



南開大學
Nankai University

计算机学院和网络空间安全学院
软件工程实验报告

软件设计—机票预订系统

姓名：魏伯繁

学号：2011395

专业：信息安全

2023 年 4 月 9 日

目录

1 场景描述	2
2 用例图	2
2.1 用例图简介	2
2.2 用例图绘制	2
3 活动图	3
3.1 活动图简介	3
3.2 活动图绘制	4
4 类图	5
4.1 类图介绍	5
4.2 类图绘制	6
5 顺序图	6
5.1 顺序图简介	6
5.2 顺序图绘制	7
6 协作图	8
6.1 协作图简介	8
6.2 绘制协作图	8
7 状态图	9
7.1 状态图简介	9
7.2 状态图的绘制	10
8 构件图	10
8.1 构件图简介	10
8.2 构件图绘制	11
9 部署图	11
9.1 部署图简介	11
9.2 部署图绘制	12

1 场景描述

机票预订的系统功能包括: 完整的存储、查询、核对、打印机票的功能。比如: 在此系统中, 旅客或工作人员通过系统查询为旅客安排航班, 打印取票通知和账单, 旅客在起飞前凭取票通知交款取票, 系统校验无误即可打印机票给旅客

该机票预订系统的用户主要为旅客和工作人员, 通过绘制用例图、活动图、类图、顺序图、协作图、状态图、构件图、部署图来描述整个系统的功能和运行逻辑。

2 用例图

2.1 用例图简介

UML 用例图是一种很常用的分析和设计工具, 用于描述软件系统与外界的人、其他系统或设备之间的交互, 以及系统内部各个部分与其它部分之间的交互。它为用户和开发人员提供了一个通用的、无形式的语言, 以便描述和理解系统的行为或功能。

用例图的基本要素包括: 参与者、用例、关系 (关系有包含、泛化、扩展、包含、注释等)。

1. 参与者: 参与者是指与系统进行交互的外在实体, 如用户、其他软件系统、硬件设备等。
2. 用例: 用例是系统提供的功能或服务, 它描述了系统的行为方式。
3. 关系: 关系是指用来说明参与者、用例、其他用例元素之间的关系, 比如继承、扩展、包含等。

2.2 用例图绘制

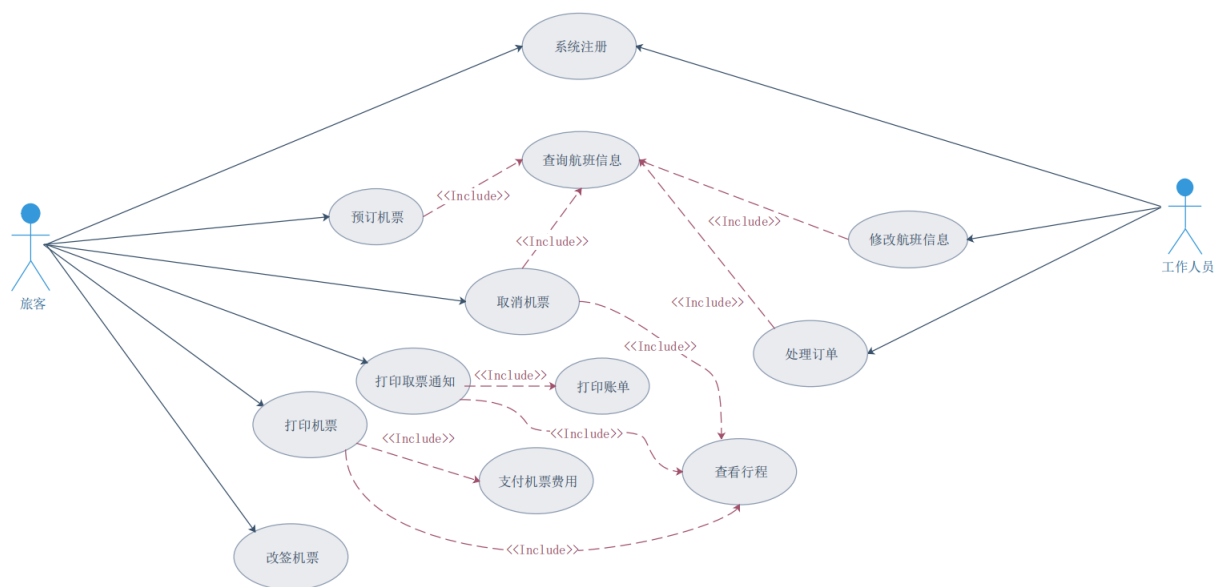


图 2.1: 用例图

上图是以系统处理功能流程为核心的用例图展示, 其主要目的为充分展示机票预订系统的功能, 清晰描述参与者与用例之间的关系。在机票预订系统中, 旅客和工作人员都需要进行系统的注册登录后

