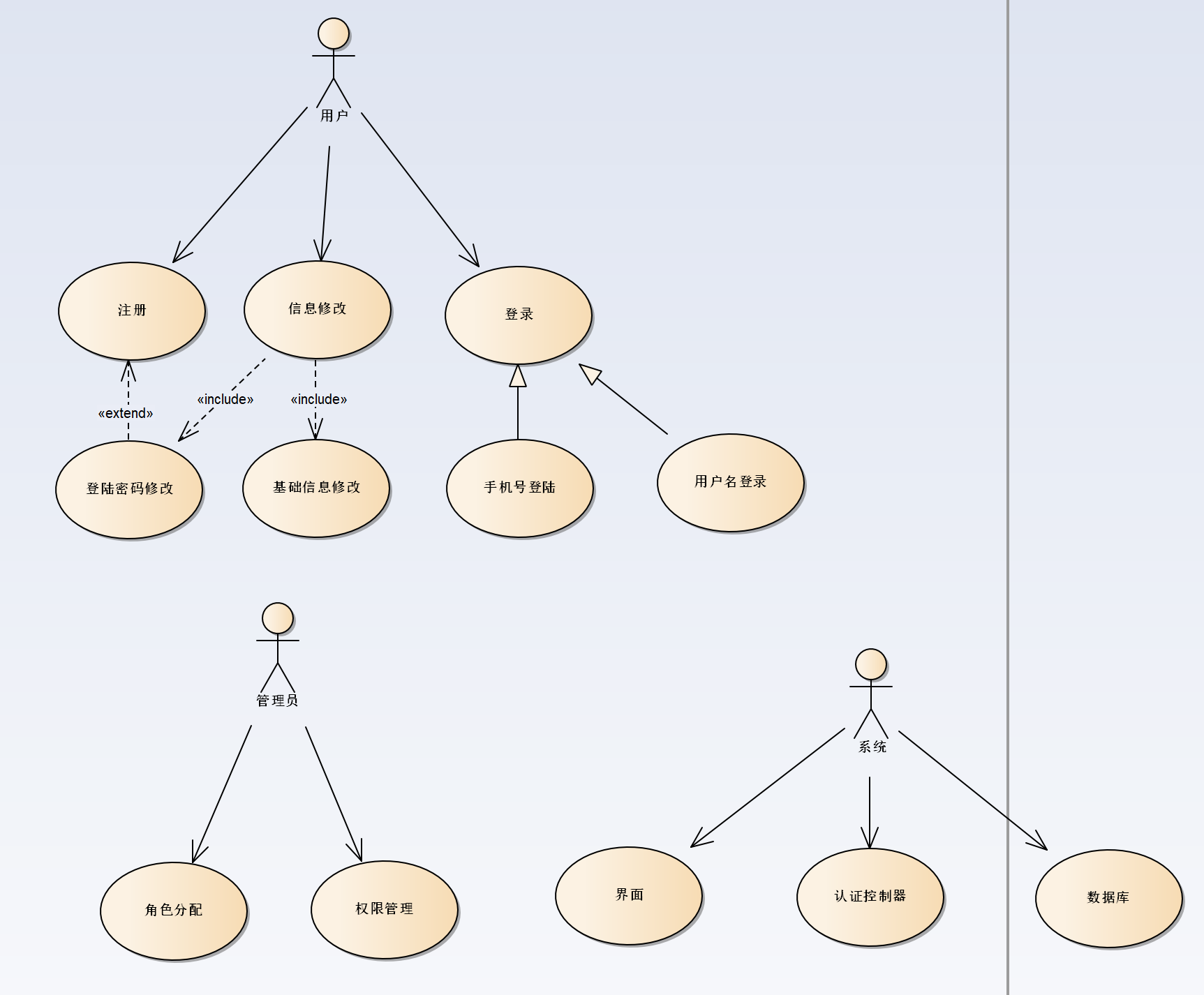
用户管理

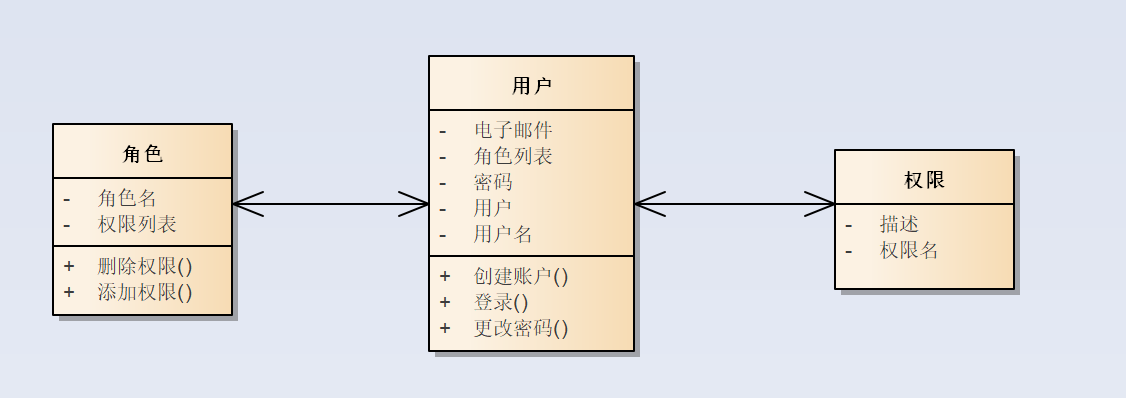
1. 用例图

参与者（Actors）:  
用户（User）：直接与系统交互的人，使用系统的基本功能。  
管理员（Admin）：负责管理和维护系统的人，使用高级功能。  
用例（Use Cases）:  
注册（Register）：用户创建新账户的过程。  
登录（Login）：用户进入系统的过程。  
密码管理（Manage Password）：用户更改或重置密码的功能。  
角色分配（Assign Roles）：管理员为用户分配角色的功能。  
权限管理（Manage Permissions）：管理员管理角色权限的功能。  
关系（Relationships）:  
用户与用例的关系：用户与注册、登录和密码管理用例相关联。  
管理员与用例的关系：管理员与角色分配和权限管理用例相关联。



