

# 哈尔滨师范大学 学士学位论文

哈尔滨师范大学

2022 年 5 月

# 目 录

1 项目概述.....	1
1.1 目的和意义.....	1
1.2 国内外发展现状.....	1
1.3 课题背景.....	1
1.4 研究内容.....	2
2 需求分析和用例.....	3
2.1 需求分析.....	3
2.2 需求用例.....	3
3 核心业务模块设计.....	6
3.1 架构设计.....	6
3.2 用户账户模块.....	8
3.3 教务平台交互模块.....	13
3.4 任务清单模块.....	15
3.5 校园地图模块.....	16
3.6 超级管理员设备鉴权模块.....	17
3.7 数据统计模块.....	20
3.8 测试用户鉴权模块.....	21
4 系统实现.....	25
4.1 Android APP 实现.....	25
4.2 服务端核心接口实现.....	29
4.3 数据库建设.....	35
4.4 系统部分功能实现效果.....	36
5. 系统运行与测试.....	40
5.1 运行环境配置.....	40
5.2 项目运行数据.....	40
5.3 系统测试.....	41
6 结论.....	43
参考文献.....	44

# 微哈师校园生活服务平台的设计与实现

康广慧

**摘 要:** 由于我校学生人数众多,由此会有一些比较典型的问题,例如每到教务平台访问高峰期时教务平台服务器会有比较大的压力,且每次操作过程略繁琐。

微哈师校园生活服务平台以微哈师 QQ 小程序、微哈师 Android 客户端、管理员平台、超级管理员平台、全平台超级管理员授权工具客户端、超级管理员设备管理客户端六类用户平台为载体,服务于普通用户、班干部、班级管理员、超级管理员、项目 Owner 五种用户角色,使用较多的先进技术实现了用户部分课程表查询、成绩查询、校园地图、任务清单和管理部分身份鉴权、用户数据统计、通知和问卷的发布与管理、轮播管理等共四十余个功能点。

项目在长期的开发迭代和运营中积累了较为可观的用户量,成为了一个开发和运营体系较为完善的软件系统。

**关键词:** MVVM 架构; SpringBoot; Redis; 全栈项目

## 1 项目概述

### 1.1 目的和意义

由于我校学生人数众多，由此会有一些比较典型的问题，例如每到教务平台访问高峰期时教务平台服务器会有比较大的压力，且每次操作过程略繁琐<sup>[1]</sup>，为了解决这些问题，搭建微哈师校园生活服务平台，设计便捷的课程表、成绩数据查询和校园地图、任务清单等功能为用户提供更加便利和高效的校园信息服务。

在学校官方的平台查数据步骤较多且服务偶尔不稳定的情况下，微哈师校园生活服务平台可以缩短查询所需数据的操作路径以及通过本地存储分担官方平台访问压力。此外通过本项目也可以参与一个成体系的平台的开发和运营，有很强的实践意义。

### 1.2 国内外发展现状

随着移动生态的发展，市场上出现了非常多的校园主题的软件和小程序，目前用户量最多的是以“超级课程表 APP”为代表的综合性软件，可以查询很多学校的课程表、学业成绩、四六级成绩、普通话成绩，以及具有很多社交方面的功能；还有一类是针对一个学校开发的平台，比较典型的有“our 山大 QQ 小程序”、“山科小站微信小程序”等，开发者大多为在校的学生，对于目标用户的需求更加了解，从而开发出用户切实需要的功能。

国外具有同类功能的 APP 的实现方案是由学校官方接入相关的信息平台，学生使用信息平台的 APP，选择自己的行政区和学校进行登录和功能的使用。依照国外相关法律，第三方存储和传输相关数据会存在一定的法律风险。

### 1.3 课题背景

本项目起始于 2019 年 11 月，项目初始版本作为 2019 QQ 小程序应用开发大赛参赛作品，实现了每日打卡、任务清单、系统消息、课表查询和成绩查询五项核心功能。原始版本小程序使用第三方框架搭建 UI 界面，服务端使用 Java 和 PHP 后端技术实现。

2020 年初，在新版教务平台即将启用的背景下，项目进行了大幅度重构，先于其他第三方教务查询工具适配了新版教务平台，重新独立设计了小程序 UI 风格，新增了班级系统、问卷和任务清单设置提醒的功能，后端主要基于 SpringBoot 进行开发，开发语言由 PHP+Java 改为主要使用 Kotlin，仅在个别业务场景中使用 PHP，开发了管理员平台和超级管理员平台以及 Android 端和 Windows 端的授权工具。在后面的时间里，项目不断完善，教务功能解析算法不断优化。

2021 年 3 月微哈师 Android 客户端启动开发，经过历时 9 个月的开发，公测版本于 2021 年 12 月上架应用商店。

2022 年 2 月超级管理员设备权限验证机制升级，并随之开发了 iOS 和 Mac 端超级管理员授权工具，实现了针对全平台的超级管理员设备授权功能。本文对项目的介绍基于当前线上可用的功能和服务。

本项目 QQ 小程序、Android 客户端、服务端均已进行软件著作权登记，QQ 小程序登记号：2020SR1271518，Android 客户端登记号：2021SR1664477，服务端登记号：2021SR1664476。

#### **1.4 研究内容**

微哈师校园服务平台分为用户部分和管理部分，用户部分实现课程表查询、成绩查询、校园地图、任务清单等功能，管理部分实现身份鉴权、用户数据统计、通知和问卷的发布与管理、轮播管理等功能，项目整体涵盖 QQ 小程序、Android 客户端、Web 前端、Windows 客户端和 Java 后端等多个领域内的技术。QQ 小程序基于 QQ 小程序官方 API 开发，实现用户登录、绑定小程序、课程表查询、成绩查询、任务清单、系统消息等功能。Android 客户端使用 Android 官方提供的 JetPack 工具库中的相关组件基于 MVVM 架构实现用户登录、绑定小程序、课程表查询、成绩查询、校园地图、系统消息、社会化分享等功能。Web 前端基于 LayUI 框架实现数据概览、用户管理、测试授权、轮播管理、后台账户管理、通知发布、通知管理、问卷发布、问卷管理等页面。服务端基于 SpringBoot 框架，使用 MySQL 数据库和 Redis 数据库进行业务数据存储和访问，实现项目中所有的业务接口。

## 2 需求分析和用例

### 2.1 需求分析

微哈师校园生活服务平台包括五种用户角色和六类用户平台，五种用户角色包括：普通用户、班干部、班级管理员、超级管理员、项目 Owner，六类用户平台包括：微哈师 QQ 小程序、微哈师 Android 客户端、微哈师管理员平台、超级管理员平台、全平台超级管理员授权工具客户端、超级管理员设备管理客户端，针对各用户角色需要实现如下需求。

(1)普通用户可以使用 QQ 小程序和 Android 客户端中的课表查询、成绩查询、校园地图等基础功能进行课程数据查询、成绩数据查询、查阅校园地图、制定任务提醒等操作。

(2)班干部相对于普通用户拥有更高的权限，除基础功能外，班干部可以使用 QQ 小程序班级模块中的通知发布和问卷发布功能向班级成员发布通知或问卷。

(3)班级管理员可以使用管理员平台查看班级数据、管理班级成员、发布与管理班级通知和问卷。

(4)超级管理员可以使用超级管理员授权工具申请和验证超级管理员权限，使用超级管理员平台查看和管理整个项目的数据、修改项目配置。

(5)项目 Owner 可以使用超级管理员设备管理客户端控制超级管理员设备的权限。

### 2.2 需求用例

普通用户用例图如图 1 所示。

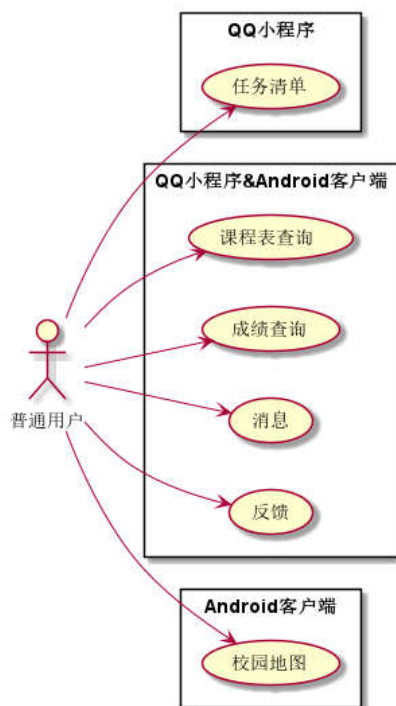


图 1 普通用户用例图

班干部用例图如图 2 所示。

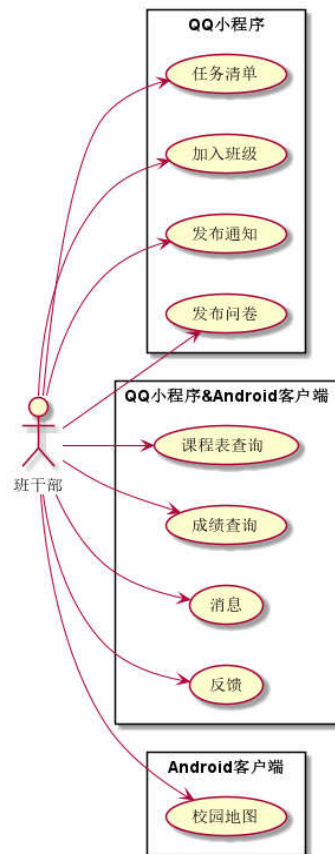


图 2 班干部用例图

班级管理员用例图如图 3 所示。

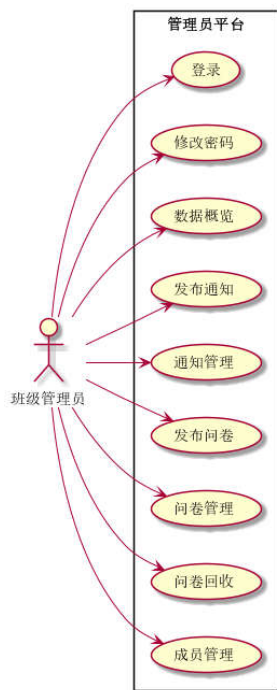


图 3 班级管理员用例图

超级管理员用例图如图 4 所示。

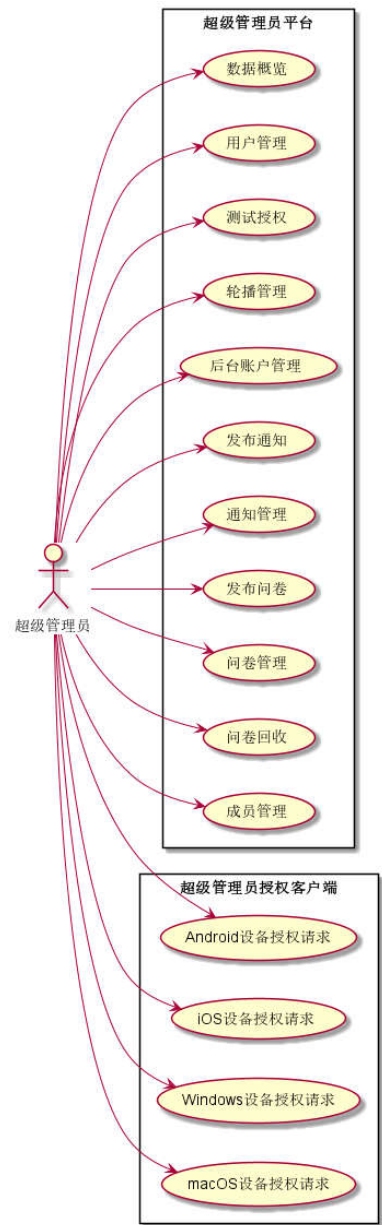


图 4 超级管理员用例图

项目 Owner 用例图如图 5 所示。

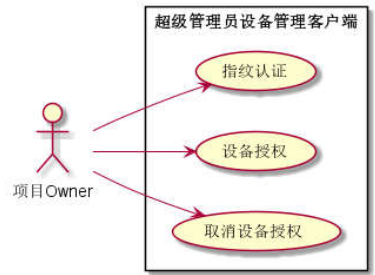


图 5 项目 Owner 用例图



### 3 核心业务模块设计

#### 3.1 架构设计

项目业务架构图如图 6 所示。

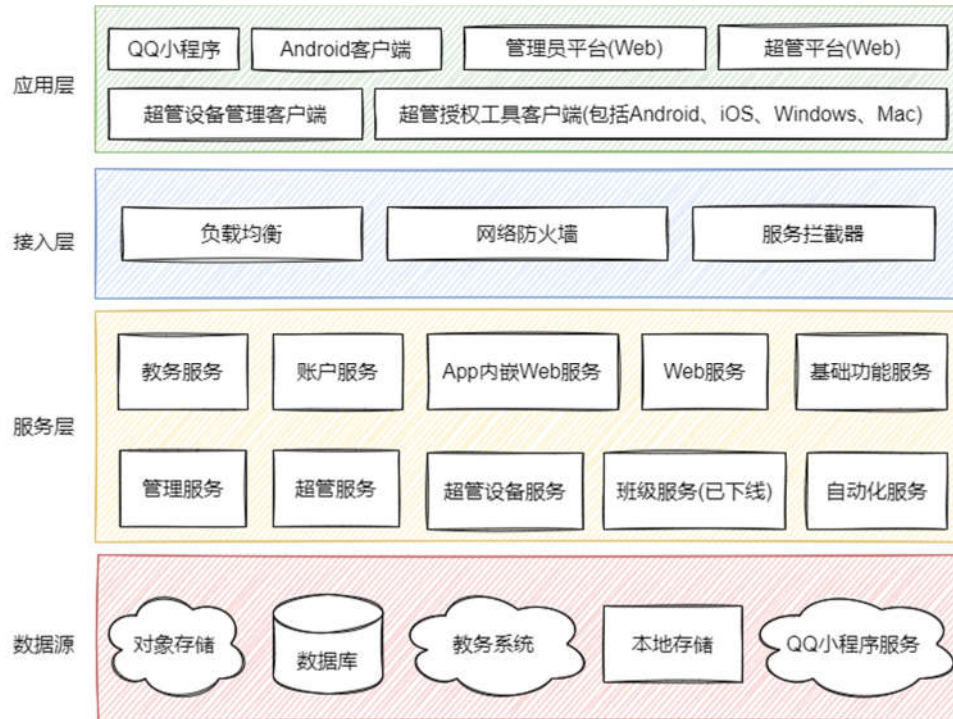


图 6 项目业务架构图

项目业务架构自底向上分为数据源、服务层、接入层、应用层。

数据源负责存储和提供数据，该层中的对象存储负责存储静态业务资源，如 Banner 图片、消息图片等；数据库负责存储业务数据；教务系统作为教务数据来源；本地存储负责存储临时文件；QQ 小程序服务作为用户的 QQ 账户信息来源。

