AI-Assisted Thai Language Learning Mobile Application

Master of Science

Information Technology

Mahanakorn University of Technology

Project Title	AI-Assisted Thai Language Learning Mobile Application
Student Name	Liang JianYu
Student ID	6717740001
Program	Master of Science in Information Technology
Year	2025
Advisor	Assoc.Prof.Dr.Mudarmeen Munlin

Contents

Chapter 1 Introduction

1.1 Problems and Motivation

With the growing economic and cultural exchanges between China and Thailand, the demand for Thai language learning among Chinese speakers has significantly increased. However, most existing language learning applications are designed primarily for English speakers, lacking localized teaching content and pronunciation feedback tailored to Chinese learners. This leads to inefficiencies in tone recognition and limited progress in pronunciation accuracy, which are crucial in tonal languages such as Thai.

Research has demonstrated that tonal languages like Thai, with five distinctive tones, require specialized automatic speech recognition systems with robust tone detection capabilities (Kaur & Kumar, 2020). Traditional general-purpose ASR systems often fail to accurately assess tonal pronunciation, which is critical for meaning differentiation in Thai.

This project aims to design and implement an AI-assisted Thai Language Learning Mobile Application that integrates artificial intelligence (AI), speech recognition, and cloud technologies to create an intelligent, personalized, and efficient learning platform. The system focuses on solving key challenges in Thai learning for Chinese speakers, including pronunciation accuracy, vocabulary memorization, and personalized learning path generation.

1.2 Objective

1. Develop an integrated mobile learning system covering Thai alphabet, vocabulary, sentence construction, pronunciation assessment, and reading comprehension.
2. Utilize AI-driven pronunciation evaluation to improve tone recognition and phoneme accuracy.
3. Implement a personalized content recommendation system based on user learning progress and behavior.
4. Provide cloud-based real-time synchronization to ensure seamless learning experiences across devices.
5. Enhance learner engagement through interactive exercises and gamified progress tracking.

1.3 Scope

The proposed system covers the following core modules, designed in accordance with the Use Case Diagram and Class Diagram :

1.3.1 Alphabet Learning Module

1. Introduces all 44 consonants and 32 vowels through interactive flashcards.
2. Provides stroke-order animations and native pronunciation playback.
3. Includes quizzes and matching exercises to reinforce character recognition.

1.3.2 Vocabulary Learning Module

1. Organizes over 1,000 Thai words into thematic categories such as “Food”, “Family”, and “Travel”.
2. Integrates spaced repetition algorithms to optimize memory retention.
3. Provides native speaker audio and tone practice exercises.

1.3.3 Sentence Construction and Grammar Practice

1. Supports sentence-building exercises based on grammar patterns and templates.
2. Provides drag-and-drop activities and fill-in-the-blank exercises for reinforcement.
3. Offers instant feedback and correction suggestions.

1.3.4 Pronunciation Evaluation System

1. Employs AI-powered tone and phoneme analysis to evaluate pronunciation accuracy.
2. Displays tone curves and quantitative feedback for real-time correction.
3. Enables learners to record, replay, and compare their pronunciation with native samples.

1.3.5 Reading Comprehension Module

1. Provides graded reading materials categorized by difficulty levels.
2. Offers interactive vocabulary support and comprehension questions.
3. Enables learners to save unfamiliar words to their personal review list.

1.3.6 User Management and Cloud Synchronization

1. Provides user registration, authentication, and profile management.
2. Synchronizes learning progress across multiple devices using real-time cloud storage.
3. Integrates with AI recommendation engines for continuous learning optimization.

1.4 Tools

Layer	Technologies Used
Front-end	React Native, TypeScript, Zustand
Back-end	Node.js, Express.js, Tencent CloudBase
AI Services	Tencent Speech Oral Processing (SOP), Tencent Text-to-Speech (TTS)
Database	CloudBase Real-time Database
Storage	Cloud Object Storage (COS)
Development Tools	Visual Studio Code, Android Studio, Xcode

1.5 Benefit of the Project

This system addresses identified gaps in technology-enhanced language learning for tonal languages. Mobile-assisted language learning (MALL) has demonstrated medium-to-large positive effect sizes on learning outcomes across multiple meta-analyses, with mobile devices providing unique affordances for personalized, anytime-anywhere learning (Burston, 2015). By integrating evidence-based pedagogical approaches—including AI-powered pronunciation assessment and scientifically-validated spaced repetition algorithms—with mobile technology, this project aims to deliver measurable improvements in Thai language acquisition for Chinese speakers.

This system provides a localized and intelligent learning experience for Chinese learners of Thai. By leveraging AI-driven pronunciation assessment and personalized recommendations, it enhances tone acquisition and language retention efficiency. The cross-platform mobile design ensures accessibility, while cloud synchronization guarantees data consistency. Overall, this project contributes to improving the quality and effectiveness of Thai language learning through modern AI and mobile technologies.

1.6 Project Schedule

Phase	Description	Duration
1	Requirement Analysis & Proposal	August – September
2	Theory and Background Research	September – October
3	System Analysis and Design	October – November
4	Implementation	November – January
5	Testing and Evaluation	January – February
6	Final Report & Presentation	February – March

Chapter 2 Theory and Background

2.1 Introduction

This chapter provides an overview of the core technologies and theoretical frameworks that support the design and implementation of the Thai language learning system. The project follows a mobile-first architecture with cloud integration, focusing on scalability, modularity, and responsiveness. React Native serves as the cross-platform framework, while Tencent CloudBase provides the backend infrastructure. AI services such as Tencent SOP and TTS enhance pronunciation and audio synthesis functionalities.

2.2 Key Front-end Technologies

2.2.1 React Native Framework

React Native is an open-source framework developed by Meta for building cross-platform mobile applications using JavaScript and React [1]. It allows developers to write one codebase that runs on both iOS and Android, improving development efficiency and reducing maintenance costs. The component-based design of React Native enhances reusability and modularization, ensuring smooth user interfaces and fast performance.Figure 2.1 illustrates the working principle of React Native.

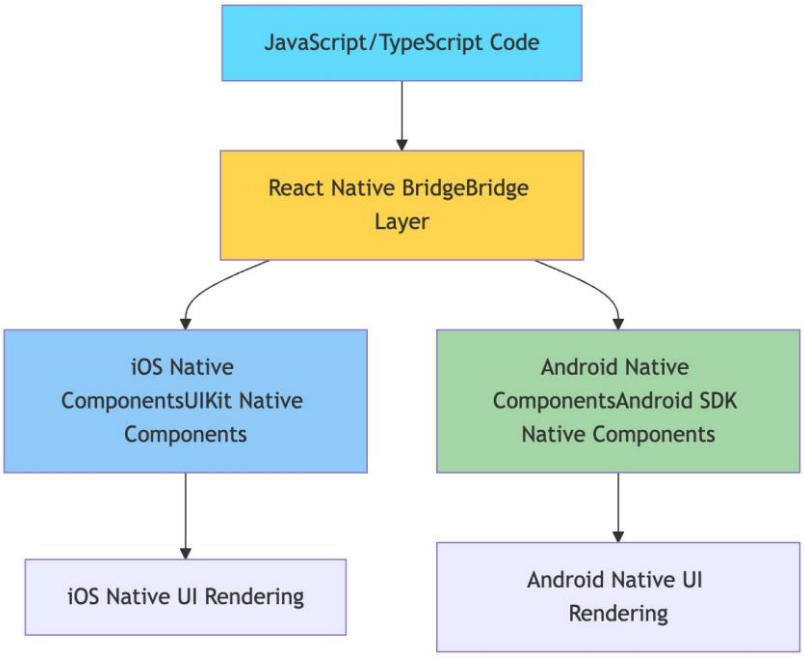

Figure 2.1 React Native Working Principle

2.2.2 TypeScript

TypeScript is a statically typed superset of JavaScript that improves code reliability and maintainability [2]. It enforces type safety, reducing runtime errors and improving development efficiency, particularly for complex systems with extensive data models and API interactions.

2.2.3 Zustand for State Management

Zustand is a lightweight state management library for React applications. As show at Figure 2.2,it provides a simpler alternative to Redux by offering a hook-based API and minimal boilerplate code. Zustand is responsible for managing user states, progress tracking, and personalized learning settings.

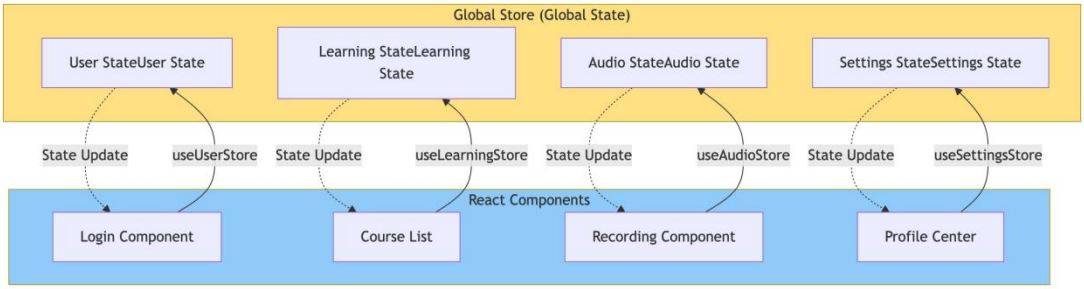

Figure 2.2 Zustand State Management Architecture

2.2.4 Web Audio API

The Web Audio API enables audio analysis and manipulation directly in the browser or within React Native via bridging [3]. It supports waveform visualization and real-time tone comparison, which are essential for pronunciation training modules.Figure 2.3 shows the workflow of web Audio.

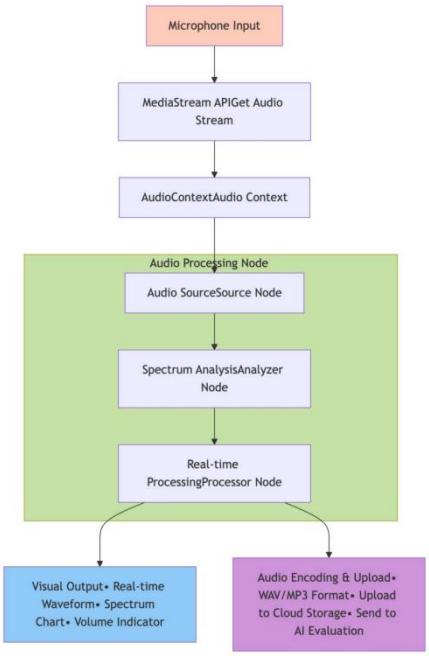

Figure 2.3 Audio Processing Flow

2.3 Key Back-end Technologies

2.3.1 Tencent CloudBase

Tencent CloudBase is a serverless backend-as-a-service (BaaS) platform integrating databases, authentication, and storage [4]. It simplifies backend maintenance by automating scaling and resource allocation. The system employs Cloud Functions for business logic, real-time databases for synchronization, and COS for audio and image storage.Through the cloud function workflow in Figure 2.4, we can intuitively see the working process of Tencent CloudBase.

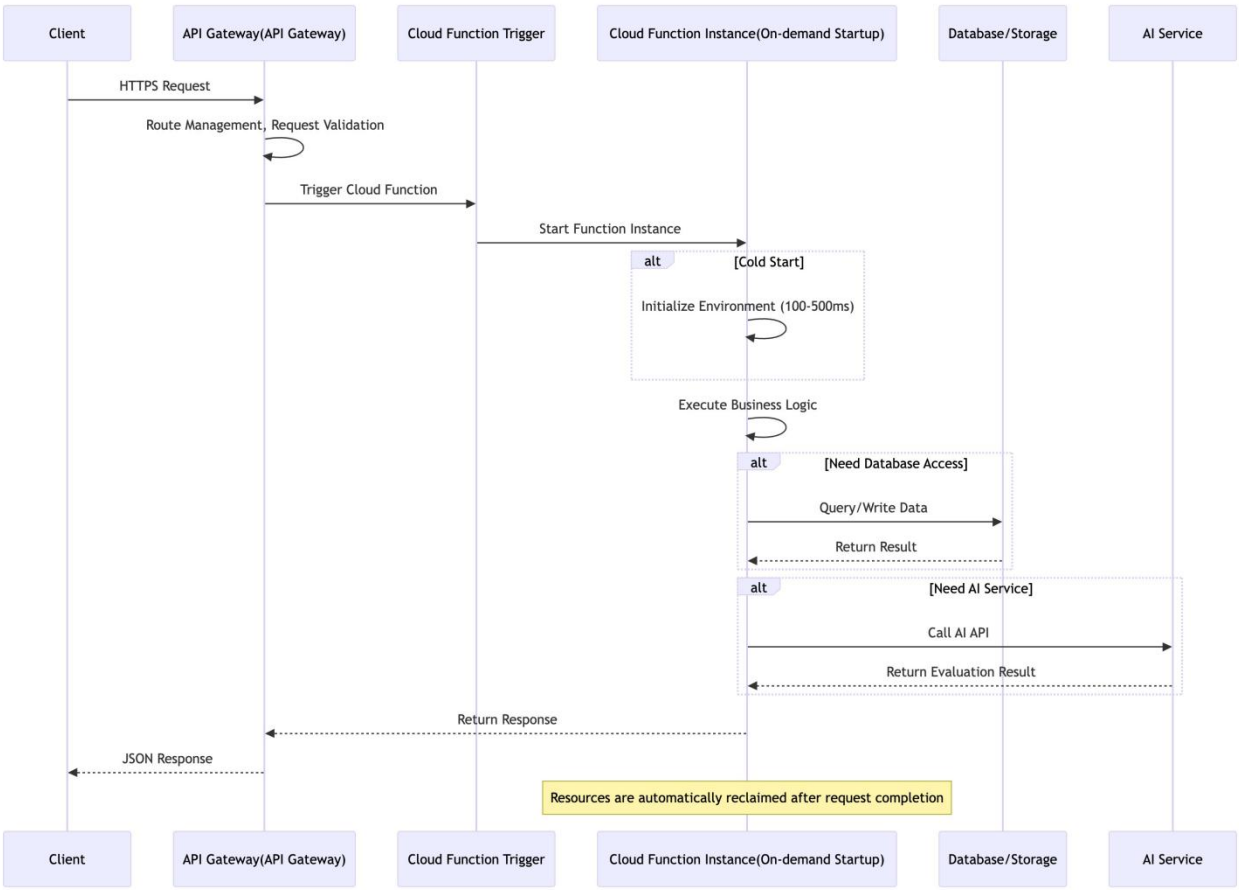

Figure 2.4 Cloud Function Workflow

2.3.2 Real-time Database

The CloudBase database supports real-time data synchronization through WebSocket connections. This ensures immediate propagation of user progress and learning records across devices, maintaining data consistency and supporting seamless cross-device experiences.As show in Figure 2.5, it is the architecture of the Real-time Database.

