

同济微博系统

软件测试项目 系统测试文档

1751324 贺思奇

1752079 尤 皓

1551531 吴国栋

1652657 刘 寅

# 目录

[1. 目录 1](#_Toc44372113)

[2. 引言 2](#_Toc44372114)

[2.1 文档目的 2](#_Toc44372115)

[2.2 预期读者 3](#_Toc44372116)

[3. 项目概述 4](#_Toc44372117)

[3.1 页面和路由 4](#_Toc44372118)

[3.2 API 4](#_Toc44372119)

[登录API 4](#_Toc44372120)

[注册API 4](#_Toc44372121)

[设置API 4](#_Toc44372122)

[微博首页API 5](#_Toc44372123)

[个人主页API 5](#_Toc44372124)

[广场页API 5](#_Toc44372125)

[@页API 5](#_Toc44372126)

[3.3 组件 6](#_Toc44372127)

[3.4 架构设计 6](#_Toc44372128)

[3.5 数据库设计 6](#_Toc44372129)

[4. 系统功能分析 7](#_Toc44372130)

[4.1 功能需求 7](#_Toc44372131)

[4.2 性能需求 8](#_Toc44372132)

[4.3 安全需求 9](#_Toc44372133)

[4.4 错误处理 9](#_Toc44372134)

[4.5 其他需求 9](#_Toc44372135)

[5. 测试背景 9](#_Toc44372136)

[5.1 测试方法 9](#_Toc44372137)

[5.2 测试环境 10](#_Toc44372138)

[硬件环境 10](#_Toc44372139)

[软件环境 10](#_Toc44372140)

[网络环境 11](#_Toc44372141)

[5.3 测试对象 11](#_Toc44372142)

[6. 测试用例分析与设计 11](#_Toc44372143)

[6.1 用例1：注册 11](#_Toc44372144)

[6.2 用例2 登录 12](#_Toc44372145)

[6.3 用例3：设置基本信息 13](#_Toc44372146)

[6.4 用例4：修改密码 14](#_Toc44372147)

[6.5 用例5：关注、粉丝列表、关注列表 15](#_Toc44372148)

[6.6 用例6：发表+个人空间+广场页+首页 16](#_Toc44372149)

[6.7 用例7：回复+@提到我的 17](#_Toc44372150)

# 引言

## 文档目的

本文档为“微博系统”系统测试活动提供范围、方法、资源和进度方面的指导。主要描述如何进行系统测试活动，如何控制系统测试活动，及策划功能测试活动的流程等。在测试的过程中，尽可能多地发现功能上的缺陷和错误，对系统进行不断地优化和改进，保证系统能够正确实现其预期功能并能够满足客户当前的需求。

在整个系统投入运行之前，对系统的各元素进行组装和确认测试，确保在系统实际运行时 软件、硬件、外设、网络等系统元素能够相互配合并正常工作。 除了验证系统的功能外，还会涉及安全性、性能压力、 可用性、可靠性、可恢复性等方面的测试 。

## 预期读者

本文档的预期读者包括但是不限于：

1. 测试人员：指导和核查单元测试人员的工作，对后续的测试人员提供测试资料，以帮助其更快地明确测试重点，设计测试方案。
2. 开发人员：指导项目的开发者修正项目缺陷和错误，补全需求文档中列举的功能点，指导后续的开发过程。
3. 项目经理：对整个项目的流程进行监管，方便经理对测试工作进行系统地安排。

我们强烈地希望读者能够在一定程度上掌握如下的相关知识，这能够帮助读者更好地理解本文档。

（1）熟悉计算机的基本网络操作。

（2）熟悉当前常用的测试工具。

# 项目概述

## 页面和路由

* 注册 **/register**
* 登录 **/login**
* 首页 **/**
* 个人主页 **/profile/: username**
* @页 **/atMe**
* 广场 **/Square**
* 设置 **/setting**
* 错误页 **/error**
* 404 **/\***

## API

### 登录API

* 登录 **/api/user/login**

### 注册API

* 注册 **/api/user/register**
* 用户名是否存在 **/api/user/isExist**

### 设置API

* 修改个人信息 **/api/user/changeInfo**
* 图片上传 **/api/ utils/upload**
* 修改密码 **/api/user/changePassword**
* 退出登录 **/api/user/logout**

### 微博首页API

* 创建微博**/api/blog/create**
* 图片上传 **/api/utils/upload**
* 加载更多 **/api/blog/loadMore/:pageIndex**

### 个人主页API

* 加载更多 **/api/profile/ loadMore/:username/:pageIndex**
* 关注 **/api/profile/follow**
* 取消关注 **/api/ profile/unFollow**

