Assignment 1: Jodel Alert

Software Engineering - Design
- 2DV603

Name: Patrik Hermansson
Email: ph222md@student.lnu.se
Name: Michael Wagnberg
Email: mw222uu@student.lnu.se
Name: Benjamin Svärd
Email: bs222et@student.lnu.se
Name: Christofer Nguyen
Email: cn222hn@student.lnu.se
Name: Jonathan Walkden
Email: jw222qi@student.lnu.se

Contents

1 Design Document

1.1 Purpose

This document will describe the entire design and decisions as well as rationales regarding Jodel application API, work-arounds, client-server and core functionality and vital parts of Jodel Alert.

1.2 General priorities

One of our highest priorities is simplicity. The core problem we are trying to solve, the project idéa, is in itself simple and small, which makes it reasonable to design everything with simplicity in our minds. The alert that raises awareness of a post that Linnéstudenterna is interested in is the important result. There are several ways to accomplish that, you can have an advanced GUI for the customer, or different ways to change the properties of keywords and email. We have designed everything to just solve the core problem, the alert. The Jodel Alert will run in the background on a server, and send alerts (emails) when keywords found. We have also focused on reuseability as a priority in our design, this is merely because of our projects uniqueness on the Jodel application. Everything will be general and not suited just for our customer, this makes other companies potential customers.

The customers of Linnéstudenterna are the users of Jodel. Our customer/stakeholder is Linnéstudenterna. Our customer has ordered an application to handle relationship with the students asking questions on Jodel.

1.3 Design issues

One of the first things that had to be resolved was of course the Jodel API. We did not know if that API was open or not accessible for the general public. We planned accordingly to cover both situations, so we were ready to design with the API or without. After several tries of contacting the developers of Jodel we understood that no answer would arrive.

The environment used will be a PC running Windows which the application will be run upon on. This computer must be up and running and online twenty-four seven in order for the application to work properly. A mail-server will run in the background of the application which will send email to the chosen recipients in case of a registered keyword gets posted.
Application will be written and developed in Java. This makes it versatile and can work on all platforms needed.
The application Jodel is already in place and we are going to base our work around the API of that program and create a new application. Nothing else is in place, there is no half-done project that covers this problem, so everything will be created from scratch. If the API is not accessible, we are going to use mitmproxy to listen to the traffic and save data which we can process in order to produce the required email.

1.4 Tasks and procedures currently performed

Today Linnéstudenterna manually checks Jodel from time to time in order to capture some of the questions and answering them, but this is too time consuming and there is no way

they can monitor the feed all the time. They currently do not use any technology in aid other than the app itself.

1.5 Competing software

Jodel is kind of alone on the market with its nische, and tapping on to their feed in order to capture keywords is not something that exists right now. Other software that taps into feeds and derives statistics and data are Google Alert and Meltwater.

1.6 Product placement

This application can be used by Linnéstudenterna in order to connect with the students when they have questions. Other aspects that this application can cover can also be in the commercial market.
Companies can tap in and scan the feed and see what is said about that company, and because Jodel is an anonymous application, people tend to say the truth, or completely the opposite. This can build a knowledge about the customers using a specific brand or a chain and companies can act upon that information in marketing campaigns or changing the way they act against customers.

1.7 Stakeholder's vision for the future

Our stakeholder's desiderata is to have a web based interface with functions like managing keywords and emails, and this application can be a start of something bigger in the future with more added functionality. Our customer want to be able to answer to a post directly from the web interface.

2 Scenarios

These scenarios are based on the meeting we had with our customer.

2.1 Scenario 1

A Jodel user will make a post on the Jodel app. Once this post is submitted the Jodel Alert app will recieve an access token to be able to scan the post for any specified keywords from a matching database. If a keyword is found, the Jodel Alert app will send an email to the administrator which contains the keyword used and the content of the post in which it was used. If any errors occur they will be logged and an email will be sent to the list of emails in the database.

2.2 Scenario 2

If the administrator of the Jodel Alert app would like to add/remove keywords, they can access the database manually and perform the changes there. The same goes for adding or removing the list of email addresses that will recieve notification when a matching keyword is found.

3 Functional requirements

1. When a keyword is used in a Jodel post, Linnestudenterna will receive a mail
2. The application will request Jodel posts every 15 minutes.
3. Keywords must be able to be removed
4. Keywords must be able to be added
5. Keywords must be able to be changed
6. Added keywords saved between sessions (i.e saved on local host if you restart the app)
7. Recipient email must be able to be removed
8. Recipient email must be able to be added
9. Recipient email must be able to be changed

4 Non-functional requirements

1. When a keyword has been found, email should be sent within 2 seconds

5 Use Cases

The following use cases are derived from the scenario.

