



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

软件工程

机票预定系统

穆禹宸 2012026

年级：2020 级

专业：信息安全、法学双学位班

指导教师：徐思涵

2023 年 4 月 25 日

目录

一、用例图	1
二、活动图	4
三、类图	9
四、顺序图	12
五、协作图	18
六、状态图	21
七、构件图	23
八、部署图	24

一、 用例图

首先，根据题目要求，在本系统之中至少应该有如下用例：

- 注册账号：旅客或工作人员可以通过注册功能创建一个新账户，以便于使用系统的其他功能。
- 登录账号：用户可以通过登录功能进入系统。用户需要输入他们的用户名和密码才能够登录。
- 查询航班信息：用户可以通过查询航班信息功能来查找可用的航班信息。用户可以根据航班号、起点、终点、出发日期等信息进行查询。查询结果会显示出符合条件的航班列表。
- 选择航班：用户可以根据查询结果中显示的航班信息选择一条适合自己的航班。用户可以在航班列表选择一个航班，并查看航班的详细信息。
- 预订航班：用户可以通过预订航班功能进行机票的预订。用户需要输入旅客的信息，如姓名、出生日期、联系方式等。预订成功后，系统将生成一个订单号。
- 打印取票通知和账单：系统将自动生成一份取票通知和账单，用户可以通过打印功能将其打印出来。
- 交款取票：用户需要在起飞前凭取票通知交款取票。系统将校验取票通知和账单的信息是否一致，一致后即可打印机票给旅客。
- 核对机票：工作人员可以通过核对机票功能，检查乘客的机票信息是否正确。
- 退票：如果用户需要退票，可以通过退票功能来申请退票。用户需要输入订单号并填写退票原因。系统将根据航空公司的退票规定计算退票费用，然后将退款金额返回给用户。
- 改签：如果用户需要改签，可以通过改签功能来修改预订信息。用户需要输入订单号并选择新的航班信息。系统将根据航空公司的改签规定计算费用，然后提示用户是否同意支付费用。如果用户同意支付费用，则系统将修改预订信息并生成新的订单号。

经过设计之后，得到的用例图如下所示：

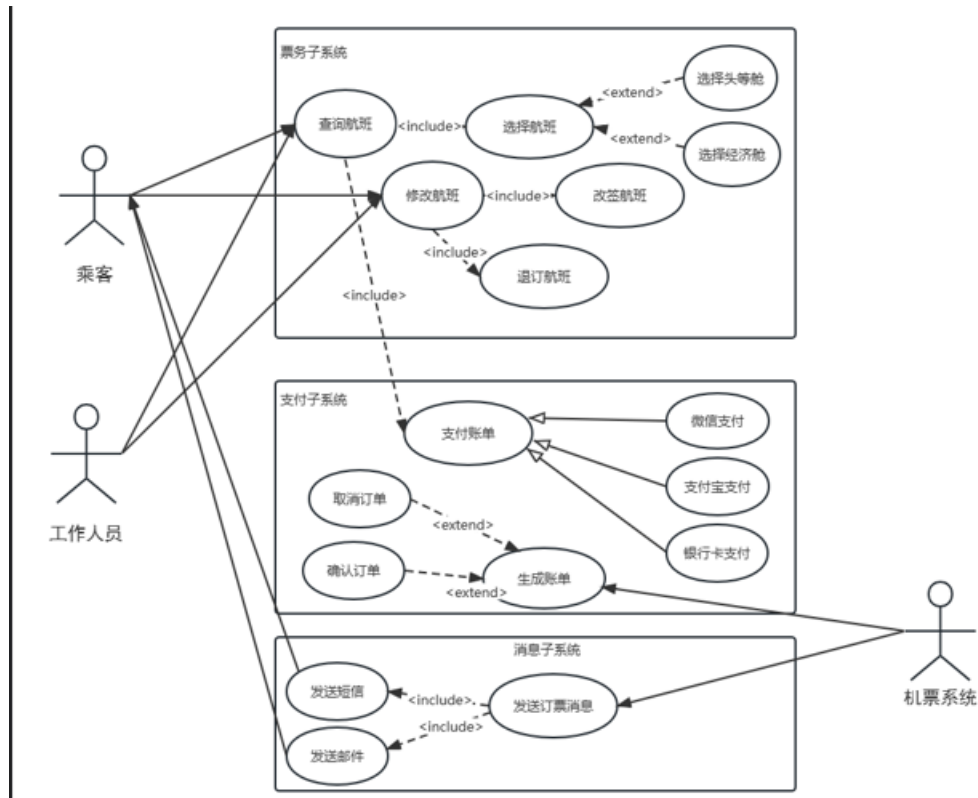


图 1: 用例图 1

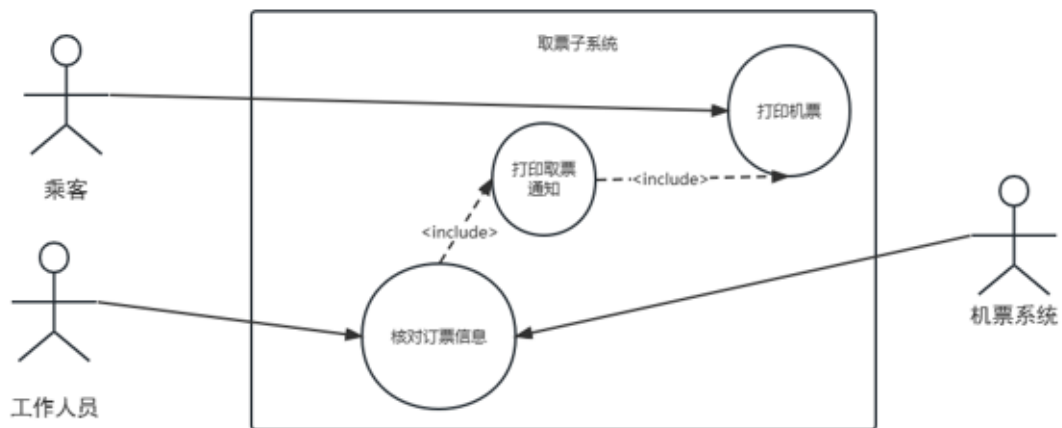


图 2: 用例图 2

用例图说明

用例关系	说明	特性	表示
关联	参与者与用例关系	-	→
泛化	参与者之间或用例之间关系	父子关系, 子继承父行为和结构	←
包含	用例之间关系	子用例无条件发生	---▶
包含	用例之间关系	子用例有条件发生	◀---

图 3: 用例图说明

具体说明如下所示:

票务子系统

该子系统主要负责机票的查询、预订、退订等功能。用户可以通过该子系统查询航班信息, 选择航班并进行预订, 也可以在需要的时候进行退订。该子系统还需要与航空公司的系统进行交互, 以获取最新的航班信息和座位情况。

支付子系统

该子系统主要负责用户的支付操作。当用户完成机票预订后, 需要通过该子系统进行支付。该子系统需要支持多种支付方式, 如信用卡、支付宝、微信等。在用户完成支付后, 该子系统还需要与票务子系统进行交互, 以更新用户的订单状态。

消息子系统

该子系统主要负责向用户发送各种消息, 如订单确认、支付成功、航班变更等。该子系统需要支持多种消息发送方式, 如短信、邮件、APP 推送等。在用户进行机票预订、支付、退订等操作时, 该子系统还需要向用户发送相应的消息, 以提供及时的反馈和服务。

取票子系统

该子系统主要负责机票的取票操作。当用户完成机票预订和支付后, 需要通过该子系统进行取票。用户可以选择在机场柜台或自助取票机上进行取票。该子系统需要与航空公司的系统进行交互, 以获取用户的订单信息和机票信息。在用户进行取票操作时, 该子系统需要验证用户的身份信息, 并核对订单和机票信息。如果信息匹配成功, 该子系统会向用户发放机票, 并更新订单状态。如果信息匹配失败, 该子系统会提示用户重新输入或联系客服解决问题。

二、 活动图

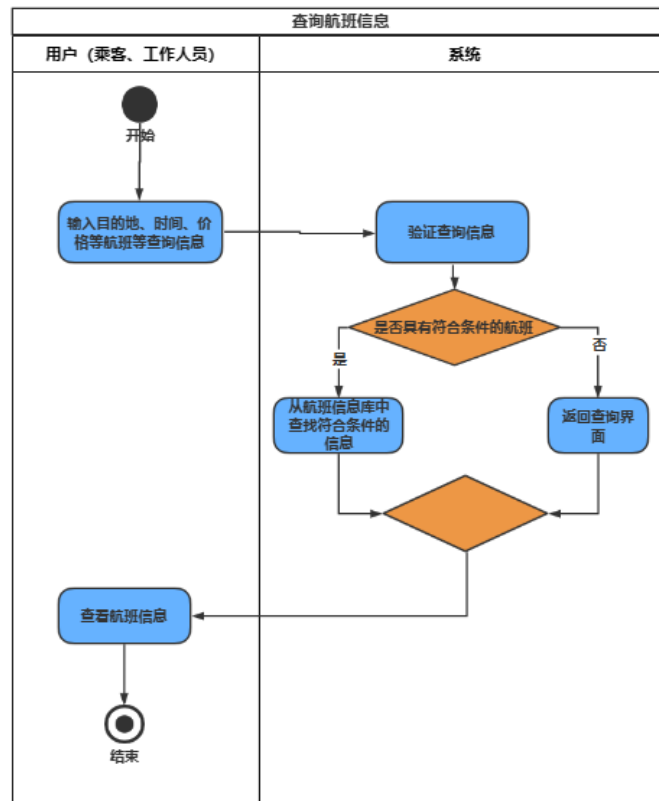


图 4: 查询航班信息活动图

用户需要进入机票预订系统，然后点击“查询航班信息”按钮。接着，系统会显示一个查询页面，用户可以在该页面上输入出发地、目的地、出发时间等信息。用户输入完毕后，点击“查询”按钮。

