

# 目录

机票预订系统开发文档..... 错误!未定义书签。

前言 ..... 4

1 可行性研究 ..... 5

1.1 系统概述 ..... 5

1.2 问题定义 ..... 6

1.3 初步的系统分析设计 ..... 6

1.3.1 系统流程图 ..... 7

1.4 技术可行性分析 ..... 9

1.5 经济/效益可行性分析 ..... 10

1.5.1 开发成本 ..... 10

1.5.2 运行费用 ..... 10

1.5.3 经济效益 ..... 11

1.6 操作可行性分析 ..... 12

1.7 系统开发计划 ..... 12

2 需求分析..... 13

2.1 总体目标 ..... 13

2.2 具体目标 ..... 13

2.3 系统数据建模 ..... 13

2.4 系统功能建模 ..... 16

2.5 数据字典 ..... 18

3 总体设计..... 22

3.1 系统结构图 ..... 22

3.2	数据库设计 .....	23
3.2.1	逻辑结构设计 .....	23
3.2.2	物理结构设计 .....	28
3.3	系统模块 IPO 表 .....	29
3.3.1	用户登录主页模块 .....	29
3.3.2	用户查询航班模块 .....	30
3.3.3	用户订购航班模块 .....	31
3.3.4	用户修改个人信息模块 .....	32
3.3.5	用户安全退出模块 .....	33
3.3.6	用户退票模块 .....	33
3.3.7	用户改签模块 .....	34
3.3.8	管理员登录模块 .....	35
3.3.9	管理员管理管理员信息模块 .....	36
3.3.10	管理员管理订单模块 .....	37
3.3.11	管理员管理航班模块 .....	38
4	<u>详细设计 .....</u>	<u>40</u>
4.1	用户登录模块 .....	40
4.2	用户个人中心模块 .....	41
4.3	用户查询、预订航班模块 .....	43
4.4	管理员后台登录模块 .....	44
4.5	管理员管理航班信息功能模块 .....	45
4.6	管理员密码修改功能模块 .....	46
4.7	用户注销登陆模块 .....	47
4.8	用户添加联系人模块 .....	47
5	<u>系统实现 .....</u>	<u>48</u>
5.1	开发平台和开发环境介绍 .....	48
5.2	前台界面与后台云数据库连接代码 .....	50
5.3	各模块功能的实现 .....	51

5.3.1	主界面模块实现.....	51
5.3.2	注册功能实现.....	52
5.3.3	登陆功能实现.....	55
5.3.4	订票功能实现.....	58
5.3.5	机票改签功能实现 .....	60
5.3.6	退票功能实现.....	62
5.3.7	后台添加航班功能实现 .....	67
5.3.8	后台删除航班功能实现 .....	68
5.3.9	后台修改航班功能实现 .....	68
5.3.10	查询航班功能实现 .....	70
<b>5.4</b>	<b>测试.....</b>	<b>76</b>
5.4.1	软件测试的目标.....	76
5.4.2	软件测试的准则.....	77
5.4.3	软件测试的步骤.....	77
5.4.4	单元/集成/验收测试.....	78
5.4.5	测试过程.....	80
<b>6</b>	<b><u>维护.....</u></b>	<b><u>86</u></b>
6.1	系统维护过程.....	86
6.2	系统维护策略 .....	86
<b>7</b>	<b><u>体会.....</u></b>	<b><u>88</u></b>

# 前言

本文档是根据软件工程的基本原理、开发方法以及开发过程进行了多次的讨论与实践的成果，本系统按照软件生命周期的各个阶段相应的任务进行开发，主要包括可行性研究、需求分析、总体设计、详细设计、编码、测试等内容，每一章节都图文并茂地阐述了具体的设计过程，使得该文档具有较高的可读性，易于开发人员进行设计和维护。

# 1 可行性研究

## 1.1 系统概述

机票预订系统是一个基于微信开发者工具作为开发工具，小程序代码编写一样使用 JS 来完成，但是结构和样式采用分别采用 WXML—是微信设计的一套标签语言，他和 HTML 类似，全名叫：Weixin Markup Language，WXSS—是一套样式语言，用于定义样式 和 CSS 类似，被认是 CSS 的子集 全名：Weixin Style Sheets。后端采用云开发技术，无需搭建服务器，即可使用云端能力，云开发为开发者提供完整的原生云端支持和微信服务支持，弱化后端和运维概念，无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现快速上线和迭代，同时这一能力，同开发者已经使用的云服务相互兼容，并不互斥。

每个航空公司都必须解决的问题就是如何提高飞机票售票效率，更好的方便客户，摆脱机械化的管理模式，想要轻松解决这个问题不妨尝试采用机票预订系统使用户在线就能够预订到机票，问题得到解决的同时，还能做到飞机票售出途径多元化，方便乘客购票，提高航空公司的工作效率。同时，一个完整的机票预订系统还应当能使后台的管理员方便高效地管理用户、航班等数据信息。

网上机票预订系统属于应用系统，用到机票预订系统的用户有很多，当然，这个系统提供的功能也有很多，对于用户而言，需要实现的功能有用户登陆、查询航班、预订机票、在线退票、改签航班、个人信息修改等，这些都是用户可以完成的，实现在线预订可以为用户提供便捷的订票渠道，使用户能够通过互联网了解航班与航线的详细情况。对管理员而言，还可以实现制定航班、管理信息等功能。具体而言，管理员登陆本系统之后，可以对管理员的信息进行修改，除此以外，管理员还可以对一些基础信息进行管理，基础信息主要包括航班信息，同时还可以对航班信息进行详细的预览，管理员还可以对订单进行管理，主要包括对用户提交的订票、退票信息进行查看。

用户进行登录操作无需注册可以通过微信账号自动登录，只需要允许读取微信信息即可，用户登陆系统之后，查看并修改个人的基本信息同时可以添加联系人，通过选择起始城市、终点城市以及选定的日期来进行机票预订，同时可以在

个人中心查看自己已经预订的机票，对其进行退票改签等操作。

## 1.2 问题定义

当今时代是飞速发展的信息时代。在各行各业中离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用它能够进行信息管理。使用计算机进行信息控制，不仅提高了工作效率，而且大大的提高了其安全性。因此，对机票预订系统进行分析并澄清该问题定义是非常必要的。

预订机票是一个流程相对固定但繁琐的过程，而预订机票的旅客又非常多。在该过程中，因为数据信息处理的工作量非常大，而且牵扯到很多的数据管理和查询，如果采用人工操作的方式来实现这一切，不仅容易出错，而且效率也非常低下、数据也容易丢失且不易查找。

基于上述分析，这整个相对固定的流程意味着其能够以较低的成本通过计算机系统来实现，而且该系统的实现非常有必要。这样就能使预订机票工作规范化，系统化，程序化，避免预订机票的随意性，同时也能提高信息处理的速度和准确性，能够及时、准确、有效地完成机票预订工作。

机票预订系统需要满足两方面的需求，其分别是旅客、航空公司。

## 1.3 初步的系统分析设计

机票预订系统应当包含以下几个系统模块：

1. 查询航班模块：该系统模块主要实现航班信息的查询，用户可以根据行程的起始地、出发地以及具体日期，查看当天的所有航班情况。
2. 预订机票模块：该系统模块主要实现机票的预订、退票以及改签功能。用户在找到自己想要的航班后，可以进行航班的预订、在订单页实现后续的退票、改签等功能。
3. 管理员模块：该系统模块主要描述了系统管理员特有的功能。具有对系统中已有的航班、个人信息的查询以及修改功能。
4. 用户模块：该系统模块主要描述了用户登录、查询修改个人信息等功能。用户可根据用户名和密码进行登录，登录后可以修改个人信息，密码等。未注

册用户可进行注册。

### 1.3.1 系统流程图

系统流程图是概括得描绘物理系统的传统工具。其基本思想是利用图形符号以黑匣子的形式描绘组成系统的各个部件。其强调的是数据在系统各部件间的流动情况，选择性地暂时忽略对各数据加工处理的控制过程。以一种物理数据流图的形式展现系统的各个功能运作流程，为人们初步理解项目系统提供帮助。基于上述分析的五个系统模块，本项目系统的业务流程图如下图所示。

#### (1) 用户-登录注册业务

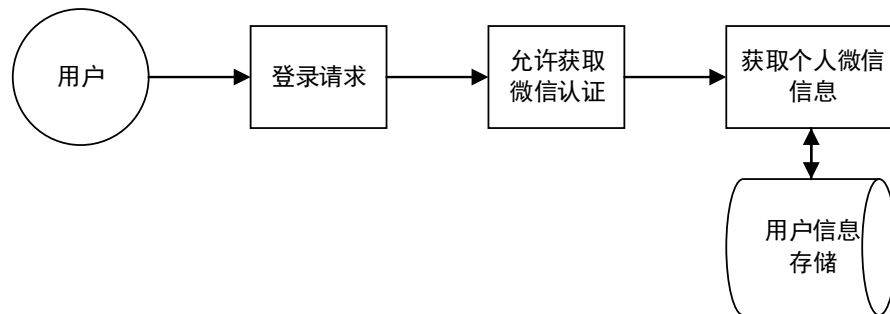


图 1. 1 用户-登录流图

#### (2) 用户预订退订改签机票业务

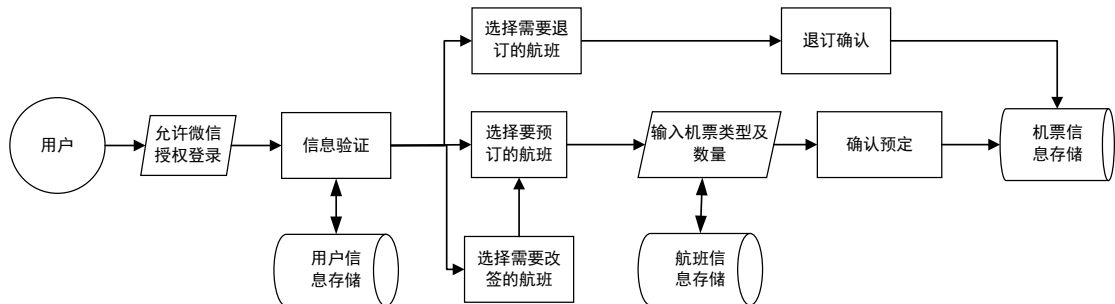


图 1. 2 用户预订退订改签机票业务流图

#### (3) 用户查询航班业务

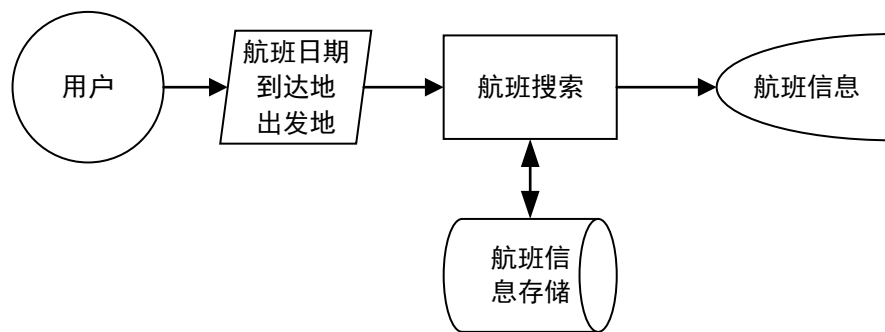


图 1. 3 用户查询航班业务流程图

(4) 用户查看已预订航班业务

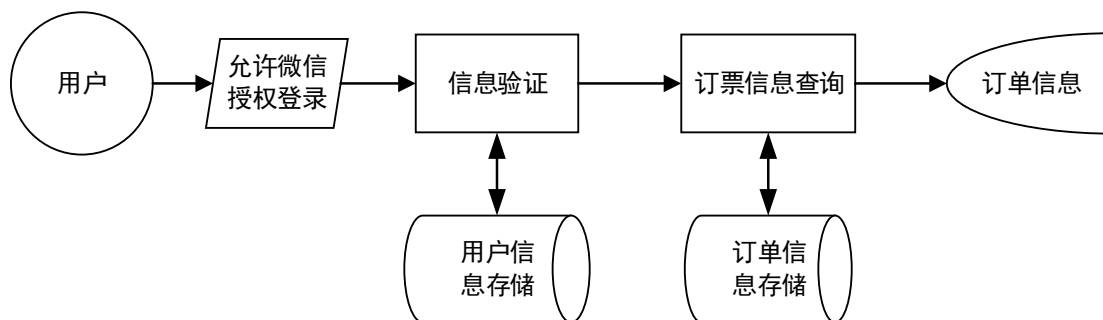


图 1. 4 用户查看已预订航班业务流程图

(5) 管理员对用户、航班等信息管理业务

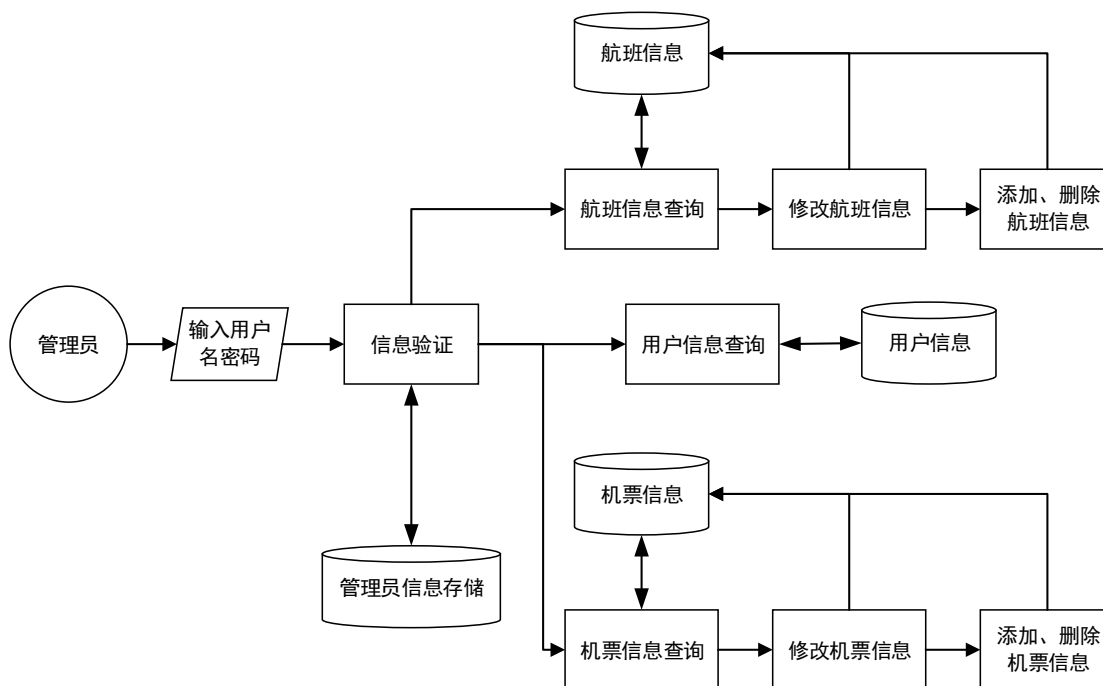




图 1. 5 管理员对用户、航班、订单等信息管理业务流程图

## 1.4 技术可行性分析

结合问题描述和问题定义来看，该问题具有相对固定的流程，能够以较低的成本通过现有的计算机系统来实现。经过 1.3 节中的分析，说明该系统能够以结构化的分析模型表示出来，所以该系统是可行的。最终我们决定将该系统做成微信小程序，因为目前市面上已经有了很多类似订票业务的微信小程序，比如“智行火车票”火车票抢票小程序，所以利用腾讯提供的微信小程序开发工具是完全能够胜任该机票预订系统的开发工作。除了在前述技术角度上，该系统的实现是可行的，使用微信小程序来开发该系统亦具有诸多优点。

