软件设计模式

**I M**

**架构设计文档**

|  |  |  |
| --- | --- | --- |
| **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、 文档简介 1**](#_Toc4201)

[1． 编写目的 1](#_Toc18971)

[2． 文档概述 1](#_Toc11129)

[3． 术语定义 1](#_Toc21299)

[4． 参考资料 2](#_Toc21240)

[**二、 项目背景 2**](#_Toc8060)

[**三、 系统建设目标 2**](#_Toc1128)

[**四、 系统环境 2**](#_Toc11371)

[基本硬件要求： 2](#_Toc19832)

[基本软件环境要求 3](#_Toc1727)

[**五、 架构设计 3**](#_Toc3730)

[1． 逻辑架构 3](#_Toc338)

[2． 拓扑架构 4](#_Toc23254)

[3． 部署架构 4](#_Toc23254)

[4． 模块划分 4](#_Toc1966)

1. 文档简介
   1. 编写目的

本文档旨在说明即时通讯工具的架构设计方案，包括高层设计与部分详细设计， 为后续开发提供指导，使开发人员明确系统设计，从而降低项目风险、提高开发效 率。

* 1. 文档概述

本文档首先介绍项目背景，然后说明系统的核心功能需求与非功能需求，并介 绍系统的运行环境，最后重点阐述系统的架构设计，包括架构分析、设计思想、系 统视图、模块划分等部分，对系统的概要设计和部分详细设计进行说明。

* 1. 术语定义

1. IM：Instant message，即时通讯软件。
2. C/S：Client/Server，客户端/服务器架构。
3. MVC：Model-View-Controller，模型-视图-控制器架构。
4. ZooKeeper：为大型分布式计算提供分布式配置服务、同步服务和命名注册服务的开源框架。
5. Redis：使用ANSI C编写的开源、支持网络、基于内存、可选持久性的键值对存储数据库。
6. Spring Boot：Java平台用于创建独立的、生产级别的Spring应用程序的框架。
7. MongoDB：面向文档的数据库管理系统，用C++等语言撰写而成，以解决应用程序开发社区中的大量现实问题。
8. Vue：用于创建用户界面的开源JavaScript框架，也是一个创建单页应用的Web应用框架。
   1. 参考资料
9. 维基百科：<https://zh.wikipedia.org/>
10. crossoverJie/cim：<https://github.com/crossoverJie/cim>
11. farsunset/lvxin-im：<https://github.com/farsunset/lvxin-im>
12. 《软件系统分析与设计》课程项目文档
13. 《软件工程基础》课程项目文档
14. 项目背景

本次项目是软件设计模式课程的期末作业，要求实现一个类似微信或QQ的聊天工具，能够实现文字聊天、图片聊天和多人同时聊天的功能。我们小组选择实现web端的聊天工具。

1. 系统建设目标

根据上述聊天软件的需求分析，本多功能聊天软件借鉴了QQ、聊天室等及时聊天工具，实现了多人聊天和私聊等功能，聊天的内容包括文本和图片表情信息。本软件应包括服务器程序和客户端程序，服务器包括聊天信息的即时转发，自动关机的设置及记载，发送通知，数据库备份还原及数据库的初始化。客户端具有登录，注册，修改个人信息，聊天等功能。

1. 系统环境

由于系统将由两部分程序组成，服务器和客户端

* 1. 客户运行环境

客户端子系统的运行要求：

|  |  |
| --- | --- |
| CPU ： | 1.0GHZ 及其以上 |
| 内存： | 128M 及其以上 |
| 硬盘空间： | 2G 及其以上 |
| 网卡 | 10M/S |
| 显示设备 | 13寸及其以上彩显 |

