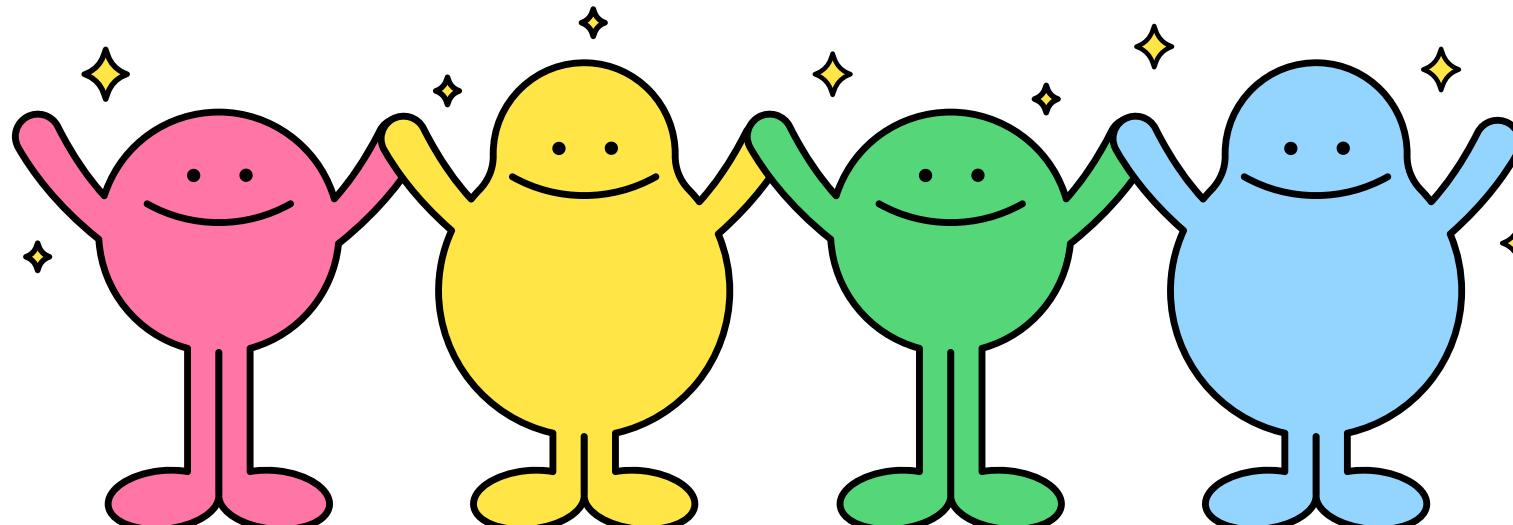




English Word Learning with Vision Object Recognition

Chalkak



21102042 Seungwoo Baik
21102052 Jeongyun Lee
21102043 Suyong Sim

Contents

01 Background & Motivation

02 Role & Responsibilities

03 App Features

04 Demonstration Video

05 Technical Methodology



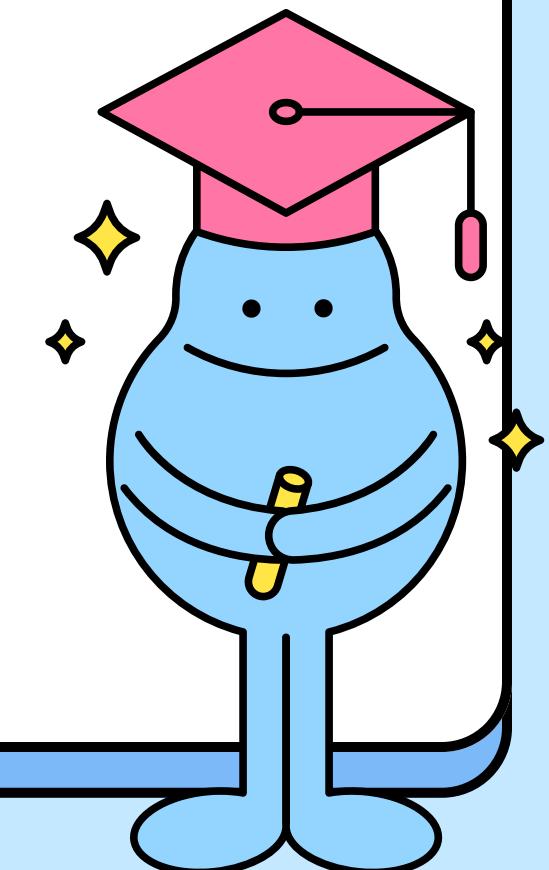
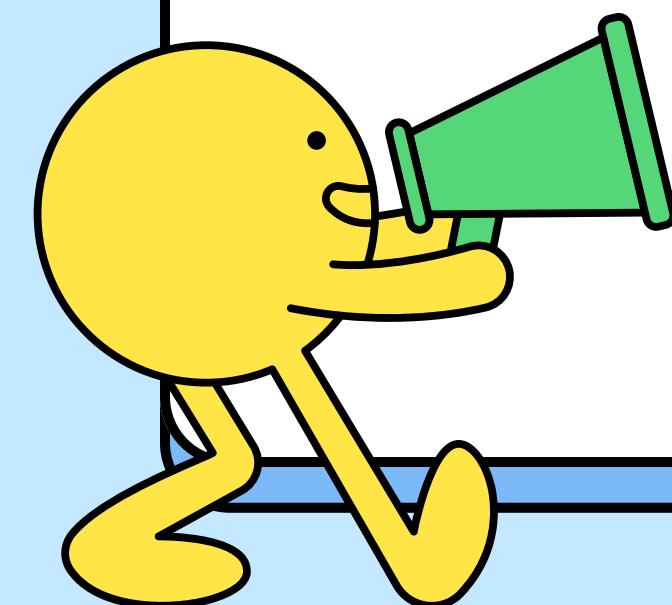
Background & Motivation

Learning English words through pictures

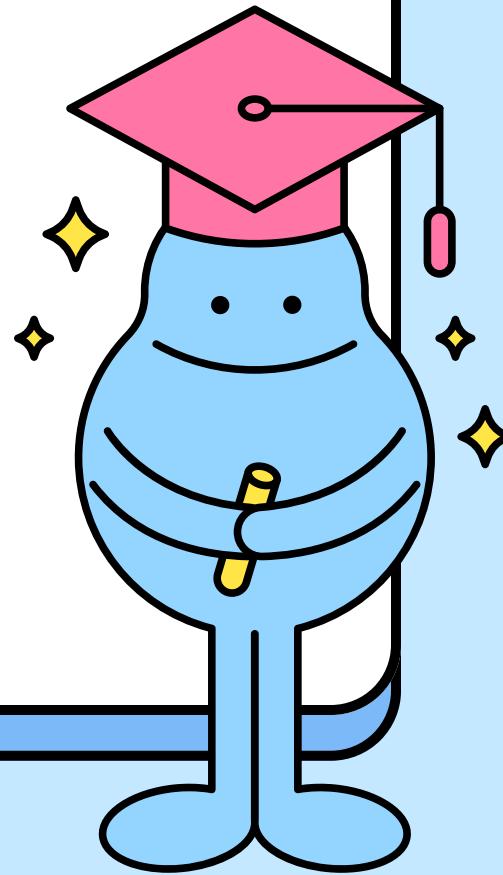
Users directly generate learning materials

System for continuous learning and review

Learn anytime, anywhere using smartphones



Role & Responsibility



Baik Seungwoo:

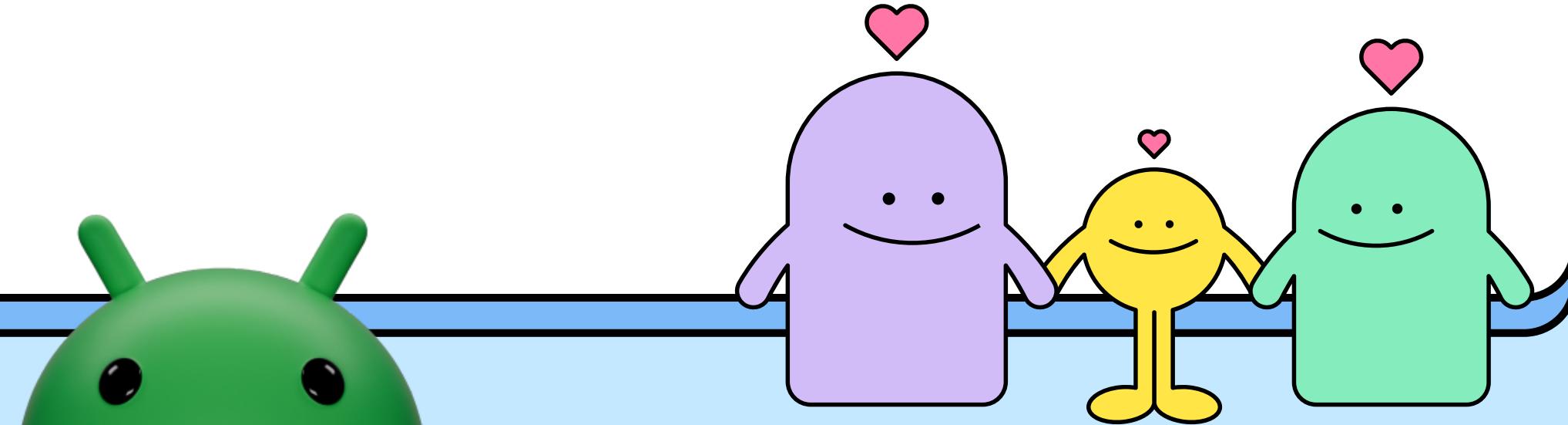
- Object Detection Model(Integrating & Using with camera/file system)
- TTS Functionality
- Sensor(accelerometer) Functionality

