

**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](#_Toc146018492)

[1.1 教材 1](#_Toc1169270020)

[1.2 作业选题背景知识 1](#_Toc377378890)

[1.3 辅助工具软件 2](#_Toc1277000567)

[1.3.1 需求分析 2](#_Toc210889652)

[1.3.2 工程管理 2](#_Toc750923367)

[1.3.3 软件建模 2](#_Toc1739758955)

[1.3.4 版本管理 2](#_Toc1180711624)

[1.3.5 自动化测试工具 2](#_Toc275269842)

[1.4 编程语言和集成调试环境 2](#_Toc815556450)

[1.5 想法 3](#_Toc2062273874)

[2 问题定义及可行性分析 4](#_Toc1295441733)

[2.1 界面操作形式 4](#_Toc656012091)

[2.2 预期达到功能 4](#_Toc1162253802)

[2.3 使用者角色描述(用户画像) 4](#_Toc2047016860)

[2.3.1 常规使用人员 4](#_Toc855891617)

[2.3.2 数据统计人员 4](#_Toc1474257332)

[2.3.3 软件维护人员 4](#_Toc1809506173)

[2.4 系统环境描述 5](#_Toc1220059923)

[2.4.1 硬件环境 5](#_Toc503290612)

[2.4.2 软件环境 5](#_Toc1815141672)

[2.5 系统的性能需求 5](#_Toc991027707)

[2.5.1 数据库任务/交易分布表 6](#_Toc1737592142)

[2.5.2 交易混合表 7](#_Toc1313137153)

[2.5.3 数据库性能指标 7](#_Toc2025174845)

[2.6 可行性分析 7](#_Toc1067892241)

[3 需求分析 8](#_Toc360166780)

[3.1 功能定义 8](#_Toc830434873)

[3.2 角色定义 9](#_Toc1033840510)

[3.3 设计系统的逻辑关系 10](#_Toc246661018)

[3.4 系统接口 10](#_Toc2003215039)

[3.4.1图形接口 10](#_Toc1179859002)

[3.4.2 数据库接口 10](#_Toc1415931038)

[3.5 系统接口 10](#_Toc233110281)

[3.6 结构化分析 10](#_Toc309375922)

[3.6.1 数据库逻辑结构设计 10](#_Toc1626820690)

[3.6.2 数据字典表格 12](#_Toc984033649)

[3.6.3 数据流图 17](#_Toc2049134877)

[3.6.4 E-R图 18](#_Toc660048666)

[3.6.5 系统状态转换图 18](#_Toc1259303491)

[3.7 系统性能 19](#_Toc717207679)

[4 概要设计与详细设计 20](#_Toc574838892)

[4.1 系统功能模块划分 20](#_Toc407261577)

[4.2 系统架构方案 20](#_Toc1373219770)

[4.3 系统操作流程 20](#_Toc1737092694)

[4.3.1 系统总体操作流程图 20](#_Toc306794789)

[4.3.2 系统各部分操作流程图 20](#_Toc81627740)

[4.4 前端实现方案 20](#_Toc1063866379)

[4.5 后端实现方案 20](#_Toc2116300963)

[4.5.1 系统操作表格图 20](#_Toc1301687663)

[4.5.2 E-R图 20](#_Toc1567156991)

[4.5.3 数据流图 20](#_Toc1783958987)

[4.5.4 数据库逻辑结构设计 20](#_Toc145231722)

[4.5.6 数据表格设计 20](#_Toc1157265486)

[4.5.7 数据库事务管理 20](#_Toc949612492)

[4.6 设计模式 20](#_Toc22922919)

[4.7 注释及命名规范 20](#_Toc77674079)

[5 测试报告 21](#_Toc1309779272)

[5.1 测试环境 21](#_Toc853357793)

[5.2 功能测试 21](#_Toc1111514589)

[5.3 压力测试 21](#_Toc1556440290)

[6 项目管理 22](#_Toc709089184)

[6.1 项目计划 22](#_Toc143889944)

[6.2 项目实际执行状况 23](#_Toc824887680)

[6.3 版本迭代情况 23](#_Toc942199465)

[6.3.1 代码迭代情况 23](#_Toc453265866)

[6.3.2 文档迭代情况 23](#_Toc304224722)

[6.4 资源使用情况 23](#_Toc1926233114)

[6.5 项目交付情况 23](#_Toc354917095)

[7 课程总结 24](#_Toc964273388)

