Hand Me Down Clothing Documentation

Table of Contents

1 Informative Part . . . 1
1.1 Team . . . 1
1.2.1 Current Situation . . . 2
1.2.2 Need . . . 3
1.3.1 Scope & Span . . . 3
1.3.2 Synopsis . . . 4
1.4 Other Activities than Just Developing Source Code . . . 5
1.5 Derived Goals . . . 8
2.1 Domain Description . . . 8
2.1.1 Domain Rough Sketches . . . 10
2.1.2 - Terminology . . . 11
2.1.3 Domain Terminology in Relation to Domain Rough . . . 12
2.1.4 Domain Narrative . . . 13
2.1.5 Events, Actions, and Behaviors . . . 14
2.1.6 - Function Signatures . . . 16
2.2.1 Epics, Features, and User Stories . . . 19
Epics . . . 19
Features . . . 20
User Stories . . . 21
2.2.2 - Personas . . . 21
2.2.3 Domain Requirements . . . 24
2.2.4 Interface Requirements . . . 25
2.2.5 Machine Requirements . . . 26
2.3 Implementation . . . 28
2.3.1 Selected Fragments of the Implementation . . . 31
3.1 Concept Formation and Analysis . . . 38
3.2 Validation and Verification . . . 39
Logbook . . . 41

1 Informative Part

1.1 Team

The team is organized into key functional areas with dedicated leads overseeing documentation and requirement completion, authentication, listings, and search and map integration, under the guidance of three project managers.

Managers

- Anthony Martinez
- Alma Piñeiro
- Jahsyel Rojas

Team 1 - Documentation & Requirements

- Joshua Dávila (Lead)
- Ojani Figueroa
- Giovanny García
- Juan Iranzo

Team 2 - Authentication & User Accounts

- Lorenzo Pérez (Lead)
- Jessy Andújar
- Gabriel Marrero
- Luis Marrero
- Ángel Villegas

Team 3 - Listings

- Kevin Gómez (Lead)
- Leanelys González
- Karina López
- Nicolás Rivera
- Jachikasielu Uwakweh

Team 4 - Search & Map Integration

- Jorge De León (Lead)
- Devlin Hahn

- Alejandro Marrero
- Kian Ramos
- Januel Torres

Team 5 - UI/UX & Branding

- Fabiola Torres (Lead)
- Yamilette Alemañy
- Daniella Melero
- Andrea Segarra
- Kenneth Sepúlveda

1.2.1 Current Situation

Landfills and textile waste in Puerto Rico

- High waste volume: Approximately 250 million pounds of clothing and textiles are sent to Puerto Rico's landfills annually.
- High landfill rate: Similar to the global and U.S. trends, a very high percentage of discarded textiles, around 85%, end up in landfills, despite being largely recyclable.
- Low recycling rates: Puerto Rico's overall recycling rate is notably low, with some reports estimating it to be less than 10%. This is significantly lower than the U.S. national average.
- Overwhelmed landfills: Puerto Rico's landfills are facing a serious crisis, with many already at or over capacity. The high volume of textile waste contributes to this problem.

Sources:

- investpr.org
- theenvironmentalblog.org

Poverty

- Overall Poverty: The poverty rate in Puerto Rico is alarmingly high, at around 41.7% as of 2022. This is over three times the U.S. national average.
- Child Poverty: A staggering 54.3% to 57.6% of children under 18 in Puerto Rico live in poverty. This is more than any U.S. state and indicates that a vast number of families cannot afford to consistently provide their children with properly fitting, weather-appropriate clothing and shoes.
- Persistent Poverty: All 78 municipalities in Puerto Rico are classified as "persistent poverty counties," meaning they have maintained a poverty rate of 20% or more for at least 30 years.

Sources:

- centropr.hunter.cuny.edu

- geopoliticaleconomy.com

Homelessness:

- Homeless Population: Recent counts show the homeless population in Puerto Rico to be around 2,096 individuals. For these individuals, clothing is a constant and critical need.

Sources:

- periodismoinvestigativo.com

1.2.2 Need

The purpose of this section is to establish the fundamental needs that motivate the Hand Me Down project, expressed independently of any system-to-be. These needs are grounded in the resale domain and must reflect the concerns of students and families who participate in secondhand exchanges. The articulation of these needs will guide the subsequent development of domain descriptions, requirements, software architecture, and testing activities.

Stakeholders in this domain shall be understood as students and families seeking opportunities for affordable, accessible, and trustworthy secondhand exchanges. Their needs are not for a platform itself, but for solutions to the problems they encounter when attempting to exchange goods in local communities.

The following distinct needs are identified:

- Students and families must have affordable access to secondhand goods that support daily life, education, and well-being.
- Stakeholders must be able to rely on transparent information about the condition and history of pre-owned items.
- Exchanges shall be conducted in a manner that establishes trust, fairness, and safety between participants.
- Opportunities for accessibility and inclusivity must be available so that all families and students, regardless of economic background, will participate in the resale domain without barriers.
- Developers shall have clear requirements, descriptions, and architecture to build upon, since no structured system currently exists to organize this resale context.

These needs form the foundation for further project work. They are deliberately expressed at the domain level, independently of any particular solution, to ensure that subsequent design and implementation activities will remain aligned with the stakeholders underlying motivations.

1.3.1 Scope & Span

Scope

The Hand Me Down project will operate in the broad domain of online resale marketplaces. It will

address the general problem of enabling individuals and communities to exchange secondhand goods in a structured, reliable, and sustainable manner. The scope will cover activities in domain engineering, requirements engineering, and software architecture to ensure a well-founded solution.

The project will emphasize the following areas:

- Domain:: Resale of pre-owned items across categories such as clothing and accesories.
- Requirements:: Identifying user needs related to affordability, sustainability, accessibility, and usability.
- Architecture:: Defining a framework that supports secure and scalable interactions between sellers and buyers.
- Project Activities:: Documentation, validation, and design processes that must accompany implementation.

Span

The span narrows the focus of the Hand Me Down project to **specific concerns and audiences** within the general resale domain. The platform must primarily serve individuals and families who wish to exchange items affordably, students and young adults seeking budget-friendly goods, and community members interested in sustainable consumption.

The span includes the following project-specific aspects:

- User Interaction:: Individuals must be able to list, browse, and search for secondhand items.
- Categorization:: Items will be organized into categories that facilitate discovery.
- Transaction Support:: The system must provide structured means for creating and viewing listings, including optional prices or donation markers. Negotiation and exchange are arranged outside the platform.
- Trust and Transparency:: Item conditions and relevant metadata must be clearly described to support informed decisions.

1.3.2 Synopsis

This Synopsis provides overview of the Hand Me Down project from the perspective of students and families engaged in secondhand exchanges. It articulates the domain, affordability, accessibility, trust, safety and states that stakeholders must be able to discover, evaluate, and exchange pre-owned goods with transparent information about item condition and history. The project shall be conducted through structured domain acquisition to produce a domain description, a requirements prescription that specifies goals, constraints, and quality attributes with traceability to stakeholder needs and a software architecture that evaluates alternatives and justifies decisions, prototyping where necessary to mitigate risk. Component design and iterative implementation will realize prioritized capabilities while preserving traceability. Verification and validation shall include a test plan that covers functional fitness, usability, and trust/safety concerns, supported by versioned documentation, change control, risk tracking, and metrics.

1.4 Other Activities than Just Developing Source Code

This project will not be limited to writing source code. To satisfy the needs identified for affordability, accessibility, trust/safety, and transparency in the Mayagüez/UPRM context, the team shall execute the following activities in addition to implementation. Each activity is mandatory, tied to a stakeholder need and justified by the current situation.

Domain engineering

The team shall elicit and model the domain of donation and resale of clothing and accessories for students and families (actors, workflows, vocabulary, constraints).

- **Need satisfied:** developers must share a precise vocabulary and mental model to preserve accessibility, affordability, and transparency.
- **Current situation:** rough sketches are based on team self-observation; no external interviews conducted yet; stakeholder roles are not enumerated; internal terminology seems consistent but remains unvalidated in the field.
- **Contributions to date:** condition labels and verification practices were researched; the **Sell** vs **Donate** category structure was standardized; initial personas were drafted to ground concepts.
- **Planned outcomes:** domain glossary, context diagram, concrete exchange workflows, and domain verification norms (tag photo, full-item photos, defect call-outs).

Requirements engineering

The team shall prescribe goals, user-level functional requirements (listing, browse/search, donation/resale flows, offer/negotiation) and quality attributes (trust/safety, transparency, usability, accessibility) with explicit traceability to needs and acceptance criteria.

- **Need satisfied:** developers must clearly understand system functionality and nonfunctional expectations that build trust and reduce effort.
- **Current situation:** no consolidated requirements baseline or acceptance criteria exist.
- **Contributions to date:** epics were linked to user stories (buyer/seller perspectives) against stakeholder needs; interface requirements at the system boundary were authored; measurable machine requirements (response time, uptime, user capacity) were drafted.
- **Planned outcomes:** requirements set with SHALL statements and acceptance criteria; traceability matrix (Need → Requirement → Test); nonfunctional thresholds made testable.

Software architecture

The team shall select and justify an architecture addressing security, privacy, modifiability, and campus-scale usage; architectural views and decision records will be maintained.

- **Need satisfied:** stakeholders must receive a reliable, maintainable basis for transparent, safe exchanges.
- **Current situation:** direction is leaning toward Supabase for authentication (Google sign-in compatibility) with a full custom backend; no UPRM hosting/privacy constraints identified; ADRs and C4 views are not yet written.
- **Contributions to date:** search and map integration approaches were evaluated and

geolocation/privacy considerations documented; authentication backends (Firebase vs. Supabase) and session-management implications were analyzed; page-level layouts were produced to inform view composition and navigation flows.

- **Planned outcomes:** C4 views (context/container), Architectural Decision Records (auth, data, map/search), quality-attribute scenarios, and targeted risk spikes where uncertainty is high.

Component design

The team shall define modules, interfaces, and data contracts to preserve testability and changeability (catalog/search, profiles/auth, exchange/offer, reporting/moderation, “items circulated” metrics).

- **Need satisfied:** maintainable, verifiable components are required to deliver transparency (clear item/condition data) and accessibility (predictable flows).
- **Current situation:** boundaries and interfaces are only partially documented.
- **Contributions to date:** the user/auth schema (roles, profile fields, donation/sell history) was initiated and APIs for login/registration/logout were defined; search behavior and map-related data interactions were documented; wireframes and page designs (Homepage, About, Clothes Listing, Individual Item, Favorites, Checkout, Log In/Sign Up, Profile) clarify interface responsibilities and data needs.
- **Planned outcomes:** module responsibilities, interface specs (inputs/outputs/preconditions/postconditions), initial schemas with migration notes, and example queries.

Implementation planning

The team will establish a delivery roadmap, Definition of Done, contribution standards, and a branching/CI strategy suitable for a student-run service.

- **Need satisfied:** predictable, reviewable progress must be ensured to realize stakeholder value without regressions.
- **Current situation:** a branch is created per team with controlled pull requests and merges to main when working; no CI pipeline is in place; review checklist/PR template usage is minimal.
- **Contributions to date:** the AsciiDoc documentation structure and conventions were established; documentation issues with acceptance criteria were created; a step-by-step Node.js/npm installation and verification guide was produced to bootstrap the development environment.
- **Planned outcomes:** roadmap with dates/owners, Definition of Done, CONTRIBUTING guidelines, PR template, and CI workflow for lint/tests on pull requests.

Testing and validation

The team shall produce a test plan spanning unit, integration, end-to-end, and usability/acceptance with students and families; trust/safety validations shall be included (e.g., prohibited items policy, condition/fit etiquette).

- **Need satisfied:** stakeholders must have confidence that behavior and quality attributes match the prescription.
- **Current situation:** no automated tests exist and the test plan is not started;

usability/acceptance testing is deferred to later milestones; recruitment will be informal ("ask around UPRM"); top priority requirements for acceptance scenarios are not selected yet.

- **Contributions to date:** machine requirements were expressed in measurable terms (e.g., support ~100 simultaneous users) to anchor performance testing; trust-building practices (verification methods, condition labels) were documented to translate into validation checks.
- **Planned outcomes:** test plan, seeded test suites, acceptance scenarios (Given–When–Then) mapped to requirements, trust/safety validations, and a defect taxonomy with triage protocol.