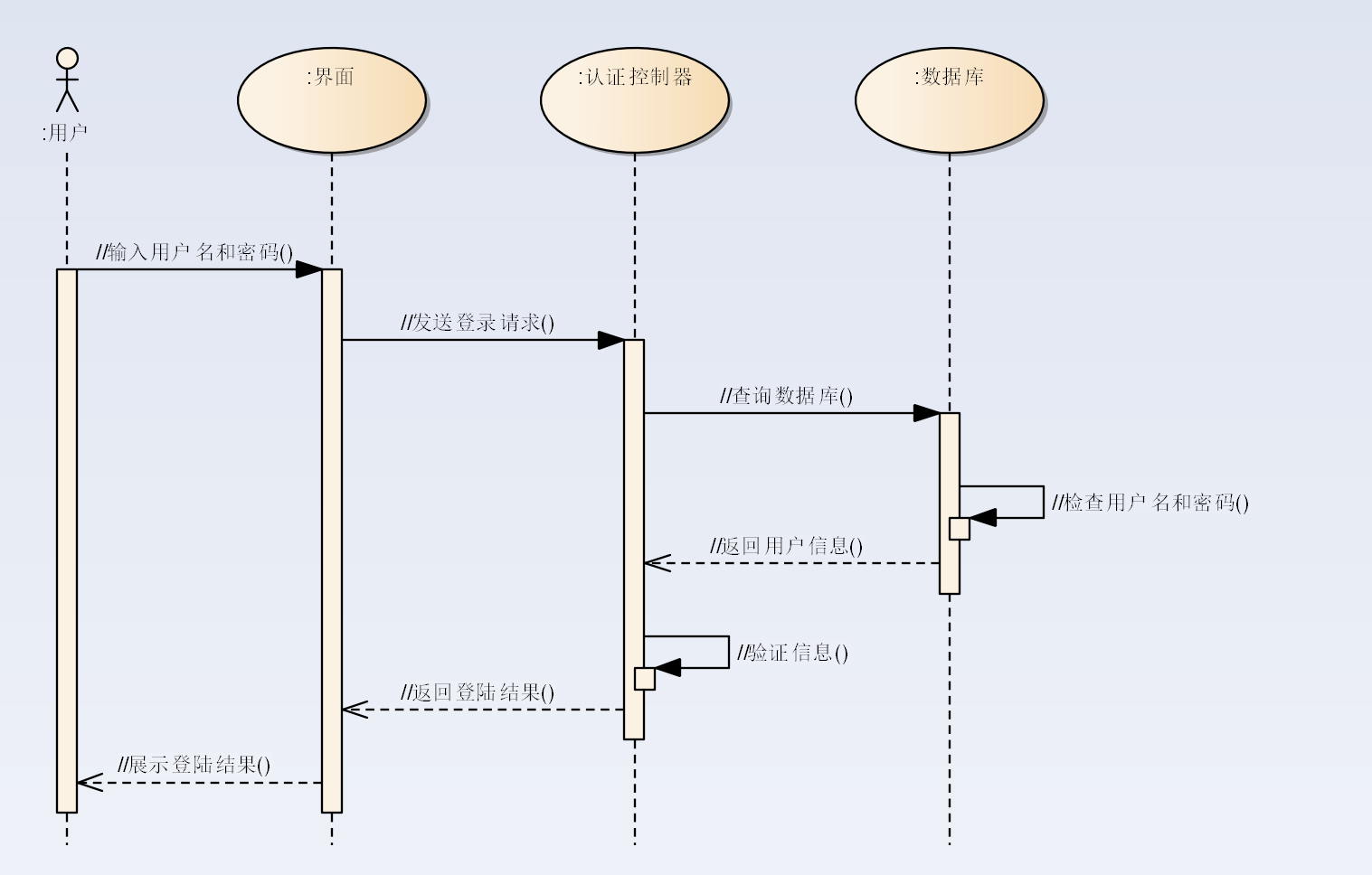
1. 类图

类（Classes）:  
用户（User）:  
属性：用户名（username）、密码（password）、电子邮件（email）、用户ID（userId）、角色列表（roles）。  
方法：创建账户（createAccount）、登录（login）、更改密码（changePassword）。  
角色（Role）:  
属性：角色名（roleName）、权限列表（permissions）。  
方法：添加权限（addPermission）、删除权限（removePermission）。  
权限（Permission）:  
属性：权限名（permissionName）、描述（description）。  
方法：无特定方法，主要用于访问控制。  
关系（Relationships）:  
用户与角色：多对多关系（一个用户可以有多个角色，一个角色可以分配给多个用户）。  
角色与权限：多对多关系（一个角色可以拥有多个权限，一个权限可以分配给多个角色）。



1. 用户登录过程—顺序图

参与者和对象：  
用户（User）：与系统交互的人。  
界面（UI）：用户界面，用户与系统交互的入口。  
认证控制器（Authentication Controller）：处理登录请求的后端控制器。  
数据库（Database）：存储用户数据的数据库。  
交互过程：  
用户在界面上输入用户名和密码。  
界面将登录请求发送给认证控制器。  
认证控制器查询数据库，检查用户名和密码。  
数据库返回用户信息给认证控制器。  
认证控制器验证信息，然后向界面返回登录结果（成功或失败）。



1. 用户登录过程—活动图

活动和决策点：

开始：表示流程的开始点。

用户输入用户名和密码：用户在界面上输入登录凭据。

系统验证凭据：系统检查输入的用户名和密码是否匹配。

决策点 - 凭据是否有效？

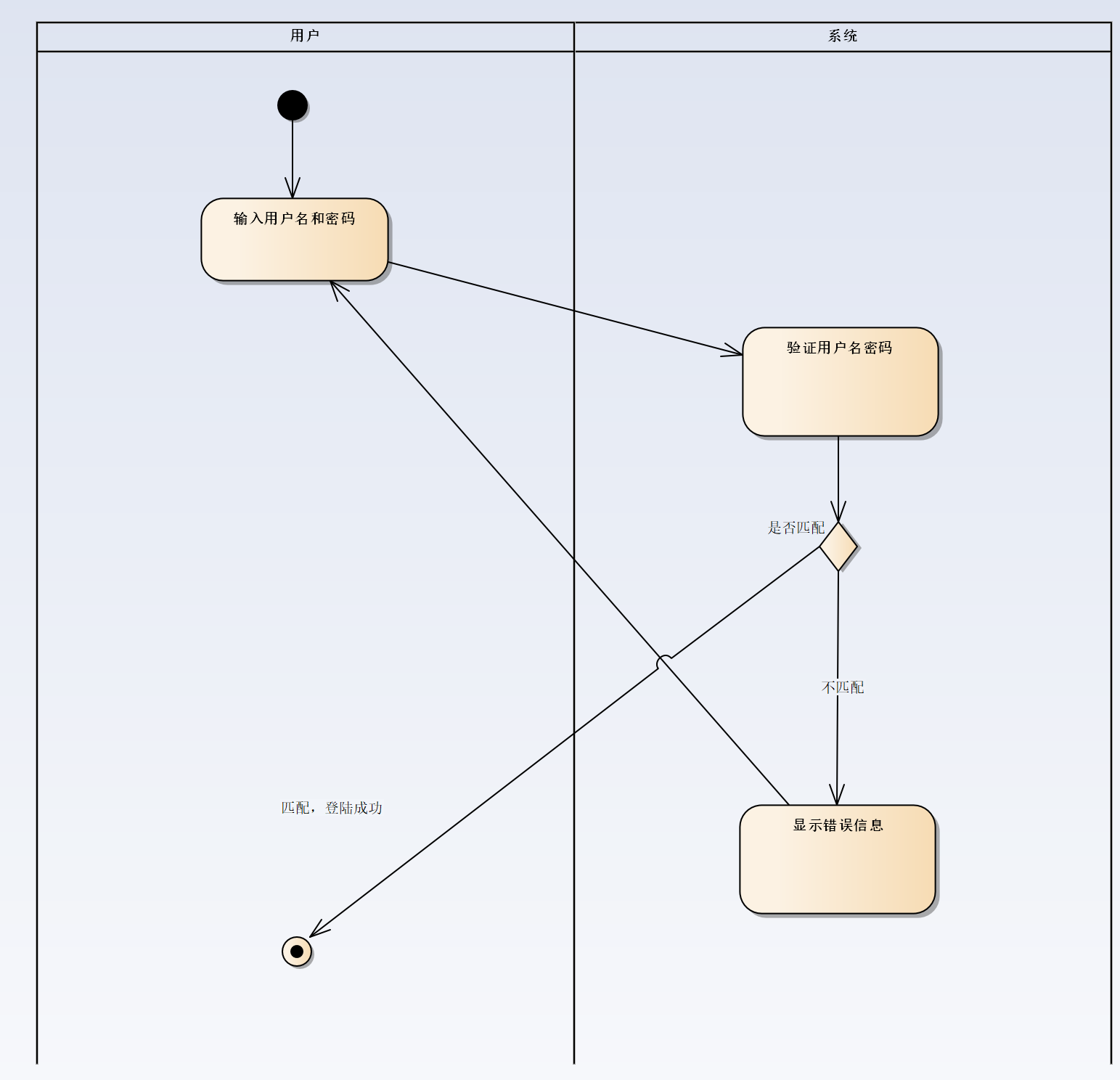
如果是（有效），则进行下一步。

如果否（无效），则转到“显示错误消息”。

显示错误消息：如果登录信息无效，显示错误消息并返回到“用户输入用户名和密码”。

登录成功：用户成功登录系统。

结束：表示流程的结束点。



订单管理

1. 用例图

参与者（Actors）:

客户（Customer）：在系统中购物并创建订单的人。

销售人员（Sales Staff）：负责订单的处理和管理。

库存管理人员（Inventory Staff）：处理与订单相关的库存问题。

用例（Use Cases）:

创建订单（Create Order）：客户选择产品并创建新订单。

查看订单（View Order）：客户和销售人员查看订单详情。

修改订单（Modify Order）：客户或销售人员更改订单内容。

取消订单（Cancel Order）：客户取消已创建的订单。

支付订单（Pay for Order）：客户对订单进行支付。

订单发货（Ship Order）：销售人员处理订单发货。

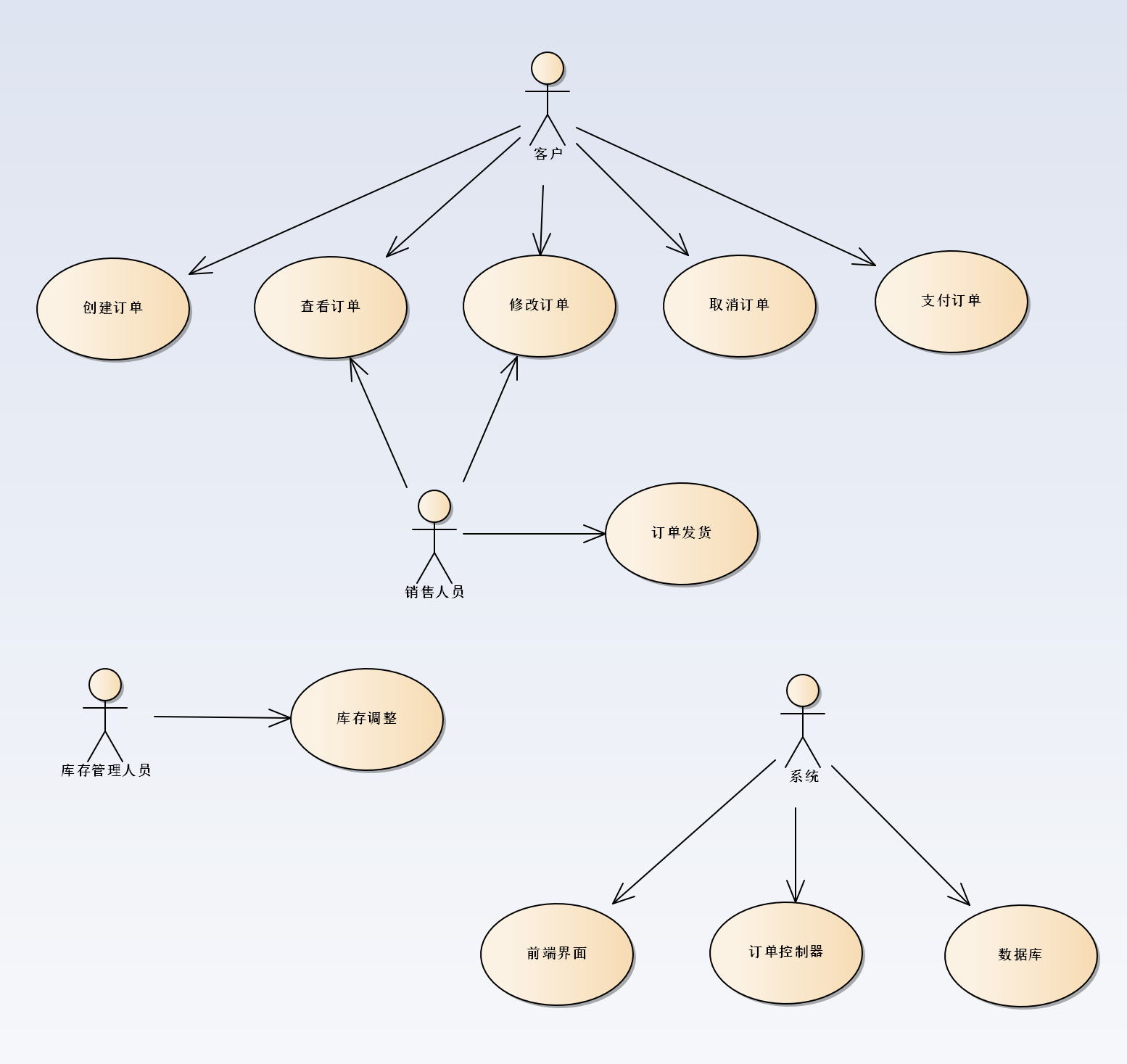
库存调整（Adjust Inventory）：库存管理人员根据订单调整库存。

关系（Relationships）:

客户与用例的关系：客户主要与创建订单、查看订单、修改订单、取消订单和支付订单相关联。

销售人员与用例的关系：销售人员参与查看订单、修改订单、订单发货等。

库存管理人员与用例的关系：库存管理人员涉及库存调整。



1. **类图**

类（Classes）:

订单（Order）:

属性: 订单ID（orderId）、订单日期（orderDate）、订单状态（status）、客户ID（customerId）、总金额（totalAmount）。

方法: 创建订单（create）、更新订单（update）、取消订单（cancel）、获取订单详情（getDetails）。

客户（Customer）:

属性: 客户ID（customerId）、姓名（name）、地址（address）、电子邮件（email）。

方法: 创建客户信息（create）、更新客户信息（update）、获取客户信息（getDetails）。

订单项（OrderItem）:

属性: 订单项ID（orderItemId）、订单ID（orderId）、产品ID（productId）、数量（quantity）、单价（unitPrice）。

方法: 添加订单项（addItem）、更新订单项（updateItem）、删除订单项（removeItem）。

产品（Product）:

属性: 产品ID（productId）、名称（name）、价格（price）、库存数量（inventoryQuantity）。

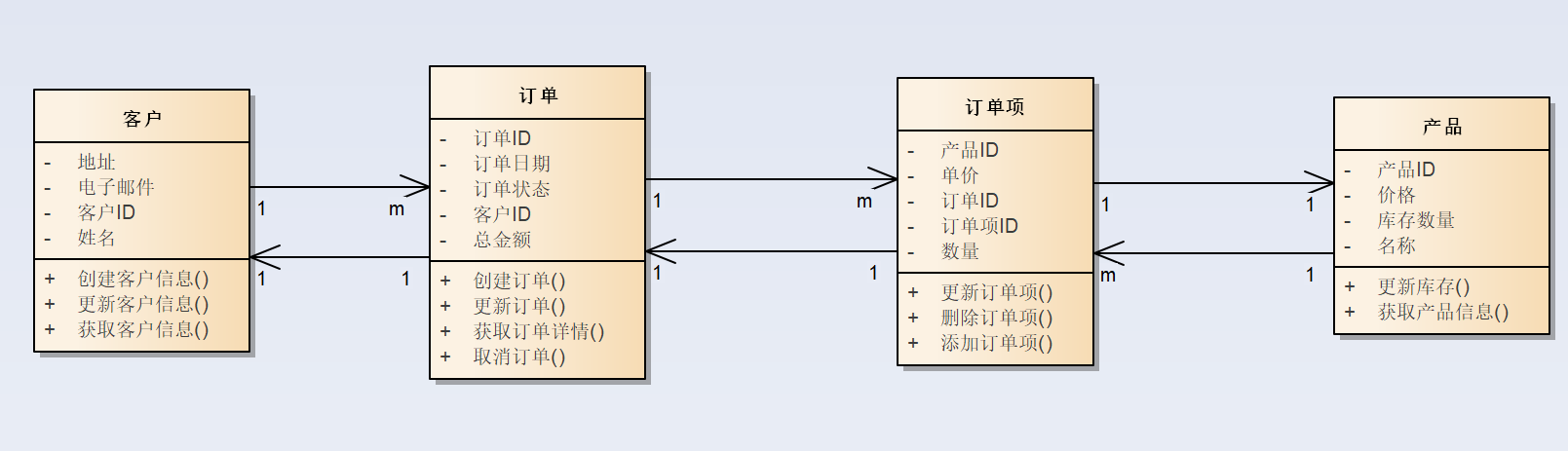
方法: 更新库存（updateInventory）、获取产品信息（getDetails）。

关系（Relationships）:

订单与客户：一个客户可以拥有多个订单，但每个订单只属于一个客户（一对多关系）。

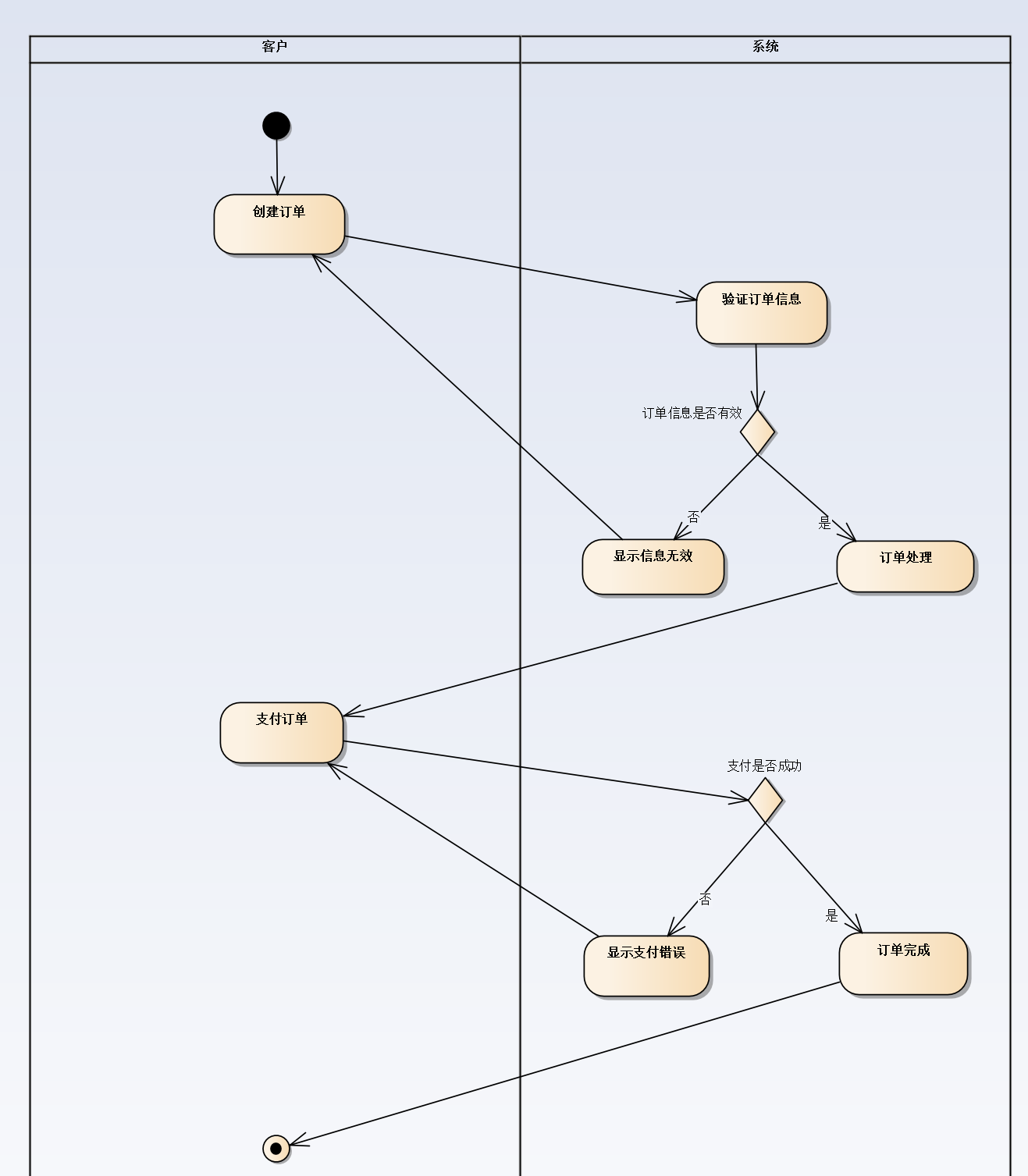
订单与订单项：一个订单可以包含多个订单项，每个订单项只属于一个订单（一对多关系）。

订单项与产品：每个订单项对应一个产品，产品可以在多个订单项中出现（多对一关系）。



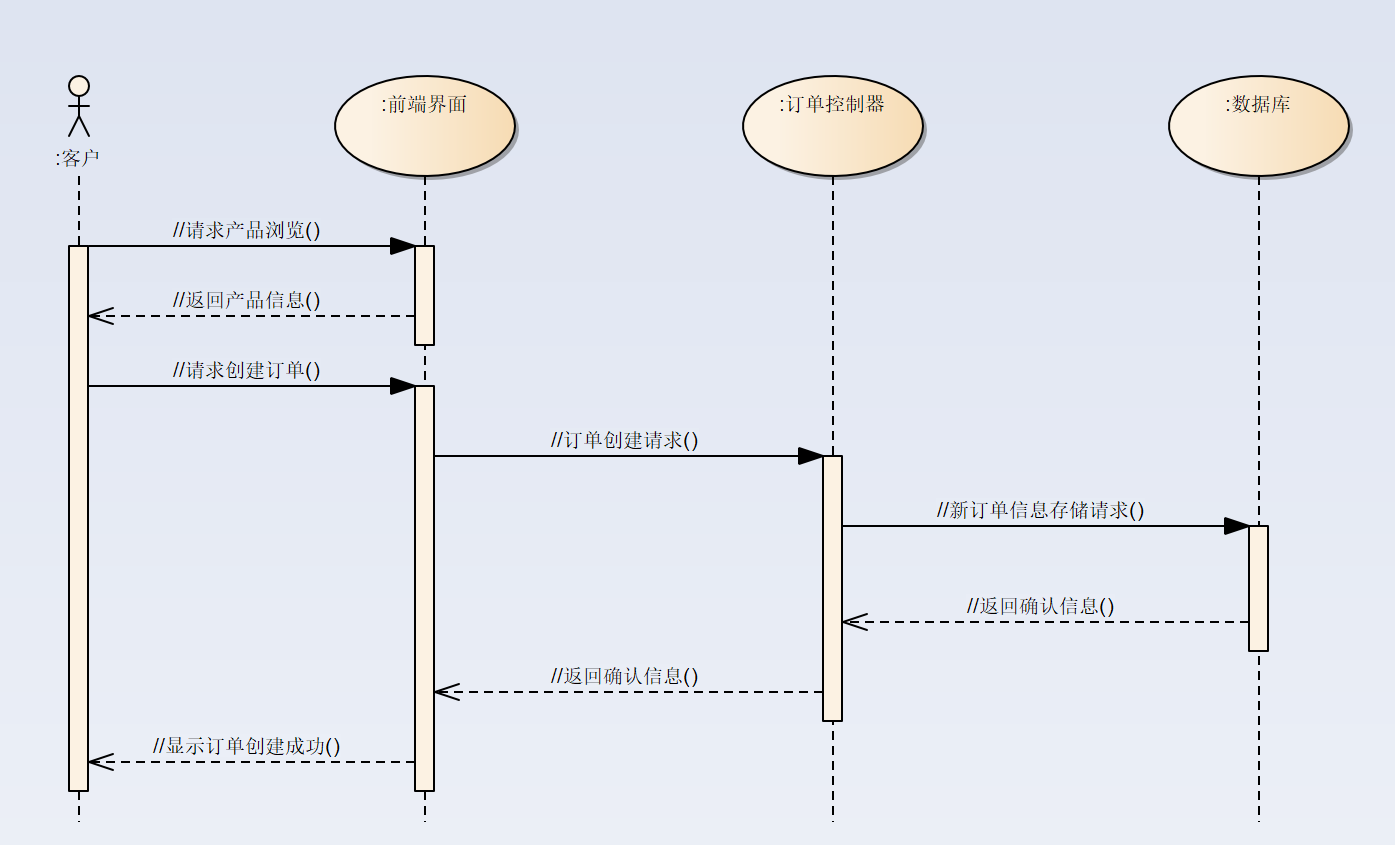
1. **订单创建和处理—活动图**

主要活动和决策点：  
开始：表示流程的开始点。  
客户创建订单：客户在系统中选择产品并发起订单。  
系统验证订单信息：系统检查订单信息是否完整和正确。  
决策点 - 订单信息是否有效？  
如果是（有效），则进入下一步。  
如果否（无效），则转到“显示错误消息”。  
显示错误消息：如果订单信息无效，向客户显示错误消息，并返回到“客户创建订单”。  
系统处理订单：系统对订单进行处理，如计算总价、更新库存等。  
订单支付：客户对订单进行支付。  
决策点 - 支付是否成功？  
如果是（成功），则进入下一步。  
如果否（失败），则转到“显示支付错误”。  
显示支付错误：如果支付失败，向客户显示错误消息，并返回到“订单支付”。  
订单完成：订单处理完成，系统更新订单状态。  
结束：表示流程的结束点。



1. **订单创建和处理—顺序图**

参与者和对象：  
客户（Customer）：发起订单创建的用户。  
前端界面（Frontend Interface）：客户与系统交互的界面。  
订单控制器（Order Controller）：后端服务，处理订单相关的逻辑。  
数据库（Database）：存储订单和客户数据。  
交互过程：  
客户在前端界面上选择产品并发起创建订单的请求。  
前端界面将创建订单的请求发送到订单控制器。  
订单控制器请求数据库以存储新订单信息。  
数据库确认订单信息已存储并返回确认信息给订单控制器。  
订单控制器将订单确认信息发送回前端界面。  
前端界面向客户显示订单创建成功的消息。



