Software Requirements Specification

for

DHYAN - AI Powered Robot for Visual and Audio Sensing

Version 1.0 approved

Prepared by: Qasim Ali, Ali Raza, Abdullah Awan, Sara Bint Bilal

Ghulam Ishaq Khan Institute

Supervisor **Date: 20 Nov, 2025**

Dr. Muhammad Hanif

Table of Contents

Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for **DHYAN (AI-Powered Robot for Visual and Audio Sensing)**, a low-cost, socially assistive robotic platform designed to support autism spectrum disorder (ASD) intervention in Pakistan.
This SRS covers:

- The robot's hardware-software integrated behaviour, including sensing, processing, actuation, and interaction capabilities.
- The therapist/caregiver-facing control interface.
- All therapeutic modules (e.g., joint attention, imitation etc).
- Requirements for system safety, privacy, cultural adaptation, and real-world deployment.

The document translates the full research proposal into actionable engineering requirements.

1.2 Document Conventions

- "**FR-X.Y**" denotes functional requirements.
- "**NFR-X.Y**" denotes non-functional requirements.
- "**UI-X**", "**HW-X**", and "**SW-X**" denote user, hardware, and software interface requirements respectively.
- Priorities: **H (High)**, **M (Medium)**, **L (Low)**.
- Terminology such as ASD, SAR, ROS, and TTS follow definitions in the Glossary.

1.3 Intended Audience and Reading Suggestions

This document is intended for robotics engineers, software developers, AI/ML specialists, therapists, researchers, and project managers involved in the DHYAN project. Engineers will rely on this SRS for system architecture, module specifications, and interface definitions. Therapists and ASD specialists may focus on the therapeutic modules and interaction behaviors. Researchers can use this document to understand the system for replication or academic evaluation.
Readers should begin with Sections 1 and 2 for high-level comprehension, proceed to Section 3 for system interfaces, Section 4 for detailed features, and Section 5 for non-functional requirements.

1.4 Product Scope

The scope of the **DHYAN-AI Powered Robot for Visual and Audio Sensing** is to develop a low-cost, socially assistive robotic system capable of delivering structured therapeutic interactions for children with autism spectrum disorder (ASD). The robot will combine computer vision, audio sensing, AI-driven perception, and expressive robotic behaviours to simulate the kinds of social, communicative, and cognitive learning opportunities typically offered in human-led therapy sessions. DHYAN aims to become an accessible and scalable intervention tool for Pakistani educational and clinical environments, particularly in contexts where trained therapists, diagnostic facilities, and therapeutic resources remain limited or inaccessible.

At its core, DHYAN addresses the shortage of practical, consistent, and evidence-based ASD intervention resources in Pakistan. Many families lack access to specialized therapy due to high treatment costs, geographical constraints, or long waiting lists in major urban centers. The system will allow children to engage in developmentally appropriate therapeutic activities such as joint attention, imitation, turn-taking, and emotion recognition through an interactive robotic companion. Equipped with visual sensing, DHYAN can detect a child's face, observe gaze behaviour, and identify basic emotional expressions. Through audio sensing, it can process vocal cues and deliver spoken prompts using culturally adapted Urdu and English voices.

The DHYAN platform will provide real-time interactivity by enabling children to respond to the robot's gestures, voice instructions, and social cues while receiving immediate feedback through expressive motions, affirmations, or gamified reinforcement. This promotes active learning, engagement, and social skill development beyond passive or screen-based tools. The therapist or caregiver will retain complete control of the session through a dedicated interface that allows module selection, manual overrides, difficulty adjustments, and performance monitoring.

A significant advantage of DHYAN lies in its **safety, affordability, and risk-free therapeutic environment**. Unlike traditional therapy sessions that may require specialized equipment or highly trained practitioners, DHYAN can be used in homes, schools, therapy centers, and community settings without exposing children to physical risks or requiring complex setup. The robot provides a predictable, patient, and non-judgmental interaction partner, an essential quality for children who may struggle with human social cues.

Cost-effectiveness is a central design principle. By relying on off the shelf microcontrollers, 3D-printed components, and open-source software frameworks such as ROS, DHYAN significantly lowers the financial barriers associated with conventional robotic therapy tools. This makes it feasible for public schools, low-income families, and NGOs to adopt the system without the prohibitive expenses typically associated with advanced assistive robots.

The platform is intentionally built with **future scalability** in mind. While the current version focuses on core ASD-related therapeutic modules, the modular architecture allows expansion into additional developmental domains, age groups, or disabilities. Future iterations may integrate more advanced emotion recognition, natural language understanding, adaptive learning models, or additional social-behavioural training modules. The robot can also be adapted for broader educational or healthcare applications, transforming DHYAN into a versatile and extensible assistive technology platform.

