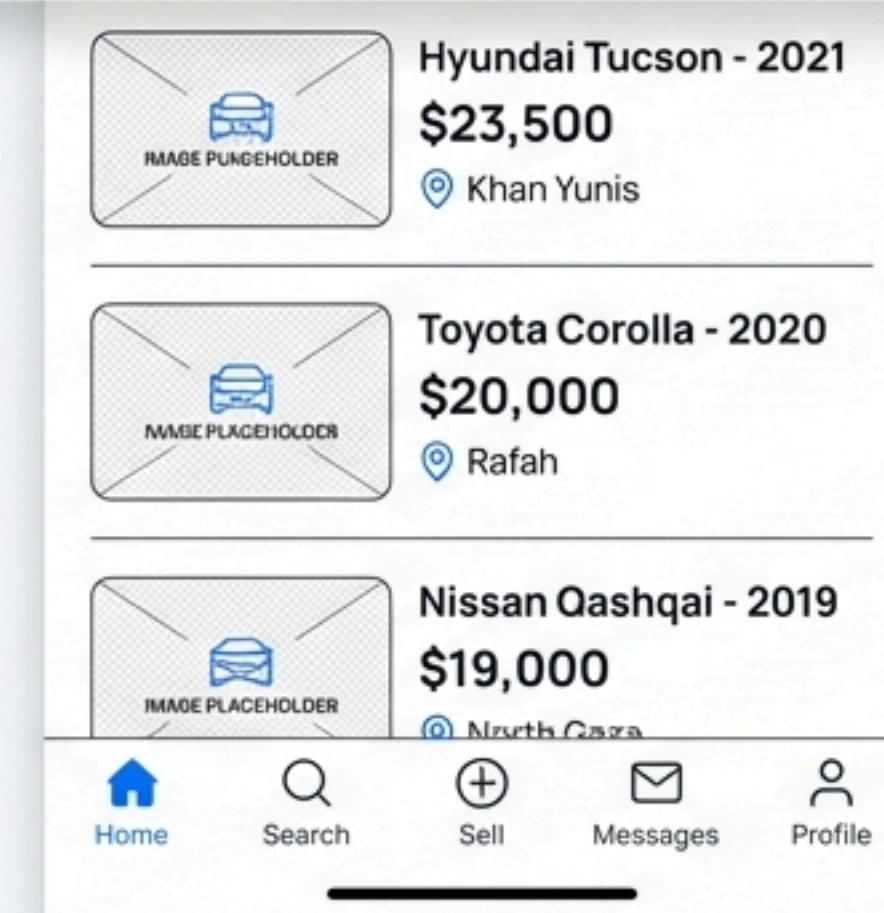


Gaza Cars: A Blueprint for the Digital Automotive Marketplace

Translating the Software Requirements Specification into a Tangible User Experience



A Specialized Marketplace for Gaza's Automotive Sector

Our Purpose

Connecting Buyers and Sellers

To provide a comprehensive and secure mobile platform that serves as the primary digital marketplace for the automotive sector in Gaza. The application is designed to be the definitive reference for development, stakeholders, and quality assurance.

The Technical Scope

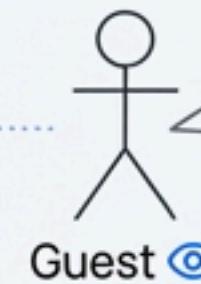
A Serverless, Scalable System

Gaza Cars is a client-server application built on a robust, modern stack. The Flutter mobile app communicates directly with a Google Firebase backend, ensuring real-time data synchronization, scalability, and high availability.

Designed for Every Role in the Marketplace

Guest User

Can browse all public car listings.
Read-only access encourages registration for more features.



Guest

Gaza Cars System

View Listings

Search & Filter

Register/Login

Manage Profile

Add Car Listing

Chat with Seller

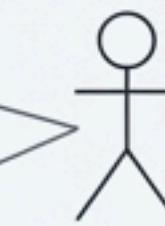
Moderate Listings

Ban Users

A blue shield icon with a white outline.

Administrator

Platform moderator with full access to manage users and content. Ensures the quality and safety of the marketplace through a dedicated dashboard.



