



## Design Documentation

For

# Speakify

**SENG 31242 - System Design Project**

*Submitted By:*

**SE/2020/018 - Sandalika Liyanage**

**SE/2020/051 - Januda Bethmin**

**SE/2020/055 - Supuni Abeysinghe**

**SE/2020/057 - Thevindu Kevin**

**SE/2020/062 - Naduni Tashana**

On 21th of Dec 2024

*A report submitted in partial fulfillment of the requirements for the degree*

*Of*

**BACHELOR OF SCIENCE HONOURS IN SOFTWARE ENGINEERING**

**UNIVERSITY OF KELANIYA**

## **DECLARATION**

We hereby certify that this project and all the artifacts associated with it are our own work and it has not been submitted before nor is currently being submitted for any other degree program.

Full name of the student: Sandalika Liyanage

Student No: SE/2020/018

Signature of the student: 

Date: 2025/02/09

Full name of the student: Januda Bethmin

Student No: SE/2020/051

Signature of the student: 

Date: 2025/02/09

Full name of the student: Supuni Abeyasinghe

Student No: SE/2020/055

Signature of the student: 

Date: 2025/02/09

Full name of the student: Thevindu Kevin

Student No: SE/2020/057

Signature of the student: 

Date: 2025/02/09

Full name of the student: Naduni Tashana

Student No: SE/2020/062

Signature of the student: 

Date: 2025/02/09

## **ACKNOWLEDGEMENTS**

We would like to express our sincere gratitude to the Software Engineering Teaching Unit (SETU) for providing us with the opportunity to work on the Speakify project. As we complete the design phase, this experience has already been invaluable in applying our theoretical knowledge to real-world challenges. The guidance and support from SETU have played a crucial role in shaping our vision and refining our approach.

We also extend our appreciation to the panel of judges for their insightful feedback during our idea pitching. Their constructive suggestions have helped us enhance our design and ensure that Speakify is well-structured to address the challenges of language learning.

A special thanks to our fellow batchmates for their continuous encouragement and collaborative discussions. Their feedback and shared experiences have been instrumental in strengthening our design and preparing us for the next phases of development.

We are deeply grateful to our parents and families for their unwavering support and motivation. Their encouragement and involvement in our research have been essential in shaping the direction of Speakify.

Finally, we extend our heartfelt thanks to everyone who has contributed, directly or indirectly, to this project so far. Your insights, feedback, and encouragement have been invaluable as we lay the foundation for Speakify.

As we move forward from the design phase into development, we are excited about the journey ahead. This project has already been a tremendous learning experience, and we look forward to bringing our vision to life with the continued support of our mentors, peers, and loved ones.

## **ABSTRACT**

Speakify is an innovative language learning platform designed to make learning Sri Lankan languages, such as Sinhala and Tamil, more interactive and engaging. The project addresses the lack of personalized and immersive language-learning experiences in existing solutions by incorporating AI-driven features, gamification, and real-life conversation practice. Speakify aims to bridge the gap between traditional learning methods and modern technological advancements, making language acquisition more accessible and enjoyable.

The platform integrates speech recognition and AI-generated voice interactions to provide users with real-time pronunciation feedback and adaptive learning paths. By leveraging ElevenLabs AI for speech synthesis and analysis, Speakify enhances spoken language proficiency, ensuring learners receive a dynamic and interactive learning experience. Additionally, the system employs a structured curriculum tailored to different proficiency levels, allowing users to progress at their own pace. Gamification elements, such as achievements and interactive challenges, further enhance engagement and motivation.

Speakify's development is guided by extensive research on language learning methodologies and user needs. The system incorporates Next.js for the frontend, Python for AI-driven functionalities, and PostgreSQL (via NeonDB) for data storage, ensuring a seamless and scalable architecture. The platform is hosted on Vercel, enabling efficient deployment and performance optimization.

This document presents the system's design, technical architecture, and implementation details, along with an analysis of its feasibility and future improvements. Speakify not only offers a novel approach to language learning but also sets the foundation for expanding AI-driven education solutions tailored to regional and underrepresented languages.

## TABLE OF CONTENTS

<b>1. Introduction</b>	
<b>1.1 Project Background .....</b>	<b>8</b>
<b>1.2 Problem Definition .....</b>	<b>8</b>
<b>1.3 Aims and Objectives .....</b>	<b>9</b>
<b>1.3.1 Aim .....</b>	<b>9</b>
<b>1.3.2 Objectives .....</b>	<b>9</b>
<b>1.4 Scope with Clear Boundaries .....</b>	<b>10</b>
<b>1.4.1 In-Scope .....</b>	<b>10</b>
<b>1.4.2 Out-of-Scope .....</b>	<b>11</b>
<b>1.5 Organization of the Document .....</b>	<b>11</b>
<b>2. System Analysis</b>	
<b>2.1 Data Collection Techniques .....</b>	<b>11</b>
<b>2.1.1 Surveys .....</b>	<b>11</b>
<b>2.1.2 Interviews .....</b>	<b>11</b>
<b>2.1.3 Pilot Testing and Feedback .....</b>	<b>11</b>
<b>2.2 Detailed Analysis of Gathered Facts .....</b>	<b>12</b>
<b>2.2.1 Key Insights .....</b>	<b>12</b>
<b>2.2.2 Challenges Identified .....</b>	<b>13</b>
<b>2.2.3 Use Case Diagram and Descriptions .....</b>	<b>14</b>
<b>2.2.3.1 Use Case 01: Register .....</b>	<b>15</b>
<b>2.2.3.2 Use Case 02: Login .....</b>	<b>15</b>
<b>2.2.3.3 Use Case 03: Manage Profile .....</b>	<b>16</b>
<b>2.2.3.4 Use Case 04: Choose Language .....</b>	<b>16</b>
<b>2.2.3.5 Use Case 05: Choose Course .....</b>	<b>17</b>
<b>2.2.3.6 Use Case 06: Choose Units .....</b>	<b>17</b>
<b>2.2.3.7 Use Case 07: Choose Lessons .....</b>	<b>18</b>
<b>2.2.3.8 Use Case 08: Do Challenges .....</b>	<b>18</b>
<b>2.2.3.9 Use Case 09: Track Progress .....</b>	<b>19</b>
<b>2.2.3.10 Use Case 10: Update Leaderboard .....</b>	<b>19</b>

<b>2.2.3.11 Use Case 11: Earn Hearts &amp; Points .....</b>	<b>20</b>
<b>2.2.3.12 Use Case 12: Lose Hearts .....</b>	<b>20</b>
<b>2.2.3.13 Use Case 13: Out of Hearts .....</b>	<b>21</b>
<b>2.2.3.14 Use Case 14: Upgrade to Pro .....</b>	<b>21</b>
<b>2.2.3.15 Use Case 15: Process Payments .....</b>	<b>22</b>
<b>2.2.3.16 Use Case 16: Activate Pro Features .....</b>	<b>22</b>
<b>2.2.3.17 Use Case 17: Practice with Chatbot .....</b>	<b>23</b>
<b>2.2.3.18 Use Case 18: Manage Learning .....</b>	<b>23</b>
<b>2.2.3.19 Use Case 19: Save Learning History .....</b>	<b>24</b>
<b>2.2.3.20 Use Case 20: Generate Answers .....</b>	<b>24</b>
<b>2.3 Software Requirement Specification .....</b>	<b>26</b>
<b>2.3.1 Functional Requirements .....</b>	<b>26</b>
<b>2.3.2 Non-Functional Requirements .....</b>	<b>27</b>
<b>2.3.3 External Interface Requirements .....</b>	<b>28</b>
<b>2.4 Alternative Ways of Satisfying the Requirements .....</b>	<b>29</b>
<b>2.4.1 Gamified User Authentication .....</b>	<b>29</b>
<b>2.4.2 AI-Based Learning Analytics .....</b>	<b>30</b>
<b>2.4.3 Mobile App Usage Tracking .....</b>	<b>31</b>
<b>2.4.4 Direct API Integration .....</b>	<b>31</b>
<b>2.5 Feasibility Study .....</b>	<b>32</b>
<b>2.5.1 Technical Feasibility .....</b>	<b>32</b>
<b>2.5.2 Operational Feasibility .....</b>	<b>33</b>
<b>2.5.3 Economic Feasibility .....</b>	<b>33</b>
<b>3. System Design</b>	
<b>3.1 User Interface Wireframes .....</b>	<b>34</b>
<b>3.1.1 Desktop View .....</b>	<b>35</b>
<b>3.1.2 Mobile View .....</b>	<b>43</b>
<b>3.2 System Architecture .....</b>	<b>44</b>
<b>3.2.1 Context Diagram .....</b>	<b>47</b>
<b>3.2.2 Container Diagram .....</b>	<b>48</b>
<b>3.2.3 Component Diagram .....</b>	<b>49</b>

<b>3.2.4 Class Diagram .....</b>	<b>50</b>
<b>3.2.5 System Architecture Overview .....</b>	<b>51</b>
<b>4. Conclusion</b>	
<b>    4.1 Evaluation of Project Objectives .....</b>	<b>52</b>
<b>    4.2 Usability, Accessibility, and Reliability .....</b>	<b>53</b>
<b>    4.3 Limitations and Drawbacks .....</b>	<b>54</b>
<b>    4.4 Future Improvements .....</b>	<b>54</b>
<b>5. Appendices</b>	
<b>    Appendix A: Figma Design Link .....</b>	<b>56</b>
<b>    Appendix B: Survey Link .....</b>	<b>56</b>

## **LIST OF FIGURES AND TABLES**

<b>Use Case Diagram .....</b>	<b>14</b>
<b>Use Case 01 .....</b>	<b>15</b>
<b>Use Case 02 .....</b>	<b>15</b>
<b>Use Case 03 .....</b>	<b>16</b>
<b>Use Case 04 .....</b>	<b>16</b>
<b>Use Case 05 .....</b>	<b>17</b>
<b>Use Case 06 .....</b>	<b>17</b>
<b>Use Case 07 .....</b>	<b>18</b>
<b>Use Case 08 .....</b>	<b>18</b>
<b>Use Case 09 .....</b>	<b>19</b>
<b>Use Case 10 .....</b>	<b>19</b>
<b>Use Case 11 .....</b>	<b>20</b>
<b>Use Case 12 .....</b>	<b>20</b>
<b>Use Case 13 .....</b>	<b>21</b>
<b>Use Case 14 .....</b>	<b>21</b>
<b>Use Case 15 .....</b>	<b>22</b>
<b>Use Case 16 .....</b>	<b>22</b>
<b>Use Case 17 .....</b>	<b>23</b>
<b>Use Case 18 .....</b>	<b>23</b>
<b>Use Case 19 .....</b>	<b>24</b>
<b>Use Case 20 .....</b>	<b>24</b>
<b>Activity Diagram .....</b>	<b>25</b>
<b>Context Diagram .....</b>	<b>47</b>
<b>Container Diagram .....</b>	<b>48</b>
<b>Component Diagram .....</b>	<b>49</b>
<b>Class Diagram .....</b>	<b>50</b>

## 1. INTRODUCTION

### 1.1 Project Background

Language learning is a vital skill that enhances communication, education, and career opportunities. However, most existing language-learning platforms focus on globally dominant languages, neglecting regional languages such as Sinhala and Tamil. As a result, individuals who wish to learn these languages face a lack of structured, engaging, and interactive resources.

