软件设计模式

**I M**

**需求文档**

|  |  |  |
| --- | --- | --- |
| **项目组成员信息** | | |
| **小组联系人** | **李析航** | |
| **学号** | **姓名** | **主要承担的工作内容** |
| 17373103 | 李析航 | 需求文档引言部分 |
| 17373095 | 唐玥 | 架构设计系统环境部分 |
| 17373116 | 徐洪晟 | 架构设计系统建设目标部分 |
| 17373471 | 刘卓月 | 架构设计文档结构设计部分 |
| 17373530 | 徐永鑫 | 架构设计文档结构设计部分 |
| 17373249 | 李景熙 | 需求文档非功能需求部分 |
| 17373463 | 袁梓晽 | 需求文档项目概述部分 |
| 17373031 | 张欣玥 | 需求文档功能需求部分 |

2020-04

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2020.04.15 | 李析航 | 李析航 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

**[一、 引言 2](#_Toc21351)**

[1． 编写目的 2](#_Toc2768)

[2． 背景 3](#_Toc24524)

**[二、 项目概述 3](#_Toc6238)**

[1． 项目目标 3](#_Toc3205)

[2． 系统基本技术 3](#_Toc18730)

[3． 软件系统约束 3](#_Toc12254)

[4． 总体流程 4](#_Toc11781)

**[三、 功能需求 4](#_Toc17464)**

[1． 登录系统： 4](#_Toc6492)

[2． 好友以及群聊系统 4](#_Toc13738)

[3． 聊天系统 4](#_Toc9405)

**[四、 非功能需求 4](#_Toc1537)**

[1． 用户界面需求 4](#_Toc10095)

[2． 性能需求 5](#_Toc16741)

[3． 系统需求 5](#_Toc20213)

[4． 可靠性需求 5](#_Toc20835)

[5． 语言需求 5](#_Toc20590)

1. 引言
   1. 编写目的

此文档是聊天软件开发过程中需求分析、产品开发、产品设计和测试方案的基础。

* 1. 背景

该项目为软件设计模式课程的课程作业。在计算机网络发展如此迅速的今天，无论 是人们的日常生活还是学习工作，都越来越依赖于互联网快捷的信息传输交流功能， 该聊天软件可以满足用户在日常生活中的交流需求，能够让用户之间的沟通变得更 加便利。

1. 项目概述
   1. 项目目标

实现一个类似微信或者QQ的聊天工具,能完成文字聊天、图片聊天、多人同时聊天这些基本功能。

* 1. 系统基本技术

技术栈是vue+springboot+websocket+mongodb。

利用websocket实现双向调用，客户端注册服务端要调用的方法，服务端注册客户端要调用的方法。

* 1. 软件系统约束

设计约束主要分三部分：系统的基本设计约束、数据库设计约束、外部交互设计约束。

* + 1. 系统的基本设计约束

1. 减少跨系统的交互，一个系统尽量只管理自己业务域内的数据库，不要跨域去操作其它应用的数据
2. 尽量减少IO的访问，将多次的调用整合在一次操作中完成，尽量减少IO资源的浪费
3. 禁止把操作数据或者外部接口的操作放在循环里，尽量做成批量接口调用
4. 系统间的调用最好是制度，系统之间的修改用事件或者消息来实现比较靠谱
5. 对于系统的配置文件，数据库字段的修改，或者其它显示复杂逻辑修改的操作，尽量采用增加的操作，而少用update的操作；update永远比insert成本大很多
6. 系统必须支持横向扩展，底层数据库及上层应用本身都需要支持扩展来满足未来业务的增长需求
7. 强一致性在微服务架构下不合适，互联网公司一般会采取基于消息的一致性事务的解决方案
8. 系统需要有区分主次的功能，对于主要功能需要加日志层面或者监控程名的告警逻辑
9. 尽量充分地利用CPU的资源，在很多情况下，一个应用的CPU资源都利用不充分。瓶颈往往在于IO层面，所以可以多引入线程池，让CPU的使用率最大化
10. 系统之间交互，拉的效果往往比推来的稳定性高，选择只读API，而不是写API，写部分尽量采用事件驱动或消息驱动
11. 往往内存中的复杂数据结构组装要优先于数据库的链接
    * 1. 数据库设计约束

