
DOCUMENTATION OF SOCIAL NETWORK PROJECT

June 26, 2017

He Yan

Contents

1	Introduction	1
1.1	Main Features	1
1.2	Components	1
2	Environment	1
3	Data Structure	2
3.1	Entity-Relationship Diagram	2
3.2	MySQL Table	2
4	Division of Labor	5
5	Kernel Codes	5
5.1	Google reCaptcha	5
5.2	Two-step friends	7
6	Website Preview	7
7	References	7

1 Introduction

This project aims to build a social network with JSP and MySQL.

1.1 Main Features

- *Compulsory:*
 - Sign up & in
 - Search for contacts & Post status and reply
 - 30 secs refreshment
- *Optional:*
 - Email address regex check
 - Ajax
 - Add Google's reCaptcha¹ validation
 - Two-step friends

1.2 Components

- Apache, Tomcat, Apache-Tomcat-Connector
- MySQL, MySQL Connector/J (JDBC)

Visit our project site at [Database Course Project](#).

2 Environment

This project is hosted on Amazon Linux AMI server provided by AWS. To build the environment for running our website, we took steps as below.

1. install OpenJDK-1.8.0
2. install and configure Apache (httpd) & Tomcat

¹Completely Automated Public Turing test to tell Computers and Humans Apart

3. link Apache and Tomcat with Apache Tomcat Connector ²

4. install MySQL ³ and prepare MySQL connector/J in WEB-INF/lib

Note: Our project has been hosted at GitHub. Visit our project at <https://github.com/PKU-2017-Database/Social-Network>.

3 Data Structure

3.1 Entity-Relationship Diagram

Here is an English version of ER Diagram redrawn by L^AT_EX.

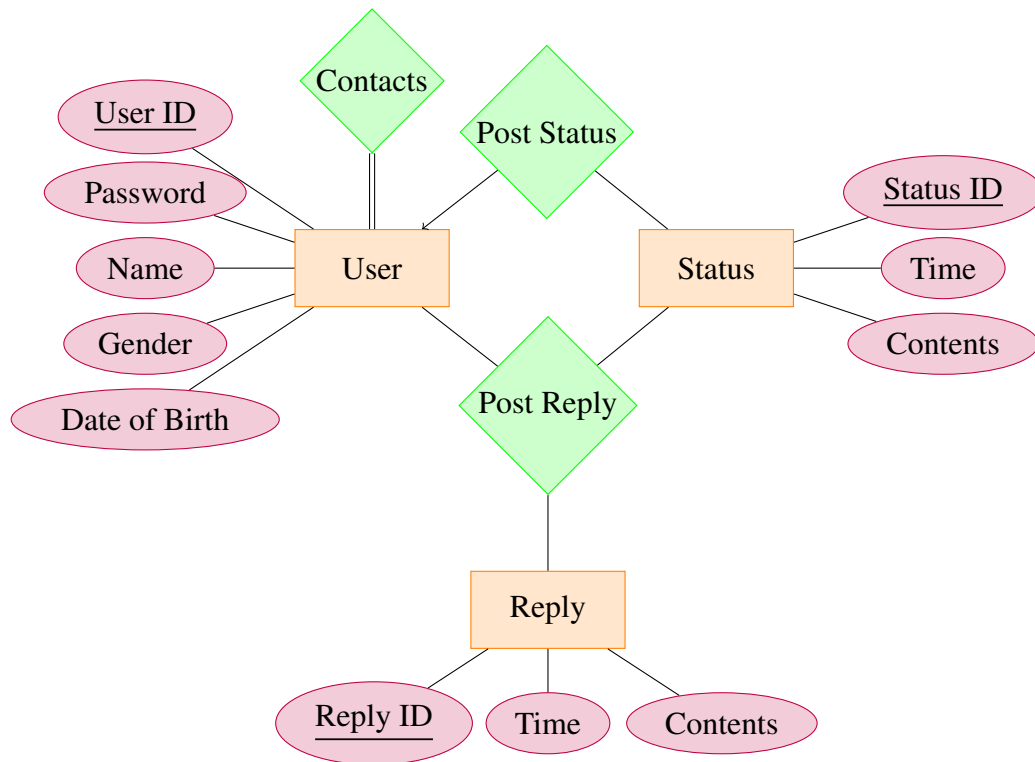


Figure 1: Entity-Relationship Diagram

3.2 MySQL Table

According to the above ER Diagram, we've designed MySQL tables as below.