Administrator



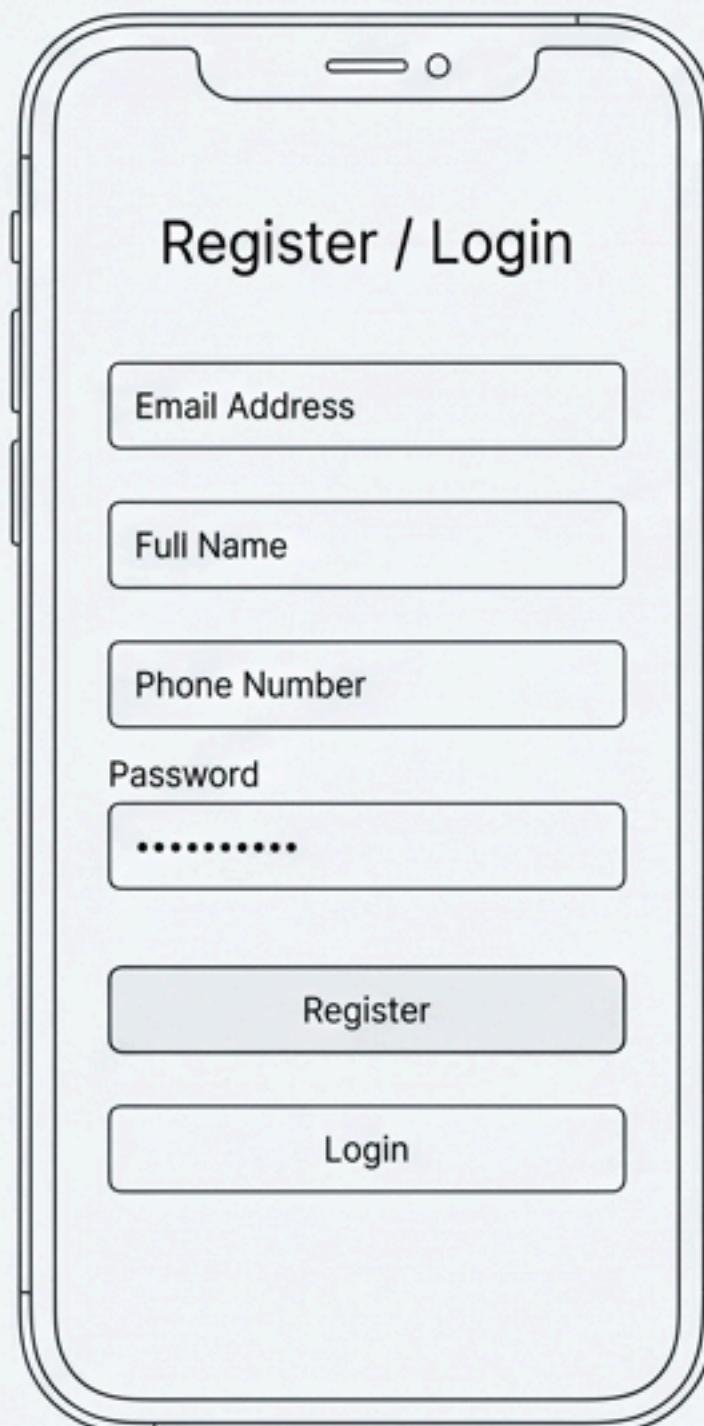
Registered User

A stick figure icon with a person symbol at the bottom.

Registered User (Buyer/Seller)

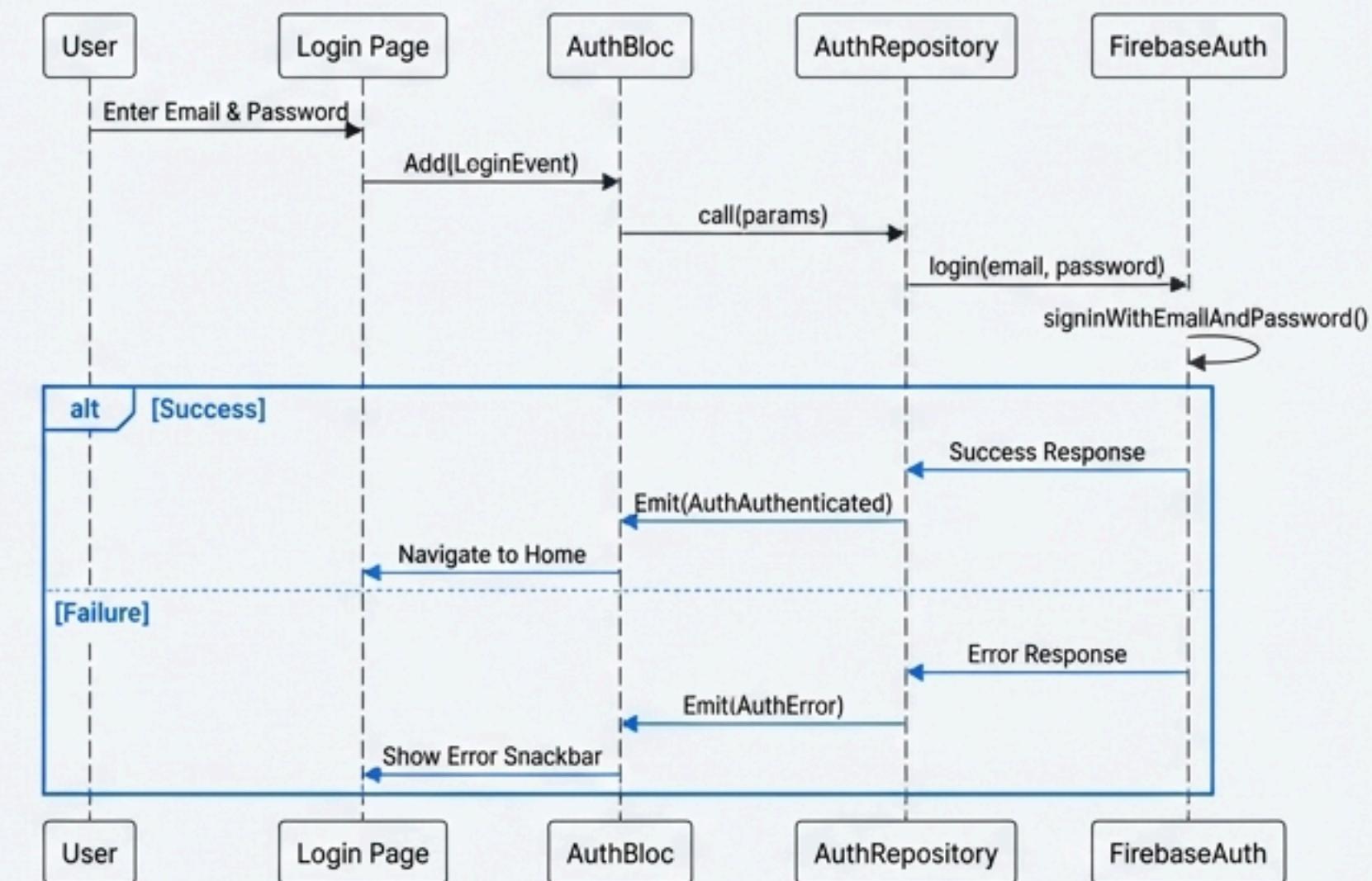
The core user. Can list cars for sale, manage their profile, save favorites, and communicate directly with other users via real-time chat.