Sim Suyong:

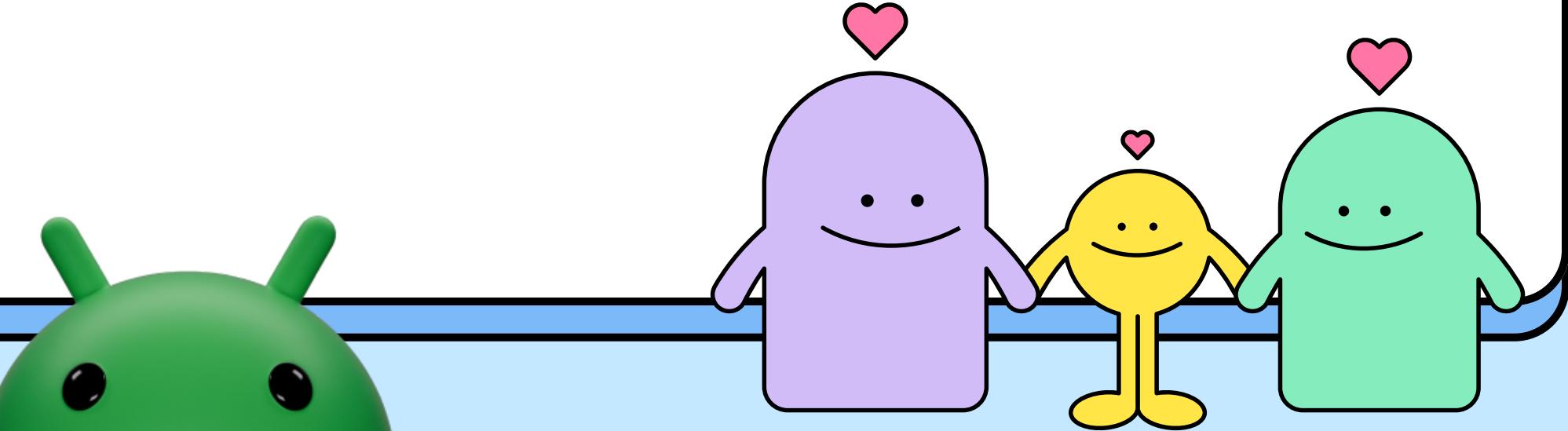
- Local RoomDB
- Open API for using ChatGPT API
- Google Firebase (Cloud server, Authentication, Notification)

Lee Jeongyun:

- UI/UX Implementation
- STT Functionality

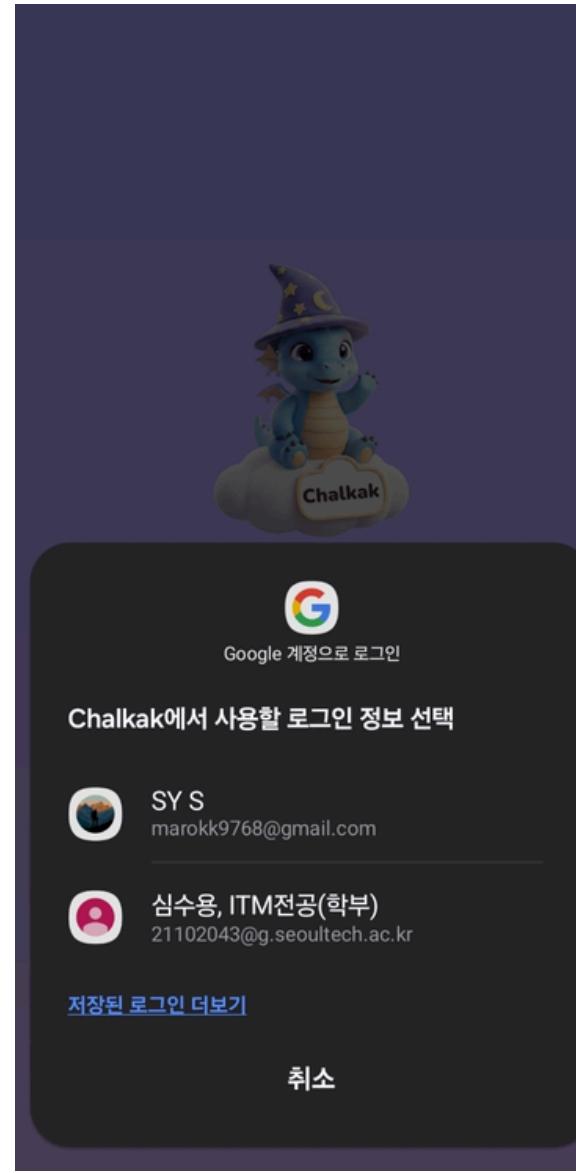
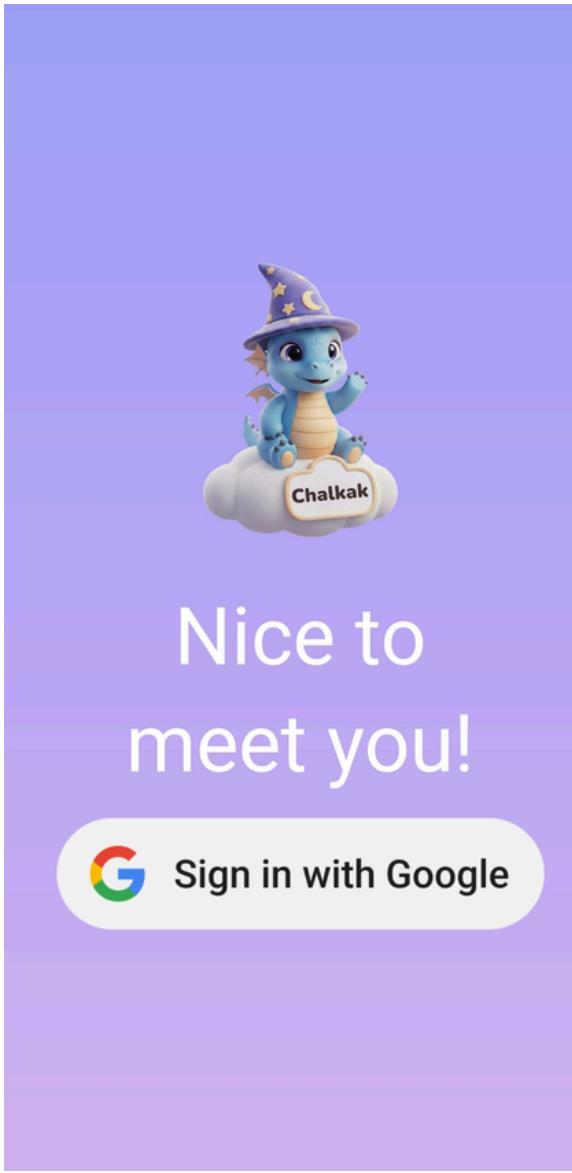


App Features





Login / Register



Architecture

- Auth: Firebase Auth
- User DB: Firestore

Steps

1. Start login via Credential Manager / Google Sign-In
2. Authenticate with Firebase
3. Retrieve FCM Token
4. Save user info & token to Firestore
5. Go to main screen & update local/session state





Home Screen

Welcome back, protruser

Magic Adventure

Learning Activity • 1 days streak

Learned Not Learned

Learning Calendar

12월 2025

일	월	화	수	목	금	토
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Learned Not Learned