Traditional language-learning methods often rely on rote memorization, which fails to provide learners with real-world conversational skills. Additionally, many existing platforms do not cater to personalized learning needs, making it difficult for learners to track progress, practice pronunciation, and engage in interactive exercises.

Recognizing these challenges, the Speakify project aims to develop an innovative AI-powered language learning platform specifically designed for Sinhala and Tamil learners. The platform will integrate gamification, AI-driven feedback, and real-life conversation practice to create an engaging and effective learning experience.

### 1.2 Problem Definition

Current language-learning platforms are primarily designed for major global languages, offering limited support for regional languages. This creates several challenges for learners:

- **Lack of Engaging Content:** Existing resources are often outdated, text-heavy, and do not incorporate modern interactive techniques.
- **Absence of Personalized Learning:** Most platforms use a generic approach, failing to adapt to individual learners' strengths and weaknesses.
- **Limited Conversational Practice:** Many applications focus on vocabulary memorization rather than real-life speaking exercises.
- **Insufficient AI-Driven Feedback:** Learners struggle with pronunciation and grammar due to a lack of automated, instant feedback mechanisms.

- **Minimal Support for Regional Languages:** Sinhala and Tamil learners have very few structured learning platforms compared to widely spoken languages like English or Spanish.

Speakify seeks to address these issues by integrating AI-powered feedback, gamification techniques, and adaptive learning paths, making it easier for users to learn, practice, and retain a new language.

### **1.3 Aims and Objectives**

#### **Aim**

The aim of Speakify is to develop an interactive and AI-driven language-learning platform that makes learning Sinhala and Tamil engaging, personalized, and effective.

#### **Objectives**

- **Personalized Learning Paths:** Implement AI-based adaptive learning techniques to tailor lessons to individual learners.
- **Real-World Conversational Practice:** Enable users to engage in simulated conversations using speech recognition and AI-driven responses.
- **Gamification for Engagement:** Introduce badges, leaderboards, and interactive exercises to motivate continuous learning.
- **AI-Powered Pronunciation Feedback:** Use speech analysis technology to provide instant feedback on pronunciation and fluency.
- **Cultural Integration:** Incorporate real-life scenarios, stories, and dialogues relevant to Sri Lankan culture to enhance contextual learning.
- **User Progress Tracking:** Offer dashboards with performance analytics, enabling learners to track progress and adjust learning goals.

## 1.4 Scope with Clear Boundaries

The Speakify platform is designed as a digital language-learning solution focused on Sinhala and Tamil. It aims to provide an immersive and interactive experience while maintaining a clear set of functional boundaries.

### In-Scope

- AI-driven adaptive learning to customize lessons based on user progress.
- Speech recognition and pronunciation analysis for real-time feedback.
- Gamification elements such as rewards, challenges, and streaks.
- Conversation simulations using AI chatbots and real-world dialogues.
- Multimedia content including audio, video, and interactive exercises.
- User analytics and progress tracking through detailed reports.

### Out-of-Scope

- The platform does not replace human tutors but supplements learning with AI-driven features.
- It will not offer translations for all global languages, as the primary focus is Sinhala and Tamil.
- No offline mode in the initial version; all functionalities require an internet connection.
- No formal certification will be provided, although users will receive achievement badges.

## 1.5 Organization of the Document

This document is structured as follows:

- **Chapter 1 - Introduction:** Provides an overview of the project background, problem statement, aims, objectives, and scope.
- **Chapter 2 - System Analysis:** Covers data collection techniques, user requirements, use case diagrams, and feasibility studies.
- **Chapter 3 - System Design:** Explains the high-level system architecture, UI wireframes, component diagrams, and technology stack.

- **Chapter 4 - Implementation and Testing:** Details feature implementation, testing methodologies, and system validation.
- **Chapter 5 - Conclusion and Future Enhancements:** Summarizes the key outcomes, challenges, limitations, and potential improvements for Speakify.

## 2. SYSTEM ANALYSIS

### 2.1 Data Collection Techniques

To better understand the needs and requirements for a **language-learning platform** for Sinhala and Tamil, various data collection methods were employed. These methods were aimed at identifying the challenges, preferences, and pain points of potential users, primarily students and individuals interested in learning these regional languages. The data collection process involved **surveys, interviews, and feedback from pilot users**.

- **Surveys:**

Surveys were distributed to university students, language enthusiasts, and learners who have shown an interest in Sinhala and Tamil. The goal was to gather quantitative and qualitative data on their existing language-learning experiences, preferences, and challenges. The surveys included both multiple-choice and open-ended questions to cover a wide range of topics such as:

- Learners' current language skills and goals.
- Previous experiences with language-learning apps.
- Challenges faced in learning Sinhala and Tamil.
- Desired features in a language-learning platform.
- Interest in using AI-driven language tools (e.g., speech recognition, real-time feedback).

- **Interviews:**

In-depth interviews were conducted with a select group of learners to explore specific pain points not captured in the survey. The interviews provided deeper insights into:

- Preferred learning methods (audio, video, text, interactive).

- The importance of real-life conversation practice.
  - Frustrations with current language learning methods (e.g., limited interaction, lack of personalized content).
  - The role of gamification in enhancing learning engagement.
- **Pilot Testing and Feedback:**

After the initial design and development phase, a **pilot version of Speakify** was tested by a small group of users. Their feedback helped refine features such as:

- **User interface design** and ease of navigation.
- Effectiveness of AI feedback on pronunciation and grammar.
- Engagement with gamification elements like points, rewards, and levels.
- Functionality of real-time conversation practice with the AI chatbot.

Together, these methods provided a comprehensive understanding of the learners' needs, preferences, and challenges, guiding the design and features of the Speakify platform.

## 2.2 Detailed Analysis of Gathered Facts

Following the data collection process, a detailed analysis was performed to identify key requirements for the Speakify platform. This analysis categorized the findings into **functional** and **non-functional requirements**. The key insights were translated into a set of use cases and features that Speakify should incorporate.

### Key Insights:

From the surveys and interviews, the following core insights emerged:

1. **Personalized Learning:** Many learners expressed the need for a platform that adapts to their learning pace, helping them focus on areas where they are weakest. AI-driven content that customizes lessons based on user progress was identified as crucial for better learning outcomes.
2. **Conversational Practice:** There was a strong desire for real-life **speaking exercises** where learners could practice conversations in Sinhala and Tamil. Many learners felt that traditional language apps lack the focus on practical conversation skills.

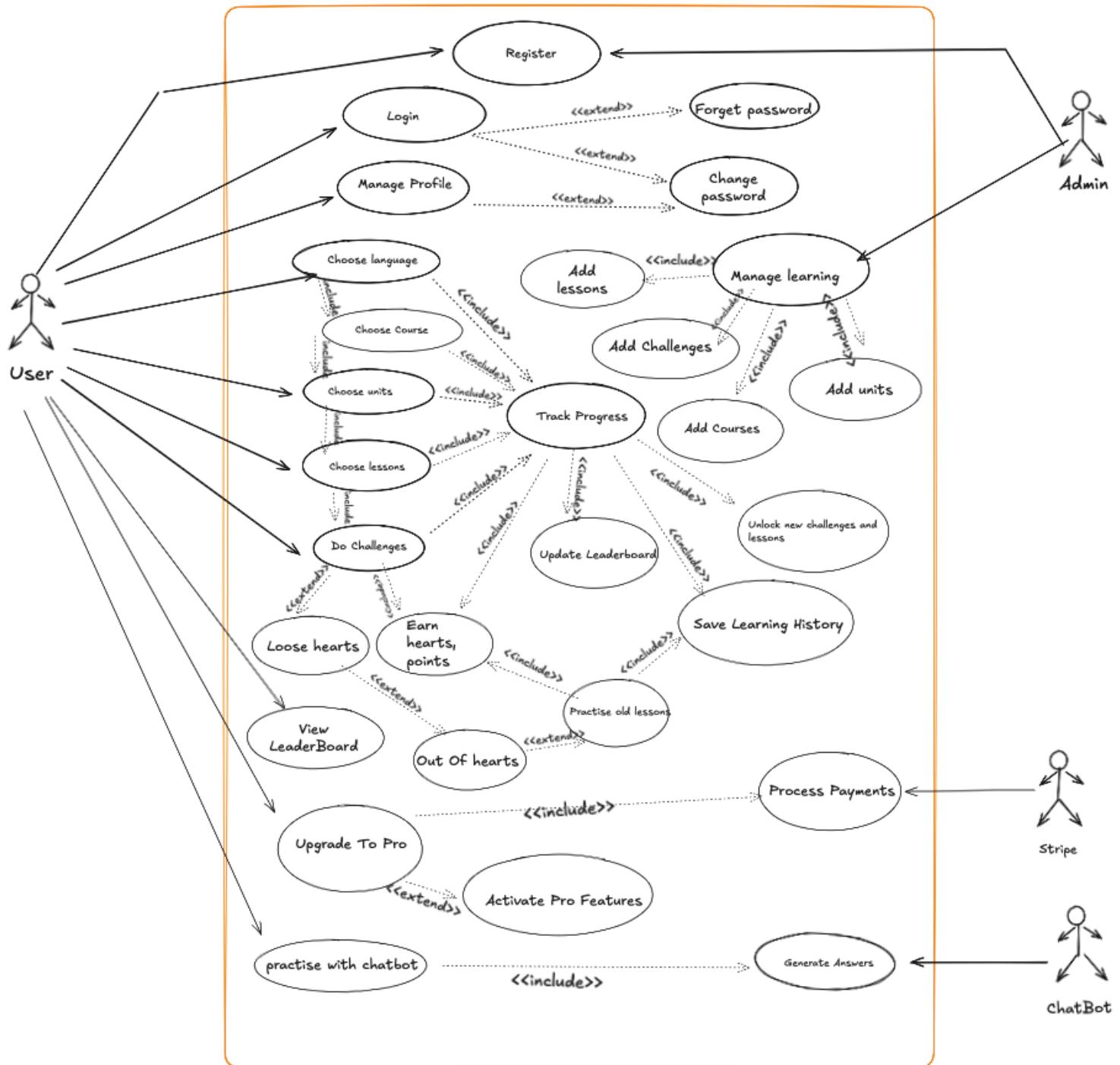
3. **Gamification:** Respondents showed a high level of interest in using gamified elements to make learning more engaging and rewarding. Features such as **badges, leaderboards, and levels** were popular suggestions to motivate users to continue learning.
4. **Real-Time Feedback:** Feedback on pronunciation and grammar is critical for learners. Many participants wanted a system that could **provide instant feedback** as they practice speaking, helping them correct mistakes in real-time.
5. **Ease of Use:** The user interface (UI) needed to be simple and intuitive, as many learners were overwhelmed by complex designs in other language learning platforms.