5.1 Use Case 1

Use Case 1 - Application scans and sends mail - Functional requirement 1
Primary Actor: Jodel Alert Application

Pre condition

1. Jodel Alert has been started
2. Jodel Alert has received permission for get request

Post condition

An email has been sent to the receiver

Post condition - Alternate scenario

An email has been sent to the receiver

Main scenario

1. Jodel Jodel Alert received recent list of post
2. Jodel Alert will process the post
3. Jodel Alert will perform a scan on each post
4. Jodel Alert will find matching keyword in post from database
5. Mail server on Jodel Alert will send away an email to the specified email recipient

Alternate scenario

1. Jodel Alert received recent list of post
2. Jodel Alert will process the post
3. Jodel Alert will perform a scan on each post
4. Keywords does not exist in database
5. No actions will be taken

5.2 Use Case 2

Use Case 2 - Application can not send email - Functional requirement 1
Primary Actor: Jodel Alert Application

Pre condition

1. Jodel Alert has been started

2. Jodel Alert has received permission for get request

Post condition

An email has not been sent to the recipient

Main scenario

1. Jodel Jodel Alert received recent list of post
2. Jodel Alert will process the post
3. Jodel Alert will perform a scan on each post
4. Jodel Alert will find matching keyword in post from database
5. Mail server on Jodel Alert will try to send away an email to the specified email recipient
6. Mail server can not send mail due to error
7. Session timed out
8. Generic respond will be generated to be sent away, logged to error text

5.3 Use Case 3

Use Case 3 - Restart application - Functional requirement 2
Primary Actor: Jodel Alert Application

Pre condition

Jodel Alert is running

Post condition

1. Jodel Alert is running again
2. Email and Keyword list will be loaded back to Jodel Alert

Main scenario

1. Restart Jodel Alert

5.4 Use Case 4

Use Case 4 - Remove keyword - Functional requirement 3
Primary Actor: Administrator

Pre condition
Database list is available and has been opened

Post condition
Newly removed keyword will not exist on the list of keywords

Main scenario

1. Manually remove existing keyword from the database text file
2. Restart Jodel Alert

5.5 Use Case 5

Use Case 5 - Add keyword - Functional requirement 4
Primary Actor: Administrator

Secondary Actor: Jodel Alert application

Pre condition
Database list is available and has been opened

Post condition
Newly added keyword will be added to the list of keywords

Main scenario

1. Manually add new keyword to database text file
2. Restart Jodel Alert

5.6 Use Case 6

Use Case 6 - Change keyword - Functional requirement 5
Primary Actor: Administrator

Secondary Actor: Jodel Alert application

Pre condition
Database list is available and has been opened

Post condition
Newly changed keyword will be changed in database text file

Main scenario

1. Manually change existing keyword to database text file
2. Restart Jodel Alert

5.7 Use Case 7

Use Case 7 - Scan Jodel Traffic - Non Functional requirement 1
Primary Actor: Jodel Alert application

Pre condition
Jodel Alert has been started

Post condition
Jodel Alert has scanned each post

Post condition - Alternative flow
Jodel Alert has not scanned each post

Main scenario

1. Jodel Alert request recent list of post
2. Jodel Alert will process the post
3. Jodel Alert will perform a scan on each post

Alternate scenario

1. Jodel Alert request recent list of post
2. Jodel Alert does not get permisssion to recent list of post
3. Jodel Alert will not process the post
4. Jodel Alert request list of post until permission granted

5.8 Use Case 8

Use Case 8 - Remove email recipient - Functional requirement 7
Primary Actor: Administrator

Secondary Actor: Jodel Alert application

Pre condition
Database list is available and has been opened

Post condition
Newly removed email recipient will be removed from the database text file

Main scenario

1. Manually remove email recipient from database text file
2. Restart Jodel Alert

5.9 Use Case 9

Use Case 9 - Add email recipient - Functional requirement 8
Primary Actor: Administrator

Secondary Actor: Jodel Alert application

Pre condition
Database list is available and has been opened

Post condition
Newly added email recipient will be added to the database text file

Main scenario

1. Manually add email recipient to database text file
2. Restart Jodel Alert

5.10 Use Case 10

Use Case 10 - Change email recipient - Functional requirement 9
Primary Actor: Administrator

Secondary Actor: Jodel Alert application

Pre condition
Database list is available and has been opened

Post condition
Newly changed email recipient will be changed in the database text file

Main scenario

1. Manually change email recipient to database text file
2. Restart Jodel Alert

6 Admin Scenario

7 Participating Objects

Domain model containing the participating objects.

Figure 7.1: Jodel logo