服务层负责处理业务逻辑，该层中的各项服务负责处理其对应的业务逻辑。

接入层负责网络请求的接入控制，该层中的负载均衡用于控制请求访问的服务实例，从而保障服务的可用性；网络防火墙负责控制请求访问的端口，从而保障服务的安全性<sup>[2]</sup>；服务拦截器用于对请求的合法性进行校验。

应用层负责为用户提供操作平台，该层中的 QQ 小程序和 Android 客户端适用于普通用户，管理员平台适用于班级管理员，超级管理员平台和超级管理员授权工具客户端适用于超级管理员，超级管理员设备管理客户端适用于项目 Owner。

项目技术架构图如图 7 所示。

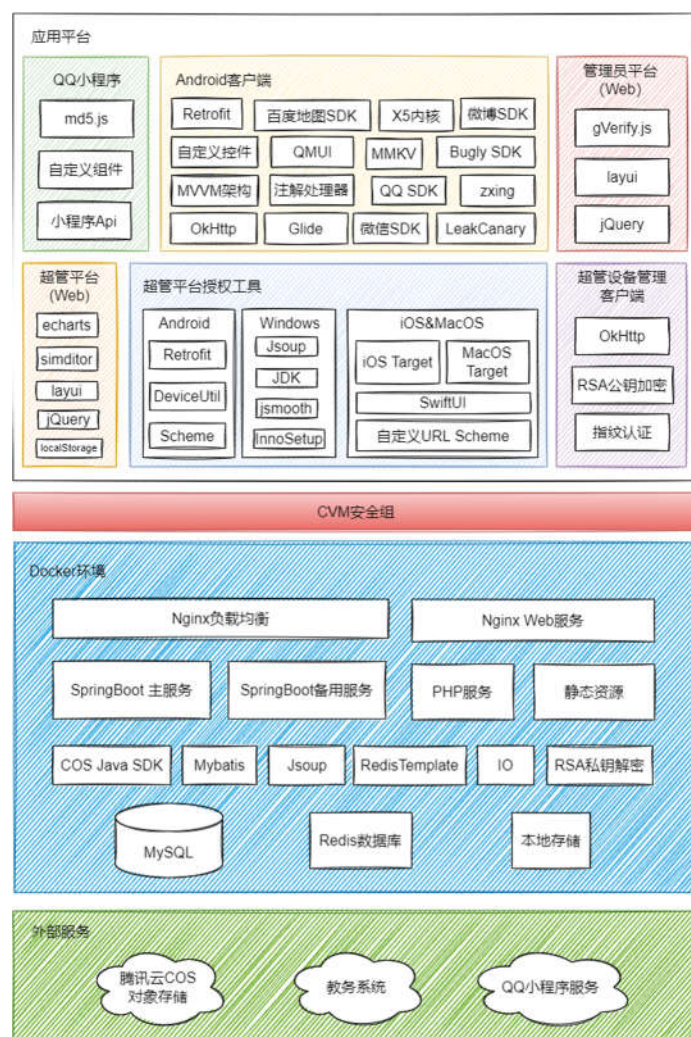


图 7 项目业务架构图

项目业务架构自底向上分为外部服务、Docker 环境、CVM 安全组、应用平台。

外部服务的功能由第三方实现，项目只需接入后进行调用即可。

Docker 环境是所有服务的基础运行环境<sup>[3]</sup>，内部可以大致分为四层，第四层和外部服务综合对应技术架构中的数据源，第三层和第二层综合对应业务架构中的服务层，第一层和 CVM 安全组综合对应业务架构中的接入层。第四层中 MySQL 和 Redis 数据库综合对应业务架构中数据源的数据库部分。第三层为数据操作中间层，其中的 COS Java SDK 是腾讯云官方提供的用于操作 COS 对象存储的工具；Mybatis 用于与 MySQL 交互；Jsoup 用于请求外部服务<sup>[4]</sup>；Redis Template 用于与 Redis 数据库交互；IO 负责操作本地存储；RSA 私钥解密负责业务数据的解密<sup>[5]</sup>。第二层为服务主体，其中的 SpringBoot 主服务和 SpringBoot 备用服务承载了绝大多数的业务逻辑，PHP 承载了个别业务逻辑。第一层中 Nginx 负载均衡可在 SpringBoot 主服务不可用时将请求发送到 SpringBoot 备用服务，Nginx Web 服务用于支持 PHP 服务和静态资源请求。

CVM 安全组负责服务器端口的访问控制，对应业务架构中接入层的网络防火墙，功能由腾讯云实现，只需在控制台进行配置。

用户平台对应业务架构中的应用层，由于项目涉及的技术过多，技术架构图中仅列出了该部分的一些主要技术，更多技术应用将在下文介绍。

### 3.2 用户账户模块

每一个用户使用微哈师 QQ 小程序和微哈师 Android APP 时都需要登录自己的账号，在自己的账户下绑定教务平台账号和邮箱才能使用相关功能。

#### 1. 登录

账户登录涉及用户角色为普通用户、班级管理员，涉及技术平台为 QQ 小程序、Android 客户端、管理员平台、服务端、MySQL 数据库、QQ 小程序服务、QQ 服务，共涉及四个过程：小程序登录、APP 登录态校验、APP 登录、管理员平台登录。

小程序每次启动时会调用小程序框架的登录 API 获得一个登录凭证，小程序将登录凭证交给服务端，服务端使用登录凭证请求小程序服务接口换取 openid，小程序服务接口响应后服务端将请求结果透传给小程序，小程序对结果进行解析以取得 openid，拿到 openid 后调用小程序框架 API 检查是否有获取用户信息的权限，如无权限则调起授权页引导用户授权，授权完成后调用小程序框架获取用户信息 API 获取用户头像和昵称，获取成功后使用上述数据请求服务端，服务端将用户信息存储到数据库中并查询一些业务数据返回给小程序供小程序进行功能初始化<sup>[6]</sup>。

Android 端登录逻辑相对于小程序有一定区别，不同点在于小程序登录每次都需要执行完整的登录流程，而 Android 端在登录完成后可以获得一个 token，后续可通过校验 token 来校验登录态。Android 端在已登录的情况下再次启动 APP 会先触发用户信息更新流程，更新完成后执行登录态校验流程，APP 调用 QQ SDK 的 API 校验登录态，QQ SDK 请求 QQ 服务接口校验 token，校验通过后正常进入主页。APP 启动时如校验登录态失败或未登录会进入登录页，用户点击登录按钮会调起 QQ 授权界面，点击授权按钮会请求 QQ 服务接口进行授权，授权完成后执行更新用户信息的逻辑。

班级管理员进入管理员平台登录页后，管理员平台会调用 gVerify.js 生成验证码图片。用户输入用户名、密码、验证码，点击“登录”按钮，管理员平台会校验验证码，通过后请求服务端登录，服务端校验用户名和密码匹配后生成并更新用户 token，返回成功响应给管理员平台，否则返回失败响应。管理员平台接收到响应后判断登录是否成功，如果登录未成功会显示错误提示，如果登录成功则存储 token 等信息并跳转到平台主页。管理员平台请求服

务端获取管理员信息，服务端校验 token 正确后查询 MySQL 数据库中的管理员信息返回给管理员平台。管理员平台接收到成功响应后将管理员信息存储到本地存储。

### 2.绑定教务平台

绑定教务平台涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、教务平台服务，时序图如图 8 所示。

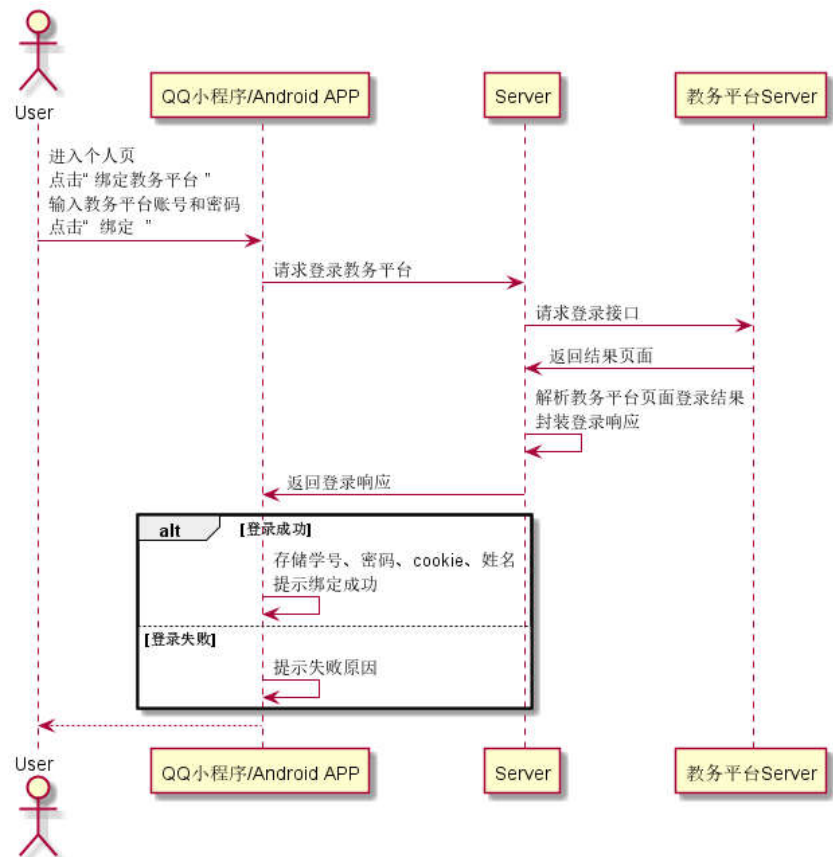


图 8 绑定教务平台功能时序图

绑定教务平台的逻辑小程序和 APP 通用，用户进入个人页，点击“绑定教务平台”后在未绑定状态下会弹出教务平台账号密码输入框，用户输入账号和密码后小程序/APP 会将上述数据交给服务端，服务端使用上述数据请求教务平台登录接口，服务端得到登录结果后返回结果响应给小程序/APP，小程序/APP 得到结果后进行相应处理。

服务端请求教务平台接口登录的过程将在下文教务登录部分详细介绍。

### 3.解绑教务平台

解绑教务平台涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、教务平台服务。小程序端在已绑定状态下点击“绑定教务平台”会自动使用存储的账号和密码执行教务平台登录流程，如登录未成功会自动清除本地存储的全部教务数据，登录成功则展示绑定信息和解绑按钮，用户点击解绑按钮可以手动清除全部教务数据。Android

端在已绑定状态下点击“绑定教务平台”会直接展示绑定信息和解绑按钮，点击解绑按钮的效果与小程序相同。

4.绑定邮箱

绑定邮箱涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、服务端、MySQL 数据库。用户进入个人页在未绑定的状态下点击“绑定邮箱”会弹出邮箱输入对话框，用户输入邮箱后点击“获取验证码”小程序会使用邮箱请求服务端，服务端会生成一个 6 位的随机验证码并将验证码填充到邮件模板，调用邮件服务将验证码发送到邮箱<sup>[7]</sup>后会将邮箱和验证码存入 MySQL 数据库，用户收到验证码后填写到小程序输入框点击“提交”后小程序将邮箱和验证码交给服务端，服务端在 MySQL 数据库中匹配查询邮箱和验证码，查询到结果为绑定成功，否则失败，如绑定成功服务端会更新用户的绑定邮箱并删除对应的验证码记录。

每个邮件验证码的有效时间为 5 分钟，服务端定时服务会每分钟执行一次删除创建时间达到 5 分钟的验证码。

5.解绑邮箱

解绑邮箱涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、服务端、MySQL 数据库，时序图如图 9 所示。