### 广场页API

* 加载更多 **/api/square/ loadMore/: pageIndex**

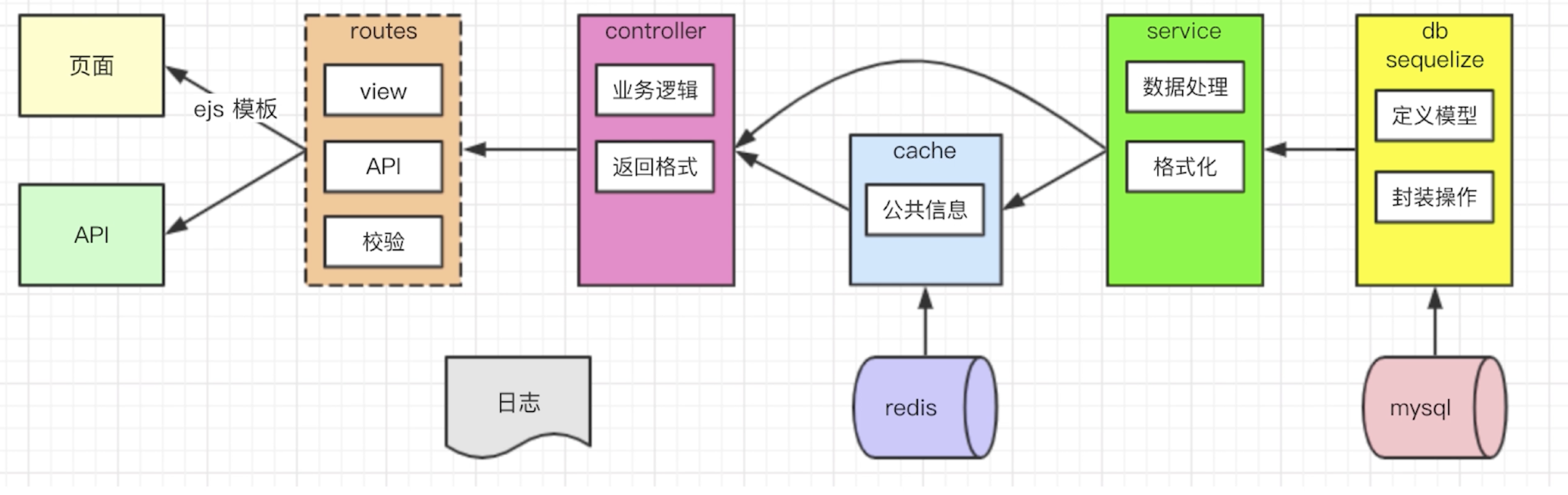
### @页API

* 创建微博 **/api/blog/create**
* 图片上传 **/api/utils/upload**
* 加载更多 **/api/atMe/loadMore/: pageIndex**