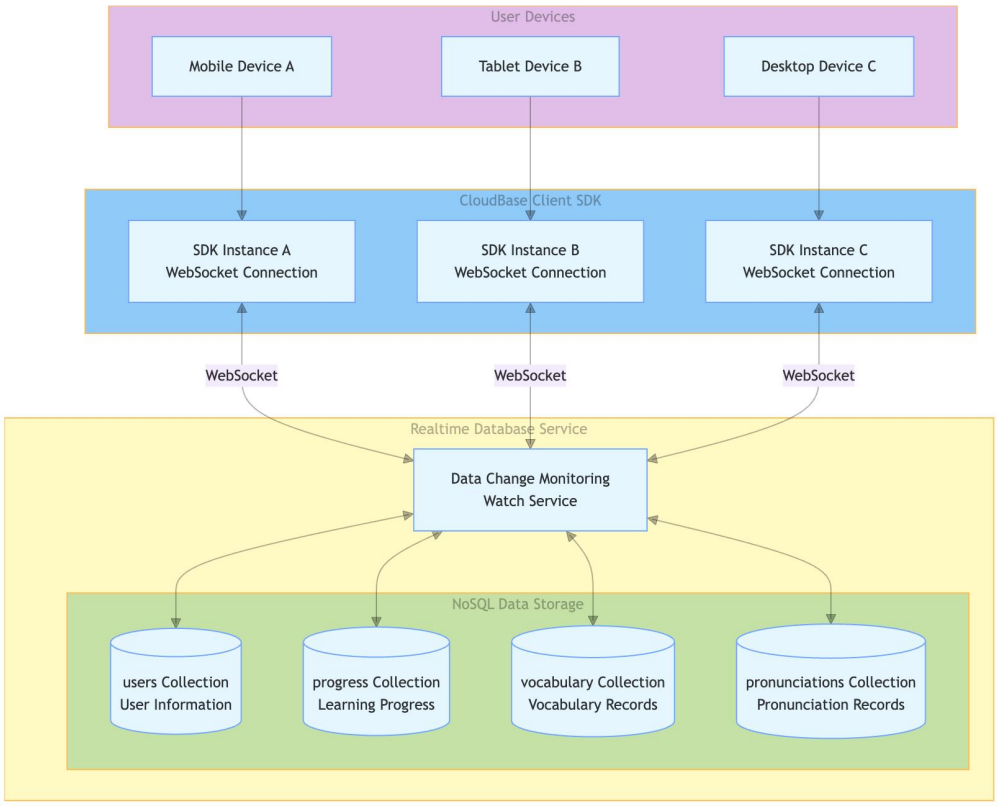

Figure 2.5 Real-time Database Architecture

2.4 AI and Speech Technologies

2.4.1 Tencent Speech Oral Processing (SOP)

Pronunciation assessment is a core function of this application and the primary technical solution for addressing the "tonal learning difficulties" identified in Chapter 1. This project adopts Tencent Cloud's Speech Oral Processing (SOP). Its complete working process is shown in Figure 2.6. This service is specifically optimized for language learning scenarios, providing stable, low-latency assessment capabilities[5]. A backend cloud function calls the SOP interface to conduct a multi-dimensional analysis of the user's uploaded audio, including "tone accuracy" and "phoneme integrity." It then returns the quantitative score and corrective feedback to the client, thereby forming a complete learning feedback loop.

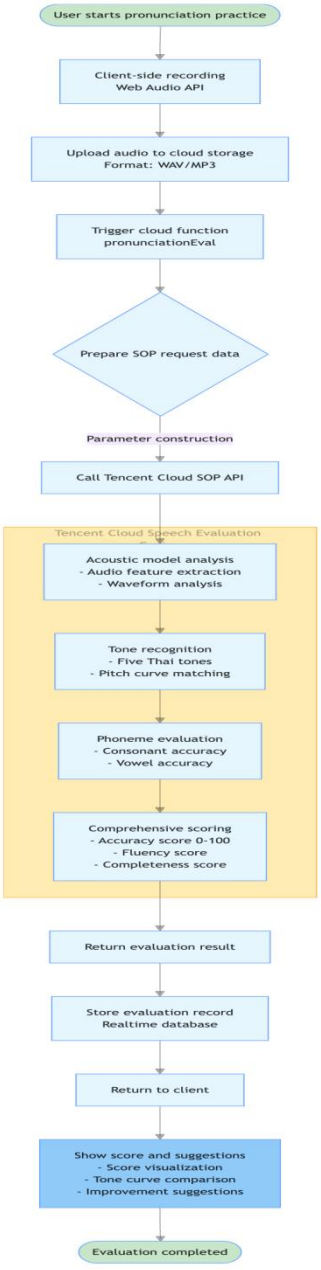

Figure 2.6 Complete Process of SOP Speech Evaluation

2.4.2 Tencent Text-to-Speech (TTS)

To provide standard and natural Thai pronunciation examples (e.g., for vocabulary and sentences), the system will utilize Tencent Cloud's Text-to-Speech (TTS) service.Tencent TTS synthesizes high-quality native Thai speech with customizable voice options. It ensures consistent and authentic pronunciation models for learners [6]. It also supports speech rate adjustment. This ensures that learners are exposed to accurate pronunciation models from the outset.Figure 2.7 shows the Speech Synthesis Process of the TTS.

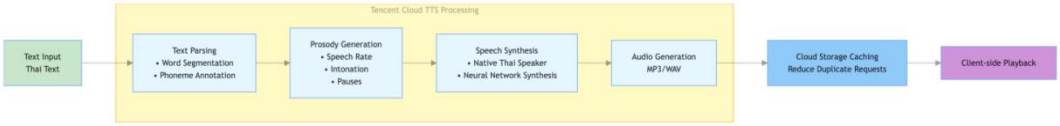

Figure 2.7 TTS Speech Synthesis Process

2.5 Cloud Object Storage

Figure 2.8 shows the Cloud storage architecture. The CloudBase solution has built-in COS capabilities, which are deeply integrated with Cloud Functions and the database. This integration simplifies the processes for file uploading, downloading, and permission management.All media assets, such as user recordings, images, and audio clips, are stored in Tencent Cloud Object Storage (COS), ensuring scalability, durability, and easy access [7].

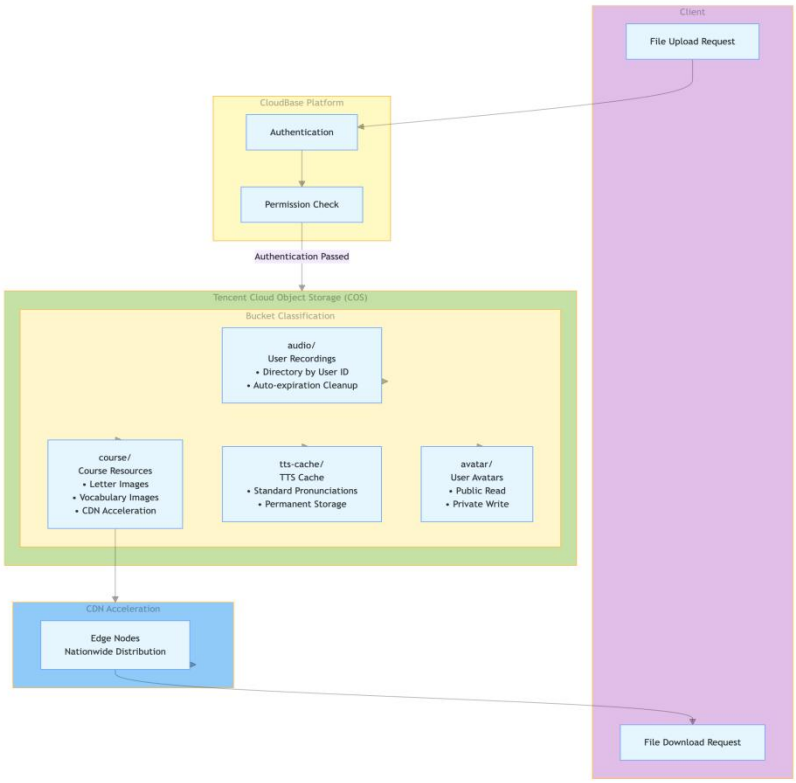

Figure 2.8 Cloud storage architecture

Chapter 3 System Design

3.1 Introduction

This chapter presents the system design and analysis of the AI-assisted Thai Language Learning Mobile Application, which serves as the foundation for implementation. The design aims to ensure that the system is scalable, user-friendly, and capable of providing intelligent, personalized learning experiences for Thai language learners. The chapter describes the overall system architecture, use case diagram, class diagram, and sequence diagram, followed by the detailed design of each component.

3.2 System Architecture

The architecture of the Thai Learning App is based on a client–server model utilizing cloud computing infrastructure. The client side, developed using React Native, provides the cross-platform user interface for both Android and iOS devices [1]. The backend utilizes Tencent CloudBase, a serverless platform integrating authentication, database, and file storage services [2]. AI services such as Tencent Speech Oral Processing (SOP) and Text-to-Speech (TTS) are employed for pronunciation assessment and voice synthesis [3][4].

The architecture is divided into three main layers:

1. Presentation Layer: Handles user interaction, interface rendering, and state management using React Native and Zustand.
2. Application Layer: Contains cloud functions and APIs responsible for processing user input, performing AI evaluation, and managing logic flow.
3. Data Layer: Consists of CloudBase databases and Cloud Object Storage (COS), ensuring secure storage and real-time synchronization of user progress.

This layered design facilitates modularity and scalability, allowing each module to evolve independently. Additionally, serverless deployment reduces maintenance overhead and ensures seamless scaling under fluctuating user loads [2][5].

The architectural choices reflect established principles in mobile-assisted language learning and speech technology. The mobile-first design leverages research-validated benefits of MALL, including accessibility, personalization, and opportunistic learning during otherwise unproductive time (Burston, 2015). The integration of specialized ASR for tonal language pronunciation assessment addresses identified challenges in Thai speech recognition, where tone detection systems significantly improve evaluation accuracy compared to general-purpose ASR (Kaur & Kumar, 2020). The serverless architecture through Tencent CloudBase ensures scalability while minimizing operational overhead, allowing focus on pedagogical features rather than infrastructure management.

3.3 Use Case Diagram

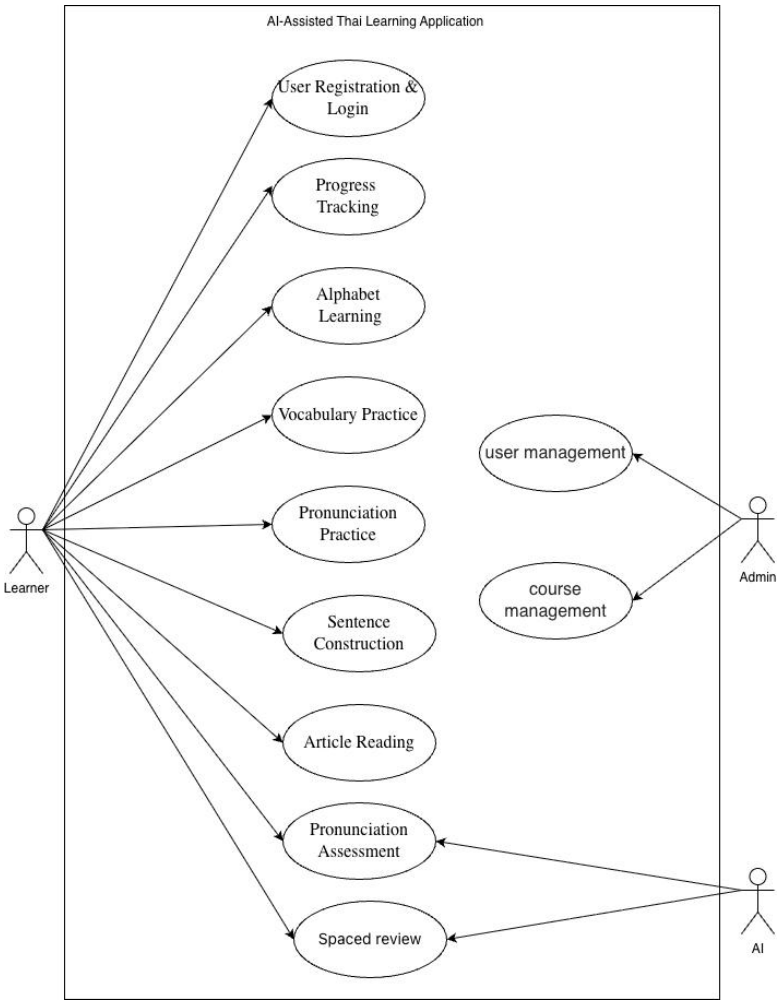

Figure 3.1 System Use Case Diagram

The system provides 10 core use cases organized into four functional modules: Learning Module, User Module, Intelligent Features Module, and Content Management Module. These use cases represent the essential functionalities required to support the complete learning lifecycle from initial alphabet study to advanced article reading and assessment.

