USKY机票预订系统

开发计划

|  |  |  |
| --- | --- | --- |
| 项目组成员信息 | | |
| 小组名称 | USKY机票预订系统开发小组 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 14061003 | 刘重伟 | 文档大部分内容的编写及检查（70%） |
| 14061009 | 张子轩 | 风险管理部分（12%） |
| 14061015 | 张文镔 | 成本估算部分（8%） |
| 14061018 | 李佳锴 | 引言部分（10%） |
|  |  |  |
|  |  |  |
|  |  |  |

2016-09

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2016/10/16 | 刘重伟 | 李佳锴 | 本版为最终提交版本 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1 引言 6](#_Toc7898)

[1.1软件标识 6](#_Toc754)

[1.2项目概述 6](#_Toc12483)

[1.3文档概述 6](#_Toc24309)

[1.4 与其他计划之间的联系 6](#_Toc7069)

[1.5 基线 7](#_Toc1369)

[2 引用文件 7](#_Toc18617)

[3 产品交付 7](#_Toc7630)

[3.1程序 7](#_Toc7712)

[3.2文档 7](#_Toc18156)

[3.3服务 8](#_Toc5617)

[3.4非移交产品 8](#_Toc14484)

[3.5验收标准 8](#_Toc7623)

[3.6最终交付期限 8](#_Toc21065)

[4 需求与约束 8](#_Toc26143)

[4.1 软件需求与约束 8](#_Toc15735)

[4.2开发系统需求与约束 9](#_Toc18966)

[4.3项目文档编制的需求与约束 9](#_Toc19130)

[4.4项目进度及资源需求 9](#_Toc31224)

[4.5 监督和控制机制 9](#_Toc10922)

[4.6 其他需求 9](#_Toc10110)

[5 实施整个软件开发的计划 10](#_Toc26039)

[5.1软件开发流程 10](#_Toc10402)

[5.2软件开发总体计划 11](#_Toc21517)

[5.2.1软件开发方法 11](#_Toc28278)

[5.2.2软件产品标准 11](#_Toc7596)

[5.2.3处理关键性需求 11](#_Toc22377)

[5.3软件开发计划表 12](#_Toc19466)

[5.3.1任务分配表 12](#_Toc24782)

[5.3.2开发进度表 12](#_Toc508)

[6 实施详细软件开发活动的计划 14](#_Toc6579)

[6.1 系统需求分析 14](#_Toc21436)

[6.1.1 用户输入分析 14](#_Toc2368)

[6.1.2 运行概念 14](#_Toc18501)

[6.1.3 系统需求 14](#_Toc3382)

[6.2 系统设计 14](#_Toc5719)

[6.2.1 系统级设计决策 14](#_Toc6473)

[6.2.2 系统体系结构设计 15](#_Toc742)

[6.3 软件需求分析 15](#_Toc18024)

[7 进度表和活动网图 16](#_Toc13252)

[7.1 进度表 16](#_Toc22839)

[7.2 活动网图 17](#_Toc8382)

[8 项目组织和资源 17](#_Toc24887)

[8.1 项目资源 17](#_Toc24203)

[8.1.1人力资源情况 17](#_Toc24194)

[8.1.2其他资源情况 18](#_Toc6711)

[9培训 18](#_Toc2555)

[9.1 项目技术要求 18](#_Toc4342)

[9.2培训计划 18](#_Toc31839)

[10 估算 18](#_Toc28547)

[10.1人力资源估算 18](#_Toc4777)

[10.2成本估算 19](#_Toc7540)

[10.2.1 规模估算 19](#_Toc8851)

[10.2.2 成本估算 19](#_Toc23590)

[11 风险管理 20](#_Toc26697)

# 1 引言

## 1.1软件标识

文档标识号：BUAA2016-09-26

文档标题：软件设计说明

项目/产品中文全称：USKY机票预订系统

项目/产品英文全称：USKY air ticket booking system

项目/产品英文简称：USKY ATBS ver.1.0

项目/产品编码：20160926

## 1.2项目概述

该网站是机票预订网站的简化版，面向对象为广大群众，该网站能够满足客户查询航班、预订和退订机票，以及航空公司出售机票等基本功能。

该网站之前没有开发、运行和维护的历史。

该网站目前开发方为北京航空航天大学计算机学院软件工程课程Team-24小组，计划用户为广大群众无投资方，无需求方，无支持机构。

## 1.3文档概述

该文档为“USKY”机票预订网站的规划书，主要说明了该网站的开发计划、开发网站所需要的软件硬件支持、资金预算以及对开发人员编程能力的要求，由于在该网站的开发过程中会需要用户体验，从而涉及到部分个人隐私，同时，防止源代码外泄，我们将提出以下几点保密性要求：