[参考文献 25](#_Toc1038052958)

[附录 26](#_Toc1072124775)

[文件说明 26](#_Toc1539112280)

# 

# 1 选题

## 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》

## 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

## 辅助工具软件

### 需求分析

思维导图：XMind 8

### 工程管理

MS Project

### 软件建模

MS Visio：UML，数据流图，算法流程图等

### 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

### 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

## 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

## 想法

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。

# 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

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

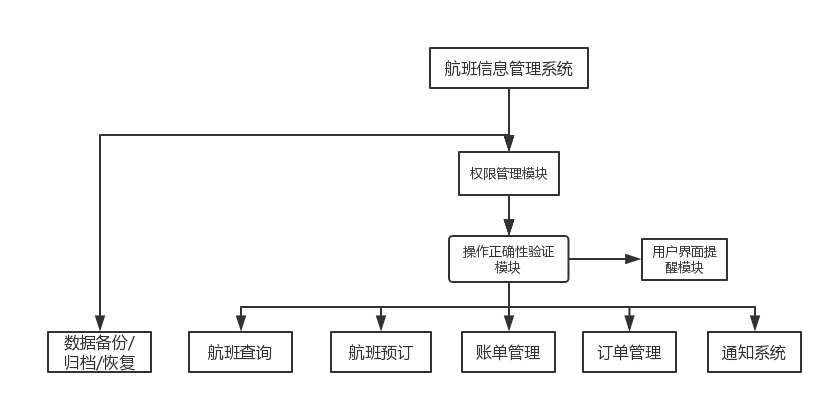


图3-1 系统功能模块图

## 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 |

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 |  |  | √ |

## 3.3 设计系统的逻辑关系

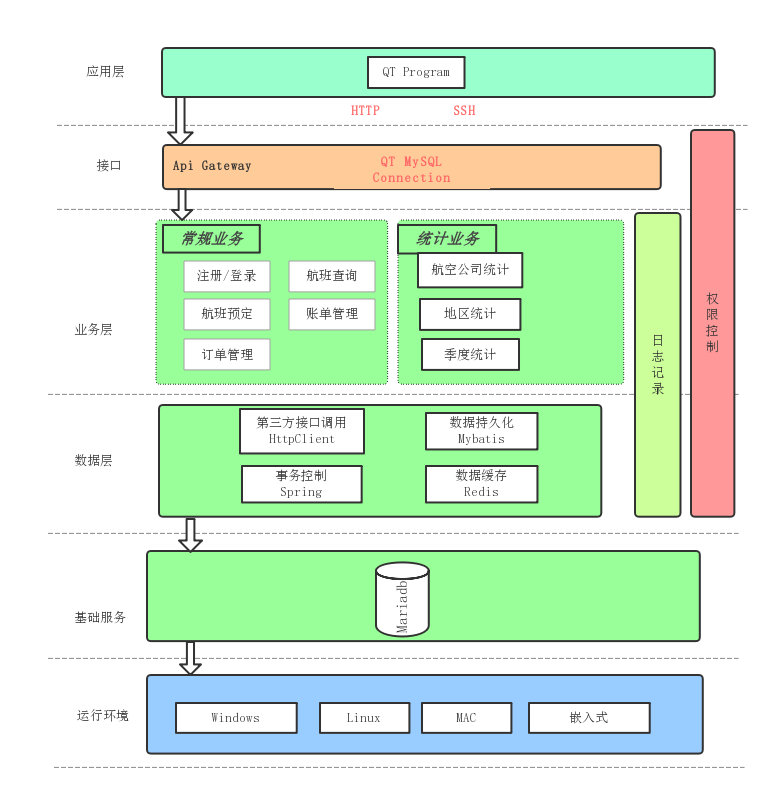


图3-1 系统整体架构图

## 3.4 系统接口

### 3.4.1图形接口

### 3.4.2 数据库接口

## 3.5 系统接口

## 3.6 结构化分析

### 3.6.1 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

### 3.6.2 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) |

### 3.6.3 数据流图

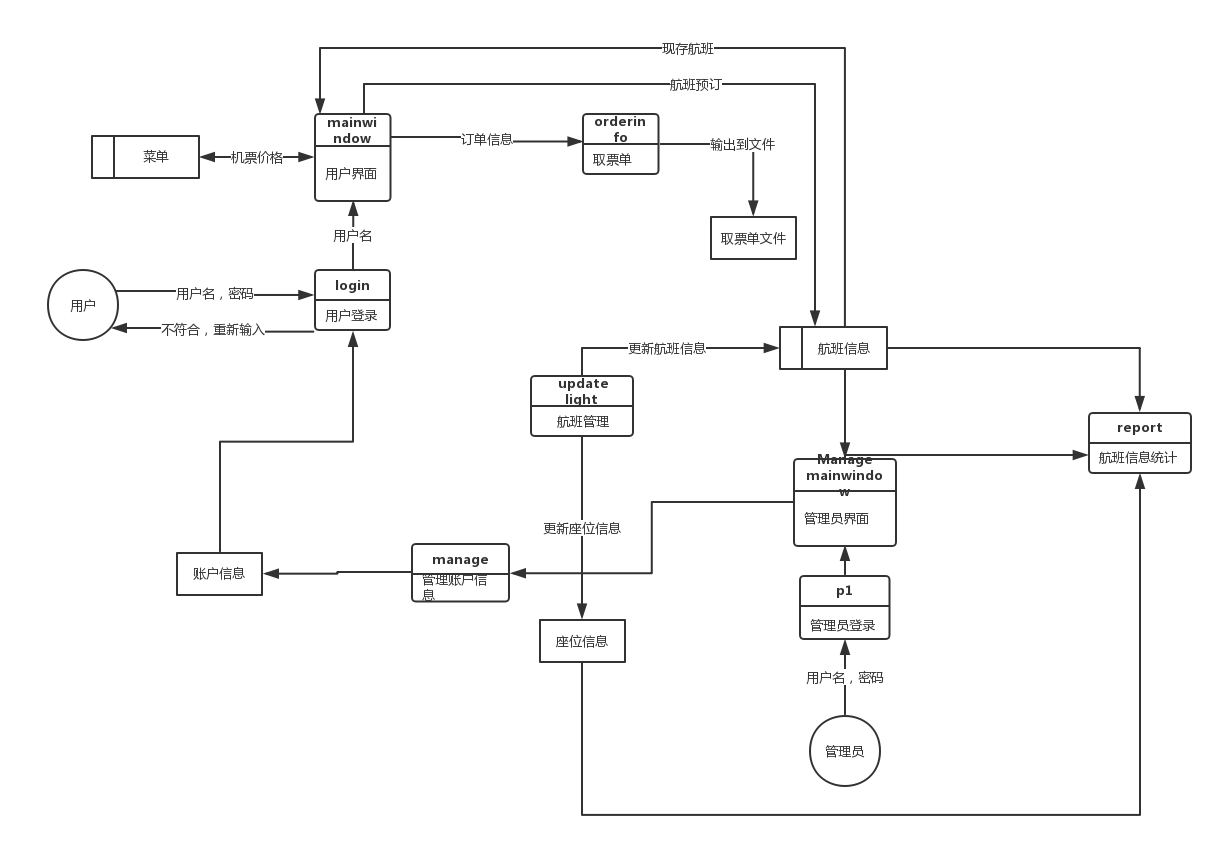


