

**软件工程报告**

|  |  |
| --- | --- |
| 综合设计题目： | 航班信息管理系统 |
| 姓 名： | 潘翔 |
| 学 院： | 计算机科学与技术学院 |
| 专 业： | 物联网工程 |
| 班 级： | IOT1601 |
| 学 号： | U20161489 |
| 指导教师： |  |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2018 年 9月 22日

**目 录**

[1 选题 1](#_Toc683028622)

[1.1 教材 1](#_Toc1693923508)

[1.2 作业选题背景知识 1](#_Toc995094925)

[1.3 辅助工具软件 2](#_Toc665394065)

[1.3.1 需求分析 2](#_Toc1209546617)

[1.3.2 工程管理 2](#_Toc1486764284)

[1.3.3 软件建模 2](#_Toc651550557)

[1.3.4 版本管理 2](#_Toc1498191597)

[1.3.5 自动化测试工具 2](#_Toc1846433546)

[1.4 编程语言和集成调试环境 2](#_Toc685153109)

[2 问题定义及可行性分析 3](#_Toc288480052)

[2.1 界面操作形式 3](#_Toc346709811)

[2.2 预期达到功能 3](#_Toc1506397292)

[2.3 使用者角色描述(用户画像) 3](#_Toc980444147)

[2.3.1 常规使用人员 3](#_Toc1950997768)

[2.3.2 数据统计人员 3](#_Toc32779193)

[2.3.3 软件维护人员 3](#_Toc1429631396)

[2.4 系统环境描述 4](#_Toc1011960907)

[2.4.1 硬件环境 4](#_Toc1758356825)

[2.4.2 软件环境 4](#_Toc479899104)

[2.5 系统的性能需求 4](#_Toc333837504)

[2.5.1 数据库任务/交易分布表 5](#_Toc694871880)

[2.5.2 交易混合表 6](#_Toc623284940)

[2.5.3 数据库性能指标 6](#_Toc793406046)

[2.6 可行性分析 6](#_Toc1791376749)

[3 需求分析 7](#_Toc1915287574)

[3.1 功能定义 7](#_Toc815276009)

[3.1.1 模块功能描述 7](#_Toc1479706765)

[3.1.2 模块优先级评价 7](#_Toc1118593196)

[3.2 角色定义 8](#_Toc1221018020)

[3.3 设计系统的逻辑关系 8](#_Toc1584909580)

[3.4 系统接口 9](#_Toc1801621818)

[3.4.1 图形接口 9](#_Toc767457880)

[3.4.2 软件接口 10](#_Toc432520858)

[3.4.3 通讯接口 10](#_Toc319532235)

[3.5 结构化分析 11](#_Toc1977004497)

[3.5.1 数据库逻辑结构设计 11](#_Toc1919285142)

[3.5.2 数据字典表格 12](#_Toc971082792)

[3.5.3 数据流图 16](#_Toc1327712447)

[3.5.4 E-R图 16](#_Toc1618235041)

[3.5.5 系统状态转换图 17](#_Toc1656235902)

[3.6 系统性能 17](#_Toc1616192499)

[4 概要设计与详细设计 18](#_Toc1964944852)

[4.1 系统功能模块划分 18](#_Toc1015149546)

[4.2 系统架构方案 18](#_Toc449152998)

[4.3 系统操作流程 19](#_Toc1768458973)

[4.3.1 常规使用人员操作流程 19](#_Toc1047928739)

[4.3.2 数据统计人员操作流程 19](#_Toc1878784395)

[4.3.3 软件维护人员操作流程 19](#_Toc632936232)

[4.4 前端实现方案 20](#_Toc658801917)

[4.5 后端实现方案 20](#_Toc211199851)

[4.5.1 数据库事务管理 20](#_Toc966773736)

[4.5.2 历史数据管理 20](#_Toc1353673797)

[4.6 部件详细设计 20](#_Toc834484792)

[4.6.1 航班事务管理 20](#_Toc1760179782)

[4.6.2 订单事务管理 22](#_Toc997566898)

[4.6.3 权限系统 24](#_Toc602288718)

[4.6.4 报表系统 24](#_Toc427972143)

[4.6.5 通用数据显示模块 25](#_Toc329790015)

[4.7 设计模式 26](#_Toc1720881914)

[4.8 注释及命名规范 27](#_Toc1648990163)

[5 测试报告 29](#_Toc1914699595)

[5.1 测试环境 29](#_Toc1375020084)

[5.2 软件文档测试 29](#_Toc268964395)

[5.3 源代码一致性检查 29](#_Toc199736805)

[5.4 界面(外观)测试 29](#_Toc1694552319)

[5.5 功能测试 29](#_Toc98485244)

[5.5.1 航班事务测试 29](#_Toc2119021948)

[5.5.2 订单事务测试 32](#_Toc518151464)

[5.6.3 报表测试 36](#_Toc1426197691)

[5.6 稳定性(强度)测试 36](#_Toc1589773341)

[5.7 性能测试 36](#_Toc26903718)

[5.8 强壮性(恢复)测试 37](#_Toc894906543)

[5.9 逻辑性测试 37](#_Toc1407234545)

[5.9 破坏性测试 37](#_Toc1042053264)

[5.10 安全性测试 37](#_Toc1344059541)

[6 项目管理 38](#_Toc1028209870)

[6.1 项目计划 38](#_Toc2089982004)

[6.1.1项目整体实施流程 38](#_Toc1075360288)

[6.1.2 项目进度计划 38](#_Toc1661146102)

[6.1.2 项目进度计划 38](#_Toc601300273)

[6.2 项目所需人员 43](#_Toc1286560140)

[6.3 项目功能点估算 43](#_Toc480436190)

[6.3.1 功能点定义 43](#_Toc1954974070)

[6.3.2 功能点计算 44](#_Toc2121044932)

[6.3.3 功能点调整 46](#_Toc93132324)

[6.4 项目成本估算 46](#_Toc805057320)

[6.4 项目版本迭代情况 51](#_Toc575850002)

[6.3.1 代码迭代情况 51](#_Toc521104467)

[6.3.2 文档迭代情况 53](#_Toc1134847335)

[6.5 资源使用情况 53](#_Toc149248269)

[6.6 项目交付情况 53](#_Toc22610982)

[7 课程总结 54](#_Toc902063282)

[参考文献 55](#_Toc1524268353)

[附录 56](#_Toc291575377)

# 

# 1 选题

## 1.1 教材

教材： 《软件工程(第4版)》

课外阅读：《设计模式：可复用面向对象软件的基础》

## 1.2 作业选题背景知识

1. 选题题目：航班信息管理系统
2. 所需知识：
   1. 不同航空公司的航班编号格式
   2. 航班预订流程
   3. 航班退票流程
   4. 航班选座流程
   5. 联程航空的查询及购买
   6. 航班管理系统不同的模块
   7. 航空公司对于用户信息的分析处理
3. 了解途径
   1. 自身航班工具的使用经理
   2. 不同APP和网站的体验使用
   3. 已有航班管理系统的调研
   4. 航班管理系统的相关设计文档和论文
   5. 对于能够接触到的航班工作人员的调查访问
4. 需求分析
   1. 进行用户行为约束， 对于普通用户， 某些操作一旦进行无法更改：
      1. 航班选座
      2. 订单确认
   2. 用户行为的时间约束：
      1. 旅客在飞机起飞前一天凭取票通知交款取票
      2. 用户检票时间约束
   3. 系统查询设计， 完成满足指定要求的查询如航班查询： 进行航班的查询操作
      1. 模糊查询： 利用时间， 地点， 价格等条件查询航班
      2. 精确查询： 利用航班号查询
   4. 航班预订： 进行航班预订的相关操作
      1. 座位选取
      2. 订单生成和确认
      3. 与订单管理模块， 账单管理模块交互
   5. 订单管理： 进行订单的相关操作
      1. 管理订单状态， 已经完成的订单和约束订单无法操作
      2. 更新订单状态
   6. 账单管理： 生成系统的账单

## 1.3 辅助工具软件

### 1.3.1 需求分析

思维导图：XMind 8

### 1.3.2 工程管理

MS Project

### 1.3.3 软件建模

MS Visio：UML，数据流图，算法流程图等

### 1.3.4 版本管理

如果是团队集成开发小型采用ant而大型团队使用[cruisecontrol](https://sourceforge.net/projects/cruisecontrol/)，对于单人开发的项目，通常git足够使用

1. 本地：git version 2.18.0
2. 在线：GitHub

### 1.3.5 自动化测试工具

对于QT/C++来说，常见的自动化测试工具为以下，因为不涉及B/S架构，故无须大规模访问的web压力测试，使用数据库，需要对数据库的压力进行一定程度的测试，但由于数据库底层实现的安全机制，故数据安全性和一致性有一定的保障

1. QTest
2. Google Test

## 1.4 编程语言和集成调试环境

编程语言： C++

操作系统： ArchLinux x64

IDE: Qt Creator

编译器： 8.1.1 20180531 (GCC)

数据库： 10.1.33-MariaDB MariaDB Server

因 Qt 框架的跨平台特性， 可以发行各平台的发行版本

# 2 问题定义及可行性分析

## 2.1 界面操作形式

本航班管理系统作为面向航班客服人员的管理软件，为图形化界面，采用常规的PC端UI设计，并提供清晰的UI接口，操作流程符合直觉，登录需输入用户名和密码，之后点击上方各个功能栏目进行相应的功能操作，同时规范化tab键操作，可以使用tab键进行下一个个选项填写，同时回车点击，故在无鼠标情况下可以使用。

## 2.2 预期达到功能

1. 每个航班信息的输入。
2. 每个航班的坐位信息的输入；
3. 当旅客进行机票预定时，输入旅客基本信息，系统为旅客安排航班，打印取票通知和帐单；
4. 旅客在飞机起飞前一天凭取票通知交款取票；
5. 旅客能够退订机票；
6. 能够查询每个航班的预定情况、计算航班的满座率。

## 2.3 使用者角色描述(用户画像)

### 2.3.1 常规使用人员

常规使用人员为机场的客服人员，具有基本的计算机使用技能，能够使用软件完成乘客需求。

### 2.3.2 数据统计人员

数据统计人员为专人或机场管理人员，能够使用软件进行统计报表分析，财务状况分析，机场运营分析

### 2.3.3 软件维护人员

软件维护人员应掌握基本的数据库管理技能，在软件发生故障的时候，能够利用软件的备份对关键数据进行恢复，同时根据数据库数据对于硬件进行调整。

## 2.4 系统环境描述

### 2.4.1 硬件环境

1. 服务器硬件环境

大于或等于下列参数配置

CPU: Xeon E5-2690×2

Mem: 32G [DDR3](https://www.baidu.com/s?wd=DDR3&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "/home/hover/Documents\\x/_blank) REG ECC

Disk: SSD 512G

Structure: 2U机架式

Net: 百兆网卡+百兆局域网

1. 客户机硬件环境

CPU: E7500

Mem: 2G内存

Disk: 128G硬盘

Net: 百兆网卡+百兆局域网

### 2.4.2 软件环境

1. 服务器软件环境

操作系统：支持MariaDB的操作系统

数据库： mariadb 10.1.37-MariaDB

命令行： mycli 1.18.0

网络环境：考虑多台客户端和数据库的局域网内远端同步

百兆局域网络连接支持

1. 客户机软件环境

操作系统: 支持QT的桌面端操作系统(Windows/Linux/Mac)

支持QT的基于Linux的嵌入式设备

开发语言: C/C++

编辑器: Visual Studio Code 1.27.2 x64

编译器: g++ (GCC) 8.2.1 20180831

调试器: GNU gdb (GDB) 8.2

构建工具: GNU Make 4.2.1

网络环境：考虑多台客户端和数据库的局域网内远端同步

百兆局域网络连接支持

## 2.5 系统的性能需求

对于一个航班管理系统来说，所需要处理的通常为一个地区的航班信息，且面向客服人员，对于数据库的访问请求的并发处理机制由数据库进行处理。

### 2.5.1 数据库任务/交易分布表

删除操作为更改订单状态，订单仍然保留于数据库中

对于一个机场进行数据统计：

增加：为机场订票情况

查询：为订票或者航班查询情况

更改：为检票或者改签情况

删除：为退订

生成报表：为当日数据统计

数据备份：为数据库系统备份

表2-1 数据库任务/交易分布表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 时间  交易量  事务 | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 |
| 增加 |  |  |  |  | 100 | 200 | 450 | 350 | 300 | 250 |  |  |  |
| 查询 |  |  |  |  | 300 | 390 | 450 | 240 | 360 | 300 |  |  |  |
| 更改 | 300 | 400 | 600 | 1000 | 1500 | 3000 | 3000 | 3000 | 3000 | 3000 | 2000 | 1000 | 500 |
| 删除 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 生成报表 | 20 | 25 | 23 |  |  |  |  |  |  |  |  | 30 |  |
| 数据备份 | 10 | 20 | 30 | 50 |  |  |  |  |  |  |  |  |  |

### 2.5.2 交易混合表

表2-2 交易混合表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 交易名称 | 日常业务 | 高峰期业务 | 客户机负载 | 数据库负载 | 数据机密 | 商业风险 |
| 登录 | 10/h | 20/h | **高** | 低 | **高** | 大 |
| 生成订单 | 200/h | 500/h | 中 | 中 | 中 | 中 |
| 更新订单 | 2000/h | 3500/h | 中 | 中 | 中 | 中 |
| 归档 | 20/h | 30/h | 低 | **高** | 中 | 中 |
| 数据备份 | 0/h | 2/h | 低 | **高** | **高** | 小 |

### 2.5.3 数据库性能指标

MariaDB(MySql)的性能指标为：

1. Query： 3w/s
2. Insert： 2w/s
3. Update： 8000/s
4. Delete： 8000/s

故可以满足常规的业务需求

## 2.6 可行性分析

软件采用QT进行开发，依赖mariadb作为数据库，基于QT 的控件支持和跨平台特性以及主流服务器操作系统对于mariadb的支持，能够较好的适应各种实际的应用场景。

对于业务压力，mariadb作为开源数据库，其性能能够满足正常需求，如果有更高的数据安全性需求，软件预留了清晰的接口，在不更换客户端程序的情况下，仅仅需要对于服务器进行数据迁移和客户端更新。

# 3 需求分析

根据用户的需求，给出系统分析的结果，明确给出系统实现后的功能定义，角色定义等，设计系统的逻辑关系，系统接口以及达到系统性能的措施及解决办法等。采用结构化分析方法时要有系统的详细的数据流图和核心数据的数据字典，必要时进一步画出系统的E-R图和状态转换图。采用面向对象分析方法时画出系统详细的用例模型和对象模型，必要时进一步写出系统的动态脚本或画出系统的事件跟踪图。

