FINAL YEAR PROJECT REPORT

BCSF21M011	Muhammad Haider Hamayoun
BCSF20M015	Tuba Saleem
BCSF21M018	Husna Sarwar
BCSF21M003	Muhammad Umar Qureshi

Project Advisor:
Dr. Asif Sohail asif.sohail@pucit.edu.pk

Project Title:
Voice-Driven AI Chatbot for Early Childhood Education.

ABSTRACT OF THE PROPOSAL:

This project aims to develop a voice-enabled AI chatbot to facilitate early childhood education. By incorporating voice recognition technology and integrating the Gemini API, the system converts voice queries into text, generates responses, and delivers answers via voice in an engaging, dynamic web environment.

FYDP OVERVIEW

Table 1 Project Proposal Summary

FYDP Goals: To develop a voice-enabled AI system that facilitates real-time, interactive educational dialogue for children, converting voice queries into text-based responses and delivering answers via voice, within a dynamic web environment.
FYDP Objectives: Implement voice recognition technology to capture and transcribe children's spoken questions into text**.**
Integrate the Gemini API to create an AI chatbot capable of analyzing the transcribed queries and generating accurate, text-based responses.
Develop a system to convert the AI-generated text responses back into voice, creating an interactive, speech-driven communication loop.
Build an interactive, web-based platform using the MERN stack that adapts to the nature of children's queries, displaying an engaging, visually dynamic environment to enhance learning experiences.
FYDP Success Criteria: Successful implementation of voice-to-text, AI-based responses, and dynamic, interactive web platform.
Assumptions: Children's queries will be basic, and the system will work in a controlled web environment.
Risks & Obstacles: Challenges may include AI response accuracy and ensuring smooth integration of voice features.
Organization Address: FCIT, University of the Punjab, Lahore, Pakistan
Target End Users
Suggested Project Supervisor: Sir Asif Sohail
Approved By:

Table of Contents

List of Tables

CHAPTER 1
PROJECT PROPOSAL

1. INTRODUCTION

1.1. Background:

Children are increasingly interacting with digital platforms, but most existing systems do not effectively cater to early childhood education needs. A voice-driven, AI-based solution can significantly enhance learning experiences by offering interactive, voice-based dialogues.

1.2. Problem Statement:

There is a lack of AI-driven, voice-enabled tools for young children that provide real-time educational assistance. This project aims to fill that gap by developing an AI chatbot tailored to children's educational needs.

1.3. Stakeholders & Interests

- **Children (3-7 years old):** Primary users, benefiting from personalized educational content.
- **Parents and Educators:** Indirect stakeholders interested in enhancing learning resources for children.

1.4. Objectives:

- Develop a system that converts children's spoken queries into text.
- Integrate the Gemini API to analyze queries and generate responses.
- Implement text-to-speech functionality to complete the interactive loop.
- Build a web-based, interactive learning platform using MERN stack.

1.4. Scope:

This project focuses on voice recognition and AI-driven education for children.

The solution will be a web-based platform that evolves based on children's learning themes.

1.5. Assumptions:

Children's queries will be basic, and the system will work in a controlled web environment.

1.6. Risks:

Challenges may include AI response accuracy and ensuring smooth integration of voice features.

1.7. Success Criteria:

Successful implementation of voice-to-text, AI-based responses, and dynamic, interactive web platform.

1.8. TOOLS, LIBRARIES AND TECHNOLOGIES WITH REASONING

Table 2 Tools Technologies and Libraries

Tools, Libraries, And Technologies	**Tools**	**Version**	**Rationale**
	MERN Stack (MongoDB, Express, React, Node.js)	Latest	To create a scalable, interactive, web- based platform for real-time communication.
	Libraries	**Version**	**Rationale**
	Gemini API	N/A	For AI chatbot integration, capable of analyzing queries and generating appropriate responses.
	Speech-to-Text API	N/A	To capture and convert children's spoken queries into text, enabling the voice interaction

		functionality.
Text-to-Speech API	N/A	To convert AI-generated text responses into speech, closing the interactive communication loop.
Technology	**Version**	**Rationale**
Node.js	Latest	For building the backend logic and efficiently handling Server-side requests and real- time communication.
React	Latest	To develop the dynamic frontend, creating an engaging, interactive user interface that adapts to children's queries.
MongoDB	Latest	For database management, ensuring scalability and efficient storage of user data and interactions.

	Express.js	Latest	As part of the MERN stack, for routing and middleware integration in the web application.

1.8. WORK DIVISION:

Table 3 Project Team Members Work Division

Sr. No.	Roll Number	Name	Role Assignment & Work Division
1.	BCSF21M003	Muhammad Umar Qureshi	Frontend Development (React, UI/UX)
2.	BCSF20M015	Tuba Saleem	Backend Development (Node.js, Express)
3.	BCSF21M018	Husna Sarwar	AI Integration (Gemini API, Chatbot)
4.	BCSF21M011	Muhammad Haider	Speech Recognition and Web Deployment

CHAPTER 2
PROJECT PROPOSAL

2.1. Literature Overview:

The integration of AI in education has focused on personalizing learning experiences through chatbots and voice-driven systems. Research highlights advancements in platforms like Duolingo and IBM Watson, emphasizing benefits such as instant feedback and accessibility. However, these systems predominantly target older students, leaving gaps in adaptability and engagement for young learners. Studies suggest the need for AI solutions that incorporate age-appropriate design, interactive storytelling, and voice-based learning to address the cognitive needs of children

2.2. Gap Analysis:

Current solutions lack adaptability for younger age groups and do not offer dynamic, voice-based interaction that enhances learning experiences.