Seamless Onboarding and Secure Access



- **Quick Registration:** Users onboard with essential details: Email, Name, Phone, and a secure password. (FR-AUTH-01)
- **Input Validation:** Ensures valid email formats and passwords of at least 6 characters for enhanced security. (FR-AUTH-02)
- **Persistent Sessions:** Secure token storage keeps users logged in for a frictionless return experience. (FR-AUTH-03)

User Login Sequence Diagram



Technical Flow: This diagram illustrates the step-by-step interaction between the UI, business logic (Bloc), and Firebase Auth for a secure login.

Exploring the Digital Showroom

Price Range:

Min - Max

Year Range:

Min - Max



Toyota Camry
\$25,000

[view](#)



Honda CR-V
\$32,500

[view](#)



Honda CR-V
\$35,000

[view](#)



Ford F-150
\$45,000

[view](#)

 Loading more...



Intuitive Browsing: Guests and registered users can effortlessly scroll through available cars.



Powerful Filtering: Users can precisely narrow results by:

- Price Range (Min/Max) (FR-SRCH-01)
- Year Range (FR-SRCH-02)



Optimized Performance: Search results are paginated and 'lazy loaded' to ensure a fast, smooth experience even with thousands of listings. (FR-SRCH-03)

A read-only experience for guests provides a compelling preview, driving user registration.

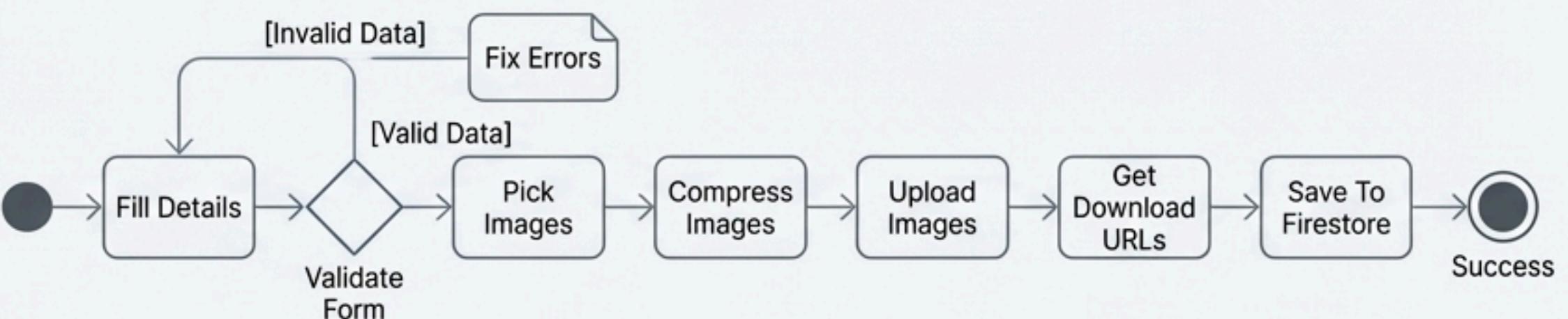
A Seller's Toolkit for Creating the Perfect Listing

The mobile application interface for adding a car listing includes:

- Upload Photos (Max 10)**: A section for uploading up to 10 images, featuring three placeholder boxes with a plus sign.
- Add a Car**: A main category for listing details.
- Make**: A text input field for the vehicle make.
- Model**: A text input field for the vehicle model.
- Year**: A text input field for the vehicle year.
- Price**: A text input field for the vehicle price.
- Location**: A section for selecting the listing location.
- Select Location**: A dropdown menu showing options like Gaza City and Rafah.
- Publish Listing**: A large button at the bottom to publish the listing.