Deployment considerations

The team will define dev/staging environments, configuration, seed/reset data, rollback procedures, and a minimal operations runbook for a student-operated service.

- **Need satisfied:** availability and safe adoption are necessary for accessibility and affordability benefits to materialize.
- **Current situation:** no documented path to deploy, recover, or roll back; backend selection work is informing environment and secrets management but is not yet consolidated in docs.
- **Contributions to date:** Docker usage was explored to standardize developer environments and setup was documented; backend evaluations (Supabase/Firebase) inform environment and secret management decisions.
- **Planned outcomes:** environment definitions, secrets/config guidance, release/rollback steps, seed scripts, and a basic ops runbook.

Stakeholder liaison and feedback (cross-cutting)

The team shall schedule and document periodic touchpoints with students and families in Mayagüez to validate assumptions early (quotes/anecdotes shall be recorded in §2.1.1).

- **Need satisfied:** continuous alignment is required to keep requirements correct and trust high.
- **Current situation:** no formal liaison cadence is defined; external interviews are not planned at this time; consent/ethics approach is undefined.
- **Planned outcomes:** contact cadence, feedback and decision logs, and lightweight consent notes for any future interactions.

Documentation & governance (cross-cutting)

The team shall maintain versioned documentation, change control, risk tracking, and metrics to ensure durable traceability across activities.

- **Need satisfied:** traceability is required to justify decisions and onboard contributors without rework.
- **Current situation:** a docs index/navigation page is considered established (initial draft); changelog and risk register are not started; project metrics beyond "items circulated" are not yet defined.
- **Contributions to date:** the docs/ layout and AsciiDoc style were standardized and most documentation issues were created; branding (logo, color palette, typography) was

established to keep artifacts consistent and legible.

- **Planned outcomes:** docs index and navigation (maintained), changelog, risk register (e.g., technology choice risk, schedule slip, data/privacy misconfiguration), and basic metrics (e.g., items circulated as a primary signal).

1.5 Derived Goals

In addition to the primary goals (needs, scope, and core functionality), the project shall pursue secondary outcomes that respond to the realities of Mayagüez and the UPRM community. These outcomes are substantially different from core system behavior and emphasize long-term social, educational, and community benefits for students and families.

- Promote sustainability literacy and circular practices in Mayagüez:: The project shall normalize reuse, repair, and responsible disposal behaviors among students and families through donation and resale norms. **Broader impact:** item lifecycles will be extended and textile waste pressure will be reduced without prescribing any specific technical solution.
- Strengthen community engagement and mutual aid through UPRM-led outreach:: The project will cultivate equitable sharing practices (donation, fair resale) centered on UPRM as the primary touchpoint. **Broader impact:** social capital will increase and households will respond more effectively to clothing and accessory needs across semesters and seasons.
- Raise awareness of affordability and access constraints faced by local households:: The project shall make visible how structured sharing reduces acquisition cost and effort for students and families. **Broader impact:** schools and neighborhood groups will make more informed choices about drives, sizing priorities, and targeted outreach.

These derived goals will guide outreach, education, and validation activities alongside the primary objectives, and shall not be construed as mandates for any specific platform or implementation approach.

2.1 Domain Description

The domain of **hand-me-down clothing exchange** in Puerto Rico is shaped by social, environmental, and economic realities. It exists independently of any digital platform or technical system and can be understood through the people, practices, artifacts, and norms that sustain the circulation of clothing and accessories among students, families, and local communities.

At its core, the domain revolves around two primary actors:

- **Sellers**: Individuals or households who post garments they no longer need, offering them for reuse or resale.
- **Buyers**: Individuals who discover these garments through listings and arrange exchanges directly with the seller, outside the system.

The central entity in this domain is the **Piece**, any individual article of clothing or accessory that circulates between actors. Each Piece carries attributes such as:

- **Type**: The category of clothing.

- **Condition Rating**: A measure of quality or usability.

Pieces move between states through events:

- **Listing Published**: Occurs when a Seller makes a garment visible to potential Buyers.
- **Interest Expressed**: Occurs when a Buyer contacts a Seller regarding a listed garment.
- **Listing Closed**: Occurs when a garment has been exchanged offline or removed from visibility.
- **Discard Event**: Occurs when a Piece leaves circulation, feeding into the wider **Textile Waste Stream**.

The domain is sustained by informal practices and behaviors:

- **Discovery Flows**: The recurring sequence in which garments are listed, browsed, and handed over in person.
- **Condition Disclosure Norms**: The expectation that Sellers will show tag photos, highlight defects, and represent garments honestly.
- **Informal Price Bands**: Symbolic or suggested valuations (typically USD $8–$15), distinguishing resale from donation or commercial sale.
- **Dormant Stock**: Clothing stored in homes awaiting redistribution, resale, or disposal.
- **Meetup Spots**: Semi-public locations (e.g., campus benches, apartment lobbies) chosen for exchanges.
- **Ad-hoc Channels**: Informal digital venues (e.g., Facebook Marketplace, WhatsApp groups, Instagram stories).
- **Trust Cues**: Bilingual communication, recognizable names, or clear photos that influence whether an exchange proceeds.
- **Seasonal Demand Pulses**: Cyclical increases in demand, such as back-to-school or weather-driven surges for uniforms or outerwear.

These elements form an interconnected web of events, actions, and behaviors. Sellers and Buyers rely on ad-hoc digital channels for discovery, agree on terms through direct contact, and meet in semi-public spaces to exchange items offline. Exchanges are underpinned by implicit rules of trust and fairness, while also constrained by larger social and environmental forces such as poverty rates, limited recycling infrastructure, and overflowing landfills.

From a functional perspective, the domain can be represented through abstract operations:

- `publishListing(Piece, Seller, Locale) → ListingPublished`
- `expressInterest(Listing, Buyer) → InterestExpressed`
- `rate(Piece, ConditionRating) → ConditionRating`
- `review(Seller, Buyer, Review) → ReviewSubmitted`
- `categorize(Piece, Type) → Piece`
- `closeListing(Listing) → ListingClosed`

Each function captures an action that transforms the state of a **Listing** or **Piece** within the domain,

producing observable changes such as entering circulation, connecting interested parties, or leaving visibility once the exchange occurs offline.

In sum, the domain of secondhand clothing discovery in Puerto Rico is defined by the **visibility of Pieces**, the **actors who list and find them**, the **events that mark transitions**, and the **behaviors and norms** that make these exchanges trustworthy, affordable, and sustainable. This description provides a foundation for later requirements and design work, while remaining independent of any specific system or implementation.

2.1.1 Domain Rough Sketches

This section documents raw, observable examples of clothing exchange and discovery as they occur in daily life. These anecdotes are early insights and do not interpret or generalize; they simply record what was seen or described.

Example 1: Adriana's Search for Affordable Outfits

Adriana, a 20-year-old UPRM student, needs new outfits for an upcoming presentation but cannot afford retail prices. She opens a local Facebook group called "UPRM Ropa y Accesorios" and scrolls through the feed. She spots a post titled **"Blazer, lightly used, $10"** with photos showing the tag and sleeves. Adriana messages the seller in Spanish: **"¿Todavía lo tienes disponible?"** The seller responds quickly and confirms that the blazer is still available. They agree to meet at the main campus entrance the next afternoon. Adriana checks the garment in person, pays in cash, and the seller later marks the post as "Sold." The entire flow—scrolling, messaging, coordinating, and closing—happened outside any dedicated app, relying solely on informal trust and visible cues like clear photos and quick replies.

Example 2: Manuel's Donation After Semester End

Manuel, a middle school teacher and UPRM alumnus, sorts his closet at the end of the semester. He finds several shirts in good condition but no longer wears them. He takes photos, labels them **"Free for pickup near UPRM apartments"**, and posts them in a WhatsApp group where local students trade items. Within hours, a student replies: **"Can I pick up tomorrow morning?"** Manuel agrees, places the items in a bag labeled "Free clothes," and leaves them at his apartment lobby. By afternoon, the bag is gone. No money exchanged hands, and no platform intervention was needed — only quick, direct communication and mutual trust.

Observed Raw Facts

- Listings often use informal phrases like "lightly used," "like new," and "free."
- Meetups happen in semi-public, familiar places.
- Sellers use photos as proof of honesty (tags, condition, size).
- Prices are symbolic and flexible ($8–$15 typical range).
- Listings and messages are bilingual; Spanish often dominates the first contact.
- Exchanges rely on visibility and responsiveness rather than any formal guarantee.

These examples illustrate the recurring pattern of discovery → contact → agreement → exchange that underlies the entire secondhand clothing domain.

2.1.2 - Terminology

The following terminology consolidates entities, events, functions, and behaviors in the domain. Each entry specifies the type of concept it represents and the phase in which it is introduced (domain, requirements, design, implementation). This approach avoids circular definitions and ensures alignment with both domain knowledge and system concerns.

Term	Concept Type	Phase Introduced	Definition / Notes
Seller	Entity	Domain	A person who posts a clothing item on the platform to make it visible for reuse or resale.
Buyer	Entity	Domain	A person who browses listings and contacts sellers to express interest in an item.
Piece	Entity	Domain	An individual clothing item, defined independently of the system.
Listing	Representation	Design	A published representation of a Piece in the platform, visible to potential buyers.
Listing Published	Event	Domain	Instantaneous occurrence when a Seller makes a clothing item publicly visible.
Interest Expressed	Event	Domain	Instantaneous occurrence when a Buyer contacts a Seller about a listed item.
Listing Closed	Event	Domain	Occurrence when a Listing is manually closed by the Seller after the exchange is completed offline or the item is withdrawn.
Condition Rating	Attribute / Function	Domain	A measure (e.g., scale 1–10) of quality for a Piece.
Review	Artifact	Domain	Feedback associated with a completed offline exchange between Buyer and Seller.
Locale	Entity	Domain	Physical location where exchanges typically occur (e.g., campus area, public spot).
Type	Attribute	Domain	Category of clothing (dress, pants, shirt, etc.).
rate(Piece, ConditionRating) → ConditionRating	Function	Design	Updates the condition rating of a Piece using the new rating value; no pre-existing rating is required.

Term	Concept Type	Phase Introduced	Definition / Notes
publishListing(Piece, Seller, Locale) → ListingPublished	Event Function	Design	Function that triggers the publication of a new Listing by a Seller at a given Locale.
expressInterest(Listing, Buyer) → InterestExpressed	Event Function	Design	Function that triggers the event when a Buyer contacts a Seller about a Listing.
closeListing(Listing, Seller) → ListingClosed	Event Function	Design	Function that closes a Listing, indicating that the exchange has been completed offline or the item is no longer available.

2.1.3 Domain Terminology in Relation to Domain Rough

This section explains how several of the terms defined in **2.1.2 – Terminology** were derived through the analysis of the material captured in **2.1.1 – Domain Rough Sketch**. Its purpose is to make explicit the reasoning that transformed informal notes, phrases, and anecdotes into the concepts that organise our description of the hand-me-down clothing domain. It does not repeat the glossary presented in 2.1.2; instead, it narrates the steps that led from raw wording to stable vocabulary.

The rough sketch included an observation that students commonly **"look up an item, contact the seller, agree on price, size, and place, then meet to exchange the item."** This line was analysed as evidence of a recurring, recognisable structure in how people organise exchanges. Rather than a random set of moves, it revealed a social routine for arranging garment handoffs: identify an article, reach out to whoever offers it, discuss its details, and complete the handoff face-to-face. From this analysis emerged the term **Exchange Flow**, which designates the sequence of actions that shapes informal clothing listings and exchanges in the Mayagüez/UPRM setting.

Several fragments in the sketch referred to sellers showing **a photograph of the size tag and full images of the garment, including any flaws**. These snippets suggested an implicit rule: before an exchange, the seller is expected to disclose enough detail to reassure the buyer about size, hygiene, and hidden damage. By organising these remarks, we formulated the term **Condition Disclosure Norm**, a name for the informal convention that garments be represented honestly, with visual evidence, before a meeting occurs. This term encapsulates how participants try to manage uncertainty and build trust.