2. Overall Description

2.1 Product Perspective

DHYAN is conceived as a standalone socially assistive robotic system that integrates hardware, software, and AI-based perception modules into a unified therapeutic platform. It is not an extension of an existing commercial robot but is designed from the ground up as a contextualized solution for the Pakistani ASD intervention landscape. The robot operates as part of a broader ecosystem consisting of visual and audio sensing modules, a therapist-controlled interface, and an interaction engine capable of delivering structured behavioural therapy. The system functions autonomously during therapy sessions while allowing therapist intervention through a web-based console. DHYAN's design follows a modular architecture its perception layer (camera, microphone), cognition layer (face detection, gaze estimation, emotion recognition), and action layer (gestures, speech, reinforcement behaviours) interact through ROS-based message passing. This modularity ensures scalability and future adaptability to additional therapeutic functions or expanded subject domains.

2.2 Product Functions

DHYAN provides a comprehensive set of functions centered around ASD-related therapeutic goals. The robot performs real-time face detection, gaze estimation, and voice activity detection to monitor the child's engagement. It delivers verbal prompts and expressive gestures to support skill-building activities such as joint attention, imitation, turn-taking, and emotion identification. Through its integrated text-to-speech capabilities, DHYAN communicates using appropriate tone and phrasing. The system logs session data including response times, engagement metrics, and behavioural outcomes for later review by therapists. The therapist interface allows configuration of therapy modules, manual triggering of robot behaviours, and real-time monitoring of system status. These integrated functions work together to create a safe, interactive, and engaging environment for child participants.

2.3 User Classes and Characteristics

DHYAN is intended for three primary user classes: children with ASD, therapists, and caregivers.

- **Children with ASD** represent the core users and may exhibit limited verbal communication, atypical eye contact, and varied sensory sensitivities. DHYAN is designed to provide predictable, patient, and non-threatening interactions aligned with their developmental needs.
- **Therapists**, including clinical psychologists and special educators, interact with the therapist console to configure sessions, supervise interactions, and interpret performance data. They possess domain expertise but may not have advanced technical skills; thus, the interface must remain simple and intuitive.
- **Caregivers**, such as parents or teaching assistants, may use the system in home or school settings to supplement therapy. They require minimal training and rely on guided presets rather than detailed customization. Each user class interacts with the system differently, but all benefit from DHYAN's emphasis on accessibility, safety, and ease of use.

2.4 Operating Environment

DHYAN is designed to operate in indoor environments such as homes, therapy centers, special education classrooms, and clinics. The robot runs on a Raspberry Pi or similar single-board computing platform with Linux-based operating systems. It requires stable power supply, Wi-Fi connectivity for the therapist interface, and a quiet space for effective audio sensing. The software environment includes Linux operating system, OpenCV libraries, ONNX/TensorFlow Lite for lightweight AI inference, and a web-based backend for the control interface. The robot interacts with physical surroundings only through gestures and voice output; therefore, it does not require complex environmental mapping or navigation. The system is optimized for low-resource environments and must maintain reliable performance despite variable lighting, modest hardware capability, and limited network infrastructure

2.5 Design and Implementation Constraints

DHYAN is subject to several constraints arising from cost limitations, hardware capabilities, operational environments, and ethical considerations. From a financial standpoint, the system must remain affordable for under-resourced Pakistani schools, therapy centers, and households; therefore, the robot relies on low-cost, locally available components such as consumer-grade cameras, basic servo motors, low-power microcontrollers, and 3D-printed chassis elements. This affordability requirement significantly influences design decisions, limiting the complexity of mechanical assemblies and restricting the use of advanced sensors or high-precision actuators that are common in commercial socially assistive robots.

Hardware limitations also impose strict bounds on processing power, memory usage, thermal performance, and real-time responsiveness. Since the system operates on a Raspberry Pi–class edge device without GPU acceleration, all AI models including face detection, gaze estimation, and emotion recognition must be lightweight, quantized, and optimized for embedded inference. These constraints require careful algorithmic choices, efficient resource allocation, and fallback behaviours in case of computation overload, low light conditions, or background noise interference. Power consumption must also be minimized to allow battery-based operation without overheating or rapid discharge, further limiting continuous motor activity and intensive perception loops.

