Ollscoil Teicneolaíochta an Oirdheiscirt

South East Technological University

LingoLizard Design Document

Name: Michael Cullen

Student ID: C00261635

Supervisor: Dr Jamal Tauseef

Date: 4/12/2024

Contents

Introduction

LingoLizard seeks to fulfil the growing need for effective language education by employing innovative real-time conversational learning techniques. The application focuses on fixing common issues with spelling and grammar and providing feedback that enhances the learning experience by integrating interactive practice, LingoLizard ensures that users not only learn the rules of language but also apply them in real-world scenarios.

LingoLizard aims to create an engaging and effective language learning experience that empowers users to improve their communication skills confidently and competently.

Document Purpose

This document will outline LingoLizard's internal structure. It will describe the internal components and website and how they will function as a final product.

I will include a list of technologies I am using and how they are planned to be used, along with details of the alpha released for the first iteration.

System Overview

LingoLizard operates through a structured conversational model, guiding users through role-playing exercises in real-world scenarios. The bot tracks user progress and adjusts its responses based on proficiency levels. Key components include:

- **Bot Framework**: Handles dialogues and interactions.
- **User State Management**: Saves user preferences and progress.
- **Scenario-Based Learning**: Engages users in practical language exercises.
- **Translation & NLP Processing**: Provides corrections and feedback.

Core Components

- **Conversational Flow**: The system follows a waterfall model where users select a language, set their proficiency, and engage in guided conversations.
- **State Management**: Saves learning progress and adjusts feedback accordingly.
- **Scenario-Based Learning**: Includes Taxi, Hotel, and Job Interview scenarios, allowing users to practice relevant conversations.
- **Translation & Feedback**: Uses Azure Translator and sentiment analysis to refine user input and provide learning insights.

Technologies

Python

A versatile, high-level programming language used for LingoLizard's backend development, API handling, and data processing. Python enables seamless integration with Azure services and supports the chatbot's conversational logic and NLP features.

Git

A distributed version control system that tracks changes in the code during development. Git is used to manage the codebase, collaborate with team members and ensure the project integrity across iterations.

Postman

A popular API testing tool that makes sending requests and debugging API endpoints easy. Postman is used during LingoLizard development to test the APIs used.

Azure Services

LingoLizard integrates various Azure services for scalability and AI enhancements:

- **Azure Bot Services**: Handles bot interactions and deployment.
- **Azure Translator API**: Provides real-time language translation and grammar correction.
- **Azure Cognitive Services**: Supports NLP features like sentiment analysis and intent recognition.
- **Azure App Services**: Manages cloud-based hosting and deployment.

Alpha release

Language Selection

The user can choose the language they wish to practice, which determines the language in which all interactions with the bot will occur. The selected language dynamically influences the bot's responses, intents, and scenarios.

Proficiency Selection

This option allows the user to specify their level of proficiency in the selected language from beginner, intermediate and advanced. This selection adjusts the difficulty of the scenarios, tailoring the prompts and challenges to suit the user’s skill level for an effective learning experience.

Scenario Selection

Allows the user to choose a real-life situation they want to practice. Scenarios include Taxi, Restaurant, Hotel Check-In, and Asking for Directions. The selected scenario provides a focused context for language practice, making the learning experience immersive and practical.

Django Interface

Displays a real-time chat window where users can interact with the bot. The chat interface shows the user’s messages and the bot’s responses.

Beta Release

Scenarios Selection

Now, scenarios are chosen based on difficulty. As of the beta release date, beginner is the only working option, which will start the taxi scenario. Intermediate will start the hotel check-in scenario in the future, and advanced will be a job interview.

Input Feedback

Initial grammar corrections and feedback have been implemented. However, improvements are ongoing, and feedback accuracy may vary depending on user input complexity.