首先，微信已经在事实上成为了中国一个流量巨大的载体，而该机票预订系统就是提供机票预订服务，倘若没有流量，又该给谁提供服务呢？就这一点来看，使用微信小程序来开发该系统就具有很强大的天然优势。其次，微信小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。它不需要用户去下载 APP 或客户端，只需要在微信内部搜索小程序即可，用户使用成本极低。而且在当今时代，人们的手机已经下满了 APP，眼花缭乱不说，其还会占据用户手机的大量资源，而又有很多 APP 用户的使用它们的频率又并不高。所以，我们小组对该系统的定位就是：即搜即得，用完即走。所以微信小程序是该系统开发的不二选择。

除上述之外，微信小程序还拥有良好的开发生态环境，其提供了一个简单、高效的应用开发框架和丰富的组件及 API，帮助开发者在微信中开发具有原生 APP 体验的服务。网页

<https://developers.weixin.qq.com/miniprogram/dev/framework/>即为微信小程序开发的官方文档支持，其涵盖了微信小程序开发的方方面面，而且微信小程序还有官方社区，不仅有其他开发者会帮助我们答疑解惑，腾讯官方的开发者也会出来答疑解惑。

再者就是微信小程序的云开发能力了，这是不得不提的一点，在过去，想要创建一个能上线的软件，需要各方企业和软件提供支持，通常我们还需要将其连接起来，配置各种环境。而在微信小程序中，开发者可以使用云开发开发微信小

程序、小游戏，无需搭建服务器，即可使用云端能力，比如云数据库、云存储、云函数、云调用等等。而且微信小程序的云数据库不再是我们之前所学的关系型数据库了，其使用的是 JSON 数据库，一种非关系型数据库，不仅查询效率高，而且查询也非常简单。故使用微信小程序开发软件不需要安装、连接额外的软件，一个微信小程序开发工具就能全部搞定了。

为了高质量完成本软件系统开发工作，还需要遵循如下准则：

- (1) 做好数据的规划，建立稳定的信息模型；
- (2) 在功能模型的划分上，按组织机构来划分子系统或模块；
- (3) 应用软件的开发设计，要充分考虑应用软件的适应性，建立友好的人机界面。

综上所述，使用微信小程序来开发该机票预订系统在技术上是完全可行的。

## 1.5 经济/效益可行性分析

### 1.5.1 开发成本

机票订购系统主要涉及的开发成本来源于项目前期准备、软件设计、编码实现以及产品测试人员的人力消耗，开发过程中所使用的集成开发平台、SSH 开发框架均属于开源性质。

采用代码行技术对项目成本进行简单估算，根据以往开发基于微信小程序项目的经验，我们可以预估出程序的最小规模  $a=3490$  行，最大规模  $b=5000$  行，最可能的规模  $m=3750$  行，利用代码行计算公式可以求出代码规模的估计值：

$$L = \frac{a+4m+b}{6} = 3915 \text{ 行}$$

设定每行代码的平均成本为 2 元，可以推出开发成本的估计值为 7830 元。

### 1.5.2 运行费用

该项目的运行费用主要来自系统运营人员的工资和 API 使用的租赁费用。后期系统运营人员的工作主要涉及定期的数据备份、网站安全的维护以及对使用者的基本培训等方面，是项目正常运行必不可少的一环。按目前市场行情来看，考虑到地区之间的差异性，维护与本项目相类似的小型网站的月工资大概在 5000

人民币左右。

部分功能的 API 的租赁费用属于基本的开销。每天有一定数量的航班往返，所以机票订购系统的用户基数会多于一些常见的校内小系统，而且要支持用户购票、退票的并发操作。因此，该项目对于 API 使用的需求量较大。按当前的市场价位估计，该项目在 API 上的开销大约在 10000 元左右。

### 1.5.3 经济效益

经过前面的分析可以得到此项目的当前投资成本约为 22830 元，根据网上查阅的系统资料可知，类似的交通票务订购系统每年约可收益 12000 元。从经济角度出发判断此项目的效益不能直接把当前投资成本与现在市场行情下的收益进行比较，还应考虑货币的时间价值。

目前主要以利率的形式表示货币的时间价值。假设年利率为*i*，如果*n*年后能收入*G* 元，那么这些钱的现在价值为：

$$P = G/(1 + i)^n$$

查阅资料可知目前存款的基准年利率为 1.5%。假设该项目的生命周期为 5 年。根据上面的收入数据可以计算出机票订购系统每年预计收益的钱的现在价值，如表 1.1 所示：

表 1.1 将来收益折算成现在价值

年	将来值（元）	$(1+i)^n$	现在值（元）	累积现在值（元）
1	12000	1.015	11823	11823
2	12000	1.030	11648	23471
3	12000	1.046	11476	34947
4	12000	1.061	11306	46253
5	12000	1.077	11139	57392

根据累积现在值，可以得出项目的预计纯收入为 34562 元，投资回收期为 1.47 年，可以求得项目投资回收率为 41.97%。该项目投资回收期较短，回收率较高，在经济效益上具有较好的可行性。

## 1.6 操作可行性分析

因为该系统不需要具备很强的计算机专业知识，只需要对操作人员进行简单的技术培训，操作人员即可快速上岗，推动整个系统的使用。同时，该系统也可以使工作人员摆脱繁琐的工作流程，从而可以使他们高效、准确地完成对应的工作。所以，该系统在操作上是可行的。

## 1.7 系统开发计划

在开始我们正式的系统开发前，我们小组进行了大量的讨论与探索，所以共花了两周时间来完成机票预定系统的可行性分析，需求分析和总体设计部分。把基本结构搭出来后，后面四周我们进行了系统设计的详细分工和代码的编写。最后两周我们整合了前面六周的工作，然后完成了程序的测试，具体实现了各模块的功能，最后完成了我们最终的机票预定系统开发文档报告。

## 2 需求分析

### 2.1 总体目标

本项目的目标在于依托网络平台将航班信息与旅客用户信息集合在统一的订票管理系统下，简化旅客订购机票的流程，方便航空公司对航班线路、用户信息和订单信息等的管理，实现旅客与航空公司之间的互利共赢。对订票用户而言，如何快捷地查询航班信息，如何方便地选择航班订票、改签、退票是该项目要重点考虑的实现目标；对航空公司管理员而言，本项目主要的实现目标考虑的是如何合理地管理用户、客机和航班信息以及直观地查看航班订单情况，从而为航班制订提供依据，以实现更高的收益。

### 2.2 具体目标

机票预订系统具体包含以下几个模块：

1. 查询航班模块：该系统模块主要实现航班信息的查询，用户可以根据行程的起始地、出发地以及具体日期，查看当天的所有航班情况。
2. 预订机票模块：该系统模块主要实现机票的预订、退票以及改签功能。用户在找到自己想要的航班后，可以进行航班的预订。
3. 管理员模块：该系统模块主要描述了系统管理员特有的功能。具有对系统中已有的用户、航班信息的增删改查功能。
4. 用户模块：该系统模块主要描述了用户登录、查询修改个人信息等功能。用户可根据用户名和密码进行登录，登录后可以修改个人信息，密码等。未注册用户可进行注册。
5. 订单详情模块：该系统主要实现了在用户完成订单后查询自己订单的详情信息，并且对订单进行修改（订单取消、订单改签等）。

