Shaping Ideas Through Code, Curiosity, and Continuity

Pramuditha Muhammad Ikhwan

iOS Developer

Highlighted Projects from 2025

Pramuditha Muhammad Ikhwan

iOS Developer

A growth-minded iOS developer passionate about turning real-world challenges into intuitive digital experiences.

Educational Background

Apple Developer Academy @ Infinite Learning
Learner Cohort 6
Feb 2025 - Dec 2025

Universitas Negeri Yogyakarta
Majoring in Information Technology
Aug 2021 - May 2025

Bangkit Academy
Mobile Development Cohort
Sep 2024 - Jan 2025

Professional Background

4TITU PTE LTD
Full Stack Developer
Aug 2025 - Present

Apple Developer Academy @ Infinite Learning
iOS Developer
Feb 2025 - Dec 2025

PT Widya Inovasi Indonesia
System Analyst
Aug 2023 - Dec 2023

Selected Works as
iOS Developer.

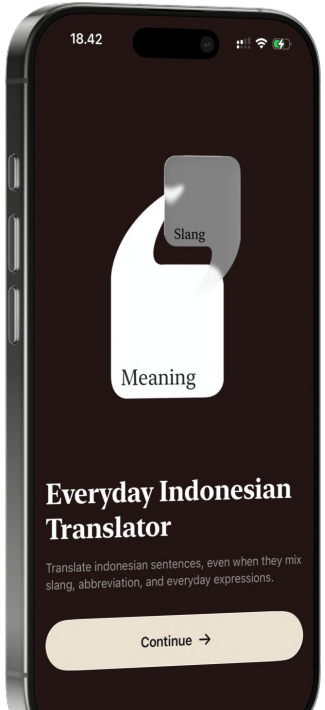
Building SLNG: My Journey to Craft a **Fast, Native, Playful App**

SLNG

An app to help foreign students understand informal Indonesian

Foreign students struggle to understand the meaning and social context of Indonesian slang and slang evolves quickly, which makes it hard for them to keep up.

From a developer’s perspective, the interesting problem wasn’t just the slang itself—it was how to design an iOS experience that handles unpredictable, fast-moving language gracefully.

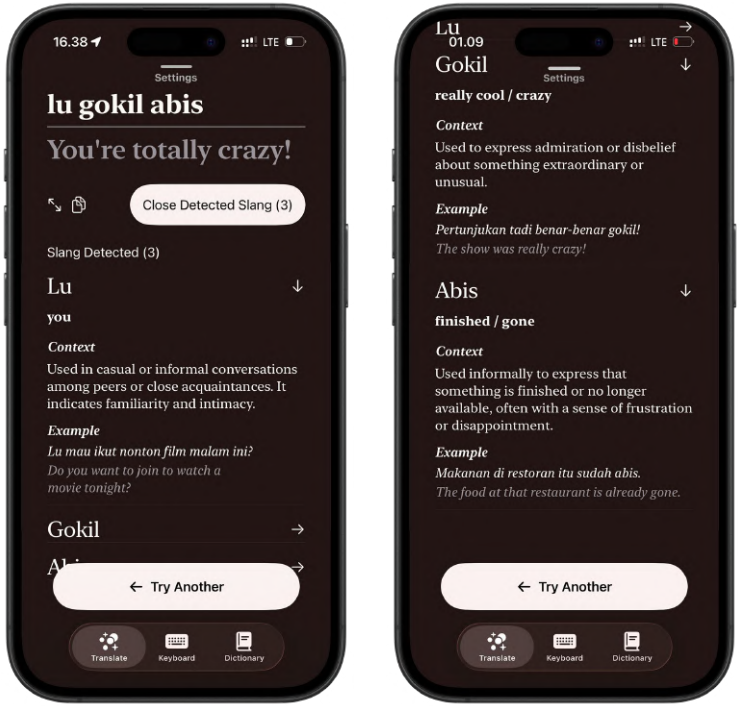


PROBLEM STATEMENT

Foreign students struggle to grasp meaning + social context, not just words.

Many foreign students could recognize Indonesian slang, but not the intention behind it. Meaning shifts with tone, sentiment, and social context. One word can change entirely depending on how it's used.

