



南開大學
Nankai University

计算机学院
软件工程实验报告

机票预定系统软件设计

姓名：徐文斌

学号：2010234

专业：计算机科学与技术

2023 年 4 月 9 日

目录

1	任务概述	2
2	用例图	2
2.1	用例图	2
2.2	用例规约	3
3	活动图	7
3.1	用户注册活动图	7
3.2	用户登录活动图	8
3.3	用户管理个人信息活动图	9
3.4	旅客订票活动图	10
3.5	旅客退票活动图	11
3.6	旅客打印取票通知活动图	12
3.7	旅客取票活动图	13
3.8	工作人员管理航班信息活动图	14
4	类图	16
4.1	类信息介绍	16
4.2	类间关系	18
5	顺序图	18
5.1	用户登录顺序图	18
5.2	旅客订票顺序图	19
5.3	旅客退票顺序图	20
5.4	旅客取票顺序图	21
5.5	工作人员修改航班信息顺序图	22
6	协作图	23
6.1	旅客登录协作图	24
6.2	旅客订票协作图	24
6.3	旅客退票协作图	24
6.4	旅客取票协作图	25
6.5	工作人员修改航班信息协作图	25
7	状态图	25
7.1	旅客状态图	25
7.2	工作人员状态图	26
8	构件图	27
9	部署图	28

1 任务概述

1. 用户：旅客、工作人员。

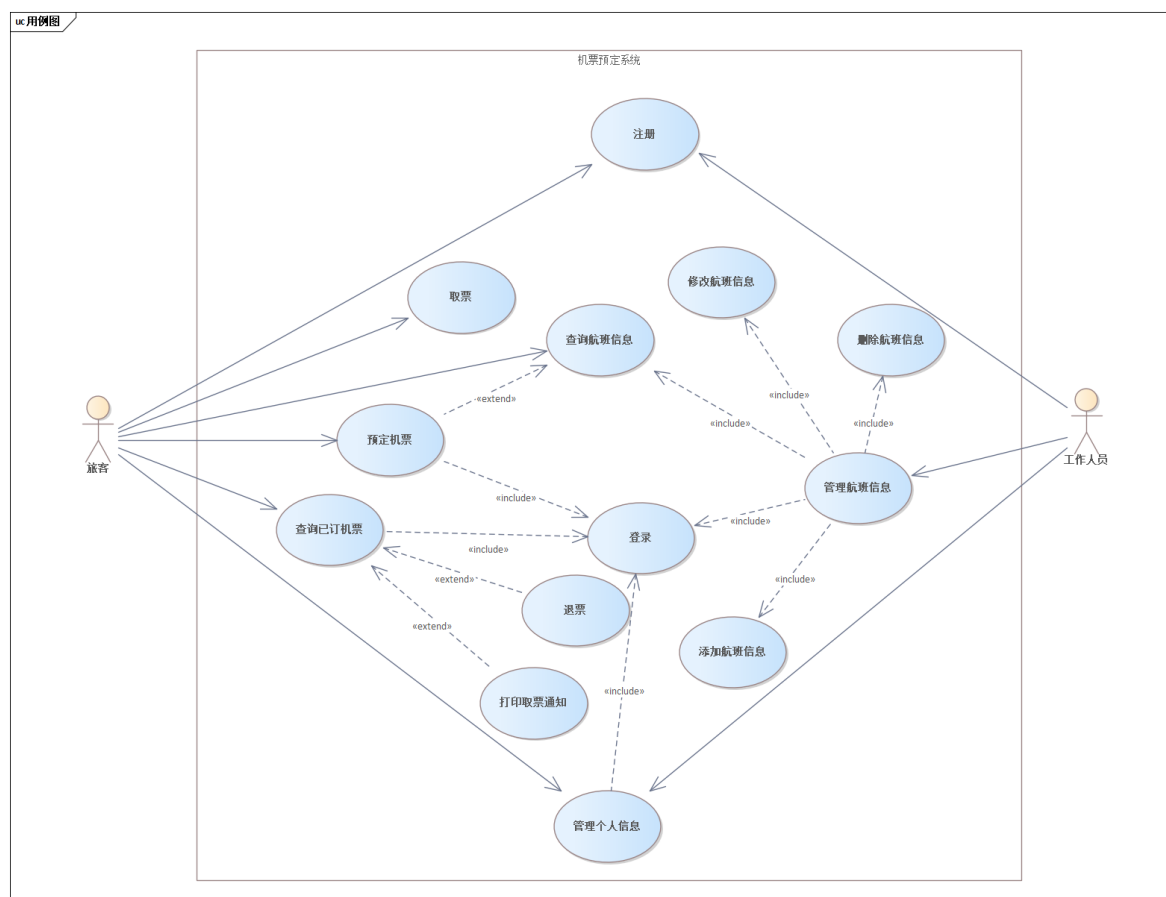
2. 功能需求：

- (a) 登录和注册功能：旅客和工作人员首先需要注册自己的账号，并使用该账号登录机票预定系统，登录成功后才可以进行更进一步的操作。
- (b) 个人信息管理功能：已经注册的用户可以在系统中修改自己的个人信息。
- (c) 航班查询功能：用户可以在系统中查询航班信息。
- (d) 订票退票功能：旅客通过航班查询功能确定自己期望的航班后，可以进行订票操作。之后旅客可以对所有自己已经预定的机票进行查阅，并在满足条件的情况下进行退票操作。
- (e) 票据打印功能：旅客可以打印取票通知、账单等票据。
- (f) 取票功能：在起飞前旅客可以凭借取票通知来交款取票、经系统检验无误后打印机票给旅客。
- (g) 航班信息管理功能：工作人员可以对航班信息以及机票信息进行增删改查等一系列的管理操作，对航班信息进行更新，并保证信息的正确性。

2 用例图

2.1 用例图

根据系统需要提供的功能，设计用例图如下图所示。



- 用例：注册、登录、查询航班信息、预定机票、查询已订机票、退票、打印取票通知、取票、管理航班信息、添加航班信息、删除航班信息、修改航班信息、管理个人信息
- 参与者：旅客、管理人员

2.2 用例规约

用例规约如下：

用例名称：	注册	
用例 ID：	1	
角色：	未注册旅客、未注册工作人员	
用例说明：	用例主要功能是实现用户的注册功能	
前置条件：	进入系统的注册页面	
基本事件流：	参与者动作	系统响应
	1. 用户输入基本信息 2. 用户点击注册按钮	3. 系统查找数据库，看用户名是否已被注册，如果是，则转至 3.1.1；如果不是，则将用户输入信息存入数据库，注册成功；若未输入用户名或密码，转至 3.1.2
其它事件流：	无	
异常事件流：	参与者动作	系统响应
	3.1.1 注册用户名重复 3.1.2 未输入用户名或密码	3.1.1 提示用户名重复 3.1.2 提示未输入用户名或密码
后置条件：	注册成功，跳转至登录页面	

用例名称：	登录	
用例 ID：	2	
角色：	旅客、工作人员	
用例说明：	用例主要功能是实现用户的登录功能	
前置条件：	进入系统的登录页面	
基本事件流：	参与者动作	系统响应
	1. 用户输入用户名和密码 2. 用户点击登录按钮	3. 系统查找数据库，看该用户是否在数据库中。若存在且密码正确则登录成功，进入主页面；若不存在转至 3.1.1；若用户名存在，但密码错误，转至 3.1.2；若未输入用户名或密码，转至 3.2.1
其它事件流：	无	
异常事件流：	参与者动作	系统响应

	3.1.1 用户名不存在 3.1.2 用户所输入的密码错误 3.2.1 未输入用户名或密码	3.1.1 提示用户名不存在 3.1.2 提示密码错误 3.2.1 提示未输入用户名或密码
后置条件:	登录成功，进入主界面	

用例名称:	查询航班信息	
用例 ID:	3	
角色:	旅客、工作人员	
用例说明:	用例主要功能是实现用户对航班信息的查询功能	
前置条件:	进入系统的航班查询页面	
基本事件流:	参与者动作	系统响应
	1. 用户输入查询的条件，如航班出发时间、起止地点等 2. 用户点击查询按钮	3. 系统查找数据库，根据用户输入的条件来过滤查找数据库中存储的航班信息，并将符合条件的航班信息返回给用户
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	无	无
后置条件:	进入航班查询结果显示页面，显示航班查询结果	

用例名称:	预订机票	
用例 ID:	4	
角色:	旅客	
用例说明:	用例主要功能是实现旅客预订机票功能	
前置条件:	旅客得到查询航班信息的结果	
基本事件流:	参与者动作	系统响应
	1. 用户点击机票的预订按钮	2. 系统查找数据库判断对应航班的机票是否已经被预订完，如果被预订完，则转至 2.1.1；否则订票成功，在数据库中记录用户的订票信息。
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	2.1.1 用户期望预订的机票已没有剩余	2.1.1 提示用户机票已被预订完
后置条件:	提示用户订票成功	

用例名称:	查询已订机票	
用例 ID:	5	
角色:	旅客	

用例说明:	用例主要功能是实现旅客对于已订机票的查询功能	
前置条件:	进入系统的已订机票查询页面	
基本事件流:	参与者动作	系统响应
	1. 用户点击查询按钮, 查询所有已订机票	2. 系统查找数据库, 得到对应用户所有已订机票的信息, 将信息返回给用户; 如果用户没有预订过任何机票, 则转至 2.1.1
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	2.1.1 用户没有预订过机票	2.1.1 提示用户没有预订过机票
后置条件:	进入订票查询结果页面, 显示用户已订的所有机票	

用例名称:	退票	
用例 ID:	6	
角色:	旅客	
用例说明:	用例主要功能是实现旅客对于已订机票的退票功能	
前置条件:	用户得到已订机票的查询结果	
基本事件流:	参与者动作	系统响应
	1. 用户点击机票的退票按钮	2. 系统根据用户选择的机票进行退票操作
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	无	无
后置条件:	转至退票成功页面	

用例名称:	打印取票通知	
用例 ID:	7	
角色:	旅客	
用例说明:	用例主要功能是实现旅客打印已订机票的取票通知功能	
前置条件:	用户得到已订机票的查询结果	
基本事件流:	参与者动作	系统响应
	1. 用户点击机票的打印取票通知按钮	2. 系统打印用户所选择的机票的取票通知
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	无	无
后置条件:	转至打印成功页面	

用例名称:	取票
-------	----

用例 ID:	8	
角色:	旅客	
用例说明:	用例主要功能是实现旅客的取票功能	
前置条件:	用户已打印机票的取票通知	
基本事件流:	参与者动作	系统响应
	1. 用户向系统提交机票的取票通知, 发起取票请求	2. 系统获得取票通知的内容, 判断所取机票是否已截止取票, 若是, 则转至 2.1.1, 否则打印机票给用户
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	2.1.1 用户所取机票已截止取票	2.1.1 提示用户截止取票
后置条件:	系统打印机票	

用例名称:	管理个人信息	
用例 ID:	9	
角色:	旅客、工作人员	
用例说明:	用例主要功能是实现用户对个人信息的管理功能	
前置条件:	用户登录系统, 进入个人信息管理页面	
基本事件流:	参与者动作	系统响应
	1. 用户修改自己的个人信息, 并点击修改按钮	2. 系统判断用户的修改内容是否合法, 若合法则修改数据库中用户信息的记录; 否则转至 2.1.1
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	2.1.1 个人信息修改不合法	2.1.1 提示用户信息修改不合法
后置条件:	提示个人信息修改成功	