### 2.3 系统数据建模

系统数据模型包含三种互相关联的信息：数据对象、数据对象的属性和数据对象彼此间相互连接的关系。本项目使用实体-联系图来清晰、准确地描述用户

的数据要求。E-R 图属于面向问题的数据模型，即概念性数据模型。它从用户角度出发看待数据，反映用户的现实环境，与软件系统中的实现方法无关。

全局的 E-R 图设计如图 2.1 所示：

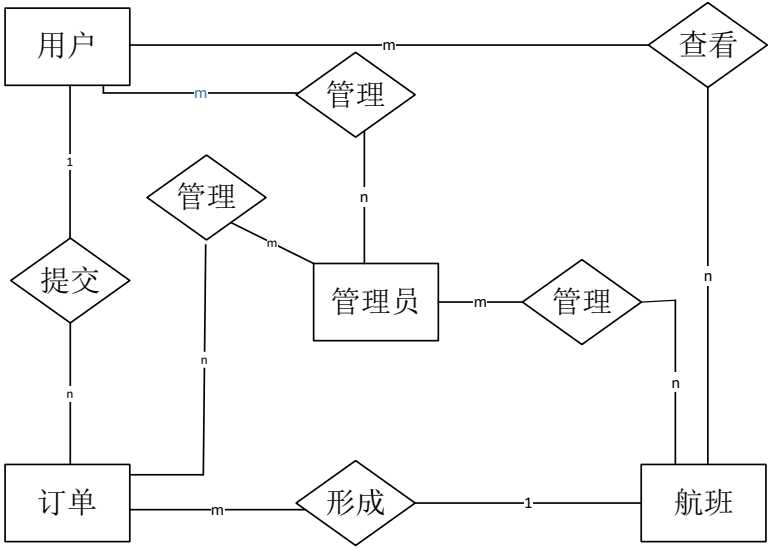


图 2. 1 全局 E-R 图

对全局 E-R 图 2.1 进行分解，得到以下局部 E-R 图

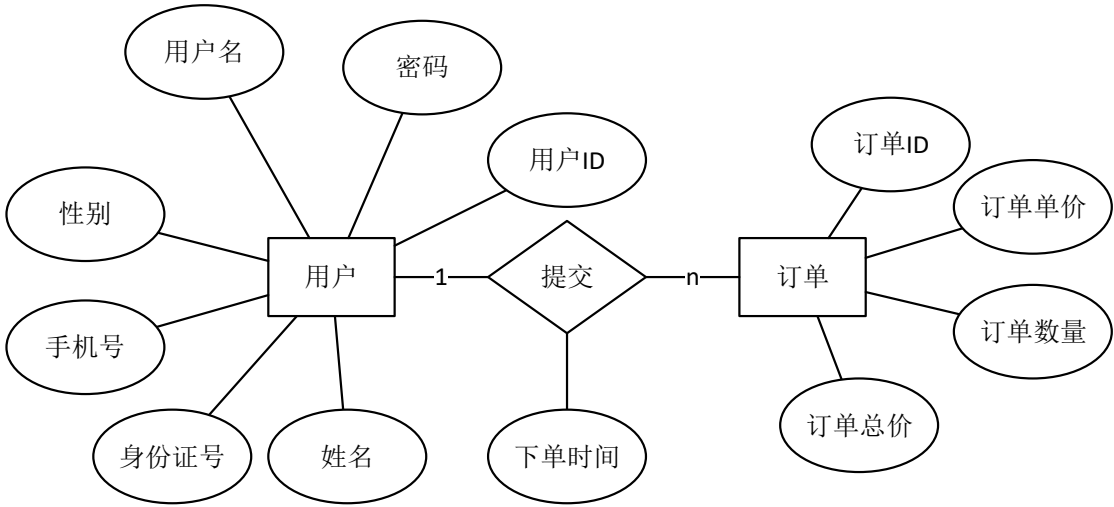


图 2. 2 用户提交订单

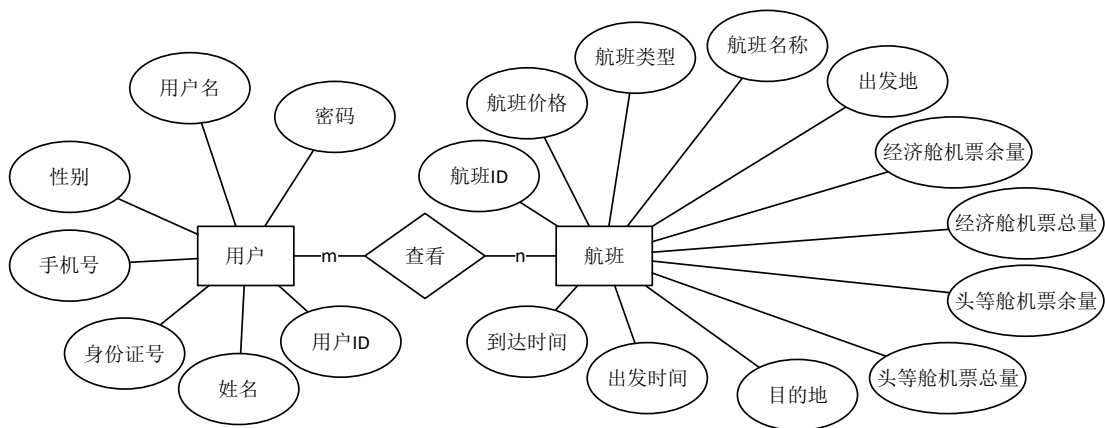


图 2. 3 用户查看航班

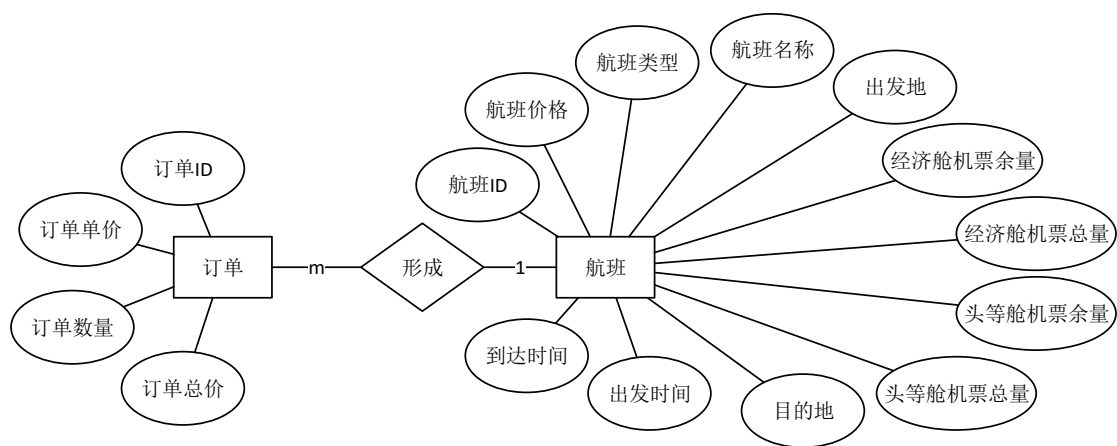


图 2. 4 航班形成订单

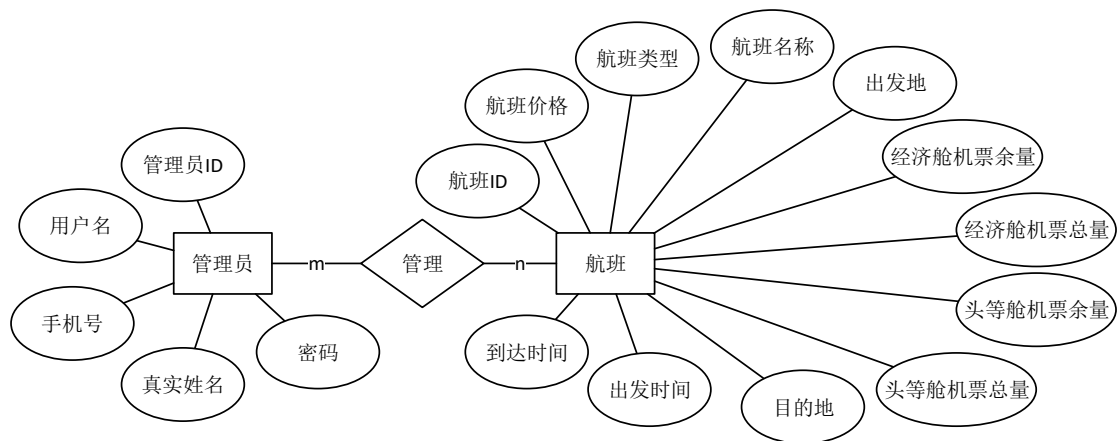


图 2. 5 管理员管理航班

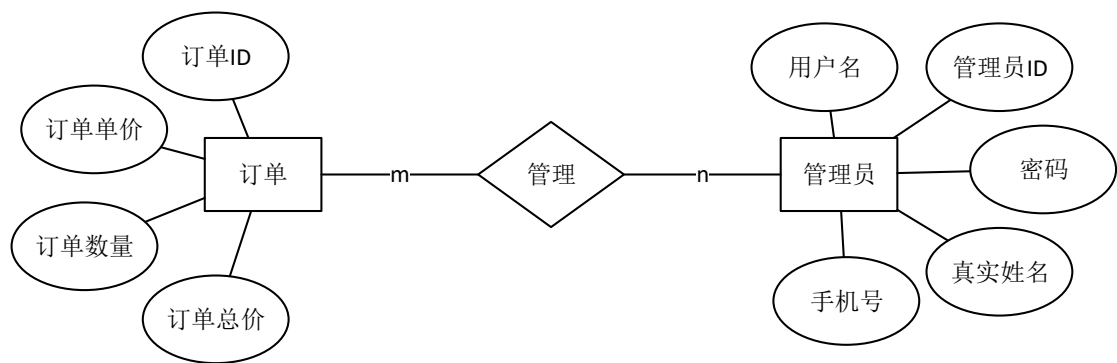


图 2. 6 管理员管理订单

## 2.4 系统功能建模

如下图 2.10 顶层数据流图所示，该系统整体上可以分为两个实体和一个处理过程。其实体分别为 E0 用户、E1 管理员，处理过程为 P0 机票预订系统。由于系统主要功能在可行性分析阶段已经进行了严谨的分析讨论，因此在该阶段可以直接沿用之前的顶层数据流图。该数据流图以人机交互思想为指导，从整体观念出发，展示了流入和流出系统的所有数据信息。

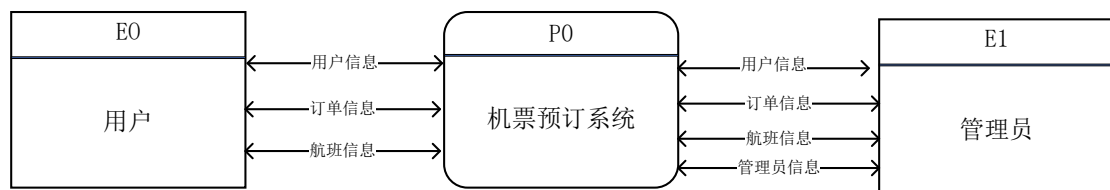


图 2.8 顶层数据流图

0 层数据流图 2.11 将 E0 用户、E1 管理员之间的消费信息数据流进行第一步细化。其中，将处理过程 P0 机票预订系统细分为 P0 管理员与用户信息管理、P1 订单查询及修改、P2 航班信息管理、P3 航班信息查询、P4 订单处理；并引入了多个数据存储，如用户表、管理员表、订单表、航班表。由于本系统在可行性研究阶段已经进行了详细的分析论证，因此 0 层数据流图可在需求分析阶段直接沿用。



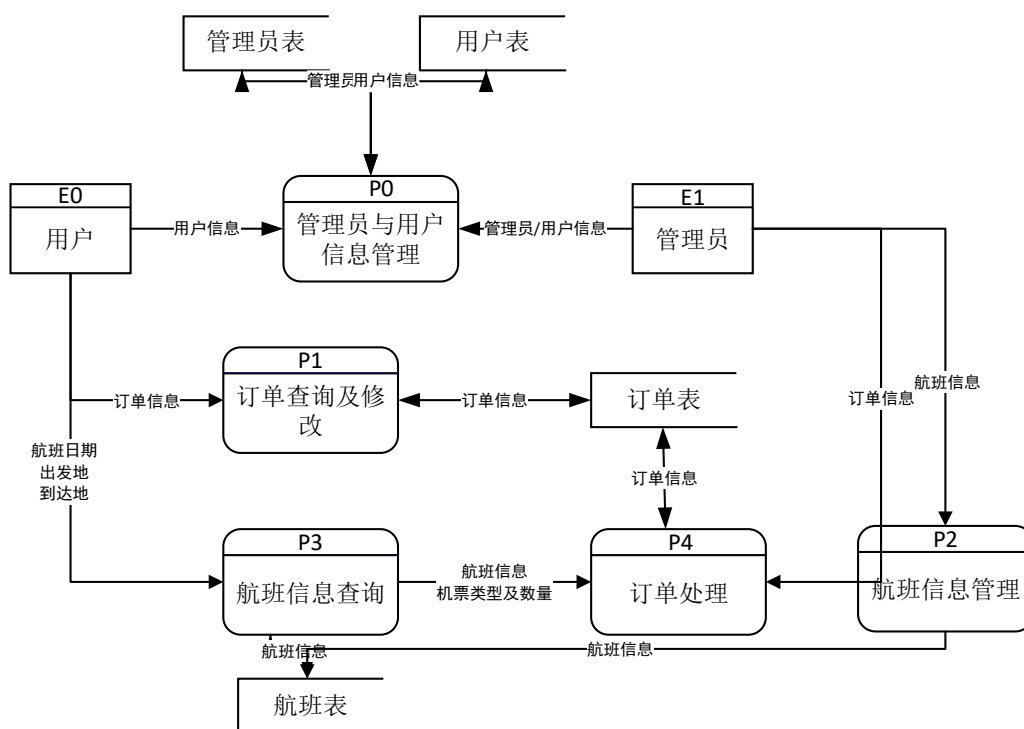


图 2. 9 0 层数据流图

在系统数据、功能的基础上，我们需要进一步建立起系统的行为模型来描述作为外部事件结果的软件行为。我们使用状态转换图来描绘机票订购系统在面对用户登录购票、管理员管理航班、用户信息等事件发生时所产生的动作。

用户登录后，可以根据起始地、目的地以及起飞时间查询航班信息，选定想买的航班后选择想乘坐的舱位类型进行付款购票。对于已购的机票，倘若后续无法乘坐对应的航班，还可以选择退票、改签操作。用户层面的状态图如下图所示：

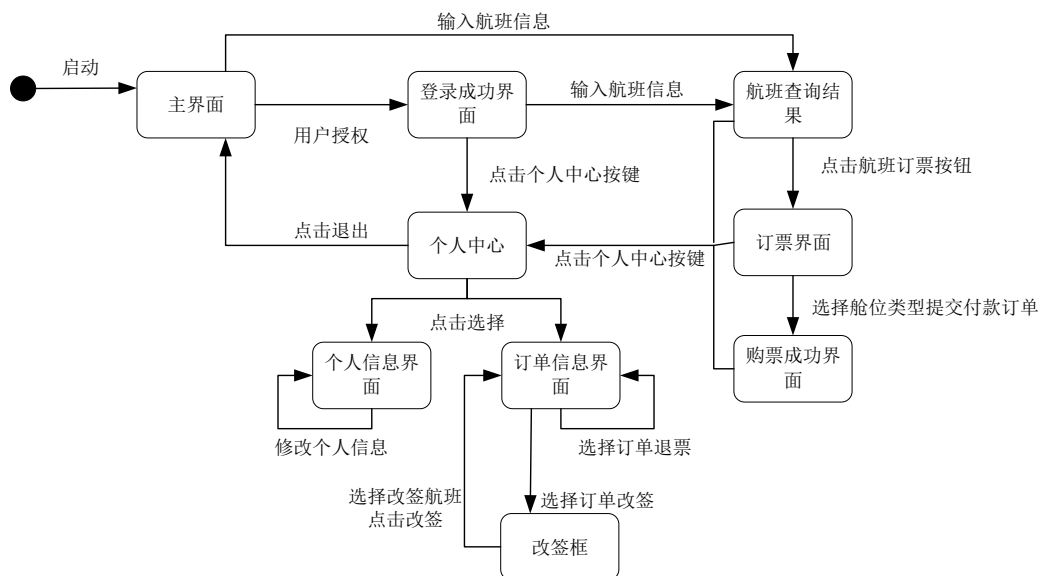


图 2. 5 用户层面状态图

管理员登录后，可以对用户信息、航班信息和订单信息进行管理，还可以修改管理员自身的密码信息。管理员层面的状态图如下图所示：

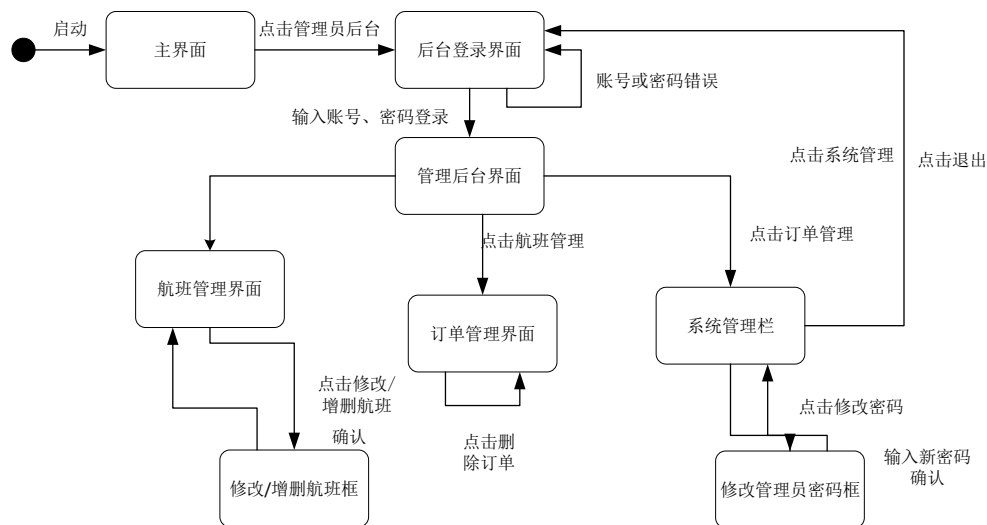


图 2. 6 管理员层面状态图

## 2.5 数据字典

数据字典最重要的用途是作为分析阶段的工具，典型的情况是，在数据字段中记录数据元素的下列信息：一般信息（名称，描述），定义（数据类型（微信小程序的特殊数据类型为 **number**：数值、**string**：字符串、**boolean**：布尔值、**object**：对象、**function**：函数、**array**：数组、**date**：日期、**regex**：正则），

长度，结构等），使用特点（值的范围，使用频率，使用方式—输入、输出、本地、条件值等），控制信息（来源，用户，使用它的程序，改变权，使用权等）等。在可行性研究中的初步分析的基础上，我们认为本系统应当实现航空公司对于每个航班所使用的飞机机型进行管理的需求；其次，管理员与普通用户在本系统中属于不同层次的用户，考虑到系统后续拓展的方便，应当将管理员与用户信息采用不同的表进行存储。综上所述，我们需要建立用户表、航班信息表、管理员表、订单表等。对在可行性分析研究阶段得到的数据字典进行补充完善如下：

表 2. 1 用户数据字典

名称:用户信息
描述:使用机票预订系统的用户
定义:用户=用户 ID+用户 Openid+虚拟网址+电子邮箱+性别+身份证号码+语言+真实姓名+昵称+手机号
字段:用户 ID= string 用户 Openid=string 虚拟网址=string 电子邮箱=string 性别= 【0 男   1 女】 身份证号码=1 {string} 18 语言=string 真实姓名=1 {string} 20 昵称=1 {string} 20 手机号=1 {string} 11
位置:用户表

表 2. 2 航班数据字典

名称:航班信息
描述:记录每个航班的信息
定义:航班=航班 ID+目的地+到达时间+出发地+出发时间+经济舱机票总量+经济舱价格+经济舱机票余量+头等舱机票总量+头等舱机票价格+头等舱机票余量+航班名称

字段:航班 ID= string 目的地=1 {string} 到达时间=date 出发地=1 {string} 出发时间=date 经济舱机票总量=number 经济舱价格=number 经济舱机票余量=number 头等舱机票总量=number 头等舱机票价格=number 头等舱机票余量=number 航班名称=1 {string} 20
位置: 航班表

表 2. 3 订单数据字典

名称:订单信息
描述:记录每笔交易订单的信息
定义:订单= <u>订单 ID</u> +订单总价+ <u>用户 ID</u> +用户名+用户手机号码+航班( <u>航班 ID</u> , 目的地, 到达时间, 出发地, 出发时间, 经济舱机票总量, 经济舱价格, 经济舱机票余量, 头等舱机票总量, 头等舱机票价格, 头等舱机票余量, 航班名称)+ <u>随行联系人</u> +下单时间+舱位类型
字段: 订单 ID=string 订单总价=number 用户 id=string 用户名=string 用户手机号码=string 航班=object 航班 ID=string 随行联系人=string 下单时间=date

舱位类型=【经济舱   头等舱】
位置: 订单表

表 2. 4 管理员数据字典

名称: 管理员信息
描述: 使用机票预订系统的管理员
定义: 管理员=管理员 ID+用户名+密码+真实姓名+手机号
字段: 管理员 ID=string 用户名=1 {string} 20 密码=1 {string} 20 手机号=1 {string} 11 真实姓名=1 {string} 20
位置: 管理员表

表 2. 5 联系人数据字典

名称: 联系人信息
描述: 用户订单中添加的随行的联系人
定义: 用户=联系人 ID+联系人 openID+电子邮箱+身份证号码+真实姓名+手机号+用户关系
字段: 联系人 ID= string 联系人 Openid=string 电子邮箱=string 身份证号码=1 {string} 18 真实姓名=1 {string} 20 手机号=1 {string} 11 昵称=1 {string} 20
位置: 联系人表

### 3 总体设计

总体设计的基本目的是回答“概括地说，系统应该如何实现”的问题，因此总体设计又称为概要设计或初步设计。通过这个阶段的设计将划分出的物理元素——程序、文件、数据库、人工过程、文档等，但每个物理元素仍然处于黑盒子级，这些黑盒子里的具体内容将在以后仔细设计。总体设计阶段的另一任务是设计软件的结构，也就是要确定系统中每个程序是由哪些模块组成的，以及这些模块之间的关系。总体设计的两大阶段是：系统设计阶段和结构设计阶段。

由于本项目相对简单，无需进行方案的选择，因此直接进入功能分解、软件结构的设计、数据库的设计等阶段，并且由于面向数据流的设计方法需要相当精细的数据流图，因此本项目直接采用自顶向下逐步求精的方法进行软件结构的设计。根据软件设计“高内聚，低耦合”的设计原理，用层次图表示出相互之间独立的模块，并用 IPO 图描述每个模块的具体功能；然后根据需求分析阶段的数据模型 ER 图，建立数据库。完成总体设计阶段的任务。

