Pronunciation Coach – First Milestone

Fabian L. Velez Ocasio; Team 1

19 September 2025

Table of Contents

1. 1. Informative Part

1.1. 1.1 Teams

Team	Member Name	Role & Key Contributions
1	Alondra Arce	Team Leader & Architecture Specialist. Project coordination, architecture design, routing and state management.
1	Fabian Velez	Researcher & Documentation Lead. Domain research, UI/UX component system design, requirements analysis.
1	Julian A. Toro	Security & Research. Research on secure authentication methods and pipeline design.
1	Aryam Z. Diaz	Research & Quality Assurance. Research on login UI structure, input validation, and CI/CD/performance strategies.
1	Jaydiemar Vazquez	Architecture & Design. Research on clean architecture, state management, and home screen visual design.
1	Kevin Lara	Backend Research. Research on backend-as-a-service solutions (Firebase vs. Supabase).
1	Kevin Ruiz	Data & Offline Strategy. Research on data fetching, caching, and offline strategies for the Home Page.

1.2. 1.2 Current Situation, Needs, Ideas

1.2.1. 1.2.1 Current Situation

For native Spanish speakers, learning and pronouncing English words and phrases presents a significant challenge due to fundamental differences between the two languages. English, with its influences from Latin, Germanic, and Nordic languages, features complex and often inconsistent rules, which can be daunting for learners. The primary difficulties are rooted in pronunciation, where specific phonetic elements in English simply do not exist in Spanish. These challenges, while not always a complete barrier to being understood, can hinder professional integration and career advancement for ambitious non-native speakers.

Source: "Why is English pronunciation difficult for a Spanish speaker?", London Speech Workshop, https://londonspeechworkshop.com/why-is-english-pronunciation-difficult-spanish-speaker/

1.2.2. 1.2.2 Needs

- Need for accessible pronunciation practice outside classroom settings
- Requirement for immediate, objective feedback on pronunciation attempts
- Need for structured progression through phonetically challenging sounds
- Demand for visual reinforcement to complement auditory learning
- Need for tracking personal progress and identifying persistent difficulties
- Requirement for motivational systems to maintain consistent practice habits

1.2.3. 1.2.3 Ideas

- Interactive pronunciation exercises with instant feedback mechanism
- Phoneme-focused practice modules targeting specific sound challenges
- Visual comparison interface between learner and native speaker pronunciation
- Progressive difficulty system that adapts to user improvement
- Achievement system to encourage regular practice and milestone completion
- Personalized practice recommendations based on performance analytics

1.3. 1.3 Scope, Span, and Synopsis

1.3.1. 1.3.1 Scope and Span

Scope: Mobile-based language learning application specializing in pronunciation improvement. The first milestone focuses exclusively on building the front-end visual and architectural foundation.

Span: English pronunciation training for Spanish-speaking adults. This initial phase spans the development of the application's shell, including login and home screens, a reusable component library, and core app infrastructure (routing, state management).

The project encompasses mobile development with Flutter, initial architecture setup, and UI/UX design.

1.3.2. 1.3.2 Synopsis

Pronunciation Coach is a Flutter-based mobile application designed to help Spanish-speaking adults improve their English pronunciation through targeted exercises and AI-driven feedback. The project involves developing a robust architecture for handling authentication flows, creating a comprehensive component system for consistent UI/UX, implementing speech analysis functionality, and designing an engaging activity-based learning progression.

The solution aims to make pronunciation practice accessible, effective, and engaging through technology-enabled learning tools.

1.4. 1.4 Other Activities (Beyond Coding)

- Domain Engineering: Research on English phonetics and common Spanish speaker challenges.
- Requirements Analysis: User needs assessment and feature prioritization for the UI/UX.
- Architecture Design: Design of application routing structure, state management, and project organization following clean architecture principles.
- Research: Comprehensive analysis of secure authentication, backend solutions, data caching, CI/CD, and visual design.
- Documentation: Management of project plans, research findings, and technical specifications.

1.5. 1.5 Derived Goals

- Development of a reusable Flutter component library for educational applications.
- Establishment of a scalable and maintainable codebase using clean architecture principles.
- Creation of a robust authentication flow that can be integrated with a secure backend.
- Implementation of a responsive and accessible design system.

2. 2. Descriptive Part

2.1. 2.1 Domain Description

2.1.1. 2.1.1 Domain Rough Sketch

(This section is a preliminary sketch based on the research made by team 1)

- User: Spanish-speaking adult, motivated to learn, may be frustrated with current tools.
- Goal: Improve English pronunciation.
- Activity: Logs into app, sees progress, selects a practice module, records their voice, receives feedback, tracks improvement.
- System: Mobile app, requires login, has home dashboard, practice sections, profile.
- Data: User account, authentication tokens, progress data, practice history.

2.1.2. 2.1.2 Terminology

- Phoneme: The smallest unit of sound in a language that can distinguish words (e.g., /θ/ in "think" vs. /s/ in "sink").
- Authentication: The process of verifying a user’s identity (e.g., via email and password).
- JWT (JSON Web Token): A compact, URL-safe means of representing claims to be transferred between two parties, used for securing authentication.
- State Management: The handling of the state of the application (e.g., whether a user is logged in or not) in a predictable way.

- Component Library: A collection of reusable UI elements (buttons, input fields, cards) that ensure design consistency.
- Routing/Navigation: The mechanism for moving between different screens in the application.

