



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

软件工程实验报告

软件设计-机票预定系统

2011763 黄天昊

年级：2020 级

专业：计算机科学与技术

指导教师：李起成

2023 年 4 月 7 日

摘要

机票预定系统是一种常见的软件系统，它允许用户在预订机票时进行搜索，比较航班和价格，并最终选择他们喜欢的选项并完成预订。机票预定系统的优点包括方便快捷、节省时间和成本，用户可以根据自己的需求和预算自主选择机票和支付方式。然而，机票预定系统也存在一些缺点，如航班信息和价格的实时性问题、用户个人信息和支付安全问题等，需要系统开发者和用户共同努力来解决。

软件设计的目的是为了确保开发的软件系统能够满足用户需求、具有可维护性、可扩展性、可靠性和安全性等特性。软件设计可以将系统需求转化为可实现的设计，确定系统的整体结构和各个组件之间的关系，为软件开发提供了指导和框架。

关键字：软件设计；机票预定系统；UML 图；功能需求；系统结构

目录

一、 引言	1
(一) 编写目的	1
(二) 项目背景	1
二、 用例图	1
(一) 主要组成及功能简述	1
1. 主要组成	2
2. 功能简介	2
(二) 用例图实现	2
1. 旅客用例图实现	3
2. 工作人员用例图实现	3
三、 活动图	4
(一) 工作人员活动图	4
(二) 旅客活动图	5
四、 类图	6
(一) 简易类图	6
(二) 详细类图	8
五、 顺序图	10
(一) 旅客预定顺序图	11
(二) 工作人员信息管理顺序图	13
(三) 旅客账户管理顺序图	14
六、 协作图	15
(一) 旅客预约管理协作图	16
(二) 旅客账户管理协作图	16
(三) 工作人员信息管理协作图	17

七、 状态图	18
(一) 机票信息状态图	18
(二) 公共信息服务状态图	18
八、 构件图	19
九、 部署图	20
十、 总结	21

一、 引言

（一） 编写目的

进行机票预订系统软件设计的主要目的是为了创建一个方便快捷的平台，使用户可以在线上购买机票并进行预订。此外，这个系统还可以帮助航空公司和旅行代理商自动化和简化预订和票务管理流程，提高效率和准确性，降低人力成本。

机票预订系统软件设计的目的可以包括以下几点：

- 提供一个方便快捷的平台，使用户可以轻松搜索和比较不同航班和价格，进行预订和购票，并方便地查看订单和票据。
- 管理和处理所有的机票预订和购票，包括收集、存储和处理用户信息、支付信息、航班信息等。
- 自动化票务管理流程，包括航班调度、座位分配、电子票务生成等，提高工作效率和准确性，降低错误和纸质票据的成本和风险。
- 提供数据分析和报告功能，帮助航空公司和旅行代理商了解和分析销售和预订趋势，调整营销策略和价格政策，提高盈利能力。

机票预订系统软件设计的目的是为了提供一个方便、高效和准确的平台，帮助航空公司和旅行代理商提高运营效率和盈利能力，同时为旅客提供更好的购票和旅行体验。

（二） 项目背景

随着互联网的普及和民航业的发展，越来越多的人选择通过在线预订机票来进行旅行。传统的售票方式需要亲自前往航空公司售票处或旅行社，这不仅浪费时间，还可能存在排队等待和信息不准确等问题。因此，机票预定系统应运而生，为旅客提供了一站式服务，包括机票预订、航班查询、机场信息、航班状态查询、退改签等。这个在线旅游服务平台能够快速便捷地满足旅客的需求，同时提高了航空公司的管理和服务水平。因此，该项目的目标是为用户提供方便快捷的机票预订服务，并促进旅游行业的高效发展。

二、 用例图

在软件设计中，用例图是一种描述软件系统功能和用户需求的图形化工具。它主要用于展示系统的用例（即用户场景），以及系统中各个参与者的角色和交互。

用例图能够帮助软件开发团队理解系统的核心功能和用户需求，从而有助于确保系统开发的正确性和完整性。也可以作为软件开发团队与业务人员、用户和其他相关利益相关者之间沟通和交流的一种工具。通过用例图，各方可以更容易地理解系统的功能和交互，并提供反馈和建议。

（一） 主要组成及功能简述

首先我们先对机票预定系统的总体流程进行分析与设计，根据旅客与工作人员的用户特点以及系统任务的分析描述，设计分析系统业务流程、系统的主要组成以及系统所需要实现的概念。

1. 主要组成

机票预定系统通常由以下组成部分构成：

1. 用户界面

这是用户与机票预订系统交互的界面。它包括网站和移动应用程序。用户可以搜索和选择机票，购买机票，管理订单以及查看相关信息，如航班时刻表，机场信息等等。

2. 数据库

该系统需要一个数据库来存储航班信息，航空公司信息，机场信息以及用户信息。在用户预订机票时，系统需要从数据库中检索相关信息，并在用户购票后更新数据库。

3. 订单处理系统

订单处理系统是机票预订系统的核心组件。它处理用户提交的订单并生成订单确认，发票和机票信息。订单处理系统应该能够处理多种支付方式，并且与多家银行或支付平台进行集成。它还应该与预订系统的数据库相连接以及向用户提供订单跟踪功能。

4. 系统管理后台

工作人员可以通过该后台管理系统来添加，删除，更新航班信息、航空公司信息、机场信息以及促销信息等等。该后台系统还可以监控机票预定系统的性能和处理异常，如网络延迟，支付故障等等。

2. 功能简介

机票预定系统根据旅客和工作人员的业务流程的分析，基本需要实现以下功能：

1. 搜索和选择机票

用户可以根据出发地，目的地，日期等等来搜索机票。在搜索结果中，用户可以选择自己满意的机票并进行下一步购买操作。

2. 购票

用户可以在系统中购买机票并选择支付方式。系统应该能够支持多种支付方式，例如信用卡，借记卡，支付宝，微信等等。

3. 订单确认和机票信息

购票后，系统应该发送订单确认信息和机票信息给用户。用户也可以在系统中查看订单和机票信息。也可以将已经确认的订单信息进行打印取出，并以此为凭证登机。

4. 订单跟踪

用户可以在系统中跟踪订单状态，例如机票出票，航班取消等等。

5. 系统管理和监控

管理员可以添加和更新航班信息，航空公司信息，机场信息以及促销信息等等。管理员还可以监控机票预订系统的性能并处理系统异常。

(二) 用例图实现

我们根据不同的用户，也就是旅客和工作人员将用例图的实现转化为两个部分，也就是旅客部分以及工作人员部分。

1. 旅客用例图实现

旅客在机票预定系统中的行为通常包含：旅客通常会输入出发地、目的地、出发时间和乘客数量等信息来搜索适合他们的航班。他们可能会使用过滤器来缩小搜索结果的范围，例如航空公司、价格、起飞时间等。