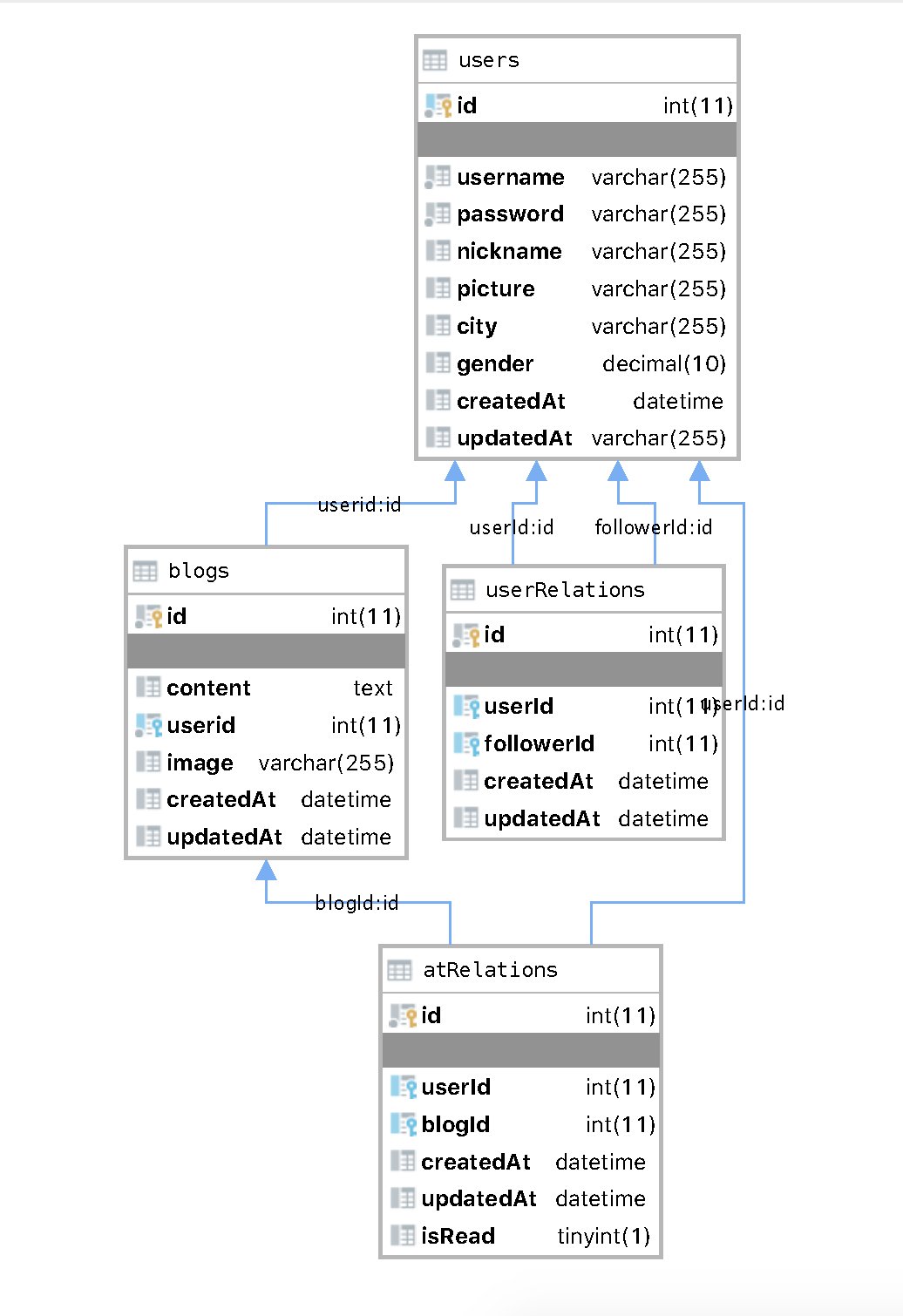
## 组件

* 发布博客输入框
* 博客列表
* 加载更多
* 个人信息
* 粉丝列表
* 关注人列表

## 架构设计



## 数据库设计



# 系统功能分析

## 功能需求

* 用例1：注册

描述：用户注册新的账号

* 用例2 登录

描述：用户登录系统

* 用例3：首页

描述：用户可以看到自己和关注的人发的微博，可以加载更多，看到个人信息、粉丝数、关注数等。

* 用例4：发表

描述：发布一个新的微博，可以插入图片。

* 用例5：回复

描述：可以引用评论某条微博，并@微博的发布者。

* 用例6：个人空间

描述：查看自己发布的所有微博（我的空间），或者某个人发布的所有微博，

