Pronunciation Coach Proposal

First Milestone

Prof. Marko Schütz

Introduction to Software Engineering – INSO 4101

Sec. 080

Professor Marko Schütz

Wednesday, September 17th, 2025

Table of Contents

1. 1. Informative Part

1.1 Teams

Team 1:

Name	**Role**	**Tasks**
Alondra Arce	Team Leader	Architecture specialist
Fabian Velez	Developer	Documentation Lead
Julian Toro	Developer	Security & Research
Jaydiemar Vazquez	Developer	Architecture & Design
Kevin Lara	Developer	Backend Research
Kevin Ruiz	Developer	Data & Offline Strategy
Aryam Diaz	Developer	Research & Quality Assurance

Team 2:

Name	Role	Tasks
Jorge L. Luna	Team leader	Project coordination, research analysis of phonemes, implement Vosk as speech-to-text for testing
Daniel Reyes	Developer	Research speech-to-text models, provide findings to team, implement Whisper (OpenAI) as speech-to-text for testing
Uriel Rosado	Developer	Research pronunciation comparison methods, evaluate Montreal Forced Alignment models
Claudia Guzmán	Developer	Explore AI-based user feedback (chatbot), study phonetic dictionaries
Diego Rios	Developer	Investigate playback-based pronunciation training, creation of phonetic dictionaries
Omar Cordero	Developer	Analyze speech-to-text libraries, develop algorithms for phoneme-level pronunciation scoring
José Valentín	Developer	Evaluate STT options, research English dictionaries
Noel Colón	Developer	Research accent variation in pronunciation, build orthographic dictionary

Team 3:

Name	Role	Tasks
Alex Morales	Team Leader	Led project coordination and research, conducted UX research on gamification strategies for dashboards and quizzes to boost engagement; created Flutter UI components for progress dashboards showing learning metrics.
Jorge Belgodere	Manager	Monitored project progress, facilitated communication across teams, provided guidance, and ensured timely delivery of research and development goals.
Pedro Morales	Manager	Monitored project progress, facilitated communication across teams, provided guidance, and ensured timely delivery of research and development goals.
Segmar Rosado	Manager	Monitored project progress, facilitated communication across teams, provided guidance, and ensured timely delivery of research and development goals.
Abdiel Velazquez	Developer	Researched adaptive difficulty algorithms and recommended the Elo rating system to personalize pronunciation challenges based on user accuracy.
Ignacio Gomez	Developer	Designed regional/cultural name pronunciation packs using IPA-based TTS and native recordings; developed a Flutter daily challenge dashboard widget and page with XP and streak rewards.
Enrique Vilela	Developer	Designed and implemented a daily streak and points tracking system with gamification features; researched and recommended best practices for daily challenge score and streak systems.
Gabriel Visbal	Developer	Researched sourcing native pronunciation audio, recommending YouGlish integration; built a simple UI for users to select their preferred learning pace.

Ivan Morales	Developer	Explored backend progress analytics options with xAPI and open-source LRS for flexible tracking; recommended progressive disclosure UX for adaptive dashboards to reduce cognitive overload.
Jan Davey	Developer	Researched real-time pronunciation feedback using MFCC and Integral Approximation for similarity scoring; implemented a confirmation page that displays the user's chosen learning pace after selection.
Bruno Vergara	Developer	Investigated Flutter mic/audio packages for real-time speech processing and visualization; developed a daily challenge prompt UI to motivate users with daily tasks.

Team 4:

Name	Role	Tasks
Joy Martinez	Team Leader	➢ [Research] Randomization in Multiple Choice Quiz: - Goal: Research how to add randomized letters into a phonetics quiz - Result: Using the math library returned the best results ➢ [Feature] Script to Check English Validation for Input Words: - Goal: Create a simple to read Python function that checks if a word exists in English and return a Boolean result- Result: Python Script was reported functional alongside test cases
Yediel Acosta	Developer	➢ [Research] Text → IPA Translator: - Goal: Find ways to Translate input text into its respective phonetic symbols - Result: Given Python's CMU Dictionary, one can translate Text → ASCII → IPA ➢ [Feature] Speech → Text Translator: - Goal: Allow a User to verbally communicate with the app, with it returning Text translation - Result: Dart Script was reported functional and tested for accuracy

		➢ [Documentation] Creating Team 4's Documentation for Milestone #1 ➢ [Presentation] Participating in Team 4's Video for Milestone #1
Adriel Bracero	Developer	➢ [Research] Stress Placement and Basic Word Intonation: - Goal: Research a simple method that detects stress and word intonation for words and phrases - Result: Feature-based stress detection can be lightweight and accurate, useful for Flutter ➢ [Research] On-Device Pitch Extraction & Normalization: - Goal: Research for low-cost methods to extract fundamental frequency on mobile devices- Result: Autocorrelation based techniques are low cost and work well, used for shorter recordings ➢ [Research] Noise Robustness and Preprocessing: - Goal: Research into lightweight preprocessing pipelines for noise-handling techniques - Result: Spectral gating is easy to implement and fast; RNNoise provides stronger noise suppression with low cost & real-time capable
Diego Hernandez	Developer	➢ [Research] Slow-Motion for Pronunciation Practice: - Goal: Research how to allow slow-motion playback for audios - Result: just_audio supports 0.75 & 0.5 speeds in Flutter; Back End with Python/Librosa allows offline audio slow-down
Iralys Sanchez	Developer	• **Iralys Sanchez:** ➢ [Research] Validation of Spoken Word Input: - Goal: Research methods to validate spoken word input - Result: CMUdict, Wiktionary, and Free Dictionary API offer tools for input validation ➢ [Research] Methods for Building Quiz Word/Question Bank: - Goal: Research for quiz word bank structure design to store easy to

		load data for Python & Flutter/Dart - Result: JSON, CSV, and SQLite offer notation tools or lightweight database responses
Daniel Reyes	Developer	➢ [Research] Whisper Speech → Text Setup: - Goal: Research Whisper’s AI Speech → Text functionality and setup in Python - Result: Whisper is a reliable choice if accuracy is prioritized over latency

1.2 Current Situation, Needs, Ideas

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 Needs

Team 1.

- 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

Team2.