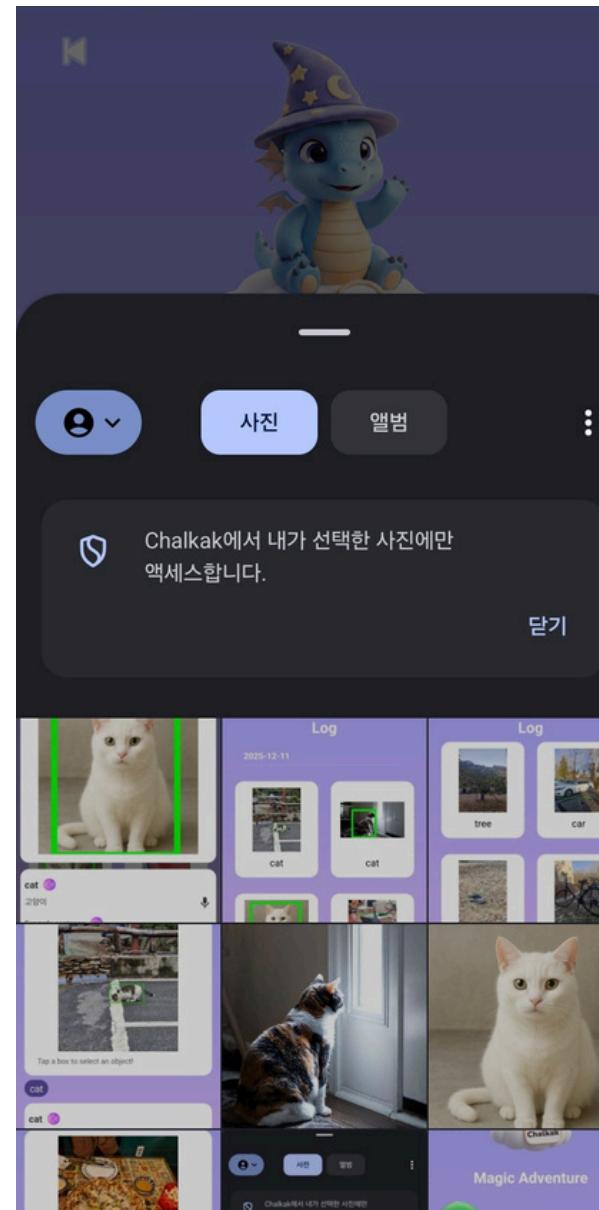
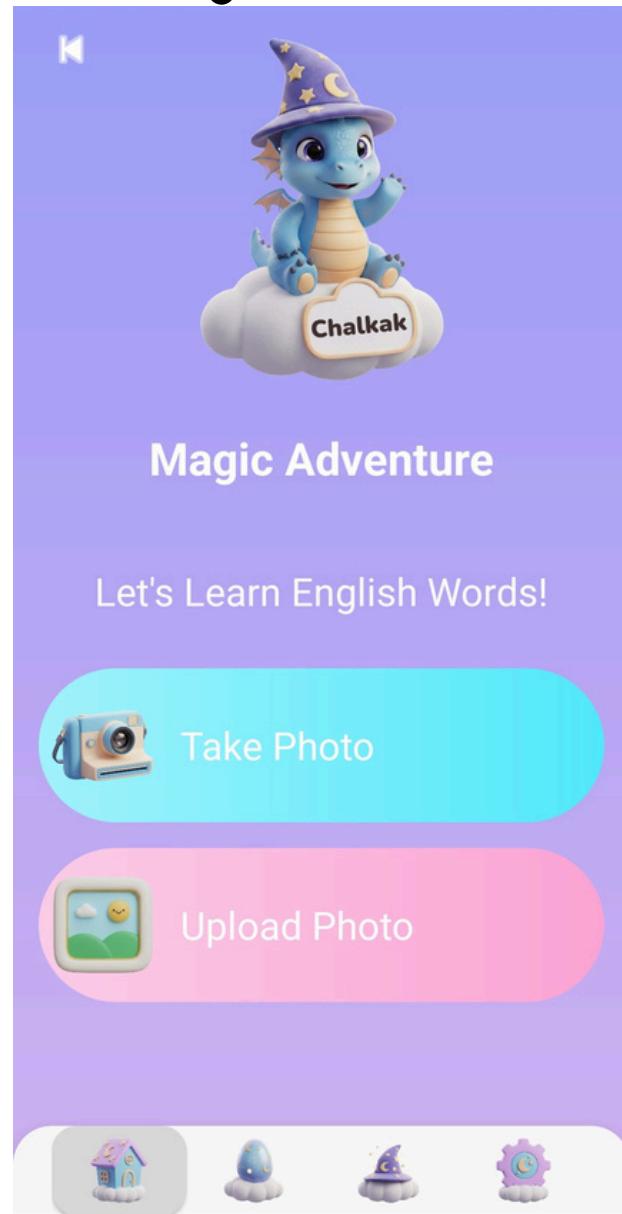
Functionalities

- Greeting with user nickname
- Access to Magic Adventure (image-based vocab)
- Last 7-day learning streak
- Calendar showing learned / not learned days
- Quick Snap access point





Learning English Words w/ Pictures (1/2)

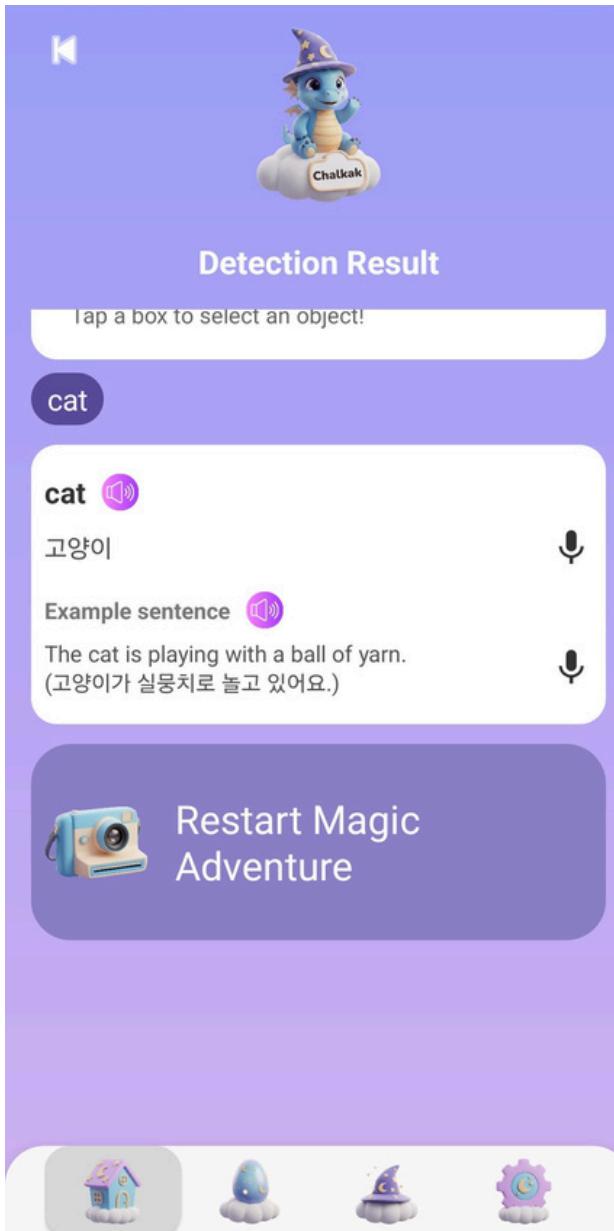
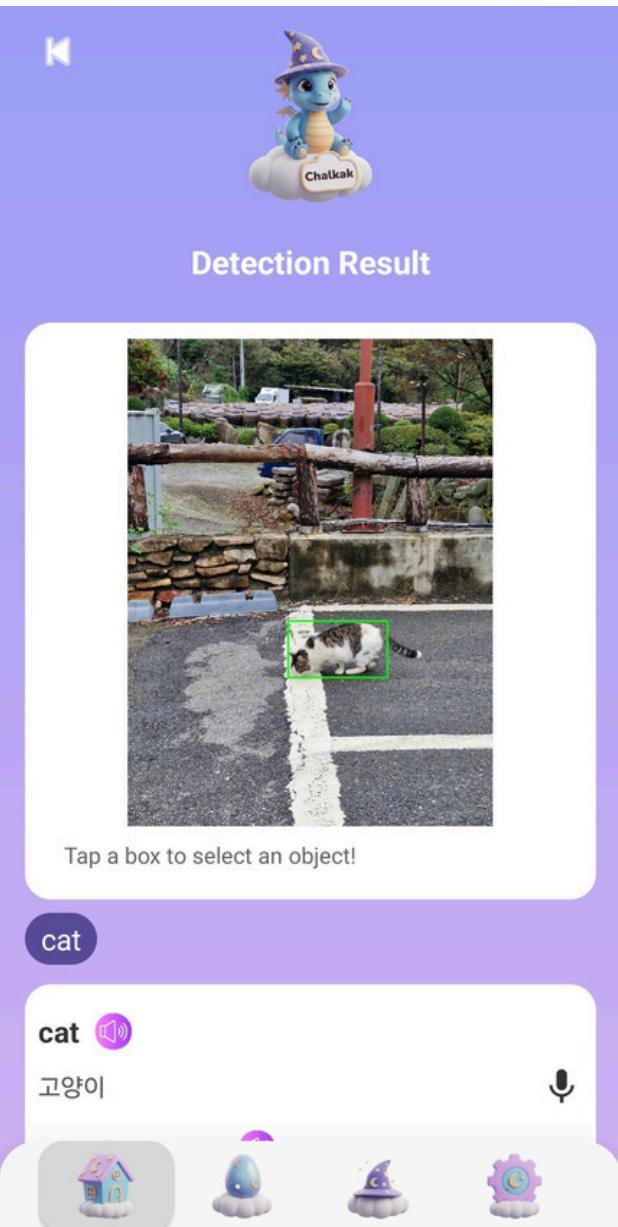


Functionalities

- Select a image by choosing Camera or Gallery Access.
- Utilize a EfficientDet-Lite0 object recognition model.