²This makes it possible to run static web pages on Apache and dynamic ones on Tomcat.

³MySQL is case insensitive for Windows and MacOS, but that's not true for Linux.

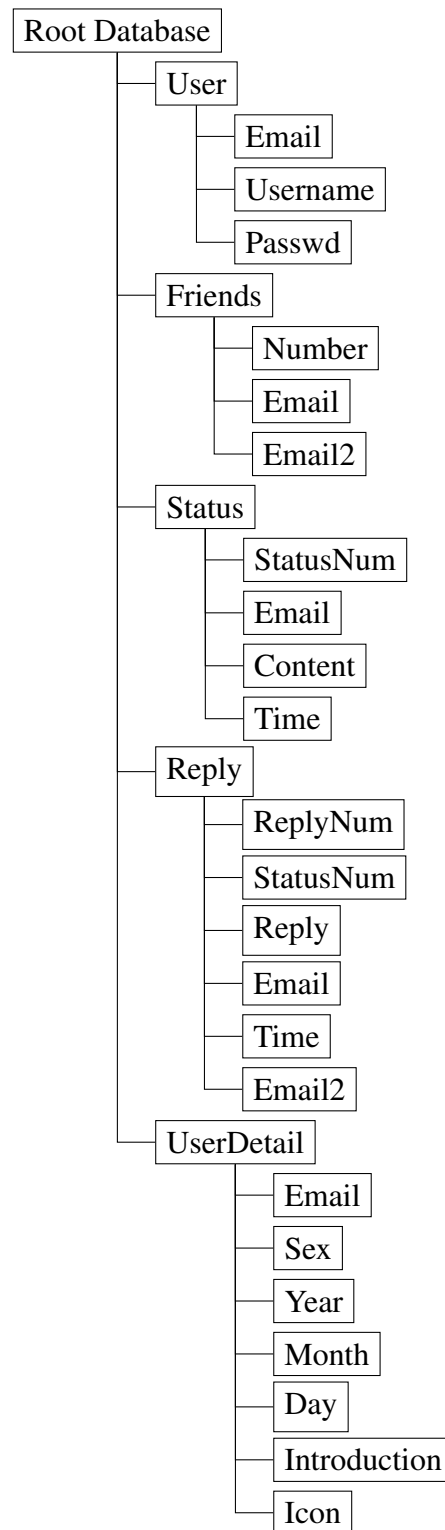


Figure 2: MySQL Table Structure

Details about tables and attributes:

- User - deal with signing up & in
 - Email: primary key to identify users in registration and log-in
 - Username: nickname, which can be edited after registration
 - Passwd: password to validate a user
- Friends - record friend relationships
 - Number: auto-increment primary key for identification
 - Email: follower's email
 - Email2: followee's email
- Status - store posted statuses
 - StatusNum: auto-increment primary key for identification
 - Email: poster's email
 - Content: posted contents
 - Time: posting time
- Reply - store posted replies to status
 - ReplyNum: auto-increment primary key for identification
 - StatusNum: replied status number
 - Reply: reply contents
 - Email: replier's email
 - Time: replying time
 - Email2: repliee's email
- UserDetail - store user details
 - Email: primary & foreign key pointing to User.Email
 - Sex: user's sex
 - Year: user's year of birth
 - Month: user's month of birth

-
- Day: user's day of birth
 - Introduction: simple introduction to the user
 - Icon: user's avatar

4 Division of Labor

Our group members:

Name	Student ID	Mobile	Email
He Yan	1400015464	15910670278	heyan@pku.edu.cn
Sun Meng	1500012867	15010189739	1400017665@pku.edu.cn
Wu Chuchuan	1500062802	18811788416	wuchuchuan@pku.edu.cn

