Assignment 1: Jodel Alert Design Document

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.

The Jodel Alert Application will scan Jodel post feed, derive keywords desiderata and notify an email recipient(requirement ID 1). It will do this every 15 minutes (requirement ID 2). Possible failure conditions can be that Jodel Alert will not get an authenticated token, the server which Jodel Alert runs upon is down, or wrong format of email recipient. This will be handled by error messages to the user.

The Jodel Alert Application user will be able to add and change email recipient as well as keywords(requirement ID 4,5,6,7)

The Jodel Alert Application admin will be able to remove email recipient and keywords(requirement 3,8) as well as the ordinary user actions.

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.

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. Instead we moved towards a program called mitmproxy, that deals with listening to the traffic from Jodel by imitating an ordinary client cellphone. We can then extract that data and use it in our favour.

Choosing database was another possible constraint or issue. Regarding to the technical aspect, maybe no one in the group have competence to use a powerful database (like MySQL) in collaboration with the never used program mitmproxy. Another aspect is budget and time for implementing MySQL as a database.

We have choosen a design that fits a "database", and we can easily choose between a commercial database or just a plane text file, depending on difficulties and other constraints that may show up.

Application will be written and developed in Java. This makes it versatile and can work on all platforms needed. This and better knowledge of the language is the rationale of choosing Java.

1.4 Outline of the design
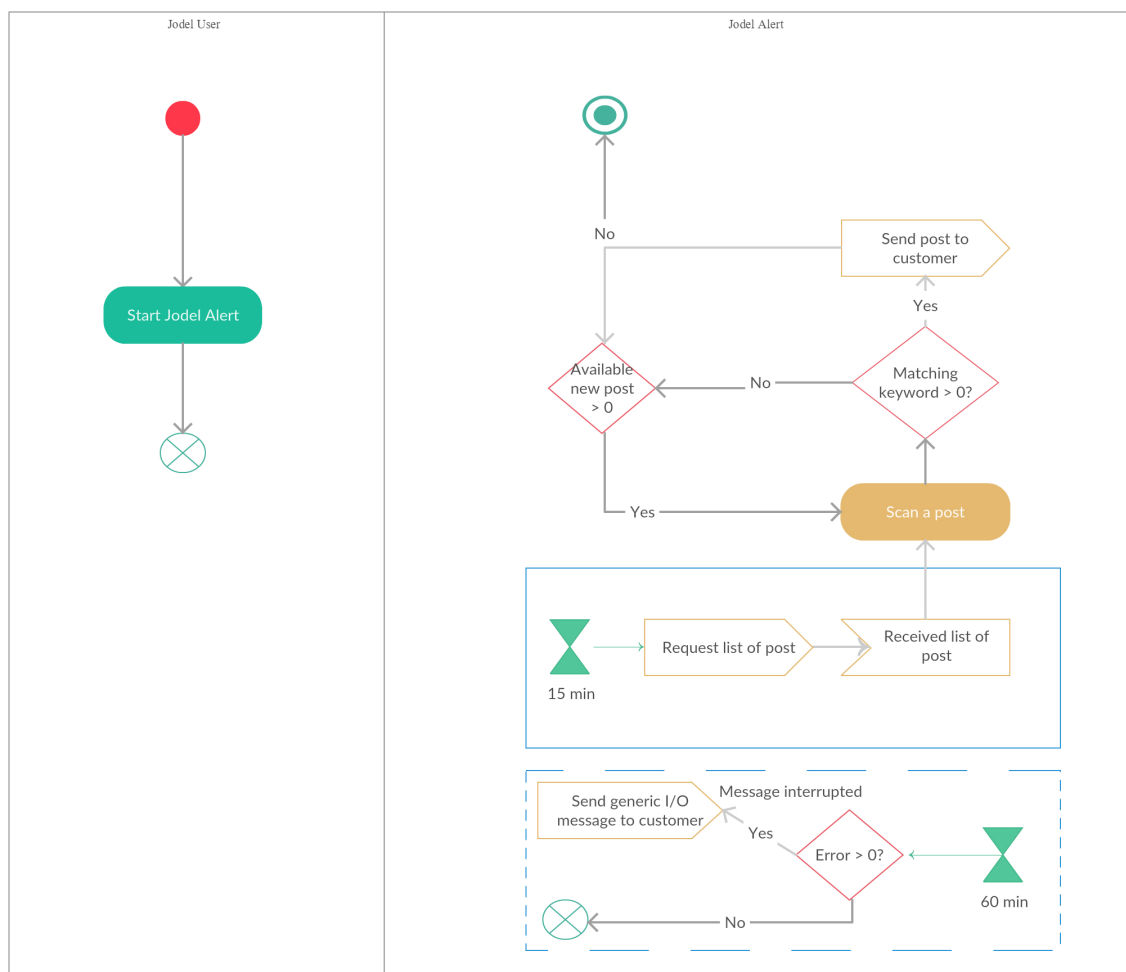
All key components for this application is listed here.