Learning English Words w/ Pictures (2/2)

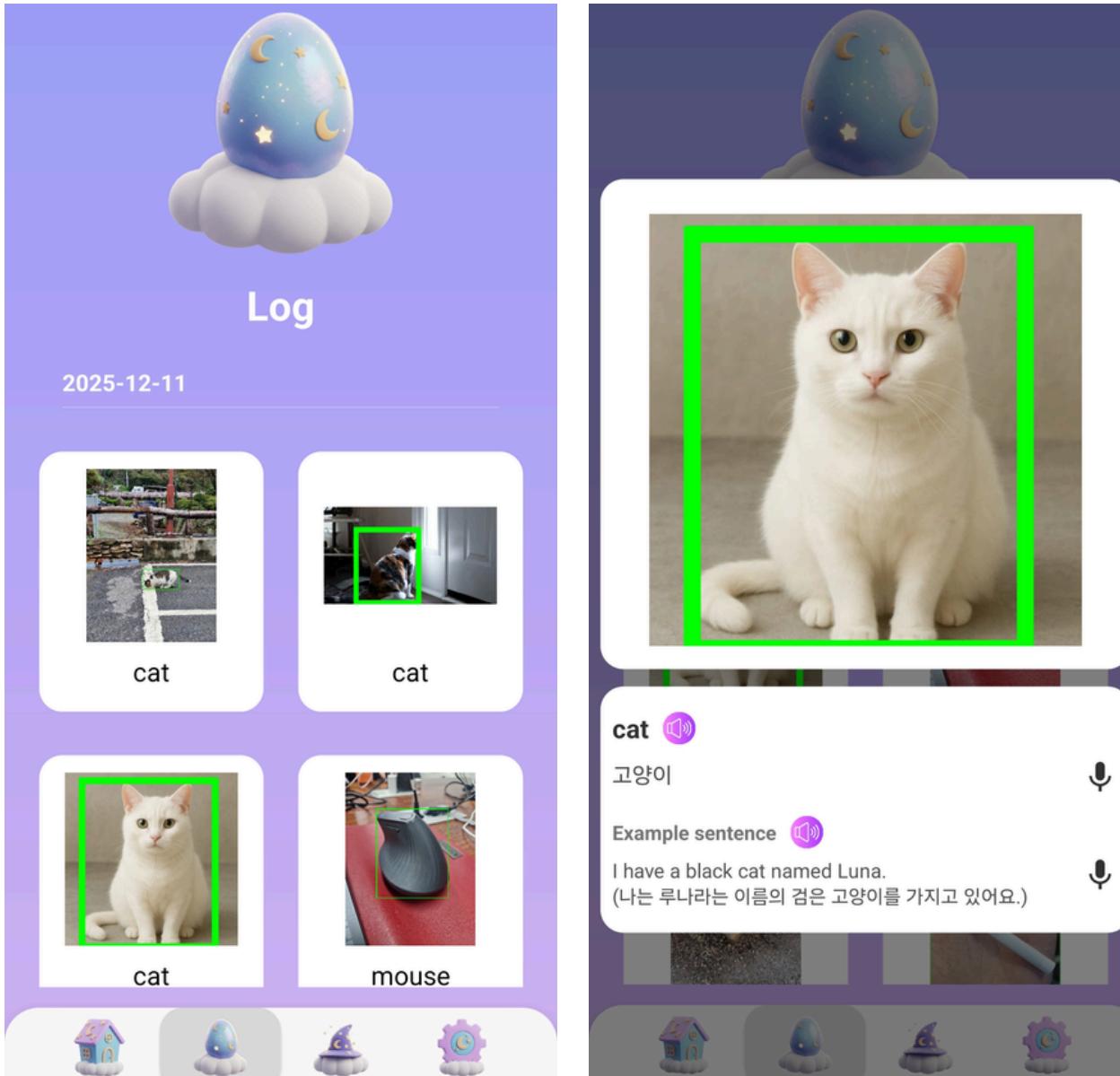


Functionalities

- Display the recognized object as a Box on the original image.
- Provide word and an example sentence with Korean meaning.
- Provide the pronunciation of the word using TTS.
- Provide user's accuracy of their pronunciation using STT.



Log



Functionalities

- Vocabulary-photo mapping & automatic word card generation
- Multi-word detection → multi-card creation
- Grid-view cards with date-grouped organization
- Card tap → meaning + example sentences
- TTS/STT for pronunciation and speaking practice





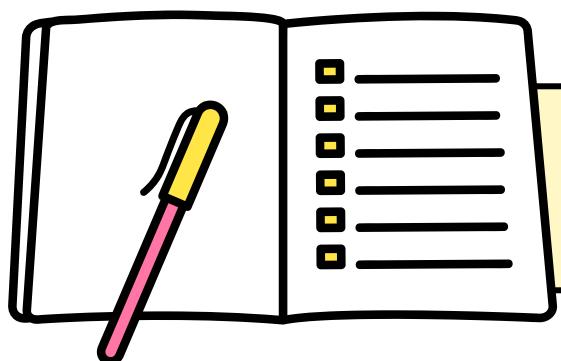
Quiz

The screenshot shows a mobile application interface. At the top left is a purple cloud icon with a wizard's hat containing stars and a crescent moon. Below it is the text "Quiz Time". A purple star icon indicates "Today's Magic Quiz" with the note "The magic hat has prepared it!". Below this, the text "18 words" is displayed, followed by "Magic spells that need review". At the bottom, there are three metrics: "100% Review Rate", "18 Total Learning", and "1 days Consecutive Days". A large purple button labeled "Start Magic Quiz" with the subtext "Review 18 magic spells" is centered at the bottom. To the right, a quiz card for question 8 is shown, featuring a potted plant image, the word "potted plant" with a microphone icon, its Korean meaning "화분에 심어 키운 식물", an example sentence "I like to decorate my balcony with colorful potted plants.", and its Korean translation "(나는 다양한 색상의 화분에 심어 키운 식물로 발코니를 장식하는 것을 좋아한다.)". Below the card are four options: "black", "mouse", "potted plant" (which is highlighted with a green checkmark), and "dining table". A progress bar at the top of the card shows "8 / 18" and "44%".

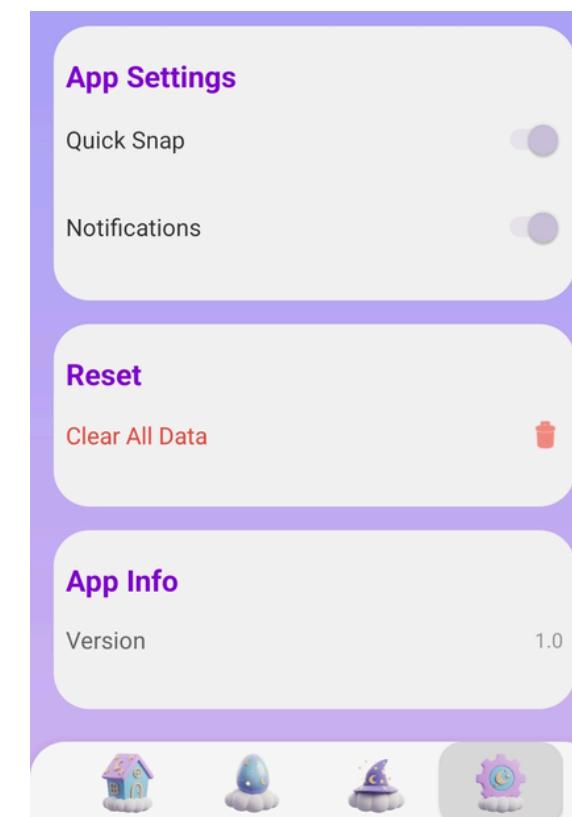
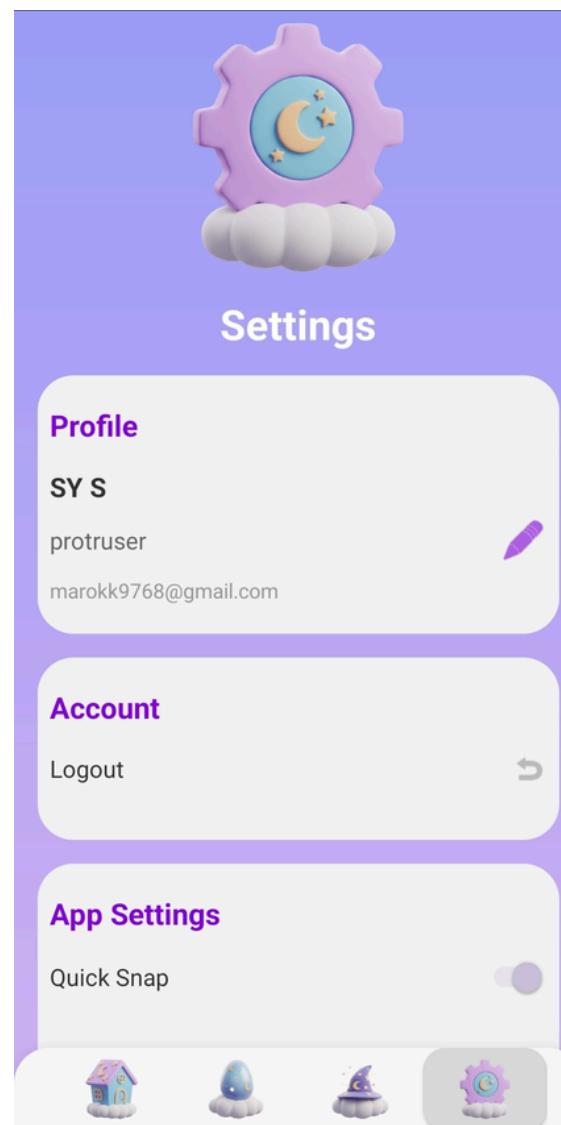
Functionalities

- Daily target vocabulary count
- Quiz attempt count & overall review rate
- Consecutive quiz-day streak
- Four-choice quiz format with instant feedback
- Correct answer → word, meaning, example sentence, TTS/STT
- Progress indicator at the top during quizzes





