sensor user case 3, Change Boundary Values*

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 set as the interval.
Alternative flows:	
Postconditions:	The generated values are stored in the database
Special Requirements:	

Table 34. *Sensor user case 4, Publish feeds*

Use case name:	**SUC4: Set data send frequency**
Participating Actors:	Simulator
Preconditions:	
Basic Flow:	1. The simulator user fills in the time interval (in milliseconds) text box. 2. The simulator validates the inserted value
Alternative flows:	
Postconditions:	The text box value is updated.
Special Requirements:	

Table 35. *Sensor user case 5, Set data send frequency*

7 List of Tables

8 List of Figures