Ethical and privacy constraints further shape the system. DHYAN processes sensitive behavioural, emotional, and engagement data, requiring compliance with child-protection and data-security principles. Raw audio or video recording is disabled by default and may only be enabled with explicit informed consent. Data storage must be encrypted, anonymized, and limited to essential metrics necessary for therapy progress tracking. The system must not make clinical diagnoses or decisions; it always operates under human supervision to avoid ethical misuse or misinterpretation of behavioural cues.

Finally, environmental constraints such as unreliable Wi-Fi connectivity, fluctuating indoor lighting, and background noise in typical Pakistani classrooms or clinics influence system robustness. DHYAN must function effectively offline, adapt to various lighting conditions, filter environmental noise during audio sensing, and maintain stable performance despite inconsistent infrastructure. Taken together, these design and implementation constraints ensure that DHYAN remains affordable, safe, culturally relevant, ethically compliant, and technically feasible within Pakistan’s resource-limited educational and therapeutic settings.

2.6 User Documentation

The complete DHYAN system will include a comprehensive suite of user documentation designed to support therapists, caregivers, educators, and technical staff. This documentation will consist of a quick start guide outlining hardware setup, connectivity, and initial calibration steps; a therapist console manual detailing module configuration, session control features, and data interpretation; and a troubleshooting guide covering common operational issues such as connectivity failures, perception errors, and servo alignment problems. Additionally, maintenance instructions will be provided to help users replace modular components such as servo motors, camera modules, or 3D-printed shell parts. Contextual help will be available within the user interface through tooltips or an embedded help panel, ensuring immediate guidance during operation. All documentation will be provided in both English and Urdu to improve accessibility across diverse educational and clinical environments in Pakistan.

2.7 Assumptions and Dependencies

The development and successful deployment of DHYAN depend on several critical assumptions and external dependencies. First, it is assumed that institutions using the system—such as therapy centers, inclusive schools, or special education classrooms—possess a minimally stable indoor environment with access to standard electrical outlets and at least basic Wi-Fi or local network connectivity. Although DHYAN is designed to operate offline for core interactions, real-time session control and performance monitoring require a functional connection between the robot and the therapist console. It is also assumed that sessions will take place in reasonably quiet and well-lit rooms, as extreme variability in lighting and background noise can impair visual and audio sensing accuracy.

The system assumes that all robot-child interactions will occur under supervision, whether by therapists, caregivers, or trained educators. Supervisors should have sufficient familiarity with ASD therapy processes to interpret the child's behaviour and intervene if necessary. DHYAN is not intended to function autonomously or replace human judgment; its role is to augment professional intervention, not substitute for it. It is further assumed that children will be willing to engage with the robot and will not display severe sensory sensitivities that could be triggered by auditory or visual stimuli.

DHYAN is dependent on several open-source software frameworks, including ROS for inter-process communication, OpenCV for computer vision processing, ONNX Runtime or TensorFlow Lite for AI inference, and open-source TTS libraries for speech synthesis. Any major version changes, deprecated functions, or licensing restrictions in these frameworks may necessitate updates to the DHYAN software stack. Similarly, the project depends on the availability of hardware components i.e. servo motors, single-board computers, USB cameras, microphones, battery packs, and 3D-printing materials. Supply chain disruptions, import restrictions, or price increases could significantly impact production and scalability.

Furthermore, the project assumes ongoing access to collaborative partnerships with ASD centers, therapists, and special education institutions for pilot testing, feedback, and iterative improvement. The quality and validity of therapeutic modules depend heavily on continuous input from domain experts. Finally, ethical and privacy assumptions include the expectation that families and institutions will provide informed consent for the collection of behavioural metrics and that all data collection will comply with local privacy standards. Proper data handling practices by users—such as secure storage, restricted access, and responsible sharing are implicitly assumed for the system to remain compliant with ethical guidelines.

3. External Interface Requirements

3.1 User Interfaces

The DHYAN system provides a therapist-facing, browser-based graphical user interface designed to facilitate seamless configuration, control, and monitoring of therapy sessions. The interface is responsive and compatible with desktops, laptops, and touchscreen tablets, as these devices are commonly available in Pakistani schools and therapy centers. The UI is structured into distinct sections, including a dashboard displaying module options; a live session screen showing real-time engagement metrics such as detected face presence, gaze direction, emotion classification, and voice activity; and a settings panel allowing customization of language, volume, gesture intensity, and session duration.

To accommodate therapists and caregivers with varying levels of technical expertise, the UI incorporates large icons, bilingual labels (English and Urdu), and logically grouped controls. Real-time robot status indicators such as battery level, CPU temperature, camera connectivity, and network strength are displayed prominently to prevent disruptions during sessions. Pop-up alerts notify users of critical issues like servo overload, poor lighting, or microphone malfunction. Contextual help is embedded in the form of tooltips, modal guides, and a support panel linking to documentation and troubleshooting steps. The design follows principles of usability, accessibility, and minimal cognitive load, ensuring that clinicians can focus on the child rather than managing the interface.