1.5 Handle Jodel Post

Figure 1.1: Jodel Alert Activity Diagram Handling Jodel Post
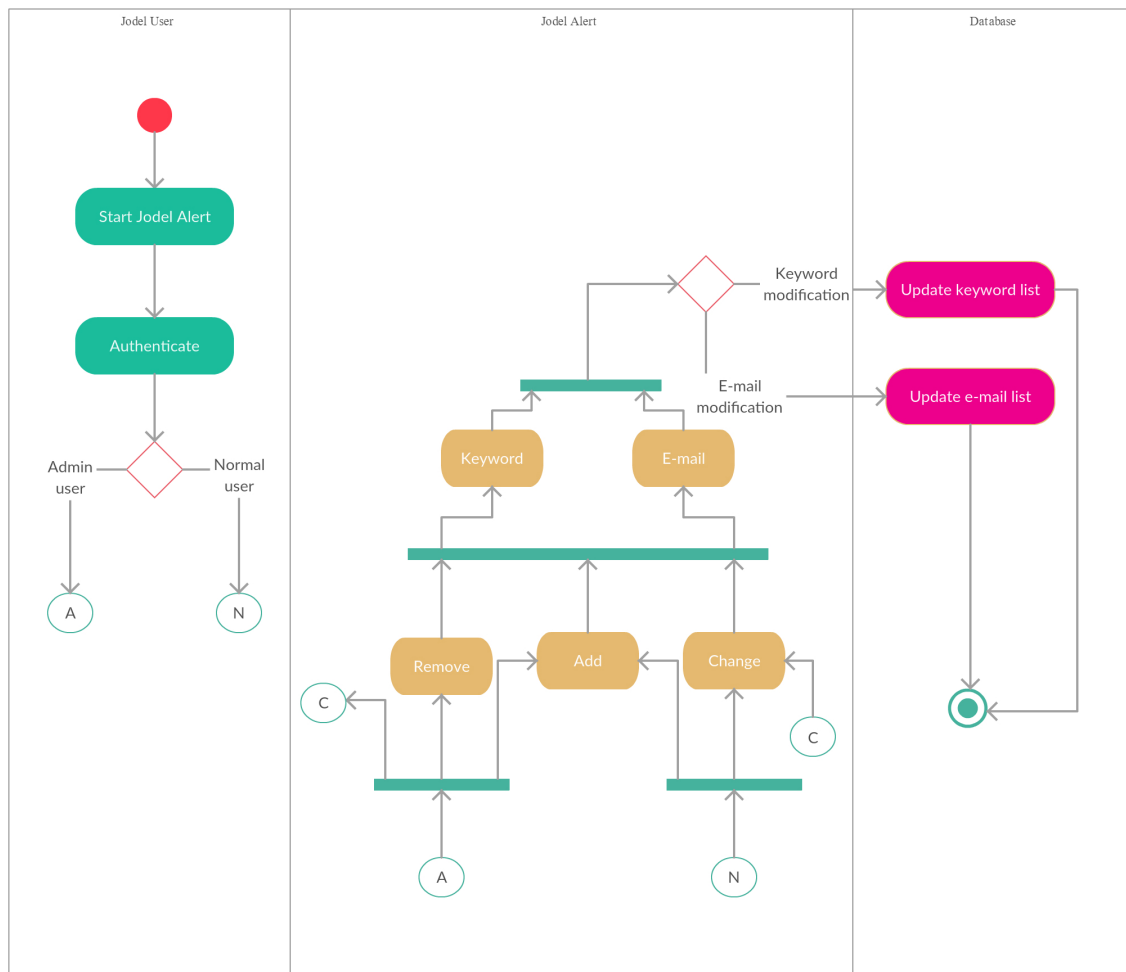
1.6 Modifying Database

