**资源计划**

人员分配：（5人）

项目经理 1

安卓前端 1

微信小程序 1

后台 2

硬件分配：

服务器N台

软件资源：

开发环境：

安卓前端：Android studio

微信小程序：微信web开发工具

后台：Ideal

**风险分析与控制**

本计划主要针对项目开发设计到的风险，包括在项目开发周期中可能出现以及项目实施过程中外部环境的变化可能引起的风险进行评估。在文中对所提到的风险一一做了分析，并提出相应的回避措施。由于风险是在项目开始之后才对项目发起的负面影响，所以风险分析不足，或者回避措施不得力，都可能造成项目开发的失败。风险分析是在事前进行的一种估计，凭借一定技术手段和丰富的经验，基本能对项目的风险做出比较准确的估计，经过慎重的考虑提出可行的风险回避措施，是避免损失的重要环节。

1. 风险条目表

2.1产品规模风险

产品初定活跃用户6000人；

软件接口：校园卡支付系统

2.2需求风险

对长期活跃用户缺少确定的把握；

与餐厅部门沟通不协调；

分析员对业务了解不够全面；

用户的需求不断变化，由于不确定的需求导致新的市场。

2.3相关性风险

开发团队财务资源有限；

项目经理管理经验不足；

不可抗力造成的危害；

项目时间把控不准确，无法在规定时间内提交。

2.4技术风险

开发团队经验不足；

数据加密技术不够完善；

用户需求中的特殊功不能及时交付；

数据库过小不能满足需求；

设计缺陷导致编码不能正常进行；

缺少测试计划；

缺少质量跟踪。

2.5开发环境风险

所使用的开发工具质量问题；

设计工具不合适；

数据库各子模块对接困难；

设备固定折损；

备份环境不稳定。

风险定性分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 潜在风险事件 | 风险发生的概率等级 | 后果影响等级 | 综合风险指数 |
| 产品规模风险 | 产品初定活跃用户6000人 | 高 | 严重 | 5 |
|  | 软件接口：校园卡支付系统 | 高 | 严重 | 3 |
| 需求风险 | 对长期活跃用户缺少确定的把握 | 高 | 轻度 | 9 |
|  | 与餐厅部门沟通不协调 | 高 | 轻度 | 18 |
|  | 分析员对业务了解不够全面 | 中 | 轻度 | 11 |
|  | 用户的需求不断变化，由于不确定的需求导致新的市场 | 中 | 轻度 | 2 |
| 相关性风险 | 开发团队财务资源有限 | 中 | 轻度 | 11 |
|  | 项目经理管理经验不足 | 极高 | 严重 | 3 |
|  | 不可抗力造成的危害 | 低 | 灾难性 | 1 |
|  | 项目时间把控不准确，无法在规定时间内提交 | 极高 | 灾难性 | 2 |
| 技术风险 | 开发团队经验不足 | 高 | 中度 | 11 |
|  | 数据加密技术不够完善 | 极高 | 灾难性 | 1 |
|  | 用户需求中的特殊功不能及时交付 | 中 | 轻度 | 11 |
|  | 数据库过小不能满足需求 | 低 | 轻度 | 11 |
|  | 设计缺陷导致编码不能正常进行 | 中 | 严重 | 6 |
|  | 缺少测试计划 | 低 | 轻度 | 14 |
|  | 缺少质量跟踪 | 低 | 轻度 | 9 |
| 开发环境风险 | 所使用的开发工具质量问题 | 中 | 严重 | 6 |
|  | 设计工具不合适 | 低 | 轻度 | 19 |
|  | 数据库各子模块对接困难 | 中 | 严重 | 6 |
|  | 设备固定折损 | 低 | 轻度 | 14 |
|  | 备份环境不稳定 | 中 | 严重 | 6 |

**风险应对措施**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 风险意识 |  | 应对措施 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目管理过程 | 潜在风险事件 | 风险发生后果 | 应急措施 | 预防措施 |
| 产品规模风险 | 产品初定活跃用户6000人 | 系统不稳定 | 追加资源 | 使用大服务器 |
|  | 软件接口：校园卡支付系统 | 数据库不能共享 |  | 优化几口 |
| 需求风险 | 对长期活跃用户缺少确定的把握 | 系统崩溃 | 修改系统 | 引入并发 |
|  | 与餐厅部门沟通不协调 | 不能满足餐厅需求 | 立即沟通 | 制定完整的访问计划 |
|  | 分析员对业务了解不够全面 | 系统不能满足用户需求 | 根据业务修改 | 让用户确定需求报告 |
|  | 用户的需求不断变化，由于不确定的需求导致新的市场 | 项目完成困难 | 项目讨论，决定保留内容 | 建立范围变更 |
| 相关性风险 | 开发团队财务资源有限 | 项目不能按时完成 | 追加成本 | 提前做好资源分配计划 |
|  | 项目经理管理经验不足 | 项目延期 | 加班 | 制定详细的迭代计划 |
|  | 不可抗力造成的危害 |  |  |  |
|  | 项目时间把控不准确，无法在规定时间内提交 | 不能满足用户需求 | 追加模块 | 与用户沟通协调 |
| 技术风险 | 开发团队经验不足 | 系统功能不能完全保证 | 培训 | 培训 |
|  | 数据加密技术不够完善 | 数据泄露 | 备份数据 | 加强安全管理 |
|  | 用户需求中的特殊功不能及时交付 | 用户需求不嫩满足 | 追加模块 | 沟通机制 |
|  | 数据库过小不能满足需求 | 数据溢出 | 将现有数据备份 | 分布式 |
|  | 设计缺陷导致编码不能正常进行 | 质量问题 | 修改设计 | 设计提交之前评审 |
|  | 缺少测试计划 | 项目延期 | 追加测试计划 | 事前评审测试计划 |
|  | 缺少质量跟踪 | 项目质量问题 | 追加质量跟踪 | 事前评审质量跟踪计划 |
| 开发环境风险 | 所使用的开发工具质量问题 | 项目延期 | 更换开发工具 | 选择正版软件 |
|  | 设计工具不合适 | 项目延期 | 更换设计工具 | 选择适用的工具 |
|  | 数据库各子模块对接困难 | 数据不能统一 |  |  |