#### 3.1 系统结构图

因为本系统在前面需求分析阶段已经得到了较为完善可行的顶层数据流图、0 层数据流图，所以在本总体设计阶段直接沿用上述得到的图即可，具体详见图 3.1-图 3.3。

层次图可以用来描述软件的层次结构，适用于在自顶向下设计软件的过程中使用，层次图其中的一个矩形框代表一个模块，方框间的连线表示调用关系而不像层次方框图那样表示数据的组成关系。本系统分别从用户端、管理员端；两个方面绘制系统结构图，具体如下所示。

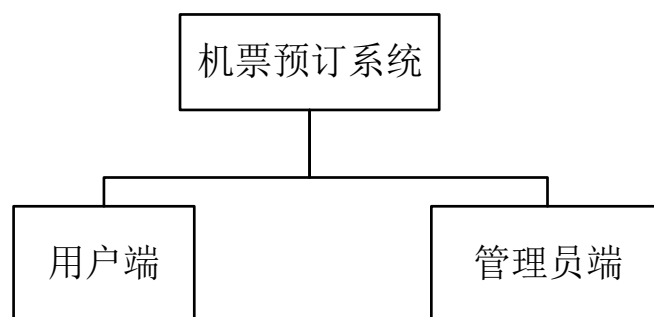


图 3. 1 总体层次结构图

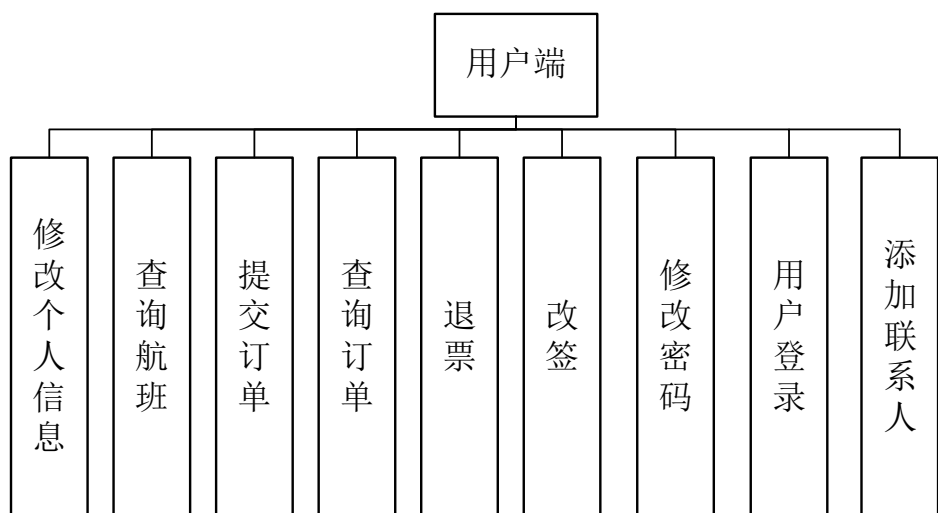


图 3. 2 用户端层次结构图

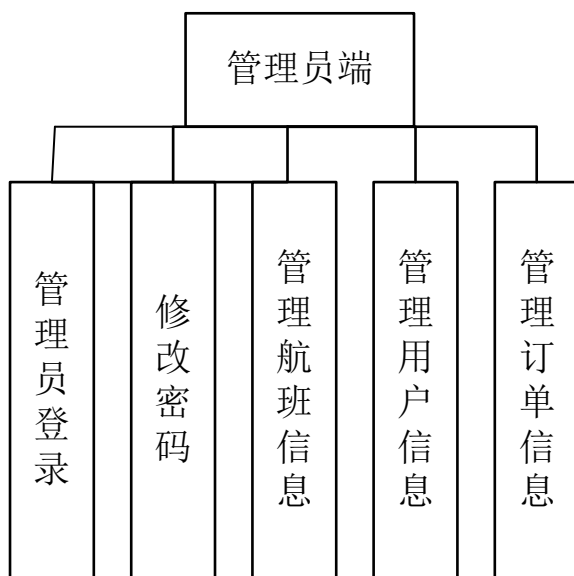


图 3. 3 管理员端层次结构图

上述设计的功能模块最高程度的耦合为数据耦合，并均为功能内聚，因而符合“高内聚，低耦合”的设计理念。

## 3.2 数据库设计

### 3.2.1 逻辑结构设计

#### (1) 关系模式设计

系统的关系模式可由系统数据建模中的 E-R 图分析设计而得。其中 E-R 图模式中的实体和联系都可以转化成关系型，其属性可以转化为关系型的属性，实体

型的主键也可以转化为关系型的主键。而对实体集之间的联系转换受到其自身类型的影响，需要具体问题具体分析。针对该项目中的联系转换，本项目作出以下几点调整：

(1) 对于 1:1 型的联系转换：先将两个实体型分别转换为对应的关系模式后，再将其中一个实体的主键属性加入另一个实体当中，即可完成相关转换。

(2) 对于 1:n 型的联系转换：除了需要将实体自身的属性直接进行转换，对在 n 端的实体，需要将在 1 端实体的主键放入其中。需要按其修改的联系有——用户和订单间的联系，需要将用户表的主键即用户 ID 加入订单表中；航班和订单之间的联系，需要将航班表的主键即航班 ID 加入订单表中；客机和航班之间的联系，需要将客机 ID 加入航班表中。

(3) 对于 n:m 型的联系转换：除了需要将实体自身的属性直接进行转换，在设计时还需要将两者间的联系转换为一个关系型。该关系型的属性需要包含其对应两个实体的主键属性和联系自身的属性。

(4) 对于某些特殊的 n:m 型的联系转换：在本项目中多对多联系有管理员与用户之间的联系、管理员与航班之间的联系、管理员与订单之间的联系、管理员与飞机之间的联系、用户与航班之间的联系；由于上述与管理员相关的联系仅仅是一种权限方面产生的联系，不需要进行存储，因此没有必要为这些联系建立新的数据表；同样，用户与航班之间的多对多联系仅仅只是一种由于使用产生的联系，同样不需要建立新的数据表。

按上述要求建立系统的关系模式，其主要内容如下所示，其中，下划线表示该关系型的主键，波浪线表示该关系型的外键。

- ① 用户表（用户 ID，用户 Openid(微信小程序开发特有)，虚拟网址(微信小程序特有)，电子邮箱，性别，身份证号码，语言，真实姓名，昵称，手机号)
- ② 管理员表（管理员 ID，管理员姓名，密码，电子邮箱，手机号)
- ③ 航班信息表（航班 ID，目的地，到达时间，出发地，出发时间，经济舱机票总量，经济舱价格，经济舱机票余量，头等舱机票总量，头等舱机票价格，头等舱机票余量，航班名称)
- ④ 联系人表（联系人 ID，联系人 openID，电子邮箱，身份证号码、真实姓名、手机号，用户关系)



- ⑤ 订单信息表（订单 ID，订单总价，用户 ID，用户名，用户手机号码，航班  
（航班 ID，目的地，到达时间，出发地，出发时间，经济舱机票总量，经济  
舱价格，经济舱机票余量，头等舱机票总量，头等舱机票价格，头等舱机票  
余量，航班名称），随行联系人，下单时间，舱位类型）

## (2) 数据类型定义

接下来对关系模式设计中的属性定义类型、长度和约束，得到表 3.1-表 3.5

表 3.1 用户表

数据项名	英文名	数据类型	长度	实体完整 性约束	参照完整 性约束	用户自定 义性约束
用户 ID	User_ID	String	/	√	/	not null
用户 Openid	User_open	String	20	/	/	not null
虚拟网址	Avatar	String	100	/	/	not null
电子邮箱	User_email	String	40	/	/	/
性别	gender	Number	/	/	/	0(男)/1(女)
身份证号 码	User_idnum	String	18	/	/	/
语言	User_language	String	20	/	/	/
真实姓名	User_name	String	20	/	/	/
昵称	Nickname	String	20	/	/	/
手机号	User_Phone	String	11	/	/	/

表 3.2 管理员表

数据项名	英文名	数据类型	长度	实体完整 性约束	参照完整 性约束	用户自定 义性约束
管理员	Manage_ID	String	/	√	/	not null

ID						
真实姓名	Manage_Name	String	20	/	/	not null
密码	Password	String	20	/	/	not null
电子邮箱	Eamil	String	20	/	/	/
手机号码	Phonenum	String	20	/	/	/

表 3. 3 订单表

数据项名	英文名	数据类型	长度	实体完整性约束	参照完整性约束	用户自定义性约束
订单 ID	Order_ID	String	/	√	/	not null
总价	totalprice	Number	/	/	/	not null
用户 ID	User_id	String	/	/	√	not null
用户手机号码	User_phone	String	11	/	√	not null
航班:	Flight	Object	/	/	√	
航班 ID	flight_ID	String	/	√	√	not null
目的地	Tocity	String	20	/	√	/
到达时间	Totime	Date	/	/	√	/
出发地	fromcity	String	20	/	√	/
出发时间	Fromtime	Date	/	/	√	/
经济舱总量	ecTicketTotal	Number	/	/	√	/
经济舱余量	ecTicketRemain	Number	/	/	√	not null
经济舱价格	ecPrice	Number	/	/	√	not null
头等舱总量	fcTicketTotal	Number	/	/	√	not null

头等舱余量	fcTicketRemain	Number	/	/	√	/
头等舱价格	fcPrice	Number	/	/	√	not null
航班名称	Name	String	20	/	√	not null
随行联系人序号	Contasts_No	Number	/	/	/	/
下单时间	Order_	date	/	/	/	/
舱位类型	Spacetype	String	20	/	/	/

表 3.4 联系人表

数据项名	英文名	数据类型	长度	实体完整性约束	参照完整性约束	用户自定义性约束
联系人ID	Contasts_ID	String	20	√	/	not null
联系用户	User_ID	String	20	/	√	not null
电子邮箱	Contasts_Email	String	20	/	/	not null
联系人身份证号	Contasts_idnum	String	18	/	/	not null
真实姓名	Contasts_name	String	20	/	/	not null
手机号	Contasts_phone	String	11	/	/	not null
用户关系	Relationship	String	10	/	/	/

表 3. 5 航班表

数据项名	英文名	数据类型	长度	实体完整性约束	参照完整性约束	用户自定义性约束
------	-----	------	----	---------	---------	----------

航班 ID	flight_ID	String	/	√	/	not null
目的地	Tocity	String	20	/	/	/
到达时间	Totime	Date	/	/	/	/
出发地	fromcity	String	20	/	/	/
出发时间	Fromtime	Date	/	/	/	/
经济舱总量	ecTicketTotal	Number	/	/	/	/
经济舱余量	ecTicketRemain	Number	/	/	/	not null
经济舱价格	ecPrice	Number	/	/	/	not null
头等舱总量	fcTicketTotal	Number	/	/	/	not null
头等舱余量	fcTicketRemain	Number	/	/	/	/
头等舱价格	fcPrice	Number	/	/	/	not null
航班名称	Name	String	20	/	/	not null

### 3.2.2 物理结构设计

数据库最终是存储在物理设备上的，数据库在物理设备上的存储结构和存取方法就称为数据库的物理结构，它依赖于具体的计算机系统。所谓数据库的物理结构设计就是为一个给定数据库的逻辑结构选取一个最适合应用环境的物理结构和存取方法的过程，其目的是为了提高数据库的访问速度并有效地利用存储空间。

#### (1) 聚簇设计

聚簇是将有关数据元组集中存放在一个物理块内或若干相邻物理块内或同一柱面内，以提高查询效率的数据存储结构。建立聚簇有如下原则：

1. 当对一个关系的某些属性列的访问是该关系的主要应用，而对其他属性的访

问很少或是次要应用时，可以考虑对该关系在这些属性列上建立聚簇。

2. 如果一个关系在某些属性列上的值重复率很高，则可以考虑对该关系在这些组属性列上建立聚簇。
3. 如果一个关系一旦装入数据，某些属性列的值很少修改，也很少增加或删除元组，则可以考虑对该关系在这些组属性列上建立聚簇。

## (2) 分区设计设计

磁盘分区设计的本质是确定数据库数据的存放位置，其目的是提高系统性能，是数据库物理设计的内容之一。

磁盘分区设计的一般原则：

1. 减少访问冲突，提高 I/O 并发性。多个事物并发访问同一磁盘时，会产生磁盘访问冲突而导致效率低下，如果事务访问数据均能分布于不同磁盘上，则 I/O 可并发执行，从而提高数据库访问速度。
2. 分散热点数据，均衡 I/O 负担。在数据库中数据访问的频率是不均匀的，那些经常被访问的数据成为热点数据，此类数据宜分散存在于不同的磁盘上，以均衡各个磁盘的负荷，充分发挥多磁盘的并行操作的优势。
3. 保证关键数据快速访问，缓解系统瓶颈。在数据库中有些数据如数据字典等的访问频率很高，为保证对它的访问不直接影响整个系统的效率，可以将其存放在某一固定磁盘上，以保证其快速访问。

本实验的机票预订系统由于程序较小，所以不进行分区设计。

## 3.3 系统模块 IPO 表

### 3.3.1 用户登录主页模块

#### (1) 功能描述

对登录系统的用户进行身份验证。

#### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	用户登录	模块编号	01
作者		日期	2021/11/18

模块描述	对登录系统的用户进行身份验证。	
调用模块	数据库模块	
被调用模块	用户模块	
输入	项目	授权信息
	格式	手动授权
处理	1、用户对登录进行授权即可登陆成功	
输出	用户登录成功/失败。	
局部数据元素	数据库表	
注释	需要数据库用户信息表	

### (3) 主要算法

【登录】按钮：验证顾客的合法性。

## 3.3.2 用户查询航班模块

### (1) 功能描述

为用户提供查询航班的功能。

### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	用户查询	模块编号	02
作者		日期	2021/11/18
模块描述	为用户提供查询航班的功能。		
调用模块	数据库模块		
被调用模块	用户模块		
输入	项目	出发地点，到达地点，出发日期	
	格式	帐号：String 密码：String 出发日期：滑动	
处理	1、用户输入出发地点，到达地点，出发日期 2、模块获得出发地点，到达地点，出发日期等信息 3、模块调用数据库表进行遍历搜索 4、模块返回符合条件的航班信息		

输出	航班详细信息
局部数据元素	数据库表
注释	需要数据库航班信息表

### (3) 主要算法

【搜索】按钮：遍历数据库当中符合条件的数据库表项。

【重置】按钮：清空已输入的信息

## 3.3.3 用户订购航班模块

### (1) 功能描述

为用户提供订购航班的功能。

### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	订购机票	模块编号	03
作者		日期	2021/11/18
模块描述	为用户提供订购航班的功能		
调用模块	数据库模块		
被调用模块	用户模块		
输入	项目		
	格式	订购数量：number；舱位类型：单选框输入；航班： 通过按钮选择输入	
处理	1、用户输入订购数量、舱位类型 2、模块获得出订购数量、舱位类型、选中的航班等信息 3、模块调用数据库表进行遍历搜索 4、模块将信息存储在订单表中 5、模块修改航班的信息 6、模块返回提示信息		
输出	订购成功或失败的信息		
局部数据元素	数据库表		

