

**CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

PronounceIT: English Vocabulary/Pronunciation E-Learning System for
Kindergarten

Change History

Date	Version	Description	Authors
3/12/2025	1.00	Initial Draft	Kho, Antonio A. Alico, Christian Barry R. Laude, Raymund Christian A. Saceda, Rhandulf Q. Pobadora, Rommel John L.

Table of Contents

Change History	2
Table of Contents	3
1. Introduction	4
1.1. Purpose	4
1.2. Scope	4
1.3. Definitions, Acronyms and Abbreviations	5
1.4. References	5
2. Overall Description	6
2.1. Product perspective	6
2.2. User characteristics	6
2.4. Constraints	7
2.5. Assumptions and dependencies	7
3. Specific Requirements	9
3.1. External interface requirements	9
3.1.1. Hardware interfaces	9
3.1.2. Software interfaces	9
3.1.3. Communications interfaces	10
3.2. Functional requirements	11
Module 1	Error! Bookmark not defined.
Module 2	Error! Bookmark not defined.
3.4 Non-functional requirements	61
Performance	62
Security	62
Reliability	62

1. Introduction

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a detailed description of the **PronounceIT** software, an AI-driven English Vocabulary and Pronunciation E-Learning System designed for kindergarten students. This document defines the system's functionalities, constraints, and user requirements. It will serve as a comprehensive reference for the developers, project managers, and stakeholders to ensure the development of a product that meets customer needs.

In short, this SRS provides a comprehensive overview of the **PronounceIT** software, including its objectives, parameters, and intended functionalities. It details the system's target users—students, parents and teachers—and outlines the user interface, hardware, and software requirements. The document defines the perspectives of all stakeholders, including the development team, and how they envision the product functioning. Moreover, it serves as a guide for developers and designers throughout the software development lifecycle (SDLC), ensuring a streamlined process from conception to deployment.

1.2. Scope

PronounceIT is a web and mobile application designed to assist kindergarten students in improving their pronunciation skills and expanding their English vocabulary. The system leverages AI-driven voice recognition technology to provide immediate, accurate feedback on pronunciation attempts, making language learning more engaging and effective for young learners. The system will utilize AI voice recognition to evaluate pronunciation accuracy and provide real-time feedback. The system provides audio assistance with correct pronunciation from trusted sources like Merriam-Webster. It includes displaying vocabulary words visually with supporting images to reinforce learning. The system also incorporates gamification elements to maintain young learners' interest and motivation and provide personalized feedback and progress tracking for children, parents, and educators.

The system will not:

- Support languages other than English in the initial release.
- Provide advanced grammar or sentence structure training.
- Replace traditional classroom instruction but rather supplement it.

The scope of this SRS focuses on defining the key features of **PronounceIT** for both user and administrative users, outlining the essential software requirements. This document also serves as a guide for developers, providing a detailed foundation for the system's implementation. While primarily targeting the **PronounceIT** project, this SRS can be used as a model for developing similar e-learning systems in other languages.

Additionally, this SRS provides flexibility in its application, allowing it to assist in the selection of

software solutions—whether developed in-house or commercially. Although it does not prescribe a specific methodology or toolset for preparing an SRS, it establishes a standard that can be adapted for project-specific needs.

1.3. Definitions, Acronyms and Abbreviations

PronounceIT	The name of the web and mobile application.
Admin	An administrator with the ability to manage users, content, category and lessons.
SRS	Software Requirements Specification
AI	Artificial Intelligence, used for voice recognition and pronunciation feedback.
E-Learning	Electronic learning, using digital platforms for educational purposes.
Speech Recognition	AI technology that analyzes spoken words and provides feedback.
Gamification	The application of game-design elements in non-game contexts to enhance user engagement.
COPPA	Children's Online Privacy Protection Act, a data protection regulation ensuring child safety online.

1.4. References

Component	Name	Description
Problem Statement	PronounceIT Problem Statement	This document defines the core issue that PronounceIT aims to solve, outlining the challenges related to pronunciation learning and how the software addresses them.
Software Project Proposal	PronounceIT Software Project Proposal	This proposal provides an overview of the PronounceIT project, detailing its objectives, features, target audience, and the technologies used for development.

2. Overall Description

2.1. Product perspective

The pronunciation learning app is an **interactive educational tool** designed to help young children (ages 3 to 5) develop clear speech by improving their pronunciation skills. It leverages **speech recognition technology** to provide immediate feedback, helping children recognize and correct pronunciation errors. The app is designed as a **standalone mobile or tablet application** but can also be integrated into **early childhood education programs**. It incorporates **gamification elements** such as rewards, animations, and interactive exercises to keep children engaged.

The application is divided into four core modules: User Authentication, Profile Management, Gameplay, and Admin Management. These modules interact seamlessly within the application to provide a cohesive learning experience for the user while allowing administrators to monitor and manage content and progress effectively.

Overall System Architecture

The system is composed of the following main modules:

- **User Authentication and Authorization:** Manages user sign-up, login, and session handling, ensuring secure access to the application. It supports both regular users and admin roles, with different levels of access control.
- **User Profile Management:** Allows users to manage personal information, track progress, and update their profile as needed.
- **Gameplay:** The core of the learning experience. Users interact with the system by selecting a category (represented by a letter, e.g., A, B, C) and then choosing a lesson within that category. Each lesson follows a specific theme related to the chosen letter (e.g., "Lesson 1: Animals"). Users attempt to pronounce words correctly, receiving feedback based on speech recognition accuracy, reinforced with visual and audio cues.

2.2. User characteristics

1. User (Parent and Child)

- **Role:** Engage with gamified pronunciation exercises, monitor progress, select category and lessons, track performance and progress.
- **Privileges:** Access lessons; no configuration rights.
- **Needs:** Simple, icon-driven interface with voice-guided navigation.

2. Teacher (Admin)

- Role: Add lesson, word management (such as add, edit and delete), track user performance, and group users.
- Privileges: Access analytics dashboards and modify lessons and word content.

2.3 Constraints

The following constraints must be considered during development:

- **Regulatory Policies** – Must comply with **COPPA (Children's Online Privacy Protection Act)** and **GDPR-K (General Data Protection Regulation for Kids)** to ensure safe data handling.
- **Hardware Limitations** – The app must run on **low-end mobile devices** with limited processing power and memory.
- **Speech Recognition Accuracy** – Variability in children's speech patterns and background noise can impact pronunciation analysis.
- **Interfaces to Other Applications** – May need to **integrate with educational platforms or school management systems** for data sharing.
- **Parallel Operation** – Should allow **multiple child profiles** on a single device, enabling shared use in classrooms or households.
- **Audit Functions** – Parents and educators should be able to **review pronunciation history** and performance reports.
- **Control Functions** – Must include parental control settings for **screen time limits** and **content accessibility**.
- **Reliability Requirements** – Speech recognition must function with at least **90% accuracy** to provide meaningful feedback.
- **Criticality of the Application** – Although **not life-critical**, it significantly impacts **early childhood language development** and should ensure **minimal errors in pronunciation feedback**.
- **Safety & Security Considerations** – No **personal identifying information (PII)** should be stored, and **voice recordings must not be shared externally**.

2.4 Assumptions and dependencies

Assumptions:

- Children will use the app **with supervision from parents or educators**.
- The **device has a functioning microphone** for voice input.
- Users have access to **basic internet connectivity** for software updates and cloud-based analytics.
- Speech recognition **algorithms will continuously improve** with usage data.

Dependencies:

- **Operating System** – The app will be developed for **Android (version 10 and above)**.
- **Speech Recognition API** – The system relies on **Google's Speech-to-Text API**.
- **Regular Updates** – The app will require **ongoing updates** to refine pronunciation feedback and introduce new exercises.
- **Third-Party Libraries** – Uses **text-to-speech (TTS)** and **gamification frameworks** to enhance learning engagement.

3. Specific Requirements

3.1. External interface requirements

3.1.1. Hardware interfaces

1. **Supported Devices:**

The system will support the following devices:

- **Tablets:** Android devices with touchscreen capabilities.
- **Desktop/Laptop Computers:** Windows systems with a minimum screen resolution of 1024x768.
- **Smartphones:** Android devices with touchscreen capabilities.

2. **Configuration Characteristics:**

- **Ports:** The system will use USB ports/Audio Jack for external microphones and headphones.
- **Instruction Sets:** The system will support ARM and x86/x64 instruction sets for compatibility with mobile and desktop devices.

3. **Device Support:**

- **Tablets/Smartphones:** The system will support touchscreen input and accelerometer-based interactions (e.g., tilting/rotating the device for certain activities).
- **Desktop/Laptops:** The system will support keyboard and mouse input for navigation and interaction.

3.1.2. Software interfaces

This section describes the software components and their interaction with the system.

1. **Operating System Compatibility:**

The system will be compatible with the following operating systems:

- **Mobile:** Android (version 10.0 and above).
- **Desktop:** Windows (version 10 and above).

2. **Required Software Products:**

- **Web Browser:** The system will support modern web browsers (e.g., Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) for web-based access.
- **Mobile App:** The system will have mobile applications for Android, developed using Android Studio.

- **Backend System:** The backend will be developed using **Spring Boot (Java)** for handling user authentication, data processing, and API integration. It ensures scalability, security, and efficient request handling.
- **Database Management System:** The system will use a relational database to store user data, progress, and learning materials.
- **Audio Processing Library:** The system will be integrated with an audio processing library (e.g., Web Audio API for web) for pronunciation analysis.

3. Interfaces with Other Systems:

- **Cloud Storage:** The system will use cloud storage (e.g., AWS S3, Google Cloud Storage) for storing multimedia content (e.g., audio, video, images).

4. APIs:

- The system will provide RESTful APIs for third-party integrations, such as:
 - **Authentication:** Integration with OAuth 2.0 for secure login (e.g., Google, Facebook).
 - **Analytics:** Integration with analytics platforms (e.g., Google Analytics) to track user engagement and learning progress.
 - **Voice Recognition Options:** Integration with audio processing library (e.g. Web Audio API on Web and Android APIs on mobile) to capture and process voice input in real time.
 - **Text-to-Speech:** Integration with text-to-speech engines (e.g. Google Cloud Text-to-Speech, Amazon Polly, Microsoft Azure Text-to-Speech) to deliver audio pronunciation, feedback, and instructions.

3.1.3. Communications interfaces

This section describes the communication protocols and network requirements for the system.

1. Network Protocols:

The system will use standard network protocols for communication:

- **HTTP/HTTPS:** For secure web-based communication between the client and server.
- **WebSocket:** For real-time communication during interactive activities (e.g., live pronunciation feedback).
- **TCP/IP:** For reliable data transmission between the system and external services.

2. Cloud Communication:

The system will communicate with cloud-based servers for:

- **Content Delivery:** Fetching learning materials (e.g., videos, audio files) from cloud storage.
- **Real-Time Feedback:** Sending pronunciation data to the server for analysis and receiving feedback in real-time.

3. Security Protocols:

The system will use the following security measures:

- **SSL/TLS encryption:** For secure communication between the client and server.
- **Data Encryption:** For encrypting sensitive user data (e.g., passwords, payment information) stored in the database.
- **Firewall:** To protect the system from unauthorized access.

3.2. Functional requirements

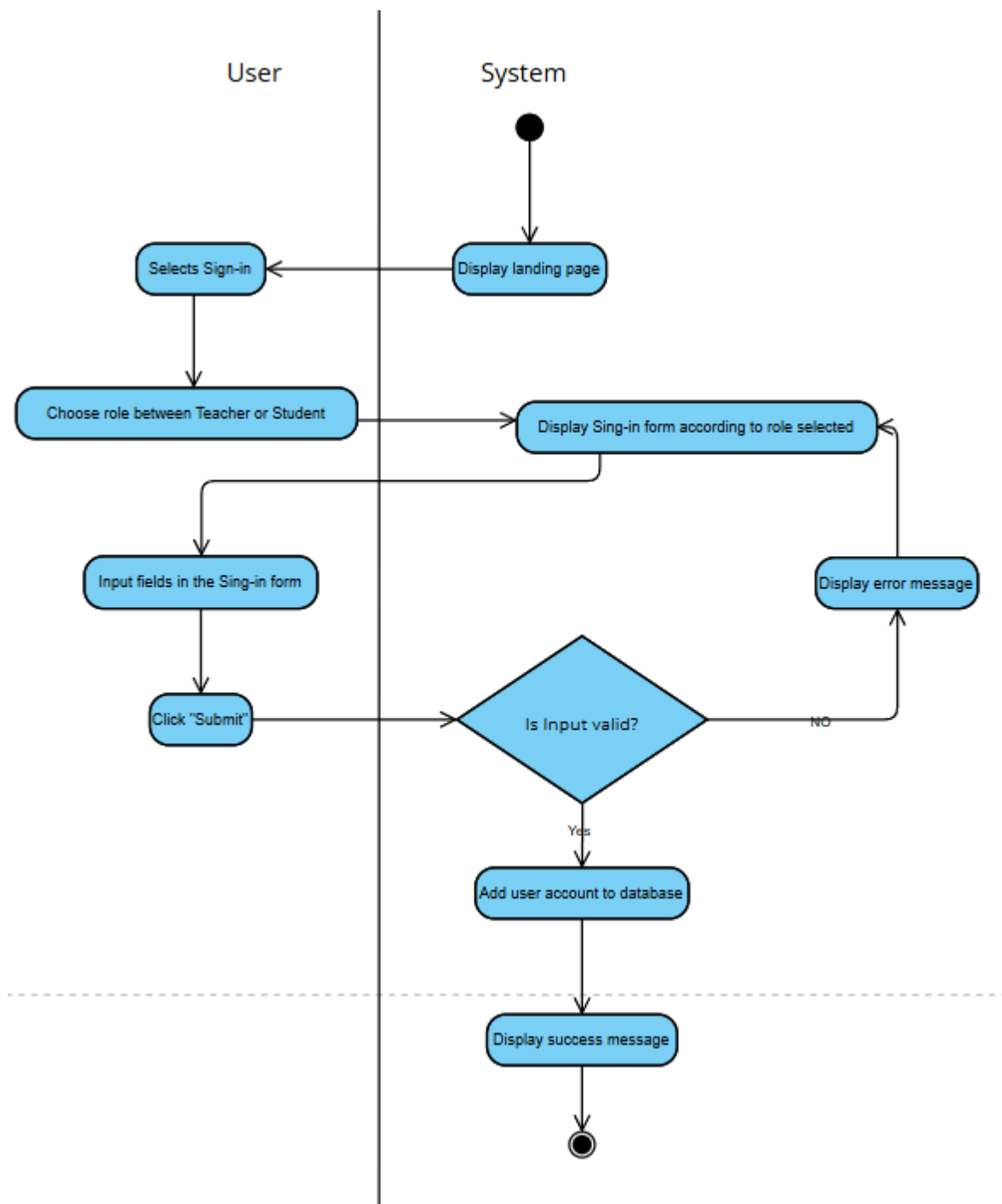
3.2.1 Module 1: User Authentication and Authorization

3.2.1.1 User Sign-In

Use Case	User Sign-In
Description	A new user creates an account by providing the required information.
Actors	User, System
Module	User Authentication and Authorization
Preconditions	1. The user is on the landing page.
Basic Sequence	<ol style="list-style-type: none">1. The user clicks on the sign in button.2. The System displays a selection screen with options: "Teacher," and "Student".3. The User clicks on their respective role.4. The user is on the sign-in page.5. User fills out required fields of the sign-in form (username, password, email).6. User clicks on the Submit button.7. The system validates the input data.8. The system creates a new user record in the database.

	9. System displays a success message ("Account successfully created").
Exceptions	<ol style="list-style-type: none">1. Invalid Input: If required fields are missing or invalid, the system displays an error message and prompts the user to correct the input.2. Username Already Exists: If the chosen username is taken, the system prompts the user to select a different username.
Postconditions	<ol style="list-style-type: none">1. The user's account is created and stored in the database.2. The user can now log in to the system.

