

**《软件技术方向综合设计（二）》**

**项目报告**

|  |  |
| --- | --- |
| 学 号： | 20242160A1311 |
| 姓 名： | 刘灿 |
| 班 级： | 计信2213 |
| 项目名称： | 飞克尔聊天系统 |
| 指导老师： | 吴海子 |

20 25 年 4 月 15 日至20 25 年 6 月 6 日

**湖北师范大学制**

**目录**

[第一章 研究背景与意义 1](#_Toc200202551)

[1.1 研究背景与意义 1](#_Toc200202552)

[1.2 国内外研究现状 1](#_Toc200202553)

[1.2.1 国外研究现状 1](#_Toc200202554)

[1.2.2 国内研究现状 1](#_Toc200202555)

[1.3 研究内容 2](#_Toc200202556)

[第二章 相关理论与技术基础 3](#_Toc200202557)

[2.1 IDEA相关简介 3](#_Toc200202558)

[2.2 WEB 应用服务器 Tomcat 3](#_Toc200202559)

[2.3 Java 语言简介 3](#_Toc200202560)

[2.4 MySQL数据库简介 4](#_Toc200202561)

[2.5 SpringBoot框架简介 4](#_Toc200202562)

[2.6 Mybatis框架简介 5](#_Toc200202563)

[2.8 MyBatis-Plus框架简介 5](#_Toc200202564)

[2.9 Netty框架简介 6](#_Toc200202565)

[2.10 VUE 前端框架简介 6](#_Toc200202566)

[2.11 uniapp前端框架简介 7](#_Toc200202567)

[第三章 系统需求分析 8](#_Toc200202568)

[3.1 系统可行性分析 8](#_Toc200202569)

[3.1.1 技术可行性研究 8](#_Toc200202570)

[3.1.2 运行可行性研究 8](#_Toc200202571)

[3.2 系统总体需求 8](#_Toc200202572)

[3.2.1 系统总体概述 8](#_Toc200202573)

[3.2.2 系统业务流程分析 8](#_Toc200202574)

[3.3 系统功能需求 9](#_Toc200202575)

[3.3.1用户管理功能模块 9](#_Toc200202576)

[3.3.2 好友管理功能模块 9](#_Toc200202577)

[3.3.3 群组管理功能模块 9](#_Toc200202578)

[3.3.4 聊天功能模块 10](#_Toc200202579)

[3.4 非功能性需求 10](#_Toc200202580)

[第四章 系统设计 12](#_Toc200202581)

[4.1 系统总体设计 12](#_Toc200202582)

[4.1.1 系统总体架构 12](#_Toc200202583)

[4.1.2 系统功能模块设计 12](#_Toc200202584)

[4.1.3 系统网络拓扑架构设计 12](#_Toc200202585)

[4.2 数据库设计 12](#_Toc200202586)

[4.2.1 数据库的概念模型 12](#_Toc200202587)

[实体关系 14](#_Toc200202588)

[4.2.2 数据库物理结构设计 15](#_Toc200202589)

[4.3 系统详细设计 17](#_Toc200202590)

[4.3.1 用户管理功能详细设计 17](#_Toc200202591)

[**4.3.2 好友管理功能详细设计** 18](#_Toc200202592)

[4.3.3 群组管理功能详细设计 18](#_Toc200202593)

[4.3.4 聊天功能详细设计 19](#_Toc200202594)

[第五章 系统实现 20](#_Toc200202595)

[5.1 系统实现环境 20](#_Toc200202596)

[5.2 用户管理功能实现 20](#_Toc200202597)

[5.3好友管理功能实现 21](#_Toc200202598)

[5.4 群组管理功能实现 21](#_Toc200202599)

[5.5 聊天功能实现 21](#_Toc200202600)

[第六章 系统测试 23](#_Toc200202601)

[6.1 测试目的 23](#_Toc200202602)

[6.2系统测试结果 23](#_Toc200202603)

[第七章 总结与展望 23](#_Toc200202604)

[7.1 研究工作总结 23](#_Toc200202605)

## 第一章 研究背景与意义

### 1.1 研究背景与意义

在当今数字化时代，人们的沟通方式变得越来越多样化，即时通讯软件成为了人们日常生活中不可或缺的一部分。随着移动设备的普及，人们几乎可以随时随地与他人进行交流，分享生活的点滴。然而，在享受这些便捷的同时，我们也时常会遇到一些小困扰，例如当我们想要与朋友进行深度交流时，却发现由于双方使用的手机操作系统不同，某些通讯应用无法互相兼容。这种情况下，要么一方妥协下载另一方推荐的应用程序，要么只能选择那些较为通用但可能功能有限的应用来维持联系。这样的经历不仅降低了沟通效率，有时也会让人感到些许无奈。

飞克尔聊天系统正是基于对这种普遍现象的理解与洞察而应运而生的。它以其卓越的跨平台兼容性，成功解决了这一难题。无论是苹果 iOS 系统、谷歌 Android 系统还是传统的个人电脑（PC），飞克尔聊天系统都能够完美适配，真正实现了无缝连接。这意味着，无论你身处世界的哪个角落，无论你的朋友使用的是哪一种设备，你们都可以轻松实现即时通讯，无需担心因技术壁垒而被隔绝开来。