才能使用机票预订系统中的一系列功能。旅客可以使用的功能包括预订机票、打印取票通知和账单、打印机票、支付机票费用、改签机票、退订机票等，其中预订机票、取消机票、改签机票都需要查询航班信息以预订和旅客预期出行日期以及目的地一致的机票（改签机票会在下一张图更加详细的展示）。而取消机票、打印取票通知和打印机票都需要查看个人行程以确定打印正确日期的机票。打印取票通知和打印账单存在包含关系，因为必须有账单才能缴费，缴费后才能凭借取票通知单进行取票，同理取票的前提也是乘客必须已经进行了缴费，否则无法正常打印机票。

除了旅客以外，机票预订系统还有被授权的工作人员可以使用系统的一系列功能。其中，工作人员可以使用的功能主要包括航班信息管理、机票管理、订单查询和客户服务等。航班信息管理主要负责更新和维护航班的相关信息，例如航班号、机型、起飞时间、到达时间、航班班次等。而机票管理则负责添加、修改和删除机票，以满足旅客的不同需求。订单查询则为工作人员提供了便利的方式来查询和确定旅客的订单详细信息，如支付情况、乘机人信息、机票信息等等。除此之外，客户服务则提供了一种快速响应客户需求的途径，使得工作人员可以高效地处理旅客的问题。

需要注意的是，机票预订系统中的所有功能都必须遵守相关的规则和限制。例如，一张机票在规定的时间内只能被预订一次，而且同一张机票不能同时被多个旅客预订。此外，系统也需要保护用户的隐私信息，例如，工作人员不能随意查询、修改或删除用户的个人信息。机票预订系统需要满足相关的安全性、稳定性以及性能等方面的要求，以保证系统能够稳定、高效、可靠地运行。

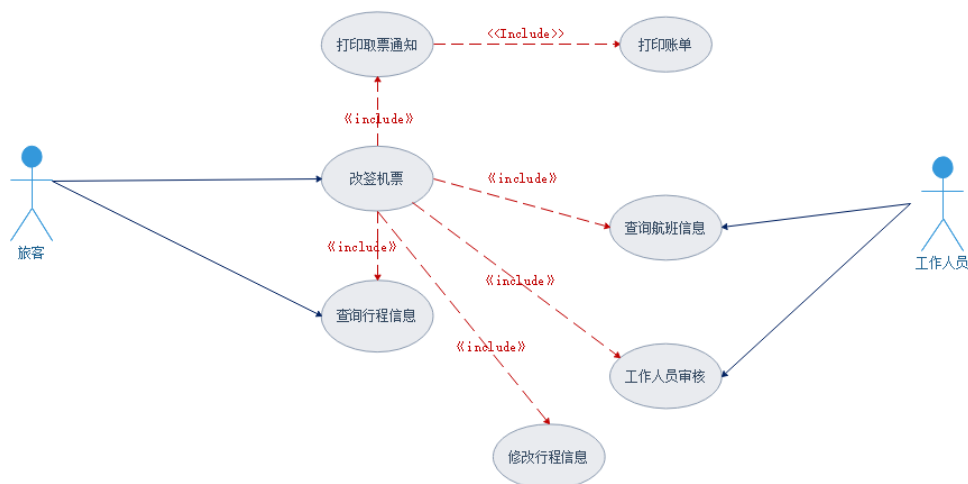


图 2.2: 用例图

上图是以改签机票为核心，着重体现机票预订系统中用例间彼此联系、交互的用例图示。在改签机票时，客户需要查询航班信息以确定想改签哪一个班次的航班，需要查询客户的个人行程已确定想要改签哪个已经预订的班次，并提交给工作人员审核。审核成功后个人行程才会被对应修改，并且会重新打印取票通知和账单，旧的账单自动作废。

3 活动图

3.1 活动图简介

UML 活动图是一种很常用的流程图，用来表示活动的执行过程。活动是指系统中的一个操作或流程，而活动图则用于说明活动的过程中涉及的各种动作、条件和并发控制，以及活动的交互和协同。

活动图可以帮助用户和开发人员全面、直观地了解系统的执行过程，是系统的设计和应用过程中

必不可少的工具。在实际操作中，活动图常常用于完成以下几个任务：

1. 描述系统的活动过程：通过定义活动图中的活动、控制流和对象流来帮助用户和开发人员描述系统的活动过程。
2. 确定系统的上下文：活动图可以描述活动之间的并发性、同步性和交互，帮助用户和开发人员明确系统的上下文。
3. 明确系统的条件：活动图的分支和合并可以帮助用户和开发人员明确活动过程中出现的不同条件。
4. 识别系统需求：通过定义活动图，用户和开发人员可以识别系统的不同需求，包括功能需求、非功能性需求、数据需求等。
5. 优化系统执行过程：通过定义活动图，用户和开发人员可以优化系统的执行过程，包括优化并发性、同步性和交互。