2.3. Related Work:

Explore existing AI chatbots and voice-driven systems in education, identifying gaps in interactivity and adaptation for young learners.

2.4. Summary:

While AI-powered educational tools offer personalized learning and interactivity, they often lack features tailored for younger learners. Addressing gaps in adaptability, engagement, and real-time learning dynamics can significantly enhance AI's role in early childhood education.

CHAPTER 3: SOFTWARE REQUIREMENT SPECIFICATION

Requirements Analysis

Introduction to SRS

This Software Requirements Specification (SRS) document provides a detailed overview of the system's requirements for an AI-based, voice-enabled educational platform tailored for children aged 3-7 years. The platform integrates voice recognition, AI chatbot responses, and text-to-speech functionalities to enhance the learning experience. The SRS captures functional and non-functional requirements, user classes, and requirement identification techniques to support the development of the system using the MERN stack and Gemini API.

User Classes and Characteristics

The users of the system are divided into the following classes, each with their distinct characteristics:

User Class	**User CharacteristicS**
Children (3-7 years)	Primary users who will interact with the system through voice queries. They require a simple, engaging, and interactive user interface with responsive feedback that is easy to understand.
Parents	Indirect users who may assist their children. They are interested in monitoring their child's learning progress and ensuring the educational content is appropriate.
Educators	They may use the system to guide children's learning and track educational progress. They require analytics and insights into children's interaction with the system.

Requirement Identifying Technique

To identify the system's requirements, the following techniques have been selected:

1. Use Case Technique

- Use Case Name: Child Query Interaction
- Associated Requirements:
- The child should be able to speak their query into the microphone.
- The system should transcribe the spoken query into text using a speech-to-text API.

- The system should analyze the query using the Gemini API and generate an accurate response.
- The system should convert the text-based response back into speech using the text-to-speech API.

Use Case Diagram:

A diagram representing the flow of interaction between the child, the system, and the voice APIs.

2. Storyboarding Technique

This technique is used to provide a graphical representation of the interaction flow between the child and the system, displaying how the child inputs a query and receives a voice-based response.

Functional Requirements

The system's functional requirements are defined by feature, with each feature broken down into its associated specific functional requirements.

Functional Requirement 1: Voice-to-Text Query

Identifier	**FR-1**
Title	Voice Query Input
Requirement	The child shall be able to speak their question into the system's microphone.
Source	System designed for voice-based interaction.
Rationale	Provides an accessible interface for young children, allowing hands-free interaction.
Business Rule	N/A
Dependencies	Speech-to-text API must be functional
Priority	High

Functional Requirement 2: Query Processing Using Gemini API

Identifier	**FR-2**
Title	Query Processing
Requirement	The system shall send the transcribed text to the Gemini API for response generation.
Source	AI-driven query analysis.

Rationale	Analyzes the child's query to generate an appropriate response.
Business Rule	N/A
Dependencies	Dependent on the successful transcription of the child's query.
Priority	High

Functional Requirement 3: Text-to-Speech Response

Identifier	**FR-3**
Title	Voice Response
Requirement	The system shall convert the AI-generated response into speech using a text-to-speech API.
Source	Speech-based interaction model.
Rationale	Provides the child with an auditory response, completing the interactive loop.
Business Rule	N/A
Dependencies	Dependent on Gemini API successfully generating a text response.
Priority	High

Functional Requirement 4: Interactive Learning Interface

Identifier	**FR-4**
Title	User Interface Interaction
Requirement	The system shall display a dynamic and engaging interface in response to the child's query.
Source	Interactive educational environment.
Rationale	Helps maintain the child's attention and enhances the learning experience.
Business Rule	N/A
Dependencies	N/A

Priority	Medium

Non-Functional Requirements

Reliability

- The system should maintain a Mean Time Between Failures (MTBF) of at least 99.9% uptime to ensure uninterrupted learning sessions.
- Mean Time to Recovery (MTTR): If the system fails, it should recover and restart operations within 5 minutes.
- A backup of children's queries and system responses will be saved every 10 minutes to ensure data consistency.

Performance

- The response time for voice-to-text conversion should be less than 2 seconds.
- The AI-based response generation should not exceed 3 seconds.
- Text-to-speech conversion should occur in under 1 second to maintain a smooth interactive experience.

Usability

- The system must be intuitive and easy to use, especially for the primary user class (children aged 3-7).
- The interface will feature large, easy-to-interpret buttons and feedback mechanisms to confirm that a child's query has been understood and is being processed.

Security

- The system shall ensure that children's data is stored securely, following privacy guidelines.
- Personal data shall not be collected without parental consent, and any information related to children's interactions will be anonymized.

Usability Requirements

Usability requirements focus on ensuring that users, especially children aged 3-7, can efficiently and effectively use the system with minimal effort and errors. These requirements guide the user interface design to create an optimal user experience.