Price information appeared in multiple places. Notes such as **"symbolic prices between eight and fifteen dollars are common reference points"** hinted at an unwritten valuation practice. The figures were neither arbitrary nor fixed by any authority; they functioned as a social guideline for what counts as reasonable in low-cost exchanges among students. Processing this evidence

produced **Student Resale Price Band**, a phrase describing the monetary interval that frames conversations and distinguishes a low-cost resale from a donation.

Other material dealt with what happens when garments do not re-enter circulation. Mentions of **discarded clothes left in open areas or delivered directly to landfill, with only rare opportunities for recycling** highlighted a backdrop of disposal routes. These lines were interpreted as documenting the “exit channels” of clothing once it ceases to be worn. We named this phenomenon **Textile Waste Stream**, referring to the set of pathways — municipal collection, careless dumping, or limited recycling — through which clothing leaves everyday use. This term anchors the environmental dimension of the domain.

A different cluster of notes described **bags of outgrown clothing kept at home while owners decided whether to give them away or sell them**. Here the sketch captured a liminal state: garments were no longer needed, yet had not been reassigned. Analysing this condition produced **Dormant Stock**, a concept for clothing retained in domestic spaces after its initial life, awaiting a new role or final disposal.

The sketch also documented preferred locations for handing over items: students mentioned **campus benches, apartment lobbies, and other familiar public corners**. Rather than isolated remarks, these examples pointed to a shared concern for safety and practicality. We consolidated them under the term **Meetup Spot**, which denotes semi-public environments chosen because they balance accessibility, visibility, and comfort during an exchange.

Another strand involved the digital places where clothing is discovered. Notes cited **Facebook Marketplace, WhatsApp groups, and Instagram stories** as typical sources. Instead of treating each separately, we recognised a broader category — lightly moderated online venues where offers, requests, and quick negotiations happen. From this reasoning came the expression **Ad-hoc Channel**, describing the informal communication spaces that enable the visibility of available garments.

Scattered remarks highlighted how participants assess reliability. Seeing clear pictures, recognising a name, or receiving bilingual messages were all said to make people more comfortable proceeding with an exchange. Bringing these hints together led to the concept **Trust Cue**, a label for the small but influential signals that reduce perceived risk in peer-to-peer exchanges.

Finally, the rough sketch posed questions about changes in activity during **back-to-school periods, semester starts, or seasonal weather**. Even without full data, the presence of these queries suggested that demand for particular items is not constant. To represent this dynamic aspect we coined **Seasonal Demand Pulse**, a term for the predictable fluctuations in which garments are offered or sought as academic and climatic cycles progress.

By articulating these derivations, this section clarifies the analytical bridge between the exploratory material of the rough sketch and the structured vocabulary presented in the Terminology section. Understanding this path is essential for tracing how domain knowledge was built and for ensuring that later requirements remain anchored in the observed environment.

2.1.4 Domain Narrative

In Puerto Rico, informal clothing exchanges are a normal part of student and family life.

Individuals frequently use social media to find or offer clothing items and arrange in-person meetups. The domain captured by this project represents those existing behaviors, listing, discovery, communication, and offline exchange rather than creating new ones.

Sellers create posts or listings to share available garments, often including details such as size, condition, and photos. Buyers search or browse listings, then contact sellers through direct messages to express interest. Once both parties agree, they meet in person to complete the exchange. Afterward, sellers may close their listings, and buyers may leave a review describing their experience.

All exchanges happen directly between users, outside the platform. The platform's role is limited to:

- making listings visible
- enabling efficient search and filtering
- supporting communication through messaging.

The domain also includes secondary practices such as saving listings for later, checking reviews for trust, and adjusting a listing's visibility (active, reserved, closed). Together, these elements describe a clear, observable cycle of activity: **publish → discover → contact → exchange offline → close**. The system reflects this real-world rhythm without modeling payment, ownership, or physical handoffs.

2.1.5 Events, Actions, and Behaviors

Each domain concept below is explicitly annotated as **Entity**, **Action**, **Event**, or **Behavior**. Entities provide structure, Actions trigger Events, and Behaviors describe recurring patterns observed in exchanges. This annotation clarifies the "diverse notions" called out in feedback and keeps domain vs. system concerns separate.

Listing and Discovery

Entity

A **Listing** — **Entity** — represents an item made visible by a seller for reuse or resale. It includes attributes such as title, description, category, condition, and images.

Associated Actions

- **Action** — Publishing a listing with sufficient information (title, category, photos).
- **Action** — Editing details (price marker, condition, description).
- **Action** — Deactivating or closing a listing once the item is no longer available.

Events

- **Listing Published** — **Event** — occurs when a seller makes an item visible to the public.
- **Listing Updated** — **Event** — occurs when a listing's information has just been modified.
- **Listing Closed** — **Event** — occurs when the seller marks the listing as reserved or removed.

Behavior

Listing Lifecycle — **Behavior** — the continuous cycle in which sellers create, update, and close

listings to maintain an accurate catalog of available items.

Aggregate Definition

Role	Kind	Notes
Piece	Aggregate Root	Discoverable item (listing) and single entry point for changes.
ConditionRating	Value Object	Integer in `[1..10]`; updated via root method only.
Review	Child Entity	`{id, reviewer, rating (1□5), comment, createdAt}`; belongs to exactly one Piece. // reviewer (concept), not just an ID
Reservation	Value Object	**Conceptually associates a Buyer with a Piece**; storage may use an internal identifier (e.g., buyerId) but the domain concept is **Buyer**.
Images	Collection of Value Objects	≥ 1 image required when publishing (URI + metadata).
Status	Value Object	One of: `Active
Reserved	Closed`.	Seller
Reference (Primitive/VO)	Owner identity outside the aggregate boundary (storage key may be `sellerId`).	Locale

NOTE	**Domain Purity:** Read-model/presentation types such as `Page<PieceSummary>` belong to query/view layers. They must not be part of the **Piece** aggregate itself.

Invariants

Invariant	Enforced In	Temp. Violation?	Rationale
A Piece requires ≥ 1 image at publish time.	Factory / publish step	No	Visual trust for discovery.
`ConditionRating` □ `[1..10]`.	`rateCondition()`	No	Consistent quality scale.
A `Review` belongs to exactly one Piece.	`addReview()`	No	Prevents cross-linking reviews.
Status transitions follow `Active` □ `Reserved` → `Closed` (no reopen).	Root methods	Yes (within method only)	Coherent lifecycle; history integrity.
`reserveFor(buyer)` only when status = `Active`.	`reserveFor()`	No	Avoid double holds; associates with **Buyer** (concept), not just a raw ID.

Invariant	Enforced In	Temp. Violation?	Rationale
`close(reason)` only when status ∈ {`Active`,`Reserved`}.	`close()`	No	Forbids illegal re-closing.
Editing details is blocked when status = `Closed`.	Root guards	No	Preserve record of completed exchanges.

Root Operations (Conceptual Surface)

- `publish()` — (if drafts are modeled) make the Piece publicly visible.
- `editDetails(updates)` — update title/description/attributes (forbidden when `Closed`).
- `rateCondition(newRating)` — set/adjust condition in `[1..10]`.
- `reserveFor(buyer)` — `Active → Reserved` (hold while parties coordinate offline).
- `releaseReservation()` — `Reserved → Active`.
- `close(reason)` — `Active/Reserved → Closed` after offline handoff or withdrawal.
- `addReview(reviewer, rating, comment)` — attach feedback post-close.

Repository (Conceptual)

A repository operates at the **aggregate level** (reads may provide projections for lists): - `findById(pieceId) → Piece?` - `save(piece) → void` (persists the whole aggregate and its members) - Read-model queries **(projections)** for UI lists, e.g.: - `findActiveByFilters(filters) → Page<PieceSummary>` - `findBySeller(seller) → Page<PieceSummary>`

2.1.6 - Function Signatures

Objective

Define domain-level function signatures that describe how actions are carried out, including inputs, outputs, and possible changes in the domain.

Description

Function signatures specify how actors in the system interact through domain actions. They define the logical relationships between inputs, outputs, and resulting state changes:

- **The name of the function**: The action being performed.
- **The input parameters**: The information required by the action.
- **Output**: Type of data the function produces.
- **State changes**: How the action affects the domain.

The general format of a function signature is:

- FunctionName: Input1 >< Input2 >< ... → OutputType

- **Description** – what the function does.
- **Preconditions** – what must be true or available for the function to execute.
- **Postconditions** - what becomes true afterwards.
- **Notes** - clarifications about scope, limits, or non-covered cases.

Examples

publishListing : Piece >< Seller >< Locale → Either[ValidationError,ListingPublished]

Description

A Seller publishes a Piece in a given Locale, producing a ListingPublished event.

Preconditions

Seller and Piece exist and are active; Piece has required metadata (title, condition, images).

Postconditions

Listing becomes visible; `ListingPublished` recorded.

Notes

Discovery-only; no in-platform payments.

expressInterest : Listing >< Buyer → Either[ContactError, InterestExpressed]

Description

A Buyer signals interest in a Listing to start contact with the Seller.

Preconditions

Listing is open; Buyer is authenticated.

Postconditions

`InterestExpressed` links Buyer and Seller.

Notes

Returns `ContactError` if listing is closed/unavailable.

rate : Piece >< ConditionRating → Option[ConditionRating]

Description

Assigns or updates a Piece's condition rating.

Preconditions

Piece exists; rating within 1–5.

Postconditions

Piece's stored rating updated.

Notes

Returns `None` if Piece missing/inactive.

review : Seller >< Buyer >< Review → Either[ReviewError, ReviewSubmitted]

Description

Records a review after an offline exchange.

Preconditions

A prior interaction/closed listing exists between the parties.

Postconditions

`ReviewSubmitted` attached to reviewer/reviewee.

Notes

Returns `ReviewError` if no eligible exchange is found.

closeListing : Listing >< Seller → Either[ClosedError, ListingClosed]

Description

Seller closes an active Listing after exchange or withdrawal.

Preconditions

Seller owns the Listing; Listing is active.

Postconditions

Listing state becomes "closed"; 'ListingClosed' recorded.

Notes

Not callable by Buyers.

categorize : Piece >< Type → Option[Piece]

Description

Assigns a category type to a Piece.

Preconditions

Piece exists; Type is valid.

Postconditions

Piece's Type attribute updated.

Notes

Returns `None` if Piece missing/inactive. Does not change visibility.

discard : Piece → Option[Void]

Description

Removes a Piece from active circulation.

Preconditions

Piece exists and is active.

Postconditions

Piece marked as discarded/inactive.

Notes

Returns `None` if Piece missing/inactive.

Example Scenario: From Listing to Offline Exchange

1. Seller publishes a listing using `publishListing(Piece, Seller, Locale)` → produces `ListingPublished`.
2. Buyer browses available items and finds one of interest.
3. Buyer triggers `expressInterest(Listing, Buyer)` → produces `InterestExpressed`.
4. Both users coordinate offline to exchange the item.
5. Seller closes the listing via `closeListing(Listing, Seller)` → produces `ListingClosed`.
6. Buyer leaves a review using `review(Seller, Buyer, Review)` → produces `ReviewSubmitted`.

2.2.1 Epics, Features, and User Stories

Epics

Epics are a higher-level overview of goals that are large enough in scope that they can be broken down into multiple sprints. They provide direction and group related work together.

Buyer Epics

1. Listing Discovery
    a. As a buyer, I want to search for listings by category, filters, and keywords, so that I can quickly find clothing items that meet my needs.
2. Saved Listings
    a. As a buyer, I want to save or bookmark listings I am interested in, so that I can revisit them later when deciding whether to contact the seller.
3. Trust and Transparency
    a. As a buyer, I want to view detailed seller profiles and leave reviews after exchanges, so that I can make informed and confident decisions about future interactions.

Seller Epics

1. Listing Management
    a. As a seller, I want to create, edit, and close listings with detailed information, so that I can effectively manage the items I am offering for reuse or resale.
2. Interest Notifications
    a. As a seller, I want to receive notifications when someone shows interest in my item, so that I can promptly respond and arrange the exchange offline.

3. Seller Profile and Trust
    a. As a seller, I want to maintain a profile with personal and location details, so that I can establish credibility and attract buyers interested in my listings.

Features

Features specify the functionality required to deliver the goals described in the epics to the user. They serve to provide more concrete goals related to development.