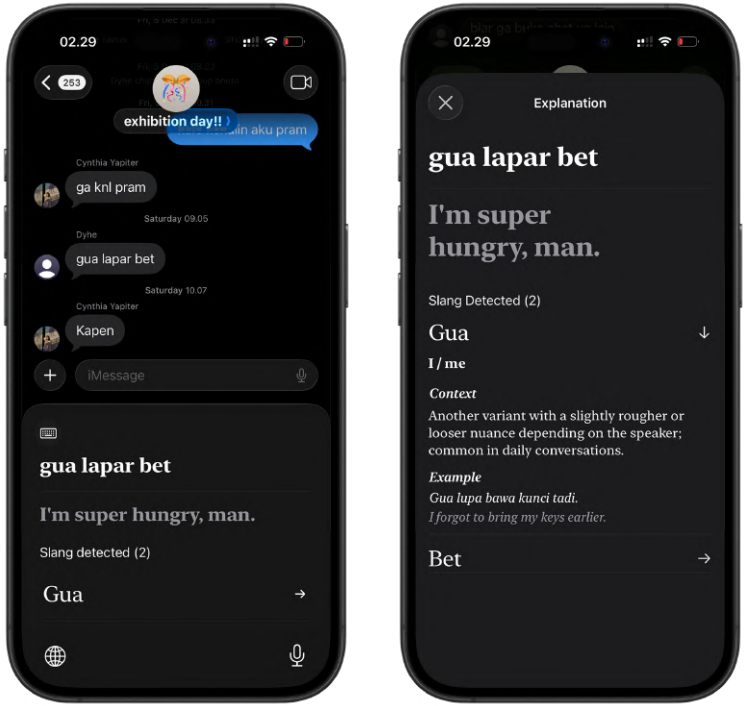
I built SLNG's data model to capture that nuance. Each slang has contextual variants, sentiment labels, and bilingual examples, paired with a matching algorithm that handles spelling variations and tone alignment. This lets SLNG detect not just the slang, but the meaning in that moment, making translations far more intuitive.



Designing a translator that captures nuance, not just definitions.

Instead of adding more features, I focused on what actually helps users interpret meaning in real conversations.

SLNG is built around multiple entry points, including the main app, a custom keyboard, and a system-wide Share Extension, ensuring slang can be understood wherever it appears.



Avoiding a flat UX by playing with motion, sounds, and haptics.

SLNG needed a personality, something more expressive than a static dictionary UI. I shaped the experience with motion, sound, and haptics to match the lively nature of slang.

Interactions use fluid spring animations, subtle haptic cues, and light audio touches that make actions feel tactile without becoming gimmicky. Everything follows iOS-native principles: responsive, intentional, and lightweight. The goal wasn't visual noise, but an experience that feels expressive and genuinely fun to use.

Each round of testing reshaped flow, interactions, and stability.

SLNG evolved through fast iterations shaped by real user behavior. While testers noted minor inconsistencies in gestures and transitions, a key insight came from foreign students who often heard slang without knowing how to spell it. This led to the addition of Speech-to-Text and a refined input flow for real-world voice conversations.

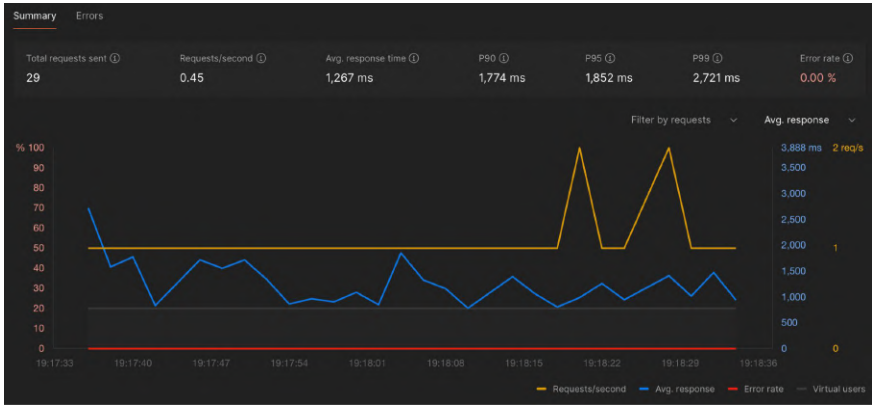
Iterating Based on Real Feedback

Migrating to Go to halve response times and stabilize concurrency.

As usage scaled, backend latency became a clear bottleneck. The initial Node.js prototype worked for early validation, but it struggled with consistent response times under concurrency.

