High-Level Design - Eco Closet

1. Introduction

1.1 Purpose

EcoCloset is a simple-to-use app designed to make buying and selling clothes easy, fast, and enjoyable. We connect people within their local communities, promoting sustainability and affordability in fashion.

1.2 Scope

In Scope:

- **Concept:** EcoCloset is about sustainable fashion that's quick and convenient for users.
- **Features:** Easy listings, personalized suggestions, and seamless local transactions.
- **Technology:** Built with Flutter (mobile app) and Firebase (backend), ensuring smooth performance.
- **User Journey:** Designed for effortless use, from signing up to buying or selling clothes.

1.3 References

1. **High-Level-Design-HLD-Document-Template.docx**
2. **Product Concept_ EcoCloset.docx**
3. **MVP & Project Plan.docx**
4. **Pages description.docx**
5. **Flutter Documentation** (docs.flutter.dev)
6. **Firebase Documentation** (firebase.google.com/docs)

2. Concept & Product Overview

2.1 Vision and Goals

EcoCloset was designed to revolutionize the second-hand clothing industry by:

- • **Promoting Sustainability:** Extending clothing lifecycles and significantly reducing fashion waste.
- • **Empowering Connections:** Facilitating fast, effortless transactions within local communities.
- • **Encouraging Individuality:** Making unique fashion accessible for every user.
- • **Ensuring Affordability:** Making expensive fashion affordable and available to everyone.

2.2 Target Audience

- **Primary Users**: Young, eco-conscious urban individuals (ages 18–35) interested in sustainable fashion, affordability, and convenience.
- **Secondary Users**: Fashion-forward shoppers looking for unique and budget-friendly pieces that align with their values of individuality and sustainability.

2.3 Unique Selling Points

1. **Localized Community Buying**: Quick local transactions significantly reduce buying times and create easy exchanges.
2. **Personalized Discovery**: Advanced “For Me” filtering options to present highly relevant and personalized clothing suggestions tailored to users' preferences and proximity.
3. **AI-Assisted QuickUpload**: Innovative AI-powered item listing technology auto-generates tags and descriptions, simplifying and accelerating the selling process.
4. **Intelligent Personalization**: Integrated AI technology learns user preferences, understands individual styles, and dynamically personalizes the explore page.
5. **Smart Notifications:** Sending timely push notifications and tailored promotions, keeping users engaged with relevant new listings and exclusive deals.

2.4 Market & Competitive Landscape

While existing second-hand platforms like Depop, Vinted, or Facebook Marketplace offer broad marketplace capabilities, they frequently fail to offer convenient, localized transaction experiences. EcoCloset distinctly addresses this gap by providing an engaging, localized, and highly personalized shopping experience, simplifying buying and selling within local communities.

3. System Overview

3.1 System Description

From a top-level perspective, EcoCloset is an advanced mobile marketplace designed for intuitive and engaging user experiences, integrating:

- **Flutter Application**: Cross-platform interface for Android and iOS, ensuring seamless, responsive interactions.
- **Firebase Suite**: Comprehensive backend infrastructure offering user authentication, real-time database management, robust analytics, secure data storage, and rapid scalability.
- **AI Tagging & Personalization Services**: Utilizing Gemini AI for intelligent item tagging, personalized content recommendations, and proactive user engagement.

3.2 Key Functional Areas

User Accounts & Profiles

- Simple, secure registration and login using email or phone number.
- Highly customizable user profiles allowing users to define style preferences, sizes, favorite brands, and more.
- User dashboard with activity tracking for transactions, listings, and preferences.
- Integrated social sharing for easy connection with friends and community members.

Catalog & AI-Assisted Item Listings

- Fast, intuitive item uploading with automatic AI-generated tags, descriptions, and optimized titles using advanced Gemini AI.
- Support for uploading multiple high-quality images with future capability for interactive videos and AR virtual try-ons.
- Specialized catalog sections such as “Trending,” “Recommended for You,” and curated seasonal collections.

Advanced Search & Personalized Filtering

- Real-time search capabilities powered by Firebase’s Cloud Firestore for instant, relevant item discovery.
- Comprehensive filtering options allowing users to search precisely by style, size, brand, price range, condition, seller ratings, and proximity.
- “For Me” filter providing highly personalized suggestions based on user preferences, past behavior, and location.
- Additional search enhancements, including voice and visual search features powered by AI.

Local Transactions & Secure Payments

- Clear display of approximate seller locations to encourage convenient local buying and selling.
- Multiple secure payment methods integrated directly into the platform, including local and global solutions such as Bit, Paybox, Apple Pay, Google Pay, and cash transactions.
- Planned future integration of secure escrow payments to build trust and further streamline transactions.
- Future enhancements include logistics support, delivery options, and scheduled pickups for greater convenience.