### 1.2 国内外研究现状

#### 1.2.1 国外研究现状

国外在即时通讯领域的研究和发展相对成熟，有许多知名的即时通讯软件，如 Facebook Messenger、WhatsApp 等。这些软件具有广泛的用户基础和强大的功能，在全球范围内得到了广泛的应用。例如，Facebook Messenger 支持多种语言，具备语音通话、视频通话、文件传输等功能，并且与 Facebook 社交平台深度集成，方便用户与好友进行交流和互动。WhatsApp 则以其简洁的界面和高效的通讯功能受到了众多用户的喜爱，尤其在一些发展中国家拥有庞大的用户群体。

#### 1.2.2 国内研究现状

国内的即时通讯市场也非常活跃，腾讯的 QQ 和微信是国内最具代表性的即时通讯软件。QQ 拥有丰富的功能和个性化的设置，深受年轻用户的喜爱。微信则以其简洁易用的界面和强大的社交功能，成为了国内用户使用最广泛的即时通讯工具之一。此外，国内还有一些针对特定领域或用户群体的即时通讯软件，如企业级即时通讯软件钉钉，为企业提供了高效的沟通和协作平台。

### 1.3 研究内容

本实训主要围绕飞克尔聊天系统展开，研究内容包括系统的需求分析、设计、实现和测试等方面。具体来说，需要完成以下几个任务：

1. 对飞克尔聊天系统进行需求分析，明确系统的功能需求和非功能需求。
2. 设计系统的架构和数据库，包括客户端、服务器和数据库的设计。
3. 开发系统的客户端和服务器端，实现系统的各项功能。
4. 对系统进行测试，确保系统的稳定性和可靠性。

## 第二章 相关理论与技术基础

### 2.1 IDEA相关简介

IDEA 全称 IntelliJ IDEA，是 JetBrains 公司开发的一款集成开发环境（IDE），主要用于 Java 语言的开发。它具有强大的代码编辑、调试、重构等功能，能够大大提高开发效率。IDEA 支持多种编程语言和框架，如 Java、Scala、Groovy 等，并且提供了丰富的插件和工具，方便开发人员进行开发和调试。此外，IDEA 还具有智能代码提示、代码自动补全、代码分析等功能，能够帮助开发人员快速编写高质量的代码。

### 2.2 WEB 应用服务器 Tomcat

Tomcat 是一个开源的、免费的 Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。Tomcat 由 Apache 软件基金会开发和维护，它支持最新的 Servlet 和 JSP 规范，能够提供高效、稳定的 Web 服务。Tomcat 的特点包括：

1. 技术先进：采用了最新的 Servlet 和 JSP 规范，能够提供高效、稳定的 Web 服务。
2. 性能稳定：经过了大量的测试和优化，具有较高的性能和稳定性。
3. 免费开源：可以免费使用和修改，并且有丰富的社区资源和文档支持。
4. 易于使用：具有简单的配置和管理界面，方便开发人员进行使用和维护。

### 2.3 Java 语言简介

Java 是一种广泛使用的面向对象编程语言，由 Sun Microsystems 公司开发，现在由 Oracle 公司维护。Java 具有跨平台、面向对象、安全性高等特点，能够在不同的操作系统和硬件平台上运行。Java 的主要特点包括：

1. 跨平台性：Java 程序可以在不同的操作系统和硬件平台上运行，只需要安装相应的 Java 虚拟机（JVM）即可。
2. 面向对象：Java 是一种面向对象的编程语言，支持封装、继承和多态等面向对象的特性。
3. 安全性高：Java 具有严格的安全机制，能够有效地防止病毒和恶意软件的攻击。
4. 丰富的类库：Java 提供了丰富的类库，包括字符串处理、文件操作、网络编程等方面的类库，方便开发人员进行开发和编程。

### 2.4 MySQL数据库简介

MySQL 是一种开源的关系型数据库管理系统，由 MySQL AB 公司开发，现在由 Oracle 公司维护。MySQL 具有高性能、可靠性高、易于使用等特点，被广泛应用于各种 Web 应用和企业级应用中。MySQL 的主要特点包括：

1. 高性能：采用了先进的存储引擎和优化算法，能够提供高效、稳定的数据库服务。
2. 可靠性高：具有完善的备份和恢复机制，能够有效地保证数据的安全性和可靠性。
3. 易于使用：具有简单的配置和管理界面，方便开发人员进行使用和维护。
4. 开源免费：可以免费使用和修改，并且有丰富的社区资源和文档支持。

### 2.5 SpringBoot框架简介

SpringBoot 是 Spring 家族的一个子项目，它是一个用于简化 Spring 应用开发的框架。SpringBoot 通过自动配置和约定大于配置的原则，减少了 Spring 应用的配置工作量，提高了开发效率。SpringBoot 的主要特点包括：

1. 自动配置：SpringBoot 能够根据项目的依赖和配置自动进行配置，减少了开发人员的配置工作量。
2. 约定大于配置：SpringBoot 采用了约定大于配置的原则，通过默认的配置和约定来简化开发过程。
3. 嵌入式服务器：SpringBoot 内置了 Tomcat、Jetty 等 Servlet 容器，能够方便地将应用打包成可执行的 JAR 文件，直接运行。
4. 微服务支持：SpringBoot 与 Spring Cloud 等微服务框架集成，能够方便地开发和部署微服务应用。