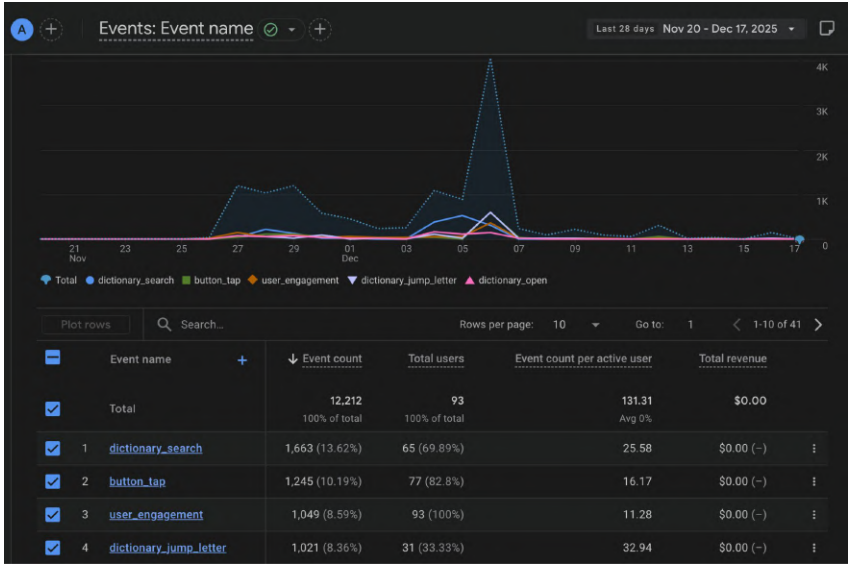
Rather than patching around the problem, I rebuilt the backend in Go. The migration enabled lightweight routing, stronger concurrency handling, and more efficient memory usage. Average response times dropped from ~3 seconds to ~1.5 seconds, effectively cutting latency in half.

This wasn't just an engineering improvement. The faster backend translated directly into a more fluid, responsive, and reliable user experience.



Tracing Real User Behavior with Firebase Analytics

To inform future product decisions, I used Firebase Analytics to trace which features users actively engaged with. Feature-level usage data revealed where users found real value, allowing me to prioritize iterations and focus development effort on areas that mattered most.