## 3.1 功能定义

### 3.1.1 模块功能描述

系统总体分为：航班查询，航班预订，订单管理，账单管理，通知系统

1. 航班查询：进行航班的查询操作
   1. 模糊查询：利用时间，地点，价格等条件查询航班
   2. 精确查询：利用航班号查询
2. 航班预订：进行航班预订的相关操作
   1. 座位选取
   2. 订单生成和确认
   3. 于订单管理模块，账单管理模块交互
3. 订单管理：进行订单的相关操作
   1. 管理订单状态，已经完成的订单和约束订单无法操作
   2. 更新订单状态
4. 账单管理：生成系统的账单
5. 通知系统：进行相关查询用户的状态提醒

### 3.1.2 模块优先级评价

表3-1 模块优先级评价表

|  |  |  |
| --- | --- | --- |
| 交易名称 | 优先级 | 评价 |
| **航班查询** | 高 | 高频 |
| **航班预订** | 高 |  |
| **账单管理** | 中 |  |
| **订单管理** | 高 |  |
| **通知系统** | 低 |  |
| **数据备份** | 高 | 风险 |
| **数据归档** | 低 |  |

## 3.2 角色定义

总共为三级角色权限

表3-1 角色定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能  角色 | 操纵方式 | 物理对象 | 数据对象 |
| 客服人员 | GUI | 客户机 | 所属地区数据 |
| 数据统计人员 | GUI | 客户机 | 所属地区数据/统计数据 |
| 软件维护人员 | CLI/GUI | 客户机/服务器 | 所有数据 |

## 3.3 设计系统的逻辑关系

表3-2 角色权限表

|  |  |  |  |
| --- | --- | --- | --- |
| 角色  权限 | 客服人员 | 数据统计人员  (主管) | 软件维护人员 |
| 登录 | √ | √ | √ |
| 当前地区信息 | √ | √ | √ |
| 全局地区信息 |  | √ | √ |
| 其他用户数据 |  | √ | √ |
| 账户/密码 |  |  | √ |
| 历史超时数据 |  |  | √ |

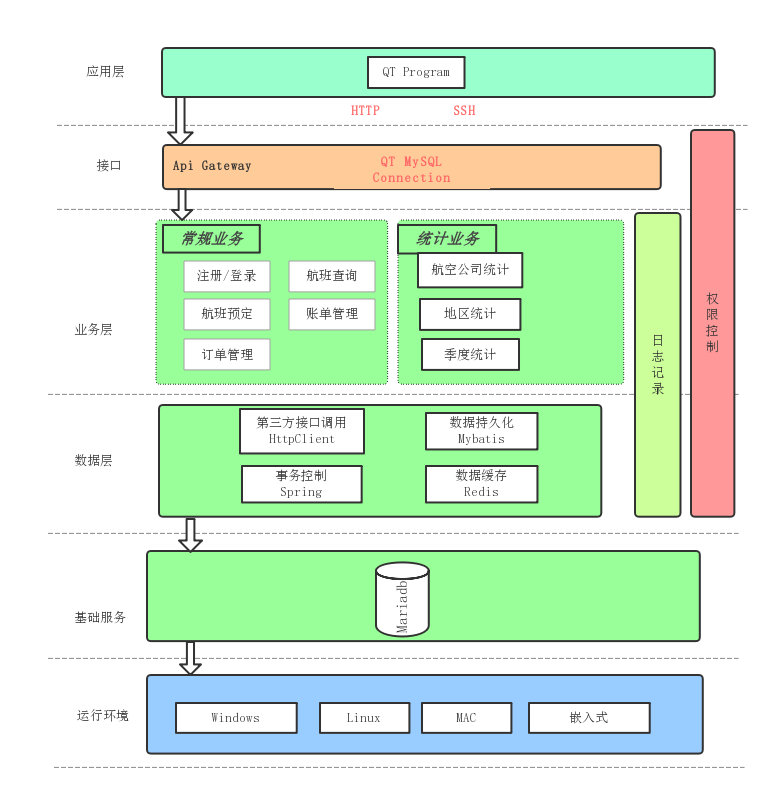


图3-1 系统整体架构图

## 3.4 系统接口

### 3.4.1 图形接口

采用QT的图形化布局工具，进行布局

1. 提供两种配色方案：
   1. 默认配色
   2. 夜间模式
2. 采用类浏览器tab的风格进行布局
   1. 标题栏
   2. 主体窗口进行选择
   3. 下方为操作按钮对选择项进行操作
3. 弹窗提示用户操作结果
   1. 弹窗标题为操作结果
   2. 窗口主体为详细提示信息
4. 最上方提供导航链接和相应的快捷键
5. 时间格式
   1. 时间格式：12h格式 hh::mm AM/PM
   2. 日起格式：mm/dd/yy
6. 字体：
   1. Cantarell
   2. 界面字体大小为10

### 3.4.2 软件接口

操作系统：支持MariaDB的操作系统

数据库： mariadb 10.1.37-MariaDB

命令行： mycli 1.18.0

操作系统: 支持QT的桌面端操作系统(Windows/Linux/Mac)

支持QT的基于Linux的嵌入式设备

开发语言: C/C++

编辑器: Visual Studio Code 1.27.2 x64

编译器: g++ (GCC) 8.2.1 20180831

调试器: GNU gdb (GDB) 8.2

构建工具: GNU Make 4.2.1

QT所需工具集：

core gui

sql

webengine

webenginewidgets

charts

### 3.4.3 通讯接口

1. 人机交互：
   1. 输入：鼠标键盘
   2. 输出：屏幕
   3. 必要时可采用加密

即系统的标准输入输出std::cin/cout

为同步通讯机制

1. 客户端服务器交互：

QSqlDatabase类进行数据库访问

为同步通讯机制

1. 客户端打印机交互：

printer\_pixmap调用打印机进行打印

为异步通讯机制

## 3.5 结构化分析

### 3.5.1 数据库逻辑结构设计

1. 用户权限表：对于用户的行为和权限进行约束管理，同时便于统计用户信息
   1. 用户名 UID
   2. 用户姓名 UNAME
   3. 密码 PASSWORD
   4. 用户权限
2. 航班信息表：用于航班信息的管理
   1. 航班号 FID
   2. 航空公司 Flight
   3. 飞机机型 FMODEL
   4. 航班状态：FSTATE
      1. 已降落 landed
      2. 延误 delayed
      3. 登机口 gate
      4. 登机 boarding
      5. 登机手续办理 [check-in](https://www.baidu.com/s?wd=check-in&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "/home/wings/Documents\\x/_blank)
   5. 起飞城市：FFROM
   6. 飞往城市：FTO
   7. 起飞日期：DATA
   8. 起飞时间：TIME
   9. 座位状态：FSTATUS
3. 用户订单/账单表：用于用户订单的管理，账单的生成
   1. 订单号 OID
   2. 航班号 FID
   3. 用户号 UID
   4. 座位号 SID
   5. 订单时间 OTIME
   6. 订单金额 OAMOUNT
   7. 订单状态 OSTATE
4. 航班座位表
   1. 航班号 FID
   2. 座位号 SID
5. 航班模型表
   1. 飞机型号 FMODEL
   2. 舱位设置 矩阵形式

### 3.5.2 数据字典表格

1. 用户权限表

用户权限表用作用户的登录验证，进入系统后的权限验证。

表3-1 用户权限表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 约束 | 数据类型 | 说明 |
| UID | Primary key | int | 用户编号 |
| USERNAME | unique | varchar | 用户名 |
| PASSWORD |  | varchar | 密码 |
| Root |  | tinyint(bool) | 是否是管理员 |

1. 航班信息表

航班信息表用作航班信息管理，和航班状态的维护，便于用户查询预订航班。

表3-2 航班信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 约束 | 数据类型 | 说明 |
| FID | Primary key | int | 航班编号 |
| Flight | unique | varchar | 用户名 |
| FMODEL | 外码：  Reference 飞机型号FSTATUSinfo(FMODEL) | varchar | 飞机机型 |
| FSTATE |  | varchar | 航班状态 |
| FFROM |  | varchar |  |
| FTO |  | varchar |  |
| FDATE |  | Date |  |
| FTIME |  | Time |  |
| ARRDATE |  | Date |  |
| ARRTIME |  | Time |  |
| FSTATUS |  | varchar | 航班座位情况  (Not Full,FULL) |

1. 飞机型号表：

用于记录飞机型号，存储飞机座位排布图和座位排布信息。

表3-3 飞机型号表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 约束 | 数据类型 | 说明 |
| FMODEL | Primary key | varchar | 飞机机型 |
| Fimage | unique | LONGBLOB | 机型图片 |
| FirstClassScale |  | varchar | 头等舱规模  （如：2排\*每排4个人  2\*4） |
| BussinessClassScale |  | varchar | 商务舱规模  同上 |
| EconomyClassScale |  | varchar | 经济舱规模  同上 |

1. 航班座位表

用于存储航班的座位信息，判断当前航班时候还有位置，并存储座位的等级信息，用于报表输出

表3-4 航班座位表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 约束 | 数据类型 | 说明 |
| FID | 外码：参照航班信息表  FLIGHTinfo(FID) | int | 航班编号 |
| SID | 由FID和SID复合主码 | varchar | 座位号 |
| USABLE |  | Tinyint(bool) | 座位是否可用 |
| SeatRank |  | varchar | 座位等级  (头等舱F  商务舱C  经济舱Y) |

### 3.5.3 数据流图

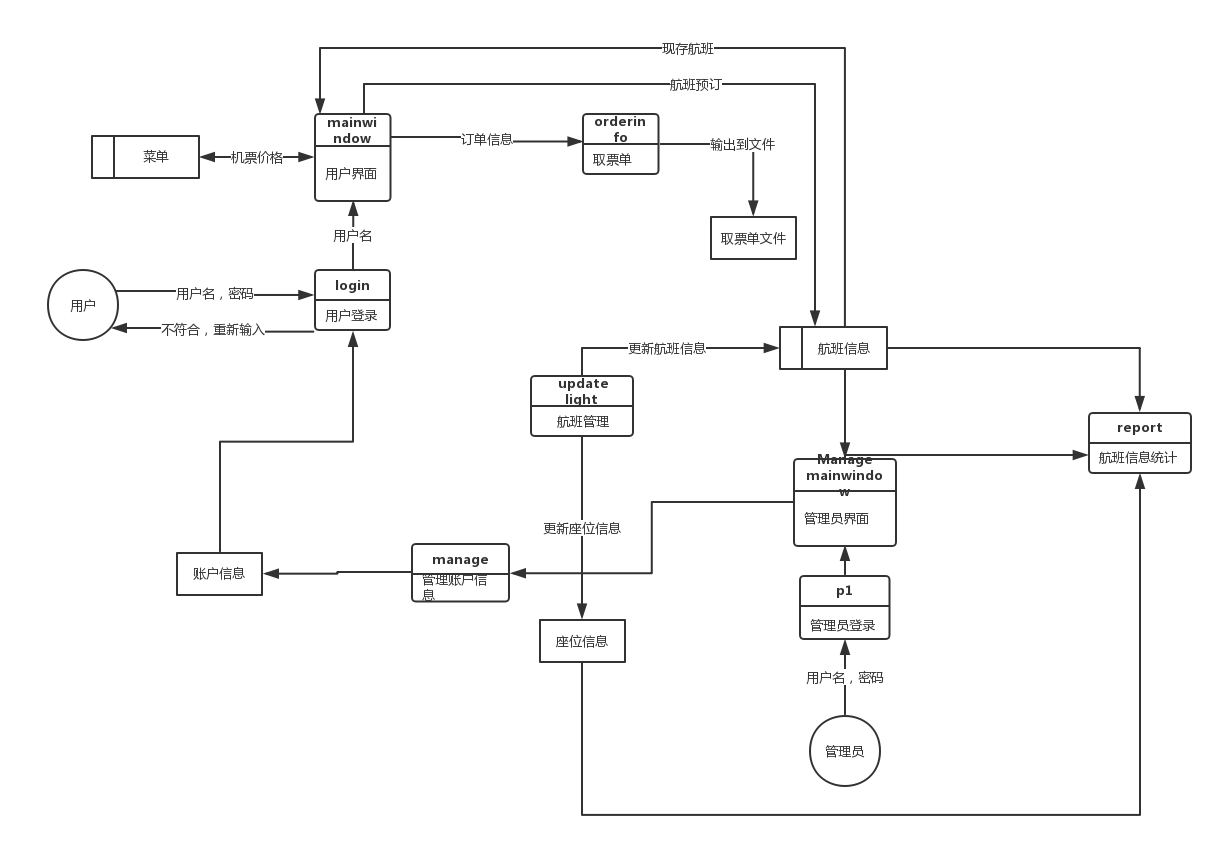


图3-1 数据流图

### 3.5.4 E-R图

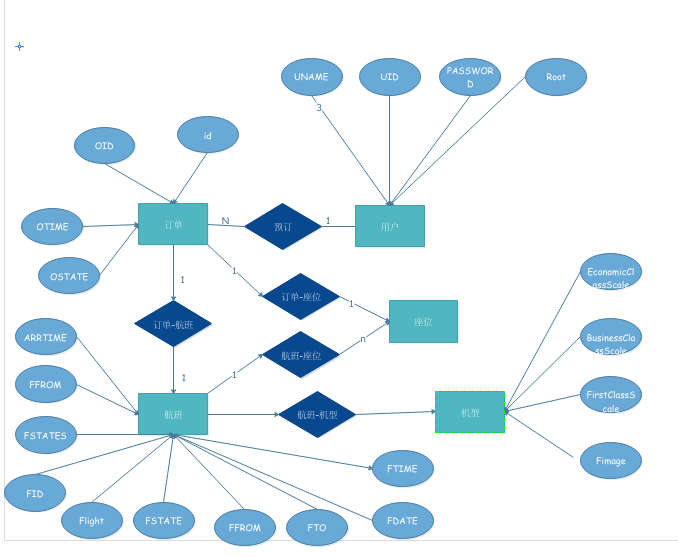


图3-1 E-R图

### 3.5.5 系统状态转换图

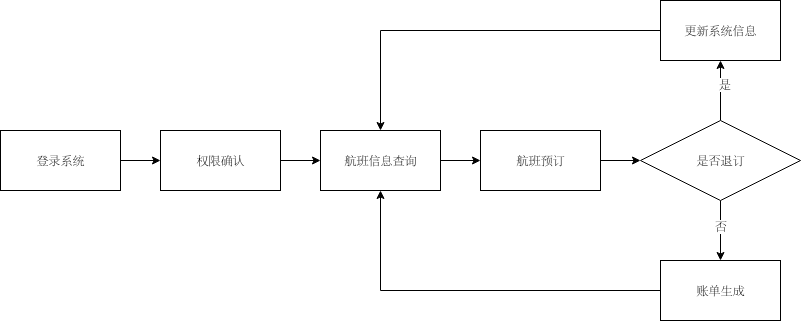


图3-1 系统操作状态转换图

## 3.6 系统性能

MariaDB(MySql)的性能指标为：

1. Query： 3w/s
2. Insert： 2w/s
3. Update： 8000/s
4. Delete： 8000/s

在满足服务器配置的情况下，常规操作可以在秒级完成。

在低峰时间，进行数据备份，可以在不影响系统正常运作的情况下，完成备份需求。

# 4 概要设计与详细设计

以数据为中心的思想，分析系统的数据结构设计，系统的命名规格定义等内容。

设计出系统的功能模块图或系统结构图。

结构化设计时采用过程模型工具（例如流程图、盒图或PAD图等）设计核心模块内容或算法描述。采用面向对象方法时给出核心类的核心服务的详细内容或算法描述，设计出类之间的关联，指明继承关系。

