

# Introduction to Software Engineering

# Architectural Design

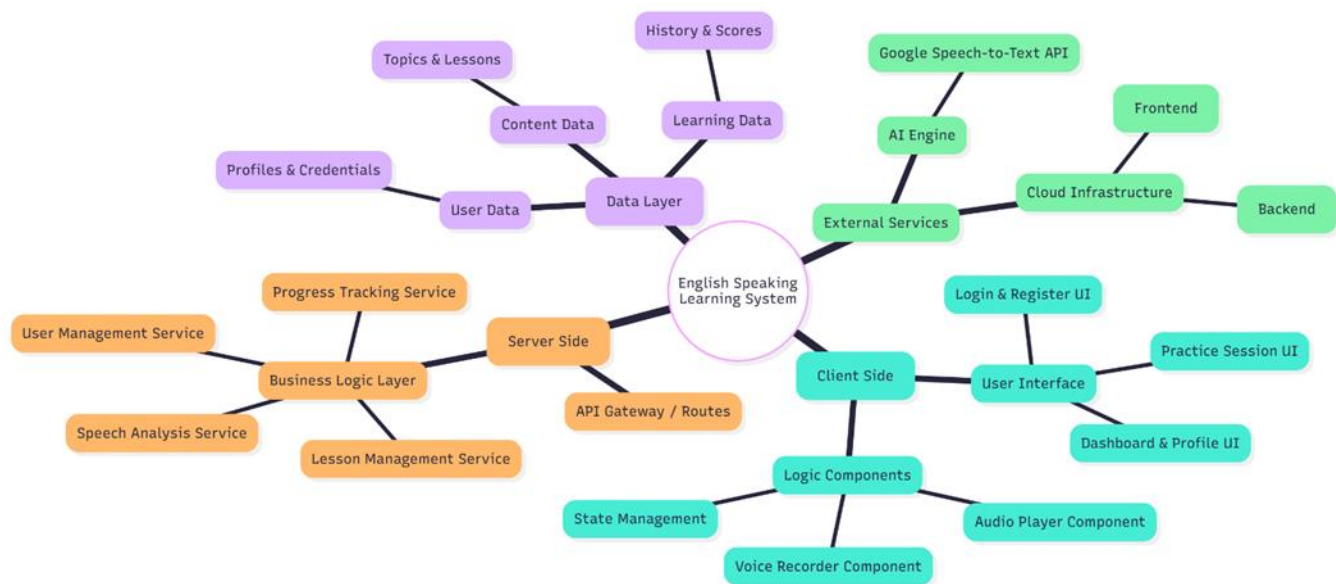


Software Engineering Department  
Faculty of Information and Technology  
University of Science

## 1.1 Architecture Diagram

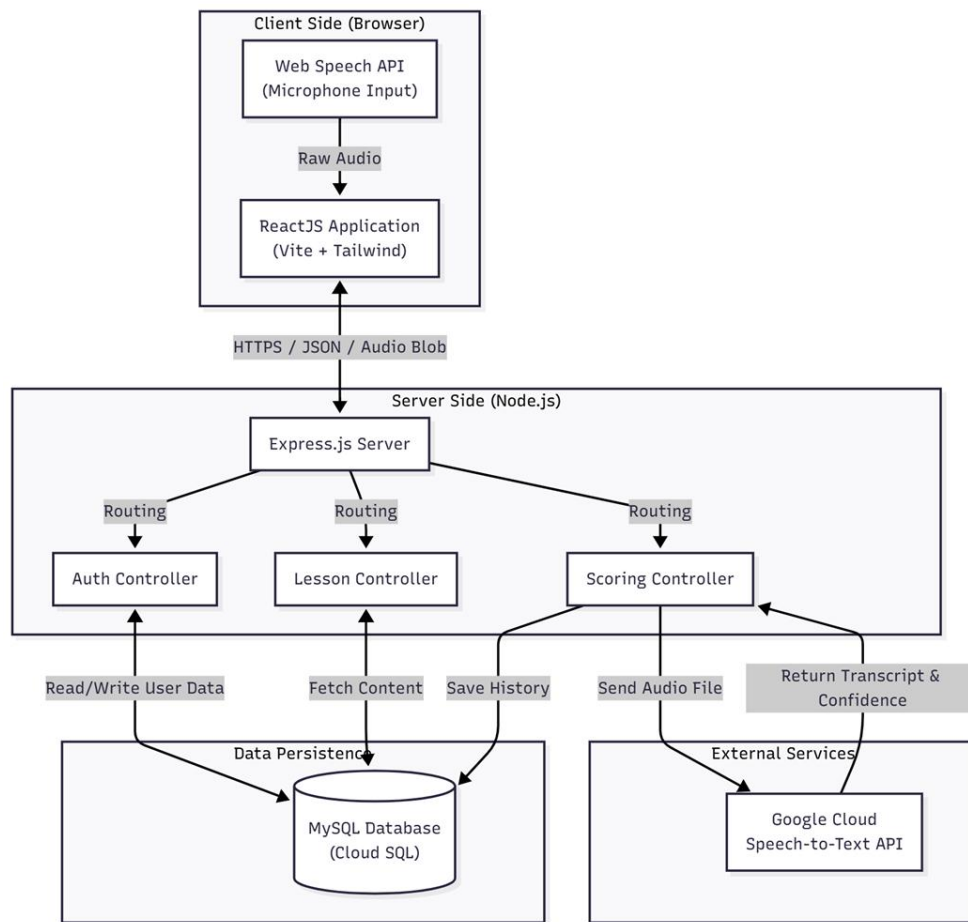
### 1.1.1 System Decomposition Tree Diagram