Buyer Features

1. Listing Discovery
    a. Filtering (clothing type, size, color, condition, category, price/free marker, etc.).
    b. Keyword search.
    c. Sorting options (newest first, alphabetical, etc.).
2. Saved Listings
    a. Ability to bookmark or save listings for later consideration.
    b. Saved listings persist across sessions and devices.
3. Trust and Transparency
    a. Seller profile page (location, account age, bio).
    b. Seller ratings & reviews system.
    c. Buyer-to-seller review submission flow.
    d. Reporting mechanism for problematic listings or suspicious behavior.

Seller Features

1. Listing Management
    a. Create listing form (title, description, tags, price/free marker, category).
    b. Upload multiple images per listing.
    c. Edit listing details (update description, condition, images, price).
    d. Close or deactivate a listing once the exchange is complete offline.
2. Interest Notifications
    a. Push/email/in-app notifications when a buyer expresses interest in a listing.
    b. Notification center showing recent messages or contact attempts.
3. Seller Profile and Trust
    a. Editable seller profile (profile picture, name, location, short bio).
    b. Seller dashboard displaying active and closed listings, as well as reviews and ratings.

User Stories

User stories are derived from Features, breaking them down into smaller, individual tasks to be added to the backlog. These stories focus on user needs and help make development more user-focused.

Buyer User Stories

1. As a buyer, I want to browse listings by category to find a specific type of item I want.
2. As a buyer, I want to filter my search by size, condition, and category to tailor my search to my preferences.
3. As a buyer, I want to search for listings using keywords to find specific items that may not fit predefined categories.
4. As a buyer, I want to save interesting listings so I can revisit them later.
5. As a buyer, I want to view a seller's information such as location, reviews, and account details to feel confident before contacting them.
6. As a buyer, I want to leave reviews for sellers after an offline exchange so others can trust the process.

Seller User Stories

1. As a seller, I want to create listings for my items with options such as adding multiple pictures, a description, and a condition rating so interested buyers can find me easily.
2. As a seller, I want to edit my listings so I can update details whenever necessary.
3. As a seller, I want to close listings once the item is exchanged offline so the system stays accurate.
4. As a seller, I want to receive notifications when someone contacts me about a listing so I can respond quickly.
5. As a seller, I want to provide information on my profile such as my name and location to increase trust with buyers.

2.2.2 - Personas

A persona is a fictional yet plausible representation of the goals, needs, and expectations of any person within the project's domain. These don't have to just be direct users of the platform; anyone who benefits from the project's goals in one way or another can be a good persona. These personas serve to guide development in such a way where users are kept in mind when creating, developing, and refining features.

The following personas represent direct users of the platform:

Adriana Gómez

- **Age**: 20 years old
- **Occupation**: University student on financial aid working part-time on campus

- **Build / Appearance**: Dark and straight hair, low to average height, dresses in vintage clothes and kitschy accessories
- **Personality**: Creative, unique, community-driven

Adriana loves expressing herself through her fashion, yet she struggles finding clothes that feel like her. She has an affinity for older clothing, so thrift stores are a common spot for her. However, she struggles finding thrift stores in her area given that most of them don't have an online presence and, between her studies and her job, she doesn't have a lot of time to go out and look for them.

- **Goals:** Build a wardrobe that reflects her individuality without overspending
- **Pain Points**: Difficulty finding unique clothing items and accessories that don't break the bank, difficulty finding local thrift stores
- **Needs**: Access to trustworthy online listings, a way to find local thrift stores
- **Platform Interaction**: Browses listings from her phone during her breaks between classes
- **Keep Adriana in mind when:** Developing the general listings layout

Manuel Torres

- **Age**: 35 years old
- **Occupation**: Middle school teacher
- **Build / Appearance**: Brown and curly hair, average height, frequently seen in polo shirts or t-shirts with witty quotes on them.
- **Personality**: Patient, funny, good with kids, adventurous

Manuel is moving to a new location and wants a fresh start. Even if he doesn't want a lot of his belongings anymore, he'd rather not throw them away. He wants to sell whatever he has that's still in good condition to help out with the expenses involved in the move but can't find a way to do so.

- **Goals:** Simplify his moving process while giving his clothes a second life
- **Pain Points**: No low-hassle way to sell his belongings
- **Needs**: Access to a way to sell his belongings on his own terms
- **Platform Interaction**: Creates listings from his work computer for his pre-loved clothes
- **Keep Manuel in mind when:** Developing the listing creation interface and functionality

Daniela López

- **Age**: 27 years old
- **Occupation**: Nurse practitioner
- **Build / Appearance**: Long and wavy brunette hair, tall, dresses in comfortable clothes like baggy jeans or athleisure
- **Personality**: Outspoken, understanding, eco-conscious

Daniela is the type of person to own something until it falls apart. She is vehemently against fast fashion and tries to not contribute to it to the best of her ability. She believes older clothing was

made to last longer and buys second-hand whenever she can, but her favorite local thrift store has just closed down and the closest is now too far away for her to visit frequently.

- **Goals:** Achieve convenient access to good-quality clothing
- **Pain Points**: Doesn't want to contribute to fast fashion, tired of the low quality of modern clothing items
- **Needs**: A way to purchase good quality vintage clothing items
- **Platform Interaction**: Browses listings using search filters to narrow down her results to her needs
- **Keep Daniela in mind when:** Developing the search filters

As mentioned, it's not just direct users of the platform that benefit from it; other people within the domain can also benefit from the platform's goals. The following personas pertain to the project's domain:

Isidra Montoya

- **Age**: 61 years old
- **Occupation**: Thrift store owner
- **Build / Appearance**: White and straight hair in a pixie cut, short in stature, wears comfortable outfits like jeans and a t-shirt
- **Personality**: Friendly, welcoming, hard-working

Isidra runs a small thrift store that has been struggling to pull in customers as of late. Her store isn't located very close to the main highway and lacks an online presence due to Isidra's limited technical knowledge and reluctance to learn, making her rely mainly on word of mouth and the occasional curious passerby.

- **Pain Points**: Limited digital presence, lack of technical skills, poor reach
- **Needs**: An alternate way to expand her shop's reach to attract more customers
- **Platform Interaction**: Despite not interacting directly with the platform, a user uploaded her store's location to the thrift store map, expanding her store's reach and allowing her to gain the additional customers she needed.
- **Keep Isidra in mind when:** Developing the map feature and designing user friendly interfaces so intuitive that even those with poor technical knowledge can use them

Fernando Ruiz

- **Age**: 29 years old
- **Occupation**: Event organizer and nonprofit volunteer
- **Build / Appearance**: Short hair that has been dyed blue at the tips, medium build, tall, usually seen wearing light colors like whites and pastels
- **Personality**: Proactive, charismatic, enthusiastic

Fernando is very involved in volunteer work and helps host monthly clothing drives in his

community to help those in need. However, getting sufficient clothes for the drives has been a recurring issue, as they don't receive enough donations to help as many people as they'd like.

- **Pain Points**: Lack of clothing donations due to a lack of visibility
- **Needs**: More contributors/donors
- **Platform Interaction**: Despite not interacting directly with the platform, a user of the platform added the recurring clothing drive event to the map, increasing its visibility and contributing to getting more clothing donations for the drive.
- **Keep Fernando in mind when:** Developing the map feature

2.2.3 Domain Requirements

The system shall manage clothing listings according to the inherent properties of categories, items, and taxonomy structures observed in local resale and donation discovery platforms. It must allow users to browse, publish, and filter listings efficiently without facilitating in-platform payments or transactions.

- DR1: The system must classify every listing under exactly one primary category (e.g., tops, bottoms, shoes, accessories) to ensure consistent organization and discoverability across the taxonomy.
- DR2: The system must allow categories to have hierarchical subcategories, reflecting how real-world resale and reuse spaces organize clothing by type, gender, or size.
- DR3: The system must distinguish between **Resale** and **Free/Donation** listings, preserving the domain difference in motivation and visibility while keeping both accessible through a unified discovery interface.
- DR4: The system shall provide means to enforce item-to-category compatibility, ensuring that each listing aligns with an appropriate clothing category (e.g., shirts cannot be placed under footwear).
- DR5: The system must support category evolution, allowing the taxonomy to incorporate new categories or subcategories (e.g., uniforms, vintage) without invalidating existing listings.

Each of these requirements directly ties to observed domain properties: - **DR1** reflects the necessity of a single, authoritative classification per item for consistent search results. - **DR2** enforces hierarchical organization used in established discovery platforms like Facebook Marketplace or Craigslist. - **DR3** maintains the conceptual distinction between resale and giveaway behavior while allowing both to coexist. - **DR4** ensures semantic alignment between items and categories, avoiding misclassified listings. - **DR5** enables adaptability and scalability as the listing ecosystem evolves and new categories emerge.

Sources

- **Facebook Marketplace (Meta)**: https://www.facebook.com/marketplace/
- **OfferUp**: https://offerup.com/
- **Craigslist**: https://www.craigslist.org/about/sites
- **Goodwill (Donation)**: https://shopgoodwill.com/all-categories

2.2.4 Interface Requirements

Objective

Define how users interact with the system through visible UI elements, controls, states, and feedback. These **Interface Requirements** specify the behavior of forms, buttons, validation, navigation, and visibility rules. The platform facilitates listing and discovery but does not process payments or host transactions; all exchanges occur directly between buyers and sellers outside the system.

Interface Requirements

Create Listing Form - The authenticated user is implicitly the seller; **no seller field is displayed**. - Required fields: **Title**, **at least one Image**, and **Category**. Optional fields: **Price** (if absent, listing is marked **Free**), **Description**, **Condition**, and **Size**. - The **Publish** button is **disabled** until all required fields are valid. - On submit with missing or invalid fields, show **inline errors**; **focus moves** to the first invalid field. - On success, navigate to the **Listing Details** page and show a **toast**: “Listing published.”

Listing Details (Seller View) - Shows editable fields (Title, Description, Images, Price, Condition) while **Status** = **Active**. - Includes a **Status** dropdown with transitions: **Active** → **Reserved** → **Closed** (no skipping). - When **Status** = **Closed**, all edit controls are **disabled**, and a **‘Closed’ badge** is displayed. - Displays **Contact Requests** received for that listing, sorted by most recent.

Browse & Search (Buyer View) - Buyers can **search by keyword** and **filter by Category, Condition, Size, and Price/Free marker**. - Sorting options include **Newest first**, **Lowest price**, and **Condition rating**. - Each listing card shows **Title, Thumbnail, Price/Free marker, Condition, and Seller rating (if available)**. - Clicking a card opens **Listing Details**, where the **Contact Seller** button is visible if the viewer is not the owner.

Saved Listings (Buyer View) - Buyers can **Save/Unsave** a listing from both the card and details views. - Saved items persist across sessions and appear in the **Saved Listings** section under the user menu. - Saved icons visually change state (e.g., outlined vs. filled heart).

Contact Seller / Messaging - **Contact Seller** opens a message composer prefilled with the listing title. - On send, show confirmation (“Message sent”) and add it to **Message Threads**. - Seller receives a **notification badge** in the header or inbox. - Messages are displayed chronologically, grouped by listing.

Profile & Reviews - Sellers and buyers each have a **profile page** displaying their listings, ratings, and reviews. - Users can **edit their own profile** (profile picture, bio, location). - Reviews are **read-only** for the profile owner but can be submitted by others after an offline exchange. - The **average rating** and total reviews are displayed prominently at the top of the profile.

Accessibility & Feedback - All interactive elements are **keyboard-accessible** (Tab/Shift+Tab) with visible **focus outlines**. - Inline validation messages are **ARIA-live announced** to screen readers. - Toasts and banners are **non-blocking** and dismissible via keyboard. - Forms include descriptive labels and placeholder text for clarity.

Examples

1. **Create Listing**: A seller opens the **Create Listing** form. Title is empty and there are no images, so **Publish** is disabled. The user enters a title, selects a category, and uploads a photo. **Publish** becomes active. On submit, a success toast appears and the app redirects to the Listing Details page.
2. **Reserve Then Close**: A seller opens their Listing (Status = Active), selects **Reserved** from the dropdown. After completing the exchange offline, they select **Closed**, disabling all edit controls.
3. **Contact Seller**: A buyer opens a listing and clicks **Contact Seller**, writes a message, and sends it. A confirmation appears, and the seller's header shows a **1** notification badge.