1）尽可能在数据模型上控制业务对象的约束关系，如果通过程序逻辑去保证完整性于一致性，会存在一定的风险

2）数据模型总的唯一性约束，一定要在数据库层面得到控制，控制层面的幂等不太安全

3）尽量少用存储过程，将复杂的业务逻辑抽离到上层应用总，也就是尽量使用程序中的数据结构完成复杂的关系运算，避免用存储过程或者复杂的SQL语句

4）SQL语句尽量不要一句业务逻辑以及动态拼接的SQL字符串，而是采用预编译的方式

5）如果主表与子表是一对一的关系，主键尽量相同

6）数据库是各个业务系统的私有资源，其他系统对于该数据结构应该是透明的，只能通过接口和事件去访问和修改数据

* + 1. 外部交互设计约束

1）最好是把对方的数据拉过来，这样比对方推过来的稳定性会好一点

2）异步消息处理时，最好先落地到本地库再进行处理

3）系统中只能有一种异常，处理中状态等待超时或者重试次数达到最大值

* 1. 总体流程
     1. 单独聊天

单聊功能的实现流程为：点击发送按钮->服务端接受->客户端在页面挂载时请求获取当前用户的所有消息。这里需要注意，需要给输入框绑定当前聊天好友的消息变量，没有好友时单独绑定一个变量。

* + 1. 添加好友

添加好友的流程：服务端发起请求->利用vue更新好友列表->写在socket中->服务端调用->通过websocket更新对方的好友信息。

* + 1. 发图片

发图片的流程：把点击上传以后的全部图片的名字和文件对象全部存在一个数组内->把文件名字放在data-name属性内->在富文本中使用fileReader.readAsDataURL(file)方法将图片解析成字符串显示->点击发送按钮时操作DOM查找富文本中的IMG标签->通过data-name属性拿到文件名->通过文件名找到文件对象->以键值对的方式存在FormData对象中 在发送消息之前将formData对象发送给服务器->服务器存储后返回键值对->对img标签的src进行赋值->最后发送消息字符串

1. 功能需求
   1. 登录系统：
   * 用户可以用手机号或邮箱注册新账号，系统自动分配一个八位数字作为用户ID。
   * 用户登录时，需要输入ID以及密码，系统应该检查登录信息是否正确，并向客户端返回登陆信息（登陆成功或登陆失败）。
   * 用户成功登录后，可以显示该用户好友列表及登录状态，以及加入的群聊列表，并将用户在线状态同步给该用户的好友。
   * 用户登录成功后可以保存其号码，下次可直接登录。
   * 用户可以修改密码。
   1. 好友以及群聊系统
   * 用户可以通过搜索用户ID的方式添加好友。
   * 用户可以删除好友。
   * 用户可以创建群聊，系统自动分配一个八位数字作为群聊ID。
   * 用户可以通过搜索群聊ID进入群聊。
   * 用户可以退出群聊。
   1. 聊天系统
   * 用户可以与好友一对一聊天，可以在群聊中多对多聊天。
   * 聊天过程中用户可以发送文字、图片、文件。
   * 聊天记录可保存，用户可以查看聊天记录。
2. 非功能需求
   1. 用户界面需求

（1）尽可能使用列表框、选择框等鼠标操作的控件接受用户输入；

（2）系统界面可以更换主题、更改背景图片；

（3）用户进行错误的输入后系统要有相应的反馈；

（4）界面风格要统一。

* 1. 性能需求

（1）在95%的情况下，一般时段的响应时间不超过1.5秒，高峰时段不超过4秒；

（2）登录响应时间在2秒以内，刷新栏目响应时间在2秒内，刷新条目分页列表响应 时间2秒内，打开信息条目响应时间1秒内；

（3）在非高峰时段根据特定条件进行搜索其他用户，可以在3秒内得到搜索结果；

（4）信息传送时间在2秒以内，高峰时期不超过5秒；

（5）系统可以同时满足10,000个用户请求；

（6）占用的系统资源不会超过50%。

* 1. 系统需求

支持IOS，Android操作系统。Android版本8.0以上，IOS应在11.0版本以上。

* 1. 可靠性需求

（1）对输入有提示，数据有检查，防止数据异常

（2）系统健壮性强，应该能处理系统运行过程中出现的各种异常情况，如人为操作错 误、输入非法错误、硬件设备失败等，系统应该能正确处理并加以恰当的回避

（3）因软件系统的失效而造成不能完成业务的概率要小于0.5%

（4）系统缺陷每1000小时最多发生一次故障

* 1. 语言需求

支持简体中文与英文的输入，传输与显示。