系统接收到用户的查询请求后，会根据用户输入的信息，从数据库中查询符合条件的航班信息。如果查询结果为空，系统会提示用户“未找到符合条件的航班信息”，并返回查询页面，让用户重新输入查询条件。如果查询结果不为空，系统会将查询结果显示在页面上，包括航班号、出发时间、到达时间、票价等信息。

用户可以根据查询结果选择需要预订的航班，点击“预订”按钮。系统会跳转到预订页面，用户需要输入乘客信息、联系方式等信息，并选择座位类型和支付方式。用户输入完毕后，点击“确认预订”按钮。

系统接收到用户的预订请求后，会根据用户输入的信息，生成订单，并将订单信息保存到数据库中。系统会提示用户“预订成功”，并显示订单号和支付金额等信息。

(进入机票预订系统，点击“查询航班信息”按钮) -> (显示查询页面);

(在查询页面上输入出发地、目的地、出发时间等信息，点击“查询”按钮) -> (接收到用户的查询请求，从数据库中查询符合条件的航班信息);

(如果查询结果为空，提示用户“未找到符合条件的航班信息”，返回查询页面) -down-> (重新输入查询条件);

(如果查询结果不为空，将查询结果显示在页面上) -> (根据查询结果选择需要预订的航班，点击“预订”按钮);

(点击“预订”按钮) -> (跳转到预订页面);
 (在预订页面上输入乘客信息、联系方式等信息, 选择座位类型和支付方式, 点击“确认预订”按钮) -> (接收到用户的预订请求, 生成订单, 并将订单信息保存到数据库中);
 (提示用户“预订成功”, 显示订单号和支付金额等信息) -> (完成预订).

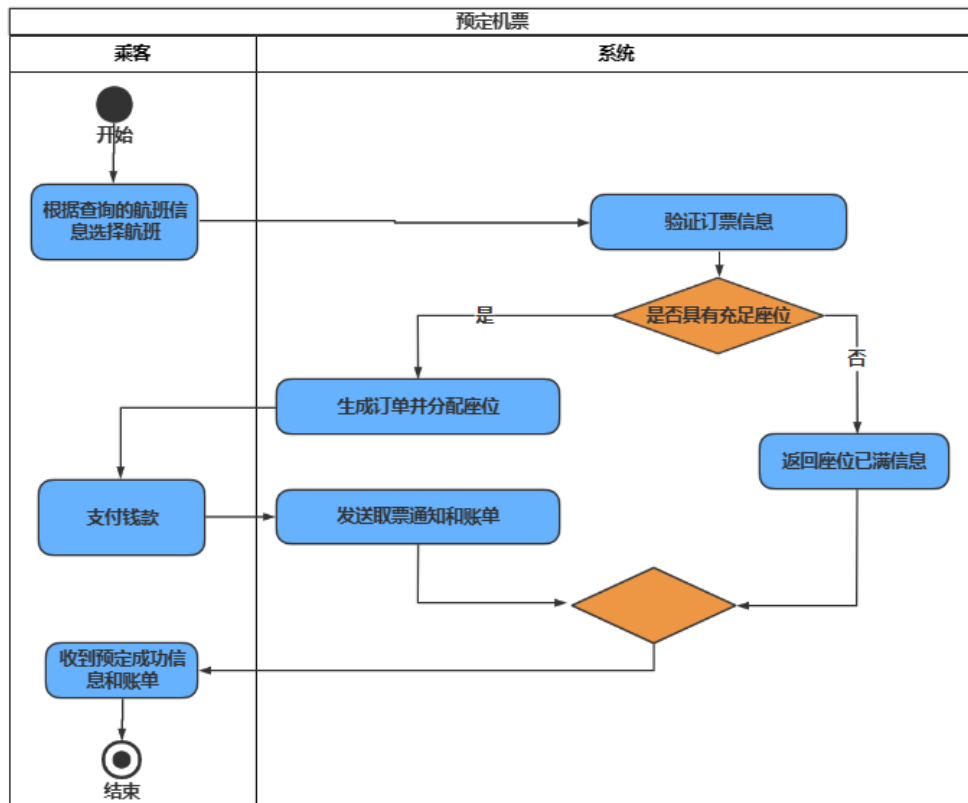


图 5: 预定机票活动图

用户需要进入机票预订系统, 然后点击“预定机票”按钮。接着, 系统会显示一个预定页面, 用户需要输入出发地、目的地、出发时间、乘客人数等信息, 并选择航班和座位类型。用户输入完毕后, 点击“确认预定”按钮。

系统接收到用户的预定请求后, 会根据用户输入的信息, 生成订单, 并将订单信息保存到数据库中。系统会提示用户“预定成功”, 并显示订单号和支付金额等信息。

用户需要选择支付方式, 并输入支付信息。用户输入完毕后, 点击“确认支付”按钮。

系统接收到用户的支付请求后, 会根据用户输入的支付信息, 完成支付, 并将支付结果保存到数据库中。系统会提示用户“支付成功”, 并显示支付金额和支付时间等信息。

(进入机票预订系统, 点击“预定机票”按钮) -> (显示预定页面);

(在预定页面上输入出发地、目的地、出发时间、乘客人数等信息, 选择航班和座位类型, 点击“确认预定”按钮) -> (接收到用户的预定请求, 生成订单, 并将订单信息保存到数据库中);

(提示用户“预定成功”, 显示订单号和支付金额等信息) -> (选择支付方式, 并输入支付信息, 点击“确认支付”按钮);

(点击“确认支付”按钮) -> (接收到用户的支付请求, 完成支付, 并将支付结果保存到数据库中); (提示用户“支付成功”, 显示支付金额和支付时间等信息) -> (完成支付)

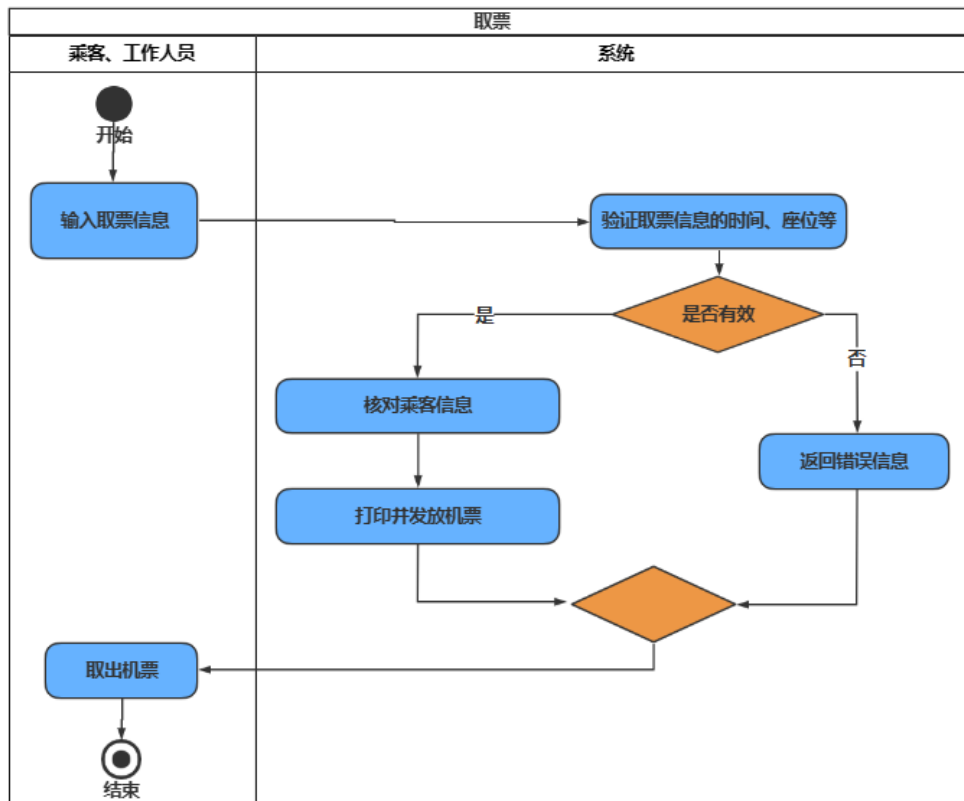


图 6: 取票活动图

用户需要进入机票预订系统，然后点击“我的订单”按钮。接着，系统会显示用户的订单列表，用户需要选择需要取票的订单，并点击“取票”按钮。

系统接收到用户的取票请求后，会根据订单信息，查询该订单的状态。如果订单状态为“已支付”，系统会生成取票码，并将取票码发送给用户。如果订单状态为“未支付”，系统会提示用户“该订单未支付，请先完成支付”。