Tech Stack of **SLNG**

Swift Data

AVFoundation

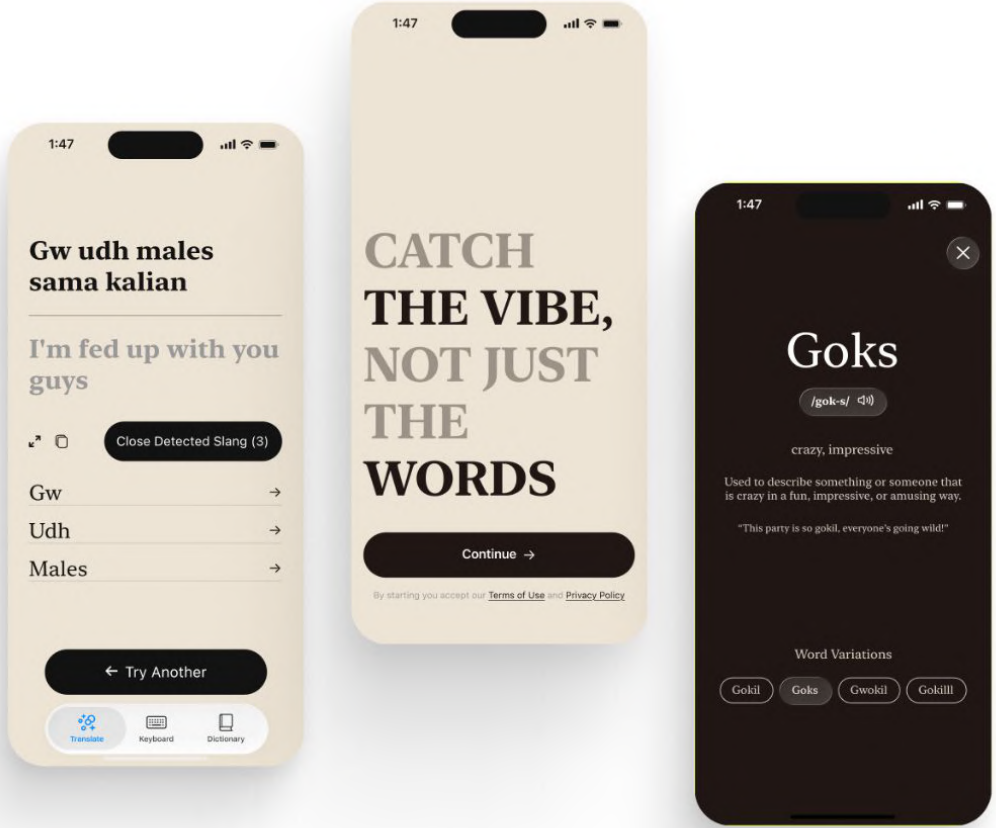
OpenAI API

Apple Speech

SLNG

Indonesia Slang Translator

Download on the App Store



Building Paintee: Crafting Face-Aware AR Experiences from First Principles

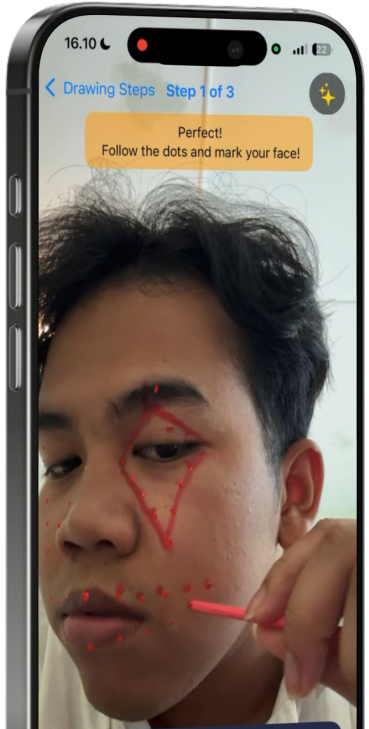
Paintee

An app to help beginner doing face painting who find it difficult to create proportional and symmetrical face painting

Beginner who apply face painting on their own faces often struggle with proportion and symmetry, especially when working from references on a flat surface.

Looking at it through a builder’s lens, the interesting problem wasn’t painting itself—it was how to design an experience that understands facial structure and guides precision in real time.

PROBLEM STATEMENT



Crafting Navigation Around a Continuous AR Session

Face painting is a continuous, hands-on activity, not a sequence of screens. Every interruption breaks focus, muscle memory, and spatial alignment.

In Paintee, navigation had to respect the session itself. Instead of resetting context at every step, the app preserves a single, continuous AR session while guiding users through structured phases—previewing, dotting, outlining, and painting. Screens change, but the session doesn't. This approach turns navigation into session control, allowing beginners to stay grounded in the same physical space while progressing with confidence and precision.

Voice Control as a First-Class Interaction

During face painting, hands are already busy and attention is split between a mirror, the brush, and facial precision. Asking users to constantly tap buttons or navigate UI breaks focus at the exact moment it matters most.

By allowing users to move between steps, toggle guidance, and progress through the experience using voice, the app removes unnecessary touch interactions. This lets users stay focused on applying the paint itself, keeping their hands, posture, and attention anchored to the task instead of the screen.

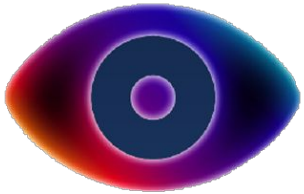
Nothing Extra Between the User and the Task

Paintee is structured as a single, linear session rather than a collection of screens. From design preview to guided application and finally capturing the result, each step advances the same underlying state instead of creating a new context. The AR session stays alive while the UI layers shift just enough to guide the next action. This keeps navigation lightweight and predictable, avoids unnecessary resets, and preserves spatial alignment throughout the process. By treating the experience as one continuous flow, the app removes friction from the interface and lets users stay focused on the physical act of face painting.