3.2 活动图绘制

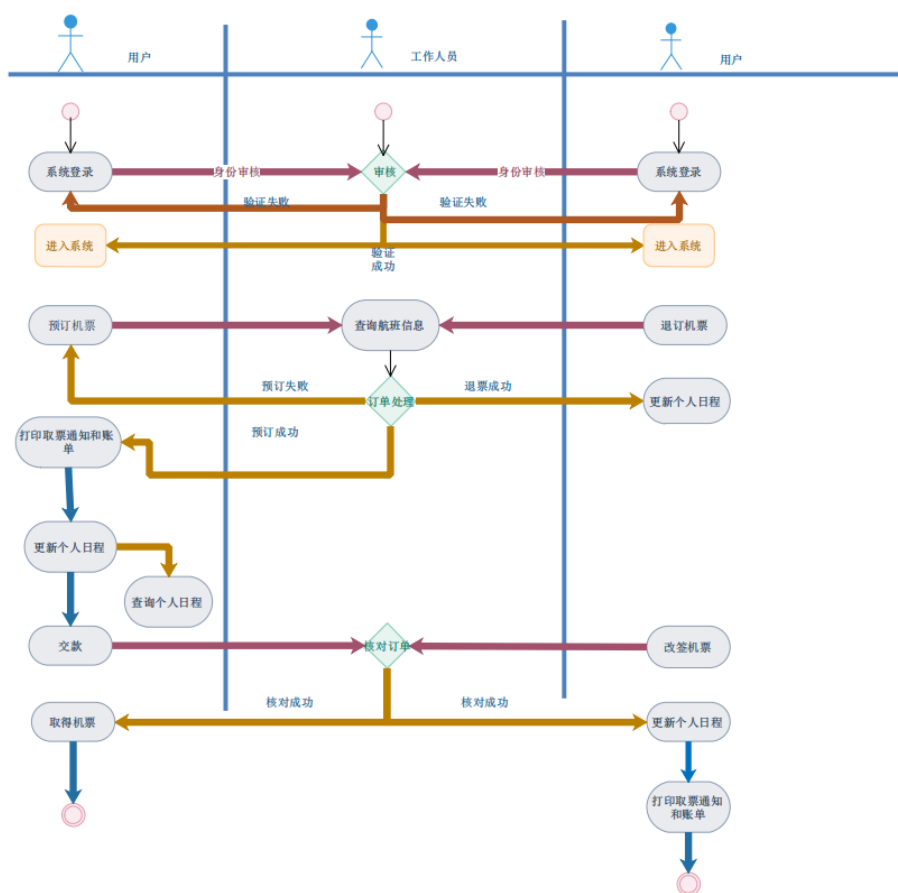


图 3.3: 活动图

上图展示了不同身份的用户在使用机票预订系统的不同功能时的交互与协作过程，用户的登录信息需要被审核已确定是正常的访问登录，在进入系统后，用户可以发起预订或退订机票，而机票信息的来源是由工作人员控制的航班信息系统，在提交预订/退订请求后系统会审核请求。请求成功后会系统

会修改用户的个人行程信息，如果是预订机票则还需要打印出取票通知和账单已待用户交款取票。改签机票同样需要交由工作人员审核待审核成功后更新个人日程并重新打印新的取票通知和账单。这张活动图旨在将不同身份的参与者在系统中的功能分割开并尽可能详细的描述活动间的交互过程。

机票预订系统是一个复杂的系统，需要不同身份的用户实现协作和交互才能正常运转。首先，用户的登录信息需要进行审核，以确保系统的安全性。用户进入系统后可以发起预订或退订机票，但是机票信息是由工作人员控制的航班信息系统提供的，因此需要系统审核请求才能成功。一旦预订或退订请求成功，系统会修改用户的个人行程信息，如果是预订机票，则需要打印出取票通知和账单，待用户交款后才能拿到机票。改签机票同样需要经过工作人员的审核，并更新用户的个人日程。这样，即使出现问题也可以方便地联系到用户。这张活动图将不同身份的参与者清晰地分割开来，尽可能详细地描述了活动间的交互过程，并对整个系统的安全和稳定运行起到极大的促进作用。

