# 项目总结报告

## 一、项目概述

### 项目名称

同伴课堂

### 开发周期

11周（需求分析到测试与交付）

### 项目目标

本项目旨在开发同伴课堂平台——一个在线教育平台，专为支持和增强同伴教学法在实际教学中的应用而设计。该平台利用现代信息技术，为教师和学生提供一个便捷、高效、互动的学习环境。

## 二、项目技术架构

### 前端技术栈

* **网页框架**：Vue 3，用于构建用户界面，结合 Vite 提升构建效率。
* **移动端框架**：Fluttr，用于高效管理前端状态。

### 后端技术栈

* **框架**：Spring Boot 提供服务，支持高性能和快速开发。
* **数据库**：
  1. Mysql，存储平台数据，支持事务管理与复杂查询。
  2. oss数据库，存储照片等复杂信息。
* **身份认证**：JWT，用于用户身份验证和权限管理。

### 接口设计

* 使用Apifox进行API文档管理和测试，确保开发工具链的高效性。

## 三、项目完成的功能

### 学生端

* **注册与登录**：学生可以通过邮箱完成注册，使用密码登录。
* **课程管理**：学生通过课程码加入课程，并查看课程信息。
* **课堂互动**：支持课堂签到、课堂答题和数据统计查看。
* **消息通知**：接收课程公告和活动提醒。

### 教师端

* **课程管理**：教师创建和管理课程，包括添加课程简介和成员。
* **课堂管理**：创建课堂，发布课堂问题、实时监控学生答题情况。
* **题库管理**：增删查改题库中的题目，为课堂活动准备测试题。
* **数据统计**：查看学生课堂表现和学习数据。

## 四、项目执行情况

### 总体情况

* **接口设计**：所有同学垂直分工，共同完成。
* **前端开发**：负责学生端APP和教师端的网页界面设计与功能实现。
* **后端开发**：完成数据库设计与服务部署。
* **测试与优化**：团队所有成员参与测试，修复问题并优化功能。

### 资源使用情况

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 | 计划时间 | 实际完成时间 | 偏差原因 |
| 需求分析与任务分解 | 1 周 | 1 周 | 无 |
| 开发基础知识学习 | 3 周 | 3 周 | 按计划完成 |
| 概要设计与UI设计 | 1 周 | 1 周 | 按计划完成 |
| 详细设计与接口编写 | 1 周 | 2 周 | 数据表定义不完整，接口定义需要反复修改 |
| 前后端开发 | 4 周 | 4 周 | 前后端连接时需要时间协同 |
| 软件测试 | 1周 | 1周 | 按计划完成 |
| 项目汇报 | 1周 | 1周 | 按计划完成 |

## 五、风险与问题回顾

### 1. 前后端联调问题

* **问题描述**：项目初期，前后端接口文档不够详细，导致接口定义不一致，前后端开发人员在调用接口时存在理解偏差，联调过程中多次出现问题，导致开发进度拖延。
* **解决措施**：
  1. **补充接口文档**：为了确保接口一致性，我们重新审视了需求文档，完善了接口定义，加入了详细的请求和响应字段说明。
  2. **引入 API 验证工具**：使用 **Apifox** 工具进行接口文档的管理和测试，前后端分别连接**Apifox**模拟接口，提前验证接口的正确性，减少了联调阶段的错误。
  3. **定期沟通**：开发过程中，我们设置了前后端同步小组会议，每天及时讨论和解决接口问题，确保开发进度同步，联调问题可以第一时间解决。

### 2. 图像上传功能问题

* **问题描述**：图像上传功能在初步开发时，由于没有考虑到不同大小和格式的图像，导致部分文件上传时出错。
* **解决措施**：
  1. **增加文件校验**：在图像上传前，加入了文件格式和大小的校验，限制上传的图像大小，避免出现过大文件导致的错误。
  2. **使用云存储**：将上传的图片存储在 **云服务器** 上，减轻了本地服务器的压力，提高了上传和存储的可靠性。

### 3. 用户权限管理问题

* **问题描述**：在项目的初期，用户权限管理功能不够完善，导致部分用户（如学生和教师）能够访问不该访问的模块或数据，造成了安全隐患。
* **解决措施**：
  1. **权限角色细化**：我们将用户权限进一步细化，增加了 **学生端**、**教师端** 的角色管理，确保每个角色只能访问自己的权限范围。
  2. **JWT 身份验证**：使用 **JWT** （JSON Web Token） 实现了前后端分离后的身份认证和权限验证，确保每个请求都经过授权验证。

### 4. 用户界面设计与交互体验问题

* **问题描述**：初期的用户界面设计较为简陋，学生和教师端在功能操作时难以找到所需内容，用户体验差，界面不美观，功能不完善。
* **解决措施**：
  1. **优化界面设计**：重新编写用户界面，并完善功能以及状态。
  2. **逐步迭代**：开发者在一次次使用中，及时察觉逻辑漏洞，和功能缺失的问题并及时纠正。