用例名称:	添加航班信息	
用例 ID:	10	
角色:	工作人员	
用例说明:	用例主要功能是实现工作人员添加航班的功能	
前置条件:	用户登录系统, 进入航班信息添加页面	
基本事件流:	参与者动作	系统响应
	1. 用户输入航班信息 2. 用户点击添加按钮	3. 系统判断输入的航班信息是否合法, 若合法则添加航班信息, 否则转至 3.1.1
其它事件流:	无	
异常事件流:	参与者动作	系统响应

	3.1.1 用户输入航班信息不合法	3.1.1 提示用户输入信息不合法
后置条件:	提示添加航班信息成功	

用例名称:	修改航班信息	
用例 ID:	11	
角色:	工作人员	
用例说明:	用例主要功能是实现工作人员修改航班信息的功能	
前置条件:	用户登录系统, 进入航班信息修改页面	
基本事件流:	参与者动作	系统响应
	1. 用户输入航班信息 2. 用户点击修改按钮	3. 系统判断修改的航班信息是否合法, 若合法则修改航班信息, 否则转至 3.1.1
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	3.1.1 用户输入航班信息不合法	3.1.1 提示用户输入信息不合法
后置条件:	提示修改航班信息成功	

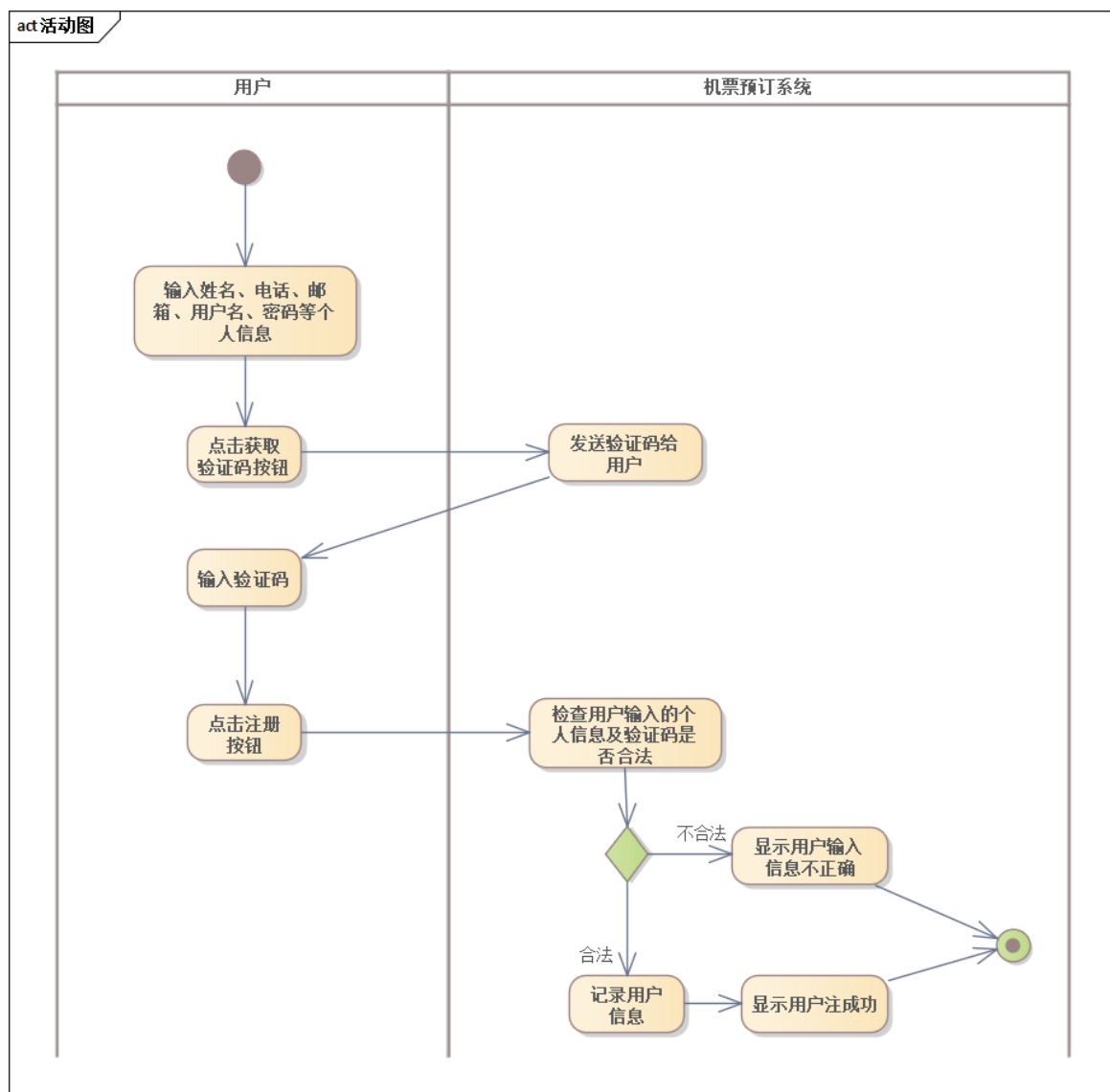
用例名称:	删除航班信息	
用例 ID:	12	
角色:	工作人员	
用例说明:	用例主要功能是实现工作人员删除航班的功能	
前置条件:	进入航班查询页面	
基本事件流:	参与者动作	系统响应
	1. 用户点击航班删除按钮	2. 系统从数据库中删除对应的航班信息
其它事件流:	无	
异常事件流:	参与者动作	系统响应
	无	无
后置条件:	转至航班删除成功页面	

3 活动图

我们根据用户的常用功能绘制活动图, 具体如下所示。

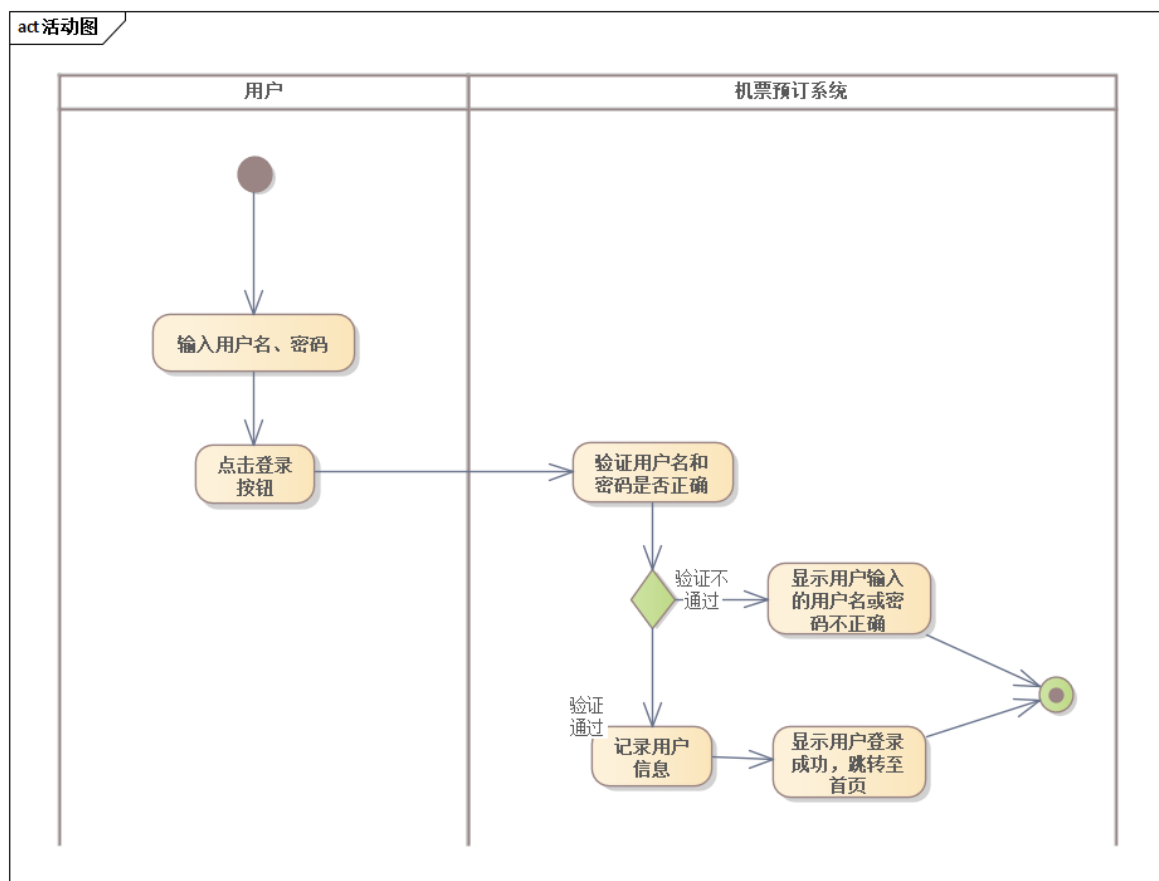
3.1 用户注册活动图

首先是对用户注册功能用例的描述, 如下图所示, 首先用户填写注册需要的姓名、电话、邮箱、用户名、密码等个人信息; 然后用户点击获取验证码按钮。之后系统会向用户发送一个验证码。用户输入验证码并点击注册按钮后系统会对用户的输入数据进行验证, 如用户名是否重复、验证码是否正确等等, 如果验证通过, 则记录用户的个人信息并提示用户注册成功; 否则提示用户注册失败。