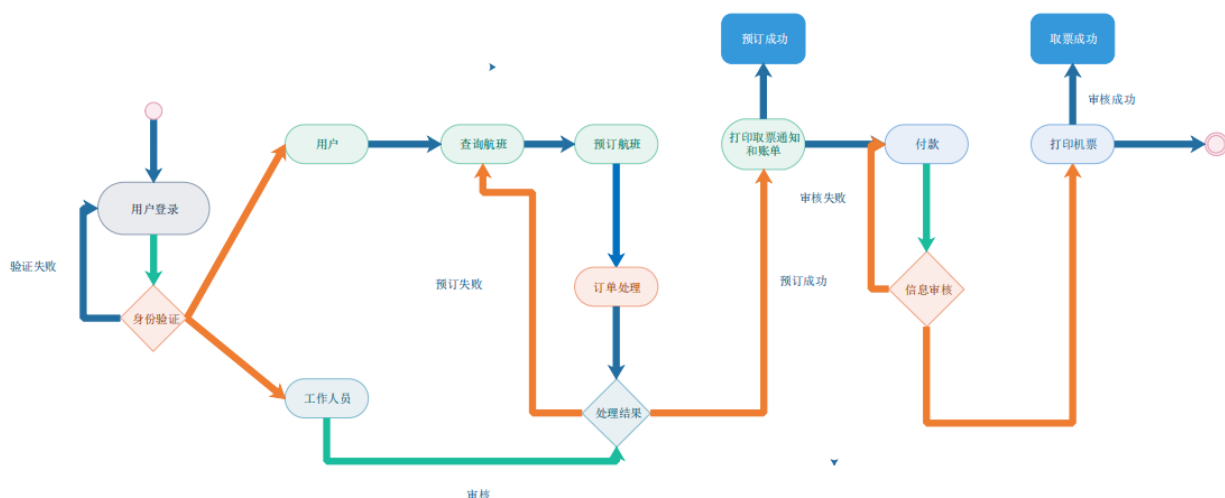


图 3.4: 活动图

与前一张活动图不同，本张活动图旨在清晰的描述功能流程与状态迁移过程，其中蓝色的矩形表示了现在的状态。上图清晰的描述了从系统登录到查看航班信息再到查询已有的行程信息，随后提交机票预订请求、打印取票通知和账单最后付款打印机票的全过程，是机票预订系统中最常使用的一套流程。该图可以帮助读者构建机票预定系统的基本功能流程框架。

登录后可以将不同的用户分为两类，一种为用户，用户可以进行一些普通的功能访问：比如机票预订、机票取票通知的打印、账单打印、缴费和最后的取票。其中包含两个非常重要的状态一个是预订成功状态，该状态在成功打印取票通知和账单后触发，第二个状态是取票成功状态，意味着一个预订机票的流程周期已经结束。

4 类图

4.1 类图介绍

UML 类图是一种很常用的分析和设计工具，用于描述软件系统中的类或对象及其之间的关系。它为用户和开发人员提供了一个通用的、无形式的语言，以便描述和理解系统的结构和行为。

类图的基本要素包括：类、关系（继承、实现、关联、聚合、组合等）和属性和方法等。

1. 类：类是指一个具有共同特性和行为的实体，如人、车、订单等。

2. 关系：关系是指类之间的联系，包括继承、实现、关联、聚合、组合等多种形式。
3. 属性和方法：属性和方法是类的组成要素之一，属性指类的特征、状态等，方法指其行为、操作等。