### 5. 二维码信息编入签到返回的问题

* **问题描述**：
  1. **无法识别问题**:二维码在生成后无法正确编码课程ID或签到信息，导致学生扫描二维码后，无法识别签到对应的课程或签到状态。
  2. **数据同步问题**：二维码生成时，后台生成的签到信息没有同步到数据库，学生扫码签到时，系统未能找到对应的签到记录，出现了签到状态不更新的问题。
* **解决措施**：
  1. **访问二维码信息编码**:在二维生成过程中，我们保证二维码信息只包含必要的数据字段，比如**课程ID**、**签到ID**。同时限制二维码的大小，保证二维码的大小适中，用于扫描和解析。
  2. **实时同步信息**： 在数据库中加入“签到状态”字段，确保每个学生签到后，状态能够及时更新，并返回当前签到状态。

## 六、成果与经验教训

### 主要成果

1. 平台完成了需求范围内的所有功能，涵盖用户管理、课程管理、题库管理、课堂互动和统计分析等核心模块，并设置了验证码注册，二维码扫码等多个特色功能，APP流畅精美，网页功能完善。
2. 后端采用高效架构，及时处理用户请求并相应，可存储并管理大量信息。

### 经验教训

1. **充分前期规划与设计的重要性**： 项目初期没有对一些关键环节进行充分的规划，导致后期频繁出现问题。例如，前后端接口文档的不一致、权限管理的设计不完善、二维码信息编入的问题等。这些问题在项目后期不仅增加了开发的难度，还影响了开发进度和用户体验。对于系统中涉及到核心功能的模块（如接口、权限管理、数据库设计等），必须在项目初期就进行详细设计，并明确每个模块的责任和接口定义。文档应全面详细，确保团队成员能够对项目有一致的理解，减少开发中的反复修改。
2. **需求变更与及时沟通**： 在开发过程中，部分功能需求发生了变化，尤其是在前后端联调和用户界面设计方面。虽然这种需求变更是不可避免的，但没有及时同步到团队成员，导致了不必要的重复工作和浪费。团队内前期缺乏有效的沟通，特别是前后端、开发与设计之间的沟通不够紧密，导致了接口定义的偏差和用户体验的不足。为避免此类问题，应该在项目早期就建立明确的沟通机制，确保所有团队成员对需求变更和进度有所了解，并根据需要灵活调整开发计划。
3. **测试与优化的重要性**： 测试和性能优化是项目成功的关键环节，比如二维码信息同步机制没有充分考虑到实时性，导致签到时出现数据延迟或错误。这些问题如果在早期的测试中得到及时发现，就能避免后期开发和维护过程中的大量返工。因此，团队可以项目初期就制定严格的测试和性能优化计划，包括单元测试、集成测试、负载测试等，确保系统的稳定性和性能在高负载和复杂场景下也能保证。
4. **数据一致性与安全性**：在开发过程中，我们遇到了数据同步和安全性的问题，特别是在二维码信息同步和验证码验证功能上。由于缺乏有效的数据同步机制，导致二维码扫码后，学生签到信息未能及时更新到数据库；同时，验证码的有效期和校验机制也存在一定的疏漏。为避免这种情况，系统设计中必须考虑到数据的一致性和安全性，确保每一个操作都能够得到及时反馈，并且所有关键数据都能够正确同步、正确存储。

## 七、收获与感悟

通过本次项目的开发，团队成员在多个方面获得了成长，并且深刻体会到了团队合作、技术应用和项目管理的重要性。以下是我们在项目开发过程中的主要收获与感悟：

在本次项目的开发过程中，团队合作的重要性得到了充分的体现。项目的顺利推进离不开每位成员的紧密协作，尤其是在前后端开发和技术实现方面。前后端开发团队通过清晰的沟通和共同解决问题，不仅提高了开发效率，也促进了团队协作能力的提升。特别是在前后端联调的过程中，我们通过详细设计接口文档并保持充分的沟通，最终克服了复杂性带来的挑战。每个团队成员都在自己负责的模块上发挥了重要作用，确保了项目各部分能够顺利对接和实现。

在技术方面，我们也取得了很多突破。通过使用 **Flutter**，我们成功实现了跨平台应用开发，能够同时支持 **Android**、**iOS** 和桌面端的需求。在后端开发上，我们熟练使用了 **Spring Boot** 等技术，搭建了高效的数据访问层，确保了系统的稳定性和扩展性。在遇到技术困难瓶颈时，团队不断沟通，协调，这个过程中让团队成员的责任感和解决问题的能力得到了进一步加强。

在时间安排方面，团队成员也面临了不小的挑战。由于部分成员正在准备考研，另一些成员需要同时兼顾实习任务，大家的空闲时间并不完全重合，这给项目的整体推进带来了很大的压力。软件开发并非完全并行，某些模块的完成依赖于前序模块的交付。在这种情况下，我们进行多轮任务调整，确保每位成员能够在有限的时间内完成任务。

这次项目经历不仅提升了我们的技术能力，也让我们更加深刻地理解了团队合作和时间管理的重要性。每位成员的努力和支持，使得我们能够在困难中找到解决方案，推动项目最终顺利完成。

### 总结

通过本次项目开发，团队成员不仅在技术上获得了显著成长，更在团队协作、沟通与解决问题的能力上得到了提升。通过前后端分离的开发模式，我们在实现 **Vue 3** 和 **Flutter** 的同时，也进一步加深了对 **Spring Boot** 后端技术的理解。通过对高并发问题、性能优化、数据一致性等方面的解决，我们为后续的项目开发积累了宝贵经验。