Figure 3.1 illustrates the use case diagram of App, showing the relationship between actors (Learner, Admin and AI Service) and the various use cases. The diagram employs standard UML to represent dependencies and mandatory relationships between use cases.

3.3.1 Learner Use Case Description

Use Case Description 1

Use Case Name	User Registration and Login
Use Case ID	UC1
Actors	Learner
Preconditions	None

Postconditions	User successfully authenticated and redirected to home page
Main Flow	1. Learner opens the application 2. Learner chooses to register a new account or login 3. For registration: Learner provides email, password, and display name 4. For login: Learner provides email and password 5. System validates credentials and creates/authenticates user 6. System redirects learner to home page
Alternative Flows	3a. Third-party authentication (OAuth) - future feature
Exception Flows	- Invalid input format - Authentication failure - Network connectivity issues
Business Rules	- Email must be unique in the system - Password minimum 8 characters - Session expires after 7 days of inactivity
Related Classes	User, UserService, UserStore, LoginPage

Use Case Description 2

Use Case Name	Progress Tracking
Use Case ID	UC2
Actors	Learner
Preconditions	User is authenticated
Postconditions	Progress statistics displayed with visual charts
Main Flow	1. Learner navigates to profile/progress section 2. System retrieves and displays learning statistics: - Current proficiency level - Content completion counts by type - Total score and study time - Learning streak days 3. System presents progress visualization (charts/graphs) 4. Progress synchronized to cloud storage
Alternative Flows	2a. Use cached data if available, sync in background 3a. Learner drills down into detailed statistics by topic
Exception Flows	- Data retrieval failure → display cached data - Sync failure → queue for background retry

Business Rules	- Progress syncs every 5 minutes or on significant events - Completion rate = completed items / total course items - Level progression based on completion rate thresholds
Related Classes	LearningProgress, ProgressService, ProfilePage

Use Case Description 3

Use Case Name	Alphabet Learning
Use Case ID	UC3
Actors	Learner
Preconditions	User authenticated and course selected
Postconditions	Alphabet study session completed, progress updated
Main Flow	1. Learner selects alphabet learning module 2. System presents Thai character with: - Visual representation - Pronunciation guide - Example word - Category (consonant/vowel) 3. Learner plays standard pronunciation audio 4. Learner studies the character 5. Learner navigates to next/previous character or exits 6. System records completion and updates progress
Alternative Flows	3a. Skip audio playback 5a. Practice pronunciation (→ UC05)
Exception Flows	- Audio playback failure → use fallback or skip - Progress save failure → cache locally for sync
Business Rules	- 76 total Thai characters (44 consonants, 32 vowels) - Characters organized by category for systematic learning
Related Classes	Alphabet, LearningService, LearningPage, AudioStore

Use Case Description 4

Use Case Name	Vocabulary Practice
Use Case ID	UC4
Actors	Learner
Preconditions	User authenticated, course and difficulty selected
Postconditions	Vocabulary practiced, mastery level updated, review scheduled

Main Flow	1. Learner enters vocabulary practice module 2. System presents vocabulary item with: - Thai word and translation - Pronunciation audio - Example sentence 3. System presents interactive exercise (multiple choice/fill-blank/listen-select) 4. Learner completes exercise and submits answer 5. System provides immediate feedback with correct answer 6. System updates vocabulary mastery level based on performance 7. System creates/updates spaced repetition schedule if mastered
Alternative Flows	3a. AI recommends vocabulary based on weak areas 5a. Correct answer → increase mastery 5b. Incorrect answer → add to review queue
Exception Flows	- Empty vocabulary list for selected difficulty - Progress sync failure → local cache
Business Rules	- Mastery levels: NOT_LEARNED → LEARNING → REVIEWING → MASTERED - Mastery progression based on consecutive correct answers - AI prioritizes weak vocabulary items
Related Classes	Vocabulary, MasterLevel, LearningService, ReviewService, AIRecommendationEngine

Use Case Description 5

Use Case Name	Pronunciation Practice
Use Case ID	UC5
Actors	Learner
Preconditions	User authenticated and viewing learning content
Postconditions	Pronunciation recorded and ready for assessment
Main Flow	1. Learner initiates pronunciation practice from learning content 2. System displays target text 3. System plays standard pronunciation (optional) 4. Learner records pronunciation using microphone 5. Learner can replay recording, re-record, or submit for assessment 6. If submitted → proceed to UC08 (Pronunciation Assessment)
Alternative Flows	3a. Skip standard audio playback 5a. Re-record multiple attempts
Exception Flows	- Microphone permission denied → guide to settings - Recording save failure → prompt retry
Business Rules	- Maximum 3 minutes recording duration - Audio format: WAV 16kHz mono
Related Classes	AudioStore, AudioPlayer, AudioData, LearningPage

Use Case Description 6

Use Case Name	Sentence Construction
Use Case ID	UC6
Actors	Learner
Preconditions	User authenticated, course and difficulty selected
Postconditions	Sentence exercise completed, progress updated
Main Flow	1. Learner enters sentence construction module 2. System presents Thai sentence with translation and grammar pattern 3. System plays sentence audio (optional) 4. System presents construction exercise: - Drag-and-drop word ordering, or - Fill-in-blank completion 5. Learner constructs sentence and submits 6. System validates and provides detailed feedback: - Correctness indication - Grammar explanation - Highlighted vocabulary 7. System records completion and updates progress
Alternative Flows	4a. Listen-only mode (audio-first construction) 6a. Partial correctness with specific feedback on errors
Exception Flows	- Content loading failure → retry option - Validation error → allow resubmission
Business Rules	- Sentences organized by grammar pattern - Difficulty increases with sentence length and complexity - Correct answers trigger review schedule creation
Related Classes	Sentence, LearningService, LearningPage, ProgressService

Use Case Description 7

Use Case Name	Article Reading
Use Case ID	UC7
Actors	Learner
Preconditions	User authenticated, course and difficulty selected
Postconditions	Article read, comprehension tested, progress updated

Main Flow	1. Learner selects article reading module 2. Learner browses and selects an article 3. System displays article with: - Thai text - Optional Chinese translation - Difficulty level and word count 4. Learner can: - Tap words for instant definitions - Play full article or paragraph audio - See highlighted known vocabulary 5. After reading, learner marks completion 6. System presents comprehension questions (optional) 7. System records completion and updates progress
Alternative Flows	4a. Save unfamiliar words to personal vocabulary list 4b. Follow-along (read-aloud) mode with recording 6a. Skip comprehension test
Exception Flows	- Article loading failure → show cached articles - Audio unavailable → continue with text-only
Business Rules	- 50+ articles across difficulty levels - Word count: 100-500 words - Articles categorized by topic and cultural relevance
Related Classes	Article, LearningService, LearningPage, ProgressService

Use Case Description 8

Use Case Name	Pronunciation Assessment
Use Case ID	UC8
Actors	Learner, AI Service (Tencent SOP)
Preconditions	User has recorded pronunciation (UC05 completed)
Postconditions	Assessment report generated and saved

Main Flow	1. Learner submits recorded audio for assessment 2. System sends audio to AI service for evaluation 3. AI service analyzes: - Tone accuracy - Phoneme correctness - Fluency 4. System receives detailed assessment results 5. System displays comprehensive feedback: - Overall score with breakdown - Visual tone curve comparison - Specific improvement suggestions 6. System saves assessment record 7. Learner can review details, practice again, or continue
Alternative Flows	3a. Use local AI model if cloud service unavailable (reduced accuracy) 7a. View historical assessments and improvement trends
Exception Flows	- AI service unavailable → save for later assessment - Poor audio quality → prompt re-recording with tips - Save failure → local cache with sync flag
Business Rules	- Assessment score: 0-100 with component breakdown - Score weights: Tone (40%), Phoneme (35%), Fluency (25%) - Average processing time: <3 seconds
Related Classes	PronunciationRecord, PronunciationService, TencentSOP, CloudBase

Use Case Description 9

Use Case Name	Spaced Repetition Review
Use Case ID	UC9
Actors	Learner, AI System
Preconditions	User has learned content with active review schedules
Postconditions	Review completed, schedules updated per SM-2 algorithm
Main Flow	1. System notifies learner of due reviews on home page 2. Learner initiates review session 3. System presents due items in priority order 4. Learner reviews each item (similar to original learning) 5. Learner self-assesses recall quality (1-5 scale) 6. System calculates next review date using SM-2 algorithm: - Easy recall (4-5): Extend interval significantly - Medium recall (3): Maintain interval - Poor recall (1-2): Reset to short interval 7. System updates review schedule and progress 8. System displays review completion summary

Alternative Flows	3a. AI prioritizes items based on weakness analysis 3b. Quick review mode with flashcard interface
Exception Flows	- No reviews due → suggest new content - Schedule update failure → local cache with sync flag
Business Rules	- SM-2 algorithm with intervals: 1d, 3d, 1w, 2w, 1m, 3m... - Ease factor range: 1.3-2.5 - Maximum interval: 180 days
Related Classes	ReviewSchedule, ReviewService, AIRecommendationEngine, LearningPage

3.3.2 Admin Use Case Description

Use Case Description 10

Use Case Name	User Management
Use Case ID	UC10
Actors	Administrator
Preconditions	Administrator authenticated with ADMIN role
Postconditions	User accounts managed per administrator actions
Main Flow	1. Administrator accesses user management interface 2. System displays user list with key information 3. Administrator performs management operations: - Create new user accounts - Edit user information (name, email, role) - Assign or change user roles - View detailed user learning statistics - Delete user accounts (with confirmation) 4. System validates and executes operations 5. System displays operation results
Alternative Flows	2a. Search/filter users by email, role, or registration date 3a. Batch operations for multiple users
Exception Flows	- Duplicate email on creation - Delete user with active learning data → suggest deactivation - Update failure → rollback changes
Business Rules	- Only ADMIN role can manage users - Cannot delete self or last admin - User deletions cascade to associated data
Related Classes	User, UserRole, UserService, CloudBase

Use Case Description 11

Use Case Name	Course Management
Use Case ID	UC11

Actors	Administrator
Preconditions	Administrator authenticated with ADMIN role
Postconditions	Course catalog and content managed per administrator actions
Main Flow	1. Administrator accesses course management interface 2. System displays course list with metadata and statistics 3. Administrator performs course operations: - Create new courses with metadata - Edit course information - Add learning content (alphabet/vocabulary/sentences/articles) - Upload or generate (TTS) audio files - Edit or delete existing content - Activate/deactivate courses 4. System validates and executes operations 5. System displays operation results
Alternative Flows	3a. Generate audio using TTS service 3b. Batch import content from CSV/Excel 3c. Preview course as learner
Exception Flows	- Duplicate course name - Audio upload failure → retry or defer - Delete course with learner data → prevent or warn
Business Rules	- Only ADMIN role can manage courses - Deactivated courses hidden from learners - Course deletion requires no active learners - Audio files stored in cloud object storage
Related Classes	Course, Alphabet, Vocabulary, Sentence, Article, CourseService, TencentTTS, CloudBase

3.4Class Design

The Thai Language Learning Application employs a comprehensive object-oriented architecture consisting of 35 core classes organized across three distinct layers: Entity Layer, Service Layer, and Presentation Layer. The class diagram design adheres to SOLID principles and industry-standard design patterns to ensure maintainability, scalability, and testability. The system is specifically architected to support AI-powered pronunciation assessment and personalized learning recommendations for Chinese speakers learning Thai.

The architecture separates concerns through clear layer boundaries: the Entity Layer encapsulates domain models and business rules; the Service Layer implements core business logic and external service integration; and the Presentation Layer handles user interface components and client-side state management. This separation enables independent evolution of each layer while maintaining well-defined interfaces for communication.

3.4.1 Architecture Overview

As shown in Figure 3.2， the overall system architecture follows a three-tier pattern that clearly delineates responsibilities and dependencies:

Entity Layer (10 classes)

The Entity Layer represents the core domain model with persistent data structures:

- Learning Content Entities: Course, Alphabet, Vocabulary, Sentence, and Article represent the hierarchical learning materials. Each entity contains Thai script, Chinese translations, pronunciation data, and audio URLs.
- User Progress Entities: User, LearningProgress, PronunciationRecord, and ReviewSchedule track learner activities, assessment results, and spaced repetition schedules.
- Data Transfer Object: AudioData encapsulates audio content with format metadata for pronunciation playback and recording.

Service Layer (7 classes)

The Service Layer implements business logic and orchestrates external service integration:

- Core Services: UserService, CourseService, and LearningService handle basic CRUD operations and content delivery.
- Intelligent Services: PronunciationService integrates Tencent SOP for AI-powered speech assessment; AIRecommendationEngine analyzes learning patterns to generate personalized recommendations; ReviewService implements the SM-2 spaced repetition algorithm.
- Progress Management: ProgressService synchronizes learning statistics across devices in real-time.

Presentation Layer (15 classes)

The Presentation Layer manages user interactions and application state:

- Page Components: LoginPage, CourseListPage, LearningPage, and ProfilePage implement the primary user interfaces.
- State Management: UserStore, LearningStore, and AudioStore leverage Zustand for reactive state management with persistent caching.
- Utility Classes: ApiClient, ErrorHandler, DataTransformer, Router, AudioPlayer, and LocalStorage provide cross-cutting infrastructure concerns.