* 开发人员定期维护数据库并且对网站及数据库安全进行测试。
* 开发人员不得将用户个人信息泄露给非开发人员。
* 体验人员不得将程序代码泄露给非开发方人员。

## 1.4 与其他计划之间的联系

本项目为独立的开发项目，与其他开发项目之间不存在关系。

## 1.5 基线

软件开发过程中，每个阶段可能由不同类型的角色和人员来完成，因此有必要清晰地标识上一阶段完成的成果和下一阶段开始工作的基础，这种标识活动就是建立基线。基线包括以下四部分：

* 需求基线：包括需求分析规格和功能清单。其中功能清单有用户注册、用户登录、用户注销、用户个人资料、用户评价、用户付账、航班查询、活动通知、订单查询、订单完成等。
* 设计基线：总体设计规格、详细设计规格、数据库设计规格。
* 测试基线：手动测试、单元测试、整体测试、自动化测试、撰写测试报告、修改问题、再测试等。
* 发布基线：软件产品信息描述、用户手册、软件产品质量报告等。

2 引用文件

[1]计算机文档编制规范

# 3 产品交付

## 3.1程序

需要提交的程序为网页前端代码和数据库代码，其中网页前端为顾客提供各种方便的功能，包括机票预订系统以及最新的机票打折讯息等。后端使用MySQL数据库，其中包括航班信息、机场信息、客户信息、航空公司信息等。为保护客户的信息安全以及保证网站的正常使用，本网站会有专门的维护人员定期维护数据库，并且，对数据库安全性进行评估。

## 3.2文档

* 需求规格文档：该文档仅供软件开发者使用，开发者需根据此文档了解用户需求，并作相应功能的扩展，用户无需掌握此文档。
* 设计规格文档：该文档仅供软件开发者使用，开发者需根据此文档了解软件的架构，其中包括总体设计规格.doc、详细设计规格.doc、数据库设计.pdm。
* 测试规格文档：该文档是开发人员完成开发后，将软件系统交给测试人员测试时对之前所有开发成果的标识。
* 软件开发计划：该文档用于软件开发过程中各种计划的安排以及软件开发日程的管理。

## 3.3服务

* 航班查询服务：为顾客提供某时某刻某地到某地航班的具体信息，方便顾客了解航班动态。
* 机票预订服务：顾客不仅能够为自己订票还能够为关系组群中的人订票，不仅能够预定单程票还能预定返程票。
* 降价通知服务：顾客会在网站首页醒目的位置看到航班的降价通知，方便顾客选择更经济实惠的机票。

## 3.4非移交产品

* Web服务器端程序：产品发布后，所有产品的安装根据用户需求从已经发布的版本中选择或者进行增量开发，不能直接将源码编译后交付用户
* 数据库配置：产品发布后，用户没有权限更改数据库中的数据，更没有权限了解或更改数据库结构。

## 3.5验收标准

* 最后在交付客户之前进行小组内评审，代码编写符合软件项目开发类标准，与文档说明保持一致，文档书写规范，格式统一，没有语病歧义。
* 最终产品和软件开发规划中的描述大体一致，能实现软件开发规划中的大部分功能，用户体验良好、操作流畅。
* 代码书写风格统一，简洁明了，易于维护，采用标准规范。
* 没有因软件设计缺陷造成数据泄漏甚至丢失，没有因为操作不当造成的系统崩溃，保证程序具有较高的安全性。

## 3.6最终交付期限

2016年12月5日。

# 4 需求与约束

## 4.1 软件需求与约束

* 前端：Eclipse Jee / Macromedia Dreamweaver 8等能够编写JSP的软件
* 后端：MySQL数据库

## 4.2开发系统需求与约束

* 操作系统：Windows 7/Windows 8/Windows 8.1/Windows 10或类似操作系统。
* 处理器：Intel Core i5/i7或性能相当的处理器。
* IE内核：IE 8以上。

## 4.3项目文档编制的需求与约束

为使软件文档能起到多种桥梁的作用，使它有助于程序员编制程序，有助于管理人员监督和管理软件的开发，文档的编制必须保证一定的质量。

* 针对性：分清读者对象。按不同的类型、不同层次的读者，决定怎样适应他们的需要。
* 精确性：文档的行文应当十分确切，不能出现多义性的描述。清晰性：文档编写应力求简明，如有可能，配以适当的图表，以增强其清晰性。
* 完整性：任何一个文档都应当是完整的、独立的，它应自成体系。
* 灵活性：各个不同软件项目，其规模和复杂程度有着许多实际差别，能一律看待。