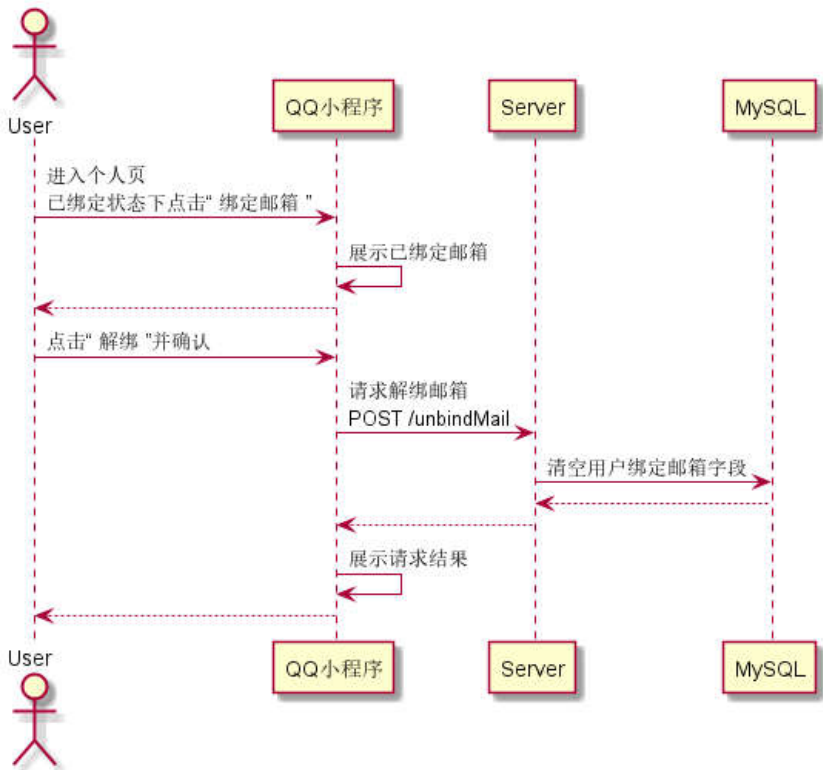


图 9 解绑邮箱功能时序图

用户进入个人页在已绑定的状态下点击“绑定邮箱”会弹出已绑定邮箱对话框，用户点击



解绑按钮并确认后小程序会请求服务端清空用户的绑定邮箱。

6.清除数据

清除数据涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端，共涉及两个过程：清除单项数据和清除全部数据。用户进入设置页，点击“清除数据”可以选择清除某一项数据，确认后小程序/APP 会清除该项数据的本地存储。如果用户点击“清除全部数据”并确认，小程序和 APP 都会清空本地存储的教务数据，小程序会另外请求服务端的清除数据接口，但此接口目前暂时不执行任何逻辑，所以两端清除全部数据的逻辑本质上是一致的。

7.修改登录密码

修改登录密码涉及用户角色为班级管理员，涉及技术平台为管理员平台、服务端、MySQL 数据库，时序图如图 10 所示。

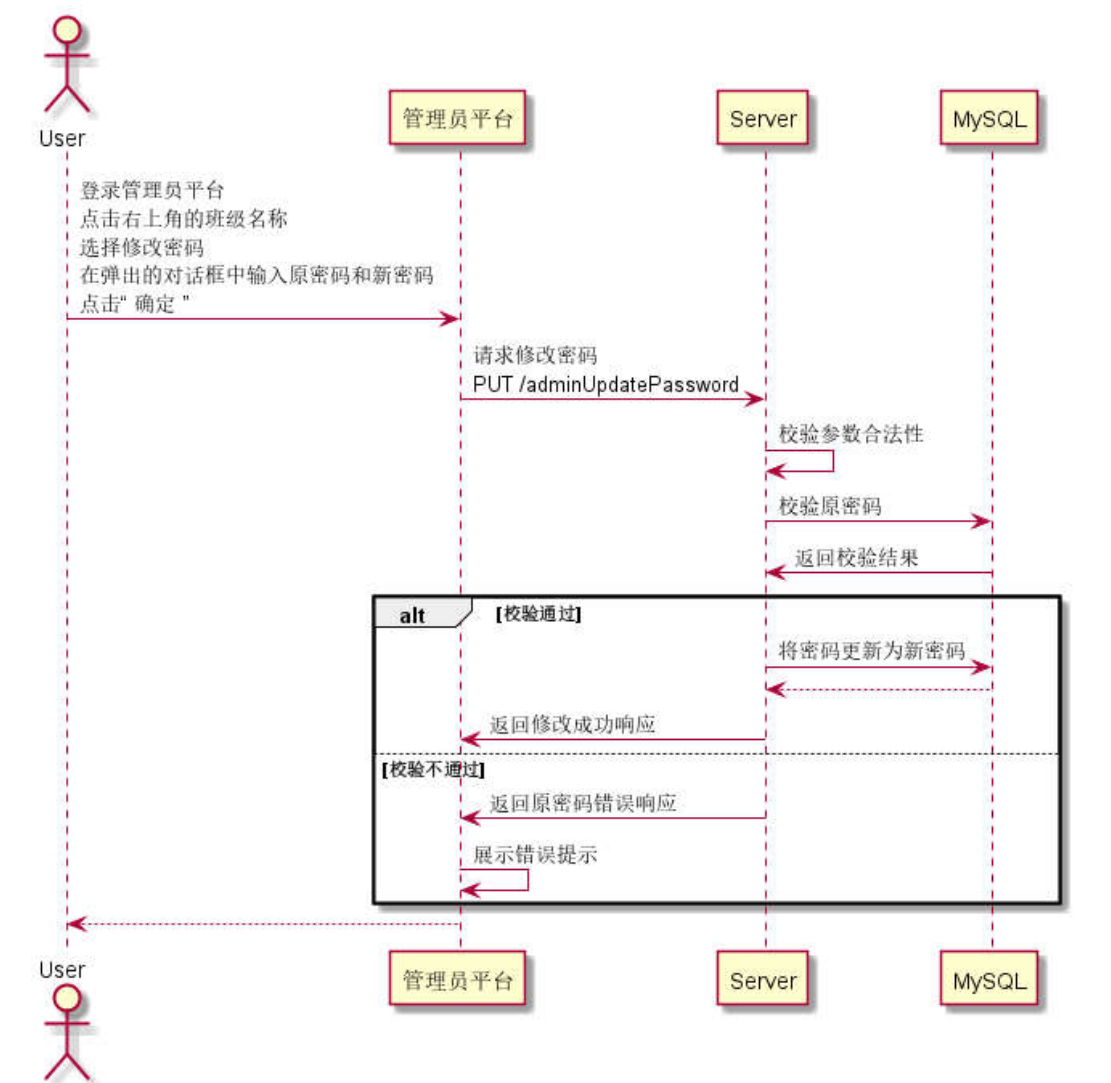


图 10 修改登录密码时序图

班级管理员登录管理员平台，点击右上角的班级名称，在下拉菜单中选择修改密码，在弹出的对话框中输入原密码和新密码，点击“确定”按钮，管理员平台会请求服务端修改密码。服务端校验参数合法性和原密码后将密码更新为新密码，管理员平台收到响应后提示修改结果。

8.用户管理

用户管理涉及用户角色为超级管理员，涉及技术平台为超级管理员平台、服务端、MySQL 数据库，时序图如图 11 所示。

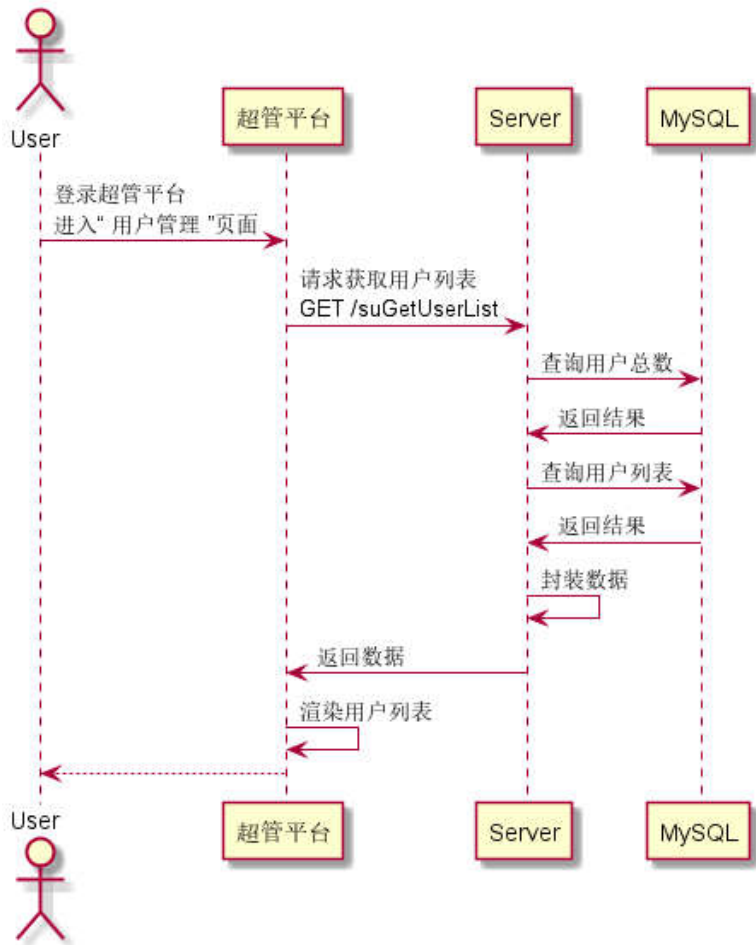


图 11 用户管理功能时序图

超级管理员登录超级管理员平台进入“用户管理”页面，超级管理员平台会分页请求服务端获取用户列表，服务端查询 MySQL 数据库中的用户总数和指定页的列表数据，经过封装后返回给超级管理员平台，超级管理员平台收到响应后将数据渲染到用户列表上。

9.后台账户管理

后台账户管理涉及用户角色为超级管理员，涉及技术平台为超级管理员平台、服务端、MySQL 数据库，共涉及六个过程：生成账号、获取学院列表、修改学院账号密码、获取班

级列表、修改班级账号密码、删除班级账号。

超级管理员登录超级管理员平台进入后台账户管理页输入账户名称，选择账号类型，如果账号类型是班级账号还需选择归属的学院，点击“生成账号”按钮后超级管理员平台会请求服务端生成账号，服务端随机生成账号 id、用户名和密码保存到对应的表中并返回给超级管理员平台。

超级管理员登录超级管理员平台，进入后台账户管理页，超级管理员平台会请求服务端获取学院列表，服务端查询 MySQL 数据库中的班级列表并返回给超级管理员平台，超级管理员平台收到响应后将数据渲染到学院列表上。

超级管理员点击学院列表中的密码单元格可以修改学院账号的密码，修改完成后超级管理员平台会请求服务端更新 MySQL 数据库中学院账号的密码。

超级管理员点击学院列表中的“班级账号管理”按钮，超级管理员平台会分页请求服务端获取学院下的班级列表，服务端查询班级总数和指定页的班级列表数据，经过封装后返回给超级管理员平台，超级管理员平台将数据渲染到班级列表中。

超级管理员点击班级列表中的密码单元格可以修改班级账号的密码，修改完成后超级管理员平台会请求服务端更新 MySQL 数据库中班级账号的密码。

超级管理员点击班级列表中的“删除”按钮并确认，超级管理员平台会请求服务端删除班级账号，服务端会依次删除班级账号关联的所有问卷、通知、成员，最后删除账号记录。

### 3.3 教务平台交互模块

教务平台交互模块负责处理与教务平台相关的业务逻辑，包括登录教务平台、查询课程表数据和查询成绩数据。

#### 1. 登录教务平台

登录教务平台涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、教务平台服务，时序图如图 12 所示。



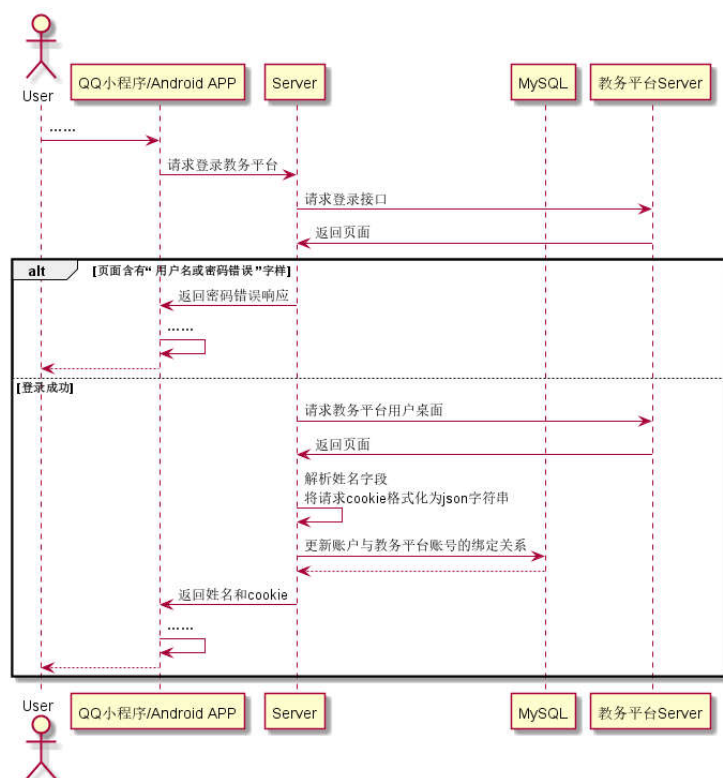


图 12 教务平台登录功能时序图

教务登录的逻辑为小程序和 APP 通用，用户触发登录逻辑后小程序/APP 会将教务平台用户名和密码交给服务端，服务端使用 Jsoup 请求教务平台登录接口，教务平台会返回一个页面，如果密码错误页面里会有“用户名或密码错误”字样，如没有此文案则视为登录成功，如登录失败会直接响应到小程序/APP，登录成功则会使用登录获得的 Cookie 请求教务平台用户桌面页，服务端取得返回的页面后从页面内解析用户姓名，之后将姓名和 Cookie 封装成 json 并更新账户和教务平台的关联关系，将封装后的数据返回给小程序/APP，小程序/APP 获得响应后执行其他逻辑。

## 2.课表查询

课表查询涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、教务平台服务。

用户在小程序/APP 触发课表查询逻辑后，小程序/APP 会先检测是否已绑定教务平台，如未绑定会跳转到个人页引导用户绑定，已绑定状态下 APP 会检测目前是否存在手动添加的课程，如有则弹窗用户确认覆盖，后面小程序/APP 会从本地存储获取上次登录成功的 Cookie，使用 Cookie 请求服务端课表查询接口，如 Cookie 的登录态已失效服务端会返回需要重新登录的响应，小程序/APP 重新执行登录流程后使用新的 Cookie 请求再次课表查询接口，服务端使用有效的 Cookie 请求教务平台课表页，教务平台响应后服务端从页面源码里

