# System Design Document for the KudoMessage Project (SDD)

**Contents**

**1 Introduction 1**

* 1. Design goals.. 2
  2. Definitions, acronyms and abbreviations 2

**2 System design 3**

2.1 Overview 3

2.2.1 The model functionality 3

2.2.2 Rules . 3

2.2.3 Unique identifiers, global loop-ups 3

2.2.4 Spaces 3

2.2.5 Event handling 4

2.2.6 Internal representation of text 4

2.2 Software decomposition 4

2.2.1 General 3

2.2.2 Decomposition into subsystems. 3

2.2.3 Layering 3

2.2.4 Depoendecy analysis 3

2.3 Concurrency issues 4

2.4 Persistent data management 4

2.5 Access control and security 4

2.6 Boundary conditions 4

**3 References 5**

**Version:** N.0.0.B

**Date:** Today

**Author:** Group 21

This version overrides all previous versions.

**1 Introduction**

**1.1 Design goals**

derp

**1.2 Definitions, acronyms and abbreviations**

* GUI, graphical user interface
* Java EE, platform independent programming language (EE stands for Enterprise Edition)
* Glassfish, a web server to host java files
* JSF, (Java Server Faces) The HTML pages with connection to java classes.
* Hustler, the push server
* OAuth 2.0, gives us the possibility to access server resources on behalf of the resource owner
* GCM (Google Cloud Messaging), a service that helps developers to send data from servers to their Android applications on Android devices.

**2 System design**

**2.1 Overview**

The android application will use a MVC model, even though it won’t have a comprehensive view. The frontend in the web application will be built using HTML and CSS, with java as backend.

**2.1.1 The model functionality**

Although we believe that usability is important for the application, this isn’t where we put our main focus. The functionality of the application is relatively simple, seen from a user perspective, and we therefore believe that we won’t have to dedicate any greater amount of time for this.

We will, nevertheless, put time aside to make a user guide for the program. It will include the whole process, from installing the background server to your smart phone, to using the web application.

**2.2.2 Reliability**

NA

**2.2.3 Performance**

It is hard to guarantee a perfect performance for this application due to the dependency on how well your own telecommunication operator works. The application is the intermediary between the user and the operator and are therefore not to blame for poor network connection. We can’t guarantee that there wont be any delay due to the communication with GCM.

The user should, however, get a notification that clarifies if the SMS was delivered successfully or not.

**2.2.4 Supportability**

The web application will suit all platforms, as it is accessible through a regular web browser. There are plans to develop native applications that are tailored to suit a specific operating system later on, as mentioned earlier. This shouldn’t be hard to implement when it Iphone server app.

**2.2.5 Implementation**

The reason we’re developing the app as a web application is so that it can be used from all devices without having to be installed first. It will be quick and easy to use.

**2.2.6 Packaging and installation**

**2.2.7 Legal**

Följer googles användarvillkor

**2.3 Application models**

**2.3.1 Use case model**

**2.3.2 Use case priority**

**2.3.3 Analysis model**

**2.3.4 User interface**

**2.2 References**