Settings

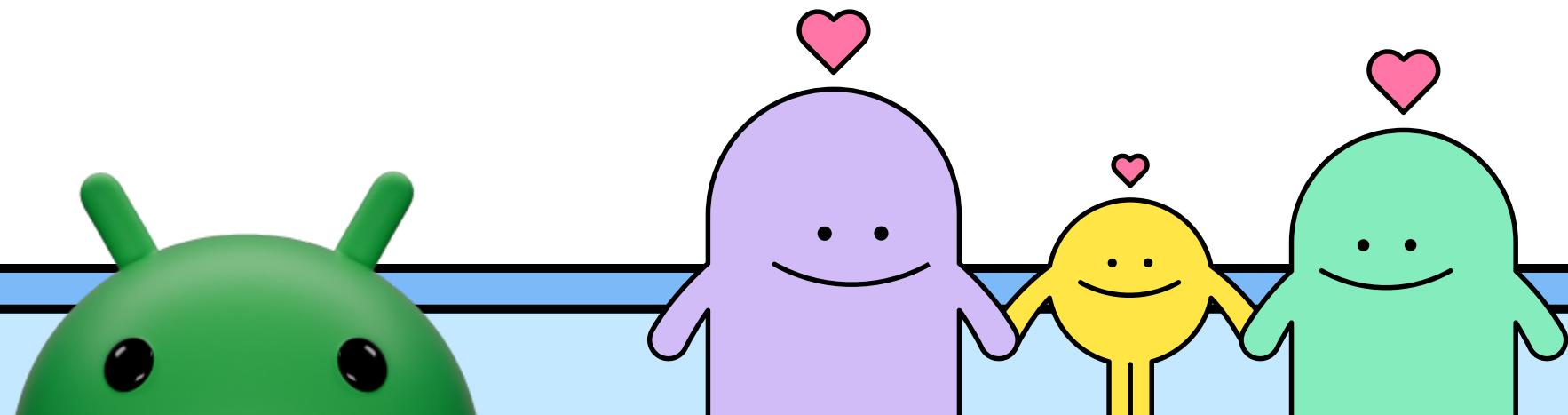


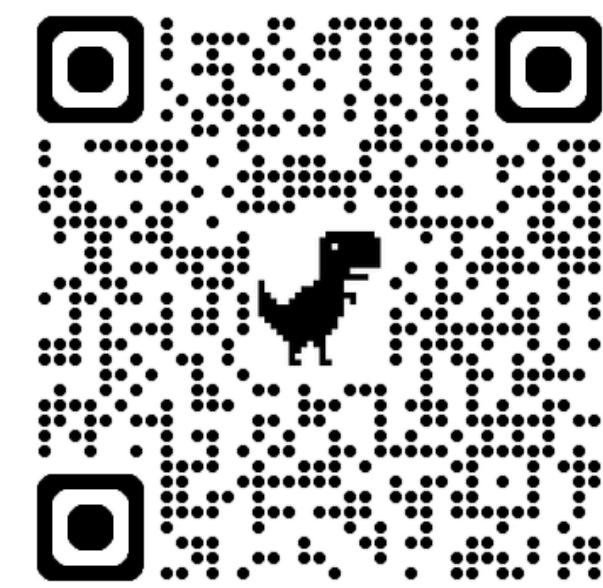
Functionalities

- Nickname customization
- Logout option
- Quick Snap toggle
- Notification toggle
- App data reset function

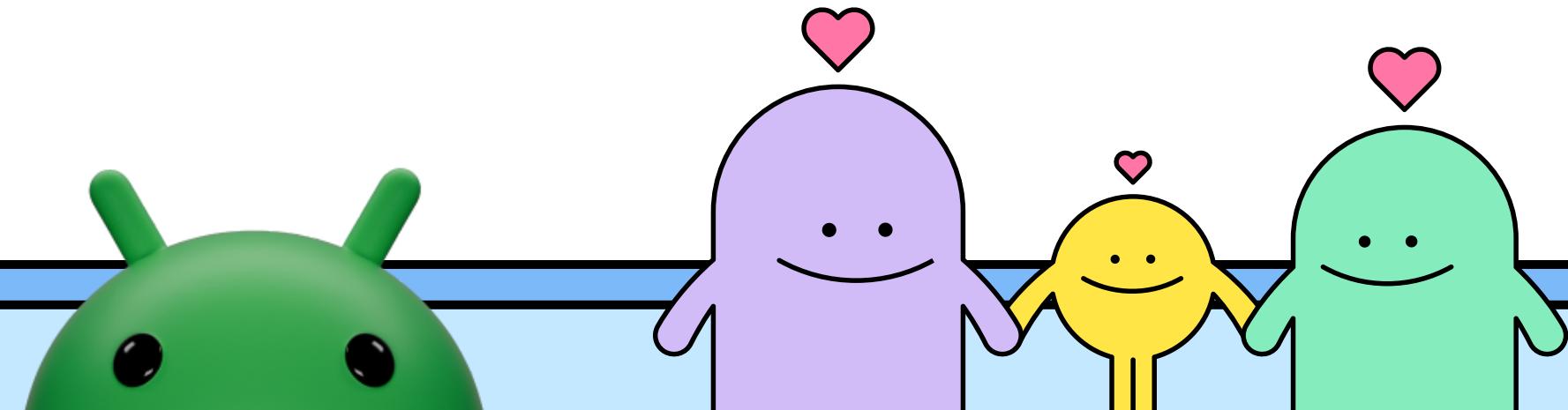


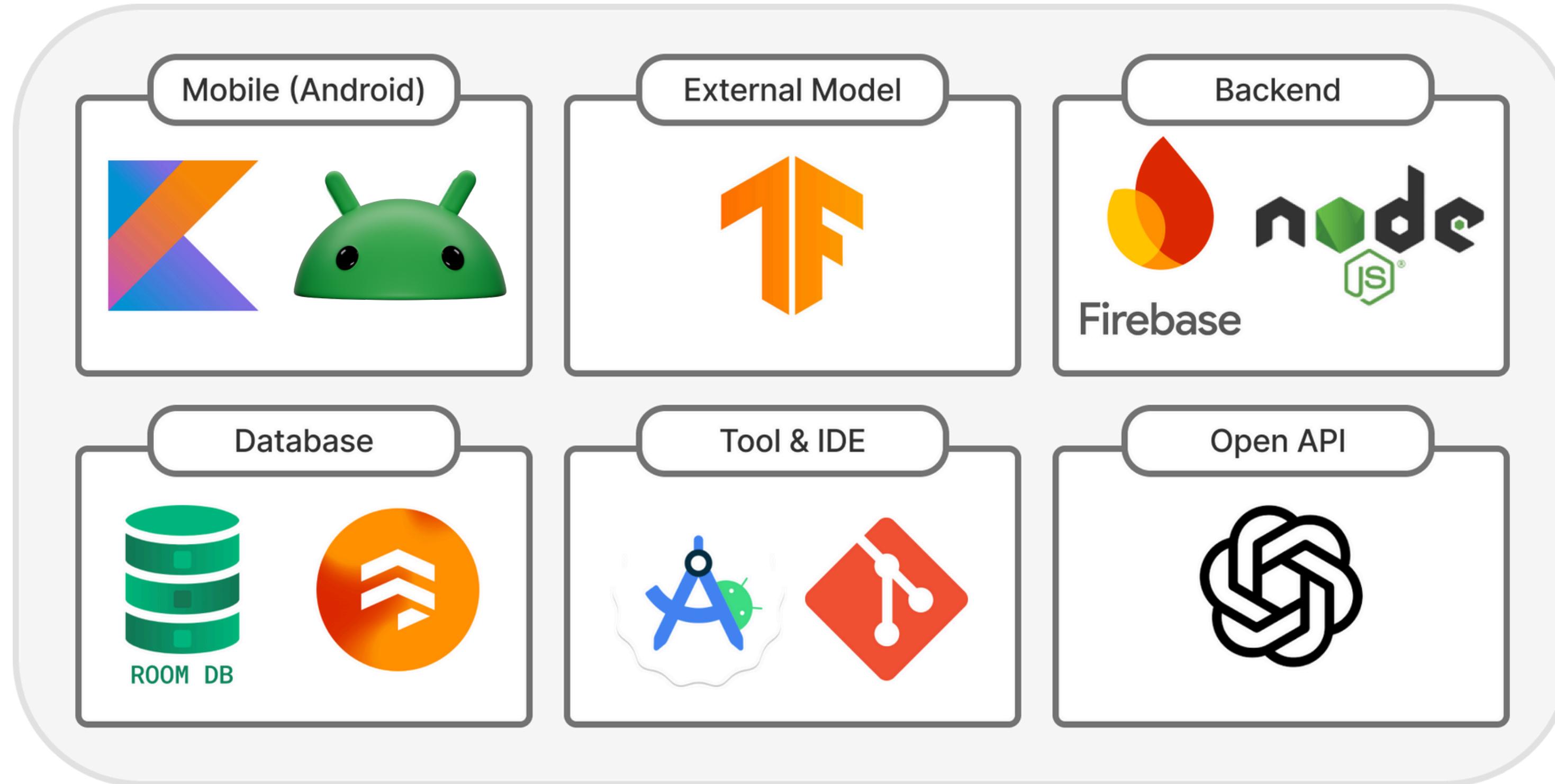
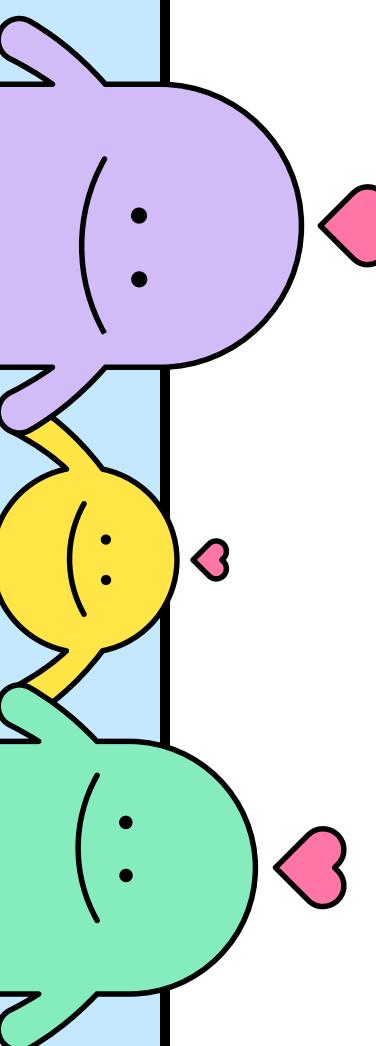
Demonstration Video



