在线购票平台

软件项目计划

**（简化版）**

版本 <3.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2021.7.12 | 1.0 | 初始版本 | 李天策 |
| 2021.7.18 | 2.0 | 修改了风险分析 | 胡尧、闫星羽 |
| 2021.9.8 | 3.0 | 调整了开发计划 | 任浩天、李天策 |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

2. 项目概述 4

2.1 项目的目的、规模和目标 4

2.2 假设与约束 4

2.3 项目的可交付成果 4

3. 项目组织 4

4. 项目计划 4

4.1 风险分析 4

4.2 方法和工具 5

4.3 开发计划 5

4.4 质量保证计划 5

4.5 项目沟通计划 5

4.6 培训计划 5

5. 附录 5

软件项目计划

# 简介

## 目的

确定详细的项目实施范围、定义递交的工作成果、评估实施过程中主要的风险、制定项目实施的时间计划、成本和预算计划、人力资源计划，合理地估算软件项目开发所需的资源 、经费和开发进度，并控制软件项目开发过程按此计划进行。

## 范围

《在线购票平台》项目，以及其开发过程中产生的相关代码、文档。

## 定义、首字母缩写词和缩略语

## 参考资料

《软件工程原理》 沈备军、陈昊鹏、陈雨婷，

《软件项目计划》模板，

# 项目概述

## 项目的目的、规模和目标

目的：开发并部署一个稳定且高性能的在线购票平台。

规模：能够支持2000人的同时在线使用。

目标：能够实现账户管理、信息查询、购票等基本功能,并支持高并发下的系统的稳定性与正确性。

## 假设与约束

假设： 1.项目提供预算充足。

2.项目成员能够快速掌握新技术。

3.项目风险能有效规避。

预算：1200元

人员：《互联网产品设计与开发》第三小组所有成员（李天策、任浩天、胡尧、闫星羽）

设备：4台电脑及可能的云服务器

时间表：8月6日进行第一次项目集中检验，9月13日进行项目的最终答辩。

## 项目的可交付成果

|  |  |
| --- | --- |
| 序号 | 可交付成果 |
| 1 | 《软件项目计划》 |
| 2 | 《迭代计划》 |
| 3 | 《迭代评估报告》 |
| 4 | 《SRS文档》和用例模型 |
| 5 | 《软件架构文档》和基于UML的分析设计模型 |
| 6 | 《测试用例》和《测试报告》 |
| 7 | 《项目总结报告》 |
| 8 | 源代码和可执行代码 |
| 9 | 演示视频文件（包括安装、运行、功能等） |
| 10 | 演示PPT |

# 项目组织

1. 李天策（组长）：数据爬取，图数据库设计和查询实现
2. 闫星羽（组员）：导入数据库数据，实现购票业务逻辑
3. 任浩天（组员）：app界面设计，实现用户身份验证和信息管理
4. 胡尧（组员）：数据库结构设计，实现票务查询业务逻辑

# 项目计划

## 风险分析

1. 技术风险，如成员掌握新技术速度较慢，框架、工具选择错误等情况，如果出现，则会大大加重之后的工作压力。而且，若在项目进度过半的情况下出现技术选型无法实现需求等问题，则需要
2. 需求风险，支持高并发并维持系统正确运行是本项目中较为关键且实现难度较高的需求，需要投入大量学习时间和不断尝试，并测试其稳定性。
3. 进度风险，本次项目开发在第一次集中检验前仅有四周时间，整个项目开发过程仅有八周时间，需要合理安排每轮迭代的计划和任务。
4. 预算风险，本次项目初始预算为1111元，用以购买华为云服务，大约可以维持两个月，存在预算不足以支持项目最终部署的风险。

|  |  |  |  |
| --- | --- | --- | --- |
| **优先级** | **风险名称** | **风险描述** | **风险缓解措施** |
| 高 | 技术学习风险 | 项目成员学习新技术花费时间较长，无法跟上开发进度 | 项目内进行培训 |
| 高 | 需求实现风险 | 无法实现高并发、高性能需求的风险 | 合理安排迭代计划，提早验证并解决风险。 |
| 高 | 数据获取风险 | 爬取数据失败，无法获得所需的数据资源 | 迭代初期提前进行测试，或购买第三方API |
| 中 | 进度缓慢风险 | 在八周时间内无法完成项目目标的风险 | 合理规划迭代计划，及时变更以适应开发进度。 |
| 低 | 预算不足风险 | 预算不足以支撑部署经费的风险 | 合理使用预算，节源。 |

## 方法和工具

IDE: IDEA、Eclipse

测试工具：xUnit

版本管理工具：Git

项目管理工具：Git

## 开发计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **迭代名称** | **起止时间** | **所缓解的风险** | **需完成的任务** | **需提交的成果** |
| 环境配置和前端搭建 | 7.12-7.20 | 技术风险和需求风险 | 项目环境搭建，选择架构类型，学习相关技术 | 《软件项目计划》、《迭代计划》、《迭代评估报告》、《SRS文档》和用例模型、源代码和可执行代码 |
| 后端基本功能实现、数据获取 | 7.21-8.6 | 需求风险和数据获取风险 | 设计前端界面；后端基本功能实现 | 《迭代计划》、《迭代评估报告》、《软件架构文档》和基于UML的分析设计模型、源代码和可执行代码 |
| 并发和安全性优化 | 8.19-9.12 | 进度缓慢风险和需求风险 | 保证用户账号安全、保证高并发抢票的准确性 | 《迭代计划》、《迭代评估报告》、源代码和可执行代码、《测试用例》和《测试报告》、《项目总结报告》、演示视频文件、演示PPT |

## 质量保证计划

需求评审：每次迭代的开始以及迭代过程中不断进行，针对当前需求的适用性进行评估不断改进完善，由小组所有成员共同进行。

设计评审：在设计完成后由出设计人员外的其他项目组成员共同进行，针对设计的可靠性，可用性以及合理性等方面进行评估反馈，并传达给负责设计的小组成员

代码评审：每次代码PULL Request时以及迭代评审时进行，pull request时由负责其他模块的成员进行，迭代评审由全体成员负责，代码评审主要对代码的质量以及编程规范等方面进行评审并反馈。

单元测试：单元测试由小组所有成员共同进行，在每次迭代前对此次迭代中自己所负责模块的代码进行测试。

集成测试：集成测试在所有的单元测试通过后根据当前进度安排由所有小组成员来共同负责进行，验证软件的所有功能和性能及其他特性是否与用户的预期要求一致。

系统测试：系统测试在集成测试通过后通过在实际的运行环境中对所有功能依次进行测试，验证软件各项需求是否实现，以及软件的性能水平

## 项目沟通计划

每日例会 9：15-9：30， 定期，每日

不定期向助教及老师汇报项目进度

# 附录