



# 华中科技大学

## 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

<b>1 选题</b>	<b>1</b>
1.1 教材	1
1.2 作业选题背景知识	1
1.3 辅助工具软件	2
1.3.1 需求分析	2
1.3.2 工程管理	2
1.3.3 软件建模	2
1.3.4 版本管理	2
1.3.5 自动化测试工具	2
1.4 编程语言和集成调试环境	2
<b>2 问题定义及可行性分析</b>	<b>3</b>
2.1 界面操作形式	3
2.2 预期达到功能	3
2.3 使用者角色描述(用户画像)	3
2.3.1 常规使用人员	3
2.3.2 数据统计人员	3
2.3.3 软件维护人员	3
2.4 系统环境描述	4
2.4.1 硬件环境	4
2.4.2 软件环境	4
2.5 系统的性能需求	4
2.5.1 数据库任务/交易分布表	5
2.5.2 交易混合表	6
2.5.3 数据库性能指标	6
2.6 可行性分析	6
<b>3 需求分析</b>	<b>7</b>
3.1 功能定义	7
3.1.1 模块功能描述	7
3.1.2 模块优先级评价	7
3.2 角色定义	7
3.3 设计系统的逻辑关系	8
3.4 系统接口	9
3.4.1 图形接口	9
3.4.2 软件接口	10

3.4.3 通讯接口.....	10
3.5 结构化分析.....	11
3.5.1 数据库逻辑结构设计.....	11
3.5.2 数据字典表格.....	12
3.5.3 数据流图.....	16
3.5.4 E-R 图.....	16
3.5.5 系统状态转换图.....	17
3.6 系统性能.....	17
<b>4 概要设计与详细设计.....</b>	<b>18</b>
4.1 系统功能模块划分.....	18
4.2 系统架构方案.....	18
4.3 系统操作流程.....	19
4.3.1 常规使用人员操作流程.....	19
4.3.2 数据统计人员操作流程.....	20
4.3.3 软件维护人员操作流程.....	21
4.4 前端实现方案.....	22
4.5 后端实现方案.....	22
4.5.1 数据库事务管理.....	23
4.5.2 历史数据管理.....	23
4.6 部件详细设计.....	23
4.6.1 航班事务管理.....	23
4.6.2 订单事务管理.....	25
4.6.3 权限系统.....	27
4.6.4 报表系统.....	27
4.6.5 通用数据显示模块.....	27
4.7 设计模式.....	29
4.8 注释及命名规范.....	30
<b>5 测试报告.....</b>	<b>31</b>
5.1 测试环境.....	31
5.2 软件文档测试.....	31
5.3 源代码一致性检查.....	32
5.3.2 代码规范检查.....	32
5.4 界面(外观)测试.....	1
5.5 功能测试.....	1
5.5.1 航班事务测试.....	1
5.5.2 订单事务测试.....	4

5.6.3 报表测试.....	8
5.6 稳定性测试.....	9
5.7 性能测试.....	9
5.8 恢复测试.....	9
5.8 安全测试.....	9
<b>6 项目管理.....</b>	<b>10</b>
6.1 项目计划.....	10
6.1.1 项目整体实施流程.....	10
6.1.2 项目进度计划.....	11
6.2 项目所需人员.....	16
6.3 项目功能点估算.....	16
6.3.1 功能点定义.....	16
6.3.2 功能点计算.....	17
6.3.3 功能点调整.....	1
6.4 项目成本估算.....	2
6.4 项目版本迭代情况.....	2
6.3.1 代码迭代情况.....	2
6.3.2 文档迭代情况.....	4
6.5 项目交付情况.....	4
<b>7 课程总结.....</b>	<b>5</b>
<b>参考文献.....</b>	<b>6</b>

## 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：
  - a) 不同航空公司的航班编号格式
  - b) 航班预订流程
  - c) 航班退票流程
  - d) 航班选座流程
  - e) 联程航空的查询及购买
  - f) 航班管理系统不同的模块
  - g) 航空公司对于用户信息的分析处理
- 3) 了解途径
  - a) 自身航班工具的使用经理
  - b) 不同 APP 和网站的体验使用
  - c) 已有航班管理系统的调研
  - d) 航班管理系统的相关设计文档和论文
  - e) 对于能够接触到的航班工作人员的调查访问
- 4) 需求分析
  - a) 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：
    - i. 航班选座
    - ii. 订单确认
  - b) 用户行为的时间约束：
    - i. 旅客在飞机起飞前一天凭取票通知交款取票
    - ii. 用户检票时间约束
  - c) 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
    - i. 模糊查询：利用时间，地点，价格等条件查询航班
    - ii. 精确查询：利用航班号查询
  - d) 航班预订：进行航班预订的相关操作
    - i. 座位选取
    - ii. 订单生成和确认

- iii. 与订单管理模块， 账单管理模块交互
- e) 订单管理： 进行订单的相关操作
  - i. 管理订单状态， 已经完成的订单和约束订单无法操作
  - ii. 更新订单状态
- f) 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, 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) 每个航班的座位信息的输入；
- 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 REG ECC

Disk: SSD 512G

Structure: 2U 机架式

Net: 百兆网卡+百兆局域网

#### 2) 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

#### 2) 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

#### 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
  - a) 模糊查询：利用时间，地点，价格等条件查询航班
  - b) 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
  - a) 座位选取
  - b) 订单生成和确认
  - c) 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
  - a) 管理订单状态，已经完成的订单和约束订单无法操作
  - b) 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

角色 \ 功能	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			√

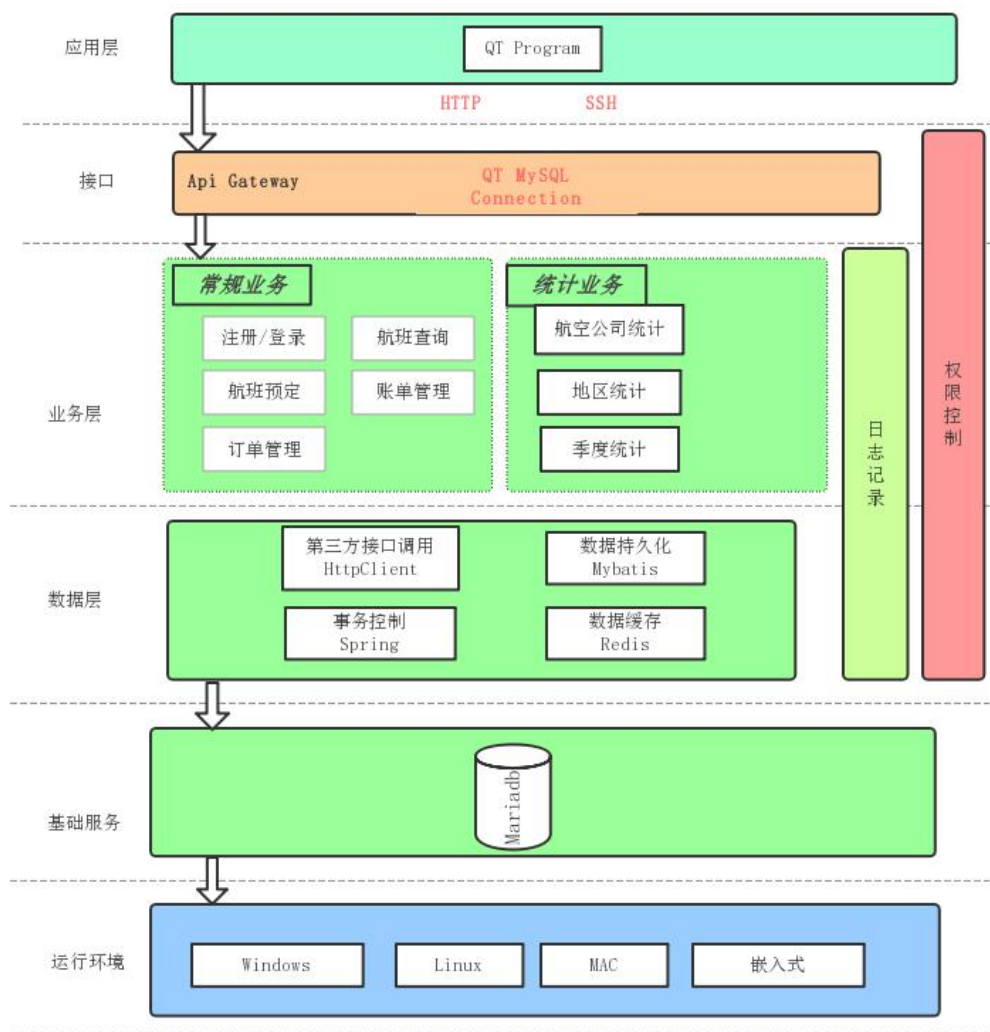


图 3-1 系统整体架构图

### 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：
  - a) 默认配色
  - b) 夜间模式
- 2) 采用类浏览器 tab 的风格进行布局
  - a) 标题栏
  - b) 主体窗口进行选择
  - c) 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

- a) 弹窗标题为操作结果
- b) 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
  - a) 时间格式: 12h 格式 hh::mm AM/PM
  - b) 日起格式: mm/dd/yy
- 6) 字体:
  - a) Cantarell
  - b) 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:
  - a) 输入: 鼠标键盘
  - b) 输出: 屏幕
  - c) 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
- 2) 客户端服务器交互:

QSqlDatabase 类进行数据库访问

为同步通讯机制

3) 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
  - a) 用户名 UID
  - b) 用户姓名 UNAME
  - c) 密码 PASSWORD
  - d) 用户权限
2. 航班信息表: 用于航班信息的管理
  - a) 航班号 FID
  - b) 航空公司 Flight
  - c) 飞机机型 FMODEL
  - d) 航班状态: FSTATE
    - i. 已降落 landed
    - ii. 延误 delayed
    - iii. 登机口 gate
    - iv. 登机 boarding
    - v. 登机手续办理 check-in
  - e) 起飞城市: FFROM
  - f) 飞往城市: FTO
  - g) 起飞日期: DATA
  - h) 起飞时间: TIME
  - i) 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
  - a) 订单号 OID
  - b) 航班号 FID
  - c) 用户号 UID
  - d) 座位号 SID
  - e) 订单时间 OTIME
  - f) 订单金额 OAMOUNT
  - g) 订单状态 OSTATE

4. 航班座位表
  - a) 航班号 FID
  - b) 座位号 SID
5. 航班模型表
  - a) 飞机型号 FMODEL
  - b) 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



## 2) 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)

3) 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)
BussinessClass Scale		varchar	商务舱规模 同上
EconomyClass Scale		varchar	经济舱规模 同上