用户需要凭借取票码到机场柜台或自助取票机上取票。用户到达机场后，需要在柜台或自助取票机上输入取票码，并按照提示操作，完成取票流程。

(进入机票预订系统，点击“我的订单”按钮) -> (显示用户的订单列表);

(选择需要取票的订单，并点击“取票”按钮) -> (接收到用户的取票请求，查询该订单的状态);

(如果订单状态为“已支付”，生成取票码，并将取票码发送给用户) -> (收到取票码);

(如果订单状态为“未支付”，提示用户“该订单未支付，请先完成支付”) -down-> (完成支付);

(凭借取票码到机场柜台或自助取票机上取票) -> (用户在柜台或自助取票机上输入取票码，并按照提示操作，完成取票流程)

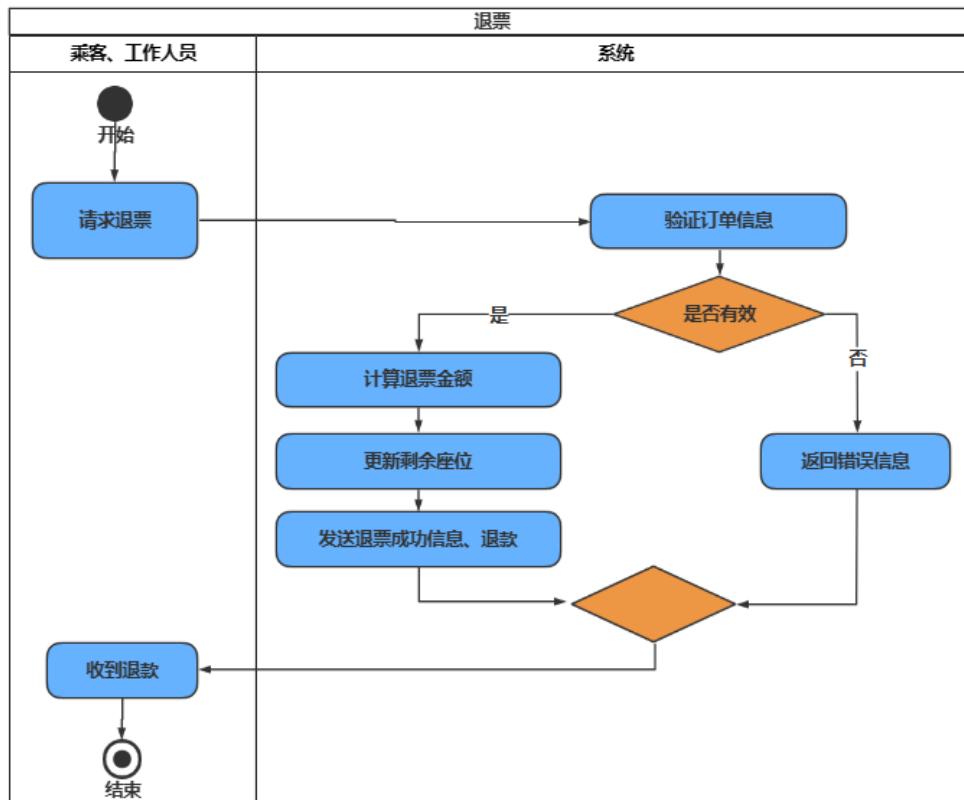


图 7: 退票活动图

整体流程如下:

- 起始节点
- 输入节点: 用户输入退票请求
- 决策节点: 系统检查用户是否有资格退票
- 如果用户有资格退票, 输出节点: 系统显示退票费用, 并等待用户确认
- 如果用户确认退票, 子活动图: 处理退票请求
 - a. 系统检查用户是否已经支付了退票费用
 - b. 如果用户已经支付了退票费用, 子活动图: 处理退款请求
 - * i. 系统检查用户的退款方式
 - * ii. 系统向用户确认退款方式, 并等待用户确认
 - * iii. 系统开始处理退款请求, 并向用户显示退款成功的消息
 - c. 如果用户还没有支付退票费用, 子活动图: 要求用户支付退票费用
 - * i. 系统向用户显示退票费用, 并等待用户确认
 - * ii. 系统检查用户的支付方式
 - * iii. 系统向用户确认支付方式, 并等待用户确认
 - * iv. 系统开始处理支付请求, 并向用户显示支付成功的消息

- 如果用户没有资格退票，输出节点：系统向用户显示无法退票的消息
- 输出节点：系统向用户显示退票成功的消息
- 结束节点

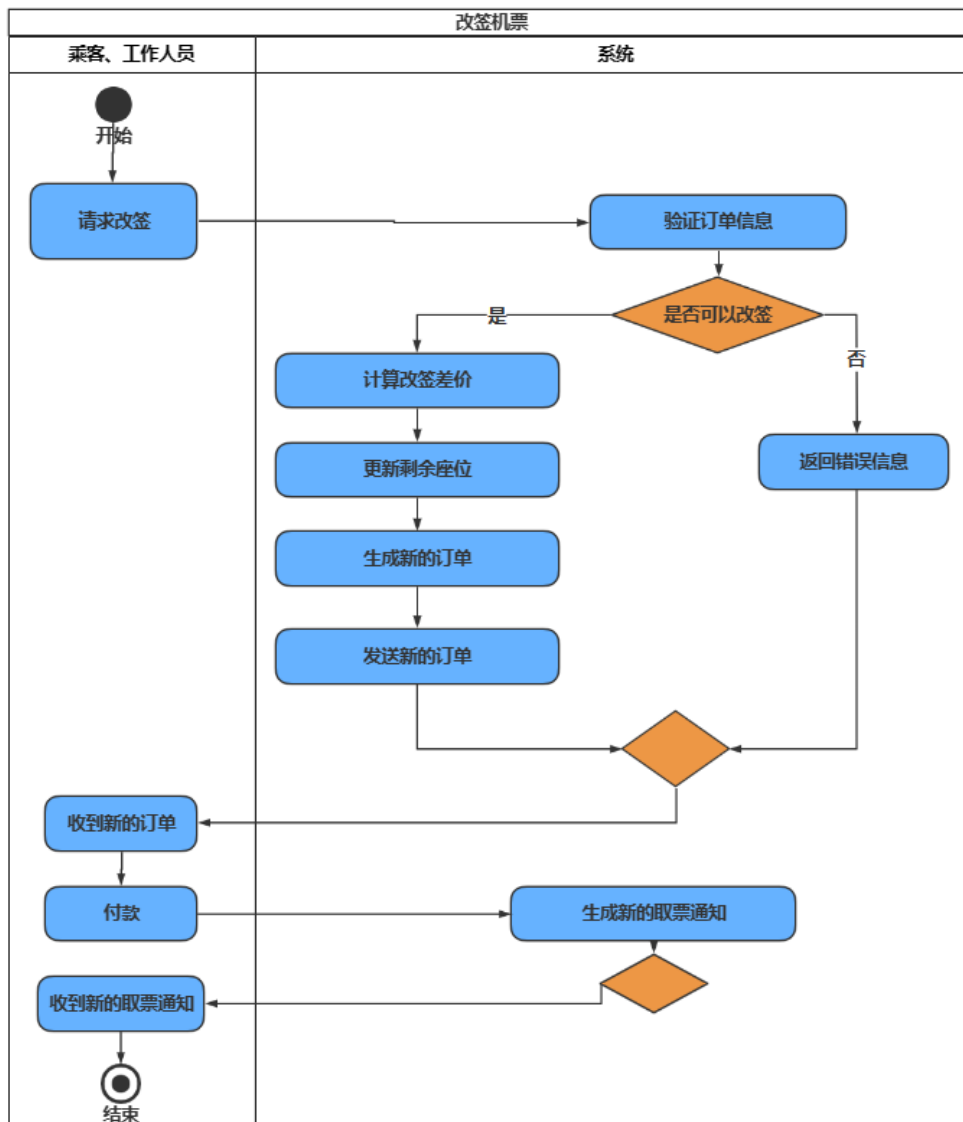


图 8: 改签活动图

活动图中有三个参与者：旅客、机票预订系统和航空公司。旅客可以通过机票预订系统预订机票，并可以在预订后进行改签操作。改签操作需要与航空公司进行交互，以确认改签是否成功。

