

# 大学生选修课评价系统——评课菌

## 软件需求规格说明书

日期：2025.10.21

## 文档变更历史记录

[illegible]

# 目录

1. 引言 .....	4
1.1 编写目的 .....	4
1.2 读者对象 .....	4
1.3 软件项目概述 .....	4
1.4 文档概述 .....	4
1.5 定义 .....	5
1.6 参考资料 .....	6
2. 软件的一般性描述 .....	7
2.1 软件产品与其环境之间的关系 .....	7
2.2 限制与约束 .....	8
2.3 假设与前提条件 .....	9
3. 软件功能需求描述 .....	10
3.1 软件功能概述 .....	10
3.2 软件需求的用例模型 .....	13
3.3 软件需求的分析模型 .....	16
4. 其它软件需求描述 .....	17
4.1 性能要求 .....	17
4.2 设计约束 .....	17
4.3 界面要求 .....	18
4.4 进度要求 .....	19
4.5 交付要求 .....	19
4.6 验收要求 .....	20
5. 软件原型 .....	20

# 1. 引言

## 1.1 编写目的

本文档旨在明确“大学生选修课智慧评分系统”的功能需求、性能要求、设计约束等核心内容，为开发团队提供清晰的开发依据，为项目验收提供客观标准，同时确保项目组与潜在用户对系统功能达成共识。

## 1.2 读者对象

本文档的读者包括：项目研发小组全体成员（项目经理、技术总监、研发工程师、测试工程师、美工）、项目管理与验收人员（指导老师）、项目用户代表。

## 1.3 软件项目概述

**项目名称：**大学生选修课智慧评分系统（代号：评课菌）

**用户单位：**本校全体在校大学生

**开发单位：**旺铺招组

**大致功能和用途：**本系统是一款基于微信小程序的课程评价与查询平台。主要功能包括：允许学生匿名对已修读课程进行评价、打分和分享；提供强大的课程搜索与筛选功能，方便学生快速查找课程信息及其历史评价；通过数据可视化展示课程的综合评分，辅助大学生做出更科学、更个性化的选课决策。

## 1.4 文档概述

本文档首先在“引言”中明确文档目的、读者对象及项目基础信息；随后通过“软件的一般性描述”宏观分析产品与外部环境的关系，并界定开发与使用的限制约束及前提假设；进而，在核心章节“软件功能需求描述”中，详细说明功能概述、用例模型及分析模型，以精确界定各项功能需求；最后，在“其它软件需求描述”中全面补充性能、设计约束、界面、进度、交付及验收等非功能性要求，确保需求定义的完整性与一致性，从而为项目的设计、开发、测试与验收提供清晰、可靠的依据。