Figure 1.2: Jodel Alert Activity Diagram Modifying Database
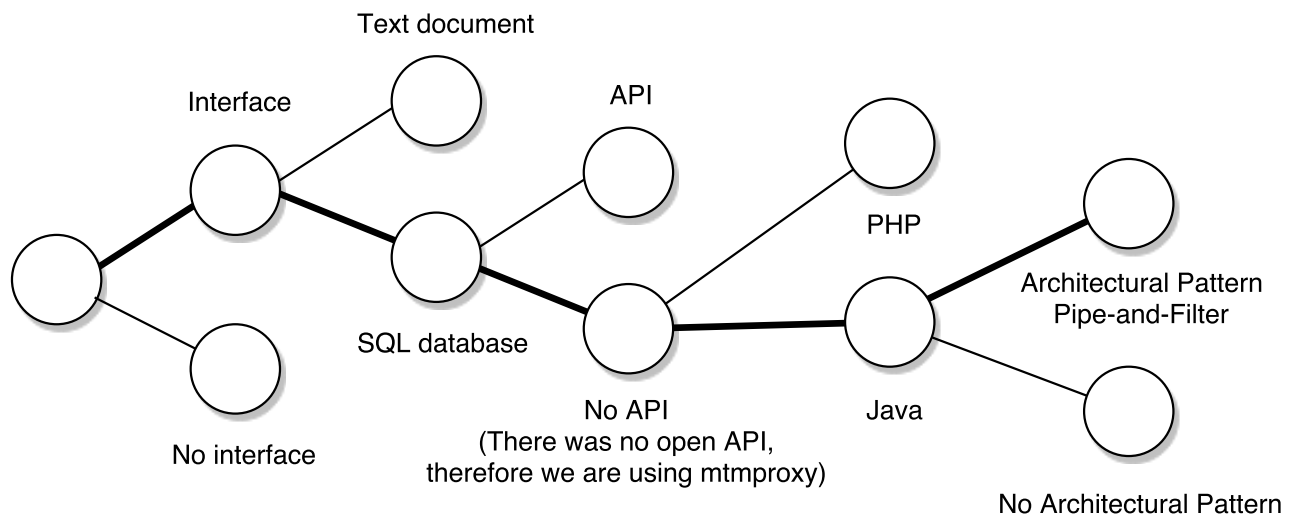
1.7 Constraints