注释	需要数据库航班信息表
----	------------

### (3) 主要算法

【订购】按钮：选择数据库中的表项，判定是否为合法购买，如果机票余量不足，返回错误信息，如果为合法购买，修改航班表、订单表表项

## 3.3.4 用户修改个人信息模块

### (1) 功能描述

为用户提供修改个人信息的功能，包括修改邮箱、密码、手机号、性别、身份证号、真实姓名、用户名等信息。

### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	修改个人信息	模块编号	04
作者		日期	2021/11/18
模块描述	为用户提供修改个人信息的功能		
调用模块	数据库模块		
被调用模块	用户模块		
输入	项目	密码、手机号、性别、身份证号、真实姓名、用户名	
	格式	密码: String; 性别: String; 身份证号: String; 真实姓名: String; 用户名: String; 手机号: String	
处理	1、用户登录进入个人信息模块 2、用户选择要更改的个人信息 3、用户进行个人信息修改并提交 4、系统接收信息进行校验，符合要求后写入数据库 5、系统弹出信息修改成功对话框		
输出	修改信息成功/失败		
局部数据元素	数据库表		



注释	需要数据库用户信息表
----	------------

### 3.3.5 用户安全退出模块

#### (1) 功能描述

退出当前用户的登陆状态

#### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	用户退出模块	模块编号	05
作者		日期	2020/11/18
模块描述	用于用户退出登陆		
调用模块	数据库模块		
被调用模块	用户模块		
输入	项目	无	
	格式	无	
处理	1、用户登录进入服务界面 2、用户注销登录状态 3、系统返回未登录的界面		
输出	注销成功界面		
约束条件	无		
注释	无		

#### (3) 主要算法

【注销】按钮：注销用户

【取消】按钮：关闭注销界面

### 3.3.6 用户退票模块

#### (1) 功能描述

将用户已经预订的机票做退票处理

#### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	用户退票模块	模块编号	06
作者		日期	2020/11/18
模块描述	用于用户退订机票		
调用模块	数据库模块		
被调用模块	用户模块		
输入	项目	退票点击按钮	
	格式	无	
处理	1、用户登录进入退单界面 2、用户点击订票管理按钮 3、用户选择需要退订的订单 4、系统弹出退款成功的对话框及修改数据库		
输出	退款成功的对话框或输出历史退单		
约束条件	无		
注释	需要数据库航班、订单信息表		

### (3) 主要算法

【退票】按钮：退订已预订的机票并修改数据库的订单表与航班表

## 3.3.7 用户改签模块

### (1) 功能描述

将用户已经预订的机票做改签处理

### (2) 模块 IPO 表

系统名称	机票预订系统		
模块名称	用户改签模块	模块编号	07
作者		日期	2020/11/18
模块描述	将用户已经预订的机票做改签处理		
调用模块	数据库模块		
被调用模块	用户模块		

输入	项目	改签点击按钮
	格式	无
处理	1、用户登录进入个人中心界面 2、用户点击订票管理按钮 3、用户选择需要改签的订单 4、系统弹出改签成功的对话框及修改数据库	
输出	改签成功的对话框	
约束条件	无	
注释	需要数据库订单、航班信息表	

### (3) 主要算法

【改签】按钮：改签已预订的机票并修改数据库的订单表与航班表

## 3.3.8 管理员登录模块

### (1) 功能描述

对登录系统的管理员进行身份验证。

### (2) 模块 IPO 图

系统名称	机票预订系统		
模块名称	管理员登录	模块编号	8
作者		日期	2020/11/18
模块描述	用于系统对管理员的登录进行身份验证		
调用模块	数据库模块		
被调用模块	无		
输入	项目	管理员用户名，密码	
	格式	管理员用户名：String，管理员密码：String	
处理	1、用户输入用户名、密码 2、模块获得用户名、密码等信息 3、模块调用数据库表进行比较校验 4、模块返回登录信息		

输出	管理员登录成功或登录失败。
局部数据元素	数据库表
注释	需要数据库管理员表

(3) 主要算法

【登录】按钮：验证管理员的合法性。

### 3.3.9 管理员管理管理员信息模块

(1) 功能描述

用于管理员对自己信息进行修改。

(2) 模块 IPO 图

系统名称	机票预订系统		
模块名称	管理员管理管理 员信息	模块编号	9
作者		日期	2020/11/18
模块描述	用于管理员对自己信息进行修改		
调用模块	数据库模块		
被调用模块	无		
输入	项目	管理员用户名，管理员密码，	
	格式	管理员用户名：String，管理员密码：String	
处理	1、管理员登录系统进入管理员界面 2、管理员修改自身信息 3、管理员提交更改后的自身信息 4、系统接收信息写入数据库中 5、系统弹出更新成功对话框		
输出	更新成功对话框		
约束条件	无		
注释	需要数据库管理员表		

(3) 主要算法

【修改】按钮：管理员修改自身信息。

【提交】按钮：管理员提交更改的信息。

### 3.3.10 管理员管理订单模块

#### (1) 功能描述

用于管理员对订单信息进行查询、修改、删除。

#### (2) 模块 IPO 图

系统名称	机票预订系统		
模块名称	管理员管理订单信息	模块编号	11
作者		日期	2020/11/18
模块描述	用于管理员对订单信息进行查询、更改、删除		
调用模块	数据库模块		
被调用模块	无		
输入	项目	订单 ID	
	格式	订单 ID: String	
处理	1、管理员登录系统进入管理员界面 2、管理员修改订单信息 3、管理员添加订单信息 4、管理员提交更改后的订单信息 5、管理员删除订单 6、系统接收信息写入数据库中 7、系统弹出更新成功对话框		
输出	更新成功对话框		
约束条件	无		
注释	需要数据库订单表		

#### (3) 主要算法

【修改】按钮：管理员修改订单信息。

【添加】按钮：管理员添加订单信息。

【删除】按钮：管理员删除订单。

【提交】按钮：管理员提交更改的信息。

### 3.3.11 管理员管理航班模块

#### (1) 功能描述

用于管理员对航班信息进行查询、修改、删除账户。

#### (2) 模块 IPO 图

系统名称	机票预订系统		
模块名称	管理员管理航班信息	模块编号	12
作者		日期	2020/11/18
模块描述	用于管理员对航班信息进行查询、更改、删除		
调用模块	数据库模块		
被调用模块	无		
输入	项目	航班 ID	
	格式	航班 ID: String	
处理	1、管理员登录系统进入管理员界面 2、管理员修改航班信息 3、管理员添加航班信息 4、管理员提交更改后的航班信息 5、管理员删除航班 6、系统接收信息写入数据库中 7、系统弹出更新成功对话框		
输出	更新成功对话框		
约束条件	无		
注释	需要数据库航班表		

#### (3) 主要算法

【修改】按钮：管理员修改航班信息。

【添加】按钮：管理员添加航班信息。

**【删除】按钮：**管理员删除航班。

**【提交】按钮：**管理员提交更改的信息。

## 4 详细设计

详细设计阶段的根本目标是确定应该怎样具体地实现所要求的系统，即不仅在逻辑上正确地实现每个模块的功能，更重要的是设计出的处理过程应该尽可能简明易懂。描述程序处理过程的工具称为过程设计的工具，它们可以分为程序流程图、盒图、问题分析图（PAD）图等，由于这些工具各有其优缺点，我们将在下面使用各种不同的过程设计工具对各个细分的模块进行详细设计。

### 4.1 身份选择模块

该模块用于登录前选择身份，具体如下程序流程图所示：

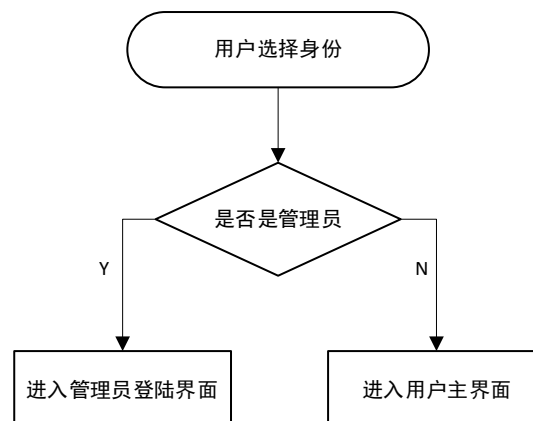


图 4.1 身份选择模块程序流程图

#### 伪代码设计

```
Begin
    等待用户点击按钮选择身份
    if (用户选择管理员身份)
        进入管理员登陆界面
    else
        进入用户主界面
End
```

### 4.2 用户登录模块

该模块用于用户进行机票预订系统的登录，具体如下程序流程图所示：



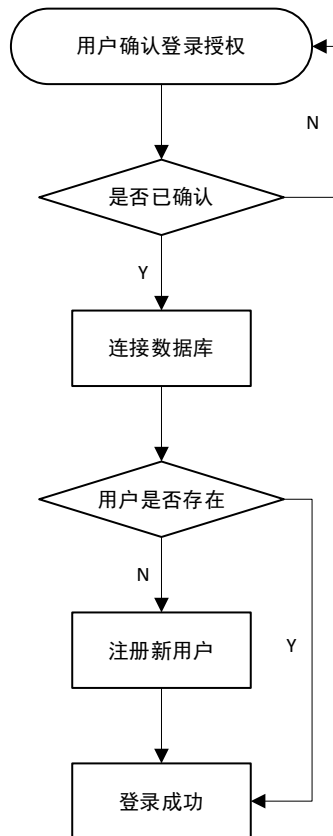


图 4.1 用户注册模块程序流程图

### 伪代码设计

```

Begin
    弹出登录信息，用户端进行授权登陆
    if (用户确认登录授权)
        连接数据库
        if (用户不存在)
            注册新用户
        提示登录成功，进入主界面
    else
        等待用户确认授权
End
  
```

## 4.3 用户个人中心模块

用于用户查看自己的信息并进行修改，还可以查看自己的订票管理情况，进行改签或退票，具体如下盒图所示：

个人主页面				
点击修改资料	点击添加联系人	点击航班		点击退出登录
跳转至个人信息	跳转至联系人信息	跳转至订票管理		
显示个人信息	输入联系人的相关信息	显示订票情况		
点击完善或修改		点击退票	点击改签	
输入修改信息	点击添加按钮	点击确认	弹出改签选框	退出回到主界面
点击确认			选择改签航班	
修改成功反馈	添加完成	退票成功反馈	点击确认	
			改签成功反馈	

图 4.2 用户个人中心模块盒图

## 伪代码设计

```

Begin
    进入个人主页面
    if (用户点击“修改资料”)
        连接数据库
        显示相关个人信息
        if (用户点击某项信息)
            输入修改的信息
            if (用户点击“提交”)
                连接数据库
                修改相关个人信息
                弹出修改成功提醒
            return Begin
    if (用户点击“修改联系人”)
        连接数据库
        显示联系人信息
        if (用户点击添加按钮)
            弹出联系人添加界面
            输入联系人的相关信息
        if (用户点击“提交”)
            连接数据库
            添加联系人信息
  
```

```

        弹出添加成功提醒
        return Begin
    if (用户点击某个联系人信息)
        输入修改的信息
        if (用户点击"提交")
            连接数据库
            修改相关联系人信息
            弹出修改成功提醒
            return Begin
    if (用户点击"退出登录")
        退出个人中心，跳至主界面
End

```

## 4.4 用户查询、预订航班模块

用于用户查询航班，选择航班进行订票，具体如下 PAD 图所示：

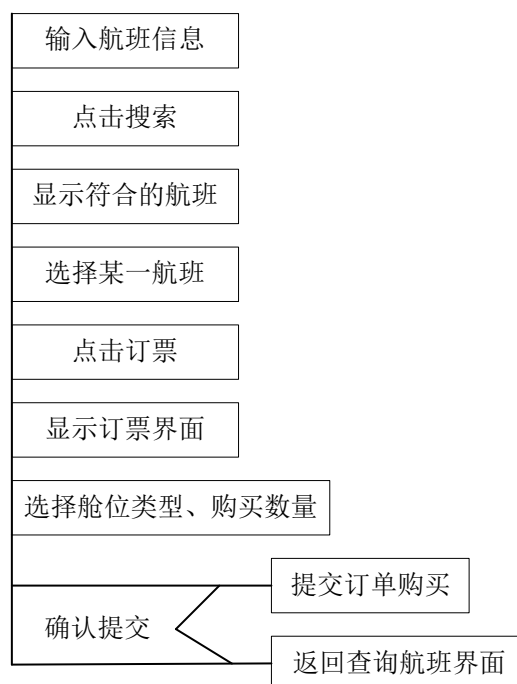


图 4.3 用户查询、预订航班 PAD 图

### 伪代码设计

```

Begin
    显示查询主界面
    用户输入要查询航班的相关信息
    if (用户点击"查询")
        连接数据库
        显示符合的航班
        if (用户选择某一航班点击)

```

```

    显示订票界面
    用户填写订票信息
    if (用户点击“提交订单”)
        连接数据库
        增加相应记录
        弹出订票成功提醒
    return Begin
End

```

## 4.5 管理员后台登录模块

用于管理员登录后台，具体如下程序流程图所示：

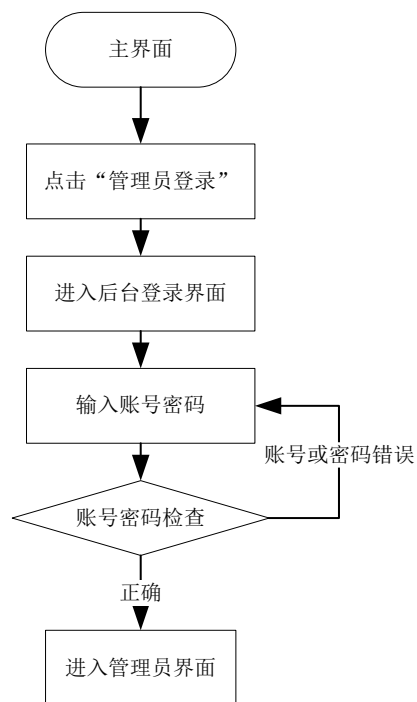


图 4.4 管理员后台登录程序流程图

### 伪代码设计

```

Begin
    进入管理员后台登录界面
    输入账号、密码
    if (账号密码为空)
        弹出错误提醒
    else
        连接数据库
        if (账号及密码都正确)
            进入管理员后台界面
        else
            提示账号或密码错误，要求重新输入
    
```

## 4.6 管理员管理航班信息功能模块

用于管理员管理航班信息，具体如下盒图所示，在图中，将某一用户/客机/航班信息简称为某一记录。



图 4.5 管理员管理用户/客机/航班信息盒图

### 伪代码设计

```

Begin
    进入航班管理界面
    if (管理员点击添加按钮)
        跳出新建航班信息输入框；
        输入航班信息
        if (管理员点击“提交”)
            连接数据库
            增加相应记录
            return Begin
    if (管理员选择某一航班点击)
        连接数据库
        显示航班信息
        管理员进行信息修改
  
```

```

        if (管理员点击“提交”)
            连接数据库
            修改相应记录
            return Begin
    if (管理员在相应航班左滑)
        出现“删除”滑块
        if (管理员点击“删除”)
            连接数据库
            删除相应记录
End