类图可以帮助用户和开发人员全面、直观地理解系统的结构和行为，是系统的设计和应用过程中必不可少的工具

4.2 类图绘制

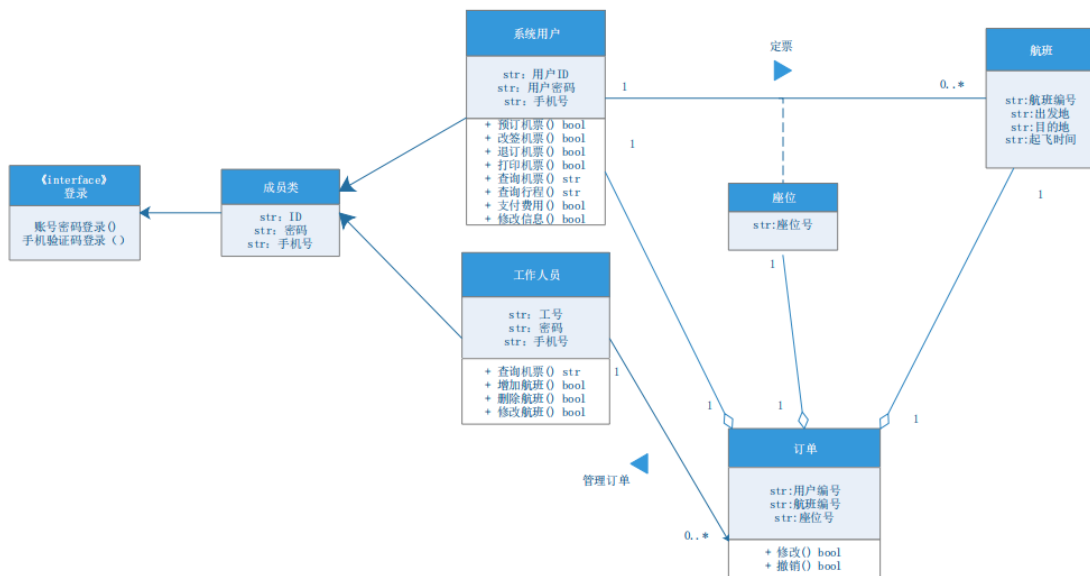


图 4.5: 类图

在本场景下，我的设计原则遵照奥卡姆剃刀原理：如无必要，勿增实体，尽可能简介的用最少的类完成整个机票预订系统的运转。用户方面最重要的类是两个：系统用户也就是旅客，他们的核心功能是要实现预订机票、打印账单、缴费、改签、退票等一系列操作的函数实现，而工作人员需要实现的是对机票系统的增删改操作。航班类标记了航班的编号、起飞时间、目的地和起飞地。每个用户可以预订 0 张或任意更多张机票，每次预订客户都会挑选一个座位，可以用座位号来标记。用户编号、航班编号。座位号一起组成了一个订单，他们之间的关系应该是聚合，因为他们的之间的逻辑关系更接近教师、学生和学院之间的关系，而不是像汽车零件和汽车一样的组合关系。一个工作人员可以管理任意多的订单，也可以不管理订单。

5 顺序图

5.1 顺序图简介

UML 顺序图是一种很常用的交互图，用于展示对象之间的交互以及它们之间传递消息的顺序。在顺序图中，系统中的对象表示为图中的垂直线，每个对象执行的操作则表示为序列图中的水平箭头，箭头按照时间顺序从上到下排列。

顺序图的基本要素包括：对象、生命线、执行条和消息等。

1. 对象：对象是指系统中的实体，如购物车、订单等。
2. 生命线：生命线是指建立在对象上的连续虚线，表示对象在一段时间内存在的情况。
3. 执行条：执行条是指生命线上的竖线，代表对象执行某个功能或操作。
4. 消息：消息是指对象之间相互关联的事件，例如：请求、响应、时间、异常等。