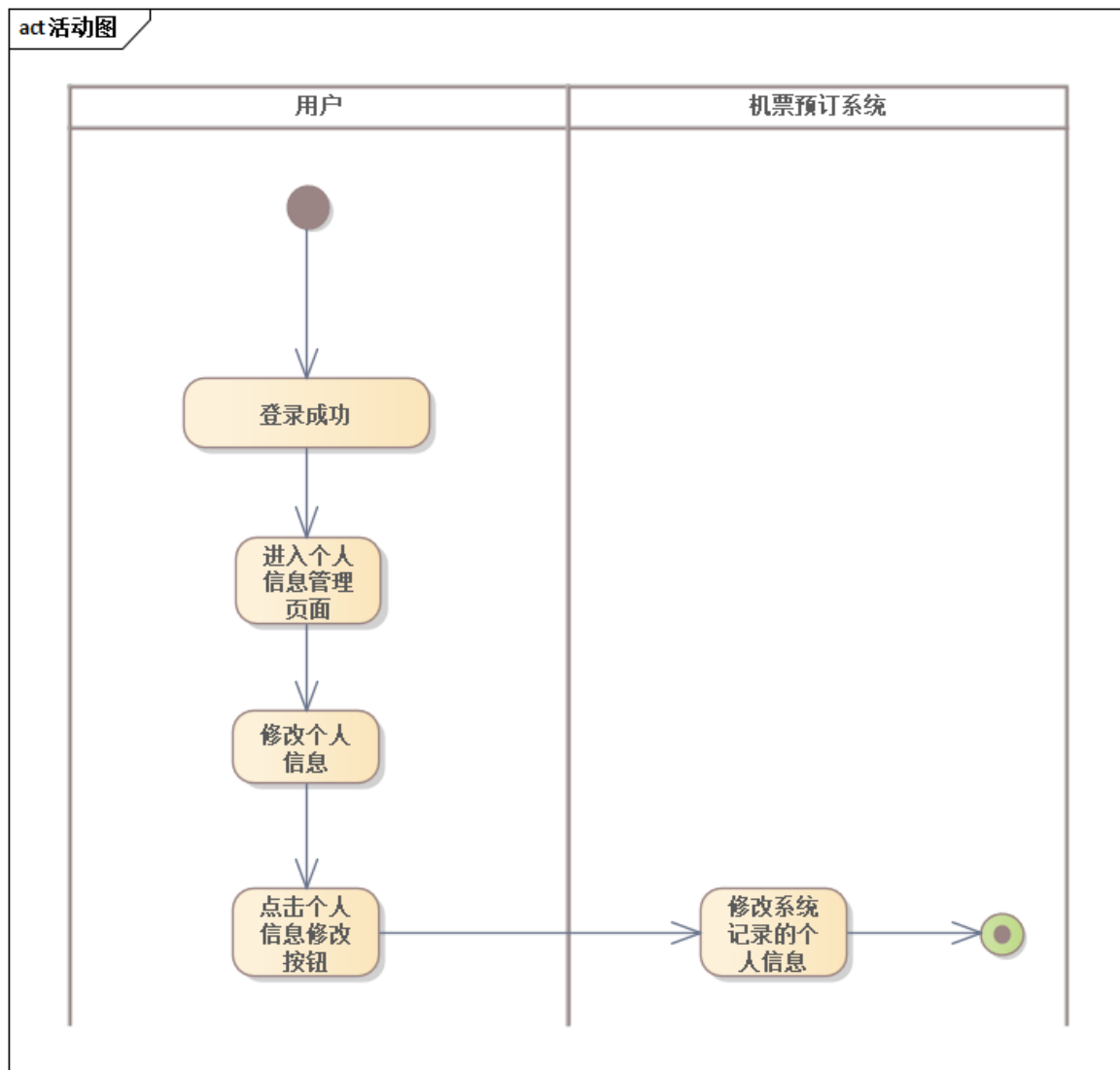
3.2 用户登录活动图

用户登录活动图如下，用户首先输入用户名和密码，然后点击登录按钮。之后系统判断用户输入的用户名和密码是否正确，如果正确则登录成功，转至系统首页，否则提示用户用户名或密码错误，登录失败。



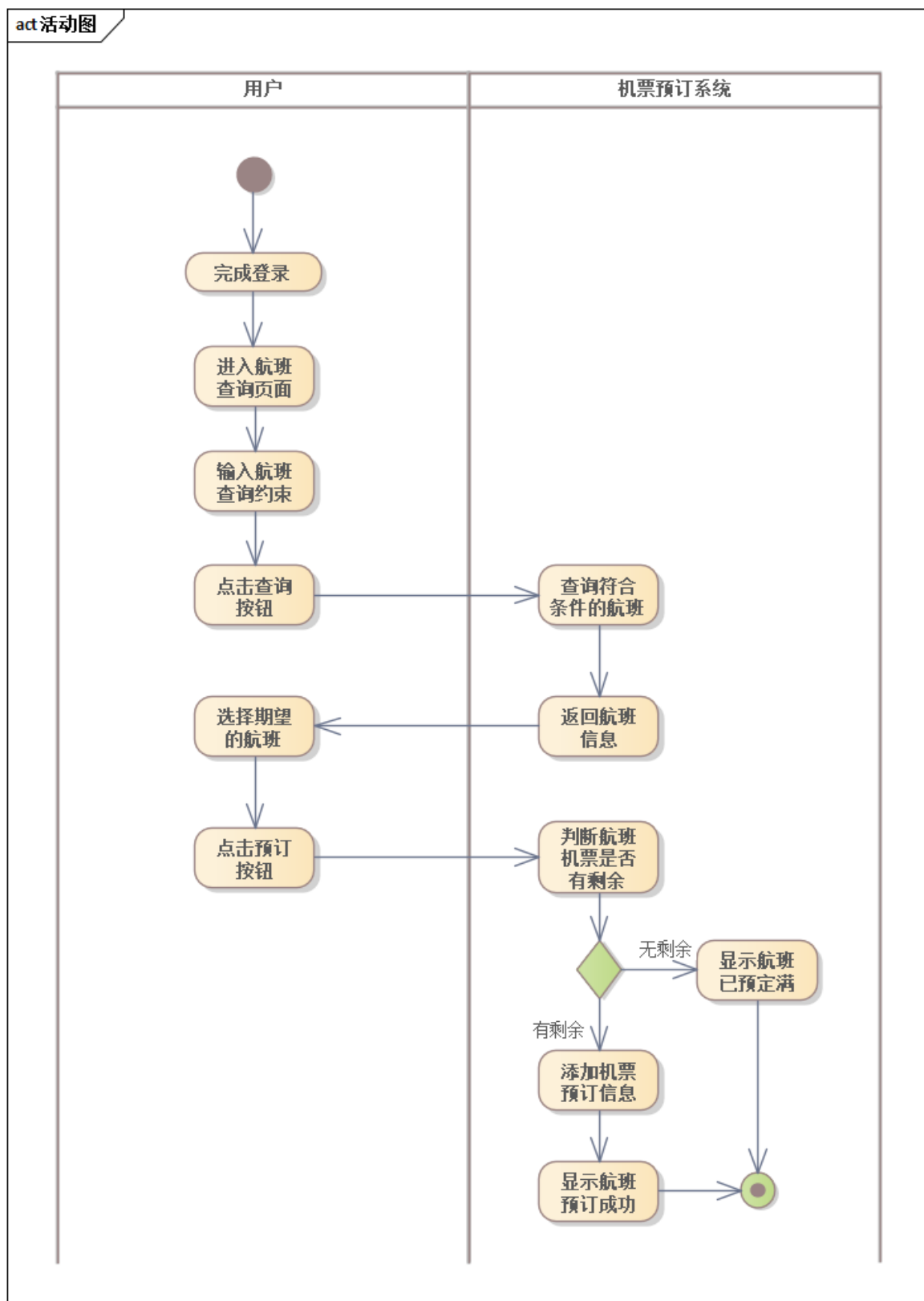
3.3 用户管理个人信息活动图

用户管理个人信息活动图如下，用户首先完成登录，然后进入个人信息管理页面。之后用户可以对个人信息进行修改，修改完成后点击个人信息修改按钮。最后系统接收到用户的修改数据，修改系统中记录的用户信息。



3.4 旅客订票活动图

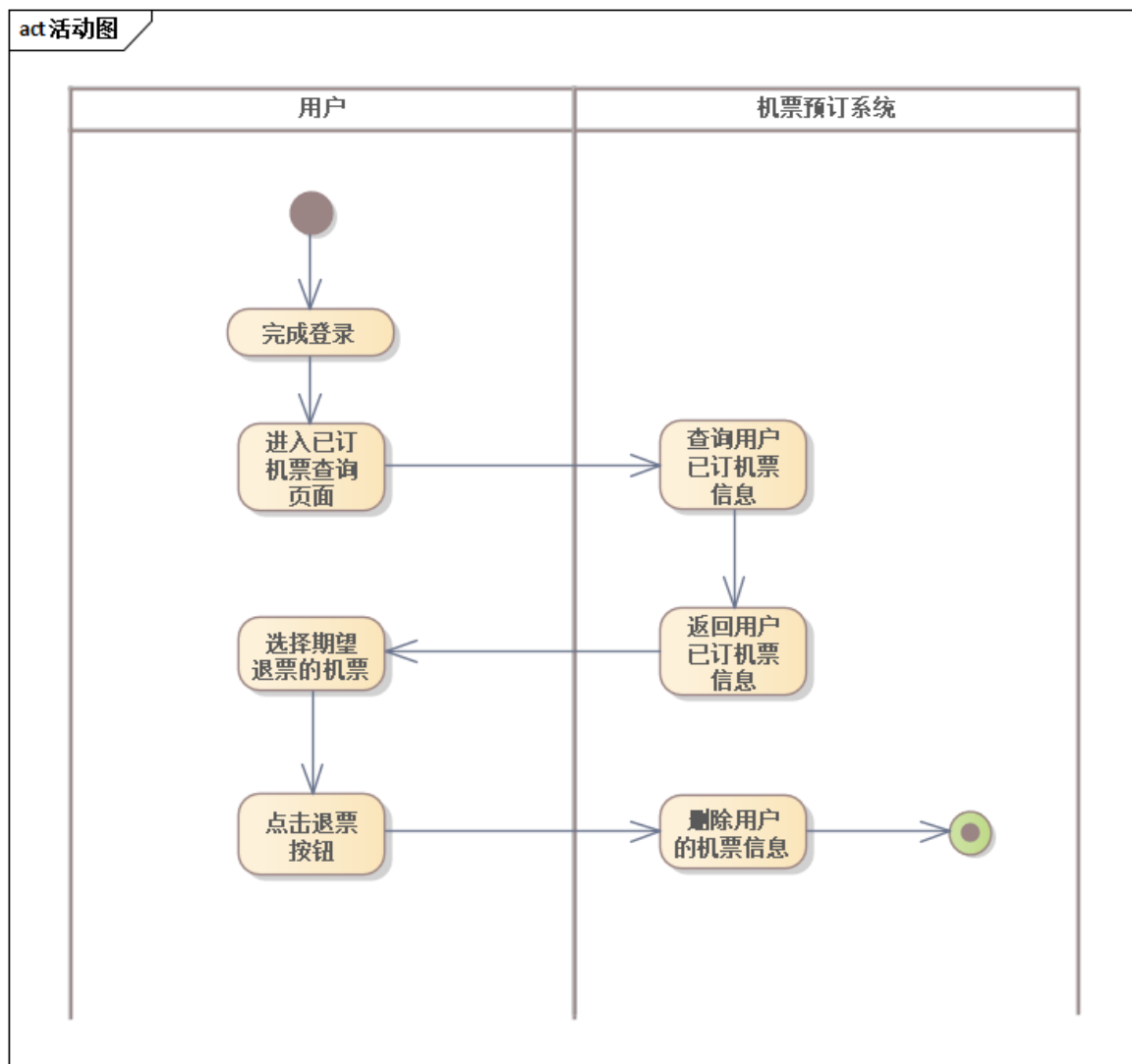
旅客订票活动图如下，用户首先完成登录，然后进入航班查询页面并输入自己期望的航班的一些查询约束信息。点击查询按钮后，系统会查询所有符合用户要求的航班的信息，并将这些航班信息返回给用户。之后用户可以选择自己期望预订的航班，点击预订按钮后，系统会判断对应的航班是否还有剩余座位，如果没有，则提示用户航班已预订满；否则记录机票预订信息，并提示用户航班预订成功。