## 1.5 定义

### 1.5.1 业务角色与核心对象

**学生用户：**本系统的服务对象，特指通过“微信授权 + 学校学籍邮箱验证”完成身份认证的在校大学生。拥有课程搜索、评分、发布匿名评价/标签、参与讨论、查看榜单、收藏课程等权限。

**管理员：**负责系统运营与内容合规的角色，通过后台管理系统执行以下操作：审核用户发布的评价/标签/讨论内容、维护课程/教师基础信息、统计系统数据、处理恶意用户（如封禁账号）。

**基础课程库：**系统存储的课程基础信息集合，初始数据来源于学校教务系统导入或人工录入，包含课程名称、任课教师、开设学院、课程类型、学分等字段，需与学校官方数据保持一致。

**可视化榜单：**系统按预设规则（如评价数量、评分均值、标签热度）自动生成的课程排名模块，包含“全校最受欢迎课程榜”“学院给分最慷慨课程”等多维度榜单，支持按全校/学院/专业筛选。

**自由讨论区：**学生用户交流课程信息的功能模块，分为两部分，一是评价区：用户发布与星级评分、标签关联的文字形式上课体验；一是问答区：用户提出选课疑问并接收其他用户的文字回复。

**个人中心：**学生用户管理个人数据的功能入口，支持查看 / 编辑 / 删除本人发布的评价、标签及讨论内容，提供“课程收藏”功能（可将感兴趣课程加入列表）。

### 1.5.2 技术相关术语

**微信小程序：**本系统的运行载体，依赖微信客户端启动，需遵循微信小程序开发规范（如代码包大小 $\leq 2\text{MB}$ ），通过调用微信开放接口（wx.login、wx.request等）实现登录、网络请求等功能。

**API 通信：**小程序前端与后端服务器的交互方式，基于 HTTPS 协议传输数据，后端通过 API 接口处理前端请求（如用户认证、评价数据读写），保障功能实现与数据安全。

**关系型数据库：**系统可选的数据存储方案之一，用于持久化存储用户信息、课程数据、评价内容等业务数据，需满足选课高峰期的读写性能需求（响应时间

≤2 秒)。

**云数据库：**系统可选的数据存储方案之一，功能与关系型数据库一致，优势为无需自建服务器、降低初期运维成本，需结合项目资源约束选择。

**后台管理系统：**管理员专属操作界面，与学生用户使用的“小程序前端”完全隔离，支持内容审核、课程信息管理、数据统计等管理类操作。

### 1.5.3 定性描述术语

**选课高峰期：**系统并发访问量显著升高的时间段，特指学校官方规定的“学生选修课报名周期”。此场景下需满足：页面加载时间≤3 秒，核心 API 接口响应时间≤2 秒。

**高优先级功能：**对系统核心目标起关键支撑作用的功能，即文档中标注“最高”或“高”优先级的功能（如身份认证、课程评分、榜单展示），需优先完成开发与测试。

**中优先级功能：**提升用户体验但非必需的功能，即文档中标注“中”优先级的功能（如个人中心的评价编辑、课程收藏），可在核心功能完成后推进。

**有效数据：**支撑榜单参考价值的评价数据，指未被管理员屏蔽/删除的真实、合规的课程评价与标签，需满足“数量充足”的前提以反映课程真实口碑。

**恶意用户：**多次发布违规内容（如恶意刷屏、侮辱性评价）或被举报且核实违规的学生用户，管理员可对其执行账号封禁操作。

### 1.5.4 首字母缩写词和缩略语

**MVP：**Minimum Viable Product（最小可行产品），指本项目 4 个月内需完成的核心功能集合，确保上线后满足用户评价课程、参考选课的核心需求。

**API：**Application Programming Interface（应用程序编程接口），小程序前端与后端服务器的交互接口，是实现登录、课程查询、评价提交等功能的技术桥梁。

## 1.6 参考资料

《微信小程序开发指南》，腾讯科技（深圳）有限公司，2025 年更新版。

## 2. 软件的一般性描述

### 2.1 软件产品与其环境之间的关系

本产品“大学生选修课评分评价系统”是一款基于微信平台的轻量级应用（小程序）。其运行环境与外部实体的关系如下图所示，核心关系描述如下：

#### 1. 与用户（大学生）的关系：

用户通过个人学生账号授权登录本系统，是本系统最主要的使用者和服务对象。

用户通过微信客户端与小程序的**前端界面**进行交互，完成查看课程、发布评价、浏览排行榜等操作。

#### 2. 与微信生态的关系：

小程序完全依赖于**微信客户端**作为其运行环境，遵循微信小程序的开发规范和技术约束。

调用微信提供的**开放接口**，如 `wx.login`（微信登录）、`wx.request`（网络请求）、`wx.getUserProfile`（获取用户信息）等，以实现核心功能。

#### 3. 与后端服务器的关系：

小程序的**前端逻辑**与自主研发的**后端服务器**通过 HTTPS 协议进行 API 通信。

后端服务器负责处理核心业务逻辑，包括用户认证、评价数据的存储与读取、排行榜的计算与生成等。

#### 4. 与数据存储环境的关系：

后端服务器与**数据库**进行交互，将所有业务数据（用户信息、课程数据、评价内容、标签等）进行持久化存储。

## 5. 与外部数据源的关系：

在初始阶段，本系统需要从学校**官方教务系统**（或通过人工录入）导入基础的课程列表、教师信息等数据，以构建评价的客体。本系统与教务系统为**单向数据依赖关系**，不向其回写数据。

