电子商务平台用例图设计与分析报告

### E-commerce Platform: Use Case Diagram Design & Analysis Report

(开始演示时，将完整的图展示在屏幕上)

(When starting the presentation, display the complete diagram on the screen)

大家好！今天我为大家展示的是我为电子商务平台设计的最终版用例图。我的核心设计思想是，**不只把它看作一个功能列表，而是把它看作一个连接了多方参与者的、动态的业务生态系统。**

Hello everyone! Today I'll be presenting the final version of the Use Case Diagram I designed for an e-commerce platform. My core design philosophy was **not to view it as just a feature list, but as a dynamic business ecosystem that connects multiple participants.**

(手指或激光笔指向图的右下角，从普通用户开始)

(Point a finger or laser pointer to the bottom right of the diagram, starting with the regular user)

我们故事的起点，是普通用户 **user Jerry**。他是平台的核心，他的目标很简单：完成一次愉快的购物。他可以 Browse and Search Products (浏览商品)，可以 Manage Shopping Cart (管理购物车)，最终 Place Order (下订单)。

The starting point of our story is the regular user, **user Jerry**. He is the core of the platform, and his goal is simple: to have a pleasant shopping experience. He can Browse and Search Products, Manage Shopping Cart, and ultimately Place Order.

这里，我用 <<include>> 关系强调了下单流程的**强制步骤**：用户**必须** User Login (登录)，**必须** Confirm Shipping Address (确认地址)，也**必须** Make Payment (进行支付)。

Here, I've used the <<include>> relationship to emphasize the **mandatory steps** of the ordering process: a user **must** User Login, **must** Confirm Shipping Address, and **must** Make Payment.

同时，为了提升用户体验，我还设计了两个 <<extend>> 关系，代表**可选操作**：用户可以 Apply Coupon (使用优惠券) 或者 Select Shipping Method (选择配送方式)。这些 extend 关系上我还特意标注了触发条件，比如只有当 User enters a valid coupon code (用户输入了有效优惠券码) 时，优惠券的逻辑才会被激活。

At the same time, to enhance the user experience, I also designed two <<extend>> relationships representing **optional actions**: the user can Apply Coupon or Select Shipping Method. I've specifically noted the trigger conditions on these extend relationships; for instance, the coupon logic is only activated when the User enters a valid coupon code.

(手指移动到左下角，讲解VIP用户)

(Move the pointer to the bottom left to explain the VIP user)

在这个平台上，我们还有 **VIP user Smith**。我用了一条**泛化箭头**从Smith指向Jerry，这在UML里表示 ‘VIP用户 **是一种** 普通用户’。这意味着Smith能做Jerry能做的一切，我们就不需要重复连线了，让图表非常简洁。同时，Smith还连接了一个他专属的用例：Access VIP Service (享受VIP服务)，体现了他的特权。

On this platform, we also have **VIP user Smith**. I used a **generalization arrow** from Smith to Jerry, which in UML means 'a VIP user **is a type of** regular user'. This implies Smith can do everything Jerry can, so we don't need to draw redundant lines, keeping the diagram clean. Meanwhile, Smith is also connected to his exclusive use case: Access VIP Service, reflecting his special privileges.

(手指移动到图的右侧，讲解外部系统)

(Move the pointer to the right side of the diagram to explain the external systems)

一个真实的电商平台不是孤立的。因此，我引入了两个**外部系统参与者**来体现它与真实世界的连接。

A real e-commerce platform is not isolated. Therefore, I've introduced two **external system actors** to show its connection to the real world.

第一，当用户 Make Payment (进行支付) 时，这个动作会触发 Ensure Payment (确保支付) 用例，而这个用例需要与外部的 **Payment Gateway** (支付网关) 进行交互。这保证了交易的真实与安全。

First, when a user makes a payment, this action triggers the Ensure Payment use case, which in turn needs to interact with an external **Payment Gateway**. This ensures the authenticity and security of the transaction.

第二，当用户 Select Shipping Method (选择配送方式) 后，系统需要 Confirm Shipping Method (确认配送方式)，这个过程则需要与外部的 **Logistics Provider** (物流供应商) 系统通信，以获取实时的运费和时效信息。

Second, after a user selects a shipping method, the system needs to Confirm Shipping Method, a process that requires communication with an external **Logistics Provider** system to get real-time shipping costs and delivery estimates.

(手指移动到图的中间，连接用户和后台，这是报告的亮点)

(Move the pointer to the center of the diagram, connecting the user and the backend—this is the highlight of the report)

用户可以对产品发布评论。商家会在Manage Comments中对用户的评论进行查看和回复，对不当管理进行处理；而管理员的Monitor for Fraudulent Activity也会对该用户的评论进行监控分析，如果系统算法认为这条评论有‘刷单’或‘虚假好评’的嫌疑，它就会被自动标记，并通知管理员。

这个设计，让前台用户的简单操作和后台复杂的管理、监控逻辑，形成了一个动态的、实时的联动。

This design creates a dynamic, real-time link between a simple front-end user action and the complex management and monitoring logic on the backend.

(手指移动到左上角和中上部，总结后台角色)

(Move the pointer to the top left and top center to summarize the backend roles)

最后，我们来看后台的两位管理者。**shopowner Tom** 负责所有日常经营，包括 Manage Products (商品), Manage Orders (订单) 等。他的工作与用户的行为紧密相连。而 **System Administrator Tim** 则是平台的守护者。他不管具体的某个订单，而是负责整个平台的生态健康，比如我们刚才提到的**监控欺诈活动**，以及定义平台核心盈利模式的 **Manage Platform Commission Rules** (管理平台佣金规则)。

Finally, let's look at the two administrators on the backend. **shopowner Tom** is responsible for all daily business operations, including Manage Products, Manage Orders, and so on. His work is closely tied to user activities. In contrast, **System** Administrator Tim is the platform's guardian. He doesn't deal with specific orders but is responsible for the overall health of the platform's ecosystem, such as the \*\*