Pedagogical Needs:
• Clear and actionable feedback on pronunciation beyond a binary "correct/incorrect."
• Independent practice opportunities without constant teacher supervision.
• Adaptability to different accents and common pronunciation challenges.
• Motivation and progress tracking to support long-term learning.

Accessibility Needs:
• Affordable solutions that reduce or eliminate subscription costs.
• Offline availability, allowing use without continuous internet access.
• Simple, intuitive tools that can be used by learners of different ages and technical skills.

User Needs:
• Tools for learners to monitor their own progress over time.
• Mechanisms to highlight frequent pronunciation errors for focused practice.
• Transparent and explainable evaluation methods, showing why certain pronunciations are
marked as incorrect or needing improvement.

Team 3.

Primary Needs:
- Language learners need consistent daily practice opportunities to improve pronunciation skills- Language learners need immediate feedback on pronunciation accuracy to correct mistakes early
- Language learners need motivation systems to maintain long-term engagement with pronunciation practice
- Language learners need personalized learning experiences that adapt to their skill level and available time

- Language learners need progress tracking to visualize improvement over time

Secondary Needs:
- Educators need tools to supplement classroom pronunciation instruction
- Language learning institutions need data on student engagement and progress
- Native speakers need platforms to contribute authentic pronunciation samples

Team 4.

Due to the problem at hand, the necessity for an app that focuses on practicing English speech and allow for a broad practice for pronunciation, unlike most apps. An app of this style would also require cultivating discipline and motivation for the student as they use it, as a learning app that doesn't captivate a student's attention will fail as a learning app.

1.2.3 Ideas

Team 1.

- 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

Team 2.

Pedagogical Features:
• Provide detailed feedback that highlights mispronounced words or phonemes, with suggestions for improvement.
• Offer sentence and word practice modes to let learners focus on specific areas.
• Support accent-aware evaluation so that learners with different linguistic backgrounds receive fair and useful feedback.

• Include progress tracking dashboards that visualize learner improvement over time.

Accessibility Features:
• Maintain a low-cost model by relying on open-source speech recognition and feedback methods.
• Design a simple, user-friendly interface suitable for both beginners and advanced learners.

User Features:
• Allow learners to monitor their own progress and identify areas for focused practice.
• Provide clear visualizations of frequent pronunciation errors.
• Ensure transparency in feedback by showing how evaluations are derived (e.g., highlighting words or phonemes instead of giving only scores).
• Enable exporting progress reports for personal review or sharing with tutors/mentors.

Team 3.

Core System Features:
- Daily challenge system with streak tracking to maintain consistent practice habits
- Adaptive difficulty system using Elo rating to personalize pronunciation challenges
- Multi-pace learning options (Casual: 5min/day, Standard: 15min/day, Intensive: 30min/day)
- Real-time pronunciation feedback using MFCC analysis and similarity scoring
- Regional pronunciation packs focusing on culturally relevant names and phrases
- Gamified progress tracking with XP points, badges, and visual progress indicators

Implementation Approach:
- Flutter-based mobile application for cross-platform accessibility
- Local data storage for offline capability and privacy
- Progressive disclosure UI to reduce cognitive overload for beginners
- Integration with native pronunciation audio sources like YouGlish

Team 4.

• To fulfill these needs, the team will develop a domain description, requirements prescription, software design, and implementation. A new platform called "Pronunciation Coach" will allow students to study and learn by themselves English speech by various exercises and activities. Additionally, the app will reward interaction to improve the students' motivation and build upon a habit to learn for discipline.

1.3 Scope, Span, and Synopsis

1.3.1 Scope and Span

Team 1.

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.

Team 2.

Scope: A digital language learning app, specifically tools designed to help learners improve spoken English. This includes general language apps, pronunciation tools, and speech analysis technologies.

Span: The Pronunciation Coach focuses on a specific segment of this domain: an application that provides learners with detailed, actionable feedback on their pronunciation at both word and phoneme levels. The project emphasizes accessibility, low-cost solutions, and visual progress tracking for independent learners.

Team 3.

Scope: This project involves the research and development of a functional pronunciation learning application. It involves both front-end and back-end work, focusing on features that enhance user motivation and productivity, provide personalized learning experiences, and accurately track user progress. The scope involves exploring best practices in gamification, adaptive difficulty, speech processing, and data analytics to develop a comprehensive and effective pronunciation learning app.

Span: Within this scope, the project so far targets the implementation of a Daily Challenge system incorporating score accumulation, streak mechanics, and bonus rewards to encourage consistent practice. It also focuses on real-time pronunciation feedback using advanced speech analysis techniques, the creation of culturally relevant Regional Name Pronunciation Packs, and the establishment of a robust backend progress analytics infrastructure. The span ensures detailed attention to the integration of these components to deliver a seamless, motivating, and personalized user experience.

Team 4.

Scope: The “Pronunciation Coach” project will create an app for English speech practice, project which will be completed by the end of November 2025. Said app will test English speech and construct the student’s discipline. Therefore, the project yearns for students to improve their English speech and keep them interested in learning by constant use of the application. The target audience is whoever is willing to receive individual coaching. To access the “Pronunciation Coach”, the student must have internet access, a clean environment with minimal background noise, and any mobile app (IOS or Android) or web browser (Chrome, Microsoft Edge, Firefox, etc.)

Span:
• Time Frame: 4 months and 3 weeks, to submit at the end of November 2025
• Front End: Flutter and Dart
• Back End: Python
• Architecture: Client-Server

1.3.2 Synopsis

Team 1.

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.

Team 2.

The Pronunciation Coach is a software tool aimed at helping language learners improve their English pronunciation. By leveraging open-source speech-to-text models, the application evaluates user speech at the word and phoneme levels, highlights errors, and provides clear, actionable feedback. The tool is designed to track progress over time to motivate continued practice. This solution addresses the gap in current language learning tools that often provide minimal or non-specific pronunciation feedback.

Team 3.