* 用例7 ：广场

描述：查看所有人发布的微博。

* 用例8：设置基本信息

描述：修改自己的昵称，城市、头像。

* 用例9：修改密码

描述：修改自己的账户密码。

* 用例10：@提到我的

描述：查看@到我的消息。

* 用例11：关注功能

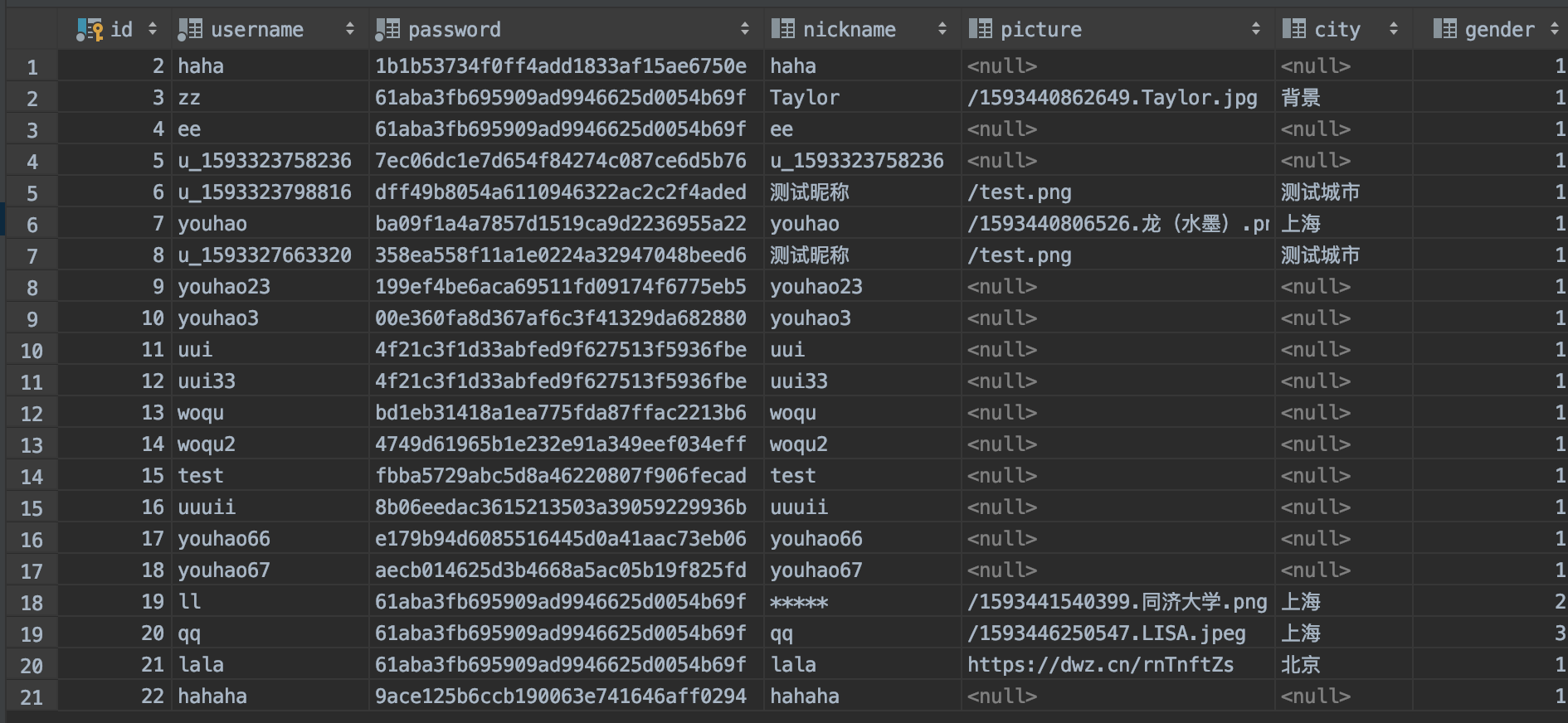
描述：关注某一个用户。

## 性能需求

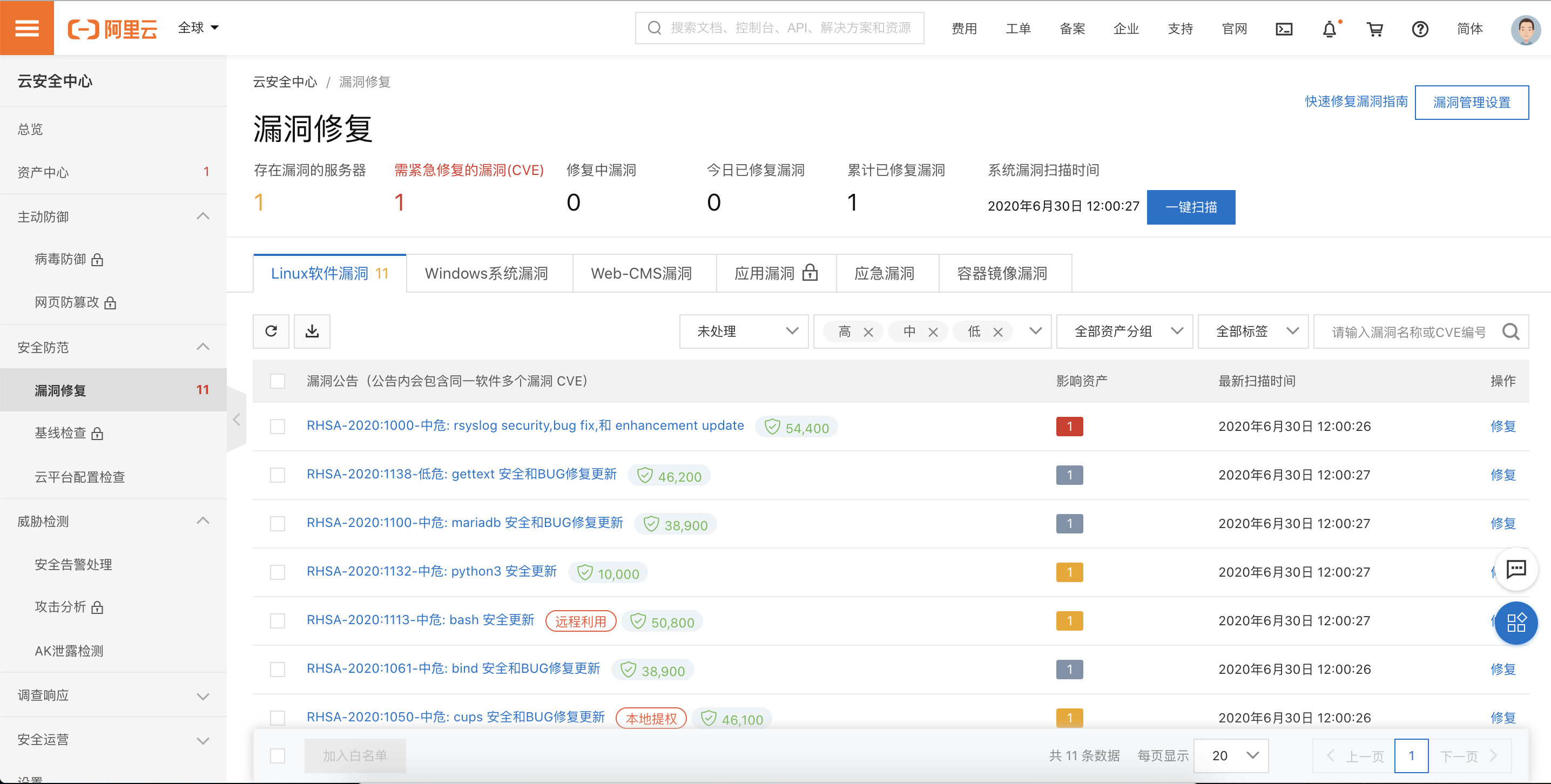
* 大容量数据测试:测试当网数据库中插入大量数据的时候，前端的渲染与显示所需要的时间在 允许的范围内。
* 响应时间:如果刷新时响应时间超过 30s，系统再次执行刷新。