图3-1 数据流图

### 3.6.4 E-R图

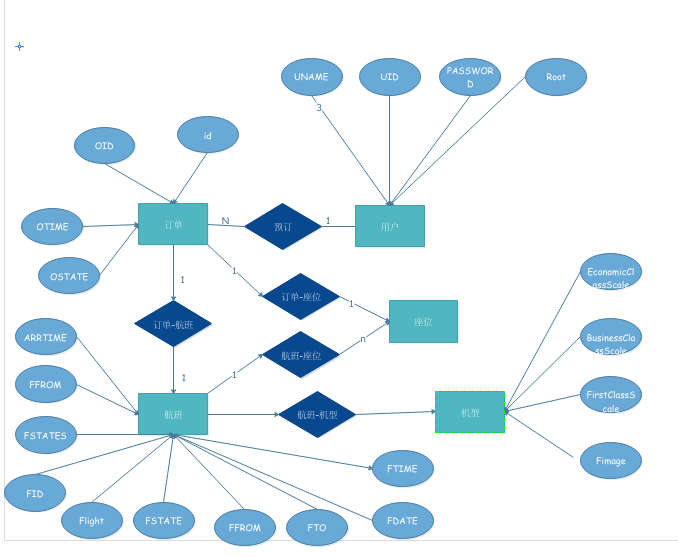


图3-1 E-R图

### 3.6.5 系统状态转换图

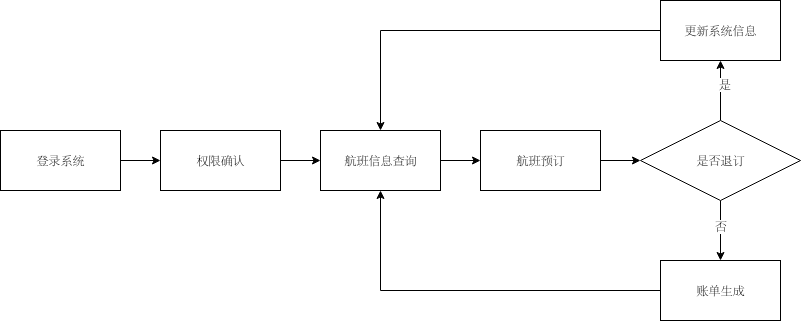


图3-1 系统操作状态转换图

## 3.7 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

# 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

## 4.2 系统架构方案

## 4.3 系统操作流程

### 4.3.1 系统总体操作流程图

### 4.3.2 系统各部分操作流程图

## 4.4 前端实现方案

## 4.5 后端实现方案

### 4.5.1 系统操作表格图

### 4.5.2 E-R图

### 4.5.3 数据流图

### 4.5.4 数据库逻辑结构设计

### 4.5.6 数据表格设计

### 4.5.7 数据库事务管理

## 4.6 设计模式

## 4.7 注释及命名规范

# 5 测试报告

## 5.1 测试环境

## 5.2 功能测试

黑盒测试