4) 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

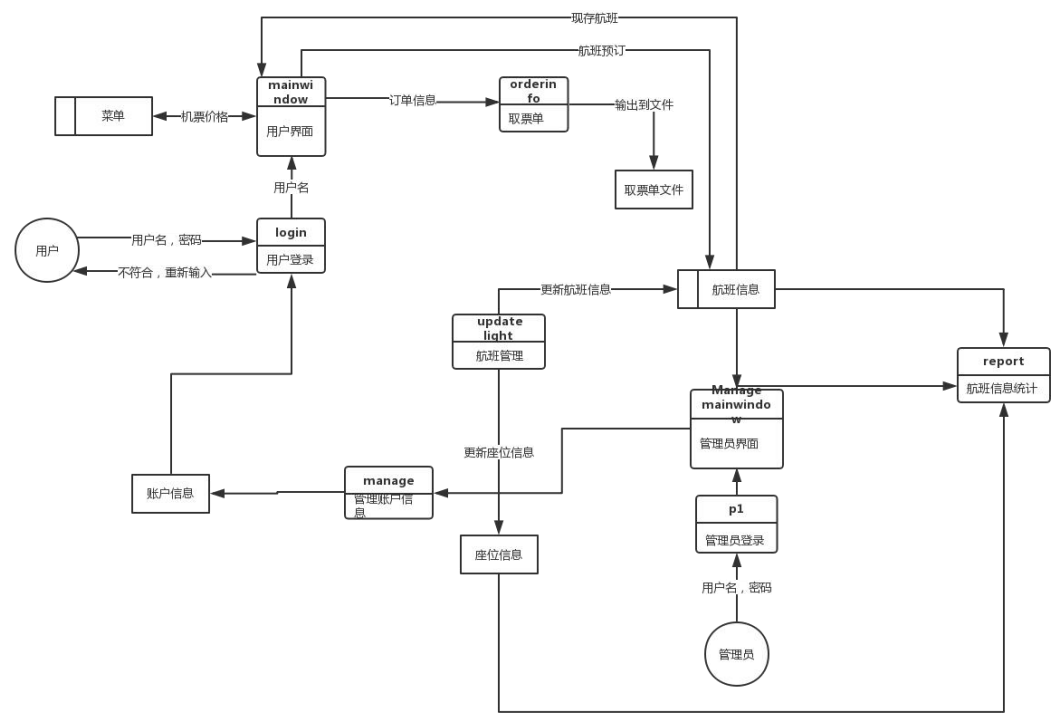


图 3-1 数据流图

3.5.4 E-R 图

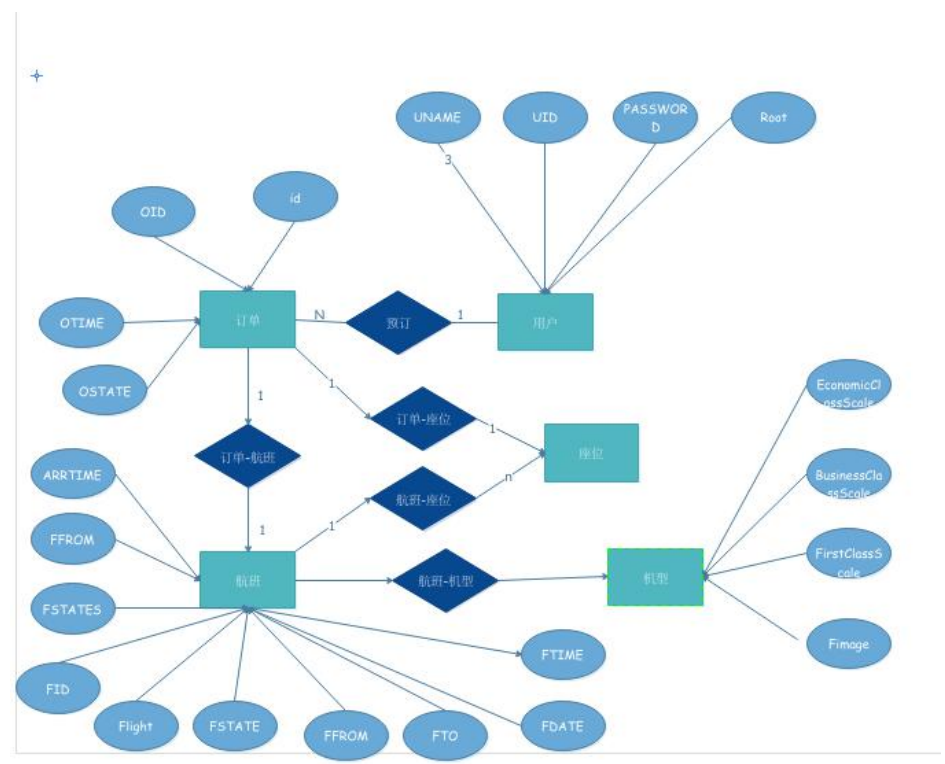


图 3-1 E-R 图

### 3.5.5 系统状态转换图

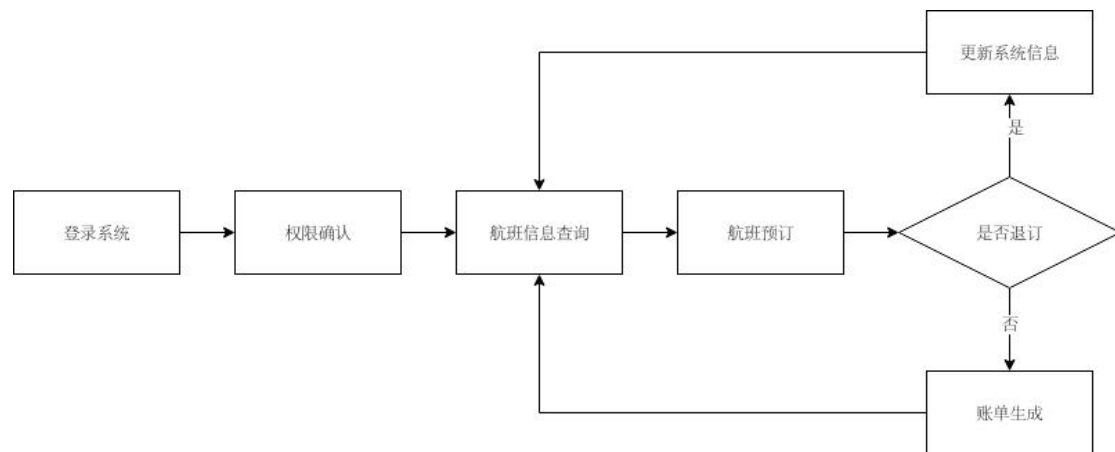


图 3-1 系统操作状态转换图

## 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

### 4.1 系统功能模块划分

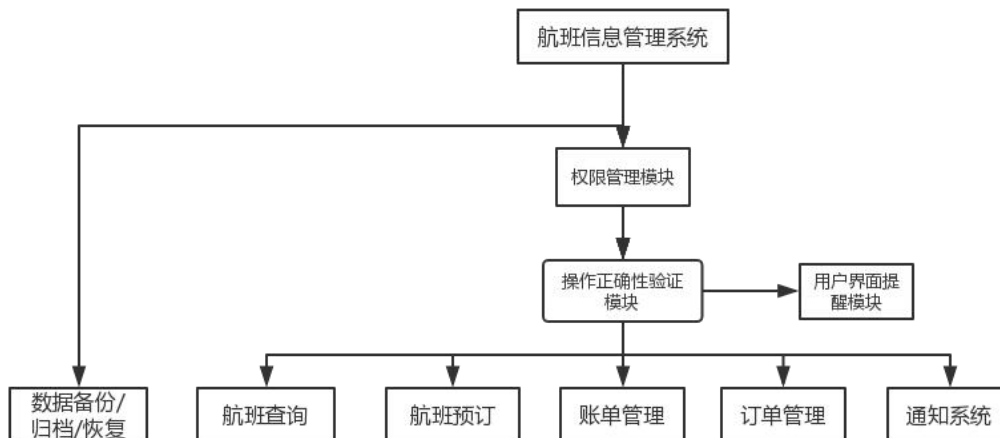


图 4-1 系统功能模块图

### 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

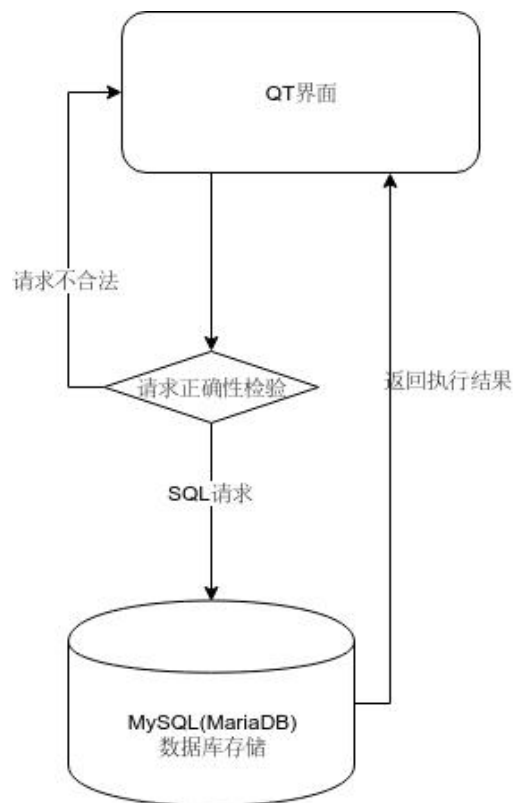


图 4-2 系统 C/S 架构图

## 4.3 系统操作流程

### 4.3.1 常规使用人员操作流程

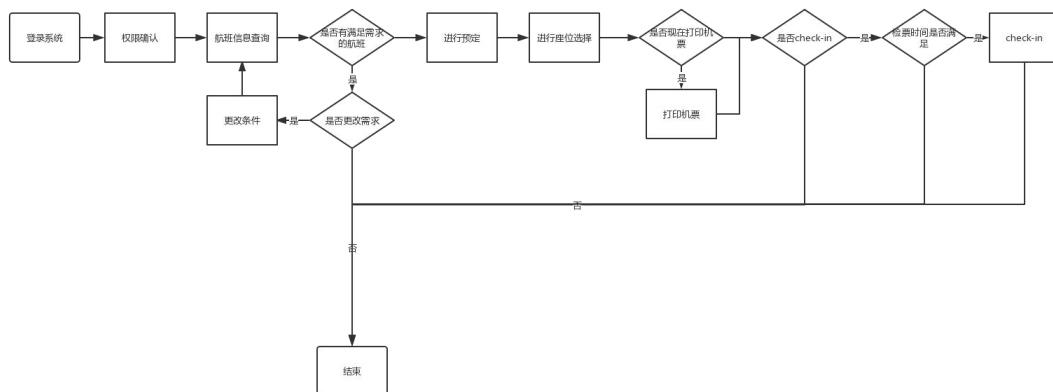


图 4-3 常规使用人员操作流程图

#### 4.3.2 数据统计人员操作流程

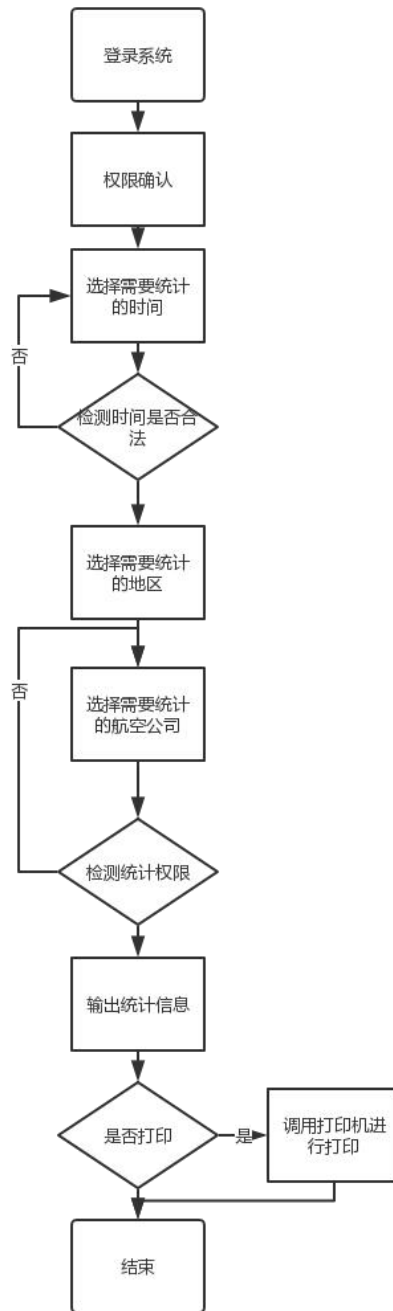


图 4-4 数据统计人员操作流程



### 4.3.3 软件维护人员操作流程

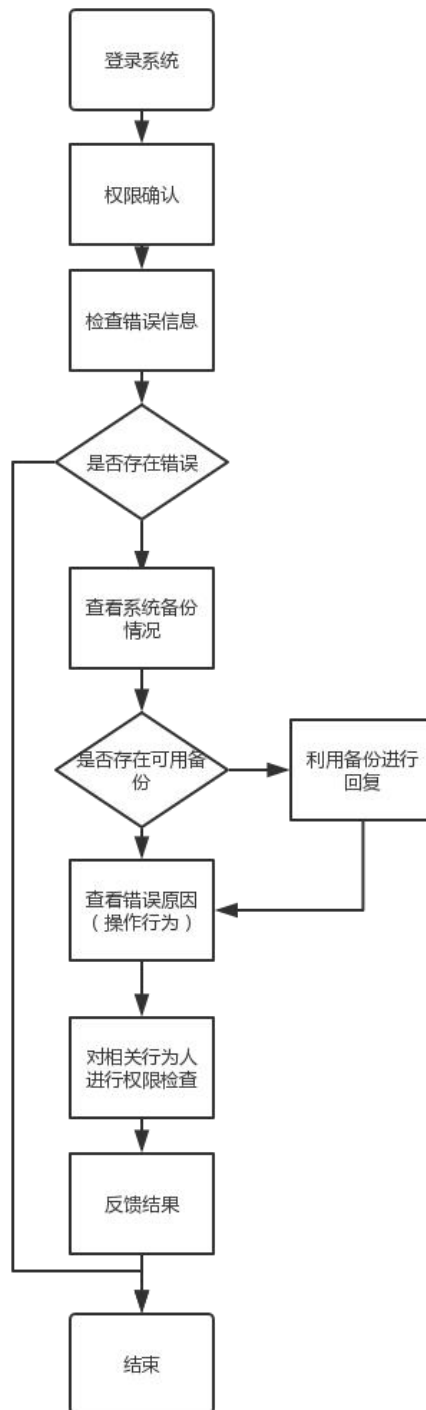


图 4-5 软件维护人员操作流程图

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。

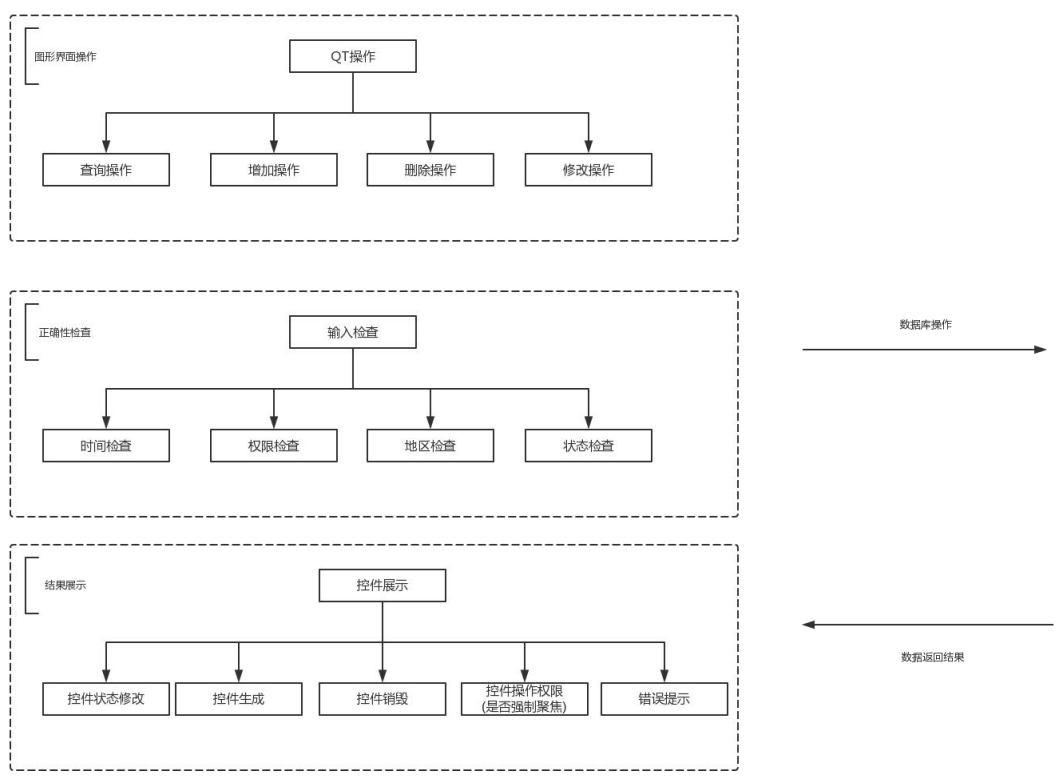


图 4-6 前端实现方案架构图

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。