Relation to Domain Requirements

- Domain Requirements define what must exist (e.g., Listing entity, User, Saved association).
- Interface Requirements define how users **create, view, modify, and interact** with those entities through UI components.
- Each interface rule directly supports a domain rule by providing a visible, testable interaction.

Justification

This section ensures that UI behaviors are intuitive, consistent, and verifiable. It separates internal representation (domain) from what users experience, providing a clear mapping between actions and visible states. These behaviors reinforce trust and usability while aligning with accessibility and real-world discovery practices.

Testing Plan

- Test each form and button for proper enable/disable states and validation.
- Confirm that "Publish" and "Close" actions trigger correct navigation and feedback messages.
- Verify that saved listings persist across sessions.
- Ensure that editing is disabled once a listing is closed.
- Check accessibility by tab-navigating forms and validating screen reader output.
- Confirm that notifications appear when a buyer sends a contact message.

2.2.5 Machine Requirements

Objective

The web server must support reliable and efficient operation of a React + JavaScript clothing discovery platform, enabling users to create, browse, filter, and save listings. Requirements are defined in measurable, testable terms to ensure responsiveness, reliability, and scalability. The platform does not handle payments, carts, or checkout operations; all exchanges between buyers and sellers occur offline after contact through the system.

Requirements

Performance - The system shall return **search results within 2 seconds** on average, with a maximum of **4 seconds** under peak load (defined as 150 concurrent active users). - The system shall support **200 simultaneous browsing users** without response time exceeding 4 seconds, including actions such as searching, filtering, saving listings, and viewing listing details. - Listing creation and updates (e.g., uploading images, editing descriptions, updating status) shall complete within **3 seconds** on average.

Reliability - The web server shall maintain an uptime of **99.7% per month**, allowing no more than **2.1 hours** of unscheduled downtime. - Core operations — creating, editing, saving, and closing listings — must be processed reliably, with **no more than 0.1% of requests** failing due to server errors. - Scheduled maintenance shall be limited to **3 hours per month**, announced with at least **48 hours of notice**.

Scalability - The system shall scale to support **500 concurrent users** performing mixed actions (browsing, searching, creating listings, saving favorites, sending messages) while maintaining average response times ≤ 3.5 seconds. - The system shall handle a database of up to **50,000 active listings** without significant degradation (response time increase ≤ 20% compared to baseline). - The system shall support both **vertical scaling** (increasing server resources) and **horizontal scaling** (adding more servers or instances) without major architectural changes.

Clarifications

- “Peak load” is defined as ≥ **150 concurrent active users** browsing or searching, with at least **10% performing listing creation or updates**.
- “Minimal outages” means ≤ **2.1 hours per month** of unplanned downtime.
- “Acceptable performance” means ≤ **3.5 seconds response time** for 95% of requests.
- No transaction throughput metrics are required, as **the system does not process financial operations**.

Areas for Refinement

- Stress tolerance for **extreme traffic spikes** (≥ 1000 users during semester openings or local donation drives) remains under evaluation.
- Optimization for **mobile devices under slow network conditions** needs further benchmarking.
- Additional testing required for concurrent **image upload** operations and cache efficiency under high demand.

Justification

The clothing discovery platform requires fast, consistent responses to maintain usability and trust. Defined thresholds ensure users can browse and post listings smoothly, even during busy periods such as the start of a semester or community events. These benchmarks align with standard web performance expectations for lightweight, image-based listing platforms similar to Facebook Marketplace or OfferUp.

Testing Plan

- Conduct **load tests** at 150, 250, and 500 concurrent users simulating real-world actions (searching, filtering, creating, editing, and saving listings).
- Perform **stress tests** with simulated traffic spikes to evaluate degradation and recovery patterns.
- Monitor uptime logs to confirm 99.7% reliability.
- Benchmark database performance against a dataset of **50,000+ listings** with image attachments.
- Measure latency on both desktop and mobile clients to validate acceptable response times across devices.

Load Definitions

Refined load definitions include: - **Peak Concurrent Users**: 500 users performing actions simultaneously. - **Average Concurrent Users**: 200 users during regular hours.

User Mix Scenarios

1. **Scenario 1: Browsing Listings**
    - 60% of users are browsing listings.
    - Actions: Filtering, searching, and viewing item details.
2. **Scenario 2: Creating Listings**
    - 20% of users are creating new listings.
    - Actions: Uploading photos, entering item details, and submitting listings.
3. **Scenario 3: Messaging**
    - 15% of users are messaging sellers or buyers.
    - Actions: Sending and receiving messages.
4. **Scenario 4: Administrative Actions**
    - 5% of users are performing administrative tasks.
    - Actions: Moderating listings, resolving disputes.

2.3 Implementation

Objective

Describe the overall architecture and design of the clothing repurposing application, showing how the React frontend, JavaScript logic, and Supabase database/storage/auth components work together. Include mockups and example code fragments to complement the explanation.

Description

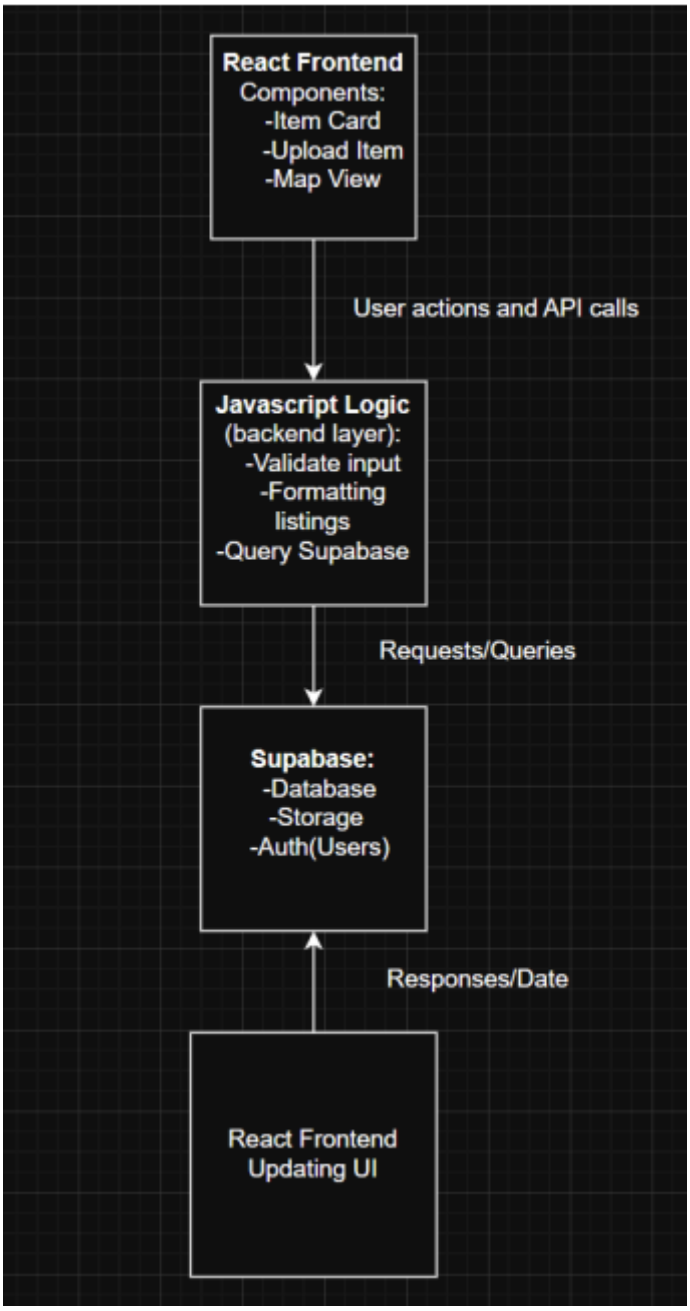
The application is structured as follows:

- Frontend (React)
  - Handles the user interface, including browsing clothing listings, saving favorites, uploading new listings, basic messaging/contact, and viewing nearby donation centers.
  - Planned pages/screens include:
    - Home Page: shows recent listings and quick navigation to other pages.
    - Clothing Listing Page: displays a scrollable list of available clothing items with filters and sorting options.
    - Upload Page: allows sellers to publish clothes as **listings** (with price or **free** marker); no checkout or payment is provided.
    - Saved Listings Page: lets buyers review items they bookmarked.
    - Map Page: integrates with Leaflet to display donation centers.
  - Styling is implemented with CSS modules or CSS-in-JS for maintainability.
- Backend Logic (JavaScript functions)
  - Manages business logic through modular JavaScript functions that act as an interface between the frontend and Supabase.
  - Responsibilities include:
    - Validating form inputs for listing creation and user registration.
    - Handling image uploads to Supabase Storage.
    - Authenticating users via Supabase Auth.
    - Querying and updating data in Supabase Database (listings, user profiles, reviews, saved listings).
    - Enforcing invariants such as mandatory seller ID and required listing fields.
    - Managing contact flows (e.g., expressing interest, message thread creation) without processing payments.
- Database & Storage (Supabase)
  - PostgreSQL database stores structured application data, including:
    - User profiles (name, email, role, reputation score).
    - Clothing listings (item ID, seller ID, title, description, category, size, condition, price/free marker, status).
    - Reviews and ratings for trust-building.
    - Saved listings (buyer listing associations).
    - Donation center metadata.
  - Supabase Storage manages image uploads and retrieval.
  - Supabase Auth handles authentication, role assignment, and secure access.
- External Services
  - Leaflet.js provides interactive map functionality for donation center locations.
  - Optional geocoding APIs convert addresses into map coordinates when new donation

centers are added.

Architecture Diagram

The architecture of the application is designed to separate concerns between the frontend, backend logic, and Supabase services. This structure ensures maintainability, scalability, and clarity for both developers and users.



1. Architecture Overview
    a. React Frontend
        - Provides the user interface where users can browse, search, and filter clothing listings, upload new items, save favorites, contact sellers, and view donation centers on a map.
        - User actions (e.g., creating a listing, saving an item, updating profile info, sending a message) trigger requests to backend functions.
    b. JavaScript Logic (Backend Logic)

- Bridges communication between the React frontend and Supabase services.
- Validates input (e.g., required fields for new listings).
- Formats data before insertion into the database.
- Handles error responses gracefully, ensuring the frontend receives useful feedback.

c. Supabase Services

- Database (PostgreSQL) stores listings, user profiles, reviews, saved associations, and donation center data.
- Cloud Storage securely manages clothing images, keeping them accessible for display in listings.
- Auth enforces authentication and authorization, ensuring data privacy and correct user roles.

d. Data Flow

- Requests flow downward from the React frontend through JavaScript logic into Supabase.
- Responses flow upward, updating React state dynamically so users see fresh listing data.
- The platform **does not** perform or mediate payments; messaging/contact supports offline coordination only.
- This separation ensures that the UI, application logic, and data persistence remain decoupled and maintainable.