## 4.4项目进度及资源需求

采用自顶向下的设计，按照需求分析和功能分析文档，将项目分成若干小项目，每个小项目严格按照线性顺序模型(Requirement-Design-Implementation)执行。

## 4.5 监督和控制机制

每周组员需要向组长汇报当前项目的进度，组内对当前项目验收合格后才能进行下一步计划，这些项目包括文档和程序。文档应该在检查符合格式要求后才准许通过程序应该在完成相应功能并进行测试后才准许通过。在项目进展过程中使用Microsoft Office Project进行项目进度的管理。

## 4.6 其他需求

* 知识需求：需要各成员掌握JSP/Servlet、SQL、JavaScript、jQuery等技术。
* 可靠性需求：由于小组成员水平有限所以不可避免会有设计错误，缺乏对特殊情况和错误处理的考虑等。众所周知程序代码越长，结构越复杂，其可靠性越难保证。我们会在软件开发的各个阶段进行测试，尽量减少错误，同时进行代码优化，提高系统可靠性。
* 扩展性需求：该订票系统在开发过程中，应该充分考虑以后的可扩展性。因为对于用户查询的需求也会不断的更新和完善，一些新的功能也可以逐渐嵌入。所以，要求系统提供足够的手段进行功能的调整和扩充。

# 5 实施整个软件开发的计划

## 5.1软件开发流程

该软件开发主要包括以下流程，该流程主要参考瀑布模型：

* 需求分析：该流程主要结合调查报告，分析用户对该机票预订网站有什么需求，在该流程中采取组内讨论的方式，按照自顶向下的方法将用户需求进行整理归纳，形成确切的需求分析，最终由文档撰写人员撰写需求分析报告。
* 系统设计：在该流程中主要结合之前撰写的需求分析文档设计软件的总体架构，该部分由前端架构人员和后端架构人员负责，按照自底向上的方法进行设计，最终由文档撰写人员撰写系统设计报告。
* 程序设计：在该过程中采用面向对象的程序设计方法，按照之前的系统设计将软件总体分解为若干功能模块，描述实现具体模块所涉及到的主要算法、数据结构、类的层次结构及调用关系，之后对人员进行分工，编写自己负责的功能模块。
* 编写代码：开发者根据之前程序的详细设计中计划的对数据结构、算法分析和模块实现等方面的设计要求，开始具体的编写程序工作，分别实现各模块的功能。
* 单元测试：每个功能模块程序编写完后由测试人员进行测试，确保模块基本功能实现。
* 系统测试：所有模块编写并测试完毕后进行系统整体的功能测试，测试完成后有文档编写人员撰写软件测试报告。
* 用户测试：选择部分同学使用该软件，记录使用过程中遇到的问题以及用户反馈的意见以便日后进行升级和维护。
* 运行维护：不定期对软件进行更新维护并收集用户评价。

## 5.2软件开发总体计划

### 5.2.1软件开发方法

* 面向对象的开发方法：这是一种自底向上和自顶向下相结合的方法，而且它以对象建模为基础，从而不仅考虑了输入、输出数据结构，实际上也包含了所有对象的数据结构。
* 信息隐蔽方法(Parnas方法)：在概要设计时列出将来可能发生变化的因素，并在模块划分时将这些因素放到个别模块的内部。这样，在将来由于这些因素变化而需修改软件时，只需修改这些个别的模块，其它模块不受影响。信息隐蔽技术不仅提高了软件的可维护性，而且也避免了错误的蔓延，改善了软件的可靠性。
* 软件重用和组件连接：软件重用是利用已有的软件成份来构造新的软件。它可以大大减少软件开发所需的费用和时间，且有利于提高软件的可维护性和可靠性。

### 5.2.2软件产品标准

* 代码编写标准：主要编程语言为JSP，代码风格要简洁明了、易于调试，代码要有缩进。变量命名方式按照匈牙利命名法。
* 程序测试标准：进行测试时，测试样例不仅要保证代码覆盖率最大化，而且，程序也要有一定的抗压力性。
* 软件验收标准：该网站能够实现需求说明文档上的大部分需求，并且，界面简洁明了，操作流畅，没有因该软件程序造成的Crash。