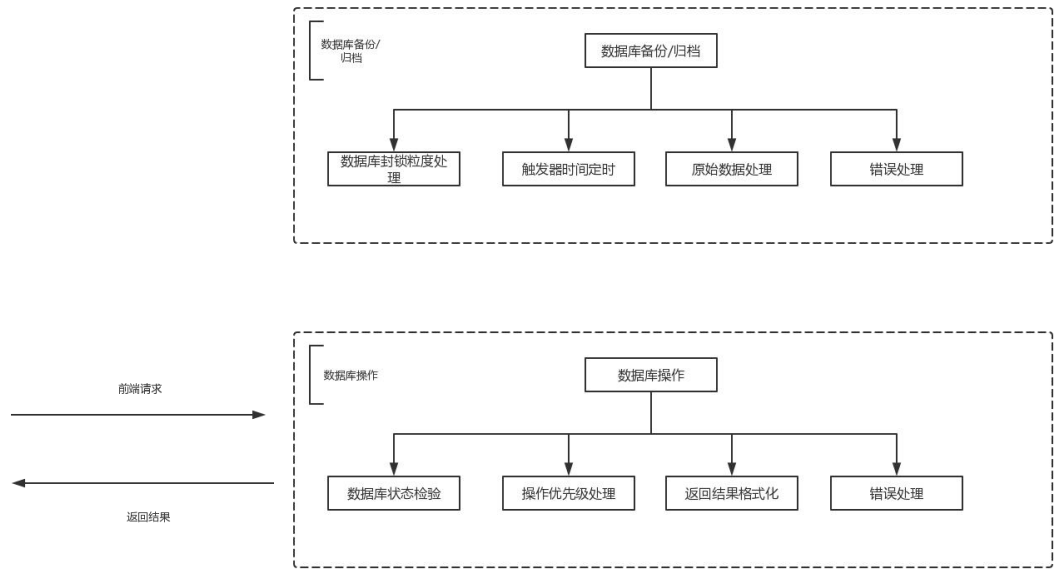


图 4-7 后端实现方案架构图

#### 4.5.1 数据库事务管理

数据库主要的事务管理主要为定时机制对数据进行备份处理和差错提醒。

#### 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。

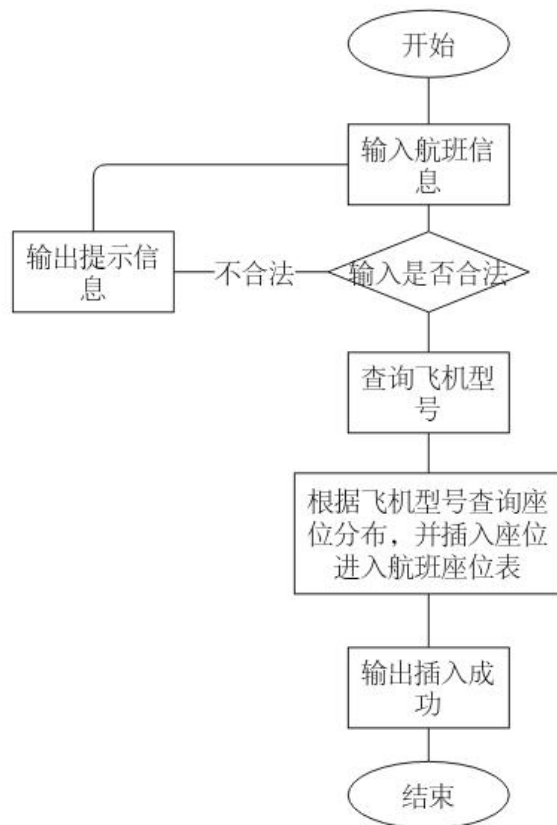


图 4-8 航班添加流程图

## (2) 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。

## (3) 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。

## (4) 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。

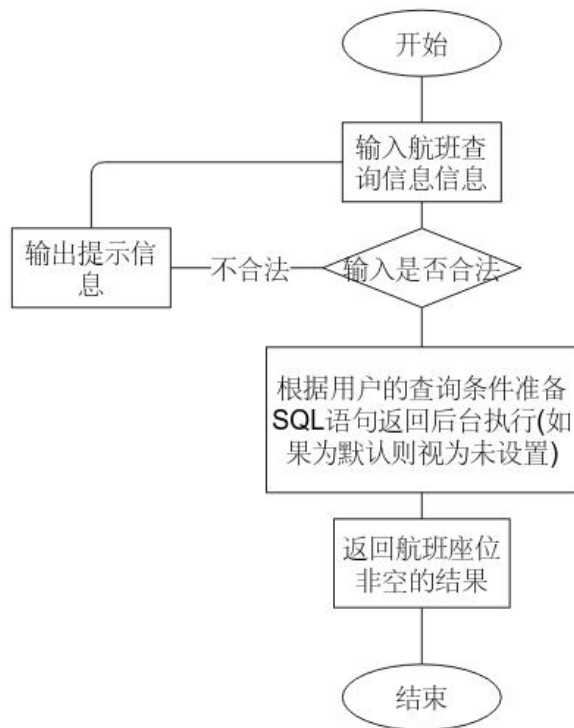


图 4-9 航班查询流程图

#### 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：

根据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。

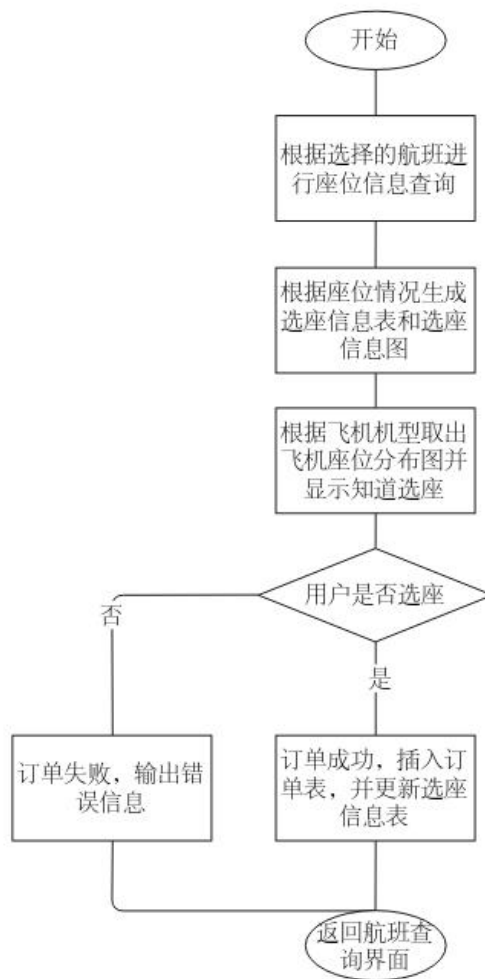


图 4-10 选座生成订单流程图

## 2) 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。

## 3) 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。

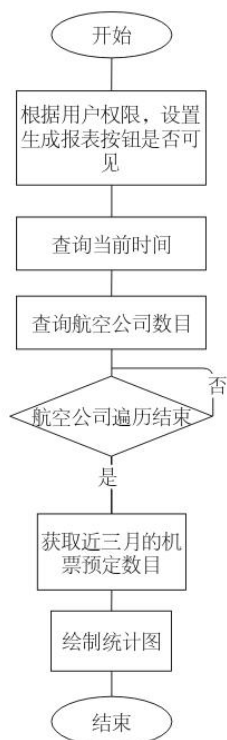


图 4-11 统计图生成模块流程图

### 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 `sqltable` 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