The Pronunciation Coach project develops an interactive mobile application that addresses pronunciation learning challenges through structured daily challenges, real-time feedback, and gamified progress tracking. The project follows software engineering best practices, beginning with comprehensive domain analysis of language learning behaviors and pronunciation coaching methodologies.
The development process includes detailed requirements engineering to capture user stories and personas, followed by software architecture design emphasizing modular components for audio processing, user interface management, and progress tracking. Implementation utilizes Flutter framework for cross-platform mobile deployment, with careful attention to offline capability and user privacy through local data storage. The project incorporates research-driven features including MFCC-based pronunciation analysis, Elo rating systems for adaptive difficulty, and regional pronunciation packs with native speaker audio. Validation

involves user testing of gamification elements, pronunciation feedback accuracy, and long-term engagement metrics. The end result provides language learners with a comprehensive tool supporting consistent practice habits and measurable pronunciation improvement.

Team 4.

A lot of people who are interested in learning English have been impacted by multiple factors when it comes to pronunciation, a key aspect of language. Some of the obstacles are the lack of immediate feedback on classrooms, the cost of entrance for tutors, and the absence of speech depth by apps. These negatively affect a student's motivation, which leads to inconsistent practice and, ultimately, failure. To dimmish this problem, the team has started the Pronunciation Coach project, which allows students to learn and practice their English speech by a collection of activities, which rewards the student to improve motivation, create a habit and thus, learning discipline. For implementation, the Pronunciation Coach app will be compromised by a Flutter and Dart Front End and a Python Back End, working with a Client-Server Architecture. This project shall be developed in 4 months, finishing development by the end of November 2025The target audience for this project will be any student looking to improve their English speech or practice it. It's expected by users to have internet connection to access the app, and a minimally noisy environment to improve learning and comprehension.

1.4 Other Activities (Beyond Coding)

Team 1.

- 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.

Team 2.

Domain Engineering:
• Studying language learning techniques, phonetics, and pronunciation challenges.
• Reviewing existing STT models (Whisper, Vosk) and their suitability for offline evaluation.
3• Exploring Montreal Forced Alignment (MFA) for phoneme-level alignment and error detection, evaluating its potential for accurate feedback in pronunciation learning.

Requirements Analysis:
• Identifying user needs (learners) and mapping them to feature ideas.
• Defining system requirements for accuracy, offline performance, and usability.
• Exploring algorithms that will identify the user’s errors.

Architecture
• Designing the software architecture to integrate recording, STT processing, feedback generation, and progress tracking.
• Planning for modularity to allow swapping or updating speech recognition models.

Testing
• Conducting usability tests with learners to evaluate comprehension and effectiveness.
• Comparing STT outputs with target phrases to validate accuracy.
• Evaluating performance across different accents and age groups.

Deployment
• The Pronunciation Coach application should be lightweight, running smoothly on typical consumer devices such as laptops, tablets, and smartphones without excessive CPU or memory usage.
• The user interface should be intuitive and user-friendly, allowing learners of varying ages and technical proficiency to navigate recording, transcription, and playback easily.

• The system should support offline operation for core functions (recording, transcription, playback) to ensure accessibility in environments with limited connectivity.
• Packaging and installation should be simple, requiring minimal setup for learners to start practicing immediately.

Team 3.

Domain Engineering: Comprehensive analysis of pronunciation learning patterns, gamification psychology, and language acquisition methodologies to establish foundational understanding of the problem space.
Requirements Engineering: Development of detailed user stories, personas, and functional requirements through stakeholder analysis and user research to ensure the system meets actual learner needs.
Software Architecture: Design of scalable, maintainable system architecture supporting modular audio processing, state management, and data persistence components.
Component Design: Detailed design of individual system components including pronunciation analysis engines, progress tracking systems, and user interface elements.
Testing Strategy: Development of comprehensive testing approaches including unit tests for business logic, integration tests for audio processing pipelines, and user acceptance testing for gamification effectiveness.
Deployment Planning: Analysis of mobile app store requirements, platform-specific considerations, and update distribution strategies.

Team 4.

• Multiple Research on topics that can be implemented in the "Pronunciation Coach" app as the domain allows it
• Discuss adjacent topics that allow for a domain expansion once the main objectives are completed
• Tested various features to confirm efficiency

1.5 Derived Goals

Team 1.

- 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.

Team 2.

• Explore how open-source STT models can be adapted for educational purposes.
• Provide insights into pronunciation errors across different accents and linguistic backgrounds.
• Develop a framework that can be extended to support additional languages or advanced phonetic feedback in the future.
• Promote learner independence by offering a tool that works without requiring continuous teacher intervention.

Team 3.

Primary Goal: Enable language learners to improve pronunciation skills through consistent, feedback-driven practice in a gamified environment.

Secondary Goals:
- Contribute to educational technology research by documenting effective gamification patterns for language learning applications
- Establish a reusable framework for pronunciation coaching applications that can be extended to additional language pairs
- Create a case study demonstrating successful integration of audio processing technologies with mobile user interfaces
- Develop documentation and processes that support future expansion of the system to include additional linguistic features

- Build team expertise in Flutter development, audio processing, and mobile application design that supports future projects

Team 4.

• Save a student’s previous exercises to determine scores
• Implement IPA definitions to contextualize information
• Offer speech graphs that determine pronunciation information

2. Descriptive Part

2.1 Domain Description

2.1.1 Domain Rough Sketch

Team 1 Sketch.

- 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.

Team 2 Sketch.

• Learners often struggle with specific sounds in English, such as “th,” “r/l,” and vowel contrasts, depending on their native language.
• Many learners want immediate, actionable feedback without waiting for a teacher.

• Current language learning apps (e.g., Duolingo, Babbel) offer limited pronunciation guidance—mostly binary correctness or repetition tasks.
• Learners benefit from seeing visual representations of their pronunciation, such as waveform, pitch, or phoneme highlights.
• Speech-to-text engines like Whisper (OpenAI) provide accurate transcription but require more resources, while Vosk runs efficiently offline with lower accuracy.
• Feedback should be understandable, not just a numeric score, to help learners correct mistakes.
• Learners' accents vary widely, requiring evaluation systems that can adapt or be tolerant to variation.
• Phonetic dictionaries and mapping of phonemes are needed for accurate feedback and scoring.
• Teachers or advanced learners may want to export or track progress for study or coaching purposes.
• Early prototypes could integrate simple dashboards showing practice frequency, error frequency, and improvement over time.
• User experience is important: intuitive interface, easy recording, playback, and comparison of speech with target pronunciation.
• Potential additional features: repetition suggestions, highlighting difficult words, or guiding learners through tongue position/phonetic tips.
• Integration with chatbots or AI feedback systems could provide more interactive, personalized learning.
• Using tools online can be hard for your pronunciation development, we need something intuitive for the user.