解析课表数据并封装成 json 后返回给小程序/APP，小程序/APP 将课表数据存储到本地并渲染到课表组件上。

### 3.成绩查询

成绩查询涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、教务平台服务。成绩查询的逻辑为小程序和 APP 通用，用户在成绩页触发成绩查询逻辑后，小程序/APP 会先检测是否已绑定教务平台，如未绑定会跳转到个人页引导用户绑定，已绑定状态下小程序/APP 会从本地存储获取上次登录成功的 Cookie，使用 Cookie 请求服务端成绩查询接口，如 Cookie 的登录态已失效服务端会返回需要重新登录的响应，小程序/APP 重新执行登录流程后使用新的 Cookie 请求再次成绩查询接口，服务端使用有效的 Cookie 请求教务平台成绩页，教务平台响应后服务端从页面源码里解析课表数据，服务端再次使用 Cookie 请求教务平台用户桌面页，解析出页面内的入学年份，再次使用 Cookie 请求教务平台培养方案页，解析出培养方案列表并结合入学年份推算已修成绩的具体学期，将所得数据与教务平台成绩页解析出的数据取并集，封装成 json 后返回给小程序/APP，小程序/APP 将成绩数据存储到本地并渲染到成绩列表上。

### 3.4 任务清单模块

通过任务清单功能，用户可以设定一个定时提醒任务，可以设置的提醒方式包括推送消息提醒和邮件提醒。

#### 1.新建任务

新建任务涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、服务端、MySQL 数据库。

用户点击任务清单页右下方的“+”图标，小程序会弹出表单对话框，用户填写任务名称、选择提醒方式、点击“添加”按钮，如果设置了模板消息提醒要求时间不能超过一周，原因在于用于推送模板消息的 formid 有效期为一周，如果设置了邮箱提醒还需检查邮箱绑定状态，如果用户未绑定邮箱小程序会引导用户绑定邮箱，设置提醒的条件满足后小程序会请求服务添加任务接口，服务端会为任务生成一个 id，将任务保存到数据库。

任务提醒的实现依赖于服务端的定时任务，服务端定时任务每分钟会查询数据库中需要执行提醒的任务，如果设置了模板消息提醒，服务端会构造模板消息，请求 QQ 小程序服务接口进行消息推送，如果设置了邮件提醒则调用邮件服务将提醒邮件发送到用户的绑定邮箱。

在请求 QQ 小程序服务接口进行消息推送时需要携带小程序的 access\_token，该参数由

服务端定时任务每两小时请求 QQ 小程序服务接口进行获取。

2.删除任务

删除任务涉及用户角色为普通用户，涉及技术平台为 QQ 小程序。用户在任务清单页点击任务列表中的任务项，小程序会弹出任务详情对话框，用户点击对话框内的“删除”按钮并确认后小程序会将存储在本地存储的任务记录删除，如果任务设置了提醒，当任务被删除后提醒并不会被取消。

3.5 校园地图模块

用户可以通过校园地图功能了解学校内一些重要地点的位置,如果用户在对应的校区内还可以查询到指定地点的路线。

1.数据更新

POI 数据更新涉及用户角色为普通用户，涉及技术平台为 Android 客户端、服务端，时序图如图 13 所示。

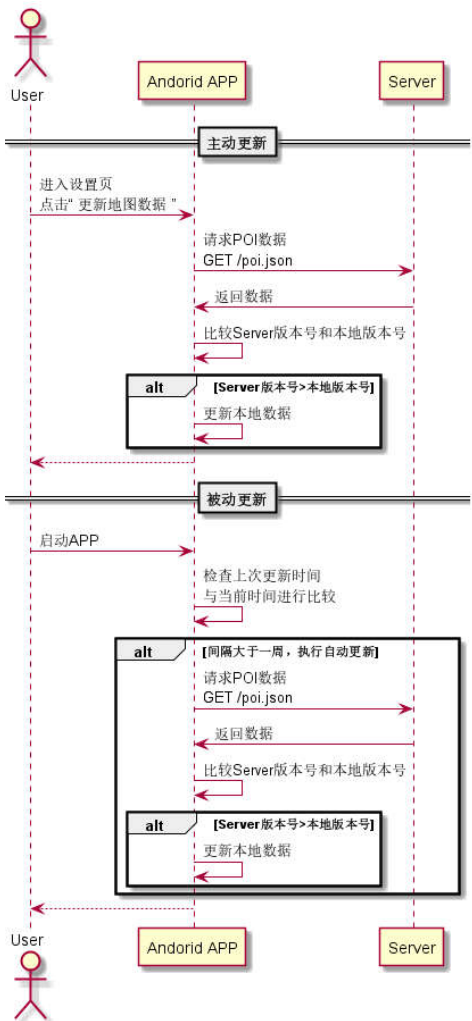


图 13 数据更新功能时序图

POI 数据更新共有两个过程：主动更新和被动更新。

APP 自身有一份内置的 POI 数据，用户点击设置页的“更新地图数据”，APP 会请求服务端的静态 POI 数据文件，接收到服务端返回的 POI 数据后通过比对 POI 数据的版本号是否有更新，如有更新则将最新的数据存储到本地，在下一次更新时会将服务端版本号与本次存储数据的版本号进行比对，如有更新则覆盖本地存储数据。

在用户启动 APP 时，APP 会检查地图数据上次更新时间是否已超过一周，如已超一周则执行被动更新，自动请求更新数据。

## 2. 地图展示

地图展示涉及用户角色为普通用户，涉及技术平台为 Android 客户端，共涉及三个过程：百度地图 SDK 初始化、显示地图、导航。

在用户启动 APP 时，APP 会自动进行 SDK 的初始化。

用户点击首页的“校园地图”版块时，由于地图相关功能需要定位支持，APP 会先检查是否具有定位权限，如没有权限会弹窗提醒用户授权，若用户拒绝授予权限将无法使用地图相关功能，进入地图页后 APP 会先渲染地图组件、从本次读取地图数据渲染到 POI 列表并初始化定位<sup>[8]</sup>，如果用户当前的位置在江南校区或江北校区，APP 会显示用户当前的实际定位，如果用户在校外则会显示定位为江北校区行知楼。

用户选中 POI 列表项时，如果用户在选中地点所属校区内，APP 会调用百度地图 SDK 查询用户当前位置到选中地点的步行路线，并将最优路线渲染到地图上，如果用户在校区外仅会标注选中地点的位置。

## 3. POI 上报

POI 上报涉及用户角色为普通用户，涉及技术平台为 Android 客户端、POI 上报页面、服务端、MySQL 数据库。用户点击地图页右上角的“上传”图标，APP 会跳转到内置的浏览器页面加载 POI 上报页面，用户在 POI 上报页面填写 POI 信息并点击“上报”后，POI 上报页面会将表单提交到服务端，为了保证服务的安全性，POI 上报接口添加了频控策略，限制每个用户每天最多可上报 5 条 POI 数据，仅当日上报次数未达上限时才会将用户提交的 POI 信息存储到数据库中。

### 3.6 超级管理员设备鉴权模块

超级管理员登录超级管理员平台的认证方式为设备认证，即只有通过授权的设备才能够登录超级管理员平台，超级管理员设备鉴权模块负责处理设备鉴权相关的业务逻辑。

### 1.设备权限验证

权限验证涉及用户角色为超级管理员、项目 Owner，涉及技术平台为超级管理员平台、超级管理员设备管理客户端、服务端、MySQL 数据库，共涉及两个过程：超级管理员设备权限验证、超级管理员设备客户端权限验证。

超级管理员访问超级管理员平台页面需要通过超级管理员设备验证，进入超级管理员平台任意页面使用任意功能时超级管理员平台会尝试从本地获取设备 id，将设备 id 放入请求的 Header 中，服务端超级管理员拦截器会尝试从 Header 中取得设备 id，如果 Header 中不存在设备 id，服务端会生成一个设备 id 返回给超级管理员平台供超级管理员平台请求授权，如果 Header 中存在设备 id，服务端会将设备 id 格式化为 key，查询 Redis 数据库中是否存在该 key，如不存在则验证不通过，否则放通请求到对应接口，超级管理员平台如果在响应中收到了服务端生成的设备 id 会将其存储到本地存储供后续授权使用。

项目 Owner 在使用超级管理员设备管理客户端时会接受更加严格的验证，在客户端启动时会调用指纹传感器要求项目 Owner 验证指纹，只有当指纹验证后客户端才会执行主要逻辑<sup>[9]</sup>，每次请求服务端时，客户端会使用客户端内置的 RSA 公钥对当前时间戳进行加密，将时间戳密文放入请求的 Header 中，服务端设备管理拦截器会尝试从 Header 中获取时间戳密文，使用服务端的 RSA 私钥对密文进行解密，将解密结果与服务端当前时间戳进行对比，如果相差 10 秒内则验证通过，放通请求到相应接口，否则验证失败，返回 403 错误。

### 2.申请设备授权

申请授权涉及用户角色为超级管理员，涉及技术平台为超级管理员平台、全平台超级管理员设备授权工具、服务端、MySQL 数据库、Redis 数据库。超级管理员平台收到无权限响应后会跳转到 403 错误页，该页面内版权信息部分为超链接，页面会从本地存储获取设备 id，将其拼接到超链接的参数部分，超级管理员点击超链接后浏览器会启动对应平台的授权工具，授权工具解析超链接获取其中的设备 id，请求服务端将设备 id 加入设备列表供管理员授权使用，如果设备列表内已有该设备 id，服务端会将设备 id 格式化为 key，查询 Redis 数据库中是否存在该 key，如果存在则更新对应的设备信息，如果设备列表内不存在该设备 id 则将设备 id 存入 MySQL 数据库的设备列表中。

### 3.授权设备管理

授权管理涉及用户角色为项目 Owner，涉及技术平台为超级管理员设备管理客户端、服务端、MySQL 数据库、Redis 数据库，时序图如图 14 所示。

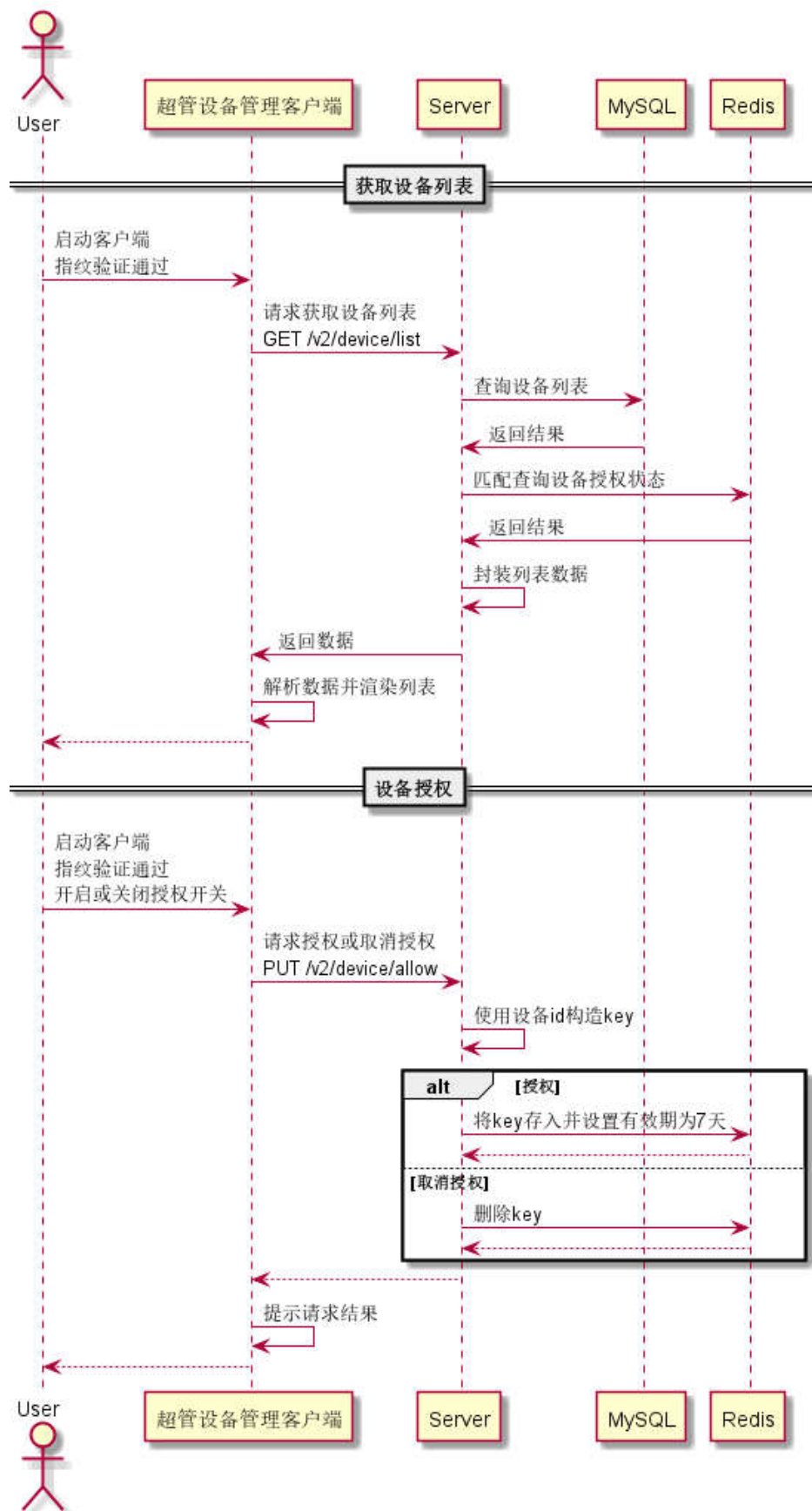


图 14 超级管理员权限授权管理功能时序图

授权管理共有两个过程：获取设备列表、设备授权。

项目 Owner 进入客户端主页，客户端会请求服务端获取设备列表，服务端从 MySQL 数据库查询设备列表，从 Redis 数据库查询设备列表内设备的授权状态，经过封装后返回给客户端，客户端将数据渲染到设备列表上。

