



LingoLizard

FUNCTIONAL SPECIFICATION

Name: Michael Cullen

Student ID: C00261635

Supervisor: Dr Jamal Tauseef

Date: 10/10/2024

CONTENTS

Introduction.....	3
Project Scope.....	3
Project Objectives	3
Target Audience	3
Problem and Proposition.....	4
Problem Statement.....	4
Value Proposition	4
System Overview.....	5
System Context Diagram	5
Assumptions.....	5
Dependencies.....	5
Requirements	6
Functionality	6
Core features.....	6
additional features	6
Usability	6
Reliability.....	7
Performance	7
Security	7
Supportability.....	7
Use case Diagram	8
Detailed use case.....	9
Brief Use Cases	10
References	11

INTRODUCTION

This specification provides all functionalities and requirements for this application. It offers detailed descriptions and illustrations to aid in directing the creation and validation of LingoLizard.

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.

Project Scope

PROJECT OBJECTIVES

The objective of this project is to conduct each of the requirements and functions mentioned in this document. To create a good user experience and encourage users to want to use this website and build a habit of practising their language of choice.

TARGET AUDIENCE

This project's target audience is users trying to learn a language but do not know anyone who can speak that language they can practice.

Other language applications might focus more on learning languages by learning basic words without context, it can be hard to know where to use what you learn in certain situations.

This project will be useful for people who are not interested in being fluent and want to learn simple travelling phrases.

PROBLEM AND PROPOSITION

Problem Statement

Acquiring proficiency in a foreign language is exceedingly challenging, and regrettably, there is no silver bullet for this problem. Being fluent in a Language involves an understating of grammar, vocabulary and pronunciation. Language learning can be accomplished in a variety of ways, including immersion and school-based approaches. For people wanting to learn through immersion, they need to be surrounded by the language therefore without travelling, they will need to listen to movies, songs or podcasts in their target language. This is helpful but they are unable to practice and develop conversational fluency effectively. Research says, “As your brain learns how to process a second language, it can even change and improve how it processes your first language.”[1] This shows it is beneficial to be bilingual.

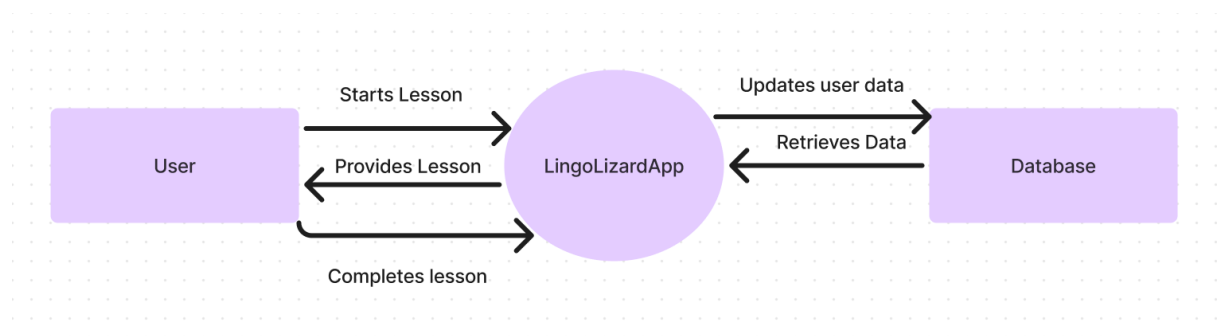
Value Proposition

LingoLizard addresses this challenge by offering a solution that combines personalised lessons, real-time conversations and gamification elements to keep the user engaged. LingoLizard aims to help any user improve regardless of their skill level. Unlike other language apps that focus on grammar drills and vocabulary memorisation, LingoLizard aims to provide a unique learning experience for each user no matter what level of fluency they are. Whether a user is advanced, or a beginner LingoLizard’s versatile approach will help anyone looking to improve their language skills.

SYSTEM OVERVIEW

This section provides a high-level view of the LingoLizard system. It includes a diagram showing how the system will work, along with assumptions and dependencies.

System Context Diagram



Assumptions

- Users are comfortable with text-based communication.
- Users have reliable internet access.