### **Challenges Identified:**

Several challenges were highlighted by the data analysis that Speakify aims to address:

1. **Lack of Personalized Learning Paths:** Many users feel that language learning platforms are too generic, offering the same content to all users regardless of their individual learning pace and needs.
2. **Limited Focus on Speaking Skills:** While reading and writing are covered extensively in most platforms, speaking skills, particularly **pronunciation** and **conversation practice**, were identified as weak points.
3. **Boredom with Traditional Learning Methods:** A significant portion of the respondents felt that conventional language learning apps were too monotonous, leading to a lack of sustained engagement.
4. **Cultural Context:** Users emphasized the importance of learning the cultural aspects of a language, especially for languages like Sinhala and Tamil, which have rich cultural contexts that can affect the meaning of words and phrases.
5. **Lack of Real-Time Feedback:** Many respondents complained about the absence of immediate feedback on pronunciation and grammatical errors while practicing speaking.

## 2.2.1 Use Case Diagram and Descriptions



### **2.2.1.1 Use Case 01 : Register**

<b>Use Case ID</b>	<b>UC 001</b>
Description	User registers a new account
Actors	User
Preconditions	None
Postconditions	User account created
Main Flow	<ol style="list-style-type: none"><li>1. User enters registration details.</li><li>2. System validates input.</li><li>3. System creates a user account.</li><li>4. Confirmation message is shown.</li></ol>
Alternative Flow	2a. If input is invalid, prompt the user to re-enter details.

### **2.2.1.2 Use Case 02 : Login**

<b>Use Case ID</b>	<b>UC 002</b>
Description	User logs into the system
Actors	User
Preconditions	User must be registered
Postconditions	User is logged in
Main Flow	<ol style="list-style-type: none"><li>1. User enters credentials.</li><li>2. System verifies credentials.</li><li>3. User is granted access.</li></ol>

Alternative Flow	2a. If credentials are invalid, show error message
------------------	--

### 2.2.1.3 Use Case 03 : Manage Profile

<b>Use Case ID</b>	<b>UC 003</b>
Description	User updates personal information
Actors	User
Preconditions	User must be logged in
Postconditions	Profile updated
Main Flow	<ol style="list-style-type: none"> <li>1. User navigates to profile settings.</li> <li>2. User update information.</li> <li>3. System saves changes.</li> </ol>
Alternative Flow	2a. If invalid data, show an error message.

### 2.2.1.4 Use Case 04 : Choose Language

<b>Use Case ID</b>	<b>UC 004</b>
Description	User selects a language to learn
Actors	User
Preconditions	User must be logged in
Postconditions	Language selection saved

Main Flow	<ol style="list-style-type: none"> <li>1. User selects the language.</li> <li>2. System updates preferences.</li> </ol>
-----------	---

#### 2.2.1.5 Use Case 05 : Choose Course

<b>Use Case ID</b>	<b>UC 005</b>
Description	User selects a course in a chosen language
Actors	User
Preconditions	User must have selected a language
Postconditions	Course selection saved
Main Flow	<ol style="list-style-type: none"> <li>1. User selects course.</li> <li>2. System updates preferences.</li> </ol>

#### 2.2.1.6 Use Case 06 : Choose Units

<b>Use Case ID</b>	<b>UC 006</b>
Description	User selects specific learning units
Actors	User
Preconditions	User must have selected a course
Postconditions	Unit selection saved
Main Flow	<ol style="list-style-type: none"> <li>1. User selects the unit.</li> <li>2. System updates preferences.</li> </ol>

### **2.2.1.7 Use Case 07 : Choose Lessons**

<b>Use Case ID</b>	<b>UC 007</b>
Description	User selects specific lessons
Actors	User
Preconditions	User must have selected a unit
Postconditions	Lesson selection saved
Main Flow	<ol style="list-style-type: none"><li>1. User selects the lesson.</li><li>2. System updates preferences.</li></ol>

### **2.2.1.8 Use Case 08 : Do Challenges**

<b>Use Case ID</b>	<b>UC 008</b>
Description	User completes language challenges
Actors	User
Preconditions	User must have selected a lesson
Postconditions	Challenge completion recorded
Main Flow	<ol style="list-style-type: none"><li>1. User starts a challenge.</li><li>2. System verifies the answers.</li><li>3. System provides feedback.</li></ol>
Alternative Flow	2a. If the answer is incorrect, prompt the user to retry.

### **2.2.1.9 Use Case 09 : Track Progress**

<b>Use Case ID</b>	<b>UC 009</b>
Description	System tracks user learning progress
Actors	User
Preconditions	User must have completed lessons
Postconditions	Progress updated
Main Flow	<ol style="list-style-type: none"><li>1. System records completed lessons.</li><li>2. Progress is updated.</li></ol>

### **2.2.1.10 Use Case 10 : Update Leaderboard**

<b>Use Case ID</b>	<b>UC 010</b>
Description	Leaderboard is updated based on user points
Actors	System
Preconditions	User must have completed challenges
Postconditions	Leaderboard updated
Main Flow	<ol style="list-style-type: none"><li>1. User earns points.</li><li>2. System updates leaderboard.</li></ol>

### **2.2.1.11 Use Case 11 : Earn Hearts & Points**

<b>Use Case ID</b>	<b>UC 011</b>
Description	User gains hearts and points for activities
Actors	User
Preconditions	User must complete challenges
Postconditions	Hearts/points awarded
Main Flow	<ol style="list-style-type: none"><li>1. User completes the challenge.</li><li>2. System updates rewards.</li></ol>

### **2.2.1.12 Use Case 12 : Lose Hearts**

<b>Use Case ID</b>	<b>UC 012</b>
Description	User loses hearts when failing challenges
Actors	User
Preconditions	User must be participating in challenges
Postconditions	Hearts deducted
Main Flow	<ol style="list-style-type: none"><li>1. User fails challenge.</li><li>2. System deducts hearts.</li></ol>
Alternative Flow	2a. User can regain hearts via purchases or practice.

### **2.2.1.13 Use Case 13 : Out of Hearts**

<b>Use Case ID</b>	<b>UC 013</b>
Description	User runs out of hearts and is restricted
Actors	User
Preconditions	User must have 0 hearts
Postconditions	User is restricted from further lessons
Main Flow	<ol style="list-style-type: none"><li>1. System checks heart count.</li><li>2. If 0, restrict access.</li></ol>

### **2.2.1.14 Use Case 14 : Upgrade to Pro**

<b>Use Case ID</b>	<b>UC 014</b>
Description	User subscribes to Pro version
Actors	User
Preconditions	User must have a valid payment method
Postconditions	Pro features activated
Main Flow	<ol style="list-style-type: none"><li>1. User selects the upgrade.</li><li>2. System processes payment.</li><li>3. System activates Pro.</li></ol>
Alternative Flow	<ol style="list-style-type: none"><li>2a. If payment fails, show an error message.</li></ol>

### **2.2.1.15 Use Case 15 : Process Payments**

<b>Use Case ID</b>	<b>UC 015</b>
Description	Handles user payments
Actors	User
Preconditions	User must have a valid payment method
Postconditions	Payment processed
Main Flow	<ol style="list-style-type: none"><li>1. User enters payment details.</li><li>2. Stripe processes transactions.</li></ol>
Alternative Flow	2a. If payment fails, show error message

### **2.2.1.16 Use Case 16 : Activate Pro Features**

<b>Use Case ID</b>	<b>UC 016</b>
Description	Unlocks Pro features for user
Actors	User
Preconditions	User must have upgraded to Pro
Postconditions	Pro features enabled
Main Flow	<ol style="list-style-type: none"><li>1. System verifies Pro status.</li><li>2. Unlocks premium content like unlimited hearts and points.</li></ol>

### **2.2.1.17 Use Case 17 : Practice with Chatbot**

<b>Use Case ID</b>	<b>UC 017</b>
Description	User interacts with AI chatbot for practice
Actors	User, Chatbot
Preconditions	User must be logged in
Postconditions	Chat session recorded
Main Flow	<ol style="list-style-type: none"><li>1. User opens the chatbot.</li><li>2. User asks questions based on language.</li><li>3. System generates responses.</li></ol>

### **2.2.1.18 Use Case 18 : Manage Learning**

<b>Use Case ID</b>	<b>UC 018</b>
Description	Admin manages courses, lessons, challenges
Actors	Admin
Preconditions	Admin must be logged in
Postconditions	Learning content updated
Main Flow	<ol style="list-style-type: none"><li>1. Admin adds/updates courses.</li><li>2. System saves changes.</li></ol>

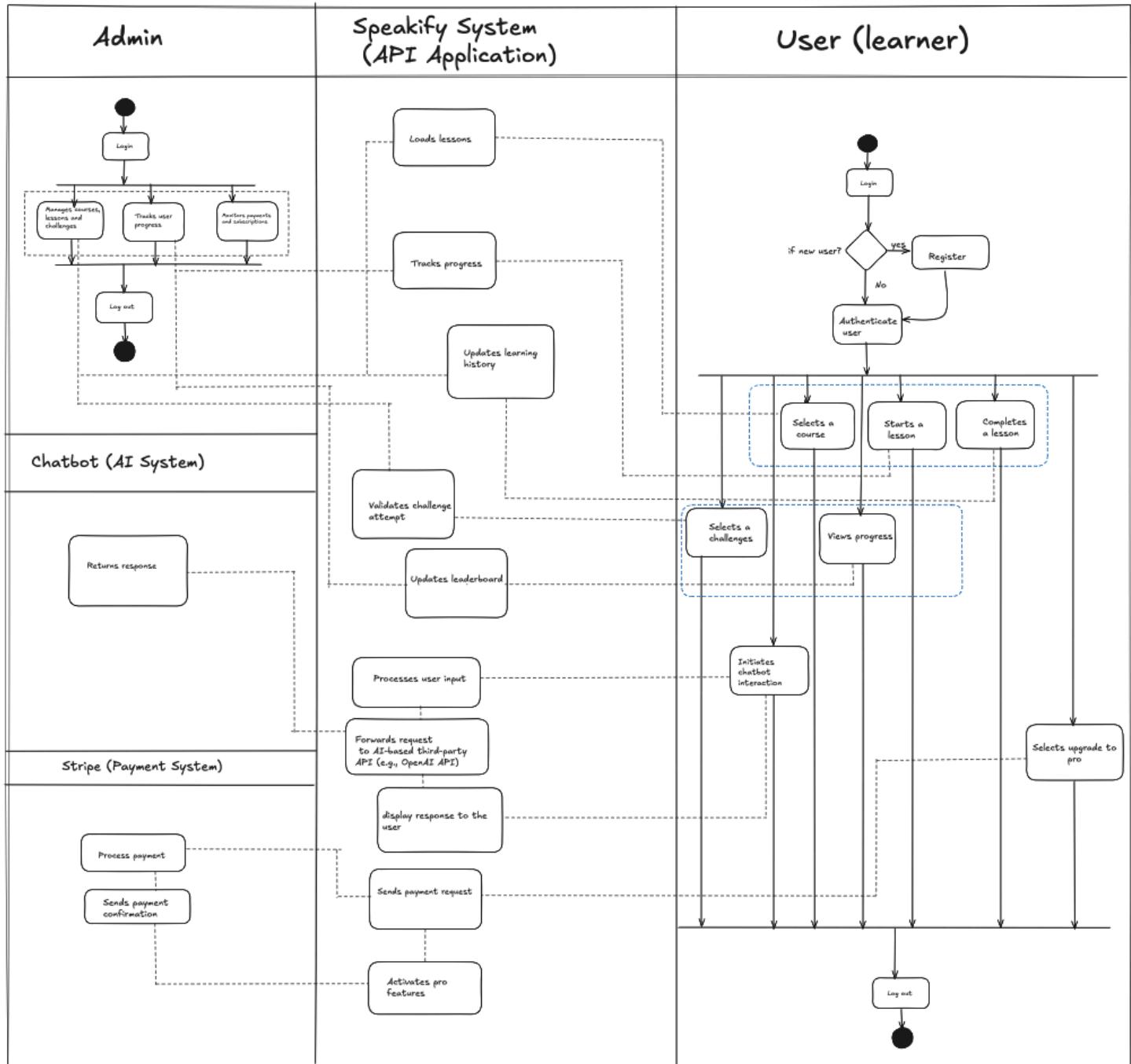
### **2.2.1.19 Use Case 19 : Save Learning History**