旅客会浏览搜索结果，比较航班时间、价格、航空公司、航班号、座位等级等信息，并选择最符合他们需求的航班。

旅客完成支付，机票就被预订并生成电子机票。此时，旅客可以选择打印机票或者将机票保存在电子设备上以备查验。

根据上述的分析，实现旅客样例图如下，这里使用的是亿图图示进行绘制：

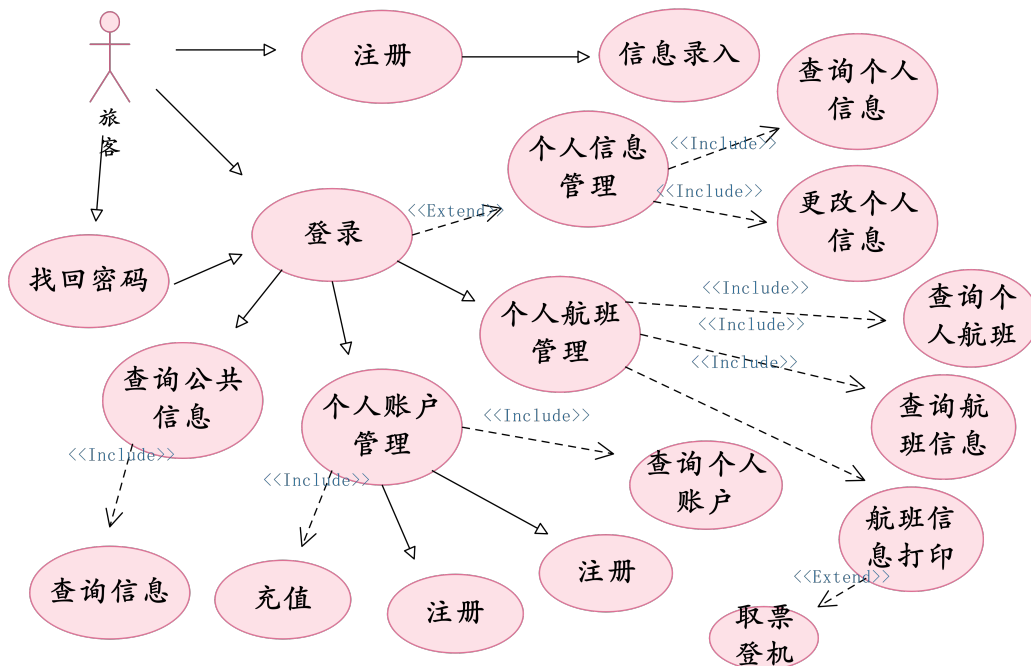


图 1: 旅客样例图

2. 工作人员用例图实现

工作人员在机票预定系统中的行为通常包含：工作人员需要通过机票预定系统搜索符合乘客要求的航班信息，包括出发地、目的地、出发时间、航班时刻、航班舱位等等。在搜索过程中，工作人员需要考虑不同的乘客需求，如价格、航班时刻、舱位等，以便给乘客提供最佳的选择。

工作人员需要为乘客查询机票价格，了解各种舱位的优劣，并为乘客预订机票。在此过程中，工作人员需要熟练掌握机票预定系统的操作流程，以及各类机票的预订规则和操作流程。

一旦工作人员为乘客预订机票，就需要确认订单信息并进行修改。如果出现特殊情况，如航班延误、改变行程等，工作人员需要及时调整订单，以确保乘客的出行计划不受影响。

工作人员需要为乘客进行机票支付，并及时为乘客处理退款等相关问题。在此过程中，工作人员需要严格遵守相关规定，确保乘客的权益得到保障。

工作人员需要与客户保持良好的沟通，及时回复客户的咨询和投诉，并提供优质的客户服务。这可以帮助机票预定系统提高客户满意度和忠诚度，从而提高公司的业绩。

实现工作人员样例图如下：

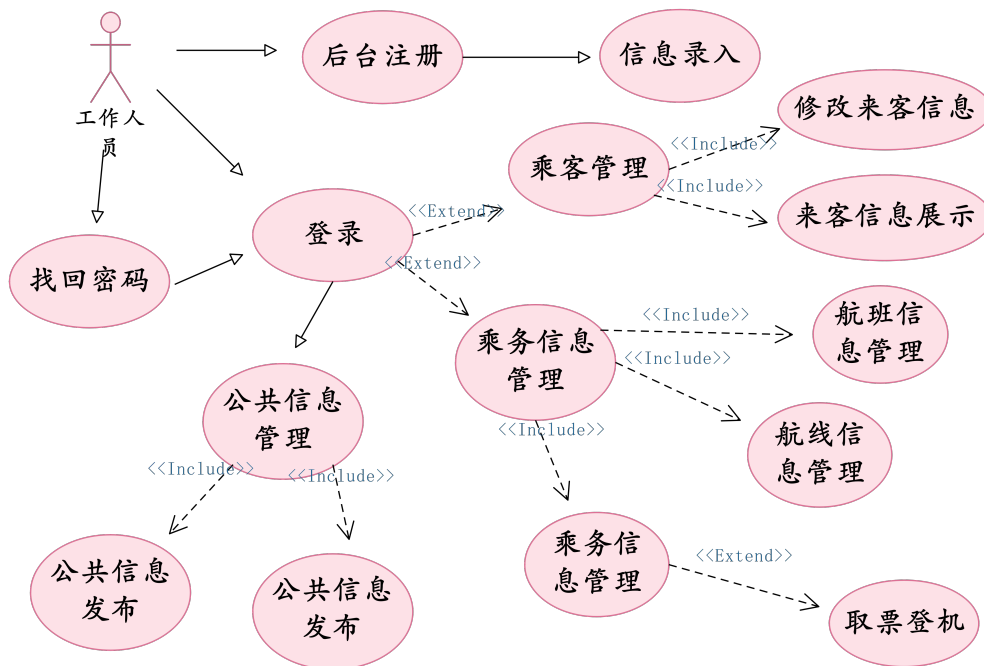


图 2: 工作人员样例图

三、 活动图

活动图是一种用于描述软件系统中的业务流程、用例、场景等的图形化建模工具。活动图的主要功能是帮助软件设计者将系统中的业务流程可视化，并对其进行分析和设计。

活动图通常由一系列的活动节点和控制流组成。活动节点表示系统中的业务活动或任务，而控制流则表示这些活动之间的先后关系和条件约束。通过使用活动图，设计者可以更加清晰地了解系统中的各个业务流程，并通过对控制流的定义，使得这些流程可以按照预期的顺序或条件执行。

活动图通常被用于以下几个方面：

- 描述用例和场景：活动图可以被用于描述软件系统中的用例和场景，以便于更好地理解系统的功能和业务流程。
- 分析和设计：活动图可以帮助设计者对系统中的业务流程进行分析和设计，以便于发现流程中的问题或瓶颈，并对其进行优化。
- 系统实现：活动图可以被用于系统实现的过程中，作为设计文档和开发指导，以便于确保系统的实现与设计的一致性。

活动图是一种非常实用的建模工具，在软件设计的过程中扮演着重要的角色。

(一) 工作人员活动图

在这一部分之中主要进行介绍的是工作人员管理航班旅客信息以及打印展示机票信息等活动的流程分析。

1. 登录系统：工作人员首先需要在系统中登录自己的账号，以便进行后续操作。
2. 工作人员需要对客户的订单进行确认，包括核对客户的信息和机票的信息是否一致，确认订单后系统会生成订单号和电子机票。
3. 工作人员将客户的电子机票发送给客户，并提醒客户注意登机时间和地点。
4. 工作人员根据旅客信息安排航班。
5. 管理员根据机票价格收费，打印机票与账单。
6. 工作人员需要定期维护系统，包括更新航班信息、价格信息等，以保证系统的稳定性和可靠性。

实现工作人员活动图如下：

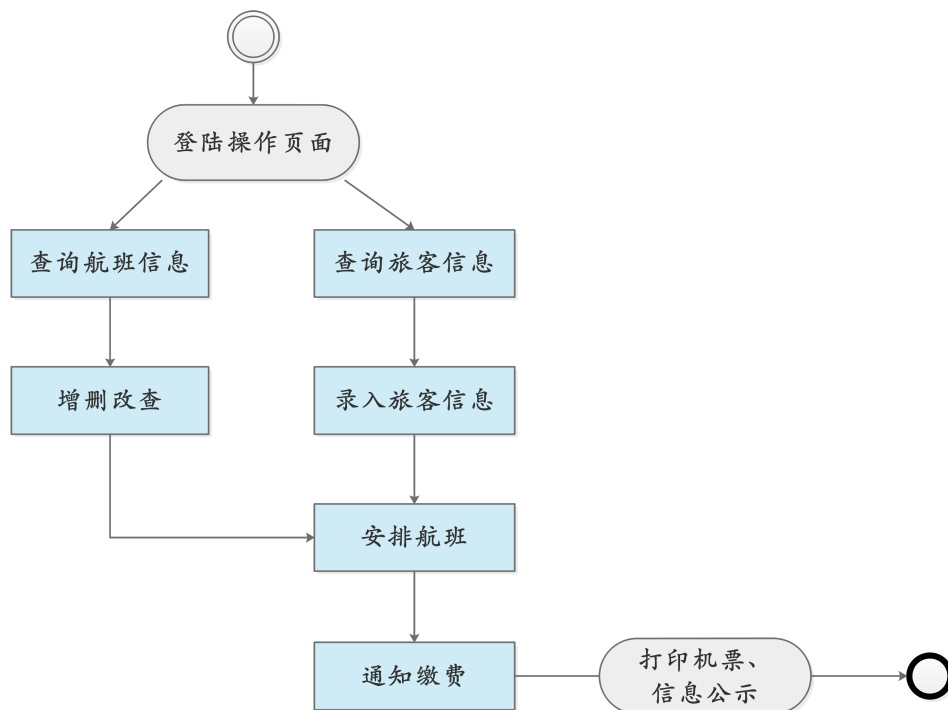


图 3: 工作人员活动图

(二) 旅客活动图

在这一部分之中主要进行介绍的是旅客预约机票，确认订单，缴费以及打印机票信息等活动的流程分析。

1. 输入查询信息：旅客在系统中输入出发地、目的地、出发日期、乘客人数等查询信息。
2. 查询航班信息：系统根据旅客输入的查询信息，查询可用航班，并将结果展示给旅客。
3. 选择航班：旅客根据系统展示的结果，选择自己满意的航班。
4. 输入乘客信息：旅客需要输入自己和同行乘客的个人信息，包括姓名、出生日期、联系方式等。