### 2.6 Mybatis框架简介

MyBatis 是一款轻量级的 Java 持久层框架，最初由 Clinton Begin 于 2002 年创建，最初命名为 iBATIS，2010 年被 Apache Software Foundation 收购并更名为 MyBatis。它提供了将 Java 对象和 SQL 语句之间的映射，使得 Java 开发人员可以更加便捷地访问数据库。

1. 灵活性高：采用 XML 文件来管理 SQL 语句，开发人员可以根据需要自由编写 SQL 语句，从而实现对数据库的灵活操作。
2. 可配置性强：支持将 Java 对象与数据库表进行映射，开发人员可以通过 XML 文件来配置 Java 对象与数据库表之间的映射关系，从而实现对数据库表的操作。
3. 易于维护：将 SQL 语句抽象为可重用的组件，可以在不同的 SQL 语句中重复使用这些组件，从而提高程序的可重用性和可维护性。
4. 性能高：采用连接池技术，能够对数据库连接进行有效的管理，从而提高程序的性能和可维护性。
5. 易于集成：与 Spring 框架等其他常用框架集成比较容易，可以与其他框架一起使用，从而实现更加高效的开发。
6. 支持多种数据库：支持多种数据库，包括 Mysql、Oracle、Sql server 等，可以根据不同的需求选择适合的数据库。

### 2.8 MyBatis-Plus框架简介

MyBatis-Plus（简称 MP）是一个 MyBatis 的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。

1. 简化开发：提供了丰富的功能，使得开发人员可以更快地完成数据库操作，减少了重复代码的编写。
2. 性能优化：自带了多种优化手段，如缓存、批量操作等，能够显著提高系统的性能。
3. 业务逻辑分离：允许开发人员将业务逻辑与数据库操作分开，使得代码更加清晰易读。
4. 简化配置：提供了默认的配置，使得配置文件更加简洁，减少了配置的复杂度。
5. 适用场景：适用于需要进行 CRUD 操作的场景，如用户管理模块、商品管理系统、订单管理系统、博客系统、后台管理系统等。

### 2.9 Netty框架简介

Netty 是一个基于 Java 的高性能、异步事件驱动的网络应用框架，主要用于开发高并发、高吞吐量的网络服务器和客户端。

1. 异步和事件驱动：基于异步非阻塞的 I/O 操作，通过事件驱动的架构实现高效的网络通信。这种架构使得 Netty 能够轻松处理大量的并发连接，适用于开发高性能的网络服务器和客户端。
2. 高性能和低延迟：针对性能进行了高度优化，例如通过零拷贝技术减少内存复制的开销，提供高吞吐量和低延迟的网络通信能力。其高效的内存管理和多线程处理模型使得 Netty 成为开发高性能网络应用的首选框架之一。
3. 丰富的协议支持：支持多种传输协议，包括 TCP、UDP，以及高级协议如 HTTP、WebSocket、SSL/TLS 等。开发者还可以基于 Netty 框架轻松实现自定义协议，这使得 Netty 在复杂的网络应用场景中非常灵活。
4. 高度可定制化：提供了强大的可定制性，从线程模型到协议栈，都可以根据应用的具体需求进行调整。这种灵活性使得 Netty 能够适应各种不同的应用场景，无论是轻量级的客户端还是复杂的服务器端应用。
5. 良好的社区支持：作为一个成熟的开源项目，Netty 拥有一个活跃的社区和丰富的文档资源。无论是学习入门，还是深入开发，Netty 社区都能提供强大的支持。

### 2.10 VUE 前端框架简介

VUE 是一个用于构建用户界面的渐进式 JavaScript 框架，由尤雨溪开发。VUE 具有简单易用、灵活高效等特点，能够帮助开发人员快速构建交互式的 Web 应用。VUE 的主要特点包括：

1. 简单易用：VUE 的语法简单易懂，学习成本低，适合初学者使用。
2. 灵活高效：VUE 采用了虚拟 DOM 和响应式数据绑定等技术，能够提供高效、灵活的渲染性能。
3. 组件化开发：VUE 支持组件化开发，能够将页面拆分成多个小的组件，提高代码的复用性和可维护性。
4. 生态丰富：VUE 拥有丰富的插件和工具，如 Vue Router、Vuex 等，能够方便地进行路由管理和状态管理。

### 2.11 uniapp前端框架简介

uni-app 是一个基于 Vue.js 框架的跨平台应用开发框架。它支持使用 Vue 语法进行开发，并可以将代码编译为微信小程序、H5、iOS 和 Android 等多个平台的原生应用，实现了一套代码多端运行的目标。

1. 一套代码多端运行：采用了统一的组件和 API，开发者只需编写一套代码，就能够在不同的平台上运行，极大地简化了开发流程。
2. 性能优化：通过对底层的渲染机制进行优化，提升了应用的性能表现，使得跨平台应用在用户体验上更加流畅。
3. 生态丰富：拥有庞大的开发者社区和丰富的生态系统，开发者可以轻松找到各种插件和组件来满足不同需求。
4. 灵活扩展：支持原生代码扩展，开发者可以根据需要编写原生代码，并与 uni-app 结合使用，实现更复杂的功能。

## 第三章 系统需求分析

### 3.1 系统可行性分析

#### 3.1.1 技术可行性研究

从技术可行性的角度来看，开发飞克尔聊天系统所涉及的技术包括 Java 语言、SpringBoot 框架、VUE 前端框架、MySQL 数据库等，这些技术都是成熟的、广泛应用的技术。此外，市场上有许多开源的框架和工具可以供开发使用，能够大大提高开发效率和降低开发成本。因此，从技术可行性的角度来看，开发飞克尔聊天系统是可行的。