<b>Use Case ID</b>	<b>UC 019</b>
Description	System saves user progress
Actors	User
Preconditions	User must be logged in
Postconditions	Progress saved
Main Flow	<ol style="list-style-type: none"><li>1. User completes activities.</li><li>2. System records progress.</li></ol>

### **2.2.1.20 Use Case 20 : Generate Answers**

<b>Use Case ID</b>	<b>UC 020</b>
Description	Chatbot provides answers during interaction
Actors	User, Chatbot
Preconditions	User must ask a question
Postconditions	Answer displayed
Main Flow	<ol style="list-style-type: none"><li>1. User inputs query.</li><li>2. Chatbot generates responses.</li></ol>

## 2.2.2 Activity Diagram



## **2.3 Software Requirement Specification**

### **2.3.1 Functional Requirements of the Group Admin:**

1. Group Admin should be able to successfully log in to the system.
2. Group Admin should be able to create and manage learner groups.
3. Group Admin should be able to assign and remove users within a group.
4. Group Admin should be able to manage access levels and permissions for different users.
5. Group Admin should be able to monitor group progress and individual learning analytics.
6. Group Admin should be able to create and assign learning paths or challenges.
7. Group Admin should be able to view real-time engagement metrics and learner activity.
8. Group Admin should be able to access a comprehensive dashboard with visual analytics.
9. Group Admin should be able to generate reports on learning progress and engagement.
10. Group Admin should be able to send notifications and announcements to the group.
11. Group Admin should be able to deactivate or archive a learning group if necessary.

### **2.3.2 Functional Requirements of the Regular User:**

1. Regular User should be able to successfully log in to the system.
2. Regular User should be able to access personalized learning materials.
3. Regular User should be able to track their own learning progress.
4. Regular User should be able to participate in interactive exercises and gamified challenges.
5. Regular User should be able to compare progress with other learners in their group.
6. Regular User should receive real-time feedback on quizzes and exercises.
7. Regular User should be able to receive notifications for reminders, milestones, and updates.
8. Regular User should be able to interact with AI-driven language practice features.
9. Regular User should be able to update their profile and learning preferences.

### **2.3.3 Non-Functional Requirements for Speakify:**

#### **1. Performance:**

- The system should respond to user interactions within a maximum delay of 3 seconds.
- The system should support up to 1,000 simultaneous users without performance degradation.
- The platform should consume minimal device resources while running.

#### **2. Scalability:**

- The system should be able to scale horizontally to accommodate increasing user loads.
- The architecture should allow easy integration of additional language modules.

#### **3. Reliability:**

- The system should have an uptime of 99.9% to ensure continuous access to learning materials.
- In case of service disruptions, data should be synced within 2 minutes after reconnection.

#### **4. Usability:**

- The user interface should be intuitive, requiring minimal training for new users.
- Users should be able to complete key actions (e.g., starting a lesson, tracking progress) within three clicks.
- The platform should be optimized for both web and mobile (iOS and Android).

#### **5. Security:**

- The system should support multi-factor authentication for user logins.
- Role-based access control should restrict administrative features to authorized users.
- User data should be encrypted both in transit and at rest.

#### **6. Maintainability:**

- The system should allow updates with minimal downtime.
- Logs should be retained for at least 12 months for auditing and troubleshooting.

## **7. Portability:**

- The application should be accessible on major web browsers (Chrome, Firefox, Safari) and mobile platforms.
- The system should support cloud-based and on-premise deployments.

## **8. Availability:**

- The system should operate 24/7, with scheduled maintenance windows not exceeding 1 hour per month.
- Offline mode should allow learning activities to continue without the internet for up to 24 hours.

## **9. Data Integrity:**

- Learning progress and user data should be consistently synced across devices.
- The system should perform automated data integrity checks every hour.

## **10. Legal and Compliance:**

- The platform should comply with data privacy regulations such as GDPR and CCPA.
- AI-generated content should adhere to ethical language learning standards.

### **2.3.4 External Interface Requirements:**

#### **1. Hardware Interfaces:**

- The system should be compatible with devices running iOS, Android, Windows, and macOS.
- It should support smartphones, tablets, desktops, and laptops.

#### **2. Software Interfaces:**

- The system must integrate with third-party authentication services (e.g., OAuth, Google Authenticator).
- Users should be able to export learning progress data in CSV or PDF format.
- Future enhancements should include integration with AI voice recognition APIs.

#### **3. Communication Interfaces:**

- The system should support real-time notifications via push notifications (Firebase, Apple Push Notification Service).

- Users should receive email notifications for key events (lesson reminders, progress reports).

#### **4. User Interfaces:**

- The web and mobile app should have a responsive design optimized for various screen sizes.
- The dashboard should include interactive data visualizations for progress tracking.
- The learning interface should provide engaging, intuitive interactions, including gamified elements.

## **2.4 Alternative Ways of Satisfying the Requirements**

Several alternative solutions can address the functional requirements of **Speakify**, such as monitoring user engagement, managing subscription billing, and ensuring a fair distribution of platform resources. Each solution has its own strengths and limitations, which must be considered in light of the specific needs of an adaptive language learning platform.

### **1. Gamified User Authentication and Progress Tracking**

One possible alternative is using a **gamified authentication system** that requires users to engage with small interactive tasks before accessing learning content. This method can encourage active participation while tracking user engagement.

- **Functionality:** When users log in, they are prompted to complete a micro-challenge (such as identifying a word or pronouncing a simple phrase) before proceeding to the main learning modules. Their progress is recorded, ensuring engagement before access.
- **Advantages:**
  - **Enhanced Engagement:** Encourages active participation rather than passive login behavior.
  - **No Additional Installations:** Users can access features seamlessly without extra downloads.
  - **Centralized User Tracking:** The system provides real-time insights into user engagement levels.

- **Disadvantages:**
  - **Potential User Friction:** Some users may find repeated challenges unnecessary or intrusive.
  - **Infrastructure Requirements:** Requires a robust backend to track progress and authenticate users dynamically.

## 2. AI-Based Learning Analytics for Progress Monitoring

Another potential solution involves **leveraging AI-driven analytics** to monitor language learning progress and provide personalized recommendations.

- **Functionality:** The AI system analyzes user performance, identifying learning patterns and adjusting lesson difficulty accordingly. Administrators and users can access reports for tracking improvements.
- **Advantages:**
  - **Automated Learning Path Adjustments:** AI personalized content based on individual progress.
  - **Comprehensive Data Insights:** Enables Speakify to track learning efficiency and optimize curriculum delivery.
  - **User-Friendly Monitoring:** Eliminates the need for manual tracking by offering real-time analytics.
- **Disadvantages:**
  - **High Initial Development Effort:** Requires a well-trained AI model and robust data collection.
  - **Data Privacy Concerns:** User data must be handled securely to maintain trust and compliance.

### **3. Mobile App Usage Tracking for Personalized Feedback**

A third alternative is **integrating mobile-based usage tracking** to assess learning habits and provide tailored feedback.

- **Functionality:** A mobile app tracks user activity, including session duration, pronunciation attempts, and quiz performance. This data is then processed to offer adaptive learning recommendations.
- **Advantages:**
  - **Detailed Learning Insights:** Users receive personalized feedback based on actual usage patterns.
  - **Customizable Learning Alerts:** Notifications encourage users to stay on track with their learning goals.
  - **Cross-Platform Accessibility:** Works across smartphones and tablets for a seamless experience.
- **Disadvantages:**
  - **User Compliance Required:** Users must consistently engage with the app for accurate tracking.
  - **Resource Usage Considerations:** Continuous tracking may impact battery life and device performance.

### **4. Direct API Integration with Language Proficiency Platforms**

A more advanced alternative involves **integrating Speakify with external language proficiency platforms** to track and validate learning progress.

- **Functionality:** Speakify partners with third-party assessment platforms (such as CEFR-based testing services) to provide official proficiency tracking and certification.
- **Advantages:**
  - **Accurate and Standardized Evaluation:** Provides users with industry-recognized language proficiency metrics.
  - **Minimal Maintenance for Speakify:** Reduces the need for complex in-house evaluation systems.

- **Disadvantages:**
  - **Third-Party Dependency:** Speakify relies on external providers for assessment accuracy.
  - **Integration Complexity:** Requires API development and ongoing coordination with assessment providers.

Each alternative presents a viable way to enhance Speakify's adaptive learning experience, but the best approach depends on the platform's strategic focus. Gamified authentication enhances engagement, AI-based analytics provide deep insights, mobile tracking personalized feedback, and API integration ensures standardized assessments.

## 2.5 Feasibility Study

The feasibility of the Speakify system, along with potential alternative solutions, is evaluated based on the following criteria: technical feasibility, operational feasibility, and economic feasibility.

### 1. Technical Feasibility

- **Speakify Platform:** Speakify leverages AI-driven speech recognition, natural language processing, and gamification techniques to enhance language learning. The backend, built using Python and PostgreSQL, ensures scalability and efficient data handling. The use of Elevenlabs AI for voice synthesis provides an engaging and realistic conversation experience. The frontend, developed in Next.js and deployed on Vercel, ensures responsiveness and a smooth user experience across devices. However, real-time speech processing and pronunciation evaluation may require optimization to minimize latency.
- **Traditional Language Learning Apps:** Most existing apps rely on pre-recorded lessons and structured exercises, making them technically viable but less interactive. AI integration is minimal, limiting real-time feedback.
- **Video-Based Learning:** Platforms like YouTube provide extensive language learning content, but they lack interactive and adaptive learning features. This approach is technically straightforward but does not personalize the learning experience.