项目 Owner 点击设备列表中的授权开关，客户端会请求服务端授权或取消授权，如请求授权服务端会将设备 id 格式化为 key，存入 Redis 数据库，请求取消授权服务端会删除 Redis 数据库中对应的 key。

### 3.7 数据统计模块

通过对项目运行过程中用户总数、日活跃用户数量、各端日启动总次数等运行指标进行统计并展示到超级管理员平台，超级管理员可以对项目运行情况有更加清晰的了解。

#### 1. 数据采集

数据采集涉及用户角色为普通用户，涉及技术平台为 QQ 小程序、Android 客户端、服务端、MySQL 数据库、Redis 数据库。用户在触发小程序的用户信息更新流程时，服务端会将用户的 openid 存储到 Redis 数据库内的日活 HyperLogLog 字段内，将 Redis 数据库内的日启动次数计数值增加 1。

在每日 0 时时，服务端定时任务会将上述两项数据和前一日的时间戳插入对应字段的列表并将上述两个字段的值清零。

其余统计指标为调取时实时查询，无需采集。

#### 2. 数据面板

数据面板涉及用户角色为班级管理员、超级管理员，涉及技术平台为管理员平台、超级管理员平台、服务端、MySQL 数据库、Redis 数据库，共涉及两个过程：管理员平台数据面板、超级管理员平台数据面板。

班级管理员进入管理员平台概览页，管理员平台会请求服务端获取统计数据，服务端从 MySQL 数据库中查询班级成员人数和班级已发布通知数，封装后返回管理员平台，管理员平台将数据渲染成数据块或图表<sup>[10]</sup>。

超级管理员登录超级管理员平台进入概览页，超级管理员平台会请求服务端获取统计数据，服务端从 MySQL 数据库中查询用户总数、已绑定教务平台人数、已发布通知总数、已发布问卷总数、班级总数，从 Redis 数据库查询今日日活人数，今日日启动次数、近一周日活人数、近一周日启动次数，经过封装后返回给超级管理员平台，超级管理员平台将数据渲染到页面上。

### 3.8 测试用户鉴权模块

微哈师 Android APP 正式发布前会进行小范围的用户测试，只有得到测试授权的用户才能正常登录测试版本的微哈师 Android APP。

#### 1. 测试权限验证

权限验证涉及用户角色为普通用户，涉及平台为 Android 客户端、服务端、MySQL 数据库，时序图如图 15 所示。

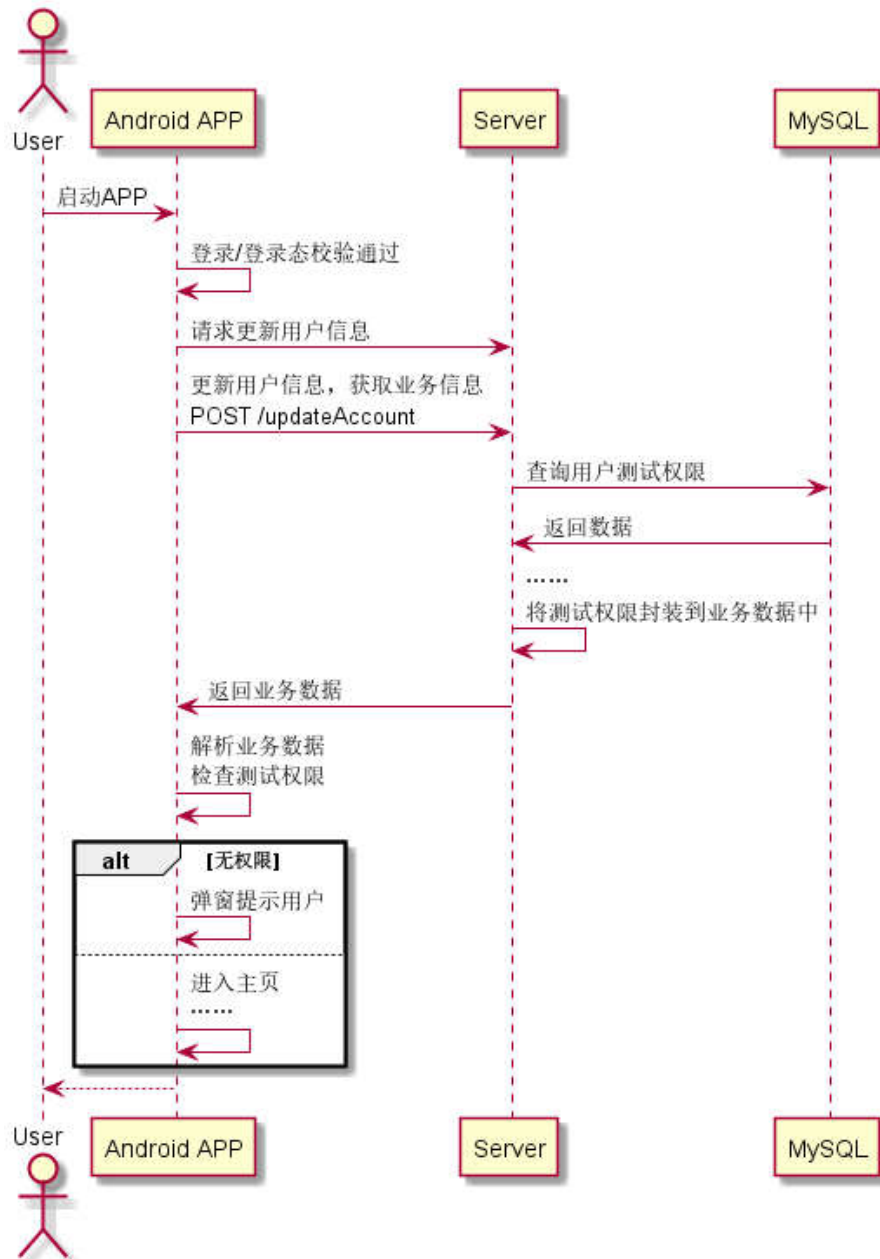


图 15 APP 测试权限验证功能时序图

用户启动测试版 APP 时 APP 会在更新用户信息阶段从服务端返回的响应中取得用户的测试权限标志，如果用户没有测试权限将弹窗引导用户申请测试权限，如果用户有测试权限，



APP 会正常跳转到主页。

2.申请测试授权

申请授权涉及用户角色为普通用户，涉及平台为 Android 客户端、权限申请 Web 页面，待审核 Web 页面、已提交 Web 页面、服务端、MySQL 数据库，时序图如图 16 所示。

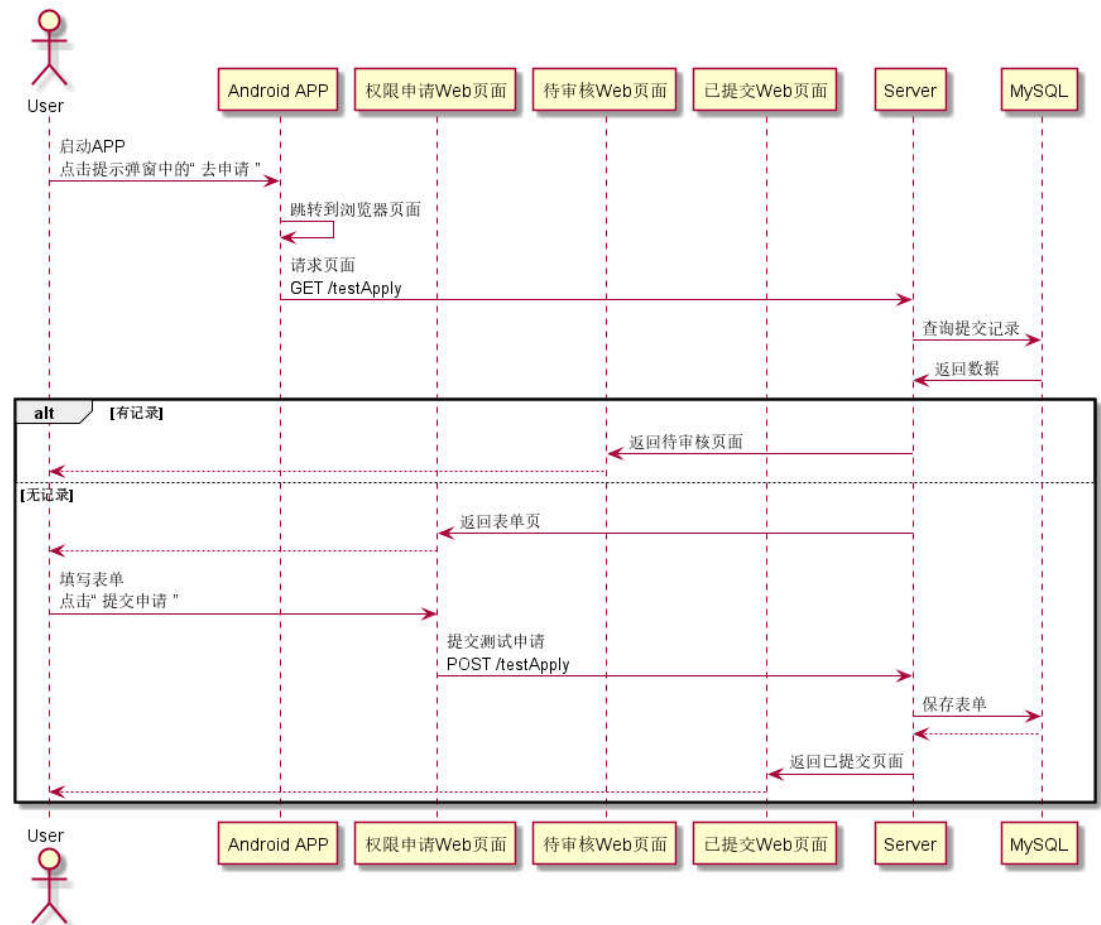


图 16 APP 测试权限申请功能时序图

用户点击引导弹窗中的“去申请”，APP 会跳转到浏览器页面，请求服务端权限申请 Web 页面，服务端查询 MySQL 数据库中的申请提交记录，如用户已提交过申请会返回待审核页面，如用户未提交过申请会返回权限申请页面，用户填写表单点击“提交申请”按钮后页面会请求服务端提交申请，服务端将申请信息保存到 MySQL 数据库中，返回已提交页面。

3.测试授权管理

授权管理涉及用户角色为超级管理员，涉及平台为超级管理员平台、服务端、MySQL 数据库，时序图如图 17 所示。

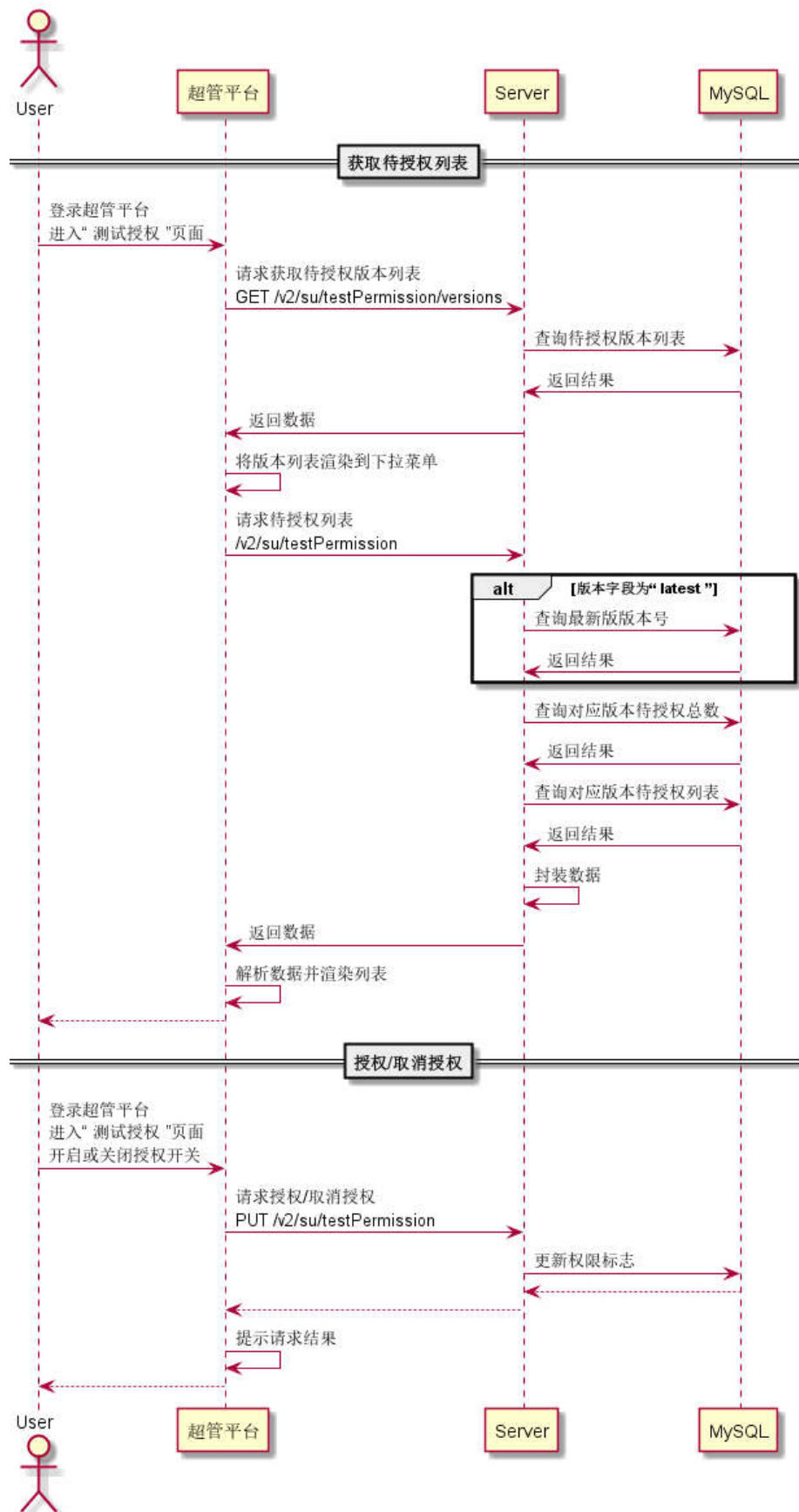


图 17 超级管理员设备授权管理功能时序图

授权管理共有两个过程：获取待授权列表、授权/取消授权。

超级管理员登录超级管理员平台进入测试授权页,超级管理员平台会请求服务端获取版本号列表,服务端查询 MySQL 数据库中的版本号列表返回给超级管理员平台,超级管理员平台将版本号列表渲染到下拉菜单中后再次请求服务端获取待授权列表,服务端核对超级管理员平台请求的版本号,如果是“latest”服务端会从 MySQL 数据库中查询出最新的版本号,然后服务端使用最终版本号查询待授权列表返回给超级管理员平台,超级管理员平台将待授权列表数据渲染到待授权列表上。

