

HiTutor 好会帮 —— 家教信息对接平台

一、软件概述

1.1 软件简介

HiTutor 好会帮是一个纯公益的家教信息对接共享平台，致力于连接学生与专业家教老师。平台仅提供家教信息发布、搜索、对接、沟通等中介信息服务，不提供任何家教教学，不参与任何教学辅导、培训等相关活动。平台提供移动端客户端和后端 API 接口，通过信息匹配、在线沟通、预约对接、评价反馈等全流程数字化信息服务，提升家教信息对接的透明度和信息质量，降低双方的沟通和匹配成本。

本软件采用前后端分离架构，前端使用 Flutter 框架开发移动端应用，支持 iOS 和 Android 双平台，后端使用 Spring Boot 框架提供 RESTful API 接口服务。通过现代化的技术架构和人性化的交互设计，为用户提供便捷、高效、安全的家教信息对接服务。平台的核心价值在于打破信息壁垒，让家教老师和学生家长能够直接对接，减少中间环节，提高匹配效率。

平台的设计理念是简单、高效、安全。简单体现在用户界面简洁明了，操作流程清晰易懂，用户无需复杂的学习成本即可快速上手。高效体现在信息匹配精准，搜索功能强大，用户可以快速找到合适的家教或需求。安全体现在数据传输加密，身份认证严格，用户信息安全有保障。

平台的技术架构采用微服务思想，将系统划分为多个独立的服务模块，每个模块负责特定的功能。这种架构设计使得系统具有良好的可扩展性和可维护性，可以方便地添加新功能或修改现有功能。平台还采用缓存机制，提高系统响应速度，优化用户体验。

1.2 软件特点

平台具有以下显著特点：第一，纯公益性质，不收取任何中介费用，所有功能免费开放给用户使用。平台致力于为家教老师和学生家长提供一个免费、公平、透明的信息对接平台，打破传统家教中介的高收费模式，让更多家庭能够负担得起优质的家教服务。

第二，信息透明公开，所有家教信息和学生需求都经过审核认证，确保信息真实可靠。平台建立了完善的信息审核机制，家教老师需要上传教师资格证、学历证书等相关证明材料，经过人工审核后才能发布家教信息。学生需求也需要经过基本的格式验证，确保信息的完整性和有效性。

第三，地理位置智能匹配，基于高德地图提供附近家教和需求的精准搜索。平台使用高德地图 SDK，支持 GPS 定位和网络定位，可以准确获取用户当前位置。用户

可以在地图上查看附近的家教或需求，系统会根据用户当前位置计算距离，并按距离远近排序显示结果，方便用户选择就近的家教或需求。

第四，实时在线沟通，支持文字和图片消息的即时聊天功能。平台使用 WebSocket 技术实现实时通信，消息可以实时推送给接收方，接收方可以立即收到消息通知。用户可以随时与家教老师或学生家长进行沟通，了解对方的情况，确认预约细节，提高对接效率。

第五，完善的评价体系，对接完成后可以进行星级评分和文字评价，帮助其他用户做出更好的选择。平台建立了科学的评价机制，用户可以在完成对接后对对方进行评价，评价内容包括星级评分和文字评价。评价记录会公开显示，其他用户可以查看评价记录，了解家教老师或学生的服务质量，为选择提供参考。

第六，智能推荐算法，根据用户的历史行为和偏好，推荐合适的家教或需求。平台使用机器学习算法，分析用户的浏览记录、搜索记录、收藏记录等数据，推荐符合用户偏好的家教或需求，提高匹配精度，减少用户搜索时间。

第七，安全可靠的支付系统，支持多种支付方式，保障交易安全。平台集成主流支付平台，支持微信支付、支付宝等多种支付方式，用户可以选择自己习惯的支付方式。支付过程采用 SSL 加密，确保交易信息安全，防止数据泄露。

第八，灵活的预约管理，支持预约时间确认、修改、取消等操作。用户可以在预约管理页面查看所有预约信息，包括预约的家教或学生、预约时间、预约状态等。用户可以对预约进行确认或取消操作，系统会及时通知对方，确保双方都能及时了解预约状态的变化。

1.3 适用人群

本软件主要面向两类用户群体：一类是寻找家教的学生和家长，他们可以通过平台发布学生需求，搜索合适的家教老师，查看家教信息和评价记录，与家教老师在线沟通，完成预约对接；另一类是提供家教服务的专业家教老师，他们可以通过平台发布家教信息，展示自己的教学经验和资质认证，接收学生的需求申请，与学生在线沟通，完成预约对接。

二、系统架构

2.1 技术架构

本软件采用前后端分离的架构设计，前端使用 Flutter 3.2.6 框架开发移动端应用，编程语言为 Dart 3.2.6，使用 Provider 6.1.0 进行状态管理，HTTP 1.1.0 进行网络请求，Shared Preferences 2.2.3 进行本地数据存储，AMap Flutter Map 3.0.0 提供地图服务，Geolocator 11.1.0 提供地理位置服务，Permission Handler 11.0.0 处理权限请求。

后端使用 Spring Boot 3.1.0 框架开发 RESTful API 接口，编程语言为 Java 17，使用 MyBatis Plus 3.5.4.1 进行数据库操作，数据库采用 MySQL 8.0.33，使用 JWT 0.11.5 进行身份认证，使用 WebSocket 实现实时通信。前后端通过 HTTP 协议进行数据交换，数据格式采用 JSON。

2.1.1 前端技术栈详解

Flutter 3.2.6: Flutter 是 Google 推出的开源 UI 工具包，用于构建跨平台应用。采用 Dart 语言开发，具有热重载、自绘渲染引擎等特点，可在 iOS、Android 等平台上构建高质量的原生界面。

Dart 3.2.6: Dart 是一种由 Google 开发的面向对象编程语言，具有强类型、异步编程、垃圾回收等特性，适合移动应用开发。Dart 语言支持 JIT（即时编译）和 AOT（提前编译）两种编译模式，JIT 模式用于开发阶段的热重载，AOT 模式用于生产环境的性能优化。

Provider 6.1.0: Provider 是 Flutter 中常用的状态管理库，基于 InheritedWidget 实现，提供了简洁的状态管理方案。支持依赖注入、状态监听、状态共享等功能，适合中大型应用的状态管理。

HTTP 1.1.0: HTTP 是 Flutter 中用于网络请求的库，提供了发送 HTTP 请求的功能。支持 GET、POST、PUT、DELETE 等 HTTP 方法，支持设置请求头、请求体、查询参数等，适合与后端 API 进行通信。

Shared Preferences 2.2.3: Shared Preferences 是 Flutter 中用于本地数据存储的库，提供了键值对存储功能。适合存储用户配置、登录状态等轻量级数据，数据存储在设备的本地文件系统中。

AMap Flutter Map 3.0.0: AMap Flutter Map 是高德地图在 Flutter 平台的插件，提供了地图显示、定位、搜索等功能。支持地图缩放、拖动、标记等操作，适合需要地图服务的应用。

Geolocator 11.1.0: Geolocator 是 Flutter 中用于获取地理位置的库，提供了获取设备当前位置、监听位置变化等功能。支持 GPS、网络等多种定位方式，适合需要地理位置服务的应用。

Permission Handler 11.0.0: Permission Handler 是 Flutter 中用于处理权限请求的库，提供了请求各种系统权限的功能。支持位置权限、相机权限、存储权限等，适合需要访问系统资源的应用。

2.1.2 后端技术栈详解

Spring Boot 3.1.0: Spring Boot 是基于 Spring 框架的快速开发脚手架，提供了自动配置、内嵌服务器等功能，简化了 Spring 应用的开发和部署。支持 RESTful API 开发、数据访问、安全认证等功能，适合构建企业级应用。

Java 17: Java 是一种广泛使用的编程语言，具有跨平台、面向对象、安全性高等特点。Java 17 是 Java 的长期支持版本，提供了许多新特性，如密封类、模式匹配等，适合企业级应用开发。

MyBatis Plus 3.5.4.1: MyBatis Plus 是基于 MyBatis 的增强工具，提供了代码生成、分页插件、逻辑删除等功能，简化了 MyBatis 的使用。支持 Lambda 表达式、链式调用等特性，提高了开发效率。

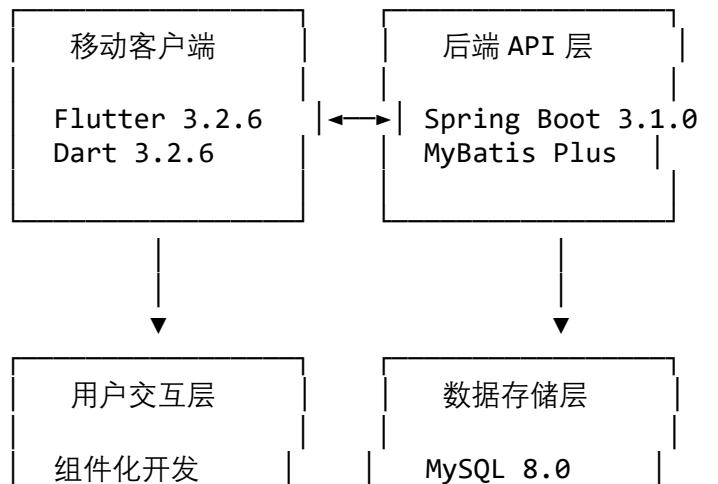
MySQL 8.0.33: MySQL 是一种常用的关系型数据库管理系统，具有可靠性高、性能好、易于使用等特点。MySQL 8.0 提供了许多新特性，如窗口函数、CTEs、JSON 支持等，适合存储结构化数据。

JWT 0.11.5: JWT (JSON Web Token) 是一种基于 JSON 的开放标准，用于在网络应用间传递声明。JWT 由三部分组成：头部、载荷和签名，可用于身份认证、信息交换等场景。

WebSocket: WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议，可实现服务器主动向客户端推送消息。适合实时通信场景，如在线聊天、实时通知等。

2.2 系统架构图

系统整体架构分为三层：表示层、业务逻辑层和数据持久层。表示层包括 Flutter 移动客户端，负责用户交互和界面展示；业务逻辑层包括 Spring Boot 后端 API，负责业务逻辑处理和数据转换；数据持久层包括 MySQL 数据库，负责数据存储和管理。



2.2.1 模块划分

前端模块划分：

1. **用户认证模块**：负责用户注册、登录、身份验证等功能，包括登录注册页面、验证码发送等。
2. **家教信息模块**：负责家教信息的发布、编辑、删除、查看等功能，包括发布家教信息页面、家教信息详情页面等。
3. **学生需求模块**：负责学生需求的发布、编辑、删除、查看等功能，包括发布学生需求页面、学生需求详情页面等。
4. **搜索匹配模块**：负责家教信息和学生需求的搜索、筛选、排序等功能，包括搜索页面、筛选条件设置等。
5. **在线沟通模块**：负责用户间的实时聊天功能，包括聊天会话列表页面、聊天详情页面等。
6. **预约管理模块**：负责预约信息的查看、确认、取消等功能，包括预约列表页面、预约详情页面等。
7. **评价反馈模块**：负责用户评价的提交、查看等功能，包括评价页面、评价列表页面等。
8. **个人中心模块**：负责用户个人信息的管理、功能入口的提供等功能，包括个人中心页面、设置页面等。

后端模块划分：

1. **用户服务模块**：负责用户相关的业务逻辑，包括用户注册、登录、个人信息管理等。
2. **家教服务模块**：负责家教信息相关的业务逻辑，包括家教信息发布、编辑、删除、查询等。
3. **学生服务模块**：负责学生需求相关的业务逻辑，包括学生需求发布、编辑、删除、查询等。
4. **搜索服务模块**：负责搜索相关的业务逻辑，包括信息检索、筛选、排序等。
5. **聊天服务模块**：负责实时聊天相关的业务逻辑，包括消息发送、接收、存储等。

6. **预约服务模块**: 负责预约相关的业务逻辑, 包括预约申请、确认、取消等。
7. **评价服务模块**: 负责评价相关的业务逻辑, 包括评价提交、查看、统计等。
8. **系统服务模块**: 负责系统相关的业务逻辑, 包括配置管理、日志记录等。

2.3 数据流程

用户通过移动客户端发起请求, 请求通过 HTTP 协议发送到后端 API, 后端 API 接收请求后进行业务逻辑处理, 如果需要访问数据库, 则通过 MyBatis Plus 进行数据库操作, 处理完成后将结果以 JSON 格式返回给前端, 前端接收响应数据后更新界面显示。对于实时聊天功能, 使用 WebSocket 建立长连接, 实现消息的实时推送和接收。

2.3.1 数据流程详细说明

1. 用户认证流程:

1. 用户在移动客户端输入手机号和密码, 点击登录按钮。
2. 客户端将登录请求通过 HTTP POST 请求发送到后端 API 的 /api/auth/login 接口。
3. 后端 API 接收请求后, 验证手机号和密码的正确性。
4. 验证通过后, 生成 JWT 令牌并返回给客户端。
5. 客户端存储 JWT 令牌, 用于后续请求的身份认证。

2. 信息发布流程:

1. 用户在移动客户端填写家教信息或学生需求, 点击提交按钮。
2. 客户端将发布请求通过 HTTP POST 请求发送到后端 API 的相应接口。
3. 后端 API 接收请求后, 验证用户身份和信息的有效性。
4. 验证通过后, 将信息存储到数据库中。
5. 后端 API 返回发布成功的响应给客户端。
6. 客户端显示发布成功的提示信息。

3. 搜索匹配流程:

1. 用户在移动客户端输入搜索关键词或设置筛选条件, 点击搜索按钮。

2. 客户端将搜索请求通过 HTTP GET 请求发送到后端 API 的相应接口。
3. 后端 API 接收请求后，根据搜索条件查询数据库。
4. 后端 API 对查询结果进行排序和分页处理。
5. 后端 API 返回搜索结果给客户端。
6. 客户端显示搜索结果列表。

4. 在线沟通流程：

1. 用户在移动客户端打开聊天详情页面，与对方建立 WebSocket 连接。
2. 用户输入消息，点击发送按钮。
3. 客户端将消息通过 WebSocket 连接发送到后端服务器。
4. 后端服务器接收消息后，存储消息到数据库中。
5. 后端服务器通过 WebSocket 连接将消息推送给接收方客户端。
6. 接收方客户端收到消息后，更新聊天界面显示。

5. 预约管理流程：

1. 用户在移动客户端查看家教信息或学生需求，点击报名按钮。
2. 客户端将报名请求通过 HTTP POST 请求发送到后端 API 的相应接口。
3. 后端 API 接收请求后，验证用户身份和报名信息的有效性。
4. 验证通过后，创建报名记录并存储到数据库中。
5. 后端 API 返回报名成功的响应给客户端。
6. 对方用户收到报名通知，可以查看报名详情并进行处理。

6. 评价反馈流程：

1. 用户在移动客户端完成对接后，进入评价页面。
2. 用户选择星级评分，输入文字评价，点击提交按钮。
3. 客户端将评价请求通过 HTTP POST 请求发送到后端 API 的相应接口。
4. 后端 API 接收请求后，验证用户身份和评价信息的有效性。
5. 验证通过后，存储评价信息到数据库中。

6. 后端 API 返回评价成功的响应给客户端。

7. 客户端显示评价成功的提示信息。

三、主要功能

3.1 用户认证模块

用户认证模块是系统的基础功能模块，提供用户注册、登录、身份验证等功能。用户可以通过手机号进行注册，支持密码登录和短信验证码登录两种方式。密码登录需要用户输入手机号和密码，系统验证通过后生成 JWT 令牌并返回给客户端，客户端保存令牌用于后续请求的身份认证。短信验证码登录需要用户输入手机号和验证码，系统验证验证码的有效性后生成 JWT 令牌并返回给客户端。

系统还提供实名认证和家教教师资质认证功能。实名认证需要用户上传身份证照片，系统进行人工审核，审核通过后用户的实名状态更新为已认证。家教教师资质认证需要家教老师上传教师资格证、学历证书等相关证明材料，系统进行人工审核，审核通过后用户的家教教师资质状态更新为已认证，可以在平台发布家教信息。

3.1.1 功能详细说明

1. 用户注册

- 功能描述：**允许新用户通过手机号注册平台账号。
- 流程说明：**用户输入手机号，获取短信验证码，设置登录密码，点击注册按钮完成注册。
- 实现细节：**
 - 手机号验证：使用正则表达式验证手机号格式是否正确。
 - 验证码发送：调用短信服务 API 发送验证码，验证码有效期为 5 分钟。
 - 密码设置：密码长度至少 8 位，包含字母和数字。
 - 用户创建：注册成功后，系统自动创建用户账号，默认角色为普通用户。

2. 用户登录

- 功能描述：**允许已注册用户登录平台。
- 流程说明：**用户输入手机号和密码，或手机号和验证码，点击登录按钮完成登录。

- **实现细节：**
 - 密码登录：验证手机号和密码是否匹配，匹配成功后生成 JWT 令牌。
 - 验证码登录：验证手机号和验证码是否匹配，匹配成功后生成 JWT 令牌。
 - 令牌生成：JWT 令牌包含用户 ID、角色信息，有效期为 7 天。
 - 令牌存储：客户端使用 Shared Preferences 存储令牌，用于后续请求的身份认证。

3. 实名认证

- **功能描述：**验证用户的真实身份信息。
- **流程说明：**用户上传身份证正面和反面照片，填写姓名和身份证号，点击提交按钮完成认证申请。
- **实现细节：**
 - 照片上传：支持 JPG、PNG 格式，单张照片大小不超过 5MB。
 - 信息验证：验证姓名和身份证号是否匹配。
 - 人工审核：后台管理员人工审核认证材料，审核通过后更新用户实名状态。
 - 认证状态：认证状态包括未认证、审核中、已认证、认证失败。

4. 家教教师资质认证

- **功能描述：**验证家教老师的教学资质。
- **流程说明：**家教老师上传教师资格证、学历证书等证明材料，填写教学经验和专长，点击提交按钮完成认证申请。
- **实现细节：**
 - 材料上传：支持 JPG、PNG、PDF 格式，单个文件大小不超过 10MB。
 - 信息填写：包括毕业院校、专业、教学年限、擅长科目等。
 - 人工审核：后台管理员人工审核认证材料，审核通过后更新用户教师资质状态。
 - 认证状态：认证状态包括未认证、审核中、已认证、认证失败。

3.1.2 技术实现

前端实现：

- **登录注册页面**：使用 Flutter 构建，包含手机号输入框、密码输入框、验证码输入框、获取验证码按钮、登录/注册按钮等。
- **实名认证页面**：使用 Flutter 构建，包含身份证照片上传组件、姓名输入框、身份证号输入框、提交按钮等。
- **教师资质认证页面**：使用 Flutter 构建，包含证明材料上传组件、教学经验输入框、专长输入框、提交按钮等。
- **状态管理**：使用 Provider 管理用户认证状态，包括登录状态、认证状态等。
- **网络请求**：使用 HTTP 库发送注册、登录、认证申请等请求。

后端实现：

- **用户控制器**：处理用户注册、登录、个人信息管理等请求。
- **认证控制器**：处理实名认证、教师资质认证等请求。
- **用户服务**：实现用户相关的业务逻辑，包括用户创建、密码验证、信息更新等。
- **认证服务**：实现认证相关的业务逻辑，包括验证码发送、认证材料审核等。
- **JWT 工具类**：实现 JWT 令牌的生成和验证。
- **短信服务**：调用第三方短信 API 发送验证码。

3.2 家教信息发布模块

家教信息发布模块允许家教老师发布自己的家教信息，包括授课科目、时薪价格、授课地点、可授课时间段等信息。家教老师需要先完成家教教师资质认证才能发布家教信息。发布家教信息时，系统会对信息进行格式验证，确保信息的完整性和有效性。家教信息发布后，其他用户可以查看和搜索这些信息。

家教老师可以对自己发布的信息进行编辑和删除操作。编辑操作可以修改家教信息的各项内容，修改后的信息需要重新提交审核，审核通过后更新显示。删除操作会将家教信息的状态设置为已删除，其他用户无法再查看和搜索到该信息。家教老师还可以查看自己发布的信息列表，了解信息的浏览量、报名人数等统计数据。

3.2.1 功能详细说明

1. 家教信息发布

- **功能描述：**允许家教老师发布家教信息。
- **流程说明：**家教老师填写授课科目、时薪价格、授课地点、可授课时间段等信息，上传个人照片和教学成果证明，点击发布按钮完成发布。
- **实现细节：**
 - 信息填写：包括授课科目、时薪范围、授课地点、可授课时间、教学经验、专长描述等。
 - 照片上传：支持上传个人照片和教学成果证明，单张照片大小不超过5MB。
 - 信息验证：系统验证信息的完整性和有效性，确保必填项都已填写。
 - 审核机制：发布的信息需要经过后台审核，审核通过后才能在平台显示。

2. 家教信息编辑

- **功能描述：**允许家教老师编辑已发布的家教信息。
- **流程说明：**家教老师进入我的发布页面，选择要编辑的家教信息，修改相关内容，点击保存按钮完成编辑。
- **实现细节：**
 - 信息修改：可以修改家教信息的各项内容，包括授课科目、时薪价格、授课地点等。
 - 重新审核：修改后的信息需要重新提交审核，审核通过后更新显示。
 - 历史记录：系统保存家教信息的修改历史，方便追溯。

3. 家教信息删除

- **功能描述：**允许家教老师删除已发布的家教信息。
- **流程说明：**家教老师进入我的发布页面，选择要删除的家教信息，点击删除按钮完成删除。
- **实现细节：**

- 软删除：系统采用软删除机制，将家教信息的状态设置为已删除，其他用户无法查看。
- 权限验证：只有家教信息的发布者才能删除该信息。
- 删除确认：删除操作需要用户确认，防止误操作。

4. 家教信息管理

- **功能描述：**允许家教老师管理自己发布的家教信息。
- **流程说明：**家教老师进入我的发布页面，查看已发布的家教信息列表，了解信息的浏览量、报名人数等统计数据。
- **实现细节：**
 - 信息列表：显示家教信息的标题、发布时间、浏览量、报名人数等。
 - 状态筛选：支持按信息状态（审核中、已发布、已删除）筛选。
 - 详情查看：点击信息可以查看详细内容和统计数据。

3.2.2 技术实现

前端实现：

- **发布家教信息页面：**使用 Flutter 构建，包含科目选择器、价格输入框、地点选择器、时间选择器、照片上传组件、提交按钮等。
- **编辑家教信息页面：**使用 Flutter 构建，包含与发布页面相同的表单组件，预填已有信息。
- **我的发布页面：**使用 Flutter 构建，包含家教信息列表，支持下拉刷新和上拉加载更多。
- **状态管理：**使用 Provider 管理家教信息发布状态和列表数据。
- **网络请求：**使用 HTTP 库发送发布、编辑、删除、查询等请求。

后端实现：

- **家教控制器：**处理家教信息发布、编辑、删除、查询等请求。
- **家教服务：**实现家教信息相关的业务逻辑，包括信息创建、修改、删除、查询等。

- **审核服务**: 实现家教信息审核相关的业务逻辑，包括审核状态更新、审核记录管理等。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作家教信息表。

3.3 学生需求发布模块

学生需求发布模块允许学生和家长发布学生需求，包括辅导科目、学生年级、时薪预算、上课地点等信息。用户注册登录后即可发布学生需求，无需额外的认证。发布学生需求时，系统会对信息进行格式验证，确保信息的完整性和有效性。学生需求发布后，家教老师可以查看和搜索这些需求。

用户可以对自己发布的需求进行编辑和删除操作。编辑操作可以修改学生需求的各项内容，修改后的请求会立即更新显示。删除操作会将学生需求的状态设置为已删除，其他用户无法再查看和搜索到该需求。用户还可以查看自己发布的需求列表，了解需求的浏览量、报名人数等统计数据。

3.3.1 功能详细说明

1. 学生需求发布

- **功能描述**: 允许学生和家长发布学生需求。
- **流程说明**: 用户填写辅导科目、学生年级、时薪预算、上课地点等信息，点击发布按钮完成发布。
- **实现细节**:
 - 信息填写: 包括辅导科目、学生年级、时薪预算、上课地点、学习情况描述等。
 - 信息验证: 系统验证信息的完整性和有效性，确保必填项都已填写。
 - 审核机制: 发布的需求需要经过后台审核，审核通过后才能在平台显示。

2. 学生需求编辑

- **功能描述**: 允许用户编辑已发布的学需求。
- **流程说明**: 用户进入我的发布页面，选择要编辑的学生需求，修改相关内容，点击保存按钮完成编辑。
- **实现细节**:

- 信息修改：可以修改学生需求的各项内容，包括辅导科目、时薪预算、上课地点等。
- 重新审核：修改后的需求需要重新提交审核，审核通过后更新显示。
- 历史记录：系统保存学生需求的修改历史，方便追溯。

3. 学生需求删除

- 功能描述：允许用户删除已发布的学生活需求。
- 流程说明：用户进入我的发布页面，选择要删除的学生活需求，点击删除按钮完成删除。
- 实现细节：
 - 软删除：系统采用软删除机制，将学生活需求的状态设置为已删除，其他用户无法查看。
 - 权限验证：只有学生活需求的发布者才能删除该需求。
 - 删除确认：删除操作需要用户确认，防止误操作。

4. 学生需求管理

- 功能描述：允许用户管理自己发布的学生活需求。
- 流程说明：用户进入我的发布页面，查看已发布的学生活需求列表，了解需求的浏览量、报名人数等统计数据。
- 实现细节：
 - 信息列表：显示学生活需求的标题、发布时间、浏览量、报名人数等。
 - 状态筛选：支持按需求状态（审核中、已发布、已删除）筛选。
 - 详情查看：点击需求可以查看详细内容和统计数据。

3.3.2 技术实现

前端实现：

- **发布学生活需求页面**：使用 Flutter 构建，包含科目选择器、年级选择器、预算输入框、地点选择器、提交按钮等。
- **编辑学生活需求页面**：使用 Flutter 构建，包含与发布页面相同的表单组件，预填已有信息。