Activity Diagram

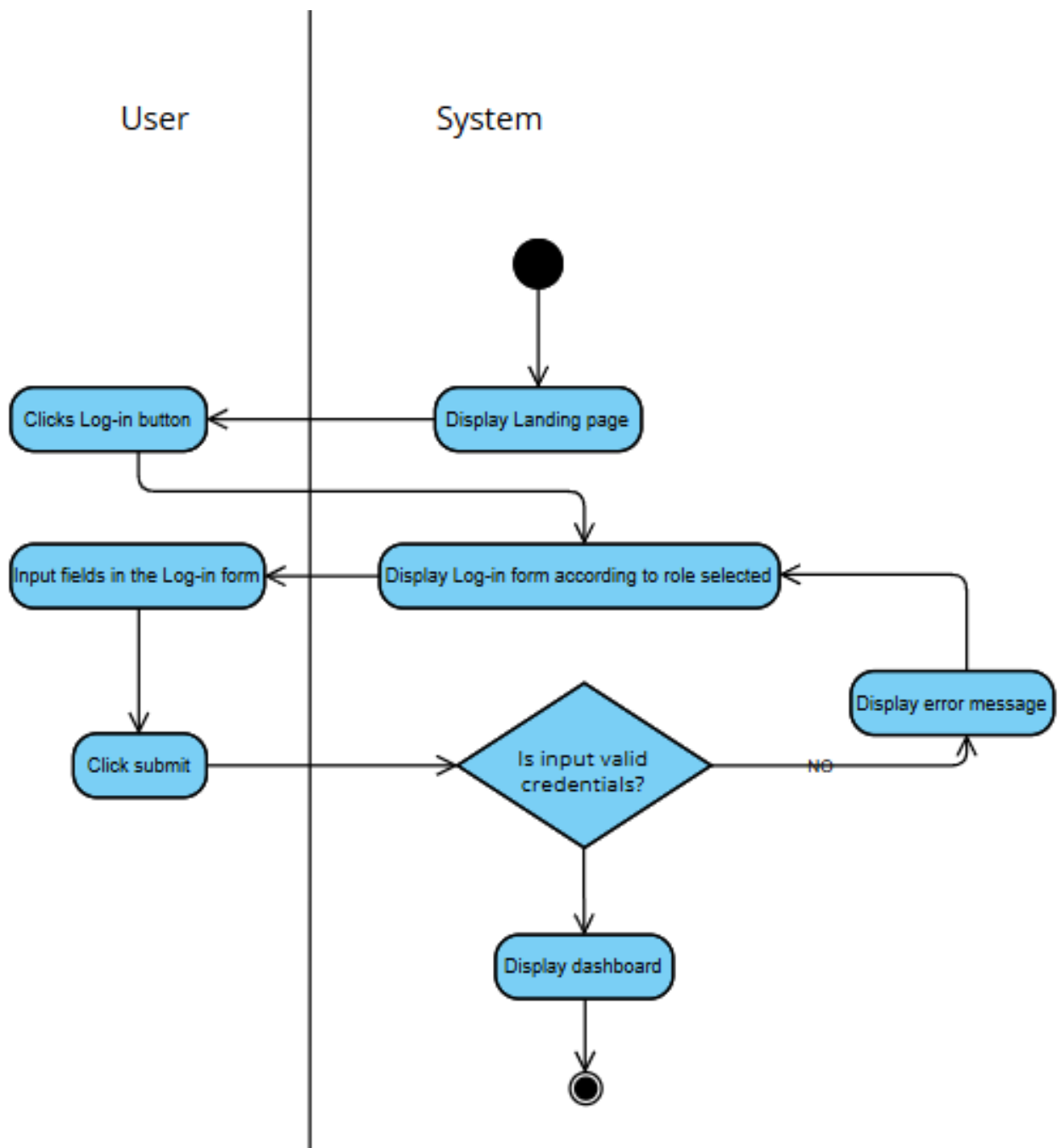


3.2.1.2 User Login

Use Case	User Login
----------	------------

Description	A registered user logs into the system using valid credentials.
Actors	User, System
Module	User Authentication and Authorization
Preconditions	<ol style="list-style-type: none">1. User has an account.2. The user is on the landing page.
Basic Sequence	<ol style="list-style-type: none">1. User clicks the login button.2. The system displays a selection screen with Log In options: "Teacher" and "Student."3. Users enter their username and password.4. User clicks login button.5. The system verifies credentials against stored data.6. The system grants access and redirects the user to the main dashboard.
Exceptions	<ol style="list-style-type: none">1. Invalid Credentials: If the username or password is incorrect, the system displays an error message and remains on the login page.
Postconditions	<ol style="list-style-type: none">1. Users are authenticated and can access system features based on their role (User or Admin).

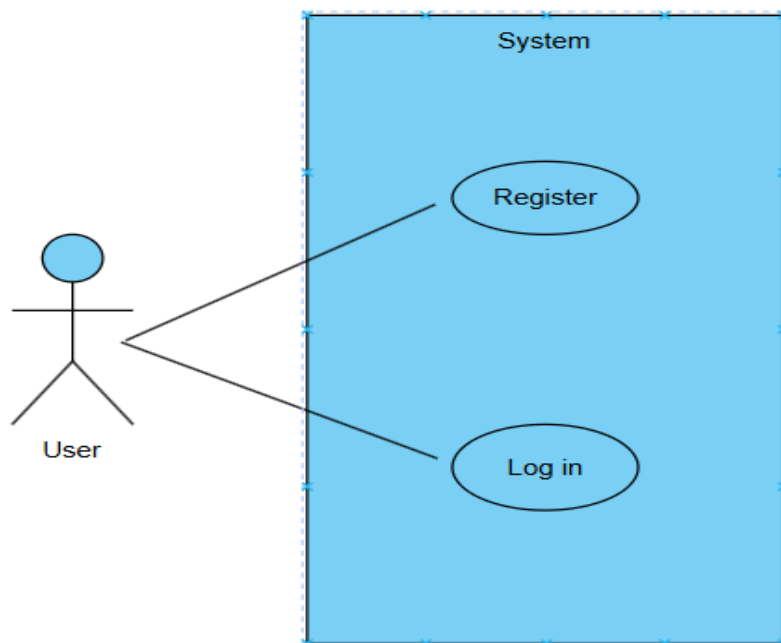
Activity Diagram



Wireframe

The wireframe shows a dark-themed interface with two main sections: 'User Login' and 'User Registration'. The 'User Login' section features the 'PronounceIT' logo, input fields for 'username' and 'password', a 'login' button, and a 'signup' button separated by an 'or' label. The 'User Registration' section includes input fields for 'username', 'email', 'password', and 'confirm password', along with a 'submit' button.

User Case Diagram

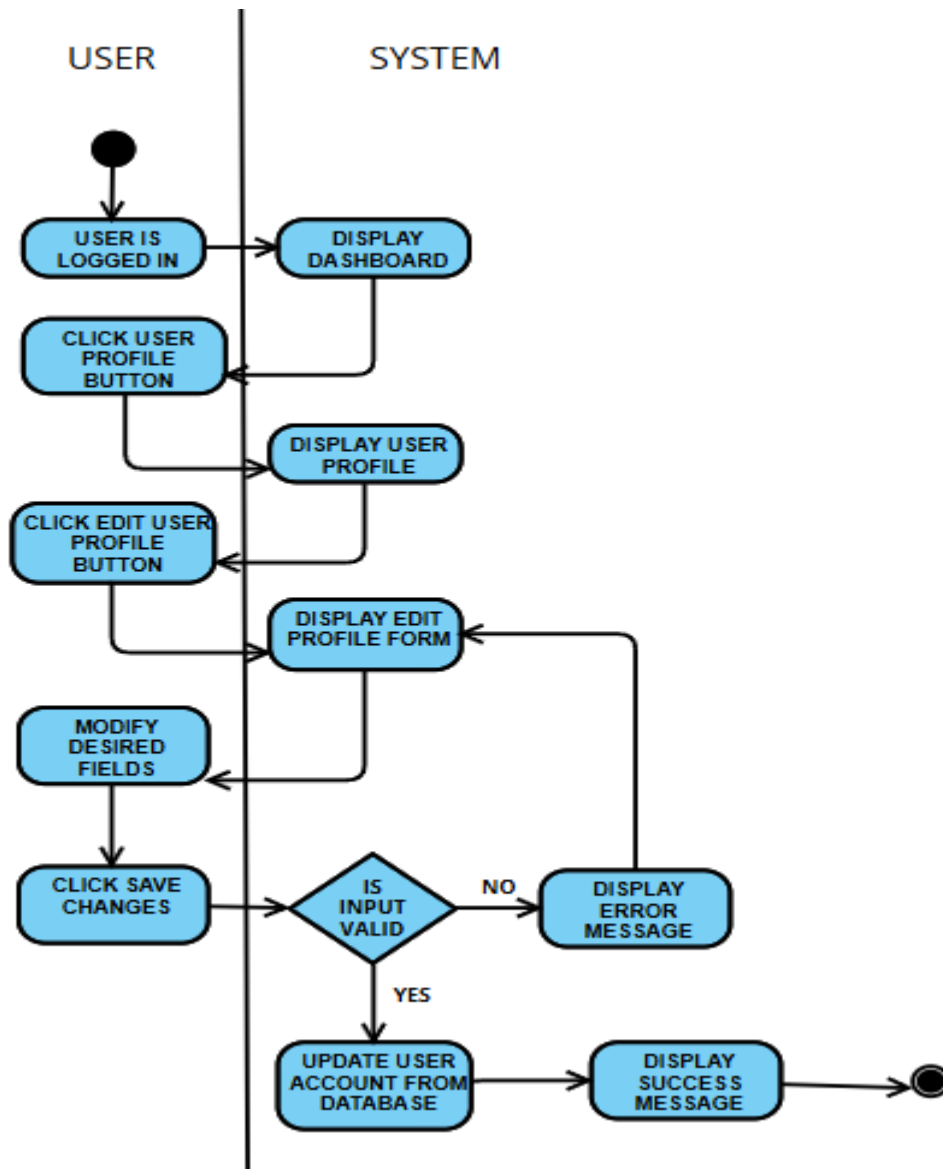


3.2.2 Module 2: User Profile Management

3.2.2.1 Edit User Profile

Use Case	Edit User Profile
Description	A user updates their personal information (eg. username, email, password).
Actors	User, System
Module	User Profile Management
Preconditions	<ol style="list-style-type: none">1. User is logged in.2. The user is on home page.
Basic Sequence	<ol style="list-style-type: none">1. The user clicks to profile to access the user profile page.2. The user clicks edit profile button.3. User modifies desired fields (e.g. Name, email, password).4. User clicks save.5. The system validates changes (e.g. Password format).6. The system updates the user record in the database.7. System displays a success message ("Profile updated successfully").
Exceptions	<ol style="list-style-type: none">1. Invalid input: If new data fails validation (e.g. Invalid email format), the system prompts the user to correct the input.
Postconditions	<ol style="list-style-type: none">1. Updated information is saved in the database.2. Changes reflect immediately on the user's profile.

- Activity Diagram

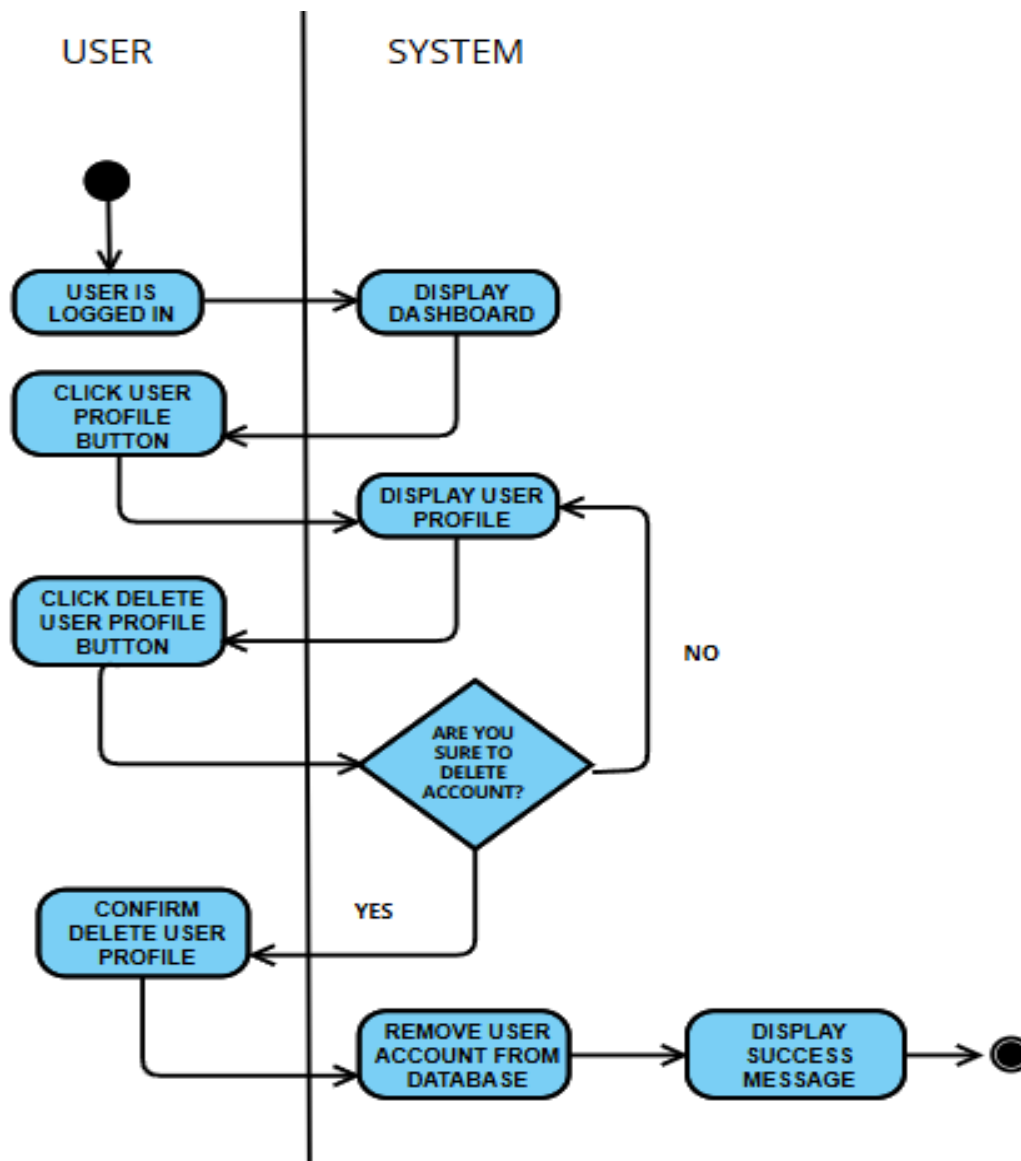


3.2.2.2 Delete User Profile

Use Case	Delete User Profile
Description	A user permanently deletes their account from the system.
Actors	User, System
Module	User Profile Management
Preconditions	1. User is logged in. 2. The user is on home page.

Basic Sequence	<ol style="list-style-type: none">1. The user clicks to profile to access the user profile page.2. The user clicks on Delete Account.3. The system asks for confirmation ("Are you sure you want to delete your account?").4. The user confirms deleting the account.5. The system deletes the user record and associated data from the database.6. The system displays a success message that the account has been deleted.7. The system redirects to the landing page.
Exceptions	<ol style="list-style-type: none">1. Cancellation: If the user cancels the deletion, no changes are made.2. System Error: If a database error occurs, the system notifies the user and instructs them to retry later.
Postconditions	<ol style="list-style-type: none">1. The user's account is removed from the database.2. The user can no longer log in with the deleted account.