Tech Stack of **Paintee**

RealityKit

SwiftUI



Vision

ARKit

PAINTEE

Now available on **App Store!**

Download on the App Store

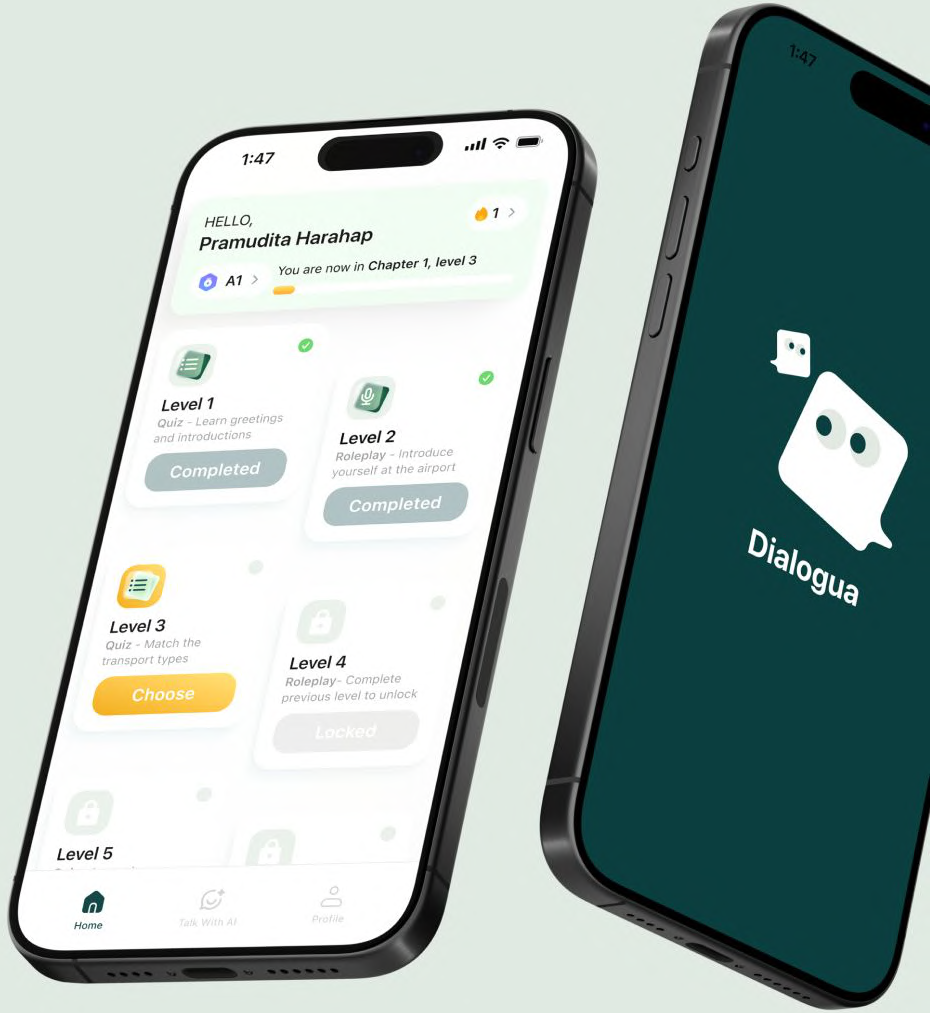
Beyond the **Case Studies.**

Dialogua

An app to help foreign students in Indonesia reduce language anxiety by providing step-by-step speaking practice in realistic social scenarios, so they can build confidence, progress from simple to complex interactions, and track their improvement.

Role: iOS Developer

Try It Now



Miruu

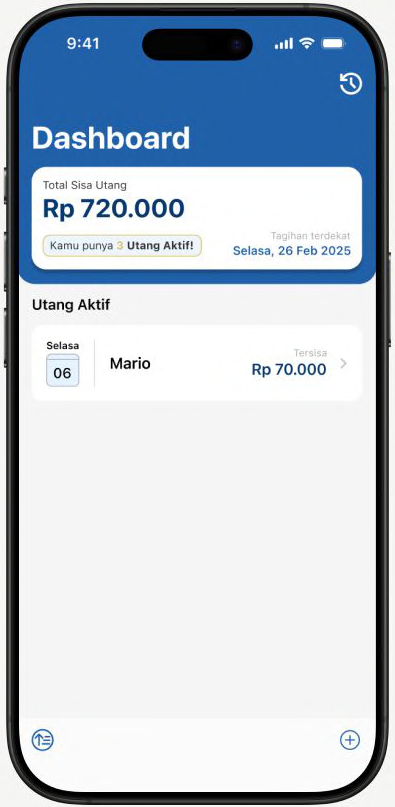
An app to turn your daily life into an RPG adventure, helping you build habits, grow skills, and track personal progress through quests, XP, and evolving characters.

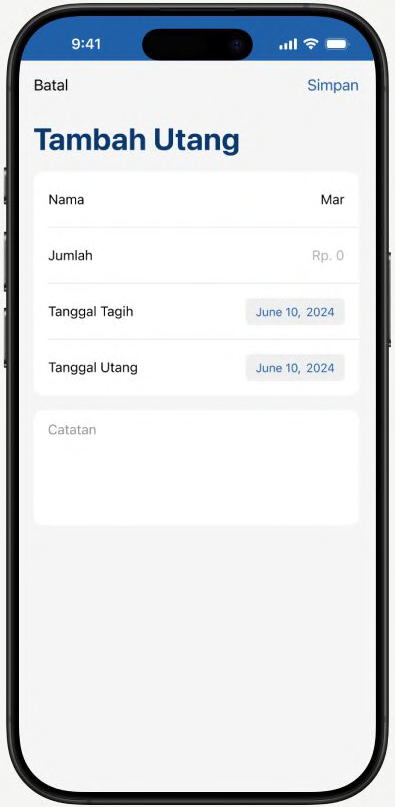
Role: iOS Developer

FreeUtang

An app to manage and track your receivables

Role: iOS Developer





Built by curiosity. Proven by shipping.

Open to iOS developer and engineering roles

linkedin.com/in/ikhwanpramuditha/

github.com/prammmoe

+62 812 4701 6022

medium.com/@prammmoe