- **我的发布页面**: 使用 Flutter 构建, 包含学生需求列表, 支持下拉刷新和上拉加载更多。
- **状态管理**: 使用 Provider 管理学生需求发布状态和列表数据。
- **网络请求**: 使用 HTTP 库发送发布、编辑、删除、查询等请求。

后端实现:

- **学生控制器**: 处理学生需求发布、编辑、删除、查询等请求。
- **学生服务**: 实现学生需求相关的业务逻辑, 包括需求创建、修改、删除、查询等。
- **审核服务**: 实现学生需求审核相关的业务逻辑, 包括审核状态更新、审核记录管理等。
- **数据访问**: 使用 MyBatis Plus 访问数据库, 操作学生需求表。



家教认证

基本信息



真实姓名



身份证号

学历信息



学历



毕业院校



专业

证书信息



教师资格证号

提交认证



家教认证



身份已认证

您的家教认证已审核通过

[查看认证信息](#)



好会帮认证证书



好会帮认证证书

已认证

真实姓名 : 李星辰

身份证号 : 310104199508154321

学历 : 硕士

毕业院校 : 华中师范大学

专业 : 数学与应用数学 (师范)

教师资格证号 : 20193100141002345

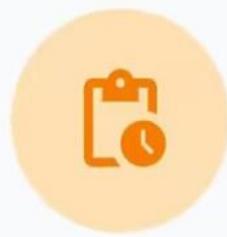
认证时间 : 2026-01-26 15:47:33



兹证明李老师已通过好会帮资格审核



家教认证



认证正在审核中

您的认证申请正在审核中，请耐心等待

3.2 家教信息发布模块

家教信息发布模块允许家教老师发布自己的家教信息，包括授课科目、时薪价格、授课地点、可授课时间段等信息。家教老师需要先完成家教教师资质认证才能发布家教信息。发布家教信息时，系统会对信息进行格式验证，确保信息的完整性和有效性。家教信息发布后，其他用户可以查看和搜索这些信息。

家教老师可以对自己发布的信息进行编辑和删除操作。编辑操作可以修改家教信息的各项内容，修改后的信息需要重新提交审核，审核通过后更新显示。删除操作会将家教信息的状态设置为已删除，其他用户无法再查看和搜索到该信息。家教老师还可以查看自己发布的信息列表，了解信息的浏览量、报名人数等统计数据。

3.3 学生需求发布模块

学生需求发布模块允许学生和家长发布学生需求，包括辅导科目、学生年级、时薪预算、上课地点等信息。用户注册登录后即可发布学生需求，无需额外的认证。发布学生需求时，系统会对信息进行格式验证，确保信息的完整性和有效性。学生需求发布后，家教老师可以查看和搜索这些需求。

用户可以对自己发布的需求进行编辑和删除操作。编辑操作可以修改学生需求的各项内容，修改后的请求会立即更新显示。删除操作会将学生需求的状态设置为已删除，其他用户无法再查看和搜索到该需求。用户还可以查看自己发布的需求列表，了解需求的浏览量、报名人数等统计数据。



需求详情



小明的需求

七年级

数学

基本信息



年级

七年级



科目

数学



授课地点

暂无地址信息

时薪范围

¥80.00-¥160.00/小时

需求描述

希望老师擅长讲解几何题型，能帮助孩子梳理理解题思路，每周课后可布置少量巩固练习

联系方式



申请后可查看联系方式

提交申请后，学生将收到通知并查看您的信息

3.4 智能搜索模块

智能搜索模块提供强大的搜索功能，帮助用户快速找到合适的家教或需求。用户可以按科目筛选，选择自己感兴趣的辅导科目，系统会返回符合条件的家教信息或学生需求。用户可以按年级筛选，选择学生的年级，系统会返回支持该年级辅导的家教信息。用户可以按时薪范围筛选，设置最低和最高时薪，系统会返回时薪在指定范围内的家教信息或学生需求。

系统还提供基于地理位置的附近搜索功能，用户可以查看附近的家教或需求，系统会根据用户的当前位置计算距离，并按距离远近排序显示结果。用户还可以使用关键词搜索，输入关键词后系统会在家教信息和学生需求的标题、描述等字段中进行模糊匹配，返回包含关键词的结果。搜索结果支持分页显示，用户可以上拉加载更多内容。

3.4.1 功能详细说明

1. 关键词搜索

- **功能描述：**根据用户输入的关键词搜索家教信息或学生需求。
- **流程说明：**用户在搜索框中输入关键词，点击搜索按钮，系统返回包含关键词的家教信息或学生需求列表。
- **实现细节：**
 - 关键词匹配：系统在家教信息和学生需求的标题、描述、科目等字段中进行模糊匹配。
 - 搜索结果排序：默认按相关度排序，相关度高的结果排在前面。
 - 分页显示：搜索结果支持分页，每页显示 20 条记录，用户可以上拉加载更多。

2. 科目筛选

- **功能描述：**根据用户选择的科目筛选家教信息或学生需求。
- **流程说明：**用户在筛选条件中选择辅导科目，系统返回符合条件的家教信息或学生需求列表。
- **实现细节：**
 - 科目分类：系统预设了语文、数学、英语、物理、化学、生物、历史、地理、政治等常见科目。

- 多选支持：用户可以选择多个科目进行筛选。
- 实时筛选：用户选择科目后，系统实时更新搜索结果。

3. 年级筛选

- **功能描述：**根据用户选择的年级筛选家教信息。
- **流程说明：**用户在筛选条件中选择学生年级，系统返回支持该年级辅导的家教信息列表。
- **实现细节：**
 - 年级分类：系统预设了小学一年级到高中三年级的各个年级。
 - 多选支持：用户可以选择多个年级进行筛选。
 - 实时筛选：用户选择年级后，系统实时更新搜索结果。

4. 时薪范围筛选

- **功能描述：**根据用户设置的时薪范围筛选家教信息或学生需求。
- **流程说明：**用户在筛选条件中设置最低和最高时薪，系统返回时薪在指定范围内的家教信息或学生需求列表。
- **实现细节：**
 - 时薪区间：系统预设了多个时薪区间，用户也可以自定义时薪范围。
 - 实时筛选：用户设置时薪范围后，系统实时更新搜索结果。

5. 地理位置搜索

- **功能描述：**根据用户的地理位置搜索附近的家教信息或学生需求。
- **流程说明：**用户开启位置权限，系统获取用户当前位置，显示附近的家教信息或学生需求列表。
- **实现细节：**
 - 位置获取：使用 Geolocator 插件获取用户当前位置的经纬度。
 - 距离计算：使用 Haversine 公式计算用户与家教/学生之间的距离。
 - 结果排序：按距离远近排序，距离近的结果排在前面。
 - 距离筛选：用户可以设置搜索半径，如 1 公里、3 公里、5 公里等。

6. 智能推荐

- **功能描述：**根据用户的历史行为和偏好，推荐合适的家教信息或学生需求。
- **流程说明：**用户登录后，系统分析用户的浏览记录、搜索记录、收藏记录等，在首页推荐符合用户偏好的家教信息或学生需求。
- **实现细节：**
 - 数据采集：系统采集用户的浏览、搜索、收藏、报名等行为数据。
 - 偏好分析：使用机器学习算法分析用户的偏好，如喜欢的科目、年级、时薪范围等。
 - 推荐算法：基于协同过滤和内容过滤的混合推荐算法，推荐符合用户偏好的家教信息或学生需求。
 - 实时更新：用户行为发生变化时，推荐结果实时更新。

3.4.2 技术实现

前端实现：

- **搜索页面：**使用 Flutter 构建，包含搜索框、筛选条件设置、搜索结果列表等。
- **筛选组件：**使用 Flutter 构建，包含科目选择器、年级选择器、时薪范围选择器、距离选择器等。
- **地图搜索页面：**使用 Flutter 和 AMap Flutter Map 构建，显示地图视图和附近的家教/学生标记。
- **状态管理：**使用 Provider 管理搜索条件和搜索结果状态。
- **网络请求：**使用 HTTP 库发送搜索请求，支持分页加载。

后端实现：

- **搜索控制器：**处理搜索请求，包括关键词搜索、筛选条件搜索、地理位置搜索等。
- **搜索服务：**实现搜索相关的业务逻辑，包括信息检索、筛选、排序等。
- **推荐服务：**实现智能推荐相关的业务逻辑，包括用户行为分析、偏好建模、推荐结果生成等。

- **数据访问**: 使用 MyBatis Plus 访问数据库，支持复杂查询和排序。
- **缓存机制**: 使用 Redis 缓存热门搜索结果，提高搜索响应速度。

系统使用 WebSocket 实现实时通信，当有新消息到达时，会实时推送给接收方，接收方可以立即收到消息通知。消息支持已读和未读状态管理，发送方可以查看消息是否已被对方阅读。图片消息需要先上传到服务器，服务器返回图片 URL 后发送给接收方，接收方点击图片可以查看大图。

3.5.1 功能详细说明

1. 聊天会话列表

- **功能描述**: 显示用户的所有聊天会话。
- **流程说明**: 用户进入消息页面，查看所有聊天会话列表，包括系统通知和用户聊天。
- **实现细节**:
 - 会话排序: 按最后一条消息的时间排序，最新的会话排在前面。
 - 未读提示: 未读消息的会话显示红点提示，显示未读消息数量。
 - 会话信息: 显示对方的头像、昵称、最后一条消息内容和时间。

2. 聊天详情

- **功能描述**: 显示与特定用户的聊天记录，支持发送和接收消息。
- **流程说明**: 用户点击会话进入聊天详情页面，查看历史消息记录，输入消息并发送。
- **实现细节**:
 - 消息列表: 显示历史消息记录，支持滚动加载更多历史消息。
 - 消息类型: 支持文字消息和图片消息。
 - 消息状态: 显示消息的发送状态，如发送中、已发送、已读。
 - 图片查看: 点击图片消息可以查看大图。

3. 实时消息推送

- **功能描述：**实时接收新消息通知。
- **流程说明：**当有新消息到达时，系统通过 WebSocket 连接将消息推送给接收方客户端。
- **实现细节：**
 - **WebSocket 连接：**客户端与服务器建立 WebSocket 长连接，保持实时通信。
 - **消息推送：**服务器将新消息通过 WebSocket 连接推送给接收方客户端。
 - **消息通知：**客户端收到新消息后，显示消息通知，包括消息内容和发送者信息。

4. 消息历史记录

- **功能描述：**查看历史消息记录。
- **流程说明：**用户在聊天详情页面向上滚动，加载并查看更早的消息记录。
- **实现细节：**
 - **分页加载：**消息历史记录支持分页加载，每次加载 20 条消息。
 - **消息存储：**消息存储在数据库中，支持长期保存。
 - **消息搜索：**支持在聊天历史中搜索特定关键词的消息。

3.5.2 技术实现

前端实现：

- **消息页面：**使用 Flutter 构建，包含聊天会话列表，支持下拉刷新和滑动删除。
- **聊天详情页面：**使用 Flutter 构建，包含消息列表、输入框、发送按钮等。
- **WebSocket 客户端：**使用 Flutter 的 `web_socket_channel` 库实现 WebSocket 连接，处理消息的发送和接收。
- **图片上传：**使用 HTTP 库实现图片上传功能，支持进度显示。
- **状态管理：**使用 Provider 管理聊天会话和消息状态。

后端实现：

- **WebSocket 服务**: 使用 Spring WebSocket 实现 WebSocket 服务器，处理客户端连接和消息推送。
- **消息控制器**: 处理消息相关的 HTTP 请求，如获取历史消息、标记消息已读等。
- **消息服务**: 实现消息相关的业务逻辑，包括消息存储、发送、推送等。
- **用户服务**: 提供用户信息查询功能，用于显示聊天对象的头像和昵称。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作消息表和会话表。

3.6 预约报名模块

预约报名模块允许用户对家教信息或学生需求发起报名申请。家教老师可以查看学生需求列表，对感兴趣的需求发起报名申请，填写自己的教学经验和优势等信息。学生可以查看家教信息列表，对感兴趣的家教发起报名申请，填写学生的学习情况和需求等信息。

报名申请提交后，对方会收到报名通知，可以查看报名申请的详细信息，包括申请人的基本信息、教学经验或学习情况等。对方可以接受或拒绝报名申请，接受后双方可以进行进一步的沟通和预约。系统会跟踪报名申请的状态，包括待处理、已接受、已拒绝等状态，用户可以查看自己的报名申请列表和收到的报名申请列表。

3.6.1 功能详细说明

1. 家教报名学生需求

- **功能描述**: 允许家教老师对学生需求发起报名申请。
- **流程说明**: 家教老师查看学生需求详情，点击报名按钮，填写教学经验和优势等信息，提交报名申请。
- **实现细节**:
 - 报名信息: 包括教学经验、专长、自我介绍等。
 - 信息验证: 系统验证报名信息的完整性和有效性。
 - 状态跟踪: 报名申请状态为待处理，等待学生方处理。

2. 学生报名家教信息

- **功能描述**: 允许学生对家教信息发起报名申请。

- **流程说明：**学生查看家教信息详情，点击报名按钮，填写学生的学习情况和需求等信息，提交报名申请。
- **实现细节：**
 - 报名信息：包括学生年级、学习情况、辅导需求等。
 - 信息验证：系统验证报名信息的完整性和有效性。
 - 状态跟踪：报名申请状态为待处理，等待家教方处理。