Dependencies

- **Web browsers:** Chrome, Firefox, Safari, and Edge for browser-based access.
- **Database management:** Hosted on PostgreSQL to store user data and progress.
- **Rasa Framework:** Used for building the conversational AI engine, providing intent recognition and real-time feedback.

REQUIREMENTS

This section covers the website's features that are necessary for it to operate as expected and deliver a good user experience.

Functionality

CORE FEATURES

- **Provide real-time conversation lessons:**
Offer lessons across multiple languages with options for different skill levels.
- **Real-time feedback and corrections:**
Use AI to detect grammar, pronunciation, and vocabulary errors and offer instant suggestions for improvement.

ADDITIONAL FEATURES

- **Track user progress:**
Maintain a user profile that visualises achievements, completed lessons, and areas for improvement.
- **Include gamification:**
Introduce streaks, leaderboards, and points.
- **Provide role-play scenarios:**
Provide everyday scenarios for practice.

Usability

- **User-friendly interface**
Keep navigation simple for users to access lessons.
- **Multi-language support:**
Ensure the website can be used by a global audience.
- **Supports mobile and desktop use:**
Make the platform available across all devices.
- **Provide reminders and notifications:**
Use notifications and reminders to remind or alert users.

Reliability

- **Ensure the bot is available 99% of the time:**
Minimise downtime to maintain a seamless user experience.
- **Back up data and log errors:**
Implement regular backups and logging of errors.

Performance

- **Keep responses fast:**
Chatbot responds in seconds to ensure a smooth user experience.