#### 3.1.2 运行可行性研究

从运行可行性的角度来看，飞克尔聊天系统的运行环境主要包括服务器、客户端和网络等。服务器可以采用云服务器或自建服务器，客户端可以是 Web 客户端或移动客户端，网络可以是局域网或互联网。通过合理的服务器配置和网络优化，飞克尔聊天系统能够在不同的运行环境下稳定运行。此外，开发团队可以提供完善的技术支持和维护服务，确保系统的正常运行。因此，从运行可行性的角度来看，开发飞克尔聊天系统是可行的。

### 3.2 系统总体需求

#### 3.2.1 系统总体概述

飞克尔聊天系统是一款仿 QQ 的即时通讯软件，旨在为用户提供高效、安全、跨平台的通讯体验。系统采用现代化技术架构设计，不仅支持基本的文字消息和文件传输功能，还具备完善的用户管理、社交网络和群组交流功能。用户可以通过飞克尔聊天系统与好友进行一对一或群组聊天，分享文字、图片、语音、视频等信息，还可以进行文件传输、表情发送等操作。此外，系统还支持用户管理、好友管理、群组管理等功能，方便用户进行社交和交流。

#### 3.2.2 系统业务流程分析

飞克尔聊天系统的业务流程主要包括用户注册、登录、聊天、文件传输、好友管理、群组管理等环节。用户首先需要注册一个账号，填写用户名、密码、手机号码等信息，然后通过手机号码或邮箱进行验证。注册成功后，用户可以使用用户名和密码登录系统。登录成功后，用户可以查看好友列表、群组列表，与好友进行聊天、文件传输等操作。用户还可以添加好友、创建群组、管理群组等。在聊天过程中，用户可以发送文字、图片、语音、视频等信息，还可以进行表情发送、文件传输等操作。系统会将用户发送的信息保存到数据库中，并及时推送给接收方。接收方可以及时查看和回复信息。

### 3.3 系统功能需求

#### 3.3.1用户管理功能模块

用户管理功能模块主要包括用户注册、登录、信息修改、修改密码等功能。用户可以通过注册功能创建一个新的账号，填写用户名、密码等信息。注册成功后，用户可以使用用户名和密码登录系统。登录成功后，用户可以修改个人信息，如头像、昵称、签名等。

#### 3.3.2 好友管理功能模块

好友管理功能模块主要包括好友添加、删除、查找、备注等功能。用户可以通过好友添加功能添加新的好友，输入好友的用户名，系统会自动查找并显示符合条件的用户。用户可以选择添加这些用户为好友。用户还可以通过好友删除功能删除已有的好友。用户可以通过好友查找功能查找指定的好友，输入好友的用户名，系统会自动查找并显示符合条件的用户。用户可以通过好友备注功能为好友设置备注名，方便识别和管理好友。

#### 3.3.3 群组管理功能模块

群组管理功能模块主要包括群组创建、加入、退出、管理等功能。用户可以通过群组创建功能创建新的群组，输入群组名称、群组简介等信息，选择群组的成员，系统会自动创建一个新的群组。用户可以通过群组加入功能加入已有的群组，输入群组的名称、群号或扫描群组的二维码等信息，系统会自动查找并显示符合条件的群组。用户可以选择加入这些群组。用户可以通过群组退出功能退出已有的群组。用户可以通过群组管理功能管理群组的成员、权限、公告等信息。

#### 3.3.4 聊天功能模块

聊天功能模块主要包括一对一聊天、群组聊天、消息发送、接收、查看、删除等功能。用户可以通过一对一聊天功能与好友进行单独的聊天，输入聊天内容，系统会自动将消息发送给对方。对方可以及时查看和回复消息。用户可以通过群组聊天功能与群组内的成员进行聊天，输入聊天内容，系统会自动将消息发送给群组内的所有成员。成员可以及时查看和回复消息。用户可以通过消息发送功能发送文字、图片、语音、视频等信息，还可以进行表情发送、文件传输等操作。用户可以通过消息接收功能接收来自好友或群组的消息，系统会自动将消息推送给用户。用户可以通过消息查看功能查看历史聊天记录，了解聊天的内容和时间。用户可以通过消息删除功能删除不需要的聊天记录。

### 3.4 非功能性需求

非功能性需求是指软件系统必须满足的非功能性要求，如安全性、性能、可靠性、易用性、可维护性等。飞克尔聊天系统的非功能性需求主要包括以下几个方面：

1. 安全性：系统需要采用安全可靠的加密算法对用户的信息和聊天内容进行加密，确保用户的隐私和数据安全。同时，系统需要提供完善的权限管理机制，对用户的操作进行授权和控制，防止非法用户的访问和操作。
2. 性能：系统需要具备高并发处理能力，能够同时处理大量用户的请求和消息，确保系统的响应速度和稳定性。同时，系统需要优化数据库和服务器的性能，提高数据的存储和查询效率。
3. 可靠性：系统需要具备高可靠性，能够在各种异常情况下正常运行，如网络故障、服务器故障等。同时，系统需要提供完善的备份和恢复机制，确保数据的安全性和完整性。
4. 易用性：系统需要具备简洁易用的用户界面，方便用户进行操作和使用。同时，系统需要提供完善的帮助文档和教程，帮助用户快速上手和使用系统。
5. 可维护性：系统需要具备良好的可维护性，能够方便地进行代码的修改和扩展。同时，系统需要采用模块化设计和分层架构，提高代码的复用性和可维护性。