3. 报名申请处理

- **功能描述：**允许用户处理收到的报名申请。
- **流程说明：**用户查看收到的报名申请列表，选择报名申请查看详情，点击接受或拒绝按钮处理申请。
- **实现细节：**
 - 申请详情：显示申请人的基本信息、报名信息、申请时间等。
 - 处理操作：支持接受或拒绝报名申请。
 - 状态更新：处理后报名申请状态更新为已接受或已拒绝。
 - 通知机制：处理后系统通知申请人处理结果。

4. 报名申请管理

- **功能描述：**允许用户管理自己的报名申请和收到的报名申请。
- **流程说明：**用户进入我的报名页面，查看自己的报名申请列表和收到的报名申请列表，了解申请状态。
- **实现细节：**
 - 申请列表：显示报名申请的标题、申请时间、状态等。
 - 状态筛选：支持按申请状态（待处理、已接受、已拒绝）筛选。
 - 详情查看：点击申请可以查看详细内容。

3.6.2 技术实现

前端实现：

- **报名页面：**使用 Flutter 构建，包含报名信息填写表单、提交按钮等。

- **报名申请列表页面**: 使用 Flutter 构建, 包含报名申请列表, 支持下拉刷新和上拉加载更多。
- **报名申请详情页面**: 使用 Flutter 构建, 包含申请详情、处理按钮等。
- **状态管理**: 使用 Provider 管理报名申请状态和列表数据。
- **网络请求**: 使用 HTTP 库发送报名申请、处理申请等请求。

后端实现:

- **报名控制器**: 处理报名申请相关的请求, 包括发起报名、处理报名、查询报名等。
- **报名服务**: 实现报名申请相关的业务逻辑, 包括申请创建、状态更新、通知发送等。
- **用户服务**: 提供用户信息查询功能, 用于显示申请人的基本信息。
- **数据访问**: 使用 MyBatis Plus 访问数据库, 操作报名申请表。
- **通知服务**: 实现报名申请处理结果的通知功能。

3.5 在线沟通模块

在线沟通模块提供用户间即时聊天功能, 支持文字和图片消息的发送和接收。用户可以查看聊天会话列表, 会话列表显示所有聊天对象的头像、昵称、最后一条消息内容和时间, 未读消息会显示红点提示。用户点击会话进入聊天详情页面, 可以查看历史消息记录, 发送新的消息。

系统使用 WebSocket 实现实时通信, 当有新消息到达时, 会实时推送给接收方, 接收方可以立即收到消息通知。消息支持已读和未读状态管理, 发送方可以查看消息是否已被对方阅读。图片消息需要先上传到服务器, 服务器返回图片 URL 后发给接收方, 接收方点击图片可以查看大图。

3.6 预约报名模块

预约报名模块允许用户对家教信息或学生需求发起报名申请。家教老师可以查看学生需求列表, 对感兴趣的需求发起报名申请, 填写自己的教学经验和优势等信息。学生可以查看家教信息列表, 对感兴趣的家教发起报名申请, 填写学生的学习情况和需求等信息。

报名申请提交后, 对方会收到报名通知, 可以查看报名申请的详细信息, 包括申请人的基本信息、教学经验或学习情况等。对方可以接受或拒绝报名申请, 接受后双

方可以进行进一步的沟通和预约。系统会跟踪报名申请的状态，包括待处理、已接受、已拒绝等状态，用户可以查看自己的报名申请列表和收到的报名申请列表。

3.7 预约管理模块

预约管理模块提供预约信息的查看和管理功能。用户可以查看自己的预约列表，包括预约的家教或学生、预约时间、预约状态等信息。预约状态包括待确认、已确认、已完成、已取消等状态。用户可以查看预约的详细信息，包括双方的基本信息、家教信息或学生需求、沟通记录等。

用户可以对预约进行确认或取消操作。确认操作会将预约状态更新为已确认，双方会收到预约确认通知。取消操作会将预约状态更新为已取消，双方会收到预约取消通知。预约完成后，双方可以对对方进行评价，评价内容包括星级评分和文字评价。用户还可以查看预约历史记录，了解过去的预约情况。

3.7.1 功能详细说明

1. 预约创建

- **功能描述：**允许用户创建预约。
- **流程说明：**用户在报名申请通过后，可以与对方协商预约时间，创建正式预约。
- **实现细节：**
 - 时间选择：用户可以选择预约的具体日期和时间段。
 - 地点设置：用户可以设置预约的具体地点。
 - 备注信息：用户可以添加预约备注，如特殊要求、注意事项等。

2. 预约确认

- **功能描述：**允许用户确认预约。
- **流程说明：**用户收到预约请求后，查看预约详情，点击确认按钮确认预约。
- **实现细节：**
 - 状态更新：确认后预约状态更新为已确认。
 - 通知机制：确认后系统通知对方预约已确认。
 - 日历同步：确认后的预约可以同步到用户的手机日历。

3. 预约取消

- **功能描述**: 允许用户取消预约。
- **流程说明**: 用户进入预约详情页面，点击取消按钮取消预约，填写取消原因。
- **实现细节**:
 - 状态更新：取消后预约状态更新为已取消。
 - 通知机制：取消后系统通知对方预约已取消。
 - 取消原因：用户需要填写取消预约的原因，方便对方了解。

4. 预约管理

- **功能描述**: 允许用户管理自己的预约。
- **流程说明**: 用户进入预约管理页面，查看所有预约列表，了解预约状态和详情。
- **实现细节**:
 - 预约列表：显示预约的标题、时间、状态等信息。
 - 状态筛选：支持按预约状态（待确认、已确认、已完成、已取消）筛选。
 - 详情查看：点击预约可以查看详细内容，包括双方信息、预约时间、地点等。

3.7.2 技术实现

前端实现：

- **预约管理页面**: 使用 Flutter 构建，包含预约列表，支持下拉刷新和上拉加载更多。
- **预约详情页面**: 使用 Flutter 构建，包含预约详情、确认/取消按钮等。
- **预约创建页面**: 使用 Flutter 构建，包含时间选择器、地点选择器、备注输入框等。
- **状态管理**: 使用 Provider 管理预约状态和列表数据。
- **网络请求**: 使用 HTTP 库发送预约创建、确认、取消等请求。

后端实现：

- **预约控制器**: 处理预约相关的请求，包括预约创建、确认、取消、查询等。
- **预约服务**: 实现预约相关的业务逻辑，包括预约创建、状态更新、通知发送等。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作预约表。
- **通知服务**: 实现预约状态变更的通知功能。

用户可以查看自己收到的评价列表，了解其他用户对自己的评价内容。评价记录会显示评价人的基本信息、评分、评价内容、评价时间等信息。用户还可以查看自己的评价统计信息，包括平均评分、评价数量等。评价系统帮助用户了解家教或学生的服务质量，为其他用户提供参考。

3.8.1 功能详细说明

1. 评价提交

- **功能描述**: 允许用户提交评价。
- **流程说明**: 用户在完成对接后，进入评价页面，选择星级评分，输入文字评价，点击提交按钮完成评价。
- **实现细节**:
 - 星级评分: 支持 1 到 5 星评分，用户可以点击星星选择评分。
 - 文字评价: 支持输入 200 字以内的文字评价。
 - 评价验证: 系统验证评价内容的完整性，确保用户已选择评分。
 - 评价限制: 每个对接只能评价一次，不能重复评价。

2. 评价查看

- **功能描述**: 允许用户查看评价记录。
- **流程说明**: 用户进入评价列表页面，查看自己收到的评价或其他用户的评价。
- **实现细节**:
 - 评价列表: 显示评价人的头像、昵称、评分、评价内容、评价时间等。
 - 分页显示: 评价列表支持分页，每页显示 20 条评价。

- 排序方式：支持按评价时间、评分高低排序。

3. 评价统计

- **功能描述：**显示用户的评价统计信息。
- **流程说明：**用户进入个人中心页面，查看自己的评价统计信息，包括平均评分、评价数量等。
- **实现细节：**
 - 平均评分：显示用户的平均星级评分，精确到小数点后一位。
 - 评价数量：显示用户收到的评价总数。
 - 评分分布：显示各星级评分的数量和比例，如 5 星评价占比、4 星评价占比等。

3.8.2 技术实现

前端实现：

- **评价页面：**使用 Flutter 构建，包含星级评分组件、文字评价输入框、提交按钮等。
- **评价列表页面：**使用 Flutter 构建，包含评价列表，支持下拉刷新和上拉加载更多。
- **评价统计组件：**使用 Flutter 构建，显示用户的评价统计信息，包括平均评分、评价数量等。
- **状态管理：**使用 Provider 管理评价状态和列表数据。
- **网络请求：**使用 HTTP 库发送评价提交、查询等请求。

后端实现：

- **评价控制器：**处理评价相关的请求，包括评价提交、查询、统计等。
- **评价服务：**实现评价相关的业务逻辑，包括评价创建、查询、统计等。
- **数据访问：**使用 MyBatis Plus 访问数据库，操作评价表。
- **缓存机制：**使用 Redis 缓存用户的评价统计信息，提高查询速度。

3.9 收藏管理模块

收藏管理模块允许用户收藏感兴趣的家教信息或学生需求。用户在浏览家教信息或学生需求时，可以点击收藏按钮将其添加到收藏列表，收藏状态会实时同步到服务器。用户可以查看自己的收藏列表，收藏列表显示所有收藏的家教信息或学生需求，包括标题、价格、地点、发布时间等信息。

用户可以取消收藏，点击取消收藏按钮后，该家教信息或学生需求会从收藏列表中移除。收藏功能帮助用户快速找到感兴趣的内容，方便后续查看和联系。收藏列表支持分页显示，用户可以上拉加载更多内容。

3.9.1 功能详细说明

1. 收藏添加

- **功能描述：**允许用户收藏家教信息或学生需求。
- **流程说明：**用户在浏览家教信息或学生需求详情页面时，点击收藏按钮添加到收藏列表。
- **实现细节：**
 - 状态切换：点击收藏按钮后，按钮状态切换为已收藏，实时同步到服务器。
 - 重复检测：系统检测用户是否已收藏该信息，避免重复收藏。
 - 收藏反馈：收藏成功后显示收藏成功的提示信息。

2. 收藏取消

- **功能描述：**允许用户取消收藏。
- **流程说明：**用户在收藏列表页面或信息详情页面，点击已收藏按钮取消收藏。
- **实现细节：**
 - 状态切换：点击取消收藏按钮后，按钮状态切换为未收藏，实时同步到服务器。
 - 收藏反馈：取消收藏成功后显示取消收藏成功的提示信息。

3. 收藏管理

- **功能描述：**允许用户管理自己的收藏。

- **流程说明：**用户进入收藏管理页面，查看所有收藏列表，了解收藏内容和状态。
- **实现细节：**
 - 收藏列表：显示收藏的家教信息或学生需求的标题、价格、地点、发布时间等信息。
 - 类型筛选：支持按收藏类型（家教信息、学生需求）筛选。
 - 详情查看：点击收藏可以查看详细内容，方便用户快速了解。

3.9.2 技术实现

前端实现：

- **收藏按钮组件：**使用 Flutter 构建，支持收藏和取消收藏功能，显示收藏状态。
- **收藏管理页面：**使用 Flutter 构建，包含收藏列表，支持下拉刷新和上拉加载更多。
- **状态管理：**使用 Provider 管理收藏状态和列表数据。
- **网络请求：**使用 HTTP 库发送收藏添加、取消、查询等请求。

后端实现：

- **收藏控制器：**处理收藏相关的请求，包括收藏添加、取消、查询等。
- **收藏服务：**实现收藏相关的业务逻辑，包括收藏创建、删除、查询等。
- **数据访问：**使用 MyBatis Plus 访问数据库，操作收藏表。

3.10 黑名单模块

黑名单模块允许用户将不喜欢的用户添加到黑名单。用户在查看其他用户信息时，可以点击添加到黑名单按钮，将该用户添加到黑名单。黑名单用户无法查看自己的信息，无法给自己发送消息，无法对自己发起报名申请。用户可以查看自己的黑名单列表，黑名单列表显示所有被拉黑的用户信息。

用户可以将用户从黑名单中移除，点击移除按钮后，该用户会从黑名单中移除，恢复正常交互。黑名单功能帮助用户避免与不喜欢的用户进行交互，提升使用体验。

3.10.1 功能详细说明

1. 添加黑名单

- **功能描述:** 允许用户将其他用户添加到黑名单。
- **流程说明:** 用户在查看其他用户信息页面时，点击添加到黑名单按钮，将该用户添加到黑名单。
- **实现细节:**
 - 操作确认：添加到黑名单前需要用户确认，防止误操作。
 - 状态更新：添加成功后，该用户被加入黑名单，无法与自己进行交互。
 - 反馈提示：添加成功后显示添加成功的提示信息。