3.5 旅客退票活动图

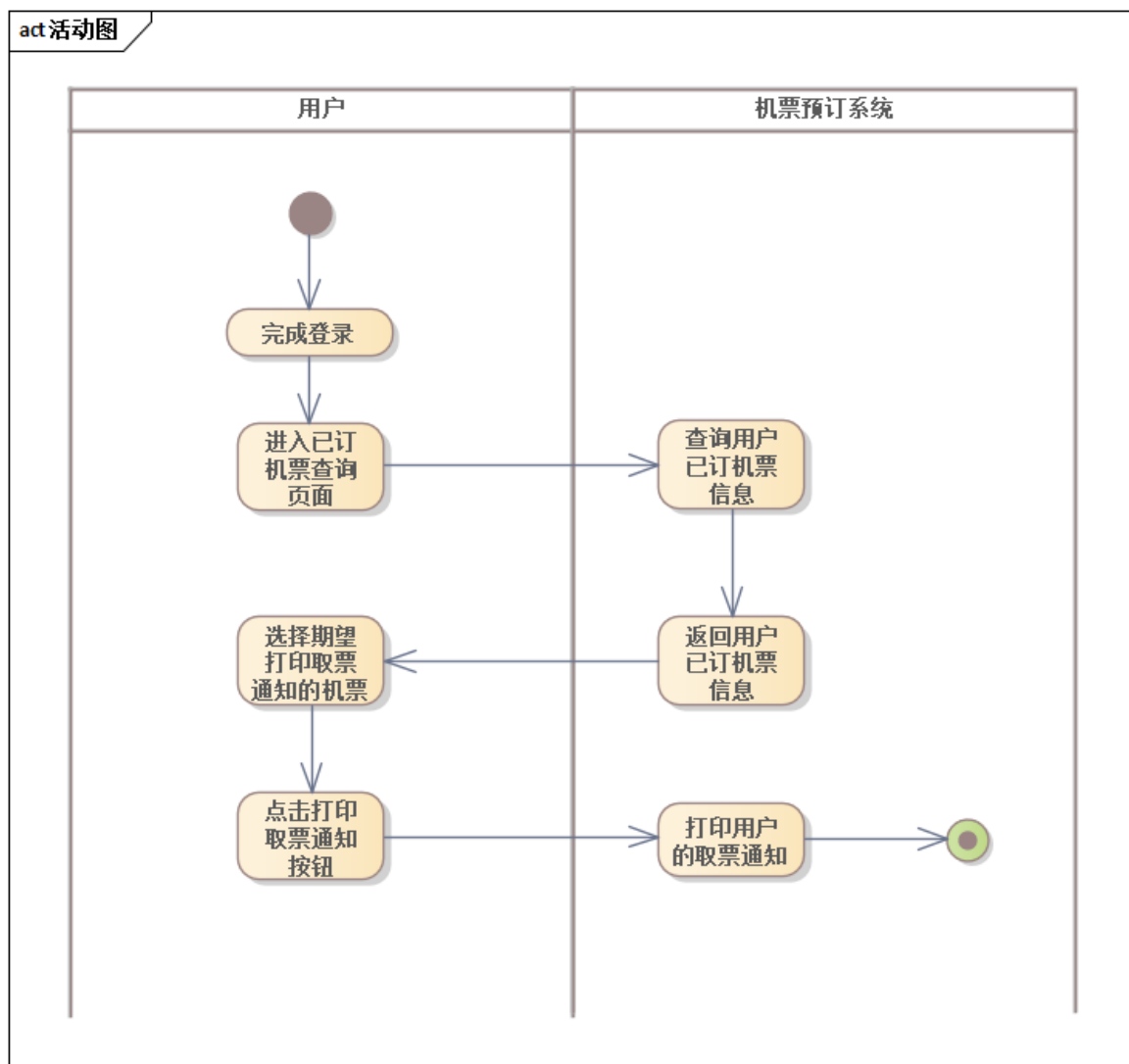
旅客退票活动图如下，用户首先完成登录，然后进入已订机票查询页面，系统将会自动将该用户已经预订的所有机票信息返回给用户。然后用户选择期望退票的机票，点击退票按钮后系统将会取消

该机票的预订。



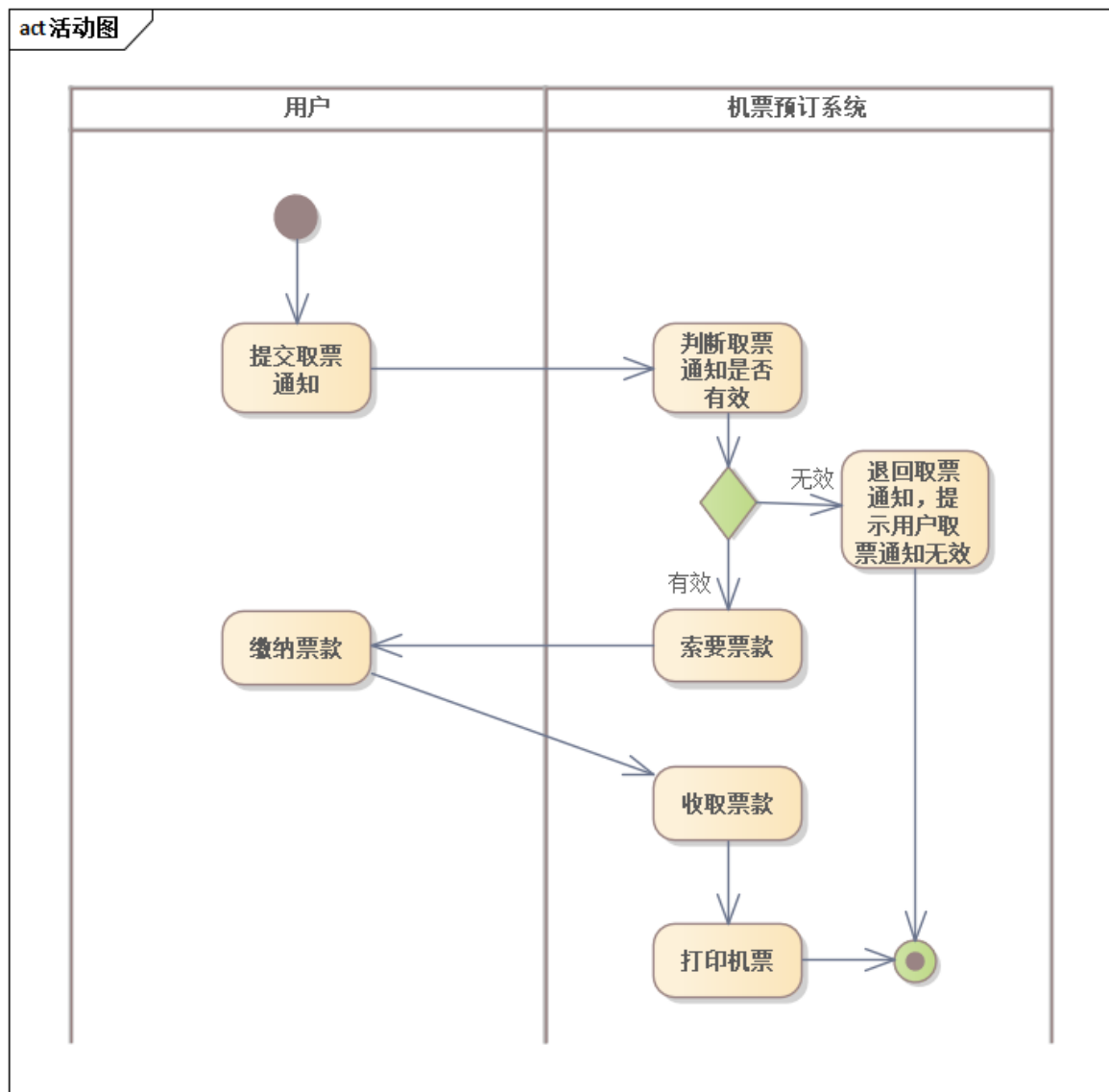
3.6 旅客打印取票通知活动图

旅客打印取票通知活动图如下，用户首先完成登录，然后进入已订机票查询页面，系统将会自动将该用户已经预订的所有机票信息返回给用户。然后用户选择期望打印取票通知的机票，点击打印取票通知按钮后系统将会打印该机票的取票通知。



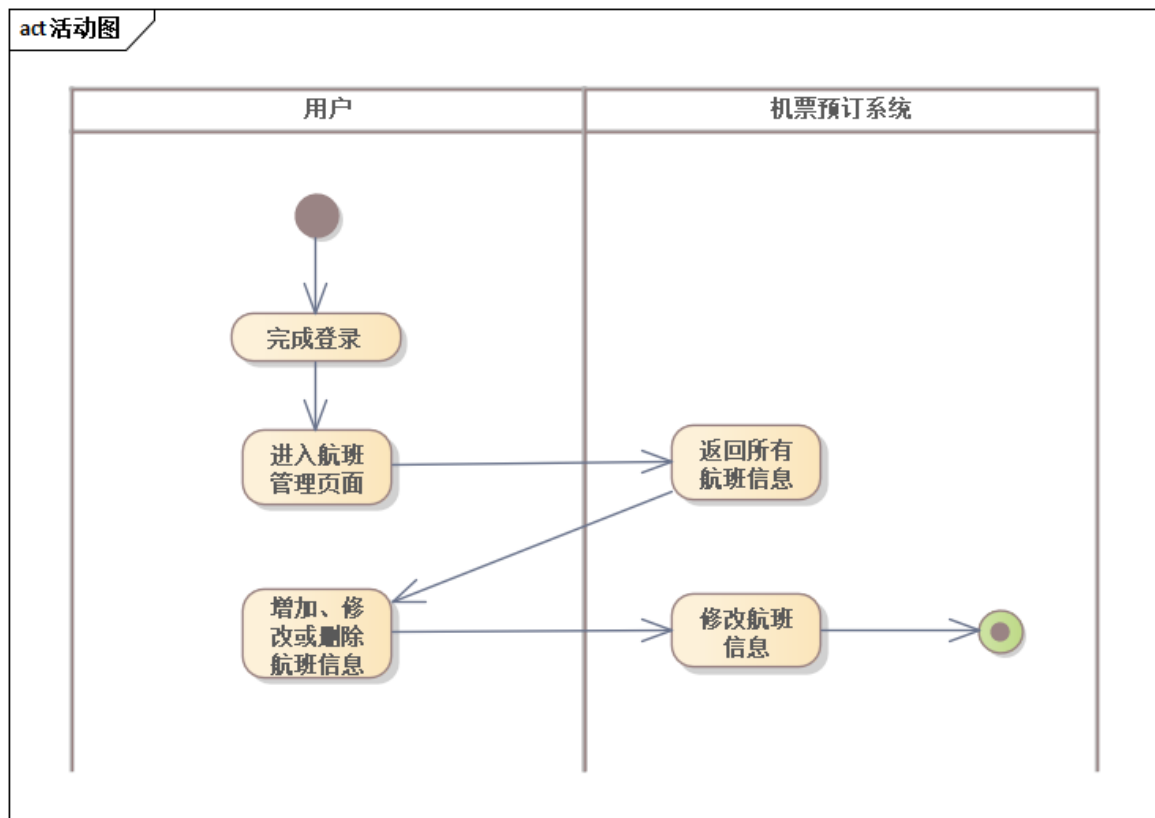
3.7 旅客取票活动图

旅客取票活动图如下，首先旅客向系统提交取票通知，系统判断取票通知是否仍然有效，若无效则提示用户取票通知无效；若有效则向用户索要票款，用户提交票款后系统打印机票给用户。