- **Ease of Learning:** The system shall allow first-time users to interact with the platform and complete basic tasks (e.g., ask a query, receive a response) without requiring prior knowledge or training. The interface will be intuitive with visual prompts and voice-based instructions.
- **Ease of Use:** The child shall be able to interact with the system using voice commands, with a maximum of 2 interactions (e.g., speak a query, receive an answer) to get a response.
- **Error Avoidance & Recovery:** The system shall automatically detect and correct common errors in voice queries, providing suggestions or clarification prompts if it cannot understand the query. It should also allow users to repeat their query or rephrase it without restarting the entire interaction.

- **Efficiency of Interaction:** The system shall provide a response to the user query within 5 seconds, ensuring smooth and quick interaction.
- **Accessibility:** The system shall include large, clearly labeled buttons for any manual interactions and use high-contrast colors to accommodate users with visual impairments. Voice interaction will be the primary method, supporting children with limited reading skills.
- **Example Usability Requirement:** The child user shall be able to ask a question and get a voice-based answer within 2 clicks or commands.

Performance Requirements

Performance requirements define the system's operational standards, focusing on speed and efficiency.

- **Voice-to-Text Conversion Time:** The system shall transcribe a child's voice query into text in under 2 seconds.
- **Query Processing Time:** The system shall process the child's query and generate a response within 3 seconds.
- **Text-to-Speech Conversion Time:** The system shall convert the generated response back to speech in under 1 second.
- **System Response Time:** The overall time from a child speaking a query to receiving a spoken response should not exceed 6 seconds.
- **Resource Utilization:** The system shall operate efficiently within the resources available on a standard Android/iOS tablet, utilizing no more than 50% CPU and 500MB RAM at peak.

Security Requirements

Security is critical to protect both the system and the sensitive data of its users.

- **Data Protection:** The system shall encrypt all communications between the client and server, ensuring that data (e.g., child queries) is secure during transmission.
- **User Privacy:** The system shall comply with COPPA (Children's Online Privacy Protection Act) regulations, ensuring that no personal information from children is collected without parental consent.
- **System Access:** Only authorized administrators (e.g., parents or educators) shall have access to the system's backend to view usage data or modify settings.
- **Break-in Resistance:** The system shall resist unauthorized access attempts, requiring high-level skill and effort to compromise. It will log all failed access attempts and notify administrators of potential threats.

External Interface Requirements

This section describes how the system will interact with external systems, such as APIs, hardware, or other software components.

User Interface Requirements

The platform will be designed for young users with a focus on simplicity and engagement.

- **GUI Standards:** The system shall follow a consistent color scheme and font size appropriate for young children, ensuring readability and ease of interaction.
- **Icons & Buttons:** Icons will be large, colorful, and recognizable to children (e.g., microphone icon for voice input). Buttons will be clearly labeled and easy to click.
- **Navigation:** Navigation will be minimal, with voice interactions guiding the child through most actions. For manual input, shortcut keys and simple prompts will be provided for easy access.
- **Accessibility:** The interface shall support screen readers and voice navigation to accommodate children with disabilities.

Software Interface Requirements

The system will interface with several external software components:

- **Speech-to-Text API:** The system will connect to a speech recognition API (e.g., Google Speech API) to transcribe the child's spoken query into text.
- **Gemini API:** The system will communicate with the Gemini AI API to process the child's query and generate an appropriate response.
- **Text-to-Speech API**: The system will use a text-to-speech API (e.g., Google Text-to-Speech) to convert the AI-generated response back into speech.
- **Database**: The system shall store user interactions and queries in a cloud-based database (e.g., MongoDB), ensuring data persistence and availability.

Hardware Interface Requirements

The system will operate primarily on mobile devices (tablets or smartphones), interfacing with:

- **Microphone:** The system shall use the device's built-in microphone to capture the child's voice for query input.
- **Speakers:** The system will output responses through the device's speakers, ensuring clarity and volume suitable for children in quiet environments.
- **Touchscreen:** For manual interactions, the system shall support the device's touchscreen for actions like tapping buttons.

Communications Interface Requirements

The system will require network connectivity to function effectively:

- **Internet Connectivity:** The system shall connect to the internet to communicate with external APIs (e.g., Gemini API, speech-to-text, text-to-speech services). It should handle network disconnections gracefully, providing fallback options if necessary.
- **Network Protocols:** The system will use HTTP/HTTPS for secure communication between the client and server.

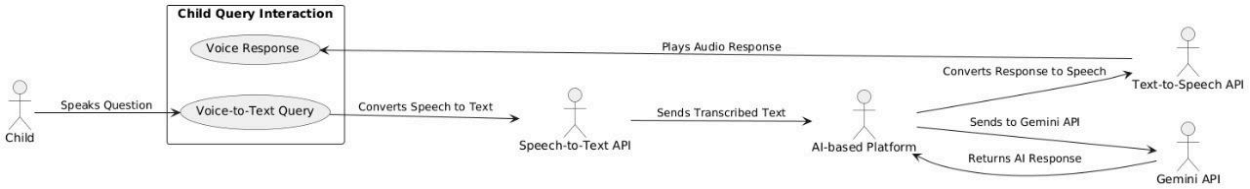
- **Data Synchronization:** The system shall synchronize data between the client application and the backend database over the internet, ensuring consistent performance across sessions.