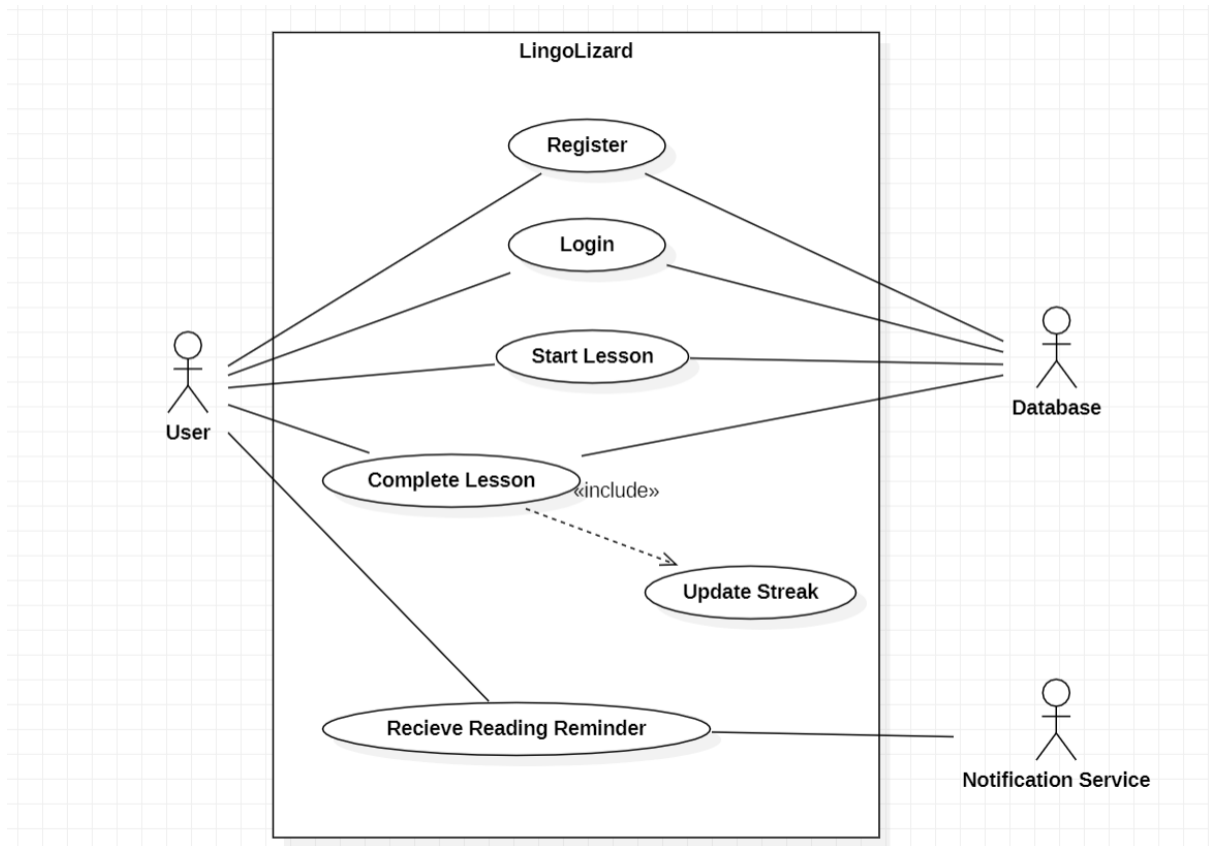
Security

- **Ensure user data is private and protected:**
Comply with GDPR to keep users' data safe.

Supportability

- **Available on multiple devices:**
Users can learn on their mobile devices or desktops.
- **Available on multiple browsers:**
Users can access the website on many browsers like Chrome, Firefox, Safari, etc.

USE CASE DIAGRAM



DETAILED USE CASE

Use Case Name:	Start Lesson	UniqueID:	UC001
Description:	The user selects a lesson from the available course options, and the system loads the lesson, customising the lesson content based on the user's progress. The chatbot provides real-time interaction, offering feedback and tracking the user's performance throughout the lesson.		
Actor:	<ol style="list-style-type: none"> 1. The user selects a lesson from the course options on the main menu. 2. The system loads the selected lesson, adapting content based on the user's progress data. 3. The system initiates the chatbot, displaying the first part of the lesson content. 4. Chatbot provides instructions and questions, collecting responses from the user. 5. The system tracks the user's responses and records performance metrics. 		
Pre-conditions:	<ul style="list-style-type: none"> • User must be registered and logged in. • Chatbot must be operational. • User must have an internet connection 		
Trigger:	The user clicks the start lesson button in the main menu.		
Main Path:	User		
Post Conditions:	The user will get access to the lesson.		

Use Case Name:	Complete Lesson	UniqueID:	UC002
Description:	The user selects "complete lesson," which updates relevant metrics such as lesson progress, skill improvement, and time spent learning. This action also contributes to the user's language learning streak and progression through their chosen course.		
Actor:	User		
Pre-conditions:	<ul style="list-style-type: none"> • User is registered and logged in. • The user is at the end of an active lesson 		
Trigger:	The User selects the "complete lesson" button.		
Main Path:	<ol style="list-style-type: none"> 1. The user finishes the lesson and clicks "Complete Lesson." 2. The system verifies that the lesson content has been completed. 3. The system updates the user's lesson completion status in their profile. 4. System increments metrics associated with lesson progress. 5. The system provides the user with feedback on lesson performance, highlighting strengths and areas for improvement. 		
Post Conditions:	The user's lesson completion, skill progression, time spent, and streak count are updated and saved.		

BRIEF USE CASES

Use Case Name:	Login	UniqueID:	UC003
Description:	The user is asked for their username and password to log in. The entered credentials are then compared to the user data that has been stored by the system. The user is given access to their account if the login credentials are accurate. The system offers options like password recovery and feedback, like an error message, if the credentials are incorrect.		
Actor:	User, Database		

Use Case Name:	Register	UniqueID:	UC004
Description:	The user is given a registration form to fill out, on which they must enter their name, email address, and password. The system verifies the data entered after the form is submitted. The system generates a new user account and saves the user's information in a database if all requirements are satisfied. The user can access all the app's features after registering.		
Actor:	User, Database		

Use Case Name:	Update Streak	UniqueID:	UC005
Description:	The system tracks the user's consecutive days of reading and updates. Their current streak count. The streak update is visually represented. In the user interface, accompanied by notifications.		
Actor:	User, System		

REFERENCES

1. Emma, 2023. *The importance of immersion: Why you should learn a language the same way you'll use it*. EF Blog. Available at: <https://www.ef.com/wwen/blog/language/the-importance-of-immersion/> [Accessed 23 October 2024].