

项目进展报告

目录

- 项目进展报告 1
- 一、引言 2
- 二、项目概述 2
 - 2.1 项目的背景 2
 - 2.2 项目目标 2
 - 2.3 整体框架 2
- 三、项目进度概况 4
 - 3.1 项目进度 4
 - 3.2 项目状态总体评估 5
- 四、关键任务和阶段性成果 5
 - 4.1.1 第一阶段 5
 - 4.1.2 第二阶段 6
 - 4.1.3 第三阶段 6
- 五、风险和问题 7
 - 5.1 接口定义不清晰 7
 - 5.2 频繁的变更需求 7
 - 5.3 前后端技术栈不匹配 7
 - 5.4 缺乏有效的沟通 7
- 六、 下一步计划 7
- 七、 总结和建议 8

一、引言

通过前期调查研究，我们选择设计了一款在线研讨室预约小程序。该小程序能随时查看校园内的研讨室使用情况，并且按需预约，将极大的方便学生的学习。本报告专用于监控小程序项目的发展情况。

二、项目概述

2.1 项目的背景

随着网络技术的发展，计算机应用的日益普及和深化，各种软件，小程序使我们的生活更加方便快捷。为了满足学生们预约使用研讨室的使用需求，我们设计了一款在线研讨室预约小程序。该小程序能随时查看校园内的研讨室使用情况，并且按需预约，将极大的方便学生的学习。

2.2 项目目标

完成研讨室预约小程序的设计与实现。为实现项目的总目标可按以下三个阶段目标来进行：

第一阶段目标：实现研讨室预约小程序的基本功能，小组各个成员进行各个模块的开发，形成初步的系统。

第二阶段目标：攻克技术上的难题，实现研讨室预约小程序的一些特殊功能，进一步完善系统。

第三阶段目标：让系统投入到实际运用中，做好系统的维护工作。

2.3 整体框架

2.3.1 前端：

- 使用 uniapp+vue 框架，包含多个页面：
 - ◆ 预约表单页面：用于用户选择日期、时间和课室，提交预约请求。
 - ◆ 课室浏览页面：显示可预约的课室列表，提供查看教室详情和进行预约的功能。

- ◆ 其他可能的页面：如用户个人中心、预约历史等。
- 通过小程序框架实现前端逻辑，包括：
 - ◆ 处理用户输入，验证和格式化数据。
 - ◆ 调用微信小程序提供的 API，如获取用户信息、发送网络请求等。
 - ◆ 与后端通过 HTTP 协议进行通信，发送预约请求、接收后端返回的数据。

2.3.2 后端

- 使用 Node.js 作为后端语言，并基于 Express.js 框架构建后端服务。
- 提供 RESTful API，包括但不限于：
 - ◆ 用户认证接口：处理用户登录、获取用户信息等。
 - ◆ 课室信息查询接口：返回可预约的课室列表。
 - ◆ 预约处理接口：处理用户提交的预约请求，检测时间冲突等。
 - ◆ 数据库操作接口：负责与数据库交互，执行相关的 CRUD 操作。
- 处理业务逻辑，包括：
 - ◆ 用户身份认知：验证用户的登录状态，确保只有授权用户可以进行预约。
 - ◆ 课室预约逻辑：检测预约时间是否冲突，更新数据库中的预约记录。
 - ◆ 数据库操作：使用 ORM（对象关系映射）或其他方式与 MySQL 数据库交互。

2.3.3 数据库

- 使用 MySQL 关系型数据库，包括但不限于以下表：
 - ◆ 用户表(User)：存储用户信息，包括用户名、密码（加密存储）、权限等。
 - ◆ 课室表(Classroom)：存储课室信息，包括课室编号、容量等。
 - ◆ 预约表(Reservation)：记录用户的预约信息，包括预约时间、用户 ID、课室 ID 等。
- 表之间通过外键关联，确保数据的完整性和一致性。