活动图描述

活动图中包含以下步骤：

- (1) 旅客登录机票预订系统，选择需要改签的机票订单。
- (2) 机票预订系统显示该订单的详细信息，包括航班号、出发时间、座位等级等。

- (3) 旅客选择需要改签的航班，并选择新的出发时间和座位等级。
- (4) 机票预订系统向航空公司发送改签请求，包括原订单信息和新的出发时间和座位等级。
- (5) 航空公司接收改签请求，并进行验证。如果验证失败，返回错误信息；如果验证成功，进入下一步。
- (6) 航空公司修改订单信息，包括出发时间和座位等级，并返回修改成功信息。
- (7) 机票预订系统接收到修改成功信息后，更新订单信息，并向旅客显示修改成功信息。

三、 类图

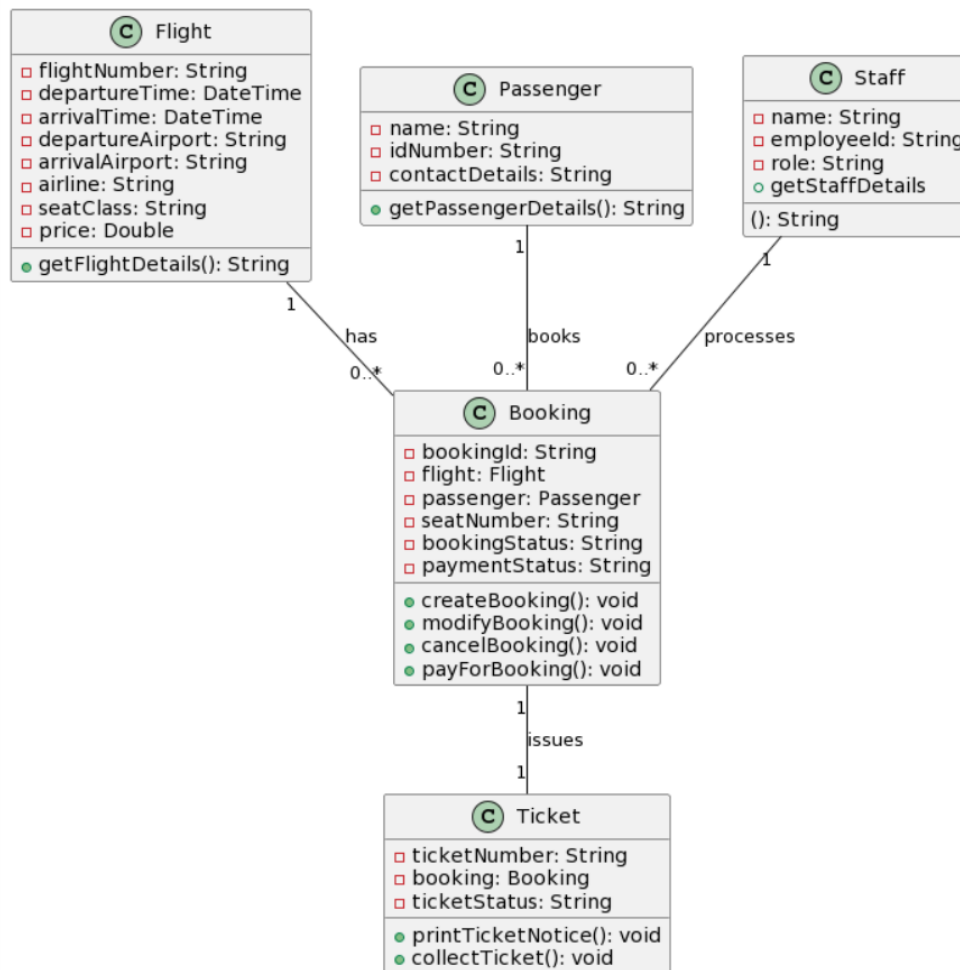


图 9: 类图

以下是类图的详细说明：

旅客类

旅客类表示系统中的旅客。旅客类有如下属性：

- (1) id: 旅客的唯一标识符。
- (2) name: 旅客的姓名。
- (3) gender: 旅客的性别。
- (4) age: 旅客的年龄。
- (5) phone: 旅客的联系电话。

旅客类有如下操作：

- (1) bookFlight(flightId: String, seatType: String, seatNum: int): void: 预订航班。flightId 为航班的唯一标识符, seatType 为座位类型(如经济舱、商务舱等), seatNum 为座位数量。
- (2) payOrder(orderId: String): void: 支付订单。orderId 为订单的唯一标识符。

工作人员类

工作人员类表示系统中的工作人员。工作人员类有如下属性：

- (1) id: 工作人员的唯一标识符。
- (2) name: 工作人员的姓名。
- (3) role: 工作人员的职称。
- (4) phone: 工作人员的联系电话。

工作人员类有如下操作：

- (1) login(username: String, password: String): boolean: 工作人员登录系统。如果用户名和密码正确, 则返回 true, 否则返回 false。
- (2) addFlight(flight: Flight): void: 添加航班。flight 为要添加的航班对象。
- (3) updateFlight(flight: Flight): void: 更新航班。flight 为要更新的航班对象。
- (4) deleteFlight(flightId: String): void: 删除航班。flightId 为要删除的航班的唯一标识符。

机票类

机票类包含了机票的基本信息, 如机票号、预定信息和机票价格等属性。其中, 机票号和预定信息是必须的属性, 而机票价格则是可选的属性。机票类还包含了获取和设置这些属性的方法。在机票类中, 预定信息是通过引用预定类来实现的, 即预定类。机票类有如下属性：

- (1) ticketNumber: String // 机票号

(2) reservation: Reservation // 预定信息

(3) price: double // 机票价格

机票类有如下操作:

(1) getTicketNumber(): String // 获取机票号

(2) getReservation(): Reservation // 获取预定信息

(3) getPrice(): double // 获取机票价格

航班类

航班类包含了航班的基本信息,如航班号、出发机场、到达机场、出发时间、到达时间、可用座位数和机票价格等属性。其中,航班号、出发机场、到达机场、出发时间和到达时间是必须的属性,而可用座位数和机票价格则是可选的属性。航班类还包含了获取和设置这些属性的方法。

(1) flightNumber: 航班号的,即航班的唯一标识符。

(2) departureTime: 航班的出发时间。

(3) arrivalTime: 航班的到达时间。

(4) departureAirport: 航班的出发地点。

(5) arrivalAirport: 航班的到达地点。

(6) airline: 航班的航线。

(7) seatClass: 座位的数量和载员情况。

(8) price: 机票价格。

航班类有如下操作:

(1) getFlightNumber(): 获取航班号

(2) setFlightNumber(flightNumber: String): 设置航班号

(3) getDepartureAirport(): 获取出发机场

(4) getAvailableSeats(): 获取可用座位数

(5) getPrice(): double // 获取机票价格

预定类

预定类包含了预定的基本信息,如预定号、预定的航班、乘客信息和座位号等属性。其中,预定号、预定的航班和乘客信息是必须的属性,而座位号则是可选的属性。预定类还包含了获取和设置这些属性的方法。在预定类中,航班和乘客信息都是通过引用其他类来实现的,即航班类和乘客类。这样可以避免在预定类中重复存储这些信息,提高了系统的效

率。预定类有如下属性：

- (1) reservationNumber: String // 预定号
- (2) flight: Flight // 预定的航班
- (3) passenger: Passenger // 乘客信息
- (4) seatNumber: String // 座位号

预定类有如下操作：

- (1) getReservationNumber(): String // 获取预定号
- (2) setReservationNumber(reservationNumber: String): void // 设置预定号
- (3) getFlight(): Flight // 获取预定的航班
- (4) setFlight(flight: Flight): void // 设置预定的航班
- (5) getPassenger(): Passenger // 获取乘客信息
- (6) setPassenger(passenger: Passenger): void // 设置乘客信息
- (7) getSeatNumber(): String // 获取座位号
- (8) setSeatNumber(seatNumber: String): void // 设置座位号

四、 顺序图

主要分析了以下四种顺序图：

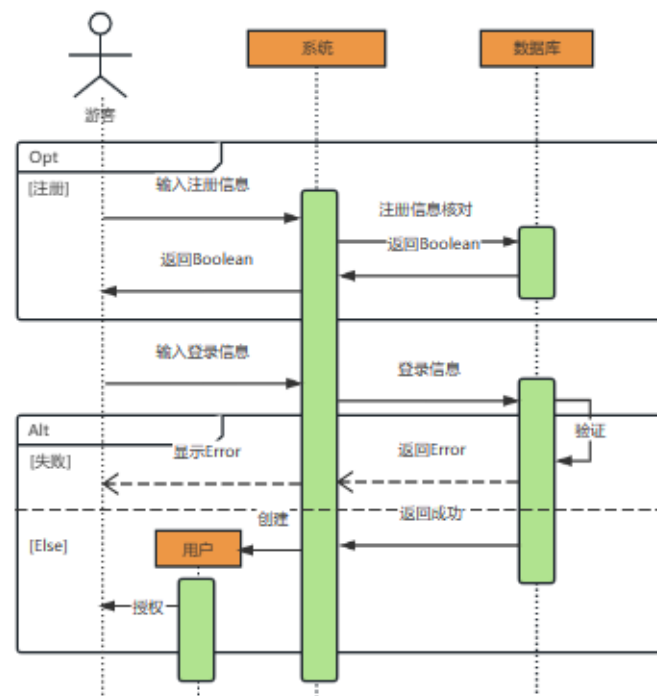


图 10: 登录顺序图

登录顺序图

新用户注册顺序图描述了用户在注册新账户时与系统之间的交互过程。用户首先向系统发送注册请求，系统显示注册页面，要求用户填写注册信息。用户填写完信息后，提交注册信息给系统。系统验证用户提交的信息是否合法，如果合法，将用户信息保存到数据库中，并显示注册成功信息；如果不合法，将显示错误信息，要求用户重新填写注册信息。用户确认注册成功信息后，顺序结束。