2. 移除黑名单

- **功能描述:** 允许用户将用户从黑名单中移除。
- **流程说明:** 用户进入黑名单管理页面，选择要移除的用户，点击移除按钮将其从黑名单中移除。
- **实现细节:**
 - 操作确认：移除黑名单前需要用户确认，防止误操作。
 - 状态更新：移除成功后，该用户从黑名单中移除，恢复正常交互。
 - 反馈提示：移除成功后显示移除成功的提示信息。

3. 黑名单管理

- **功能描述:** 允许用户管理自己的黑名单。
- **流程说明:** 用户进入黑名单管理页面，查看所有被拉黑的用户列表，了解黑名单内容。
- **实现细节:**
 - 黑名单列表：显示被拉黑用户的头像、昵称、添加时间等信息。
 - 移除操作：支持单个移除或批量移除黑名单用户。
 - 详情查看：点击用户可以查看用户的详细信息，方便用户确认。

3.10.2 技术实现

前端实现：

- **黑名单管理页面**：使用 Flutter 构建，包含黑名单用户列表，支持滑动移除。
- **用户信息页面**：使用 Flutter 构建，包含添加到黑名单按钮。
- **状态管理**：使用 Provider 管理黑名单状态和列表数据。
- **网络请求**：使用 HTTP 库发送添加黑名单、移除黑名单、查询黑名单等请求。

后端实现：

- **黑名单控制器**：处理黑名单相关的请求，包括添加黑名单、移除黑名单、查询黑名单等。
- **黑名单服务**：实现黑名单相关的业务逻辑，包括黑名单创建、删除、查询等。
- **数据访问**：使用 MyBatis Plus 访问数据库，操作黑名单表。
- **权限控制**：在相关接口中检查用户是否在黑名单中，限制黑名单用户的操作。

3.11 积分系统模块

积分系统模块提供积分获取和消耗功能。用户可以通过完成指定任务获取积分，例如完善个人信息、发布家教信息、发布学生需求、完成对接等。积分可以用于兑换平台提供的各种服务和优惠，例如提升信息曝光率、查看更多家教信息等。

用户可以查看自己的积分余额，了解当前可用的积分数量。用户可以查看积分明细记录，了解积分的获取和消耗情况，包括积分变化数量、变化原因、变化时间等信息。积分系统激励用户积极参与平台活动，提升平台活跃度。

3.11.1 功能详细说明

1. 积分获取

- **功能描述**：允许用户通过完成任务获取积分。
- **流程说明**：用户完成指定任务后，系统自动发放积分到用户账户。
- **实现细节**：
 - 任务类型：包括完善个人信息（+10 积分）、发布家教信息（+20 积分）、发布学生需求（+20 积分）、完成对接（+50 积分）等。

- 积分发放：任务完成后，系统自动计算并发放积分，显示积分到账通知。
- 任务限制：部分任务有每日或每周限制，例如每日发布信息最多获得 100 积分。

2. 积分消耗

- 功能描述：允许用户使用积分兑换服务或优惠。
- 流程说明：用户进入积分商城页面，选择要兑换的服务或优惠，点击兑换按钮使用积分兑换。
- 实现细节：
 - 兑换项目：包括提升信息曝光率（50 积分）、查看更多家教信息（30 积分）、优先推荐（100 积分）等。
 - 积分扣除：兑换成功后，系统扣除相应积分，显示积分扣除通知。
 - 兑换记录：系统保存用户的积分兑换记录，方便用户查询。

3. 积分管理

- 功能描述：允许用户管理自己的积分。
- 流程说明：用户进入积分管理页面，查看积分余额和积分明细记录。
- 实现细节：
 - 积分余额：显示用户当前的积分余额。
 - 积分明细：显示积分的获取和消耗记录，包括积分变化数量、原因、时间等。
 - 明细筛选：支持按时间、类型（获取、消耗）筛选积分明细。

3.11.2 技术实现

前端实现：

- 积分管理页面：使用 Flutter 构建，包含积分余额显示、积分明细列表等。
- 积分商城页面：使用 Flutter 构建，包含可兑换的服务和优惠列表。
- 积分通知组件：使用 Flutter 构建，显示积分到账和扣除的通知。

- **状态管理**: 使用 Provider 管理积分状态和明细数据。
- **网络请求**: 使用 HTTP 库发送积分查询、兑换等请求。

后端实现:

- **积分控制器**: 处理积分相关的请求，包括积分查询、兑换、明细查询等。
- **积分服务**: 实现积分相关的业务逻辑，包括积分发放、扣除、兑换等。
- **任务服务**: 实现任务管理相关的业务逻辑，包括任务完成检测、积分发放触发等。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作积分表和积分明细记录表。

3.12 投诉举报模块

投诉举报模块允许用户对违规行为进行投诉举报。用户在查看其他用户信息时，如果发现违规行为，可以点击投诉举报按钮，填写投诉内容，选择投诉类型，提交投诉申请。投诉类型包括虚假信息、骚扰行为、欺诈行为等。

用户可以查看自己的投诉列表，投诉列表显示所有提交的投诉申请，包括投诉对象、投诉类型、投诉内容、投诉状态等信息。投诉状态包括待处理、处理中、已处理等状态。平台管理员会审核投诉申请，对违规行为进行处理，并将处理结果反馈给投诉人。

3.12.1 功能详细说明

1. 投诉提交

- **功能描述**: 允许用户提交投诉举报。
- **流程说明**: 用户在查看其他用户信息页面时，点击投诉举报按钮，填写投诉内容，选择投诉类型，提交投诉申请。
- **实现细节**:
 - 投诉类型: 包括虚假信息、骚扰行为、欺诈行为、违规内容等。
 - 内容填写: 用户可以输入详细的投诉内容，支持上传图片证据。
 - 提交验证: 系统验证投诉内容的完整性，确保用户已填写必要信息。
 - 提交反馈: 提交成功后显示投诉提交成功的提示信息。

2. 投诉管理

- **功能描述**: 允许用户管理自己的投诉。
- **流程说明**: 用户进入投诉管理页面，查看所有提交的投诉列表，了解投诉状态和处理结果。
- **实现细节**:
 - 投诉列表：显示投诉的对象、类型、提交时间、状态等信息。
 - 状态筛选：支持按投诉状态（待处理、处理中、已处理）筛选。
 - 详情查看：点击投诉可以查看详细内容和处理结果。

3.12.2 技术实现

前端实现：

- **投诉举报页面**: 使用 Flutter 构建，包含投诉类型选择器、投诉内容输入框、图片上传组件、提交按钮等。
- **投诉管理页面**: 使用 Flutter 构建，包含投诉列表，支持下拉刷新和上拉加载更多。
- **状态管理**: 使用 Provider 管理投诉状态和列表数据。
- **网络请求**: 使用 HTTP 库发送投诉提交、查询等请求。

后端实现：

- **投诉控制器**: 处理投诉相关的请求，包括投诉提交、查询、处理等。
- **投诉服务**: 实现投诉相关的业务逻辑，包括投诉创建、查询、处理等。
- **管理员服务**: 实现管理员审核投诉的业务逻辑，包括投诉处理、结果反馈等。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作投诉表。

3.13 消息通知模块

消息通知模块提供系统消息推送功能。系统会在重要事件发生时向用户推送通知，例如报名申请通知、预约状态变更通知、评价通知等。用户可以查看消息通知列表，通知列表显示所有未读和已读的消息，包括通知类型、通知内容、通知时间等信息。

用户可以点击通知查看详细信息，查看后通知状态会更新为已读。用户可以删除通知，点击删除按钮后，该通知会从通知列表中移除。消息通知功能帮助用户及时了解重要信息，不会错过重要事件。

3.13.1 功能详细说明

1. 消息接收

- **功能描述:** 允许用户接收系统消息通知。
- **流程说明:** 系统在重要事件发生时，自动向相关用户推送消息通知。
- **实现细节:**
 - 通知类型：包括报名申请通知、预约状态变更通知、评价通知、系统公告等。
 - 通知内容：通知内容包括事件描述、相关信息、操作按钮等。
 - 通知推送：系统通过 WebSocket 或推送服务向用户推送通知。

2. 消息管理

- **功能描述:** 允许用户管理自己的消息通知。
- **流程说明:** 用户进入消息通知页面，查看所有消息通知列表，管理消息状态和内容。
- **实现细节:**
 - 消息列表：显示消息的类型、内容、时间、状态等信息。
 - 状态更新：点击消息查看后，消息状态更新为已读。
 - 消息删除：支持单个删除或批量删除消息通知。
 - 未读提示：未读消息显示红点提示，显示未读消息数量。

3.13.2 技术实现

前端实现：

- **消息通知页面:** 使用 Flutter 构建，包含消息列表，支持下拉刷新和滑动删除。
- **消息详情页面:** 使用 Flutter 构建，显示消息的详细内容和相关信息。
- **通知组件:** 使用 Flutter 构建，显示系统推送的通知，支持点击查看详情。
- **状态管理:** 使用 Provider 管理消息状态和列表数据。

- **网络请求**: 使用 HTTP 库发送消息查询、标记已读、删除等请求。

后端实现:

- **通知控制器**: 处理通知相关的请求，包括通知查询、标记已读、删除等。
- **通知服务**: 实现通知相关的业务逻辑，包括通知创建、推送、查询等。
- **事件监听**: 监听系统中的重要事件，触发通知推送。
- **数据访问**: 使用 MyBatis Plus 访问数据库，操作通知表。

3.8 评价反馈模块

评价反馈模块允许用户在完成对接后对对方进行评价。评价包括星级评分和文字评价两部分，星级评分范围为 1 到 5 星，用户可以根据对接体验选择评分，文字评价可以详细描述对接过程中的优点和不足。评价提交后会保存到系统中，其他用户可以查看评价记录。

用户可以查看自己收到的评价列表，了解其他用户对自己的评价内容。评价记录会显示评价人的基本信息、评分、评价内容、评价时间等信息。用户还可以查看自己的评价统计信息，包括平均评分、评价数量等。评价系统帮助用户了解家教或学生的服务质量，为其他用户提供参考。

3.9 收藏管理模块

收藏管理模块允许用户收藏感兴趣的家教信息或学生需求。用户在浏览家教信息或学生需求时，可以点击收藏按钮将其添加到收藏列表，收藏状态会实时同步到服务器。用户可以查看自己的收藏列表，收藏列表显示所有收藏的家教信息或学生需求，包括标题、价格、地点、发布时间等信息。

用户可以取消收藏，点击取消收藏按钮后，该家教信息或学生需求会从收藏列表中移除。收藏功能帮助用户快速找到感兴趣的内容，方便后续查看和联系。收藏列表支持分页显示，用户可以上拉加载更多内容。

3.10 黑名单模块

黑名单模块允许用户将不喜欢的用户添加到黑名单。用户在查看其他用户信息时，可以点击添加到黑名单按钮，将该用户添加到黑名单。黑名单用户无法查看自己的信息，无法给自己发送消息，无法对自己发起报名申请。用户可以查看自己的黑名单列表，黑名单列表显示所有被拉黑的用户信息。

用户可以将用户从黑名单中移除，点击移除按钮后，该用户会从黑名单中移除，恢复正常交互。黑名单功能帮助用户避免与不喜欢的用户进行交互，提升使用体验。

3.11 积分系统模块

积分系统模块提供积分获取和消耗功能。用户可以通过完成指定任务获取积分，例如完善个人信息、发布家教信息、发布学生需求、完成对接等。积分可以用于兑换平台提供的各种服务和优惠，例如提升信息曝光率、查看更多家教信息等。

用户可以查看自己的积分余额，了解当前可用的积分数量。用户可以查看积分明细记录，了解积分的获取和消耗情况，包括积分变化数量、变化原因、变化时间等信息。积分系统激励用户积极参与平台活动，提升平台活跃度。



我的积分

明细

当前积分

5

如何获得积分

→ 每日登录

+5 积分

3.12 投诉举报模块

投诉举报模块允许用户对违规行为进行投诉举报。用户在查看其他用户信息时，如果发现违规行为，可以点击投诉举报按钮，填写投诉内容，选择投诉类型，提交投诉申请。投诉类型包括虚假信息、骚扰行为、欺诈行为等。

用户可以查看自己的投诉列表，投诉列表显示所有提交的投诉申请，包括投诉对象、投诉类型、投诉内容、投诉状态等信息。投诉状态包括待处理、处理中、已处理等状态。平台管理员会审核投诉申请，对违规行为进行处理，并将处理结果反馈给投诉人。

3.13 消息通知模块

消息通知模块提供系统消息推送功能。系统会在重要事件发生时向用户推送通知，例如报名申请通知、预约状态变更通知、评价通知等。用户可以查看消息通知列表，通知列表显示所有未读和已读的消息，包括通知类型、通知内容、通知时间等信息。

用户可以点击通知查看详细信息，查看后通知状态会更新为已读。用户可以删除通知，点击删除按钮后，该通知会从通知列表中移除。消息通知功能帮助用户及时了解重要信息，不会错过重要事件。

四、操作界面

4.1 启动页面

启动页面是应用启动时展示的第一个页面，显示应用的 Logo 和名称，给用户留下第一印象。启动页面会自动跳转，如果用户已经登录，则跳转到主页，如果用户未登录，则跳转到登录页面。启动页面的设计简洁大方，突出品牌形象。