5. 选择座位：旅客可以根据自己的喜好和需要，选择合适的座位。
6. 确认订单：旅客确认订单信息，并选择支付方式，接着支付订单，旅客使用支付方式进行支付。
7. 接收电子机票：支付成功后，系统将生成电子机票发送给旅客，旅客可下载或打印机票。

根据上述的分析我们可以实现旅客活动图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

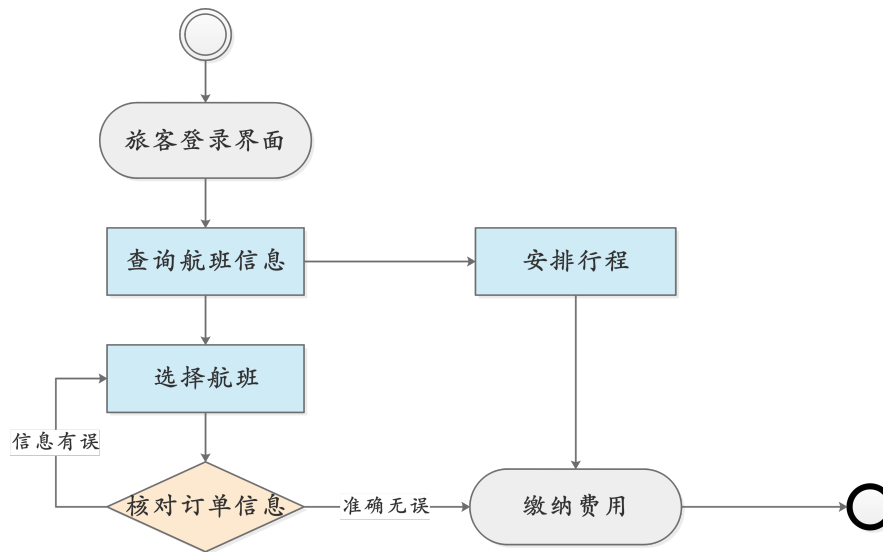


图 4: 旅客活动图

至此我们的机票预定系统的活动图就大致完成了最基础的活动实现环节。

四、 类图

软件设计中的类图是一种用于描述系统中对象和它们之间关系的图形表示法。它是一种用于面向对象分析和设计的工具，可以帮助开发人员更好地理解 and 组织系统中的各个对象，并支持系统设计和实现。

类图中的主要元素是类和关系。类是一种抽象的数据类型，用于表示系统中的对象。每个类都有一个名称和一组属性和方法。属性表示类的状态信息，而方法表示类的行为。

类图是一种重要的软件设计工具，可以帮助开发人员更好地理解系统中的对象和它们之间的关系，支持系统设计和实现，并促进团队协作和沟通。

（一） 简易类图

对于机票预定系统的总体的分析与设计而言，我们首先尝试了实现一个简易的类图，在简易类图的设计之中，我们根据业务需要设计了旅客，工作人员，机票，航班以及预约等五个类，根据具体的实现，以下给出每个类的信息：

- 旅客

该类别包括所有旅客相关的信息，如姓名、性别、年龄、身份证号、护照号码、国籍、联系电话、邮箱地址、预约信息等。

- 工作人员

该类别包括所有工作人员相关的信息，如姓名、工号、职位、联系电话、邮箱地址、部门等，应当拥有管理航班于预约等功能。

- 机票

该类别包括机票的详细信息，如航班号、起飞时间、到达时间、出发地、目的地、座位类型、票价、订票时间、支付状态等。在功能上应该给用户提供打印机票与信息展示的功能。

- 航班

该类别包括航班的详细信息，如航班号、起飞时间、到达时间、出发地、目的地、机型、机场信息等。在功能上应该给用户提供选座与取消选座的功能。

- 预约信息

该类别包括所有预约信息相关的信息，如预约号码、旅客姓名、航班号、座位类型、订票时间、支付状态等。类似于数据库的设计，在预约时，若已经进行了航班的确认，那么应当提供支付途径。