5.2 顺序图绘制

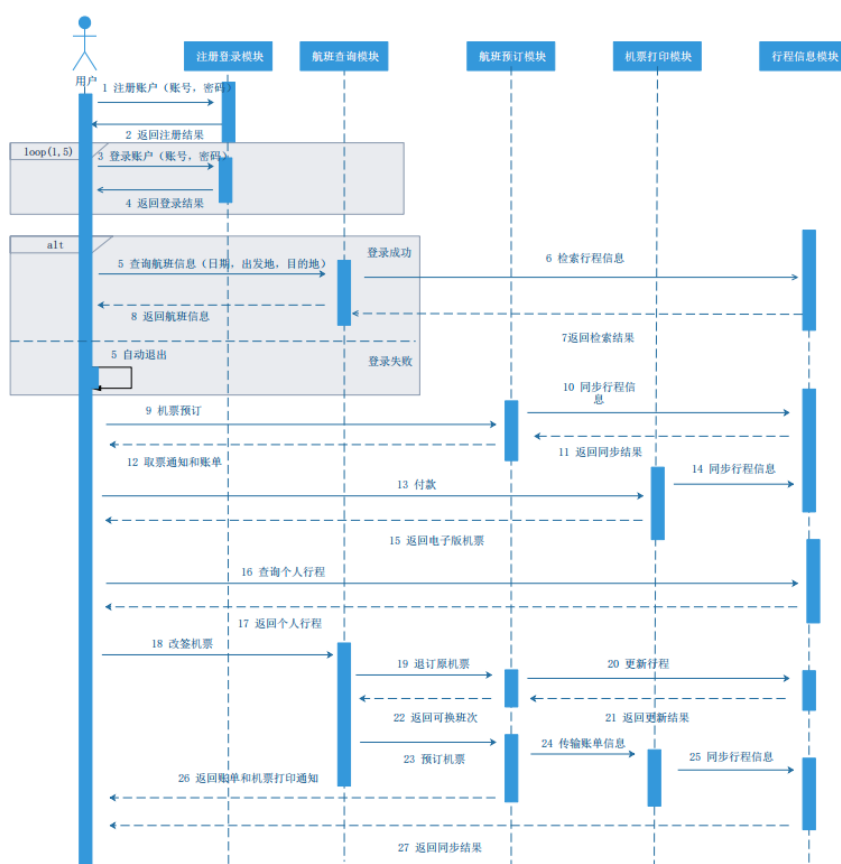


图 5.6: 顺序图

在顺序图中，首先将机票预订系统划分为五个模块，分别是注册登录、航班查询、航班预订、机票打印和行程信息模块，分别负责用户的登录注册、用户对航班信息的查询、预订、机票的打印以及对行程信息的修改以及查看。在登录阶段，允许每一名用户错误的输入他们的密码 5 次，超过五次自动冻结账号保护账号安全。用户首先发出查询航班的请求，请求抵达航班查询模块，航班查询模块根据数据库信息进行查询。与此同时他还会检测个人的行程记录以确定用户在对的时间段是否已经有机票预订而时间冲突。随后用户可以预订机票，系统会自动同步行程信息，如果用户完成缴费则可以通过机票打印模块进行机票打印。改签航班同样先进行原有机票的退订，随后查询库存机票以返回可以更改的机票信息，待用户选择后更新用户形成信息并同时返回新的账单和机票打印通知。

上图展示了正常流程下客户进行系统登录、航班信息查询、机票预订以及缴费打印机票一些列流程的协作图，再次过程中还穿插了对行程记录的查看。协作图绘制的基本流程与顺序图基本一致，按照不同的功能模块将整体的机票预订板块进行划分后就可以按照业务流程进行逐步绘制，随后即可以获得用户与不同模块交互的协作图。

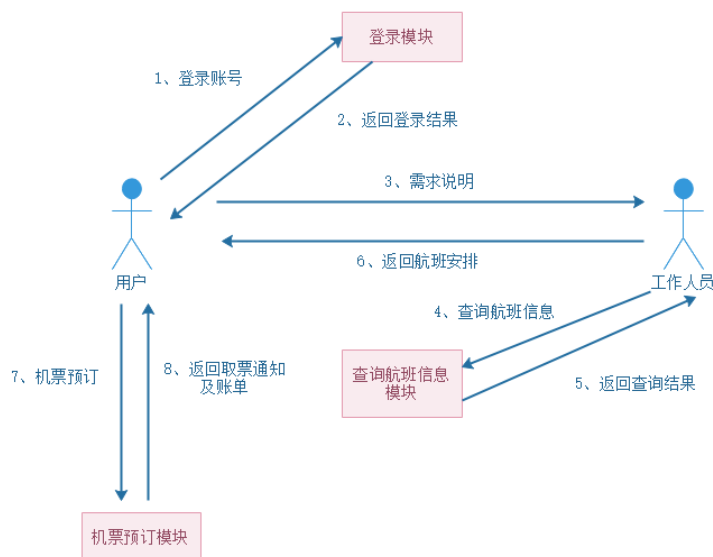


图 6.8: 协作图

上图展示了无法在顺序图中很好展示的协作图流程，因为他展示了多个对象间的交互过程，根据场景要求，工作人员除了需要对订单的有效性进行判定，还需要根据用户的需求帮助顾客返回安排行程，也就是查询航班信息挑选规划工作是由工作人员完成，客户只需确认后打印账单和取票通知，等待交费后打印机票即可。描述这样场景的协作图就蕴涵了两个不同对象的交互过程，其用顺序图表示会较为困难，更适合使用协作图表示。

7 状态图

7.1 状态图简介

UML 状态图是一种描述对象状态和状态转换的图形化语言，从而描述系统中对象的动态行为。状态是对象在一段时间内满足的条件或属性。在状态图中，系统的对象表示为带有状态的状态机或状态变化图，由状态和状态之间的转换组成。

状态图的基本要素包括：状态、转移、转移触发事件、动作等。

1. 状态：状态描述了对对象中一个确定的瞬时条件或属性，即对象的状态。状态用一个圆圈表示。
2. 转移：状态之间的转移描述了对对象在不同状态之间如何转换的过程。转移是有向边，表示从一个特定的状态到达另一个状态。
3. 转移触发事件：转移有触发事件。当对象进入一个状态时，系统在满足特定条件的情况下将转移到下一个状态。