2.3.4 主要组件之间的关系

- 前端与后端：

- ◆ 用户在小程序前端提交预约请求。
- ◆ 小程序通过 HTTP 请求调用后端提供的 RESTful API，传递用户的预约信息。
- ◆ 后端接收请求，处理业务逻辑，与数据库交互，最终返回结果给前端。
- 后端与数据库：
 - ◆ 后端通过数据库连接池与 MySQL 数据库通信，执行数据库操作。
 - ◆ 后端处理用户预约请求时，更新数据库中的预约表。
- 前端与数据库：
 - ◆ 前端通过调用后端提供的 API 与数据库间接交互。
 - ◆ 通过后端的 API，前端可以获取课室列表、用户信息等。

这样的体系结构允许各个组件专注于其核心任务，同时通过清晰定义的 API 实现了松耦合，使得系统更加灵活和可维护。

三、项目进度概况

3.1 项目进度

3.1.1 第一阶段

已完成的工作：前端管理员端、用户端 UI 界面设计完成，分为前端和后端两个组进行各个模块开发。第一次开会时，前端用户端除“我的”界面内我的历史预约信息部分还未开始编码，登录、公告、预约、我的这四个静态界面已编码完毕，后端公告、预约、登录接口已编码、测试完毕，初步预约系统已形成。

正在进行的工作：前后端联调，管理员端相关接口设计。

尚未开始的工作：管理员端静态界面编码。

3.1.2 第二阶段

已完成的工作：前端静态界面基本编码完毕，后端接口设计基本编码完毕。

正在进行的工作：前后端联调，实现用户端预约、登录、显示公告的功能，攻克技术上的难题，实现研讨室预约小程序的一些特殊功能，进一步完善系统。

尚未开始的工作：相关文档撰写。

3.1.3 第三阶段

已完成的工作：管理员端、用户端前后端联调已完成，可以实现基本的预约功能。

正在进行的工作：相关技术文档撰写。

尚未完成的工作：让系统投入到实际运用中，做好系统的维护工作

3.2 项目状态总体评估

项目各阶段进行基本符合计划。

项目已完成大部分的界面设计和编码工作，并且已经进行了一些重要功能的联调和测试。然而，仍有一些重要工作尚未完成，如系统投入实际运用以及维护工作。

下一步行动：解决技术上的挑战，确保系统功能的完善和稳定，最终将系统顺利投入实际应用中。

四、关键任务和阶段性成果

4.1.1 第一阶段

关键任务：

1. 完成前端管理员端、用户端 UI 界面设计。
2. 分别进行前端和后端各个模块的开发。
3. 编码登录、公告、预约、我的四个静态界面。
4. 设计并编码后端公告、预约、登录接口。
5. 前后端联调，准备管理员端相关接口设计。
6. 形成初步的预约系统。

阶段性成果：

1. 前端管理员端、用户端 UI 界面设计完成。
2. 完成登录、公告、预约、我的四个静态界面编码。
3. 后端公告、预约、登录接口编码并测试完毕。
4. 形成初步的预约系统。

4.1.2 第二阶段

关键任务：

1. 完善前端静态界面编码。
2. 实现用户端预约、登录、显示公告的功能。
3. 解决技术难题，特别是研讨室预约小程序的特殊功能。
4. 前后端联调，确保系统功能稳定性。
5. 进一步完善系统，提高系统的可用性和用户体验。
6. 开始相关文档的撰写，包括技术文档和用户手册等。

阶段性成果：

1. 完成前端静态界面基本编码。
2. 实现了用户端预约、登录、显示公告的基本功能。
3. 解决了一些技术上的难题，包括研讨室预约小程序的特殊功能。
4. 前后端联调，确保系统基本功能的正常运行。
5. 进一步完善系统，提高系统的稳定性和用户体验。
6. 开始了相关文档的起草，为系统上线做准备。