Activity Diagram

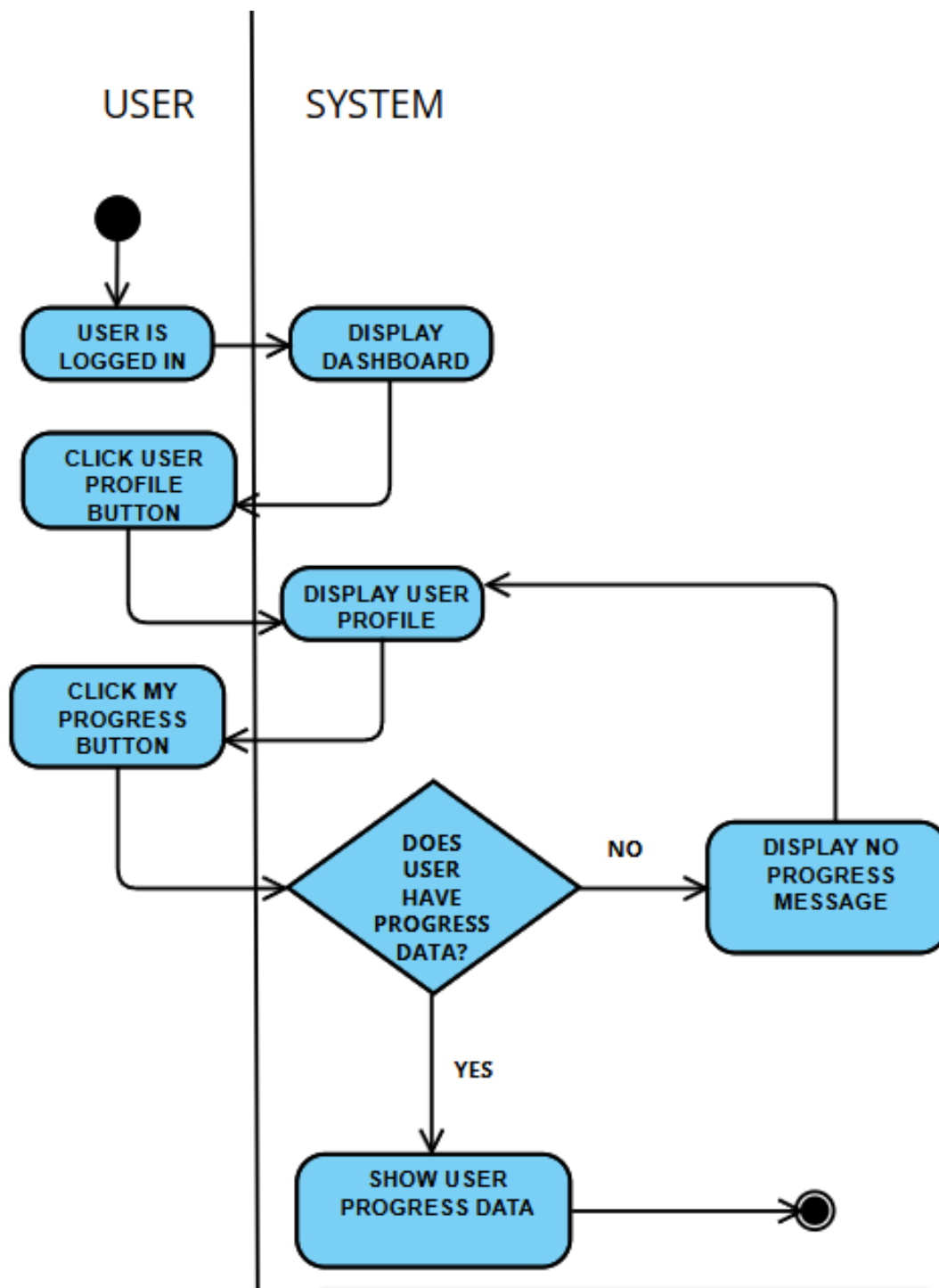


3.2.2.3 View User Progress and Scores

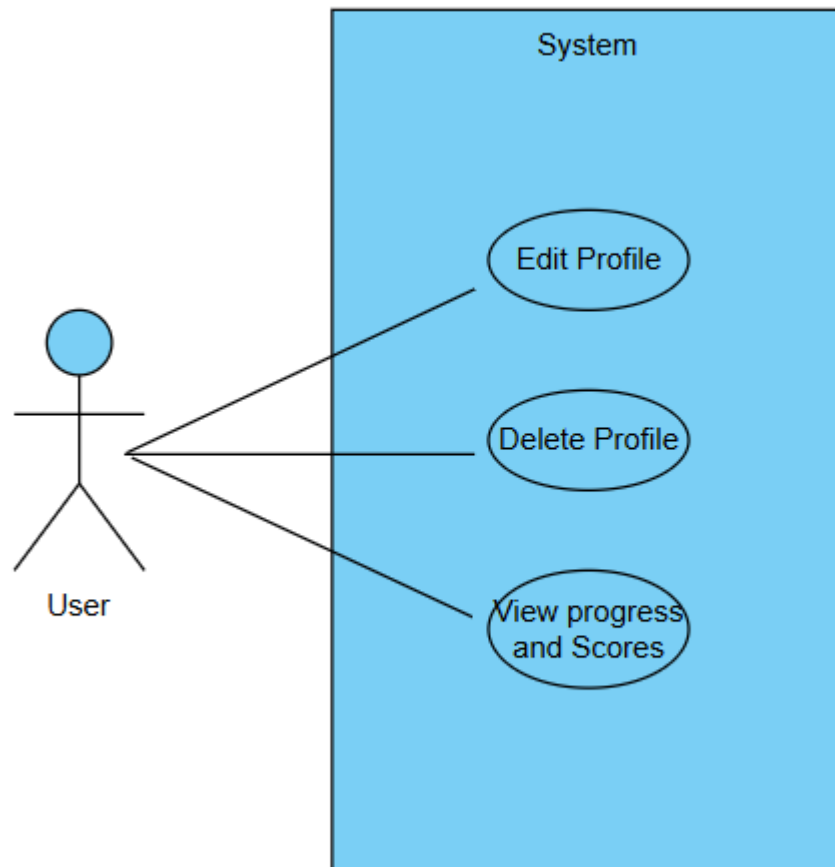
Use Case	View User Progress and Scores
Description	A user checks their progress, including scores across categories and lessons, and identifies words they struggle to pronounce.
Actors	User, System
Module	User Pronunciation

Preconditions	<ol style="list-style-type: none">1. User is logged in.2. The user is on the home page.3. User has completed at least one exercise session.
Basic Sequence	<ol style="list-style-type: none">1. User navigates to Progress / My Scores section.2. The system retrieves user's past scores from the database.3. The system displays a breakdown by category, by lesson, and highlights frequently missed words.4. The user can select a lesson and category to review
Exceptions	<ol style="list-style-type: none">1. No Completed Exercises: If the user has The system displays a breakdown by category, by lesson, and highlights frequently missed words. no completed lessons, the system shows a message indicating no data is available.2. Database Error: If retrieving scores fails, the system notifies the user to retry later.
Postconditions	<ol style="list-style-type: none">1. The user gains insight into their progress, strengths, and areas for improvement.2. Data is used to plan further practice or review specific words.

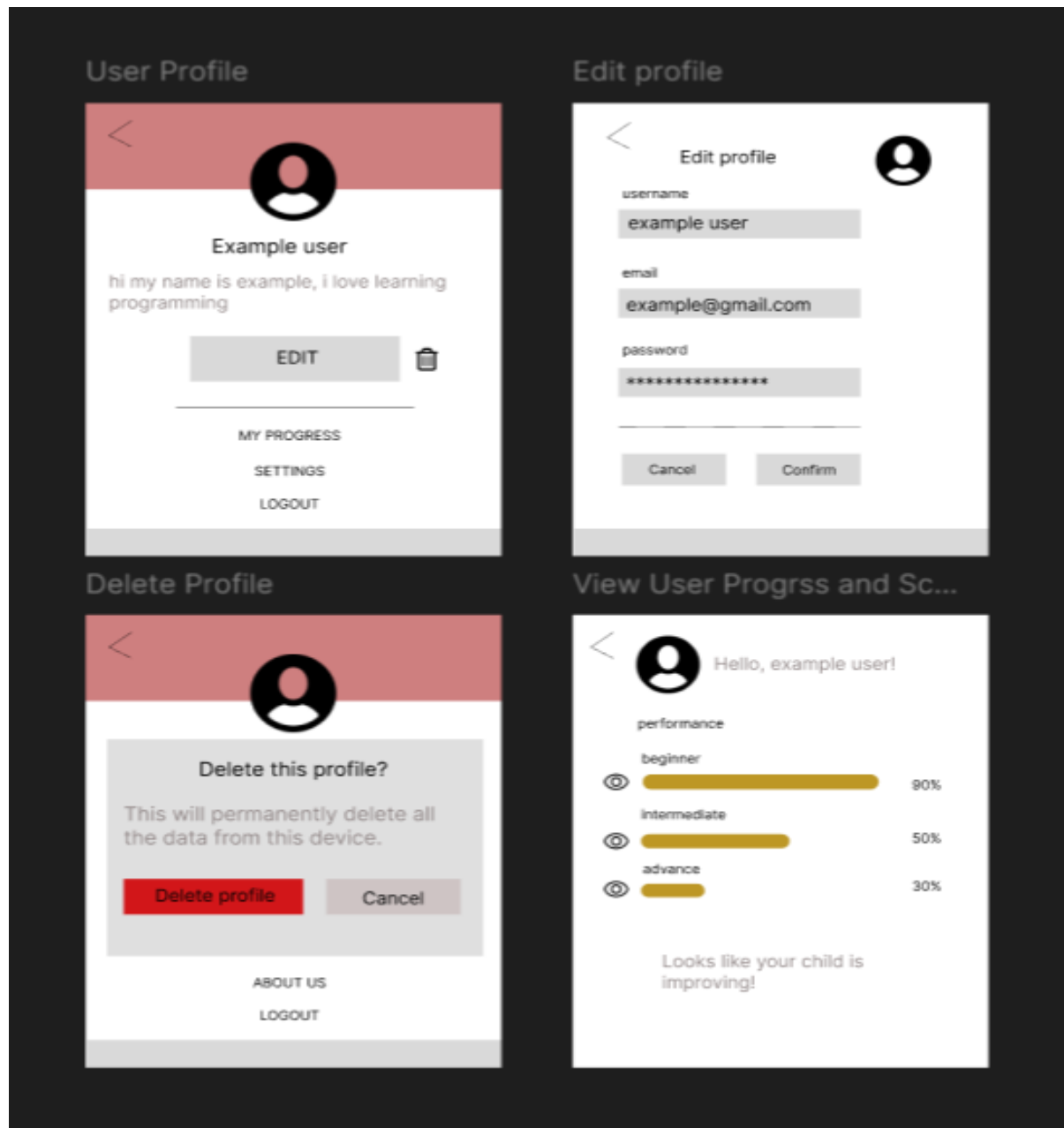
Activity Diagram



Use Case Diagram



Wireframe

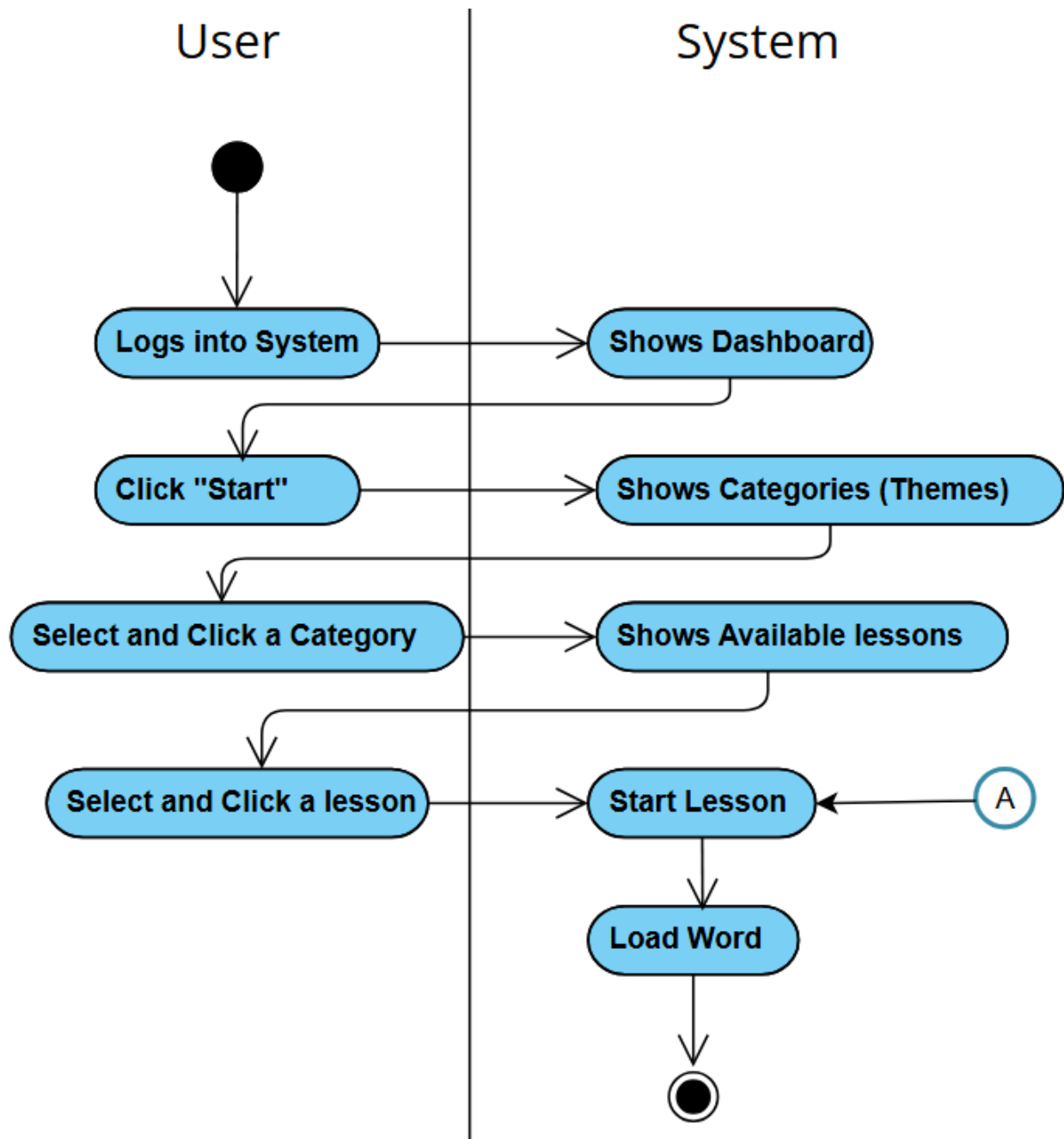


3.2.3 Module 3: Gameplay

3.2.3.1 Select Category and Lesson

Use Case	Select Category & Lesson
Description	A user chooses a category (e.g., Animals, Plants, Food) and a corresponding lesson (A, B, C, etc.) within that category to start practicing pronunciation. Locked lessons within a category must be completed in sequence.
Actors	User, System
Module	Gameplay
Preconditions	<ol style="list-style-type: none">1. User is logged in2. The system has categories (themes) and multiple lessons (letters) configured.3. Lessons beyond the first are locked until previous lessons are completed.
Basic Sequence	<ol style="list-style-type: none">1. User clicks the play button.2. User selects a category (e.g., "Animals").3. The system displays available lessons (letters) under the chosen category.4. User selects an active lesson (e.g., "Lesson A: Words that are Animals starting with the letter A").5. System loads the selected lesson's words or exercises.
Exceptions	<ol style="list-style-type: none">1. No Unlocked Lessons: If all lessons in a category are locked, the system informs the user they must complete previous lessons first.2. Configuration Error: If the system cannot load lessons, an error message is displayed, and the user is asked to retry later.
Postconditions	<ol style="list-style-type: none">1. The user's chosen category and lesson are set for the current exercise session.2. The user is ready to begin practicing words at that lesson.