- **AI Chatbots for Language Learning:** AI-driven chatbots provide real-time interaction, but they often struggle with nuanced language understanding, making conversational learning less effective than Speakify's integrated approach.

## 2. Operational Feasibility

- **Speakify Platform:** The system is designed for seamless user engagement, providing personalized lessons, interactive exercises, and real-time feedback. Users can track their progress, receive AI-generated pronunciation corrections, and engage in gamified learning challenges. The primary challenge lies in ensuring AI-generated content remains accurate and contextually relevant.
- **Traditional Language Learning Apps:** These apps are operationally feasible but lack adaptability. Users follow predefined lessons with limited interaction, making engagement levels lower.
- **Video-Based Learning:** This method requires passive learning, which may not effectively reinforce language skills. Operational feasibility is high, but the lack of interactive elements limits its effectiveness.
- **AI Chatbots for Language Learning:** While chatbots provide conversational practice, they often fail to mimic real-life interactions accurately. Ensuring AI-driven responses are contextually appropriate remains a challenge.

## 3. Economic Feasibility

- **Speakify Platform:** Speakify leverages cost-effective cloud services such as Vercel and NeonDB, minimizing infrastructure expenses. The use of AI models like Elevenlabs and open-source tools reduces development costs. The platform can generate revenue through premium features and subscription-based learning models while maintaining an affordable entry point for users.
- **Traditional Language Learning Apps:** Developing and maintaining structured courses requires significant investment in content creation, increasing costs. Monetization relies on subscription models, often making access expensive.

- **Video-Based Learning:** This is a cost-effective solution since it relies on freely available content, but it lacks an effective monetization strategy unless paired with ads or premium content.
- **AI Chatbots for Language Learning:** Implementing AI-driven chatbots involves ongoing costs related to model training, API usage, and performance optimization.

Speakify is technically, operationally, and economically feasible for deployment in the language learning market. Its AI-driven, interactive approach provides a more engaging experience compared to traditional learning methods. While alternatives like video-based learning or AI chatbots offer certain benefits, they lack the depth, personalization, and engagement provided by Speakify's integrated platform.

### 3. SYSTEM DESIGN

The system design for Speakify outlines how user requirements and functional expectations are implemented through software architecture. This chapter provides detailed insights into the **user interfaces, system components, and their interactions**. The goal is to ensure that Speakify remains **scalable, maintainable, and extensible** while providing a seamless and engaging learning experience.

---

#### 3.1 User Interface Wireframes

To create an **engaging and user-friendly experience**, Speakify is designed with **intuitive and interactive interfaces** that cater to different learning styles. The UI ensures learners can **navigate lessons, practice conversations, and track their progress** effortlessly.

These UI designs focus on making **language learning interactive, engaging, and personalized**, ensuring that Speakify meets its goal of **enhancing conversational fluency through AI-driven learning**.

## Desktop view,

**before login page**

**login-email error**

**signup**

**login-email enter**

**login-password error**

**login-password enter**

user-account...

home

learn

The screenshots illustrate the user interface of the Speakify app. The top section shows the user account screen with a dark theme, featuring a sidebar with icons for Learn, Leaderboard, Quest, Shop, and Chat Bot, and a main area with a 'Start' button and a list of five stars. The bottom section shows the home screen with a light theme, displaying 'Language Courses' for 'SINHALA' and 'TAMIL'. The right side of the bottom section shows the learn screen for 'Unit 01' of the Sinhala course, which aims to teach the basics of Sinhala. It includes a sidebar with the same navigation options as the account screen, a 'Start' button, and a list of five stars.

**quest**

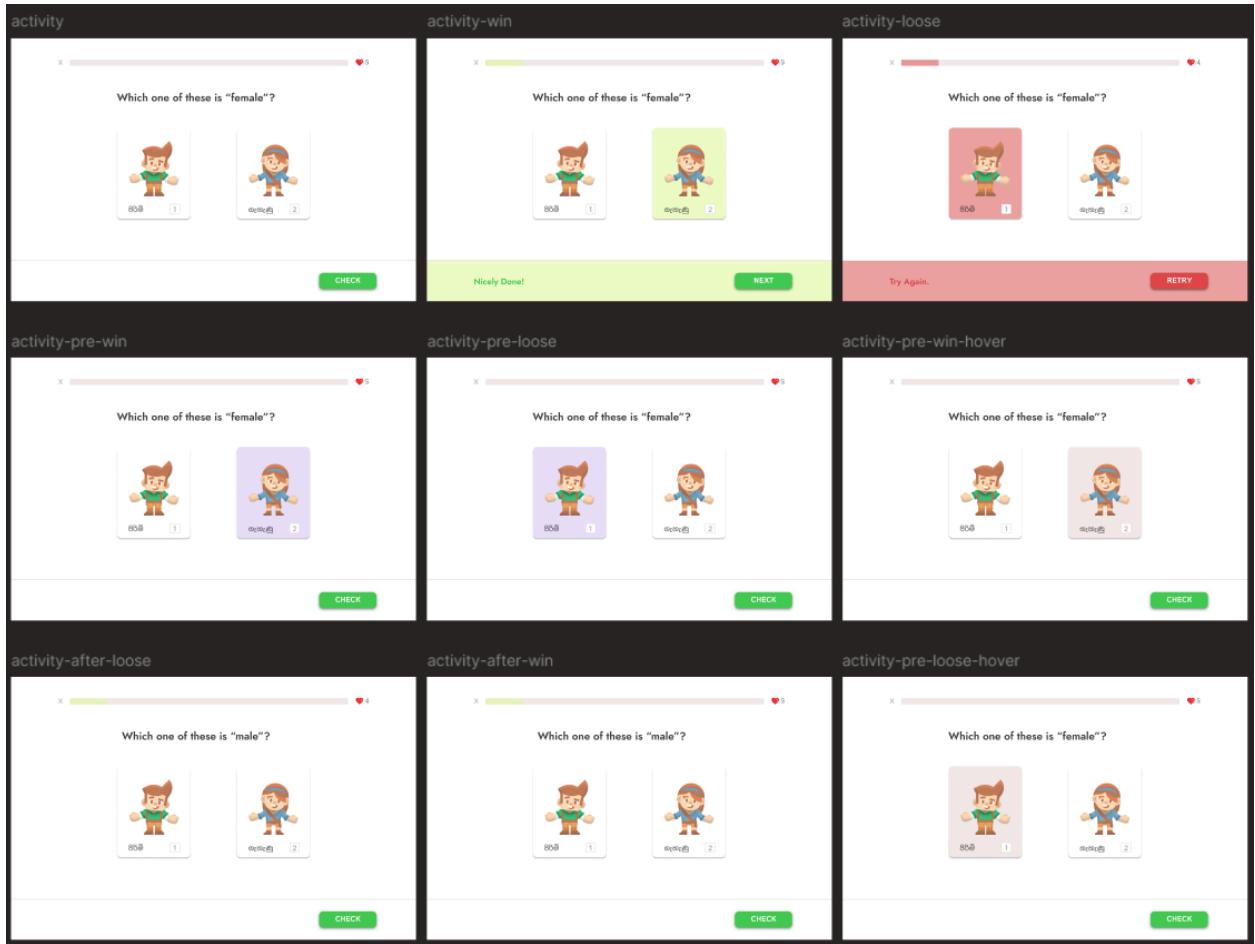
**chatbot**

**leaderboard**

**shop**

The screenshots show the following features:

- Quest:** A screen titled "Quests" showing various tasks to earn XP. It includes a sidebar with "SPEAKIFY" logo, "LEARN", "LEADERBOARD" (highlighted), "QUEST" (highlighted), "SHOP", and "CHAT BOT". A "Username" field is at the bottom.
- Chatbot:** A screen titled "ChatBot" showing a conversation history with a bot. It includes a sidebar with "SPEAKIFY" logo, "LEARN", "LEADERBOARD", "QUEST", "SHOP" (highlighted), and "CHAT BOT". A "Username" field is at the bottom.
- Leaderboard:** A screen titled "Leaderboard" showing a user's rank. It includes a sidebar with "SPEAKIFY" logo, "LEARN", "LEADERBOARD" (highlighted), "QUEST", "SHOP", and "CHAT BOT". A "Username" field is at the bottom.
- Shop:** A screen titled "Shop" showing items to purchase with points. It includes a sidebar with "SPEAKIFY" logo, "LEARN", "LEADERBOARD", "QUEST", "SHOP" (highlighted), and "CHAT BOT". A "Username" field is at the bottom.





**activity-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**CHECK**

**activity-win-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**Nicely Done!** **NEXT**

**activity-pre-win-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**CHECK**

**activity-pre-loose-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**CHECK**

**activity-loose-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**Try Again.** **RETRY**

**activity-after-win-no-heart**

Which one of these is "male"?

Bob [1] Amy [2]

**CHECK**

**activity-pre-win-hover-no-heart**

Which one of these is "female"?

Bob [1] Amy [2]

**CHECK**

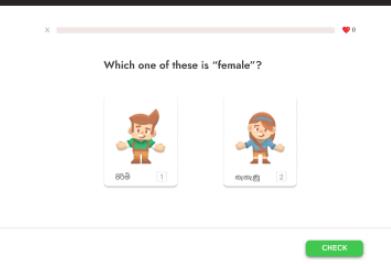
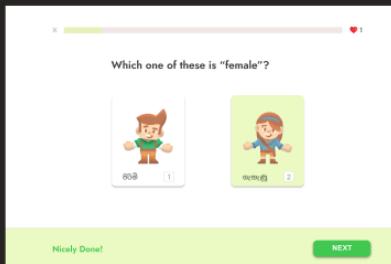
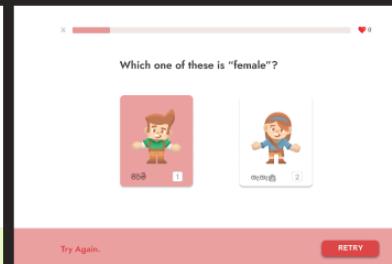
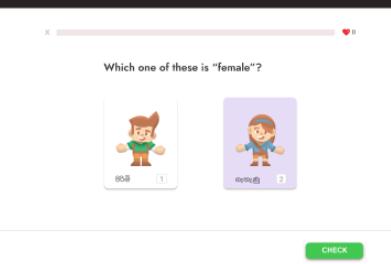
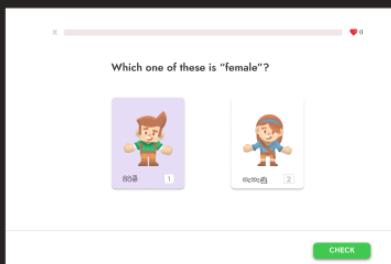
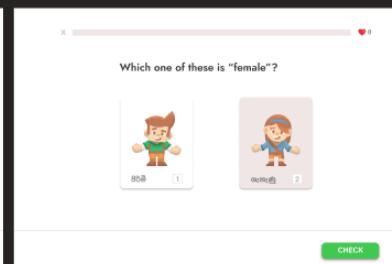
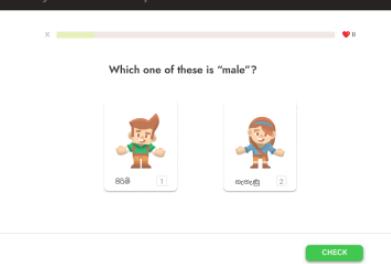
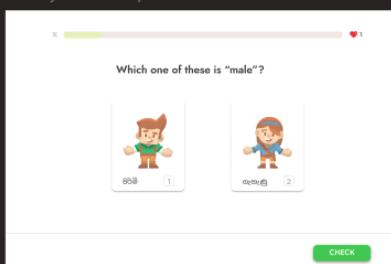
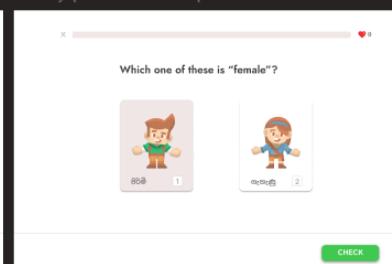
**activity-pre-loose-hover-no-heart**

Which one of these is "female"?