The system is decomposed into four main subsystems: Frontend, Backend, Database, and External Services. The decomposition tree below illustrates the hierarchical components of the English Speaking Learning System based on the functional modules defined in the proposal.



### 1.1.2 Overall System Architecture Diagram

The following diagram illustrates the high-level architecture of the system, demonstrating the relationships and data flow between the main components. The system follows a standard web application flow where the ReactJS frontend communicates with the Node.js backend via RESTful APIs, and the backend delegates voice processing to the Google Speech-to-Text API.



### 1.1.3 Architectural Design Aspects

The English Speaking Learning System employs a robust and scalable architecture designed to handle real-time audio processing and user management. The key architectural styles and patterns applied are:

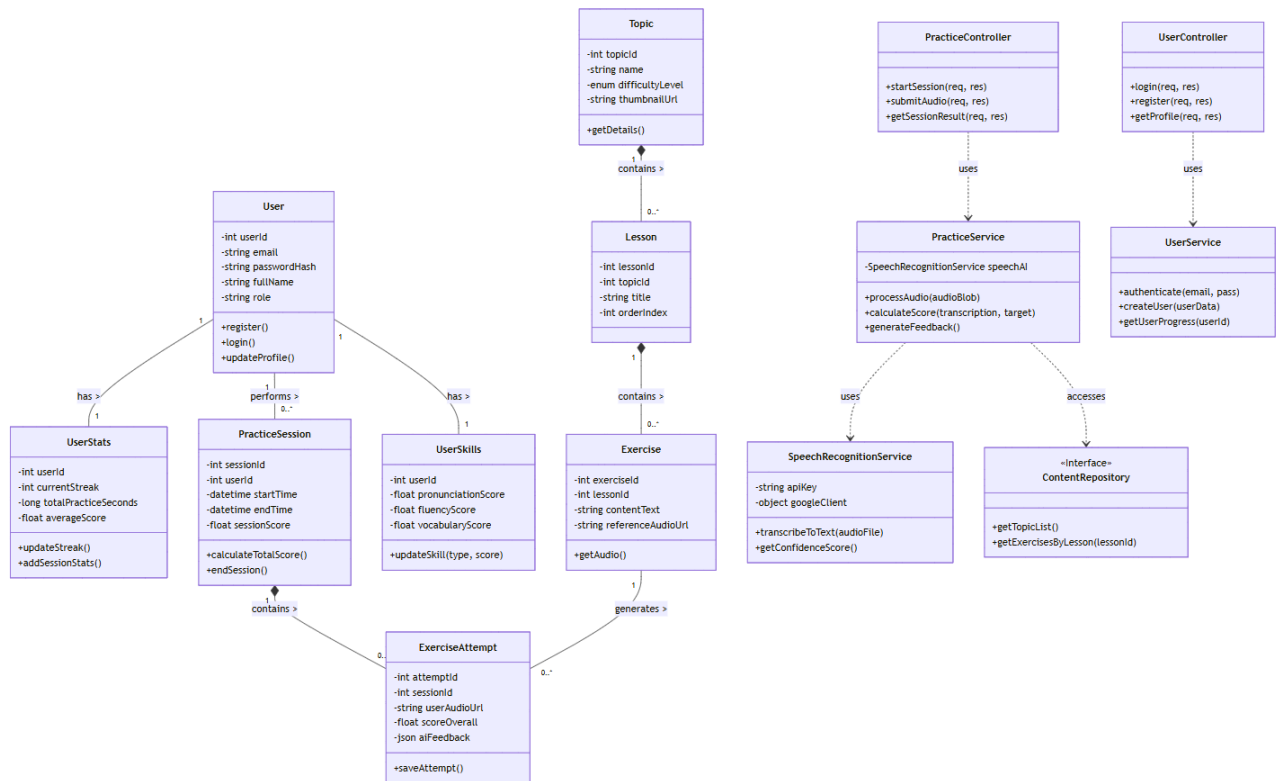
- **Client-Server Architecture**

- The system adopts a clear separation between the Client and the Server.
- **The Client (Frontend):** Built with **ReactJS**, responsible for the presentation layer, capturing user voice input via the browser's Web Speech API, and rendering feedback .
- **The Server (Backend):** Built with **Node.js (Express)**, responsible for business logic, authentication, and orchestrating calls to the database and external AI services .

- **Layered Architecture (Backend): The backend is structured into distinct layers to ensure maintainability and separation of concerns:**
  - Controller Layer: Handles incoming HTTP requests and responses
  - Service Layer: Contains the core business logic (e.g., the algorithm to compare user transcript vs. target sentence).
  - Data Access Layer: Manages direct interactions with the MySQL database.
- **Integration of Third-Party AI Services (Service-Oriented Approach)**
  - Instead of building a speech recognition engine from scratch, the system integrates the Google Speech-to-Text API. This acts as a specialized external service component. The backend acts as a secure proxy, receiving audio from the client, authenticating with Google Cloud, and processing the AI response before sending the final score back to the user.
- **Secure Data Handling**
  - Encryption: All communications between the client and server are secured over HTTPS/TLS.
  - Stateless Authentication: The system uses a stateless authentication mechanism (likely JWT - JSON Web Tokens) for user sessions, which improves scalability for the Node.js server.