## 2.2 限制与约束

本软件的开发与运行受到以下限制与约束：

### 1. 技术约束：

必须完全遵守**微信小程序的开发规范**，包括但不限于代码包大小限制（2MB 以内，分包加载后总大小有限）、指定的 API 使用方式、界面设计指南等。

前端与后端的通信必须使用 **HTTPS 协议** 以确保数据传输安全。

数据库选择需考虑成本与性能，初期可能采用关系型数据库（如 MySQL）或云数据库。

### 2. 性能约束：

在选课高峰期等并发访问场景下，系统应能保持稳定，页面响应时间应在 3 秒以内，核心 API 接口响应时间应在 2 秒以内。

排行榜数据更新频率需明确，例如采用定时任务（如每 24 小时）更新，而非实时计算，以平衡服务器压力与数据新鲜度。

### 3. 法律与合规约束：



必须严格遵守《网络安全法》和《个人信息保护法》，对用户的个人信息和评价数据予以严格保密。

系统需建立内容审核机制，对用户提交的评价内容进行敏感词过滤和人工审核，防止不当言论的传播，并建立举报与处理流程。

#### 4. 资源约束：

项目开发周期为 4 个月，团队规模为 5 人，需在此人力与时间预算内完成最小可行产品的开发与上线。

## 2.3 假设与前提条件

本项目的成功开发与有效运行基于以下假设与前提：

#### 1. 用户与环境假设：

假设目标用户群体（在校大学生）拥有智能手机并熟练使用微信及小程序。

假设大学校园内有稳定且覆盖广泛的 Wi-Fi 或移动数据网络，保障小程序的正常访问。

#### 2. 数据与内容假设：

假设能够通过合法合规的途径获取到准确、完整的基础课程库（课程名称、授课教师等）。

假设用户群体有足够的意愿分享真实的课程评价，能够产生足够数量的有效数据以支撑排行榜的参考价值。

#### 3. 开发与运维假设：

假设开发团队能够熟练掌握微信小程序开发、后端 API 开发及数据库管理等关键技术。

假设项目上线后，有持续的、低成本的服务器和域名等基础设施资源支持系统运行。

假设在项目初期，内容审核等工作可由开发团队或学生志愿者人工完成，暂不依赖全自动的 AI 审核系统。

### 3. 软件功能需求描述

#### 3.1 软件功能概述

“评课菌”小程序以“课程评价的量化表达与互动交流”为核心设计目标，围绕评分系统、自由讨论区、数据可视化三大模块设计，同时兼备多种基础辅助功能。

详细功能描述如下表格：

功能标识	功能描述	优先级	使用场景
身份登录	微信一键授权，再用学校学籍邮箱验证，确保是本校学生；评价时不显示真实名字，保护隐私。	最高	第一次用小程序，需要注册登录才能发评价、看详细内容
课程搜索与筛选	能按课程名、老师名搜课，也能按学院（如文学院、计算机学院）、课程类型（如艺术类、技术类）筛课，快速找到目标课程。	高	想选“摄影”相关课，直接搜“摄影”；或只想看本学院的选修课，用学院筛选

功能标识	功能描述	优先级	使用场景
课程评分与标签化	给课程打 1-5 星（1 星差、5 星好），还能加标签（比如“作业少”“给分高”“考试简单”）；看到别人的标签觉得对，能点赞，点赞多的标签会更显眼。	最高	上完一门课，想分享感受：给 4 星，加“老师讲课有趣”“期末考论文”标签；看某门课的特点时，直接看高赞标签
自由讨论区	分“评价区”和“问答区”：评价区可写文字分享上课体验（比如“每周要交一次小作业，不难但要及时”），和星级、标签一起展示；问答区可提问（比如“这门课需要有编程基础吗”），上过课的同学能回复。	高	选课前想了解细节，翻评价区看大家的上课体验；有疑问时在问答区提问，等学长学姐解答
可视化榜单	自动生成榜单，比如“全校最受欢迎课程”“XX 学院给分最慷慨课程”，还能按全校、学院、专业筛选。	高	不知道选什么课，看“最受欢迎课程榜”找热门好课；想快速了解某门课的核心特点，看标签墙

功能标识	功能描述	优先级	使用场景
个人中心	能看自己发过的评价、标签，想改的话可以编辑；遇到感兴趣的课，能收藏起来，后续方便找。	中	想改之前发的评价（比如漏了说“期中有小测”），去个人中心编辑；看到“短视频制作”课很感兴趣，先收藏，之后再决定要不要选
内容审核（管理员用）	管理员能看大家发的评价、标签，遇到骂人、造谣的违规内容，可删除或屏蔽；还能管理课程和老师信息，确保和学校官方数据一致	高	有用户发“老师人品差”这种人身攻击的评价，管理员看到后屏蔽；学校新增了一门“AI 入门”课，管理员把课程信息录入系统
数据统计（管理员用）	管理员能看小程序的访问量、总评价数量、各课程的评价数等数据，还能控制榜单什么时候更新、展示哪些榜单	中	学校想了解本学期大家用小程序的情况，管理员看访问量和评价总数；期末选课季前，更新“本学期热门课程榜”给学生参考