Table 1: Group Members

Division of labor:

- He Yan: write documentation & add reCaptcha to website
- Sun Meng: design of website appearance
- Wu Chuchuan: logical framework & database implementation

5 Kernel Codes

5.1 Google reCaptcha

```

1  /* Client Side */
2  <div class="g-recaptcha" data-sitekey="6
   LdDNiMUA AAAAHDPfsdqgPKAPFEy5Xi3EoGwJIXi" ></div>
3
4  /* Server Side */
5  String gRecaptchaResponse = request.getParameter("g-recaptcha-response");
6  String url = "https://www.google.com/recaptcha/api/siteverify";
7  String secret = "6LdDNiMUA AAAAACVxX9eQOV4ITsc9YApSRpb80Lle";
8  boolean check = true;
9  if (!(gRecaptchaResponse == null || "".equals(gRecaptchaResponse))) {
10     try {

```

```

11     URL obj = new URL(url);
12     HttpURLConnection con =
13         (HttpURLConnection) obj.openConnection();
14     con.setRequestMethod("POST");
15     con.setDoOutput(true);
16
17     String postParams
18         = "secret=" + secret + "&response=" + gRecaptchaResponse;
19
20     DataOutputStream wr =
21         new DataOutputStream(con.getOutputStream());
22     wr.writeBytes(postParams);
23     wr.flush();
24     wr.close();
25
26     BufferedReader in =
27         new BufferedReader(
28             new InputStreamReader(
29                 con.getInputStream()));
30     String inputLine;
31     StringBuffer rsps = new StringBuffer();
32     while ((inputLine = in.readLine()) != null) {
33         rsps.append(inputLine);
34     }
35     in.close();
36
37     JsonReader jsonReader =
38         Json.createReader(
39             new StringReader(rsps.toString()));
40     JsonObject jsonObject = jsonReader.readObject();
41     jsonReader.close();
42     check=jsonObject.getBoolean("success");
43 } catch (Exception e) {
44     e.printStackTrace();
45 }
46 }
47 ...
48 if (check) {
49     ...
50 }

```

5.2 Two-step friends

6 Website Preview

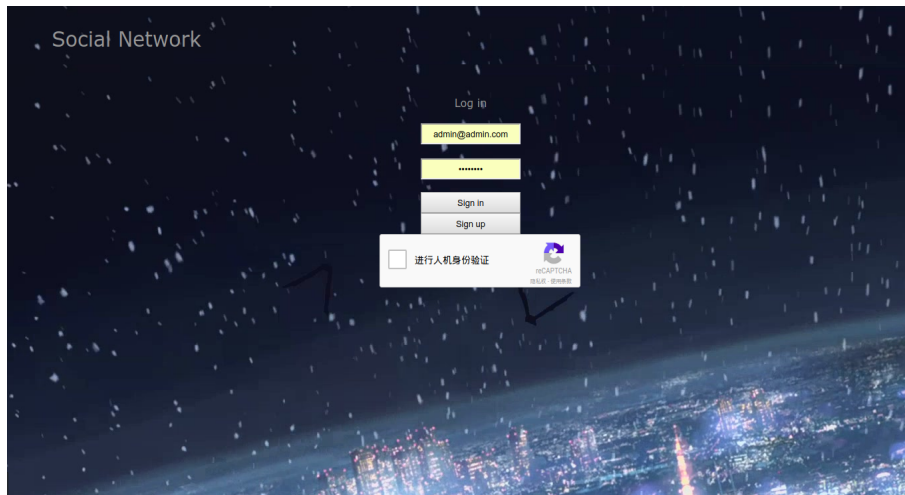


Figure 3: Log-in Interface

7 References

- Guidebook, installers and demo provided at course.pku.edu.cn
- \LaTeX template provided by [Overleaf](https://overleaf.com/)
- [mysql-connector-java-5.1.42-bin.jar](#)
- [javax.json-api-1.1.jar](#)