采用数据库存储数据时给数据库表的设计及表间关联，采用文件存储数据时给出文件的设计结构。

给出系统核心的数据结构。

## 4.1 系统功能模块划分

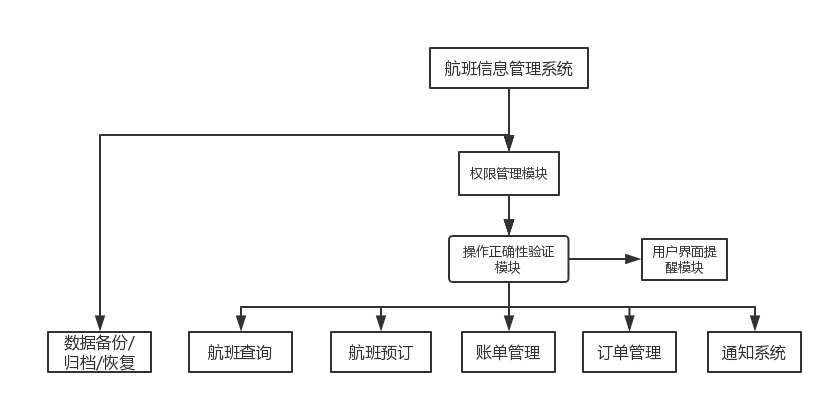


图3-1 系统功能模块图

## 4.2 系统架构方案

系统采用C/S架构，各模块之间进行较为清晰的接口定义，使模块之间尽量解耦合和独立，采用QT框架，C++语言开发，使用QSqlQuery模块与MySql(Mariadb)进行交互。

1. 用户权限：使用DB中的tabel来进行管理
2. 用户操作约束：在前端界面进行判断，SQL语句仅处理正常逻辑，便于系统实现和鲁棒性

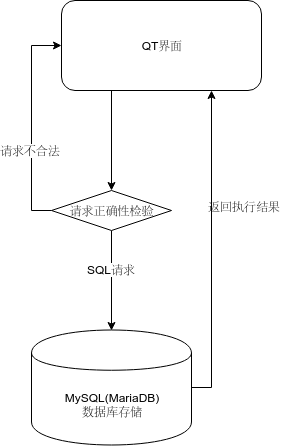


图4-1 系统C/S架构图

## 4.3 系统操作流程

### 4.3.1 常规使用人员操作流程

图4-1 常规使用人员操作流程图

### 4.3.2 数据统计人员操作流程

图4-1 数据统计人员操作流程图

### 4.3.3 软件维护人员操作流程

图4-1 软件维护人员操作流程图

## 4.4 前端实现方案

前端采用QT图形化布局，对于静态控件采用固定布局，在程序运行过程中修改控件属性达到显示需求。对于动态控件，在程序运行时进行生成，结束显示后销毁回收资源，从而尽可能的在前端资源占用和运行效率之间取得平衡。

图4-1 前端实现方案架构图

## 4.5 后端实现方案

采用mariadb进行数据库相关的操作，由QT将用户的操作转化为相应的数据库响应信息，生成数据库响应请求，后台转化为相应的SQL语句对数据库进行操作，最终返回操作结果。

图4-1 后端实现方案架构图

### 4.5.1 数据库事务管理

数据库主要的事务管理

### 4.5.2 历史数据管理

对于历史数据，航班管理系统将不再显示，故需要定时对于数据库中数据进行管理，此时设置的时间期限为一年，定期对数据库中的航班进行归档备份，同时将不可进行相关数据操作。

## 4.6 部件详细设计

### 4.6.1 航班事务管理

1. 航班添加

从飞机型号表(FMODELinfo)中利用SQL语句查询出飞机的型号，并设置型号选择款，在界面中进行输入值的约束，将输入值整理为SQL语句插入数据库，同时根据飞机型号查询座位排列，根据座位排列自动编排序号将座位插入座位信息表。

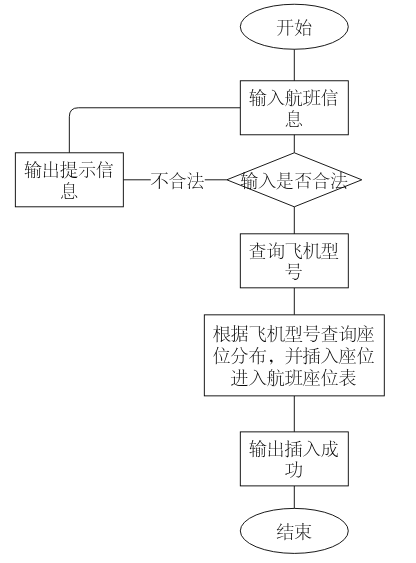


图4-4 航班添加流程图

1. 航班删除

根据删除航班的航班号FID查询是否存在用户订单为此航班，如果存在用户订单为此航班，则拒绝操作，并给出提示信息。否则，正常删除航班信息并删除座位信息。

1. 航班修改

当用户的权限为ROOT管理员权限时，将航班信息总表设置为可修改，但航班号FID为主码，设置锁定。

1. 航班查询

航班总表输出所有的航班，航班查询框根据条件筛选查询。

用户必须输入航班的始飞地和到达地，如果二者为空，给出提示信息。

用户必须输入正确的时间，如未修改时间，给出提示信息：利用初始设置时间，进行修改时间判断，是否修改。

根据输入的条件进行筛选，如果未输入，则默认条件为空。

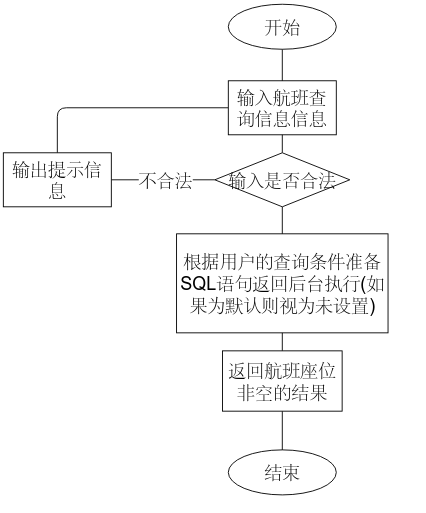


图4-5 航班查询流程图

### 4.6.2 订单事务管理

1. 预订

用户查询航班，选中想要预订的航班，根据选中的航班号FID进行座位查询，如无座位则返回。

进入座位选择界面：

根绝飞机机型的座位排布图自动生成座位的选择按钮阵列，查询飞机机型表获取机型座位分布图并显示，用户每次只能选择一个座位，且已经被预订的座位置为红色，同时用户可以根据上方列表进行选取。

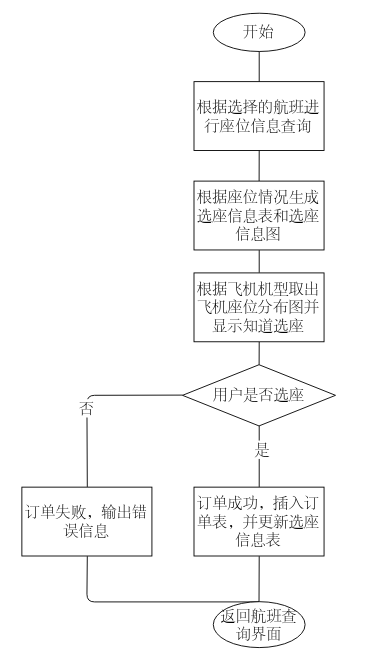


图4-6 选座生成订单流程图

1. 退订

判断订单状态，如订单状态为unfinish，则可以进行退订，同时将座位的状态设置为未占用。

如果订单状态为其他，则输出订单状态错误，无法退订错误信息。

1. 订单取票check-in

校验订单状态：如果订单状态为unfinish则继续

校验当前时间，如当前时间为订单的前一天或者订单的当天，则进行订单的checki-in，调用打印接口进行订单信息的打印输出。

### 4.6.3 权限系统

进入系统时，查询用户表user进行用户名和密码的确认，并查询是否是管理员信息进行相应设置。

如果是管理员：开放航班总表修改权限，开放航班添加删除权限。

### 4.6.4 报表系统

在进行座位选取时，进行座位信息的统计输出。

管理员可以查看所有航空公司最近三个月的统计图：

获取所有的航空公司名称并统计数目。

获取当前月份，并查询各公司在上上个月和上个月的统计情况，调用QT charts进行统计图绘制，并设置表头。

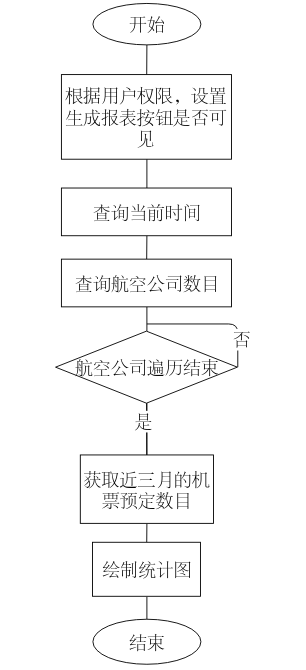


图4-7 统计图生成模块流程图

### 4.6.5 通用数据显示模块

因为QT中提供两种SQL接口，sqlmodel提供利用sql语句显示表格的少量功能，无法进行修改和选中设定一些操作，而sqltabel采用对象映射框架，完全避免的sql语句，为满足实验要求，减少模块耦合性和后期扩展性，继承sqlmodel模块进行重载，完成模块的自动化显示设置功能。

1. 中英文映射

利用SQL语句查询表头，获取表头进行自动设置，因本系统采用英文表头，且较为明晰，故未启用映射为中文。

映射模块：使用map对于表头映射为中文之后进行设置

1. 表头查询：

利用SQL语句查询表头并获得主码进行界面访问控制设置

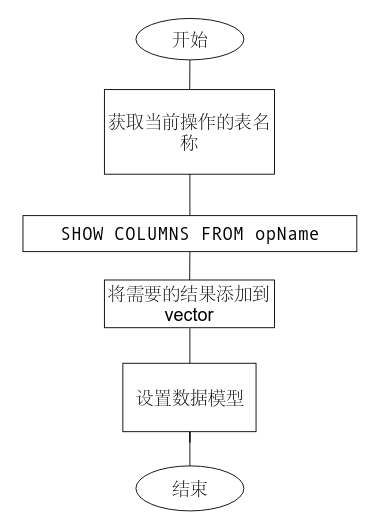


图4-8 表头查询模块流程图

1. 修改限制

利用SHOW COLUMNS FROM FLIGHTinfo where `Key`='PRI' 获取当前的主码，并设置为禁止修改

1. 用户切换窗口

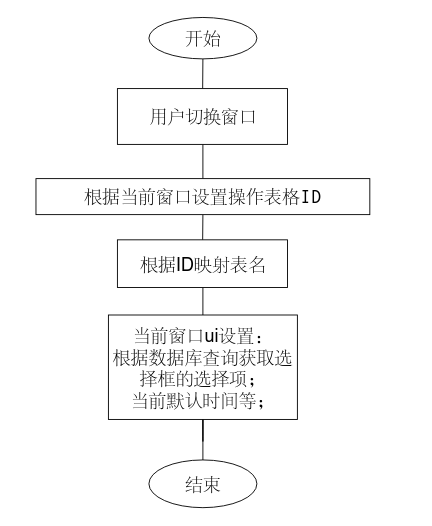


图4-9 用户切换窗口后台操作流程图

## 4.7 设计模式

整体采用C/S架构

## 4.8 注释及命名规范

采用驼峰命名法，所有的类名为当前类的功能。

* .h文件为头文件用于函数和信号槽的声明
* .cpp文件用于函数的实现和相关信号槽的绑定
* .ui文件为界面文件

整体架构

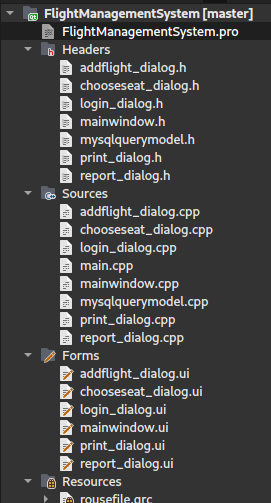


图4-1 整体文件架构图

* addflight\_dialog 添加航班相关事务的处理和界面绘制
* chooseseat\_dialog 添加航班相关事务的处理和界面绘制
* login\_dialog 登录处理和相关权限管理及系统初始化
* Mainwindow 主体窗口绘制和相关响应
* Mysqlquerymodel mysql相关操作的封装和继承重载
* print\_dialog 机票打印和行程单确认相关确认
* report\_dialog 统计报表相关事务的处理和界面绘制
* rousefile.qrc 界面的样式文件及图标文件

# 5 测试报告

## 5.1 测试环境

## 5.2 软件文档测试

## 5.3 源代码一致性检查



进行代码规范检查

## 5.4 界面(外观)测试

对照界面规范(在软件需求规格说明书中规定，或者由软件工程规范中给出)和界面表(在概要设计中给出)，检查各界面设计是否规范，包括：界面风格、表现形式、组件用法、字体选择、字号选择、色彩搭配、日期表现、计时方法、时间格式、对齐方式等等，是否符合规范、是否协调一致、是否便于操作。