1.8 Design options

Figure 1.3: Jodel Alert Design Space

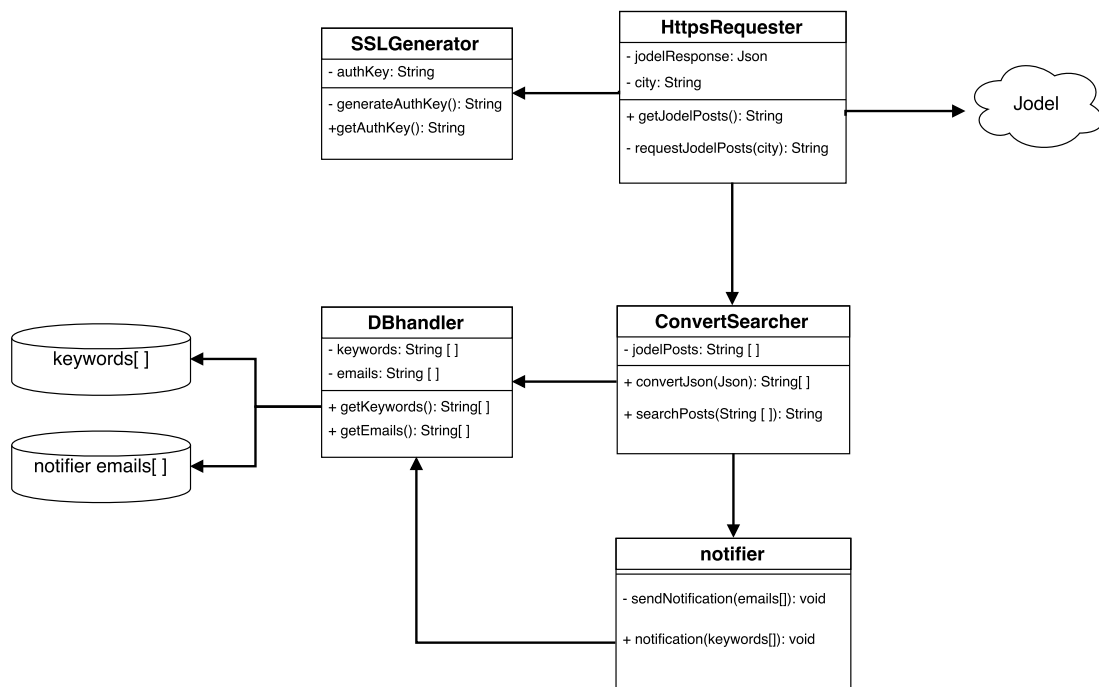# 2 Risk mitigation

3 Class diagram

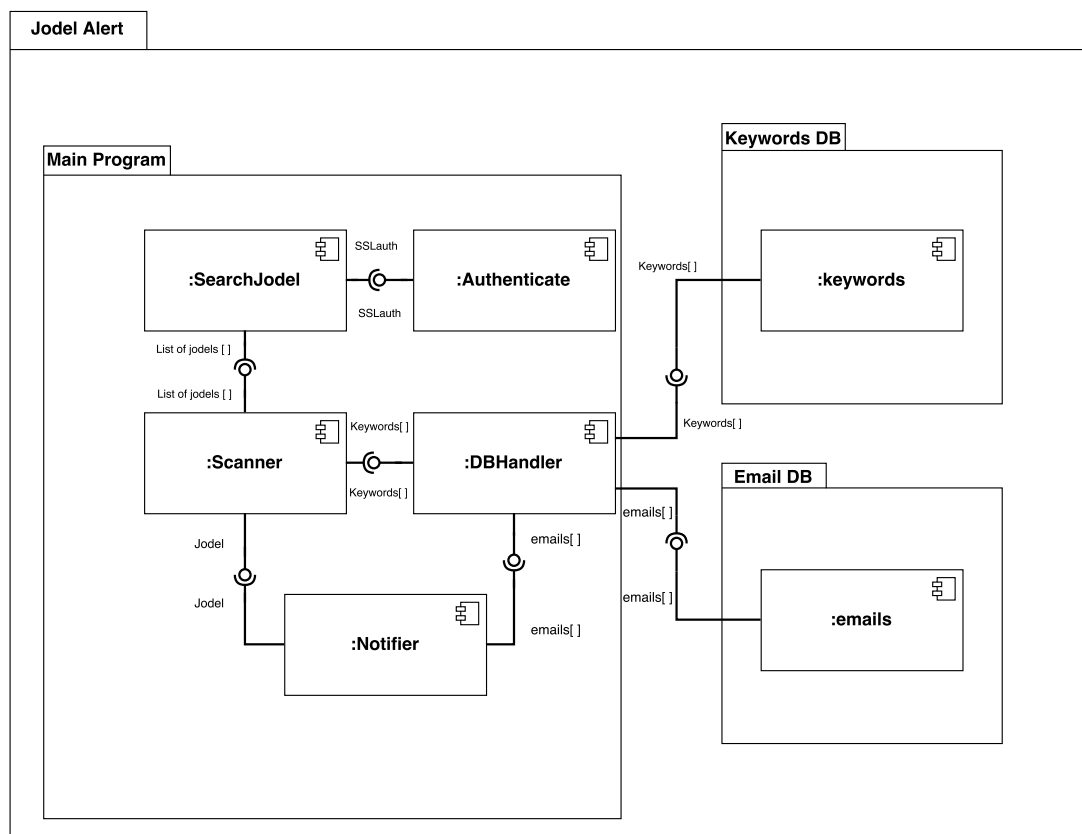Figure 3.4: Jodel Alert Class Diagram

4 Component Diagram

Figure 4.5: Jodel Alert Component Diagram

Figure 4.6: Jodel logo