External Services

Three external service classes integrate Tencent Cloud infrastructure:

- TencentSOP: Provides AI-powered pronunciation evaluation with tone, phoneme, and fluency analysis specifically optimized for Thai language.
- TencentTTS: Generates native Thai speech audio from text with configurable voice types and speech rates.
- CloudBase: Abstracts the serverless backend platform, providing database operations, file storage, cloud functions, and real-time data synchronization via WebSocket.

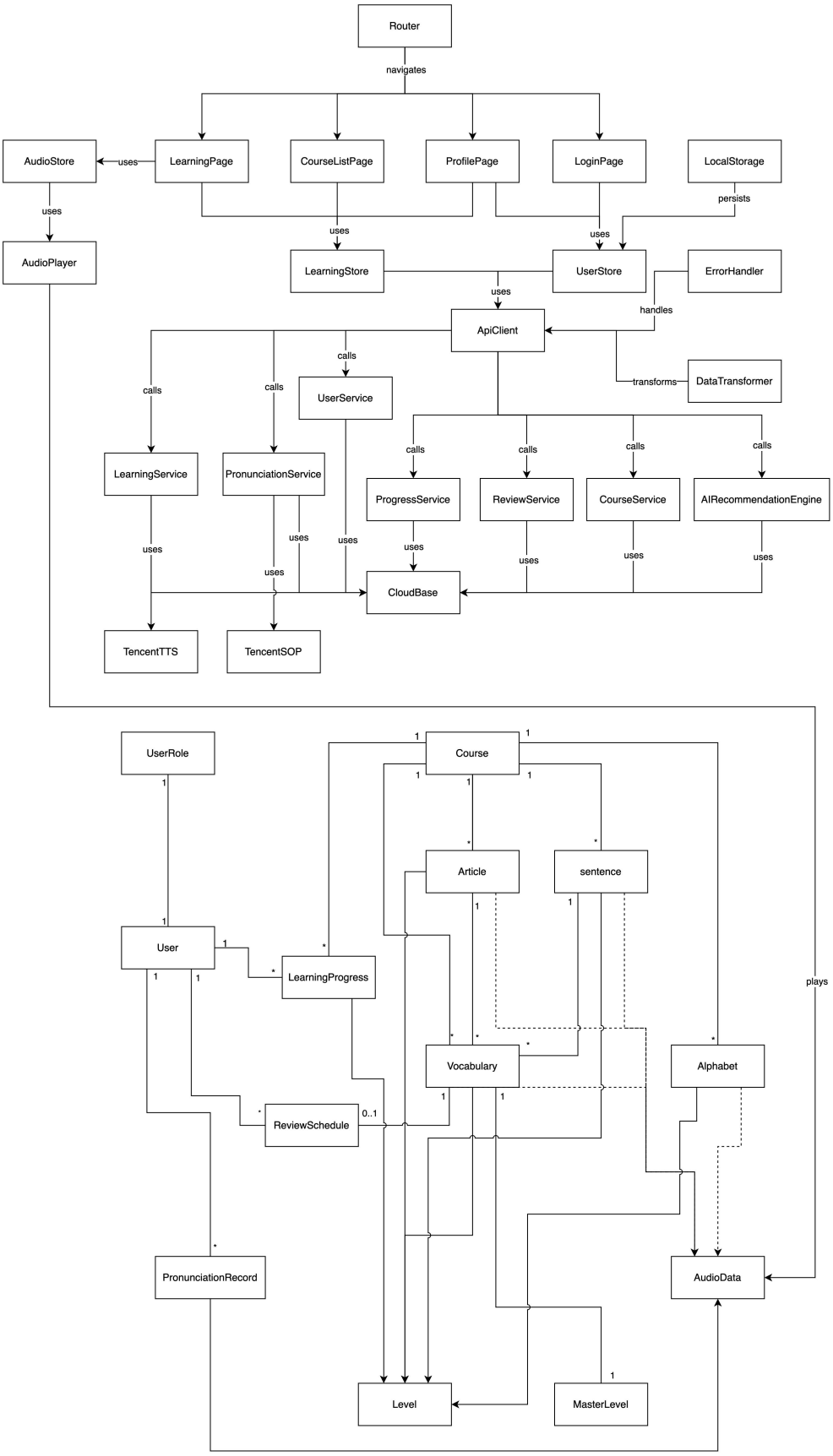

Figure 3.2 Class Diagram of Thai Learning App

3.4.2 Backend Architecture

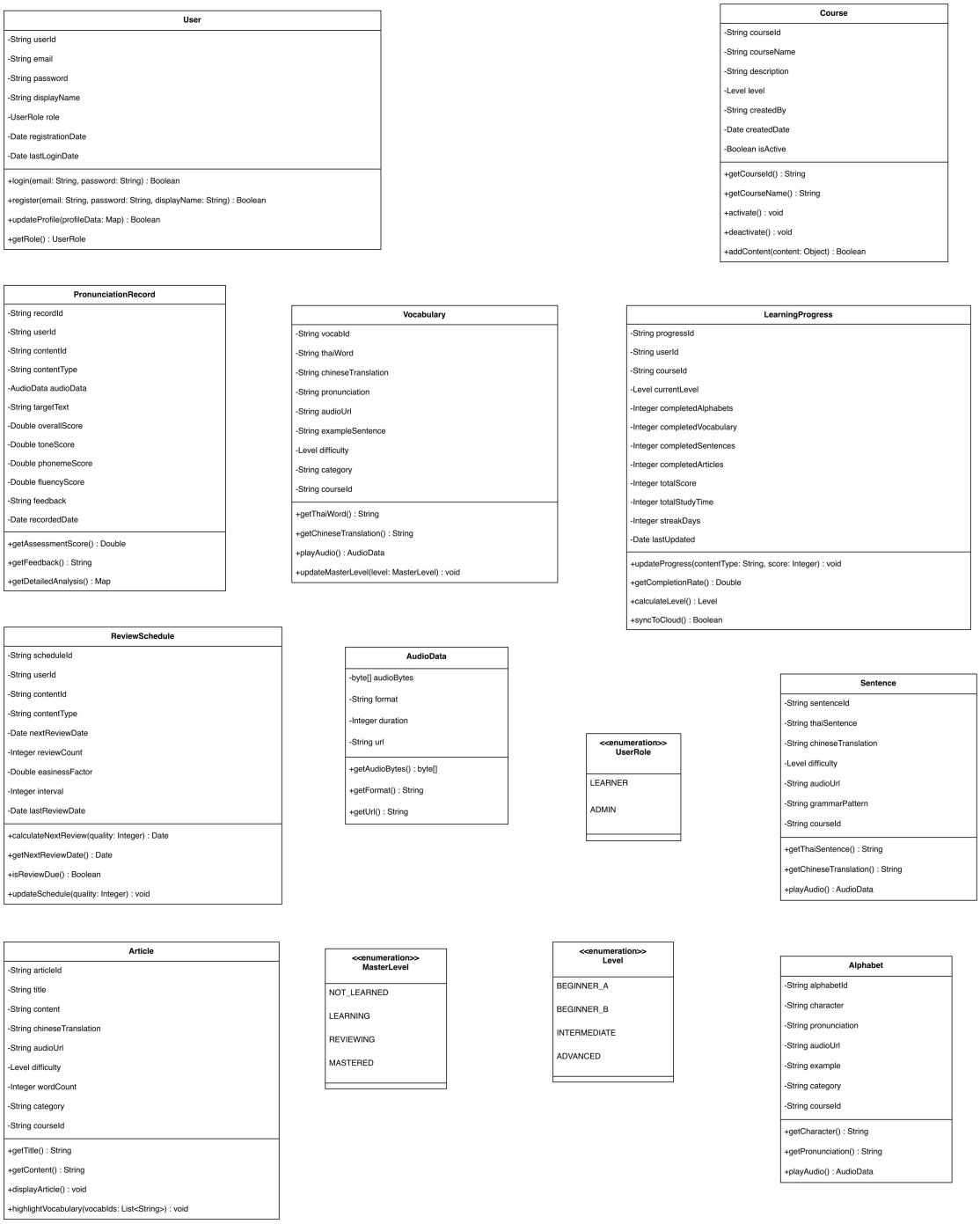

Figure 3.3 Class Diagram Details of Backend

As shown in Figure 3.3 , it mainly shows the details of backend classes, which include entity classes and server design:

Key Entity Classes

- **User:** Central entity for learners, holding auth credentials, role (LEARNER/ADMIN), and registration data. Maintains 1:1 composition with LearningProgress and 1:many links to PronunciationRecord/ReviewSchedule.

- **Vocabulary:** Represents Thai words with metadata (Thai script, Chinese translation, pronunciation, examples, difficulty, audio URL). Tracks MasterLevel (NOT_LEARNED→LEARNING→REVIEWING→MASTERED) for spaced repetition, with 1:0..1 association to ReviewSchedule (only for active review items).
- **LearningProgress:** Aggregates learning stats (proficiency level, completed content counts, study time, streak days). Uses calculateLevel() to promote learners through tiers (BEGINNER_A→BEGINNER_B→INTERMEDIATE→ADVANCED) via completion rates/scores.
- **ReviewSchedule:** Implements SM-2 algorithm for memory retention, with attributes (nextReviewDate, interval, easinessFactor: 1.3-2.5). calculateNextReview(quality) adjusts intervals—failed reviews reset to 1 day (lowered easiness), successful ones extend exponentially (1→3→7→14 days).

Service Layer Design

- **PronunciationService (Core Innovation):** Orchestrates AI assessment: receives AudioData/target text → calls TencentSOP.evaluatePronunciation("th") → processes results (overall score 0-100, tone:40%, phoneme:35%, fluency:25%) → generates PronunciationRecord (with pitch visualization/feedback) → saves to CloudBase. Supports millisecond-level tone curve analysis vs. native speakers.
- **AIRecommendationEngine (Core Innovation):** Enables personalization via three algorithms:
  1. Weakness Analysis: K-means clustering for high-error topics (≥30% error, recency-weighted).
  2. Learning Style Detection: Analyzes study time, session duration, content preference (visual/audio/reading).
  3. Difficulty Optimization: Follows ZPD theory (recommends current level +1).
  4. Composite score (0.4×WeaknessMatch + 0.3×DifficultyFit + 0.2×StyleMatch + 0.1×Recency) selects top 10 items with context.
- **CloudBase Integration:** Facade for serverless infrastructure, offering invokeFunction (cloud logic), query/save/updateData (NoSQL ops), uploadFile (media storage), syncData (real-time WebSocket, <500ms latency), and subscribeToChanges (cross-device reactive updates).

3.4.3 Front-end Design

As shown in Figure 3.4 the frontend uses Zustand for lightweight state management via three core stores:

1. **UserStore**: Manages auth state (isAuthenticated, authToken, currentUser), handles async login flow (credential validation → API call → token persistence → state update), and provides checkAuth() as a route guard for protected pages.
2. **LearningStore**: Tracks session state (currentCourse, progress, completedContent), implements optimistic updates for progress (local state update first, then async CloudBase sync), and exposes getRecommendations() to fetch personalized content via ApiClient.
3. **AudioStore**: Handles audio recording/playback, using the Web Audio API to capture mic input, encode to 16kHz mono WAV format for pronunciation assessment, and manages isPlaying to prevent concurrent playback.

Key utility classes include:

- **ApiClient**: Centralizes HTTP communication with request/response interceptors (auto-inject authToken, handle 401 redirects, retry network failures, log requests/errors) and REST methods (get/post/put/delete) returning Promises.
- **Router**: Manages navigation, history, and route guards—setRouteGuard() enforces auth checks, redirecting unauthenticated users to login while preserving the intended route.
- **LocalStorage**: Abstracts browser storage with JSON serialization, storing authToken (for auto-login), user preferences (volume, theme), and offline queues (pending syncs), enabling cross-session state persistence for UserStore.