Use Cases

Use Case UC1 - Child Query Interaction

UC Identifier	UC1
Requirements Traceability	FR-1, FR-2, FR-3
Purpose	To enable children aged 3-7 to ask questions verbally and receive accurate, voice-based responses from the educational platform.
Priority	High
Preconditions	● The system is operational. ● The child is authenticated and using the platform. ● The device's microphone is functional
Post conditions	● The child receives a voice response. ● The system logs the query and response for future reference and analytics.
Actors	● Primary Actor: Child (3-7 years) ● Secondary Actor: System (AI-based platform) ● External Actor: Speech-to-Text API, Gemini API, Text-to-Speech API
Extends	N/A
Main Success Scenario	● The child speaks a question into the device's microphone. ● The system transcribes the spoken query into text. ● The system sends the transcribed text to the Gemini API for processing. ● The AI generates a response. ● The system converts the response into speech using the Text-to-Speech API. ● The system plays the voice response back to the child.
Alternate Flows	● If the system cannot understand the spoken query, it prompts the child to repeat or rephrase their question. ● If the response generation fails, the system informs the child and suggests trying again.

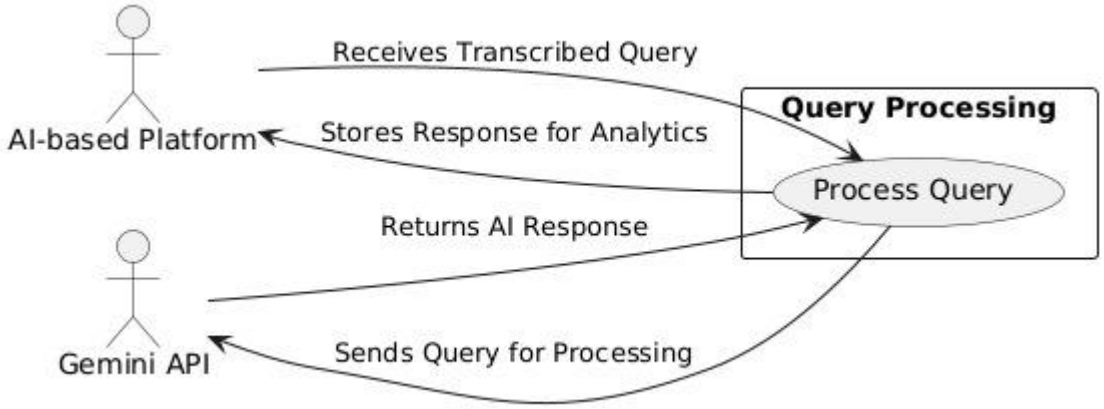
Exceptions	● The system fails to transcribe the speech due to background noise. ● The system cannot connect to the Gemini API, resulting in an inability to provide a response.
Includes	UC2 - Query Processing



Use Case Description for Use Case UC2 - Query Processing

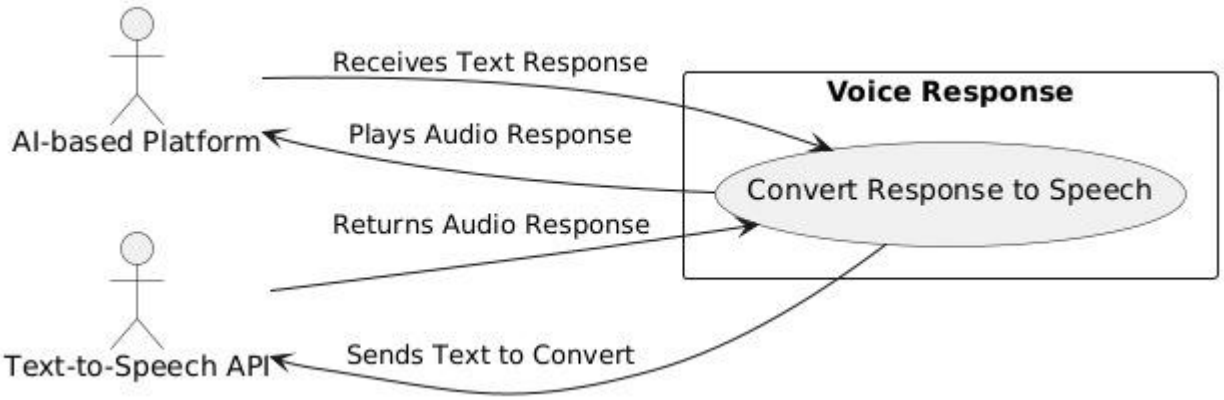
UC Identifier	UC2
Requirements Traceability	FR-2
Purpose	To process the child's transcribed voice query and generate an appropriate AI-based response.
Priority	High
Preconditions	● The child's voice query has been successfully transcribed into text. ● The system is connected to the Gemini API.
Post conditions	The AI generates a text-based response to the child's query.
Actors	● **Primary Actor:** System (AI-based platform) ● **External Actor:** Gemini API
Extends	N/A
Main Success Scenario	● The system receives the transcribed query from the voice input. ● The system sends the query to the Gemini API. ● The Gemini API processes the query and returns a response. ● The system stores the response for analytics.
Alternate Flows	● If the Gemini API returns an error, the system prompts the child to ask another question. ● If the processing takes too long, the system displays a loading indicator.
Exceptions	● Network issues prevent communication with the Gemini API.