3.2 Hardware Interfaces

The DHYAN robot integrates multiple hardware components that work together to support perception, computation, actuation, and interaction. The system uses a hybrid computing architecture consisting of a Raspberry Pi 5 and an NVIDIA Jetson Nano, connected through a local network or a direct wired interface. The Raspberry Pi serves as the primary controller, responsible for managing ROS nodes, coordinating robot behaviours, controlling servo motors, processing audio input, and communicating with the therapist interface. The Jetson Nano functions as a dedicated AI inference accelerator, executing computationally intensive tasks such as real-time face detection, gaze estimation, and emotion recognition using GPU-optimized deep learning models.

A USB webcam acts as the main visual sensing device, providing a live video stream to the Jetson Nano for accelerated GPU-based inference. The webcam connects via USB 2.0 or 3.0 and exposes video frames using standard Linux drivers to ensure compatibility with both the Jetson Nano and the Raspberry Pi. A digital microphone, also connected through USB, captures audio input including vocal activity, keywords, and environmental noise. Audio data is processed on the Raspberry Pi for voice activity detection, turn-taking recognition, and simple keyword spotting. A portable speaker connected through a 3.5mm audio jack or USB output provides text-to-speech output, sound cues, and reinforcement effects during therapy modules.

The system uses PWM-controlled servo motors to enable head movements and expressive motions. All servos interface with the Raspberry Pi via GPIO or I2C, and motion ranges are limited to ensure safe operation around children. Both computing boards and peripheral components are mounted within a 3D-printed chassis designed for durability and child-safe interaction. The structure provides isolated internal compartments for wiring and electronics, preventing accidental tampering.

Power is supplied through separate regulated power modules: the Raspberry Pi and Jetson Nano require independent, stable 5V supplies capable of delivering adequate current, while servo motors are powered through an isolated high-current 5–6V supply to prevent electrical noise from affecting computation. Optional components such as LEDs, cooling fans, or additional sensors may interface through the GPIO pins or USB ports of either board. The hardware interface design emphasizes modularity, enabling individual components such as the webcam, servos, microphone, or Jetson Nano to be replaced, upgraded, or expanded without redesigning the entire system.

3.3 Software Interfaces

The DHYAN system employs a distributed software architecture built across both the Raspberry Pi and the NVIDIA Jetson Nano. These two computing units communicate through ROS topics, services, and a local network bridge. The Jetson Nano executes GPU-accelerated AI inference tasks—such as face detection, gaze estimation, and emotion recognition using ONNX Runtime. The results of these inference tasks are published back to the Raspberry Pi through ROS topics such as **/vision/face_data**, **/vision/gaze_output**, and **/vision/emotion_output**.

The Raspberry Pi handles behaviour control, gesture execution, session management, and communication with the therapist console. ROS topics on the Pi include **/robot/motion_cmd**, **/audio/vad**, **/tts/request**, and **/session/log**, while ROS services support module switching, session start/stop, and real-time adjustments. The Pi also manages the audio pipeline using ALSA/PyAudio interfaces for microphone input and speaker output.

The therapist UI communicates with the robot through a REST API and WebSocket endpoints hosted on the Raspberry Pi. The REST API supports system configuration, module selection, and parameter updates, while Web Sockets provide low latency streaming of telemetry such as engagement metrics, detected emotions, robot health status, and session data.

The system also relies on several external software libraries, including OpenCV (video processing), TensorRT/TensorFlow Lite (model inference), PyAudio (audio input processing), and Google TTS for speech synthesis. Session data is stored in a lightweight SQLite or JSON-based log structure managed by the Raspberry Pi. Each software interface adheres to strict message schemas and error-handling rules to maintain reliability in real-time human–robot interaction.

3.4 Communications Interfaces

DHYAN uses a local network communication model where all data exchange between the Jetson Nano, Raspberry Pi, and therapist console occurs within a secure LAN environment. The Raspberry Pi acts as the primary communication node, exposing REST endpoints, WebSocket channels, and ROS master services. Communication between the Raspberry Pi and therapist console takes place over Wi-Fi using HTTP/HTTPS for command requests and WebSocket for real-time telemetry.

The interface supports bidirectional communication: the therapist console sends commands such as “start session,” “trigger gesture,” or “change module,” while the robot streams back continuous perception data, robot status, and behavioural metrics. All WebSocket messages are structured, timestamped, and optimized for minimal latency, ensuring smooth human–robot interaction.