## 5.3 压力测试

白盒测试

# 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

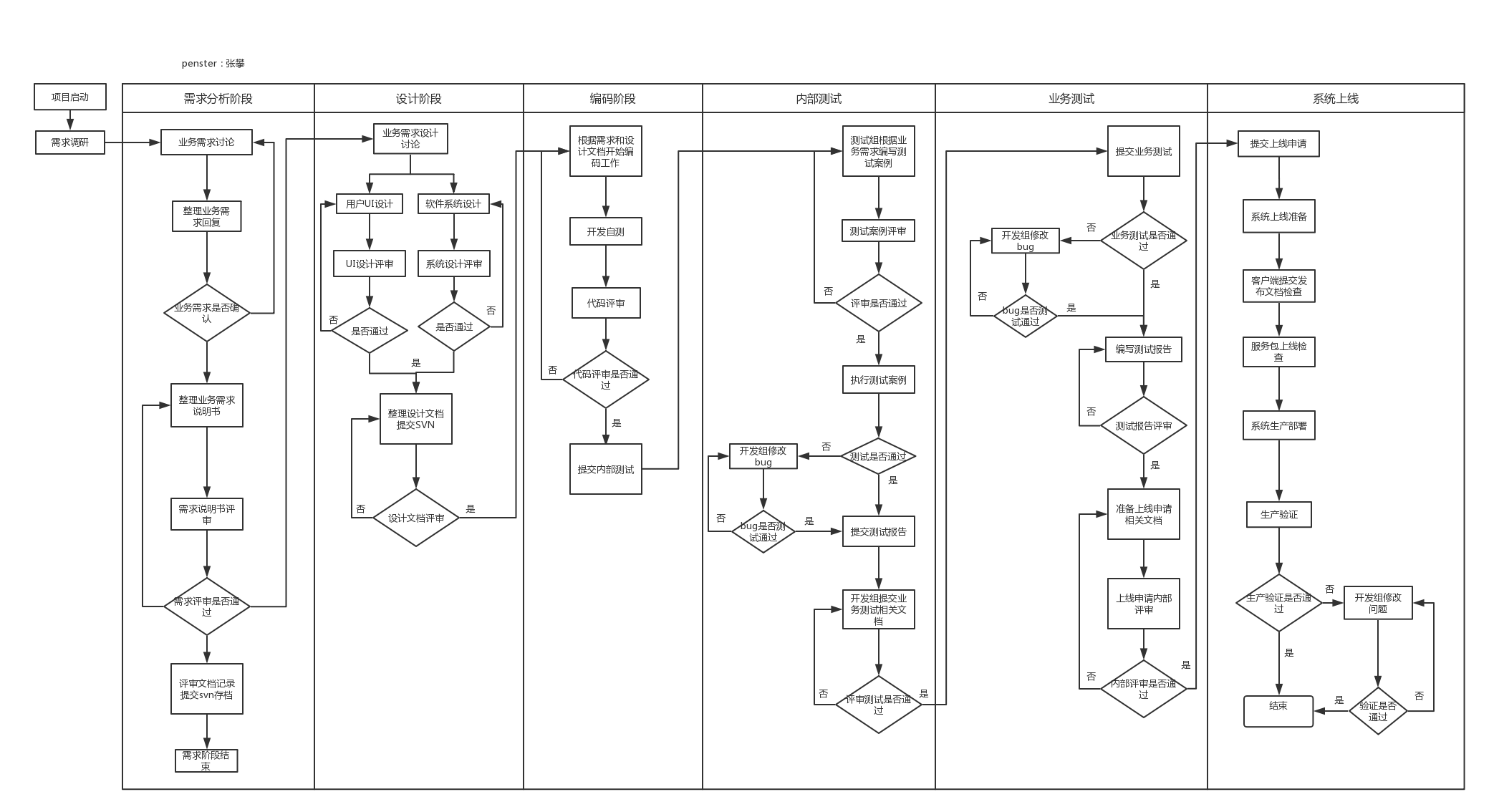


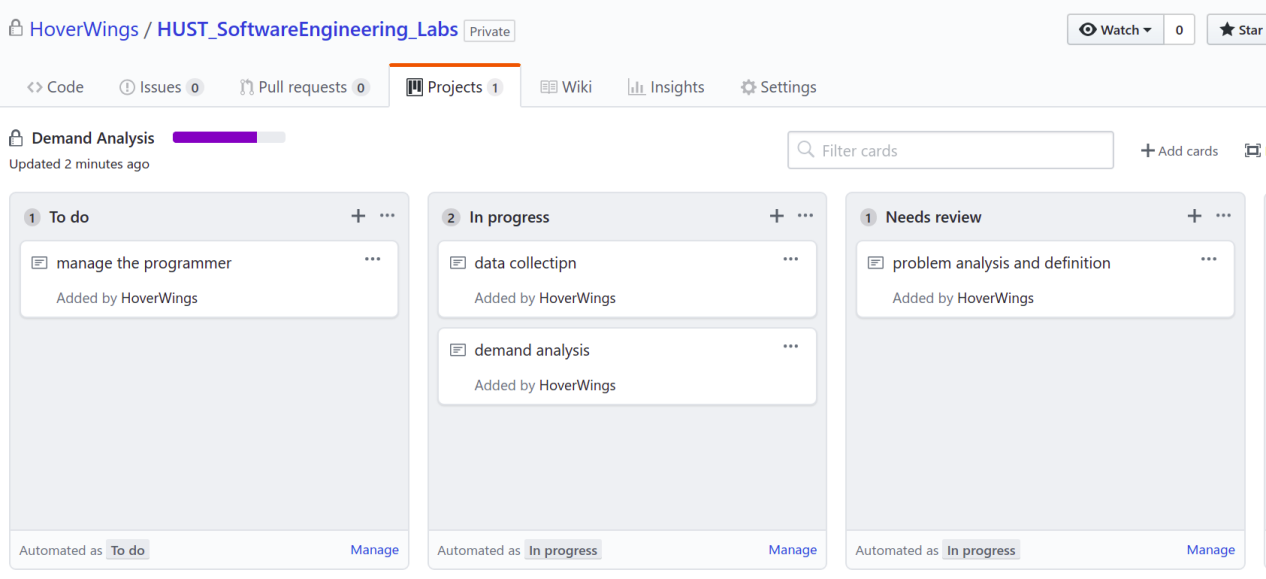
图6-1 项目整体实施流程图

## 6.2 项目实际执行状况

## 6.3 版本迭代情况

### 6.3.1 代码迭代情况

### 6.3.2 文档迭代情况



## 6.4 资源使用情况

## 6.5 项目交付情况

# 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

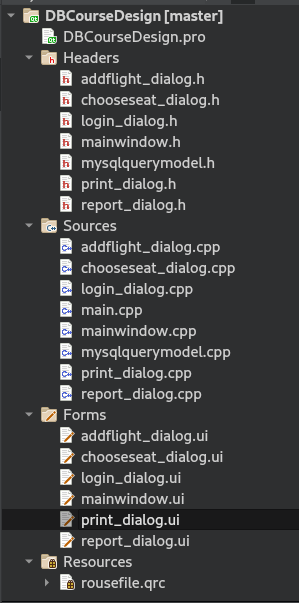
# 附录

## 文件说明

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



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