## 安全需求

* 处理xss
* 使用ORM防止SQL注入（Sequelize）
* 加密敏感信息（避免明文存储密码）



* 漏洞扫描



## 错误处理

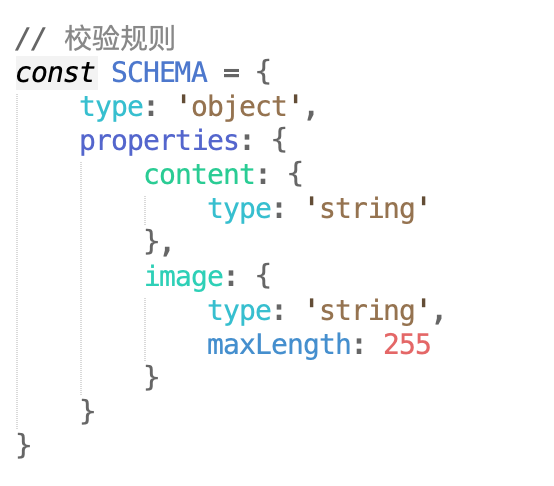
* 规范错误数据：ErrorInfo.js
* 对输入数据进行Schema校验

## 对输入输出的规定

User校验规则



Blog信息校验：



### 错误码定义



## 其他需求

* 记录日志
* 多进程
* 进程守护
* Nginx代理
* 分服务（MySQL、Redis）

# 测试背景

## 测试方法

系统从如下方面进行测试分析并结合具体的用例设计技术(主要是黑盒测试技术)来设计测试用例:

(1)系统级功能;  
(2)系统业务流;  
(3)系统级别的接口;  
(4)系统级别的输入和输出;  
(5)系统级别的状态转换;  
(6)系统级别数据;  
(7)系统非功能。

由于运用这些方法分析并设计的测试用例具有交叉重复的可能，因此，我们对测试用例集进行了严格评审，除去不合理的和重复的测试用例。另外，可能由于没有达到某种覆盖要求或覆盖不足还需要补充一些测试用例。

## 测试环境

### 硬件环境

用户端： Chrome（或使用IE8以上浏览器、Safari、Firefox）

服务端：内存：8G Nginx PM2

### 软件环境

* JAVA（1.8）
* Node （v 12.12.0）

数据库：

* MySQL（v 8.0.17）
* Redis（v 5.0.6）

### 网络环境

(1) 客户端:中国移动、中国电信、中国网通之一等

(2) 服务器:个人服务器即可

## 测试对象

微博系统（应用软件版本）

# 测试用例分析与设计

## 用例1：注册



## 用例2 登录



## 用例3：设置基本信息



## 用例4：修改密码



## 用例5：关注、粉丝列表、关注列表



## 用例6：发表+个人空间+广场页+首页



## 用例7：回复+@提到我的