### 5.2.3处理关键性需求

* 安全性保证：本网站不会对用户计算机上的软件和硬件造成任何损害，该网站不会泄露用户的个人信息也不会窃取用户的个人信息。为了软件开发安全性保证，我们会以构建软件的环境模型，建立自环境、经系统到软件的层次化分析结构为基本出发点。
* 保密性保证：工作在不同环境的软件对其安全、保密的要求显然是不同的。应当把这方面的需求恰当地做出规定，以便对所开发的软件给予特殊的设计，使其在运行中其安全保密方面的性能得到必要的保证。我们所做的图书管理系统仅仅用于学习交流，对于保密性的需求程度没有很严格的要求。
* 私密性保证：我们的机票预订网站对各种用户的权限做出了限制。普通客户只能查询和订票，相关工作人员如果取得授权就能够修改客户的订票数据，为客户退票。航空公司能够发布航班的相关信息。这些权限的限制为该软件提供了私密性保障。

## 5.3软件开发计划表

### 5.3.1任务分配表

|  |  |
| --- | --- |
| 任务 | 人员 |
| 需求分析 | 全体人员 |
| 系统设计 | 刘重伟 |
| 程序设计 | 刘重伟 张文镔 |
| 程序测试 | 刘重伟 张文镔 |
| 文档编写 | 李佳锴 张子轩 |
| 总结 | 全体人员 |

表格 1

### 5.3.2开发进度表

|  |  |  |
| --- | --- | --- |
| 任务名称 | 开始日期 | 结束日期 |
| 制定软件开发计划 | 2016年10月1日 8:00 | 2016年10月10日 17:00 |
| 进行需求调研 | 2016年10月1日 8:00 | 2016年10月5日 17:00 |
| 撰写软件开发计划文档 | 2016年10月6日 8:00 | 2016年10月10日 17:00 |
| 撰写需求分析文档 | 2016年10月6日 8:00 | 2016年10月12日 17:00 |
| 学习相关知识 | 2016年10月17日 8:00 | 2016年10月23日 17:00 |
| 学习JSP编程 | 2016年10月17日 8:00 | 2016年10月18日 17:00 |
| 学习HTML编程 | 2016年10月19日 8:00 | 2016年10月20日 17:00 |
| 学习MySQL数据库 | 2016年10月21日 8:00 | 2016年10月23日 17:00 |
| 撰写软件设计文档 | 2016年10月24日 8:00 | 2016年10月30日 17:00 |
| 完成软件设计文档数据库部分 | 2016年10月24日 8:00 | 2016年10月26日 17:00 |
| 完成软件设计文档前端部分 | 2016年10月27日 8:00 | 2016年10月30日 17:00 |
| 完成数据库初步架构 | 2016年10月31日 8:00 | 2016年11月6日 17:00 |
| 完成客户相关数据结构 | 2016年10月31日 8:00 | 2016年11月2日 17:00 |
| 完成航空公司相关数据结构 | 2016年11月3日 8:00 | 2016年11月4日 17:00 |
| 完成机场以及业务员相关数据结构 | 2016年11月5日 8:00 | 2016年11月6日 17:00 |
| 数据库架构测试 | 2016年11月7日 8:00 | 2016年11月13日 17:00 |
| 客户相关数据结构测试 | 2016年11月7日 8:00 | 2016年11月9日 17:00 |
| 航空公司相关数据结构测试 | 2016年11月10日 8:00 | 2016年11月11日 17:00 |
| 机场以及业务员相关数据结构测试 | 2016年11月12日 8:00 | 2016年11月13日 17:00 |
| 数据库架构初步完成 | 2016年11月13日 8:00 | 2016年11月13日 8:00 |
| 完成网页前端基本框架 | 2016年11月14日 8:00 | 2016年11月20日 17:00 |
| 完成网页前端客户部分 | 2016年11月14日 8:00 | 2016年11月16日 17:00 |
| 完成网页前端航空公司部分 | 2016年11月17日 8:00 | 2016年11月18日 17:00 |
| 完成网页前端机场及业务员部分 | 2016年11月19日 8:00 | 2016年11月20日 17:00 |
| 完成网页前端基本功能测试 | 2016年11月21日 8:00 | 2016年11月27日 17:00 |
| 网页前端客户部分测试 | 2016年11月21日 8:00 | 2016年11月23日 17:00 |
| 网页前端航空公司部分测试 | 2016年11月24日 8:00 | 2016年11月25日 17:00 |
| 网页前端业务员部分测试 | 2016年11月26日 8:00 | 2016年11月27日 17:00 |
| 完成前端部分 | 2016年11月27日 8:00 | 2016年11月27日 8:00 |
| 撰写测试文档 | 2016年11月28日 8:00 | 2016年12月4日 17:00 |
| 完成测试文档数据库部分 | 2016年11月28日 8:00 | 2016年11月30日 17:00 |
| 完成测试文档前端部分 | 2016年12月1日 8:00 | 2016年12月4日 17:00 |

