****

### 课程推荐APP项目WBS任务分解结构



项目名称 课程推荐app

小组组长 陈灵灵

小组成员 戚子奕

小组成员 程瑞慧

小组成员 黎婉莹

小组成员 林炜莹

指导老师 杨枨

目录

[课程推荐APP项目WBS任务分解结构 1](#_Toc193028315)

[**一、项目阶段划分** 3](#_Toc193028316)

[**二、需求工程阶段划分** 3](#_Toc193028317)

[**三、WBS任务分解结构** 3](#_Toc193028318)

**一、项目阶段划分**

* 1. 需求调研与分析阶段
* 2. 系统设计与原型开发阶段
* 3. 前后端功能实现阶段
* 4. 测试优化阶段
* 5. 部署推广阶段

**二、需求工程阶段划分**

* 1. 需求调研与分析阶段
* 2. 需求规格说明书编制阶段

**三、WBS任务分解结构**

| **编号** | **任务名称** | **任务描述** |
| --- | --- | --- |
| 1 | 需求调研与分析 | 通过问卷调查、访谈等方式收集学生、教师对课程推荐的需求和期望，分析现有教务系统的局限性，明确新系统的目标和功能要求 |
| 1.1 | 用户需求收集 | 设计调查问卷，对学生、教师进行问卷调查，了解他们在选课过程中的痛点和对推荐功能的期望 |
| 1.2 | 用户需求整理 | 对收集到的问卷数据进行整理和分析，提炼出用户的核心需求和功能点 |
| 1.3 | 现有系统分析 | 对学校现有教务系统的选课流程、数据结构、功能模块进行详细分析，找出其在课程推荐方面的不足 |
| 1.4 | 需求优先级评估 | 根据用户需求的重要性和紧急性，对各项需求进行优先级排序，为后续开发提供指导 |
| 2 | 需求规格说明书编制 | 根据需求调研与分析的结果，编写详细的需求规格说明书，明确系统的功能需求、性能需求、用户界面需求等 |
| 2.1 | 功能需求定义 | 详细描述系统的各项功能，包括课程推荐、用户管理、数据对接等，明确每个功能的输入、输出和处理逻辑 |
| 2.2 | 性能需求定义 | 规定系统的性能指标，如响应时间、并发处理能力、数据处理效率等，确保系统能够满足用户的使用要求 |
| 2.3 | 用户界面需求定义 | 设计系统的用户界面布局、交互方式、视觉风格等，确保界面友好、操作简便、符合用户的使用习惯 |
| 2.4 | 安全与保密需求定义 | 明确系统的安全要求，包括用户认证、数据加密、权限控制等，保护学生的个人信息和选课数据的安全 |
| 3 | 系统设计与原型开发 | 根据需求规格说明书，进行系统的总体设计和详细设计，开发系统原型，验证需求的可行性和技术方案的合理性 |
| 3.1 | 总体设计 | 设计系统的架构，包括前后端分离架构、数据库设计、接口设计等，确定系统的整体框架和技术选型 |
| 3.2 | 详细设计 | 对系统的每个功能模块进行详细设计，包括算法设计、数据结构设计、界面设计等，为开发提供具体的实现方案 |
| 3.3 | 原型开发 | 使用原型工具（如Axure、Sketch等）开发系统的原型，展示系统的界面布局、交互流程和主要功能，与用户进行沟通和确认 |
| 3.4 | 设计评审 | 组织项目团队、用户代表、技术专家等对系统设计和原型进行评审，检查设计的合理性和完整性，提出改进意见和建议 |
| 4 | 前端功能实现 | 基于React/Vue框架进行前端开发，实现系统的用户界面和交互功能，确保前端代码的高效性、兼容性和可维护性 |
| 4.1 | 用户界面开发 | 根据设计稿，开发系统的各个页面，包括首页、课程推荐页、课程详情页、用户个人中心等，确保界面美观、布局合理、操作便捷 |
| 4.2 | 交互功能开发 | 实现用户的登录、注册、课程搜索、筛选、排序、评论、点赞等交互功能，通过API与后端进行数据交互，确保交互的流畅性和数据的实时性 |
| 4.3 | 响应式设计 | 对前端页面进行响应式设计，使其能够自适应不同尺寸的设备（如手机、平板、电脑等），提供一致的用户体验 |
| 4.4 | 前端性能优化 | 对前端代码进行优化，包括代码压缩、图片优化、缓存策略等，提高页面的加载速度和渲染性能，提升用户体验 |
| 5 | 后端功能实现 | 基于Node.js搭建后端服务，实现系统的业务逻辑和数据处理功能，确保后端的稳定性和高效性 |
| 5.1 | 用户管理模块开发 | 实现用户的注册、登录、信息修改、权限管理等功能，对用户数据进行存储和管理，确保用户数据的安全性和完整性 |
| 5.2 | 课程管理模块开发 | 实现课程信息的存储、查询、更新、删除等功能，包括课程的基本信息、教学大纲、授课教师、课程评价等数据的管理 |
| 5.3 | 推荐算法模块开发 | 实现动态权重算法，根据用户的个人兴趣、学习目标、历史选课记录等多维度数据，为用户提供个性化的课程推荐 |
| 5.4 | 数据对接模块开发 | 实现与学校教务系统的数据对接，获取课程信息和学生选课数据，确保数据的准确性和实时性 |
| 6 | 测试优化 | 对系统进行全面的测试，包括功能测试、性能测试、安全测试等，发现并修复问题，优化系统性能和用户体验 |
| 6.1 | 功能测试 | 对系统的各项功能进行测试，确保功能的正确性和完整性，包括课程推荐的准确性、用户交互的流畅性、数据的正确性等 |
| 6.2 | 性能测试 | 对系统的响应时间、并发处理能力、数据处理效率等性能指标进行测试，分析性能瓶颈，进行优化调整 |
| 6.3 | 安全测试 | 对系统的安全性进行测试，包括用户认证、数据加密、权限控制等方面，确保系统的安全性和数据的保密性 |
| 6.4 | 用户体验测试 | 邀请部分学生和教师进行实际使用测试，收集他们的反馈和意见，对系统的界面设计、交互流程、功能实用性等方面进行优化 |
| 7 | 部署推广 | 将系统部署到生产环境，进行上线推广，确保系统能够稳定运行，并为用户提供良好的服务 |
| 7.1 | 环境搭建 | 在服务器上搭建系统的运行环境，包括操作系统、数据库、Web服务器等，确保环境的稳定性和兼容性 |
| 7.2 | 系统部署 | 将开发好的前端和后端代码部署到服务器上，进行配置和调试，确保系统能够正常启动和运行 |
| 7.3 | 用户培训 | 为学生和教师提供系统的使用培训，包括操作指南、视频教程等，帮助他们快速掌握系统的使用方法 |
| 7.4 | 推广宣传 | 通过校园网、微信公众号、宣传海报等方式对系统进行推广宣传，提高系统的知名度和使用率 |