1. 用户向系统发送注册请求。
2. 系统显示注册页面，要求用户填写注册信息。
3. 用户填写注册信息，包括用户名、密码、电子邮件地址等。
4. 用户提交注册信息。
5. 系统验证用户提交的信息是否合法。
6. 如果信息合法，系统将用户信息保存到数据库中，并显示注册成功信息。
7. 如果信息不合法，系统将显示错误信息，要求用户重新填写注册信息。
8. 用户确认注册成功信息后，顺序结束。

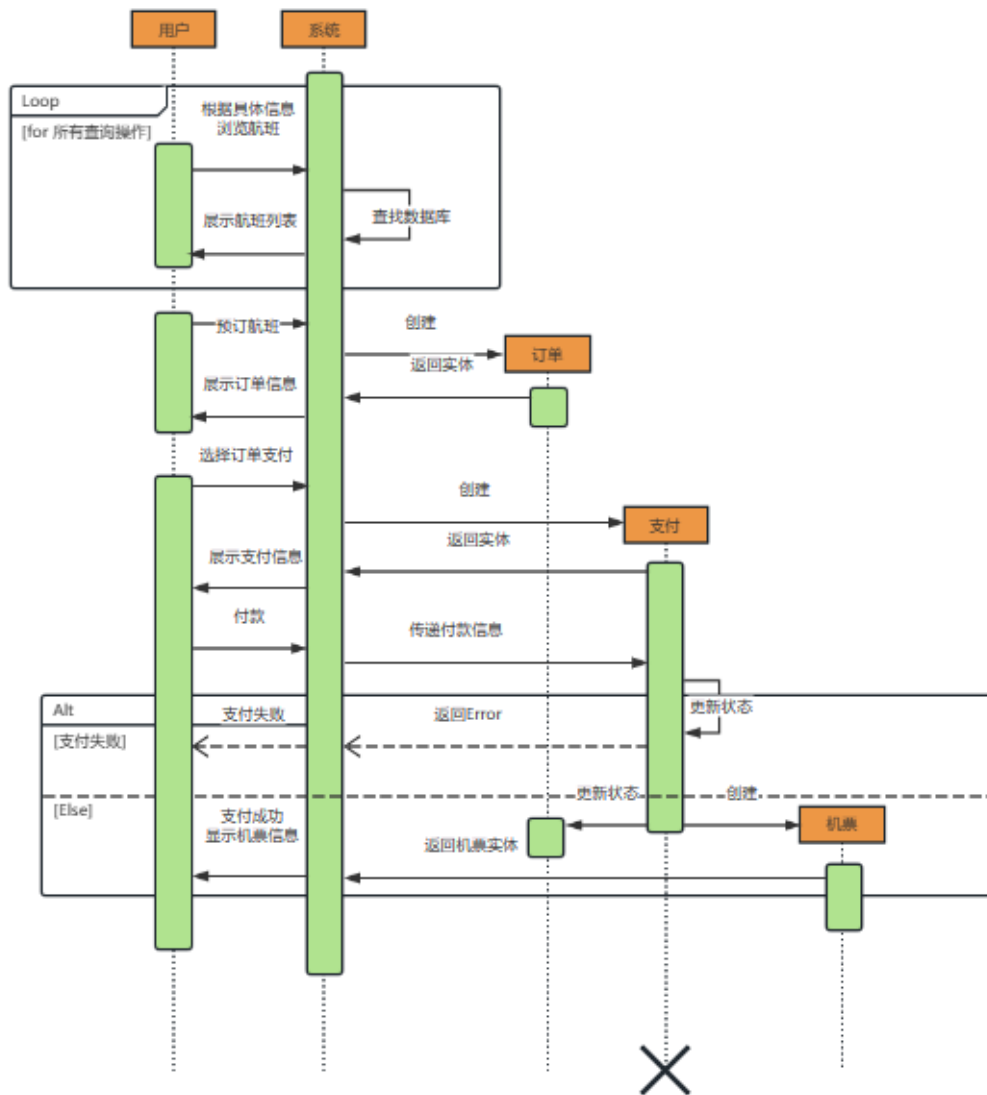


图 11: 订票顺序图

订票顺序图

订票顺序图描述了用户在订票时与系统和航空公司之间的交互过程。用户首先向系统发送查询航班请求，系统显示航班查询结果。用户选择要预定的航班后，向系统发送预定请求。系统显示预定页面，要求用户填写预定信息。用户填写完信息后，提交预定信息给系统。系统向航空公司发送预定请求，航空公司验证预定信息是否合法。如果信息合法，航空公司将预定信息保存到数据库中，并向系统发送预定成功信息；如果信息不合法，航空公司将向系统发送预定失败信息。系统根据航空公司的响应，显示预定成功或失败信息给用户。如果预定成功，系统生成机票信息，并向用户显示机票信息；如果预定失败，系统将显示错误信息，要求用户重新填写预定信息。用户确认机票信息后，顺序结束。

1. 用户向系统发送查询航班请求。
2. 系统显示航班查询结果。
3. 用户选择要预定的航班。

4. 用户向系统发送预定请求。
5. 系统显示预定页面，要求用户填写预定信息。
6. 用户填写预定信息，包括乘客信息、座位号等。
7. 用户提交预定信息。
8. 系统向航空公司发送预定请求。
9. 航空公司验证预定信息是否合法。
10. 如果信息合法，航空公司将预定信息保存到数据库中，并向系统发送预定成功信息。
11. 如果信息不合法，航空公司将向系统发送预定失败信息。
12. 系统根据航空公司的响应，显示预定成功或失败信息给用户。
13. 如果预定成功，系统生成机票信息，并向用户显示机票信息。
14. 如果预定失败，系统将显示错误信息，要求用户重新填写预定信息。
15. 用户确认机票信息后，顺序结束。

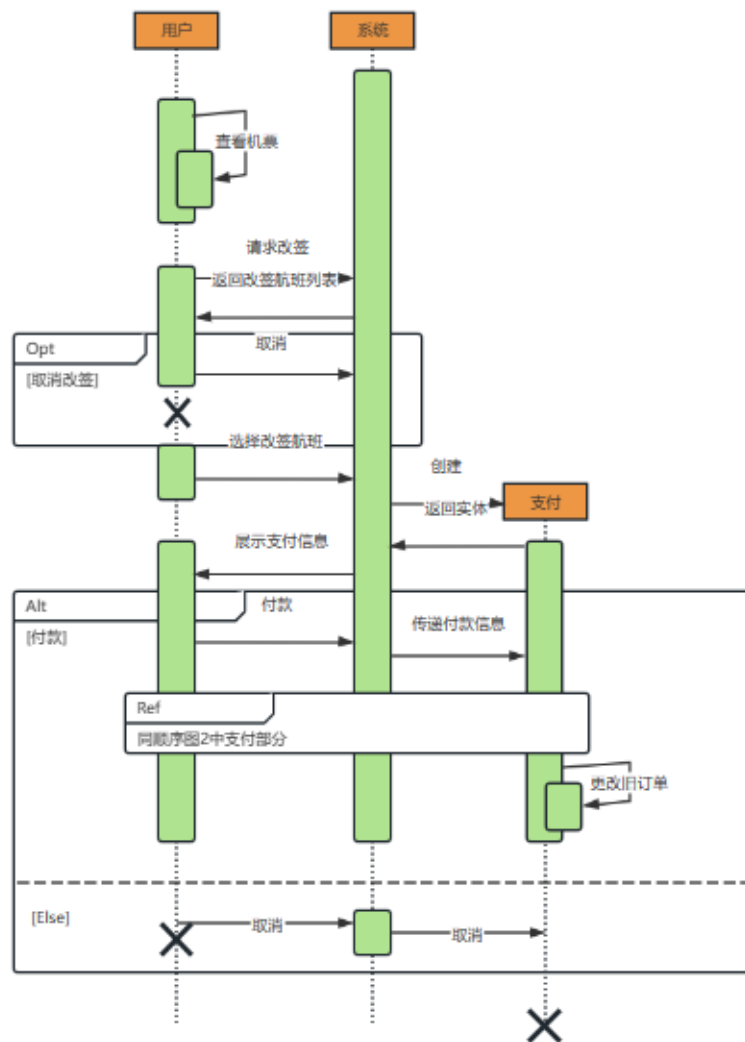


图 12: 退票、改签顺序图

退票、改签顺序图

退票、改签顺序图描述了用户在退票或改签时与系统和航空公司之间的交互过程。用户首先向系统发送退票或改签请求，系统显示退票或改签页面，要求用户填写相关信息。用户填写完信息后，提交退票或改签信息给系统。系统向航空公司发送退票或改签请求，航空公司验证退票或改签信息是否合法。如果信息合法，航空公司将退票或改签信息保存到数据库中，并向系统发送退票或改签成功信息；如果信息不合法，航空公司将向系统发送退票或改签失败信息。系统根据航空公司的响应，显示退票或改签成功或失败信息给用户。如果退票或改签成功，系统将向用户发送退票或改签成功信息，并根据用户选择的方式进行退款或改签；如果退票或改签失败，系统将显示错误信息，要求用户重新填写退票或改签信息。用户确认退票或改签信息后，顺序结束。