## 2) 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

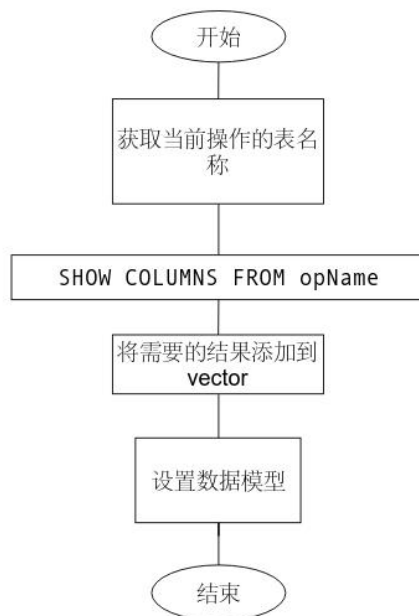


图 4-12 表头查询模块流程图

## 3) 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



#### 4) 用户切换窗口

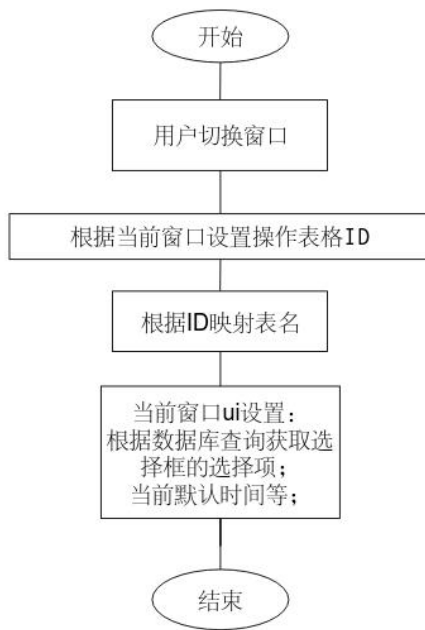


图 4-13 用户切换窗口后台操作流程

### 4.7 设计模式

整体采用 C/S 架构，对于数据库连接操作采用单例模式进行，能够维持连接同时防止反复连接，整体的架构 C/S 分离，前端进行差错检测。

对于控件的使用，在不同的控件之间做到尽量统一，以达到控件的可复用，避免频繁的申请和销毁操作，维护控件池(内存池机制)。

## 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

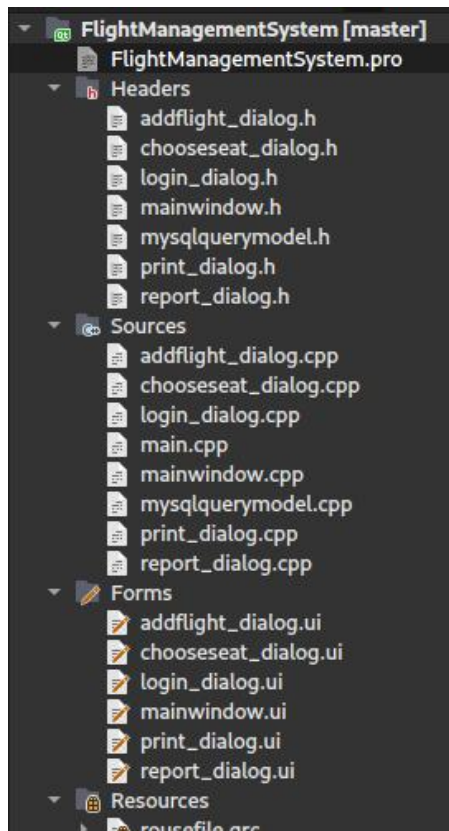


图 4-14 整体文件架构图

- |                     |                    |
|---------------------|--------------------|
| ➤ 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

编程语言: C++  
 IDE: Qt Creator  
 编译器: 8.1.1 20180531 (GCC)  
 数据库: 10.1.33-MariaDB MariaDB Server  
 OS: Manjaro 18.0.0 Illyria  
 Kernel: x86\_64 Linux 4.19.1-1-MANJARO  
 Shell: zsh 5.6.2  
 DE: GNOME  
 WM: GNOME Shell  
 Font: Noto Sans 11(System Font)  
 CPU: Intel Core i7-6700HQ @ 8x 3.5GHz  
 GPU: GeForce GTX 965M  
 RAM: 7865MiB

### 5.2 软件文档测试

表 5-1 软件文档检查表

文档内容	是否提供
项目实施计划	√
详细技术方案	√
软件需求规格说明书 (STP)(含数据字典)	√
概要设计说明书 (PDD)	√
详细设计说明书 (DDD)(含数据库设计说明书)	√
软件测试计划(STP)	√
软件测试报告(STR)	√
源程序(SCL)(不可修改的电子文档)	√
项目实施计划(PIP)	√
项目开发总结(PDS)	√

软件质量保证计划 (SQAP)	√
--------------------	---

5.3 源代码一致性检查

5.3.1 代码分析工具

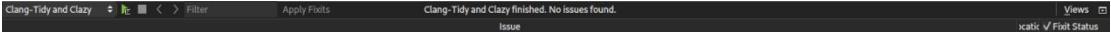


图 5-1 QT 代码分析工具无报错

5.3.2 代码规范检查

表 5-2 代码规范检查表

代码类	命名规范检查	注释检查	接口检查	数据类型检查	限制性检查	备注
addflight_dialog	√	√	√	√	√	
chooseseat_dialog	√	√	√	√	√	
login_dialog	√	√	√	√	√	
Mainwindow	√	√	√	√	√	
Mysqlquerymodel		√	√	√	√	满足 QT 原生命名规范
print_dialog	√	√	√	√	√	
report_dialog	√	√	√	√	√	数据类型为满足显示控件需要存在强转(已测试/安全检测)
rousefile.qrc						样式文件

5.4 界面(外观)测试

表 5-3 界面外观检查表

文档内容	是否规范	备注
界面风格	√	
表现形式	√	
组件用法	√	因为原生组件无法满足需求，采用派生组件完后功能(已经过测试)
字体选择	√	
字号选择	√	未在程序中统一更改字号选项 需要源码级别修改
色彩搭配	√	
日期表现	√	
计时方法	√	
时间格式	√	
对齐方式	√	

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 5-2 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 5-3 添加航班主码错误提示图