```

## 4.7 管理员密码修改功能模块

该模块用于实现管理员密码修改，具体如下程序流程图所示：

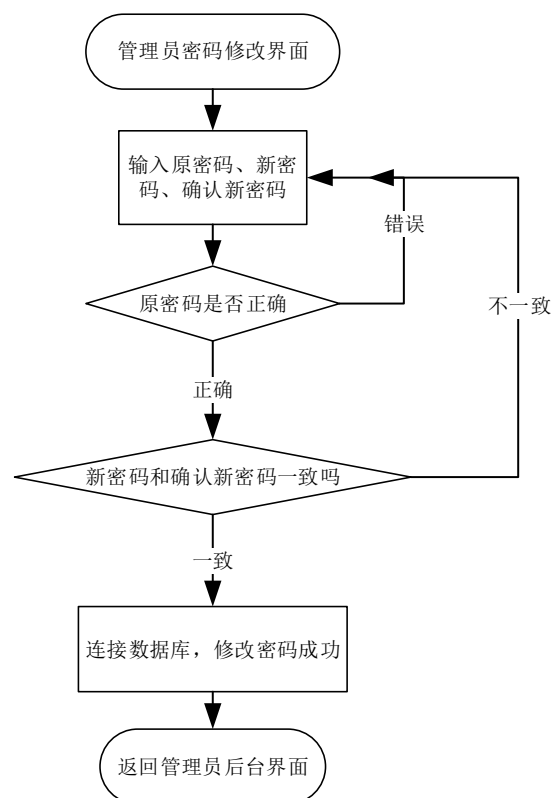


图 4.6 管理员密码修改程序流程图

### 伪代码设计

```

Begin
    进入管理员主界面
    if (管理员点击“修改密码”)
        进入修改密码界面
        输入原密码、新密码、确认新密码
        if (管理员点击“提交”)

```

```
连接数据库
if (原密码正确)
    if (两次新密码一致)
        连接数据库
        修改相应记录
        弹出修改成功提醒
        return Begin
    else
        弹出密码不一致提醒
else
    弹出密码错误提醒
End
```

### 4.8 用户注销登陆模块

该模块用于实现用户在平台上的登录注销服务，具体如下 PAD 图所示：

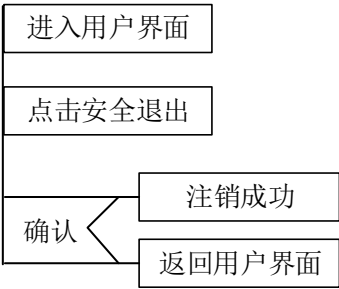


图 4.7 用户注销登录状态模块 PAD 图

### 伪代码设计

```
Begin
    进入用户主界面
    用户点击“退出登录”
    清除用户登录状态
    返回身份选择界面
End
```

### 4.8 用户添加联系人模块

该模块用于实现用户在平台上添加随行的联系人，完善联系人的各种信息，具体如下 PAD 图所示：

## 伪代码设计

```
Begin
进入用户主界面
用户点击添加联系人
IF（用户点击==添加联系人）
    显示要添加联系人的信息
    用户输入联系人信息
用户点击继续添加联系人或者添加完成
IF（用户点击==继续添加联系人）
    继续显示要添加联系人的信息
    用户输入新的联系人信息
ELSE
    添加完成
返回用户主界面
End
```

## 5 系统实现

### 5.1 开发平台和开发环境介绍

开发平台：微信开发者工具+云平台

开发环境：云开发

(1) 微信开发者工作是微信官方提供的针对微信小程序的开发工具，集中了开发，调试，预览，上传等功能。微信团队发布了微信小程序开发者工具、微信小程序开发文档和微信小程序设计指南，全新的开发者工具，集成了开发调试、代码编辑及程序发布等功能，帮助开发者简单和高效地开发微信小程序。启动工具时，开发者需要使用已在后台绑定成功的微信号扫描二维码登录，后续所有的操作都会基于这个微信的帐号。

程序调试主要有三大功能区：模拟器、调试工具和小程序操作区。

模拟器：模拟器模拟微信小程序在客户端真实的逻辑表现，对于绝大部分



的 API 均能够在模拟器上呈现出正确的状态。

调试工具：调试工具分为 6 大功能模块：Wxml、Console、Sources、Network、Appdata、Storage 以及 WxmlPannel 用于帮助开发者开发 Wxml 转化后的界面。在这里可以看到真实的页面结构以及结构对应的 wxss 属性，同时可以通过修改对应 wxss 属性，在模拟器中实时看到修改的情况。通过调试模块左上角的选择器，还可以快速找到页面中组件对应的 wxml 代码。Sources Pannel 用于显示当前项目的脚本文件，同浏览器开发不同，微信小程序框架会对脚本文件进行编译的工作，所以在 Sources Pannel 中开发者看到的文件是经过处理之后的脚本文件，开发者的代码都会被包裹在 define 函数中，并且对于 Page 代码，在尾部会有 require 的主动调用。Netwrok Pannle 用于观察和显示 request 和 socket 的请求情况。Appdata Pannel 用于显示当前项目当前时刻 appdata 具体数据，实时地反馈项目数据情况，可以在此处编辑数据，并及时地反馈到界面上。Storage Pannel 用于显示当前项目的使用 wx.setStorage 或者 wx.setStorageSync 后的数据存储情况。Console Pannel 有两大功能：开发者可以在此输入和调试代码以及微信小程序的错误输出。

小程序操作区：微信小程序操作区帮助开发者模拟一些客户端的环境操作。例如当用户从微信小程序中回到聊天窗口，会触发一个微信小程序被设置为后台的 api。

运行环境：

## (2) 云开发介绍

开发者可以使用云开发开发微信小程序、小游戏，无需搭建服务器，即可使用云端能力。云开发为开发者提供完整的原生云端支持和微信服务支持，弱化后端和运维概念，无需搭建服务器，使用警欸提供的 API 进行核心业务开发，即可实现快速上线，同时这一能力，同开发者已经使用的云服务相互兼容，并不互斥。

开发者可以使用云开发开发微信小程序、小游戏，无需搭建服务器，即可使用云端能力。云开发为开发者提供完整的原生云端支持和微信服务支持，弱化后端和运维概念，无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现快速上线和迭代，同时这一能力，同开发者已经使用的云服务相互兼容，

并不互斥。

### 编码

编码的任务是为每个模块编写程序，也就是说将详细设计的结果转换成用某种程序设计语言写的程序。编码阶段结束时交付的是不含有语法错误的程序和有关程序说明的“内部文档”。

我们用来编码的是 JS 来完成，但是结构和样式采用分别采用 WXML——是微信设计的一套标签语言，他和 HTML 类似，全名叫：Weixin Markup Language，WXSS——是一套样式语言，用于定义样式 和 CSS 类似，被认是 CSS 的子集 全名：Weixin Style Sheets。

## 5.2 前台界面与后台云数据库连接代码

在开发机票管理系统时，我们使用的是云开发数据库。其数据库连接部分主要在前端界面绑定事件，在 js 文件中定义变量连接到数据库。样例：

- ① 前端页面绑定一个事件 `getData`

```
<button type="primary" bindtap="getData">点击获取数据</button>
```

- ② 在 js 文件中写 `getData` 函数，测试是否绑定成功

```
getData() {  
    console.log(1111);  
}
```

- ③ 在 js 文件定义一个变量 `db` 连接总体的数据库

```
const db=wx.cloud.database()
```

- ④ 在 `getData` 方法中调用 `db`，连接我们名字为“demolist”的数据库，并且用 `get` 方法获取数据，在 `get` 中写一个对象，对象中写一个回调函数，打印当前对象，控制台查看数据是否有打印

```
getData() {  
    // 获取相应名字的数据库 用 get 获取该数据库的数据 回调函数  
  
    db.collection("demolist").get({  
        success:res=>{
```

```

        console.log(res)
    }
    })
},

```

⑤ 对象里有回调函数，`.doc` 用来传递数据库的独有 `id` 参数，写在数据库名字和 `get` 方法的中间

⑥ 在 `data` 数据段中定义一个空的变量 `dataObj`

```
dataObj: ""
```

⑦ 在 `getData` 方法中调用 `this.setData` 方法给 `dataObj` 赋值

```

getData() {
    // 获取相应名字的数据库 用 get 获取该数据库的数据 回调函数
    db.collection("demolist").get({
        success: res => {
            this.setData({
                dataObj: res.data
            })
        }
    })
},

```

## 5.3 各模块功能的实现

### 5.3.1 主界面模块实现

建立航空订票管理系统的主界面，该界面又分为用户登录和管理员登陆，选择相应的身份登陆，即可进入机票预定系统的小程序。



图 5.1 主登录界面模块实现效果图

### 5.3.2 用户登录功能实现

该模块用于实现订购机票的用户的登录功能，选择用户登录，对用户登录申请进行授权。完成后点击允许即可登录账号。

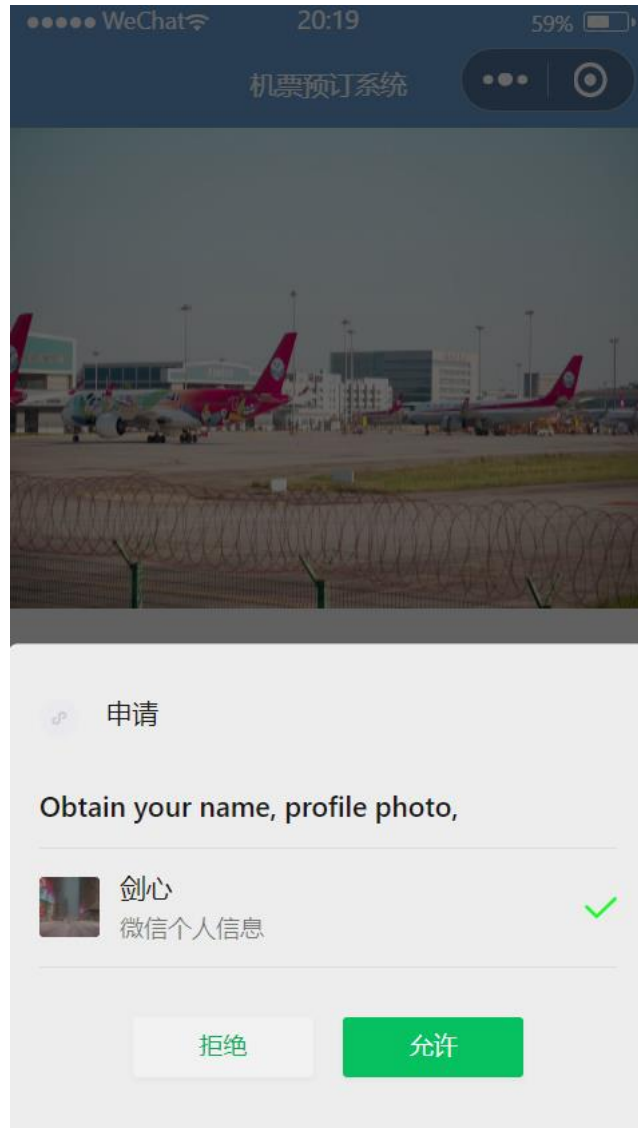


图 5.2 用户通过微信账号授权登录

如果用户选择拒绝，则不能进行登录授权，会返回到开始的主界面



图 5.3 用户取消授权登陆

如图 5.4 所示，成功登录了一个用户名为“剑心”的微信用户，进入用户界面。由于微信小程序只需要微信授权即可，我们并没有设置账号与密码。如图：

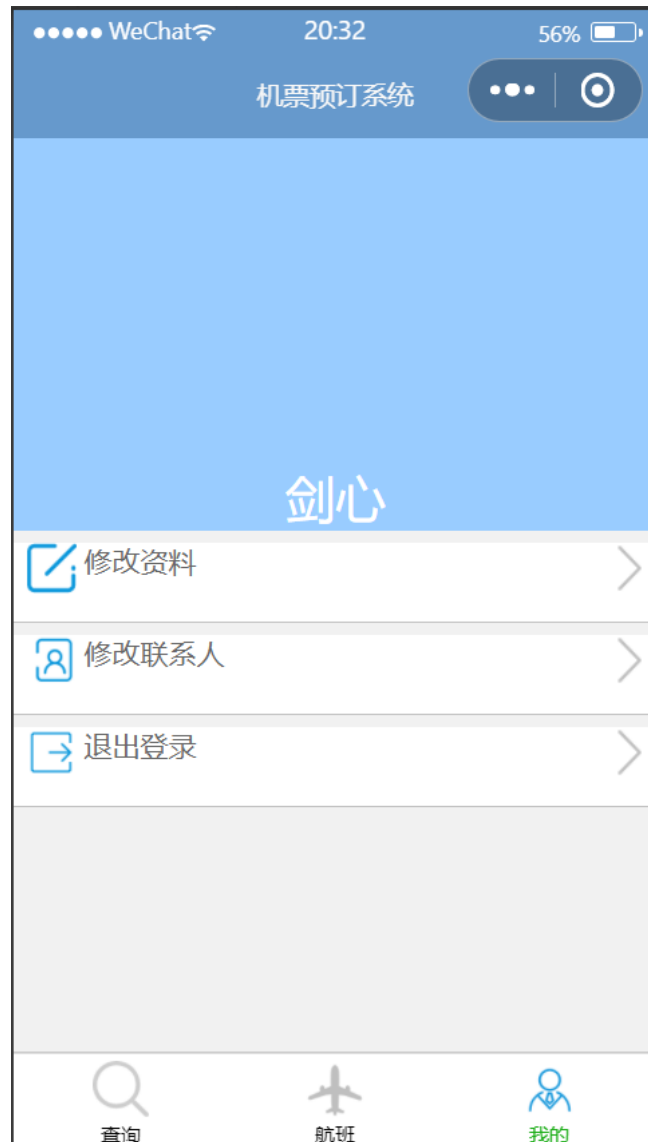


图 5.4 用户登录成功

### 5.3.3 管理员登陆功能实现

登录功能分两部分。一部分是用户登录功能（无需账号和密码），另一部分是管理员登录（需要账号和密码）。进行管理员登陆时，当输入用户名、密码后，与数据库进行校对，验证正确后即可进入。输入时，管理员账号为可现实，密码则用‘.’代替以便保护隐私。如图 5.5 是管理员的登录界面。图 5.6 是管理员登录成功后的界面。图 5.7 是管理员登录失败（账号或密码输入错误）的错误提示。



图 5.5 管理员登陆界面





图 5.6 管理员登录成功界面



图 5.7 管理员登录失败界面

#### 5.3.4 用户完善资料功能实现

用户完善资料功能，即完善用户的个人资料（真是姓名、身份证号、手机号码、邮箱）以便在用户订票时对用户的身份进行确认和订票后对用户的订单和航班信息对用户进行通告。如图为用户“剑心”对个人信息进行完善

..... WeChat 20:52 49%

< 机票预订系统 ...