Community Engagement & Notifications

- AI-powered personalized push notifications based on user preferences, browsing history, location proximity, and promotional opportunities.
- Interactive messaging platform enabling seamless buyer-seller communication and community interaction.
- Community boards and live fashion streams to foster a dynamic user community and encourage active participation.
- Reward and loyalty systems to incentivize sustainable shopping habits and community involvement.

Analytics & User Insights

- In-depth analytics providing users insights into trends, popular items, and personalized buying or selling recommendations.
- Seller-focused insights for optimizing item visibility, pricing strategies, and overall listing performance.
- Predictive analytics to enhance personalized content and guide future platform development.

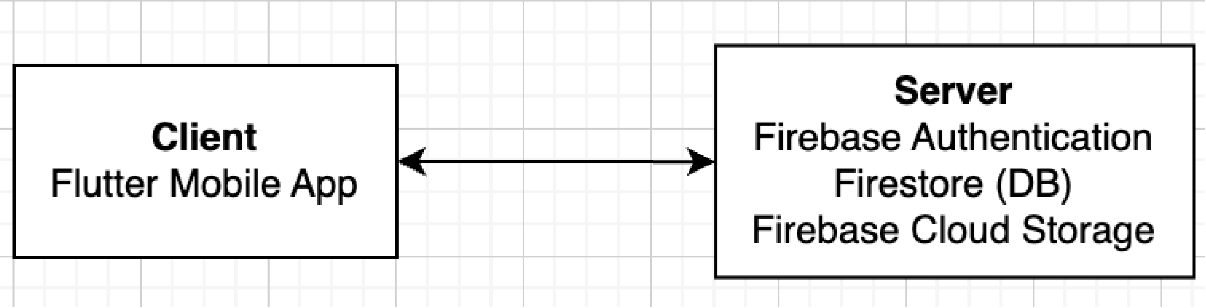
4. Architecture Design

4.1 Architectural Overview

EcoCloset follows a **client-server** approach:

- **Client (Flutter mobile app)**: A single codebase deployed on iOS and Android. Business logic is partially embedded in the client, but critical validations rely on server-side checks.
- **Backend (Firebase)**:
    - **Authentication**: Manages user sessions (email/phone sign-in).
    - **Cloud Firestore**: NoSQL DB for items, user profiles, transactions.
    - **Storage**: For item images and potential ephemeral content.
    - **Cloud Functions**: Hosting serverless logic (e.g., AI tagging calls, payment callbacks).

4.2 High-Level Component Diagram



4.3 Data Flow Overview

Core Data Flows (MVP)

- **User Login**: Mobile app securely communicates with Firebase Auth for authentication and user session management.
- **Item Upload**: App uploads images directly to Firebase Storage while simultaneously sending metadata (e.g., price, tags, description) to Cloud Firestore.
- **AI QuickUpload**: Upon item upload, Cloud Functions trigger Gemini AI to analyze uploaded images and auto-generate descriptive tags, enriching item listings instantly.
- **Search Items**: The app dynamically queries Cloud Firestore with advanced filters (style, size, trending status), delivering real-time personalized search results.
- **Notifications**: Critical notifications, updates, or alerts are managed by Cloud Functions, sent via Firebase Cloud Messaging (push alerts) directly to users.

Future Data Flows (Post-MVP)

- **Advanced Personalization**: AI services continuously analyze user behavior and preferences, proactively personalizing content and pushing notifications for items and events tailored explicitly to individual users.
- **Payment Escrow and Logistics Integration**: Cloud Functions managing payment escrow processes, integration with external logistics providers for delivery or scheduled pickups.
- **Real-Time Community Interactions**: Cloud Firestore supporting real-time chat and community board interactions, leveraging Cloud Functions for moderation, spam detection, and personalized communication suggestions.
- **Analytics-driven Engagement**: Data pipelines extracting insights from Firestore and analytics modules to predict user trends, recommend listings, and personalize user engagement strategies dynamically.

5. App Concept & User Experience

Branding and Look & Feel

EcoCloset uses a clean and modern design with earthy colors to highlight sustainability. The app feels friendly, playful, and approachable, clearly showing users that buying second-hand clothes can be simple and enjoyable.

Homepage

The homepage is the starting point for all users. Here, users immediately see personalized items based on their style, size, and previous activity. Key sections include:

- Top Deals: Best offers available right now.
- Trending Items: Most popular items in the user's local area.
- Seasonal Collections: Special collections organized by seasons or occasions.