Activity Diagram

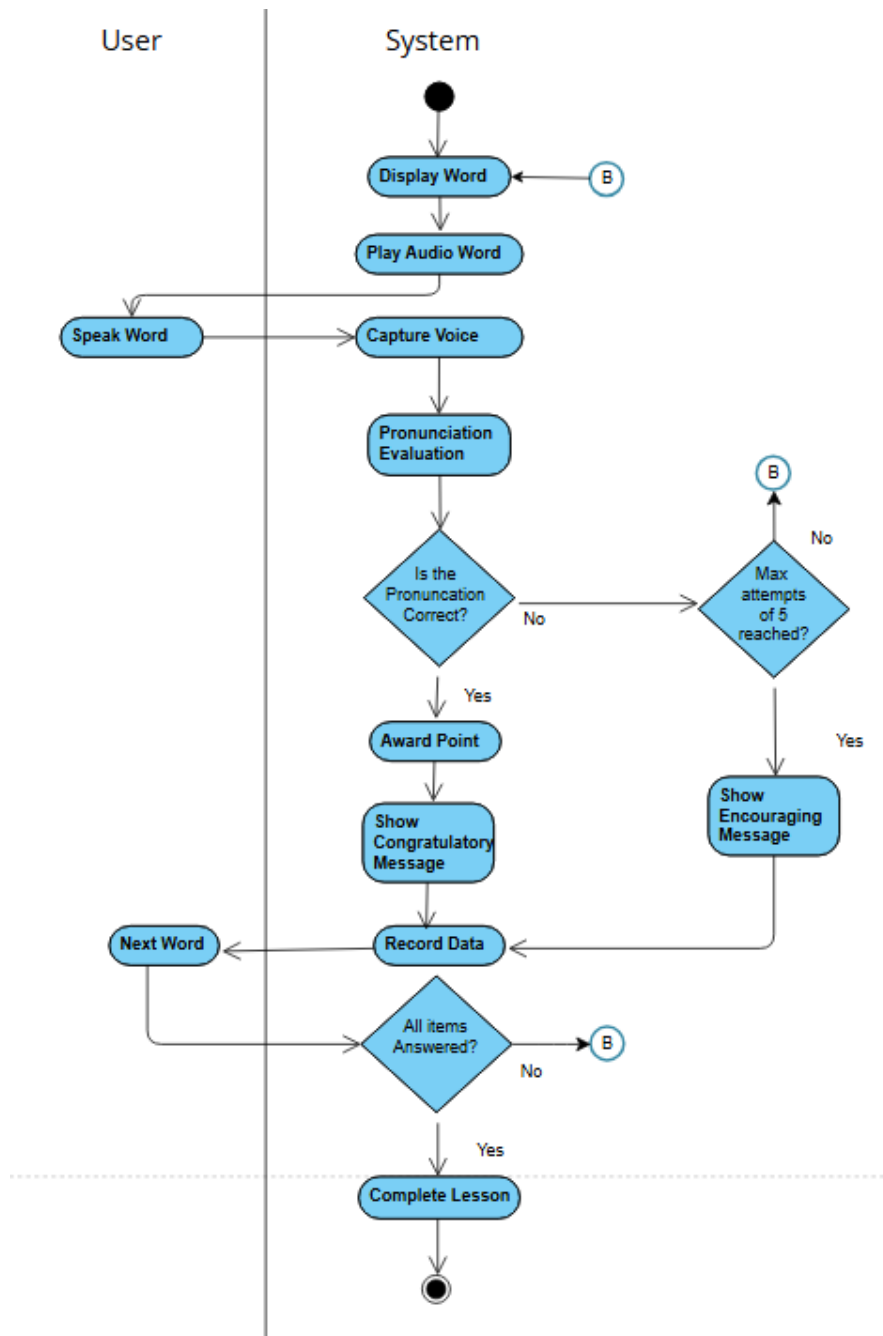


3.2.3.2 Core Gameplay Mechanics

Use Case	Core Gameplay mechanics
Description	The user pronounces a given word, and the system evaluates the accuracy of the pronunciation. If pronounced correctly, the system awards points based on precision and logs the user's progress.
Actors	User, System
Module	Gameplay
Preconditions	<ol style="list-style-type: none">1. User is logged in.2. The user selects a category.3. The user selects a lesson.
Basic Sequence	<ol style="list-style-type: none">1. The system displays or plays the word for the user.2. The user pronounces the word via microphone.3. The system analyzes the audio and recognizes it as correct.4. The system awards points to the user's total score.5. System displays positive feedback (e.g., "Correct! Good job!").6. The system updates the user's progress in the database.7. User proceeds to the next word.8. The user repeats the same process until the lesson is completed.
Exceptions	<ol style="list-style-type: none">1. Microphone Error: If no audio is captured, the system displays an error message {e.g., "Please check your microphone"}.2. Pronounce Word Incorrectly: If the user pronounces the word incorrectly, the system displays the same word and a message to encourage the user to try again.3. Five Incorrect Pronunciation Attempts: If the user pronounces the word

	incorrectly five times, the system will show an encouraging message, record data and proceeds to the next word.
Postconditions	<ol style="list-style-type: none">1. Pronounce Word Correctly2. The user pronounces the word correctly.3. The user's score is incremented.4. The system logs the correct attempt for progress tracking and analytics.

Activity Diagram

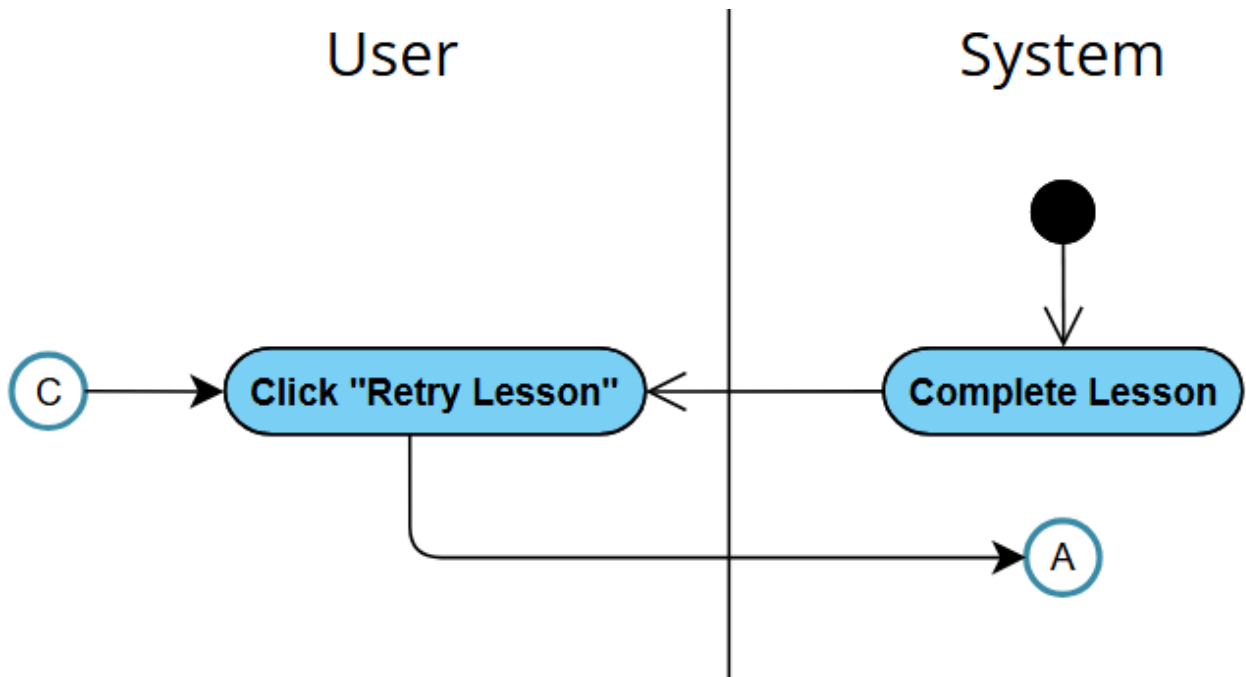


3.2.3.3 User Retry Lesson

Use Case	User Retry lesson
Description	The user can restart the current lesson and attempt pronunciation exercises again.
Actors	User, System

Module	Gameplay
Preconditions	1. The user has completed the lesson.
Basic Sequence	1. User selects the Retry Lesson button. 2. The system reloads the current lesson to start from the first word. 3. The user reattempts pronunciation exercises.
Exceptions	1. System Error: If the lesson fails to reload, the system displays an error message and prompts the user to try again later.
Postconditions	1. The user can proceed to view score details. 2. The user can proceed to the next lesson. 3. The system saves the user's score of the lesson.

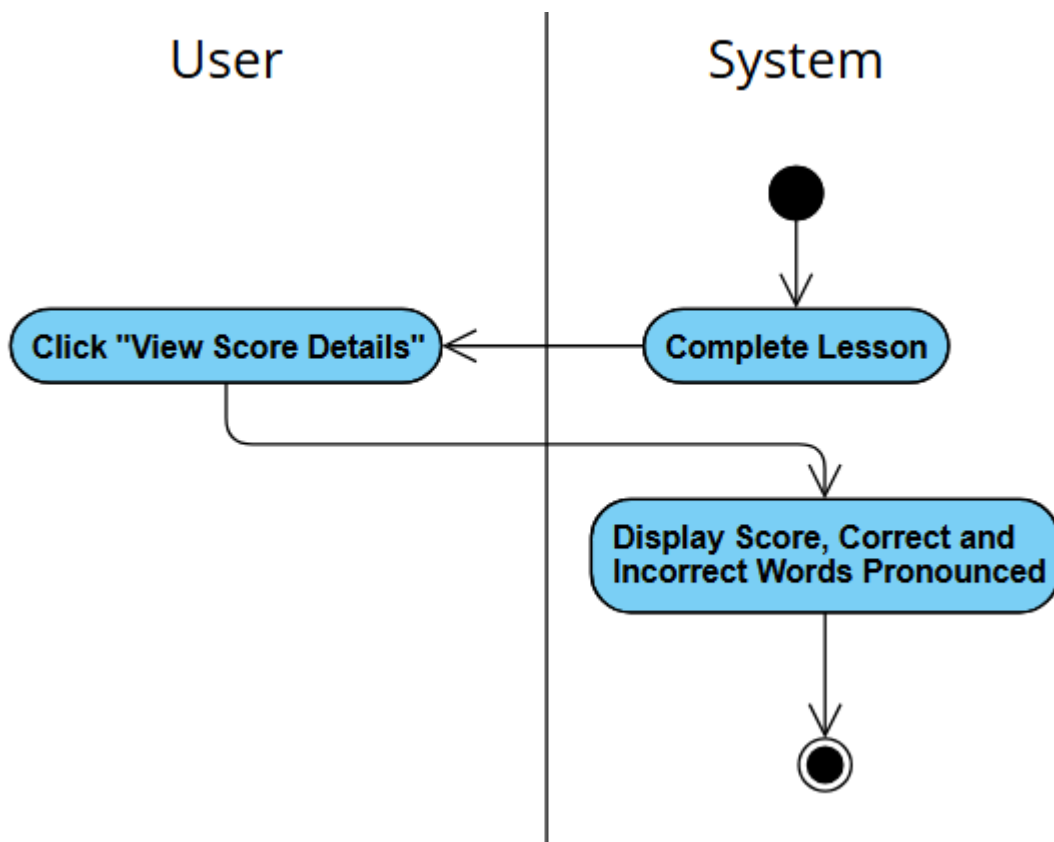
Activity Diagram



3.2.3.4 View Score Details

Use Case	View Score Details
Description	Displays the user's performance, showing which words were correctly and incorrectly pronounced.
Actors	User, System
Module	Gameplay
Preconditions	<ol style="list-style-type: none"> 1. The user has completed a lesson. 2. The system has stored pronunciation results.
Basic Sequence	<ol style="list-style-type: none"> 1. User selects View Score Details after finishing a lesson. 2. The system displays the total score, correctly pronounced words and incorrectly pronounced words.
Exceptions	<ol style="list-style-type: none"> 1. System Error: If the data fails to reload, the system displays an error message.
Postconditions	<ol style="list-style-type: none"> 1. The user gains insight into pronunciation strengths and weaknesses. 2. The user now unlocks a new lesson.

Activity Diagram

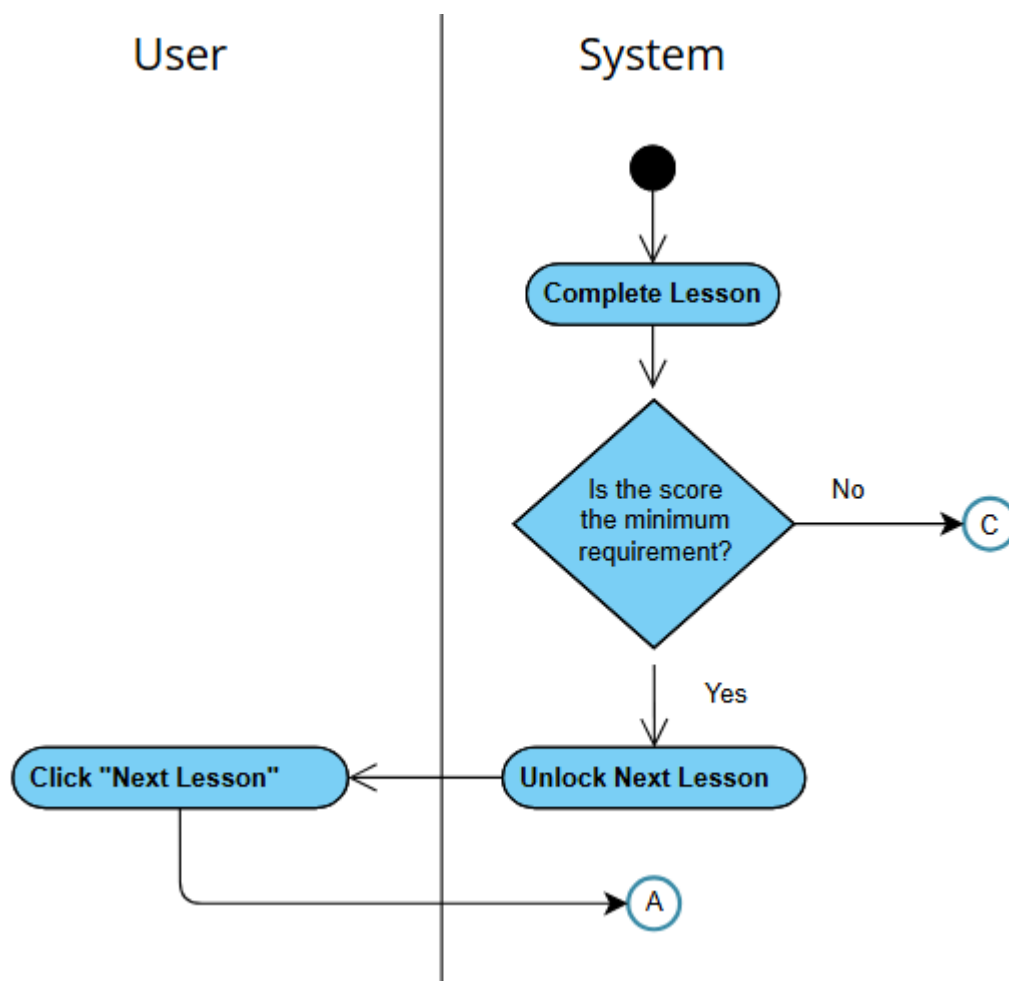


3.2.3.5 Next Lesson

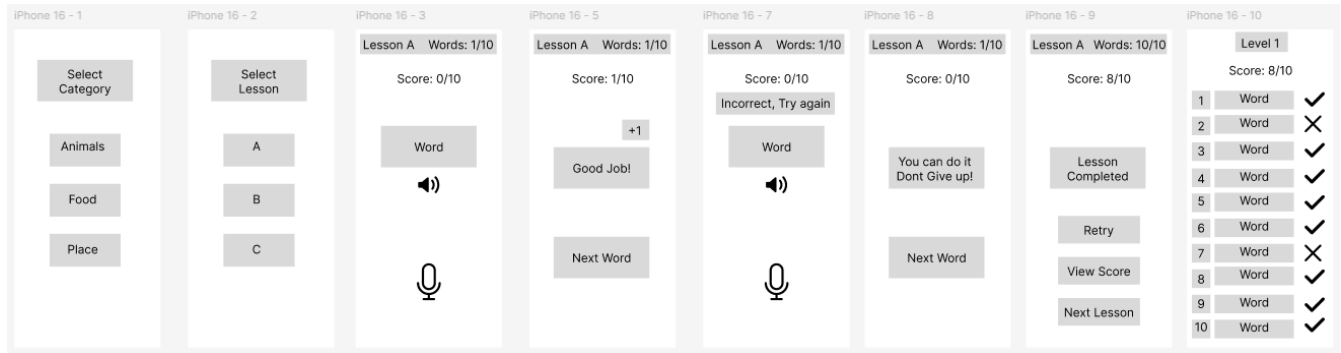
Use Case	Next Lesson
Description	Allow the user to proceed to the next lesson with a different set of words or exercises.
Actors	User, System
Module	Gameplay
Preconditions	<ol style="list-style-type: none"> 1. The user has completed a lesson. 2. The user has unlocked the next lesson.
Basic Sequence	<ol style="list-style-type: none"> 1. User selects Next lesson button. 2. If unlocked, the system proceeds to the new lesson.
Exceptions	<ol style="list-style-type: none"> 1. System Error: If the next lesson fails to

	load, the system displays an error message.
Postconditions	<ol style="list-style-type: none"> 1. The user advances to the next lesson. 2. The system saves the user's previous score