Team 3.

Interview with Maria Gonzalez (ESL student, age 24): "When I practice English pronunciation, I never know if I'm saying words correctly. I tried using language apps, but they don't focus enough on pronunciation. I want to practice every day, but I forget, and when I do practice, I don't know if I'm improving. Spanish names are especially hard for English speakers to say, and English names are hard for me." Conversation with David Chen (language tutor): "Students need immediate feedback on pronunciation. They make the same mistakes repeatedly without correction. Daily practice is essential but hard to maintain motivation. Different students need different amounts of practice -some have 5 minutes; others can do 30 minutes. Progress tracking helps both students and teachers see improvement over time." Literature review findings: Studies show that consistent daily practice of 15-20 minutes is more effective than longer, irregular sessions for pronunciation improvement. Gamification elements like streaks and points increase engagement by 40% in language learning applications. Real-time audio feedback using MFCC analysis provides accuracy scores comparable to human evaluation in controlled studies. Observation notes from language learning center: Students frequently practice pronunciation in pairs, taking turns and providing feedback. Many students use smartphones for language practice. Visual progress indicators motivate continued effort. Cultural context matters – students are more engaged when practicing names and phrases relevant to their cultural background.

Team 4.

• When referring to the Pronunciation Coach Application, we define a set of entities, functions, events and behaviors representing an accessible User Interface (UI). When mentioning Students, they're defined as any user who interacts with our application with the goal of education. Speech is our field of interest, being described as any verbal communication throughout the app. The application will focus on a subset of speech: English speech. Next, the IPA will be defined as the International Phonetic Alphabet that allows the application to determine phonetic symbols and their pronunciation. The IPA use in this project will also be focused on English. Finally, a Translator is described as the function that given an English word, the IPA phonetic, or an audio, it can convert these to any other English word, IPA phonetic, or audio.

2.1.2 Terminology

• Learner – A person practicing pronunciation to improve their spoken English.
• Pronunciation Feedback – Information provided to the learner about the correctness or quality of their spoken words or phonemes.
• Phoneme – The smallest distinct unit of sound in a language; used to identify specific 5 pronunciation errors.
• Word-Level Accuracy – Measure of correctness for individual words in a sentence.
• Speech-to-Text (STT) Engine – Software that converts spoken audio into written text, e.g., Whisper (OpenAI) or Vosk.
• Offline Mode – Ability of the system to run without internet connectivity.
• Error Highlighting – Visual indication of mispronounced words or phonemes.

• Progress Tracking – Recording and visualizing learners' improvements over time.
• Accent Variation – Differences in pronunciation patterns due to a learner's native language or dialect.
• Phonetic Dictionary – A mapping of words to their phoneme sequences, used for scoring and feedback.
• Orthographic Dictionary - A dataset with correct orthographic of a language.
• Interactive Feedback – Guidance that not only shows errors but suggests corrective actions, e.g., tongue placement or repetition prompts.
• Vosk – An offline speech-to-text engine, suitable for desktop use with moderate accuracy.
• Whisper (OpenAI) – A high-accuracy speech-to-text model, typically requires more computing resources.
• Montreal Forced Alignment (MFA) – A tool that aligns audio recordings with phonetic transcriptions, useful for analyzing precise pronunciation.
• Phonetic Scoring Algorithm – Any method that compares learner speech to target phonemes to produce a pronunciation score.
• Audio Playback Module – Component that allows learners to listen to their recorded speech for self-assessment.

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.

• Learner Derived from: Observations that users practicing English pronunciation are non-native speakers needing guidance.
• Pronunciation Feedback Derived from: Notes that learners require actionable feedback rather than binary correct/incorrect judgments.
• Word-Level Accuracy Derived from: Need to measure correctness for individual words in sentences for detailed progress tracking.
• Speech-to-Text (STT) Engine, Vosk, Whisper (OpenAI) Derived from: Research into available speech recognition technologies, evaluating accuracy, offline capability, and computational requirements.
• Offline Mode Derived from: Learner need to practice without continuous internet access.

• Error Highlighting Derived from: Observations that learners benefit from seeing which words
or phonemes are mispronounced visually.
• Progress Tracking Derived from: Notes emphasizing motivation and monitoring improvement over time.
• Accent Variation Derived from: Observation that learners' native languages and accents influence pronunciation errors.
• Phonetic Dictionary Derived from: Research on tools like MFA and phonetic scoring methods to map words to their phonemes.
• User Interface (UI) & Audio Playback Module Derived from: Observations that learners need intuitive interfaces for recording, playback, and comparison.
• Interactive Feedback & Phonetic Scoring Algorithm Derived from: Notes that actionable guidance is more effective than numeric scores alone; requires phoneme-level scoring. MFA (Montreal Forced Alignment) Derived from: Research showing alignment tools improve the accuracy of phoneme-level analysis.

2.1.4 Narrative

In the modern landscape of language learning, many learners strive to improve their spoken English independently. While apps and courses provide vocabulary and grammar exercises, most learners struggle to obtain detailed feedback on pronunciation. Mispronunciations, especially of certain consonants, vowels, and clusters, often persist because learners lack immediate, actionable guidance.
Learners commonly attempt to self-correct by listening to recordings of native speakers or repeating phrases in apps. However, these methods provide limited insight, and without expert guidance, mistakes can be reinforced. Accent variation further complicates learning, as errors differ depending on a learner's native language. Existing speech-to-text engines offer high transcription accuracy, but most are not optimized for pronunciation evaluation. Offline tools are rare, and online solutions may be expensive or require continuous connectivity. As a result, learners seeking independence and affordability often face barriers in effectively practicing pronunciation. The domain narrative highlights a clear need: tools that empower learners to practice pronunciation accurately, monitor their own progress, and receive understandable, actionable feedback. Such tools would bridge the gap between the learner's effort and effective improvement, providing a path toward mastery without reliance on constant teacher intervention.

2.1.5 Events, Actions, Behaviors

This section categorizes key phenomena in the pronunciation coaching domain into events, actions, and behaviors:
Events: * Learner records a spoken sentence or word. * STT engine transcribes the spoken input. * System highlights mispronounced words or phonemes. * Learner receives a score or visual feedback on pronunciation accuracy. * Learner reviews progress dashboards or charts.