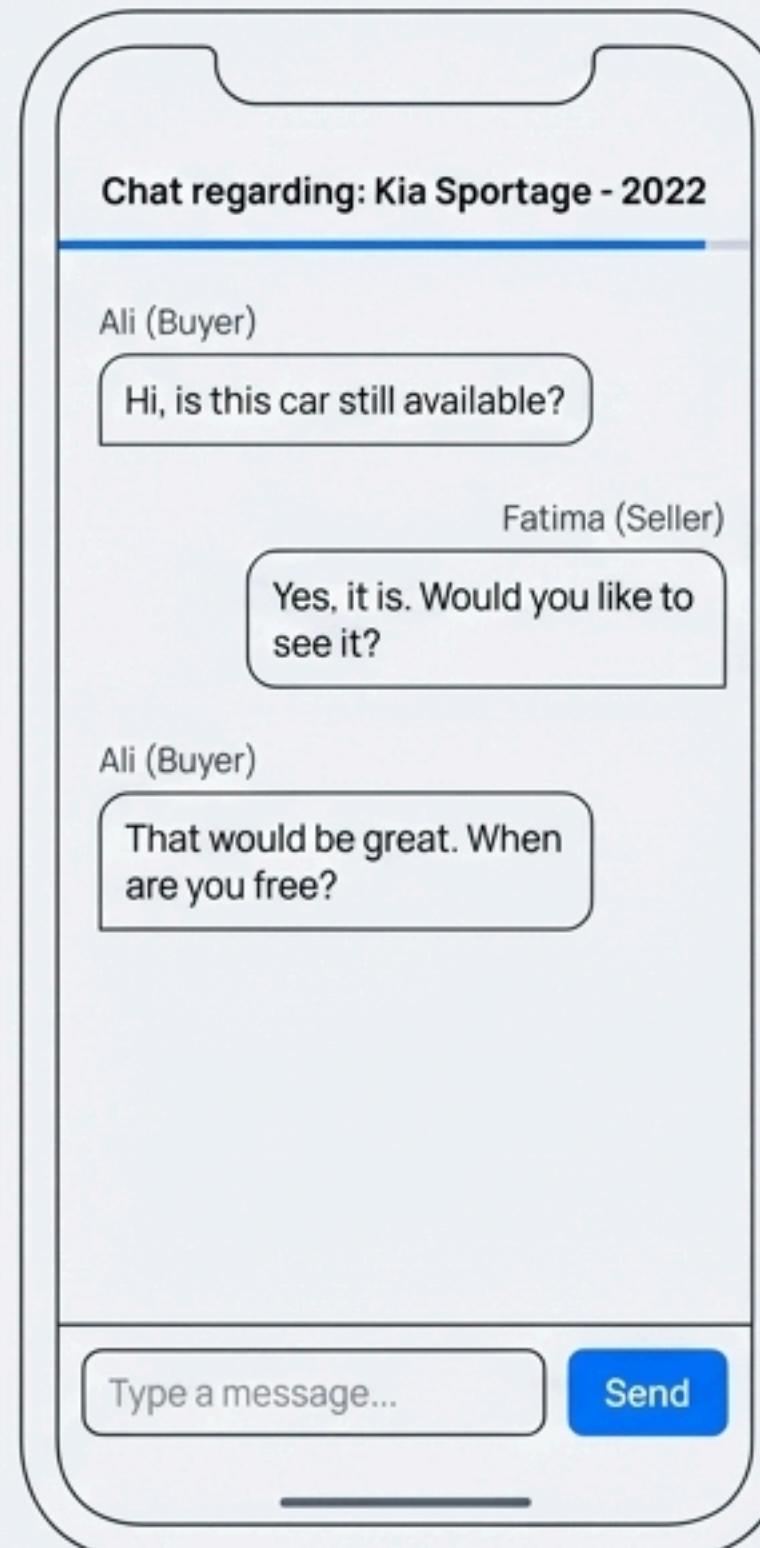
- Rich Media Uploads:** Sellers can upload up to 10 images per vehicle to showcase every detail. (FR-CAR-01)
- Bandwidth Optimization:** The system automatically compresses images to a maximum of 1080p resolution, ensuring fast uploads and viewing. (FR-CAR-02)
- Structured Data:** Sellers must provide key details, including selecting a location from a predefined list (e.g., Gaza City, Rafah). (FR-CAR-03)
- Listing Management:** Sellers can easily mark their car as 'Sold,' which removes it from active search results. (FR-CAR-04)

Activity Diagram: Adding a Car



Workflow: From form validation to image compression and database entry, this flow ensures data integrity for every new listing.