## 5.5 功能测试

### 5.5.1 航班事务测试

1. 添加航班

如过未修改时间，则进行提示：

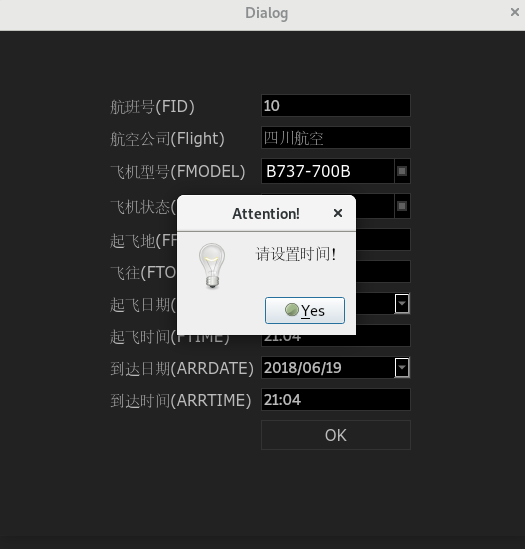


图5-10 添加航班时间错误提示图

如过数据错误(主码冲突)，则进行提示：



图4-11 添加航班主码错误提示图

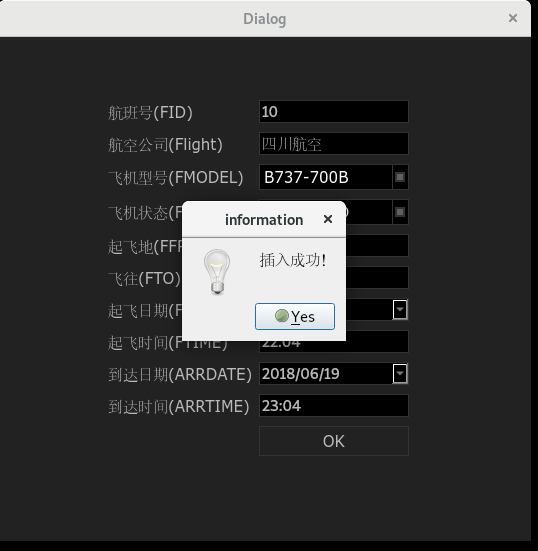


图4-12 添加航班成功提示图

1. 修改航班

管理员可以直接进行修改，如果修改不满足数据要求则不会改变，数据约束在数据库中进行。



图4-13 表格修改航班图

1. 删除航班

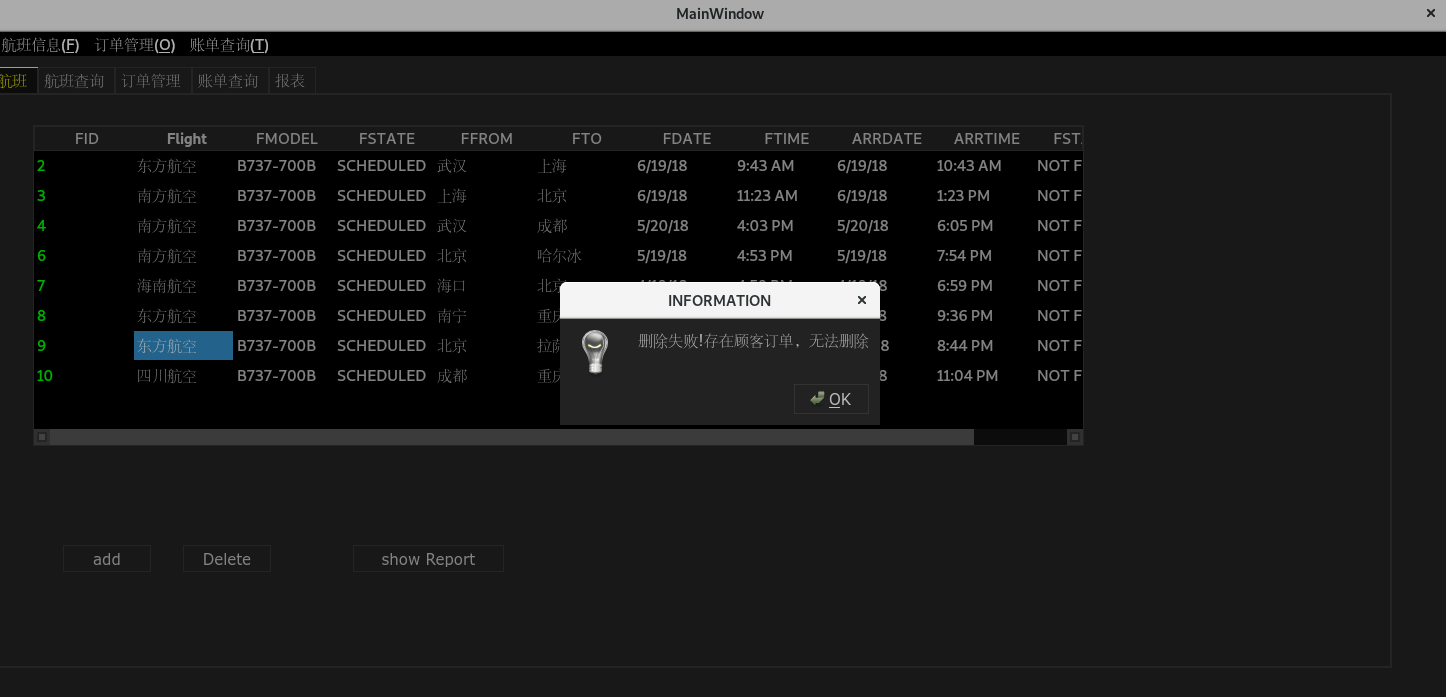


图4-14 删除航班错误图

如果正常删除，同时删除座位



图4-15 删除航班图

1. 航班查询



图4-16 航班查询图

### 5.5.2 订单事务测试

1. 预订

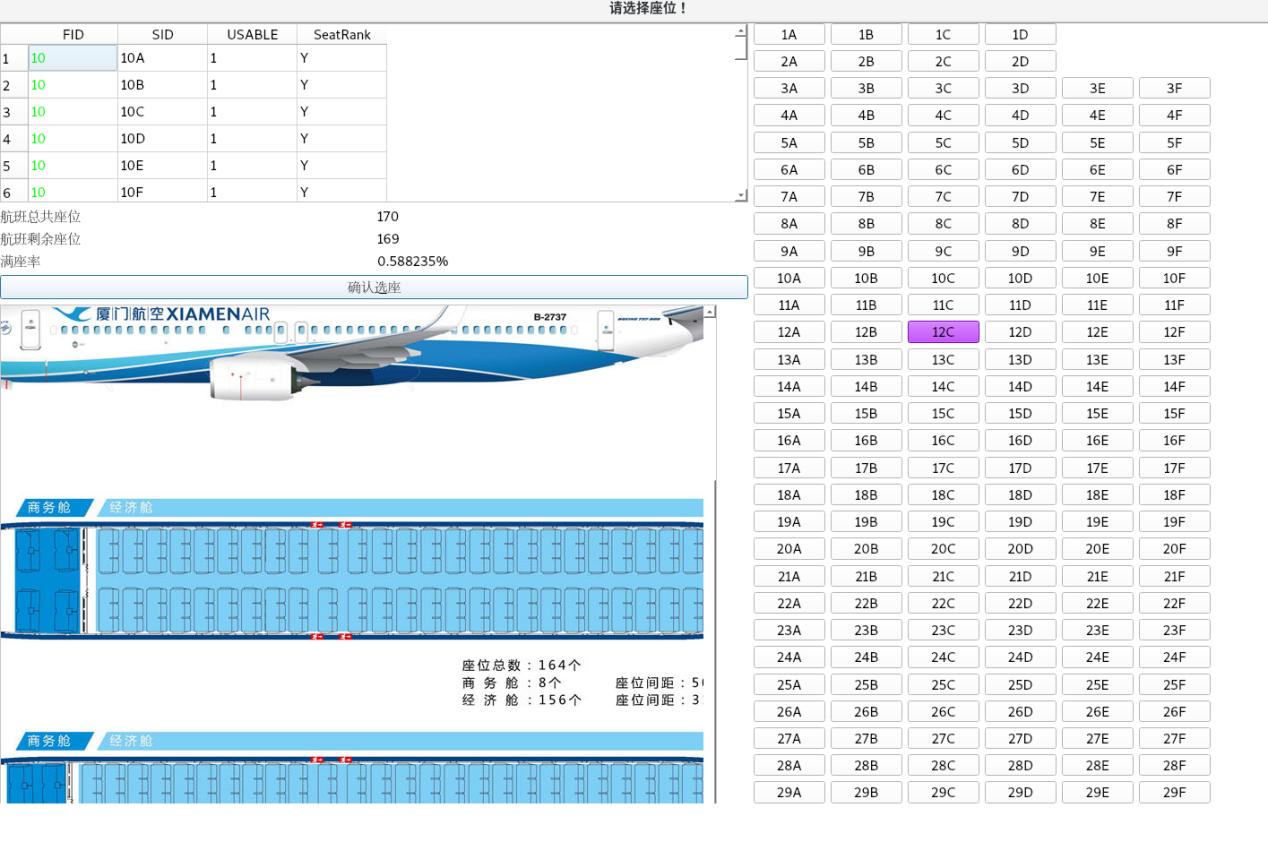


图4-17 选座系统图

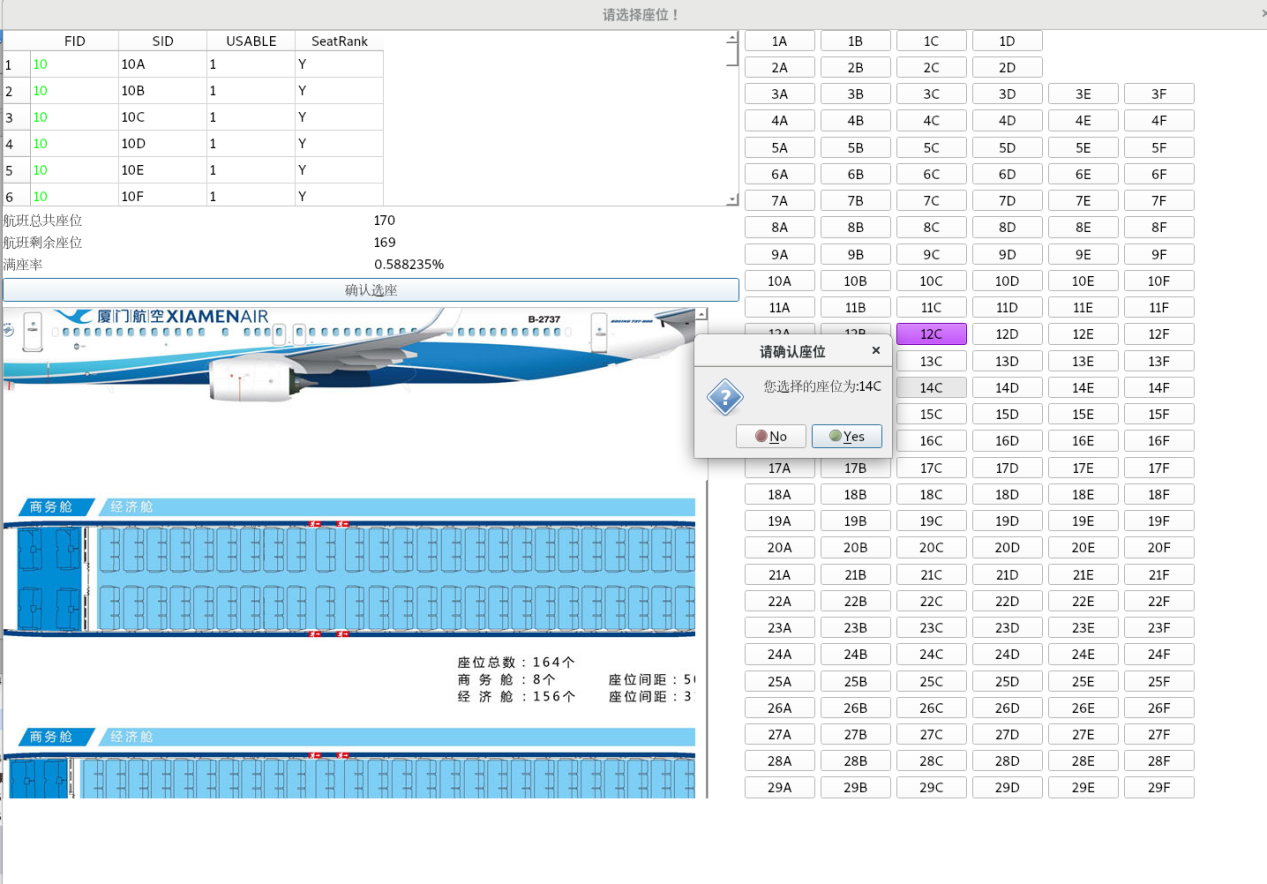


图4-18 确认选座图

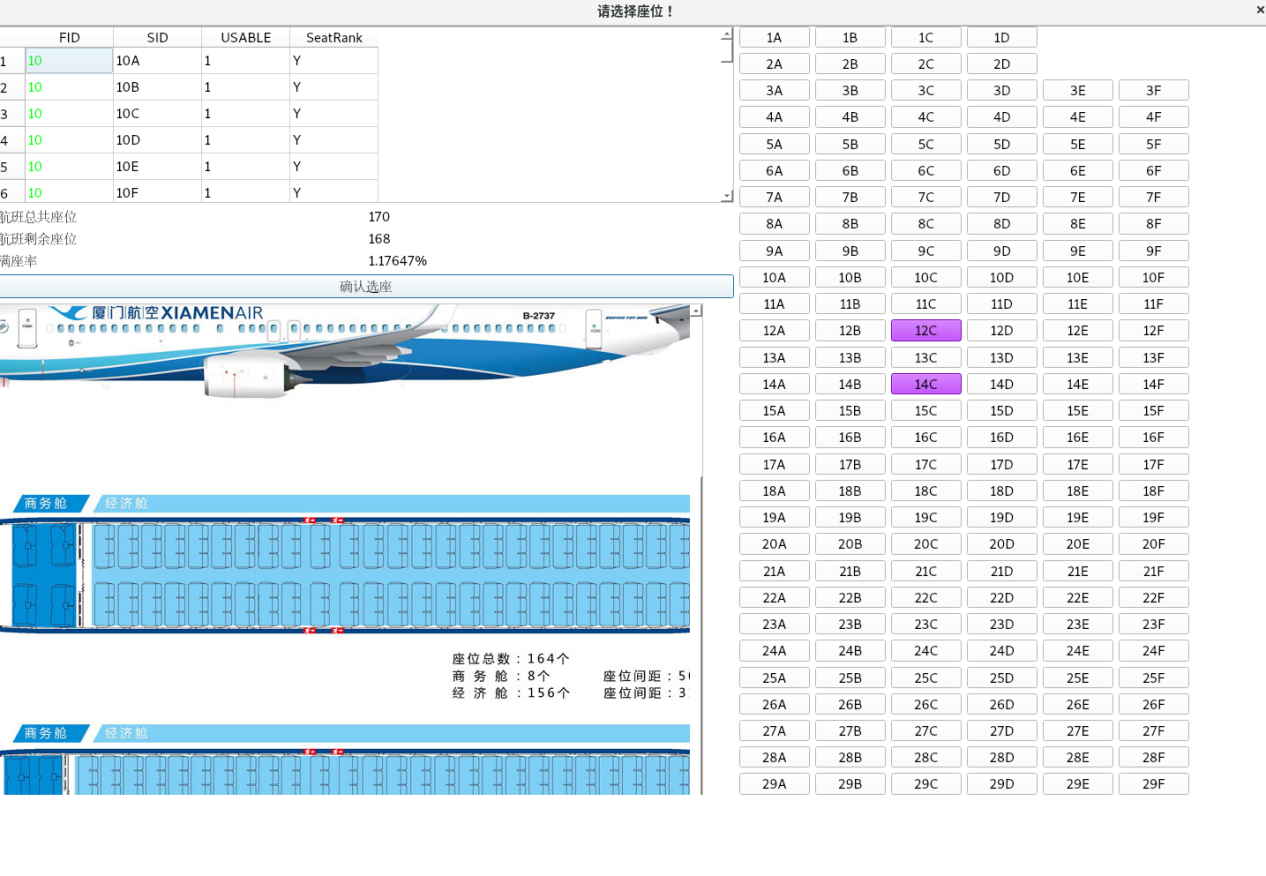


图4-19 再次查询图

1. 退订

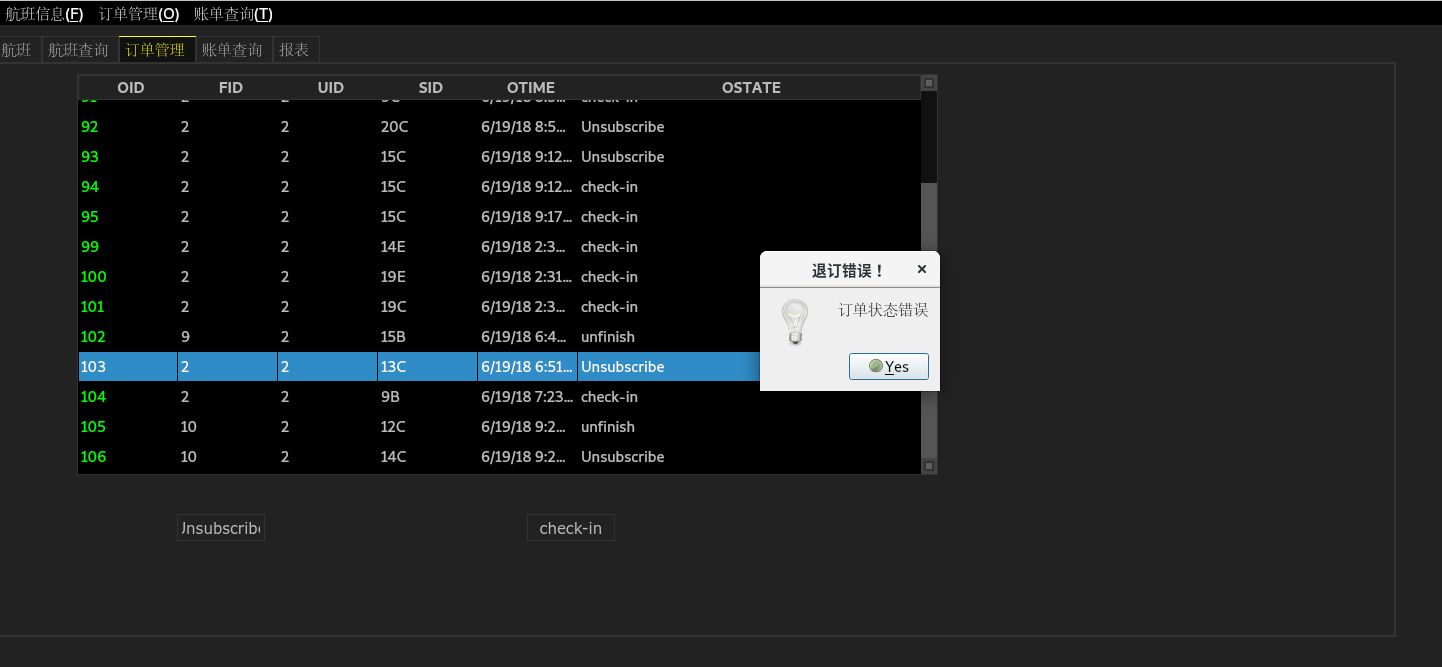


图4-20 退订错误图

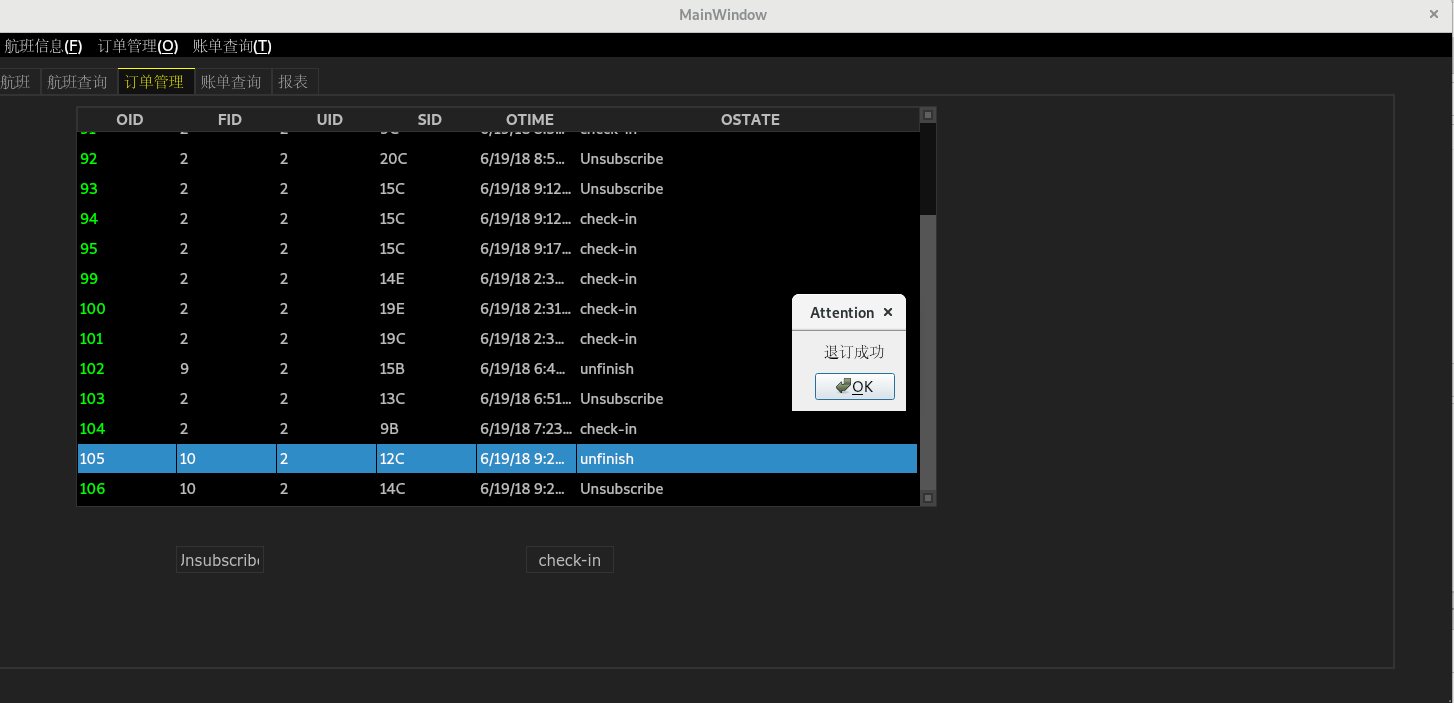


图4-21 退订成功图

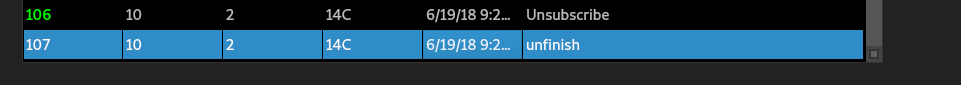


图4-22 退订再次预订图

1. 取票并输出机票

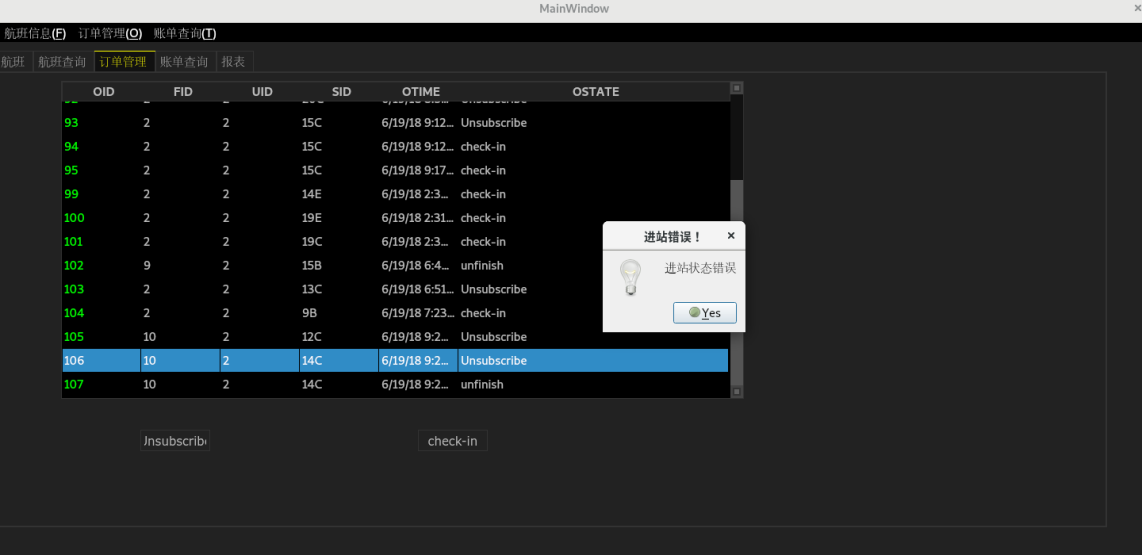


图4-23 检票错误图