超级管理员点击待授权列表中的授权开关,超级管理员平台会请求服务端授权或取消授权,由服务端更新 MySQL 数据库中用户的权限标志。

## 4 系统实现

### 4.1 Android APP 实现

#### 1.MVVM 架构搭建

MVVM 架构被应用于微哈师 Android 客户端，MVVM 架构分为 Model、View 和 ViewModel 三部分，在项目中分别起到数据抽象、业务逻辑处理、布局渲染和事件处理的作用<sup>[11]</sup>。项目中 MVVM 架构的搭建基于 JetPack 提供的一系列工具，包括 DataBinding、LiveData、ViewModel 等。

View 与 ViewModel 的交互使用 DataBinding 实现，使用 Databinding 首先需要在 app 模块下的 build.gradle 内配置启用 DataBinding，对于已有的布局可以使用 AndroidStudio 的辅助操作将其转换成适用于 DataBinding 的结构，即根布局为 layout，内部分为 data 部分和布局部分，data 部分可以引用布局所需的数据对象，布局部分即为原来的布局，但与传统布局不同的是该部分布局可以使用特定的语法单向或双向绑定布局所引用的数据对象。此外，还可以使用 BindingAdapter 注解配置绑定适配器，从而实现在布局内调用绑定适配器进行数据处理和绑定。

ViewModel 的实现基于 JetPack 的 ViewModel 组件，ViewModel 的显著特点在于它可以管理界面控制器的生命周期，能够实现布局控制器间的数据共用，而且使用方法非常简单，只需要让自定义的 ViewModel 组件继承 JetPack 的 ViewModel 类。

ViewModel 内的数据对象管理使用 LiveData 实现，LiveData 能够感知布局管理器的声明周期，基于观察者模式。因此可以使用对其注册观察者，从而实现对数据对象变化的监听，且无需显式地对观察者进行注销操作。

#### 2.网络请求封装

项目的网络请求使用 Retrofit 框架和 Kotlin 协程实现，并且进行了封装。网络接口整体分为业务类接口和配置类接口，分别运行于服务端不同的端口上。业务类接口使用 Kotlin 协程的挂起函数实现，在 Repository 内使用 withContext 扩展函数封装请求方法，并实现在该层中定义一些业务接口响应的处理异常。

接口响应数据中包括接口响应信息和实际业务数据，在实际的业务数据模型中不包含接口响应信息的相关字段，无法直接将接口响应转换成业务数据模型。在这个背景下可以将业务数据模型分为单模型和列表模型两类，基于注解、注解处理器和 JavaPoet 框架自动生成接口响应数据模型。接口响应数据模型生成模块为 Java 模块，所有接口响应数据模型继承响应数据模型基类，基类中包括错误码和消息字段及对应的 Getter 和 Setter，定义

ResponseModel 和 ResponseListModel 两个编译期类注解分别标注单模型和列表模型，实际的类生成过程由注解处理器负责，项目编译时注解处理器的 process 方法会被调用，该方法可以访问编译环境，定义类生成方法从编译环境中获得所有被模型注解标注的类，使用 JavaPoet 生成接口响应数据模型类，添加类型为业务数据模型或业务数据模型列表的 data 字段及其 Getter 和 Setter。

项目中接口的请求与响应处理由 ViewModel 负责，因此在 ViewModel 基类中封装网络请求上下文环境，在 ViewModelScope 内进行请求超时的限制和实际的请求执行，该层中对网络请求的业务异常和技术异常进行拦截，对布局控制器提供异常统一处理方法。真正的网络请求则由具体业务对应的 ViewModel 按照业务逻辑发起。

3.自定义课表组件

课表数据的显示由自定义的开源课表组件 TableView 来负责，组件使用 Android SDK 原生组件进行自定义，功能较为丰富且不会出现依赖冲突问题。组件布局整体分为日期栏、时间栏、课程区、课程块四大部分，使用相对布局、线性布局、滚动布局和帧布局嵌套实现。

组件设置了一系列属性以便对组件样式进行自定义，属性列表如表 1 所示。

表 1 TableView 自定义属性列表

属性	格式	描述
tv_count_per_day	integer	单日课程总数（必需）
tv_resolve_flags	reference	字符数组，对应周一至周日，用于解析课程，默认为["mon", "tue", "wed", "thur", "fri", "sat", "sun"]，建议自行设置此项。
tv_indicator_bg_color	color	时间栏背景颜色
tv_indicator_text_color	color	时间栏文字颜色
tv_week_bg_color	color	日期栏背景颜色
tv_week_text_color	color	日期栏文字颜色
tv_lesson_text_color	color	课程块的文字颜色

组件的部分尺寸参数可以通过覆盖 dimen 资源文件内的资源进行自定义，资源列表如表 2 所示。

表 2 TableView 可覆盖资源列表

属性	描述
height_per_lesson	每个课程块的高度，控件的高度实际由本属性和单日课程总数决定

lesson_text_size	课程块的文字大小
lesson_border_radius	课程块四个角的弧度

使用组件时可以自定义课程数据模型，但需要继承自组件的 Lesson 类或设置自定义课程数据模型和 Lesson 类间的转换方法。

#### 4. 自定义地址选择器

校园地图模块中的校内导航功能用到了地址选择器组件，由于现有开源组件无法满足项目需求，所以对地址选择器进行了自定义，地址选择器组件分为半屏浮层和地址选择器部分。

半屏浮层基于开源组件 BottomDrawerLayout 修改，原有组件基本符合项目需求，但在事件处理方面有一定缺陷，因此按照项目需求对 BottomDrawerLayout 进行了优化和定制。

地址选择器使用自定义开源组件 TreePicker 实现，该组件设计定位为通用树形结构选择组件，使用 Kotlin 语言编写，适用于所有树形结构模型的渲染。数据模型基类为树形节点，包括 name 和 next 属性，实现 Convertible 接口的 convert 方法，用于将自定义节点类型转换为基类类型。组件整体分为层级列表和节点列表，层级列表为栈结构，每次选择节点列表中的非叶子节点时，该节点会被加入层级节点栈栈顶，当层级节点栈中的非栈顶节点被点击时，该节点上方的所有节点都会被弹出栈，节点列表渲染该节点的子节点列表。为了方便对组件样式进行自定义，组件设置了一系列自定义属性和 Configuration 类，自定义属性列表如表 3 所示。

表 3 TreePicker 自定义属性列表

属性	格式	描述
tabItemPadding	dimension	tab 块的左右边距
tabHeight	dimension	tab 栏高度
tabTextSelectedColor	color	tab 栏文字选中颜色
tabTextNormalColor	color	tab 栏文字默认颜色
tabUnderline	reference(drawable)	tab 选中时的下标记
tabTextSize	dimension	tab 栏文字大小
levelTextSize	dimension	列表文字大小
levelPadding	dimension	列表 item 内边距
levelCheckDrawable	reference(drawable)	列表 item 选中标记
levelTextSelectedColor	color	列表文字选中颜色
levelTextNormalColor	color	列表文字默认颜色

## 5. 社会化分享 SDK 集成和封装

项目中集成社会化分享 SDK 按照当前需求主要用于实现 APP 的分享推广，但分享模型的设计对所有场景通用。所有的社会化分享数据都会被封装成数据分享模型 `ShareContent`，`ShareContent` 包括类型、标题、描述、链接、缩略图五个属性，其中类型包括文本、图片、音频、视频、网页等类型。所有分享渠道均实现 `Channel` 接口的 `Share` 方法，各渠道从方法的 `shareContent` 参数中取得所需的业务参数调起对应渠道的 SDK 进行分享。

## 6. 浏览器调起 APP

浏览器调起 APP 的功能应用于各端超级管理员授权工具，Android 端浏览器调起 APP 使用深层链接实现，该功能具体由 Activity 的静态 `IntentFilter` 负责过滤 Uri，需要在 `IntentFilter` 内添加 `data` 标签、`category` 标签和 `action` 标签，在 `data` 标签中声明指定的 `scheme` 和 `host` 属性值，添加 `DEFAULT` 和 `BROWSABLE` 两个类别的 `category` 标签使应用能够响应隐式的网络 Intent，添加类型为 `ACTION_VIEW` 的 `action` 标签使 `IntentFilter` 能够被网络访问，经过上述步骤后即可在浏览器中使用符合设定格式的 Uri 链接调起 APP。因授权工具需要获取到 Uri 中的参数，需要在 Activity 中使用 `getIntent()` 方法获取 Intent 对象，从而获得 Uri 对象，解析出所需的参数值。

## 7. 本地数据存储

APP 请求接口获得的课表数据和成绩数据以及 APP 使用过程中产生的其他用户数据将会被存储到本地，本地数据存储使用开源数据存储框架 `MMKV` 实现，该框架由微信客户端团队开发，并经过了微信客户端长期的验证，相比于 Android 原生的 `SharedPreferences` 具有更高的性能和更好的稳定性。项目中用到了 `MMKV` 的数据加密功能，通过在 `MMKV` 对象初始化时制定一个字符串密钥，`MMKV` 在对数据进行存储和获取时会使用该密钥对本地存储的数据进行加密和解密。

## 8. 系统权限申请

APP 中的校园地图和保存图片等功能涉及到读写存储和定位等权限的申请，权限申请使用开源权限申请框架 `PermissionX` 实现，该框架使用 Kotlin 语言编写，对新版本 Android 系统的适配比较及时，极大地简化了权限请求的代码。通过 `PermissionX` 的 `init` 方法绑定 `Context` 对象后使用 `permissions()` 方法设定要请求的权限，使用 `request` 方法发起权限申请，通过回调方法内的参数可以获得申请通过的权限列表和未申请通过的权限列表以及是否所有权限均申请通过，也可以使用 `onExplainRequestReason()` 方法和 `onForwardToSettings()` 方法向用户说明申请权限的原因和引导用户到设置页面手动开启所需权限。

## 9.内存泄漏检测

为了保证 APP 运行流畅，减少系统资源浪费，项目使用了 LeakCanary 框架进行内存泄漏检测，以便及时修复内存泄漏问题。使用 LeakCanary 只需要在项目中添加依赖，在应用 debug 版本运行时它会自动检测内存泄漏，如发生内存泄漏会显示通知提示，点击通知可查看对象的引用链和内存泄漏的详细信息<sup>[12]</sup>。

## 10.崩溃/异常监控

为了掌握 APP 在线上运行情况，及时发现和修复线上问题，项目接入了腾讯 Bugly SDK 捕捉应用异常和监控应用崩溃，使用 Bugly SDK 需要将其依赖添加到项目中，在项目清单中配置应用相关参数，在 APP 启动时初始化 SDK，此外对异常上报策略进行配置。

### 4.2 服务端核心接口实现

#### 1.登录教务平台

接口请求地址为：POST /login。接口请求参数相关信息如表 4 所示。

表 4 login 接口请求参数列表

属性	类型	必填	说明
openid	string	是	
username	string	是	学号
password	string	是	密码

接口返回值参数相关信息如表 5 所示。

表 5 login 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息
data	Object	用户信息

data 字段内参数相关信息如表 6 所示。

表 6 login 接口 data 字段参数列表

属性	类型	说明
name	string	姓名
cookies	string	教务平台 cookies

code 的合法值相关信息如表 7 所示。



表 7 login 接口错误码合法值列表

值	说明
40002	用户名或密码错误
40001	登录成功
40006	教务平台连接失败

## 2. 获取课程表

接口请求地址为：GET /getDailyLessons。请求参数如表 8 所示。

表 8 getDailyLessons 接口请求参数列表

属性	类型	必填	说明
cookieString	string	是	教务平台 cookies 字符串

接口返回值参数相关信息如表 9 所示。

表 9 getDailyLessons 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息
data	Object	填写内容

data 字段内参数相关信息如表 10 所示。

表 10 login 接口 data 字段参数列表

属性	类型	说明
weekTotal	string	总周数
lessons	Array<Lesson>	课程列表

Lesson 类型参数相关信息如表 11 所示。

表 11 Lesson 类型参数列表

属性	类型	说明
name	string	课程名
teacher	string	任课教师
bg	string	课程块 HexColor
parts	Array<Part>	课节列表