	• The Gemini API returns an invalid response format.
Includes	N/A



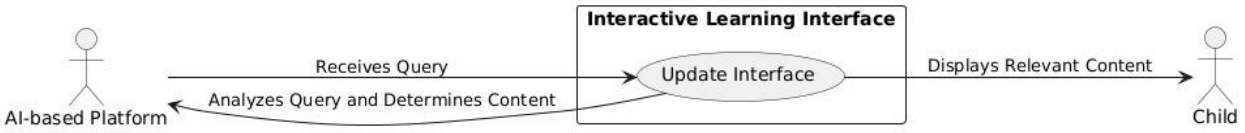
Use Case Description for Use Case UC3 - Voice Response

UC Identifier	UC3
Requirements Traceability	FR-3
Purpose	To convert the AI-generated text response into audible speech for the child to hear.
Priority	High
Preconditions	• An AI-generated text response is available. • The system is connected to the Text-to-Speech API.
Post conditions	The AI-generated response is converted to speech and played back to the child.
Actors	• **Primary Actor:** System (AI-based platform) • **External Actor:** Text-to-Speech API
Extends	N/A
Main Success Scenario	• The system receives the text response from the Gemini API. • The system sends the text to the Text-to-Speech API. • The Text-to-Speech API converts the text into speech. • The system plays the audio response back to the child.
Alternate Flows	If the Text-to-Speech API fails, the system displays an error message and suggests retrying.
Exceptions	The audio playback fails due to device issues (e.g., speaker not functioning).
Includes	N/A



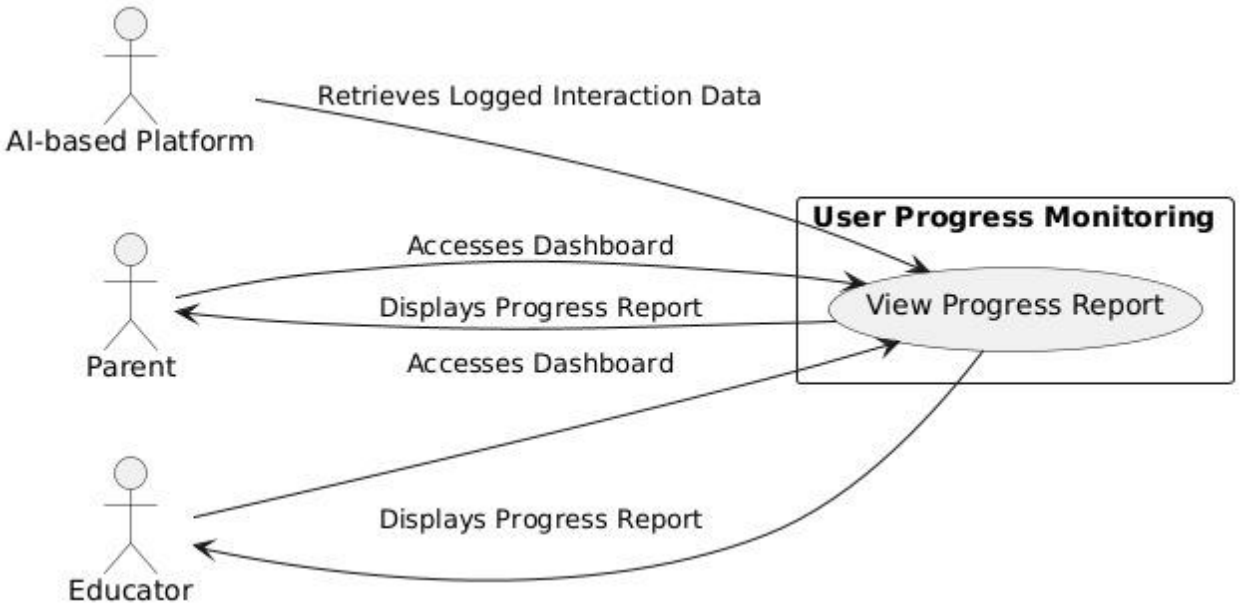
Use Case Description for Use Case UC4 - Interactive Learning Interface

UC Identifier	UC4
Requirements Traceability	FR-4
Purpose	To provide an engaging and dynamic user interface that responds to the child's queries and enhances their learning experience.
Priority	Medium
Preconditions	The system is operational and has received a query from the child.
Post conditions	The interface displays relevant content in response to the child's query.
Actors	• **Primary Actor:** System (AI-based platform) • **Secondary Actor:** Child (3-7 years)
Extends	N/A
Main Success Scenario	• The system receives a query from the child. • The system analyzes the query and determines the appropriate content to display. • The system updates the interface with engaging visuals and interactive elements. • The child interacts with the updated content.
Alternate Flows	If the content is not available, the system displays a friendly message explaining that the content is currently unavailable.
Exceptions	The system encounters a bug that prevents the interface from updating.
Includes	N/A