Connecting Buyers and Sellers in Real-Time



- **Contextual Initiation:** Users can start a conversation directly from any 'Car Details' page. (FR-CHAT-01)

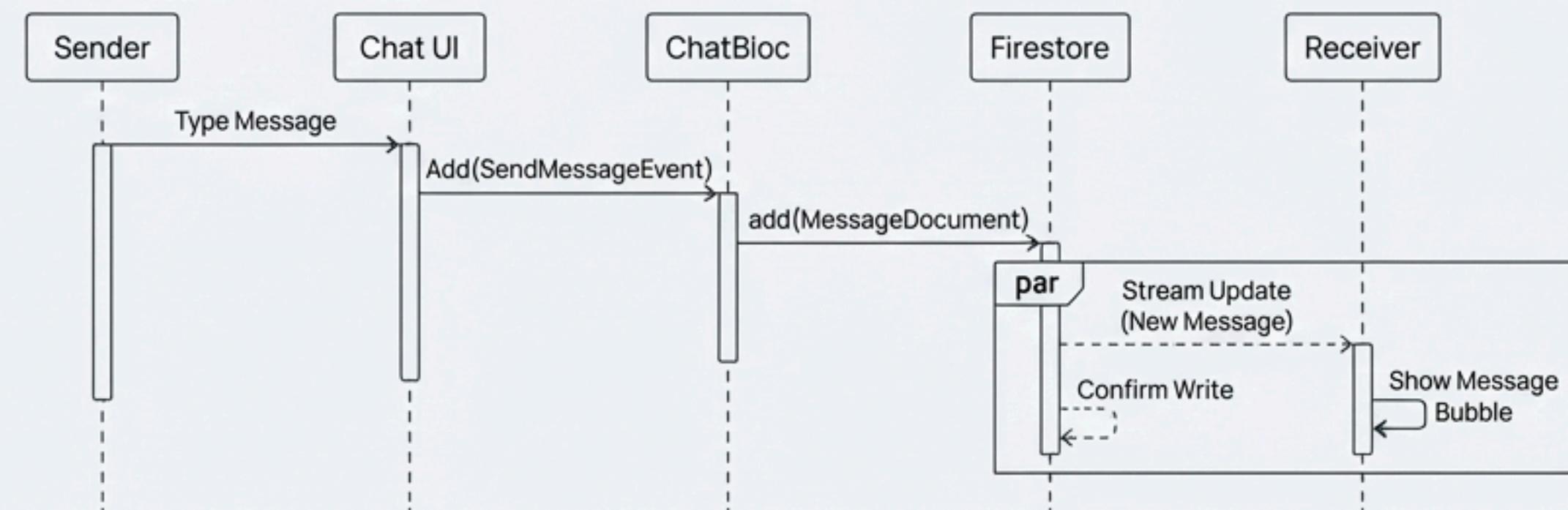


- **Instant Synchronization:** Messages are synced in real-time across all of a user's devices using Firestore's streaming capabilities. (FR-CHAT-02)



- **Centralized Inbox:** A dedicated 'Inbox' tab displays all active conversations for easy management. (FR-CHAT-03)

Sequence Diagram: Sending a Message



****Real-Time Data Flow**:** This illustrates how a message travels from the sender's UI to Firestore and is instantly reflected on the receiver's device.

The Blueprint: A Clean and Scalable Architecture

Presentation Layer

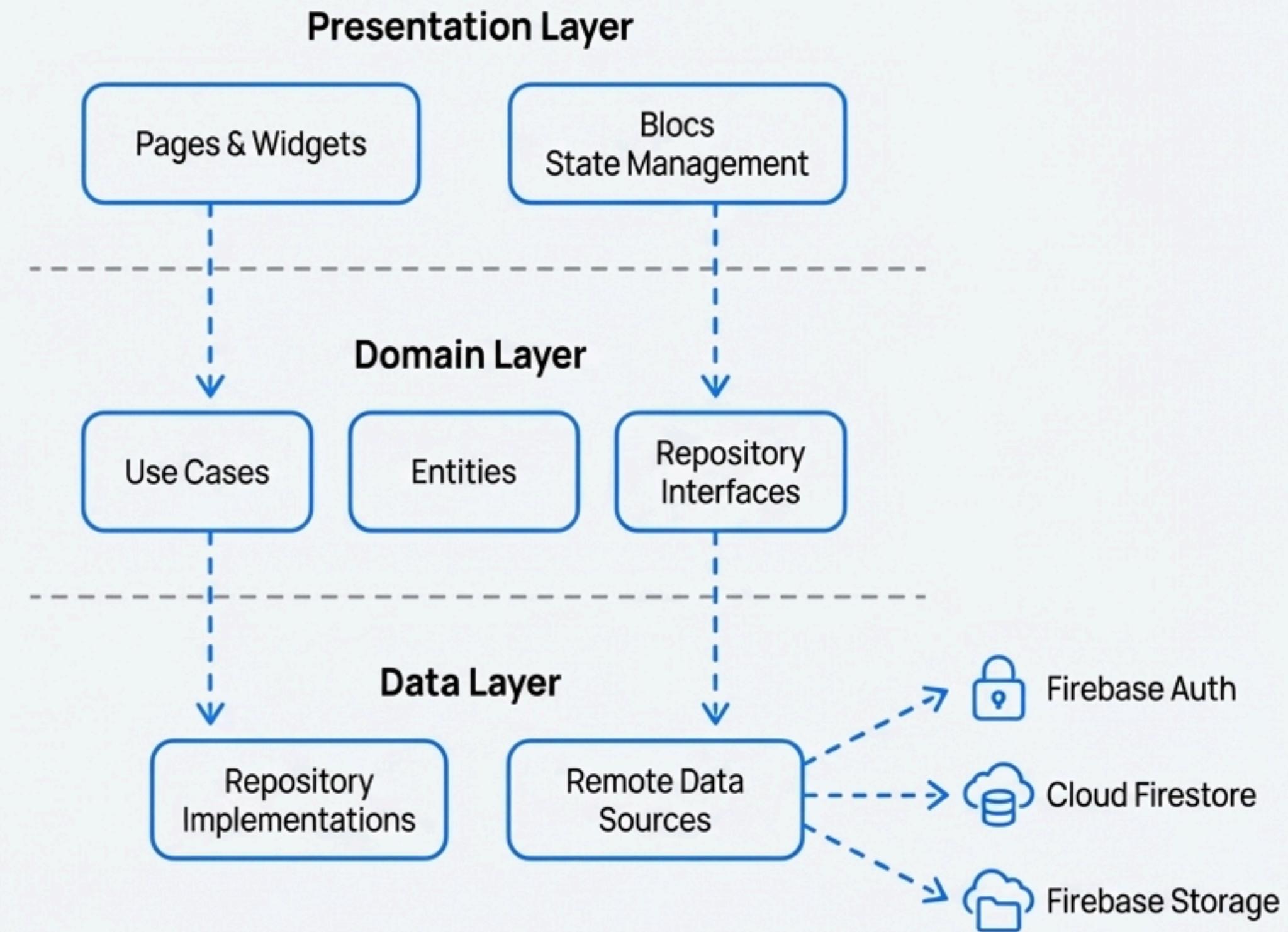
- **Technologies:** Flutter, Bloc State Management
- **Responsibility:** Renders the UI (Pages & Widgets) and handles user input. Blocs manage state and connect the UI to the business logic.