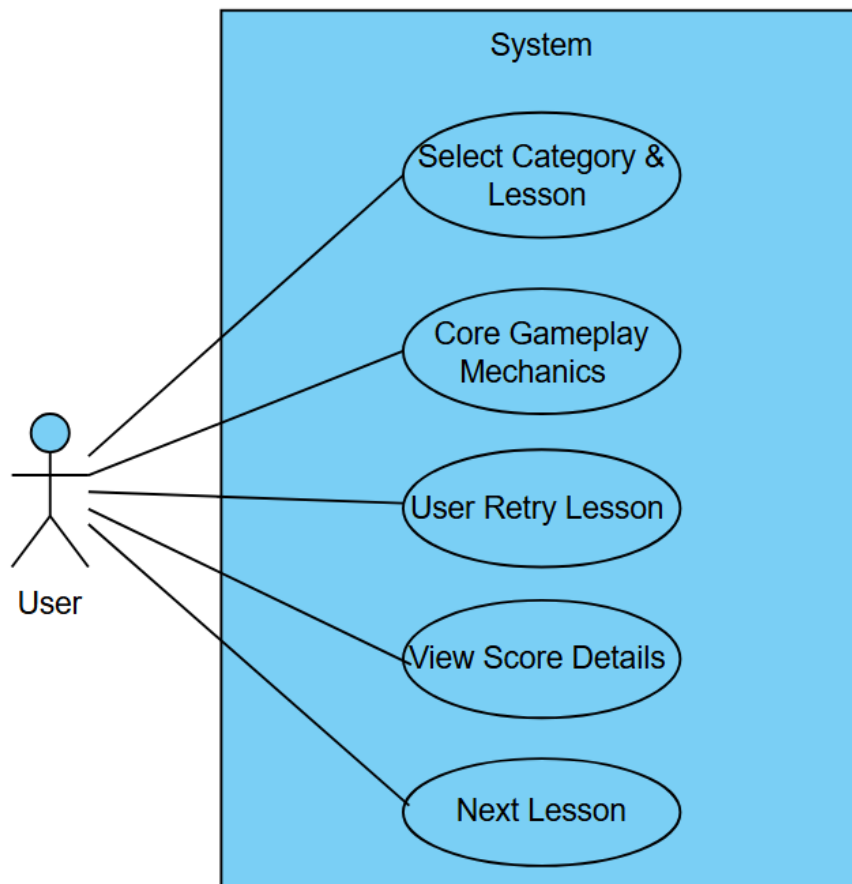
Activity Diagram



Wireframe



User Case Diagram



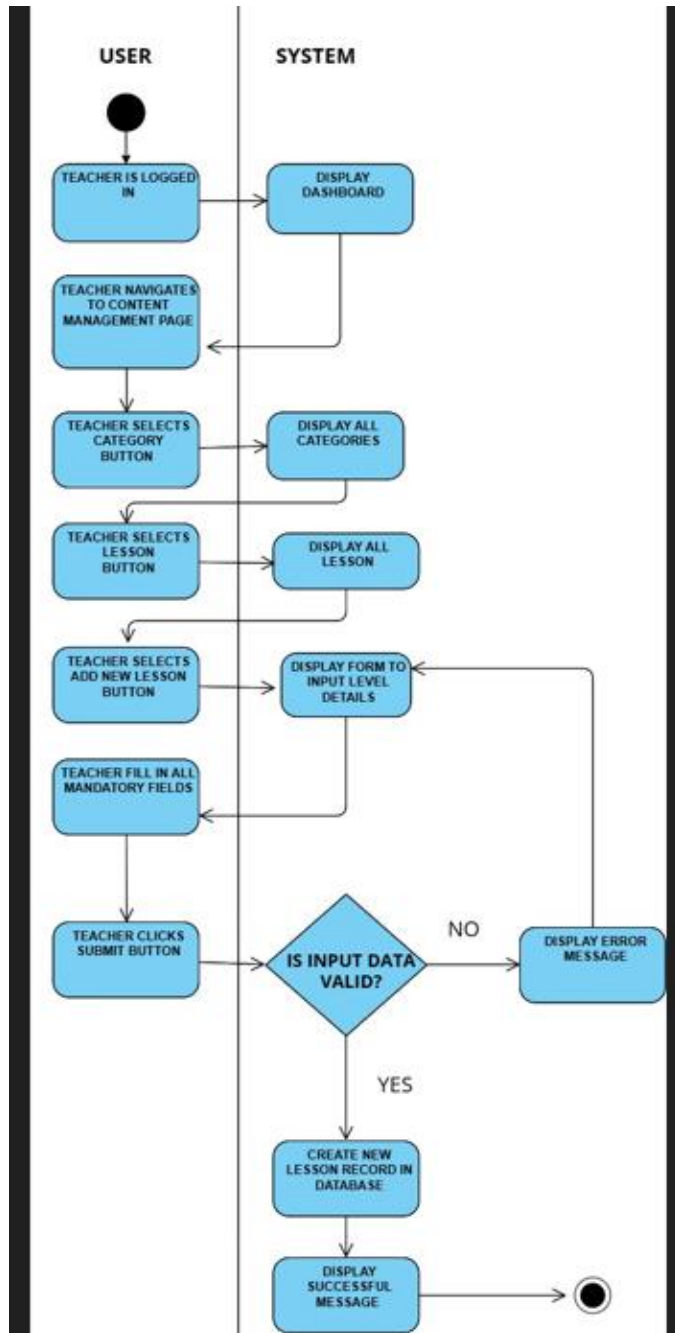
3.2.4 Module 4: Teacher Management

3.2.4.1 Teacher Adds Lesson

Use Case	Teacher Adds Lesson
Description	The teacher creates a new lesson, setting criteria such as lesson name.
Actors	Teacher, System
Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is on the teacher dashboard.
Basic Sequence	<ol style="list-style-type: none">1. The teacher navigates to the content management page.2. The teacher selects Category button3. The system displays all the categories available.4. The teacher selects a lesson.5. The system displays all levels within selected lesson.6. The teacher selects the "Add New Lesson" option from the dashboard.7. The system displays a form to input lessons details.8. The teacher fills in all mandatory fields and clicks "Submit".9. The system validates the input data.10. The system creates a new lesson record in the database.11. System displays a success message ("Lesson successfully created").12. System validates and updates the database.
Exceptions	<ol style="list-style-type: none">1. Validation error: If mandatory fields are missing or data is invalid, the system notifies the admin.2. Database error: if saving fails, the system logs, the error and notifies, the

	admin to retry.
Postconditions	<ol style="list-style-type: none">1. The new lesson is stored in the system and visible in the lesson list.2. The lesson is available once unlocked through user progress.

Activity Diagram

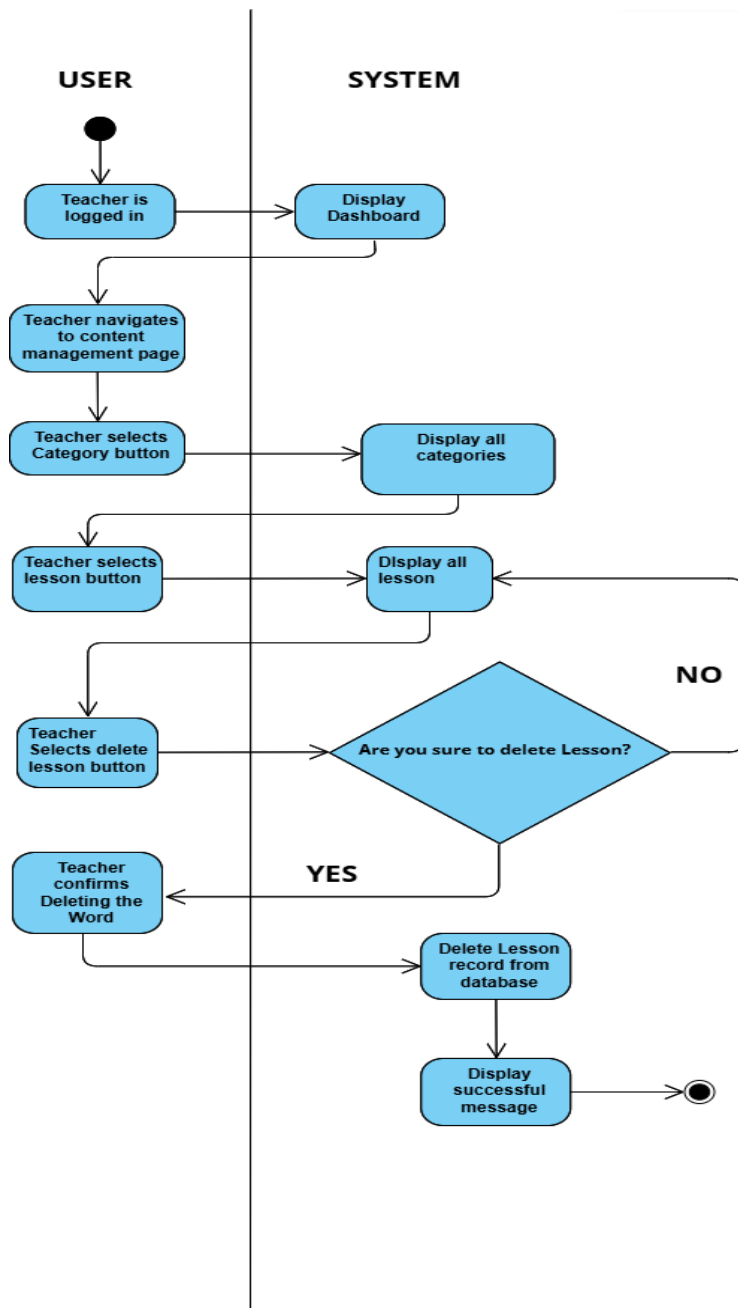


3.2.4.2 Teacher Deletes Lesson

Use Case	Teacher Deletes Lesson
Description	The teacher deletes an existing lesson from the system.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is on the dashboard.
Basic Sequence	<ol style="list-style-type: none">1. The teacher navigates to the content management page.2. The system displays all the lessons available.3. The teacher selects a lesson.4. The system displays all lessons.5. Then select the "Delete Lesson" option from the dashboard.6. The system asks for confirmation ("Are you sure you want to delete Lesson?").7. The teacher confirms deleting the word.8. The system deletes the lesson record and associate data from the database.9. The system displays a successful message that the lesson has been deleted.
Exceptions	<ol style="list-style-type: none">1. Cancellation: If the teacher cancels the confirmation, no changes are made.2. Database error: If deletion fails, the system logs the error and notifies the teacher to retry.
Postcondition	<ol style="list-style-type: none">1. The selected lesson is permanently removed from the system.2. The system updates the lesson list and adjusts any impacted user progress data.

Activity Diagram

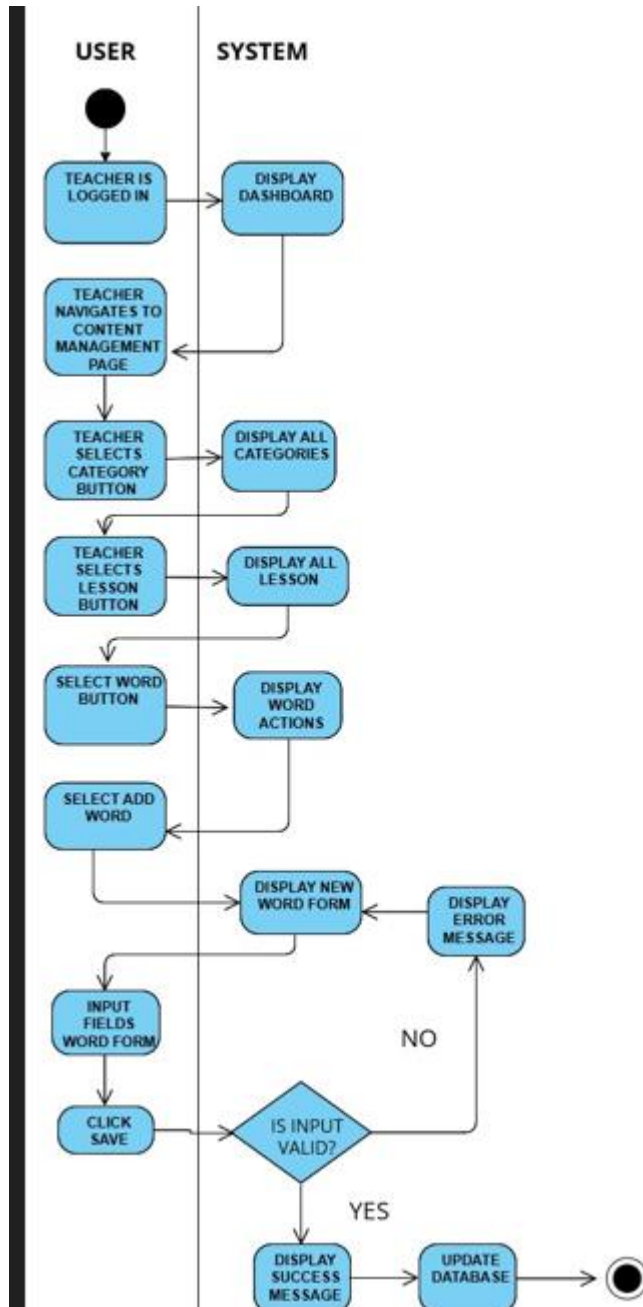


3.2.4.3 Teacher Adds Content

Use Case	Teacher Adds Content
Description	The teacher adds a new word entry along with its pronunciation guide, audio files, and associated category and lesson.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none"> 1. The teacher is logged in. 2. The teacher is in dashboard.
Basic Sequence	<ol style="list-style-type: none"> 1. The teacher navigates the content management page. 2. The teacher selects category button 3. The system displays the categories available. 4. The teacher selects the lesson button. 5. The system displays all lessons. 6. The teacher selects the word Button. 7. The system displays all words at the lesson. 8. The teacher selects "Add Word". 9. The system displays a form for the new word. 10. The teacher fills out the required fields of the new word form. 11. The teacher clicks on the save button. 12. The system validates the input data. 13. The system creates a new word record in the database. 14. System displays a success message ("Word Content successfully created"). 15. System validates and updates the database.
Exceptions	<ol style="list-style-type: none"> 1. Invalid Data: If required fields are incomplete or incorrect, the system prompts corrections. 2. Upload Failure: If the audio file upload fails, the system notifies the teacher and suggests retrying.
Postconditions	<ol style="list-style-type: none"> 1. The new word appears in the word list for the selected lesson. 2. It becomes available for use in pronunciation exercises.