## 第四章 系统设计

### 4.1 系统总体设计

#### 4.1.1 系统总体架构

飞克尔聊天系统采用分层架构设计，主要包括客户端层、服务器层和数据库层。客户端层负责与用户进行交互，接收用户的输入和操作，将用户的请求发送给服务器层。服务器层负责处理客户端的请求，对用户的信息和消息进行处理和存储，将处理结果返回给客户端层。数据库层负责存储用户的信息、聊天记录、文件等数据，为服务器层提供数据支持。

#### 4.1.2 系统功能模块设计

飞克尔聊天系统的功能模块主要包括用户管理模块、好友管理模块、群组管理模块、聊天模块等。用户管理模块负责用户的注册、登录、信息修改等功能。好友管理模块负责好友的添加、删除、查找等功能。群组管理模块负责群组的创建、加入、退出等功能。聊天模块负责一对一聊天、群组聊天、消息发送、接收等功能。

#### 4.1.3 系统网络拓扑架构设计

飞克尔聊天系统的网络拓扑架构采用客户端 - 服务器模式，客户端通过网络与服务器进行通信。

### 4.2 数据库设计

#### 4.2.1 数据库的概念模型

飞克尔聊天系统的数据库概念模型主要围绕用户、好友关系、私聊消息、群、群成员和群消息等核心实体展开，这些实体之间相互关联，共同构成了系统的数据基础。以下是对各实体及其关系的详细描述：

1. 用户（User）： 用户是系统的核心实体，代表使用飞克尔聊天系统的个体。每个用户具有唯一的标识（ID），同时包含用户名、昵称、头像、密码、性别、是否被封禁、被封禁原因、用户类型、个性签名、最后登录时间和创建时间等属性。用户名具有唯一性，用于用户的登录和识别；昵称则用于在系统中显示，方便其他用户识别；头像和头像缩略图为用户提供个性化展示；密码用于用户登录的身份验证；性别、用户类型等属性用于对用户进行分类和管理；最后登录时间和创建时间记录了用户的使用和注册信息。
2. 好友（Friend）： 好友关系是基于用户实体建立的，它表示两个用户之间的好友关联。好友实体包含唯一标识（ID）、用户 ID、好友 ID、好友昵称、好友头像、删除标识和创建时间等属性。用户 ID 和好友 ID 分别指向两个相关联的用户，通过这两个属性可以确定好友关系的双方；好友昵称和好友头像方便用户在好友列表中识别好友；删除标识用于标记好友关系是否已被删除。
3. 私聊消息（PrivateMessage）： 私聊消息实体用于记录用户之间的一对一聊天信息。它包含唯一标识（ID）、发送用户 ID、接收用户 ID、消息内容、消息类型、消息状态和发送时间等属性。发送用户 ID 和接收用户 ID 明确了消息的发送方和接收方；消息内容存储了聊天的具体信息；消息类型区分了不同形式的消息，如文字、图片、文件、语音、视频或提示等；消息状态记录了消息的读取和撤回情况；发送时间记录了消息发送的时刻。
4. 群（Group）： 群是用户的集合，用于实现多人交流的功能。群实体具有唯一标识（ID）、群名字、群主 ID、群头像、群头像缩略图、群公告、是否被封禁、被封禁原因、是否已解散和创建时间等属性。群名字用于标识群的名称；群主 ID 确定了群的管理者；群头像和群头像缩略图为群提供了可视化展示；群公告用于发布群内的重要信息；是否被封禁和是否已解散属性用于管理群的状态。
5. 群成员（GroupMember）： 群成员实体表示用户与群之间的关联关系。它包含唯一标识（ID）、群 ID、用户 ID、用户昵称、显示昵称备注、用户头像、显示群名备注、是否已退出、退出时间和创建时间等属性。群 ID 和用户 ID 分别指向所属的群和用户；用户昵称和显示昵称备注用于在群内显示用户信息；显示群名备注为用户提供了对群名的个性化设置；是否已退出和退出时间记录了用户在群内的状态变化。
6. 群消息（GroupMessage）： 群消息实体用于记录群内的聊天信息。它包含唯一标识（ID）、群 ID、发送用户 ID、发送用户昵称、接收用户 ID 列表、消息内容、被 @ 的用户 ID 列表、是否回执消息、回执消息是否完成、消息类型、消息状态和发送时间等属性。群 ID 确定了消息所属的群；发送用户 ID 和发送用户昵称明确了消息的发送者；接收用户 ID 列表指定了消息的接收对象，为空时表示发给所有成员；被 @ 的用户 ID 列表用于标记在消息中被 @ 的用户；是否回执消息和回执消息是否完成属性用于处理群消息的回执功能；消息类型和消息状态与私聊消息类似，用于区分消息的形式和状态。
7. 敏感词（SensitiveWord）： 敏感词实体用于管理系统中的敏感词汇。它包含唯一标识（ID）、敏感词内容、是否启用、创建者和创建时间等属性。敏感词内容存储了需要过滤的词汇；是否启用属性用于控制敏感词的生效状态；创建者记录了敏感词的添加者；创建时间记录了敏感词的添加时刻。