图4-24 机票确认图



图4-25 机票模拟打印图（输出到文件）

### 5.6.3 报表测试

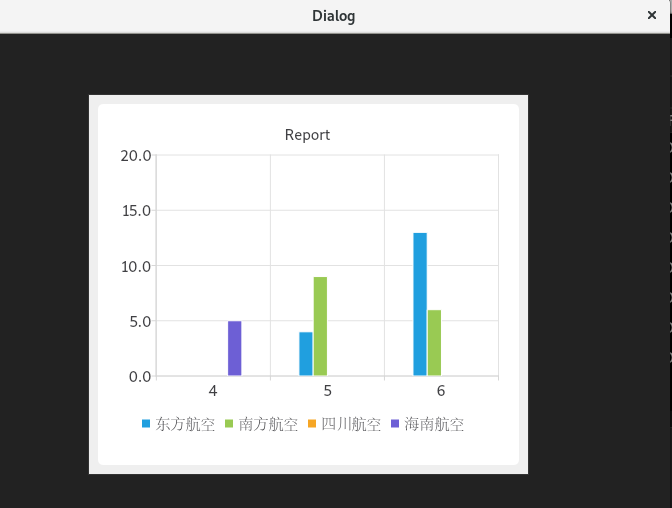


图4-26 报表测试图

## 5.6 稳定性(强度)测试

测试系统的能力最高实际限度，即检查软件在一些超负荷情况下，功能实现的情况。例如：要求软件进行某一行为的大量重复、输入大量的数据或大数值数据、对数据库进行大量复杂的查询等。

利用边界测试(最大值、最小值、N次循环)对系统进行模拟运行测试，观察其是否处于稳定状态。

## 5.7 性能测试

根据系统设计指标，或者对被测软件提出的性能指标，测试软件的运行性能，例如：传输连接最长时限、传输错误率、计算精度、记录精度、响应时限和恢复时限等。

## 5.8 强壮性(恢复)测试

采用人工的干扰使应用软件、平台软件或者系统硬件出错，中断正常使用，检测系统的恢复能力。进行强壮性测试时，应该参考性能测试相关的测试指标。

## 5.9 逻辑性测试

根据系统的功能逻辑图，测试软件是否按规定的逻辑路径运行，选择一些极限数据判断软件运行是否存在错误或非法路径，从而发现系统的逻辑错误或非法后门。

## 5.9 破坏性测试

输入错误的或非法的数据(类型)，检查系统的报错纠错的能力及稳定性。并测试可连续使用多长时间而系统不崩溃。

## 5.10 安全性测试

验证安装在系统内的保护机构确实能够对系统进行保护，使之不受各种非常的干扰，安全测试时需要设计一些测试用例试图突破系统的安全保密措施，检验系统是否有安全保密的漏洞。

说明：进行安全测试时，必须遵循相关的安全规定，并且有业主派员参加。

# 6 项目管理

自己选择一个模型，估算出系统的工作量（功能点或者源代码行数，应当有依据和过程），自己假定获得了项目的历史生产率数据（FP/PM或L/PM，元/FP,元/L等），估算出项目需要的人数，持续工期和总体成本。