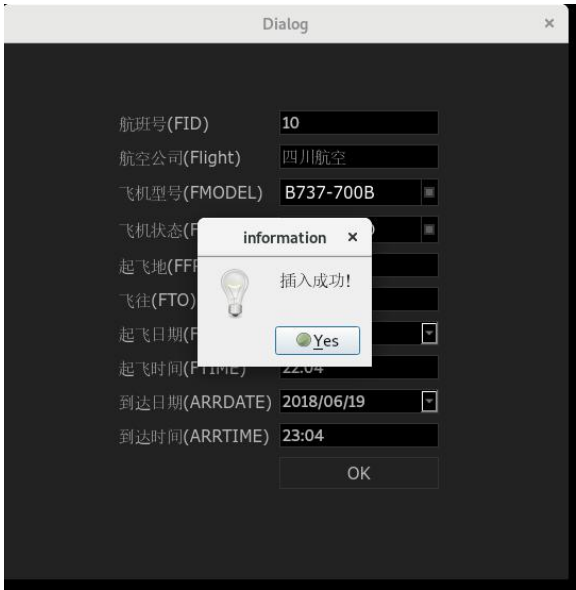


图 5-4 添加航班成功提示图

2) 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

数据库中进行。



图 5-5 表格修改航班图

3) 删除航班

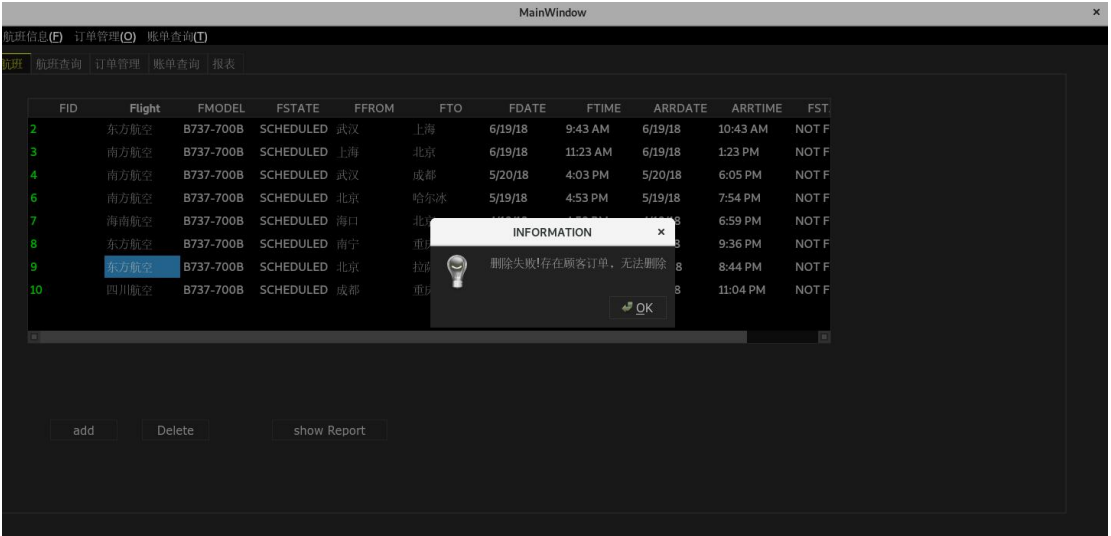


图 5-6 删除航班错误图

如果正常删除，同时删除座位



图 5-7 删除航班图



4) 航班查询



图 5-8 航班查询图

5.5.2 订单事务测试

1) 预订

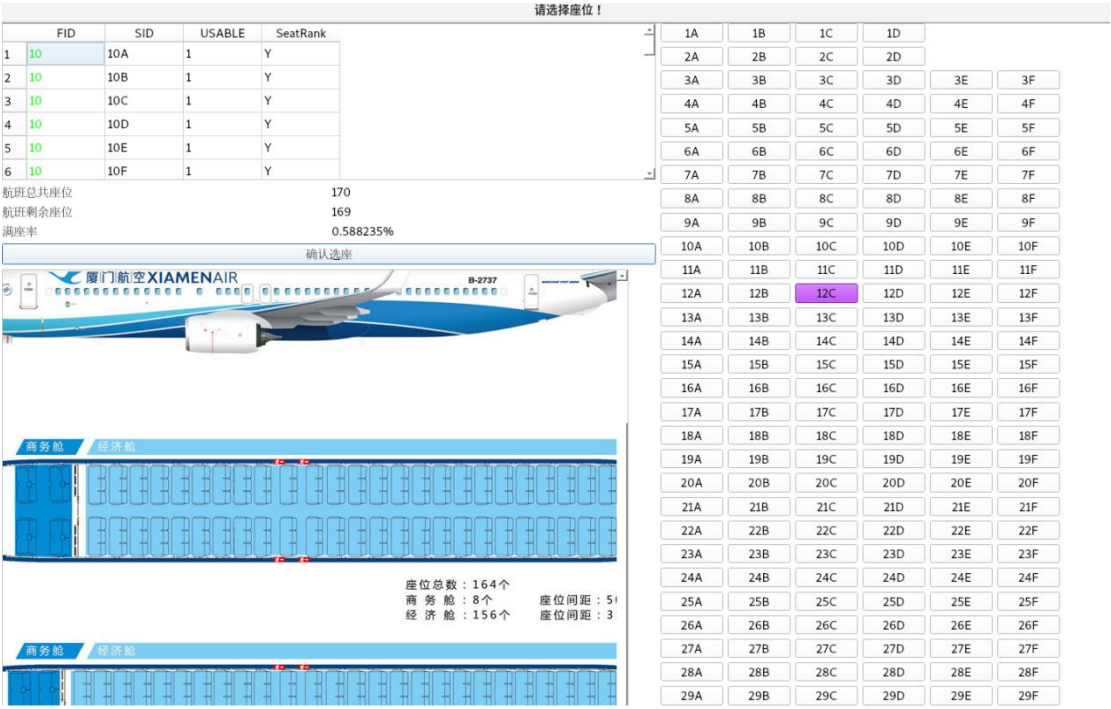


图 5-9 选座系统图

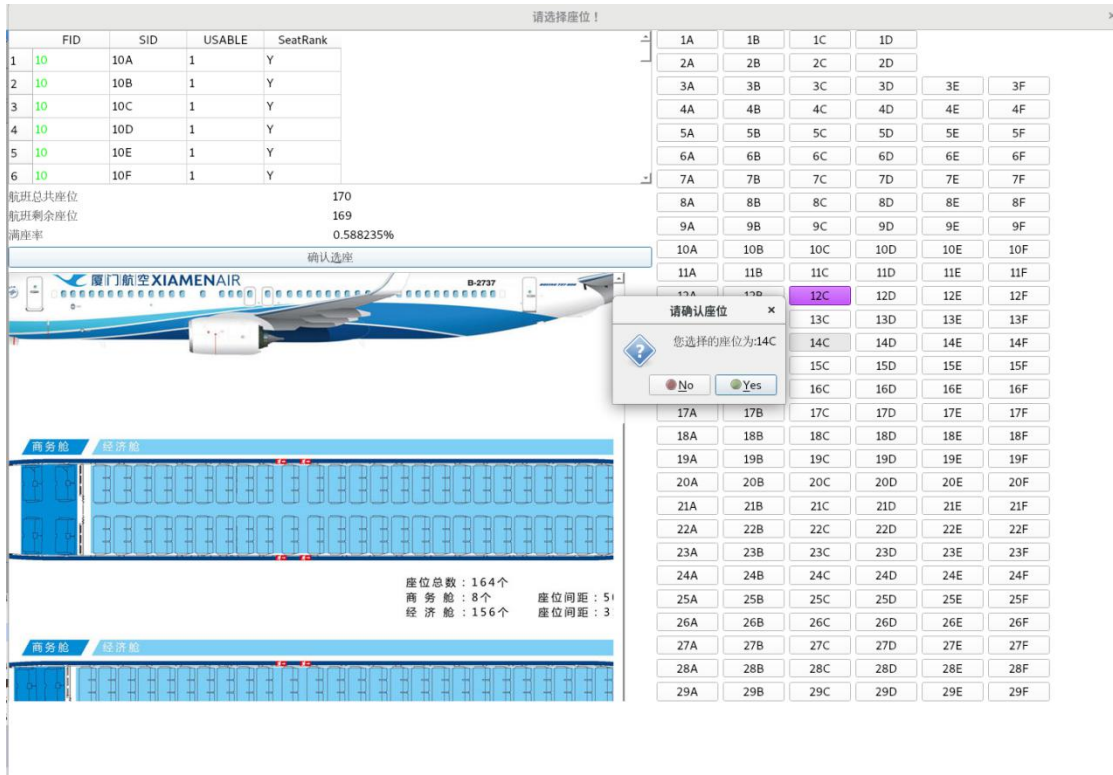


图 5-10 确认选座图

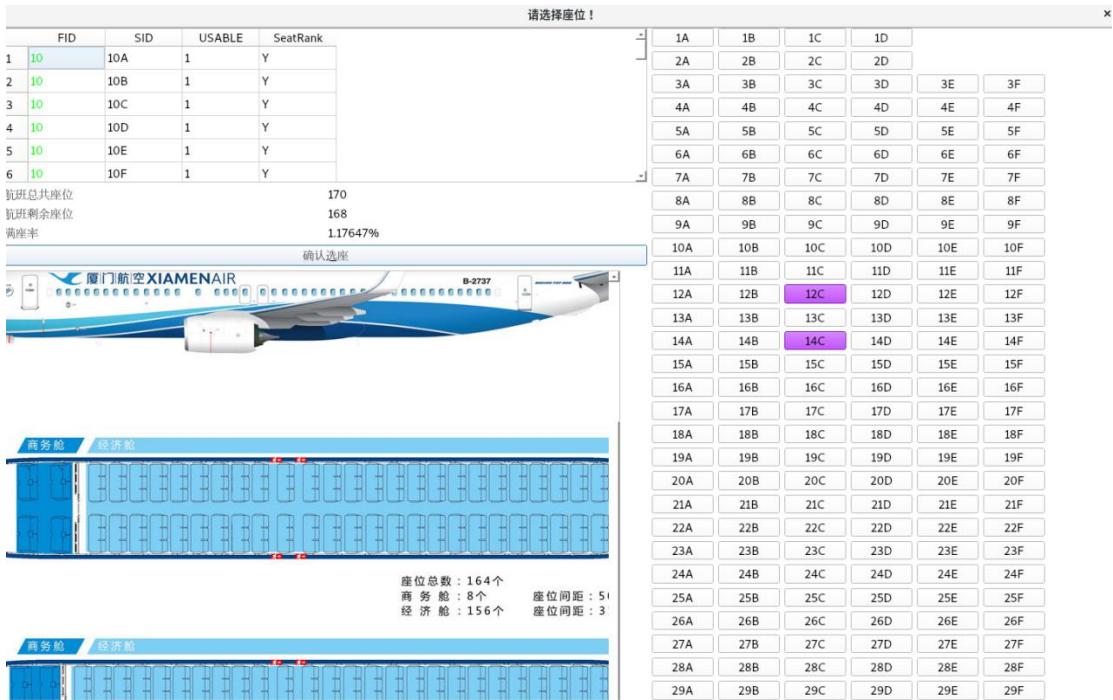


图 5-11 再次查询图

2) 退订

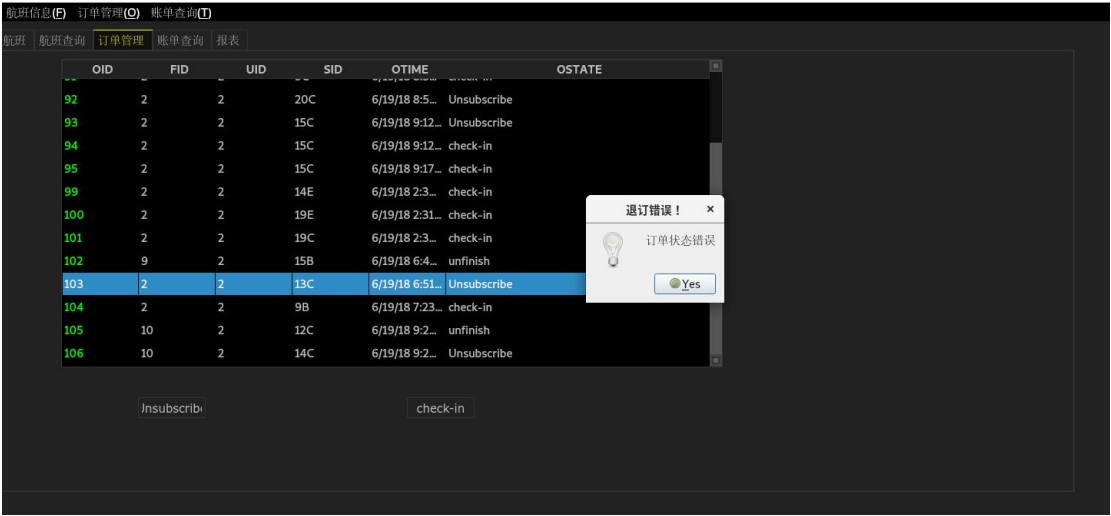


图 5-12 退订错误图

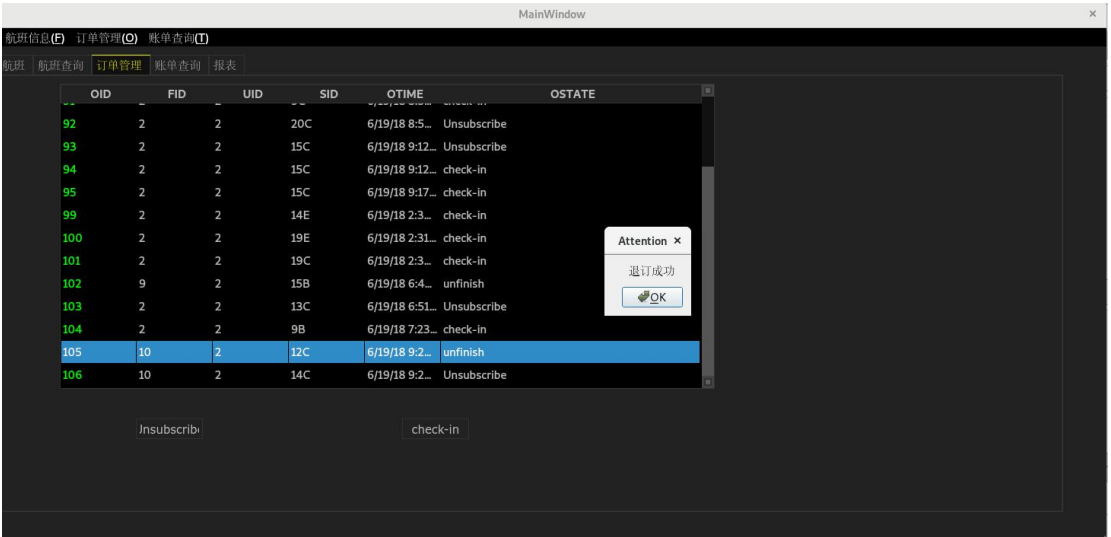


图 5-13 退订成功图

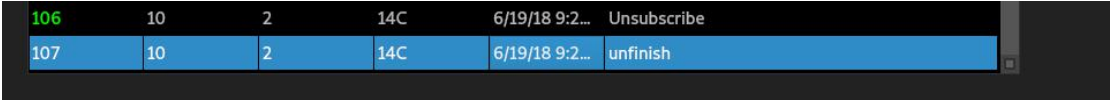


图 5-14 退订再次预订图

3) 取票并输出机票

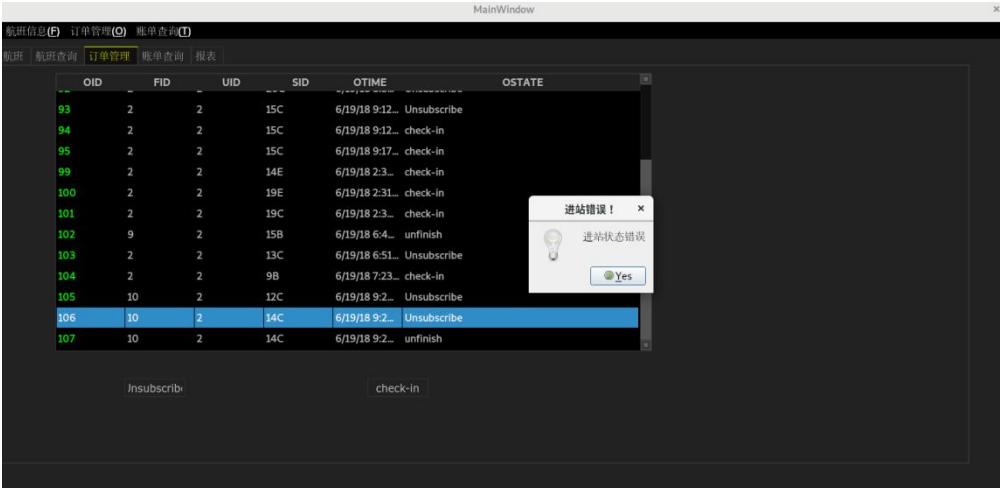


图 5-15 检票错误图



图 5-16 机票确认图



图 5-17 机票模拟打印图（输出到文件）

5.6.3 报表测试

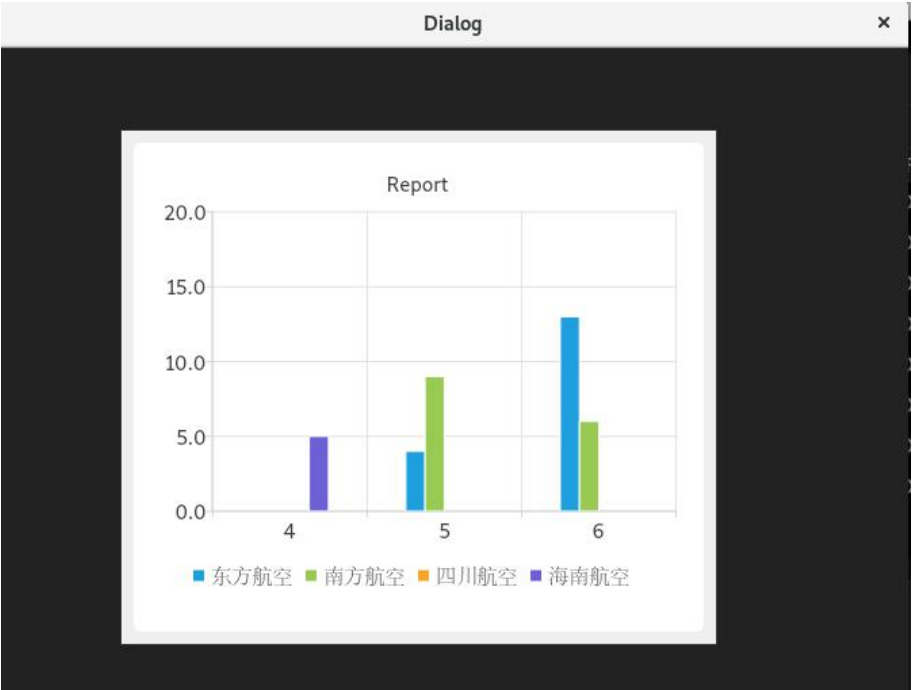


图 5-18 报表测试图

5.6 稳定性测试

表 5-4 稳定性测试表

测试内容	样例说明	测试样例	是否报错	报错信息	备注
边界测试	用户姓名	Qwertyuiop asdfghjkl;	√	用户姓名太长	
	用户密码	Qwertyuiop asdfghjkl;	√	用户密码太长	
	航公公司	西方航空啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊	√	航空公司名称太长	
	操作时间	时间不满足操作需求	√	操作提前	依赖系统时间
反复操作测试	反复预定机票			窗体限制预定	
权限测试	非管理员登录查询报表			无报表用户接口	

5.7 性能测试

Memory AVG: 20M  
I/O AVG: 1M  
Response Time: <1s

5.8 恢复测试

备份恢复测试通过，可以进行克隆安装。

5.8 安全测试

在更改数据库系统时间的情况下，可对数据库进行非法操作，造成数据库确认。  
软件启动前需要检查系统时间

## 6 项目管理

### 6.1 项目计划

#### 6.1.1 项目整体实施流程

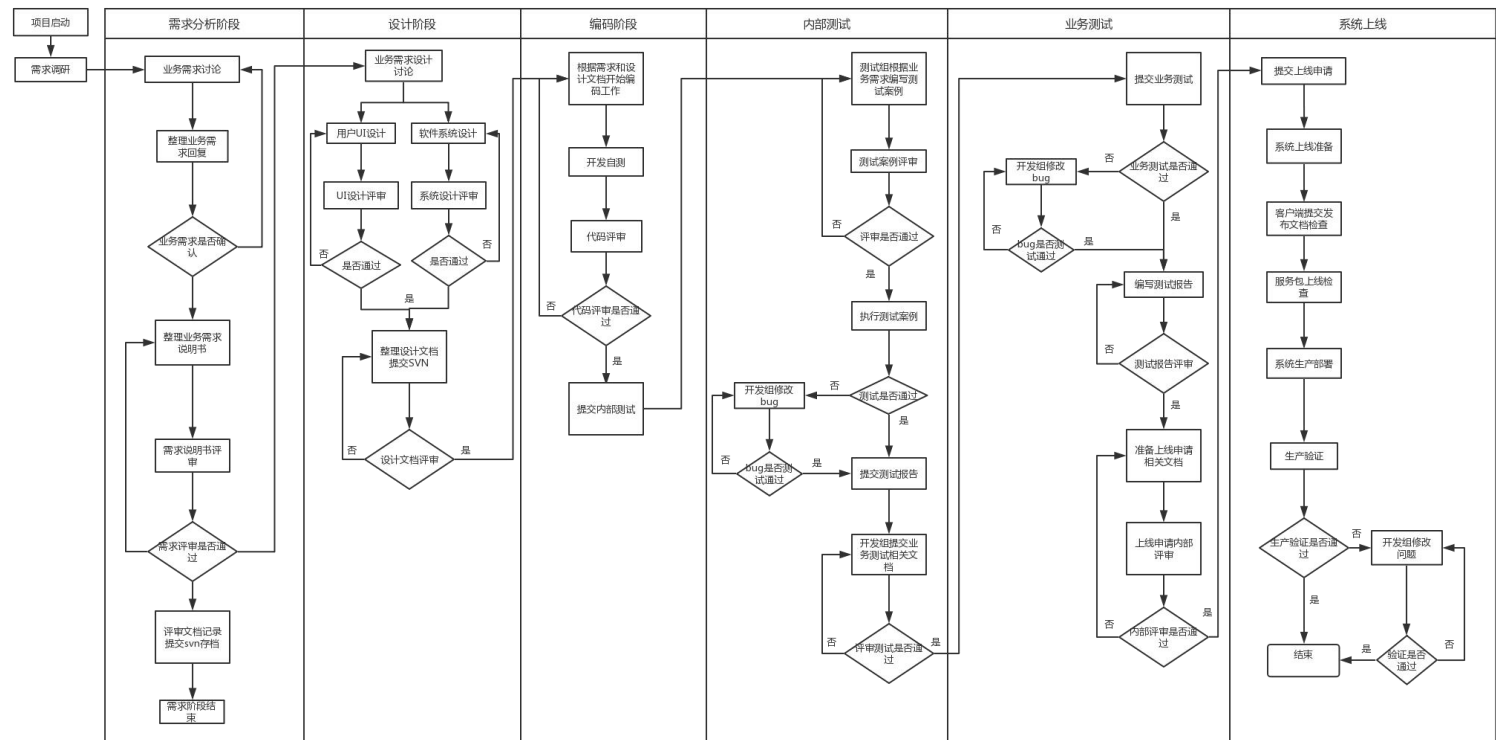


图 6-1 项目整体实施流程图

## 6.1.2 项目进度计划

采用 project 进行项目规划

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
<b>项目范围规划</b>	<b>3.5 个工作日</b>	<b>2018 年 5 月 20 日</b>	<b>2018 年 5 月 23 日</b>		
