

# Project Documentation DAT076 Web Applications for Opinion.rocks

## Group Information

Group 6

Simon Bengtsson, 9203085551, simbeng@student.chalmers.se

Caroline Kabat, 891016-4881, kabat@student.chalmers.se

Kim Kling, 930605-0973, kkim@student.chalmers.se

Simon Planhage, 900322-0036, planhage@student.chalmers.se

## Introduction

Opinion.rocks is a clean and fast web application where you can easily share your opinions and engage in discussions with other users. You simply login with your twitter account and start sharing your opinions!

## Use cases

As a user, you can:

- Login with your twitter account.
- Create new opinions (as a new post) containing text, hashtags, links, pictures and gps coordinates.
- [GUI Only] Browse the latest posted opinions (posts) on the never ending scrolling front page.
- [GUI Only] Choose to filter your front page posts by either global, local or your followers posts.
- [GUI Only] Rate other posted opinions with a Agree/Disagree system.
- Comment on opinions with a comment system.
- Follow other users.
- View users profiles, which have a short description, their names, and a picture (imported from twitter).
- Edit your own user profile.
- Search for any matching text inside opinions.
- Logout

As an anonymous user (not logged in via twitter account), you can view the latest posts and scroll through the front page.

## Components

### Backend

#### **Maven**

Both the frontend and backend uses maven for packaging and deployment of the application.

#### **JavaDB (Derby)**

For storing users, posts, users followers and some other data in the database.

## **Java Persistence**

### **JDBC**

For communicating with the database.

### **Twitter API**

For handling user login/logout, and providing the users profile page with some information from his/her twitter account.

### **Rest**

We use REST for communicating between the frontend and backend.

### **Gson**

For creating and reading JSON-objects.

## **Frontend**

### **AngularJS**

### **JavaScript**

### **HTML5**

### **CSS**

Bower is used as a package manager for front-end packages

## **Packages**

### **hage.auth**

Handles authorization for users who are logging in.

### **hage.controllers**

Handles communication between the database, authorization classes and frontend.

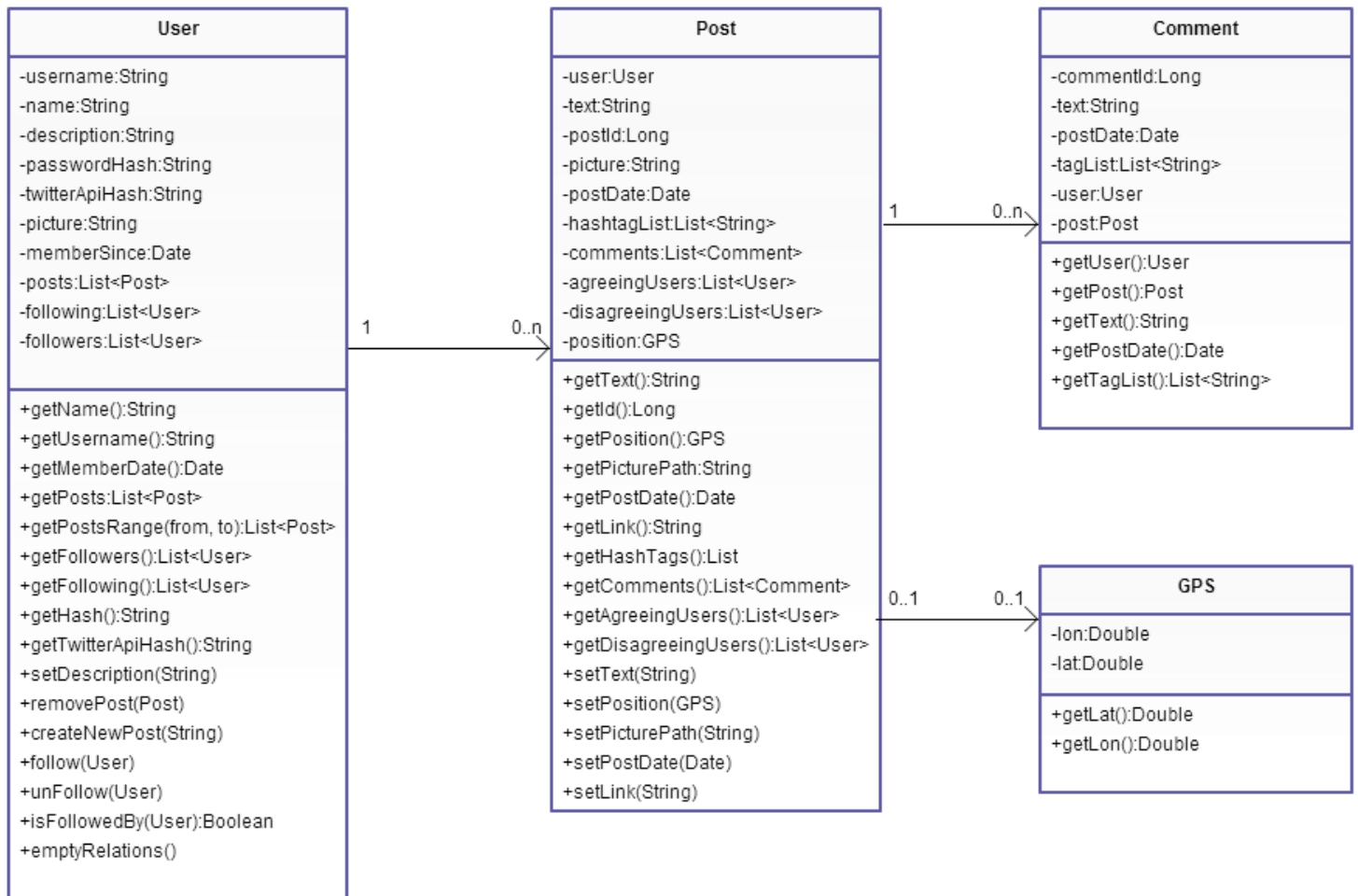
### **hage.model**

The backing model for the application. See UML diagram below.

### **hage.persistence**

Contains classes for database usage.

## **Class diagram of hage.model**



Communication between packages