拟定安排的项目人员，同时描述系统的硬件资源和软件资源。

利用甘特图或者工程网络描述项目的进度计划。

## 6.1 项目计划

### 6.1.1项目整体实施流程

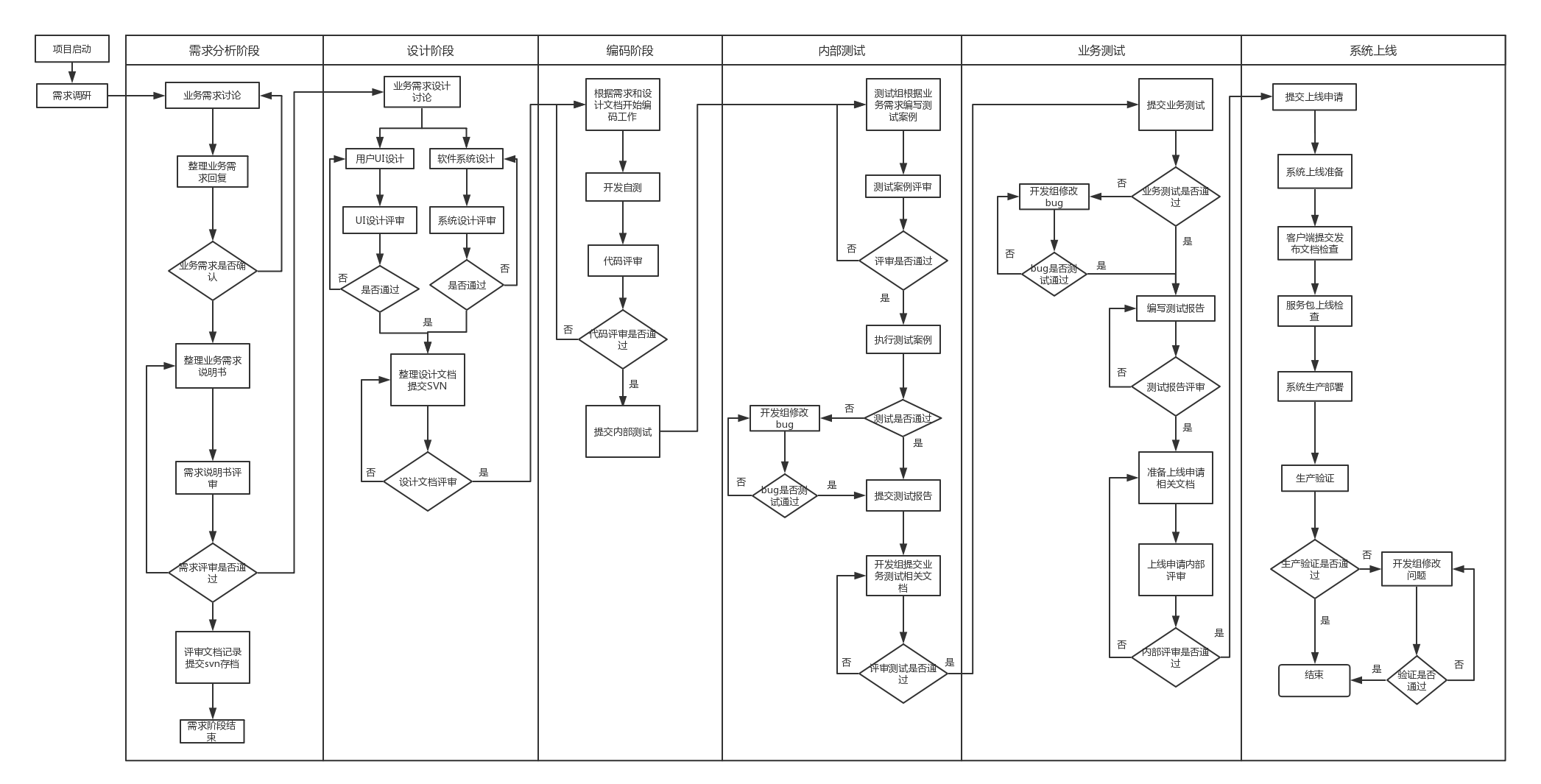


图6-1 项目整体实施流程图

### 6.1.2 项目进度计划

采用project进行项目规划

### 6.1.2 项目进度计划

采用project进行项目规划

表6-1 项目进度表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务名称 | 工期 | 开始时间 | 完成时间 | 前置任务 | 资源名称 |
| **项目范围规划** | **3.5 个工作日** | **2018年5月20日** | **2018年5月23日** |  |  |
| 确定项目范围 | 4 工时 | 2018年5月20日 | 2018年5月20日 |  | 管理人员 |
| 获得项目所需资金 | 1 个工作日 | 2018年5月20日 | 2018年5月21日 | 2 | 管理人员 |
| 定义预备资源 | 1 个工作日 | 2018年5月21日 | 2018年5月22日 | 3 | 项目经理 |
| 获得核心资源 | 1 个工作日 | 2018年5月22日 | 2018年5月23日 | 4 | 项目经理 |
| 项目范围规划完成 | 0 个工作日 | 2018年5月23日 | 2018年5月23日 | 5 |  |
| **分析/软件需求** | **9 个工作日** | **2018年5月23日** | **2018年6月5日** |  |  |
| 行为需求分析 | 1 个工作日 | 2018年5月23日 | 2018年5月24日 | 6 | 分析人员 |
| 起草初步的软件规范 | 2 个工作日 | 2018年5月24日 | 2018年5月28日 | 8 | 分析人员 |
| 制定初步预算 | 2 个工作日 | 2018年5月28日 | 2018年5月30日 | 9 | 项目经理 |
| 工作组共同审阅软件规范/预算 | 4 工时 | 2018年5月30日 | 2018年5月30日 | 10 | 项目经理,分析人员 |
| 根据反馈修改软件规范 | 1 个工作日 | 2018年5月31日 | 2018年5月31日 | 11 | 分析人员 |
| 确定交付期限 | 1 个工作日 | 2018年6月1日 | 2018年6月1日 | 12 | 项目经理 |
| 获得开展后续工作的批准(概念、期限和预算) | 4 工时 | 2018年6月4日 | 2018年6月4日 | 13 | 管理人员,项目经理 |
| 获得所需资源 | 1 个工作日 | 2018年6月4日 | 2018年6月5日 | 14 | 项目经理 |
| 分析工作完成 | 0 个工作日 | 2018年6月5日 | 2018年6月5日 | 15 |  |
| **设计** | **8.5 个工作日** | **2018年6月5日** | **2018年6月15日** |  |  |
| 审阅初步的软件规范 | 1 个工作日 | 2018年6月5日 | 2018年6月6日 | 16 | 分析人员 |
| 制定功能规范 | 2 个工作日 | 2018年6月6日 | 2018年6月8日 | 18 | 分析人员 |
| 根据功能规范开发原型 | 2 个工作日 | 2018年6月8日 | 2018年6月12日 | 19 | 分析人员 |
| 审阅功能规范 | 2 个工作日 | 2018年6月12日 | 2018年6月14日 | 20 | 管理人员 |
| 根据反馈修改功能规范 | 1 个工作日 | 2018年6月14日 | 2018年6月15日 | 21 | 管理人员 |
| 获得开展后续工作的批准 | 4 工时 | 2018年6月15日 | 2018年6月15日 | 22 | 管理人员,项目经理 |
| 设计工作完成 | 0 个工作日 | 2018年6月15日 | 2018年6月15日 | 23 |  |
| **开发** | **18 个工作日** | **2018年6月18日** | **2018年7月11日** |  |  |
| 审阅功能规范 | 1 个工作日 | 2018年6月18日 | 2018年6月18日 | 24 | 开发人员 |
| 确定模块化/分层设计参数 | 1 个工作日 | 2018年6月19日 | 2018年6月19日 | 26 | 开发人员 |
| 分派任务给开发人员 | 1 个工作日 | 2018年6月20日 | 2018年6月20日 | 27 | 开发人员 |
| 编写代码 | 15 个工作日 | 2018年6月21日 | 2018年7月11日 | 28 | 开发人员 |
| 开发人员测试(初步调试) | 7 个工作日 | 2018年6月26日 | 2018年7月5日 | 29FS-75% | 开发人员 |
| 开发工作完毕 | 0 个工作日 | 2018年7月5日 | 2018年7月5日 | 30 |  |
| **测试** | **26.75 个工作日** | **2018年6月18日** | **2018年7月24日** |  |  |
| 根据产品规范制定单元测试计划 | 1 个工作日 | 2018年6月18日 | 2018年6月18日 | 24 | 测试人员 |
| 根据产品规范制定整体测试计划 | 1 个工作日 | 2018年6月18日 | 2018年6月18日 | 24 | 测试人员 |
| **单元测试** | **5 个工作日** | **2018年7月5日** | **2018年7月12日** |  |  |
| 审阅模块化代码 | 1 个工作日 | 2018年7月5日 | 2018年7月6日 | 33,31 | 测试人员 |
| 测试组件模块是否符合产品规范 | 1 个工作日 | 2018年7月6日 | 2018年7月9日 | 31,36 | 测试人员 |
| 找出不符合产品规范的异常情况 | 1 个工作日 | 2018年7月9日 | 2018年7月10日 | 37 | 测试人员 |
| 修改代码 | 1 个工作日 | 2018年7月10日 | 2018年7月11日 | 38 | 测试人员 |
| 重新测试经过修改的代码 | 1 个工作日 | 2018年7月11日 | 2018年7月12日 | 39 | 测试人员 |
| 单元测试完成 | 0 个工作日 | 2018年7月12日 | 2018年7月12日 | 40 |  |
| **整体测试** | **8 个工作日** | **2018年7月12日** | **2018年7月24日** |  |  |
| 测试模块集成情况 | 2 个工作日 | 2018年7月12日 | 2018年7月16日 | 41 | 测试人员 |
| 找出不符合规范的异常情况 | 1 个工作日 | 2018年7月16日 | 2018年7月17日 | 43 | 测试人员 |
| 修改代码 | 3 个工作日 | 2018年7月17日 | 2018年7月20日 | 44 | 测试人员 |
| 重新测试经过修改的代码 | 2 个工作日 | 2018年7月20日 | 2018年7月24日 | 45 | 测试人员 |
| 整体测试完成 | 0 个工作日 | 2018年7月24日 | 2018年7月24日 | 46 |  |
| **培训** | **24.75 个工作日** | **2018年6月18日** | **2018年7月20日** |  |  |
| 制定针对最终用户的培训规范 | 3 个工作日 | 2018年6月18日 | 2018年6月20日 | 24 | 培训人员 |
| 制定针对产品技术支持人员的培训规范 | 3 个工作日 | 2018年6月18日 | 2018年6月20日 | 24 | 培训人员 |
| 确定培训方法(基于计算机的培训、教室授课等) | 2 个工作日 | 2018年6月18日 | 2018年6月19日 | 24 | 培训人员 |
| 编写培训材料 | 1 周工时 | 2018年7月5日 | 2018年7月12日 | 49,31,50,51 | 培训人员 |
| 研究培训材料的可用性 | 1 个工作日 | 2018年7月12日 | 2018年7月13日 | 52 | 培训人员 |
| 对培训材料进行最后处理 | 3 个工作日 | 2018年7月13日 | 2018年7月18日 | 53 | 培训人员 |
| 制定培训机制 | 2 个工作日 | 2018年7月18日 | 2018年7月20日 | 54 | 培训人员 |
| 培训材料完成 | 0 个工作日 | 2018年7月20日 | 2018年7月20日 | 55 |  |
| **文档** | **20.5 个工作日** | **2018年6月18日** | **2018年7月16日** |  |  |
| 制定“帮助”规范 | 1 个工作日 | 2018年6月18日 | 2018年6月18日 | 24 | 技术联络人员 |
| 开发“帮助”系统 | 1 周工时 | 2018年7月2日 | 2018年7月9日 | 58,29FS-50% | 技术联络人员 |
| 审阅“帮助”文档 | 3 个工作日 | 2018年7月9日 | 2018年7月12日 | 59 | 技术联络人员 |
| 根据反馈修改“帮助”文档 | 2 个工作日 | 2018年7月12日 | 2018年7月16日 | 60 | 技术联络人员 |
| 制定用户手册规范 | 2 个工作日 | 2018年6月18日 | 2018年6月19日 | 24 | 技术联络人员 |
| 编写用户手册 | 1 周工时 | 2018年7月2日 | 2018年7月9日 | 62,29FS-50% | 技术联络人员 |
| 审阅所有的用户文档 | 2 个工作日 | 2018年7月9日 | 2018年7月11日 | 63 | 技术联络人员 |
| 根据反馈修改用户文档 | 2 个工作日 | 2018年7月11日 | 2018年7月13日 | 64 | 技术联络人员 |
| 文档完成 | 0 个工作日 | 2018年7月16日 | 2018年7月16日 | 65,61 |  |
| **试生产** | **42.25 个工作日** | **2018年6月5日** | **2018年8月2日** |  |  |
| 确定测试群体 | 1 个工作日 | 2018年6月5日 | 2018年6月6日 | 16 | 项目经理 |
| 确定软件分发机制 | 1 个工作日 | 2018年6月6日 | 2018年6月7日 | 68 |  |
| 安装/部署软件 | 1 个工作日 | 2018年7月24日 | 2018年7月25日 | 47,69,66,56 | 部署小组 |
| 获得用户反馈 | 1 周工时 | 2018年7月25日 | 2018年8月1日 | 70 | 部署小组 |
| 评估测试信息 | 1 个工作日 | 2018年8月1日 | 2018年8月2日 | 71 | 部署小组 |
| 试生产工作完成 | 0 个工作日 | 2018年8月2日 | 2018年8月2日 | 72 |  |
| **部署** | **5 个工作日** | **2018年8月2日** | **2018年8月9日** |  |  |
| 确定最终部署策略 | 1 个工作日 | 2018年8月2日 | 2018年8月3日 | 73 | 部署小组 |
| 确定部署方法 | 1 个工作日 | 2018年8月3日 | 2018年8月6日 | 75 | 部署小组 |
| 获得部署所需资源 | 1 个工作日 | 2018年8月6日 | 2018年8月7日 | 76 | 部署小组 |
| 培训技术支持人员 | 1 个工作日 | 2018年8月7日 | 2018年8月8日 | 77 | 部署小组 |
| 部署软件 | 1 个工作日 | 2018年8月8日 | 2018年8月9日 | 78 | 部署小组 |
| 部署工作完成 | 0 个工作日 | 2018年8月9日 | 2018年8月9日 | 79 |  |
| **实施工作结束后的回顾** | **3 个工作日** | **2018年8月9日** | **2018年8月14日** |  |  |
| 将经验教训记录存档 | 1 个工作日 | 2018年8月9日 | 2018年8月10日 | 80 | 项目经理 |
| 分发给工作组成员 | 1 个工作日 | 2018年8月10日 | 2018年8月13日 | 82 | 项目经理 |
| 建立软件维护小组 | 1 个工作日 | 2018年8月13日 | 2018年8月14日 | 83 | 项目经理 |
| 回顾完成 | 0 个工作日 | 2018年8月14日 | 2018年8月14日 | 84 |  |
| 软件开发模板结束 | 0 个工作日 | 2018年8月14日 | 2018年8月14日 | 85 |  |