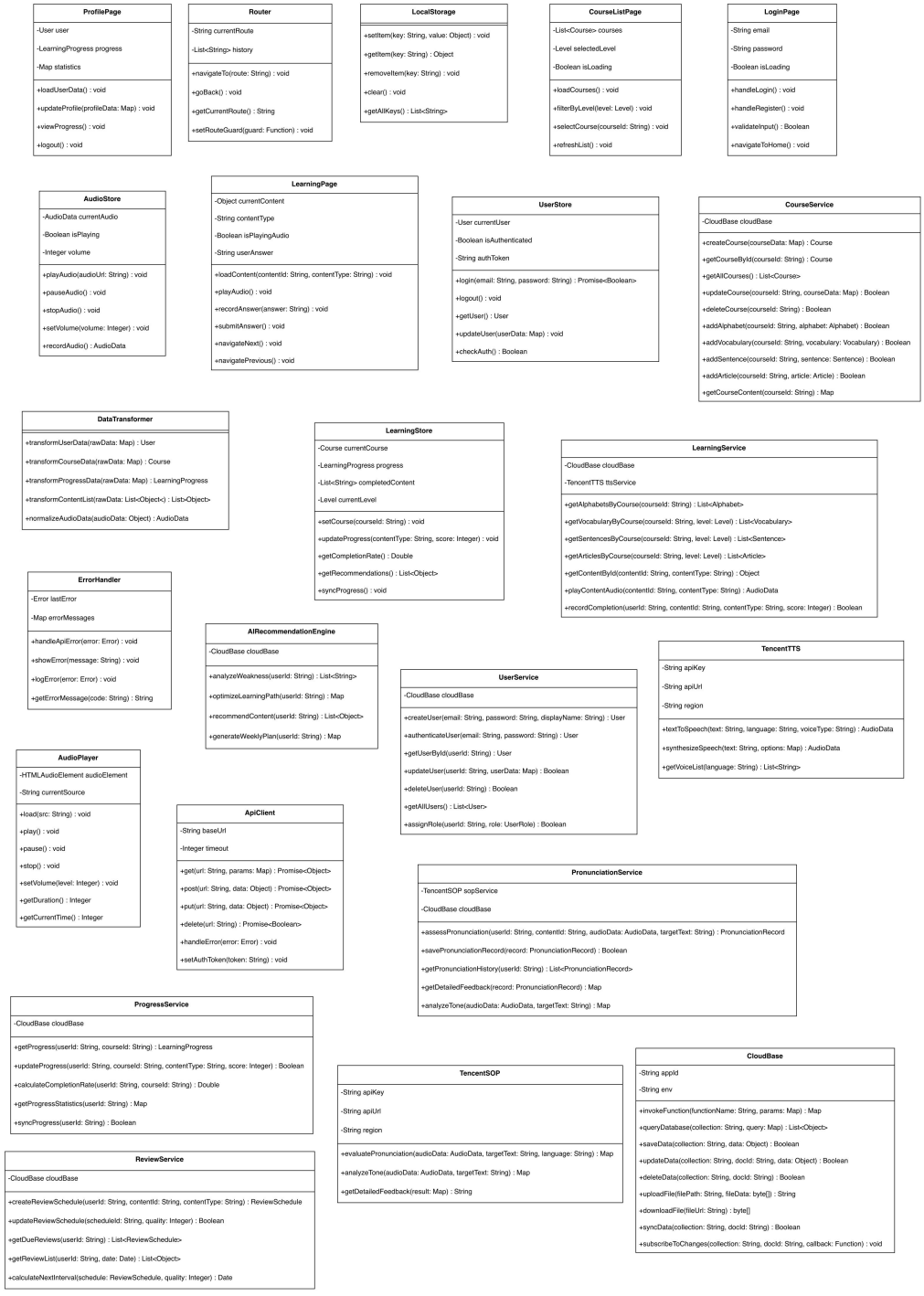

Figure 3.4 Class Diagram Details of Frontend

3.5 Sequence Diagram

3.5.1 User Sequence Diagram

1. Registration & Login Sequence Diagram

User registration demonstrates the interaction process between new users and the system when creating an account. Figure 3.5 illustrates the registration workflow. When a learner clicks "Register" on the login page, the system validates the input information through LoginPage.validateInput(). If the entered information is invalid (such as improper email format or weak password), an error message is returned to the user. If validation succeeds, the system calls UserService.createUser() to encrypt the password, generate a unique user ID, and save the user data to CloudBase. The diagram also shows the exception handling when the email already exists in the database, preventing duplicate accounts.

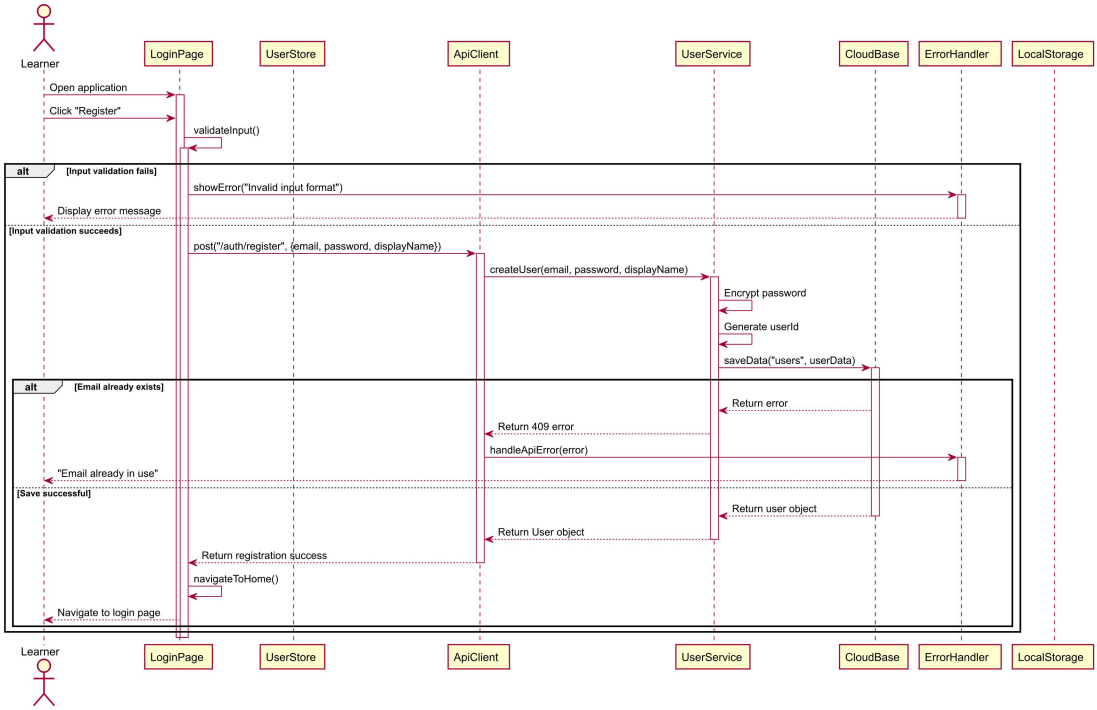

Figure 3.5 Registration Sequence Diagram

User login illustrates the authentication workflow when returning users access the application. Figure 3.6 shows the complete login sequence. After the learner enters credentials and clicks "Login", the system validates the input format and calls UserStore.login() to initiate authentication. The UserService.authenticateUser() method queries the database, verifies the password hash, and generates a JWT authentication token upon successful validation. The token is then stored in both UserStore and LocalStorage for session management. If authentication fails due to incorrect credentials, the system displays an appropriate error message and allows the user to retry. Upon successful login, the Router navigates the user to the home page.

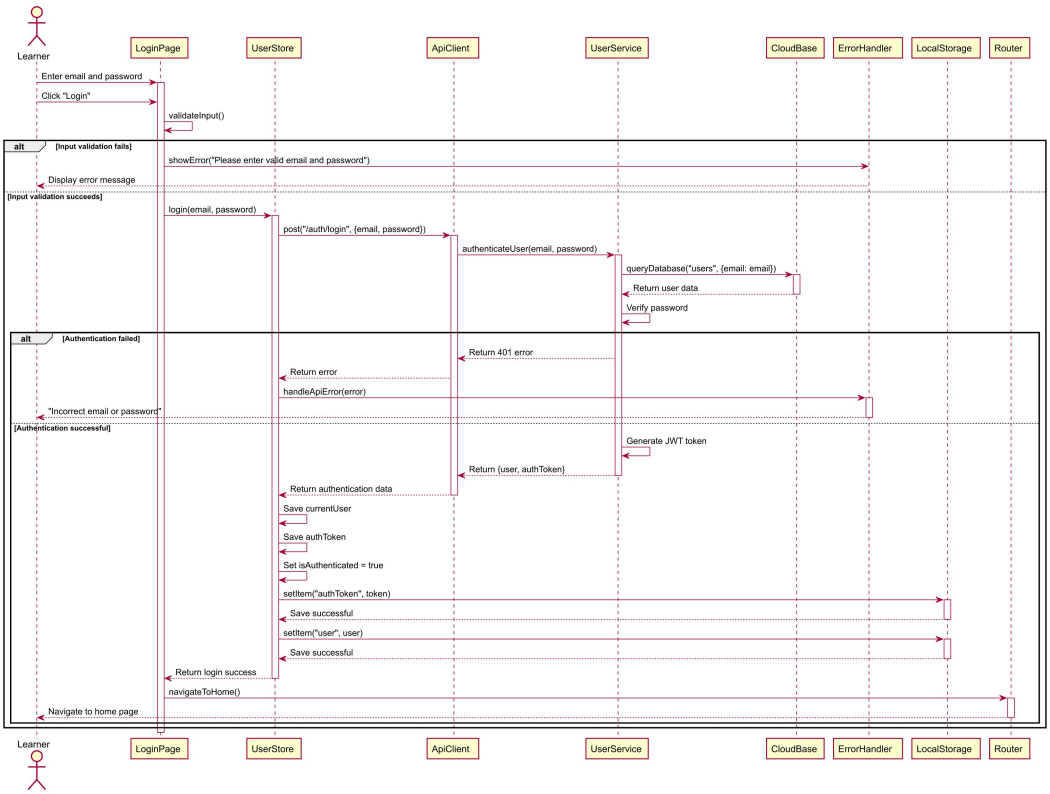

Figure 3.6 Login Sequence Diagram

2. Progress Tracking Sequence Diagram

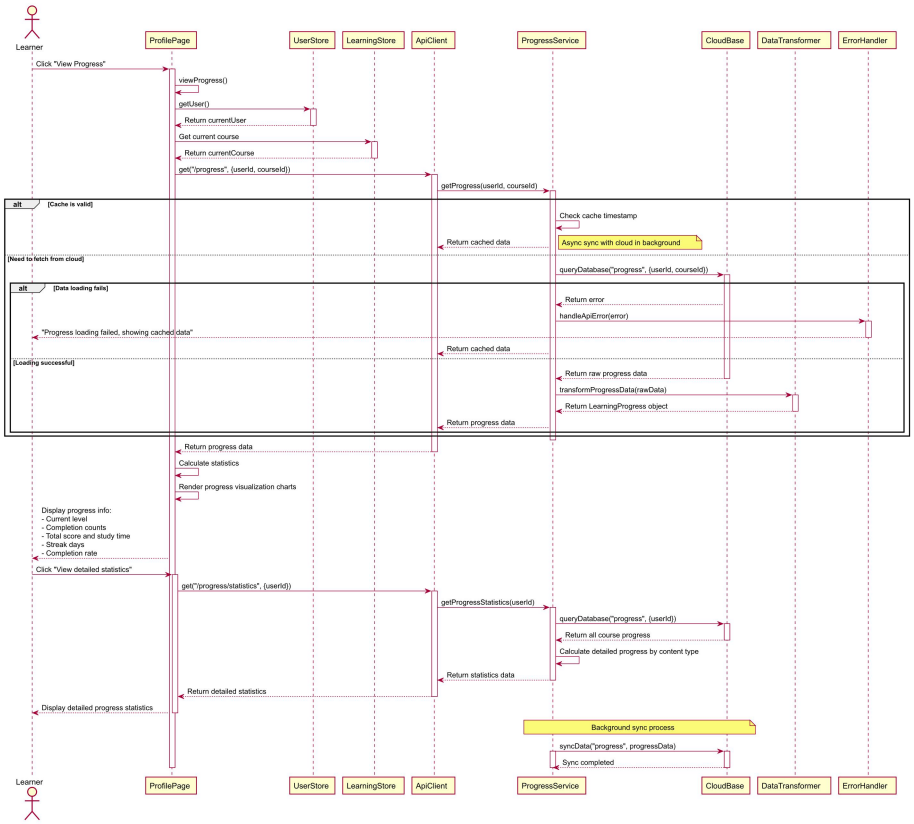

Figure 3.7 Progress Tracking Sequence Diagram

Progress tracking demonstrates how learners view their comprehensive learning statistics and achievements. Figure 3.7 illustrates the data retrieval and display workflow. When a learner clicks "View Progress" on the ProfilePage, the system calls ProgressService.getProgress() to fetch progress data from CloudBase. The diagram shows an intelligent caching strategy: if valid cached data exists locally, it is displayed immediately while the system asynchronously syncs with the cloud in the background. The DataTransformer.transformProgressData() method converts raw database records into a user-friendly format displaying current level, completion counts, total score, study time, and streak days. The visualization includes progress charts calculated by LearningProgress.getCompletionRate(). If data loading fails, the system gracefully falls back to cached data and allows the user to view detailed statistics through ProgressService.getProgressStatistics().