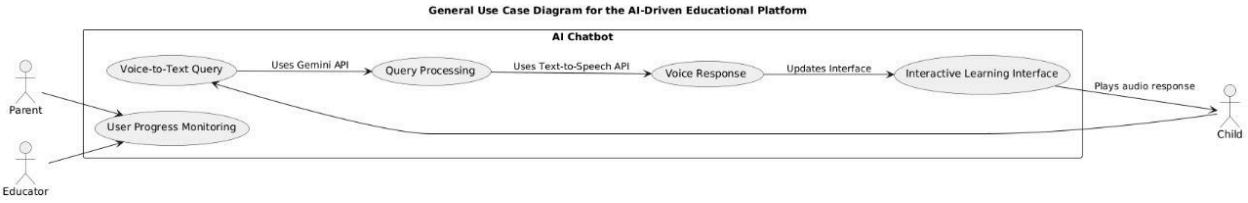


Use Case Description for Use Case UC5 - User Progress Monitoring

UC Identifier	UC5
Requirements Traceability	N/A
Purpose	To allow parents and educators to monitor children's progress and interactions with the platform, providing insights into learning habits.
Priority	Medium
Preconditions	• The child has interacted with the system. • The system has logged the child's queries and responses.
Post conditions	Parents and educators can view reports on the child's learning progress and interaction statistics.
Actors	• **Primary Actor:** System (AI-based platform) • **Secondary Actor:** System (AI-based platform)
Extends	N/A
Main Success Scenario	• The parent or educator accesses the progress monitoring dashboard. • The system retrieves the logged interaction data. • The system displays the child's progress report and analytics. • The parent or educator reviews the report to understand the child's learning journey.
Alternate Flows	If the data is not available, the system notifies the user that no data has been logged yet.
Exceptions	The system fails to retrieve data due to connectivity issues.
Includes	N/A



General Use Case Diagram



Storyboards

1. **Scene 1: Initial Interaction**
    - The child sees a welcoming screen on a tablet, with a large microphone button at the center.
    - A voice prompt asks, "What would you like to learn today?"
2. **Scene 2: Voice Input**
    - The child taps the microphone icon and asks a question like, "What is the capital of France?"
    - The microphone captures the voice and transcribes the question using the speech-to-text API.
3. **Scene 3: Processing Query**
    - The system sends the transcribed query to the Gemini API, which processes the input and generates a response.
    - A loading animation appears while the system processes the query.
4. **Scene 4: Response Delivery**
    - The system converts the response into speech using the text-to-speech API.
    - A friendly voice responds, "The capital of France is Paris."
5. **Scene 5: Interaction Feedback**

- The child can ask a follow-up question or end the interaction.
- The system logs the query and response for analytics, and the user can track progress through the platform.

Summary:

The project titled *Voice-Driven AI Chatbot for Early Childhood Education* aims to develop an interactive voice-enabled platform for children aged 3-7 years. The system uses voice recognition to capture the child's queries, processes them using the Gemini AI API, and delivers an auditory response using text-to-speech technology. The goal is to make learning more engaging and accessible for young users, ensuring the platform is intuitive and easy to use. This document outlines the Software Requirements Specification (SRS), focusing on functional and non-functional requirements, user interactions, performance, and security standards. The platform will be built using the MERN stack with voice APIs for seamless voice-based interaction.

Key features include:

- **Voice-to-text conversion** to capture the child's spoken query.
- **AI-driven query processing** to generate relevant responses.
- **Text-to-speech technology** to provide voice feedback.
- **User-friendly interface** tailored for young children with minimal manual input.

CHAPTER 4: SOFTWARE DESIGN SPECIFICATION

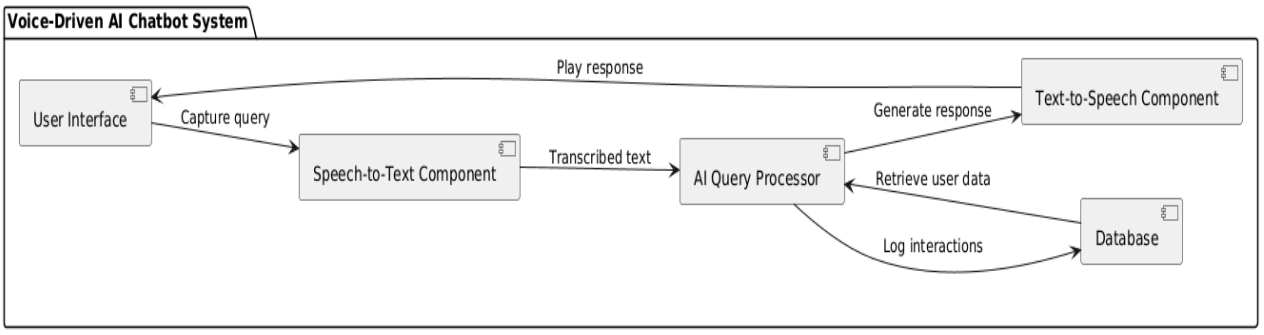
Executive Summary

The **Software Design Specification (SDS)** document outlines the technical design and implementation strategies for the **Voice-Driven AI Chatbot for Early Childhood Education**. The system is designed to provide a dynamic and engaging learning experience for children aged **3-7 years** by integrating **voice recognition**, **AI-driven query processing**, and **text-to-speech functionalities**.

This document covers the system's design considerations, architectural models, data design, and user interface specifications, ensuring that all **functional** and **non-functional requirements** identified in the **Software Requirements Specification (SRS)** are addressed.

The system leverages **modern technologies**, including the **MERN stack**, **Google APIs** for voice recognition, and **MongoDB** for efficient data storage and retrieval, to ensure scalability, responsiveness, and overall system efficiency.