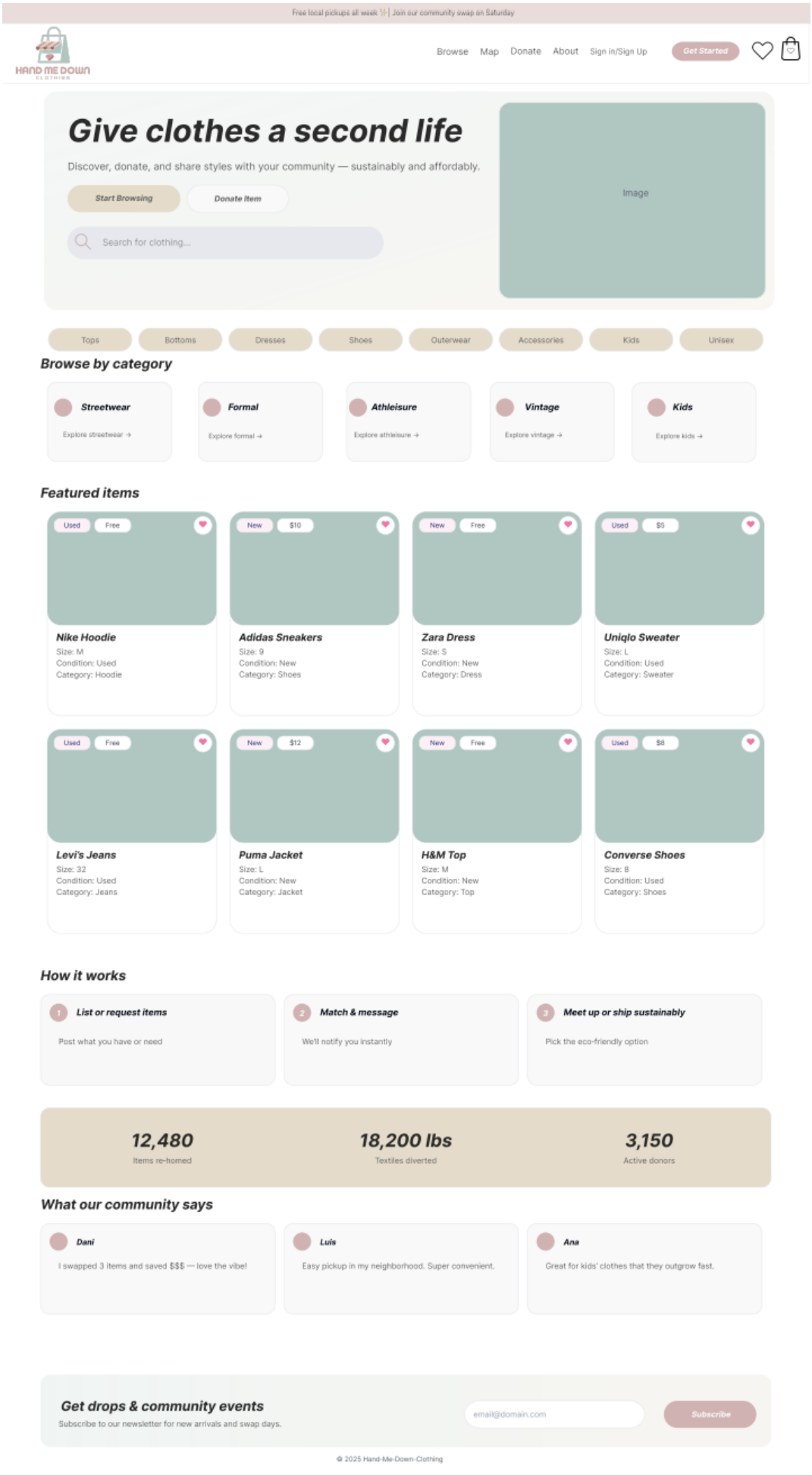
Layer Mapping (Clean Architecture / DDD Alignment)

- **Domain Layer**: Core concepts and rules (Listing, Piece, User, Review; invariants like required fields; aggregate boundaries).
- **Application Layer**: Use-case functions (publishListing, expressInterest, closeListing, saveListing), validation, orchestration of repository calls.
- **Interface/Adapters Layer**: React components/pages (forms, lists, profile, messaging UI), adapters to call application services, view models.
- **Frameworks/Drivers Layer**: Supabase (PostgreSQL, Storage, Auth), Leaflet, geocoding APIs, HTTP client libraries.

2.3.1 Selected Fragments of the Implementation

This section provides selected fragments of the implementation that complement the architecture described in Section 2.3. Instead of full code listings (not yet implemented), this section includes visual representations of the application's main pages and flows.

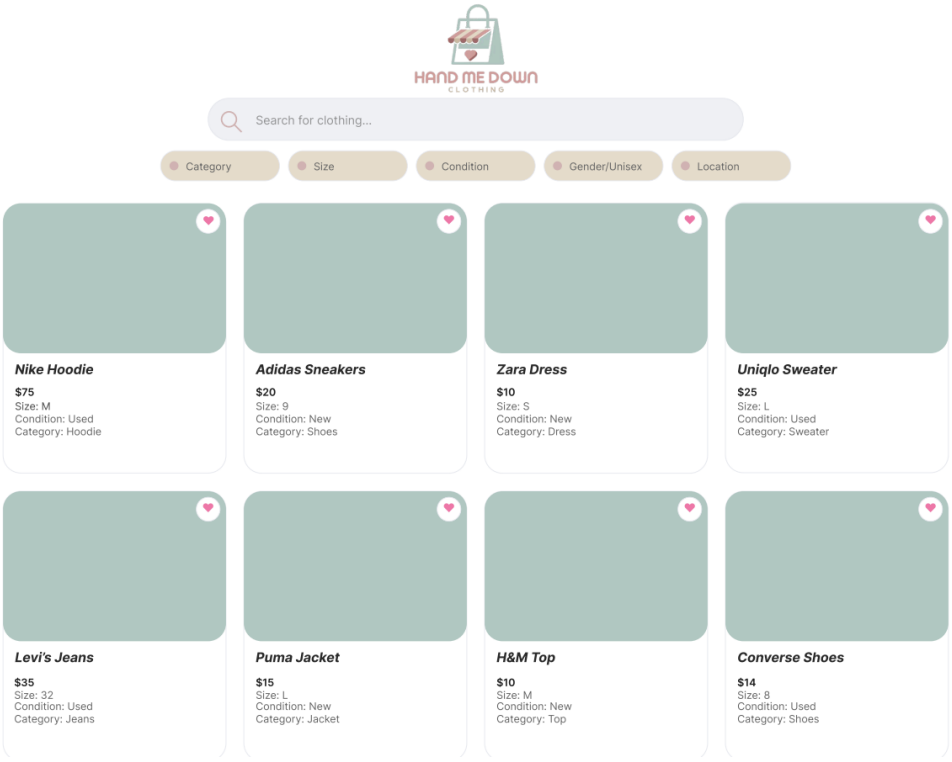
Home Page



The entry point of the application. Displays recent clothing listings and quick navigation options for

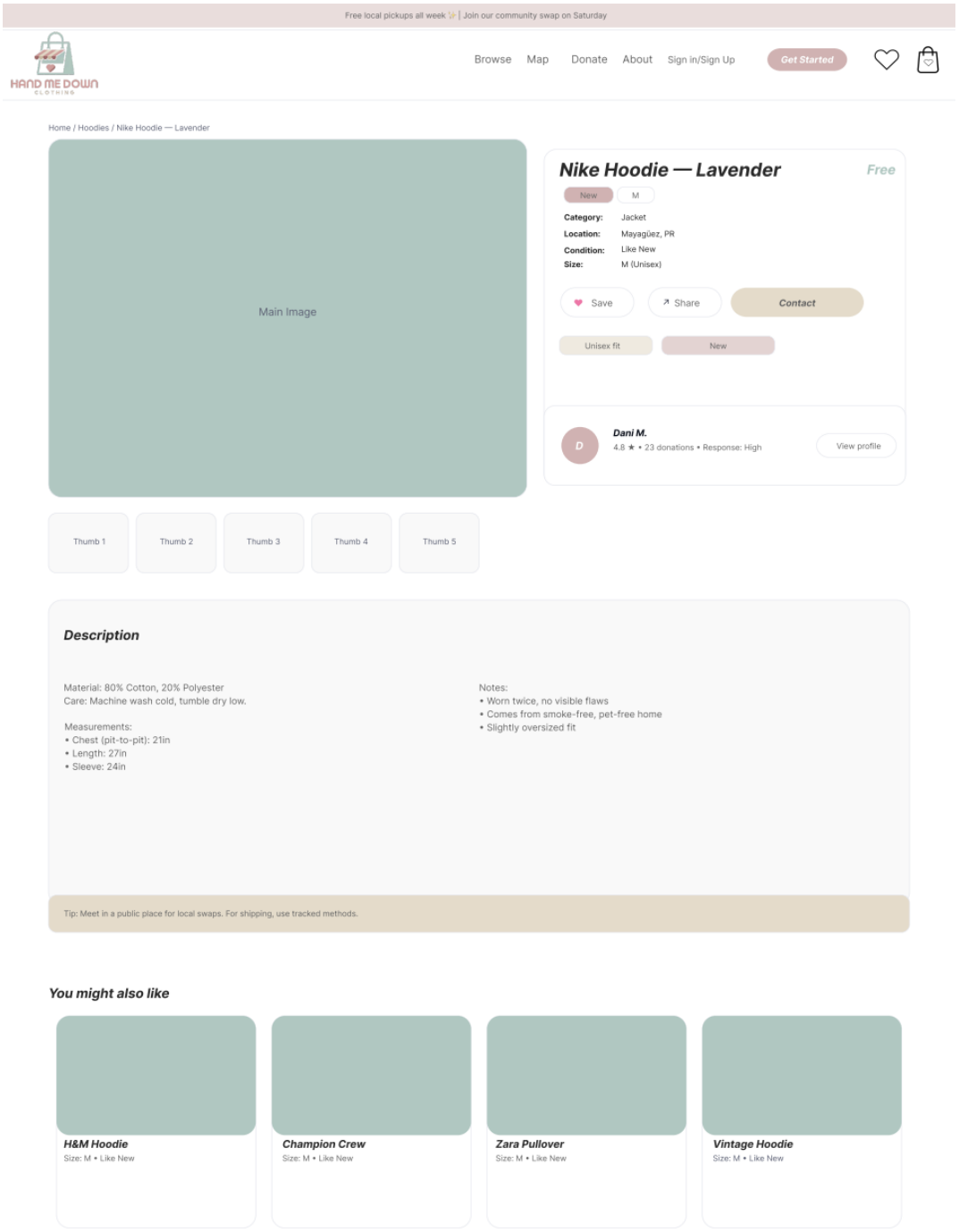
uploading new listings, browsing available items, viewing saved (liked) items, and accessing the map page for local donation centers. There is **no cart or checkout option**, since all exchanges occur directly between users outside the platform.

Clothing Listing Page



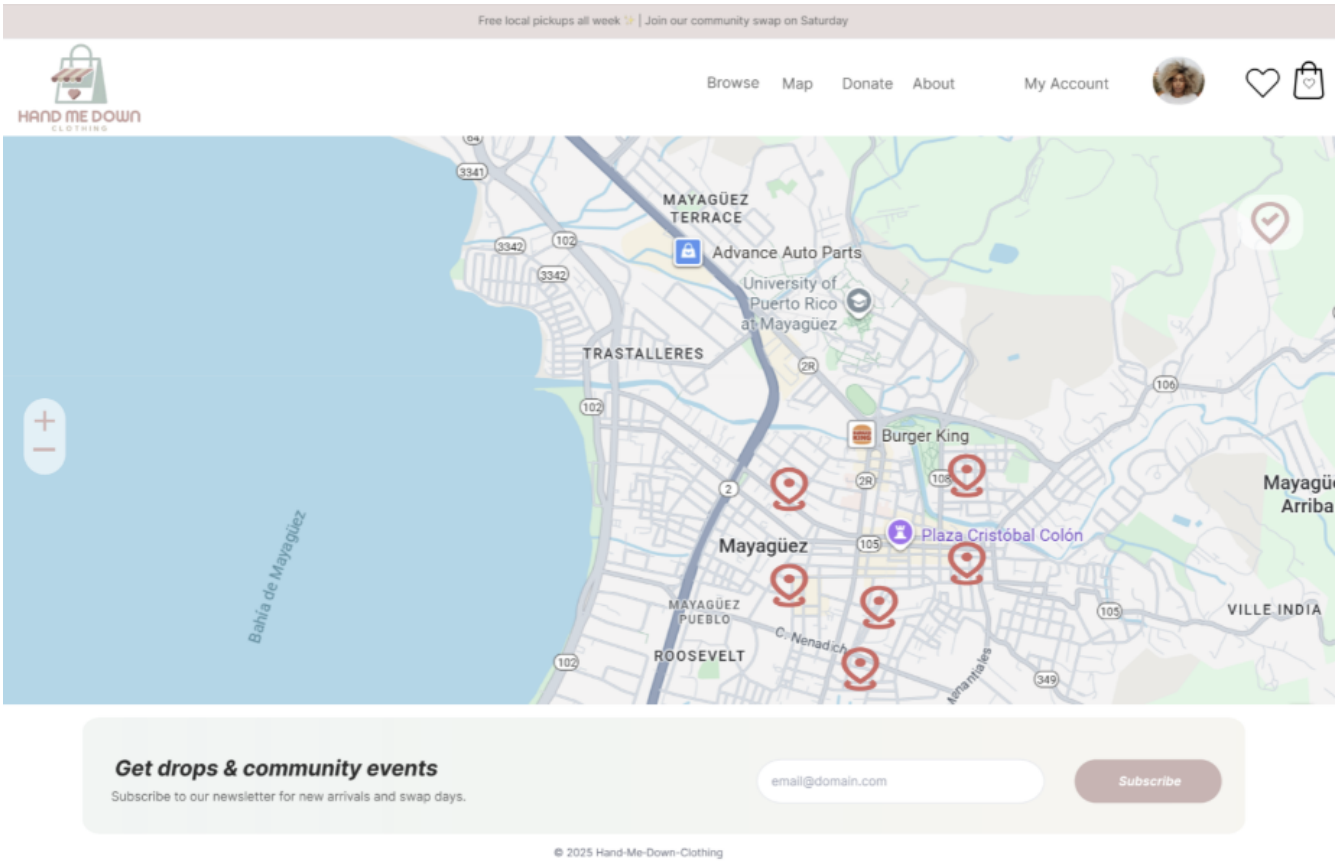
Shows a scrollable interface filled with clothing listings that include a photo, title, price (or “Free”), size, condition, and category. Each listing links to a detailed view where users can see more information or contact the seller. Search and filtering options are available by category, size, condition, location, and price/free marker. Users can also add items to a “Saved Listings” list to revisit later.