### 3.2 软件需求的用例模型

用例名	执行者	概述
身份登录	学生用户	学生用户通过微信一键授权发起登录请求，再输入学校学籍邮箱完成身份验证，系统确认其为该校真实学生后允许使用核心功能；登录后所有评价操作均以匿名形式展示，保护用户隐私，此用例为学生用户使用其他课程相关功能的基础前置条件。
课程搜索与筛选	学生用户	学生用户可在系统内输入课程名称、教师姓名进行关键词搜索，也可选择“开设学院”（如文学院、计算机学院）、“课程类型”（如艺术类、技术类）等条件进行筛选，快速定位目标选修课，减少无效信息浏览时间。
课程评分与打标签	学生用户	学生用户针对已选课或了解的课程，进行 1-5 星评分（1 星代表体验差，5 星代表体验优）；可自定义添加课程标签（如“作业少”“给分高”“考试简单”“老师讲课有趣”），同时可对其他用户添加的标签进行点赞，系统会将高赞标签优先展示在课程详情页，帮助其他用户快速识别课程核心特点。

用例名	执行者	概述
自由讨论区互动	学生用户	<p>学生用户可在课程专属的自由讨论区进行两类操作：一是在“评价区”发布文字内容，分享上课细节（如“每周 1 次小作业，难度低但需按时提交”），内容将与该用户对课程的星级评分、标签关联展示；二是在“问答区”提出选课疑问（如“这门课需要编程基础吗”），已选课或已评分的学生用户可对疑问进行文字回复，形成互助交流。</p>
查看可视化榜单	学生用户	<p>学生用户可查看系统自动生成的多维度课程榜单，包括“全校最受欢迎课程榜”“XX 学院给分最慷慨课程榜”“课程难度榜”等，且榜单支持按“全校、学院、专业”筛选；同时可查看课程专属标签墙，点赞数越高的标签位置越居中、字号越大，直观获取课程热门评价点。</p>
个人中心管理	学生用户	<p>学生用户进入个人中心后，可查看自己历史发布的课程评分、标签及讨论内容，若内容有误可进行编辑或删除；遇到感兴趣但暂未决定的课程时，可点击“收藏”按钮将课程加入收藏列表，后续可直接在个人中心快速查找该课程。</p>

用例名	执行者	概述
内容审核	管理员	管理员登录系统后台后，可查看所有学生用户提交的课程评价、标签内容，对包含人身攻击（如“老师人品差”）、造谣（如“这门课必挂科”）等违规信息的内容，执行“屏蔽”或“删除”操作；审核通过的内容才会在学生端展示，保障社区内容合规性。
数据统计与榜单管理	管理员	管理员可在后台查看系统运营数据，包括小程序总访问量、各课程评价数量、用户活跃天数等；同时可配置可视化榜单的更新频率（如每周更新 1 次）、展示范围（如仅展示本学期课程），确保榜单数据时效性与实用性。
课程 / 教师信息管理	管理员	管理员需定期同步学校官方教务数据，在系统后台执行课程、教师信息的“新增、修改、删除”操作（如学校新增“AI 入门”课程时，管理员录入课程名称、任课教师、学分等基础信息），确保系统内课程与教师信息和学校官方数据一致，避免学生获取错误信息。

用例名	执行者	概述
恶意用户处理	管理员	当管理员发现某学生用户多次发布违规内容（如恶意刷屏、发布侮辱性评价），或收到其他用户对该用户的举报并核实后，可在后台对该用户执行“账号封禁”操作，封禁期间用户无法登录系统、发布内容，维护社区健康环境。

### 3.3 软件需求的分析模型

“评课菌”小程序功能及需求相互依赖关系如模式图图 1、图 2 所示，其中图 1 为执行者为学生的软件功能与交互模式，图 2 为执行者为管理员的软件功能与交互模式。