实现简易类图如下：

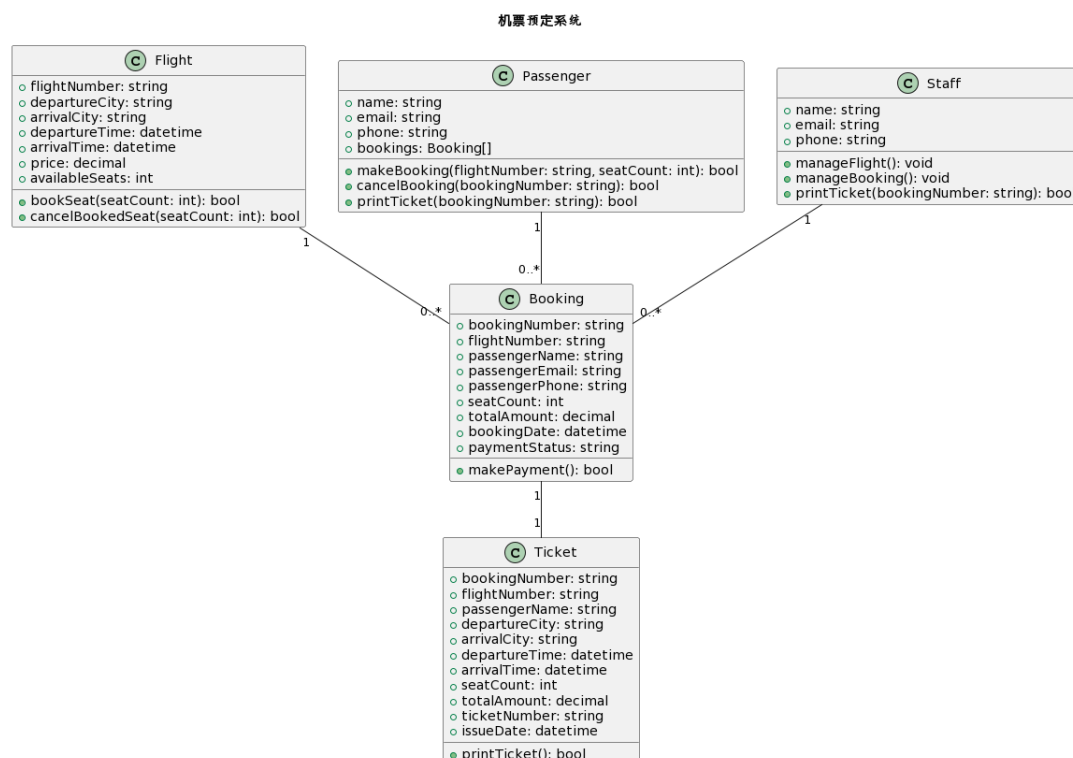


图 5: 简易类图

(二) 详细类图

我们可以在简易类图的基础之上，进行细化的设计，也就是补充一些其它的信息，为机票预定系统补充额外的功能。

在简易类图设计的基础之上，我们继续补充设计：

1. 系统维护：主要包含维护数据库与系统应用程序。
2. 财务信息：账户余额，支付，充值信息。
3. 用户信息：乘客类型，登录注册等内容。

根据补充的分析设计后我们可以实现详细类图如下，这里使用的是 PlantUML 的线上工具进行绘制：

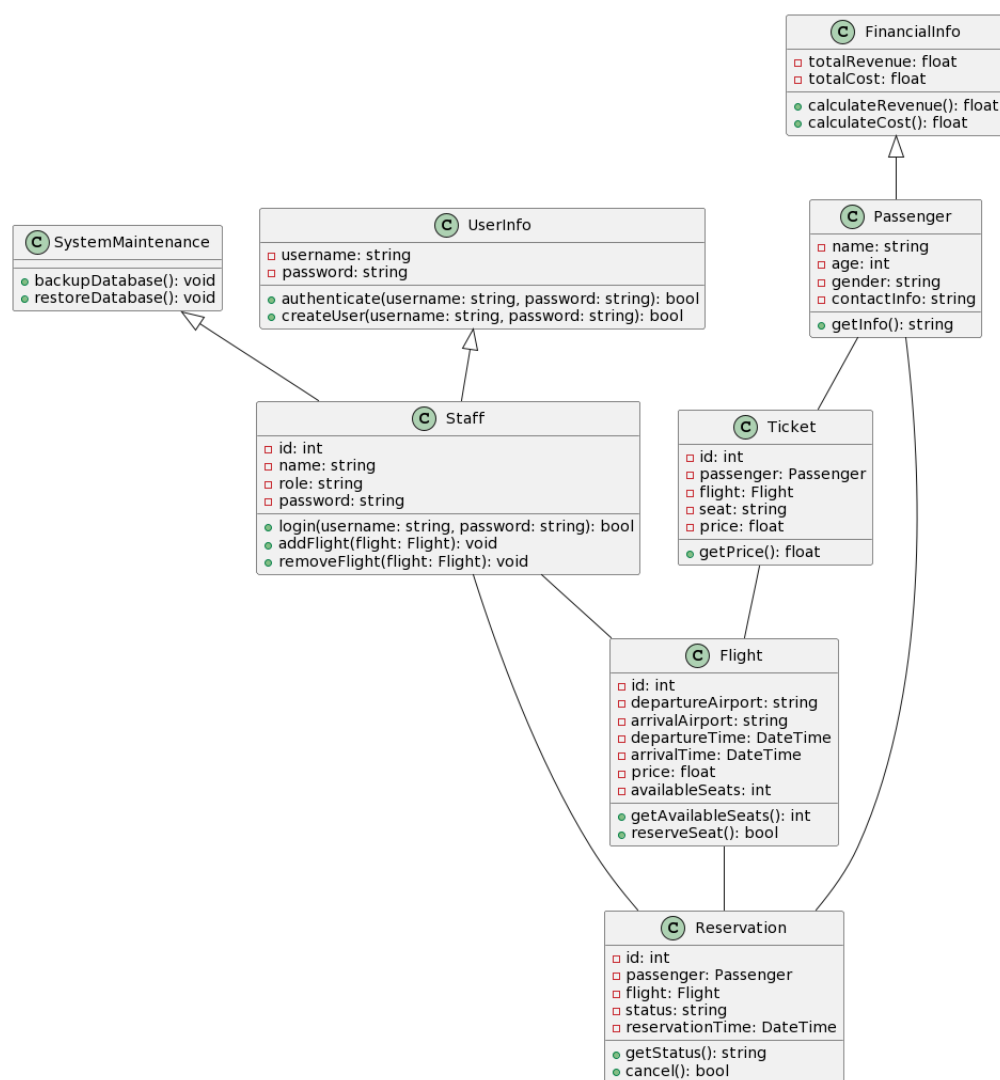


图 6: 详细类图

在基础的简易设计之上补充设计了以上的模块，得到的详细类图如上所示，因为是使用的 PlantUML 工具，给出详细类图实现的源代码如下：

详细类图实现

```
1 @startuml
2
3 class Passenger {
4     - name: string
5     - age: int
6     - gender: string
7     - contactInfo: string
8     + getInfo(): string
9 }
10
11 class Staff {
12     - id: int
13     - name: string
14     - role: string
15     - password: string
16     + login(username: string, password: string): bool
17     + addFlight(flight: Flight): void
18     + removeFlight(flight: Flight): void
19 }
20
21 class Ticket {
22     - id: int
23     - passenger: Passenger
24     - flight: Flight
25     - seat: string
26     - price: float
27     + getPrice(): float
28 }
29
30 class Flight {
31     - id: int
32     - departureAirport: string
33     - arrivalAirport: string
34     - departureTime: DateTime
35     - arrivalTime: DateTime
36     - price: float
37     - availableSeats: int
38     + getAvailableSeats(): int
39     + reserveSeat(): bool
40 }
41
42 class Reservation {
43     - id: int
44     - passenger: Passenger
45     - flight: Flight
46     - status: string
47     - reservationTime: DateTime
```

```
48     + getStatus(): string
49     + cancel(): bool
50 }
51
52 class SystemMaintenance {
53     + backupDatabase(): void
54     + restoreDatabase(): void
55 }
56
57 class FinancialInfo {
58     - totalRevenue: float
59     - totalCost: float
60     + calculateRevenue(): float
61     + calculateCost(): float
62 }
63
64 class UserInfo {
65     - username: string
66     - password: string
67     + authenticate(username: string, password: string): bool
68     + createUser(username: string, password: string): bool
69 }
70
71 Passenger — Ticket
72 Ticket — Flight
73 Passenger — Reservation
74 Flight — Reservation
75 Staff — Flight
76 Staff — Reservation
77 SystemMaintenance <|-- Staff
78 FinancialInfo <|-- Passenger
79 UserInfo <|-- Staff
80
81 @enduml
```

五、 顺序图

顺序图（Sequence Diagram）是一种用来表示对象之间交互行为的 UML 图形语言。它描述了对象之间按时间顺序传递的消息和执行的动作，是软件设计中重要的一种建模工具。

顺序图的一个重要特点是它显示了对象之间的交互顺序和时序，可以帮助开发人员深入理解系统的工作流程和交互方式，从而更好地设计和实现系统。同时，它也可以用于测试和验证系统的正确性和性能。

总的来说，顺序图在软件设计中扮演着非常重要的角色，它可以帮助开发人员更好地理解 and 描述系统的交互行为，从而提高开发效率和质量。

(一) 旅客预定顺序图

旅客预定功能是机票预定系统的核心部分，旅客通过该部分实现网上预定机票的功能，遍历乘客的出行。乘客的操作均需要涉及后台数据库，用到数据库的增删改查操作。相对应的有系统支持的各种服务功能。预定功能需要对航班信息数据库和旅客账户数据库里传递的数据。

当旅客使用机票预定系统进行机票预定时，以下是可能发生的一些步骤：

1. 旅客访问机票预定系统，输入出发地、目的地、日期和乘客人数等信息。
2. 系统根据旅客提供的信息查询航班信息，并返回符合条件的航班列表。
3. 旅客选择一个航班，并输入乘客个人信息和联系方式。
4. 系统核实旅客提供的信息，并计算机票价格。
5. 旅客确认机票预定信息，选择支付方式并完成支付。
6. 系统生成机票订单，并向旅客发送确认邮件或短信。
7. 旅客可以在机票预定系统中查询、修改或取消订单，直到出发前一定时间。