The Jetson Nano and Raspberry Pi communicate internally. The Jetson publishes high-speed perception results to the Pi at approximately 10–30 FPS depending on module load, and the Raspberry Pi subscribes to these topics to trigger robot behaviours. The communication layer includes heartbeat signals, retry mechanisms, and automatic reconnection logic to handle intermittent Wi-Fi or internal network disruptions.

Security is maintained through optional HTTPS support, local-only IP access, and configurable authentication tokens on the therapist interface. Because the system operates in classrooms and therapy centers with limited internet availability, DHYAN does not require cloud connectivity during therapy sessions. All sensitive behavioural data remains on the local network, ensuring privacy compliance. The communication stack is optimized for low bandwidth, making the system resilient even in environments with weak or unstable network infrastructure.

4. System Features

4.1 Visual Perception Feature (Face Detection, Gaze Tracking, Emotion Recognition)

4.1.1 Description and Priority

This feature enables DHYAN to perceive and interpret the child's visual behaviour through the USB webcam streamed to the Jetson Nano. The perception subsystem performs face detection, gaze estimation, head orientation detection, and basic emotion recognition using GPU-accelerated AI models. These capabilities are essential for all therapy modules because they allow the robot to determine engagement, monitor task compliance, and adjust feedback. This is a **High Priority** feature as visual sensing forms the foundation of DHYAN's behaviour.

4.1.2 Stimulus/Response Sequences

When the camera captures a frame, the Jetson Nano processes it using TensorRT-optimized models. If a face is detected, the system determines whether the child is looking at the robot or toward a referenced stimulus (e.g., an object in the joint attention task). The Raspberry Pi receives inference results and triggers the next appropriate robot behaviour—such as delivering praise, repeating an instruction, or adjusting motion.

4.1.3 Functional Requirements

REQ-4.1.1: The system shall detect human faces in real time at a minimum of 10 FPS under indoor lighting conditions.

REQ-4.1.2: The system shall estimate gaze direction with at least three states: *towards robot*, *towards stimulus*, *away*.

REQ-4.1.3: The system shall classify emotions (happy, sad, angry, surprised, neutral) based on facial features.

REQ-4.1.4: The Jetson Nano shall process each frame and publish results to ROS topics within 150 ms.

REQ-4.1.5: The system shall handle degraded lighting by reducing resolution or switching to simplified models.

REQ-4.1.6: All perception outputs shall be streamed to the therapist console in real time.

REQ-4.1.7: The system shall log perception data (face present, gaze direction, emotion) with timestamps.

4.2 Audio Sensing & Voice Activity Detection (VAD)

4.2.1 Description and Priority

This feature enables DHYAN to detect when the child is speaking or producing sound. Through a USB microphone connected to the Raspberry Pi, the robot determines voice activity and uses it to facilitate turn-taking, response timing, and natural conversation flow. This is a **High Priority** feature for interactive behaviour.

4.2.2 Stimulus/Response Sequences

When the child speaks, VAD triggers a signal indicating vocal activity. The robot immediately pauses any ongoing speech to avoid overlapping, waits for the child to finish, and then proceeds with its next action or response.

4.2.3 Functional Requirements

REQ-4.2.1: The system shall process audio input continuously at intervals no greater than 100 ms.

REQ-4.2.2: The VAD system shall correctly identify speech segments with at least 85% accuracy.

REQ-4.2.3: The robot shall stop its speech output within 200 ms when child speech is detected.

REQ-4.2.4: The system shall be able to detect prolonged silence and prompt the child accordingly.

REQ-4.2.5: The therapist console shall display voice activity status indicators (e.g., *child speaking*).

REQ-4.2.6: All vocal-response events shall be logged with timestamps.

4.3 Joint Attention Module

4.3.1 Description and Priority

Joint attention—the ability to share focus on an object—is a critical developmental skill for children with ASD. This module trains the child to follow the robot's gaze or gestures toward a target. It is a **High Priority** clinical feature.

4.3.2 Stimulus/Response Sequences

The robot verbally prompts ("Look over here!") and performs a head turn toward a predefined target. The perception system evaluates whether the child follows the gesture. If successful, the robot provides immediate reinforcement (clapping gesture, praise phrase). If the child does not respond, the robot repeats or simplifies the prompt.