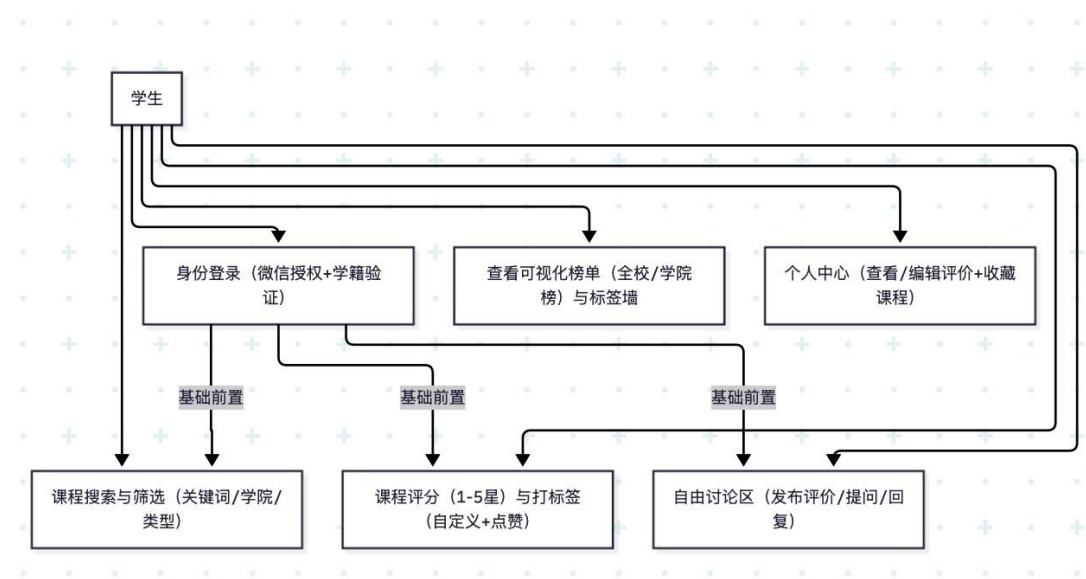


图 1





图 2

## 4. 其它软件需求描述

### 4.1 性能要求

**响应速度：**选课高峰期（学校选修课报名周期）内，页面加载时间正常，核心 API 接口（如课程搜索、评价提交、榜单加载）响应时间正常；非高峰期页面加载时间正常，核心 API 接口响应时间正常。

**数据处理：**课程评分、标签点赞等数据提交后，实时更新至数据库；可视化榜单按预设规则定时更新，更新过程不影响用户正常使用系统。

**兼容性：**适配微信客户端 iOS 版本 13.0 及以上、Android 版本 8.0 及以上，支持主流手机屏幕尺寸，界面布局无错乱、功能无失效。

### 4.2 设计约束

#### 1. 开发工具与技术栈

前端使用微信小程序原生开发框架（WXML + WXSS + JavaScript），可复用 UI 组件库。

#### 2. 安全性

用户身份认证需结合“微信授权 + 学校学籍邮箱验证”，确保仅本校在校学生可使用系统；评价内容默认匿名展示，不泄露用户真实信息。

实现敏感词过滤机制，对用户发布的评价、标签、讨论内容进行实时检测，屏蔽侮辱性、攻击性、违法违规词汇；建立内容审核日志，记录管理员操作（如删除、屏蔽内容）。

数据库定期备份，防止数据丢失；禁止存储用户明文密码等敏感数据，采用加密算法处理关键信息。

### 3. 可靠性

系统全年可用率稳定，除计划性维护外，无突发宕机情况

## 4.3 界面要求

### 1. 整体风格：

采用简约、清新的校园风格，主色调以蓝色和白色为主，字体统一使用微信默认字体，字号确保阅读清晰。

### 2. 核心页面设计：

#### 首页：

顶部设置搜索框，下方为快捷功能和可能感兴趣的贴文。

#### 排行榜：

顶部设置搜索框（支持课程名、教师名关键词搜索），下方为“热门榜单”，支持按“学院”“专业”筛选切换。

#### 讨论区：

展示课程基础信息（课程名、教师、学院、学分），下方分“评价区”和“问答区”，支持切换查看。

评价提交页：包含 1-5 星评分选项（点击对应星级高亮）、标签输入框（支持自定义标签）、文字评价输入框，底部设置“提交”按钮。

#### 个人中心：

顶部展示用户头像（微信授权获取）及昵称，下方菜单包含“我的帖子”“我的收藏”“设置”；“我的帖子”支持查看、编辑、删除本人发布的内容，“我的收藏”展示已收藏课程列表，支持快速跳转课程详情页。