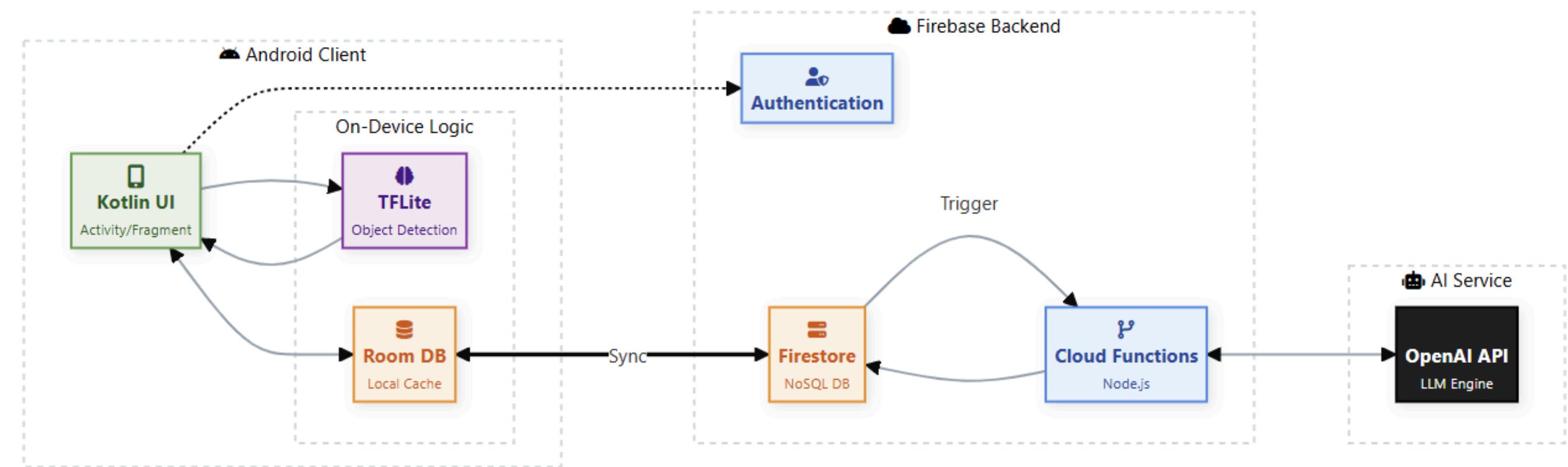


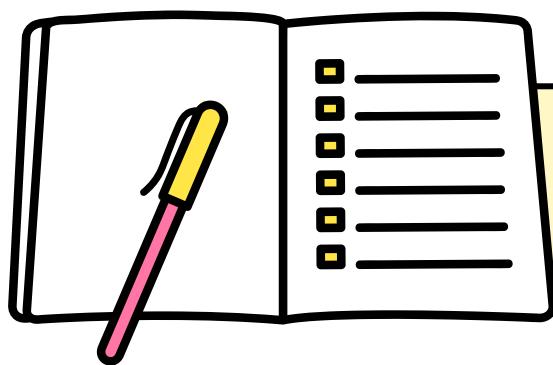
Technical Methodology



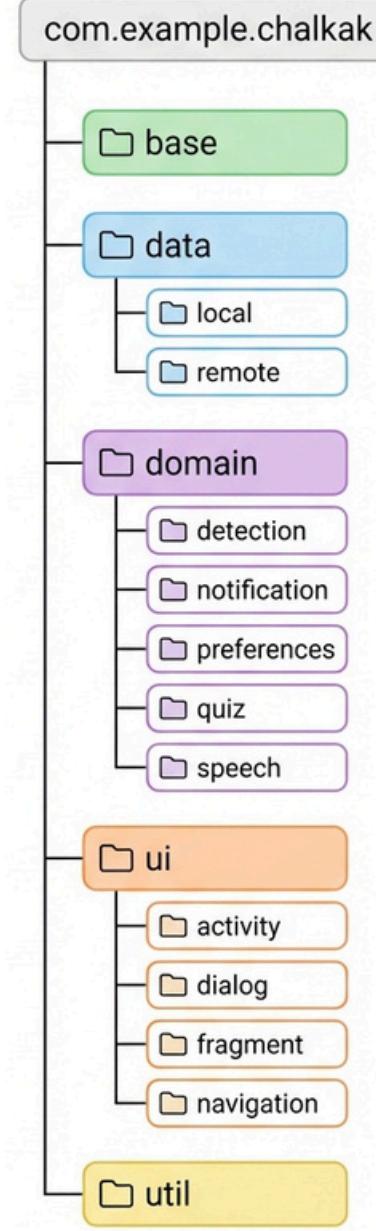


Architecture Flow





Mobile (Android) - File Structure



File Structure

- **base/**: Collection of common base classes or shared abstractions
(Providing common logic for Fragments).
- **data/**: External and internal data source layer.
 - **local/**: On-device storage like Room.
 - **remote/firestore/**: Network/backend integration (e.g., Firestore).
- **domain/**: Core domain logic and feature-specific helpers. Non-UI logic such as Detection, Notification, Settings, Quiz, and Voice.
- **ui/**: Screen layer (User Interface).
 - **activity/**, **fragment/**, **dialog/**, **navigation/**
- **util/**: General-purpose utilities (e.g., Image loading/selection, Sensor, Toast, Data loading, etc.).





Mobile (Android) – UI Architecture

App Architecture

- Uses a classic single-activity, multi-fragment architecture.
- MainActivity acts as the main container: bottom navigation, global UI, and window insets handling.
- Core screens like Home, Log, Quiz, Settings, and Magic Adventure are implemented as Fragments.

UI Pattern

- Activity: Shell of the app, global navigation, and external entry points (e.g., main screen, image preview).
- Fragment: Main screen units under bottom navigation, plus swappable sub-screens.
- DialogFragment: Temporary overlay for details or secondary actions (e.g., full-screen word detail with TTS and speech input).
- Built with XML layouts + classic View system + findViewById.
- Screen transitions use manual FragmentTransactions.





TTS

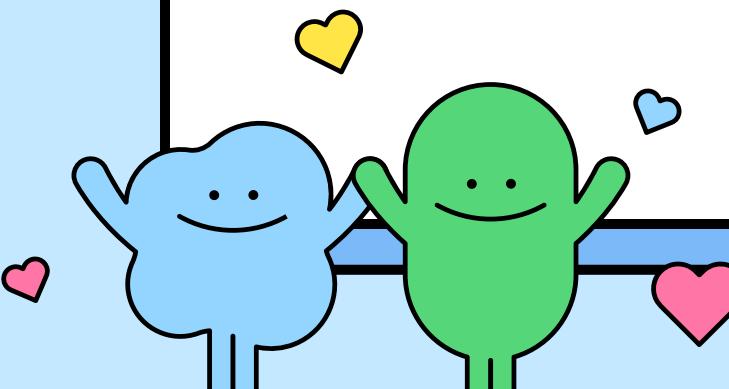
- Use Android's built-in TextToSpeech to set up an English voice (slower rate).
- Wire the word/example TTS buttons on the card view to init.
- On screen destroy, stop and shut down the TTS engine to release resources.

Mobile (Android) – TTS / STT



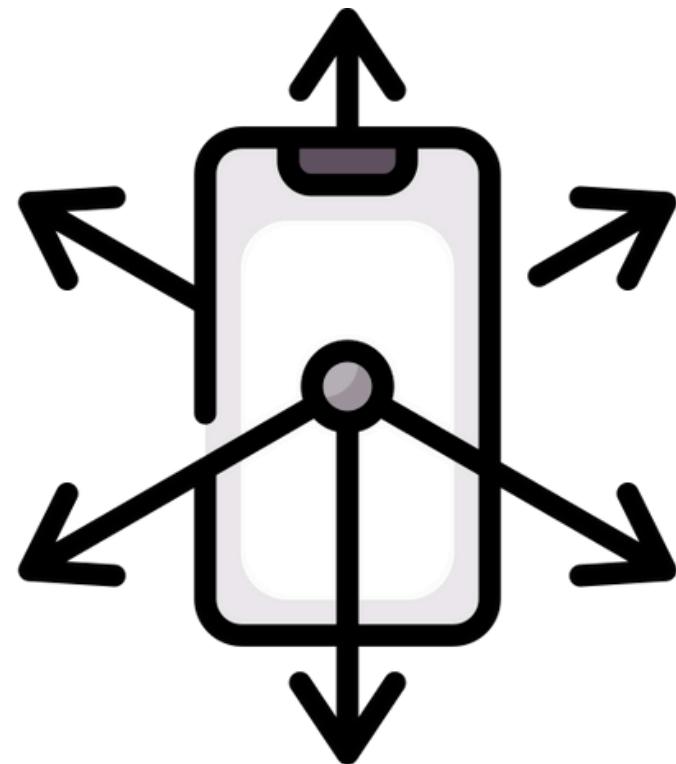
STT

- Use Android's built-in SpeechRecognizer and check mic permission
- If granted, start listening with an english - -US intent.
- Grab the two mic icons (word/example) and the texts (selected word, example) on the card on button tap, toggle listening and swap the icon between mic and stop.
- When recognition returns, compare it with the original English word and show whether it matches.