启动页面在应用启动时显示，通常显示应用 Logo、应用名称、版本号等信息。启动页面会自动跳转，不需要用户手动操作。跳转逻辑如下：首先检查本地存储中是否有 JWT 令牌，如果有令牌且令牌有效，则跳转到主页；如果没有令牌或令牌无效，则跳转到登录页面。

启动页面还负责初始化应用的全局配置，包括初始化 Provider、加载本地数据、检查网络连接等。启动页面会显示加载动画，给用户友好的视觉反馈。启动页面的加载时间通常为 1-3 秒，确保用户能够看到启动页面，避免应用启动时的空白屏幕。

启动页面的设计遵循 Material Design 设计规范，使用简洁的布局和动画效果。启动页面的背景色与应用主题色一致，Logo 居中显示，应用名称位于 Logo 下方。启动页面还支持深色模式，根据系统设置自动切换主题。

跳过 3

好会帮

上好会帮，赢好未来

4.2 登录注册页面

登录注册页面提供用户登录和注册功能。页面顶部显示应用名称和标语，中间是登录表单，包括手机号输入框和密码输入框，底部是登录按钮和辅助功能按钮。用户输入手机号和密码后点击登录按钮，系统验证通过后跳转到主页。

页面还提供验证码登录和忘记密码功能。用户点击验证码登录按钮跳转到验证码登录页面，可以输入手机号和验证码进行登录。用户点击忘记密码按钮跳转到忘记密码页面，可以通过手机号验证码重置密码。登录注册页面的设计简洁清晰，操作流程顺畅。

登录功能支持密码登录和短信验证码登录两种方式。密码登录需要用户输入手机号和密码，系统验证通过后生成 JWT 令牌并返回给客户端，客户端保存令牌用于后续请求的身份认证。短信验证码登录需要用户输入手机号和验证码，系统验证验证码的有效性后生成 JWT 令牌并返回给客户端。

注册功能需要用户输入手机号、密码、确认密码等信息，系统验证手机号是否已注册，密码是否符合安全要求，确认密码是否与密码一致。注册成功后，系统会自动登录，生成 JWT 令牌并返回给客户端。注册成功后，用户需要完善个人信息，包括头像、昵称、性别、生日等。

忘记密码功能需要用户输入手机号，系统会发送短信验证码到用户手机，用户输入验证码后可以设置新密码。新密码需要符合安全要求，包括长度、复杂度等。密码重置成功后，用户可以使用新密码登录。

登录注册页面还提供记住密码功能，用户可以选择记住密码，下次登录时自动填充密码。记住密码功能使用本地存储保存密码，密码经过加密处理，确保安全性。登录注册页面还提供第三方登录功能，支持微信、QQ、支付宝等第三方账号登录。

4.3 角色选择页面

角色选择页面是用户首次登录后显示的页面，要求用户选择自己的角色，包括学生和家长、家教老师两个角色。用户选择角色后，系统会根据角色跳转到不同的主页，展示不同的功能模块。角色选择页面的设计简洁明了，用户可以快速理解并选择自己的角色。

4.4 首页

首页是平台的主要功能入口，展示家教信息和学生需求的列表。页面顶部是搜索框和筛选按钮，用户可以输入关键词搜索家教或需求，点击筛选按钮可以设置筛选条件。页面中间是家教信息和学生需求的列表，每个条目显示家教或学生的头像、昵称、授课科目、时薪价格、地点等信息。

页面底部是导航栏，包括首页、发现、消息、我的四个标签页。用户点击标签页可以切换到不同的功能页面。首页的设计信息丰富，操作便捷，用户可以快速找到感兴趣的家教或需求。

搜索学生需求、科目...



附近学生



附近老师



我要发布



我的积分

附近学生

[查看更多](#)

小

小明的需求

数学

认真负责

¥80.00-160.00/小时

4.5 发现页面

发现页面提供更多的功能入口，包括附近家教、热门科目、最新需求等模块。附近家教模块基于用户的地理位置显示附近的家教信息，热门科目模块显示最受欢迎的辅导科目，最新需求模块显示最新发布的学生成需求。

用户可以点击模块查看更多内容，例如点击附近家教模块可以查看附近的家教列表，点击热门科目模块可以查看科目的家教列表，点击最新需求模块可以查看学生成需求列表。发现页面的设计功能丰富，帮助用户发现更多有价值的内容。

4.6 消息页面

消息页面显示聊天会话列表，包括系统通知和用户聊天。系统通知显示平台发送的各种通知，例如报名申请通知、预约状态变更通知等。用户聊天显示与用户的聊天会话，每个会话显示对方的头像、昵称、最后一条消息内容和时间。

用户点击会话可以进入聊天详情页面，查看历史消息记录，发送新的消息。未读消息会显示红点提示，用户可以快速识别未读消息。消息页面的设计简洁清晰，用户可以方便地查看和管理消息。

4.7 个人中心页面

个人中心页面显示用户的个人信息和功能入口。页面顶部显示用户的头像、昵称、积分等信息，中间显示功能菜单，包括我的发布、我的报名、我的收藏、我的评价、我的积分、设置等功能。

用户点击功能菜单可以跳转到对应的页面，例如点击我的发布可以查看自己发布的家教信息和学生成需求，点击我的报名可以查看自己的报名申请，点击我的收藏可以查看收藏列表。个人中心页面的设计功能齐全，用户可以方便地管理个人信息和功能。

用

用户38122

已认证



家教

0

家教预约

1

我的发布

0

收藏家教

5

我的积分



请家教

发布家教需求



做家教

发布家教信息

家教信息



我的预约



我的报名



我的收藏



我的评价



家教简历



家教认证



联系客服



我的举报

4.8 发布家教信息页面

发布家教信息页面允许家教老师发布家教信息。页面顶部是家教信息表单，包括授课科目、时薪价格、授课地点、可授课时间段等输入框，底部是提交按钮。家教老师填写完信息后点击提交按钮，系统验证通过后发布家教信息。

页面还提供预览功能，家教老师可以预览家教信息的展示效果。发布家教信息页面的设计表单清晰，操作简单，家教老师可以快速发布家教信息。



发布家教信息

基本信息

教授科目

数学



目标年级

学前

小学

初中

高中

时薪 (元/小时)

100

元/小时

教学信息

说点什么

可授课时间



您的地址



发布家教信息

4.9 发布学生需求页面

发布学生需求页面允许学生和家长发布学生需求。页面顶部是学生需求表单，包括辅导科目、学生年级、时薪预算、上课地点等输入框，底部是提交按钮。用户填写完信息后点击提交按钮，系统验证通过后发布学生需求。

页面还提供预览功能，用户可以预览学生需求的展示效果。发布学生需求页面的设计表单清晰，操作简单，用户可以快速发布学生需求。



发布请家教需求

基本信息

孩子称呼

孩子年级

请选择年级

请选择科目

请选择科目

时薪范围

¥50 - ¥200/小时



教学要求

教学要求

可上课时间

4.10 地图页面

地图页面基于高德地图显示附近的家教信息和学生需求。页面顶部是地图视图，显示用户当前位置和附近的家教或需求标记，底部是列表视图，显示附近的家教或需求列表。

用户可以点击地图标记查看家教或需求的详细信息，也可以点击列表条目查看详细信息。地图页面支持缩放和拖动，用户可以查看不同区域的家教或需求。地图页面的设计直观清晰，用户可以方便地查看附近的家教或需求。



李星辰 ✅ 已认证

李

数学

100.00元/小时

查看详情

4.11 聊天详情页面

聊天详情页面显示与用户的聊天记录。页面顶部是对方的基本信息，包括头像、昵称等，中间是消息列表，显示历史消息记录，底部是输入框和发送按钮。

用户可以输入文字消息，点击发送按钮发送消息。用户还可以点击图片按钮选择图片发送图片消息。消息列表支持滚动，用户可以查看历史消息记录。聊天详情页面的设计简洁清晰，用户可以方便地进行聊天。

4.12 预约页面

预约页面显示用户的预约列表。页面顶部是预约列表，显示所有预约信息，包括预约的家教或学生、预约时间、预约状态等。用户可以点击预约查看详细信息，也可以对预约进行确认或取消操作。

预约状态包括待确认、已确认、已完成、已取消等状态，不同状态用不同颜色标识。预约页面的设计信息丰富，用户可以方便地查看和管理预约。

4.13 评价页面

评价页面允许用户对对接对象进行评价。页面顶部是评价表单，包括星级评分和文字评价，底部是提交按钮。用户选择星级评分，输入文字评价后点击提交按钮，系统保存评价信息。

星级评分范围为 1 到 5 星，用户可以根据对接体验选择评分。文字评价可以详细描述对接过程中的优点和不足。评价页面的设计简洁清晰，用户可以方便地进行评价。

4.14 设置页面

设置页面提供各种设置选项，包括个人信息设置、账号安全设置、通知设置等。用户可以修改个人信息，包括头像、昵称、性别、生日等。用户可以修改账号安全设置，包括密码、手机号、邮箱等。用户可以修改通知设置，包括消息通知、系统通知等。

设置页面的设计功能齐全，用户可以方便地管理各种设置选项。

五、操作流程

5.1 家教老师操作流程

家教老师使用平台的流程如下：首先，家教老师需要注册登录平台，可以通过手机号注册，选择密码登录或验证码登录。登录后需要完善个人信息，包括头像、昵称、性别、生日、教育背景、教学经验等。完善个人信息后需要进行家教教师资质认证，上传教师资格证、学历证书等相关证明材料，等待平台审核。

资质认证通过后，家教老师可以发布家教信息，包括授课科目、时薪价格、授课地点、可授课时间段等信息。发布家教信息后，家教老师可以查看学生需求列表，对感兴趣的需求发起报名申请，填写自己的教学经验和优势等信息。学生接受报名申请后，双方可以进行在线沟通，进一步了解对方的情况。

沟通确认后，双方可以确认预约，设置预约时间。预约完成后，家教老师可以按照预约时间提供家教服务。服务完成后，家教老师可以对学生进行评价，包括星级评分和文字评价。家教老师还可以查看自己的发布列表、报名列表、预约列表、评价列表等，了解自己的服务情况。

5.2 学生家长操作流程

学生家长使用平台的流程如下：首先，学生家长需要注册登录平台，可以通过手机号注册，选择密码登录或验证码登录。登录后需要完善个人信息，包括头像、昵称、性别、生日等。完善个人信息后可以发布学生需求，包括辅导科目、学生年级、时薪预算、上课地点等信息。

发布学生需求后，学生家长可以搜索家教信息，按科目、年级、时薪范围、地理位置等条件筛选，找到合适的家教。查看家教信息后，可以对感兴趣的家教发起报名申请，填写学生的学习情况和需求等信息。家教老师接受报名申请后，双方可以进行在线沟通，进一步了解对方的情况。

沟通确认后，双方可以确认预约，设置预约时间。预约完成后，学生家长可以按照预约时间接受家教服务。服务完成后，学生家长可以对家教老师进行评价，包括星级评分和文字评价。学生家长还可以查看自己的发布列表、报名列表、预约列表、评价列表等，了解自己的对接情况。

六、技术特点

6.1 跨平台开发

本软件采用 Flutter 框架实现跨平台开发，使用一套代码同时支持 iOS 和 Android 双平台。Flutter 框架采用自绘渲染引擎，不依赖原生控件，可以保证在不同平台上的一致性体验。Flutter 框架采用热重载技术，开发过程中可以快速预览代码修改的效果，提高开发效率。

Flutter 框架采用组件化开发模式，所有界面都由组件组成，组件可以复用和组合，提高代码复用率。Flutter 框架采用声明式 UI 编程，界面状态变化时自动更新界面，减少手动操作界面的代码。Flutter 框架采用 Dart 编程语言，Dart 语言具有强类型、面向对象、异步编程等特性，适合移动应用开发。

6.2 前后端分离架构

本软件采用前后端分离的架构设计，前端使用 Flutter 框架开发移动客户端，后端使用 Spring Boot 框架开发 RESTful API 接口。前后端通过 HTTP 协议进行数据交换，数据格式采用 JSON。前后端分离架构具有以下优点：第一，前端可以独立开发和部署，提高开发效率；第二，前端可以复用后端 API，支持多种客户端；第三，前端可以独立扩展，提高系统可扩展性。

Spring Boot 框架采用约定优于配置的设计理念，减少配置工作，提高开发效率。Spring Boot 框架提供自动配置功能，根据项目依赖自动配置相关组件，减少手动配置。Spring Boot 框架提供内嵌服务器，可以直接运行项目，无需部署外部服务器。Spring Boot 框架提供丰富的生态，包括数据访问、安全认证、消息队列等组件，满足各种业务需求。

6.3 实时通信

本软件使用 WebSocket 实现实时通信功能，支持消息的实时推送和接收。

WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议，可以实现服务器主动向客户端推送消息。WebSocket 连接建立后，客户端和服务器可以随时发送消息，无需建立新的连接。

实时通信功能用于在线聊天模块，当用户发送消息时，消息通过 WebSocket 连接实时推送给接收方，接收方可以立即收到消息通知。WebSocket 连接保持长连接状态，减少连接建立的开销，提高消息传输效率。WebSocket 连接支持心跳检测，可以检测连接是否正常，及时重连断开的连接。

