Để nối các quan hệ trong sơ đồ lớp của hệ thống e-commerce dựa trên file bạn cung cấp, đây là cách các mối quan hệ giữa các lớp được biểu diễn bằng **multiplicity**, **generalization**, **aggregation**, **composition**, và **association**:

1. **Generalization (kế thừa)**:
   * **Customer** và **Admin** là hai lớp con kế thừa từ lớp **User**:
     + User ⟶ Customer (Generalization)
     + User ⟶ Admin (Generalization)
2. **Association (liên kết)**:
   * **Customer** có một **Cart** (1:1):
     + Customer ⟶ Cart (1:1)
   * **Customer** có nhiều **Order** (1

):

* + - Customer ⟶ Order (1

)

* + **Order** chứa nhiều **OrderItem** (1

):

* + - Order ⟶ OrderItem (1

)

* + **Cart** chứa nhiều **CartItem** (1

):

* + - Cart ⟶ CartItem (1

)

* + **OrderItem** chứa một **Product** (N:1):
    - OrderItem ⟶ Product (N:1)
  + **CartItem** chứa một **Product** (N:1):
    - CartItem ⟶ Product (N:1)
  + **Review** được viết bởi một **Customer** (N:1) và liên kết với một **Product** (N:1):
    - Customer ⟶ Review (1

)

* + - Product ⟶ Review (1

)

* + **Order** có một **Payment** và một **ShippingTracker** (1:1):
    - Order ⟶ Payment (1:1)
    - Order ⟶ ShippingTracker (1:1)

1. **Aggregation (tổng hợp)**:
   * **Category** chứa nhiều **Product** (1

):

* + - Category ⟶ Product (1

)

1. **Composition (thành phần)**:
   * **Order** được tạo thành từ nhiều **OrderItem** (1

), nếu **Order** bị xoá thì **OrderItem** cũng bị xoá:

* + - Order ⟶ OrderItem (Composition)

1. **Dependency (phụ thuộc)**:
   * **CartItem** phụ thuộc vào **Product** để thực hiện tính toán:
     + CartItem ⟶ Product (Usage dependency)
2. **Multiplicity**:
   * Mỗi **Customer** có thể có nhiều **Order** (1

).

* + Mỗi **Order** bao gồm nhiều **OrderItem** (1

).

* + Một **Product** có thể xuất hiện trong nhiều **OrderItem** và **CartItem** (1

).

Bố trí này cho phép tạo ra sơ đồ UML hoàn chỉnh, với các mối quan hệ chính xác giữa các lớp trong hệ thống e-commerce của bạn.