### 3. 交互体验：

评价提交、收藏课程等操作成功后，显示简短提示（如“评价提交成功”）；页面跳转采用平滑过渡动画，避免生硬切换。

## 4.4 进度要求

各阶段进度要求如下：

**需求调研（第 1-4 周）：**完成校园选修课评价需求调研，输出需求调研报告；确定项目技术方案、团队分工，完成项目计划制定。

**原型设计（第 5-8 周）：**完成小程序 UI 设计稿（含首页、课程详情页、评价提交页等核心页面）及可交互原型，组织内部评审并根据反馈优化设计方案。

**开发实现（第 9-11 周）：**完成前端小程序核心功能开发（身份登录、课程搜索、评价提交、榜单展示等）、后端 API 接口开发及数据库搭建，实现前后端联调，确保核心功能可正常运行。

**测试优化（第 12-13 周）：**完成功能测试（覆盖所有核心用例）、兼容性测试（适配不同手机型号与微信版本）、性能测试（验证高峰期响应速度与并发能力），修复测试发现的漏洞，优化页面加载速度与用户体验。

**验收（第 14-16 周）：**课堂展示

## 4.5 交付要求

### 1. 交付内容

软件产品：可正常运行的“评课菌”微信小程序

源代码与配置文件

项目文档：包含需求相关文档、设计相关文档、测试相关文档等

### 2. 交付形式

电子文件：所有源代码、配置文件上传至 Git 仓库；项目文档以 PDF 格式汇总，发送至指定邮箱。

打印材料：软件需求规格说明书、项目总结报告各装订成册后交付。

线上演示：项目验收时，团队需现场演示小程序核心功能（如身份登录、课程评价、榜单查看等）及管理员后台操作，确保功能符合需求要求。

## 4.6 验收要求

以本《软件需求规格说明书》为依据，结合项目计划、UI 设计稿、测试用例等验证软件产品是否满足需求描述的功能、性能、设计约束等要求。

### 1. 功能验收：

核心功能（身份登录、课程搜索与筛选、课程评分与标签、自由讨论区、可视化榜单、个人中心、管理员内容审核）需 100% 实现，无功能缺失或失效。