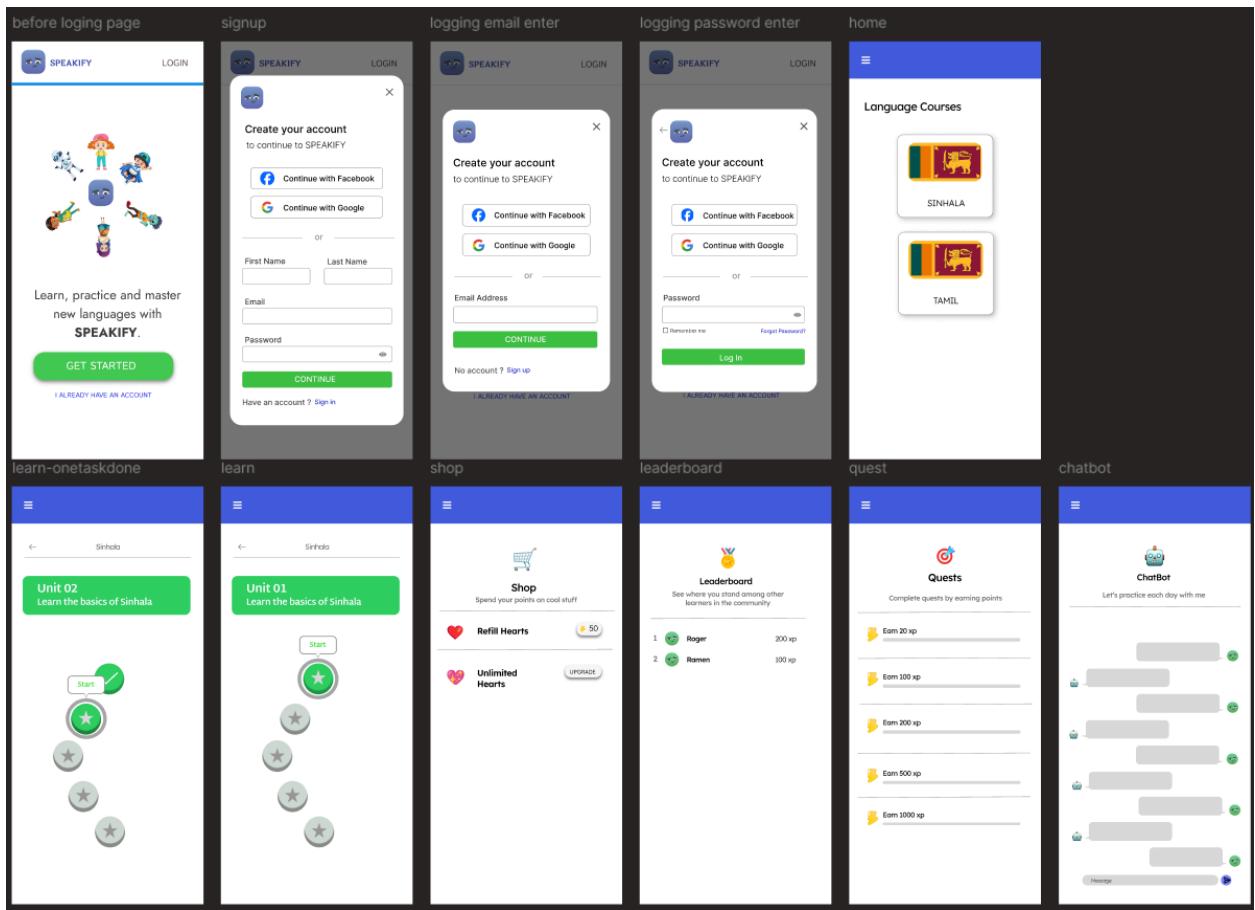
Bob [1] Amy [2]

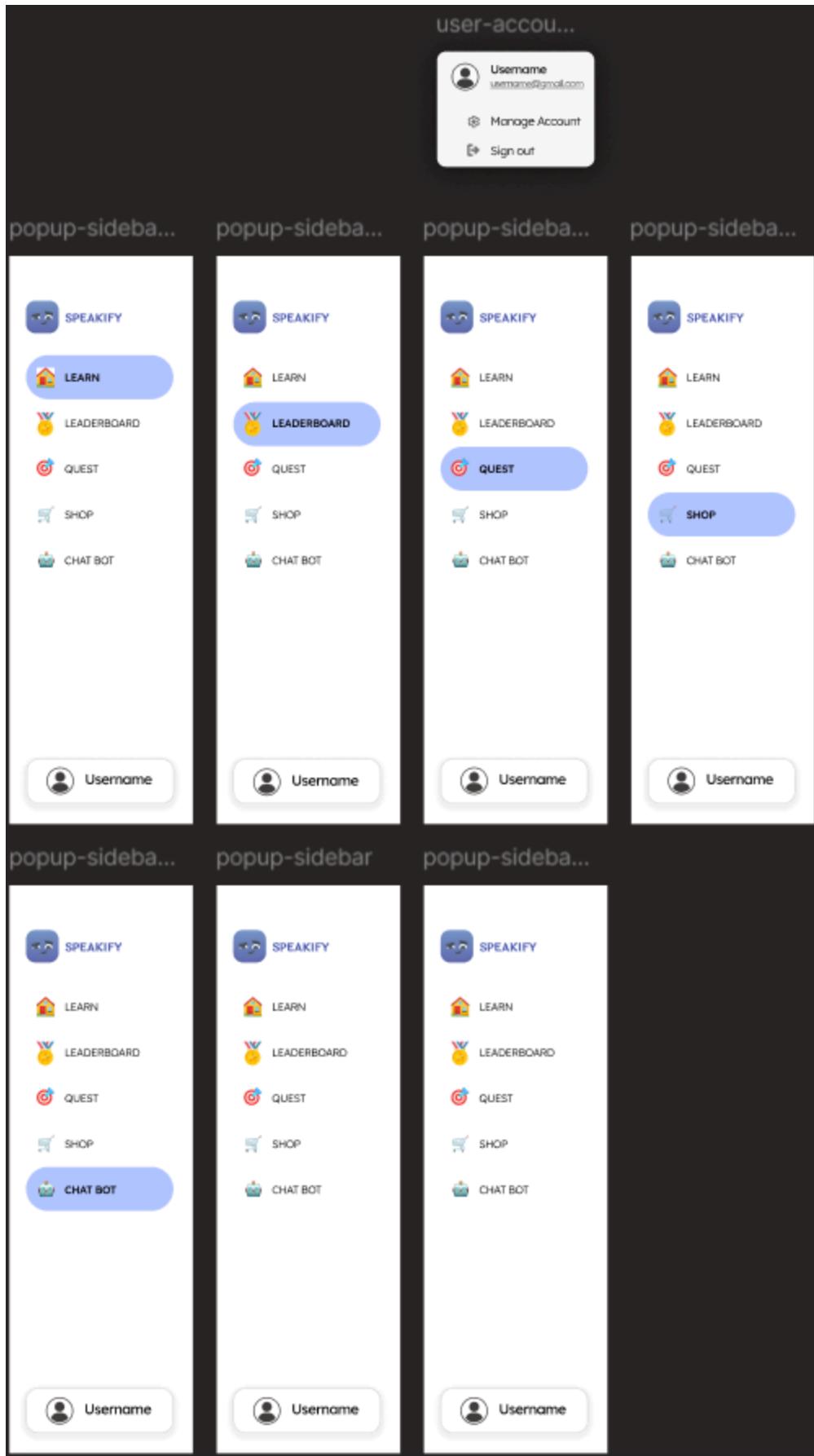
**CHECK**

activity-all-win	activity-win-all-win	activity-lose-all-win
activity-pre-win-all-win	activity-pre-lose-all-win	activity-pre-win-hover-all-win
activity-after-lose-all-win	activity-after-win-all-win	activity-pre-lose-hover-all-win

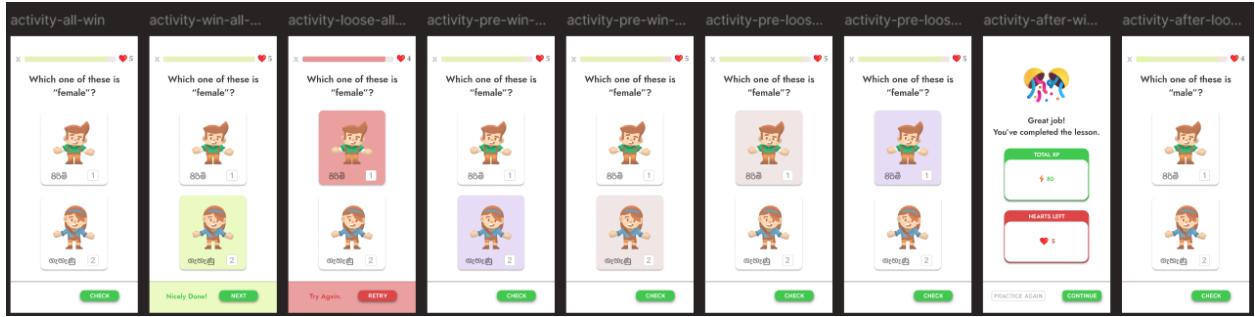
activity-practice	activity-win-practice	activity-loose-practice
		
activity-pre-win-practice	activity-pre-loose-practice	activity-pre-win-hover-practice
		
activity-after-loose-practice	activity-after-win-practice	activity-pre-loose-hover-practice
		