剑心

 电话: 15031303168

 邮箱: 2451510617@qq.com

 身份证号: 0000000000000000

 姓名: 张建鑫

提交

图 5.8 用户个人信息修改或完善

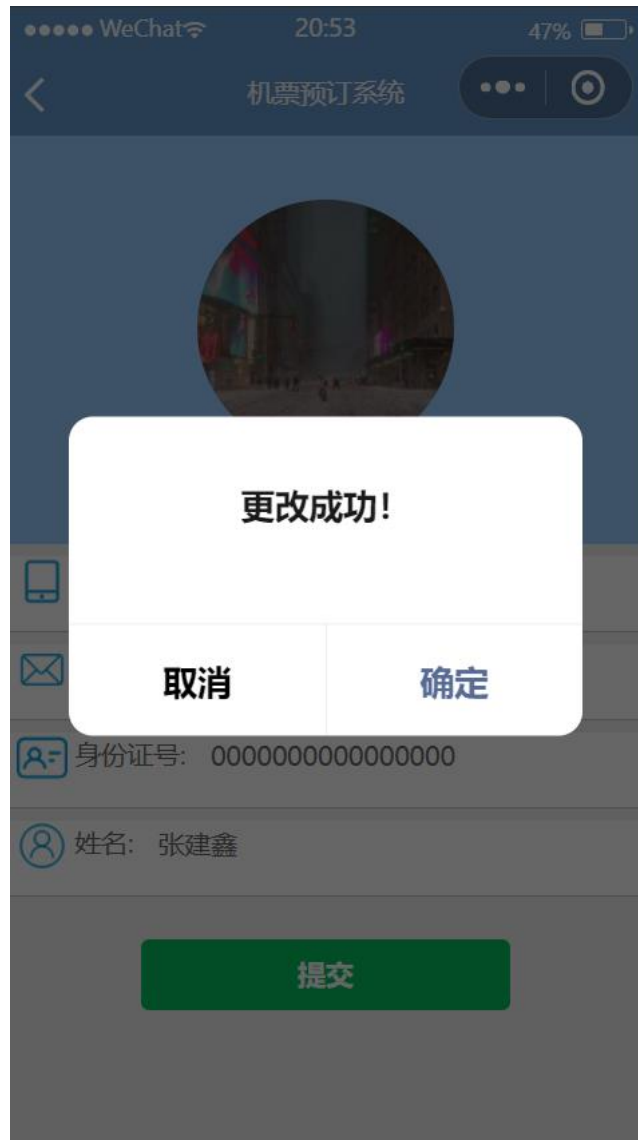


图 5.9 用户信息更改完成

### 5.3.5 添加联系人功能实现

该功能是为实现用户在订机票出行时可以为随行的联系人一起订票，对用户联系人的个人信息（联系人与用户的关系、电话、身份证号、邮箱、真实姓名）的完善，以便对联系人身份进行确认和消息的通知，如图

●●●● WeChat

21:00

46%

<

机票预订系统

⋮ | ⦿

我的联系人

👤

关 系: 朋友

☎

电 话: 15397012411

✉

邮 箱: 1620518952@qq.com

🆔

身份证号: 000000000000000000

👤

姓 名: 施展

提交

图 5.10 用户添加联系人信息



5.11 联系人添加成功

### 5.3.6 用户订票功能实现

该功能是实现用户的订票功能，用户可以根据将要出行的出发地、目的地、出发时间对航班信息进行查询，如图，图为查询结果，显示该航班的信息（出发时间、到达时间、航班编号、经济舱价格、经济舱票数量、头等舱价格头等舱数量等）



图 5.12 用户查询航班



图 5.13 满足用户要求航班信息

点击想要预定的航班即可进入订票界面，勾选相应的信息，如图



●●●● WeChat 21:14 41%

< 机票预订系统

● 航班信息

2021年11月27日 12:42  
航班号: MK780  
杭州---->北京  
票价: 经济舱 450 | 头等舱 750  
\*预计支付成功后1分钟内出票  
\*免费托运行李20公斤  
\*提供行程单

● 请选择机票类型

经济舱 ☒  
头等舱 ☐

● 请选择乘机人

☒ 施展--0\*\*\*\*\*0

● 请填写联系人

姓名: 张建鑫  
电话: 15031303168

● 保险

意外险 ¥30 ☒  
飞行有保障, 最高赔付260万元份

图 5.14 用户填写订单信息

填写完乘机信息后, 点击提交即可完成对该列航班的预定, 显示出订票所需的总金额如图:



图 5.15 用户订票完成

订票成功后点击航班，即可查看本次订票产生的订单，如图：



图 5.16 用户订单详细信息

### 5.3.7 机票改签功能实现

该功能让用户重新选择机票购买，通过输入信息查询所要改签的机票。如图 5.15 所示是改签机票界面，图 5.16 所示是改签成功显示机票信息的界面。可以看到信息已经发生更改。

### 5.3.8 用户退票功能实现

### 5.3.9 后台修改航班功能实现

该功能使管理员修改航班信息，如航班名称，出发时间，到达时间等。在管理员登陆后，点击航班界面即可查询所有的航班，点击所要修改的航班即可进入修改界面。如图 5. 是航班界面。如图 5. 所示是选择编号为 MK780 航班点击后的进入的修改界面。如图 5.25 所示是将编号为 MK780 的航班头等舱机票价格从 750 改为了 800

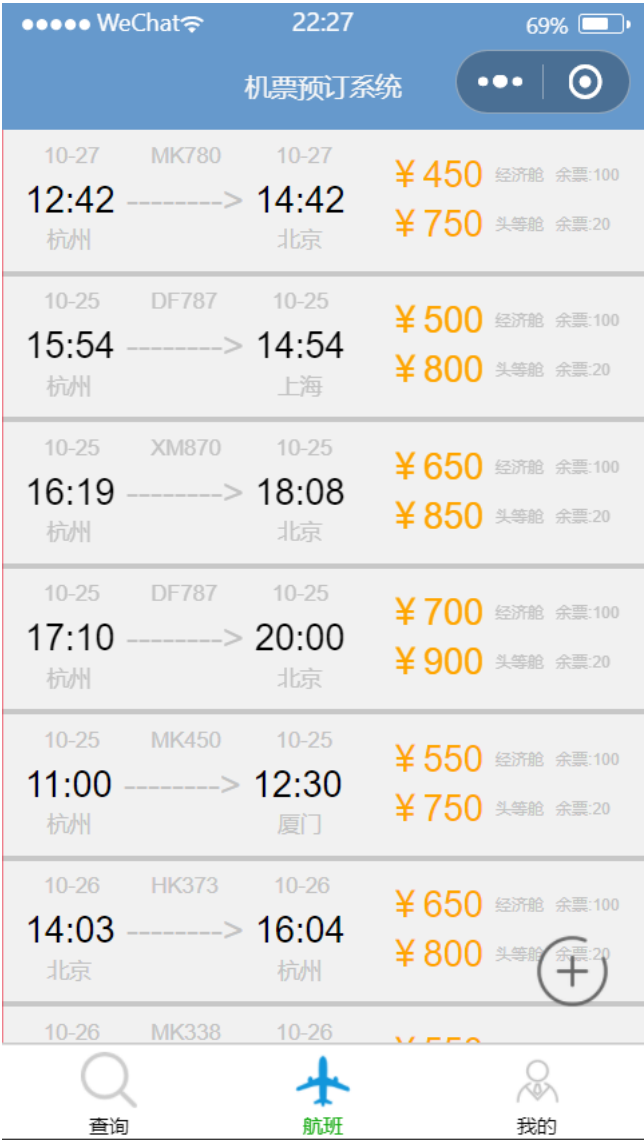


图 5. 管理员端航班信息界面

The screenshot shows a mobile interface for modifying flight information. At the top, there's a status bar with 'WeChat', '22:51', and '77%' battery. Below that is a navigation bar with a back arrow, the title '机票预订系统' (Flight Booking System), and a menu icon. The main content area is titled '航班信息修改' (Flight Information Modification). It contains a form with the following fields:

航班编号:	MK780
出发地:	杭州
目的地:	北京
出发时间:	2021/11/27 12:42:56
到达时间:	2021/11/27 14:42:51
经济舱票价:	450
经济舱总票数:	100
头等舱票价:	800
头等舱总票数:	20

At the bottom of the form is a green button labeled '提交' (Submit).

图 5. 管理员修改航班信息界面



图 5.25 将头等舱机票价格改为 800

### 5.3.10 后台添加航班功能实现

该功能使管理员在后台添加航班信息如航班名称，出发城市，目的城市，票价等。点击右侧加号，即可出现要添加的航班的信息界面，输入完整的信息后点击提交即可完成新航班的添加。如图 5.19 所示是所有航班信息的显示界面，图 5.20 是添加航班信息的界面。如图 5.21 所示添加成功。



●●●● WeChat 23:05 95%

< 机票预订系统 ...

航班信息添加

航班编号:	MK220
出发地:	杭州
目的地:	哈尔滨
出发时间:	Dec Wed 1 2021 10:40:00 GM
到达时间:	Dec Wed 1 2021 14:40:00 GM
经济舱票价:	300
经济舱总票数:	250
头等舱票价:	1000
头等舱总票数:	30

提交

图 5. 添加一个新航班





图 5.21 由编号为 MK220 的航班可知添加成功

### 5.3.11 后台删除航班功能实现

该功能使管理删除需要删除的航班信息，如图 5.22 所示，将要删除的航班信息向左滑动即可出现删除按钮，点击删除即可。如图为删除编号为 MK780 的航班。由图 5.23 所示，删除成功。



图 5.22 选择编号为 MK780 的航班待删除



图 5. 航班删除成功后的界面

### 5.3.12 管理员查询航班功能实现

该功能分为两部分，一部分是后台查询航班，另一部分是用户查询航班。图 5.26 所示是管理员查询出发地为杭州，目的地为哈尔滨的航班信息记录。

### 5.3.13 管理员修改资料功能

管理员在登录账号后，可以实现对个人信息（电话、邮箱）的更改。在管理员主界面点击“修改信息”即可进入修改界面。如图

### 5.3.14 管理员修改密码功能

管理员在必要时可以对登录的密码进行更改，以便起到更加保密的作用，在管理员主界面，点击“密码修改”即可进入密码修改界面。如图

### 5.3.15 退出系统功能

在用户和管理员操作完成后，便可以退出机票订购系统。分别在用户和管理员的主界面，点击“退出登录”即可返回最开始的主界面。如图

## 5.4 测试

### 5.4.1 软件测试的目标

软件测试是指对一个完成了全部或部分功能、模块的计算机程序在正式使用前的检测，以确保该程序能按预定的方式正确地运行。

有如下规则：

- (1) 测试是为了发现程序中的错误而执行程序的过程。
- (2) 好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案
- (3) 成功的测试是发现了至今为止尚未发现的错误的测试。

## 5.4.2 软件测试的准则

- (1) 所有测试都应该能追溯到用户需求
- (2) 应该远在测试开始之前就制定出测试计划
- (3) 把 Pareto 原理应用到软件测试中。Pareto 原理：测试发现的错误中的 80% 很可能是由程序中 20% 的模块造成的
- (4) 应该从“小规模”测试开始，并逐步进行“大规模”测试
- (5) 穷举测试是不可能的  
穷举测试就是把程序所有可能的执行路径都检查一遍的测试
- (6) 为了达到最佳的测试效果，应该由独立的第三方从事测试工作。  
“最佳效果”是指有最大可能性发现错误的测试

## 5.4.3 软件测试的步骤

- (1) 模块测试（单元测试）

把每个模块作为一个单独的实体来测试，通常比较容易设计检验模块正确性的测试方案。发现的通常是编码和详细设计的错误。

- (2) 子系统测试（集成测试 1）

把经过单元测试的模块放在一起形成一个子系统来测试，着重测试模块接口问题。

- (3) 系统测试（集成测试 2）

把经过测试的子系统装配成一个完整的系统来测试。发现的通常是设计中的错误，也可能发现需求说明中的错误。

- (4) 验收测试

也称确认测试或交付测试，把软件系统作为单一的实体进行测试，使用实际数据（系统将来要处理的信息）进行测试，通常用户来做。发现的通常是需求说明书中的错误。

- (5) 平行运行

同时运行新开发出来的系统和将被它取代的旧系统，以便比较新旧两个系统的处理结果，这样做的目的：

- (1) 可以在准生产环境中运行新系统而又不冒风险
- (2) 用户能有一段熟悉新系统的时间
- (3) 可以验证用户指南和使用手册之类的文档
- (4) 能够以准生产模式对新系统进行全负荷测试，可以用测试结果验证性能指标

#### 5.4.4 单元/集成/验收测试

##### 单元测试（模块测试）

单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

单元测试的目的： 在于发现各模块内部可能存在的各种错误，主要是基于白盒测试。

- a) 验证代码是与设计相符合的；
- b) 发现设计和需求中存在的错误；
- c) 发现在编码过程中引入的错误。

单元测试策略： 孤立的测试策略、自顶向下、自底向上的单元测试策略

##### 1) 孤立的测试策略：

- 方法：不考虑每个模块与其他模块之间的关系，为每个模块设计桩模块和驱动模块。每个模块进行独立的单元测试。

- 优点：该方法是最简单，最容易操作的。可以达到高的结构覆盖率。该方法是纯粹的单元测试。

- 缺点：桩函数和驱动函数工作量很大，效率低。

##### 2) 自顶向下的单元测试策略：

- 方法：先对最顶层的单元进行测试，把顶层所调用的单元做成桩模块。其次对第二层进行测试，使用上面已测试的单元做驱动模块。如此类推直到测试完所有模块。

- 优点：可以节省驱动函数的开发工作量，测试效率较高。
- 缺点：随着被测单元一个一个被加入，测试过程将变得越来越复杂，并且

开发和维护的成本将增加。

### 3) 自底向上的单元测试策略:

- 方法: 先对模块调用层次图上最低层的模块进行单元测试, 模拟调用该模块的模块做驱动模块。然后再对上面一层做单元测试, 用下面已被测试过的模块做桩模块。以此类推, 直到测试完所有模块。

- 优点: 可以节省桩函数的开发工作量, 测试效率较高。 缺点: 不是纯粹的单元测试, 底层函数的测试质量对上层函数的测试将产生很大的影响。

单元测试是由程序员自己来完成, 最终受益的也是程序员自己。可以这么说, 程序员有责任编写功能代码, 同时也就有责任为自己的代码编写单元测试。执行单元测试, 就是为了证明这段代码的行为和我们期望的一致。

## 集成测试

集成测试(也叫组装测试, 联合测试)是单元测试的逻辑扩展。它的最简单的形式是: 两个已经测试过的单元组合成一个组件, 并且测试它们之间的接口。从这一层意义上讲, 组件是指多个单元的集成聚合。在现实方案中, 许多单元组合成组件, 而这些组件又聚合成程序的更大部分。方法是测试片段的组合, 并最终扩展进程, 将您的模块与其他组的模块一起测试。最后, 将构成进程的所有模块一起测试。集成测试主要目的是针对详细设计中可能存在的问题, 尤其是检查各单元与其它程序部分之间的接口上可能存在的错误。

### 集成测试策略:

1) 大爆炸集成 2) 自顶向下集成 3) 自底向上集成 4) 三明治(混合式)集成 5) 基于集成 6) 分层集成 7) 基于功能的集成 8) 基于消息的集成 9) 基于进度的集成 10) 基于风险的集成

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。系统测试主要针对概要设计, 检查了系统作为一个整体是否有效地得到运行, 例如在产品设置中是否达到了预期的高性能。

## 验收测试

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪, 并且可以让最终用户将其用于执行软件的既定功能和任务。验收测试

是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。

5.4.5 测试过程

本系统采取以白盒测试方案为主，黑盒测试方案为辅的测试方法。先分别进行单元测试，再进行集成测试，遵循自底向上的策略。对于每一个测试用例都有考虑到合法和非法的测试情况。

5.4.5.1 单元测试（黑盒测试）

黑盒测试又称为功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。测试人员一般把被测程序当作一个黑盒子。但是黑盒测试并不能取代白盒测试，它是与白盒测试互补的测试方法，它很可能发现白盒测试不易发现的其他类型的错误。