Users can quickly access popular categories and their favorite collections with minimal effort.

Explore Page

The Explore Page is designed for easy browsing and discovery. Key features include:

- A smart search bar that suggests results in real-time.
- Detailed filters allowing users to narrow searches by size, price, brand, condition, style, and personalized recommendations ("For Me").
- Endless scrolling to continuously show more relevant clothing items, making discovery fast and enjoyable.

Profile Page

The Profile Page helps users manage their account details, listings, and preferences clearly and simply. Users can easily:

- View personal information, recent purchases, current sales, and saved items.
- Customize their profile with preferences such as favorite brands, sizes, and styles.
- Adjust notification, privacy, and security settings quickly and easily.
- Manage and update their active listings without confusion.

Login / Onboarding Page

Signing up is simple, fast, and hassle-free. EcoCloset offers:

- Quick registration with email or phone verification.
- An easy password recovery process.
- A short onboarding process to set initial preferences, making sure the user sees personalized content right from the start.

Item Detail Page

Each listed item has its own clear, informative page. This page includes:

- High-quality images of the clothing item.
- Clear descriptions, prices, item conditions, and AI-generated tags.
- Quick links to seller profiles, ratings, and user reviews, helping buyers feel confident.
- Future plans include interactive features like virtual try-ons and user comments, making the experience even richer.

Upload Item Page

EcoCloset makes selling clothes straightforward. Uploading items is fast and easy:

- Users can quickly select or take photos from their device.
- Simple fields clearly guide users through adding prices, condition details, and descriptions.
- AI QuickUpload automatically generates tags and detailed descriptions, significantly speeding up the listing process.
- Listings go live instantly, allowing sellers to reach potential buyers quickly.

5.2 User Journey Highlights

Discovery & Onboarding

- User effortlessly downloads EcoCloset from the app store.
- Minimal sign-up steps (email or phone verification).
- Optional, quick onboarding experience to capture preferences (style, size, brands).

Browsing & Finding Items

- Users engage with personalized homepage content or explore using advanced filters and search tools.
- AI-driven recommendations proactively display relevant and appealing items.

Listing & Selling Items

- Seller initiates an effortless "Upload Item" process.
- Seamlessly captures or selects item photos, inputs brief details, and utilizes AI QuickUpload for instant tagging and description generation.
- Item listings are reviewed instantly and published promptly, ensuring quick market presence.

Transaction & Payment

- Buyer easily contacts the seller through integrated messaging to arrange transactions.
- Transactions seamlessly recorded within Firestore, triggering real-time notifications for both buyer and seller.
- Future implementation of secure payment holding systems, integrated payment methods, and delivery scheduling for enhanced transaction ease and security.

6. Technical & Data Design

6.1 Flutter & Firebase Integration

EcoCloset leverages Flutter and Firebase to build a scalable, robust, and intuitive mobile marketplace:

- **Flutter**:
  - Single, unified codebase for seamless Android and iOS deployments.
  - Enables rapid UI prototyping, streamlined development cycles, and consistent user experiences across platforms.
  - Flexible architecture supporting future enhancements such as AR integration and advanced animations.
- **Firebase Suite**:
  - **Authentication**: Secure user authentication with support for email and phone number sign-ins, password resets, and session management.
  - **Firestore (NoSQL Database)**: Scalable database managing real-time data synchronization for user profiles, item listings, transaction logs, and community interactions.
  - **Storage**: Secure storage and retrieval of item images, optimized for performance with automatic image scaling.
  - **Cloud Functions**: Serverless backend operations, such as AI-driven auto-tagging, notification handling, transaction status updates, payment verification, and content moderation.

6.2 Data Model (Firestore Collections)

Users Collection

- **Document ID**: userId
- **Fields**:
  - Basic Info: name, email, phone, createdAt, lastLogin.
  - Preferences: size, style, favorite brands, notification settings, location data.
  - Social & Interaction: followers, following, ratings, badges (e.g., trusted seller).
  - Activity History: purchaseHistory, saleHistory, viewedItems, likedItems.

Items Collection

- **Document ID**: itemId
- **Fields**:
  - Seller Details: sellerId.
  - Item Info: title, description, size, price, tags (AI-generated and user-added), condition, brand.
  - Media: imageURLs, videoURLs (future feature).
  - Metadata: createdAt, updatedAt, views, likes, status (active/sold/archived).

