Lecture Topic Task # 311: Entities and Value Objects in the Hand-Me-Down Project

Assignee: Daniella Melero

Milestone 3

In the Hand-Me-Down clothing marketplace, several domain objects illustrate the distinction between Entities and Value Objects. Entities are defined by a continuous identity that persists even as their attributes change, while Value Objects are defined purely by their properties and are interchangeable when their values are equal.

1. Listing – Entity

A Listing represents a posted item for sale, such as "Green UPR hoodie, size M, $15." Identity continuity is essential because each listing has its own lifecycle and interactions, such as views, messages, and status updates. Even if the description, photos, or price are edited, the system must still recognize it as the same listing for analytics, favorites, and transactions. In software, this would be modeled with a stable identifier, for example Listing { id, sellerId, title, price, condition, status }, where equality depends on the id, not the mutable attributes.

2. User Account – Entity

A User is a core participant in the system and must maintain a consistent identity across time. A user may change their profile picture or email, yet their identity remains constant to preserve history such as ratings, orders, and chat records. This persistence makes the User an Entity, typically modeled as User { id, name, email, rating, ... }, where userId is the primary key ensuring the same individual is recognized even after attribute updates.

3. Order / Transaction – Entity

An Order or Transaction documents the agreement between a buyer and a seller. Identity continuity matters because each transaction can evolve through multiple states: created, pending payment, completed, or canceled. Each order's events, confirmations, refunds, or disputes—must link to the same transaction identity. This would be represented in software as Order (id, buyerId, sellerId, listingId, status, paymentInfo), with the id ensuring event consistency over time.

4. Money / Price – Value Object

Money or Price captures an amount and currency (e.g., $15 USD). Its value alone defines equality; two Money objects with the same amount and currency are indistinguishable. Identity continuity is irrelevant since $15 USD is always exchangeable with any other $15 USD. Modeled as Money (amount, currency), it is immutable and replaced when a price changes rather than updated in place.

5. Item Condition – Value Object

ItemCondition expresses the qualitative state of a product (e.g., New, Like New, Good, Fair). This classification has no independent identity—it merely describes a property of a listing. When

a seller changes an item's condition, the previous value is replaced rather than modified. Implemented as an enumeration such as enum ItemCondition (NEW, LIKE_NEW, GOOD, FAIR), equality depends entirely on the literal value, not on any unique identifier.

Reflection and Peer Review

During peer review, classifications can be challenged by re-examining whether an object's identity must persist. For example, an Address could initially be treated as a Value Object (copied into each order), but if the team later decides that users can manage and update saved addresses independently, it might instead be modeled as an Entity. The guiding principle remains:

If the object's lifecycle or history matters, it is an Entity. If only its descriptive properties matter, it is a Value Object.