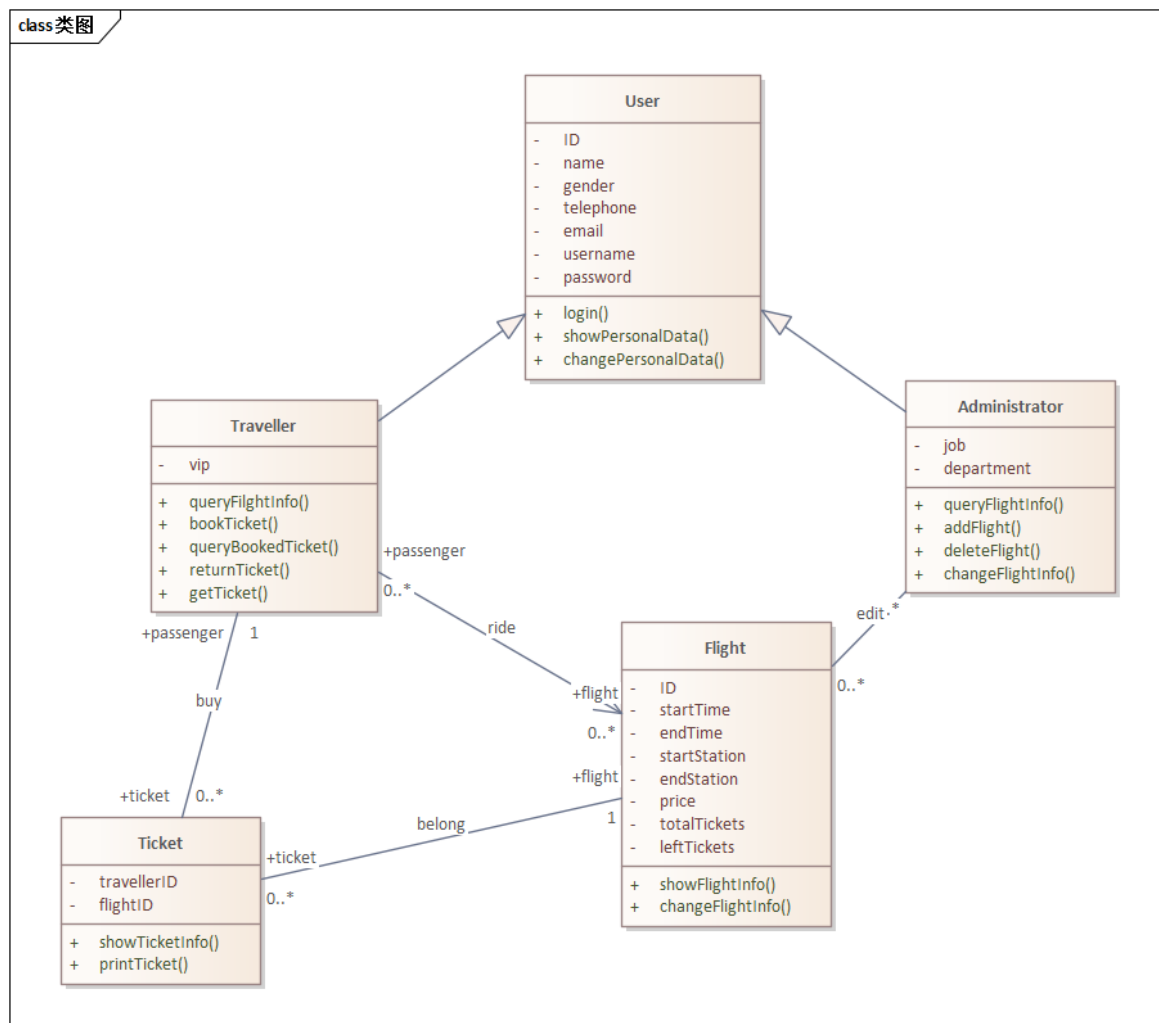


3.8 工作人员管理航班信息活动图

工作人员管理航班信息活动图如下，工作人员首先登录系统，系统将自动返回所有航班信息。然后工作人员可以选择增添、修改、删除航班信息，系统将保存对于航班信息的修改。



4 类图



类图设计如上图所示，下面对于每个类的属性和操作进行具体的解释。

4.1 类信息介绍

User 类，User 类为旅客类和工作人员类的基类，其属性和操作如下。

• 属性

- ID: 用户 ID，唯一标识一个用户
- name: 用户姓名
- gender: 用户性别
- telephone: 用户电话
- email: 用户邮箱
- username: 用户名，用于登录系统
- password: 密码，用于登录系统

- 操作

- login: 登录
- showPersonalData: 展示个人信息
- changePersonalData: 更改个人信息

Traveller 类，旅客类，属性和操作如下。

- 属性

- vip: 是否是 vip 旅客

- 操作

- queryFlightInfo: 查询航班信息
- bookTicket: 订票
- queryBookedTicket: 查询已订机票信息
- returnTicket: 退票
- getTicket: 取票

Administartor 类，工作人员类，属性和操作如下。

- 属性

- job: 职位
- department: 部门

- 操作

- queryFlightInfo: 查询航班信息
- addFlight: 添加航班
- deletFlight: 删除航班
- changeFlightInfo: 修改航班信息

Flight 类，航班类，属性和操作如下。

- 属性

- ID: 航班 ID，唯一标识一个航班
- startTime: 航班出发时间
- endTime: 航班到达时间
- startStation: 航班出发站点
- endStation: 航班到达站点
- price: 航班价格
- totalTickets: 航班总机票数

- leftTickets: 航班剩余机票数

- 操作

- showFlightInfo: 展示航班信息
- changeFlightInfo: 更改航班信息

Ticket 类, 机票类, 属性和操作如下。

- 属性

- travellerID: 旅客 ID
- flightID: 航班 ID

- 操作

- showTicketInfo: 打印机票信息
- printTicket: 打印机票

4.2 类间关系

类间关系如下:

1. Traveller 类、Administartor 类和 User 类为继承关系。后者是前二者的基类。
2. Traveller 类和 Ticket 类为关联关系, 一个旅客可以预订多张机票, 但一张机票只能被一个旅客预订。
3. Traveller 类和 Flight 类为关联关系, 一个旅客可以乘坐多次航班, 一次航班可以搭载多名旅客。
4. Administartor 类和 Flight 类为关联关系, 一个工作人员可以编辑多个航班信息, 一个航班可以被多个工作人员编辑。

5 顺序图

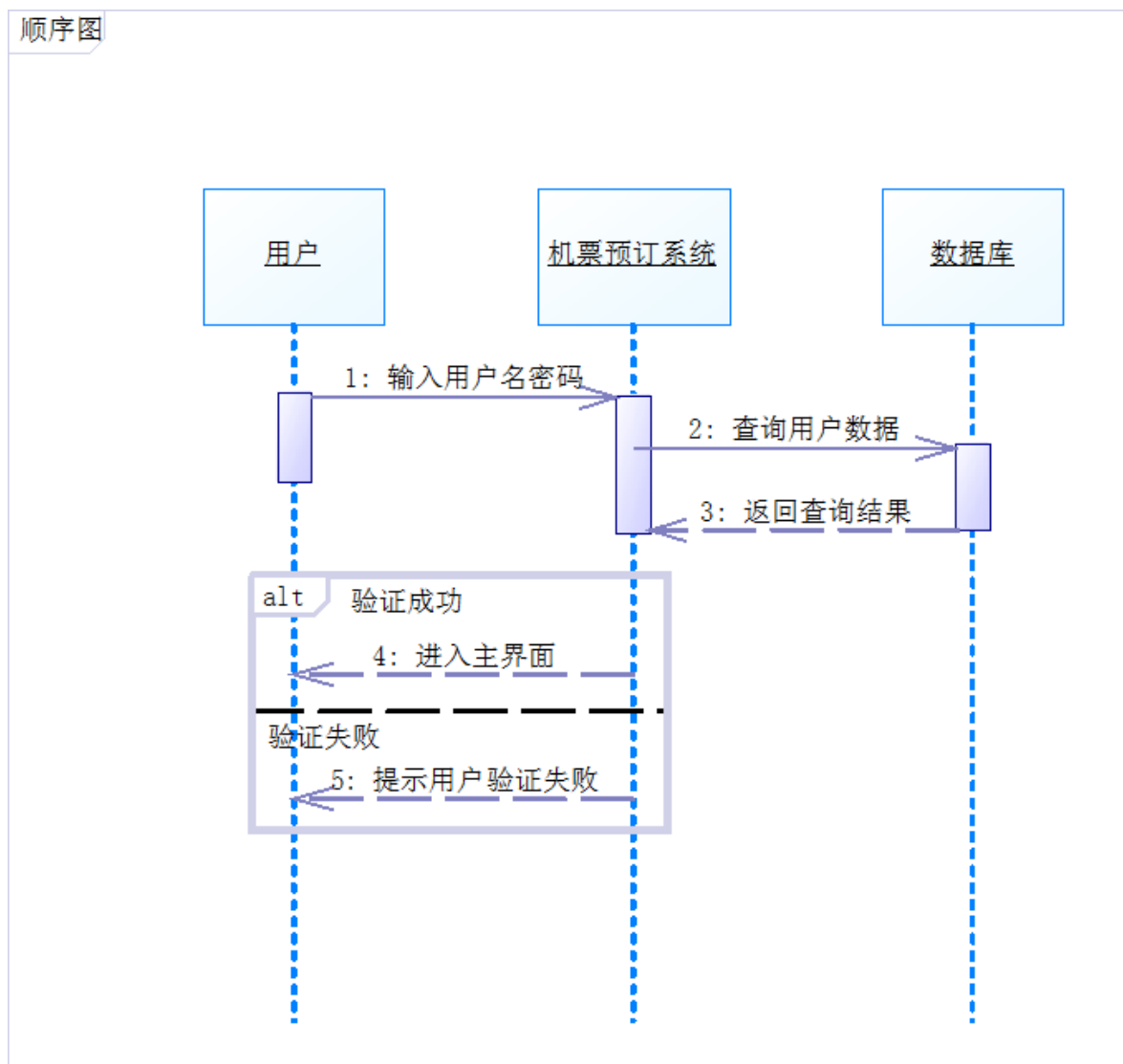
下面我们对一些关键的交互动作绘制顺序图, 如下所示。

5.1 用户登录顺序图

用户登录顺序图如下所示, 流程如下:

1. 用户向系统输入用户名和密码;
2. 系统向数据库发出查询请求, 查询对应的用户信息;
3. 数据库返回用户的信息;
4. 系统对用户输入的密码进行验证;
5. 如果验证成功, 返回给用户系统的主页面;