4. 动作：动作是指对象在状态转换时所执行的操作。它是状态转移的输出，也可以驱动状态进一步的转移。

7.2 状态图的绘制

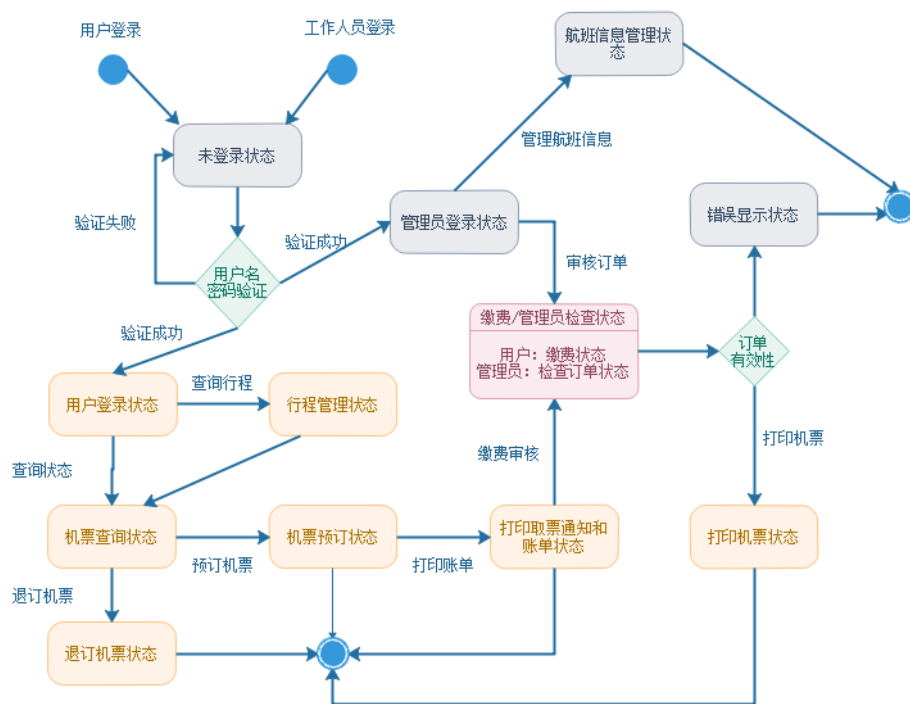


图 7.9: 状态图

在状态途中，用户以及工作人员所进行的每一步操作后的结果都被抽象为一个状态，而引起状态转移的是用户以及工作人员不同的动作。例如，从未登录状态开始，如果用户输入的用户名以及密码验证成功那么用户就可以进入用户登录状态，反之继续维持未登录状态不变，随后根据自己的需求可以选择进入机票查询状态查看所需要预订的机票，也可以进入行程管理状态来查看自己已经拥有的行程。而后也可以预订新的机票或者退订已经预订好的机票，在有新的行程被确认后，用户可以打印取票通知和账单，在对应日期进行缴费，交费后工作人员会对当前的状态信息进行检查，如果检查成功就可以成功取得机票。同理，工作人员也可以进入航班信息管理状态对航班信息进行增删改。

8 构件图

8.1 构件图简介

UML 构件图是一种用于描述软件系统中构件及其之间关系的建模方法，从而帮助开发人员和系统分析师更加直观的理解和描述系统的结构。在构件图中，软件系统中的各个构件都表示为一个单独的矩形，其中每个矩形代表系统中的一个组件或类等。

构件图的基本要素包括：构件、接口、关系等。

1. 构件：构件指软件系统中的一个组件，它可以是一个模块、类库、子系统或其他一些软件模块。在构件图中，构件通常用矩形表示。

2. 接口：接口指构件与其他构件之间的关系，它描绘了构件对外部的约束条件和规范。在构件图中，接口通常用包含一个或多个方法的半圆形表示。
3. 关系：构件图中的关系包括依赖、使用、泛化和实现等。依赖关系描述了一个构件依赖另一个构件中的某些功能；使用关系描述了构件之间的交互关系，泛化关系则表示一般与特殊关系的继承关系，实现关系表示的是一个类中的方法如何被接口所实现