表格 2

# 6 实施详细软件开发活动的计划

## 6.1 系统需求分析

### 6.1.1 用户输入分析

* 注册时需要输入：用户姓名、性别、ID、邮箱、密码。
* 登陆时需要输入：用户姓名、密码、ID、邮箱。
* 检索时需要输入：出发城市、到达城市、出发日期、返程日期、出行人数、乘客类型、航空公司、舱位。

### 6.1.2 运行概念

该网站主要为人们提供方便快捷的订票途径，为旅客们规划出行路线并提供经济舒适的出行服务，让顾客体验到最优质的乘机体验。

### 6.1.3 系统需求

* 操作系统：Windows 7/Windows 8/Windows 8.1/Windows 10或类似操作系统。
* 处理器：Intel Core i5/i7或性能相当的处理器。
* IE内核：IE 8以上。

## 6.2 系统设计

### 6.2.1 系统级设计决策

采用面向对象的设计方法，具体为自底向上归纳和自顶向下分解。

* 自底向上的归纳：从问题的陈述入手，构造系统模型。从真实系统导出类的体系，即对象模型包括类的属性，与子类、父类的继承关系，以及类之间的关联。类是具有相似属性和行为的一组具体实例（客观对象）的抽象，父类是若干子类的归纳。因此这是一种自底向上的归纳过程。在自底向上的归纳过程中，为使子类能更合理地继承父类的属性和行为，可能需要自顶向下的修改，从而使整个类体系更加合理。在对象模型建立后，很容易在这一基础上再导出动态模型和功能模型。这三个模型一起构成要求解的系统模型。
* 自顶向下的分解：系统模型建立后的工作就是分解。在软件开发过程中按服务来分解。服务是具有共同目标的相关功能的集合，如I/O处理、图形处理等。这一步的分解通常很明确，而这些子系统的进一步分解因有较具体的系统模型为依据，也相对容易。

### 6.2.2 系统体系结构设计

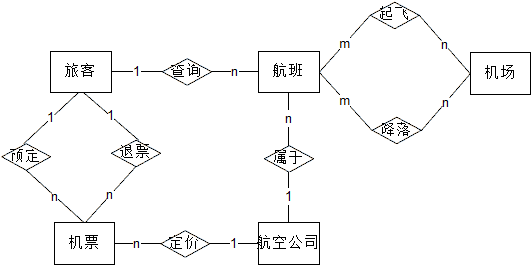


图 1

## 6.3 软件需求分析

* 顾客：顾客能够查询到特定日期两地之间所有机票，顾客能够预定机票，顾客能够为关系组群中的人预定机票。
* 业务员：业务员能够为顾客办理退票业务，并且，具有和顾客一样的查询权限。
* 航空公司：航空公司能够决定各航线飞机的数量以及机票的价格。
* 机场：机场能够查询当天到达该机场和从该机场起飞的航班的具体信息，为顾客提供指引。