Mobile view,





1412 x 400

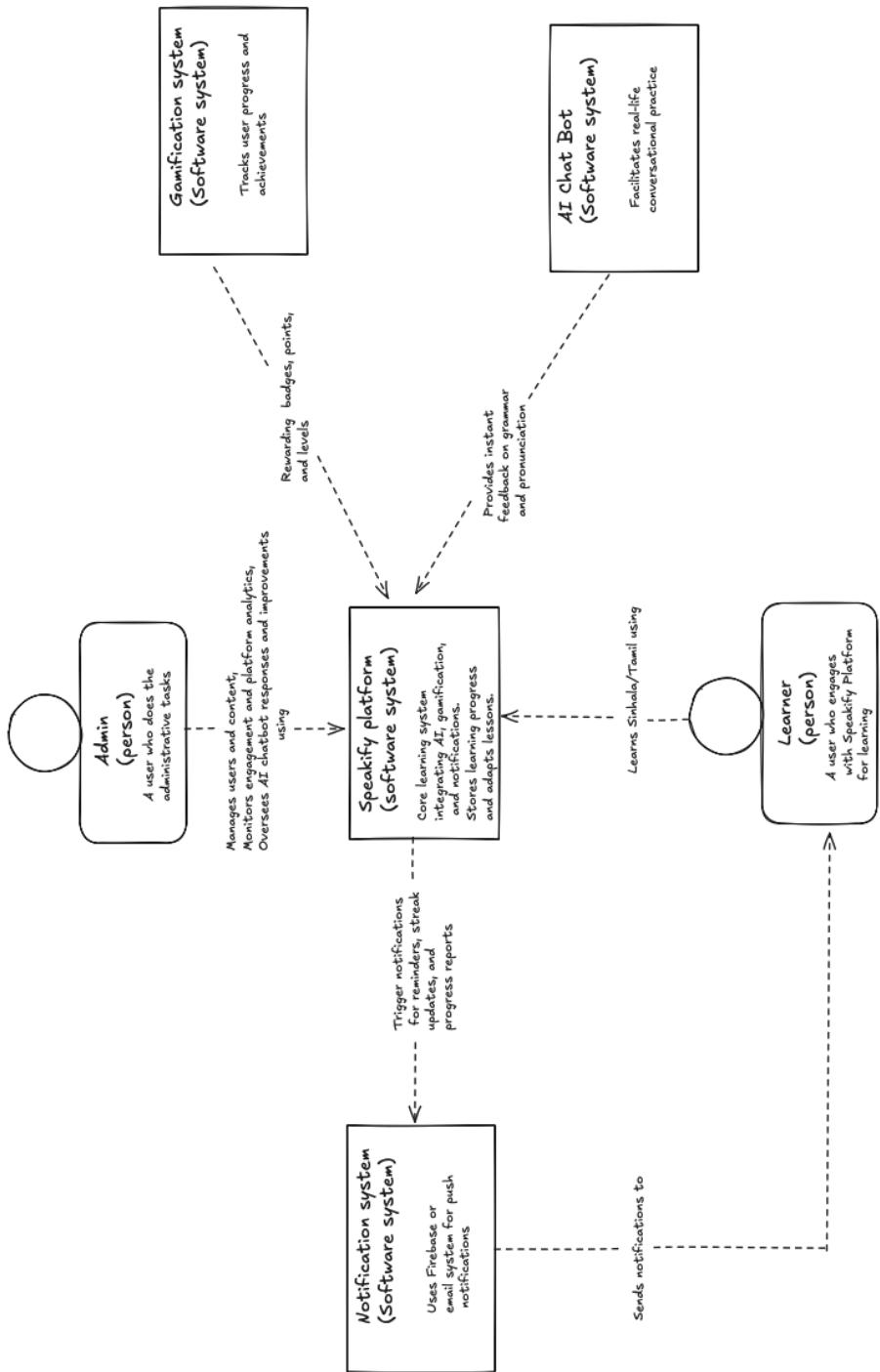


## 3.2 System Architecture

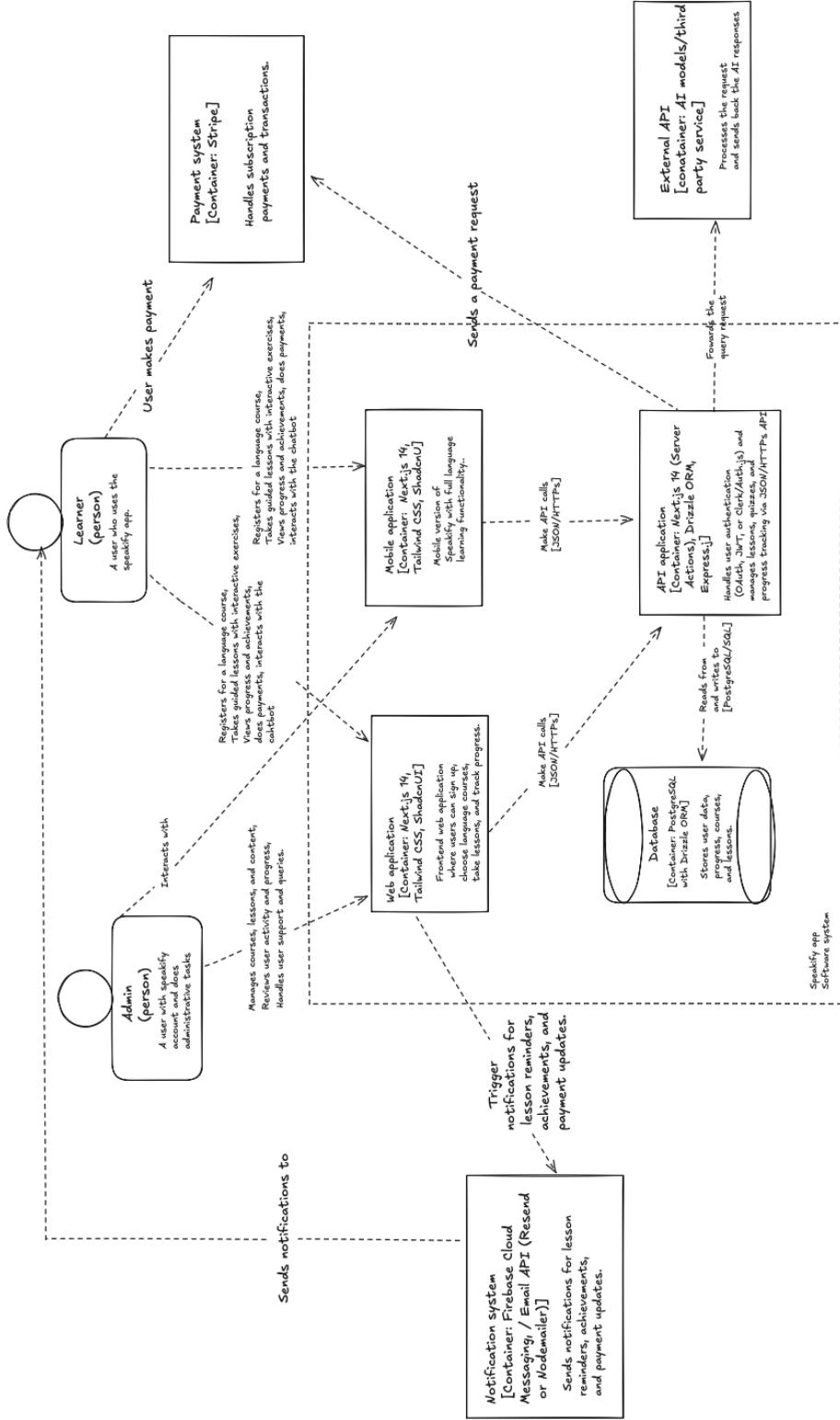
The system architecture for Speakify is designed to provide a scalable, maintainable, and efficient platform for interactive language learning. It follows the C4 model to ensure modularity, clear communication, and flexibility for future extensions. The architecture is organized into several layers, each serving a distinct purpose in the overall system. This approach ensures that Speakify can deliver personalized, AI-driven learning experiences while maintaining performance and adaptability as the platform grows. Below is the breakdown of the architecture, starting from high-level context to the individual components, followed by diagrams that visually represent the system's structure.

You can access the diagrams [here](#).