4.1.3 第三阶段

关键任务：

1. 完成管理员端、用户端前后端联调。
2. 编写相关技术文档。
3. 将系统投入到实际运用中，并做好系统的维护工作。

阶段性成果：

1. 完成了管理员端、用户端前后端联调，可以实现基本的预约功能。
2. 完成了相关的技术文档撰写。
3. 将系统投入实际应用中，并启动系统的维护工作，确保系统正常运行和及时修复问题。

五、风险和问题

5.1 接口定义不清晰

问题：前后端在定义上存在歧义，导致开发过程中的不一致。

解决方法：在项目开始阶段，确保段后端开发团队共同制定清晰的接口文档。
使用 Apifox 工具帮助定义和共享接口规范。

5.2 频繁的变更需求

问题：后端或前端频繁修改需求，导致对方需要频繁调整代码。

解决方法：在项目启动前，尽量明确项目需求，确保前后端开发团队之间的需求变更得到及时沟通和确定。

5.3 前后端技术栈不匹配

问题：前端和后端使用不同得技术栈，导致集成和协同开发时出现问题。

解决方法：在项目初期确定一致得技术栈，并确保前后端团队具备相应得技术能力。进行组内培训或知识分享以减少技术栈不匹配得问题。

5.4 缺乏有效的沟通

问题：缺乏及时、清晰得沟通可能导致误解和不必要的问题。

解决方法：确保及时回复问题，避免信息滞后。定期召开会议进行项目进度和问题的讨论。

六、 下一步计划

1. 解决当前面临的挑战：

· 针对技术难题和特殊功能，进行更深入的研究和解决方案的制定，确保系统功能的完善和稳定。

2. 推进进展并完善系统：

- 持续进行前后端联调，确保系统各功能模块协调顺畅、无故障。
- 进一步完善系统，提升用户体验和系统的可用性，确保界面友好、响应迅

速。

3. 准备系统投入实际运用：

- 制定系统上线计划，确定上线时间和过渡方案。
- 做好系统上线前的最后测试和检查，确保系统的稳定性和安全性。

4. 启动系统投入实际运用：

- 将系统正式投入实际使用中，监测系统运行情况。
- 提供必要的培训和支持，确保用户能够顺利使用系统。

5. 开始系统的维护和优化：

- 建立系统维护机制，包括定期检查、bug 修复和系统优化。
- 收集用户反馈，根据用户需求持续改进和优化系统。

6. 扩展系统功能：

- 系统功能不在局限于校园研讨室预约，额外上线图书馆预约、体育场预约、实验室预约等功能，完善整个校园预约系统。

七、 总结和建议

项目已完成了校园研讨室预约系统的基本编码，包括前端和后端的开发工作，用户界面设计、静态界面编码以及后端接口设计等任务。同时，已进行了一些前后端联调工作，取得了阶段性的成果。系统已具备基本预约功能的使用能力。然而，仍有一些关键任务需要完成，并面临着技术挑战和后续工作需要处理。此外，本项目仍然需要团队共同努力来完成剩余任务，解决挑战，并确保系统成功投入实际应用。在此基础上，持续优化和维护系统将是项目的后续重点工作，这将有助于确保系统长期稳定运行和用户满意度。

建议：

1. 完善相关文档（如技术文档、用户手册等）帮助系统的全面使用和后续的维护工作。
2. 针对研讨室预约小程序等特殊功能和技术难题，应加大研究和解决力度，争取尽快找到可行的解决方案，确保系统的完善和稳定性。
3. 虽然研讨室预约的基本编码已完成，但仍需不断优化系统，提高用户体验，确保界面友好且反应迅速。除此之外，考虑之后上线其他课室的预约，如图书馆预约、体育场预约、实验室预约等功能。

4. 准备系统上线并制定过渡方案，进行最后的测试和检查，确保系统的稳定性和安全性。一旦投入使用，要持续监测系统运行情况，并建立系统的维护机制，及时修复问题。

5. 加强团队内部的沟通与合作，确保团队成员间的协作顺畅，及时解决问题和调整方向。