# 7 进度表和活动网图

## 7.1 进度表

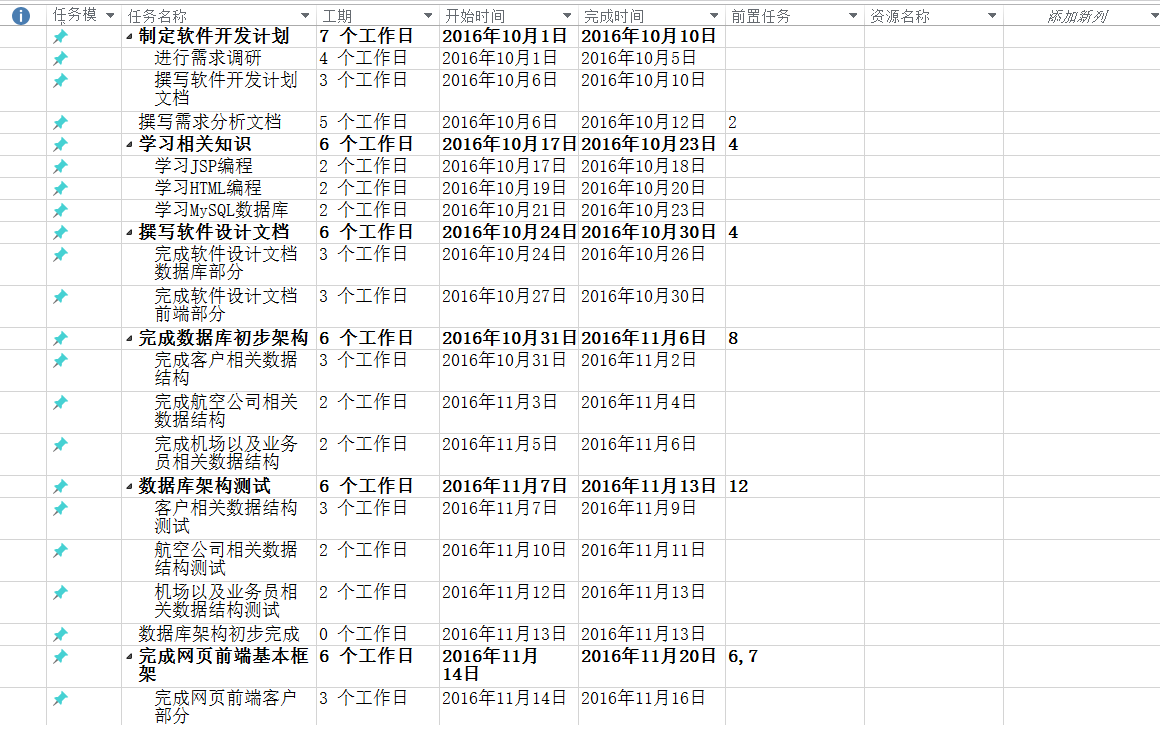


图 2

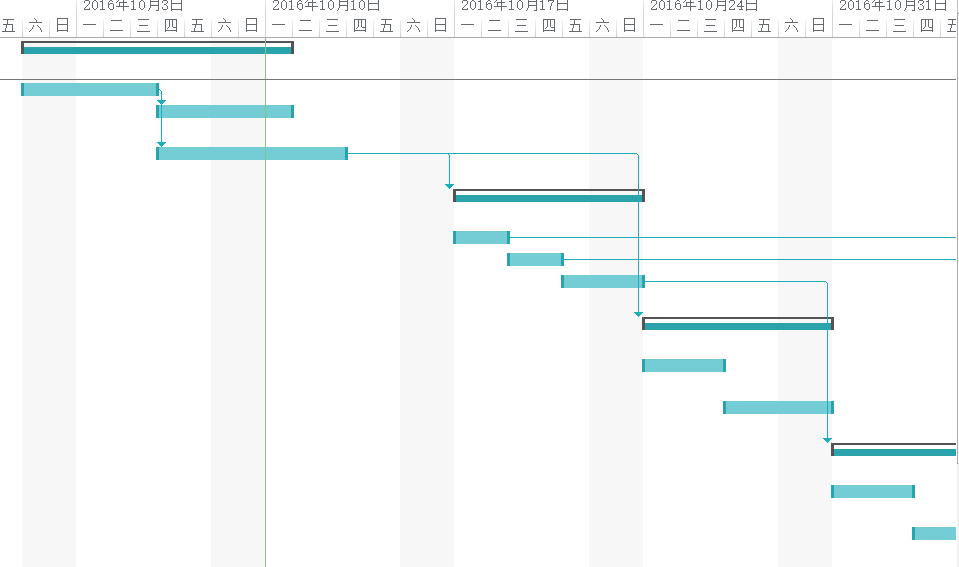


图 3

## 7.2 活动网图

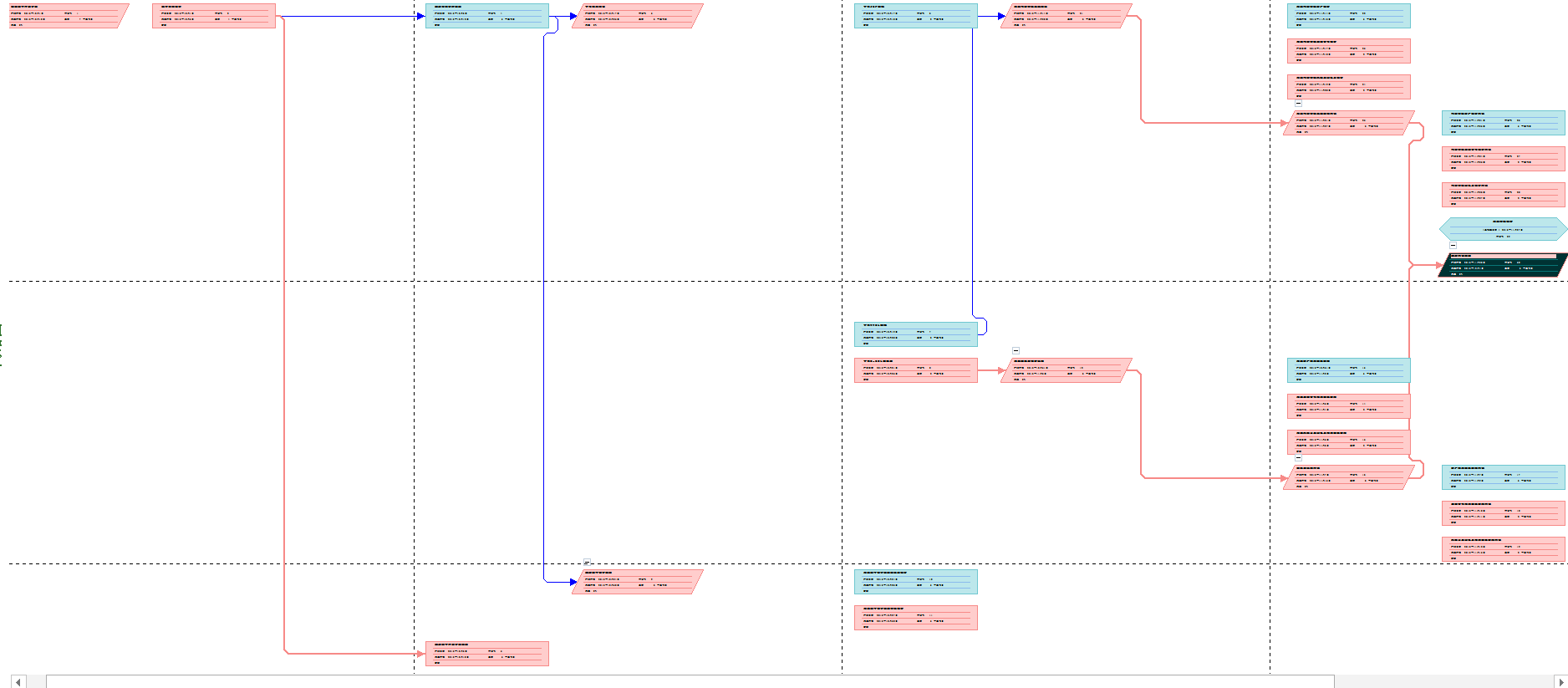


图 4

以上图表详见软件开发日程.mpp

# 8 项目组织和资源

## 8.1 项目资源

### 8.1.1人力资源情况

|  |  |  |
| --- | --- | --- |
| 成员 | 掌握技术 | 开发经历 |
| 刘重伟 | Java、C、HTML | 安卓开发，Web网页开发 |
| 张文镔 | Java、C、JSP、SQL | Java程序开发 |
| 张子轩 | Java、C、C++ | C++程序开发 |
| 李佳锴 | Java、C、SQL | Java程序开发 |

表格 3

### 8.1.2其他资源情况

|  |  |  |  |
| --- | --- | --- | --- |
|  | 硬件环境 | 软件环境 | 技术需求 |
| 已有 | Intel Core i5 以上计算机 | Windows 8以上操作系统IE 8以上浏览器 | 数据库技术 JSP开发技术 |
| 需求 | 无 | 无 | 无 |

表格 4

# 9培训

## 9.1 项目技术要求

JSP/Servlet开发技术、MySQL数据库使用技术、JavaScript、jQuery。

## 9.2培训计划

|  |  |
| --- | --- |
| JSP/Servlet的学习 | 2016/10/10-2016/10/14 |
| MySQL数据库使用技术学习 | 2016/10/15-2016/10/20 |
| JavaScript、jQuery学习 | 2016/10/21-2016/10/25 |