黑盒测试主要测到的错误类型有：不正确或遗漏的功能；接口、界面错误；性能错误；数据结构或外部数据访问错误；初始化或终止条件错误等等。

常用的黑盒测试方法有：等价类划分法；边界值分析法；因果图法；场景法；正交实验设计法；判定表驱动分析法；错误推测法；功能图分析法。

下面对各个模块进行测试如下：

用户注册模块测试：

表 5.1 “注册模块” 输入条件等价类表

输入等价类	有效等价类	无效等价类
注册信息	1. 无空值且符合各项信息的输入要求	1. 重复的用户名 2. 不一致的密码 3. 无规范的电子邮件 4. 存在空值

表 5.2 有效等价类测试用例表



测试数据	期望结果	覆盖的有效等价类
用户类型用户名(test)、密码(111)、电子邮件(123@qq.com)	输入有效	1

表 5.3 无效等价类测试用例表

测试数据	期望结果	覆盖的无效等价类
用户名(test)、密码()、确认密码、电子邮件()	空值	4
用户名(test)、密码(qwer)、确认密码(123) 电子邮件(15@qq.com)	密码不一致	2
用户名(test)、密码(qwer1234)、确认密码(qwer1234) 电子邮件(84888324)	已注册的用户名	1

用户登录模块测试:

表 5.4 “登录模块” 输入条件等价类表

输入等价类	有效等价类	无效等价类
登录信息	1. 无空值且符合各项信息的输入要求	1. 无效用户名 2. 无效密码 3. 存在空值

表 5.5 有效等价类测试用例表

测试数据	期望结果	覆盖的有效等价类
用户名(test)、密码(qwer1234)	输入有效	1

表 5.6 无效等价类测试用例表

测试数据	期望结果	覆盖的无效等价类
用户名(liuzerun)、密码(2)	无效输入	1
用户名(liuzerun123)、密码(qwer)	无效输入	2
用户名()、密码(123456)	无效输入	3

#### 5.4.5.2 集成测试（白盒测试）

设计测试方案是测试阶段的关键技术问题。所谓测试方案包括具体的测试目的，应该输入的测试数据和预取的结果。由于时间关系，小组对系统进行了部分功能的白盒测试。下面使用白盒测试方法对本系统进行测试。白盒测试包括：语句覆盖、判断覆盖、条件覆盖、判断/条件覆盖、条件组合覆盖、点覆盖、边覆盖、路径覆盖等。下面对本系统的几个模块进行测试：

##### **登录模块的测试：**

- ① 选择用户类型，输入账号、密码，系统进行匹配。
- ② 密码错误时，刷新页面，提醒用户密码需要重新输入。
- ③ 账号不存在时，刷新页面提醒用户账号需重新输入。

##### **测试路径：**

路径 1：1 ->2->3->4->6->7

路径 2：1->2->3->4->6->8

路径 3：1->2->3->4->5

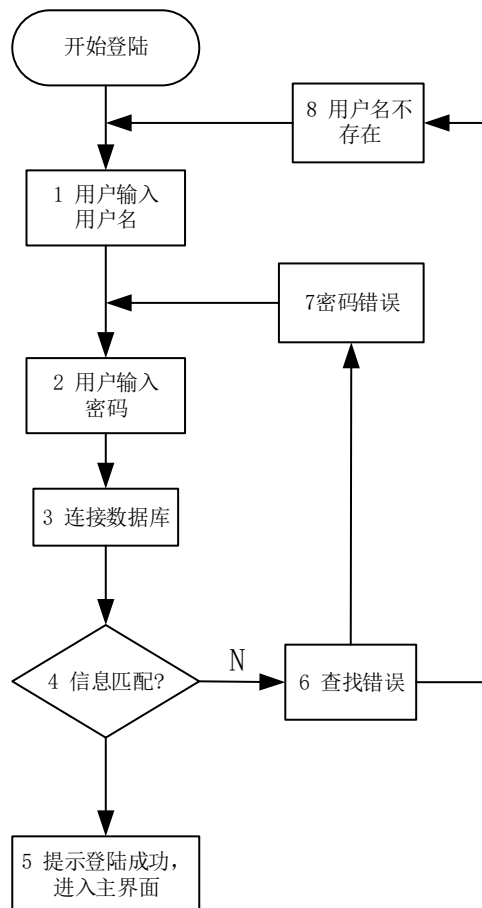


图 5.28 登录模块白盒测试实现效果图

### 机票订购模块的测试:

- ① 用户搜索航班
- ② 用户选择机票舱位与数量
- ③ 用户确认购买后，在个人信息界面查看是否购买成功

语句与判断与条件覆盖: 1 → 2 → 4 → 3 → 5 → 6 → 7 → 5 → 8

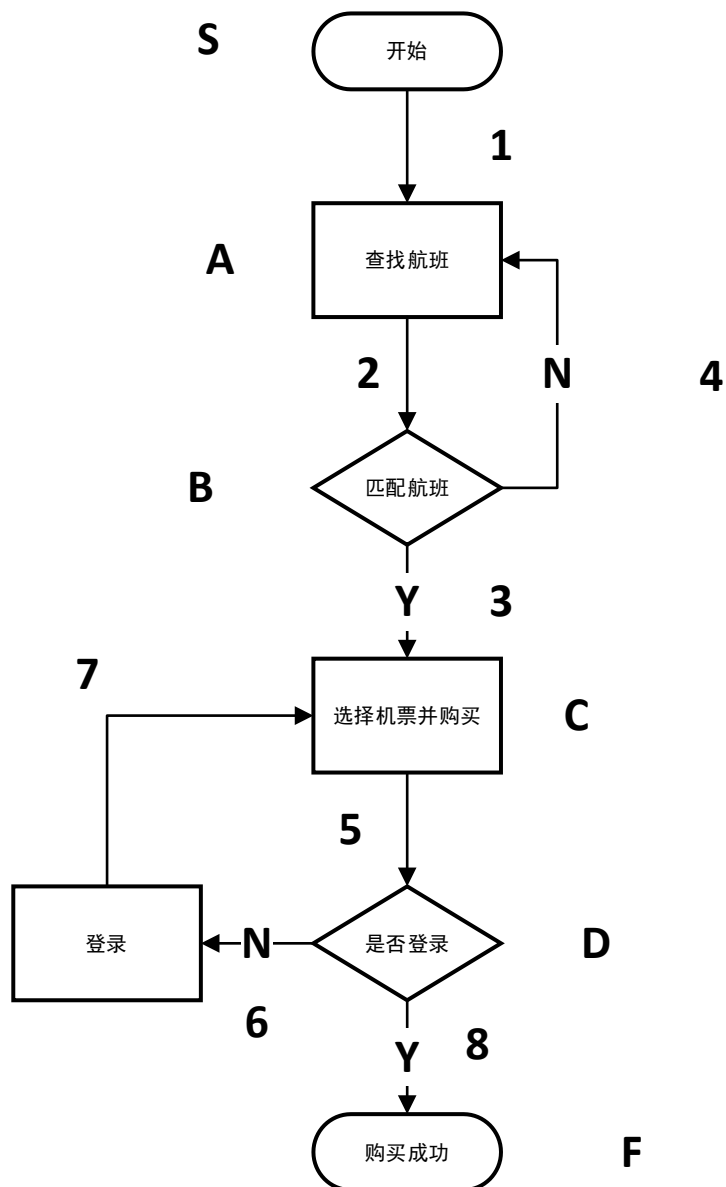


图 5.29 订购机票的流程

#### 5.4.5.3 系统测试与回归测试

完成了各个模块的测试后，要进行系统测试。系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符合或与之矛盾的地方。在系统测试的任何一个阶段，只要发现了错误，就要尽可能及时更正。更正后还要检验已经发现的缺陷有没有被正确的修改和修改过程中有没有引发新的缺陷，即回归测试。另外，每当一个新的模块被当作集成测试的一部分加进来的时候，软件环境都很发生改变，即建立起新的数据流路径，还有可能激活了新的控制逻辑。这些改变可能会使原本工作得很正常的功能产生错

误。因此在集成测试策略的环境中，要进行回归测试，就是对部分已通过测试的功能要再次进行测试，以保证系统在新环境下能正常工作。

#### 5.4.5.4 系统测试结论

经过各个模块功能，系统测试，黑盒中的边界测试，利用等价类划分得到的测试数据，容易出错的位置分析，白盒测试中路径覆盖，语句覆盖，判定覆盖，条件覆盖，以及各个小组成员的代码审查，以及要求其他同学对系统进行使用后，需求分析中所要实现的功能基本完成除了这些功能有些设计的体验不是很好之外。最终该系统成功通过测试验收。

## 6 维护

在软件产品被开发出来并交付用户使用之后，就进入了软件的运行维护阶段。这个阶段是软件生命周期的最后一个阶段。

为了清除系统运行中发生的故障和错误，产品维护人员要对软件进行必要的修改与完善；为了使软件产品适应用户环境的变化，满足新提出的需要，也要对原软件产品做些局部的更新，这些工作称为软件的维护。其任务是改正软件系统在使用过程中发现的隐含错误，扩充在使用过程中用户提出的新的功能及性能要求，其目的是维护软件系统的“正常运作”。

### 6.1 系统维护过程

系统维护工作在整个系统生命周期中常常被忽视。人们往往热衷于系统开发，当开发工作完成以后，多数情况下开发队伍被解散或撤走，而在系统开始运行后并没有配置适当的系统维护人员。这样，一旦系统发生问题或环境发生变化，最终用户将无从下手，这就是为什么有些信息系统在运行环境中长期与旧系统并行运行不能转换，甚至最后被废弃的原因。

为了有效避免以上阐述的情况。我们在设计机票预订系统的时候主要从以下两个方面做好维护：

- 1) 对代码进行注释，便于开发维护人员理解
- 2) 提供说明文档，说明了开发、维护、更新的内容

每次在发布更新的时候，我们都要在说明文档中单独标记出更新的内容以方便使用者，让使用者更好的理解和使用新版本的软件。

### 6.2 系统维护策略

机票预订系统依赖于数据库数据的实时更新，一旦数据库的数据没有及时更新到每个对应的表或者出现触发器失效的情况，应该立即启用应急的措施。具体如下：

- 1) 建立一个备用数据库，在主数据库失灵的时候可以使用备用数据来保证系统

的正确运行。

- 2) 备份数据库使用云备份
- 3) 软件在运行的时候每天自动从数据库中调用各项数据并保存在一个临时的文档中，每个小时都替换更新一次。
- 4) 每隔一周的时间会将云备份的数据转存到磁盘等固定设备上。

## 7 体会

### 心得体会 1:

我们这次的软工课程大实验是设计一个航空机票系统。通过本次大实验熟悉了开发一个系统的完整流程，从前端页面的设计，后端数据的操作以及与前端页面的交互，到最后 Web 项目的打包与部署，对今后从事开发相关的工作有一定益处。想要做好一个项目的开发，我有一下几点感悟。首先一定要进行充分的设计分析，当打好一个可靠的开头，接着做下去才可以避免许多推导重来。做好设计阶段的重要性就犹如房子打好地基一样重要。如果一开始许多逻辑思考错误，到系统实现时才发现，那这将是一场悲剧。其次，团队需要有良好的分工与配合。当各个成员分工明确，任务分配的合理，这些都大大提高团队开发效率，不至于因为各个人进度不同而影响自己的任务推进。以及，软件开发工程师需要熟练掌握相应任务所需的基本知识。因为我们并未学习过相关的前后端开发知识，同时没有充足的时间学习，导致我们实现系统的功能十分痛苦，犹如瞎子走路，强烈建议学院的负责人科学策划专业培养方案，教师团队认真准备课程作业的布置。不要让教育事业陷入恶性循环，教师不负责，学生也因这种环境，到将来成为教师后也变得漠不关心。同时还需要注意一些技巧。比如设置一个良好的进度表，有利于任务的推进，目标清晰并且明确。在系统设计时要考虑将来可能会增加什么需求，将系统的可扩展性设计的泛化一些。

### 心得体会 2:

此次实验我们小组选择的是完成机票预订系统的项目实现。因为我们并没有上过前后端相关的课程，因此前期花费了较多的时间研究学习了相关的知识。这也导致真正开始项目开发时时间较为紧迫，最终得到的系统仍有许多需要改进的地方。实验一开始我们习惯性的将重点放在编程实现上，而没有很好的完成前期的需求分析中的内容，这导致实现过程中我们对前面写的不够详细的地方出现了不同的理解并且不断发现缺失了个别功能。再重新梳理后，我们完善了这块内容，避免了后续的开发发生更大的问题。这让我们更深刻的认识到了软件逐步进行开发的重要性，并对软工的理论课程知识有了更深刻的理解。除此之外，我也认识



到了良好的团队合作以及合理的任务分配能极大的加快项目推进的速度。这次实验还让我认识到了文档的规范编写也需要花费许多精力，并能很好的促进项目规划，增强了小组之间的共识。

通过本次大实验我基本熟悉了一个系统开发的流程，锻炼了自己的动手能力、实践能力和团队合作能力，也学习到了 JS、SSH 等方面的知识，并复习了数据库的知识以及增加了 visio 的熟练度。

### **心得体会 3:**

这次我们小组软件工程课程大实验做的是一个机票预订系统，虽说无法与市面上商用化的系统相比，但对首次接触软件开发的我们来说还是一个不小的工程。从初期的可行性分析规划，对我们的机票预订系统所能提供的功能进行构想，到后续的需求分析、总体设计、详细设计，再到系统的编码实现和测试环节，我们把开发一个软件所需经历的流程完完整整地体会了一遍，虽然系统规模没有日常使用的主流软件那么大，但是软件规划设计的每一个步骤我们都认真执行了。平时理论课上抽象难懂的各种过程、各种分析图，在课程大实验实际设计、操作的过程中得以豁然开朗，在绘图的过程中我们得以了解到什么图适合什么样的分析场合，对课本所讲述的软件工程内容有了初步的体会，真可谓实践出真知啊。在机票预订系统设计的过程中，最困难的还是编码实现这一环节，这是我们首次参与完整的前后端 Web 编程开发，对用到的 SSH 框架也是从头学起。通过机票预定系统的开发，我对 Web 项目的开发流程有了初步的了解，对于系统开发而言，做好稳固的前期规划确实十分重要，我们在编码过程中，出现过发现前期规划不到位的情况，导致我们花费更多的时间来完善前期规划，调整编码实现。总而言之，此次软件工程的大作业是一次十分锻炼人的机会，我们在进行大实验的过程中既提升了个人技术，也提升了团队协作能力，受益匪浅。

### **心得体会 4:**

本次软件工程专业大作业我们组的选题为机票预订系统，我们选择用 WEB 框架进行实现。在这个过程中遇到了许多困难，比如由于是计科专业，平常在前端方面没有知识积累，导致编码耗费时间较长，学习过程中遇到很多困难；同样在服务

器部署方面由于缺乏经验，也是耗费了很大的精力才得以实现。本次大作业很好地锻炼了我的编码能力，在这个过程中也得以学到很多知识，并且对课堂上学习的软件工程方法有了更深刻、更为实际的理解，除此之外，在此次大作业中我也认识到开发一个庞大的软件，需要优秀的团队合作，明确合理的分工，以及扎实的技术基础。另外，对于机票预订系统软件来说，相对于一些更大的工程，软件工程方法学对其作用有限，不需要进行十分繁琐的分析便可以进入编码的环节，希望在以后的学习过程中能够接触到更多更大的工程项目，也许能够对软件工程方法学会有更加深刻的体会。