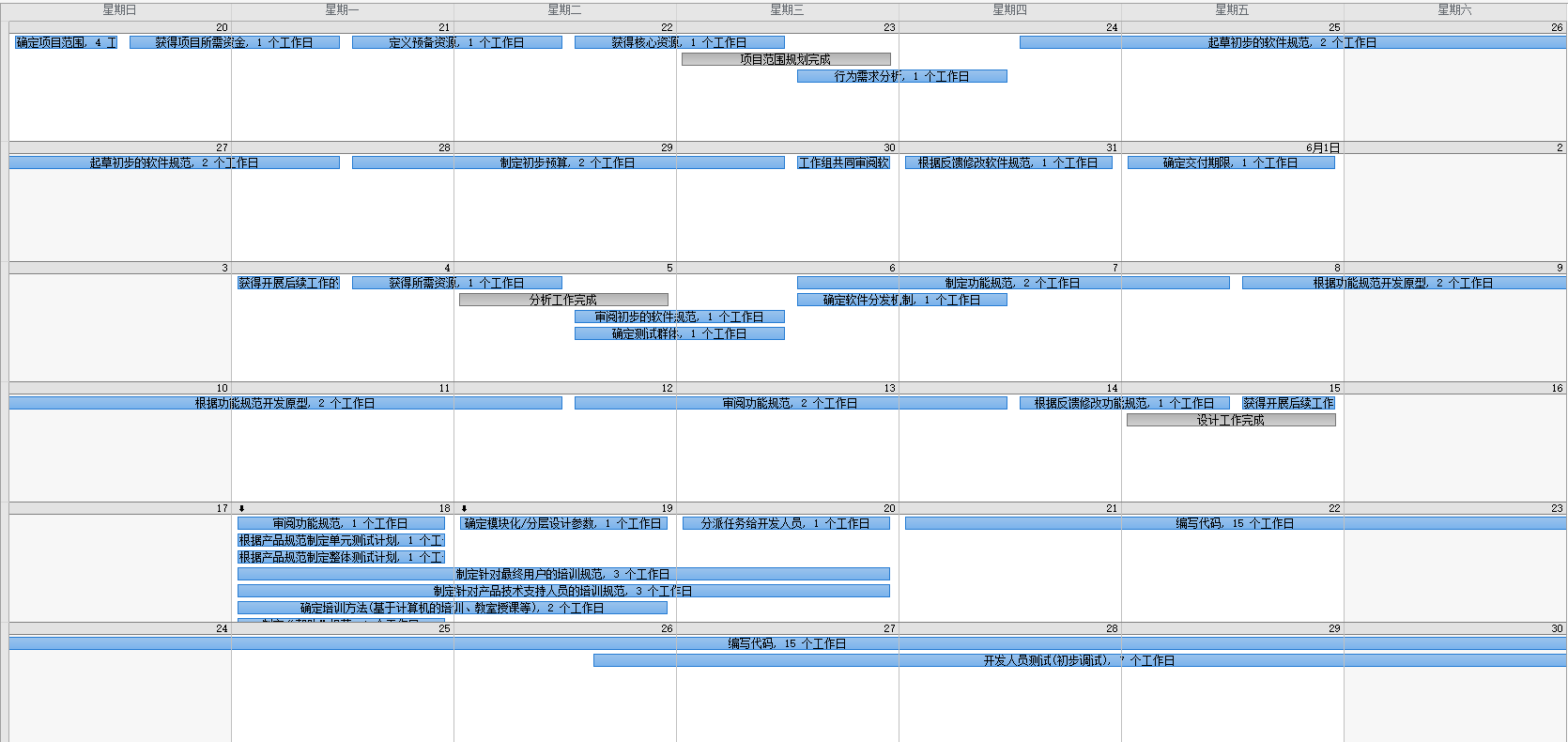


图6-1 项目进度计划图1

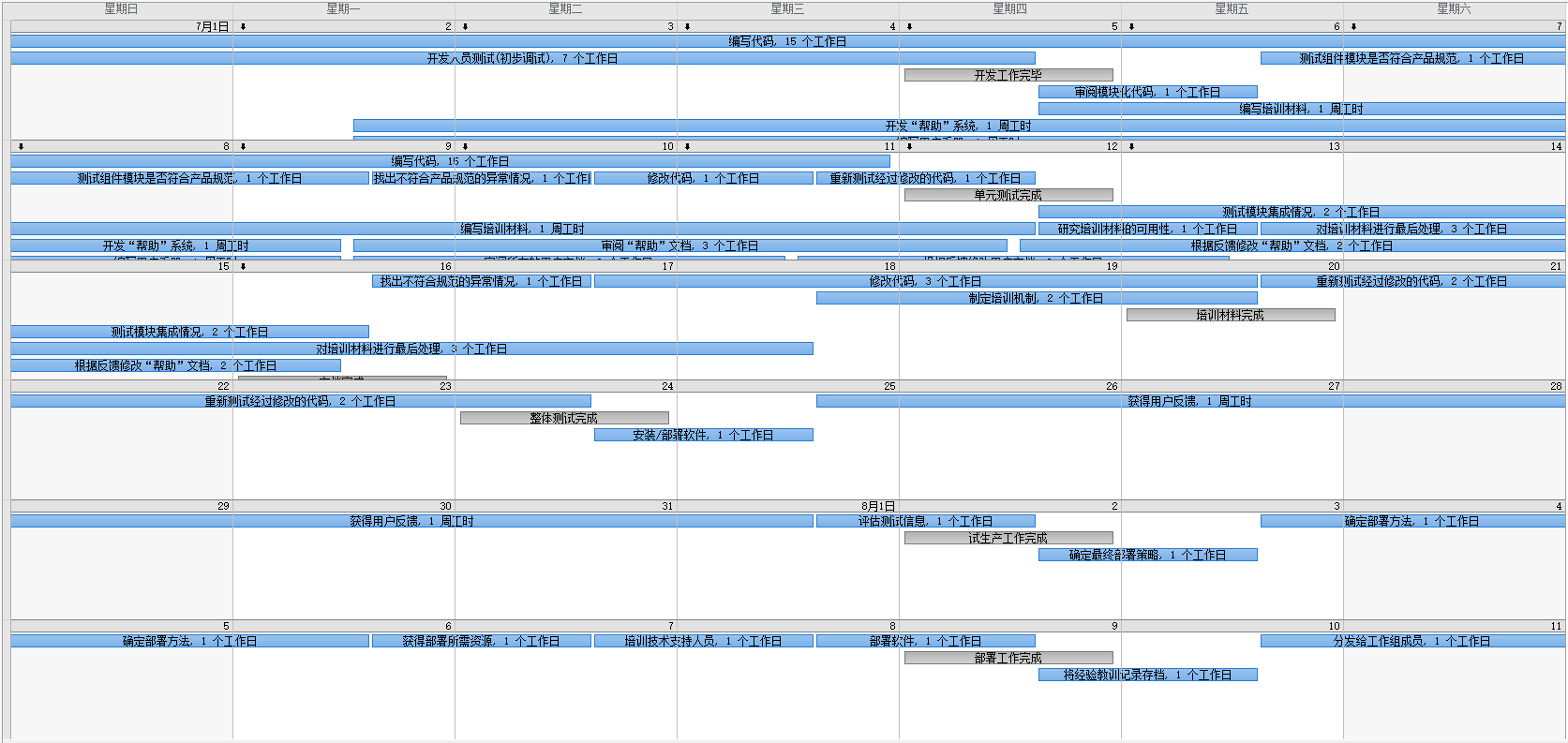


图6-1 项目进度计划图2

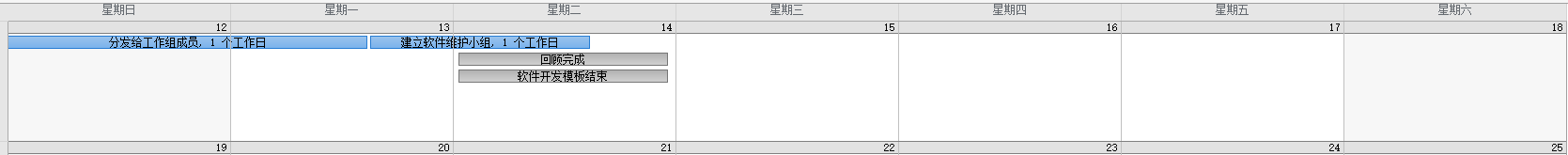


图6-1 项目进度计划图3

表6-1 项目里程碑

|  |  |
| --- | --- |
| 名称 | 完成时间 |
| 项目范围规划完成 | 2018年5月23日 |
| 分析工作完成 | 2018年6月5日 |
| 设计工作完成 | 2018年6月15日 |
| 开发工作完毕 | 2018年7月5日 |
| 单元测试完成 | 2018年7月12日 |
| 整体测试完成 | 2018年7月24日 |
| 培训材料完成 | 2018年7月20日 |
| 文档完成 | 2018年7月16日 |
| 试生产工作完成 | 2018年8月2日 |
| 部署工作完成 | 2018年8月9日 |
| 回顾完成 | 2018年8月14日 |
| 软件开发模板结束 | 2018年8月14日 |

## 6.2 项目所需人员

管理人员\*1

项目经理\*1

分析人员\*1

开发人员\*2

测试人员\*2

培训人员\*1

技术联络人员\*1

部署人员\*1

## 6.3 项目功能点估算

1. 航班查询：进行航班的查询操作
   1. 模糊查询：利用时间，地点，价格等条件查询航班
   2. 精确查询：利用航班号查询
2. 航班预订：进行航班预订的相关操作
   1. 座位选取
   2. 订单生成和确认
   3. 于订单管理模块，账单管理模块交互
3. 订单管理：进行订单的相关操作
   1. 管理订单状态，已经完成的订单和约束订单无法操作
   2. 更新订单状态
4. 账单管理：生成系统的账单
5. 通知系统：进行相关查询用户的状态提醒

### 6.3.1 功能点定义

录入订单、修改订单、删除订单是EI；

查询订单是EO

统计订单是EQ

备份文件为EIF

订单和客户是ILF

### 6.3.2 功能点计算

表6-1 ILF功能点计算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ILF内部逻辑文件 | RET | DET个数 | 复杂度 | 未调整的FP个数 |
| 用户信息 | 4  用户名 UID  用户姓名 UNAME  密码 PASSWORD  用户权限ROOT | 1 | 低 | 7 |
| 订单信息 | 部门基本信息，共1个。 | 2 | 低 | 7 |

表6-1 EIF功能点计算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EIF外部接口文件 | RET | DET个数 | 复杂度 | 未调整的FP个数 |
| 机票 | 员工基本信息、工资信息，共2个。 | 4个 | 低 | 5 |

表6-1 EI、EQ和EO的功能点数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EI | FTR | DET个数 | 复杂度 | 未调整的FP个数 |
| 添加用户信息 | 员工、部门、工资表 | 员工信息的两个标签控件内容不是用户输入的，因此不算。共16个。 部门信息与员工信息中的部门字段重复，因此一个都不算。 工资表中的员工ID和名称不能重复，因此只能算金额和单位，所以共2个。  18个 | 高 | 6 |
| 修改用户信息 | 员工、部门、工资表 | 18个 同上 | 高 | 6 |
| 删除用户信息 | 员工、部门、工资表 | 1个 员工ID | 中等 | 4 |
| 添加航班信息 | 部门 | 1个  一个标签控件的内容不是用户输入的，因此不算 | 低 | 3 |
| 修改航班信息 | 部门 | 1个  一个标签控件的内容不是用户输入的，因此不算 | 低 | 3 |
| 删除航班信息 | 部门 | 1个 部门ID | 低 | 3 |
| 添加飞机型号 | 部门 | 1个  一个标签控件的内容不是用户输入的，因此不算 | 低 | 3 |
| 修改飞机型号 | 部门 | 1个  一个标签控件的内容不是用户输入的，因此不算 | 低 | 3 |
| 删除飞机型号 | 部门 | 1个 部门ID | 低 | 3 |
| 添加订单 |  |  |  |  |
| 修改订单 |  |  |  |  |
| 删除订单 |  |  |  |  |

|  |  |
| --- | --- |
| 合计：25个 |  |

表6-1 EQ功能点数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EQ | FTR | DET个数 | 复杂度 | 未调整的FP个数 |
| 查询用户信息 | 员工、部门、工资表 | 20 | 高 | 6 |
| 查询订单信息 | 部门 | 2 | 低 | 3 |
| 查询航班信息 |  |  |  |  |

|  |
| --- |
| 合计：9个 |

表6-1 EO功能点数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EO | FTR | DET个数 | 复杂度 | 未调整的FP个数 |
| 输出机票 | 员工、工资表 | 员工ID、员工名称、年份、年薪、单位  共5个 | 低 | 4 |
| 输出统计报表 |  |  |  |  |

### 6.3.3 功能点调整

表6-1 通用系统特性及其影响程度表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 系统特性 | 分数 | | 数据通讯 | 5 | | 分布式数据处理 | 2 | | 性能 | 5 | | 大业务量配置 | 2 | | 事务处理率 | 2 | | 在线数据输入 | 5 | | 最终用户效率 | 2 | | 在线更新 | 1 | | **复杂处理** | 0 | | 可复用性 | 3 | | 易安装性 | 3 | | 易操作性 | 1 | | 多场地 | 1 | | 支持变更 | 1 |   合计：33 |
| 调整因子  =  33 \* 0.01  + 0.65  =  0.98 |