3. Alphabet Learning Sequence Diagram

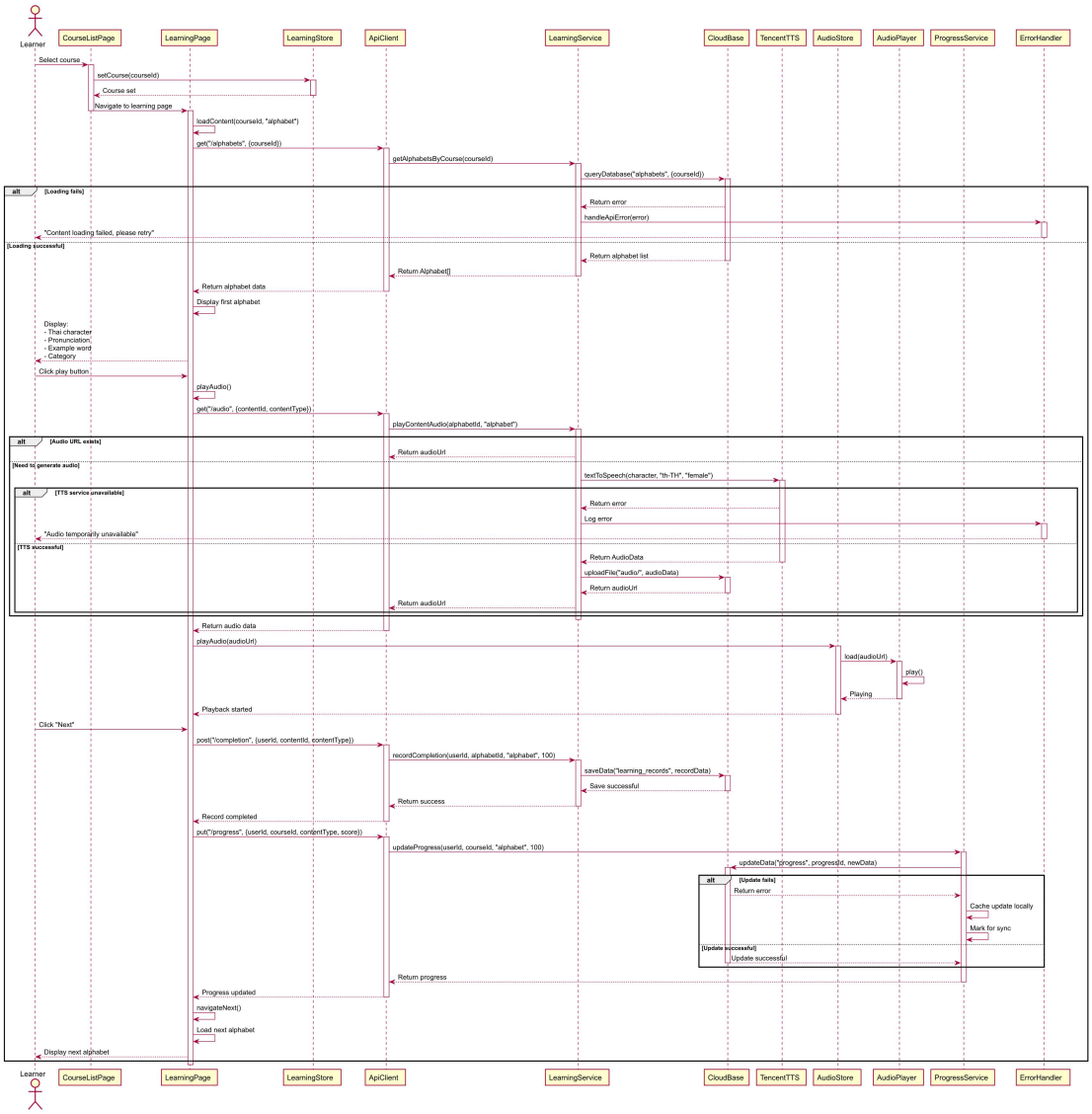

Figure 3.8 Alphabet Learning Sequence Diagram

Alphabet learning demonstrates the foundational language learning workflow for Thai characters. Figure 3.8 shows the complete interaction sequence. After the learner selects a course from CourseListPage, the system calls LearningService.getAlphabetsByCourse() to retrieve the alphabet list from CloudBase and loads the first character through LearningPage.loadContent(). The displayed information includes the Thai character, pronunciation guide, example word, and category. When the learner clicks the play button, the system either retrieves a pre-stored audio URL or generates new audio through TencentTTS.textToSpeech() and uploads it to cloud storage for future use. The AudioStore.playAudio() method streams the audio through AudioPlayer. Upon completion, LearningService.recordCompletion() logs the learning activity, and ProgressService.updateProgress() updates the user's progress statistics. The diagram includes comprehensive error handling for content loading failures, TTS service unavailability, and progress update synchronization issues.

4. Vocabulary Practice Sequence Diagram

Vocabulary practice demonstrates the interactive learning workflow with AI-powered recommendations and mastery tracking. Figure 3.9 illustrates the complete practice sequence. The learner selects the vocabulary practice module, and the system calls LearningService.getVocabularyByCourse() to retrieve vocabulary items filtered by the user's current difficulty level. The LearningPage displays comprehensive information including Thai word, Chinese translation, pronunciation annotation, example sentence, and category tags. The diagram shows two practice modes: standard sequential learning and AI recommendation mode, where AIRecommendationEngine.recommendContent() analyzes learning records to prioritize weak vocabulary items. After the learner submits an answer through LearningPage.submitAnswer(), the system validates correctness and updates the mastery level through Vocabulary.updateMasterLevel(). For correct answers, the MasterLevel progresses toward MASTERED status; for incorrect answers, items are added to the mistake collection. When vocabulary reaches MASTERED status, ReviewService.createReviewSchedule() automatically creates a spaced repetition plan using the SM-2 algorithm, setting the initial review interval to 1 day with an easiness factor of 2.5.

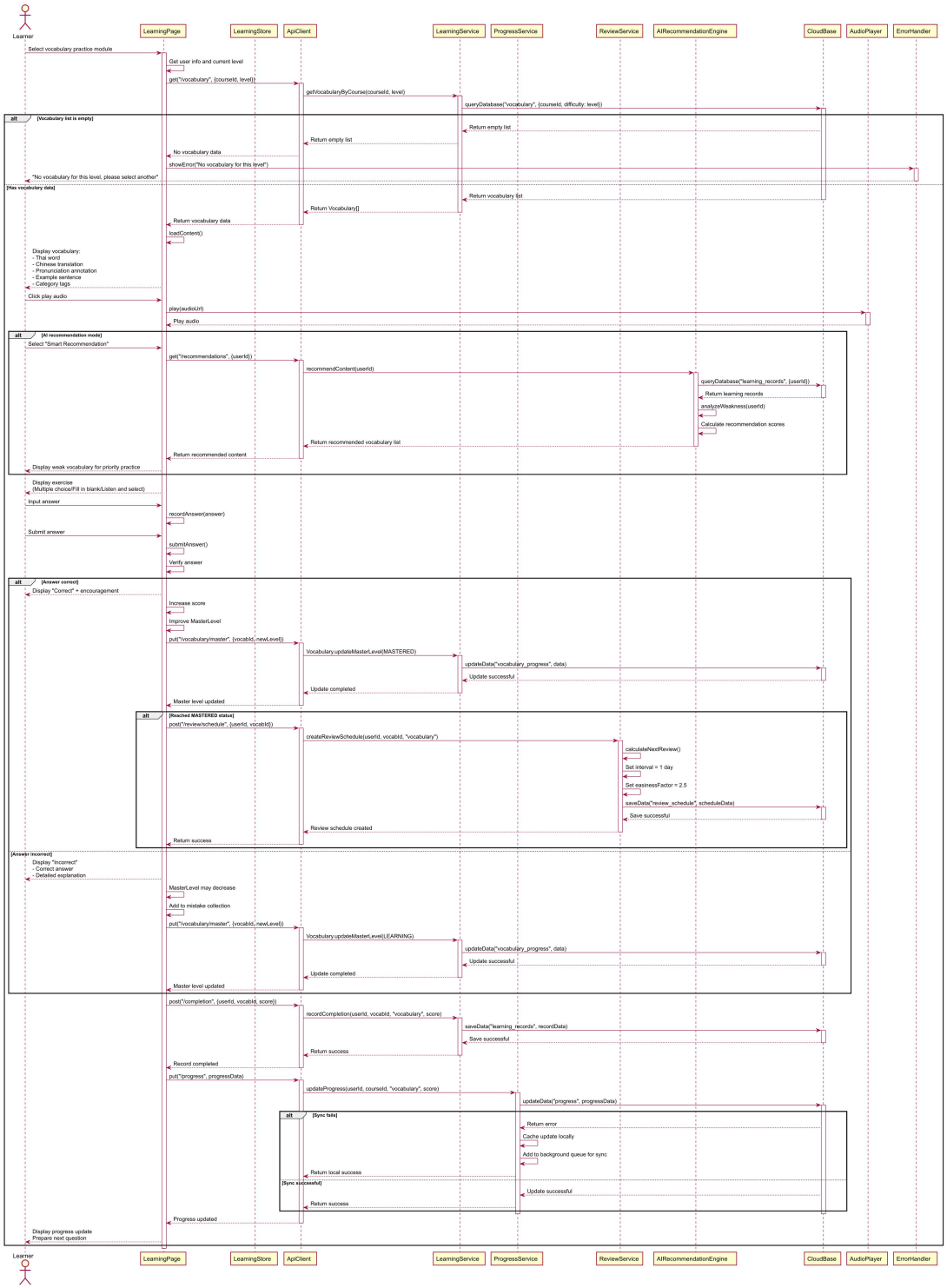

Figure 3.9 Vocabulary Practice Sequence Diagram

5. Pronunciation Practice Sequence Diagram

Pronunciation practice demonstrates the audio recording workflow that prepares user submissions for AI assessment. Figure 3.10 shows the recording interaction sequence. When a learner clicks "Practice Pronunciation" on a learning content page, the system displays the target text and plays standard pronunciation through AudioStore.playAudio(). The learner can optionally skip the standard audio playback. Upon clicking "Start Recording", the system checks microphone permissions and, if granted, calls AudioStore.recordAudio() to begin capturing audio using the Web Audio API. The diagram illustrates real-time recording status display with visual indicators like waveform animation. After clicking "Stop Recording", the system saves the AudioData and presents three options: re-record (returning to the recording state), play recording (previewing the captured audio through AudioPlayer.play()), or submit for assessment (proceeding to UC08). The exception handling covers microphone permission denial scenarios, guiding users to device settings, and recording save failures that prompt immediate retry with error logging.

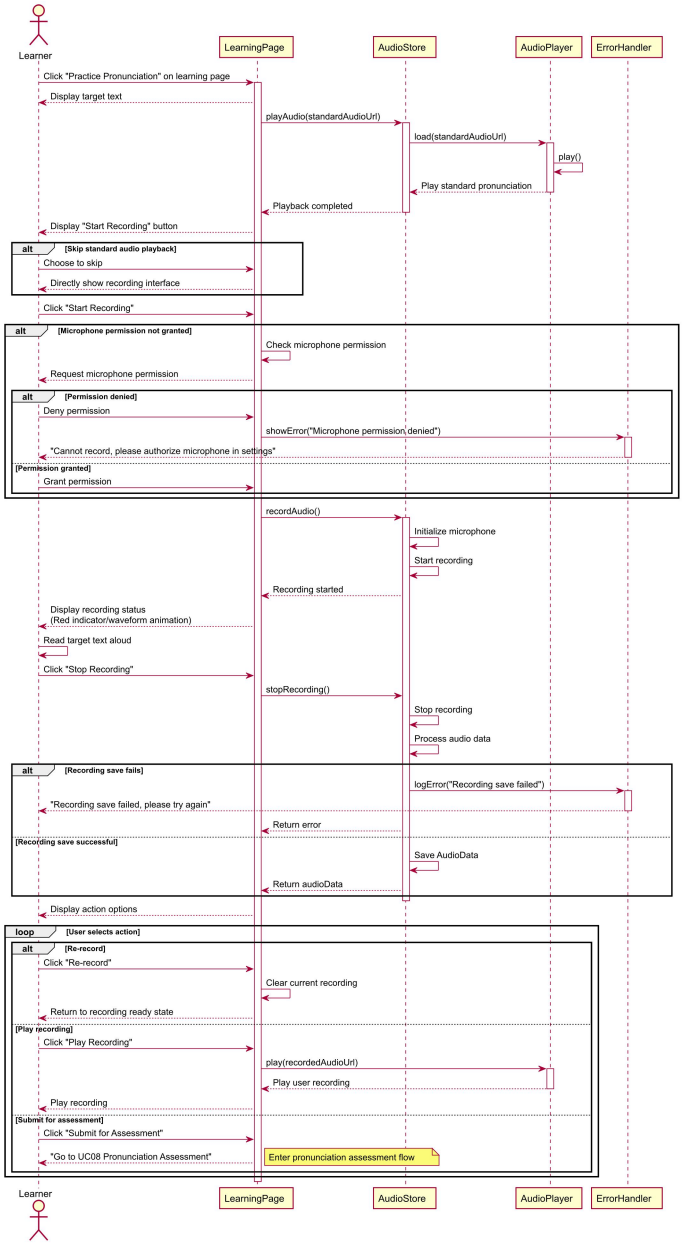

Figure 3.10 Pronunciation Practice Sequence Diagram

6. Sentence Construction Sequence Diagram

Sentence construction demonstrates the interactive grammar learning workflow with immediate feedback. Figure 3.11 illustrates the sentence building sequence. The learner selects the sentence learning module, and LearningService.getSentencesByCourse() retrieves sentences filtered by difficulty level from CloudBase. The LearningPage displays the Thai sentence, Chinese translation, grammar pattern explanation, and difficulty level. After playing the sentence audio through Sentence.playAudio(), the system presents construction exercises in either drag-and-drop format (where learners arrange vocabulary items to form sentences) or fill-in-the-blank format. The diagram includes an alternative "listen and construct" mode where audio plays without visible text, challenging learners to construct sentences purely from auditory input. When the learner submits through LearningPage.submitAnswer(), the system validates sentence structure and provides detailed feedback: completely correct answers receive full scores and automatic review plan creation; partially correct answers show marked correct/incorrect parts with partial scores; incorrect answers display the correct sentence with detailed grammar explanations and vocabulary highlighting. The LearningService.recordCompletion() method logs completion, and ProgressService.updateProgress() increments the sentence completion count.