#### 实体关系

* 用户与好友：一个用户可以有多个好友，一个好友也对应一个用户，因此用户与好友之间是一对多的关系。通过用户 ID 和好友 ID 建立关联。
* 用户与私聊消息：一个用户可以发送和接收多条私聊消息，一条私聊消息对应一个发送用户和一个接收用户，所以用户与私聊消息之间是一对多的关系。通过发送用户 ID 和接收用户 ID 建立关联。
* 用户与群成员：一个用户可以加入多个群，成为多个群的成员，一个群成员对应一个用户，因此用户与群成员之间是一对多的关系。通过用户 ID 建立关联。
* 群与群成员：一个群可以有多个成员，一个群成员只属于一个群，所以群与群成员之间是一对多的关系。通过群 ID 建立关联。
* 群与群消息：一个群可以产生多条群消息，一条群消息只属于一个群，因此群与群消息之间是一对多的关系。通过群 ID 建立关联。
* 用户与群消息：一个用户可以在群内发送和接收多条群消息，一条群消息对应一个发送用户，所以用户与群消息之间是一对多的关系。通过发送用户 ID 建立关联。

这些实体和它们之间的关系构成了飞克尔聊天系统数据库的概念模型，为系统的数据存储和管理提供了清晰的逻辑框架。

#### 4.2.2 数据库物理结构设计

飞克尔聊天系统的数据库采用 MySQL 数据库，为了实现系统的各项功能，设计了以下几张表：

1. 用户表（`user`）：用于存储用户的基本信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 用户的唯一标识 |
| user\_name | varchar(255) | 非空，唯一索引 | 用户名，用于登录和识别用户 |
| nick\_name | varchar(255) | 非空，索引 | 用户昵称，用于显示在系统中 |
| head\_image | varchar(255) | 默认 '' | 用户头像的存储路径 |
| head\_image\_thumb | varchar(255) | 默认 '' | 用户头像缩略图的存储路径 |
| password | varchar(255) | 非空 | 用户登录密码 |
| sex | tinyint(1) | 默认 0 | 用户性别，0 表示男，1 表示女 |
| is\_banned | tinyint(1) | 默认 0 | 用户是否被封禁，0 表示否，1 表示是 |
| reason | varchar(255) | 默认 '' | 用户被封禁的原因 |
| type | smallint | 默认 1 | 用户类型，1 表示普通用户，2 表示审核账户 |
| signature | varchar(1024) | 默认 '' | 用户的个性签名 |
| last\_login\_time | datetime | 可空 | 用户最后一次登录的时间 |
| created\_time | datetime | 默认 CURRENT\_TIMESTAMP | 用户账户创建的时间 |

1. 好友表（`friend`）：记录用户之间的好友关系。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 记录的唯一标识 |
| user\_id | bigint | 非空，索引 | 用户的 ID |
| friend\_id | bigint | 非空，索引 | 好友的 ID |
| friend\_nick\_name | varchar(255) | 非空 | 好友的昵称 |
| friend\_head\_image | varchar(255) | 默认 '' | 好友的头像存储路径 |
| deleted | tinyint |  | 删除标识，0 表示正常，1 表示已删除 |
| created\_time | datetime | 默认 CURRENT\_TIMESTAMP | 好友关系创建的时间 |

1. 私聊消息表（`private\_message`）：存储用户之间的私聊消息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 消息的唯一标识 |
| send\_id | bigint | 非空，索引 | 发送消息的用户 ID |
| recv\_id | bigint | 非空，索引 | 接收消息的用户 ID |
| content | text |  | 消息的内容 |
| type | tinyint(1) | 非空 | 消息类型，0 表示文字，1 表示图片，2 表示文件，3 表示语音，4 表示视频，21 表示提示 |
| status | tinyint(1) | 非空 | 消息状态，0 表示未读，1 表示已读，2 表示撤回，3 表示已读 |
| send\_time | datetime | 默认 CURRENT\_TIMESTAMP | 消息发送的时间 |

1. 群表（`group`）：管理系统中的群信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 群的唯一标识 |
| name | varchar(255) | 非空 | 群的名称 |
| owner\_id | bigint | 非空 | 群主的用户 ID |
| head\_image | varchar(255) | 默认 '' | 群的头像存储路径 |
| head\_image\_thumb | varchar(255) | 默认 '' | 群头像缩略图的存储路径 |
| notice | varchar(1024) | 默认 '' | 群公告的内容 |
| is\_banned | tinyint(1) | 默认 0 | 群是否被封禁，0 表示否，1 表示是 |

1. 群成员表（`group\_member`）：记录群的成员信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 记录的唯一标识 |
| group\_id | bigint | 非空，索引 | 群的 ID |
| user\_id | bigint | 非空，索引 | 用户的 ID |
| user\_nick\_name | varchar(255) | 默认 '' | 用户在群里的昵称 |
| remark\_nick\_name | varchar(255) | 默认 '' | 用户在群里的显示昵称备注 |
| head\_image | varchar(255) | 默认 '' | 用户的头像存储路径 |
| remark\_group\_name | varchar(255) | 默认 '' | 用户对群的显示群名备注 |