1. 用户向系统发送退票或改签请求。
2. 系统显示退票或改签页面，要求用户填写相关信息。
3. 用户填写相关信息，包括机票号、退票或改签原因等。

4. 用户提交退票或改签信息。
5. 系统向航空公司发送退票或改签请求。
6. 航空公司验证退票或改签信息是否合法。
7. 如果信息合法，航空公司将退票或改签信息保存到数据库中，并向系统发送退票或改签成功信息。
8. 如果信息不合法，航空公司将向系统发送退票或改签失败信息。
9. 系统根据航空公司的响应，显示退票或改签成功或失败信息给用户。
10. 如果退票或改签成功，系统将向用户发送退票或改签成功信息，并根据用户选择的方式进行退款或改签。
11. 如果退票或改签失败，系统将显示错误信息，要求用户重新填写退票或改签信息。
12. 用户确认退票或改签信息后，顺序结束。

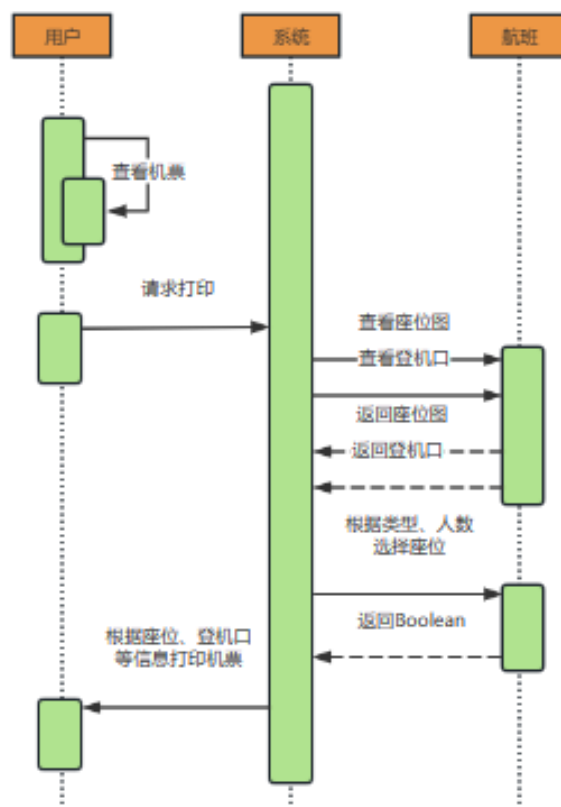


图 13: 取票活动图

取票顺序图

取票顺序图描述了用户在取票时与系统和航空公司之间的交互过程。用户首先向系统发送取票请求，系统显示取票页面，要求用户填写相关信息。用户填写完信息后，提交取票信息给系统。系统向航空公司发送取票请求，航空公司验证取票信息是否合法。如果信息合法，航空公司将机票信息发送给系统。系统根据航空公司的响应，显示机票信息给用户。用户确认机票信息后，顺序结束。

1. 用户向系统发送取票请求。
2. 系统显示取票页面，要求用户填写相关信息。
3. 用户填写相关信息，包括机票号、身份证号等。
4. 用户提交取票信息。
5. 系统向航空公司发送取票请求。
6. 航空公司验证取票信息是否合法。
7. 如果信息合法，航空公司将机票信息发送给系统。
8. 系统根据航空公司的响应，显示机票信息给用户。
9. 用户确认机票信息后，顺序结束。

五、 协作图

协作图基本与顺序图是类似的，因此进行简单的解释：

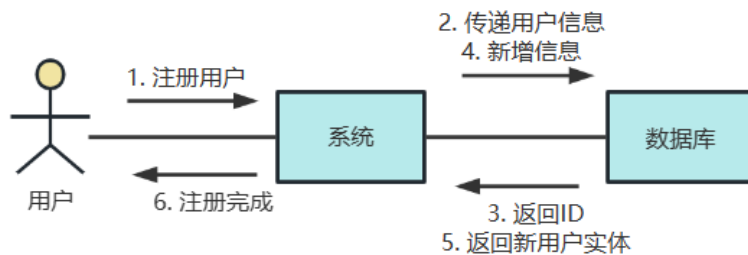


图 14: 登录协作图

参与者:

1. 用户
2. 系统
3. 数据库

协作:

用户：输入用户名和密码。

系统：接收用户输入的用户名和密码，向数据库发送登录请求，接收数据库的响应，显示登录成功或失败信息给用户。

数据库：验证用户名和密码是否匹配，将验证结果发送给系统。

登录协作图描述了用户、系统和数据库之间的协作过程。用户输入用户名和密码后，将信息提交给系统。系统接收到用户提交的信息后，向数据库发送登录请求。数据库验证用户名和密码是否匹配，将验证结果发送给系统。系统接收到数据库的响应后，显示登录成功或失败信息给用户。在这个过程中，用户、系统和数据库之间通过消息进行通信，完成了登录的协作过程。

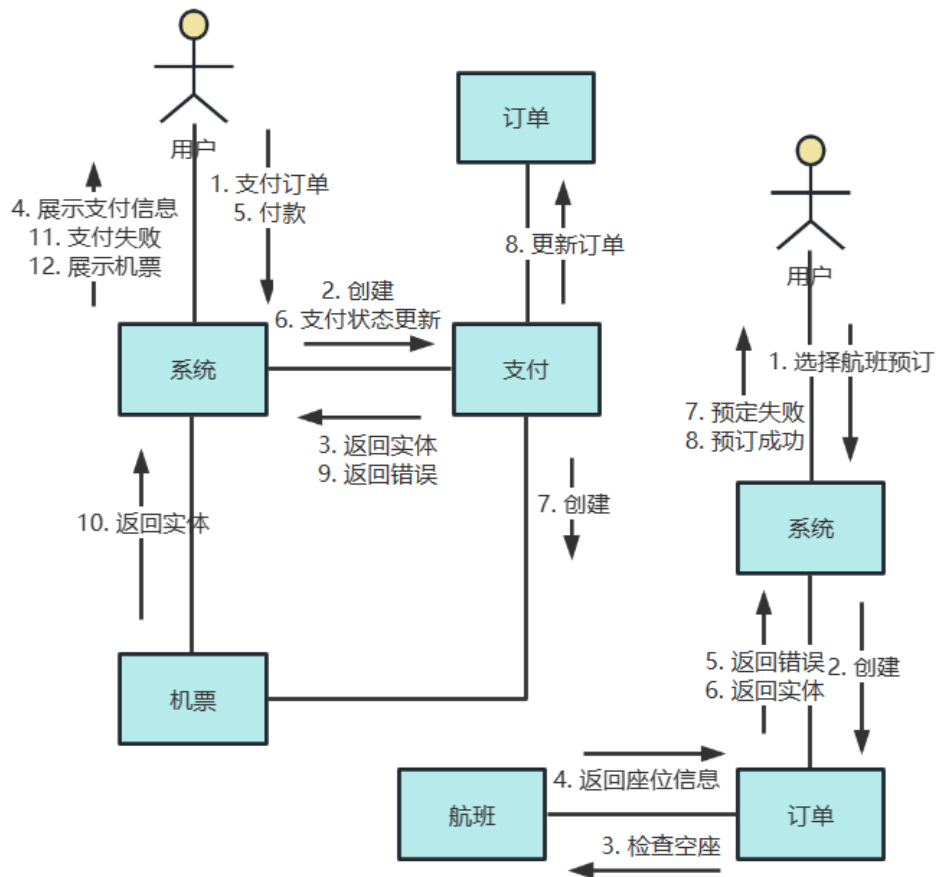


图 15: 订票活动图

参与者:

1. 用户
2. 系统
3. 航空公司
4. 数据库

协作:

用户：发送查询航班请求。

系统：接收用户发送的查询航班请求，向数据库发送查询航班请求，接收数据库的响应，将查询结果显示给用户。

用户：选择要预定的航班，发送预定请求。

系统：接收用户发送的预定请求，向航空公司发送预定请求，接收航空公司的响应，将预定页面显示给用户。

用户：填写预定信息，提交预定信息。

系统：接收用户提交的预定信息，向航空公司发送预定请求，接收航空公司的响应，将预定成功或失败信息显示给用户。验证预定信息是否合法，将预定信息保存到数据库中，将预定成功或失败信息发送给系统。