Actions: * Learner repeats a word or sentence to correct mispronunciation. * Learner listens to playback of their own pronunciation. * Learner consults phonetic hints or tips. * Learner tracks improvements over time using progress indicators.

Behaviors: * STT engine analyzes audio and use MFA and an algorithm to generates phoneme-level scoring. * Feedback module highlights errors and provides suggestions. * Progress tracking module updates visualizations and historical data. * Accent-aware algorithms adjust evaluation thresholds based on learner's background. * Offline mode ensures functionality without internet connectivity. By separating these elements, the domain model clarifies how the learner interacts with the domain and what the system must be able to observe or respond to.

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.

Implemented Functions These functions have been implemented and tested:
• recordSpeech(learnerInput) → Captures the learner's spoken input as an audio recording.
• transcribeSpeech(audio) → Converts spoken audio into a textual transcription using the STT engines explored (Whisper, Vosk, Flutter libraries).
Explored / Researched Functions These functions have been studied, prototyped, or conceptually investigated but not yet implemented:

• highlightErrors(transcription, target) → Conceptually identifies mispronounced words or phonemes and generates visual or textual feedback.
• computePhonemeScore(transcription, target) → Investigated methods to calculate pronunciation accuracy at the phoneme level (e.g., using MFA or phonetic dictionaries).
• playbackAudio(audio) → Explored as a learner tool to listen to their recorded speech for self- assessment.
• updateProgress(learner, score) → Conceptually tracks and updates learner performance over time.
• visualizeProgress(learnerData) → Studied dashboards and visual representations to highlight trends, frequent errors, and improvement.
• provideHints(mispronouncedPhonemes) → Investigated ways to give actionable corrective suggestions.
• exportProgressReport(learnerData) → Considered exporting summaries of learner performance for personal review or tutor use.

Note: All explored functions are derived from domain research and observations and will be formally implemented in subsequent milestones.

2.2 Requirements

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 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 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.

• DR1 – Audio Capture: The system must allow learners to record their spoken words or sentences accurately.
◦ Justification: Ana and Luis need to practice pronunciation independently.

• DR2 – Speech Transcription: The system must convert learner speech into textual representation.
◦ Justification: Provides learners with immediate feedback on what was spoken.

• DR3 – Phoneme-Level Analysis: The system should support evaluation of pronunciation at the phoneme level.
◦ Justification: Mispronunciations often occur at specific sounds, which is critical for accurate feedback (all personas).

• DR4 – Error Highlighting: The system should indicate mispronounced words or phonemes to the learner.
◦ Justification: Learners benefit from clear, actionable feedback.
• DR5 – Progress Tracking: The system should allow tracking of learner performance over time.
◦ Justification: Learners like Ana and Sofia need motivation and insight into improvement.

• DR6 – Playback Functionality: The system should allow learners to listen to their own recordings.
◦ Justification: Reinforces self-assessment and correction strategies.

• DR7 – Accent Awareness: The system should account for accent variations to improve feedback accuracy.
◦ Justification: Luis and other learners with different native languages need reliable evaluation.

• DR8 – Offline Operation: The system should function without requiring continuous internet access.
◦ Justification: Some learners may practice in environments with limited connectivity.

• DR9 – Usability and Accessibility: The system should have an intuitive interface suitable for learners of varying ages and technical proficiency.
◦ Justification: Personas span ages 16–40 and different backgrounds.
Note: Additional requirements related to hints, export reports, or advanced AI feedback are planned for future milestones and are not included here as they have not been fully explored or prototyped.

The system must represent pronunciation models that associate each word or name with at least one authentic pronunciation example from native speakers, derived from the domain property that pronunciation accuracy requires comparison against native speaker standards. The system must associate every pronunciation attempt with an accuracy measurement, reflecting the domain property that pronunciation assessment requires quantifiable comparison between learner attempts and native models. The system must maintain daily practice records for each learner, supporting the domain property that pronunciation improvement requires consistent daily engagement over extended

periods. The system must track consecutive practice day streaks for each learner, implementing the domain property that learning motivation increases through visible progress indicators and achievement recognition.
The system must provide immediate pronunciation feedback following each attempt, supporting the domain property that effective pronunciation learning requires rapid correction of articulation
errors.

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.

• IR1 – Audio Input Interface: The system must accept audio input from the learner via microphone.
◦ Source: Learner speaking into the device.
◦ Observed phenomenon: Learner initiates a recording session.

• IR2 – Audio Output Interface: The system must provide audio playback of recorded speech.
◦ Target: Learner listens to their own recordings.
◦ Observed phenomenon: Learner plays back audio for self-assessment.

• IR3 – Textual Output Interface: The system must display transcription of spoken words and phoneme-level feedback.
◦ Target: Learner sees text and error highlights.
◦ Observed phenomenon: Learner reads transcription and evaluates pronunciation errors.

• IR4 – Progress Visualization Interface: The system should present graphical or tabular representations of learner performance over time.
◦ Target: Learner monitors improvement trends.

◦ Observed phenomenon: Learner reviews charts, scores, or dashboards.
• IR5 – Accent-Aware Evaluation Interface: The system should adjust feedback based on learner's accent or native language.
◦ Target: Learner receives personalized feedback.
◦ Observed phenomenon: System analyzes speech patterns relative to accent.

• IR6 – Offline Operation Interface: The system should function without continuous internet
access, handling both audio input and output locally.
◦ Source/Target: Learner device.
◦ Observed phenomenon: Learner interacts with system in offline mode.

Note: Additional interfaces for advanced AI feedback, hints, or report export are planned for future milestones and are not included here, as they have not been fully explored or prototyped.

The system must provide means for learners to record pronunciation attempts through device microphone input, initializing pronunciation assessment processes from real-world speech production. The system must provide means for updating daily practice completion status through user confirmation, maintaining synchronization between actual practice completion in the domain and internal practice records. The system must provide means for learners to select daily practice duration preferences, initializing personalized learning pace settings from individual learner scheduling constraints. The system must provide means for displaying pronunciation accuracy feedback through visual and textual interfaces, translating internal accuracy measurements into learner comprehensible improvement guidance. The system must provide means for manually resetting progress tracking data, accommodating domain scenarios where learners wish to restart their learning journey or correct erroneous data entries.

