Introduction to Software Engineering

Development Plan

Software Engineering Department
Faculty of Information and Technology
University of Science

1 Development Plan

1.1 Requirements Analysis

This phase aims to identify the needs of English learners and transform them into well-defined system requirements. The team will analyze the challenges learners face in speaking practice—such as lack of feedback and limited opportunities to converse in English—and determine how technology can address them.
 The system targets students, professionals, and ESL learners who want to improve their speaking fluency, pronunciation, and confidence through AI-based interactions.

Key Objectives:

- Understand user needs and learning behaviors.
- Define system goals and scope.
- Specify functional and non-functional requirements.

Functional Requirements:

- User registration and authentication.
- Voice recording and recognition.
- Pronunciation scoring and feedback.
- Learning progress tracking and analytics.

Non-Functional Requirements:

- Speech recognition accuracy above 85%.
- Responsive design for mobile and desktop.
- Support at least 1,000 concurrent users.

Deliverable: System Requirement Specification (SRS) document.

1.2 Software Design

In this phase, the team will create the architectural blueprint of the system, ensuring that all components work together smoothly. The system will adopt a **client–server architecture**, where the frontend handles user interaction and the backend manages data processing and speech analysis.

System Overview:
The frontend will be built using **React.js**, providing an interactive interface for lessons and voice practice. The backend will use **Node.js** and **Express.js** to process user data, communicate with APIs, and store information in a **MySQL** database. The **Google Speech-to-Text API** will be integrated to handle voice recognition and pronunciation scoring.

Main Modules:

1. **User Module:** Handles login, registration, and profile management.
2. **Lesson Module:** Provides topic-based speaking exercises and simulations.
3. **Speech Analysis Module:** Evaluates pronunciation and fluency in real time.
4. **Progress Module:** Displays statistics and personalized learning paths.

Deliverable: Software Design Document (SDD) including system architecture diagram and database schema.

1.3 Implementation

The implementation stage involves translating the design into working code and integrating all system components. Development will follow an **iterative approach**, allowing feedback and improvement at each stage.

Implementation Phases:

- **Phase 1:** Environment setup and database creation.
- **Phase 2:** Backend development (authentication, API integration, scoring logic).
- **Phase 3:** Frontend development (UI screens, voice input, and feedback interface).
- **Phase 4:** Integration of all modules and preliminary testing.

Technologies Used:
React.js – Node.js – Express – MySQL – Google Speech API – Firebase

Deliverable: Working system prototype with core functionalities implemented.

1.4 Testing

The purpose of this phase is to verify that the system works correctly, efficiently, and meets user expectations. Testing will be performed continuously during and after development to ensure system stability.

Testing Types:

- **Unit Testing:** To check individual modules and components.
- **Integration Testing:** To verify communication between frontend, backend, and database.
- **User Acceptance Testing (UAT):** Conducted with real learners to assess usability and accuracy.
- **Performance Testing:** To ensure system responsiveness under load.

Testing Tools: Jest, Postman, and Lighthouse.
Success Criteria:

- No major bugs or crashes.
- Speech recognition accuracy ≥ 85%.
- Average response time < 2 seconds.

Deliverable: Testing Report and Debugged System.

1.5 Deployment and Maintainance

The final phase focuses on deploying the system to a live environment and maintaining its long-term operation. The application will be hosted on cloud infrastructure for scalability and accessibility.

Deployment Plan:

- Frontend hosted on **Firebase Hosting**.
- Backend deployed on **Google Cloud Platform**.
- Continuous Integration/Continuous Deployment (CI/CD) via GitHub Actions.

Maintenance Strategy:

- Regular updates and bug fixes.
- Monitoring system logs and user feedback.
- Adding new lessons, AI improvements, and performance optimization.

Future Enhancements:

- Real-time AI conversations.
- Support for multiple English accents.
- Mobile app version for Android/iOS.

Deliverable: Fully deployed and maintained system ready for user access.