6. 如果验证失败，提示用户验证失败。

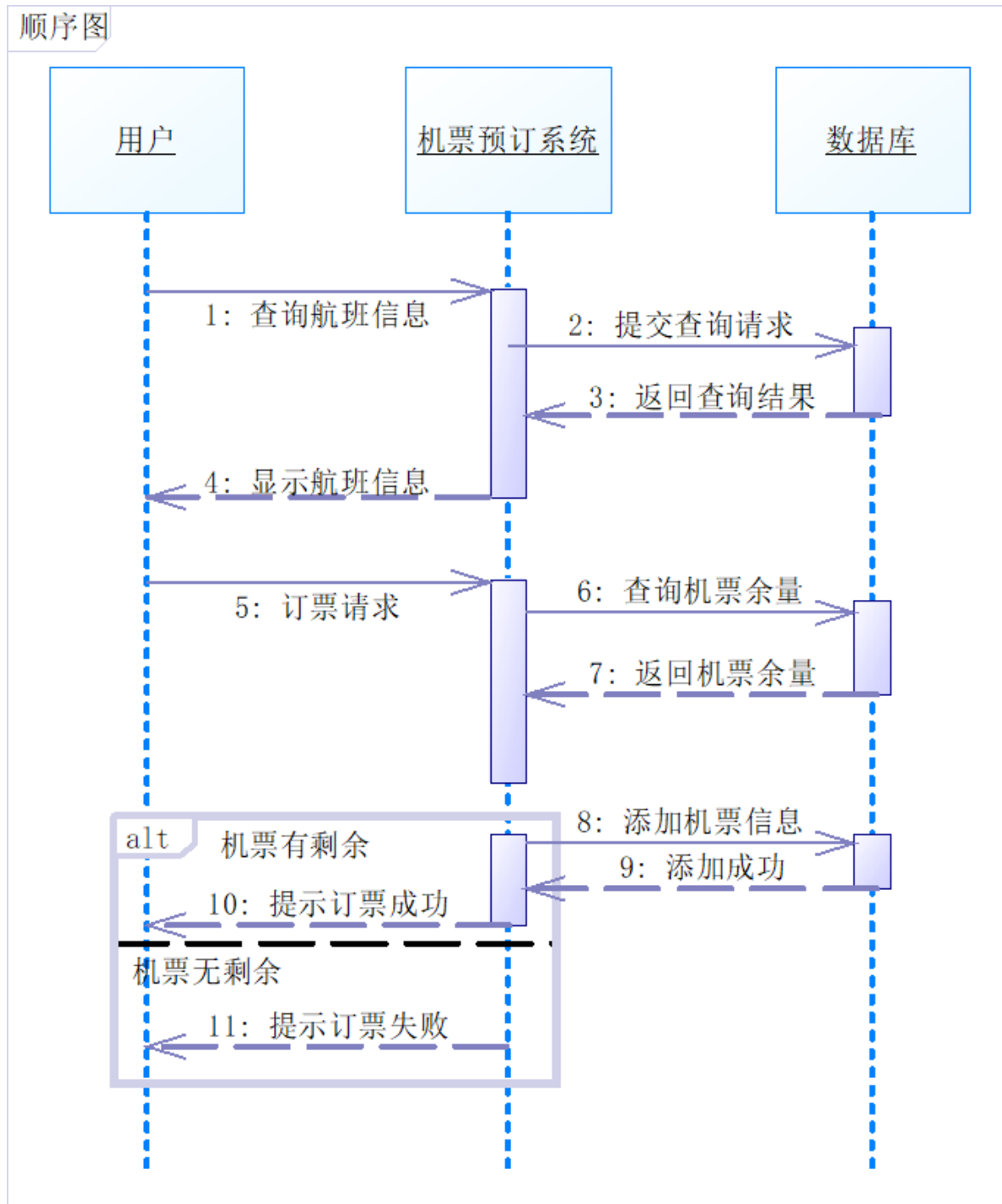


5.2 旅客订票顺序图

旅客订票顺序图如下所示，流程如下：

1. 旅客首先向系统发起航班请求查询；
2. 系统向数据库发起查询请求；
3. 数据库将所需的航班信息返回给系统；
4. 系统向用户展示航班信息；
5. 用户选择期望的航班，向系统发起订票请求；
6. 系统向数据库查询航班剩余机票数；
7. 数据库返回剩余机票数；

8. 如果剩余机票数大于零，则系统向数据库添加机票信息;
9. 机票信息添加完成后提示用户订票成功;
10. 如果剩余机票数为零，则系统提示用户订票失败。

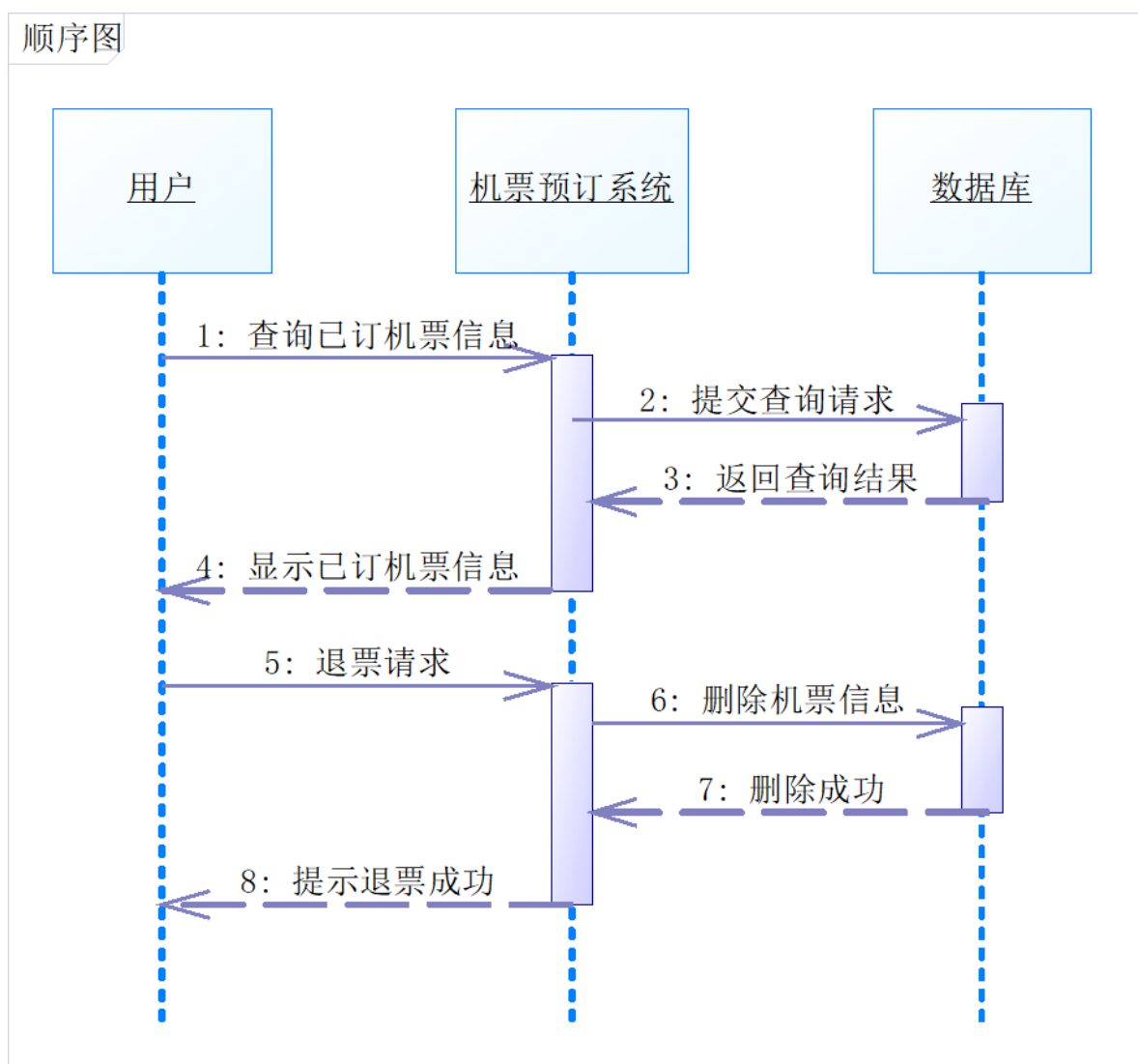


5.3 旅客退票顺序图

旅客退票顺序图如下所示，流程如下:

1. 用户向系统发起请求，查询已订机票的信息;

2. 系统向数据库发起查询请求;
3. 数据库将用户所有已订机票信息返回给系统;
4. 系统向用户展示所有已订机票的信息;
5. 用户选择期望退票的机票, 向系统发起退票请求;
6. 系统接受退票请求, 向数据库发起删除机票的请求;
7. 数据库删除完成后, 向系统返回删除成功信息;
8. 系统提示用户退票成功。



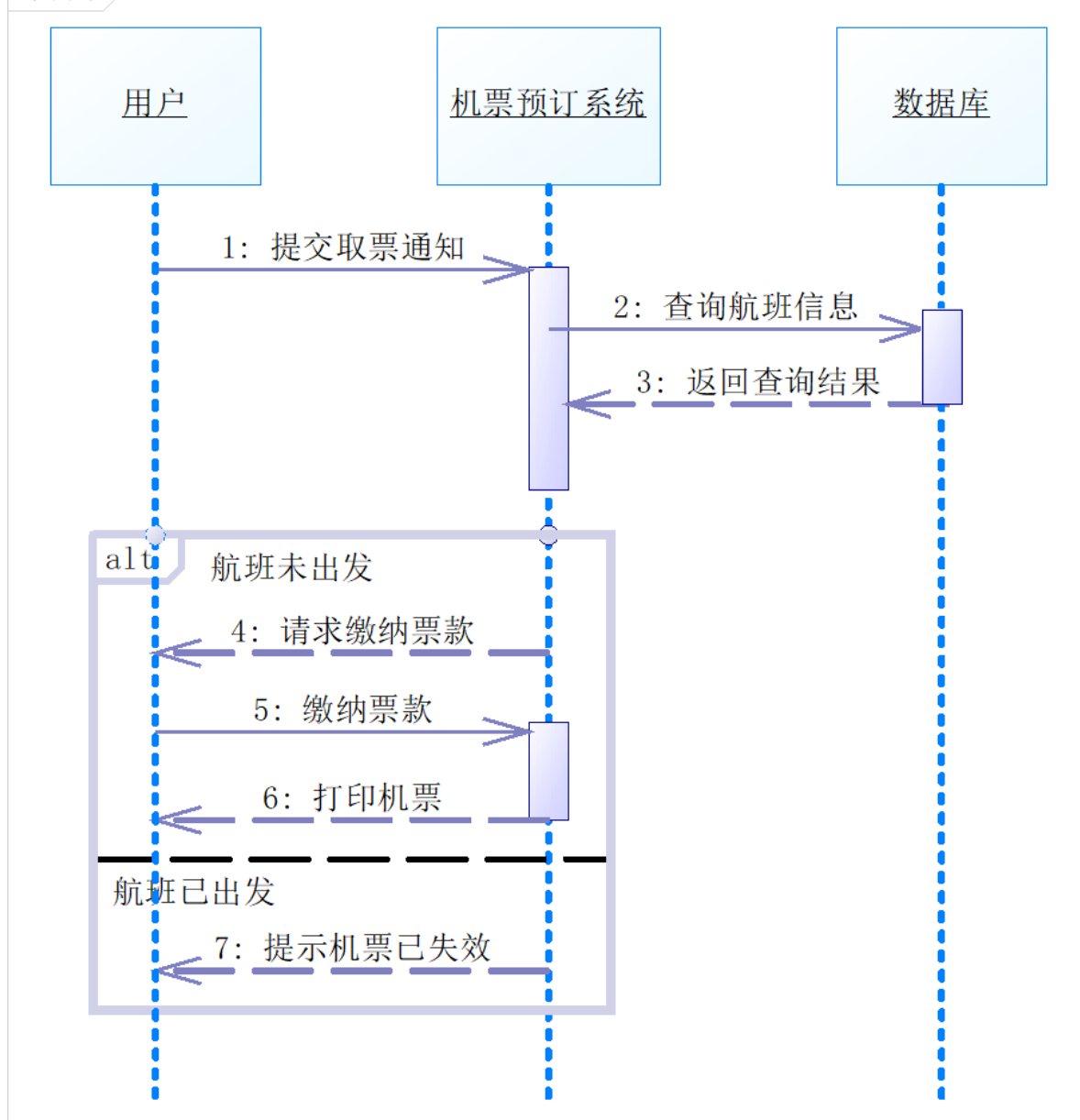
5.4 旅客取票顺序图

旅客取票顺序图如下所示, 流程如下:

1. 用户向系统提交取票通知;
2. 系统向数据库发起查询请求, 查询对应的航班信息;

3. 数据库返回给系统航班信息;
4. 系统判断航班是否已经出发, 如果未出发, 则请求用户缴纳票款;
5. 用户向系统缴纳票款;
6. 系统打印机票给用户;
7. 如果航班已经出发, 则系统提示用户机票已失效。

顺序图

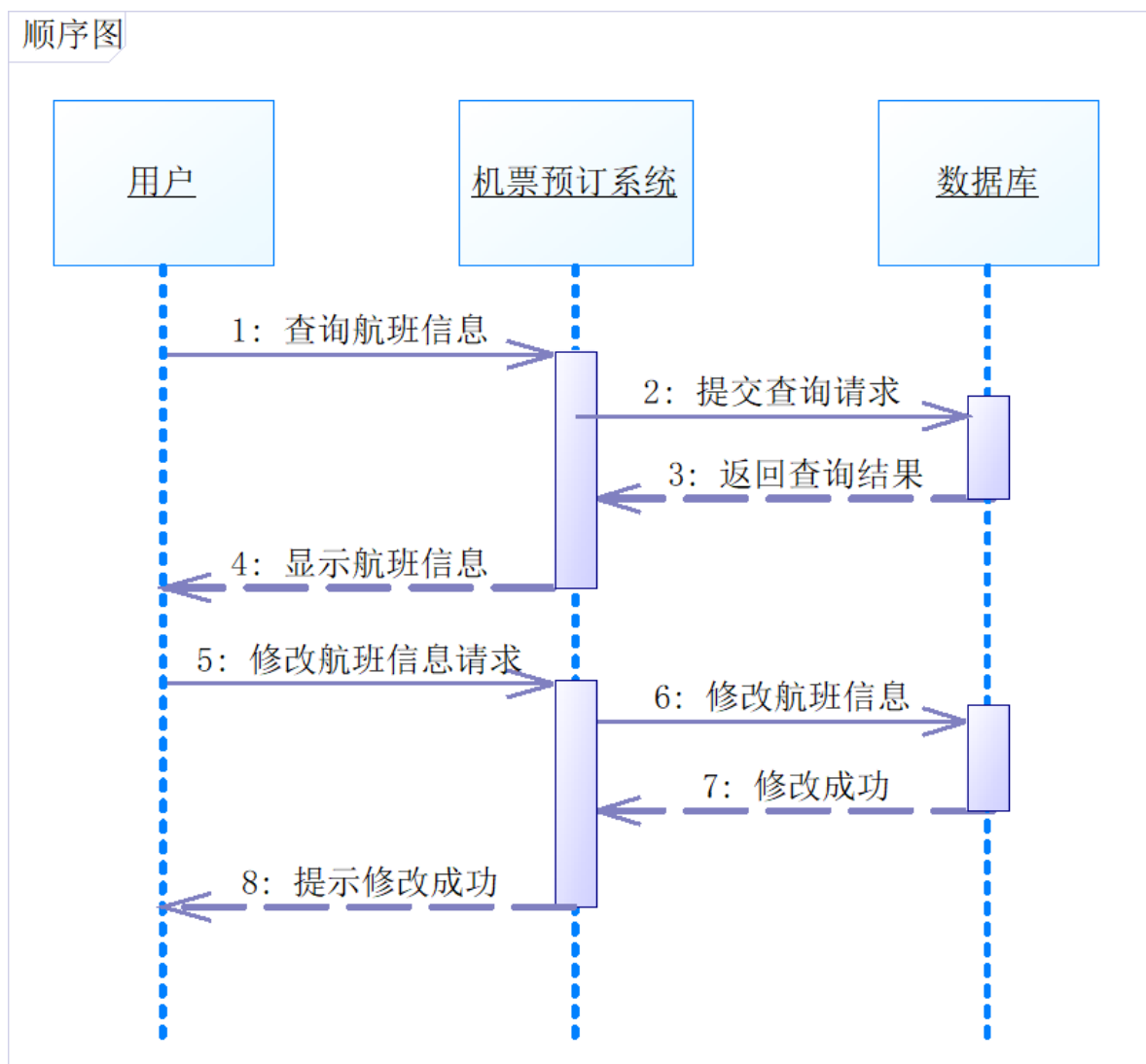


5.5 工作人员修改航班信息顺序图

工作人员修改航班信息顺序图如下所示, 流程如下:

1. 用户向系统发起查询航班信息请求;

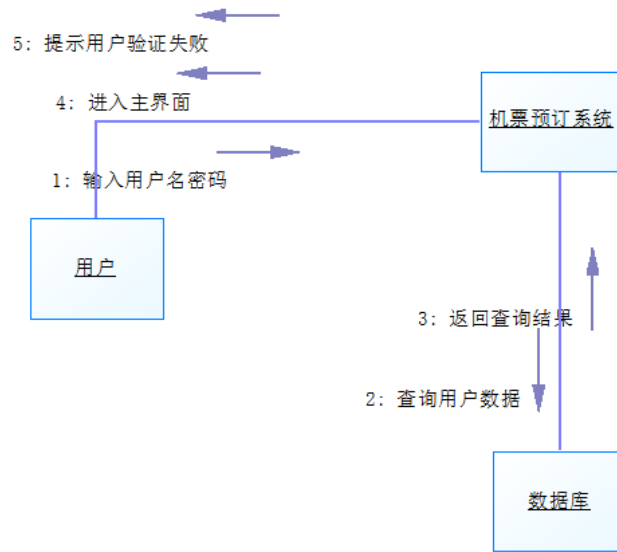
2. 系统查询数据库对应的航班信息;
3. 数据库返回航班信息给系统;
4. 系统向用户展示航班信息;
5. 用户修改航班信息, 并提交给系统;
6. 系统接受修改, 向数据库提交修改请求;
7. 数据库修改成功, 返回给系统提示信息;
8. 系统提示用户航班信息修改成功。