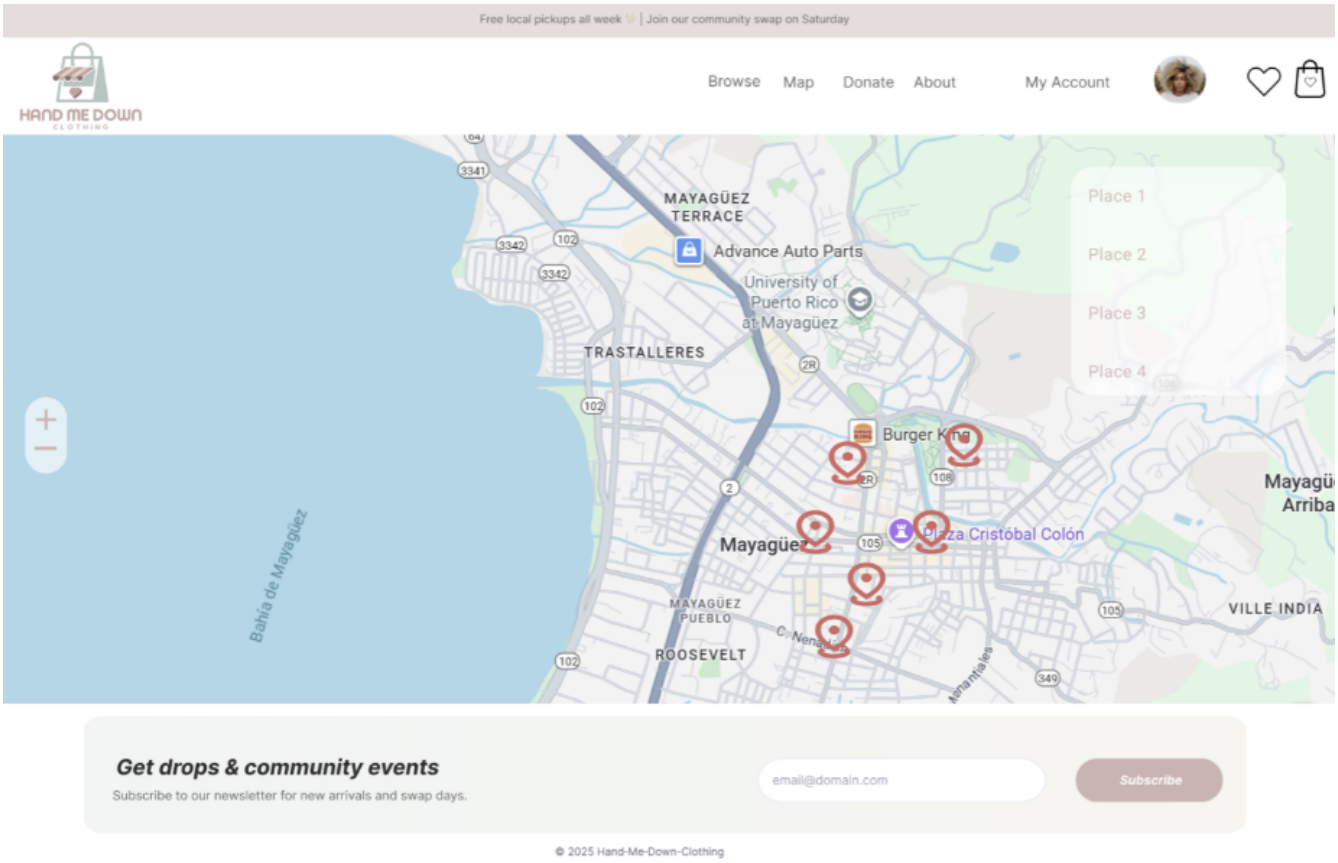
Item Detail Page

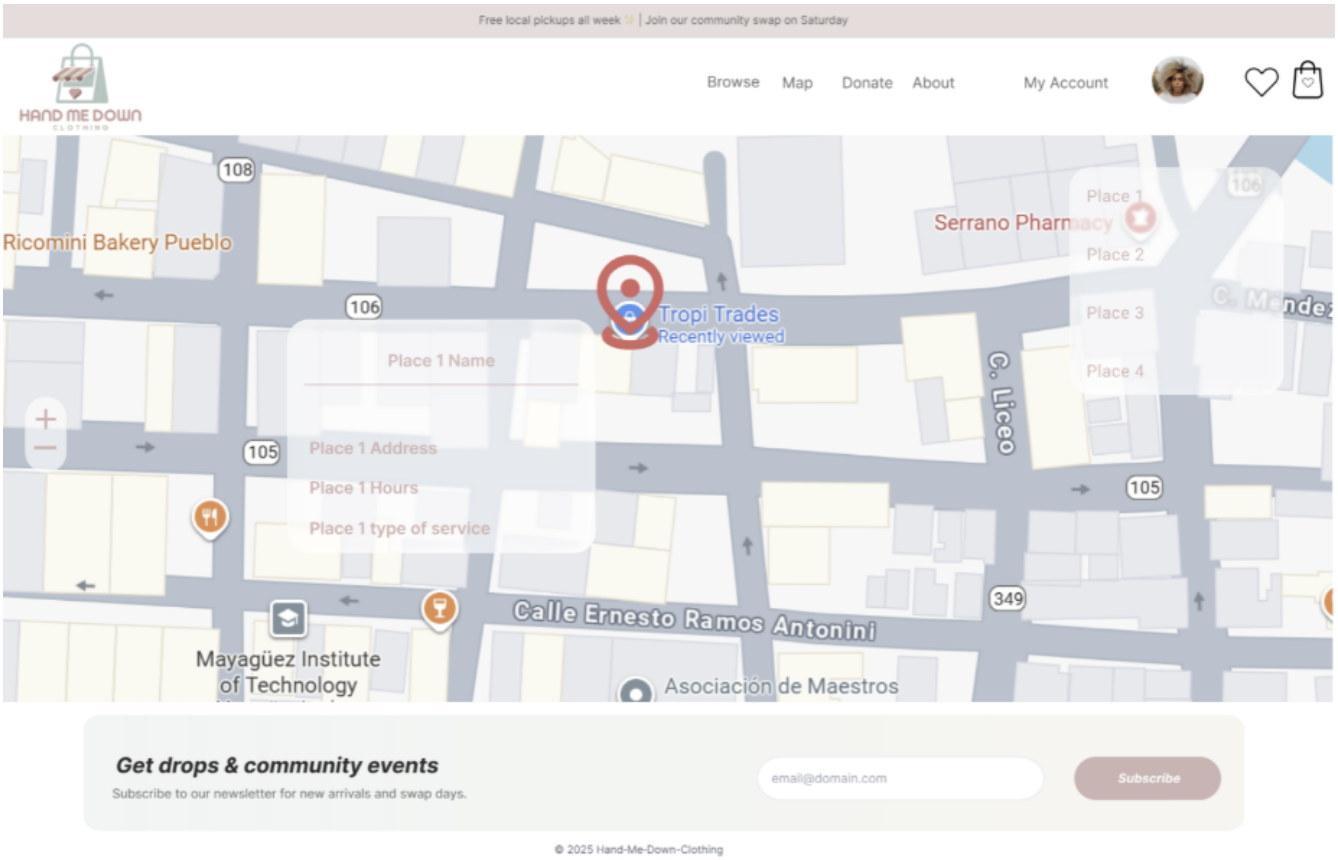


Provides detailed information about a specific clothing listing selected from the Listings Page. Displays images, descriptions, size, condition, price/free marker, and location. Includes buttons to **Save**, **Share**, or **Contact Seller**, initiating an in-app message or redirection to the seller's contact channel. Related listings are displayed at the bottom to encourage continued browsing.

Map Page


Free local pickups all week ☆ | Join our community swap on Saturday
HAND ME DOWN
CLOTHING
Browse
Map
Donate
About
My Account
MAYAGÜEZ TERRACE
Advance Auto Parts
University of Puerto Rico at Mayagüez
TRASTALLERES
Burger King
Plaza Cristóbal Colón
Mayagüez
MAYAGÜEZ PUEBLO
ROOSEVELT
VILLE INDIA
Bahía de Mayagüez
Get drops & community events
Subscribe to our newsletter for new arrivals and swap days.
email@domain.com
Subscribe
© 2025 Hand-Me-Down-Clothing