Domain Layer

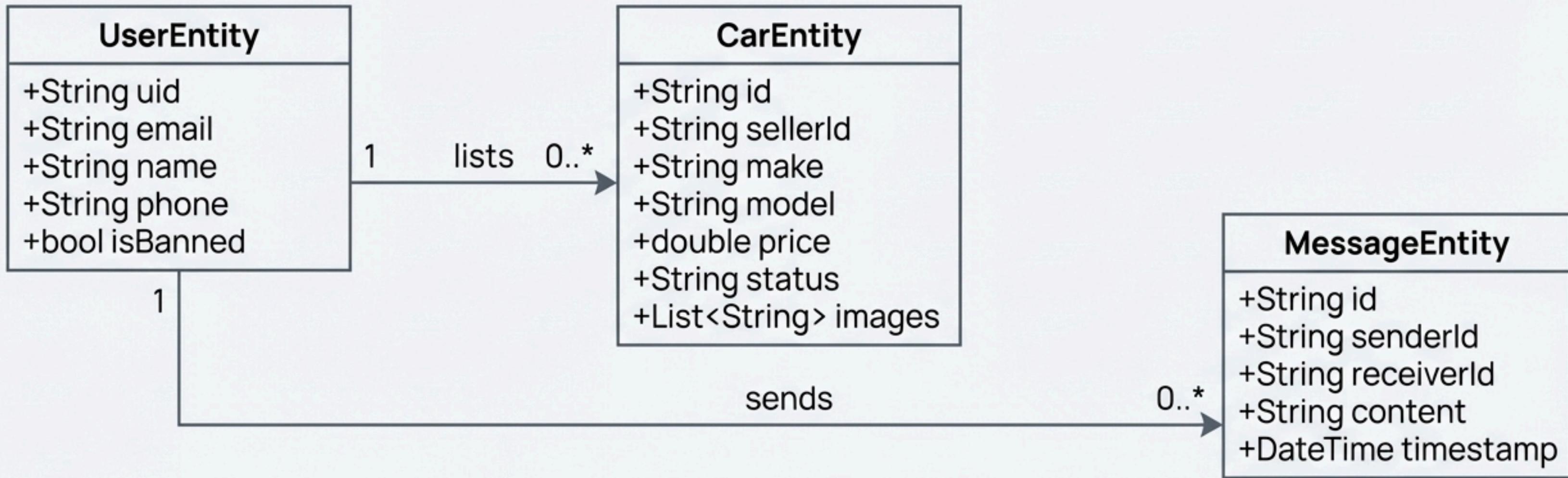
- **Technologies:** Pure Dart
- **Responsibility:** The core of the application. Contains the business logic (Use Cases), data models (Entities), and abstract interfaces for repositories. It is completely independent of UI and data sources.

Data Layer

- **Technologies:** Firebase Auth, Cloud Firestore, Firebase Storage
- **Responsibility:** Implements the repository interfaces from the Domain layer. Manages all communication with external data sources like the Firebase backend.