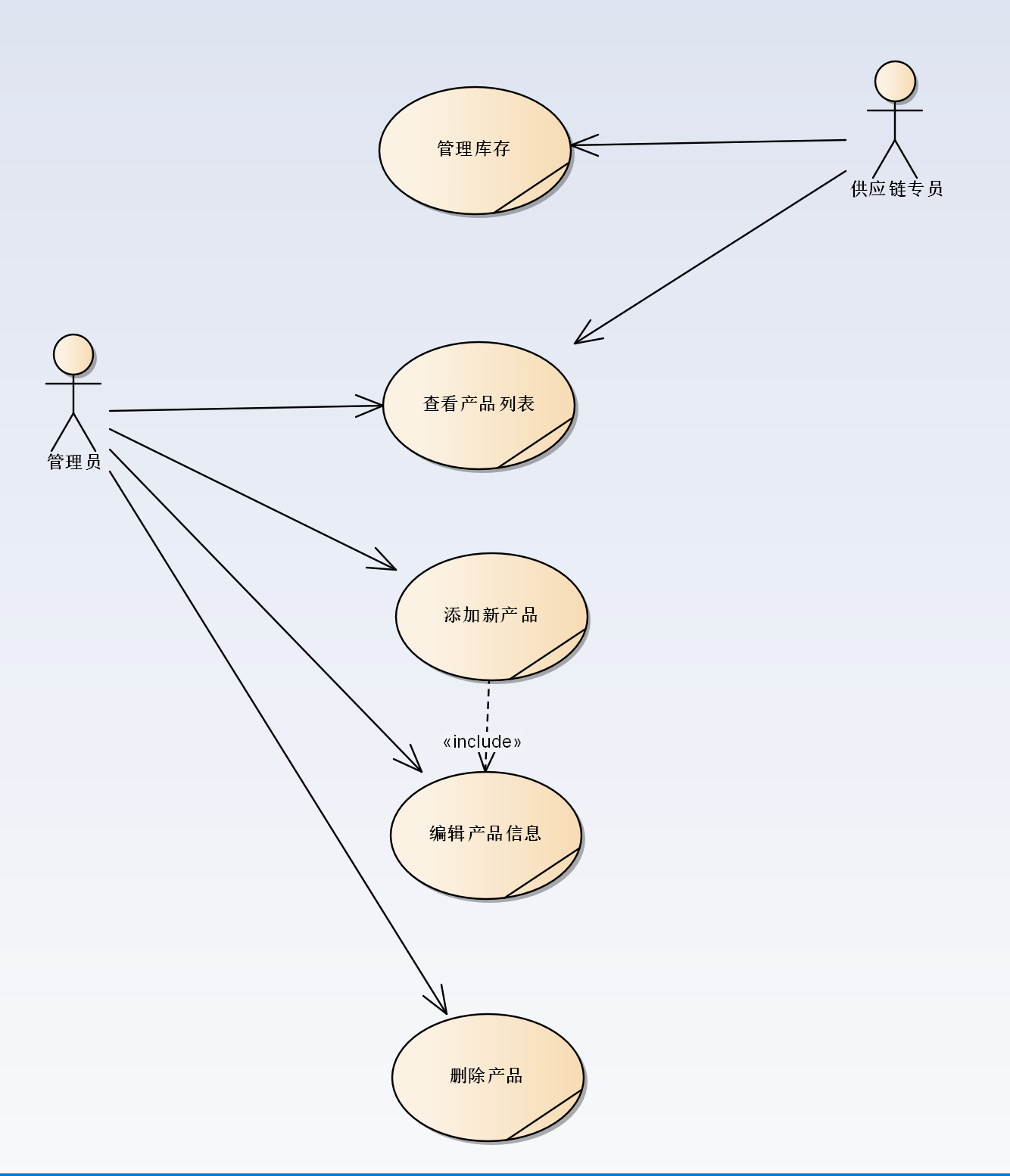
**产品管理**

1. **用例图**

参与者（Actors）:  
管理员：负责管理产品信息。  
供应链专员：需要访问库存数据来管理供应链。

用例（Use Cases）:  
添加新产品：管理员可以添加新产品到系统。  
编辑产品信息：管理员可以更新产品的价格、描述等信息。  
查看产品列表：管理员和供应链专员可以查看所有产品的列表。  
管理库存：供应链专员可以查看和更新库存信息。  
删除产品：管理员可以从系统中删除产品。

关系（Relationships）:  
关联（Associations）: 表示参与者与用例之间的交互。  
包含（Include）: 如果一个用例是另一个用例的必要部分。  
扩展（Extend）: 如果一个用例在某些条件下可以扩展或补充另一个用例。

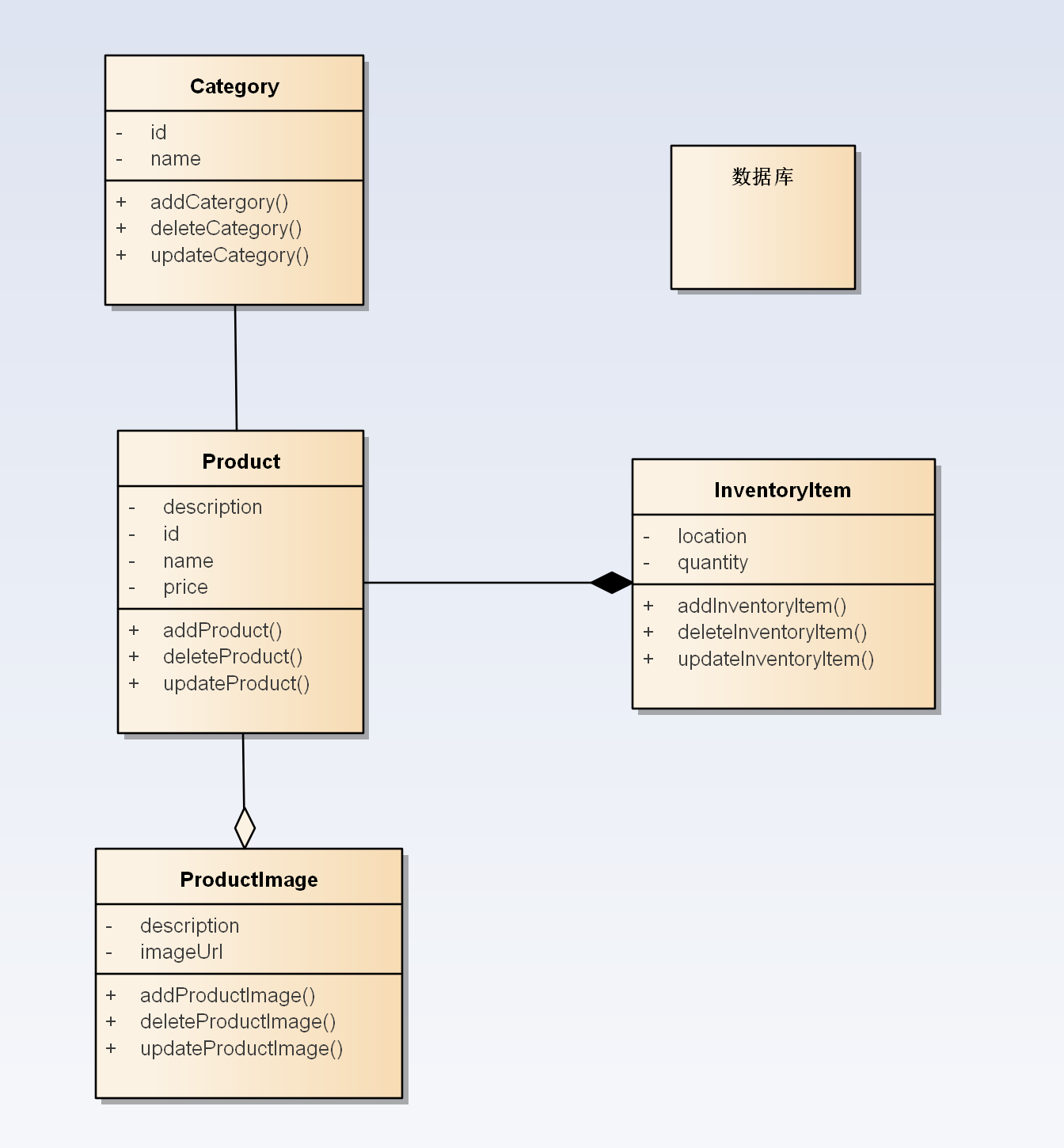


1. **类图**

主要类:  
Product：代表产品，具有属性如id、name、description、price等。  
Category：代表产品分类，具有属性如id、name。  
InventoryItem：代表库存项，与特定产品相关联，具有属性如quantity、location。  
ProductImage：代表产品图片，具有属性如imageUrl、description。