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	
<b>分析/软件需求</b>	<b>9 个工作日</b>	<b>2018 年 5 月 23 日</b>	<b>2018 年 6 月 5 日</b>		
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	
<b>设计</b>	<b>8.5 个工作日</b>	<b>2018 年 6 月 5 日</b>	<b>2018 年 6 月 15 日</b>		
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	
<b>开发</b>	<b>18 个工作日</b>	<b>2018 年 6 月 18 日</b>	<b>2018 年 7 月 11 日</b>		
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	
<b>测试</b>	<b>26.75 个工作日</b>	<b>2018 年 6 月 18 日</b>	<b>2018 年 7 月 24 日</b>		



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
<b>单元测试</b>	<b>5 个工作日</b>	<b>2018 年 7 月 5 日</b>	<b>2018 年 7 月 12 日</b>		
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	
<b>整体测试</b>	<b>8 个工作日</b>	<b>2018 年 7 月 12 日</b>	<b>2018 年 7 月 24 日</b>		
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	
<b>培训</b>	<b>24.75 个工作日</b>	<b>2018 年 6 月 18 日</b>	<b>2018 年 7 月 20 日</b>		
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	
<b>文档</b>	<b>20.5 个工作日</b>	<b>2018 年 6 月 18 日</b>	<b>2018 年 7 月 16 日</b>		
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	
<b>试生产</b>	<b>42. 25 个工作日</b>	<b>2018 年 6 月 5 日</b>	<b>2018 年 8 月 2 日</b>		
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	
<b>部署</b>	<b>5 个工作日</b>	<b>2018 年 8 月 2 日</b>	<b>2018 年 8 月 9 日</b>		
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	
<b>实施工作结束后的回顾</b>	<b>3 个工作日</b>	<b>2018 年 8 月 9 日</b>	<b>2018 年 8 月 14 日</b>		
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	