The Core Data Entities Driving the Application

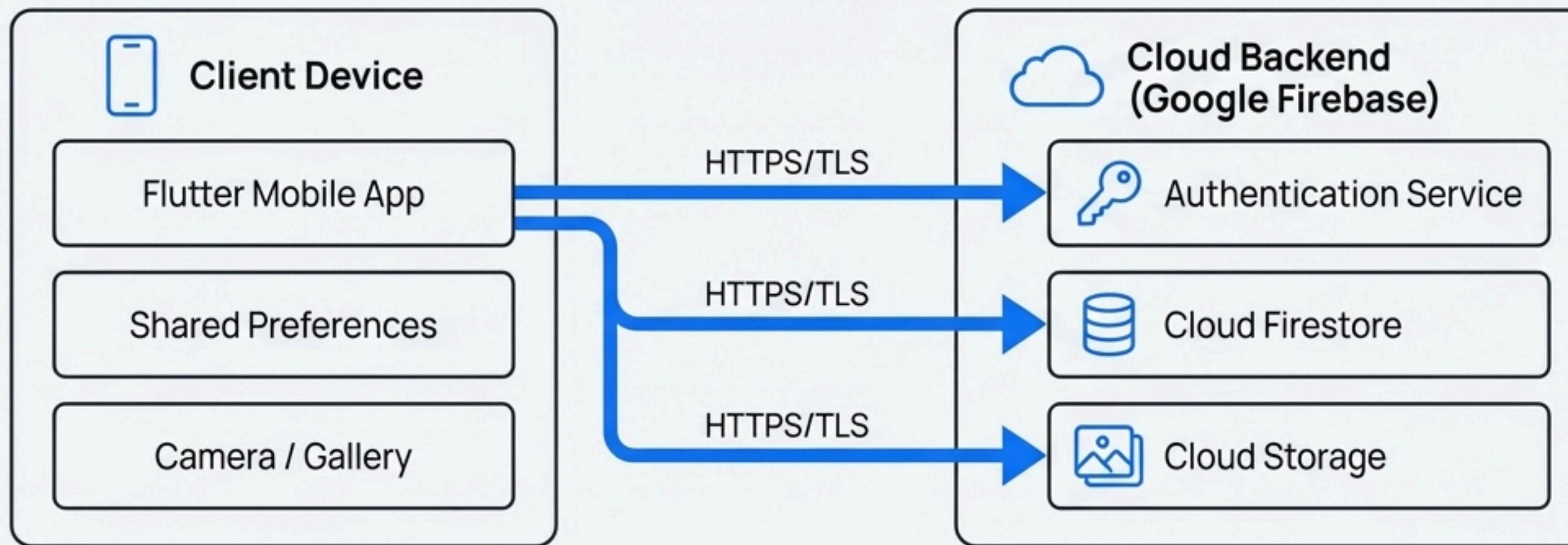


UserEntity: Represents every individual on the platform. Stores uid, email, name, phone, and moderation status like isBanned.

CarEntity: The central object of the marketplace. Contains all vehicle details, including make, model, price, status, and a list of images. Critically linked to a sellerId.

MessageEntity: A single communication packet. Stores content, timestamp, and links the senderId and receiverId to create a conversation thread.

Deployed for Performance on a Global Cloud Infrastructure



Client Device

- The Flutter Mobile App runs on the user's phone.
- Utilizes local storage (Shared Preferences) for session data and device hardware (Camera/Gallery) for image uploads.

Cloud Backend (Google Firebase)

- A fully managed, serverless backend.
- **Authentication Service:** Handles user identity and security.
- **Cloud Firestore:** The NoSQL database for all application data (users, cars, messages).
- **Cloud Storage:** Securely stores and serves all user-uploaded images.

Communication

- All data in transit between the client and backend is secured with HTTPS/TLS encryption.

Guarantees of Quality: Our Non-Functional Commitments



Performance

- < 500ms API response time on 4G networks.
- < 1s per image for on-device compression.



Reliability

- **99.9%** Uptime, backed by the Google Firebase Service Level Agreement (SLA).
- **Transactional Integrity** using Firestore transactions for critical data updates.



Security

- **SSL/TLS Encryption** for all data in transit.
- **Granular Access Control** via Firestore Security Rules, ensuring users can only modify their own data.

Empowering Administrators to Maintain a Healthy Marketplace

Admin Dashboard

Total Users **1,450**

Total Cars **2,130**

Pending Listings **15**

Pending Car Listings

Car Details	Seller	Actions
Hyundai Elantra - 2021	John Doe	<button><input checked="" type="checkbox"/> Approve</button> <button><input type="checkbox"/> Reject</button>
Hyundai Elantra - 2022	John Doe	<button><input checked="" type="checkbox"/> Approve</button> <button><input type="checkbox"/> Reject</button>
Hyundai Elantra - 2023	John Doe	<button><input checked="" type="checkbox"/> Approve</button> <button><input type="checkbox"/> Reject</button>
Hyundai Elantra - 2024	John Doe	<button><input checked="" type="checkbox"/> Approve</button> <button><input type="checkbox"/> Reject</button>
Hyundai Elantra - 2025	John Doe	<button><input checked="" type="checkbox"/> Approve</button> <button><input type="checkbox"/> Reject</button>

- At-a-Glance Oversight:** A dashboard displays key metrics like Total Users and Total Cars for a high-level view of platform health. (FR-ADM-01)
- Content Moderation:** Admins can review, 'Approve,' or 'Reject' new car listings to ensure quality and compliance. (FR-ADM-02)
- User Management:** The ability to 'Ban' malicious users, which immediately prevents them from logging in and displays a specific error message. (FR-ADM-03, FR-ADM-04)

The Road Ahead: Evolving the Gaza Cars Experience

Phase 2: Engagement

(Month 2)



- **Feature 1:** Push Notifications (FCM) to increase user re-engagement.
- **Feature 2:** Advanced & Saved Searches for power users.