关系:  
关联（Association）: 例如，Product与Category之间有关联，表示产品属于某个分类。  
聚合（Aggregation）: Product聚合多个ProductImage，表示产品可以有多张图片。  
组合（Composition）: InventoryItem与Product之间是组合关系，表示库存项直接依赖于产品。

方法:  
每个类可以有相应的方法，如addProduct(), updateProduct(), deleteProduct() 在Product类中，用于管理产品的生命周期。  
示意图描述  
Product 类位于中心，包含多个属性，如id, name, description, price。  
Category 类与 Product 类通过关联相连，表示产品分类。  
InventoryItem 类与 Product 类通过组合关系相连，表示库存项的依赖。  
ProductImage 类与 Product 类通过聚合关系相连，表示产品可以有多张图片。



1. **顺序图**

参与对象：

管理员（User）：与系统交互的用户。

前端界面（UI）：用户界面，用于输入产品数据和显示响应。

控制器（Controller）：处理前端发送的请求。

产品服务（ProductService）：负责产品相关的业务逻辑。

数据库（Database）：存储产品信息。

交互流程：

管理员发起添加产品请求：管理员在UI上填写产品信息，如名称、价格、描述，并提交。

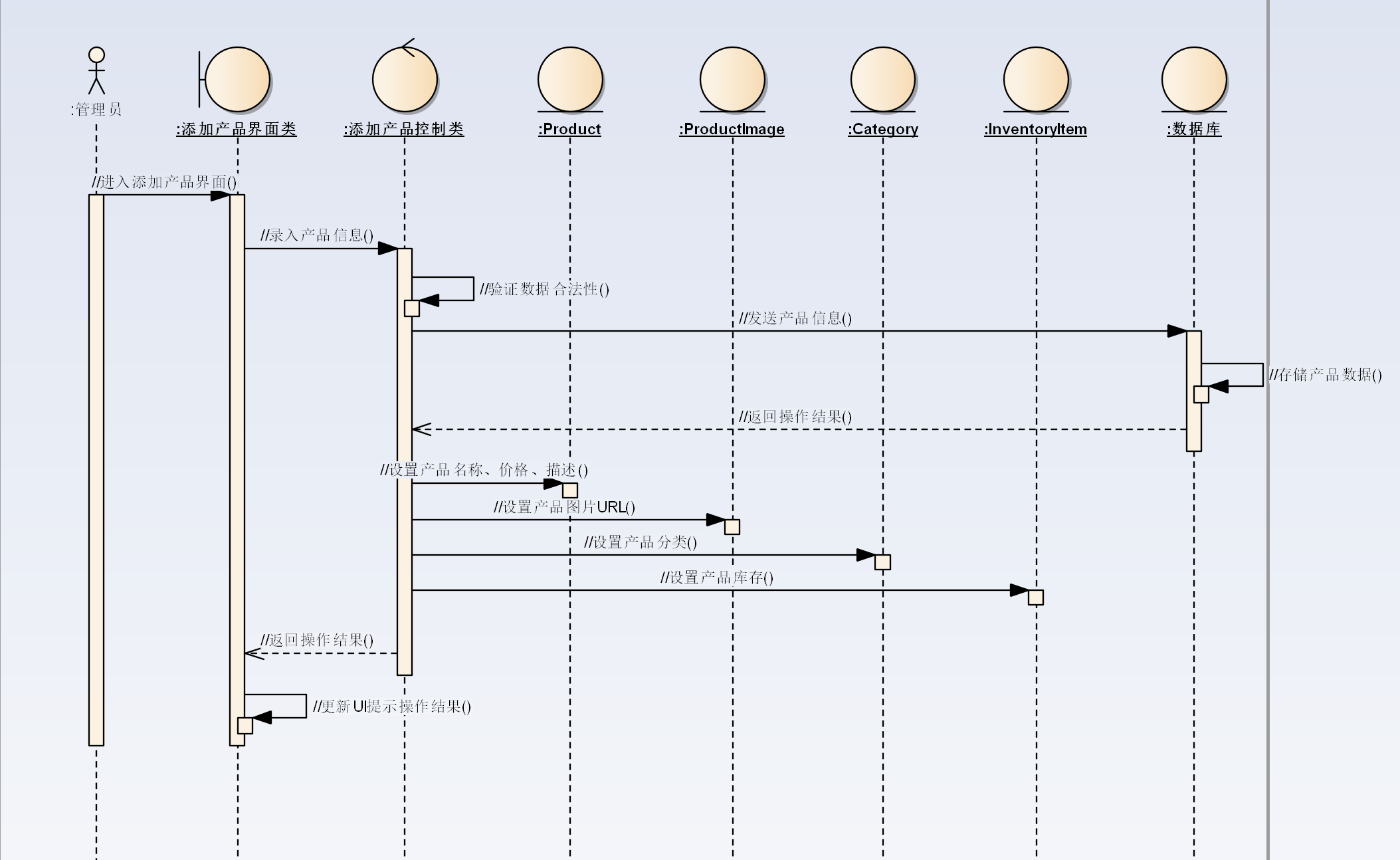
UI发送数据到控制器：UI捕获输入数据，发送请求到后端的控制器。

控制器请求产品服务处理数据：控制器接收请求，调用产品服务来处理产品数据。

产品服务与数据库交互：产品服务验证数据合法性，然后将数据存储到数据库。

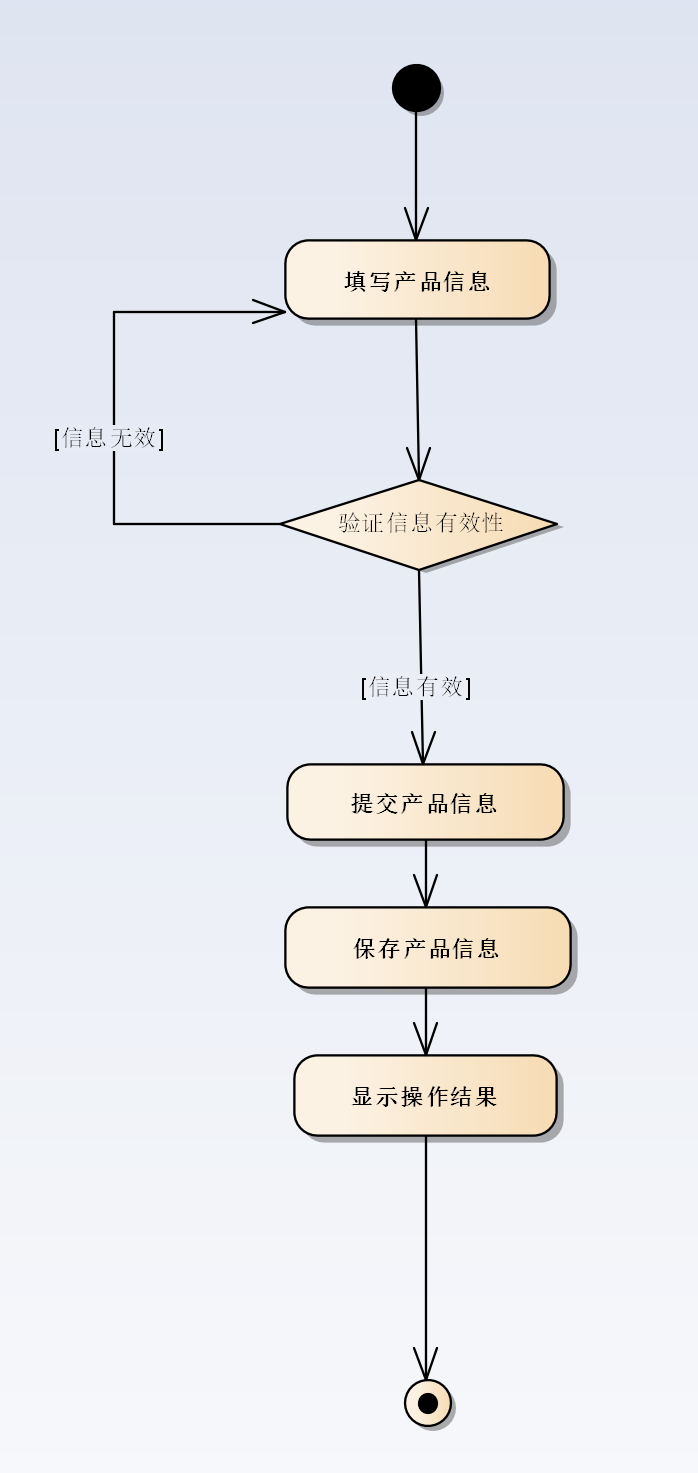
返回操作结果：一旦数据存储完成，产品服务将操作结果返回给控制器，控制器再将结果返回给UI。