Transactions Collection

- **Document ID**: transactionId
- **Fields**:
  - Participants: buyerId, sellerId.
  - Item Reference: itemId, itemTitle, itemPrice.
  - Payment Info: paymentMethod (Bit, Paybox, cash, Apple Pay, Google Pay), paymentStatus (pending, completed, canceled).
  - Transaction Details: transactionStatus (initiated, confirmed, completed, disputed), timestamps, delivery/pickup schedule (future feature).

Notifications Collection

- **Document ID**: notificationId
- **Fields**:
  - Target userId, message, type (item updates, personalized recommendations, system alerts, transaction updates).
  - Metadata: createdAt, readStatus.

Community Interaction Collection (Future)

- **Document ID**: interactionId
- **Fields**:
  - Type (message, comment, review, community post), content, authorId, targetItemId/userId, createdAt, moderatedStatus.

6.3 Data Flows

Create

- **Item Listings**: User uploads item → stores media in Storage → triggers AI tagging via Cloud Functions → writes enriched item data to Firestore.
- **Transactions**: Seller initiated transaction → new document created referencing buyer, seller, item → triggers notification updates.

Read

- **Search & Personalization**: Users query Firestore with advanced filters (size, style, price, location proximity, sustainability ratings, personalized "For Me" queries).
- **Real-time Updates**: Transactions, messages, and notifications updated in real-time, instantly reflected in the user interface.

Update

- **Item Updates**: Sellers/admins update details such as price, descriptions, availability → real-time updates to users watching or interacting with the item.
- **Profile Updates**: Users modify preferences → updates immediately reflected in personalized feeds and recommendations.

Delete

- **Item Removal**: Sellers delete items → Cloud Functions handle deletion of associated media and Firestore metadata, notifying relevant users.
- **Moderation Actions**: Admins flag/remove inappropriate content or interactions.

Transaction Management

- Comprehensive handling of buyer-seller interactions from initiation through completion.
- Integrated verification via Cloud Functions, ensuring secure payments and timely transaction updates.
- Escalation pathways for disputes or issues, maintaining transaction integrity and trust.

7. Security & Compliance

7.1 Security Requirements

- **Firebase Authentication** ensures user identity. Only authenticated users can create or buy items.
- **Cloud Firestore Security Rules** restricting read/write to item owners or admin roles.
- **HTTPS / TLS** enforced across all connections to Firebase.

7.2 Data Privacy & Compliance

- Consider local data protection regulations.
- Users can request account deletion; the system must remove personal data.

7.3 Fraud Prevention

- Admin and automated checks for suspicious behavior.
- Email/phone validation.
- User rating system to enhance trust.

8. Performance & Scalability

8.1 Performance Goals

- **Fast Loading**: Typical item listing queries returning results within 1–2 seconds.
- **Image Optimization**: Thumbnails or compressed images for faster browsing.
- **High Concurrency**: Firebase can scale to thousands of simultaneous app users.

8.2 Scalability Approach

- **Firebase Auto-Scaling**: Firestore, Storage, Functions all scale with usage.

8.3 Future Growth

- Expand to multiple regions, multi-language support.
- BigQuery exports for advanced analytics or data science.
- Payment integrations or chat features for community building.

9. Error Handling & Logging

9.1 Error Handling

- **Client-Side**: Flutter user-friendly error messages (e.g., network issues, invalid data).
- **Server-Side**: Cloud Functions logs exceptions.

9.2 Logging & Monitoring

- **Firebase Crashlytics** can collect real-time crash reports.
- **Firebase Analytics** logs user events, conversion funnels.
- **Google Cloud Logs** for Cloud Functions to debug backend events or AI tagging issues.

10. Assumptions & Dependencies

10.1 Assumptions

- Users have modern smartphones and a stable internet.
- AI services for QuickUpload are reliable or fallback is manual entry.
- Future expansions (web platform, advanced community features) remain consistent with the existing architecture.

10.2 Dependencies

- **Flutter** framework and supporting libraries.
- **Firebase** (Auth, Firestore, Storage, Functions, Analytics, FCM).
- **AI/ML Provider** (integrated for image tagging).

11. Appendix

11.1 Glossary

- **AI QuickUpload**: Automated tagging of items based on image recognition.
- **Marketplace**: A platform where multiple sellers list items for potential buyers.
- **Sustainability**: Reducing environmental impact by promoting reuse and responsible consumption.

11.2 Acronyms

- **UI**: User Interface
- **DB**: Database

11.3 Document History

- **v1.0** – Comprehensive HLD capturing concept, system design, and MVP scope.
- **v1.1** – Comprehensive HLD capturing concept, system design, and MVP scope - more accurate description with simpler explanations.