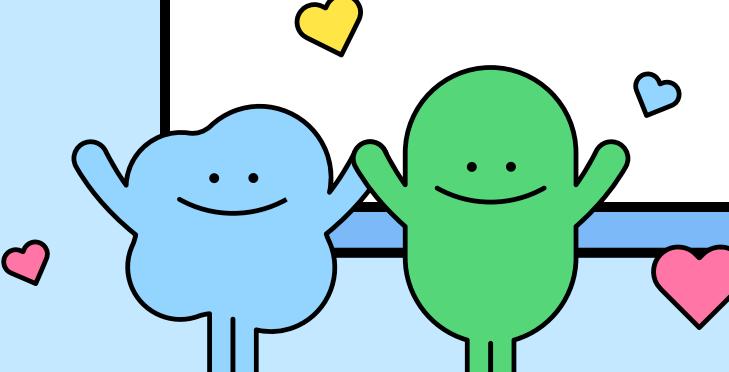




Mobile (Android) - Sensor



- Sensor Registration
 - Register listener to accelerometer via SensorManager.
- Acceleration Measurement
 - Calculate G-Force from X, Y, Z axis values.
- Shake Detection
 - Trigger when G-Force > 2.0 G
- Camera Launch
 - Execute callback to open camera screen.





Mobile (Android) - Camera Access

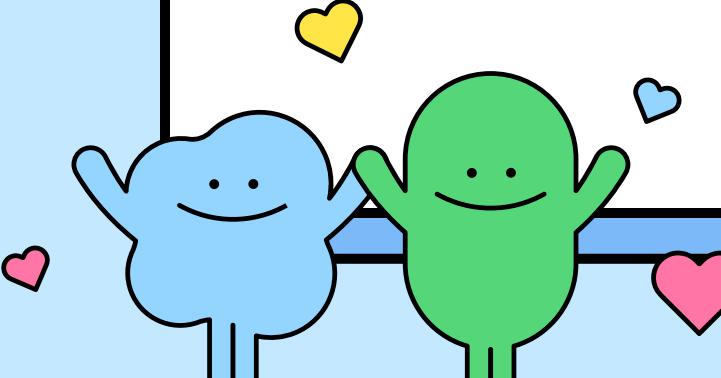


Common Mechanism: URI-based Flow

- Uses the Activity Result API: call external component → get result via callback.
- Both camera and gallery ultimately return an image Uri.
- Downstream screens load and display the image using this Uri.

Camera Flow

- The app pre-creates an empty image file in its internal cache directory.
- That file is wrapped as a content:// Uri via FileProvider and passed to the camera app.
- The camera writes the captured photo directly to that Uri location.
- On success, the callback receives the already-saved image Uri.
- Because it lives in the app's internal cache, the app can access the photo without relying on the system gallery.





Mobile (Android) - Gallery Access

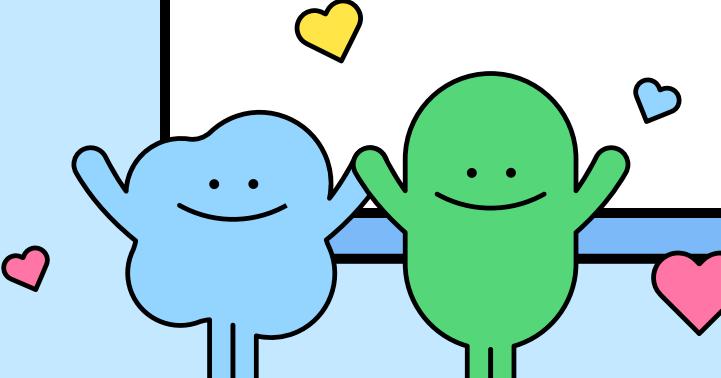


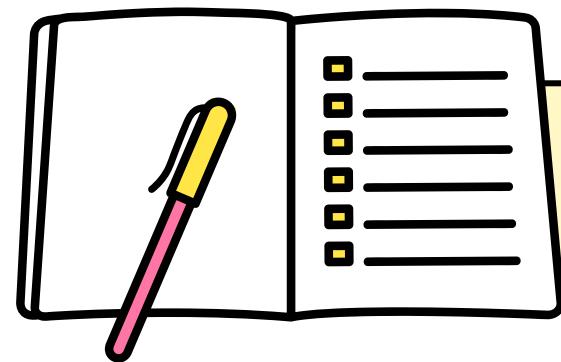
Common Mechanism: URI-based Flow

- Uses the Activity Result API: call external component → get result via callback.
- Both camera and gallery ultimately return an image Uri.
- Downstream screens load and display the image using this Uri.

Gallery Flow

- A system picker is launched in “content request” mode (e.g., `GetContent("image/*")`).
- The OS returns a Uri that represents the image selected by the user.
- The app gets read access to that Uri without needing the actual filesystem path.
- Conceptually, the app only obtains a handle to an already existing file and uses it by reference.





Mobile (Android) – Calendar View



Uses Material CalendarView to visualize learning activity with dots on each date.

Fetches learning logs from a Room database and aggregates them by date.

Date Status Visualization

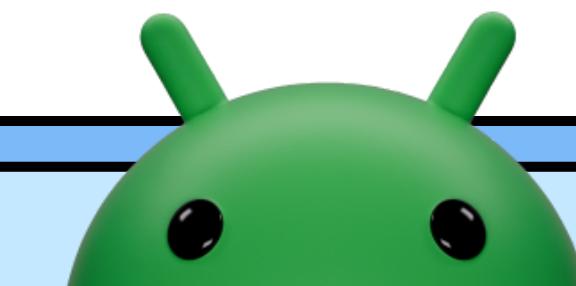
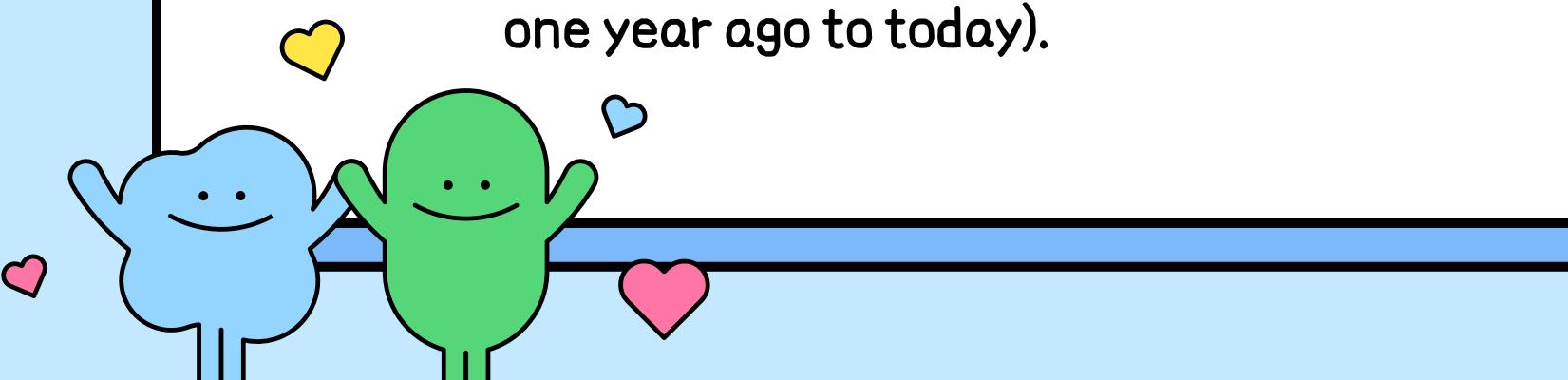
- Green dot: days with learning activity based on photo logs.
- Red dot: days with no learning between the first learning date and today.
- Calendar range is limited to the last 1 year (from one year ago to today).

User Interaction

- When a user taps a date, the app checks if it's a "learned" or "not learned" day.
- Shows a toast message: "Completed" or "No learning" accordingly.

Activity Metrics & Streaks

- Reuses the same learning logs to calculate streaks and recent activity.
- Displays current streak, recent 7-day activity, and small activity dots in the header area.





External Model (1/2)

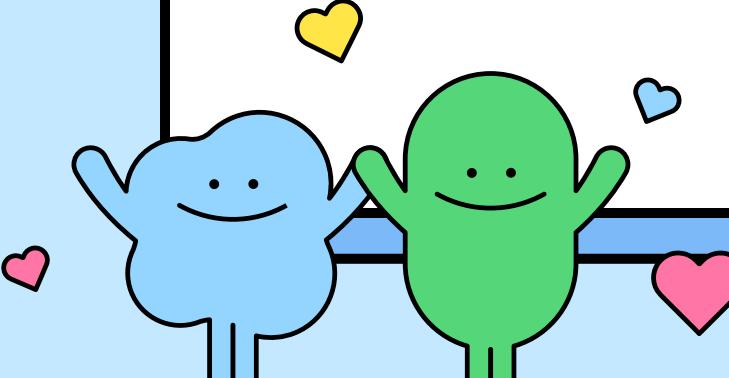


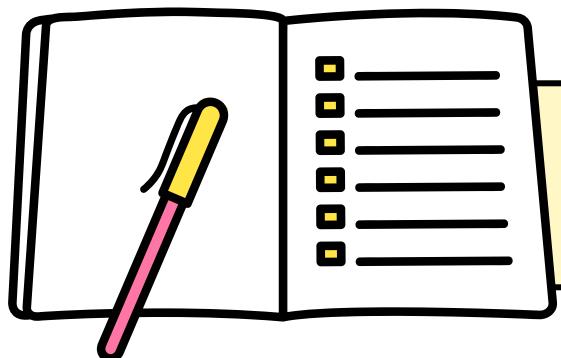
Tensorflow - EfficientDet Lite0 model

Pre-trained TensorFlow object detection model trained on the COCO dataset.