在实际应用中，可能还会涉及到一些特殊情况，比如旅客需要选择座位、办理退改签以及打印机票等操作，这些都可以通过系统中的相关功能来完成。

根据上述的分析我们可以实现旅客预定顺序图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

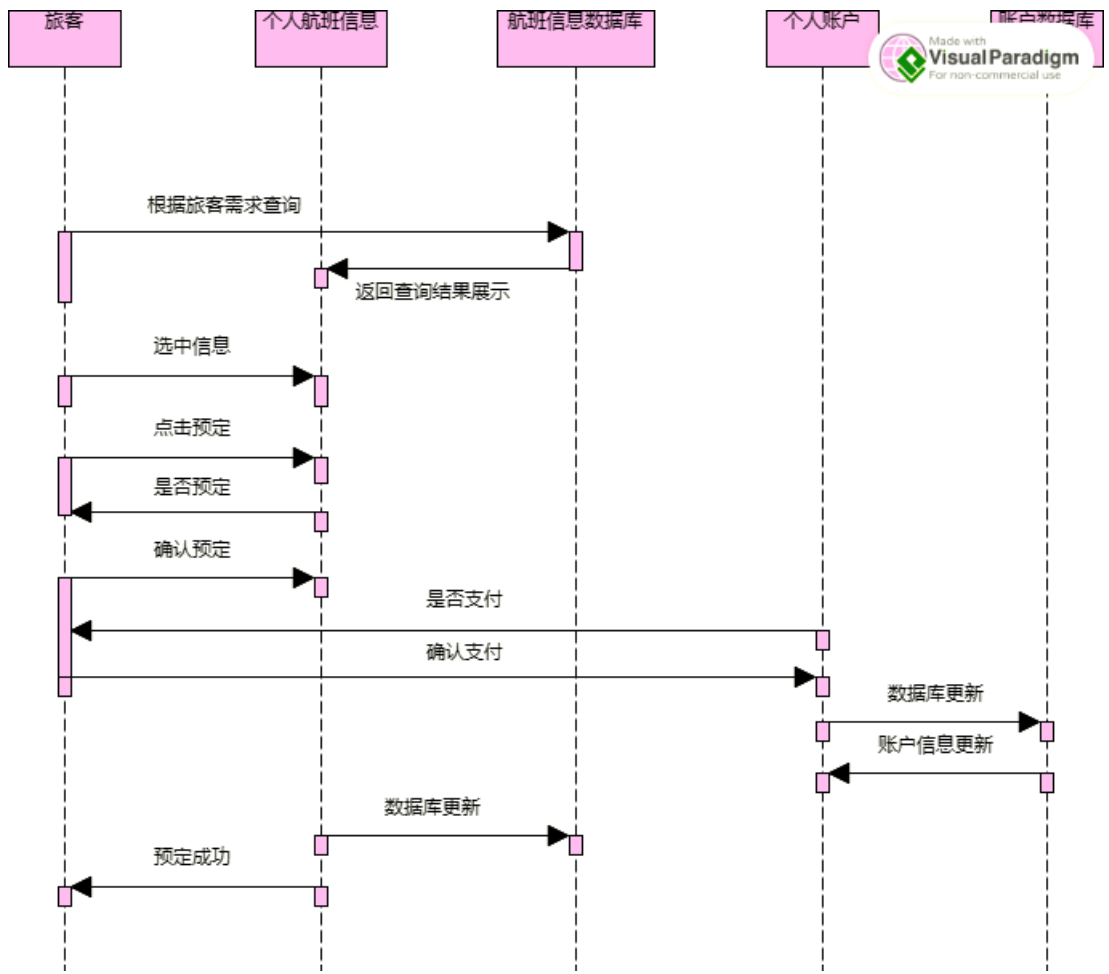


图 7: 旅客预定顺序图

对于退订等操作，我们需要将发送至数据库的消息转化为发送至个人航班信息之中进行查询，然后再决定是否要进行退订，后续的流程与预订流程类似，这里就不再赘述，下图给出旅客需要打印机票作为凭证的顺序图。

根据上述的分析我们可以实现打印航班信息顺序图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

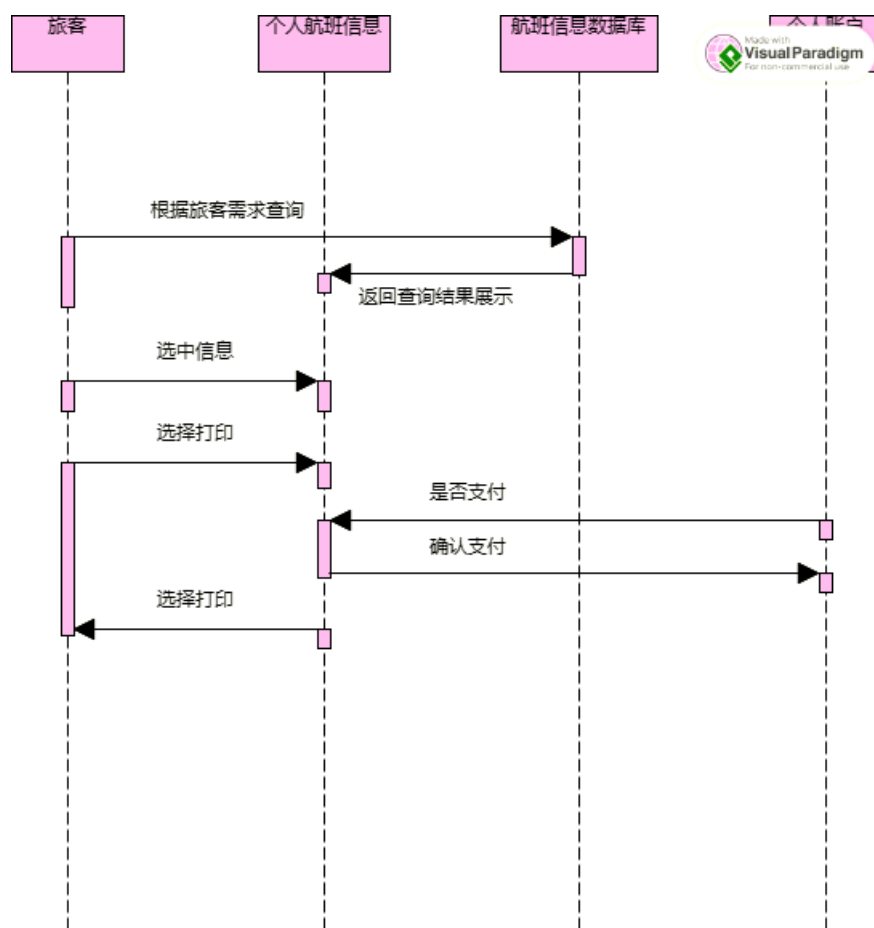


图 8: 打印航班信息顺序图

(二) 工作人员信息管理顺序图

工作人员也是机票预定系统的关键参与者,负责航空信息的管理维护操作,这里针对工作人员的具体航空信息维护操作绘制了顺序图进行分析。

当工作人员使用机票预定系统进行信息管理时,以下是可能发生的一些步骤:

1. 登录系统,登录成功后,工作人员会被带到机票预定系统的主页面。接下来,他们需要进入航班信息管理页面,这可以通过在主页面上选择相应的选项来完成。
2. 一旦进入航班信息管理页面,工作人员可以开始输入航班信息,包括航班号、起飞时间、到达时间、起飞地点、目的地等等。这些信息可以通过表格或表单的形式进行输入。
3. 在输入航班信息后,系统会对这些信息进行验证,以确保它们的准确性和完整性。如果有任何错误或不完整的信息,系统会提示工作人员进行修正。
4. 一旦所有的航班信息都被验证通过,工作人员就可以将它们保存到机票预定系统中。
5. 在航班信息被保存后,工作人员还可以随时对它们进行更新。这可以通过选择相应的航班,并进行修改或删除来完成。

根据上述的分析我们可以实现工作人员信息管理顺序图如下,这里使用的是 Visual Paradigm 的线上工具进行绘制:

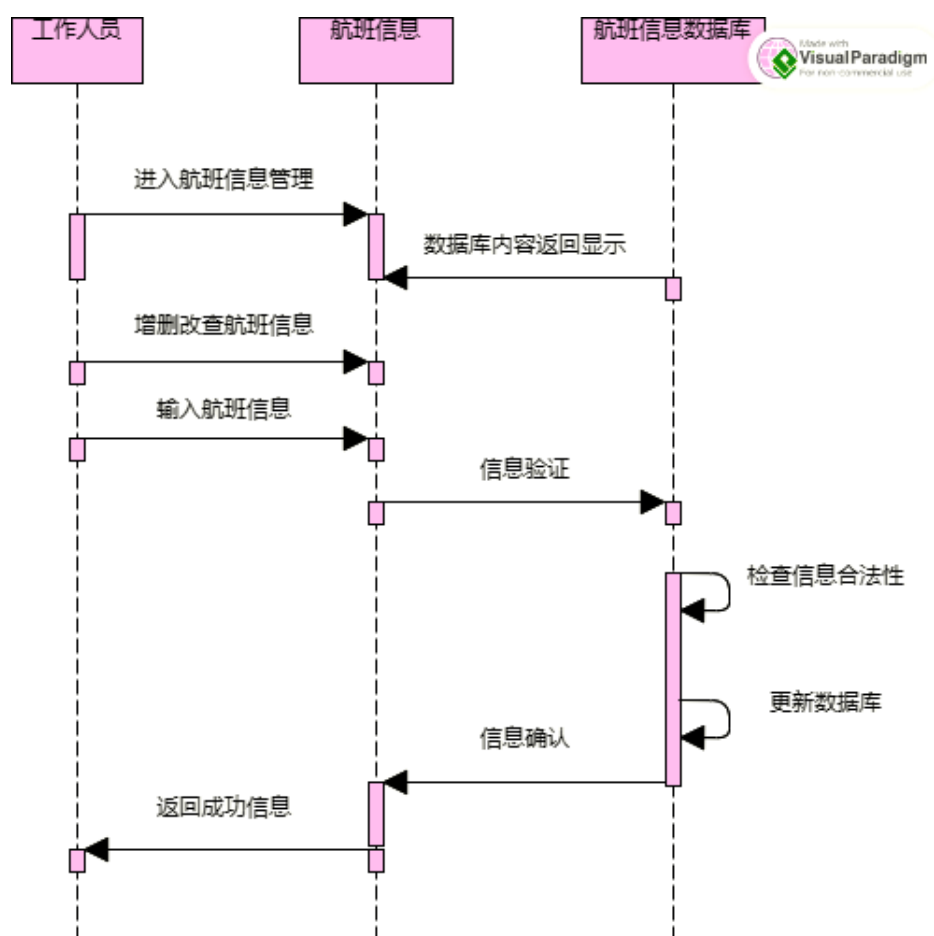


图 9: 工作人员信息管理顺序图

(三) 旅客账户管理顺序图

旅客的另一个重要的功能操作，那就是对旅客的个人账户进行管理，能够余额进行查询，也能够进行账户的充值，以及个人信息的填写等内容。

1. 首先，用户需要在系统中创建一个账户，并填写必要的个人信息，例如姓名、电子邮件地址、手机号码等等。系统验证用户的信息并创建一个唯一的用户 ID。
2. 接下来，用户需要登录到他们的账户中以访问其个人信息和进行机票预订，并输入其用户名和密码。系统验证用户的身份并授权用户访问其账户信息。
3. 接下来，用户需要登录到他们的账户中以访问其个人信息和进行机票预订，并输入其用户名和密码。系统验证用户的身份并授权用户访问其账户信息。
4. 当用户准备进行充值时，他们需要提供付款途径和付款信息，然后填写必要的旅行细节和支付信息。这些信息将被保存在用户的账户中，以便用户以后方便地查看和管理其机票订单。
5. 最后，在用户需要取消或修改其机票订单时，他们可以访问其账户并进行必要的更改。展示取消订单、更改机票日期以及退款等等信息。

根据上述的分析我们可以实现旅客账户管理顺序图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

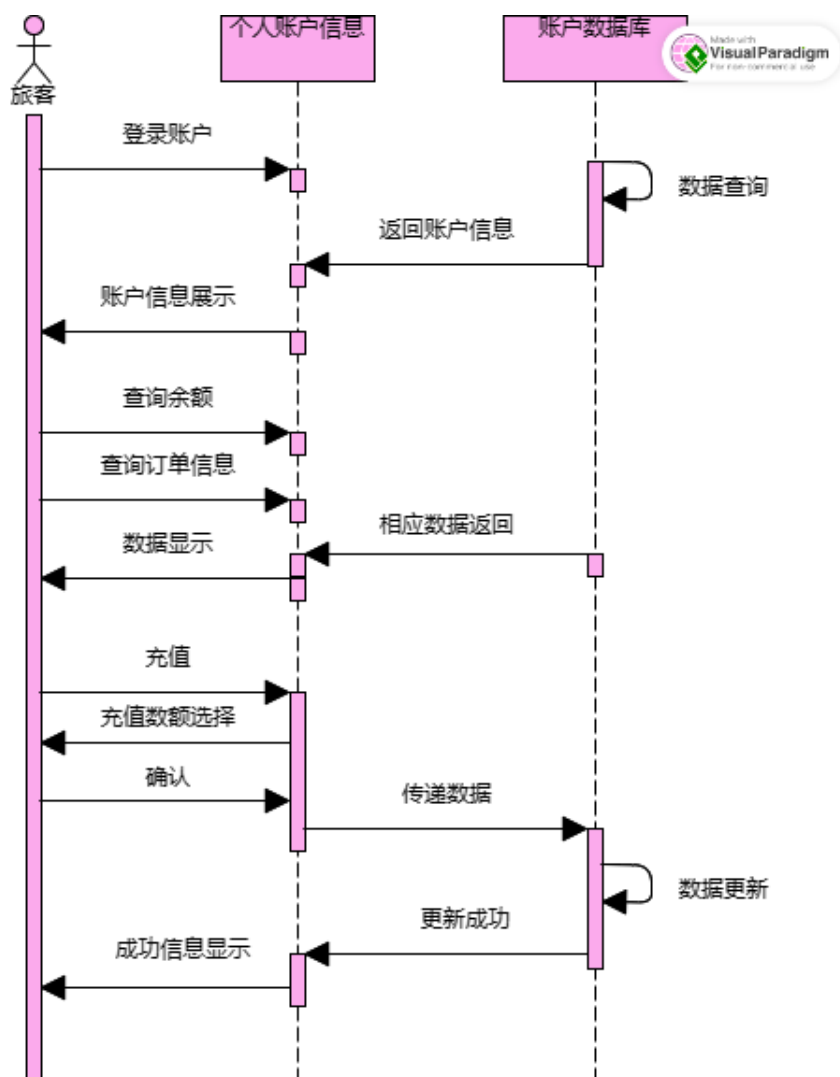


图 10: 旅客账户管理顺序图

至此我们完成了大概的业务流程的分析，并以此为根据绘制了对应的顺序图。

六、 协作图

协作图是指一种图形化的表示方法，用于描述软件系统的各个部分之间的关系和交互。协作图通常被用来描述系统的结构和行为，以及在系统中各个部分之间的消息传递。

在软件设计中，协作图可以帮助团队成员更好地理解系统的架构和交互方式，并且能够帮助设计师和开发人员更好地协作和沟通。以下是一些协作图的功能：

- 显示系统结构：协作图可以帮助团队成员更好地了解系统的组成部分，包括各个组件和它们之间的关系。
- 显示消息传递：协作图可以显示系统中各个组件之间的消息传递方式，以及它们的顺序和时机。
- 描述系统行为：协作图可以描述系统中各个组件的行为方式，以及它们之间的互动方式。
- 提高设计效率：协作图可以帮助设计师更快地了解系统的结构和行为，从而提高设计效率。

- 简化开发过程：协作图可以帮助开发人员更好地理解代码实现的意义和目的，从而简化开发过程。

(一) 旅客预约管理协作图

协作图显示对象角色之间的关系，在机票预定系统之中主要显示的就是旅客与工作人员等的关系。根据上述的分析我们可以实现旅客预约管理协作图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