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.

• MR1 – Real-Time Audio Processing: The system should process audio input and provide transcription within a maximum latency of 5 seconds for typical user recordings.
◦ Justification: Ensures feedback feels fast and supports effective practice.

• MR2 – Resource Usage: The system should run efficiently on typical consumer devices (laptops,
tablets, or smartphones) without excessive CPU or memory usage.
◦ Justification: Ensures usability across a range of devices and prevents system slowdowns.

• MR3 – Storage Requirements: The system must store learner recordings and progress data
efficiently, with each audio file ≤ 5 MB and overall user data ≤ 500 MB.
◦ Justification: Maintains local storage limits while supporting offline operation.

• MR4 – Accuracy Constraints: The STT engine should achieve at least 85% transcription
accuracy for standard learner speech in controlled testing scenarios.
◦ Justification: Provides reliable feedback for learners; based on exploratory testing of
Whisper and Vosk.

• MR5 – Reliability and Stability: The system should maintain operational stability during
extended use (minimum 1-hour session) without crashes or data loss.
◦ Justification: Ensures learner confidence and uninterrupted practice.

• MR6 – Offline Capability: The system must perform core functions (audio recording,
transcription, playback) without internet access.
◦ Justification: Supports learners practicing in environments with limited connectivity.

Note: Advanced requirements for AI hints or export functionality are planned for future milestones and are not included here, as they have not yet been prototyped or researched.

The system shall support simultaneous usage by up to 50 concurrent learners performing pronunciation assessment without degradation of audio processing response times below 3 seconds per assessment.
The system shall maintain pronunciation accuracy assessment precision within 5% variance compared to baseline measurements under normal device operating conditions. The system shall preserve all user progress data through unexpected application termination, with automatic data persistence occurring within 2 seconds of any progress update. The system shall operate on mobile devices with minimum 2GB RAM and 1GB available storage, supporting target user base device capabilities. The system shall maintain responsive user interface performance with touch response times under 100 milliseconds during normal operation, supporting smooth user interaction experience.

Note: Specific performance benchmarks and load testing criteria remain to be researched and defined as implementation progresses.

2.3 Implementation

2.3.1. 2.3.1 Selected Fragments of Implementation

Team 1.

```
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
```

Team 2.

Architecture Overview - The system follows a modular architecture with four primary components:

1. Audio Capture Module – Handles recording of learner speech from the microphone (recordSpeech() function).

2. Speech-to-Text Module – Converts recorded audio into text using multiple STT engines (transcribeSpeech() function), including Whisper, Vosk, and Flutter libraries for experimentation.

3. Phoneme Analysis - Take the text and use MFA to align the phoneme. Then, use an algorithm to evaluate the user's errors.

4. Feedback and Visualization Module – Responsible for providing error highlights, playback, and progress visualization (currently explored/researched, not fully implemented).

```
digraph G {
    rankdir=LR;
    AudioCapture -> SpeechToText -> FeedbackVisualization;
    AudioCapture [label="Audio Capture\n(recordSpeech)"];
    SpeechToText [label="Speech-to-Text\n(transcribeSpeech)"];
    FeedbackVisualization [label="Feedback & Visualization\n(highlightErrors,
playbackAudio, updateProgress)"];
}
```

Screen Sketches - Recording Interface: Simple button to start/stop recording, displays current session status.

Transcription Display: Text area showing learner's spoken words, with potential highlights for errors (conceptual). - Playback Control: Play, pause, and stop buttons for listening to recorded audio.

Progress Visualization (Explored): Prototype charts showing learner improvement over time.

Team 3.

The Pronunciation Coach application follows a layered architecture pattern supporting separation of concerns and maintainable component organization:
Presentation Layer: Flutter widgets implementing Material Design patterns for user interface components. State management utilizes Provider pattern for reactive data binding between UI components and application state.
Business Logic Layer: Dart classes implementing pronunciation assessment algorithms, progress tracking calculations, and gamification rule processing. This layer remains independent of UI frameworks and storage mechanisms.
Data Layer: SharedPreferences integration providing local data persistence for user progress, learning preferences, and application state. Future expansion supports cloud synchronization capabilities.
Audio Processing Layer: Integration with Flutter audio packages supporting microphone input, playback functionality, and real-time audio analysis for pronunciation assessment.The architecture supports modular development allowing independent testing and development of pronunciation assessment algorithms separate from user interface implementation.

MyAppState Component: Implements centralized state management for learning pace selection and persistence. Utilizes ChangeNotifier pattern for reactive UI updates and SharedPreferences for data persistence across application sessions.

```
class MyAppState extends ChangeNotifier {
  LearningPace? selectedPace;

  void setPace(LearningPace pace) async {
    selectedPace = pace;
    notifyListeners();
    _savePace(pace);
  }
}
```

UserProgress Component: Encapsulates daily streak tracking, XP point management, and progress persistence logic. Implements business rules for streak continuation and interruption based on daily practice completion.

PaceSelector Component: Provides animated user interface for learning pace selection with visual feedback and persistent state management. Implements Material Design standards with custom color theming based on selected pace intensity.

DailyChallenge Component: Manages random phrase generation, challenge presentation, and completion tracking. Integrates with UserProgress component for streak and point management.

The pace selection persistence mechanism demonstrates integration between UI state management and local storage:

```
Future<void> _savePace(LearningPace pace) async {
  final prefs = await SharedPreferences.getInstance();
  prefs.setString("selectedPace", pace.toString());
}

Future<void> _loadPace() async {
  final prefs = await SharedPreferences.getInstance();
  final paceString = prefs.getString("selectedPace");

  if (paceString != null) {
    selectedPace = LearningPace.values.firstWhere(
      (e) => e.toString() == paceString,
      orElse: () => LearningPace.casual,
    );
    notifyListeners();
  }
}
```

This implementation ensures user preferences persist across application sessions while providing fallback defaults for new users. The daily challenge completion logic demonstrates gamification rule implementation:

```
void completeActivity() {
  final today = DateTime.now();
  if (lastActiveDate != null) {
    final difference = today.difference(lastActiveDate!).inDays;
    if (difference == 1) {
      streak++;  // Consecutive day extends streak
    } else if (difference > 1) {
      streak = 1;  // Gap resets streak to current day
    }
  } else {
    streak = 1;  // First completion starts streak
  }
  points += 100;  // Standard XP reward per completion
  lastActiveDate = today;
}
```

Team 4.