表格 5

# 10 估算

## 10.1人力资源估算

|  |  |
| --- | --- |
| 设计人员 | 1人 |
| 文档人员 | 2人 |
| 开发人员 | 4人 |
| 测试人员 | 2人 |

表格 6

## 10.2成本估算

### 10.2.1 规模估算

规模估算采用功能点估算方法。其计算公式为：FP = UFC \* TCF（其中UFC是未调整功能点数，TCF为技术复杂度因子）。

UFC计算：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能项 | 权重 | | |
| 简单 | 一般 | 复杂 |
| 外部输入EI | 3 | 4 | 6 |
| 外部输出EO | 4 | 5 | 7 |
| 查询EQ | 3 | 4 | 6 |
| 外部接口文件EIF | 7 | 10 | 15 |
| 内部逻辑文件EOF | 5 | 7 | 10 |

表格 7

根据估算，本项目UFC = 5\*4+2\*5+4\*5+2\*7+2\*5=74，TCF=0.65+0.01(6\*3)=0.83。

所以FP = 74\*0.83 = 61.42

结合网上查到的java语言的LOC与FP的换算关系，得LOC = FP\*53 = 61.42\*53 = 3225.26

### 10.2.2 成本估算

|  |  |
| --- | --- |
| 用途 | 预算（元） |
| 市场调查 | 10 |
| 设备维修 | 100 |
| 人员培训 | 20 |
| 代码实现与维护 | 100 |
| 上线宣传 | 150 |
| 其他 | 100 |
| 合计 | 390 |

表格 8

# 11 风险管理

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 描述 | 发生概率 | 危害程度 | 规避措施 | 缓减策略 | 状态 |
| 1 | 过程风险（进程缓慢，后期重复工作，未预期到的风险） | 中 | 中 | 提前做好项目规划，指定软件编写标准，保证软件编写的质量与进度 | 项目规划中留有余地，分阶段交付测试产品，增加监管，的频度与力度 | 未发生 |
| 2 | 预算风险 | 中 | 中 | 事先留有余地，项目过程中控制成本 | 适当增加投入，或者接受风险 | 未发生 |
| 3 | 需求风险（需求变化，定义扩展，定义含混） | 高 | 高 | 过与用户方的技术人员和业务人员进行良好的沟通，了解业务流程、功能需求、系统构想和项目目标 | 预先规定好需求变更流程，根据具体情况决定需求的变更 | 未发生 |
| 4 | 进度风险 | 中 | 中 | 项目计划进度表，充分考虑各种潜在因素，适当留有余地 | 增加项目监控的频度和力度、多运用可行的办法保证工作质量避免返工 | 未发生 |
| 5 | 沟通不良风险 | 低 | 中 | 事先约定好沟通的渠道和方式，做好沟通，建立有效的反馈机制 | 与当事用多种方式，主动对问题进行沟通 | 已发生 |
| 6 | 人员风险（决策缓慢，技术不熟练，不熟悉环境工具） | 中 | 中 | 选用成熟稳定的技术，以免新技术本身存在问题或掌握不够深入，造成开发出的产品性能以及质量低劣 | 多上网查阅资料，多问指导老师在学习中成长，在失败中积累经验 | 已发生 |
| 7 | 计算机故障 | 低 | 低 | 平时做好计算机的保养 | 修理的同时借用其他同学或机房计算机保证进度 | 未发生 |
| 8 | 质量风险 | 中 | 高 | 预先制定标准，采用合理规范的开发流程，分阶段测试产品 | 经常交流工作成果、采用符合要求的开发流程、认真组织对产出物的检查和评审、计划和组织严格的独立测试等 | 已发生 |
| 9 | 工具风险 | 中 | 中 | 项目的启动阶段就落实好各项工具的来源或可能的替代工具 | 选择类似的替代工具 | 未发生 |
| 10 | 性能风险 | 中 | 中 | 项目开发之前先设计和搭建出系统的基础架构并进行性能测试，确保架构符合性能指标后再进行后续工作 | 对程序进行优化 | 未发生 |
| 11 | 人力资源风险 | 中 | 中 | 预先进行评估，确保每个人在合适的个岗位上。对成员进行一定的培训 | 维持开发队伍的稳定，尽可能将项目的核心工作分派给多人 | 未发生 |
| 12 | 产品风险（测试时间增加，新增功能，兼容性问题） | 中 | 中 |  |  | 未发生 |

表格 9

* 发生概率：

高：80%以上

较高：60%-80%

中：40%-60%

较低：20%-40%

低：20%以下

* 危害程度：

没有定量的指标，依靠PM个人的判断，分为“高”、“中”、“低”三级。

* 规避措施：风险未发生时，采取什么手段防止风险发生。
* 缓减策略：风险已发生时，采取什么手段减少其危害。
* 状态：该风险项当前的状态，分为“未发生”、“已发生”、“已解决”。