Activity Diagram

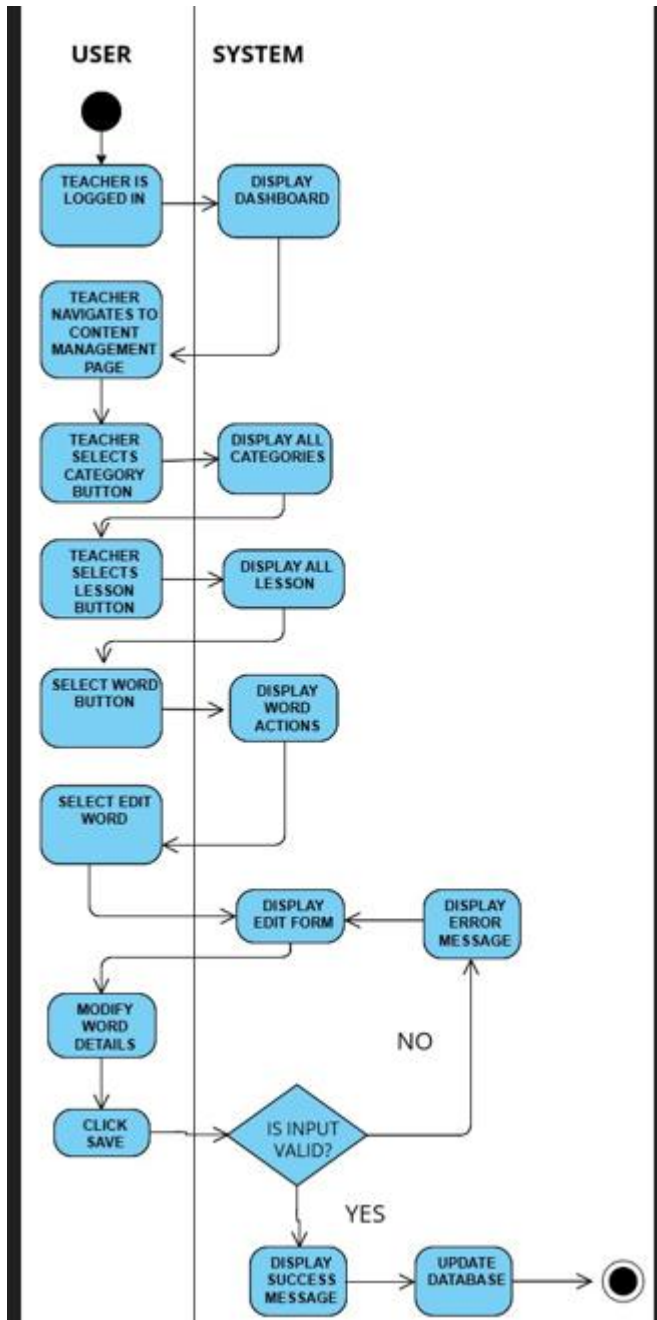


3.2.4.4 Teacher Edits Content

Use Case	Teacher Edits Word Content
Description	The teacher edits details of an existing word, including updating the text, pronunciation guide, audio file, or category and lesson assignment.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none"> 1. The teacher is logged in. 2. The teacher is in dashboard.
Basic Sequence	<ol style="list-style-type: none"> 1. The teacher navigates to the content management page. 2. The teacher selects the Category Button. 3. The system displays the categories available. 4. The teacher selects a lesson button. 5. The system displays all lesson. 6. The teacher selects a word button . 7. The system displays all words at the lesson. 8. The teacher selects a word. 9. The system displays all actions for the word. 10. The teacher selects "Update Word". 11. The system displays the "Update Word" form. 12. The teacher modifies the desired fields (e.g. word, audio). 13. Teacher saves changes. 14. The system validates and updates the database. 15. The system displays a confirmation message (e.g., "Word updated successfully").
Exceptions	<ol style="list-style-type: none"> 1. Validation Error: If the updated data is incomplete or incorrect, the system prompts the teacher to correct the input. 2. Upload/Database Error: If saving fails, the system notifies the teacher to retry.
Postconditions	<ol style="list-style-type: none"> 1. The word's updated information is saved in the database. 2. The changes are immediately reflected in the word list and related exercises.

Activity Diagram

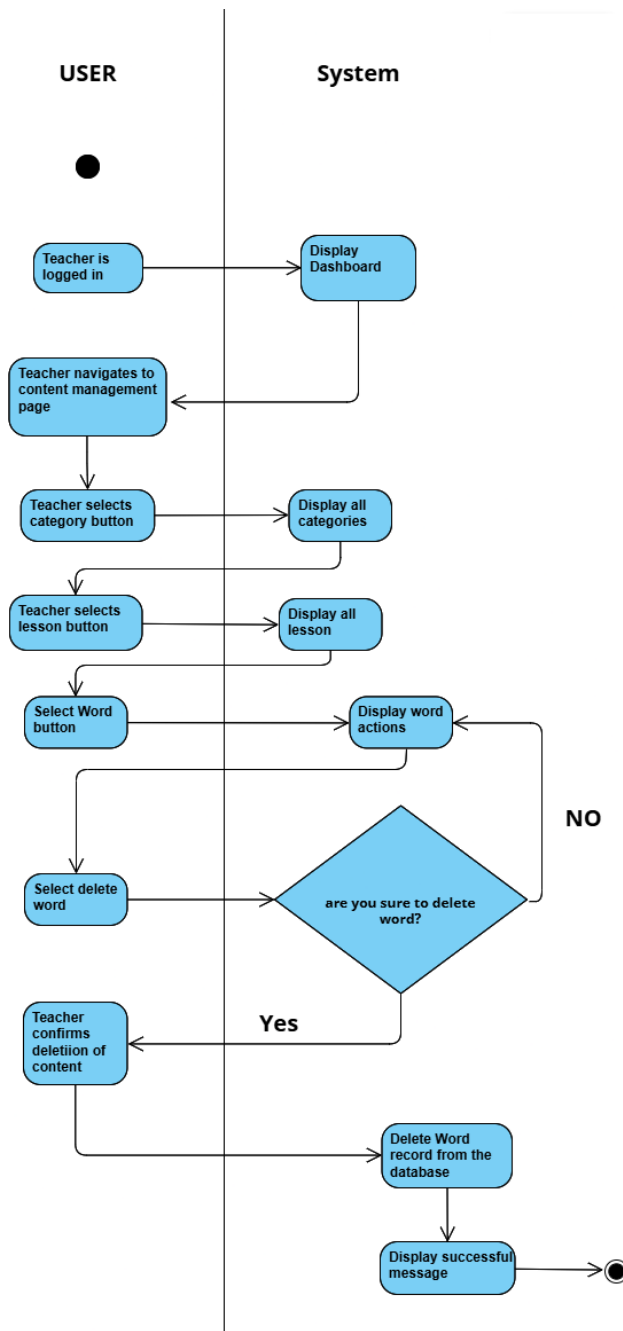


3.2.4.5 Teacher Deletes Content

Use Case	Teacher Deletes Content
Description	The teacher deletes a word and their pronunciation audio/files.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none"> 1. The teacher is logged in. 2. The teacher is in dashboard.
Basic Sequence	<ol style="list-style-type: none"> 1. The teacher navigates to the content management page. 2. The system displays the categories available. 3. The teacher selects categories. 4. The system displays all lessons in the difficulty. 5. The teacher selects the lesson. 6. The system displays all words at the lessons. 7. The teacher selects a word. 8. The system displays all actions for the word. 9. The teacher selects "Delete Word". 10. The system asks for confirmation ("Are you sure you want to delete word?"). 11. The teacher confirms deleting the word. 12. The system deletes the word record and associated data from the database. 13. The system displays a successful message that the word has been deleted.
Exceptions	<ol style="list-style-type: none"> 1. Cancellation: If the teacher cancels the confirmation, no changes are made. 2. Database error: If deletion fails, the system logs, the error and notifies the teacher to retry.
Postconditions	<ol style="list-style-type: none"> 1. The selected lesson is permanently removed from the system. 2. The system updates the lesson list and adjusts any impacted user progress data.

Activity Diagram

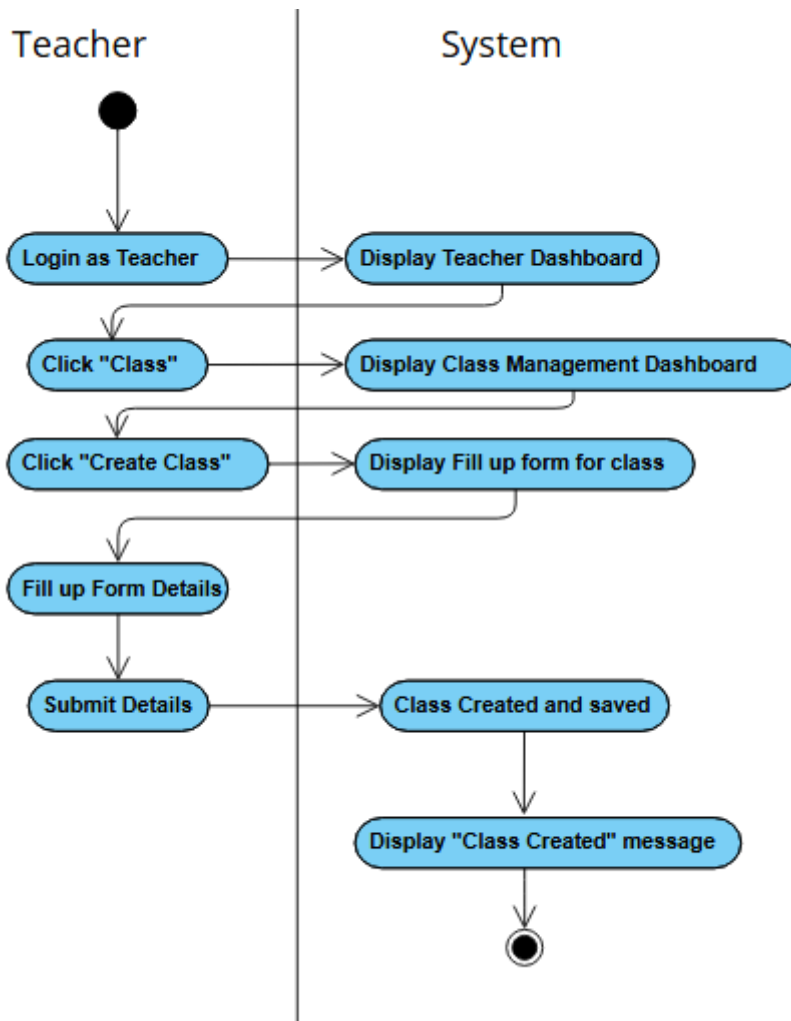


3.2.4.6 Teacher Creates Class

Use Case	Teacher Adds Class
Description	The teacher creates a class and organizes users within these groups for easier management.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is on class management dashboard.
Basic Sequence	<ol style="list-style-type: none">1. Teacher selects the "Create Class" option.2. The system displays a form to create a new class or group (e.g., class name, description).3. The teacher fills up the form with details4. System saves the class/group details5. System displays a confirmation message (e.g., "Class created successfully").
Exceptions	<ol style="list-style-type: none">1. Invalid Input: If class details are incomplete or incorrect, the system prompts for corrections.2. Database Error: If grouping fails, the system notifies the teacher to retry.
Postconditions	<ol style="list-style-type: none">1. The new class/group is created and stored in the system.

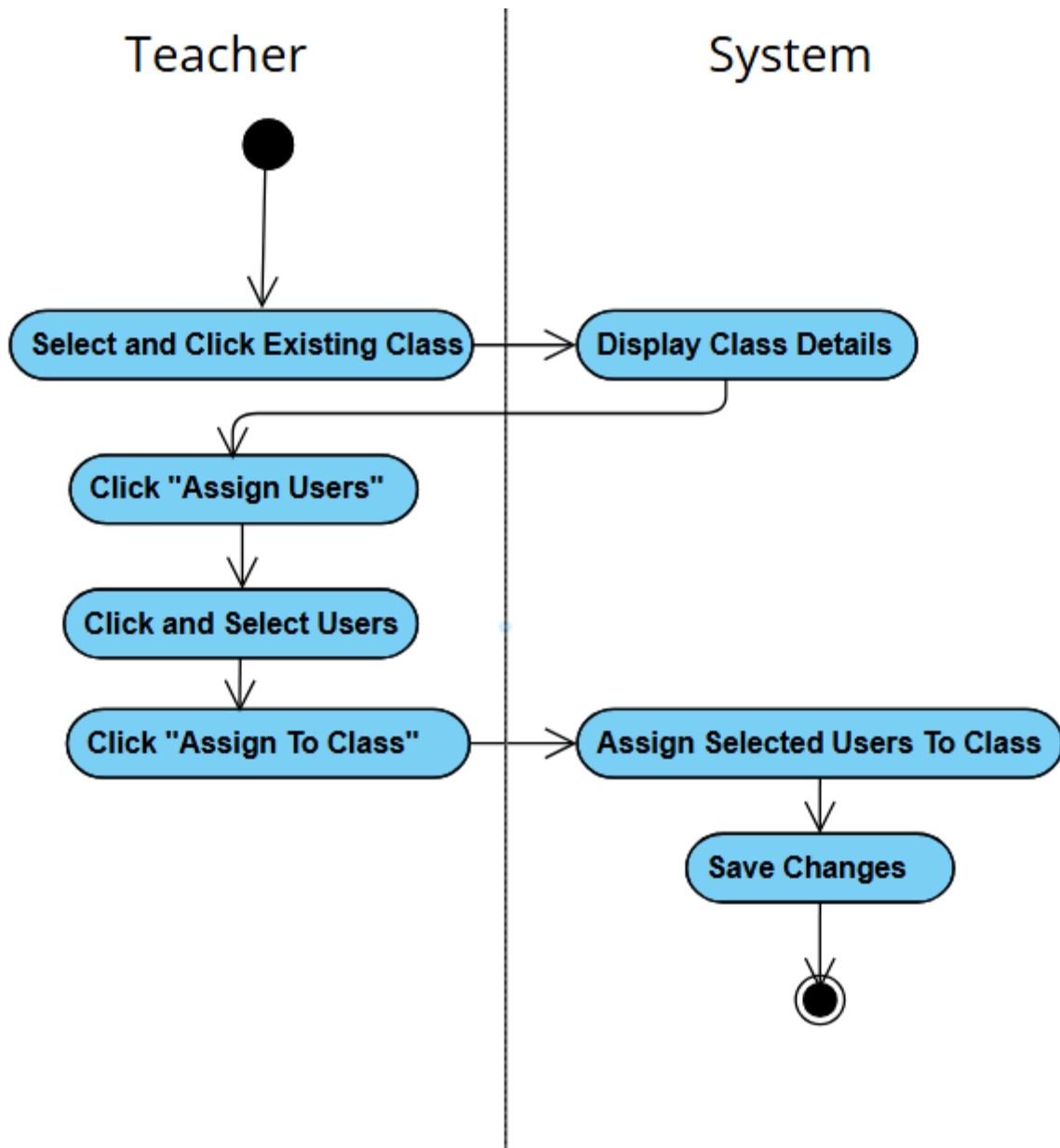
Activity Diagram



3.2.4.7 Teacher Assigns User to Class

Use Case	Teacher Assigns Student to Class
Description	The teacher assigns users (students) to a class or group for effective organization and management.
Actors	Teacher, System
Module	Teacher Management
Preconditions	<ol style="list-style-type: none"> 1. The teacher is logged in. 2. The teacher is on the class management dashboard. 3. At least one class/group exists in the

	system.
Basic Sequence	<ol style="list-style-type: none">1. The teacher selects an existing class.2. The system displays the list of available users.3. The teacher selects the users to be assigned to the class.4. The teacher confirms the assignment.5. The system updates the class details with the assigned users.6. The system displays a confirmation message (e.g., "Users successfully assigned to class").
Exceptions	<ol style="list-style-type: none">1. Invalid Selection: If no users are selected, the system prompts the teacher to choose at least one.2. Database Error: If the assignment fails due to a database issue, the system notifies the teacher to retry.
Postconditions	<ol style="list-style-type: none">2. The selected users are successfully assigned to the class/group.3. Users can now access class-related content and functionalities.