System Design



1. Design Considerations

Assumptions and Dependencies

- Internet connectivity is stable for API interactions.
- Children will primarily use voice commands with minimal manual input.
- Devices include functional microphones and speakers.

Limitations

- Requires an active internet connection for API interactions.

- Background noise may affect voice query accuracy.

Risks

- API outages affecting functionality.
- Potential biases in AI-generated responses.

2. Requirements Traceability Matrix

Requirement ID	Requirement Description	Design Specification
FR-1	Convert child's spoken query into text using Speech-to-Text API	Speech-to-Text Component
FR-2	Process transcribed queries with Gemini API	AI Query Processor
FR-3	Convert text responses into voice output using Text-to-Speech API	Text-to-Speech Component
FR-4	Provide an engaging interface for children	Interactive UI
NFR-1	Ensure system uptime of 99.9%	Reliable Backend
NFR-2	Response time under 6 seconds	Optimized API Calls and Database Queries

3. Design Models

1. Architectural Design

- **Architecture Style**: Client-Server (MERN Stack).
- **Component Diagram**: Includes UI Module, Voice Processing, AI Query Processor, and Database.

2. Data Design

- **Database**: MongoDB stores user interactions, query logs, and system responses.
- **Data Dictionary**

Entity	Type	Description

User Id	String	Unique identifier for each user.
Query Test	String	Text version of the child's query.
Response Test	String	AI-generated response to the query.
Timestamp	DateTime	Time of interaction.

3. User Interface Design

Alphabetical List of System Entities or Major Data

This is essentially a **data dictionary**, where you list all the important data entities (objects, attributes, methods, etc.) in your system.

---

2. Structured Approach: Functions and Function Parameters

If your system is not object-oriented but follows a **structured programming paradigm**, you would:

- List **all functions** in your system.
- For each function, provide:
  - Its **name**.
  - A **description** of what it does.
  - The **parameters** it takes, with their **data types** and **roles**.

Example (Structured Approach):

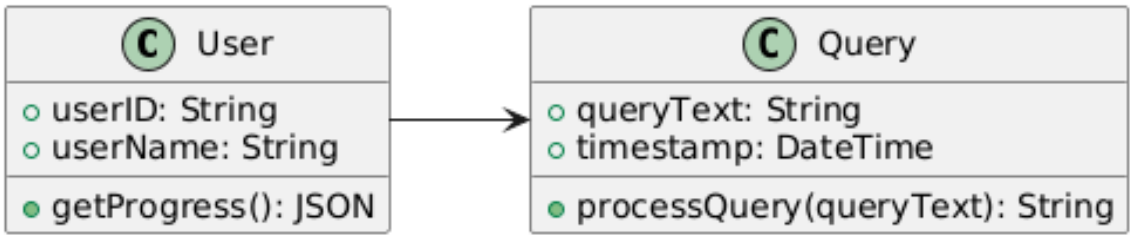
Function Name	Description	Parameters (Data Type)
processVoiceInput()	Processes the user's spoken query and converts it into text.	audioData (Binary): Captured audio input.
generateAIResponse()	Sends the transcribed text to the Gemini API and retrieves the AI response.	queryText (String): Transcribed user query.
convertTextToSpeech()	Converts the AI-generated text response into audio format.	responseText (String): Text to convert.

---

3. Object-Oriented Approach: Objects, Attributes, Methods & Method Parameters

If your system is designed using an **object-oriented approach (OO)**, you need to:

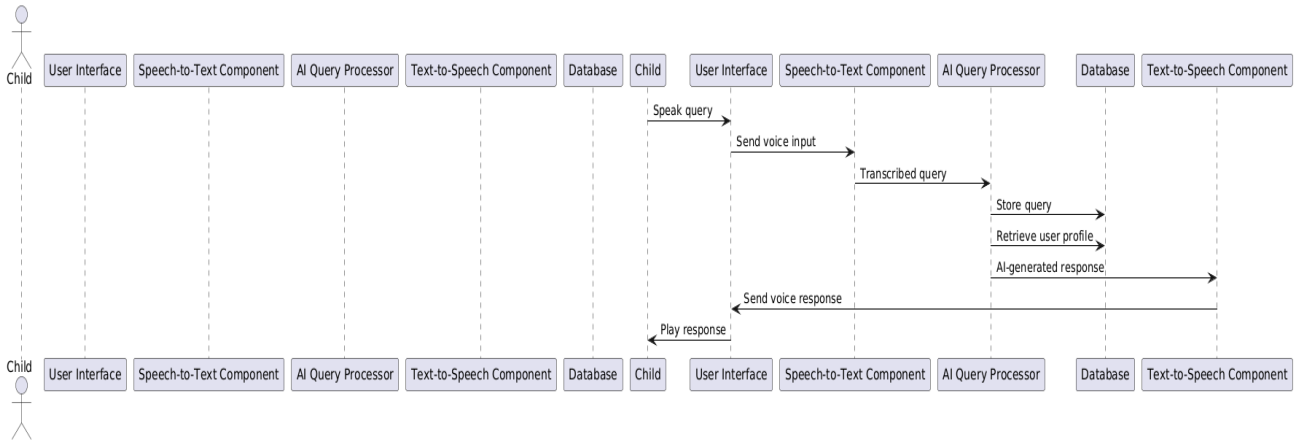
- List all **objects (classes)** in your system.