UI显示响应：UI根据控制器返回的结果，向管理员显示操作成功或错误信息。



1. **活动图**

主要活动：  
开始添加产品：这是流程的起点。  
填写产品信息：管理员在用户界面填写产品的详细信息，如名称、描述、价格、分类等。  
验证信息有效性：系统检查输入的信息是否完整且有效。  
判断信息是否有效：  
如果信息有效，继续下一步。  
如果信息无效，返回重新填写。  
提交产品信息：将有效的产品信息提交到系统。  
保存产品信息：系统处理提交的信息，并将其保存到数据库。  
显示操作结果：根据保存操作的结果，向管理员显示成功或错误消息。  
结束流程：添加产品的流程结束。

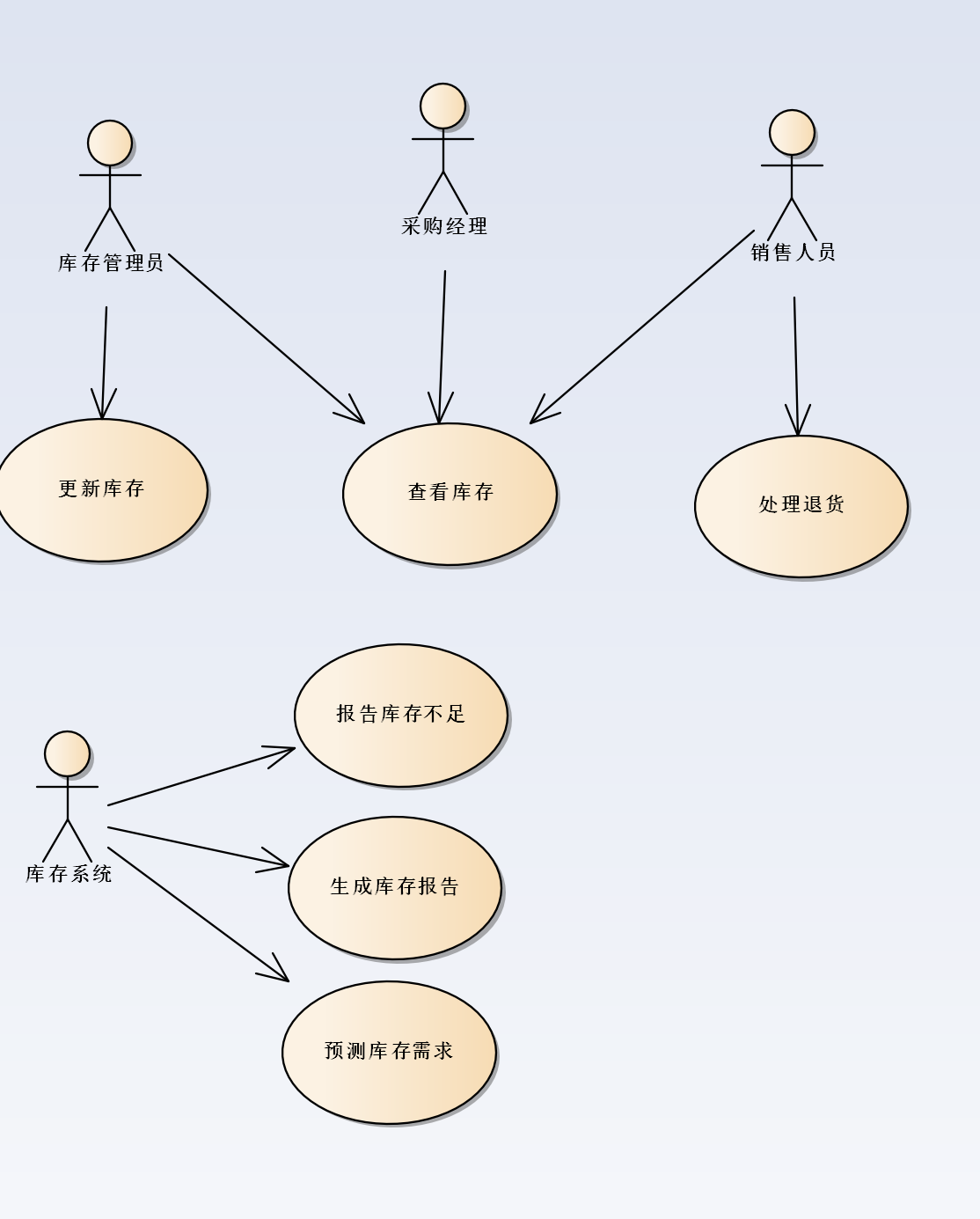


**库存管理**

1. **用例图**

参与者（Actors）  
库存管理员：负责管理库存的日常操作。  
销售人员：需要查看库存信息，以了解产品的可用性。  
采购经理：负责补充库存，需要查看库存水平和历史数据。

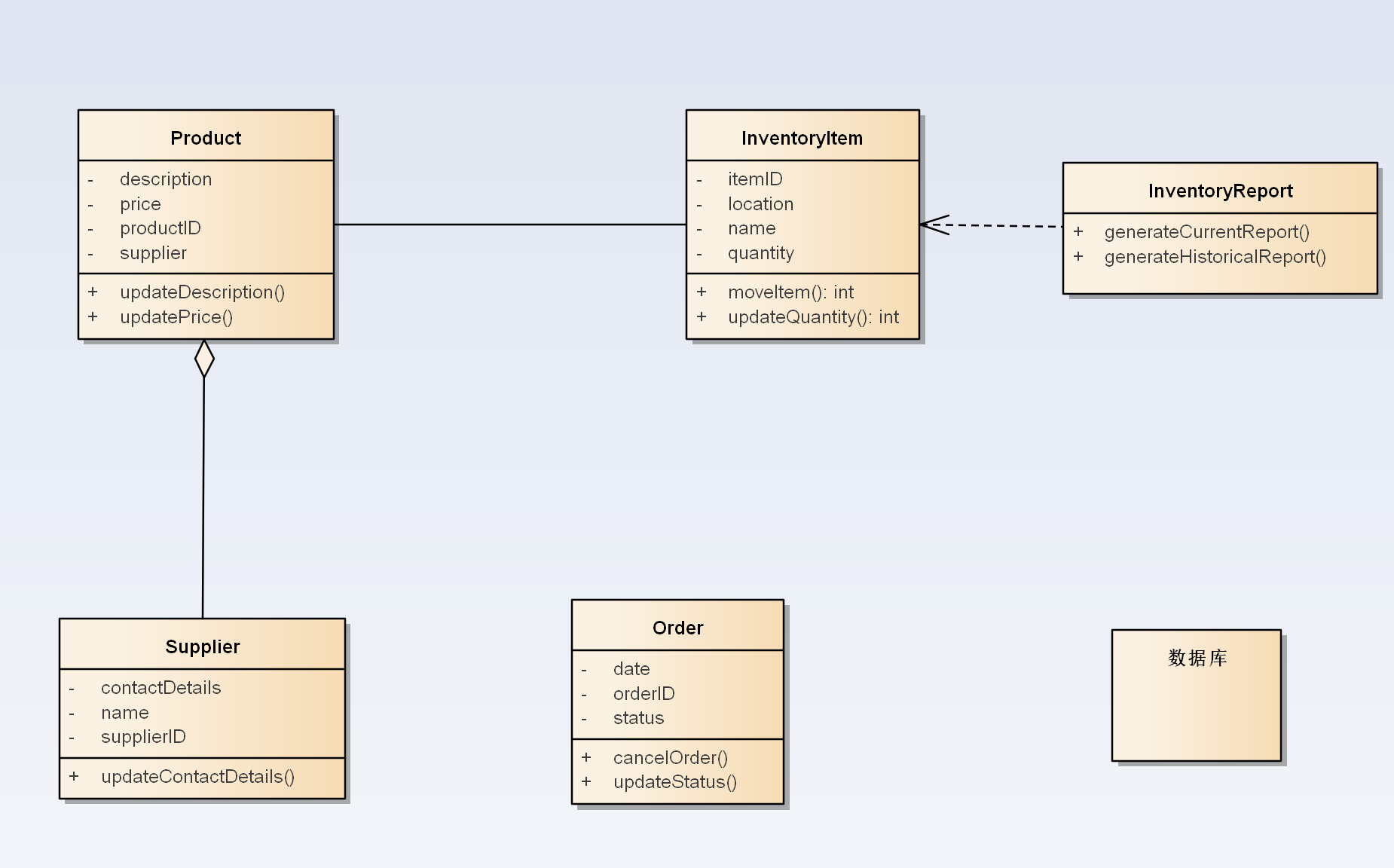
用例（Use Cases）  
查看库存：用户可以查看当前的库存水平。  
更新库存：库存管理员更新库存数量，例如在收到新货物或销售产品后。  
报告库存不足：系统自动报告低库存水平。  
生成库存报告：生成关于当前库存状态的报告。  
预测库存需求：基于历史数据预测未来的库存需求。  
处理退货：管理退回的产品并更新库存。

