Domain Object Life Cycle Specification

Table of Contents

1. Purpose

This document defines the life cycle of domain objects in the **Hand-Me-Down** project. It establishes a consistent specification for how objects (e.g., clothing items) are created, modified, stored, retrieved, archived, and deleted.

The goal is to provide a clear, shared understanding for all contributors, ensuring predictable handling of data across the system. **Note:** This document is design-only. No implementation is included here.

2. Domain Objects

The primary domain object in this project is the **Clothing Item**.

2.1. Example Schema (Conceptual)

```
{
  "id": "uuid",
  "title": "Vintage Denim Jacket",
  "description": "Lightly used, size M",
  "size": "M",
  "condition": "good",
  "images": ["https://example.com/img1.jpg"],
  "donor_id": "uuid-of-donor",
  "status": "active",
  "created_at": "2025-01-10T14:32:00Z",
  "updated_at": "2025-01-10T14:32:00Z",
  "archived_at": null,
  "deleted_at": null,
  "version": 0,
  "tags": ["jacket", "denim"],
  "price": 15.00,
  "metadata": {}
}
```

3. Life Cycle Stages

3.1. 1. Creation

- Objects must be instantiated with all **required attributes** (`title`, `status`, timestamps).
- Validation rules should cover:
  - Title cannot be empty.
  - Size and condition should follow controlled vocabularies.

    - Images, if present, must be valid URLs.
- `created_at` and `updated_at` should be initialized when the item is first recorded.

3.2. 2. Modification

- Only specific fields can be updated:
    - `title`, `description`, `size`, `condition`, `images`, `tags`, `price`, `metadata`.
- Fields such as `id`, `donor_id`, and `created_at` are **immutable**.
- Each update should increment a `version` counter.
- `updated_at` must always be refreshed on modification.

3.3. 3. Storage

- Objects are persisted in the Supabase database (`clothing_items` table).
- Recommended storage fields:
    - `uuid` primary key
    - Enum for `status` (`active`, `archived`, `deleted`)
    - GIN indexes for arrays (`tags`, `metadata`) for efficient querying.
- Images are stored in Supabase Storage (bucket: `clothing-images`).

3.4. 4. Reconstitution

- When retrieved from persistence, objects must be rehydrated into their complete schema.
- Nullables:
    - `archived_at` and `deleted_at` may be `null` until lifecycle events occur.
- JSON fields like `metadata` should default to `{}`.

3.5. 5. Archiving

- An item may be **archived** when it is no longer active but still needs to be retained.
- Rules:
    - `status` is set to `archived`.
    - `archived_at` timestamp is recorded.
    - Object remains queryable in the archive but excluded from active listings.

3.6. 6. Deletion

- Two levels:
    - **Soft delete** → mark item as `deleted`, set `deleted_at`.

    - **Permanent delete** → row removal is permitted only after archival (compliance safeguard).
- Guardrails:
    - Prevent accidental removal by requiring an explicit archive step first.
    - Permanent deletion must follow retention policies.

4. Example State Transitions

```
[*] --> Active : Create
Active --> Active : Modify
Active --> Archived : Archive
Archived --> Deleted : Soft delete
Deleted --> [*] : Permanent delete
```