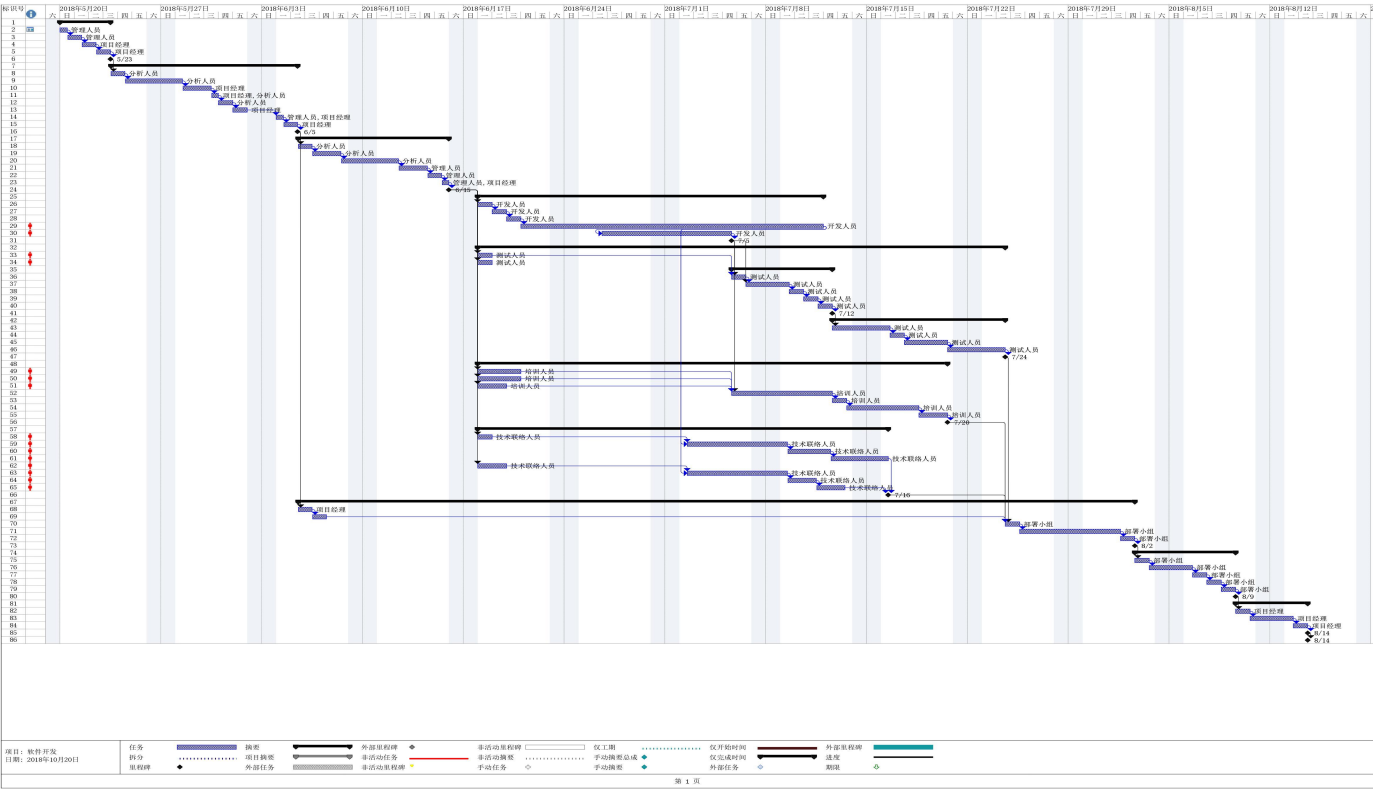


图 6-2 甘特图

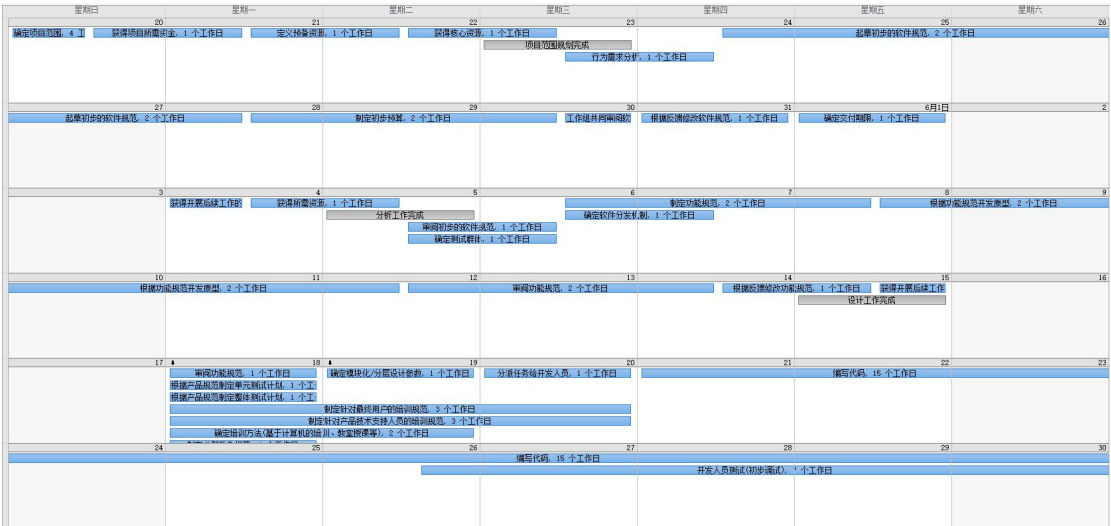


图 6-3 项目进度计划图 1

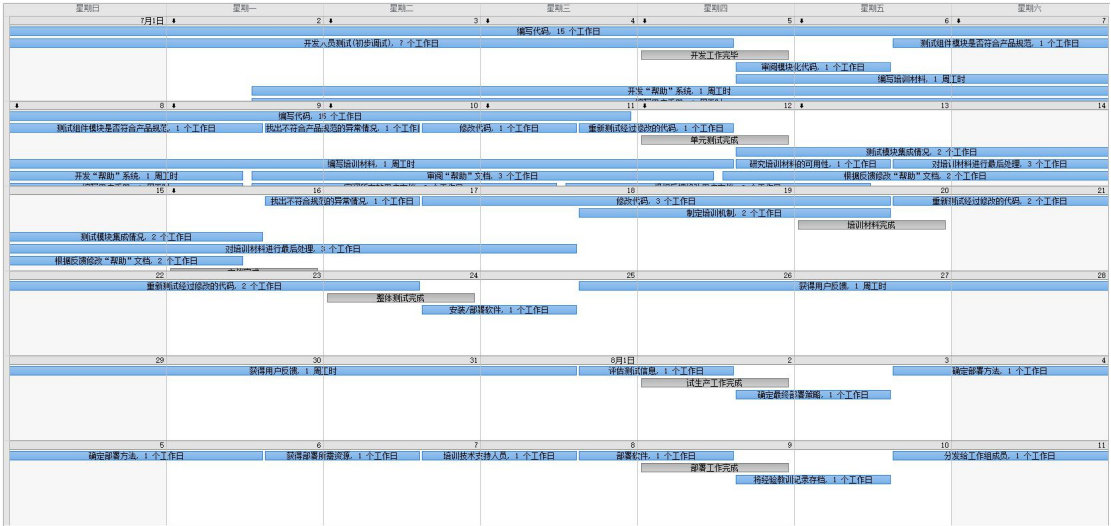


图 6-4 项目进度计划图 2



图 6-5 项目进度计划图 3

表 6-2 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

6.3.1 功能点定义

图 6-3 功能点定义表

功能点	详细内容
ILF	用户信息
	航班信息
	飞机机型信息
	航班座位信息
EIF	机票
	报表统计
EI	添加用户信息
	修改用户信息
	删除用户信息
	添加航班信息
	修改航班信息
	删除航班信息
	添加飞机型号
	修改飞机型号
	删除飞机型号
	添加订单
	修改订单
	删除订单
EQ	查询用户信息
	查询订单信息
	查询航班信息
EO	输出机票
	输出统计报表

### 6.3.2 功能点计算

表 6-4 ILF 功能点计算

ILF 内部逻辑文件	RET 个数	RET	DET 个数	DET	复杂度	未调整的 FP 个数
用户信息	1	用户账户信息	4	UID	低	7
				UNAME		
				PASSWORD		
				ROOT		
航班信息	4	航班基本信息 航班时间信息 飞机机型信息 航班座位信息	11	FID	中	10
				Flight		
				FMODEL		
				FSTATE		
				FFROM		
				FTO		
				FDATE		
				FTIME		
				ARRDATE		
				ARRTIME		
				FSTATUS		
飞机机型信息	2	飞机机型信息 机型图片信息	5	FMODEL	中	10
				Fimage		
				FirstClassScale		
				BussinessClassScale		
				EconomyClassScale		
航班座位信息	3	航班基本信息 飞机机型信息 航班时间信息	4	FID	中	10
				SID		
				USABLE		
				SeatRank		
合计：37 个						

表 6-5 EIF 功能点计算

ILF 内部逻辑文件	RET 个数	RET	DET 个数	DET	复杂度	未调整的 FP 个数
机票	4	用户 航班 机型 座位	4		低	5
报表统计	4	航空公司 订单数据 时间信息 地区信息	Nan	查询数据条数依据当前系统中的数据决定	高	10
合计：15 个						

表 6-6 EI 功能点数

EI	FTR 个数	FTR	DET 个数	复杂度	未调整的 FP 个数
添加用户信息	2	用户信息 用户报表信息	7	低	5
修改用户信息	2	用户信息 用户报表信息	7	高	10
删除用户信息	2	用户信息 用户报表信息	1	低	3
添加航班信息	3	航班 座位 机型	10	高	6
修改航班信息	3	航班 座位 机型	10	高	6
删除航班信息	3	航班 座位 机型	1	中	4
添加飞机型号	2	机型 座位	6	中	4
修改飞机型号	2	机型	6	中	4

		座位			
删除飞机型号	2	机型 座位	1	低	3
添加订单	4	用户 航班 机型 座位	8	高	6
修改订单	4	用户 航班 机型 座位	7	高	6
删除订单	4	用户 航班 机型 座位	2	中	4
合计：57 个					

表 6-7 EQ 功能点数

EQ	FTR 个数	FTR	DET 个数	复 杂 度	未调整的 FP 个数
查询用户信息	2	用户信息 用户报表信息	7	中	4
查询订单信息	3	航班 座位 机型	21	高	6
查询航班信息	3	航班 座位 机型	17	中	4
合计：14 个					



表 6-9 EO 功能点数

EO	FTR 个数	FTR	DET 个数	复 杂 度	未调整的 FP 个数
输出机票	4	用户 航班 机型 座位	7	高	6
输出统计报表	4	航空公司 订单数据 时间信息 地区信息	Nan	高	6
合计：12 个					

表 6-10 功能点统计

EO	FTR 个数
ILF	37
EIF	15
EI	57
EQ	14
EO	12
合计：135 个	

### 6.3.3 功能点调整

表 6-11 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

= (135) \* 0.98

= 132.3 个

## 6.4 项目成本估算

表 6-12 项目工时表

资源名称	工时
<b>管理人员</b>	<b>44 工时</b>
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
<b>项目经理</b>	<b>92 工时</b>
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
<b>分析人员</b>	<b>76 工时</b>
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
<b>开发人员</b>	<b>200 工时</b>
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
<b>测试人员</b>	<b>120 工时</b>
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
<b>培训人员</b>	<b>152 工时</b>
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
<b>技术联络人员</b>	<b>176 工时</b>
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
<b>部署小组</b>	<b>96 工时</b>
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 6-13 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 6-14 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