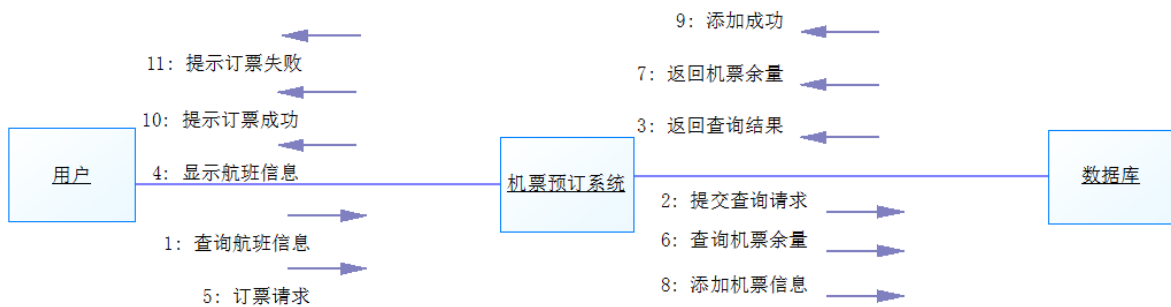
6 协作图

协作图如下, 由于已经在顺序图部分进行了较为详细的说明, 这里不再赘述协作图。

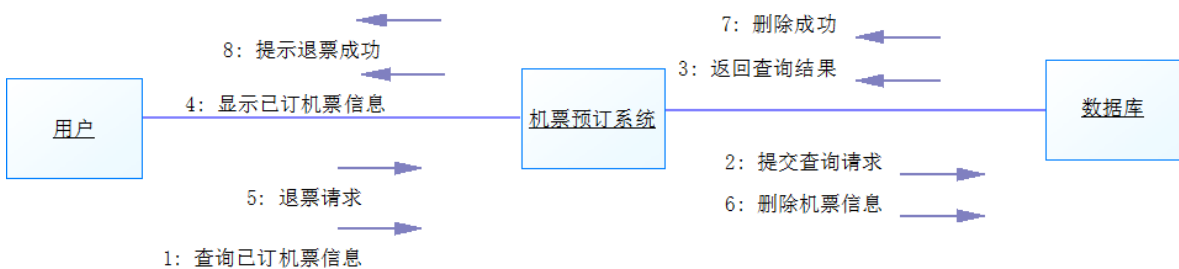
6.1 旅客登录协作图



6.2 旅客订票协作图



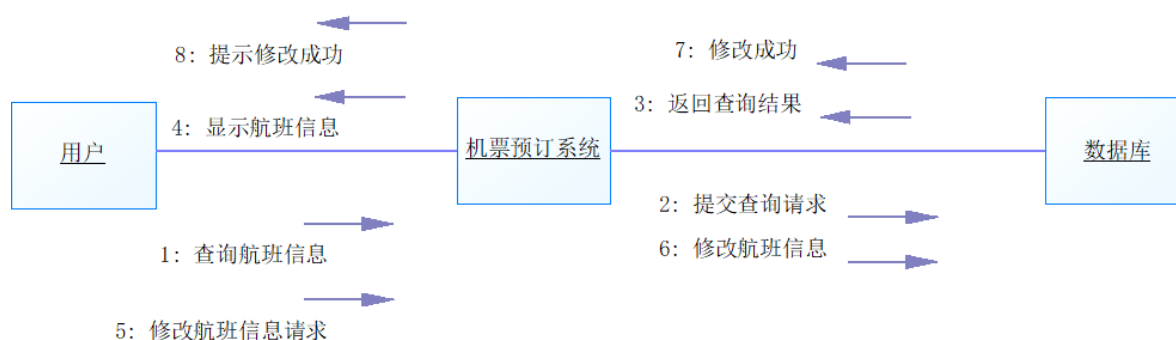
6.3 旅客退票协作图



6.4 旅客取票协作图



6.5 工作人员修改航班信息协作图



7 状态图

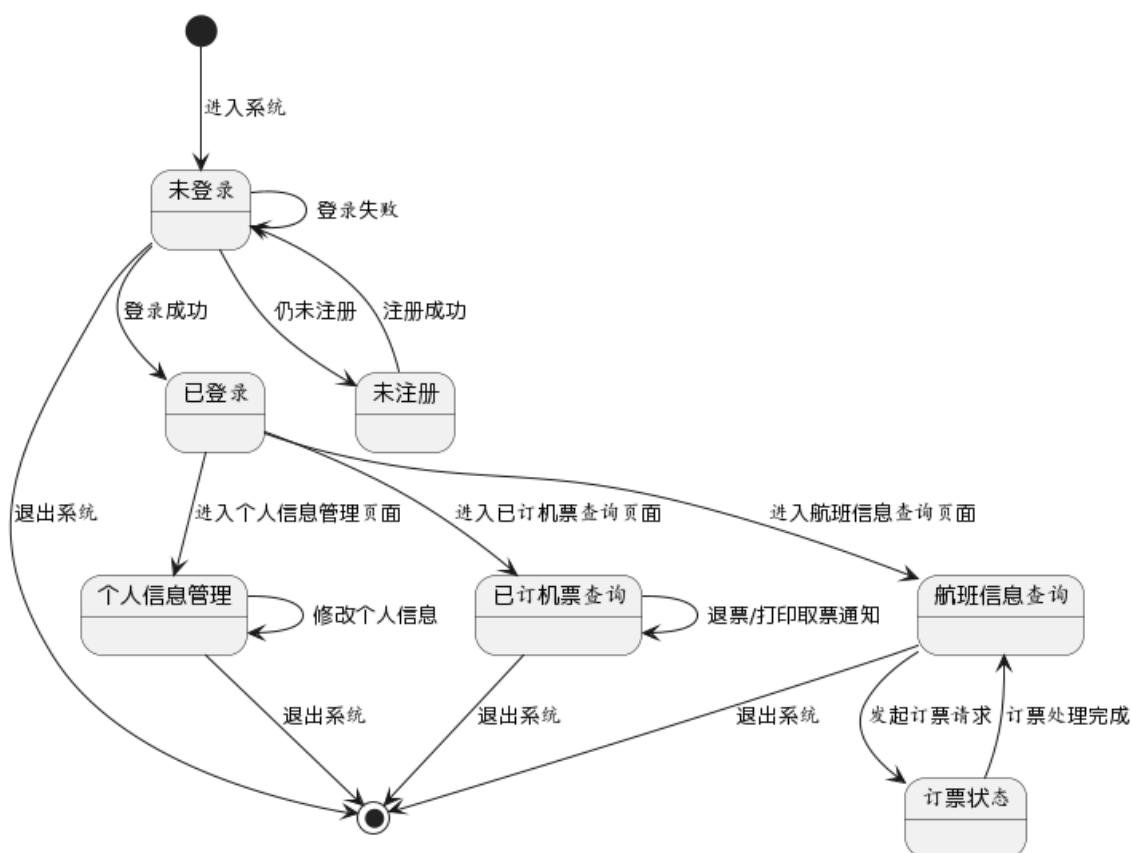
7.1 旅客状态图

旅客状态图如下，具体状态分为:

1. 未登录状态: 旅客进入系统后处于未登录状态，需要输入用户名和密码进行登录操作，如果登录成功，则进入已登录状态，否则仍处于未登录状态。如果旅客还未注册账号，则转至未注册状态。
2. 未注册状态: 旅客还未注册账号，旅客注册成功后进入未登录状态。
3. 已登录状态: 旅客成功登录进入系统，此时旅客可以进行不同的操作如进入个人信息管理页面、进入已订机票查询页面、进入航班信息查询页面，旅客的状态也会相应的转为个人信息管理状态、已订机票查询状态、航班信息查询状态。
4. 个人信息管理状态: 在此状态下，旅客可以对自己的个人信息进行查阅和修改，修改之后仍处于个人信息管理状态。

5. 已订机票查询状态：在此状态下，旅客可以浏览自己已订机票的信息，并选择自己期望退票的机票进行退票操作或者选择已订机票取票通知的打印操作，执行完这些操作后仍处于已订机票查询状态。
6. 航班信息查询状态：在此状态下，旅客可以查询自己期望的航班信息，并选择合适的航班进行订票操作，当发起订票请求后，将转至订票状态。
7. 订票状态：在此状态下，系统会对旅客的订票请求进行处理，当处理完成后，系统将结果返回给旅客，旅客回到航班信息查询状态。

机票预订系统旅客状态图



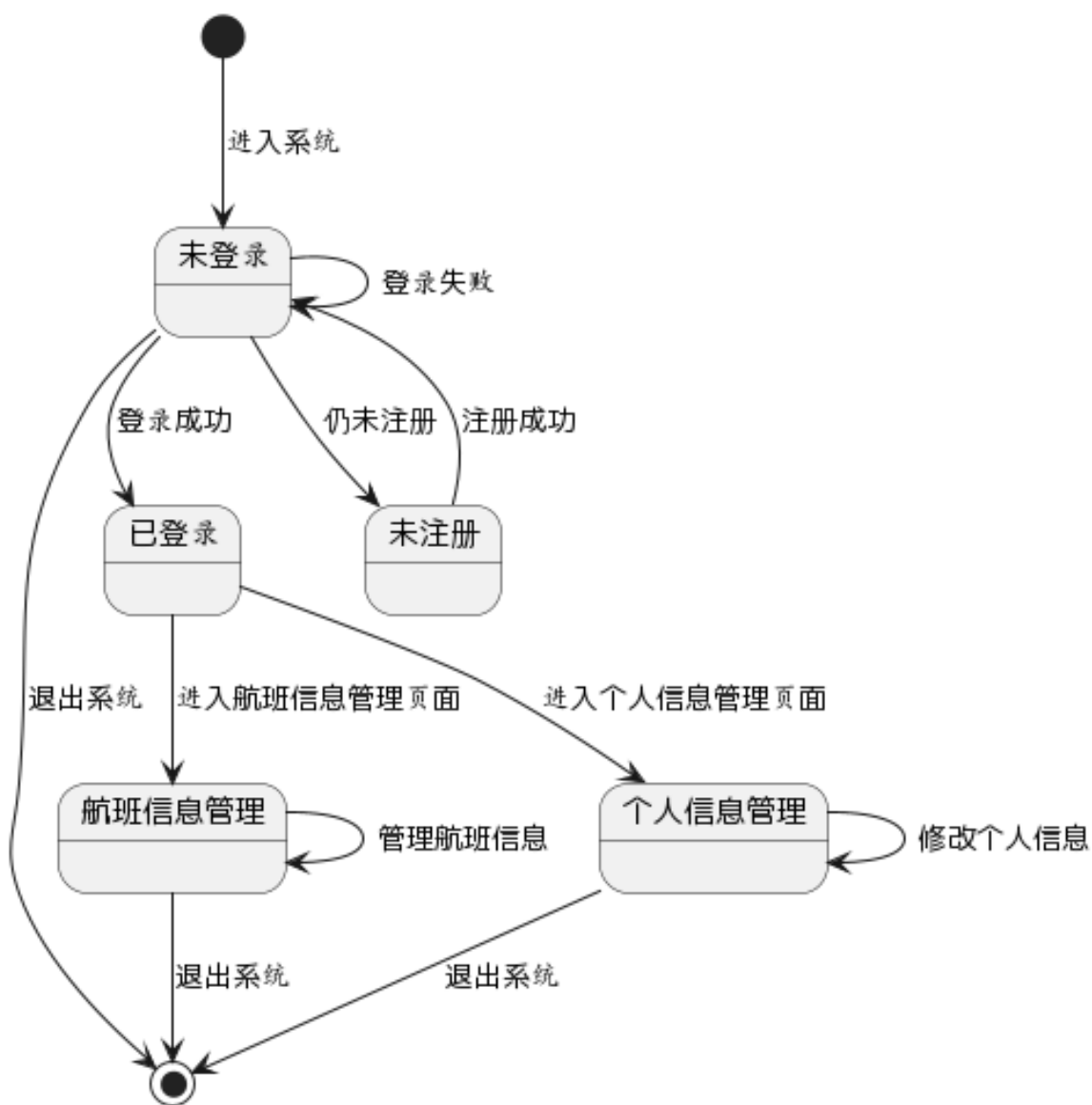
7.2 工作人员状态图

工作人员状态图如下，具体状态分为：

1. 未登录状态：工作人员进入系统后处于未登录状态，需要输入用户名和密码进行登录操作，如果登录成功，则进入已登录状态，否则仍处于未登录状态。如果工作人员还未注册账号，则转至未注册状态。
2. 未注册状态：工作人员还未注册账号，工作人员注册成功后进入未登录状态。
3. 已登录状态：工作人员成功登录进入系统，此时工作人员可以进行不同的操作如进入个人信息管理页面、进入航班信息管理页面，工作人员的状态也会相应的转为个人信息管理状态、航班信息管理状态。

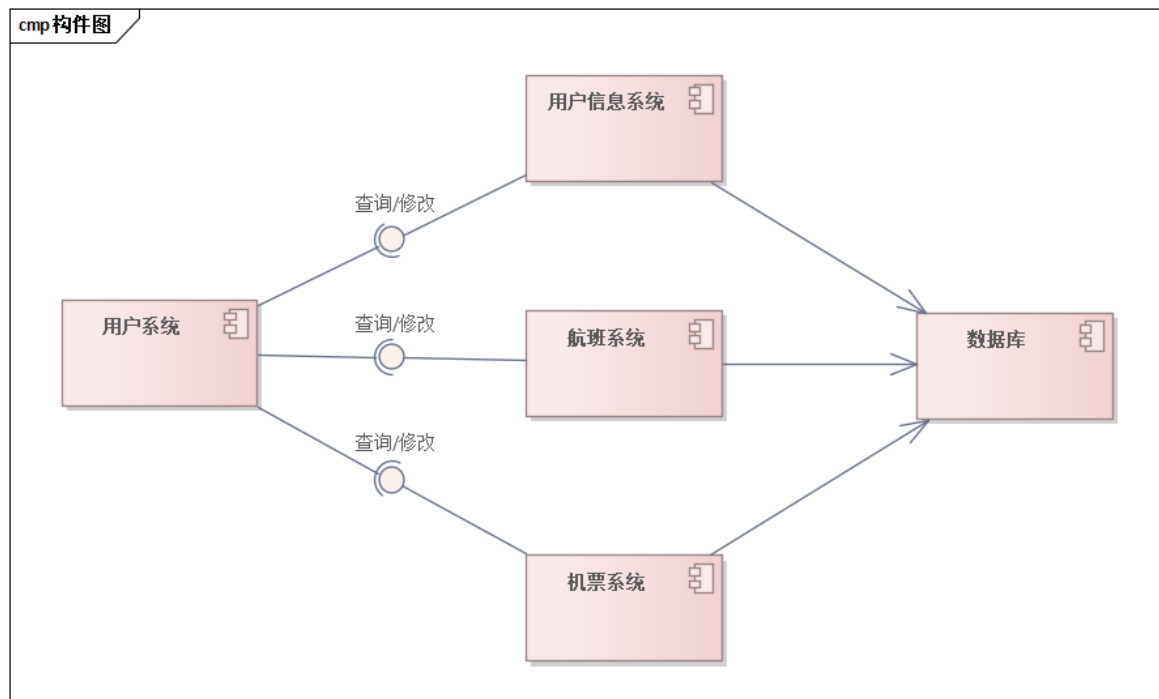
4. 个人信息管理状态：在此状态下，工作人员可以对自己的个人信息进行查阅和修改，修改之后仍处于个人信息管理状态。
5. 航班信息查询状态：在此状态下，工作人员可以浏览所有航班信息，并进行航班信息的增删改查操作。

机票预订系统工作人员状态图



8 构件图

构件图如下，整个系统分为 4 个部分，分别是用户系统、用户信息系统、航班系统、机票系统，其中用户系统依赖于其余三个系统提供的数据库接口。用户信息系统、航班系统、机票系统则依赖于数据库进行数据的存储。



9 部署图

机票预订系统使用 B/S 模型，部署图如下所示，用户通过浏览器向 web 服务器发起请求，web 服务器返回结果给浏览器。系统使用一个数据库服务器来提供数据存储服务，具体使用 MySQL 数据库，该数据库服务器和 web 服务器通过局域网连接在一起。