## 1.2 Class Diagram

The class diagram illustrates the object-oriented structure of the English Speaking Learning System, emphasizing the separation between data entities and business logic. The domain model centers on the User class, which maintains one-to-one relationships with UserStats and UserSkills for tracking progress, and a one-to-many relationship with PracticeSessions. The learning content follows a hierarchical composition pattern where a Topic contains multiple Lessons, which in turn contain Exercises. On the architectural side, the system employs a layered approach: Controllers (e.g., PracticeController, UserController) manage incoming requests and depend on Services (e.g., PracticeService, UserService) to execute core business logic. Notably, the PracticeService integrates with the SpeechRecognitionService to handle audio processing and interacts with the ContentRepository for data access.



## 1.3 Class Specifications

### 1.3.1 Class *PracticeService*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	speechAI	private	SpeechRecogniti onService	Dependency on the AI wrapper class to handle audio transcription and scoring.
2	sessionRepository	private	Repository	Component used for interacting with the PracticeSession and ExerciseAttempt data storage.
3	contentRepository	private	ContentRepositor y	Component used to fetch the target text and reference audio for the exercise.

Seq	Operation	Modifier	Constraint	Description
1	processAudio(audioBlob)	public	Return <b>ExerciseAttempt</b> object	Takes raw user audio, orchestrates transcription and scoring via <b>speechAI</b> , saves the

				attempt record, and returns the result.
2	calculateScore(transcription, target)	private	Return float (0-100)	Compares the user's transcribed text against the exercise's target text to calculate an overall accuracy score.
3	generateFeedback()	private	Return JSON	Generates structured feedback (e.g., phoneme errors, suggested improvements) based on the detailed AI analysis.
4	startSession(userId, topicId)	public	Return <b>sessionId</b>	Creates a new record in the <b>practicesessions</b> table and returns the unique ID to the frontend.

### 1.3.2 Class User

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, Primary Key, Auto Increment	Unique identifier for the user.
2	email	private	string, Unique, Not Null	User's login email address.

3	passwordHash	private	string, Not Null	Encrypted (hashed) password.
4	fullName	private	string	User's full display name.
5	role	private	Enum('USER', 'ADMIN')	Determines access privileges within the system.

Seq	Operation	Modifier	Constraint	Description
1	register()	public	return boolean	Creates a new user record in the database.
2	verifyPassword(inputPass)	public	return boolean	Compares the input password with the stored hash.
3	updateProfile(data)	public		Updates user details like name or avatar URL.



### 1.3.3 Class Topic

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	topicId	private	int, PK, Auto Increment	Unique identifier for the topic.
2	name	private	string, Not Null	Name of the topic (e.g., "Travel").
3	difficultyLevel	private	Enum('Easy', 'Medium', 'Hard')	Difficulty level of the topic.
4	thumbnailUrl	private	string, Nullable	URL to the topic's cover image.