4.3.3 Functional Requirements

REQ-4.3.1: The robot shall issue clear verbal instructions directing attention to a target.
REQ-4.3.2: The robot shall perform a corresponding gesture (head turn or pointing) within 300 ms.
REQ-4.3.3: The system shall detect whether the child shifts gaze within a configurable time interval.
REQ-4.3.4: The robot shall provide positive reinforcement upon successful attention shift.
REQ-4.3.5: The system shall adjust prompt difficulty based on previous trial accuracy.
REQ-4.3.6: Each trial's result (success or failure) shall be logged with timestamps.

4.4 Imitation Module

4.4.1 Description and Priority

Imitation is essential for learning social and motor behaviours. This module prompts the child to mimic gestures performed by the robot. It is a **Medium Priority** but important for enhancing engagement and motor imitation skills.

4.4.2 Stimulus/Response Sequences

The robot performs a gesture (e.g., waving, nodding, clapping) and waits for the child to imitate. The therapist observes the child and records whether the imitation is correct or needs improvement.

4.4.3 Functional Requirements

REQ-4.4.1: The robot shall perform predefined gestures with smooth motion

REQ-4.4.2: The therapist shall be able to mark imitation accuracy through the console.

REQ-4.4.3: The robot shall reinforce correct imitation with praise or celebratory motions.

REQ-4.4.4: The system shall allow configurable gesture sets of varying difficulty.

REQ-4.4.5: The system shall store trial results, including gesture type and therapist evaluation.

4.5 Emotion Recognition & Expression Training

4.5.1 Description and Priority

This module teaches children to identify and express basic emotions. It is a **Medium Priority** feature aimed at emotional learning and social communication.

4.5.2 Stimulus/Response Sequences

The robot may ask the child to display a specific emotion (“Show me a happy face”) or ask the child to identify the emotion being displayed by the robot. The Jetson Nano evaluates the child’s expression and provides appropriate feedback.

4.5.3 Functional Requirements

REQ-4.5.1: The robot shall verbally request the child to mimic or identify an emotion.

REQ-4.5.2: The system shall compare detected facial expression with the requested emotion.

REQ-4.5.3: The robot shall provide positive feedback for correct responses and corrective encouragement for incorrect attempts.

REQ-4.5.4: The system shall maintain a record of emotion tasks, including accuracy and latency.

REQ-4.5.5: The robot shall adapt difficulty based on child progress (e.g., increasing emotion set).

4.6 Therapist Console & Session Control

4.6.1 Description and Priority

This feature provides therapists with full administrative and operational control. It is a **High Priority** feature because therapy must always remain supervised.

4.6.2 Stimulus/Response Sequences

Therapists select modules, begin sessions, trigger manual commands, and monitor child responses. The system provides real-time visualization of perception outputs and robot status.

4.6.3 Functional Requirements

REQ-4.6.1: The console shall allow start, pause, resume, and stop functionality.

REQ-4.6.2: The therapist shall configure module settings (difficulty, duration, prompt style).

REQ-4.6.3: The console shall display real-time perception outputs (face, gaze, emotion, VAD).

REQ-4.6.4: The therapist shall trigger manual commands (speak, gesture, repeat prompt).

REQ-4.6.5: The console shall display system health metrics (CPU/GPU temperature, battery, network).

REQ-4.6.6: The console shall provide alerts for hardware errors or safety triggers.

4.7 Logging, Storage & Data Management

4.7.1 Description and Priority

The logging system captures quantitative and qualitative data from therapy sessions. It is a **High Priority** feature for research, monitoring progress.

4.7.2 Response Sequence

4.7.3 Functional Requirements

REQ-4.7.1: The system shall record all trial-level data (timestamps, successes, responses).

REQ-4.7.2: The system shall store session summaries in an encrypted local database.

REQ-4.7.3: The robot shall allow exporting session data in CSV and JSON formats.

REQ-4.7.4: The system shall anonymize all child identifiers before export.

REQ-4.7.5: The system shall not record raw video or audio unless explicitly authorized by consent.

REQ-4.7.6: Logs shall include system health information during the session.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

DHYAN must operate in real time to maintain natural and engaging interaction with children. The Jetson Nano should process visual inference tasks such as face detection, gaze estimation, and emotion recognition at a minimum of 10–20 frames per second to ensure accurate and timely behavioural responses. Audio sensing on the Raspberry Pi, including voice activity detection, must respond within 150–300 milliseconds to support smooth turn-taking and prevent the robot from interrupting the child. Text-to-speech output should begin playback within one second of receiving a TTS request to maintain conversational flow. Servo motor movements must execute within predictable time intervals, with gesture latency not exceeding 200–300 milliseconds after a command is issued. The system should remain fully functional for at least 2–3 hours on battery power and must maintain stable thermal performance, ensuring that neither the Raspberry Pi nor Jetson Nano exceeds safe operating temperatures during extended sessions.