6.4 地理位置服务

本软件基于高德地图提供地理位置服务，包括定位、地图展示、附近搜索等功能。定位功能使用 Geolocator 插件获取用户当前位置，支持 GPS 定位和网络定位。地图展示使用 AMap Flutter Map 插件显示地图视图，支持缩放、拖动、标记等操作。附近搜索基于用户当前位置计算距离，按距离远近排序显示结果。

地理位置服务用于地图页面和附近搜索功能，用户可以在地图上查看附近的家教或需求，也可以按距离筛选家教或需求。地理位置服务还用于家教信息和学生需求的地点展示，用户可以查看家教或需求的地理位置，方便选择合适的家教或需求。

6.5 数据安全

本软件采用多种措施保障数据安全。身份认证使用 JWT 令牌，用户登录后生成 JWT 令牌，客户端保存令牌用于后续请求的身份认证。JWT 令牌包含用户 ID 和角色信息，服务器验证令牌的有效性后授权访问。密码使用 BCrypt 算法加密存储，数据库中不存储明文密码，即使数据库泄露也无法获取用户密码。

数据传输使用 HTTPS 协议，所有数据在传输过程中都进行加密，防止数据被窃取或篡改。权限控制基于角色进行，不同角色具有不同的权限，例如管理员可以管理所有数据，普通用户只能管理自己的数据。数据备份定期进行，防止数据丢失，确保数据安全。

6.6 模块化设计

本软件采用模块化设计，将系统划分为多个功能模块，每个模块负责特定的功能。模块化设计具有以下优点：第一，模块之间耦合度低，修改一个模块不会影响其他模块；第二，模块可以独立开发和测试，提高开发效率；第三，模块可以复用，减少重复代码；第四，模块可以独立扩展，提高系统可扩展性。

前端采用组件化开发，将界面划分为多个组件，每个组件负责特定的界面元素。组件可以复用和组合，提高代码复用率。组件可以独立开发和测试，提高开发效率。后端采用分层架构，将系统划分为控制器层、服务层、数据访问层，每层负责特定的功能。分层架构使代码结构清晰，易于维护和扩展。

七、系统要求

7.1 硬件要求

移动客户端的硬件要求包括：处理器需要 ARM 架构，主频至少 1.5GHz；内存至少 2GB；存储空间至少 500MB 可用空间；屏幕分辨率至少 720p；支持 GPS 定位；支持摄像头和麦克风。推荐的硬件配置包括：处理器 ARM 架构，主频 2.0GHz 以上；内存 4GB 以上；存储空间 1GB 以上可用空间；屏幕分辨率 1080p 以上。

服务端的硬件要求包括：处理器 Intel Xeon E5-2680 v4 或更高；内存 64GB DDR4 ECC；存储 2TB SAS SSD；网络带宽 100Mbps 以上。推荐的硬件配置包括：处理器 Intel Xeon E5-2690 v4 或更高；内存 128GB DDR4 ECC；存储 4TB SAS SSD；网络带宽 1Gbps 以上。

7.2 软件要求

移动客户端的软件要求包括：iOS 系统版本 12.0 及以上，或 Android 系统版本 6.0 及以上；需要授予位置权限、相机权限、存储权限等权限。推荐的系统版本包括：iOS 系统版本 14.0 及以上，或 Android 系统版本 10.0 及以上。

服务端的软件要求包括：操作系统 Ubuntu Server 22.04 LTS；数据库 MySQL 8.0.33；Java 运行环境 JDK 17；Web 服务器 Tomcat 9.0 或更高；反向代理 Nginx 1.18 或更高。推荐的软件版本包括：操作系统 Ubuntu Server 22.04 LTS；数据库 MySQL 8.0.33；Java 运行环境 JDK 17；Web 服务器 Tomcat 10.0 或更高；反向代理 Nginx 1.22 或更高。

八、安装部署

8.1 客户端安装

iOS 客户端的安装方式包括：通过 App Store 搜索“HiTutor 好会帮”下载安装，或通过企业证书安装 IPA 文件。安装完成后，首次打开应用需要授予位置权限、相机权限、存储权限等权限，授予权限后即可正常使用应用。

Android 客户端的安装方式包括：通过应用商店搜索“HiTutor 好会帮”下载安装，或通过 APK 文件直接安装。安装 APK 文件需要先开启未知来源应用安装权限，然后点击 APK 文件进行安装。安装完成后，首次打开应用需要授予位置权限、相机权限、存储权限等权限，授予权限后即可正常使用应用。

8.2 服务端部署

服务端部署需要以下步骤：第一，准备服务器环境，安装 Ubuntu Server 22.04 LTS 操作系统，配置网络和安全组；第二，安装 Java 运行环境 JDK 17，配置环境变量；第三，安装 MySQL 8.0.33 数据库，创建数据库和用户，导入数据表；第四，部署后端应用，打包 Spring Boot 应用为 JAR 文件，上传到服务器，启动应用；第五，配置反向代理，安装 Nginx，配置反向代理规则，配置 SSL 证书；第六，配置域名和 DNS，将域名解析到服务器 IP 地址。

服务端部署完成后，需要测试各项功能是否正常，包括用户认证、信息发布、搜索匹配、在线沟通、预约管理、评价反馈等。测试通过后，服务端即可正式上线运行。

九、常见问题

9.1 如何修改 API 地址？

如果需要修改 API 地址，可以编辑客户端项目中的 `client/lib/services/api_service.dart` 文件，找到 `baseUrl` 常量，将其值修改为新的 API 地址。修改完成后，需要重新编译和打包客户端应用。

9.2 如何配置数据库？

如果需要配置数据库，可以编辑后端项目中的 `hitutor-backend/src/main/resources/application.yml` 文件，找到数据库连接配置，修改数据库 URL、用户名、密码等信息。修改完成后，需要重新启动后端应用。

9.3 如何运行项目？

运行后端项目需要进入后端项目目录，执行命令 `mvn spring-boot:run`，启动 Spring Boot 应用。运行前端项目需要进入前端项目目录，执行命令 `flutter run`，启动 Flutter 应用。需要确保后端应用先启动，前端应用才能正常连接后端 API。

9.4 如何构建生产版本?

构建后端生产版本需要进入后端项目目录，执行命令 `mvn clean package`，打包为 JAR 文件。构建前端生产版本需要进入前端项目目录，执行命令 `flutter build apk` 或 `flutter build ios`，打包为 APK 文件或 IPA 文件。生产版本需要经过充分测试，确保功能正常后才能发布。

9.5 如何备份数据?

备份数据需要定期执行数据库备份命令，例如使用 `mysqldump` 命令导出数据库数据，保存为 SQL 文件。备份文件需要存储在安全的位置，防止数据丢失。建议每天执行一次全量备份，每小时执行一次增量备份。

9.6 如何恢复数据?

恢复数据需要使用数据库导入命令，例如使用 `mysql` 命令导入 SQL 文件到数据库。恢复数据前需要先停止应用，恢复数据后再启动应用。恢复数据需要谨慎操作，建议先在测试环境验证恢复流程，确认无误后再在生产环境执行。

十、用户手册

10.1 软件安装

10.1.1 iOS 客户端安装

1. 打开 App Store 应用
2. 在搜索框中输入"HiTutor 好会帮"
3. 点击搜索结果中的应用
4. 点击"获取"按钮进行下载
5. 下载完成后点击"打开"按钮启动应用

10.1.2 Android 客户端安装

1. 打开应用商店（如 Google Play、华为应用市场等）
2. 在搜索框中输入"HiTutor 好会帮"
3. 点击搜索结果中的应用
4. 点击"安装"按钮进行下载
5. 下载完成后点击"打开"按钮启动应用

10.1.3 首次启动设置

1. 首次启动应用时，系统会请求位置权限、相机权限和存储权限，请点击“允许”以确保应用正常运行
2. 进入登录页面，选择“注册”按钮
3. 输入手机号码，点击“获取验证码”
4. 输入收到的验证码，设置登录密码
5. 点击“注册”按钮完成注册
6. 登录后，选择用户角色（学生和家长/家教老师）

10.2 基本操作

10.2.1 浏览家教信息

1. 登录应用后，进入首页
2. 浏览推荐的家教信息列表
3. 点击家教信息卡片查看详细信息
4. 在详细信息页面，可以查看家教的基本信息、教学经验、评价记录等
5. 点击“收藏”按钮可以收藏该家教信息
6. 点击“立即沟通”按钮可以与家教老师进行在线聊天

10.2.2 发布家教需求

1. 点击底部导航栏的“发布”按钮
2. 选择“发布家教需求”
3. 填写需求详情，包括辅导科目、学生年级、时薪预算、上课地点等
4. 点击“提交”按钮发布需求
5. 发布成功后，可以在“我的发布”中查看

10.2.3 搜索功能使用

1. 在首页顶部的搜索框中输入关键词
2. 点击搜索按钮进行搜索
3. 在搜索结果页面，可以使用筛选功能进一步缩小范围

4. 筛选条件包括科目、年级、时薪范围、距离等

10.2.4 在线沟通

1. 点击底部导航栏的"消息"按钮
2. 在消息列表中选择聊天对象
3. 进入聊天页面，输入文字消息
4. 点击"发送"按钮发送消息
5. 点击图片图标可以发送图片消息

10.3 高级功能

10.3.1 家教教师资质认证

1. 进入"我的"页面
2. 点击"家教教师资质认证"
3. 填写个人信息，包括姓名、性别、出生日期、教育背景等
4. 上传教师资格证、学历证书等证明材料
5. 点击"提交认证"按钮
6. 等待平台审核，审核结果会通过消息通知

10.3.2 预约管理

1. 进入"我的"页面
2. 点击"我的预约"
3. 查看所有预约信息，包括待确认、已确认、已完成、已取消等状态
4. 点击预约信息查看详细内容
5. 对于待确认的预约，可以点击"确认"或"取消"按钮

10.3.3 评价系统

1. 完成家教服务后，系统会提示进行评价
2. 进入评价页面，选择星级评分（1-5 星）
3. 输入文字评价，详细描述服务体验
4. 点击"提交评价"按钮

5. 评价成功后，评价内容会显示在对方的个人主页

十一、测试结果

11.1 功能测试

11.1.1 用户认证测试

测试项	测试结果	备注
手机号注册	通过	支持短信验证码注册
密码登录	通过	支持密码找回功能
短信验证码登录	通过	验证码有效期 5 分钟
实名认证	通过	需要上传身份证照片
家教教师资质认证	通过	需要上传相关证明材料

11.1.2 信息发布测试

测试项	测试结果	备注
发布家教信息	通过	需要先完成资质认证
发布学生需求	通过	注册用户即可发布
编辑家教信息	通过	修改后需要重新审核
编辑学生需求	通过	实时更新显示
删除信息	通过	支持软删除

11.1.3 搜索匹配测试

测试项	测试结果	备注
关键词搜索	通过	支持模糊匹配
科目筛选	通过	支持多选
年级筛选	通过	支持多个年级
时薪范围筛选	通过	可自定义范围
附近搜索	通过	基于高德地图

11.1.4 在线沟通测试

在线沟通功能测试主要验证消息发送、接收和管理的完整性。测试结果显示，系统能够稳定支持文字消息和图片消息的发送与接收，实时推送消息内容，并正确显示消息的已读/未读状态。同时，聊天记录能够安全存储在云端，支持离线消息推送，确保用户不会错过重要信息。

测试项	测试结果	备注
文字消息发送	通过	实时推送

测试项	测试结果	备注
图片消息发送	通过	支持图片预览
消息已读状态	通过	显示已读/未读
聊天记录保存	通过	云端存储
消息通知	通过	离线消息推送

11.1.5 预约管理测试

预约管理功能测试验证了从报名申请到预约完成的全流程。测试结果表明，系统能够成功处理报名申请的发起、接受和拒绝操作，并自动创建预约记录。预约状态的变更会及时通知相关用户，确保信息的及时同步。

测试项	测试结果	备注
发起报名申请	通过	支持填写申请理由
接受报名申请	通过	自动创建预约
拒绝报名申请	通过	可填写拒绝原因
确认预约	通过	双方收到通知
取消预约	通过	双方收到通知

11.2 性能测试

性能测试主要评估系统在不同负载下的响应速度和稳定性，确保系统能够满足用户的使用需求。

11.2.1 响应时间测试

响应时间测试验证了系统在各种操作场景下的处理速度。测试结果显示，系统在首页加载、登录操作、信息发布、搜索操作和消息发送等场景下均能保持较快的响应速度，满足用户的使用体验需求。

测试场景	平均响应时间	最大响应时间	备注
首页加载	1.2 秒	2.5 秒	包含数据请求
登录操作	0.8 秒	1.5 秒	包含验证码验证
信息发布	1.5 秒	3.0 秒	包含图片上传
搜索操作	0.9 秒	2.0 秒	包含筛选条件
消息发送	0.5 秒	1.0 秒	实时通信

11.2.2 并发测试

并发测试评估了系统在多用户同时访问时的稳定性和性能表现。测试结果表明，系统在 1000 用户并发访问时仍能基本保持稳定，在 2000 用户并发时虽然压力较大，但仍能处理大部分请求。