Seq	Operation	Modifier	Constraint	Description
1	getDetails()	public	return JSON	Returns topic details along with associated lessons.
2	getLessonOrder()	public	return List<Lesson>	Retrieves lessons sorted by their <b>orderIndex</b> .

### 1.3.4 Class Exercise

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	exerciseld	private	int, PK, Auto Increment	Unique identifier for the exercise.
2	lessonId	private	int, Foreign Key	Links to the parent lesson.
3	contentText	private	string, Not Null	The target sentence/text for the user to read/speak.
4	referenceAudioUrl	private	string, Nullable	URL for the native speaker reference audio.
5	type	private	Enum('Speaking', 'Listening')	Type of exercise.

Seq	Operation	Modifier	Constraint	Description
1	getAudio()	public	return AudioFile	Fetches the reference audio file for playback.
2	getTranscript()	public	return string	Returns the target text ( <b>contentText</b> ) for comparison.

### 1.3.5 Class ExerciseAttempt

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	attemptId	private	int, PK, Auto Increment	Unique identifier for the attempt.
2	sessionId	private	int, Foreign Key, Not Null	Links to the high-level practice session.
3	userAudioUrl	private	string, Nullable	URL to the user's recorded audio file (cloud storage).
4	scoreOverall	private	float (0-100), Not Null	Overall score generated by AI for this attempt.
5	aiFeedback	private	JSON, Nullable	Detailed AI feedback (phoneme errors, word-level confidence).

Seq	Operation	Modifier	Constraint	Description
1	saveAttempt()	public	return attemptId	Persists the attempt data (scores, audio URL, feedback) to the database.

2	getDetailedFeedback()	public	return JSON	Retrieves and formats the AI feedback for display.
---	-----------------------	--------	-------------	--

### 1.3.6 Class *SpeechRecognitionService*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	apiKey	private	string	Credentials for Google Cloud API access.
2	googleClient	private	object	Instance of the Google Speech SDK client.

Seq	Operation	Modifier	Constraint	Description
1	transcribeToText(audioFile)	public	return string	Sends the audio file buffer to the external service and returns the transcribed text.
2	getConfidenceScore()	public	return float	Retrieves the AI's confidence level associated with the transcription.
3	getPhonemeData()	public	return JSON	Fetches detailed speech metrics used

				for generating specific feedback.
--	--	--	--	-----------------------------------

### 1.3.7 Class *UserController*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userService	private	UserService	Dependency on the user business logic handler.

Seq	Operation	Modifier	Constraint	Description
1	login(req, res)	public	POST /api/login	Endpoint to handle user login request. Calls <b>UserService.authenticate()</b> .
2	register(req, res)	public	POST /api/register	Endpoint to handle new user registration.
3	getProfile(req, res)	public	GET /api/profile	Endpoint to retrieve the currently logged-in user's profile information.

### 1.3.8 Class *UserStats*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, PK, Foreign Key	Links 1-to-1 with the User.
2	currentStreak	private	int, Default 0	Current count of consecutive practice days.
3	totalPracticeSeconds	private	long, Default 0	Total time spent practicing.
4	averageScore	private	float (0-100)	Average score across all exercises.

Seq	Operation	Modifier	Constraint	Description
1	updateStreak()	public		Checks activity and increments or resets the practice streak.
2	addSessionStats(duration, score)	private		Updates total time and recalculates the running average score.

### 1.3.9 Class *UserSkills*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
1	userId	private	int, PK, Foreign Key	Links to the User.
2	skillName	private	string, PK	The specific skill being tracked (e.g., 'Pronunciation').
3	skillScore	private	float (0-100)	Current score/rating for this skill.

Seq	Operation	Modifier	Constraint	Description
1	updateSkill(type, newScore)	public		Updates the rating for a specific skill based on new attempt results.
2	getRadarData()	public	return JSON	Retrieves all skill scores formatted for the Radar Chart on the Progress Report Screen.

### 1.3.10 Class *PracticeController*

This class does not inherit from any class. It is a core service class responsible for business logic orchestration.

Seq	Property	Modifier	Constraint	Description
-----	----------	----------	------------	-------------

1	practiceService	private	PracticeService	Dependency on the core practice business logic handler.
---	-----------------	---------	-----------------	---

Seq	Operation	Modifier	Constraint	Description
1	startSession(req, res)	public	POST /api/session/start	Initializes a new practice session for the user/topic.
2	submitAudio(req, res)	public	POST /api/session/submit	Accepts the audio file blob from the client and triggers <b>PracticeService.processAudio()</b> .
3	getSessionResult(req, res)	public	GET /api/session/{id}/result	Retrieves the final results and statistics for a completed session.