5.2 Safety Requirements

Given that DHYAN interacts directly with children, safety is a critical priority. All servo motors must operate within predefined torque limits to prevent accidental injury from sudden or forceful movements. The robot's 3D-printed chassis should have smooth, rounded edges with no exposed wiring or mechanical components. Power systems must include over-current and thermal protection to prevent overheating or electrical hazards. DHYAN must restrict gesture ranges to avoid contact with the child during close-proximity interactions. Auditory output levels must remain below harmful decibel limits to protect children with sensory sensitivities. The robot should enter a safe state disabling movement and voice output if critical faults occur, such as sensor failures, low battery levels, servo overheating, or communication loss with the therapist console.

5.3 Security Requirements

DHYAN must ensure the confidentiality and integrity of all collected behavioural and session data. Communication between the therapist console and robot should be secured through HTTPS and local-network access restrictions. Authentication tokens or passwords must be required for accessing the therapist interface to prevent unauthorized control. Sensitive data including engagement metrics, emotional responses, and session logs must be stored locally on the Raspberry Pi in encrypted form to prevent misuse or unauthorized access. No raw audio or video should be recorded unless explicit informed consent is provided. External network communication is disabled during therapy sessions to minimize data exposure risks. All modules must follow strict access-control rules to ensure that only authorized supervisors can start, stop, or modify sessions.

5.4 Software Quality Attributes

DHYAN must exhibit high reliability, maintaining consistent performance throughout therapy sessions without crashes or perceptible delays. The robot's architecture must be modular, allowing individual components such as perception nodes, gesture controllers, or TTS services to be updated or replaced without affecting the rest of the system. Usability is a priority; the therapist interface must be intuitive, bilingual, and accessible to users with limited technical expertise. Maintainability is supported through clear documentation, modular code, and standardized ROS message structures. Portability is achieved by ensuring compatibility with common embedded Linux environments, allowing future upgrades to alternate hardware like Jetson Orin Nano or Raspberry Pi 5. Robustness is required so the system can gracefully degrade under poor lighting, high noise levels, or dropped network connections while continuing to provide core functionality.

5.5 Business Rules

DHYAN must always operate under human supervision and cannot initiate therapy sessions autonomously. Only registered therapists or caregivers may access the control interface and configure intervention modules. The system must follow ethical guidelines governing child–robot interactions, including restrictions on data storage and mandatory consent procedures. All behaviour modules must operate within pre-approved therapeutic frameworks, ensuring consistency with ASD intervention best practices. Updates to modules, AI models, or robot behaviours must undergo validation by an ASD specialist before deployment in clinical or educational settings.