1. 群消息表（`group\_message`）：存储群内的消息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 消息的唯一标识 |
| group\_id | bigint | 非空，索引 | 群的 ID |
| send\_id | bigint | 非空 | 发送消息的用户 ID |
| send\_nick\_name | varchar(255) | 默认 '' | 发送消息用户的昵称 |
| recv\_ids | varchar(1024) | 默认 '' | 接收消息的用户 ID 列表，逗号分隔，为空表示发给所有成员 |
| content | text |  | 消息的内容 |
| at\_user\_ids | varchar(1024) |  | 被 @ 的用户 ID 列表，逗号分隔 |

1. 敏感词表（`sensitive\_word`）：管理系统中的敏感词。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **约束** | **描述** |
| id | bigint | 主键，自增 | 敏感词记录的唯一标识 |
| content | varchar(64) | 非空 | 敏感词的内容 |
| enabled | tinyint | 默认 0 | 敏感词是否启用，0 表示未启用，1 表示启用 |
| creator | bigint | 可空 | 敏感词的创建者 ID |
| create\_time | datetime | 默认 CURRENT\_TIMESTAMP | 敏感词创建的时间 |

这些表结构的设计，能够满足飞克尔聊天系统中用户管理、好友关系管理、消息通信、群管理等核心功能的需求，同时通过合理的索引设计，提高了数据查询的效率。

### 4.3 系统详细设计

#### 4.3.1 用户管理功能详细设计

用户管理功能主要包括用户注册、登录、信息修改、密码找回等功能。具体的详细设计如下：

1. 用户注册：用户输入用户名、密码等信息，系统验证信息的合法性和唯一性。如果信息合法且唯一，系统将用户信息插入到用户表中，并返回注册成功的提示信息。
2. 用户登录：用户输入用户名和密码，系统验证用户名和密码的正确性。如果验证通过，系统生成一个唯一的令牌（token），并将令牌返回给客户端。客户端在后续的请求中需要携带该令牌，系统通过令牌验证用户的身份。
3. 用户信息修改：用户登录后，可以修改个人信息，如头像、昵称、签名等。系统将用户修改后的信息更新到用户表中，并返回修改成功的提示信息。

#### **4.3.2 好友管理功能详细设计**

好友管理功能主要包括好友添加、删除、查找、备注等功能。具体的详细设计如下：

1. 好友添加：用户输入好友的用户名、手机号码或邮箱等信息，系统查找符合条件的用户。如果找到用户，系统提示用户是否添加该用户为好友。用户确认后，系统将好友关系插入到好友表中，并返回添加成功的提示信息。
2. 好友删除：用户选择要删除的好友，系统将好友关系从好友表中删除，并返回删除成功的提示信息。
3. 好友查找：用户输入好友的用户名、手机号码或邮箱等信息，系统查找符合条件的用户，并返回查找结果。
4. 好友备注：用户选择要设置备注的好友，输入备注名，系统将备注名更新到好友表中，并返回设置成功的提示信息。

#### 4.3.3 群组管理功能详细设计

群组管理功能主要包括群组创建、加入、退出、管理等功能。具体的详细设计如下：

1. 群组创建：用户输入群组名称、群组简介等信息，选择群组的成员，系统将群组信息插入到群组表中，并将成员关系插入到群组用户表中。系统返回群组创建成功的提示信息。
2. 群组退出：用户选择要退出的群组，系统将用户关系从群组用户表中删除，并返回退出成功的提示信息。
3. 群组管理：用户可以管理群组的成员、权限、公告等信息。系统根据用户的操作更新群组表和群组用户表中的信息，并返回操作成功的提示信息。

#### 4.3.4 聊天功能详细设计

聊天功能主要包括一对一聊天、群组聊天、消息发送、接收、查看、删除等功能。具体的详细设计如下：

1. 一对一聊天：用户选择要聊天的好友，输入聊天内容，系统将消息插入到消息表中，并将消息推送给对方。对方可以及时查看和回复消息。
2. 群组聊天：用户选择要聊天的群组，输入聊天内容，系统将消息插入到消息表中，并将消息推送给群组内的所有成员。成员可以及时查看和回复消息。
3. 消息发送：用户输入聊天内容，选择消息类型（文字、图片、语音、视频等），系统将消息插入到消息表中，并将消息推送给接收方。
4. 消息接收：系统实时监听消息的推送，当接收到新消息时，将消息推送给用户。用户可以及时查看和回复消息。
5. 消息查看：用户可以查看历史聊天记录，系统从消息表中查询指定用户或群组的聊天记录，并返回给用户。
6. 消息删除：用户可以删除不需要的聊天记录，系统从消息表中删除指定的消息记录，并返回删除成功的提示信息。

## 第五章 系统实现

### 5.1 系统实现环境

系统实现环境主要包括开发环境和运行环境。开发环境主要包括开发工具、编程语言、框架等。运行环境主要包括服务器、数据库、操作系统等。具体的实现环境如下：

1. 开发环境：

* 开发工具：IntelliJ IDEA、Visual Studio Code 等。
* 编程语言：Java、JavaScript 等。
* 框架：SpringBoot、VUE 等。
* 数据库：MySQL。