- Model Loading & Utilization:
 - The model is loaded and utilized by using the methods of the classes provided by TensorFlow.
- Helper Class Implementation:
 - A dedicated Helper class was implemented for model loading and initialization.
 - Maximum number of objects to recognize: 2
 - Minimum confidence threshold for object recognition: $0.3f$

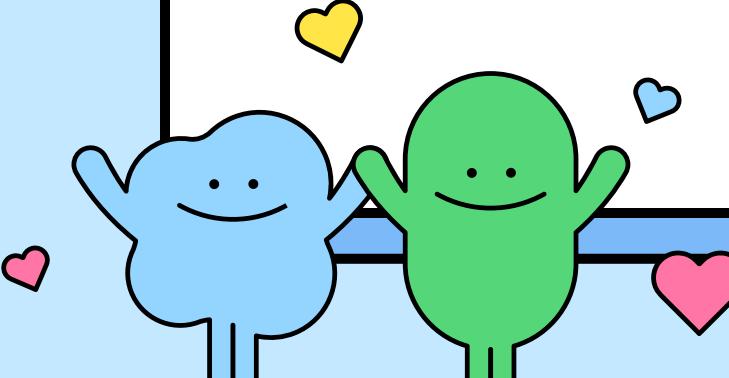


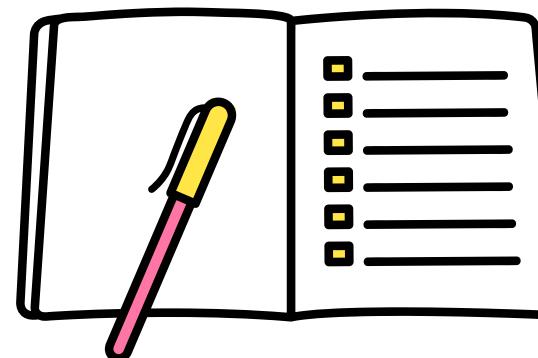


External Model (2/2)



- Model Output/Results:
 - The model returns results as a list containing the following information for each detected object:
 - Recognized Object (Class Label)
 - Confidence Score
 - Position of the object within the image
 - The image with a bounding box drawn around the detected object's location.
- Data Processing and User Delivery:
 - The returned results are stored in the database (DB).
 - The results are then provided to the user as part of the results screen.





Backend(1/2)



Authentication Flow (Google OAuth 2.0)

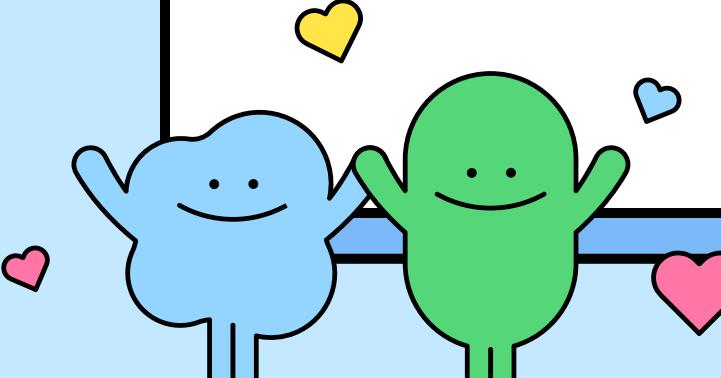
Flow: Credential Manager → Google ID Token → Firebase Auth → Session

Mechanism:



Firebase

- Credential Manager: Initialize an OAuth request through Android Credential Manager API with Authentication Intent
- Google ID Token: Receive Custom Credential result from the system, extract OAuth 2.0 token, and prepare server-side verification
- Firebase Auth: Convert ID token into Google Auth Provider credential and let Firebase Authentication validate the token signature with Google server
- Session: After verification, send UID and email to firestore and make session



Backend(2/2)



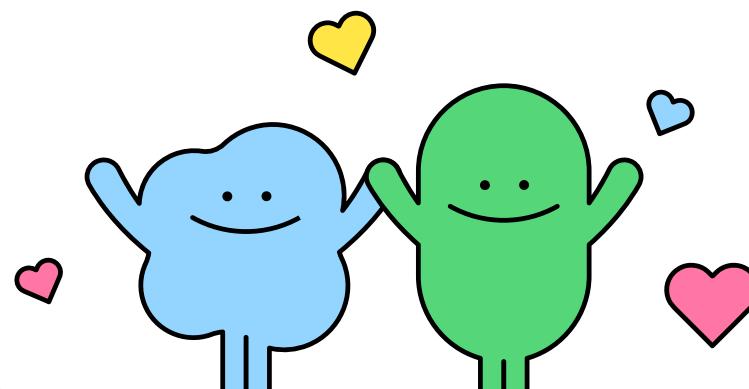
Push Notification

Flow: Token Generation → Firestore (Mapping) → Scheduler/Event → User Device mechanism:



Firebase

- Token Generation: App startup triggers the FCM SDK to create a unique device registration token.
- Firestore (Mapping): The token is saved to the user's Firestore document to bind the account to the device.
- Scheduler / Event:
 - Local: System alarms fire reliably even in Doze(절전 모드)
 - Remote: Server looks up the token and targets the specific device
- User Device:
 - Local: A system handles the alarm and shows the notification
 - Remote :A background service processes FCM payloads and wakes the app



Database



Local Database

Android Room

PhotoLog

TABLE (PARENT)

photoId	Long (PK)
localImagePath	String
createdAt	Long (Timestamp)

↓ 1 : N

DetectedObject

TABLE (CORE)

objectId	Long (PK)
@parentPhotoId	Long (FK)
englishWord	String (Index)
koreanMeaning	String
boundingBox	String "[l,t,r,b]"
lastStudied	Long

↓ 1 : N

ExampleSentence

TABLE (CHILD)

id	Long (PK)
@wordId	Long (FK)
sentence	String
translation	String

Sync Key: "English Word"



Remote Database

Cloud Firestore (NoSQL)

users

COLLECTION

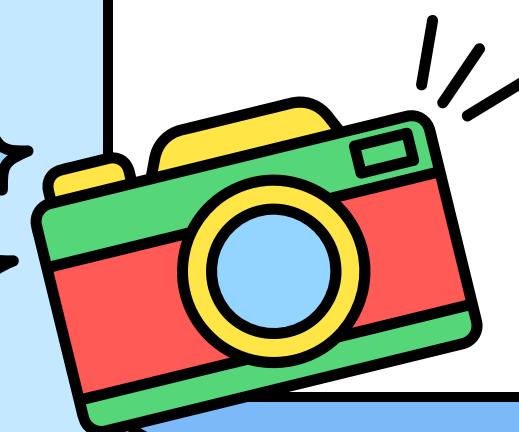
Document ID	Auth UID
email	String
nickname	String
fcmToken	String
lastStudiedAt	Timestamp/String
stats	Map { totalWordCount: Int }

words

GLOBAL CACHE

Document ID	English Word (e.g. "apple")
originalWord	String
meaning	String
examples	Array of Maps

- sentence: String
- translation: String





Tools and IDE



Android Studio

- Development based on Android 13.0, specifically targeting the S23 FE.



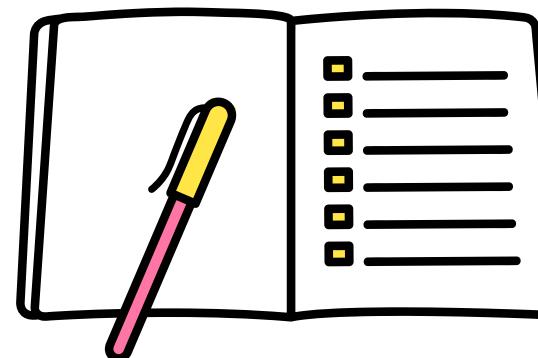
Gradle

- Control the dependencies to integrate TensorFlow Lite, Room, and Firebase

Git

- Git setup for version control and collaboration
- Integration with a GitHub Organization
- Collaboration is managed through branches and GitHub Issues.
 - <https://github.com/chalkak-itm>
 - <https://github.com/chalkak-itm/chalkak-app.git>





Open API



GPT Integration

flow: Client App → Cloud Function → GPT-3.5-turbo → Client/DB

mechanism:

- Secure Endpoint: Implemented as a Firebase Function
- JSON Enforcement: Parse the response directly as strict JSON structure
- Automated Sync: The parsed data is persisted to RoomDB and Firestore simultaneously via Dual-Write





Open API (JSON Format)



System : You are a helpful English-Korean dictionary assistant. Provide Korean translations and example sentences in JSON format.

```
{  
    "meaning": "고양이",  
    "examples": [  
        {  
            "sentence": "I have a black cat named Luna.",  
            "translation": "나는 루나라는 이름의 검은 고양이를 가지고 있어요."  
        },  
        {  
            "sentence": "The cat is playing with a ball of yarn.",  
            "translation": "고양이가 실뭉치로 놀고 있어요."  
        },  
        {  
            "sentence": "She loves to cuddle with her cat on the couch.",  
            "translation": "그녀는 소파에서 그녀의 고양이와 껴안는 것을 좋아해요."  
        }  
    ]  
}
```





Thank you
for listening



TEAM Chalkak