测试用户数	系统稳定性	平均响应时间	备注
100	稳定	1.2 秒	无错误
500	稳定	1.8 秒	无错误
1000	基本稳定	2.5 秒	个别请求延迟
2000	压力较大	4.0 秒	部分请求超时

11.3 兼容性测试

兼容性测试验证了系统在不同设备和操作系统上的运行情况，确保系统能够广泛适配各种用户设备。

11.3.1 iOS 设备测试

iOS 设备测试验证了系统在不同型号和版本的 iOS 设备上的运行情况。测试结果显示，系统在 iPhone 13、iPhone 12、iPhone 11 和 iPhone X 等设备上均能正常运行，适配效果良好。

设备型号	iOS 版本	测试结果	备注
iPhone 13	iOS 15.0	通过	完美适配
iPhone 12	iOS 14.5	通过	完美适配
iPhone 11	iOS 14.0	通过	基本适配
iPhone X	iOS 13.5	通过	基本适配

11.3.2 Android 设备测试

Android 设备测试验证了系统在不同品牌和版本的 Android 设备上的运行情况。测试结果表明，系统在小米、华为、OPPO、vivo 和三星等主流 Android 设备上均能正常运行，适配效果良好。

设备型号	Android 版本	测试结果	备注
小米 12	Android 12.0	通过	完美适配
华为 Mate 40	Android 11.0	通过	完美适配
OPPO Reno 6	Android 11.0	通过	基本适配
vivo X60	Android 11.0	通过	基本适配

设备型号	Android 版本	测试结果	备注
三星 Galaxy S21	Android 11.0	通过	基本适配

十二、开发文档

开发文档详细描述了系统的技术实现、项目结构、API 接口设计和数据库设计等内容，为系统的维护和扩展提供了技术参考。

12.1 项目结构

项目采用前后端分离的架构设计，前端使用 Flutter 框架开发跨平台移动应用，后端使用 Spring Boot 框架提供 RESTful API 服务。

12.1.1 前端项目结构

前端项目使用 Flutter 框架开发，采用模块化的目录结构，便于代码的组织和管理。主要包含应用入口、路由配置、数据模型、页面组件、服务层、状态管理和工具类等模块。

```
client/
└── lib/
    ├── main.dart                      # 应用入口
    ├── routes.dart                    # 路由配置
    └── models/
        ├── user_model.dart
        ├── tutor_model.dart
        ├── student_request_model.dart
        └── message_model.dart
    └── pages/                          # 页面组件
        ├── auth/                         # 认证相关页面
        │   ├── login_page.dart
        │   ├── register_page.dart
        │   └── role_selection_page.dart
        ├── main/                          # 主页面
        │   ├── home_page.dart
        │   ├── discover_page.dart
        │   ├── message_page.dart
        │   └── profile_page.dart
        ├── publish/                      # 发布页面
        │   ├── publish_tutor_service_page.dart
        │   └── publish_student_request_page.dart
        └── chat/                          # 聊天页面
            └── chat_detail_page.dart
    └── settings/                     # 设置页面
        └── settings_page.dart
    └── services/                     # 服务层
```

```
    └── api_service.dart      # API 接口调用
    └── socket_service.dart  # WebSocket 服务
    └── storage_service.dart # 本地存储服务
  └── providers/           # 状态管理
    └── user_provider.dart
    └── tutor_provider.dart
    └── message_provider.dart
  └── utils/                # 工具类
    └── http_util.dart
    └── location_util.dart
    └── image_util.dart
  └── assets/               # 静态资源
    └── images/              # 图片资源
    └── fonts/               # 字体资源
  └── pubspec.yaml          # 依赖配置
  └── README.md             # 项目说明
```

12.1.2 后端项目结构

```
hitutor-backend/
└── src/
  └── main/
    └── java/com/hitutor/
      └── controller/      # 控制器层
        └── UserController.java
        └── TutorController.java
        └── StudentRequestController.java
        └── MessageController.java
        └── AppointmentController.java
      └── service/          # 服务层
        └── UserService.java
        └── TutorService.java
        └── StudentRequestService.java
        └── MessageService.java
        └── AppointmentService.java
      └── mapper/            # 数据访问层
        └── UserMapper.java
        └── TutorMapper.java
        └── StudentRequestMapper.java
        └── MessageMapper.java
        └── AppointmentMapper.java
      └── model/             # 数据模型
        └── User.java
        └── Tutor.java
        └── StudentRequest.java
        └── Message.java
        └── Appointment.java
      └── config/            # 配置类
```

```
|- config/
|  |- WebSocketConfig.java
|  |- SecurityConfig.java
|  |- CorsConfig.java
|- util/          # 工具类
|  |- JwtUtil.java
|  |- PasswordUtil.java
|  |- SmsUtil.java
|- dto/           # 数据传输对象
|  |- LoginDto.java
|  |- RegisterDto.java
|  |- MessageDto.java
|- resources/
|  |- application.yml    # 应用配置
|  |- mapper/            # MyBatis 映射文件
|- test/           # 测试代码
|- pom.xml         # Maven 配置
|- README.md       # 项目说明
```

12.2 关键技术实现

12.2.1 用户认证实现

1. JWT 令牌生成

- 使用 HS256 算法生成 JWT 令牌
- 令牌包含用户 ID、角色和过期时间
- 令牌存储在客户端本地

2. 密码加密

- 使用 BCrypt 算法对密码进行加密
- 加密过程中加入随机盐值
- 验证时使用相同算法进行比对

3. 权限控制

- 基于角色的权限控制（RBAC）
- 使用 Spring Security 实现权限拦截
- 不同角色拥有不同的操作权限

12.2.2 实时通信实现

1. WebSocket 连接

- 使用 Spring WebSocket 实现服务端
- 使用 dart:io 库实现客户端
- 建立长连接实现消息实时推送

2. 消息处理

- 消息发送到服务端后，通过 WebSocket 推送给接收方
- 消息存储到数据库，支持历史消息查询
- 实现消息已读状态管理

3. 连接管理

- 维护用户与 WebSocket 连接的映射关系
- 实现心跳检测，保持连接稳定
- 处理连接断开和重连机制

12.2.3 地理位置服务实现

1. 定位功能

- 使用 Geolocator 插件获取用户位置
- 支持 GPS 定位和网络定位
- 处理定位权限请求

2. 地图展示

- 使用 AMap Flutter Map 插件显示地图
- 支持地图缩放、拖动和标记
- 显示用户当前位置和附近的家教/需求

3. 距离计算

- 使用 Haversine 公式计算两点之间的距离
- 基于距离对搜索结果进行排序
- 支持自定义搜索半径

12.3 数据库设计

12.3.1 用户表 (user)

字段名	数据类型	约束	描述
id	BIGINT	PRIMARY KEY	用户 ID
phone	VARCHAR(20)	UNIQUE NOT NULL	手机号码
password	VARCHAR(100)	NOT NULL	密码 (加密)
nickname	VARCHAR(50)	NOT NULL	昵称
avatar	VARCHAR(255)		头像 URL
gender	TINYINT		性别 (0-未知, 1-男, 2-女)
birthday	DATE		出生日期
role	TINYINT	NOT NULL	角色 (1-学生和家长, 2-家教老师)
real_name	VARCHAR(50)		真实姓名
id_card	VARCHAR(20)		身份证号
status	TINYINT	NOT NULL	状态 (0-未认证, 1-已认证, 2-认证中)
create_time	DATETIME	NOT NULL	创建时间
update_time	DATETIME	NOT NULL	更新时间

12.3.2 家教信息表 (tutor)

字段名	数据类型	约束	描述
id	BIGINT	PRIMARY KEY	家教 ID
user_id	BIGINT	FOREIGN KEY	用户 ID
subject	VARCHAR(100)	NOT NULL	授课科目
price	DECIMAL(10,2)	NOT NULL	时薪价格
location	VARCHAR(100)	NOT NULL	授课地点
description	TEXT		个人描述
teaching_experience	TEXT		教学经验
education_background	VARCHAR(100)		教育背景
certifications	VARCHAR(255)		资质证书 URL
status	TINYINT	NOT NULL	状态 (0-待审核, 1-已发布, 2-已下架)

字段名	数据类型	约束	描述
view_count	INT	NOT NULL	浏览次数
create_time	DATETIME	NOT NULL	创建时间
update_time	DATETIME	NOT NULL	更新时间

12.3.3 学生需求表 (*student_request*)

字段名	数据类型	约束	描述
id	BIGINT	PRIMARY KEY	需求 ID
user_id	BIGINT	FOREIGN KEY	用户 ID
subject	VARCHAR(100)	NOT NULL	辅导科目
grade	VARCHAR(50)	NOT NULL	学生年级
price_min	DECIMAL(10,2)	NOT NULL	最低时薪
price_max	DECIMAL(10,2)	NOT NULL	最高时薪
location	VARCHAR(100)	NOT NULL	上课地点
description	TEXT		需求描述
status	TINYINT	NOT NULL	状态 (0-待审核, 1-已发布, 2-已下架)
view_count	INT	NOT NULL	浏览次数
create_time	DATETIME	NOT NULL	创建时间
update_time	DATETIME	NOT NULL	更新时间

12.3.4 消息表 (*message*)

字段名	数据类型	约束	描述
id	BIGINT	PRIMARY KEY	消息 ID
send_user_id	BIGINT	FOREIGN KEY	发送用户 ID
receive_user_id	BIGINT	FOREIGN KEY	接收用户 ID
content	TEXT	NOT NULL	消息内容
type	TINYINT	NOT NULL	消息类型 (1-文字, 2-图片)
status	TINYINT	NOT NULL	状态 (0-未读, 1-已读)
create_time	DATETIME	NOT NULL	创建时间

12.3.5 预约表 (*appointment*)

字段名	数据类型	约束	描述
id	BIGINT	PRIMARY	预约 ID

字段名	数据类型	约束	描述
		KEY	
tutor_id	BIGINT	FOREIGN KEY	家教 ID
student_id	BIGINT	FOREIGN KEY	学生用户 ID
student_request_id	BIGINT	FOREIGN KEY	学生需求 ID
appointment_time	DATETIME	NOT NULL	预约时间
status	TINYINT	NOT NULL	状态 (0-待确认, 1-已确认, 2-已完成, 3-已取消)
create_time	DATETIME	NOT NULL	创建时间
update_time	DATETIME	NOT NULL	更新时间

十三、项目管理

13.1 开发流程

1. 需求分析

- 收集用户需求
- 分析功能可行性
- 制定需求文档

2. 设计阶段

- 系统架构设计
- 数据库设计
- 界面设计
- API 接口设计

3. 开发阶段

- 前端开发
- 后端开发
- 数据库实现
- 接口联调

4. 测试阶段

- 功能测试
- 性能测试
- 兼容性测试
- 安全性测试

5. 部署阶段

- 服务端部署
- 客户端发布
- 监控配置
- 故障演练

13.2 版本控制

- 使用 Git 进行版本控制
- 采用分支管理策略：
 - main：主分支，用于发布生产版本
 - develop：开发分支，用于集成功能
 - feature/*：功能分支，用于开发新功能
 - bugfix/*：bug 修复分支，用于修复生产问题

13.3 代码规范

13.3.1 前端代码规范

- 命名规范：使用驼峰命名法
- 缩进规范：使用 2 个空格缩进
- 代码风格：遵循 Dart 官方风格指南
- 注释规范：关键代码必须添加注释
- 文件结构：按功能模块组织文件

13.3.2 后端代码规范

- 命名规范：类名使用大驼峰命名法，方法和变量使用小驼峰命名法
- 缩进规范：使用 4 个空格缩进
- 代码风格：遵循 Java 官方风格指南
- 注释规范：类和方法必须添加 Javadoc 注释
- 异常处理：统一异常处理机制

13.4 文档管理

- 技术文档：系统架构、数据库设计、API 接口文档
- 用户文档：用户手册、操作指南
- 开发文档：开发规范、部署文档
- 测试文档：测试计划、测试用例、测试报告

十四、总结

HiTutor 好会帮家教信息对接平台是一个功能完善、技术先进的移动应用系统，采用 Flutter 和 Spring Boot 技术栈实现。平台通过信息匹配、在线沟通、预约对接、评价反馈等全流程数字化信息服务，提升了家教信息对接的透明度和信息质量，降低了双方的沟通和匹配成本。

系统具有以下核心优势：

1. **技术架构先进**：采用前后端分离架构，使用 Flutter 实现跨平台开发，Spring Boot 提供高效的后端服务
2. **功能完善**：涵盖用户认证、信息发布、智能搜索、在线沟通、预约管理、评价反馈等全流程功能
3. **用户体验优秀**：界面设计美观大方，操作流程简洁明了，响应速度快
4. **安全性高**：采用 JWT 认证、密码加密、HTTPS 传输等多种安全措施
5. **可扩展性强**：模块化设计，易于添加新功能和进行系统升级

平台的开发和部署严格遵循软件工程规范，确保了系统的稳定性和可靠性。通过本项目的实施，我们成功构建了一个纯公益的家教信息对接共享平台，为学生和家教老师提供了便捷、高效、安全的信息对接服务，取得了良好的社会效益。