2.1.3. 2.1.3 Domain Terminology vs Rough Sketch

The terminology was derived from analyzing the needs of the domain (language learning) and the technical solution (a Flutter app). Terms like **Phoneme** and **Progress** come from the educational domain, while **JWT**, **State Management**, and **Routing** are technical concepts required to build a secure and functional application shell.

2.1.4. 2.1.4 Narrative

A user, Maria, wants to improve her English pronunciation. She downloads the Pronunciation Coach app. Upon opening it, she is presented with a clean login screen. She enters her credentials and is securely authenticated. She arrives at her home page, which welcomes her and shows her current learning streak, recent progress, and suggests a new sound to practice. The app is intuitive, responsive, and makes her feel confident to start her practice session.

2.1.5. 2.1.5 Events, Actions, Behaviors

- Event: User presses the "Login" button.
- Action: The system validates the input fields and sends credentials to the authentication service.
- Behavior: If authentication is successful, the application's state changes to "authenticated," and the user is navigated to the Home screen.

2.1.6. 2.1.6 Function Signatures

(High-level domain operations, not final code)

- `authenticateUser(credentials: Credentials): AuthenticationResult` - Validates user credentials.
- `navigateTo(screen: ScreenName)` - Changes the current view of the application.
- `getUserProfile(userId: UserID): UserProfile` - Retrieves the user's data for display on the home screen.

2.2. 2.2 Requirements

2.2.1. 2.2.1 User Stories, Epics, Features

Epic: User Authentication * As a new user, I want to log in with my email and password so that I can access my personalized learning content. * As a user, I want to see clear error messages if my login fails so that I can correct my information. * As a user, I want my session to be managed securely so that my account remains protected.

Epic: Application Foundation * As a developer, I want a well-organized project structure following clean architecture so that the codebase is maintainable and scalable. * As a developer, I want a

central state management solution so that the user's authentication state can be shared across the app. * As a developer, I want a library of reusable UI components so that we can ensure design consistency and speed up development.

Epic: Home Dashboard * As a user, I want to see a welcoming home screen after logging in so that I can understand my current progress and see what to do next. * As a user, I want the app to work offline and show my cached data so that I can still see my progress without an internet connection.

2.2.2. 2.2.2 Personas

- **Maria, the Motivated Learner**: A 28-year-old professional from Mexico. She uses English at work but is self-conscious about her accent. She is tech-savvy and uses her phone for most tasks. She needs structured, feedback-driven practice she can fit into her busy schedule.
- **Carlos, the Consistent Student**: A 45-year-old teacher from Colombia preparing to move to an English-speaking country. He is dedicated but has limited time. He needs clear goals, progress tracking, and motivation to practice daily.

2.2.3. 2.2.3 Domain Requirements

1. The system must restrict access to user-specific data until identity is verified (authentication).
2. The system must provide a clear and intuitive path for the user to begin their learning activities.
3. The system must present information (progress, goals) in a motivating and visually clear way.

2.2.4. 2.2.4 Interface Requirements

- The login screen shall have input fields for email and password.
- The login screen shall have a button with the label "Login".
- The home screen shall display a welcome message containing the user's name.
- The application shall transition from the login screen to the home screen upon successful authentication.

2.2.5. 2.2.5 Machine Requirements

- The application shall render correctly on iOS and Android devices.
- The initial app startup time shall be under 400ms on a mid-range device.
- The UI shall respond to user input (e.g., button presses) within 16ms for a smooth 60fps experience.

2.3. 2.3 Implementation

2.3.1. 2.3.1 Selected Fragments of Implementation

```
lib/
```

```
├──── core/
│    ├──── constants/
│    ├──── widgets/ (reusable components)
│    └──── providers/ (state management e.g., SessionController)
├──── features/
│    ├──── auth/
│    │    ├──── screens/
│    │    │    └──── login_screen.dart
│    │    └──── widgets/
│    └──── home/
│         ├──── screens/
│         │    └──── home_screen.dart
│         └──── widgets/
└──── main.dart
```

Screen Sketches:

- **Login Screen**: Based on Research #45 and #50. Features a centered card with a header, email field, password field (with toggle), a prominent "Login" button, and secondary links ("Forgot Password?", "Sign Up").
- **Home Screen**: Based on Research #42 and #48. Features a welcome message, a "Progress Card" displaying a streak, a "Daily Challenge" card with a CTA button, and a bottom navigation bar.

3. 3. Analytic Part

3.1. 3.1 Concept Analysis

The rough sketch identified key domain entities: User, Practice, and Progress. These abstractions led to the terminology that defines both the problem space (phoneme, feedback) and the solution space (authentication, routing, state).

The narrative connects these concepts into a coherent user journey, validating that the initial implementation focus (authentication and home screen) correctly establishes the foundation for the user’s interaction with the app.

The technical research (auth security, architecture) ensures the solution space concepts are implemented robustly.

3.2. 3.2 Validation and Verification

Testing Plans: As per Research #14 and #19, testing will include:

- Unit Tests: For the SessionController state changes (login/logout logic).
- Widget Tests: For Login Screen input validation and button enabling/disabling.
- Integration Tests: For the complete flow from Login to Home navigation.

Walkthroughs: The team will conduct peer code reviews on all pull requests to verify architecture adherence and code quality.

Scenarios used for validation:

1. Happy Path: Enter valid credentials → Login button enables → Press button → Navigates to Home.
2. Validation Error: Enter invalid email format → Error message appears under field → Login button remains disabled.
3. Authentication Error: Enter incorrect credentials → SnackBar with generic error message appears.
4. Offline Scenario: With no internet, the Home screen should still render any cached data.