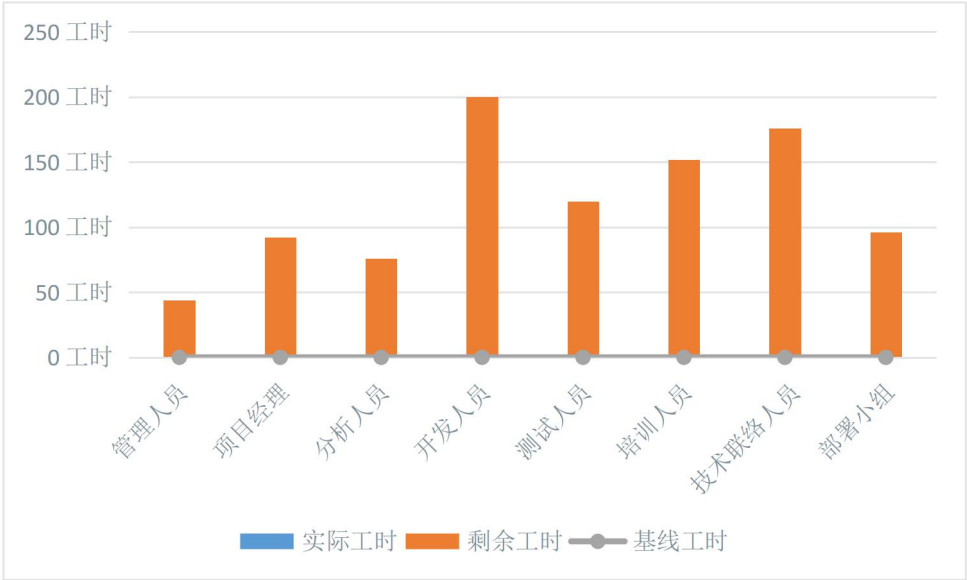


图 6-6 项目工时图

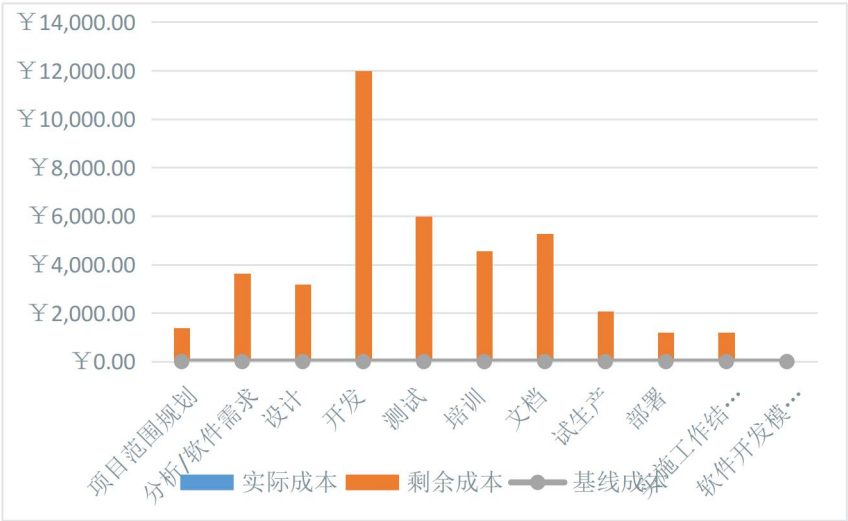


图 6-7 项目预算

表 6-15 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

Contributions to master, excluding merge commits

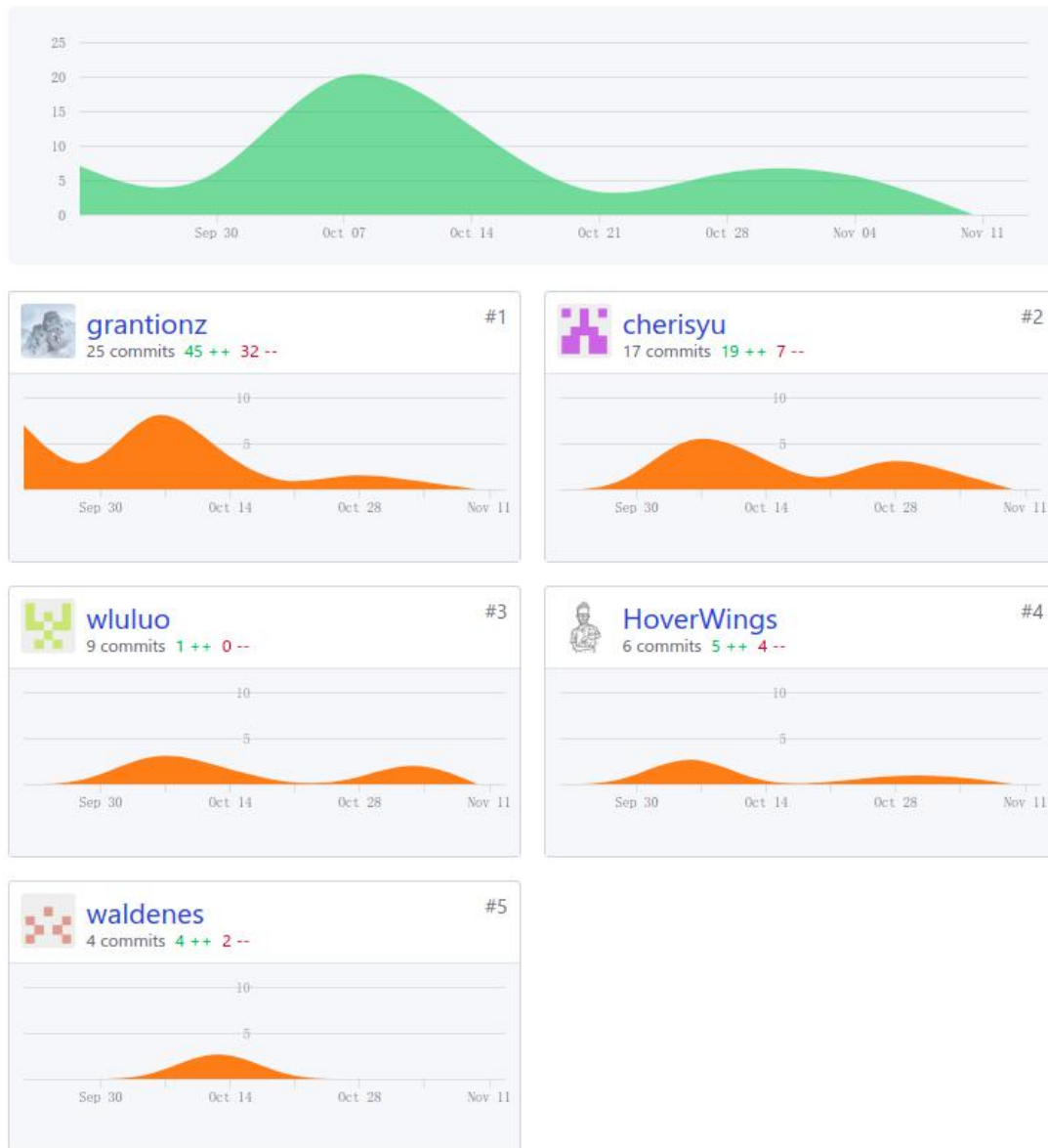


图 6-8 协同开发情况图































<input type="checkbox"/>	 yu song cherisyu	2FA 	 Private	Owner	0 teams	
<input type="checkbox"/>	 grantionz	2FA 	 Private	Owner	0 teams	
<input type="checkbox"/>	 Hover HoverWings	2FA 	 Private	Owner	0 teams	
<input type="checkbox"/>	 shijiatongxue	2FA 	 Private	Member	0 teams	
<input type="checkbox"/>	 妖精么么 SZC1024	2FA 	 Private	Member	0 teams	
<input type="checkbox"/>	 YaoWU waldenes	2FA 	 Private	Owner	0 teams	
<input type="checkbox"/>	 wluluo	2FA 	 Private	Owner	0 teams	

图 6-9 代码权限管理



图 6-10 代码迭代情况图

### 6.3.2 文档迭代情况

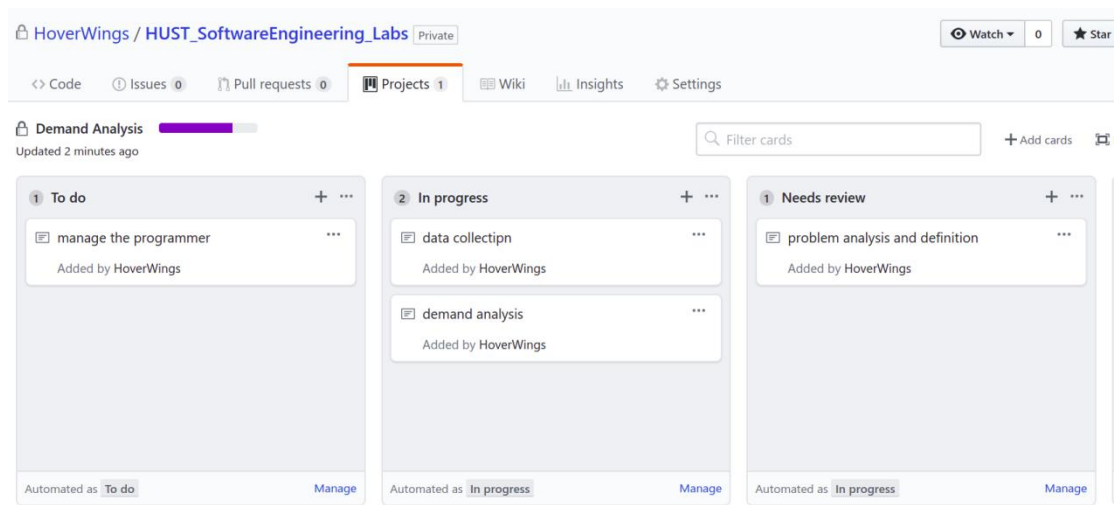


图 6-11 文档迭代情况图

### 6.5 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。

对于软件进行架构分析的时候，在设计时可以采用 UML 图等进行设计，在代码完成之后，可以采用逆向工程对代码进行分析，看是否最终满足当初的设计。

数据库设计阶段，对数据库功能、数据字典的设计，加深了对于关系模式、E-R 图、范式、数据流图等概念的理解。

数据库操纵类编写，基于 QT 信号槽的设置，设计过程中，将主要的数据库操作继承子 `qsqlquery` 基类进行重载，初期考虑进行使用 DAO 模型进行对象映射，但此时对于底层的操作可以不提供接口进行权限控制，实现更为简单，故使用 `mysqlmodel` 进行基本的数据操作类封装形成基本的 DAO 模型，并提供清晰的接口，上层只需要更新操纵表，即可以对当前表的相关属性进行自动设置。

后期测试，使用了不同的数据进行测试，且在模块编写中，对模块进行了单元测试，增强了鲁棒性，同时对文档规范和代码规范进行了检查。

整个过程中，熟悉了软件工程的基本流程，对于一些设计软件如 StarUML, visio, project 等有了一定程度掌握，同时阅读了关于 QT 的相关设计规范，对于接口(函数接口和连接接口)进行学习，便于软件部件的可复用和可拓展以及后期的维护。

多人开发方面，使用 git/github 进行软件迭代，对于 github 中的 project 管理和 code review 功能进行了探索，同时 github 听过了 task 管理，可以结合 project 做到代码的计划，对于每一次 commit 进行较为详细的注释，方便团队理解提交，同时可以在差错时方便回滚准确定位差错点。

工程实际上与真实的生产环境有一定的差距，且测试环境较为单一，希望能够有机会用较大的工程真实模拟工业生产环境，课程可考虑提供小组合作选题开发模式，用小型团队，较长周期开发，开发时间与理论课程基本同步，可以达到更好的学习效果，最后，感谢老师在课程和实验上的指导，同时感谢所参考资料的提供者。

## 参考文献

- [1] 张海藩, 吕云翔编著. 软件工程(第 4 版).北京: 人民邮电出版社, 2013 年 09 月.
- [2] Erich Gamma, Richard Helm, Ralph Johnson 著.设计模式: 可复用面向对象软件的基础.北京: 机械工业出版社.2007 年 1 月 1 日.
- [3] QT 官方文档
- [4] 功能点(Function Point FP)估算法  
<https://blog.csdn.net/wj3319/article/details/22668615>