****

1. **类图**

主要类  
InventoryItem: 代表库存中的单个物品。  
- 属性  itemID, name, quantity, location 等。  
- 方法 updateQuantity(), moveItem() 等。  
Product: 与 InventoryItem 相关联的产品详情。  
- 属性  productID, description, price, supplier 等。  
- 方法  updatePrice(), updateDescription() 等。  
Supplier: 提供产品的供应商信息。  
- 属性  supplierID, name, contactDetails 答。  
- 方法  updateContactDetails() 等。  
InventoryReport: 生成库存报告。  
- 方法  generateCurrentReport(), generateHistoricalReport() 等。  
Order: 如果库存管理涉及订单处理。  
- 属性  orderID, date, status 等。  
- 方法  updateStatus(), cancelOrder() 等。

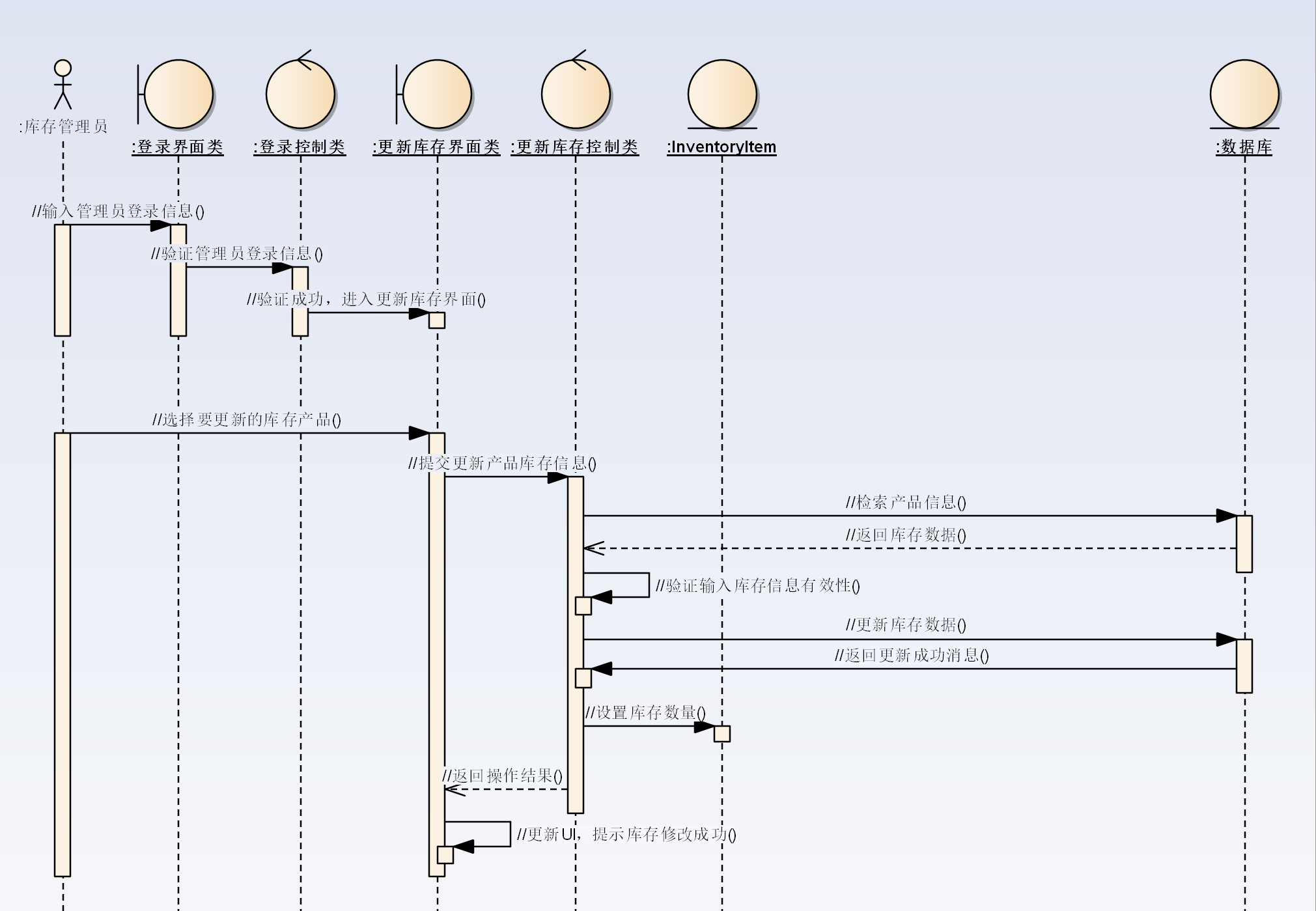
关系  
关联（Association）: InventoryItem 和 Product 之间有关联，表示库存项对应特定的产品。  
聚合（Aggregation）: Supplier 与 Product 有聚合关系，表示供应商提供多种产品。  
依赖（Dependency）: InventoryReport 依赖于 InventoryItem，以获取生成报告所需的数据。



1. **顺序图**

参与对象：  
库存管理员：执行更新库存操作的用户。  
库存管理系统：库存管理模块的主要组件，负责处理库存数据。  
产品：代表库存中的一个产品。  
数据库：用于存储库存数据。

交互流程：  
库存管理员请求更新库存：库存管理员启动更新库存操作。  
库存管理系统验证管理员身份：系统验证管理员的身份，确保他们有权限执行此操作。  
库存管理员选择产品：管理员选择要更新库存的产品。  
库存管理系统检索产品信息：系统从数据库中检索所选产品的信息，包括当前库存量。  
库存管理员更新库存数量：管理员输入新的库存数量。  
库存管理系统验证库存数量：系统验证输入的库存数量的有效性。  
库存管理系统更新库存信息：如果库存数量有效，系统更新数据库中的库存信息。  
库存管理系统生成更新确认：系统生成更新成功的确认消息。  
库存管理员接收更新确认：管理员接收并查看更新成功的确认消息。



1. **活动图**

主要活动：  
开始操作：这是流程的起点。  
登录库存管理系统：管理员登录库存管理系统以执行操作。  
选择操作：管理员选择要执行的库存管理操作，例如更新库存、生成库存报告等。  
执行操作：管理员执行所选操作的具体步骤。  
验证操作结果：系统验证操作的结果，如果成功，进入下一步。  
生成报告：如果需要，系统可以生成库存报告。  
结束操作：操作流程结束。  