调整后的功能点数量

FP Count =UFP \* VAF

=(19 + 25 + 9 + 5)\* 0.98

= 48.72个

## 6.4 项目成本估算

表6-1 项目工时表

|  |  |
| --- | --- |
| 资源名称 | 工时 |
| **管理人员** | **44 工时** |
| 确定项目范围 | 4 工时 |
| 获得项目所需资金 | 8 工时 |
| 获得开展后续工作的批准(概念、期限和预算) | 4 工时 |
| 审阅功能规范 | 16 工时 |
| 根据反馈修改功能规范 | 8 工时 |
| 获得开展后续工作的批准 | 4 工时 |
| **项目经理** | **92 工时** |
| 定义预备资源 | 8 工时 |
| 获得核心资源 | 8 工时 |
| 制定初步预算 | 16 工时 |
| 工作组共同审阅软件规范/预算 | 4 工时 |
| 确定交付期限 | 8 工时 |
| 获得开展后续工作的批准(概念、期限和预算) | 4 工时 |
| 获得所需资源 | 8 工时 |
| 获得开展后续工作的批准 | 4 工时 |
| 确定测试群体 | 8 工时 |
| 将经验教训记录存档 | 8 工时 |
| 分发给工作组成员 | 8 工时 |
| 建立软件维护小组 | 8 工时 |
| 分析人员 | 76 工时 |
| 行为需求分析 | 8 工时 |
| 起草初步的软件规范 | 16 工时 |
| 工作组共同审阅软件规范/预算 | 4 工时 |
| 根据反馈修改软件规范 | 8 工时 |
| 审阅初步的软件规范 | 8 工时 |
| 制定功能规范 | 16 工时 |
| 根据功能规范开发原型 | 16 工时 |
| **开发人员** | **200 工时** |
| 审阅功能规范 | 8 工时 |
| 确定模块化/分层设计参数 | 8 工时 |
| 分派任务给开发人员 | 8 工时 |
| 编写代码 | 120 工时 |
| 开发人员测试(初步调试) | 56 工时 |
| **测试人员** | **120 工时** |
| 根据产品规范制定单元测试计划 | 8 工时 |
| 根据产品规范制定整体测试计划 | 8 工时 |
| 审阅模块化代码 | 8 工时 |
| 测试组件模块是否符合产品规范 | 8 工时 |
| 找出不符合产品规范的异常情况 | 8 工时 |
| 修改代码 | 8 工时 |
| 重新测试经过修改的代码 | 8 工时 |
| 测试模块集成情况 | 16 工时 |
| 找出不符合规范的异常情况 | 8 工时 |
| 修改代码 | 24 工时 |
| 重新测试经过修改的代码 | 16 工时 |
| **培训人员** | **152 工时** |
| 制定针对最终用户的培训规范 | 24 工时 |
| 制定针对产品技术支持人员的培训规范 | 24 工时 |
| 确定培训方法(基于计算机的培训、教室授课等) | 16 工时 |
| 编写培训材料 | 40 工时 |
| 研究培训材料的可用性 | 8 工时 |
| 对培训材料进行最后处理 | 24 工时 |
| 制定培训机制 | 16 工时 |
| **技术联络人员** | **176 工时** |
| 制定“帮助”规范 | 8 工时 |
| 开发“帮助”系统 | 40 工时 |
| 审阅“帮助”文档 | 24 工时 |
| 根据反馈修改“帮助”文档 | 16 工时 |
| 制定用户手册规范 | 16 工时 |
| 编写用户手册 | 40 工时 |
| 审阅所有的用户文档 | 16 工时 |
| 根据反馈修改用户文档 | 16 工时 |
| 部署小组 | 96 工时 |
| 安装/部署软件 | 8 工时 |
| 获得用户反馈 | 40 工时 |
| 评估测试信息 | 8 工时 |
| 确定最终部署策略 | 8 工时 |
| 确定部署方法 | 8 工时 |
| 获得部署所需资源 | 8 工时 |
| 培训技术支持人员 | 8 工时 |
| 部署软件 | 8 工时 |

表6-1 项目工期表

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **开始时间** | **完成时间** | **剩余工时** |
| 管理人员 | 2018年5月20日 | 2018年6月15日 | 44 工时 |
| 项目经理 | 2018年5月21日 | 2018年8月14日 | 92 工时 |
| 分析人员 | 2018年5月23日 | 2018年6月12日 | 76 工时 |
| 开发人员 | 2018年6月18日 | 2018年7月11日 | 200 工时 |
| 测试人员 | 2018年6月18日 | 2018年7月24日 | 120 工时 |
| 培训人员 | 2018年6月18日 | 2018年7月20日 | 152 工时 |
| 技术联络人员 | 2018年6月18日 | 2018年7月16日 | 176 工时 |
| 部署小组 | 2018年7月24日 | 2018年8月9日 | 96 工时 |

表6-1 项目单位工资表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 资源名称 | 类型 | 材料标签 | 缩写 | 组 | 最大单位 | 标准费率 | 加班费率 | 每次使用成本 | 成本累算 | 基准日历 |
| 管理人员 | 工时 |  | 管 |  | 100% | ￥50.00/工时 | ￥55.00/工时 | ￥0.00 | 按比例 | 标准 |
| 项目经理 | 工时 |  | 项 |  | 100% | ￥50.00/工时 | ￥55.00/工时 | ￥0.00 | 按比例 | 标准 |
| 分析人员 | 工时 |  | 分 |  | 100% | ￥40.00/工时 | ￥45.00/工时 | ￥0.00 | 按比例 | 标准 |
| 开发人员 | 工时 |  | 开 |  | 100% | ￥60.00/工时 | ￥65.00/工时 | ￥0.00 | 按比例 | 标准 |
| 测试人员 | 工时 |  | 测 |  | 100% | ￥50.00/工时 | ￥55.00/工时 | ￥0.00 | 按比例 | 标准 |
| 培训人员 | 工时 |  | 培 |  | 100% | ￥30.00/工时 | ￥35.00/工时 | ￥0.00 | 按比例 | 标准 |
| 技术联络人员 | 工时 |  | 技 |  | 100% | ￥30.00/工时 | ￥35.00/工时 | ￥0.00 | 按比例 | 标准 |
| 部署小组 | 工时 |  | 部 |  | 100% | ￥30.00/工时 | ￥35.00/工时 | ￥0.00 | 按比例 | 标准 |

图6-1 项目工时图

图6-1 项目预算

表6-1 项目预算表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目名称 |  | 基准金额 | 预算金额 |  |  | 剩余金额 |
| 项目范围规划 | ￥0.00 | ￥0.00 | ￥1,400.00 | ￥1,400.00 | ￥0.00 | ￥1,400.00 |
| 分析/软件需求 | ￥0.00 | ￥0.00 | ￥3,640.00 | ￥3,640.00 | ￥0.00 | ￥3,640.00 |
| 设计 | ￥0.00 | ￥0.00 | ￥3,200.00 | ￥3,200.00 | ￥0.00 | ￥3,200.00 |
| 开发 | ￥0.00 | ￥0.00 | ￥12,000.00 | ￥12,000.00 | ￥0.00 | ￥12,000.00 |
| 测试 | ￥0.00 | ￥0.00 | ￥6,000.00 | ￥6,000.00 | ￥0.00 | ￥6,000.00 |
| 培训 | ￥0.00 | ￥0.00 | ￥4,560.00 | ￥4,560.00 | ￥0.00 | ￥4,560.00 |
| 文档 | ￥0.00 | ￥0.00 | ￥5,280.00 | ￥5,280.00 | ￥0.00 | ￥5,280.00 |
| 试生产 | ￥0.00 | ￥0.00 | ￥2,080.00 | ￥2,080.00 | ￥0.00 | ￥2,080.00 |
| 部署 | ￥0.00 | ￥0.00 | ￥1,200.00 | ￥1,200.00 | ￥0.00 | ￥1,200.00 |
|  |  |  |  |  |  |  |

## 6.4 项目版本迭代情况

### 6.3.1 代码迭代情况

采用git/github进行软件代码管理

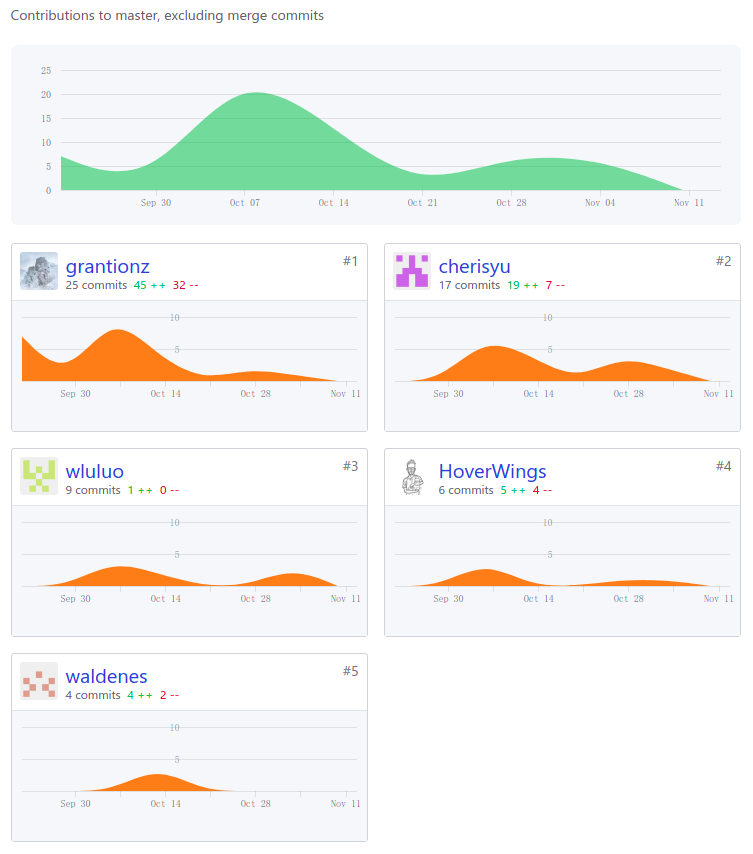


图6-1 协同开发情况图

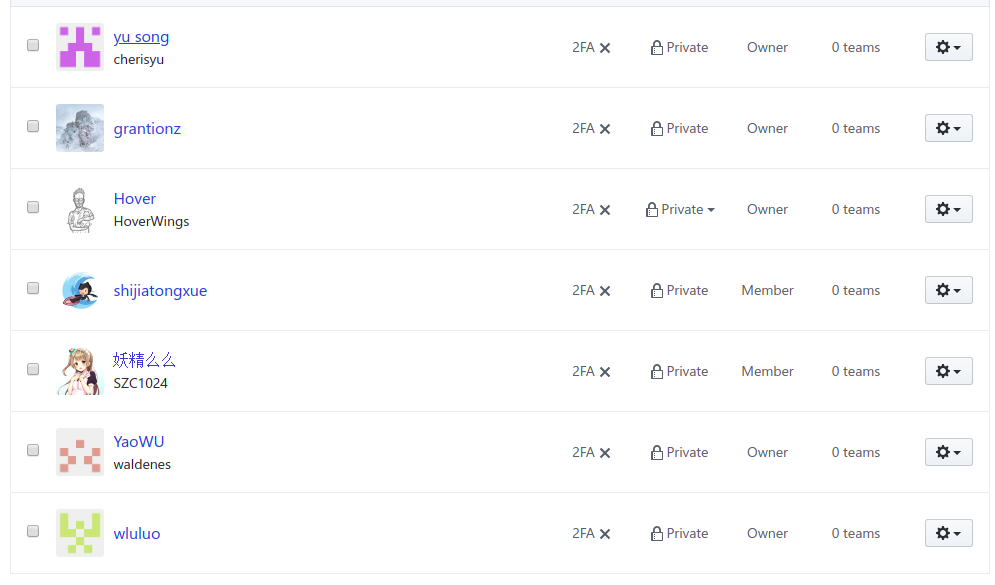


图6-1 代码权限管理

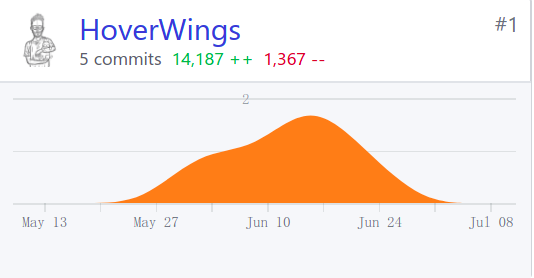


图6-1 代码迭代情况图

### 6.3.2 文档迭代情况

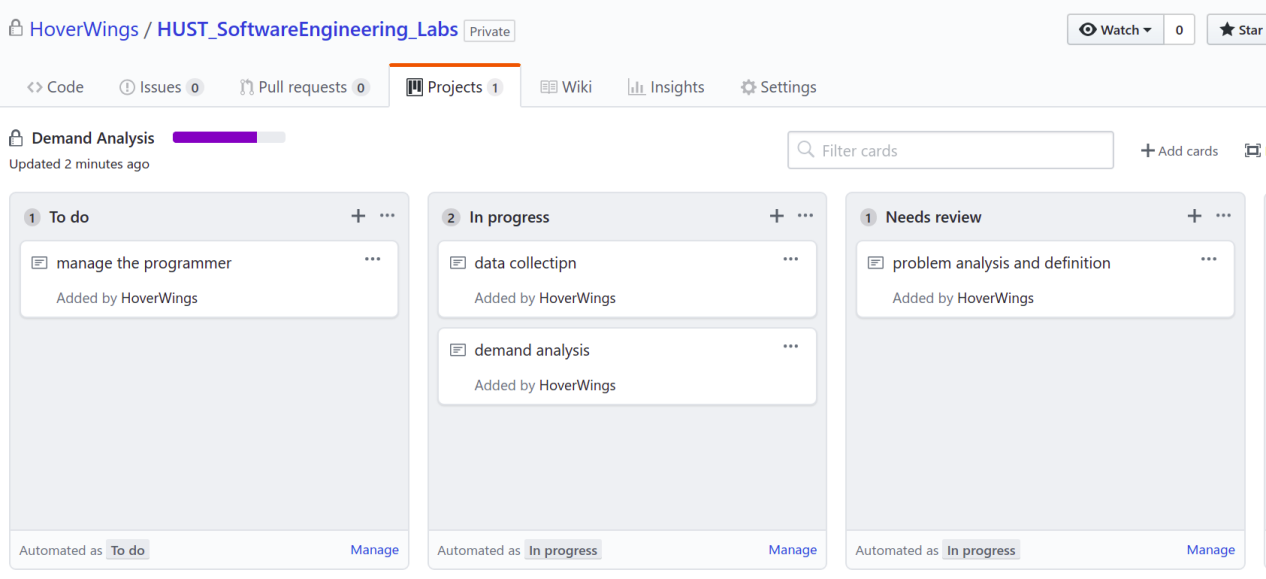


图6-1 文档迭代情况图

## 6.5 资源使用情况

## 6.6 项目交付情况

项目最终完成交付

交付件：

1. 软件源码
2. 软件Release(Linux )
3. 软件Release(Windows)
4. 软件使用说明
5. 软件测试文档

# 7 课程总结

在学习过程中，设计模式是软件工程中很重要的一个部分，尝试使用合适的设计模式对软件进行设计，可以使代码结构清晰，易于维护，并在今后的软件编写过程中尽可能合理使用设计模式和软件架构，进行软件开发和工程管理。

通常之前使用的都是较为简单的工程管理软件，而对于所开发的项目4~6人左右已经足够，但对于软件工程的学习，尝试使用不同的软件工程建模和管理工具。

# 参考文献

[1] QT官方文档

[2] CSDN[基于Qt的图表库](http://blog.csdn.net/liang19890820/article/details/54600307)

http://blog.csdn.net/liang19890820/article/details/54600307

# 附录