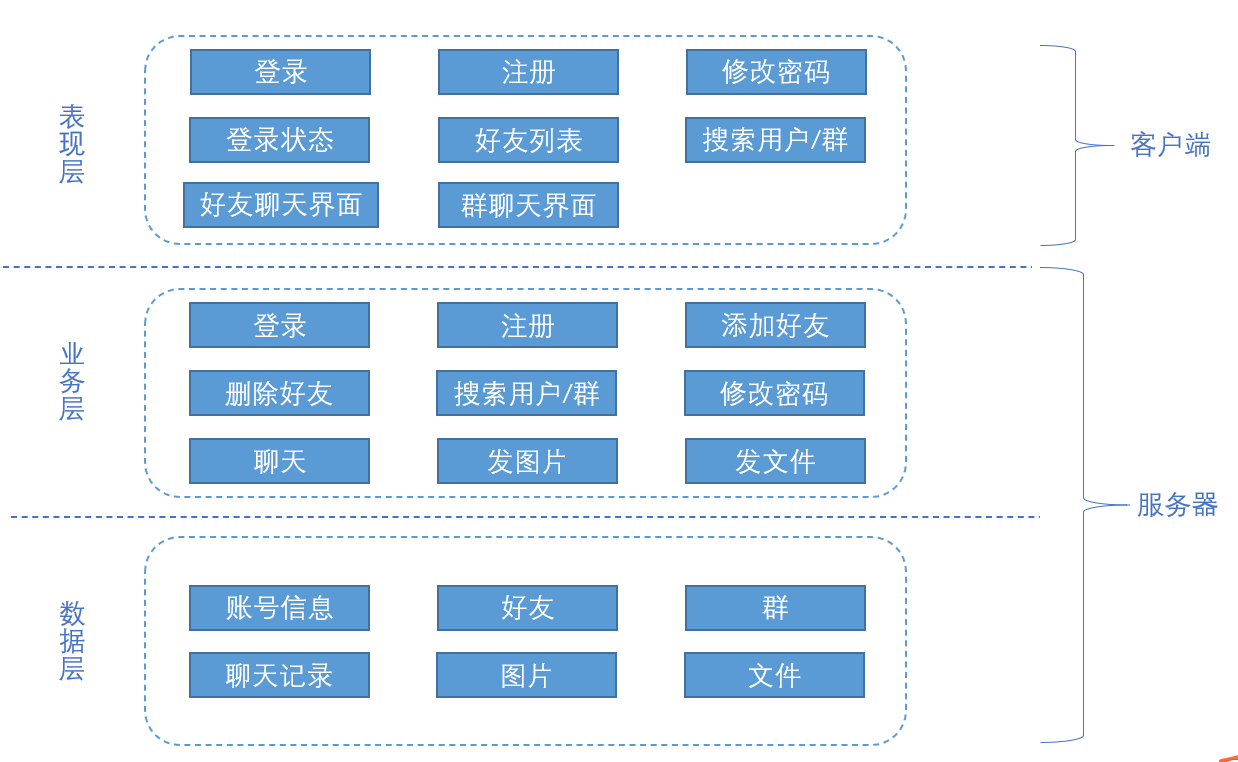
基本硬件要求：

|  |  |
| --- | --- |
| 操作系统 | Win98/win2000/winXP及其以上版本 |
| 网速 | 35.5K/S |

基本软件环境要求

* 1. 架构约束  
      <本节描述设计与实现时必须考虑的约束条件，并尽可能说明施加这些约束条件的 原因。它不影响系统的外部行为， 但必须被满足以达到技术、 商业或合同的要求。 至少应考虑两类  
      限制：一般限制、用户界面限制。  
      设计约束的描述格式为 ：  
      <需求标识 + 两个空格 +需求名称 >  
      需求描述 ： <需求详细描述 >  
      优先级 ： <参见功能需求优先级描述 >

1. 架构设计
   1. 逻辑架构



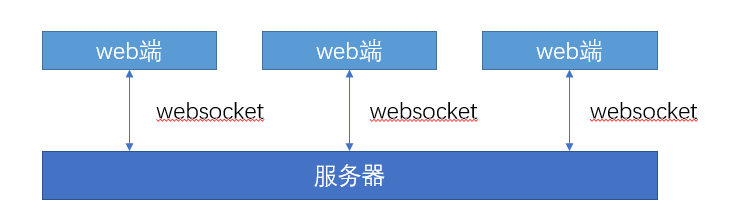
系统分为三层，分别是表现层、业务层和数据层，其中表现层位于web客户端上，其余两层在服务器上。

表现层包含登录、注册、修改密码、登录状态、好友列表、搜索用户/群、好友聊天界面和群聊天界面等组件，负责和用户交互，以及和业务层之间的数据传输。具体实现使用vue，数据通信使用websocket实现和服务器之间的双向通信。