数据库：保存预定信息。

订票协作图描述了用户、系统和数据库之间的协作过程。用户发送查询航班请求后，系统接收到请求后，向数据库发送查询航班请求，接收数据库的响应，将查询结果显示给用户。用户选择要预定的航班后，发送预定请求给系统。系统接收到用户发送的预定请求后，将预定页面显示给用户。用户填写预定信息后，提交预定信息给系统。系统接收到用户提交的预定信息后，将预定成功或失败信息显示给用户。数据库保存预定信息。在这个过程中，用户、系统和数据库之间通过消息进行通信，完成了订票的协作过程。

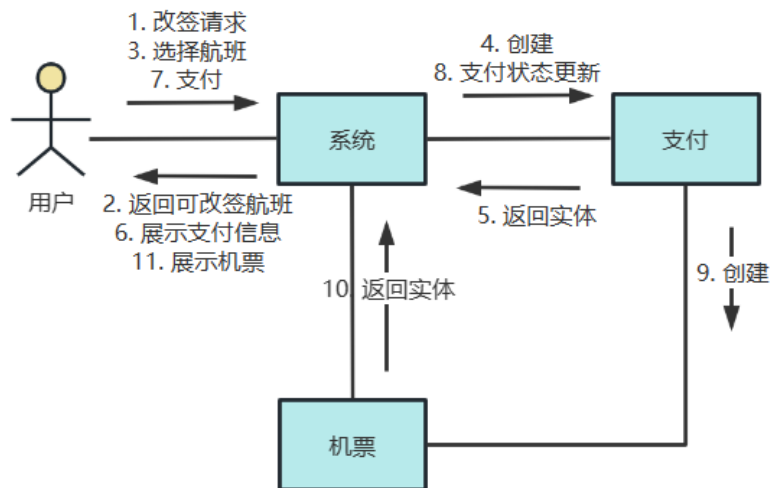


图 16: 改签、退票协作图

参与者:

1. 用户
2. 系统

协作:

用户：发送退票或改签请求。

系统：接收用户发送的退票或改签请求，向航空公司发送退票或改签请求，接收航空公司的响应，将退票或改签页面显示给用户。

用户：填写退票或改签信息，提交退票或改签信息。

系统：接收用户提交的退票或改签信息，向航空公司发送退票或改签请求，接收航空公司的响应，将退票或改签成功或失败信息显示给用户。

退票、改签协作图描述了用户、系统之间的协作过程。用户发送退票或改签请求后，系统接收到请求后，将退票或改签页面显示给用户。用户填写退票或改签信息后，提交退票或改签信息

给系统。系统接收到用户提交的退票或改签信息后，将退票或改签成功或失败信息显示给用户。在这个过程中，用户、系统之间通过消息进行通信，完成了退票或改签的协作过程。

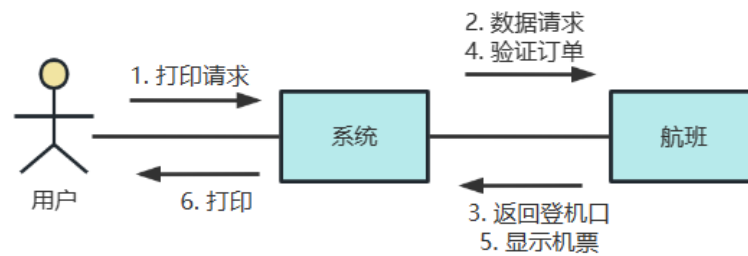


图 17: 取票协作图

参与者:

1. 用户
2. 系统

协作:

用户: 填写取票信息。

系统: 显示取票页面, 接收用户提交的取票信息, 显示机票信息给用户。验证取票信息是否合法, 将机票信息发送给系统。

取票协作图描述了用户、系统之间的协作过程。用户填写取票信息后, 将信息提交给系统。系统接收到用户提交的信息后, 验证取票信息是否合法, 如果合法, 显示机票信息给用户。在这个过程中, 用户、系统之间通过消息进行通信, 完成了取票的协作过程。

六、 状态图

状态图如下所示:

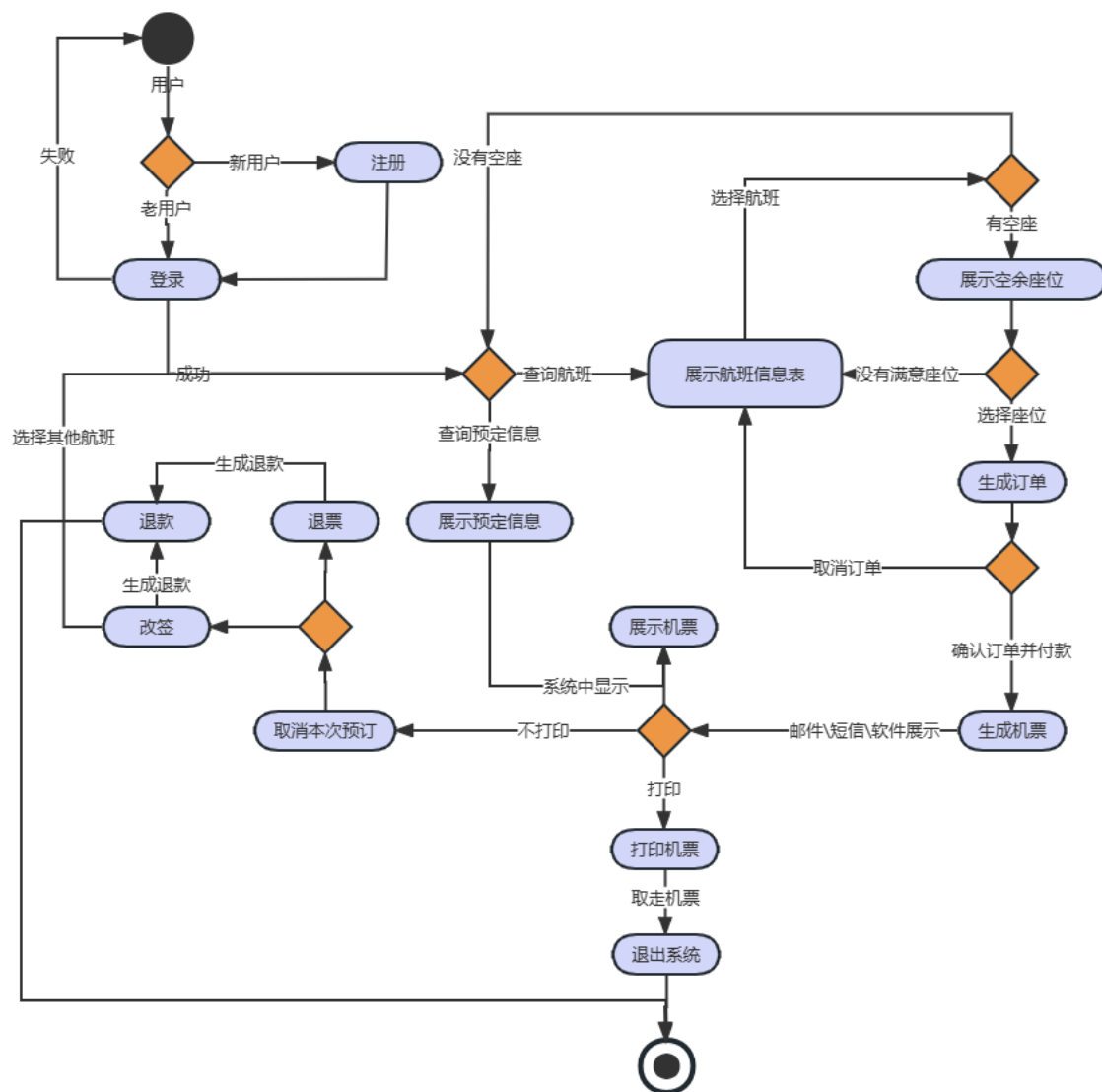


图 18: 状态图

具体分析，状态图的状态和转换关系大致应该如下：

- 初始状态：系统启动时的初始状态，等待用户进行操作。
- 登录：已注册的用户进行登录。
- 注册：新用户进行注册账号。
- 查询航班：用户进入该状态后，可以进行航班信息的查询操作。用户可以输入出发地、目的地、出发时间等信息，系统会返回符合条件的航班信息。
- 预订机票：用户进入该状态后，可以选择航班并进行预订操作。用户需要输入乘客信息、联系方式等信息，并选择支付方式进行支付。
- 支付订单：用户进入该状态后，需要完成支付操作。用户可以选择信用卡、支付宝、微信等支付方式进行支付。