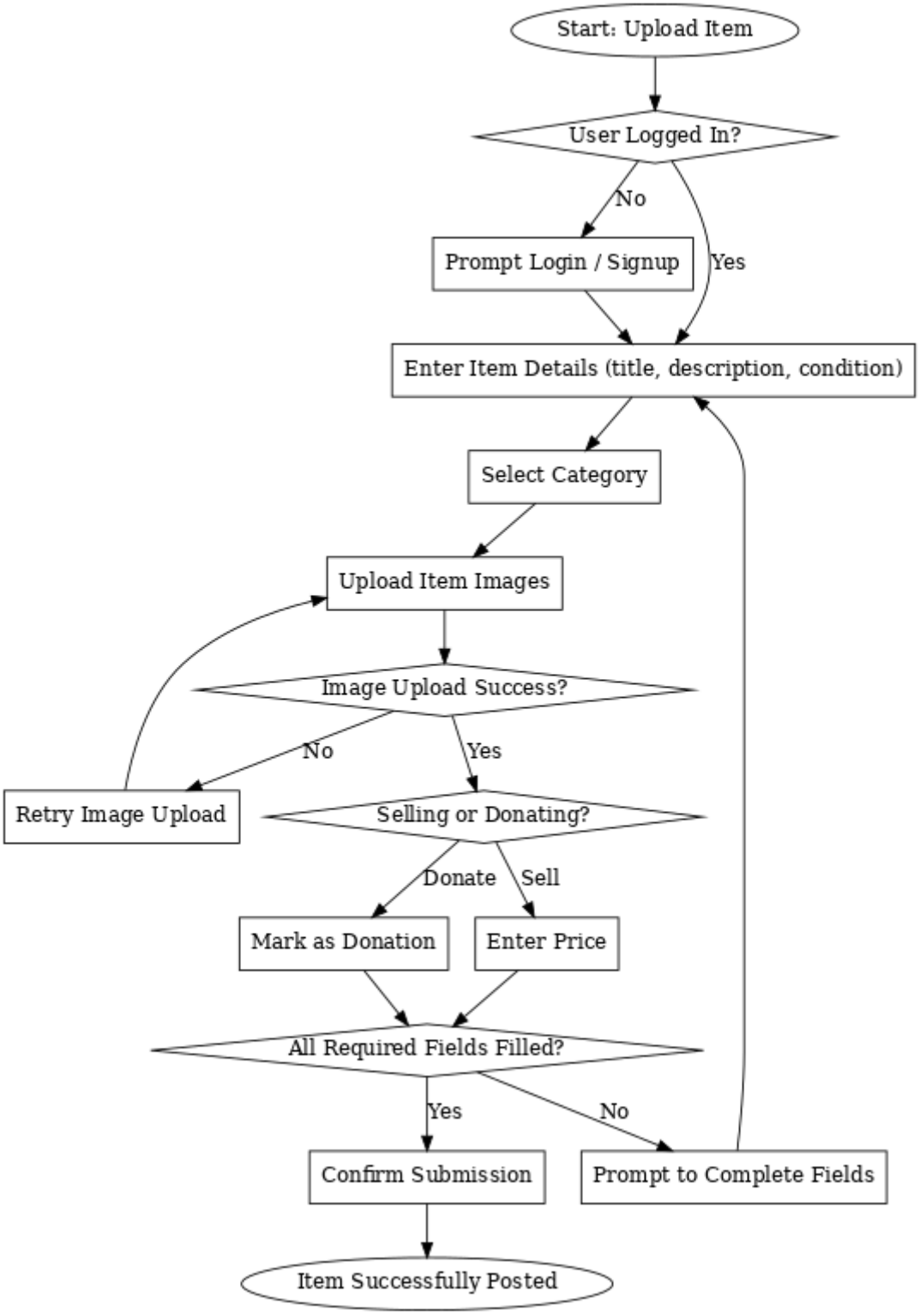

Free local pickups all week ☆ | Join our community swap on Saturday
HAND ME DOWN
CLOTHING
Browse
Map
Donate
About
My Account
Place 1
Place 2
Place 3
Place 4
MAYAGÜEZ TERRACE
Advance Auto Parts
University of Puerto Rico at Mayagüez
TRASTALLERES
Burger King
Plaza Cristóbal Colón
Mayagüez
MAYAGÜEZ PUEBLO
ROOSEVELT
VILLE INDIA
Bahía de Mayagüez
Get drops & community events
Subscribe to our newsletter for new arrivals and swap days.
email@domain.com
Subscribe
© 2025 Hand-Me-Down-Clothing



Displays nearby donation and repurposing centers using Leaflet. Locations stored in Supabase include name, address, hours, and type of service (e.g., “Donation Only,” “Donation-for-Credit”). The map helps users identify offline drop-off or pickup options near their area.

These images complement the architecture by showing how the frontend components (React pages) map to backend logic (JavaScript functions) and Supabase services (Database, Storage, Auth).

Item Upload Flowchart



Illustrates the process for creating a new listing: 1. User logs in or registers. 2. Opens the **Create Listing** form. 3. Fills in title, category, description, condition, and optional price/free marker. 4. Uploads one or more images. 5. Submits the listing. 6. Listing is stored in Supabase with its associated seller ID and displayed publicly in the marketplace.

The upload process enforces required fields but **does not create or manage any transaction**; once published, all further exchanges between users occur externally.

3.1 Concept Formation and Analysis

This section documents how the concepts identified in **2.1.2 – Terminology** emerged from the raw observations and examples in **2.1.1 – Domain Rough Sketch**. It clarifies how individual, context-specific details were abstracted into general domain entities, events, functions, and behaviors. The goal is to make the reasoning behind each abstraction traceable and verifiable.

From Observation to Abstraction

The first stories in the rough sketch (Adriana's search and Manuel's donation) revealed a repeating structure of discovery, communication, and exchange. By comparing these scenarios side-by-side, the team recognized several recurring phenomena:

- Listings (Facebook posts, WhatsApp messages) act as public representations of clothing items.
- Buyers and sellers interact through short, informal text exchanges that express **interest** and confirm details.
- The actual hand-off happens offline, after mutual agreement.
- Trust is maintained through norms such as showing garment tags, using bilingual communication, and meeting in visible places.

These repeated elements were abstracted into domain concepts — **Listing**, **Buyer**, **Seller**, **Interest Expressed**, **Condition Disclosure Norm**, and **Meetup Practice**. Each concept reflects a consistent pattern observed in multiple exchanges, rather than a one-off anecdote.

Entity Consolidation

Multiple items described in the examples (e.g., "blazer," "shirts") suggested a single core entity: **Piece**, representing any individual clothing item in circulation. The posts and messages that made those items visible became the entity **Listing**, distinct from the physical item itself. This separation mirrors how the domain actually functions: a Listing can exist, be updated, or be removed independently of the Piece it describes.

Event and Function Identification

Actions observed in the examples were distilled into domain functions:

- `publishListing(Piece, Seller, Locale)` — a seller's act of making an item visible.
- `closeListing(Listing, Seller)` — a seller's act of marking a listing inactive after the offline exchange.

Each function captures a discrete transition in the domain, corresponding to what participants **do** rather than what the platform might later automate.

Behavioral Norms and Implicit Concepts

Trust-related practices emerged as implicit but crucial aspects of the domain. For instance, showing tags or defects evolved into the **Condition Disclosure Norm**, while bilingual messaging and

recognizable names became **Trust Cues**. By identifying these implicit elements explicitly, the terminology provides a shared vocabulary that aligns both domain understanding and future requirements.

Iterative Refinement

During group analysis sessions, terms were revisited to ensure they described phenomena observable without assuming the existence of a system. This distinction kept the domain pure — describing only what happens in the world — while preparing a foundation for requirements in Section 2.2.

In conclusion, the concept analysis bridges the gap between individual real-world stories and the structured domain model. It traces how recurring actions, roles, and norms were abstracted into formal concepts that will guide subsequent design and implementation work.

3.2 Validation and Verification

Validation and verification ensure that the Hand Me Down platform meets stakeholder needs, maintains internal consistency across modules, and satisfies measurable quality standards defined in the requirements. These activities are carried out continuously and collaboratively across all sub-teams—Documentation & Requirements, Authentication, Listings, Map/Search, and UI/UX— to preserve traceability from stakeholder needs through implementation and testing.

Validation

Domain and Requirements Validation

Validation of terminology and requirements was performed iteratively through internal documentation reviews. Each update to §§ 2.1 – 2.3 was examined to confirm that domain concepts such as **Piece**, **Listing**, **Condition Disclosure Norm**, and **Listing Closed** remained coherent and aligned with stakeholder needs (§ 1.2.2). All requirements now employ definitive “shall/must” statements instead of uncertain language (“may,” “aims”) to ensure they are directly testable. No external stakeholder validation sessions have yet occurred; these will take place in later milestones to confirm usability and trust cues with students and families.

Scenario Walkthroughs

Walkthroughs were conducted for the **publishing** process, tracing data flow from form submission to Supabase storage. No inconsistencies were found. Editing, closing, and saving workflows are planned for final-phase validation once their implementations are complete.

Category and Condition Refinement

Internal validation led to refinement of the category taxonomy and condition-rating scales, resolving ambiguities from the initial model. Terminology was standardized to **Donated Piece** and **Sold Piece** to ensure semantic consistency across documentation and database layers.

Search and Map Validation

The Map & Search module was validated by cross-checking UI results against the actual data stored in Supabase. Each search query correctly displayed only listings matching its attributes, and filter walkthroughs confirmed accurate behavior for “Tops,” “Bottoms,” and “All.” Usability

was validated through manual inspection: map markers are accurately pinned, display complete popup information (name, address, hours), and maintain expected cluster and static behaviors. Search relevance and marker accuracy confirm correct data binding between UI and repository functions.

Authentication Validation

Supabase Auth integration was validated by confirming that users can **sign up, log in, and log out** successfully without confusion. The authentication flow follows Supabase and OWASP recommendations for secure web login. Manual tests were executed in Google Chrome, Firefox, and Brave on desktop devices. Error messages were displayed properly when login, signup, or logout failed. Mobile testing is scheduled for the next milestone.

UI/UX Validation

Informal validation was performed through manual walkthroughs and internal demos. Team members and classmates interacted with the interface to identify confusing or redundant elements and provided feedback on button labeling and visibility of listing-creation steps. Adjustments were applied iteratively, improving label clarity and layout alignment. Accessibility was validated manually through color-contrast and tab-navigation checks. Primary text and buttons met readability standards, and form elements followed a logical tab order. Semantic HTML was verified for buttons and labels, while full ARIA labeling and automated accessibility audits are planned for the next milestone.

Planned Stakeholder Validation

A short validation session with student volunteers will be scheduled before the final milestone to evaluate clarity of interface terminology, filter usability, and trust indicators such as condition labels and safety guidance.

Verification

Traceability and Acceptance Criteria

A preliminary **Need** → **Requirement** → **Test** mapping exists conceptually and will be formalized in `/docs/tests/traceability.adoc` for milestone 3. Each requirement includes measurable acceptance conditions: – Interface Requirements define button-state validation, inline error messaging, and toast feedback. – Machine Requirements specify response-time thresholds (≤ 2 s average; ≤ 4 s peak) and uptime ≥ 99.7 %. – Domain Requirements link directly to test cases verifying classification and visibility logic.

Unit Testing

Manual unit tests were completed for **publishListing()**, validating data verification, database insertion, and frontend feedback. All manual tests passed successfully. Additional automated unit tests for **closeListing()** and **editListing()** are scheduled for the next milestone. Authentication unit tests validated correct sign-up, login, and logout behavior, while error handling displayed expected feedback messages when failures occurred.

Integration Testing

End-to-end tests confirmed correct UI-to-database behavior: information entered in the listing form propagates through backend validation and is stored in Supabase as expected. Integration with Authentication (user ID linkage) will be completed in the final milestone. The Search

module's repository functions (`PieceRepository.getPieces()` and `filterPieces()`) were validated indirectly through accurate data synchronization between Supabase queries and the rendered results. Authentication privacy constraints were verified through **role-based access control** using Supabase **Row Level Security (RLS)** policies implemented in issue #301. Interface behaviors—such as disabled Publish buttons until form validation passes, visible toast messages, and navigation flow correctness—were manually verified against the Interface Requirements. Visual consistency was confirmed across browsers.

Load and Performance Testing

Preliminary manual observations show average search responses in ≈ **1 second** and listing creation times under **0.5 seconds**—both within the defined machine-requirement limits. Formal automated load testing using **k6** will be added to simulate concurrent usage (150 – 500 users) and confirm scalability benchmarks. Authentication latency and API response times will also be measured in the next milestone.

Data Validation and Security Checks

Map-coordinate rendering logic filters out invalid or non-finite latitude/longitude values, preventing off-map markers. All markers are non-draggable, ensuring location data remains immutable in the UI. RLS policies in Supabase protect user records by restricting read/write access based on authentication state and role. UI components were visually validated across major browsers (Chrome, Edge) to ensure consistent layout, iconography, and branding defined in the global style guide.

Continuous Verification

A GitHub Actions workflow will execute linting and unit-test jobs on pull requests to maintain consistent quality and prevent regressions once automated tests are in place.

Outcomes

– Documentation, domain model, and requirements were aligned and validated through internal review cycles. – Listing-publication backend passed all manual unit tests, achieving < 0.5 s creation time. – Search and map functionalities were validated against Supabase data, loading results in ≈ 1 s on average. – Map markers were verified for nine sample donation centers. – Authentication features (sign up, login, logout) were validated across major desktop browsers with secure RLS policies. – UI elements and flows passed internal usability and accessibility checks; no critical issues were reported. – Category and condition-rating systems were refined for accuracy and uniformity. – Traceability structure and automated CI testing are established for completion in the final milestone.

Together, these validation and verification activities confirm that the system concepts are sound, the current implementation behaves as specified, and measurable criteria are in place to ensure the platform remains reliable, scalable, and aligned with stakeholder expectations as development continues.

Logbook

Person	Sections worked on
1uismar33r0	2.3.1 (#148)
Alma-pineiro	2.3.1 (#148)
JoshDG03	1.3.1 (#190), 2.1 (#190, #214), 2.1.1 (#190, #214), 2.1.2 (#214), 2.1.3 (#214), 2.1.4 (#214, #216), 2.1.5 (#182, #190, #214, #217), 2.1.6 (#190, #214), 2.2.1 (#190, #214), 2.2.2 (#214), 2.2.3 (#214), 2.2.4 (#190, #214), 2.2.5 (#190, #214, #242), 2.3 (#190, #214), 2.3.1 (#214), 3.2 (#242)
JuanIranzo	2.1.6 (#218)
Ojani	2.3 (#247), 2.3.1 (#247), 3.2 (#246)
angelvillegas1	2.2.5 (#242)