身份认证需准确识别本校学生，非本校学生无法通过验证；评价内容需经过审核后展示，违规内容可被管理员有效屏蔽或删除；榜单数据计算准确，与实际评价数据一致。

### 2. 性能验收：

页面加载时间合格，界面无错乱、功能无异常；系统无宕机、数据丢失情况。

### 3. 文档验收：

交付的项目文档需完整、规范，内容与实际开发情况一致；文档格式统一，无明显错别字或逻辑错误；用户手册、管理员手册需清晰易懂，可指导用户正常操作。

## 5. 软件原型

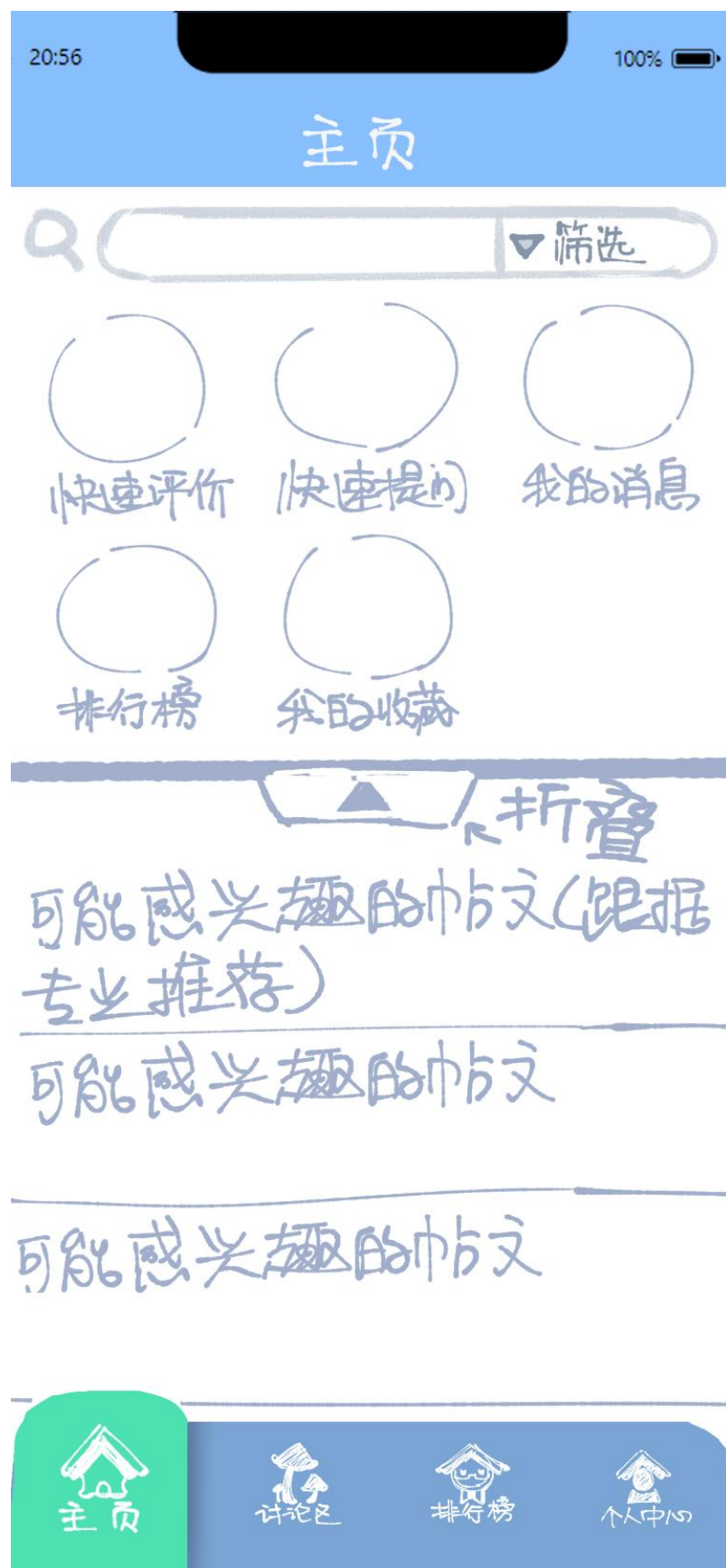


图 3 首页



图 4 讨论区

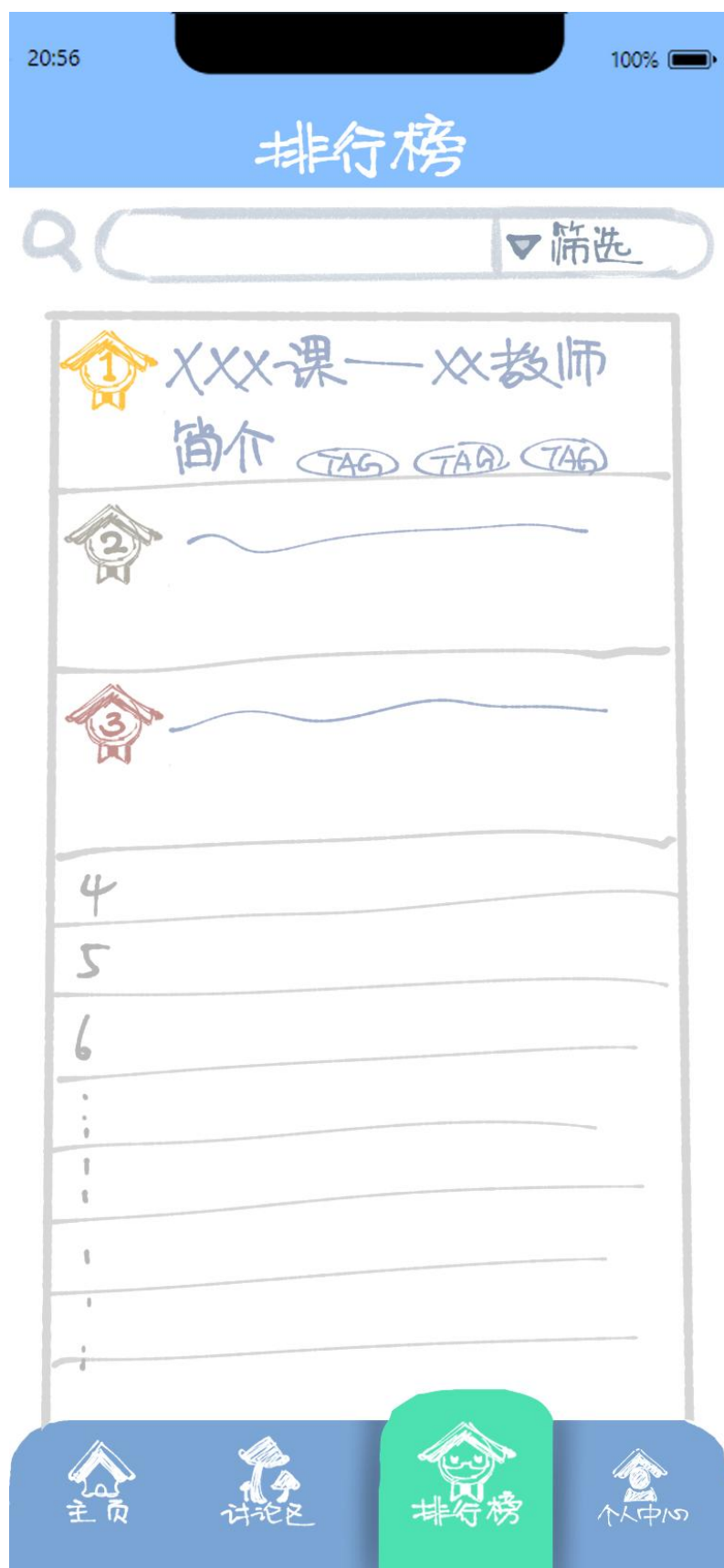


图 5 排行榜

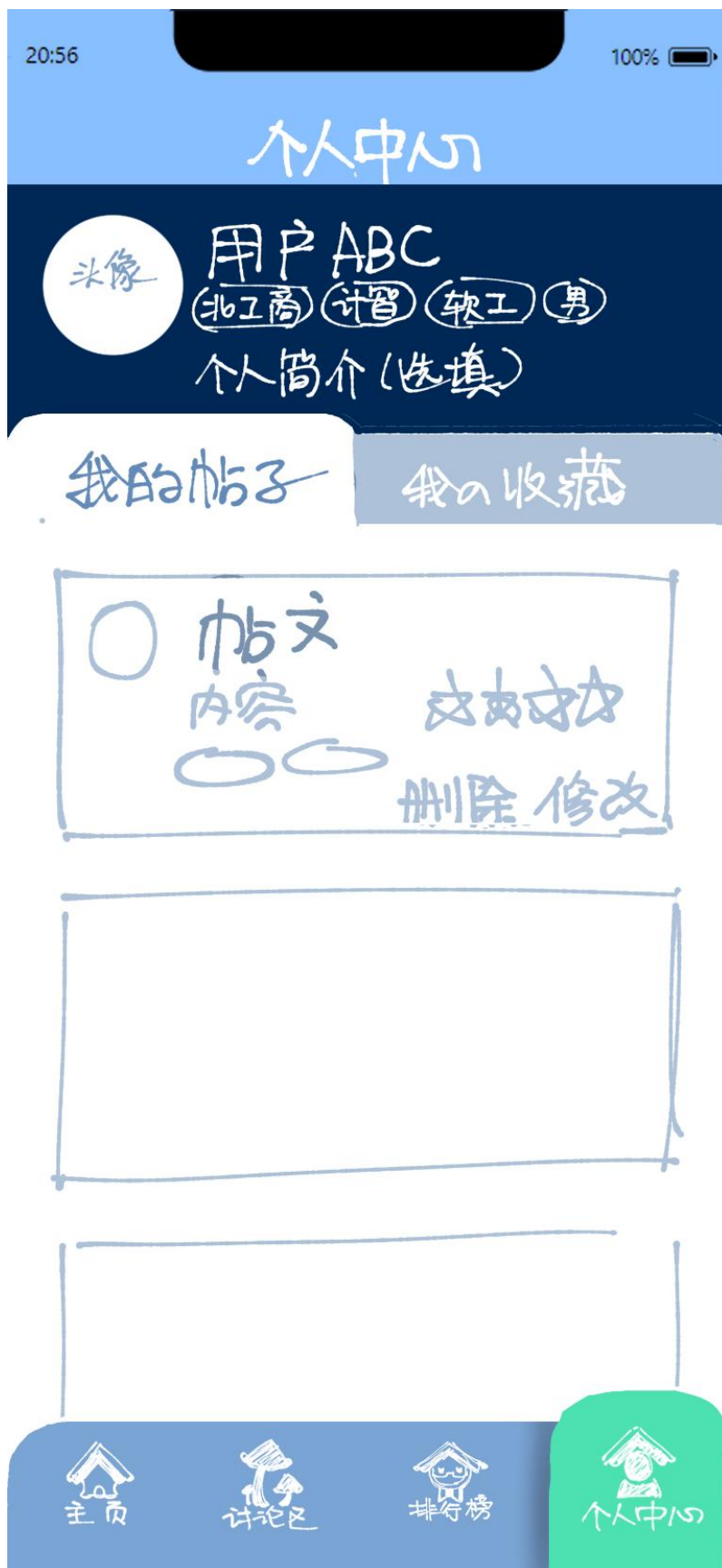


图 6 个人中心





图 7 评价区



图 8 问答区