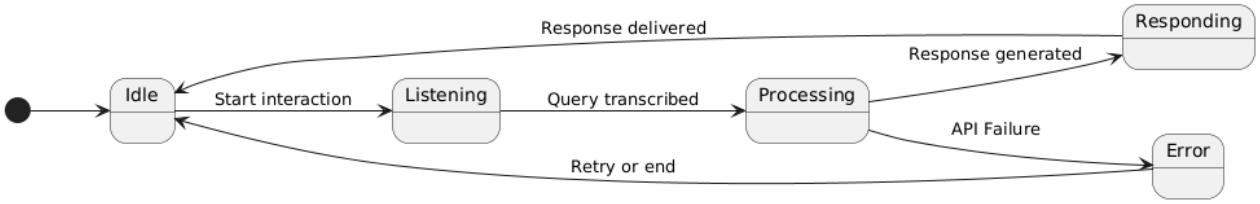
- For each object, provide:
  - Its **attributes** and their **data types**.
  - Its **methods** and their **parameters**, with descriptions of their purpose.

4. Behavioral Model

1. **Sequence Diagram:** Illustrates the interaction flow from voice input to response playback.



2. **State Diagram:** Represents system states during interaction (Idle, Processing, Responding).



- Prepare an alphabetical list of all system entities or significant data. For each entry, include the **data type** and provide a brief **description** of its purpose within the system.
- Present this information in a **two-column format**:
  - The **first column** will contain the **terminology** (e.g., object name, attribute, method, or method parameter).
  - The **second column** will provide the **description**, including data types, roles, or

any other relevant information

Structured Approach: Functions and Function Parameters:

- For systems following a structured programming paradigm, list **all functions** along with their **function parameters**. For each function, provide a description detailing its functionality, and for each parameter, specify the **data type** and its role within the function.

Object-Oriented (OO) Approach: Objects, Attributes, Methods, & Method Parameters:

- For systems using the Object-Oriented (OO) approach, list the **objects** and their **attributes**, followed by a description. Additionally, list the **methods** associated with each object, along with their **method parameters**.

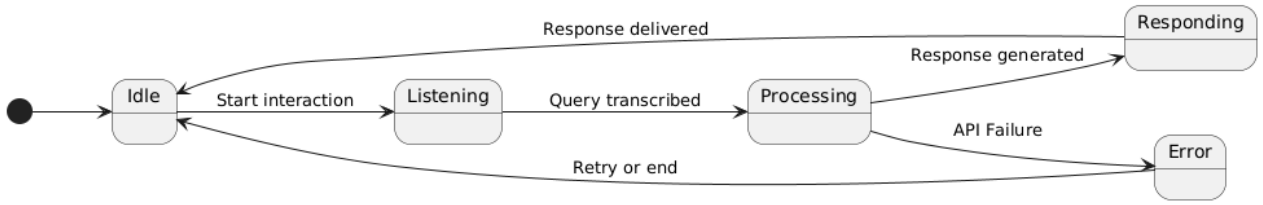
2.1 User Interface Design

Describe overview of the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user. [7]

2.2 Behavioural Model

Interaction Diagrams: Includes diagrams (Sequence or Collaboration diagram) that illustrate interactions among components

State Diagrams: Shows state transitions within the software, especially useful for systems with complex workflows or lifecycle states.[10]



3. Design Decisions

- **Architecture**: MERN stack selected for scalability and real-time updates.
- **Voice APIs**: Google APIs chosen for accuracy and multilingual support.
- **UI Framework**: React.js for a responsive and dynamic interface.

4. Summary

This SDS document provides a detailed design blueprint for the **Voice-Driven AI Chatbot for Early Childhood Education**. It aligns with the SRS requirements, focusing on scalability, performance, and usability for children aged 3-7 years. Leveraging the **MERN stack**, **Google APIs**, and **MongoDB**, the system is designed for efficient query processing and secure data management. The architectural, data, and behavioral models ensure a robust and user-friendly solution. This design forms the foundation for implementing an engaging and innovative learning platform.

References

• **MongoDB Documentation**

MongoDB, Inc. (2024). MongoDB: The Complete Developer Guide. Available at: https://www.mongodb.com/docs/

• **Node.js Documentation**

Node.js Foundation. (2024). Node.js Official Documentation. Available at: https://nodejs.org/en/docs/

• **Express.js Guide**

Express.js, Inc. (2024). Express: Fast, Unopinionated Web Framework for Node.js. Available at: https://expressjs.com/

• **React Documentation**

Facebook Open Source. (2024). React: A JavaScript Library for Building User Interfaces. Available at: https://reactjs.org/docs/getting-started.html

• **Speech-to-Text API Documentation**

Google Cloud. (2024). Google Cloud Speech-to-Text API: Documentation. Available at: https://cloud.google.com/speech-to-text/docs

• **Text-to-Speech API Documentation**

Google Cloud. (2024). Google Cloud Text-to-Speech API: Documentation. Available at: https://cloud.google.com/text-to-speech/docs

• **Gemini API Documentation**

(2024). Gemini AI API: Overview and Developer Guide. Available at: https://geminiapi.com/docs