业务层包含登录、注册、添加和删除好友、搜索用户/群、修改密码、聊天、发图片和发文件等业务，为表现层提供服务，响应表现层的请求，并根据请求操作数据层中的数据。具体实现使用springboot，数据通信使用websocket实现和服务器之间的双向通信。

数据层包含账号信息、好友、群、聊天记录、图片、文件等数据，具体实现使用MongoDB。

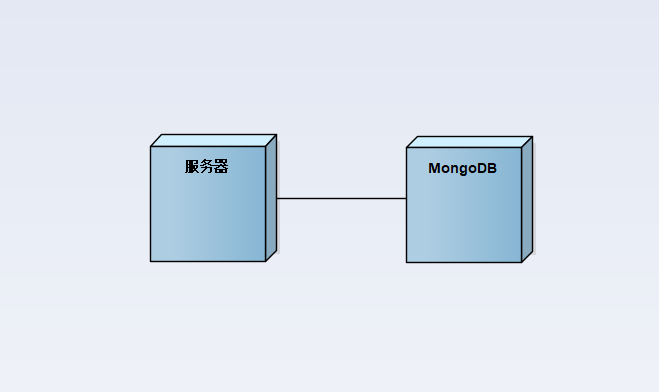
* 1. 拓扑架构



多个web端通过websocket和服务器进行双向通信。

websocket是一种在单个TCP连接上进行全双工通信的协议。websocket能够实现客户端和服务器间的长时间双向通信，不用建立多次连接，可以提高通信效率，故选择websocket。

* 1. 部署架构



MongoDB是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案，适用于本项目的web应用。MongoDB是一个介于关系数据库和非关系数据库之间的产品，它支持的数据结构非常松散，是类似json的bson格式，因此能够在支持包括账号信息等表格类数据的同时支持图片和文件等比较复杂的数据类型。

* 1. 模块划分
     1. 模块描述

客户端和服务器端均分为界面显示模块、业务处理模块和数据传送模块三大模块。

（1）界面显示模块：显示系统界面。

（2）业务处理模块：处理系统基本业务请求。

（3）数据传送模块：传送、处理系统数据。

* + 1. 模块接口
       1. 客户端

界面显示模块

登录界面：用户进入系统时自动进入登录界面，用于输入账号、密码，实现用户登录。

注册界面：用户点击“注册”按钮进入注册界面，输入基本信息可实现用户注册。

密码修改界面：点击“忘记密码”或“修改密码”进入密码修改界面，实现用户密码的修改。

主界面：用户登录后进入主界面，包含群聊、好友等基本信息，可通过主界面进入其他界面。

聊天界面：用户选择群聊或私聊后进入聊天界面，可发送、接收消息。

业务处理模块

连接服务器：用户端可以通过线程与对应的服务器端进行连接。

登录：用户在登录界面输入账号、密码并点击登录，发送请求信息到服务器端，若请求通过，则实现用户登录。

聊天：包括群聊和私聊。客户端点击“发送”按钮，将该客户端ip地址、端口号以及信息内容打包写入服务器的存储数组，其他客户端可以读取该数组中的内容，实现群聊；客户端输入私聊对象的账号及ip地址，开启端口监听，同时定义输入、输出流文件，另一客户端与该端口取得连接，任一方发送信息后，也可通过共享文件内容实现私聊。

查找：查找用户、聊天内容等基本信息。

下线：客户端向服务器端发送下线请求，断开连接，实现用户下线。

数据传送模块

用户信息：向服务器获取、操作用户信息。

聊天数据：向服务器获取、操作聊天数据。

发送数据：向服务器发送数据。

接收数据：从服务器接收数据。

* + - 1. 服务器端

界面显示模块

登录界面：管理员可通过登录界面访问服务器端。

主界面：登录后进入服务器端主界面。

业务处理模块

连接客户端：服务器端使用设定的端口进行监听，接收客户端请求，与客户端建立连接。

登录：在数据库中查找账号信息，若一致，则实现登录。

管理：对客户端进行管理。

处理客户端请求：定义一个用于存储客户端端口号及信息的数组，实现服务器、客户端共享，服务器将客户端发送的信息进行存储，并通知目标客户端进行读取，实现将信息的转发。

数据传送模块

数据库操作：连接数据库，对数据库进行操作。

发送数据：向客户端发送数据。

接收数据：从客户端接收数据。