

THANH VO HUY

Android Developer

thanh0x.com | [GitHub: Huythanh0x](https://github.com/Huythanh0x) | huythanh0x@gmail.com | [LinkedIn: huythanh0x](https://www.linkedin.com/in/huythanh0x)

SUMMARY

Android engineer with 2 years of production experience building native Android features (Kotlin, Java, MVVM) for a 2M+ user education platform, including custom UI systems, third-party SDK integration, and Flutter module embedding. Broadened into backend and AI-integrated systems (NestJS, GraphQL, Google Vertex AI) at the same company, adding a full-stack perspective on API design and system architecture. Now refocusing on Android, with hands-on Kotlin Multiplatform experience from a personal project.

SKILLS

Languages: Kotlin, Java, TypeScript

Android & Mobile: Android SDK, Jetpack (ViewModel, LiveData, Navigation, DataStore), MVVM, Clean Architecture, Compose/Kotlin Multiplatform, Flutter Module Embedding, Coroutines, Paging 3, Firebase (Crashlytics, FCM), LeakCanary

Networking & Persistence: Retrofit, Ktor Client, Room / SQLite

DI & Testing: Hilt, Koin, JUnit, Mockito

Tools & IDE: Android Studio, Cursor, Claude Code, Figma

Version Control & Workflow: Git, GitHub, Jira, Linear, Agile-Scrum

Backend: NestJS, Spring Boot, GraphQL, REST APIs, WebSocket, TypeORM, PostgreSQL, MySQL, Redis, Docker, CI/CD, LangGraph, Google Vertex AI

EXPERIENCE

Android Developer Jun 2023 – Jun 2025

Vitalify Asia Co., Ltd | Ho Chi Minh City, Vietnam

Native Android app for a 2M+ user English-learning platform connecting students with tutors through live lessons, tests, and practice flows.

- Delivered native Android features across lesson, test, and pronunciation modules using Kotlin, Java, and MVVM, growing from fresher to senior-level ownership of feature delivery and architecture decisions.
- Standardized inconsistent architecture patterns (navigation, state handling, UI lifecycle) across multiple modules, reducing navigation- and state-related bugs and making feature development more predictable for the team.
- Designed and built a custom rendering pipeline for pronunciation exercises (intonation, highlighting, dynamic API-driven content), shipping a high-complexity learning UI used by millions of learners.
- Integrated a Flutter-based mini-game into the native Android app, building a reliable native-Flutter data/state bridge without breaking existing account and business logic.
- Stabilized a third-party speech-recognition SDK (Chivox) integration across Activity/Fragment boundaries despite sparse documentation, improving reliability of speech-assessment flows in production.
- Migrated legacy Java/Volley code toward Kotlin, MVVM, and structured state management (ViewModel/LiveData), improving JSON-parsing safety and reducing crash-prone code paths.

Backend NestJS Developer Jul 2025 – Present

Vitalify Asia Co., Ltd | Ho Chi Minh City, Vietnam

Backend systems for AI-assisted training and healthcare-related products.

- Owned database schema design and GraphQL API development (NestJS, TypeScript, PostgreSQL, Redis) from early-stage product development through production release.
- Gained experience integrating AI features (LangGraph, Google Vertex AI) into a production healthcare-related product's AI-assisted workflows.
- Contributed to multi-tenant system architecture and migration, and maintained containerized services (Docker) across development and deployment.
- Brings full-stack and AI-integration experience back to Android work, applying API-design and system-architecture judgment to mobile feature development.

- Delivered browser-automation tools and a Chrome extension end-to-end, from requirement analysis through final delivery, including protocol/traffic analysis and account-scale operational tooling.

EDUCATION

University of Information Technology – VNUHCM Sep 2019 – Sep 2022

B.S. Computer Science, GPA 3.3

CERTIFICATES

- [IELTS 6.5 Academic \(2022\)](#).

PERSONAL PROJECTS

Course Deal Client — Kotlin Multiplatform

Kotlin, Compose Multiplatform

- Built a cross-platform client (Compose Multiplatform) sharing UI and business logic across Android and iOS, consuming a companion backend's validation and crawler APIs to surface verified course deals in real time.
- Implemented infinite-scroll pagination (Paging 3) for the course-deal list to handle large, continuously-growing result sets efficiently.
- Android build distributed via F-Droid due to app-store restrictions on coupon-aggregation apps.
- Setting up an automated build and release pipeline (GitHub Actions + Fastlane) for the Android client — in progress.
- Originally built as a native Android app (Kotlin, Retrofit, RoomDB, MVVM, Hilt, BiometricPrompt) with offline caching, later rebuilt as a cross-platform client.

Architecture: migrated from a native Android stack (MVVM, Hilt, Retrofit) to a shared Kotlin Multiplatform module (Compose Multiplatform UI, Ktor Client networking, Koin DI) reused across Android and iOS.

[Code](#) · [Product](#)

Course Deal Backend — Spring Boot Microservices

Java, Spring Boot, MySQL, Docker

- Designed a validation service that checks user-submitted course-discount links against the target platform's API in real time.
- Built a separate crawler/ingestion pipeline that discovers and verifies deals automatically, writing results to the database with deduplication and freshness checks.
- Hardened the link-validation endpoint against SSRF by restricting outbound requests to the target platform's own domain.
- Powers a public web app and the companion Kotlin Multiplatform client via a shared API.

Architecture: microservices — independently deployable crawler/ingestion and validation services sharing a common MySQL data layer, containerized with Docker.

[Code](#) · [API Docs](#)

Custom Rich-Text Rendering Library — Kotlin Multiplatform (*In Progress*)

Kotlin, Compose Multiplatform (Canvas)

- Rebuilding, as an independent open-source library, a custom rich-text rendering technique — mixed inline styles, phonetic sub-annotations, and intonation-curve overlays — originally developed for a custom UI feature in a production language-learning app.
- Targeting Android, iOS, and desktop through a shared Canvas-based rendering core.

Architecture: porting a View/Canvas-based rendering engine from native Android (Java/Kotlin) to a shared Kotlin Multiplatform Canvas API, enabling one rendering core across Android, iOS, and desktop.

[Code](#) · [Maven](#)