- 取票：用户进入该状态后，需要进行取票操作。用户可以选择在机场柜台或自助取票机上进行取票。系统会验证用户的身份信息，并核对订单和机票信息。如果信息匹配成功，系统会向用户发放机票，并更新订单状态。
- 改签：用户进入该状态后，可以选择需要改签的航班，并进行改签操作。系统会根据航班情况和用户需求，重新生成订单并更新订单状态。
- 退票：用户进入该状态后，可以选择需要退订的航班，并进行退票操作。系统会根据退票规则和用户需求，计算退款金额并进行退款操作。
- 完成状态：用户完成取票、改签或退票操作后，进入该状态，表示整个预订流程已经完成。

七、 构件图

部署图如下所示：

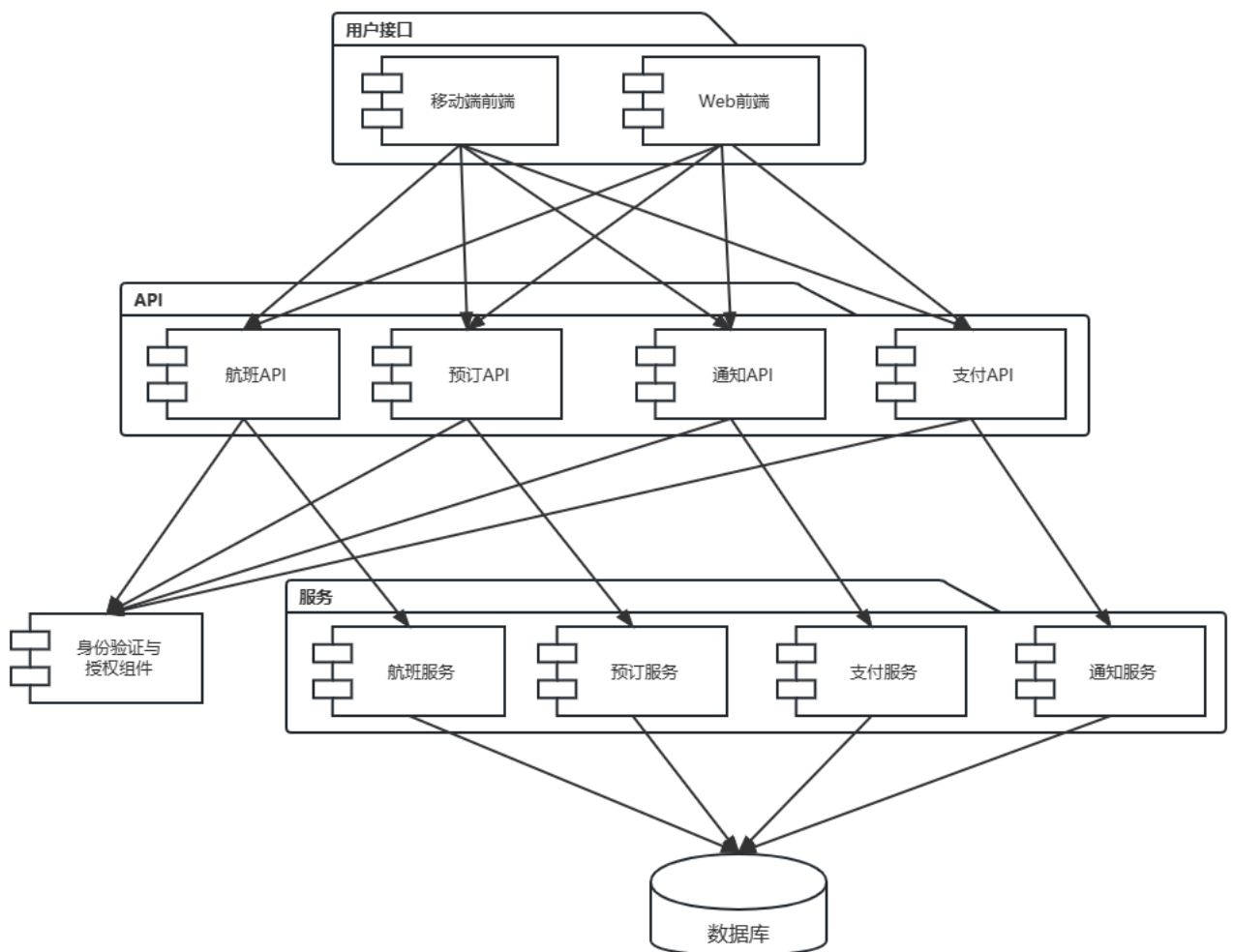


图 19: 构件图

经过分析，构件图大概有如下内容：

- 用户界面：包含 Web 前端和移动前端，为旅客和工作人员提供交互界面。

- 身份验证与授权组件：负责处理用户登录、注册、授权和验证，保证只有经过授权的用户才能访问相应功能。
- 预订 API：为前端提供创建、修改和取消预订的功能。
- 航班 API：为前端提供查询航班、获取航班详细信息和查询座位可用性的功能。
- 支付 API：为前端提供处理支付的功能，包括支持不同的支付方式。
- 通知 API：为前端提供发送通知（如：短信、电子邮件等）的功能，通知旅客有关预订和支付状态的更新。
- 预订服务：处理预订相关的业务逻辑。
- 航班服务：处理航班相关的业务逻辑。
- 支付服务：处理支付相关的业务逻辑。
- 通知服务：负责发送通知给旅客。
- 数据库：存储系统中的所有数据，包括预订、航班、旅客、支付等信息。

八、部署图

部署图如下所示：

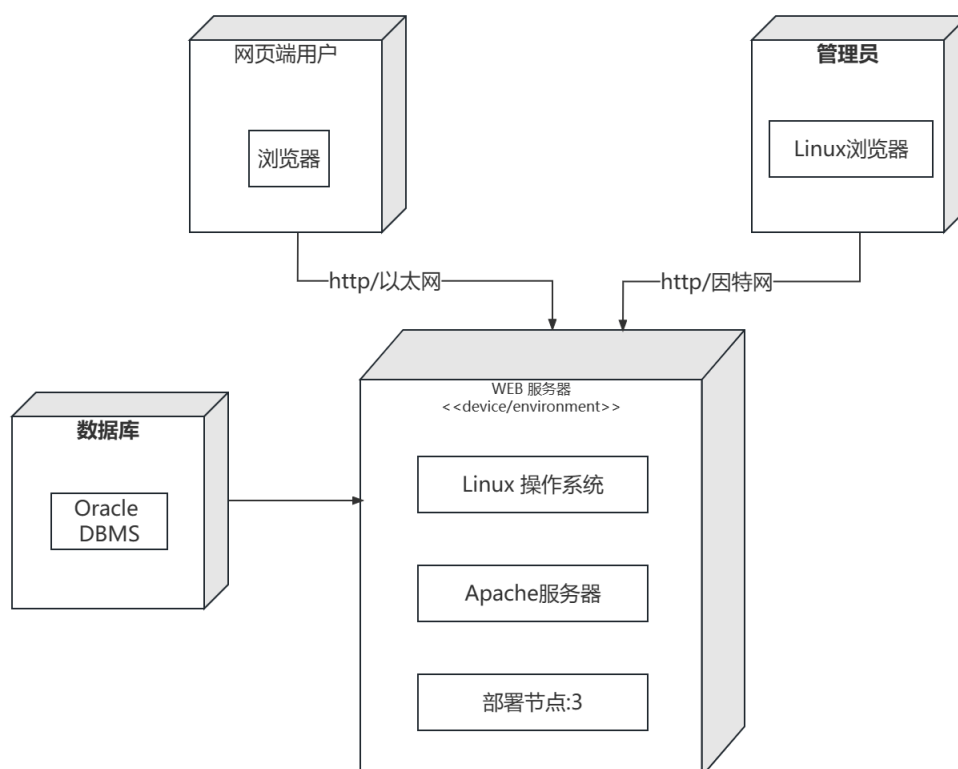


图 20: 部署图

具体内容可分为如下几点介绍：

- 系统架构：系统架构包括客户端、服务器和数据库三个部分。客户端主要负责用户界面的展示和用户输入的处理，服务器主要负责业务逻辑的处理和数据的存储，数据库主要负责数据的存储和管理。
- 硬件设备：硬件设备包括客户端设备、服务器设备和数据库设备。客户端设备可以是 PC、手机、平板等，服务器设备可以是 Web 服务器、应用服务器等，数据库设备可以是关系型数据库、NoSQL 数据库等。
- 软件组件：软件组件包括客户端应用程序、服务器应用程序和数据库管理系统。客户端应用程序可以是 Web 应用程序、移动应用程序等，服务器应用程序可以是 Java 应用程序、PHP 应用程序等，数据库管理系统可以是 MySQL、Oracle 等。
- 网络拓扑：网络拓扑包括客户端与服务器之间的通信和服务器与数据库之间的通信。客户端与服务器之间的通信可以通过 HTTP、HTTPS 等协议实现，服务器与数据库之间的通信可以通过 JDBC 等协议实现。
- 安全性：安全性包括用户身份认证、数据加密和访问控制等。用户身份认证可以通过用户名和密码、指纹识别等方式实现，数据加密可以通过 SSL、TLS 等协议实现，访问控制可以通过角色授权、IP 地址过滤等方式实现。