Part 类型参数相关信息如表 12 所示。

表 12 Part 类型参数列表

属性	类型	说明
weekday	string	周几
start	number	从第几节开始
end	number	到第几节结束
place	string	上课地点
weeks	Array<number>	上课教学周列表

code 字段的合法值相关信息如表 13 所示。

表 13 getDailyLessons 接口错误码合法值列表

值	说明
40000	需要重新登录
70006	数据解析异常
10004	获取成功
70004	未获取到本学期课程
70003	获取失败

### 3. 获取成绩

接口请求地址为：GET /getGrade。请求参数如表 14 所示。

表 14 getGrade 接口请求参数列表

属性	类型	必填	说明
cookieString	string	是	教务平台 cookies 字符串

接口返回值参数相关信息如表 15 所示。

表 15 getGrade 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息
data	Array<Grade>	成绩列表

Grade 类型参数相关信息如表 16 所示。

表 16 Grade 类型参数列表

属性	类型	说明
term	string	学期
name	string	课程名

total	string	总成绩
credit	string	学分
duration	string	学时
am	string	考核方式
ep	string	考试性质
lp	string	课程性质

code 字段的合法值相关信息如表 17 所示。

表 17 getGrade 接口错误码合法值列表

值	说明
40000	需要重新登录
70005	没有成绩数据
10004	获取成功
40006	教务平台连接失败

#### 4. 获取超级管理员设备列表

接口请求地址为：GET /getDeviceList。接口返回值参数相关信息如表 18 所示。

表 18 getDeviceList 接口请求参数列表

属性	类型	说明
devices	Array<Device>	设备列表

Device 类型参数相关信息如表 19 所示。

表 19 Device 类型参数列表

属性	类型	说明
mac	string	mac 地址
host_name	string	设备名称
ip	string	ip 地址

#### 5. 超级管理员设备授权

接口请求地址为：PUT /allowDevice。请求参数如表 20 所示。

表 20 allowDevice 接口请求参数列表

属性	类型	必填	说明
mac	string	是	mac 地址
allow	boolean	是	是否允许设备访问超级管理员后台

## 6.登录

接口请求地址为：POST /updateAccount。请求参数如表 21 所示。

表 21 updateAccount 接口请求参数列表

属性	类型	必填	说明
openid	string	是	
nickname	string	是	QQ 昵称
head	string	是	QQ 头像链接

接口返回值参数相关信息如表 22 所示。

表 22 updateAccount 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息
data	Object	用户信息

data 字段内参数相关信息如表 23 所示。

表 23 updateAccount 接口 data 字段参数列表

属性	类型	说明
email	string	邮箱地址
class_id	string	班级 id
is_cadre	string	是否为班干部
className	string	班级名称
instituteName	string	学院名称
term	string	当前学期
termStart	string	学期开始时间

## 7.获取邮件验证码

接口请求地址为：GET /getEmailCode。请求参数如表 24 所示。

表 24 getEmailCode 接口请求参数列表

属性	类型	必填	说明
openid	string	是	
email	string	是	邮箱地址

接口返回值参数相关信息如表 25 所示。

表 25 getEmailCode 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息

code 字段的合法值相关信息如表 26 所示。

表 26 getEmailCode 接口错误码合法值列表

值	说明
70002	获取失败
10004	获取成功

## 8. 绑定邮箱

接口请求地址为：POST /bindEmail。请求参数如表 27 所示。

表 27 bindEmail 接口请求参数列表

属性	类型	必填	说明
openid	string	是	
email	string	是	邮箱地址
code	string	是	验证码

接口返回值参数相关信息如表 28 所示。

表 28 bindEmail 接口返回值参数列表

属性	类型	说明
code	number	错误码
msg	string	错误信息

code 字段的合法值相关信息如表 29 所示。

表 29 bindEmail 接口错误码合法值列表

值	说明
10005	绑定成功
40005	验证码错误

## 9. 获取 openid

接口请求地址为：GET /getOpenid。请求参数如表 30 所示。

表 30 getOpenid 接口请求参数列表

属性	类型	必填	说明
code	string	是	小程序请求码

接口返回值参数相关信息如表 31 所示。

表 31 getOpenid 接口返回值参数列表

属性	类型	说明
openid	string	
session_key	string	会话秘钥
errcode	number	错误码
errmsg	string	错误信息

errcode 字段的合法值相关信息如表 32 所示。

表 32 getOpenid 接口错误码合法值列表

值	说明
-1	系统繁忙
0	请求成功
40029	code 无效
45001	请求频率过快

### 4.3 数据库建设

#### 1. 用户表

用户表中的字段包括 openid、昵称、头像、姓名、学号、邮箱、端平台、上次使用时间，详细信息如表 33 所示。

表 33 user 表结构

字段	类型	默认值	备注
openid	varchar(32)		主键
nickname	varchar(100)		用户 QQ 昵称
head	varchar(100)		用户 QQ 头像链接
name	varchar(20)	空字符串	用户姓名
studentid	varchar(10)	空字符串	用户学号
email	varchar(50)	空字符串	用户邮箱地址
platform	smallint(6)	1	用户使用的端平台，1: QQ 小程序， 2: Android 客户端
latest_use	timestamp	当前时间戳	用户上次登录时间，随记录更新而更新 新为最新的时间戳

## 2.测试权限表

测试权限表中的字段包括：openid、姓名、学号、邮箱、申请时间、申请版本、申请是否通过，详细信息如表 34 所示。

表 34 test\_application 表结构

字段	类型	默认值	备注
openid	varchar(32)		主键字段之一
name	varchar(20)		用户姓名
studentid	varchar(10)		用户学号
email	varchar(50)		用户邮箱地址
timestamp	timestamp	当前时间戳	申请提交时间
version	varchar(10)	1.0.0	申请的 APP 版本号，主键字段之一
enabled	tinyint(1)	0	申请是否通过，0：未通过，1：已通过

## 3.超级管理员设备表

超级管理员设备表中的字段包括：mac 地址、主机名、ip 地址、授权状态、设备注册时间，详细信息如表 35 所示。

表 35 su\_device 表结构

字段	类型	默认值	备注
mac	varchar(50)		mac 地址，主键
hostname	varchar(50)		设备主机名
ip	varchar(200)		设备 ip 地址
allow	tinyint(1)	0	授权状态，0：未授权，1：已授权
timestamp	timestamp	当前时间戳	设备注册时间

## 4.4 系统部分功能实现效果

### 1.QQ 小程序运行效果

启动小程序进入授权页，点击授权按钮进行授权后进入首页，点击“消息”和“我的”进入消息页和个人页，各步骤功能正常，界面显示无缺陷，视觉效果如图 18 所示。

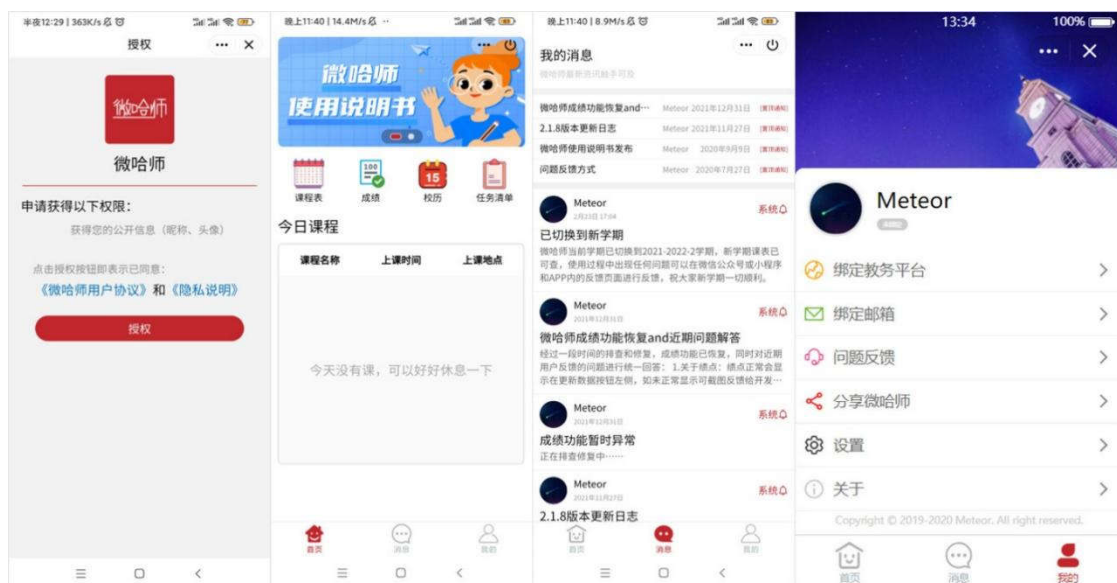


图 18 小程序启动流程视觉效果

点击个人页的“绑定教务平台”，在弹出的对话框中输入学号和正确的密码后点击“绑定”按钮完成绑定，各步骤功能正常，界面显示无缺陷，视觉效果如图 19 所示。

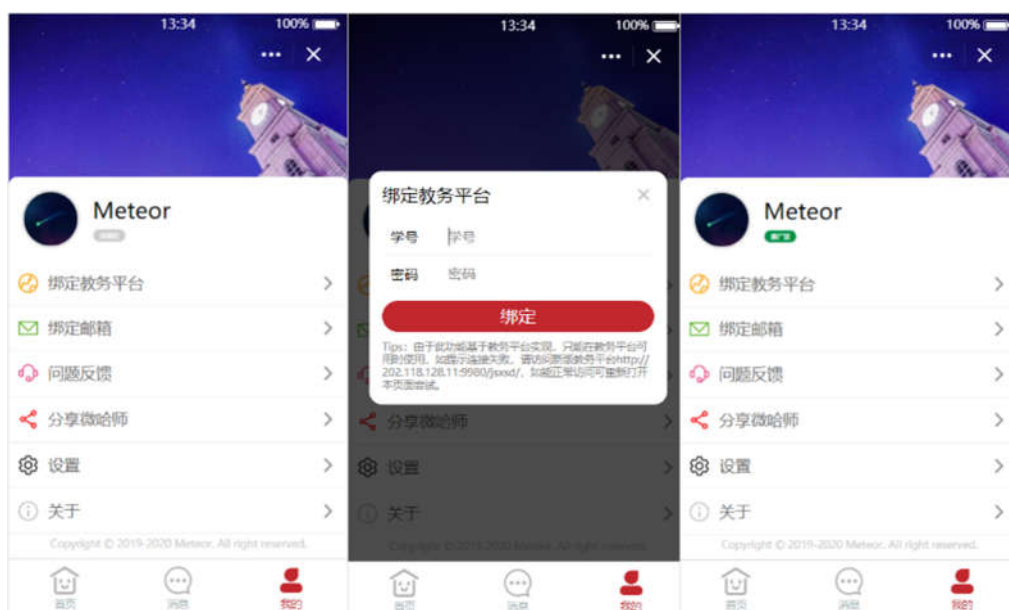


图 19 小程序绑定教务平台流程视觉效果

点击主页的“课程表”模块进入课表页，点击左上角的红色“导入”图标导入最新的课程数据，各步骤功能正常，界面显示无缺陷，视觉效果如图 20 所示。



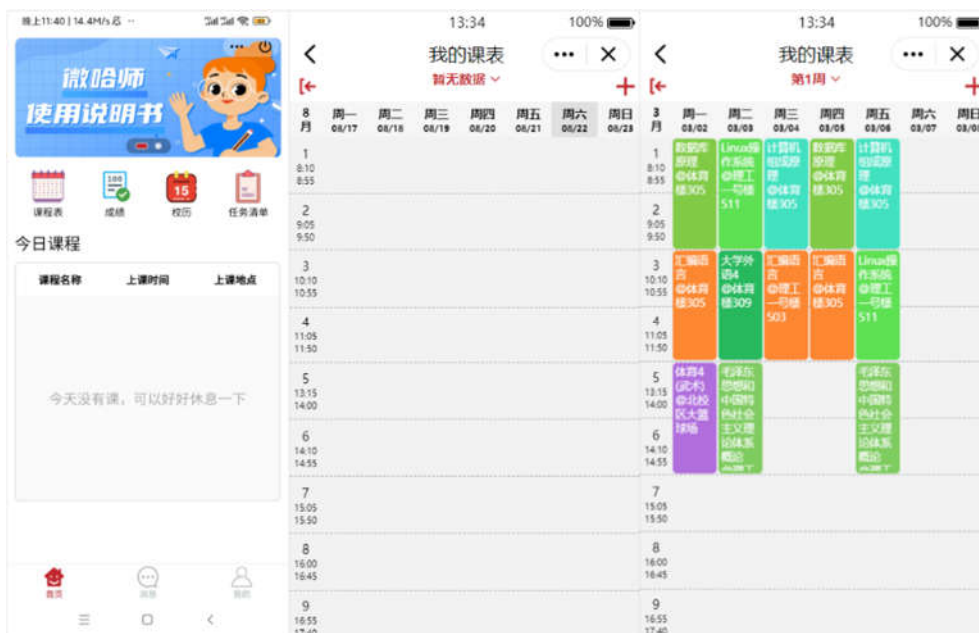


图 20 小程序查询课表流程视觉效果

点击主页的“成绩”模块进入成绩页，点击右上角的“更新课程数据”刷新成绩数据，各步骤功能正常，界面显示无缺陷，视觉效果如图 21 所示。



图 21 小程序查询成绩流程视觉效果

## 2.Android 客户端运行效果

启动 APP 进入登录页，点击“QQ 登录”按钮跳转到 QQ 授权登录页面，点击“授权登录”按钮进行授权后会返回登录页对 QQ 提供的授权信息进行处理，之后进入首页，各步骤功能正常，界面显示无缺陷，视觉效果如图 22 所示。