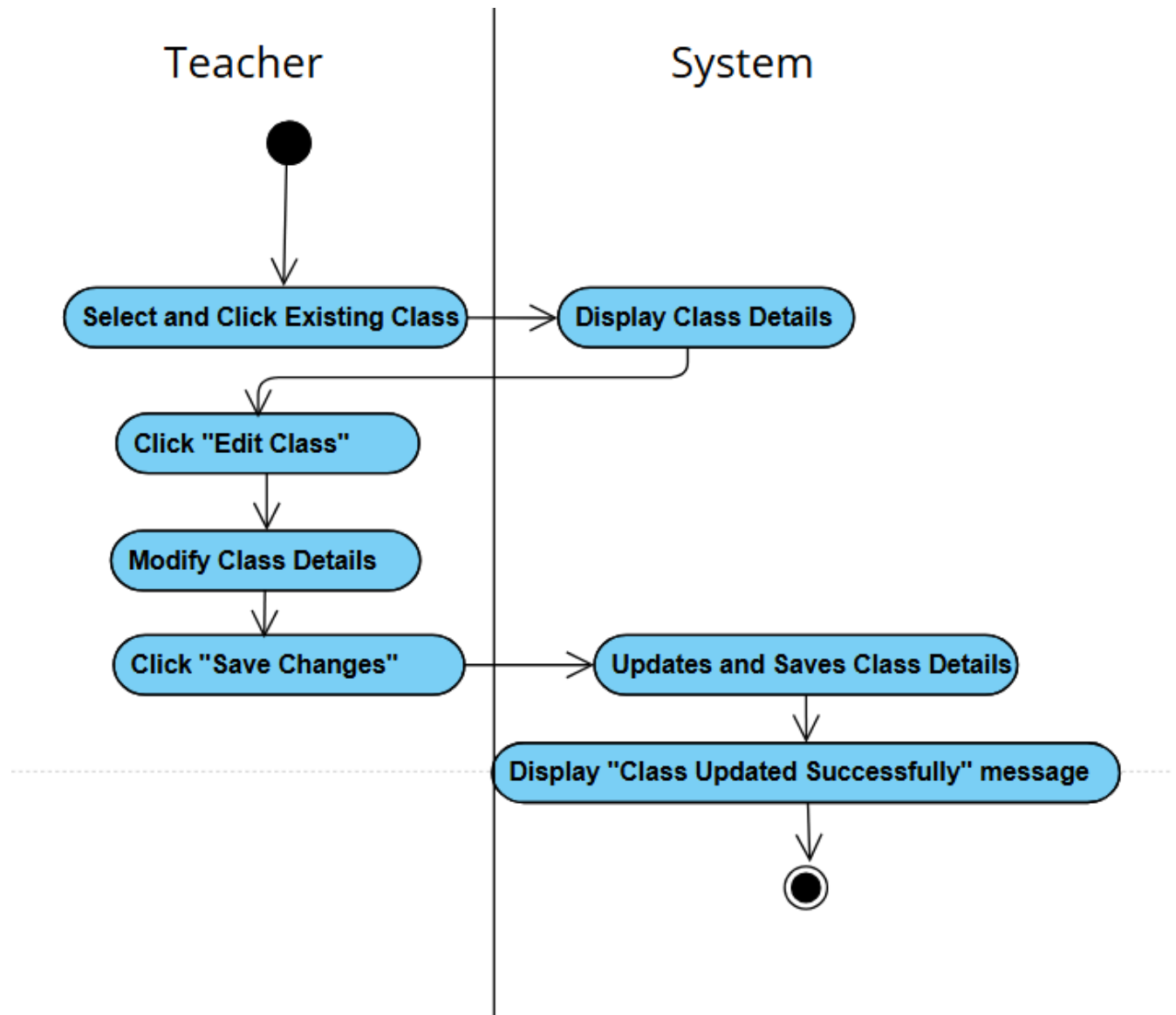


3.2.4.8 Teacher Edits Class

Use Case	Teacher Edits Class
Description	The teacher modifies details of an existing class, such as the class name, description, or assigned users.
Actors	Teacher, System
Module	Teacher Management

Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is in the class management dashboard.
Basic Sequence	<ol style="list-style-type: none">1. The teacher selects a class to edit from the class list.2. Teacher clicks the "Edit Class" option.3. The system displays a pre-populated form with the current class details (class name, description, and list of assigned users).4. The teacher updates the desired fields and/or modifies the user assignments.5. The teacher clicks "Save" to submit the changes.6. The system validates the updated information and saves the changes in the database.7. The system displays a confirmation message (e.g., "Class updated successfully").
Exceptions	<ol style="list-style-type: none">1. Validation Error: If required fields are incomplete or invalid, the system prompts the teacher to correct the input.2. Database Error: If saving fails, the system notifies the teacher and asks to retry.
Postconditions	<ol style="list-style-type: none">1. The updated class details are saved in the system.2. Changes are immediately reflected on the Class Management screen and in any related analytics.

Activity Diagram

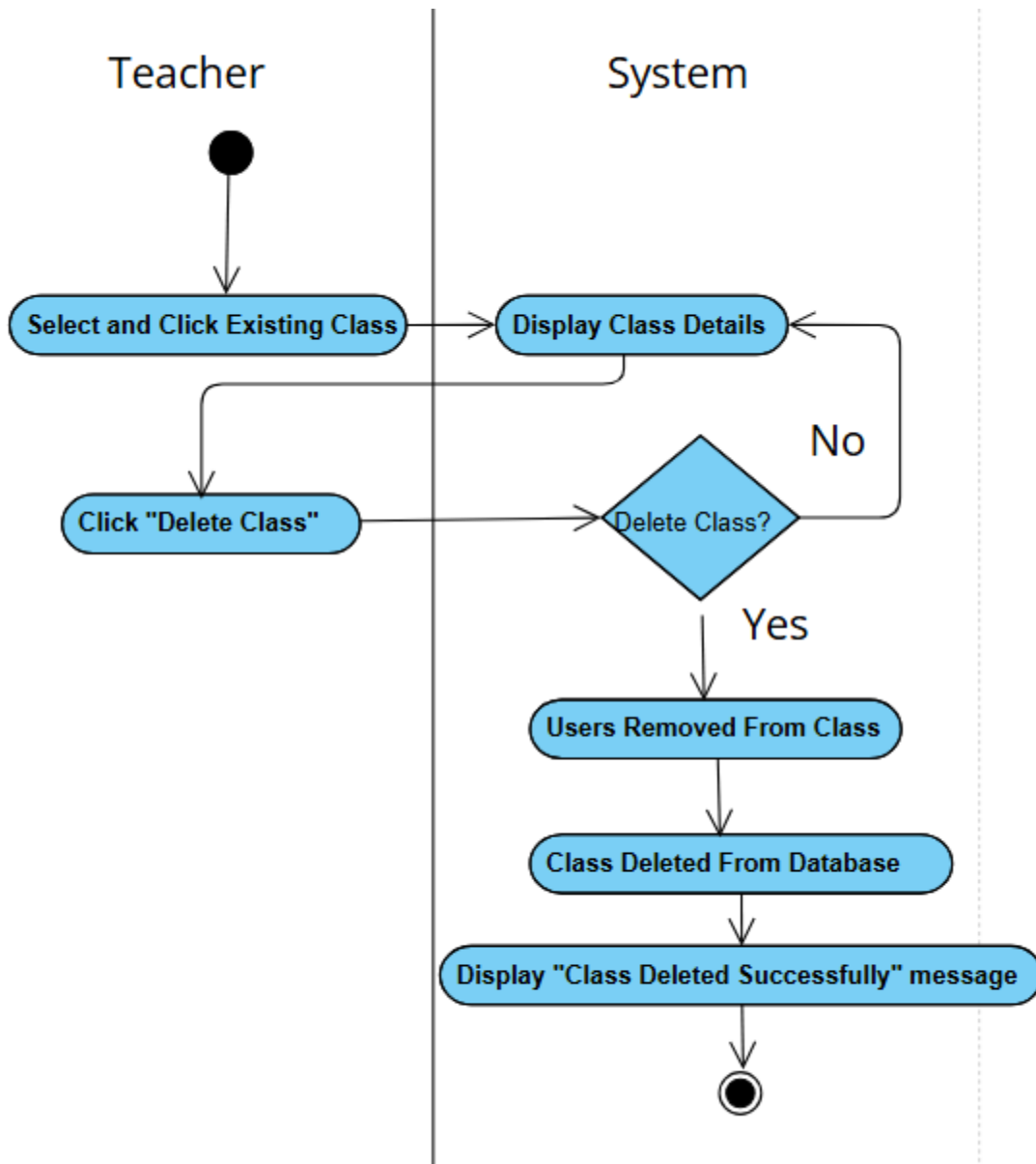


3.2.4.9 Teacher Deletes Class

Use Case	Teacher Deletes Class
Description	The teacher deletes an existing class from the system, removing the class grouping and dissociating any assigned users.
Actors	Teacher, System
Module	Teacher Management
Preconditions	1. The teacher is logged in.

	<ol style="list-style-type: none">2. The teacher is in the class management dashboard.
Basic Sequence	<ol style="list-style-type: none">1. The teacher selects the class to delete from the class list.2. The teacher clicks the "Delete Class" option for that class.3. The system prompts for confirmation (e.g., "Are you sure you want to delete this class? This action cannot be undone.").4. The teacher confirms the deletion.5. The system removes the class and dissociates any assigned users from that group in the database.6. The system displays a confirmation message (e.g., "Class deleted successfully").
Exceptions	<ol style="list-style-type: none">1. Cancellation: If the teacher cancels the confirmation, no changes are made.2. Database Error: If deletion fails, the system notifies the teacher and suggests retrying later.
Postconditions	<ol style="list-style-type: none">1. The selected class is permanently removed from the system.2. Any references to the class in user profiles or progress reports are updated accordingly.

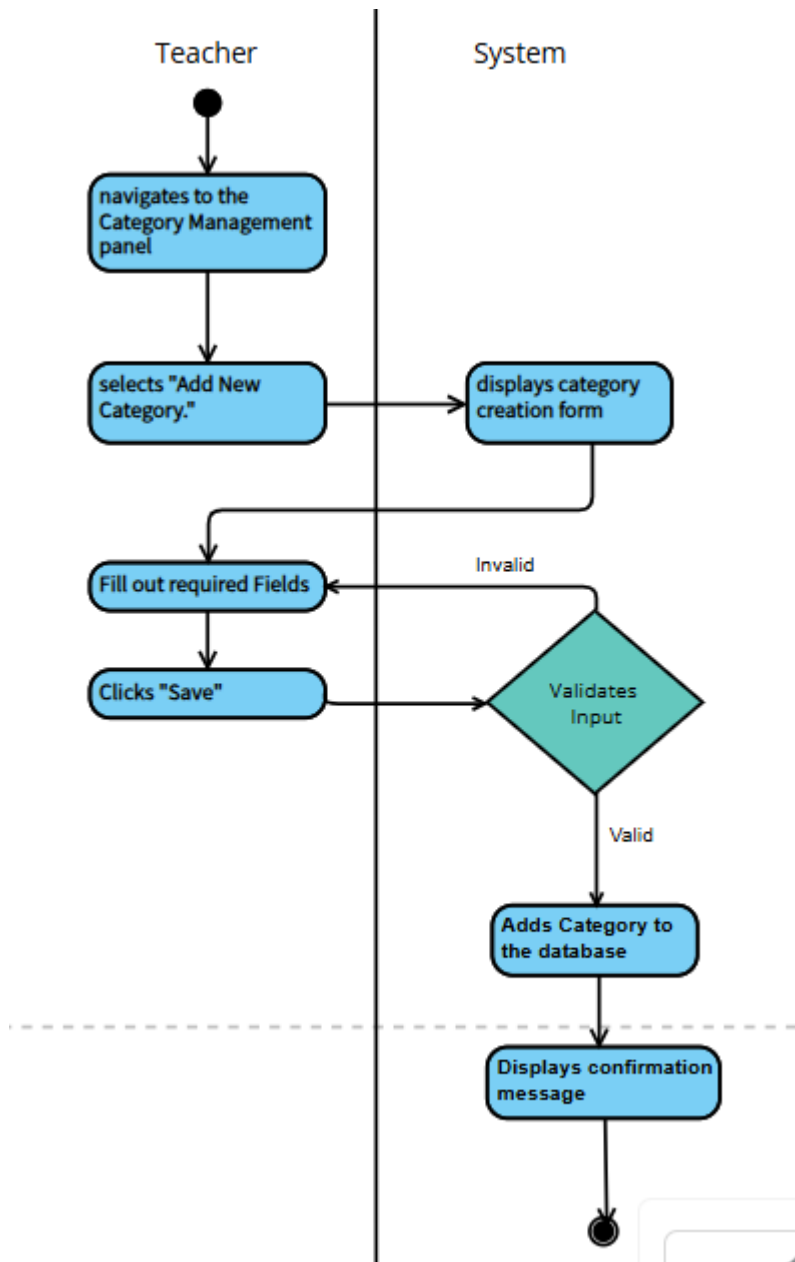
Activity Diagram



3.2.4.10 Teacher Adds Category

Use Case	Teacher Adds Category
Description	A teacher creates a new vocabulary category for students.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is in the teacher dashboard.
Basic Sequence	<ol style="list-style-type: none">1. Teacher navigates to the Category Management panel.2. Teacher selects "Add New Category."3. The system displays a form for entering the category name (e.g., "Animals that start with letter A").4. Teacher fills out required fields and clicks "Save."5. System validates input and adds the new category to the database.6. A confirmation message appears ("Category successfully created").
Exceptions	<ol style="list-style-type: none">1. If mandatory fields are missing, the system notifies the teacher.
Postconditions	<ol style="list-style-type: none">1. The new category is available for students.2. Any modifications reflect in the pronunciation exercises.