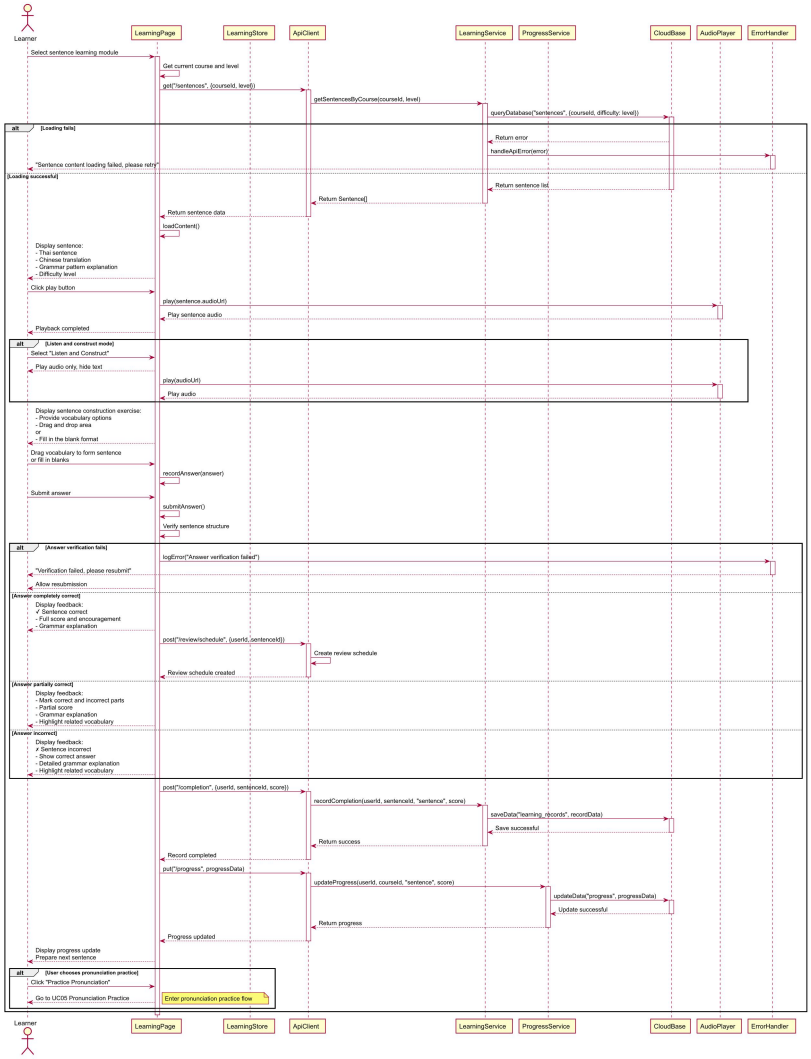

Figure 3.11 Sentence Construction Sequence Diagram

7. Article Reading Sequence Diagram

Article reading demonstrates the advanced comprehension workflow with vocabulary assistance and audio support. Figure 3.12 shows the complete reading interaction sequence. The learner selects the article reading module, and LearningService.getArticlesByCourse() retrieves articles filtered by difficulty level. After selecting a specific article, LearningPage.loadContent() fetches the article details through Article.displayArticle(), which displays the title, Thai original text, toggleable Chinese translation, category, and word count. The system enhances the reading experience by calling Article.highlightVocabulary() to emphasize previously learned vocabulary items in the text. The diagram illustrates multiple interactive features: clicking vocabulary words displays popup definitions retrieved through the API; clicking the favorite button adds words to the personal vocabulary list via CloudBase.saveData(); playing full article audio or paragraph audio through Article.playAudio() and AudioPlayer; and engaging in read-along mode where sentences play individually with pauses for learner repetition and optional recording assessment. Upon marking the article as "Completed", the system records completion through LearningService.recordCompletion() and updates progress statistics. The optional comprehension questions feature displays multiple-choice questions after reading, validates answers, and provides detailed explanations with score feedback.

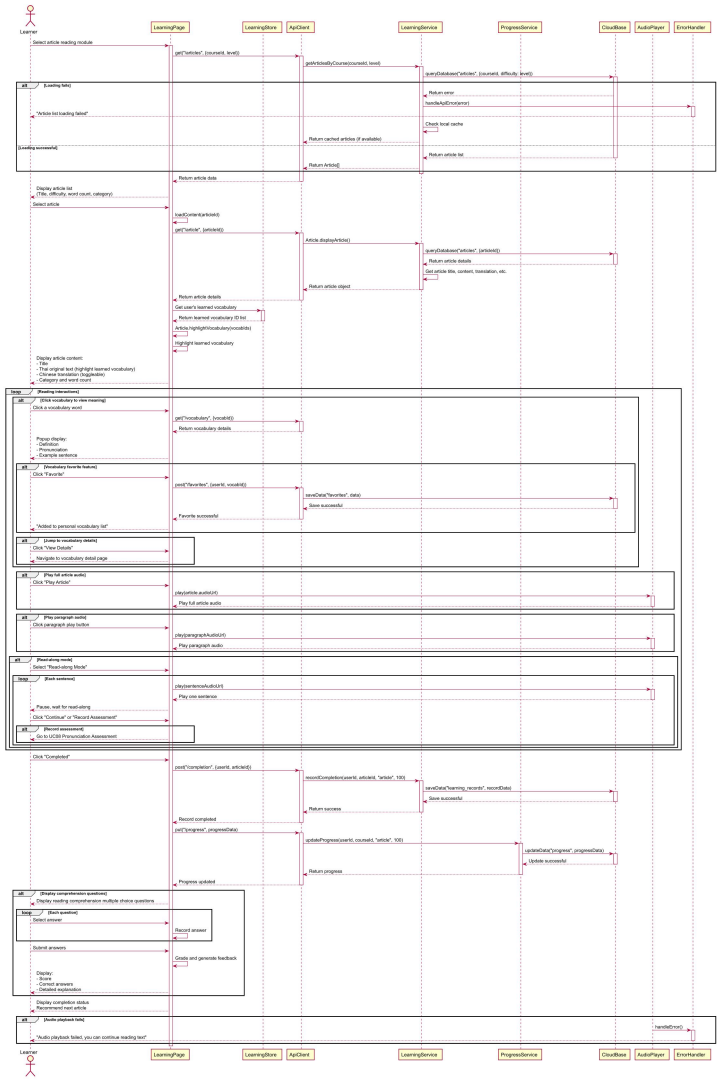

Figure 3.12 Article Reading Sequence Diagram

8. Pronunciation Assessment Sequence Diagram

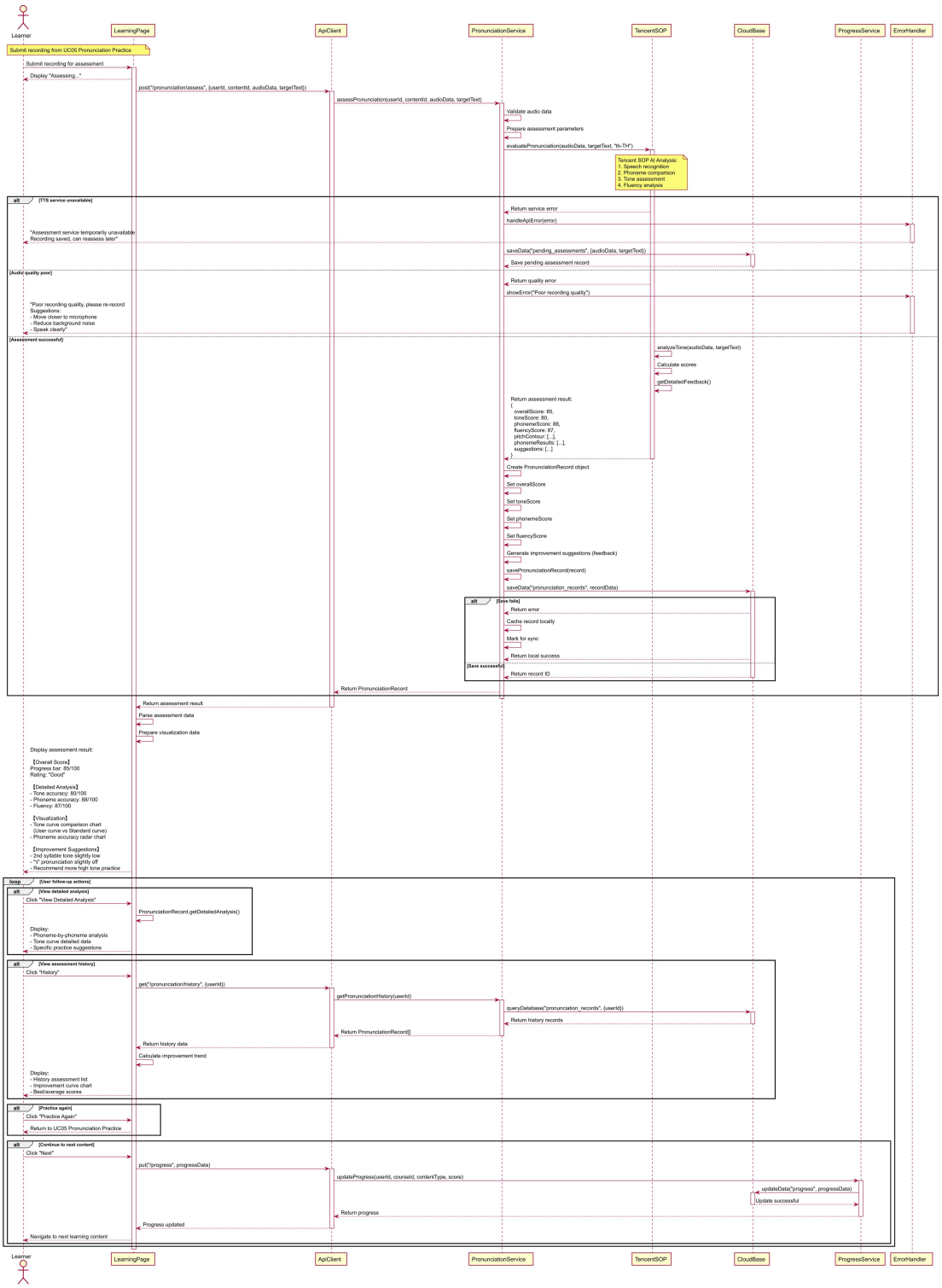

Figure 3.13 Pronunciation Assessment Sequence Diagram

Pronunciation assessment demonstrates one of the system's core innovative features: real-time AI-powered pronunciation evaluation. Figure 3.13 illustrates the comprehensive assessment workflow initiated from UC05 Pronunciation Practice. When the learner submits a recording for assessment,

PronunciationService.assessPronunciation() validates the audio data and calls TencentSOP.evaluatePronunciation() with the audio buffer, target Thai text, and language specification. The Tencent SOP AI engine performs multi-dimensional analysis including speech recognition, phoneme comparison, tone evaluation, and fluency measurement. The diagram shows three possible outcomes: if the TTS service is unavailable, the system saves the recording to pending_assessments collection for later evaluation; if audio quality is poor, the system provides specific recording improvement suggestions; if assessment succeeds, TencentSOP returns comprehensive scores including overall score (0-100), tone accuracy (weighted 40%), phoneme accuracy (weighted 35%), and fluency score (weighted 25%), along with pitch contour data and phoneme-level results. The service creates a PronunciationRecord object with all scores and improvement suggestions, saves it to CloudBase through PronunciationService.savePronunciationRecord(), and displays results on LearningPage with visualizations including progress bars, tone curve comparison charts overlaying learner's pitch with reference patterns, and phoneme accuracy radar charts. Users can view detailed analysis through PronunciationRecord.getDetailedAnalysis(), access assessment history showing improvement curves via PronunciationService.getPronunciationHistory(), practice again by returning to UC05, or continue to next content with automatic progress updates.

9. Spaced Repetition Review Sequence Diagram

Spaced repetition review demonstrates the system's second core innovative feature: scientifically-optimized review scheduling using the SM-2 algorithm. Figure 3.14 illustrates the complete review workflow. When the learner opens the application, the system automatically calls ReviewService.getDueReviews() to query the review_schedule collection for items with nextReviewDate less than or equal to today. If no reviews are due, the system displays "No review tasks" and recommends new content; otherwise, it shows a review reminder indicating the count and types of pending reviews. The diagram includes an AI smart review recommendation feature where AIRecommendationEngine.analyzeWeakness() queries learning records to calculate error rates by topic and prioritizes poorly mastered content. After the learner clicks "Start Review", ReviewService.getReviewList() retrieves and sorts items by priority using ReviewSchedule.isReviewDue(). The learner practices each item (similar to UC03/UC04/UC06), and the system grades the response, assigning a quality value from 1-5. The SM-2 algorithm implementation is shown in detail: quality 1-2 resets the interval to 1 day and decreases easinessFactor by 0.2 (minimum 1.3); quality 3 maintains current interval; quality 4-5 increases the interval by multiplying with easinessFactor (newInterval = interval × easinessFactor) and increases easinessFactor by 0.1 or 0.15 respectively. ReviewSchedule.calculateNextReview() applies the Ebbinghaus forgetting curve principles to set the nextReviewDate with a maximum interval cap of 180 days. The updated schedule is saved through ReviewService.updateReviewSchedule() with fallback to local caching if cloud sync fails, and ProgressService.updateProgress() increments the review count. Upon completing all reviews, the system displays comprehensive statistics and generates a weekly review plan through AIRecommendationEngine.generateWeeklyPlan() showing the distribution of reviews across the next 7 days.