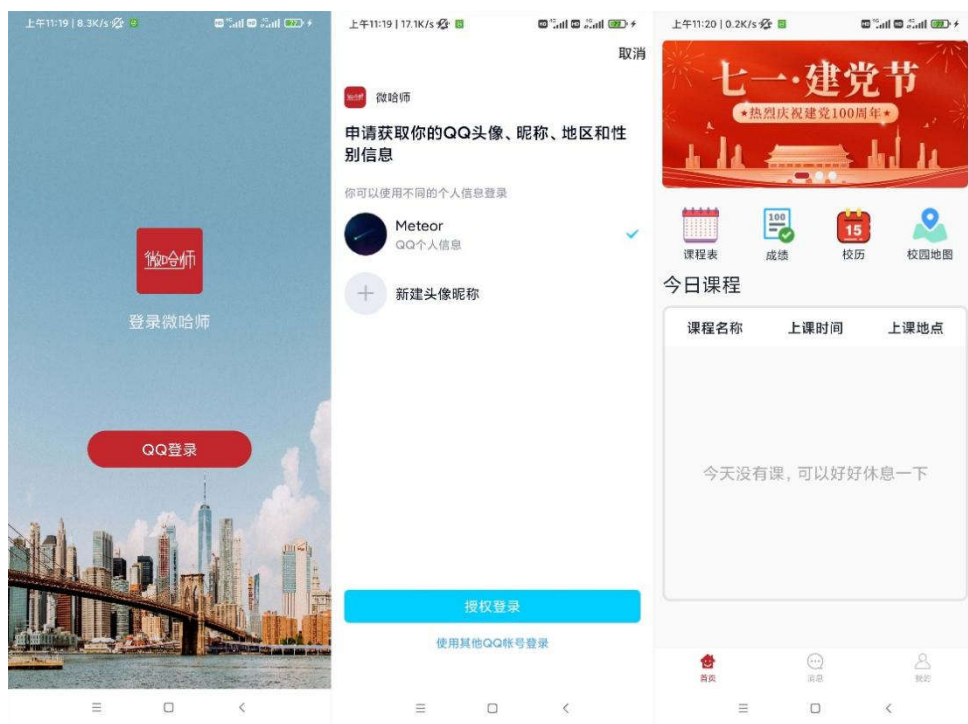


图 22 APP 登录流程视觉效果

点击首页的“校园地图”模块进入地图页即可查看校园地图，系统会根据用户所在校区初始化默认位置，如用户不在校区内初始化位置到江北校区行知楼。拖动下方滑杆可显示地点列表，选择对应项可显示导航路线（跨校区或定位在校外不显示路线）。各步骤功能正常，界面显示无缺陷，视觉效果如图 23 所示。



图 23 APP 登录流程视觉效果

## 5.系统运行与测试

### 5.1 运行环境配置

#### 1.负载均衡配置

服务端负载均衡使用 Nginx 实现，需要在 Nginx 的配置文件中的 upstream 部分配置被代理服务的地址和负载均衡的算法<sup>[13]</sup>，按照功能需求使用热备份配置。

#### 2.Redis 数据库配置

由于超级管理员设备授权和通知正文插入图片涉及到监听 Redis 数据库中 key 过期的需求，需要将 Redis 配置文件中的 notify-keyspace-events 参数设置为“Ex”来开启 Redis 的 key 过期事件通知<sup>[14]</sup>。

#### 3.测试环境配置

测试是项目开发过程中非常重要的一环，为了保证线上服务的稳定运行，测试工作应在独立的线下环境中进行。因此，需要在进行开发的计算机上进行测试环境搭建，线上环境与测试环境对比如表 36 所示。

表 36 线上环境与 QA 环境对比

服务	线上环境	测试环境
JDK	OpenJDK 8u292 Docker 镜像	JDK 1.8.0_191 + OpenJDK 8u292 Docker 镜像
MySQL	MariaDB 10.2.38 Docker 镜像	MySQL 5.7.26 - PHP study 集成环境
Redis	Redis 6.2.6 Docker 镜像	Redis 6.2.6 Docker 镜像
Web 服务	Nginx 1.21.3 Docker 镜像	Apache 2.4.39 - PHP study 集成环境
负载均衡服务	Nginx 1.21.5 Docker 镜像	单服务测试
PHP	PHP 7.4.8 Docker 镜像	PHP 7.3.4 nts - PHP study 集成环境

此外测试环境需要使用 Charles 进行服务地址映射，从而实现不修改请求地址请求测试服务。

### 5.2 项目运行数据

项目自第一个版本上线至今，据 QQ 小程序官方统计数据已有累计用户 1.5 万人以上，日活跃用户数均值达 20 人以上，日活跃用户数峰值近 500 人，日启动次数均值达 100 次以上，日启动次数峰值达 3000 次以上，近 30 日内次日活跃留存率均值达 30%以上，近 30 日内次日活跃留存率峰值达 60%以上，近 30 日内 7 次日活跃留存率均值达 20%以上，近 30 日内 7 次日活跃留存率峰值达 20%以上。

自超级管理员平台建立以来共统计 QQ 小程序和 Android 客户端用户总数 4000 余人,其中 2800 余人进行了教务平台的绑定,绑定比率达 60%以上。

### 5.3 系统测试

软件测试作为保证软件质量的重要手段,越来越引起人们的重视,而软件测试项目中存在着风险,如果能预先重视风险的评估,并对可能会出现的风险制定积极的应对计划,就可以在风险到来的时候,最大限度的避免风险或者降低风险所带来的损失<sup>[15]</sup>。在系统测试阶段使用黑盒测试从用户的角度,使用不同的测试数据或操作步骤对项目各个模块进行测试。

#### 1.QQ 小程序绑定教务平台测试

(1)启动 QQ 小程序,进入“我的”页面,用户头像和昵称显示正常,符合预期。

(2)点击“绑定教务平台”选项,弹出标题为“绑定教务平台”的对话框,对话框内有“学号和“密码”输入框以及“绑定”按钮,布局显示正常,符合预期。

(3)输入学号 2018000000,密码 123456,点击“绑定”,提示“用户名或密码错误”,符合预期。

(4)输入学号 2018020676,密码\*\*\*\*\*,点击“绑定”,提示“绑定成功”,“绑定教务平台”对话框关闭,符合预期。

(5)再次点击“绑定教务平台”选项,弹出“已绑定”对话框,对话框内显示用户真实姓名和学号以及“解绑”按钮,符合预期。

(6)QQ 绑定教务平台功能测试结束,各步骤操作结果均符合预期,测试通过。

#### 2.QQ 小程序查询课程表测试

(1)绑定教务平台后回到首页,点击“课程表”版块,进入课表页,页面未显示课表,教学周位置显示“暂无数据”,符合预期。

(2)点击左上方红色“导入”图标,页面刷新,正常显示当前教学周和当前教学周课程表,符合预期。

(3)点击任意一个课程块,弹出“课程详情”对话框,对话框内正确显示所选课程的课程名、课节、教室、教学周、教师信息,符合预期。

(4) QQ 小程序查询课程表功能测试结束,各步骤操作结果均符合预期,测试通过。

#### 3.Android 客户端校园地图权限请求和导航测试

(1)启动 Android 客户端,进入首页,点击“校园地图”版块,弹出系统授权弹窗,符合预期。

(2)点击拒绝,弹出对话框,提示“该权限为 APP 必需权限,请允许”,下方为“授权”

按钮，符合预期。

(3)点击“授权按钮”，再次弹出系统授权弹窗，符合预期。

(4)点击“拒绝且不再询问”，弹出对话框提示“请前往设置页面开启权限”，下方显示“确定”和取消按钮，符合预期。

(5)点击“确定”，跳转到系统 APP 应用信息页面，符合预期。

(6)手动开启 APP 定位权限，重启 APP，进入首页后再次点击“校园地图版块”，进入校园地图版块，符合预期。

(7)地图显示用户定位为十公寓，向上动页面下方滑杆，弹出地址选择器，选择“江北校区”-“公寓”-“八公寓”，地图正确显示起点和终点位置以及导航线路，符合预期。

(8) Android 客户端校园地图权限请求和导航功能测试结束，各步骤操作结果均符合预期，测试通过。

## 6 结论

本项目涉及用户角色和平台较多，工作量较大。经过两年多时间的更新迭代，项目目前整体涵盖了 QQ 小程序、Android 客户端、Web 前端、Windows 客户端、iOS 客户端、MacOS 客户端、Java 后端、数据库等多个领域内的前沿技术，实现了账户、教务、任务清单、校园地图、消息、班级、超级管理员设备授权、数据统计、APP 测试鉴权、Banner 和 APP 版本更新共十一个模块中包括课表查询、成绩查询、校内地图导航在内的共四十余个功能点。

在项目运营方面依照国家相关法律法规和相关平台政策规定实现了运营合规，拥有了较为可观的用户量，成为了一个开发和运营体系较为完善的软件系统。

最终项目的各端运行效果均达到设计预期，但仍有继续改进的空间，包括 QQ 小程序和 Android APP 的 UI 仍需进一步改进，服务端数据库查询语句仍可进一步优化，管理员平台学院管理员登录和管理功能尚未完善，超级管理员平台通知插入图片功能存在一定问题，教务相关接口仍有一定概率请求失败等。

此外，项目开发体验也仍有很大的提升空间，进行代码脱敏处理时由于缺乏完善的配置系统支持，将所有的配置信息存储到了本地环境变量，维护起来比较繁琐，如果能有完善的 CI/CD 系统和配置系统支持，项目开发效率会更高，后续将在这方面的学习和研发投入更多精力，争取开发一套高效的研发工具，为后续的开发工作做好准备。

## 参考文献:

- [1] 左佳丽,刘文珊,张玲乐.校园生活综合服务平台小程序的设计与实现[J].无线互联科技,2021,18(11):62-63+78.
- [2] 赵连方,韩冰.浅谈网站服务器安全防护管理[J].电子测试,2021(24):71-72+52.DOI:10.16520/j.cnki.1000-8519.2021.24.024.
- [3] 张雷.基于 Docker 的部署管理平台的研究与实现[D].中国石油大学(华东),2018.DOI:10.27644/d.cnki.gsydu.2018.001934.
- [4] 李尚林,陈宫,雷勇.基于 Java 的网络爬虫系统研究与设计[J].新型工业化,2021,11(04):74-77+80.DOI:10.19335/j.cnki.2095-6649.2021.4.029.
- [5] 高思.加密算法在电子身份标识中的应用[D].北京化工大学,2021.DOI:10.26939/d.cnki.gbhgu.2021.001390.
- [6] 朱心洲,胡誉,蒋靖妍.基于 QQ 小程序和 Flask 的高校闲置物品信息平台[J].计算机系统应用,2020,29(09):102-108.DOI:10.15888/j.cnki.csa.007565.
- [7] 卜凡港.基于 JavaMail 的 Web 邮件验证系统设计[J].电子技术与软件工程,2018(16):179-181.
- [8] 邹宗佑,关春花,潘璠,王萍.基于 Android 的校园地图设计[J].科技广场,2016(11):173-175.DOI:10.13838/j.cnki.kjgc.2016.11.042.
- [9] 王美荣,席何文.基于 Android 的指纹识别签到系统的设计实现[J].福建电脑,2019,35(03):63-65.DOI:10.16707/j.cnki.fjpc.2019.03.016.
- [10] 徐欣威.基于 ECharts 的科技统计数据可视化设计与实现[J].天津科技,2019,46(03):66-70.DOI:10.14099/j.cnki.tjkj.2019.03.018.
- [11] 崔慧娟.MVVM 模式在 Android 项目中的应用[J].信息与电脑(理论版),2021,33(06):1-3.
- [12] 何群芳,时招军.基于 Android 的内存泄漏与溢出研究[J].软件导刊,2018,17(02):50-52.
- [13] 赵琳.基于 CentOS 7 系统实现 Nginx 负载均衡[J].网络安全和信息化,2022(02):96-99.
- [14] 陈思.基于 Nginx 和 Redis 的高并发 Web 场景下缓存的研究与设计[D].东华理工大学,2021.DOI:10.27145/d.cnki.ghddc.2021.000350.
- [15] 蔡宁宁.计算机软件测试及风险管理[C]//第三十五届中国(天津)2021'IT、网络、信息技术、电子、仪器仪表创新学术会议论文集.,2021:307-309.DOI:10.26914/c.cnkihy.2021.013753.

# Design and Realization of WEIHASHI Campus Life Service Platform

Kang Guang-hui

**Abstract:** Due to the large number of students in our school, there will be some typical problems. For example, during the peak period of access to the educational affairs platform, the educational affairs platform server will have a relatively large pressure, and each operation process is slightly cumbersome.

The WEIHASHI Campus Life Service Platform bases on six types of user platforms of the WEIHASHI QQ Miniprogram, the WEIHASHI Android Client, the Administrator Platform, the Super Administrator Platform, the Full Platform Super Administrator Authorization Tool Client, and the Super Administrator Device Manager Client, serving five user roles: ordinary users, class cadres, class administrators, super administrators, and project owners. Using more advanced technologies, users can query the timetable, grades, and campus maps, task list. And identity authentication, user data statistics, publication and management of notice and questionnaire, banner management are available to administrators. A total of more than 40 function points in 11 modules were developed.

The project accumulated a considerable number of users in the long-term development iteration and operation, which has become a software system with a relatively complete development and operation system.

**Keywords:** Model-View-ViewModel; SpringBoot; Redis; full stack project