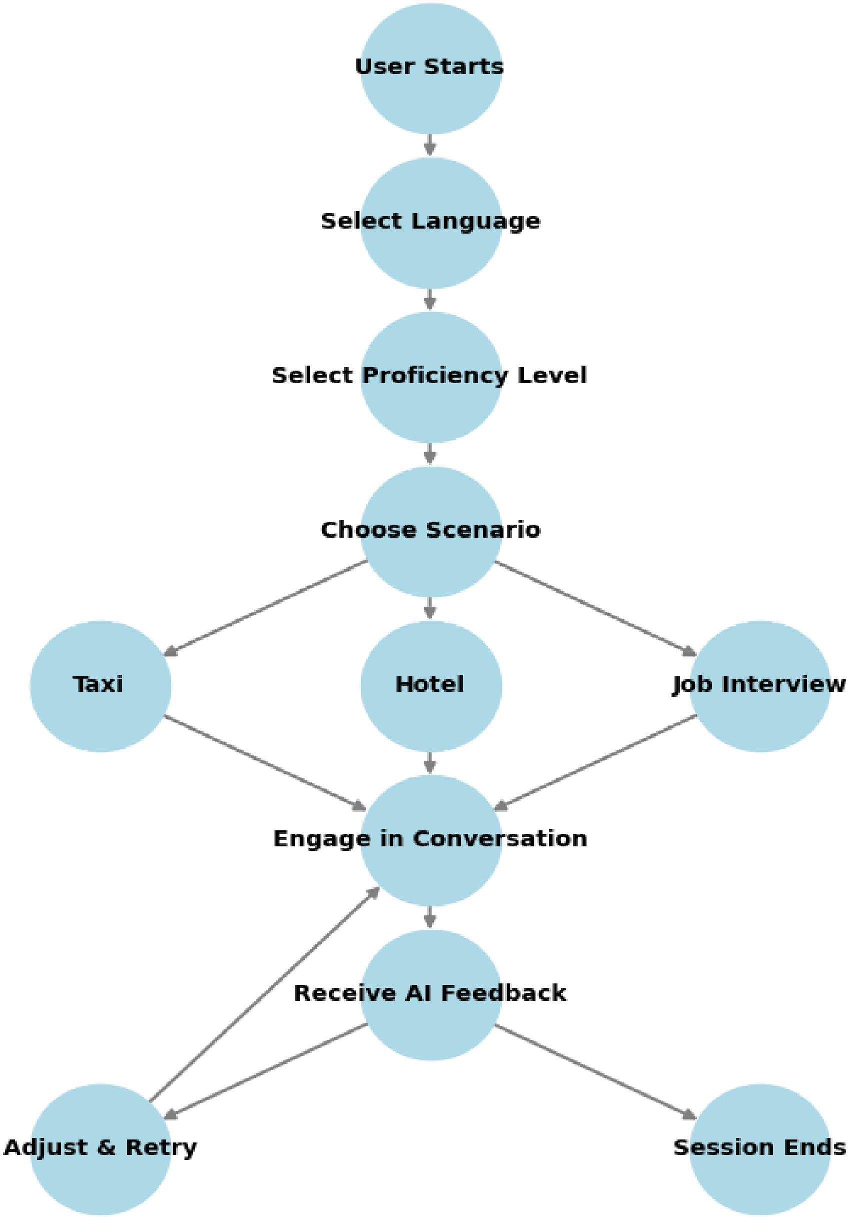
UI Improvements

The interface has been updated for better usability, but further enhancements are planned to improve navigation and user interaction. There are now also buttons for the user to select language and proficiency.

Hosting

The Beta is being hosted on Azure and can now be accessed easily. I have also adopted some DevOps principles and placed tests the code must pass each deployment to prevent the site from breaking.

Flow Chart


User Starts
Select Language
Select Proficiency Level
Choose Scenario
Taxi
Hotel
Job Interview
Engage in Conversation
Receive AI Feedback
Adjust & Retry
Session Ends