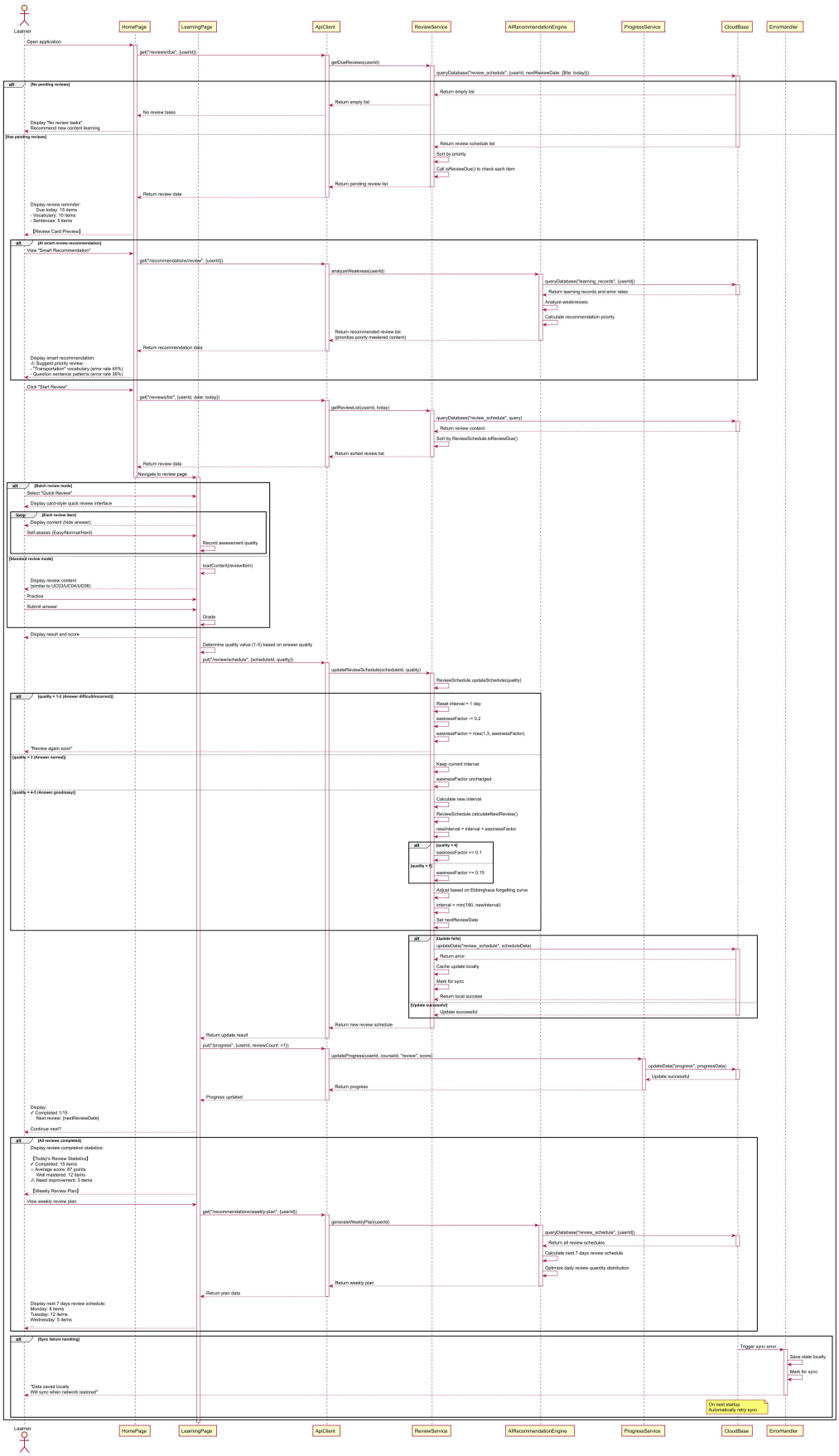

Figure 3.14 Spaced Repetition Review Sequence Diagram

3.5.2 Administer Sequence Diagram

1.User Management Sequence Diagram

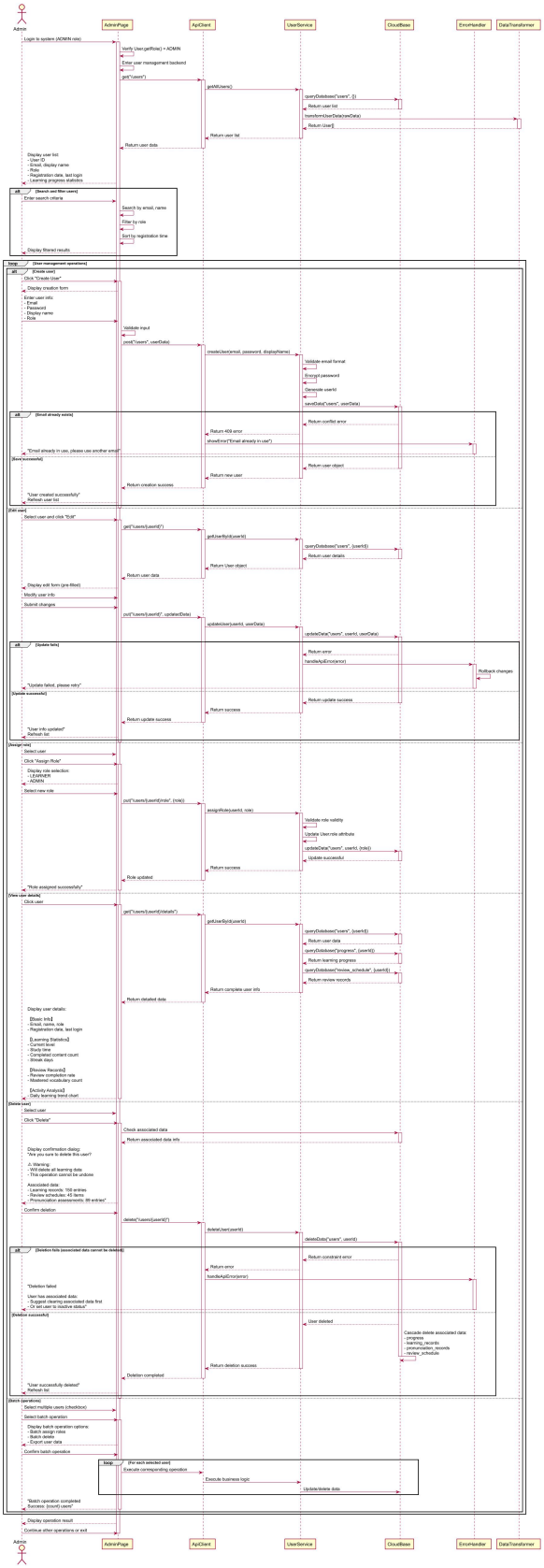

Figure 3.15 User Management Sequence Diagram

User management demonstrates the comprehensive administrative control interface for system oversight. Figure 3.15 illustrates the complete user administration workflow. The administrator logs into the system, and the system verifies User.getRole() equals ADMIN before granting access to the user management backend. UserService.getAllUsers() queries the CloudBase database to retrieve all user accounts, which are transformed through DataTransformer.transformUserData() and displayed with user ID, email, display name, role, registration date, last login, and learning progress statistics. The diagram shows five primary administrative operations with complete flows: Create User, where the administrator enters new user information, UserService.createUser() encrypts the password, generates a unique user ID, and CloudBase.saveData() persists the account (with exception handling for duplicate email addresses); Edit User, where UserService.getUserById() retrieves existing user details for modification and UserService.updateUser() applies changes with rollback capability on failure; Assign Role, where UserService.assignRole() validates and updates the User.role attribute between LEARNER and ADMIN; View User Details, where the system queries multiple collections (users, progress, review_schedule) to compile comprehensive user information including learning statistics, review records, and activity analysis displayed with trend charts; and Delete User, which checks for associated data, displays a detailed warning confirmation dialog, and calls UserService.deleteUser() to execute cascade deletion of all related records from progress, learning_records, pronunciation_records, and review_schedule collections, with constraint error handling suggesting deactivation instead of deletion when learning records exist. The diagram includes alternative flows for search and filter operations by email, name, or role, and batch operations for selecting multiple users to assign roles or delete in bulk.

2. Course Management Sequence Diagram

Course management serves as the administrative interface for comprehensive course organization, with its workflow illustrated in Figure 3.16. When administrators access the backend, CourseService.getAllCourses() queries CloudBase to retrieve all courses with details including names, descriptions, difficulty levels, creators, dates, status, and content statistics. The sequence diagram details key operations: course creation involves entering information, generating courseId via CourseService.createCourse(), saving through CloudBase.saveData(), and activating with duplicate name detection; course editing uses CourseService.getCourseById() for retrieval and CourseService.updateCourse() for modifications; content addition offers direct upload via CloudBase.uploadFile() (MP3/WAV, 10MB limit) or TTS generation where TencentTTS.textToSpeech() creates configurable Thai speech; content is saved via CourseService.addAlphabet() and similar methods. The system handles content editing through CourseService.getCourseContent(), batch importing from Excel/CSV with validation and preview, course previewing as learners, activation/deactivation via Course.activate()/deactivate(), and deletion by checking learning records, warning of cascade effects, executing CourseService.deleteCourse() or suggesting deactivation if users have learning history.

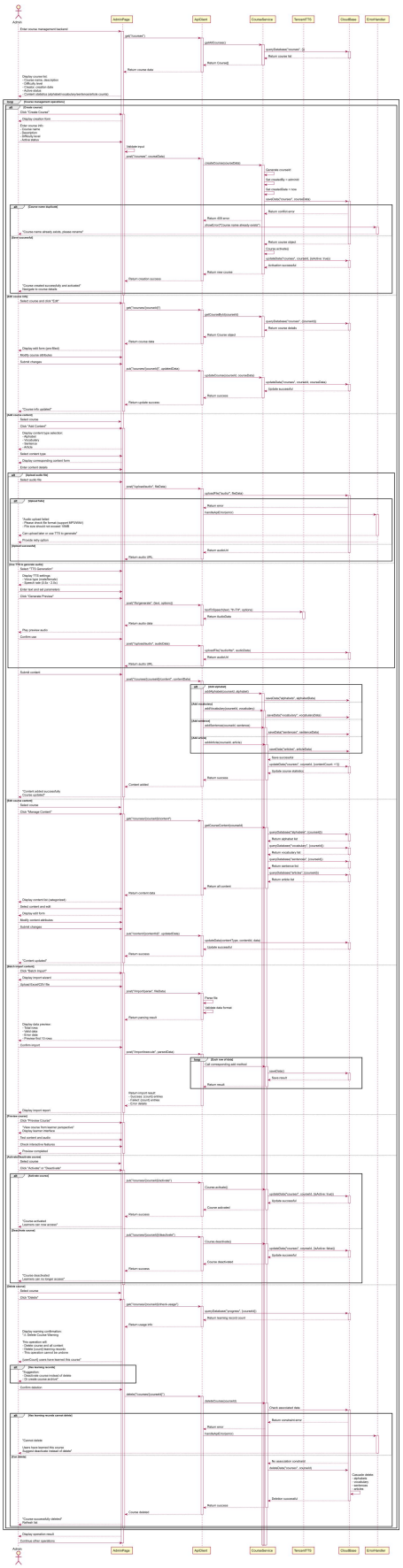

Figure 3.16　Course Management Sequence Diagram

3.6 Database Design

The database design follows a document-based schema supported by CloudBase's NoSQL database. Major collections include Users, Lessons, Exercises, Records, and Progress. Each collection stores data in JSON format, enabling dynamic updates and efficient querying [2][9].

Phase	Collection	Description
1	Users	Stores user profile, authentication credentials, and preferences
2	Lessons	Contains lesson metadata such as level, type, and difficulty
3	Exercises	Records exercises and related content links
4	PronunciationRecords	Stores AI-generated speech evaluation data
5	Progress	Tracks user performance and completion history

Data integrity and consistency are maintained through CloudBase's real-time synchronization, ensuring immediate reflection of learning progress across devices [9].

3.7 Security and Privacy Design

Security is implemented using CloudBase Authentication, which provides user identity verification through email and third-party login methods. All sensitive data, including pronunciation recordings, are securely stored in Cloud Object Storage (COS) with access control policies [10]. Communication between client and server is encrypted using HTTPS, ensuring protection against data interception or manipulation.

Moreover, user consent is required before audio data processing, complying with general data protection regulations (GDPR) principles for privacy and transparency [10].

3.8 Summary

This chapter detailed the system design, including architecture, UML diagrams, database schema, and security model. The use of React Native for cross-platform development [1], Tencent CloudBase for serverless backend [2], and AI-powered speech processing services [3][4] form the technological foundation of the application. The system's modular, scalable, and secure design ensures that it can effectively support the personalized Thai language learning experience envisioned in this project.

References

1. Burston, J. (2015). Twenty years of MALL project implementation: A meta-analysis of learning outcomes. ReCALL, 27(1), 4–20.
2. Chowdhury, S. A., Cai, Y., Sultana, N., Rasel, M. I., & Alam, F. (2023). Automatic pronunciation assessment: A review. In Findings of the Association for Computational
3. Linguistics: EMNLP 2023 (pp. 8317–8338). Association for Computational Linguistics.
4. Kaur, J., & Kumar, A. (2020). Automatic speech recognition system for tonal languages: State-of-the-art survey. Archives of Computational Methods in Engineering, 27(4),1213–1237.
5. Li, L., Zhang, Y., & Chen, H. (2023). The impact of automatic speech recognition technology on second language pronunciation and speaking skills of EFL learners: A mixed methods investigation. Frontiers in Psychology, 14, Article 1210187.
6. Lin, B., & Wang, L. (2024). AI-enhanced spaced repetition: Integrating the Ebbinghaus forgetting curve and SM-2 scheduling for English vocabulary learning. In Proceedings of the International Conference on Educational Technology. Academia.edu.

Technical Documentation:

[1] Facebook Open Source. (2023). *React Native Documentation.* Retrieved from https://reactnative.dev/

[2] Microsoft. (2023). *TypeScript: JavaScript with syntax for types.* Retrieved from https://www.typescriptlang.org/

[3] Mozilla Developer Network. (2024). *Web Audio API.* Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

[4] Tencent Cloud. (2024). *Tencent CloudBase Developer Guide.* Retrieved from https://cloud.tencent.com/document/product/876

[5] Tencent Cloud. (2024). *Speech Oral Evaluation API Documentation.* Retrieved from https://cloud.tencent.com/document/product/884/47495

[6] Tencent Cloud. (2024). *Text-to-Speech (TTS) Service Overview.* Retrieved from https://cloud.tencent.com/document/product/1073

[7] Tencent Cloud. (2024). *Cloud Object Storage (COS) Overview.* Retrieved from https://cloud.tencent.com/document/product/436

[8] Tencent Cloud. (2024). Cloud Functions Documentation. Retrieved from https://cloud.tencent.com/document/product/876/41761

[9] Tencent Cloud. (2024). CloudBase Database Overview. Retrieved from https://cloud.tencent.com/document/product/876/18441

[10] Tencent Cloud. (2024). Cloud Security and Compliance Center. Retrieved from https://cloud.tencent.com/document/product/301