1. 运行环境：

* 服务器：Tomcat、Nginx 等。
* 数据库：MySQL。
* 操作系统：Linux、Windows 等。

### 5.2 用户管理功能实现

用户管理功能的实现主要包括用户注册、登录、信息修改、密码找回等功能。具体的实现步骤如下：

1. 用户注册：使用 SpringBoot 框架搭建后端服务，接收用户的注册信息，验证信息的合法性和唯一性。使用 MySQL 数据库存储用户信息。前端使用 VUE 框架实现注册页面，与后端服务进行交互。
2. 用户登录：后端服务接收用户的登录信息，验证用户名和密码的正确性。如果验证通过，生成一个唯一的令牌（token），并将令牌返回给前端。前端使用 VUE 框架实现登录页面，与后端服务进行交互，将令牌存储在本地。
3. 用户信息修改：后端服务接收用户修改后的信息，更新用户表中的数据。前端使用 VUE 框架实现信息修改页面，与后端服务进行交互。

### 5.3好友管理功能实现

好友管理功能的实现主要包括好友添加、删除、查找、备注等功能。具体的实现步骤如下：

1. 好友添加：后端服务接收用户输入的好友信息，查找符合条件的用户。如果找到用户，提示用户是否添加该用户为好友。用户确认后，将好友关系插入到好友表中。前端使用 VUE 框架实现好友添加页面，与后端服务进行交互。
2. 好友删除：后端服务接收用户选择的好友信息，将好友关系从好友表中删除。前端使用 VUE 框架实现好友删除页面，与后端服务进行交互。
3. 好友查找：后端服务接收用户输入的好友信息，查找符合条件的用户，并返回查找结果。前端使用 VUE 框架实现好友查找页面，与后端服务进行交互。
4. 好友备注：后端服务接收用户输入的备注信息，更新好友表中的备注字段。前端使用 VUE 框架实现好友备注页面，与后端服务进行交互。

### 群组管理功能实现

群组管理功能的实现主要包括群组创建、加入、退出、管理等功能。具体的实现步骤如下：

1. 群组创建：后端服务接收用户输入的群组信息和成员信息，将群组信息插入到群组表中，并将成员关系插入到群组用户表中。前端使用 VUE 框架实现群组创建页面，与后端服务进行交互。
2. 群组加入：后端服务接收用户输入的群组信息，查找符合条件的群组。如果找到群组，提示用户是否加入该群组。用户确认后，将用户关系插入到群组用户表中。前端使用 VUE 框架实现群组加入页面，与后端服务进行交互。
3. 群组退出：后端服务接收用户选择的群组信息，将用户关系从群组用户表中删除。前端使用 VUE 框架实现群组退出页面，与后端服务进行交互。
4. 群组管理：后端服务接收用户的管理操作，更新群组表和群组用户表中的信息。前端使用 VUE 框架实现群组管理页面，与后端服务进行交互。

### 聊天功能实现

聊天功能的实现主要包括一对一聊天、群组聊天、消息发送、接收、查看、删除等功能。具体的实现步骤如下：

1. 一对一聊天：后端服务接收用户的聊天信息，将消息插入到消息表中，并将消息推送给对方。对方可以及时查看和回复消息。前端使用 VUE 框架实现一对一聊天页面，与后端服务进行交互。
2. 群组聊天：后端服务接收用户的聊天信息，将消息插入到消息表中，并将消息推送给群组内的所有成员。成员可以及时查看和回复消息。前端使用 VUE 框架实现群组聊天页面，与后端服务进行交互。
3. 消息发送：后端服务接收用户的消息信息，将消息插入到消息表中，并将消息推送给接收方。前端使用 VUE 框架实现消息发送功能，与后端服务进行交互。
4. 消息接收：后端服务实时监听消息的推送，当接收到新消息时，将消息推送给用户。前端使用 VUE 框架实现消息接收功能，与后端服务进行交互。
5. 消息查看：后端服务从消息表中查询指定用户或群组的聊天记录，并返回给前端。前端使用 VUE 框架实现消息查看页面，与后端服务进行交互。
6. 消息删除：后端服务从消息表中删除指定的消息记录，并返回删除成功的提示信息。前端使用 VUE 框架实现消息删除功能，与后端服务进行交互。

## 第六章 系统测试

### 6.1 测试目的

系统测试的目的是验证飞克尔聊天系统是否满足用户的需求和设计要求，发现系统中存在的缺陷和问题，确保系统的质量和可靠性。具体的测试目的包括：

1. 验证系统的功能是否正确实现，如用户管理、好友管理、群组管理、聊天等功能。
2. 验证系统的性能是否满足要求，如响应时间、并发处理能力等。
3. 验证系统的安全性是否可靠，如数据加密、权限管理等。
4. 验证系统的兼容性是否良好，如在不同的浏览器、操作系统和设备上的运行情况。

### 6.2系统测试结果

经过全面的系统测试，飞克尔聊天系统的各项功能均能正常实现，性能和安全性也满足设计要求。在测试过程中，发现了一些小的问题，如界面显示不美观、部分功能响应时间较长等。开发人员对这些问题进行了及时的修复，经过回归测试，问题已经解决。总体来说，飞克尔聊天系统的质量和可靠性较高，可以投入使用。

## 第七章 总结

### 7.1 研究工作总结

通过本次实训，我们完成了飞克尔聊天系统的开发和测试工作。在开发过程中，我们采用了先进的技术和框架，如 SpringBoot、VUE、MySQL 等，提高了开发效率和系统的性能。同时，我们也注重系统的安全性和可靠性，采用了安全可靠的加密算法和权限管理机制，确保用户的信息和聊天内容的安全。在测试过程中，我们使用了多种测试工具和方法，对系统的功能、性能、安全性和兼容性进行了全面的测试，发现并解决了系统中存在的问题，确保了系统的质量和可靠性。