8.2 构件图绘制

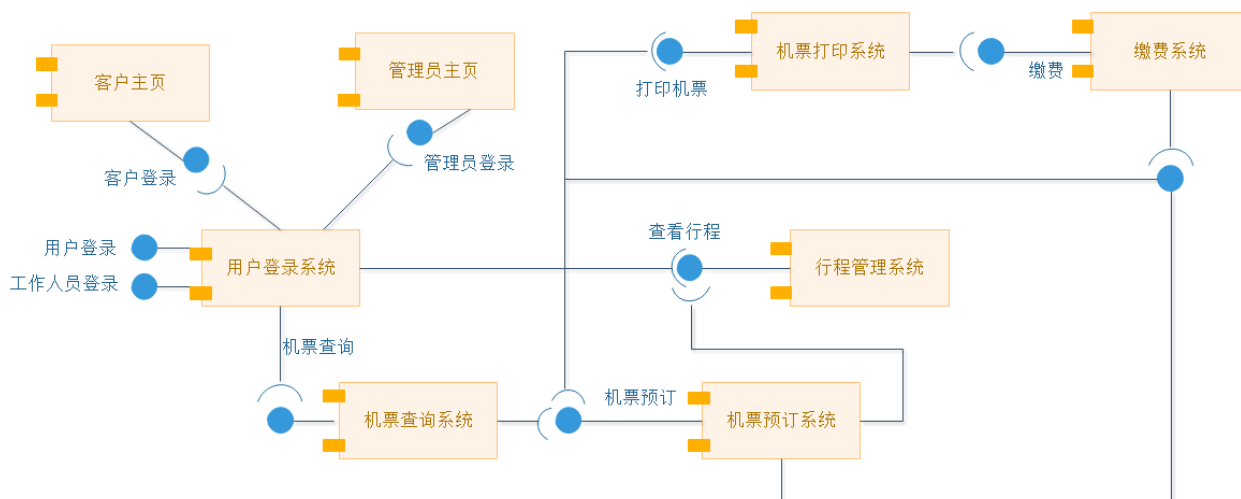


图 8.10: 构件图

在上图中，机票预订系统的各个构件间都用接口以及接口的实现的方式相互连接起来，例如用户登录系统向外提供用户登录和工作人员登录两个接口可以让使用航空订票系统的用户进行订票，同时，用户登录系统也需要登录成功后的不同显示界面，这就需要客户主页和管理员主页进行显示，同时，登录后用户也可能会选择机票查询或行程查询，所以还需要能提供这两个接口的模块，那么机票查询系统和行程管理系统可以向外提供这两个接口，查询机票后还需向外提供机票预订系统，该系统负责向外系统机票预订接口，同样该接口也可以被用户登录系统所使用。机票打印系统应该使用缴费系统的接口，因为必须确认缴费成功后才能够顺利打印机票，同样缴费系统的接口也需要接入机票预订系统和用户登录系统，保证用户无论在什么条件下都可以随时访问任意一个子系统完成操作，而无需一定要按照某种顺序才能完成完整的业务流程。

9 部署图

9.1 部署图简介

UML 部署图是一种用于描述软件系统中硬件资源和软件系统之间的关系建模方法，从而帮助开发人员和系统分析师更加直观的理解和描述应用程序的部署。在部署图中，软件系统中的各个部分都表示为一个单独的矩形，其中每个矩形代表系统中的一个节点或者硬件设备等。

部署图的基本要素包括：节点、关系等。节点指物理设备或逻辑单元，在部署图中通常表示为一个矩形，它代表了软件系统中一个硬件设备或者一个逻辑单元。关系包括连接线和通信路径。连接线用于表示连接的网络，通信路径是连接节点的路径。在部署图中，连接线和通信路径通常用线条表示。

9.2 部署图绘制

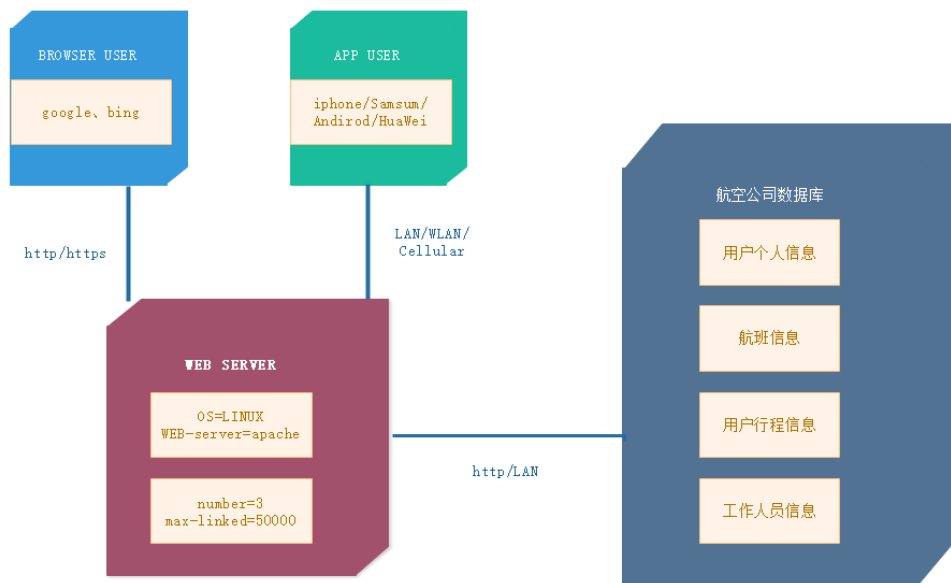


图 9.11: 构件图

上图展示了机票预订系统的部署思路，将大批量的重要信息存储在公司的数据库中，例如用户的信息、工作人员的登录信息、航班信息以及客户的具体行程信息。除此之外，负责对外相应的网页服务器计划部署在 linux 操作系统上以期待带来更好的应用性能、反应速度以及稳定性。服务器将采用分布式架构根据服务范围领域部署三个，每一个服务器设置的最大连接个数为 50000，面向用户的应用部署主要分为基于网页的访问和基于手机 app 的访问，服务器会对两种载体的访问给予回应，利用边缘计算等方式尽可能的保障每一位用户的使用体验。