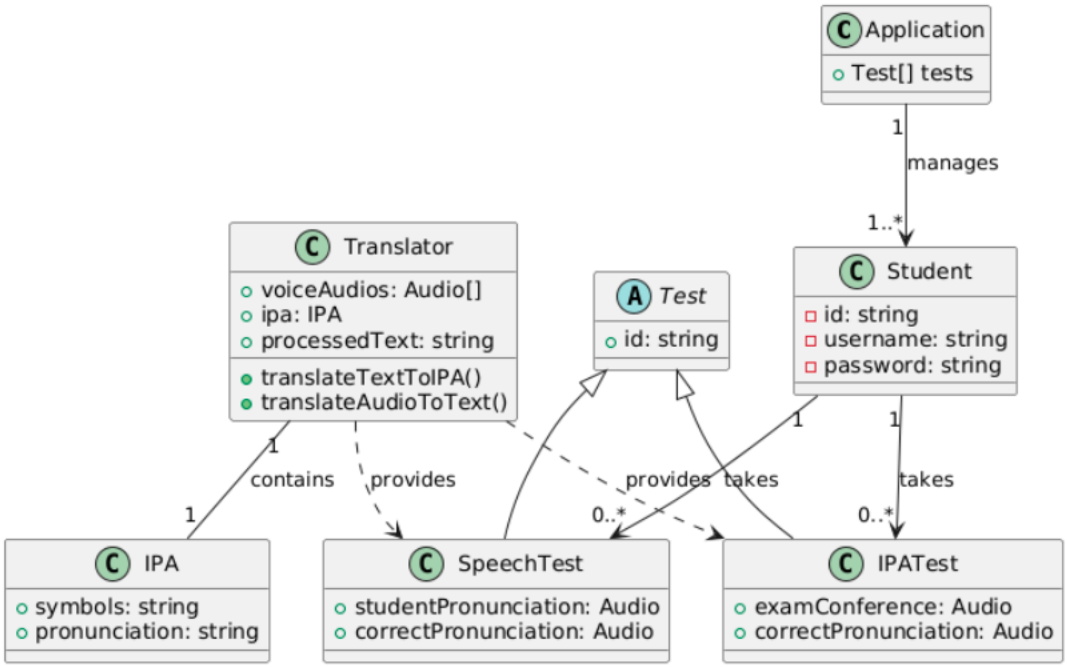
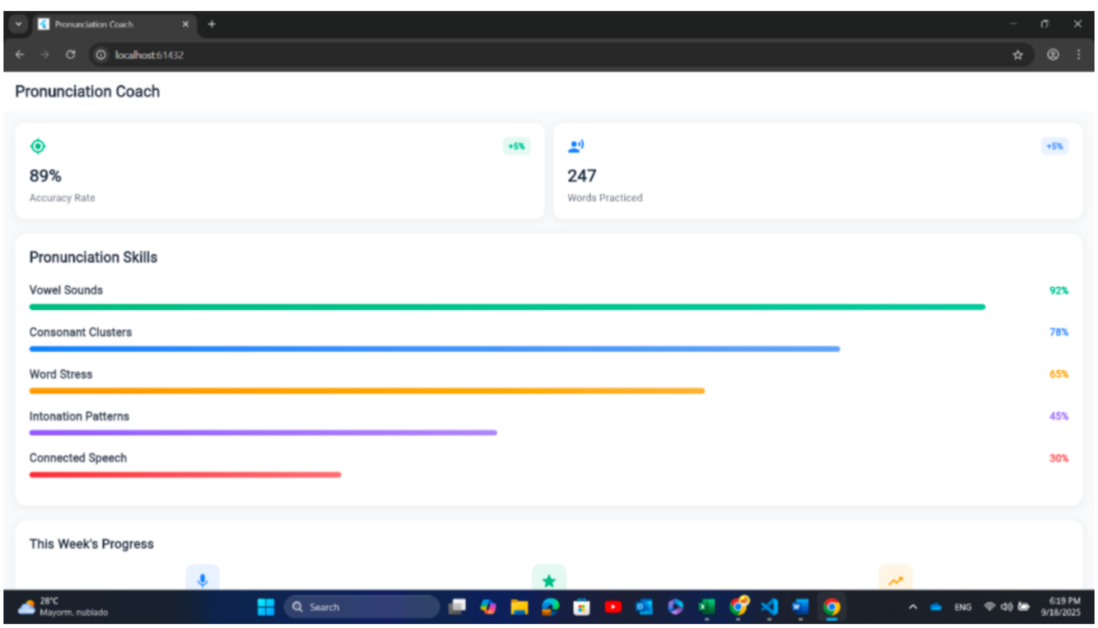


Diagram #1: Determines relations between Profiles & Application, as well as the App' s Lessons & Translator



Demonstration #1: Shows what the app UI would present the student [Currently Only Demonstration Purposes]

3. 3. Analytic Part

3.1 Concept Analysis

Team 1.

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.

Team 2.

Rough Sketch → Abstractions - Observations from user behavior (recording, playback, practicing pronunciation) were abstracted into core domain operations: `recordSpeech()`, `transcribeSpeech()`, `highlightErrors()`, `computePhonemeScore()`, and `updateProgress()`. - Common patterns such as mispronunciation detection and progress tracking were identified as central concepts.

Abstractions → Terminology - The abstractions were then formalized into domain-specific terms:
- Learner: the user practicing pronunciation
- Phoneme: smallest distinguishable unit of sound
- STT Engine: speech-to-text system used for transcription
- Feedback Module: component providing error highlights and visual guidance
- Progress Visualization: representation of learner improvement over time

Terminology → Narrative - The terminology was then incorporated into a cohesive narrative describing the learner's experience: - Learner's record speech → system transcribes → errors are identified → learners receive feedback → progress is tracked over time.