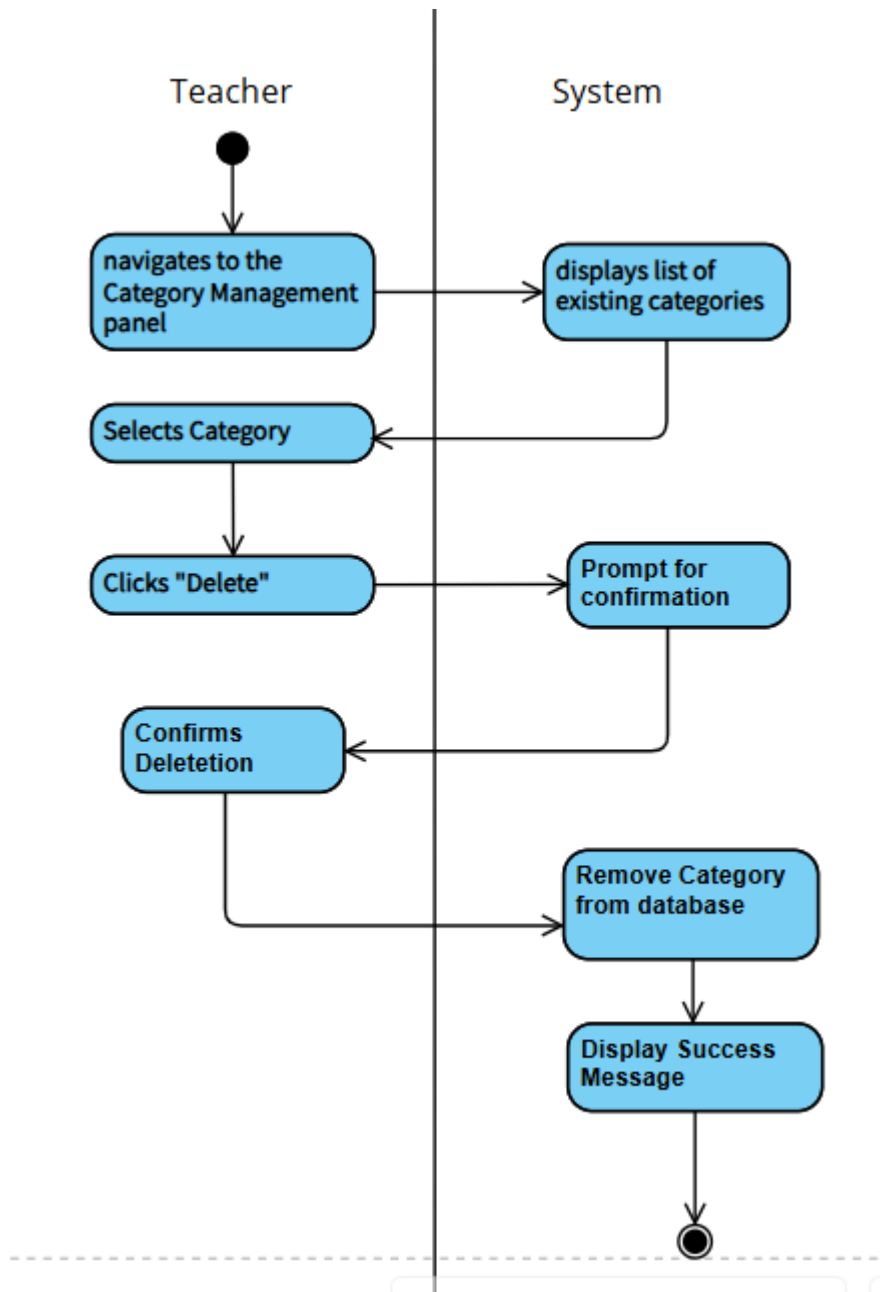


3.2.4.11 Teacher Deletes Category

Use Case	Teacher Deletes Category
Description	A teacher removes an existing vocabulary category.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is in the teacher dashboard.
Basic Sequence	<ol style="list-style-type: none">1. Teacher navigates to the Category Management panel.2. The system displays the list of existing categories.3. Teacher selects a category and clicks "Delete."4. The system prompts for confirmation.5. Teacher confirms the deletion.6. The system removes the category from the database and displays a success message.
Exceptions	<ol style="list-style-type: none">1. If a database error occurs, the system logs the error and notifies the teacher.
Postconditions	<ol style="list-style-type: none">1. The category is removed from the system.2. The category no longer appears in pronunciation exercises.

Activity Diagram

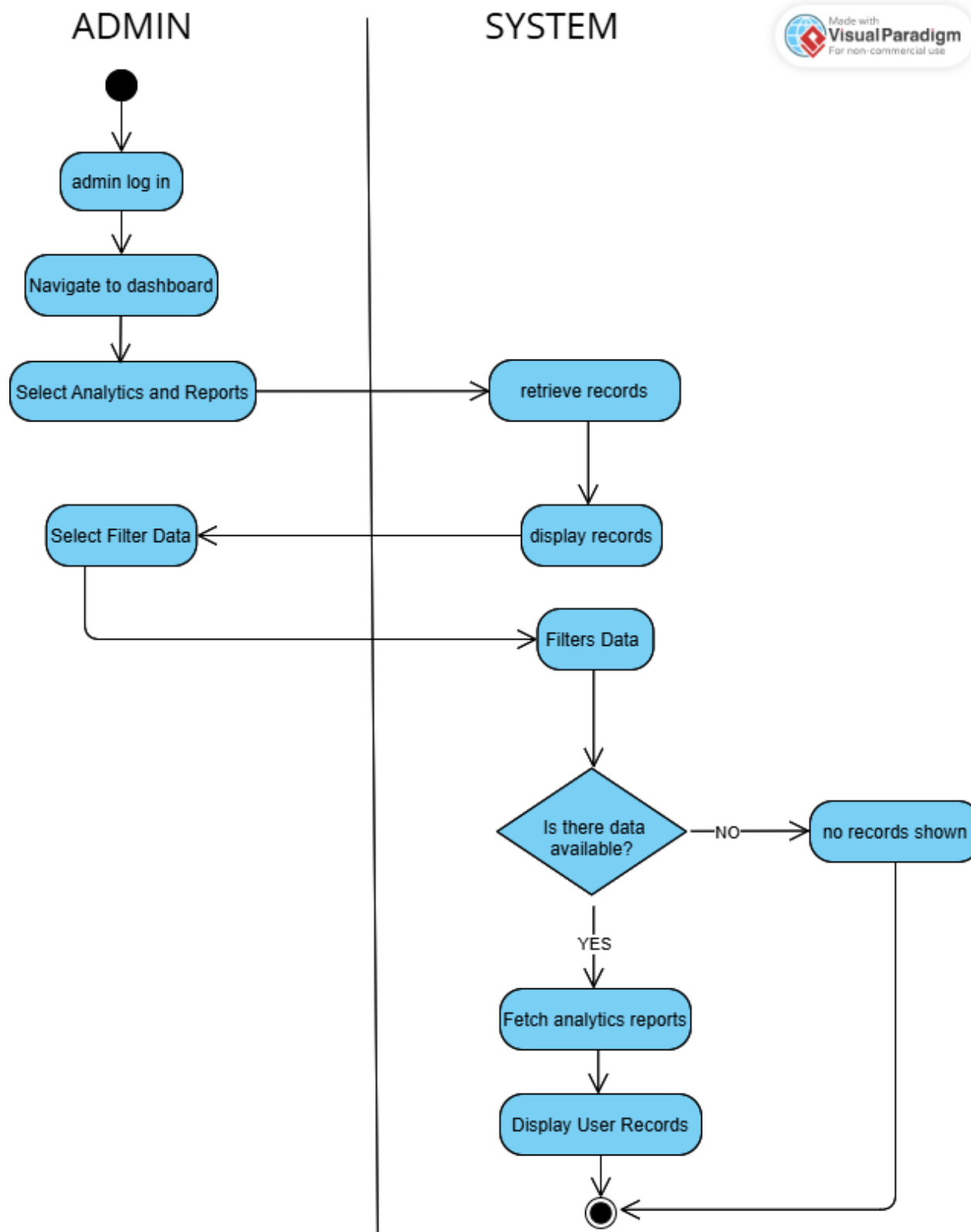


3.2.4.12 Teacher Tracks Scores and Data Analytics

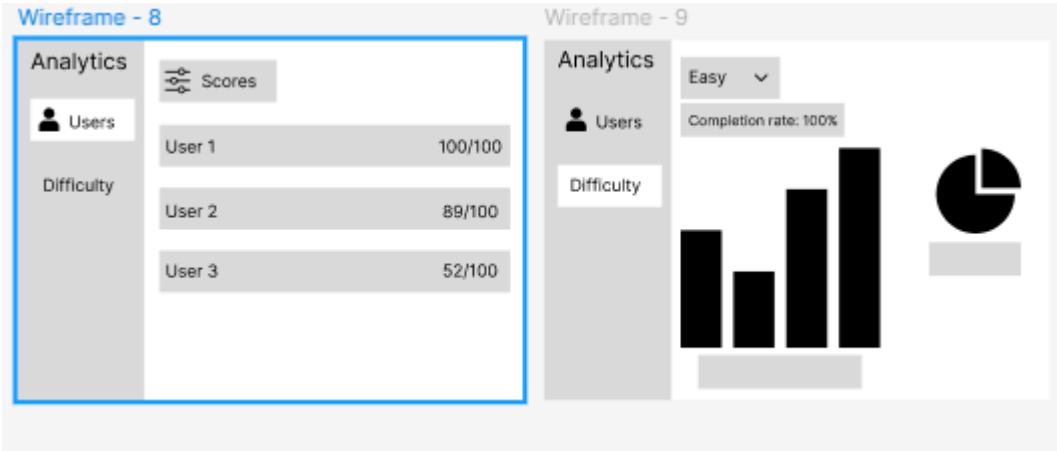
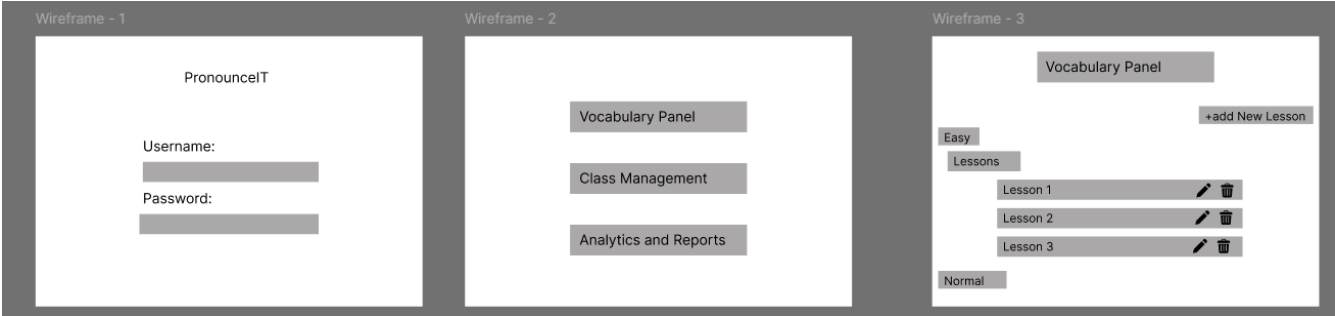
Use Case	Teacher Tracks Scores and Data Analytics
Description	The teacher reviews user performance data across all difficulties and users.
Actors	Teacher, System

Module	Teacher Management
Preconditions	<ol style="list-style-type: none">1. The teacher is logged in.2. The teacher is in the teacher dashboard.
Basic Sequence	<ol style="list-style-type: none">1. The teacher navigates to analytics page.2. The system retrieves user scores, completion rates, and frequently missed words from the database.3. The system displays aggregated charts, tables, or metrics (e.g. Average score per lesson, top 5 most-missed words).4. The teacher can filter data by difficulty or user.
Exceptions	<ol style="list-style-type: none">2. No Data: If no user has completed any lessons, the system indicates no analytics are available.
Postconditions	<ol style="list-style-type: none">3. The teacher can analyze user performance to improve or adjust content4. Data supports decisions on whether to add new lesson, modify diffusely, or provide targeted feedback.

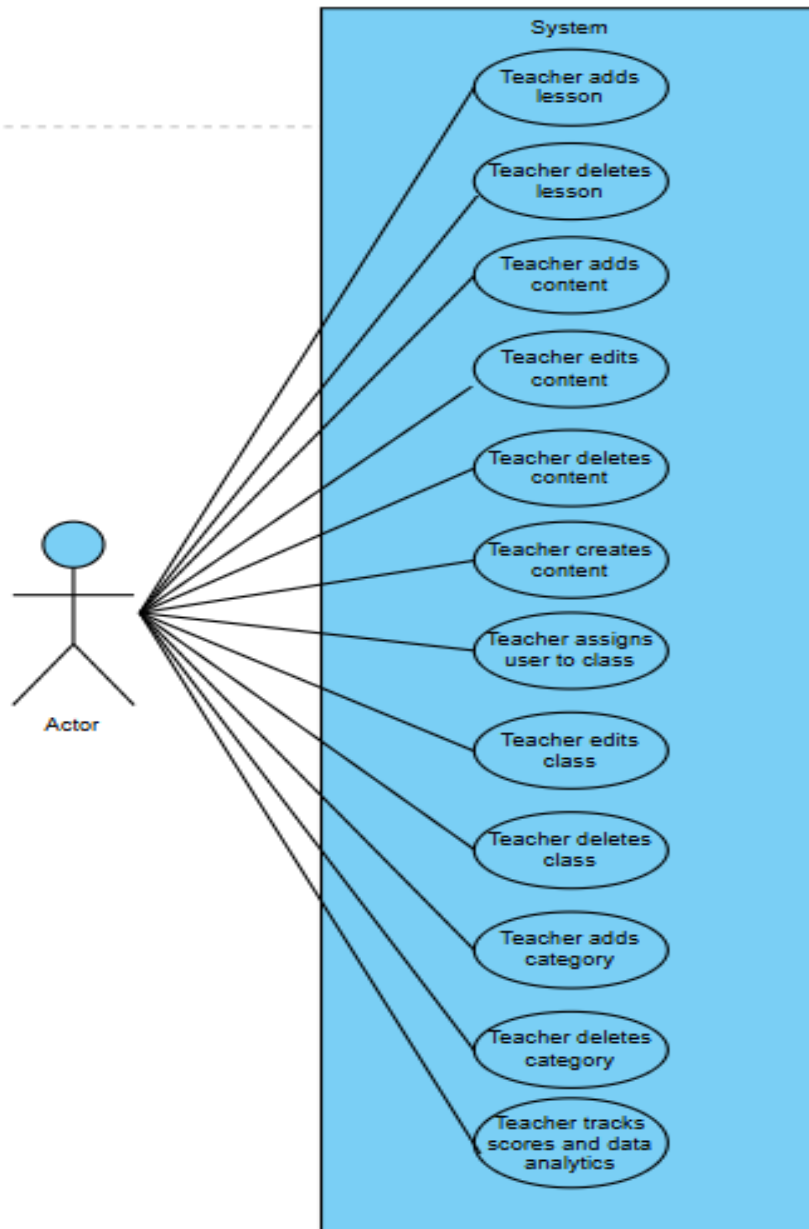
Activity Diagram



- Wireframe



User Case Diagram



3.4 Non-functional requirements

Performance

- The system must process and analyze a pronunciation attempt within **2 seconds** to provide real-time feedback.
- The app should support up to **50 concurrent users** in a classroom or shared environment without performance degradation.
- The voice recognition module should maintain at least **75% accuracy** in detecting pronunciation correctness.
- The system must maintain an average **app response time of ≤ 3 seconds** for all interactive elements, including animations and rewards.
- The application should function smoothly on:
 - **Android Smartphone/Tablets** with **touchscreen capabilities and at least 4GB RAM**.
 - **Desktop/Laptops** (Windows) with **a minimum screen resolution of 1024x768**.
 - **Android Smartphones** with **touchscreen capabilities and at least 4GB RAM**

Security

- Parents and educators should have access to **audit logs** to track system activity and student progress.
- All voice data should be processed **locally** when possible, and if stored in the cloud, it must be encrypted using **AES-256** encryption.
- The app must implement **role-based authentication** (admin and user) with secure login via OAuth 2.0.
- **No Personally Identifiable Information (PII)**, such as full names or addresses, should be stored or transmitted.

Reliability

- The system should achieve 99.5% uptime to ensure consistent availability for students, parents, and teachers.
- The speech recognition engine should handle **background noise** effectively and maintain **at least 75% accuracy** in noisy environments across Windows & Android devices.
- Progress reports and analytics dashboards must be updated **in real-time or within a maximum of 25 minutes** after a session ends.
- A **failsafe mechanism** should be implemented to retry failed API requests (e.g., speech-to-text processing) **at least 3 times** before notifying the user of an error.