CASE STUDY · MOBILE APPS

PilotChat for PilotBazar

Messaging app for the PilotBazar marketplace: chat, voice and video calls, reels.

Client	Service	Year
PilotBazar	**Mobile apps**	**2025**

Built with
Flutter · Dart · Laravel 12 · WebRTC · Pusher Channels · Firebase Cloud Messaging · SQLite · Android

6	4 mo	3.8k
prototypes before the calling stack was right	from first commit to version 1.1	lines of Dart for calls alone

THE CHALLENGE

PilotBazar wanted messaging of its own: buyers and sellers talking, calling and sharing inside the PilotBazar ecosystem, under PilotBazar accounts, instead of drifting off to other apps.

Chat was the easy part. The hard part was **calling**. A call has to ring on a locked phone, survive a bad mobile connection, get through carrier networks that block direct peer-to-peer traffic, respect the phone's silent mode, and keep running when the user switches apps. Every one of those is a separate problem on Android, and the answers change from one Android version to the next.

It all had to run well on the mid-range Android phones most people in Bangladesh carry.

WHAT WE BUILT

We did not guess our way through the calling stack. Before writing the production app we built **six separate prototypes** over two weeks, from a bare WebRTC demo to a native Android signalling client, to learn exactly how offer, answer and ICE candidates have to flow, where TURN relays are needed, and what Android allows a call to do on a locked screen. Only then did we design the real thing.

The app

- **Chat:** one-to-one and group conversations with replies, forwarding, reactions, labels to organise chats, voice messages with a live waveform, and any attachment up to 100 MB with in-app previews.
- **Calls:** WebRTC voice and video, with the TURN and STUN servers fetched from our API so they can change without an app update. Incoming calls ring on the lock screen, including on Android 14 and later, through a foreground service with a full-screen notification. Ringtone and vibration follow the phone's ringer mode, the volume button silences a call, the proximity sensor turns the screen off at the ear, picture-in-picture keeps a video call visible while the user does something else, and a call can switch from audio to video mid-way. A busy user, a missed call and a plain SIM call are all handled.
- **Reels and photos:** a short-video feed with comments, uploads that continue in the background with live progress, and server-side compression for anything over 15 MB.
- **Offline first:** chats, messages, calls, reels and photos are cached in SQLite, so the app opens instantly and works through connection drops.

Under the hood

The app is **Flutter**, Android first with the iOS project in place. It talks to a **Laravel 12 API** we built alongside it: authentication, messaging, media, reels, and the call signalling endpoints (initiate, accept, reject, signal, ICE candidate, busy, switch). **Pusher Channels** carries real-time events and **Firebase Cloud Messaging** wakes the phone for calls and messages. Server-side version control lets PilotBazar force an update or put the app into maintenance.

THE RESULT

From the first commit to version 1.1 took four months, across **168 commits** in the app and the API. The calling layer alone is about 3,800 lines of Dart, and the lessons from the six prototypes are the reason it works on real phones, on real networks, with the screen locked.

PilotBazar now owns the whole conversation: chat, calls, reels and photos under its own brand, on its own servers, with no third-party messenger in between.