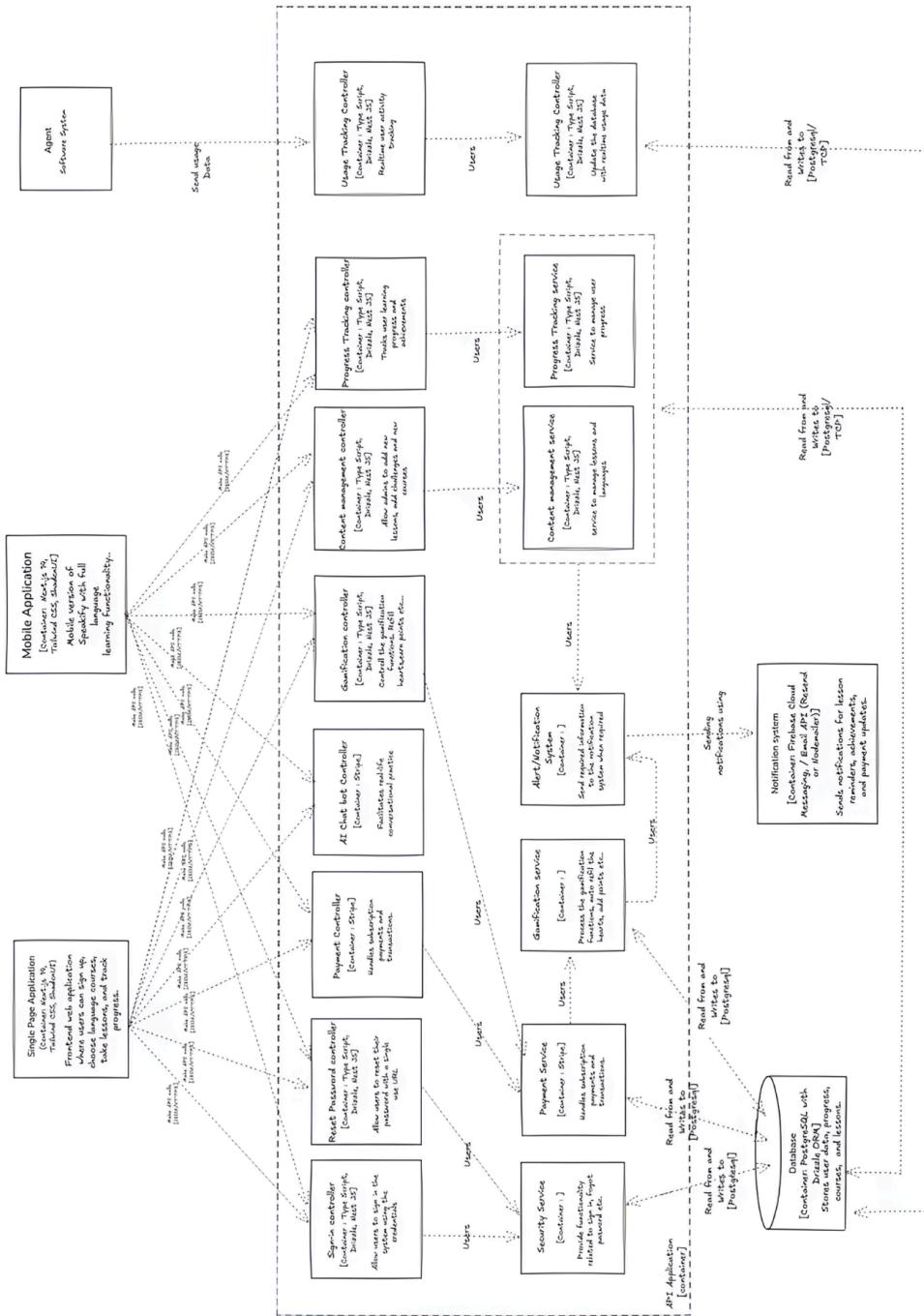
### 3.2.1 Context Diagram



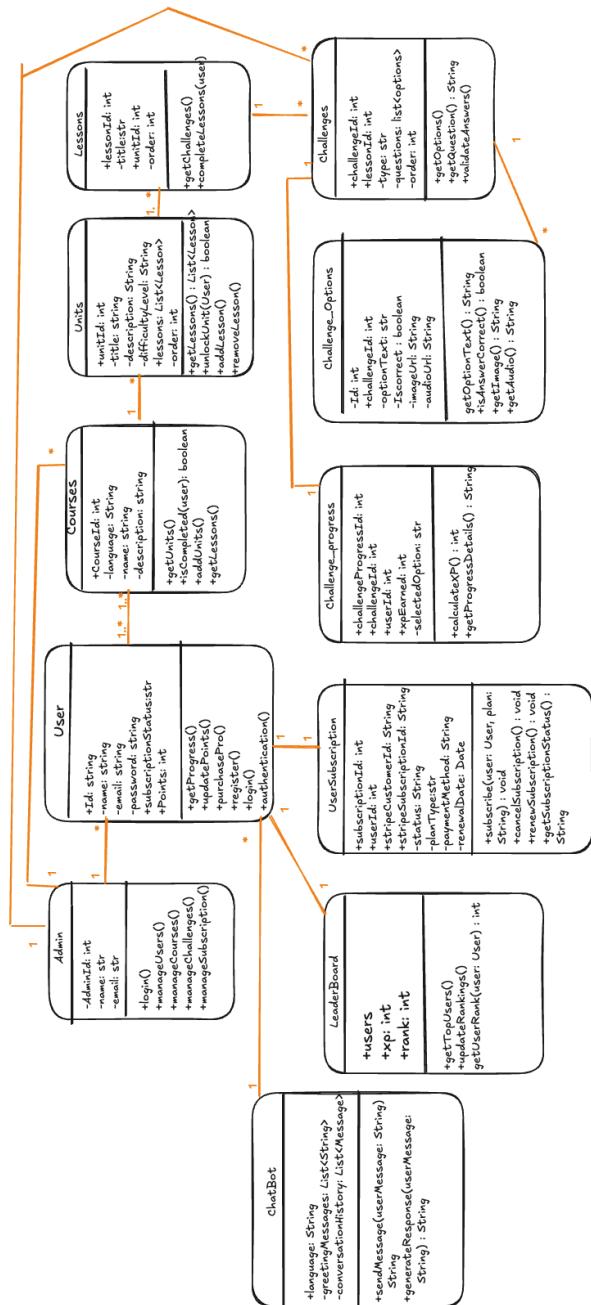
### 3.2.2 Container Diagram



### 3.2.3 Component Diagram



### 3.2.4 Class Diagram



### **3.2.5 System Architecture Overview**

The architecture of Speakify is designed around clear responsibilities distributed across multiple modules. The core components, such as User, Lesson, Progress, AIService, and Notification, represent distinct modules that manage key aspects of the language learning platform. The architecture of Speakify has the following characteristics:

- **Modular Design:**

The system is built to be modular, with each component responsible for a specific functionality, such as managing user profiles, delivering lessons, or providing real-time language assistance. This modularity ensures that the system remains scalable and easy to maintain as new features, languages, or learning modules are added.

- **Separation of Roles:**

The separation between Teacher and Student classes ensures that different functionalities are clearly defined. Teacher roles manage lesson content and progress tracking, while Student roles focus on interacting with lessons, quizzes, and receiving feedback. This clear separation of roles supports a more secure and user-friendly experience, allowing for role-based access control.

- **Data Collection and Notification:**

The DataCollection module is responsible for gathering data on student progress, quiz results, and interactions with lessons. This data is processed to offer personalized feedback and improve learning paths. The Notification module keeps users informed about important milestones such as new lessons, reminders, or progress updates, ensuring continuous engagement and motivation.

- **Adaptive Learning and AI Integration:**

The AIService module leverages Elevenlabs AI to provide personalized, real-time conversation practice and learning adjustments. This component ensures that the platform adapts to the learner's needs, offering tailored content and speech recognition that aligns with their current proficiency level.

- **Extensibility:**

The architecture is designed for future scalability. New languages, lesson types, or gamification features can be added seamlessly without disrupting existing functionality.

This extensibility is facilitated by the modular structure, which allows new components to be integrated easily while maintaining core platform stability.

- **Maintainability:**

Maintainability is a key principle, achieved by isolating functionalities into separate components. Changes to one module, such as updates to lesson content, do not affect other modules, like AI-driven learning. APIs are used to facilitate communication between components, ensuring loose coupling and easy updates.

Overall, this architecture enables Speakify to efficiently deliver personalized, engaging language learning experiences while remaining flexible and easy to extend as new features and improvements are introduced.

## 4. CONCLUSION

This chapter provides a comprehensive evaluation of the Speakify project, analyzing how well the system met its objectives, the quality of its usability, accessibility, reliability, and user-friendliness, as well as identifying limitations and potential areas for future improvements.

### 4.1 The Degree to which Project Objectives Were Met

The primary objective of Speakify was to create an interactive and adaptive language learning platform tailored to Sri Lankan languages, specifically Sinhala and Tamil. The system aimed to deliver a personalized, engaging learning experience using gamification, AI-driven features, and real-life conversation practice.

- **Personalized Learning Experience:** This objective was fully achieved by leveraging Elevenlabs AI, which adapts the learning path based on the user's progress, ensuring content is suitable for their current proficiency level.
- **Gamification and Engagement:** Speakify successfully integrated gamification features, such as progress tracking, rewards, and achievements, to keep learners motivated and engaged throughout their learning journey.
- **AI-Driven Features:** The integration of AI-driven tools, like real-time conversation practice, was successfully implemented, allowing learners to practice speaking skills in

an interactive environment, which addresses the challenges of learning a language with limited access to native speakers.

- **User Management and Notifications:** The platform provides easy-to-use user management features, allowing students to track their progress and receive notifications about milestones and new learning modules, enhancing overall user experience.
- **Cross-Platform Accessibility:** Speakify's web-based interface is accessible across multiple devices, ensuring learners can engage with the platform on smartphones, tablets, or desktops, increasing its reach and convenience.

In summary, the core objectives of the Speakify project were met effectively, laying a strong foundation for language learning in Sri Lanka.

## **4.2 Evaluation of Usability, Accessibility, Reliability, and Friendliness of the System**

- **Usability:**

Usability was a core consideration in the design of Speakify. The platform features an intuitive user interface, optimized based on user feedback, which minimizes the learning curve. Regular users and educators can easily navigate through lessons, quizzes, and progress tracking features.

- **Accessibility:**

The platform is designed with accessibility in mind, supporting multiple devices and ensuring compatibility across various screen sizes. The interface includes support for Sinhala, Tamil, and English, making it more accessible to a diverse user base in Sri Lanka.

- **Reliability:**

Speakify ensures reliable performance through real-time data synchronization between users' devices and the central database. This guarantees that progress data, lessons, and notifications are always up-to-date, even during periods of intermittent connectivity.

- **User-Friendliness:**

The system's user-friendly design caters to both learners and educators. The interface simplifies access to learning materials and progress tracking, while in-depth guides and

tooltips further support new users. The design strives to make language learning a seamless experience, even for those who are not tech-savvy.

#### **4.3 Limitations and Drawbacks**

Despite the successful implementation of Speakify, there are several limitations and areas for improvement:

- **Dependence on Internet Connectivity:**

As a web-based platform, Speakify requires a stable internet connection to function properly. In regions with poor internet connectivity, this may hinder the learning experience for users.

- **Limited Mobile Features:**

While the platform is accessible across devices, the mobile version of Speakify could be enhanced with more interactive features, such as push notifications for progress updates or rewards.

- **Language Adaptation:**

Although Speakify supports Sinhala, Tamil, and English, the platform could benefit from broader language support to cater to learners from different regions, expanding its usability.

- **Offline Access:**

Currently, offline functionality is limited, as the platform requires an internet connection for accessing most of its features. Enhancing offline access would be beneficial for learners in areas with unreliable internet.

#### **4.4 Future Improvements, Modifications, and Extensions**

To address the current limitations and further enhance Speakify, the following improvements and extensions can be considered:

- **Offline Learning Mode:**

Adding offline functionality to allow users to continue their lessons without internet connectivity would make the platform more accessible in areas with limited or unreliable internet.

- **Mobile App Enhancements:**

Developing a fully-featured mobile app with push notifications for progress reminders and new lesson availability would increase user engagement and make language learning more convenient on-the-go.

- **Additional Language Support:**

Expanding the language offerings beyond Sinhala and Tamil would help Speakify reach a larger audience in Sri Lanka and potentially in other regions. The system could also include dialect-specific lessons to cater to diverse language needs.

- **Integration of Speech Recognition:**

Enhancing AI-driven features by incorporating speech recognition tools for pronunciation feedback would add a valuable layer of interactivity, helping learners improve their speaking skills more effectively.

- **Scalability for Larger Communities:**

As the platform grows, scaling Speakify to accommodate larger cohorts of learners, particularly in educational institutions, could involve adding support for multi-user classrooms, group challenges, and performance-based assessments.

By addressing these limitations and incorporating suggested improvements, Speakify can evolve into a more comprehensive and robust language learning platform, better serving the needs of its users and continuing to make language learning more accessible and engaging.

## **APPENDICES**

### **Appendix A: Figma Design Link**

The user interface (UI) prototypes for the Speakify platform were designed using Figma. The Figma link below provides access to the complete design mockup:

[Speakify UI Design on Figma](#)

### **Appendix B: Survey Link**

To gather insights and understand user needs, a survey was conducted among university students in Sri Lanka. The survey focused on current language learning practices, the challenges faced in learning Sinhala and Tamil, and features that would make the learning experience more engaging. The survey questions can be accessed through the link below:

[Speakify User Survey](#)