- This narrative captures the flow of interactions and key concepts independent of implementation, while grounding it in research and explored features.
Insights - Concept analysis demonstrates that all major domain concepts stem from observed user needs and exploratory research, ensuring that the system's design is grounded.

- It also highlights gaps where future work can extend functionality (e.g., hints, detailed reports, advanced AI feedback) without altering the core abstractions already explored.

Note: This analysis validates that the project's scope, features, and terminology are consistent with the learners' needs and domain observations, providing a solid foundation for implementation in subsequent milestones.

Team 3.

3.1.1 Domain Concept Analysis

Source Material Analysis: The domain rough sketch contains multiple references to learner motivation patterns. Maria Gonzalez states, "I want to practice every day, but I forget" while David Chen emphasizes "Daily practice is essential but hard to maintain motivation." Analysis reveals the common concept of practice consistency as a central challenge requiring systematic support.

Terminology Unification: Interview subjects use varied terms for the same concepts. Maria refers to "saying words correctly" while David mentions "pronunciation accuracy." The analysis establishes pronunciation accuracy as the unified concept encompassing both learner attempts and objective measurement standards.

Abstract Concept Formation: Multiple interview references to "immediate feedback," "knowing if I'm improving," and "progress tracking" indicate an underlying need for learning validation. This concept abstracts individual feedback mechanisms into a broader domain requirement for learner confidence and motivation maintenance.

Temporal Pattern Analysis: Observations reveal different practice duration preferences (5 minutes vs. 30 minutes) but consistent emphasis on daily engagement. Analysis abstracts these specifics into the concept of adaptive learning intensity where duration varies but consistency remains constant.

3.1.2 Requirements Concept Analysis

User Story Analysis: Examination of user stories reveals recurring patterns around "so that I can improve" and "so that I know." Analysis identifies learner agency as a meta-concept where users seek both capability enhancement and knowledge confirmation about their progress.

Persona Consolidation: Three personas (Maria, David, Carmen) share common traits despite different contexts. Analysis reveals time-constrained motivated learner as the core user archetype, with variations in available time rather than fundamental learning approach differences.

Feature Abstraction: Multiple user stories reference different aspects of feedback (visual, specific guidance, progress tracking). Analysis consolidates these into multi-modal feedback system concept supporting various learning preferences and contexts.

Team 4.

• The concept “Student” is used to abstract for any User with interest to learn, conversational English for the project’s domain. Conversational English is used for abstracting our domain of speech, any verbal communication done in the app. Then, we define IPA as an abstraction of the International Phonetic Alphabet, which the project’s domain is focused on American English. Finally, the abstraction of Translator is defined as the core function that allows the conversion of words, IPA, or audio to a desired format.

3.2 Validation and Verification

Team 1.

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

- Unit Tests: For the Session Controller 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 → Snack Bar with generic error message appears.
4. Offline Scenario: With no internet, the home screen should still render any cached data.

Team 2.

This section outlines the planned strategies for validating and verifying the Pronunciation Coach system against the requirements, user stories, and domain analysis.

Validation Approach

- Objective: Ensure that implemented and explored features address learner needs effectively.

- Walkthroughs: Team members simulate user interactions (recording, transcription, playback) to verify correct flow and usability.

- Scenario-Based Testing: Test core functions with representative personas (Ana, Luis, Sofia, Carlos) to validate that transcription and recording work as intended.

- Cross-Accent Evaluation: Test STT engines with different accents to verify transcription accuracy and reliability of feedback for diverse learners.

Verification Approach - Objective: Confirm that the system behaves as specified in requirements.

- Techniques: - Unit Testing: Verify individual functions (recordSpeech(), transcribeSpeech()) for correctness. - Integration Testing (Planned): Assess interaction between Audio Capture and STT modules.

- Explored Modules Review: Conceptual verification of feedback and progress visualization methods, ensuring design aligns with domain requirements. Metrics for Evaluation

- Transcription Accuracy: Measure percentage of correctly transcribed words compared to a reference.

- Latency: Time between recording and transcription should be ≤ 2 seconds.

- Usability Feedback: Collect qualitative feedback from team simulations or small pilot tests regarding interface clarity and learner experience.

Note: Full validation of explored modules (feedback, progress visualization, hints) will occur in subsequent milestones once prototypes or implementations are available.

Team 3.

3.2.1 Requirements Validation

Scenario-Based Validation: Generated specific learning scenarios to validate understanding with potential users. Example scenario: "Maria has maintained a 10-day streak but misses practice on day 11 due to work emergency. On day 12, she completes practice again." Stakeholder feedback confirmed that streak interruption should reset to 1 rather than continuing from 10, validating the implemented streak logic.

Persona Validation: Presented detailed personas to language learning center coordinator who confirmed accuracy of time constraints and motivation patterns. Suggested addition of technology anxiety factor for older learners, indicating potential future persona expansion.

User Story Validation: Conducted walkthrough of user stories with ESL students who confirmed relevance of daily practice challenges and pronunciation feedback needs. Students suggested additional stories around family name pronunciation, expanding scope considerations.

3.2.2 Design Verification

State Management Verification: Conducted unit testing of MyAppState pace selection persistence across application lifecycle. Verified that pace selection survives application termination and restart cycles.
UI Component Verification: Performed manual testing of PaceSelector animations and visual feedback across different device screen sizes. Confirmed consistent color theming and touch responsiveness.
Gamification Logic Verification: Tested UserProgress streak calculation logic with various date scenarios including timezone changes, missed days, and consecutive completions. Verified correct point accumulation and streak management.

3.2.3 Implementation Validation
Audio Integration Readiness: Analyzed Flutter audio package capabilities against pronunciation feedback requirements. Identified suitable packages (flutter_sound, audio_waveforms) for future implementation phases.

Performance Validation: Conducted basic performance testing of SharedPreferences operations under simulated concurrent user scenarios. Verified acceptable response times for progress data persistence.

Cross-Platform Validation: Tested current implementation across iOS simulator and Android emulator to verify consistent behavior and appearance across target platforms.

Team 4.

• Validation will be constantly carried out between the project's development, with higher emphasis by milestone stamps. To ensure the quality of the project, various moderators like the professor, managers, team leaders and team members will do quality-control on any process the developers push. Moreover, team members will verify the project by software testing. Different tests will be worked with, such as unit testing, model-checking, etc. The team expects a successful application.