Phase 3: Monetization

(Month 3)



- **Feature 1:** Premium Listings ('Featured' or 'Top of Search') via Stripe integration.
- **Feature 2:** Verified Accounts for professional car traders.

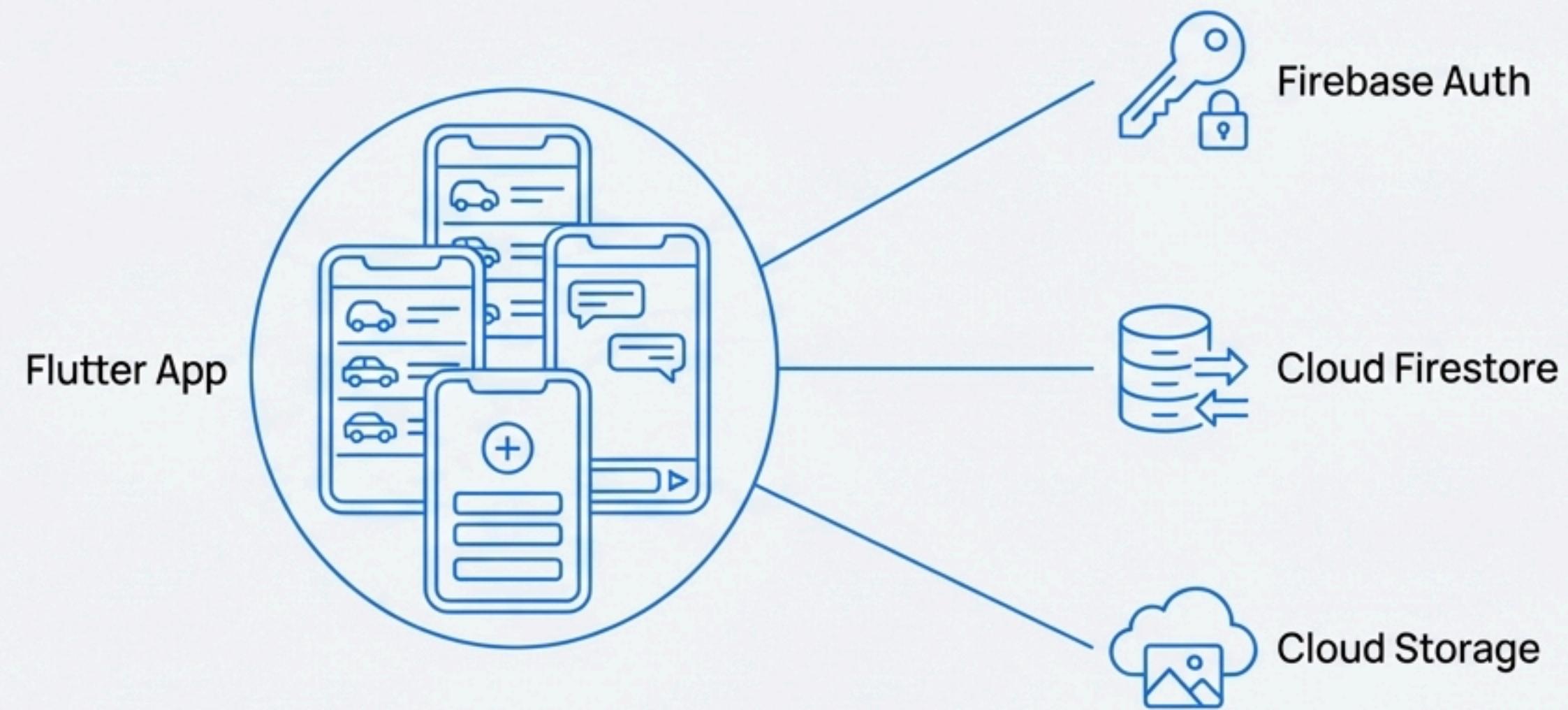
Phase 4: Expansion

(Month 6)



- **Feature 1:** A full-featured Web Application built with Flutter Web.
- **Feature 2:** AI-driven Price Prediction to help sellers and buyers.

Gaza Cars: A Complete, User-Centric Automotive Ecosystem



By bridging an intuitive user interface with a robust, scalable, and secure backend architecture, Gaza Cars is more than an application. It is a comprehensive digital ecosystem, engineered from the ground up to serve the specific needs of the Gazan automotive market. Every requirement, from the user's first tap to the underlying database rule, is purposefully designed to build a trusted and efficient marketplace.