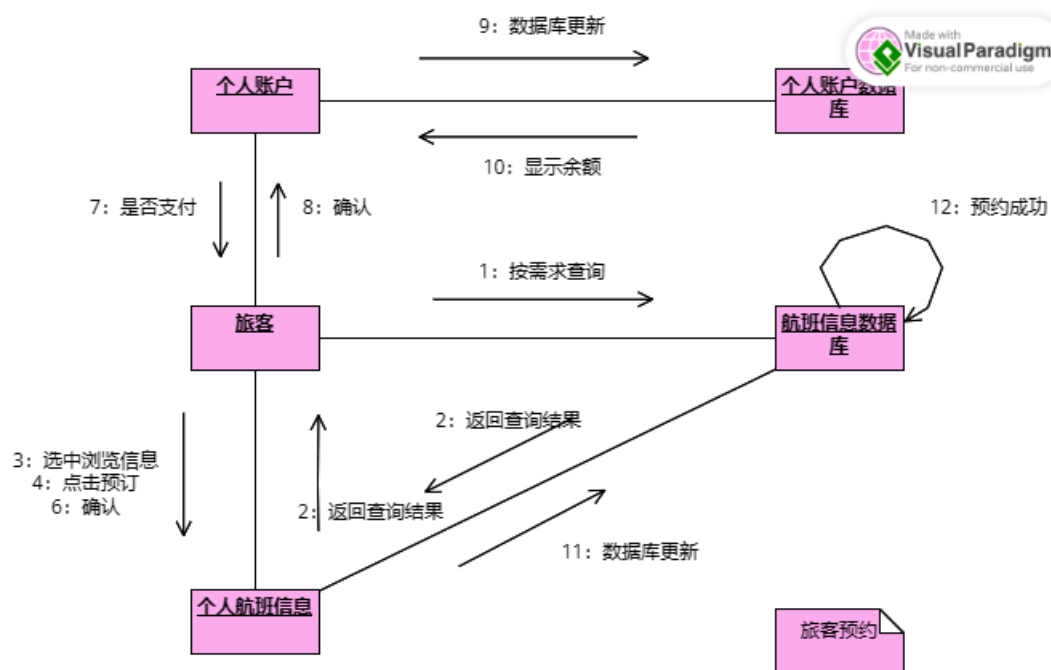


图 11: 旅客预约管理协作图

(二) 旅客账户管理协作图

根据上述的分析我们可以实现旅客账户管理协作图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

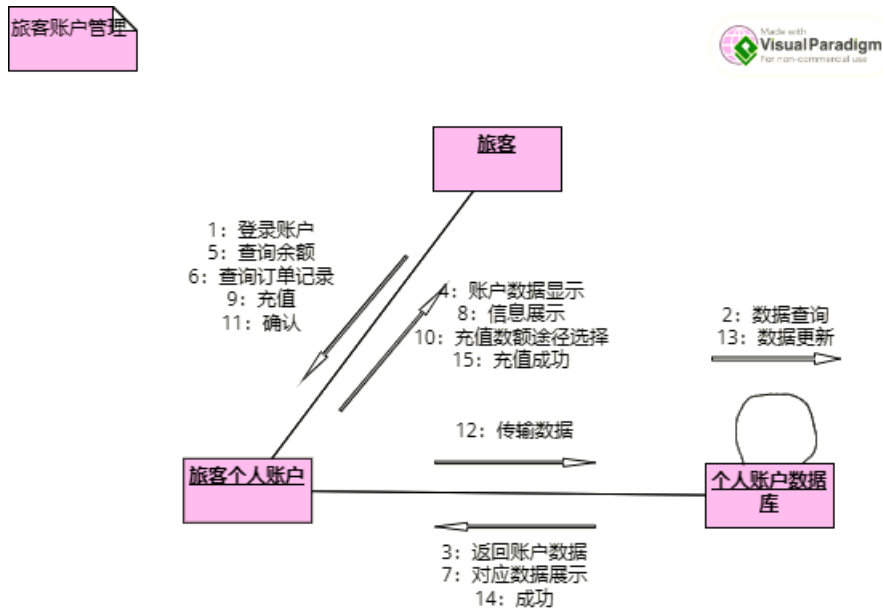


图 12: 旅客账户管理协作图

(三) 工作人员信息管理协作图

根据上述的分析我们可以实现工作人员信息管理协作图如下,这里使用的是 Visual Paradigm 的线上工具进行绘制:

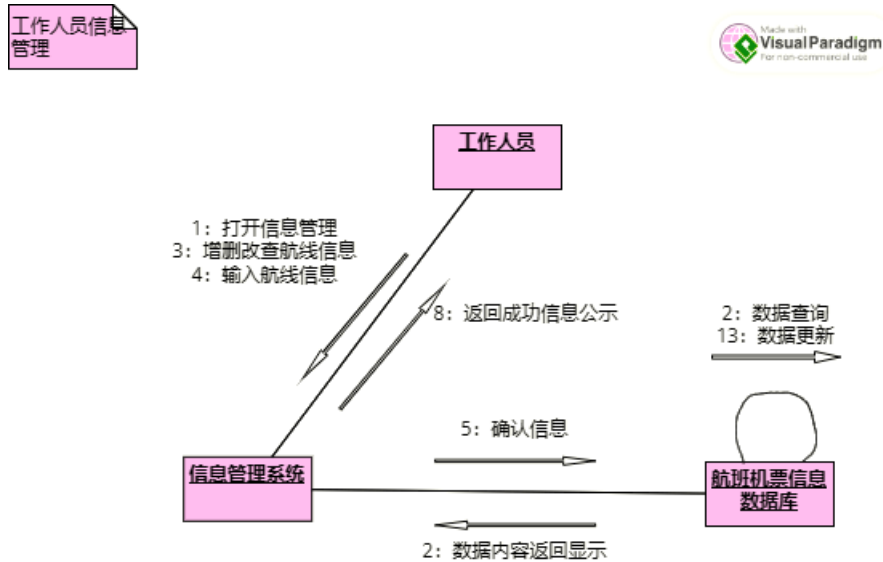


图 13: 工作人员信息管理协作图

至此,我们已经完成了机票预定系统的基础的协作图部分,协作图是软件设计中非常有用的一种工具,可以帮助团队成员更好地理解 and 协作,提高软件设计和开发的效率和质量。

七、 状态图

状态图 (State diagram) 是一种用于描述对象或系统在不同状态下的行为和转换的图形化表示方法。在软件设计中, 状态图通常被用来描述一个对象或系统在不同的状态下, 接受外界的输入或事件, 执行相应的操作, 并且可能会发生状态转换的情况。

状态是指对象或系统可能处于的不同状态; 转移是指状态之间的切换, 通常由触发事件引发; 动作是指在状态转移时需要执行的操作; 事件是指触发状态转移的外界事件或条件。

- 状态图的作用是可以帮助开发人员更好地理解 and 设计系统的行为, 并且能够提供一个清晰的视图来描述系统的状态和转换。
- 通过使用状态图, 开发人员可以更好地管理和控制对象或系统的状态, 避免错误和异常情况的发生, 同时也可以更好地进行系统的测试和调试。
- 状态图还可以用于和客户或项目经理进行交流和沟通, 以确保所有相关方对系统的行为和交互方式有一个共同的理解。

(一) 机票信息状态图

机票信息状态可以预订也可以退订, 具体实现的细节不同, 查询详细的航班信息并进行规划安排, 选定要预订或者退订的航班选择要操作, 更新航空信息数据库。

根据上述的分析我们可以实现机票信息状态图如下, 这里使用的是 Visual Paradigm 的线上工具进行绘制:

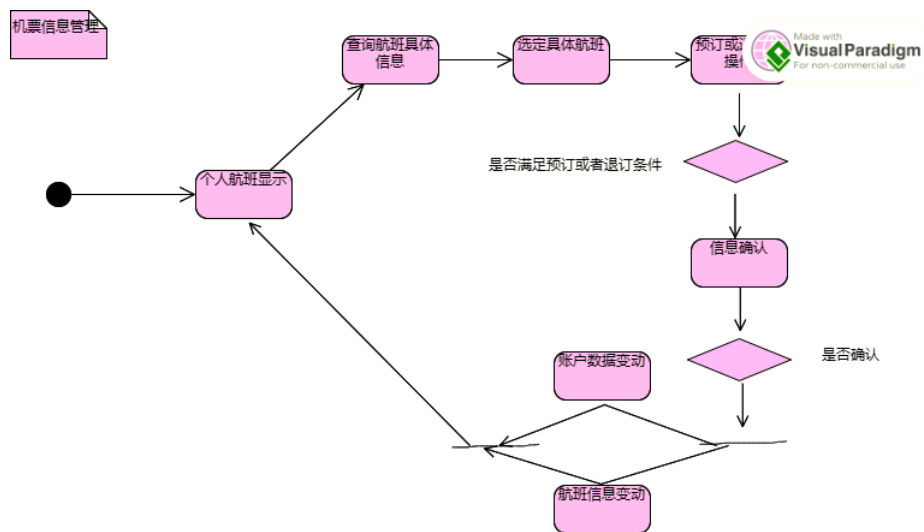


图 14: 机票信息状态图

(二) 公共信息服务状态图

公共信息模块类似于第三方服务模块, 例如信息公示, 邮件电子信息提示等功能, 便于乘客之间、航空公司之间的信息交流, 显示公共服务信息。

根据上述的分析我们可以实现公共信息服务状态图如下, 这里使用的是 Visual Paradigm 的线上工具进行绘制:

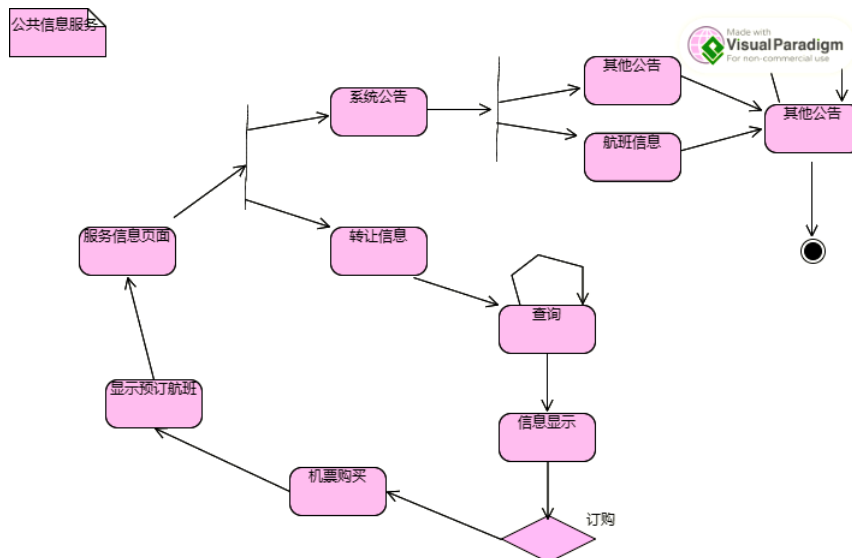


图 15: 公共信息服务状态图

八、构件图

构件图是一种用于软件设计的图形化表示法，它描述了软件系统中的各个构件以及它们之间的关系。构件图常常用于面向对象的软件设计中，它能够帮助软件设计师更好地理解系统的结构，同时也能够帮助开发者更好地实现软件系统。

构件图中的构件可以包括各种软件组件，构件图的作用主要有以下几个方面：

- 描述软件系统结构

构件图可以帮助软件设计师更好地理解系统的结构，它能够清晰地展示系统中各个构件之间的关系。

- ### • 分离关注点

通过构件图，可以将系统中各个构件的职责和功能分离出来，使得每个构件可以专注于自己的任务，提高系统的模块化程度。

- 促进复用和扩展

构件图可以明确地描述构件之间的依赖关系和组合关系，使得构件可以更加方便地进行复用和扩展，同时也能够方便地进行组件级别的测试和调试。

- 交流和协作

构件图是一种通用的软件设计语言，它可以帮助开发者进行沟通和协作，促进项目的合作和开发效率的提升。

我们的机票预定系统的设计之中，由五个组件组成，分别为客户端程序、管理员程序、服务器端程序、数据库端程序以及数据库五个部分。客户端程序与管理员端程序与服务器端程序存在着依赖关系，数据库端程序与数据库存在着依赖关系。而服务器端程序则通过一个接口连接到数据库端程序。

根据上述的分析我们可以实现系统构件图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

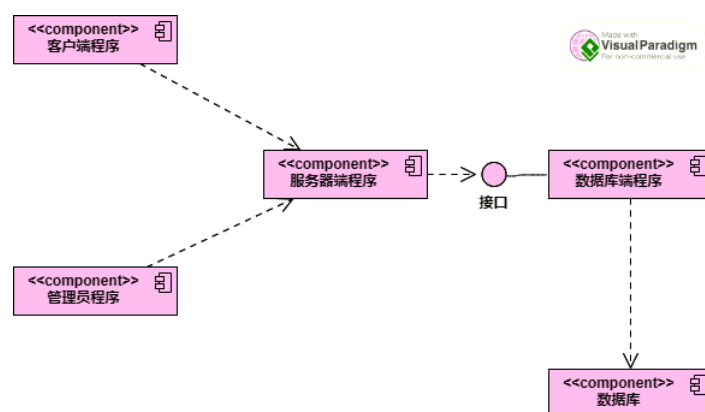


图 16: 系统构件图

至此，机票预定系统的构件图基础部分就已经完成了。

九、部署图

部署图（Deployment Diagram）是软件设计中的一种 UML 图，用于描述系统的物理结构以及系统各组件之间的部署关系。在部署图中，系统的各个组件被看作是节点（Node），节点之间通过连接线表示它们之间的通信和依赖关系。

接下来根据以上部分对于机票预定系统的分析，给出部署图的设计如下：

- 在系统的最高层次，我们可以将整个系统看作是一个单一的节点，这个节点代表整个系统的运行环境。在这个节点下，我们可以再定义一些子节点来代表不同的物理设备或服务器。
- 在机票预定系统中，通常会使用一些第三方服务来实现支付、短信通知、邮件通知等功能。这些服务通常会部署在外部的节点上，而不是部署在系统自己的节点上。因此，我们可以在部署图中添加这些外部节点，并通过连接线表示系统与这些外部节点之间的通信和依赖关系。
- 系统中的各个组件（如数据库、Web 服务器、应用服务器、客户端等）也会被部署在不同的节点上。在部署图中，我们可以为每个组件定义一个节点，并通过连接线表示它们之间的通信和依赖关系。
- 我们可以将数据库部署在一个独立的节点上，将 Web 服务器和应用服务器部署在另一个节点上，最后将客户端部署在用户的计算机上。连接线可以表示数据库节点和 Web 服务器节点之间的数据库访问关系，以及 Web 服务器节点和客户端节点之间的 HTTP 通信关系等。

总之，部署图可以帮助我们更好地理解系统的物理结构，从而更好地进行系统的部署和维护。根据上述的分析我们可以实现部署图如下，这里使用的是 Visual Paradigm 的线上工具进行绘制：

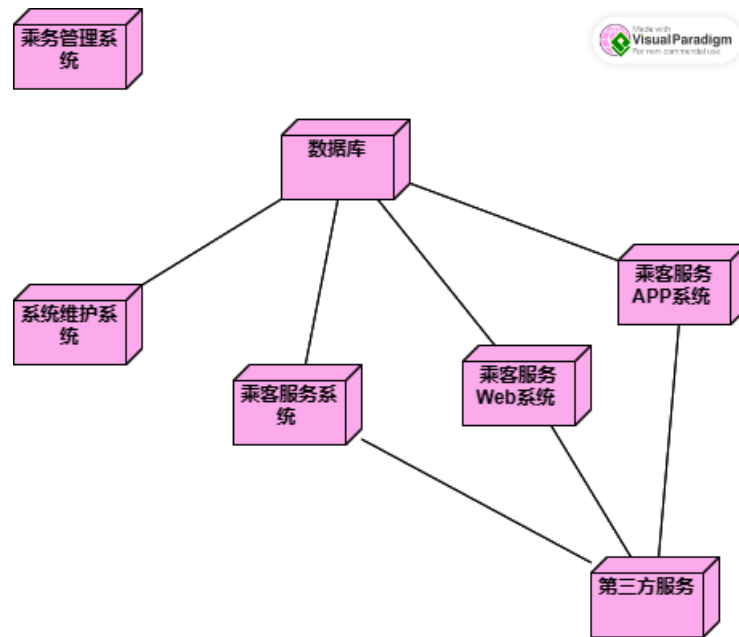


图 17: 部署图

使用部署图不仅可以显示运行时系统的结构，同时还传达了构成应用程序的硬件和软件元素的配置和部署方式。

十、 总结

通过对机票预定系统软件设计的学习和实践，我深入了解了软件设计的各个阶段，从理论到实践，每个阶段都有详细的展示，包括引言、类图和部署图的设计。这也是我之前未曾接触的领域。

通过机票预定系统的软件设计与实现，我进一步理解了 UML 等理论知识，并提升了我的分析和统计能力。这个过程让我更好地了解了如何将理论知识应用于实践。

参考文献

- [1] <https://www.planttext.com/>
- [2] <https://max.book118.com/html/2020/0910/8023040121002141.shtm>