# LinguaEase - Language Learning Mobile App

# 1. INTRODUCTION

## 1.1 Background

The increasing need for language skills in education, business, and travel has fueled the demand for accessible and effective learning solutions. Mobile apps, being ubiquitous, provide an ideal platform for language learning. LinguaEase combines AI-powered personalization, gamification, and an offline mode to offer an engaging and effective learning experience.

## 1.2 Objectives

- To create a mobile app for learning Japanese, German, and French, catering to beginners and intermediate learners.  
- To provide AI-driven features like speech recognition and personalized learning paths.  
- To gamify the language-learning process with progress tracking and rewards.  
- To enable offline access for selected modules to ensure learning without internet dependency.

## 1.3 Purpose, Scope, and Applicability

### 1.3.1 Purpose

LinguaEase is designed to simplify and enhance the language-learning process for users through interactive modules, speech analysis, and real-time feedback.

### 1.3.2 Scope

- Multilingual support for Japanese, German, and French.  
- AI-driven speech recognition and grammar improvement tools.  
- Interactive quizzes, vocabulary builders, and grammar lessons.  
- Offline access for core learning features.

### 1.3.3 Applicability

- Students seeking to expand language skills.  
- Professionals and travelers needing conversational proficiency.  
- Language enthusiasts looking for an engaging learning platform.

## 1.5 Organization of Report

This report includes:  
1. Introduction to the app and its features.  
2. Overview of technologies used in development.  
3. Requirement analysis, design, and implementation strategies.  
4. Testing, results, and conclusions.

# 2. SURVEY OF TECHNOLOGIES

- Android Studio: Development environment for mobile app creation.  
- Firebase: Backend for user authentication and data management.  
- TensorFlow Lite: Integration of AI for speech recognition and analysis.  
- Kotlin/Java: Programming languages for developing the app.  
- SQLite: Database for offline data storage.  
- Figma: Design tool for prototyping UI/UX.

# 3. REQUIREMENT AND ANALYSIS

## 3.1 Problem Definition

Language learners face the following challenges:  
- Lack of personalized content.  
- Difficulty in practicing pronunciation.  
- Limited options for offline learning.  
- Tedious and monotonous learning experiences.  
  
LinguaEase tackles these issues by incorporating AI, gamification, and offline capabilities into a mobile platform.

## 3.2 Requirement Specification

### Functional Requirements:

- User authentication and profile management.  
- AI-based speech recognition for pronunciation analysis.  
- Vocabulary and grammar modules with progress tracking.  
- Gamified quizzes and rewards system.  
- Notifications and reminders for learning consistency.

### Non-Functional Requirements:

- High responsiveness and ease of use.  
- Secure data storage and privacy compliance.  
- Optimized for low resource consumption.  
- Support for Android devices running version 6.0 and above.

## 3.3 Planning and Scheduling

| Phase | Timeline |  
|---------------------|-------------------|  
| Requirement Gathering | Week 1 - Week 2 |  
| Design Prototyping | Week 3 - Week 4 |  
| Development | Week 5 - Week 10 |  
| Testing and Debugging | Week 11 - Week 12 |  
| Deployment | Week 13 |

## 3.4 Software and Hardware Requirements

### Software Requirements:

- Android Studio (Latest Version).  
- Firebase SDK for backend integration.  
- TensorFlow Lite for AI features.  
- SQLite for local database storage.

### Hardware Requirements:

- Development machine with 8GB RAM, 256GB SSD.  
- Android device with version 6.0 or higher for testing.

## 3.5 Preliminary Product Description

LinguaEase features an intuitive interface, interactive learning modules, and AI-powered feedback for language learners. The app includes a user-friendly dashboard, progress tracking, and gamified elements like daily streaks and achievement badges.

## 3.6 Concept Models

### User Interaction Flow:

1. Onboarding: User registers/logs in → Chooses language and level → Sets daily goals.  
2. Learning Modules:  
- Vocabulary builder.  
- Grammar lessons with exercises.  
- Pronunciation practice using AI.  
- Interactive quizzes.  
3. Rewards and Progress Tracking:  
- Displays progress charts and awards badges for milestones.

### System Architecture:

1. Frontend: Designed in Android Studio using Kotlin/Java.  
2. Backend: Firebase for real-time data management and analytics.  
3. AI Module: TensorFlow Lite for speech and language processing.  
4. Offline Mode: SQLite for local storage of vocabulary and grammar content.