6. Project Design

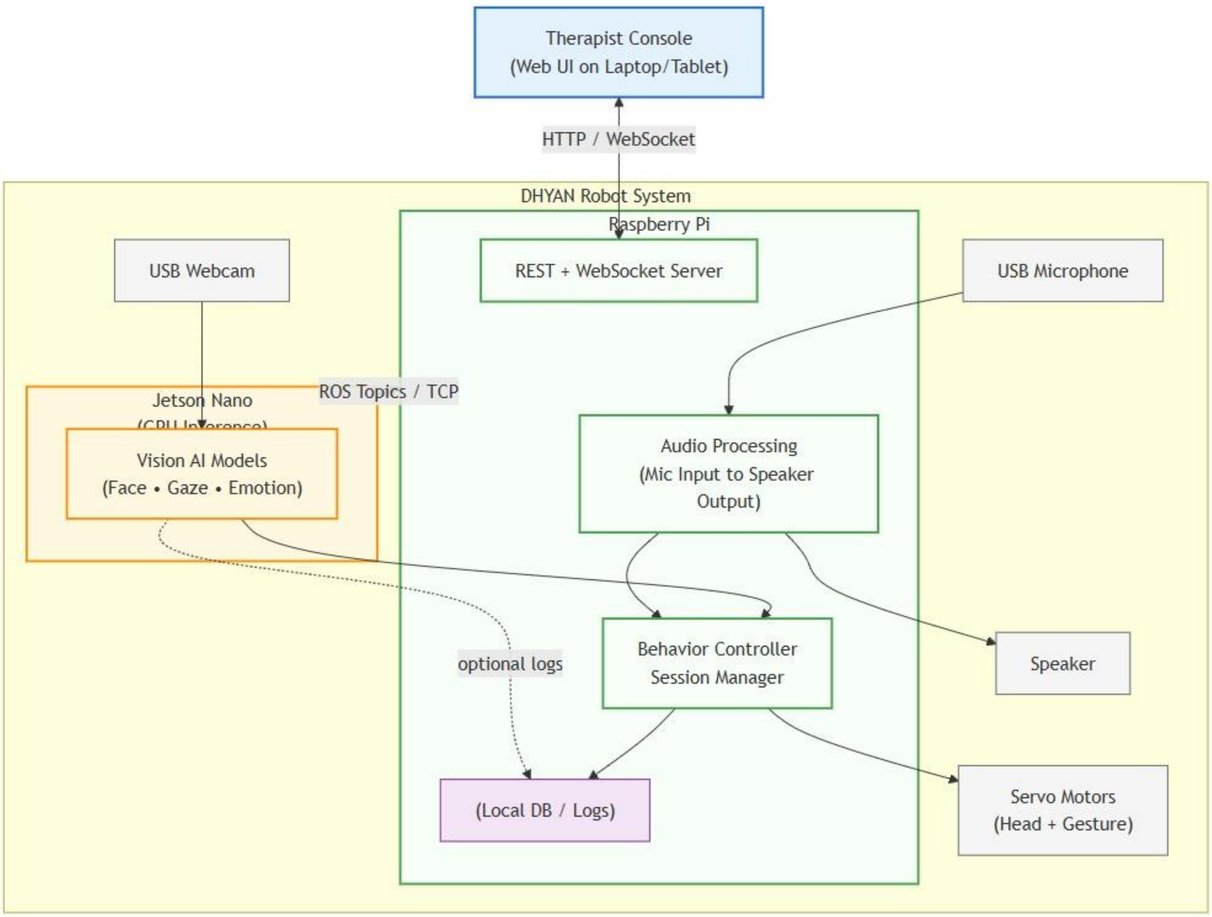


Figure 1: System Architecture Diagram

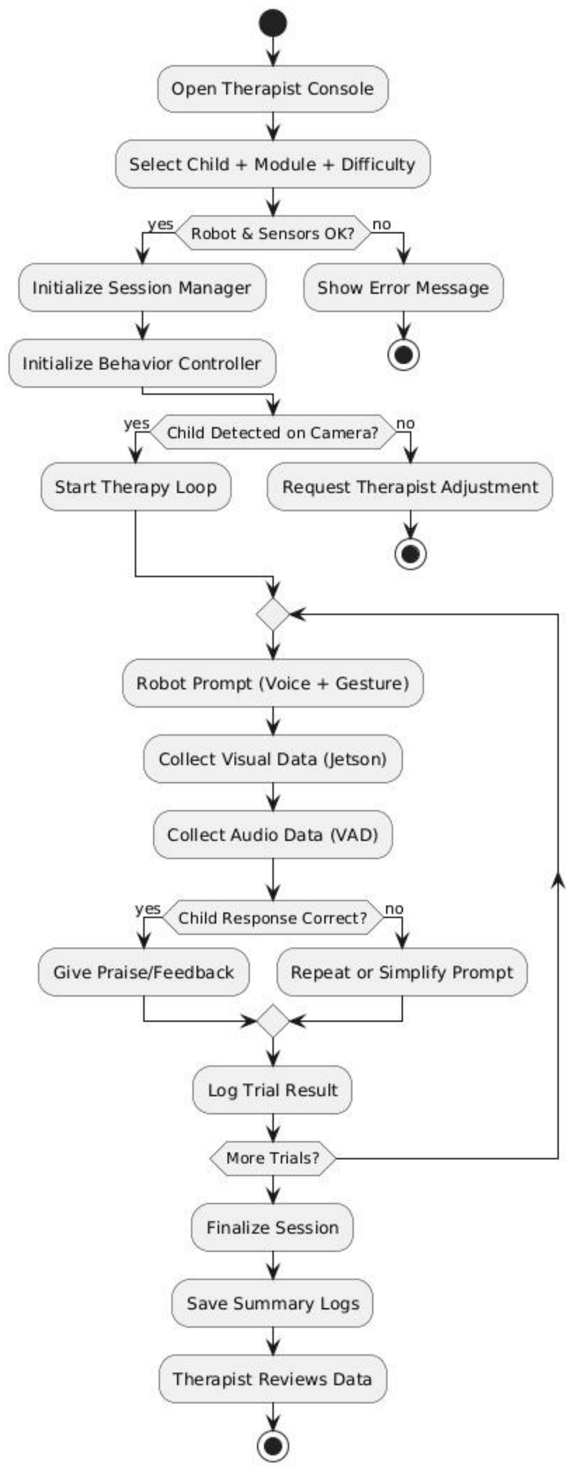


Figure 2: Activity Diagram

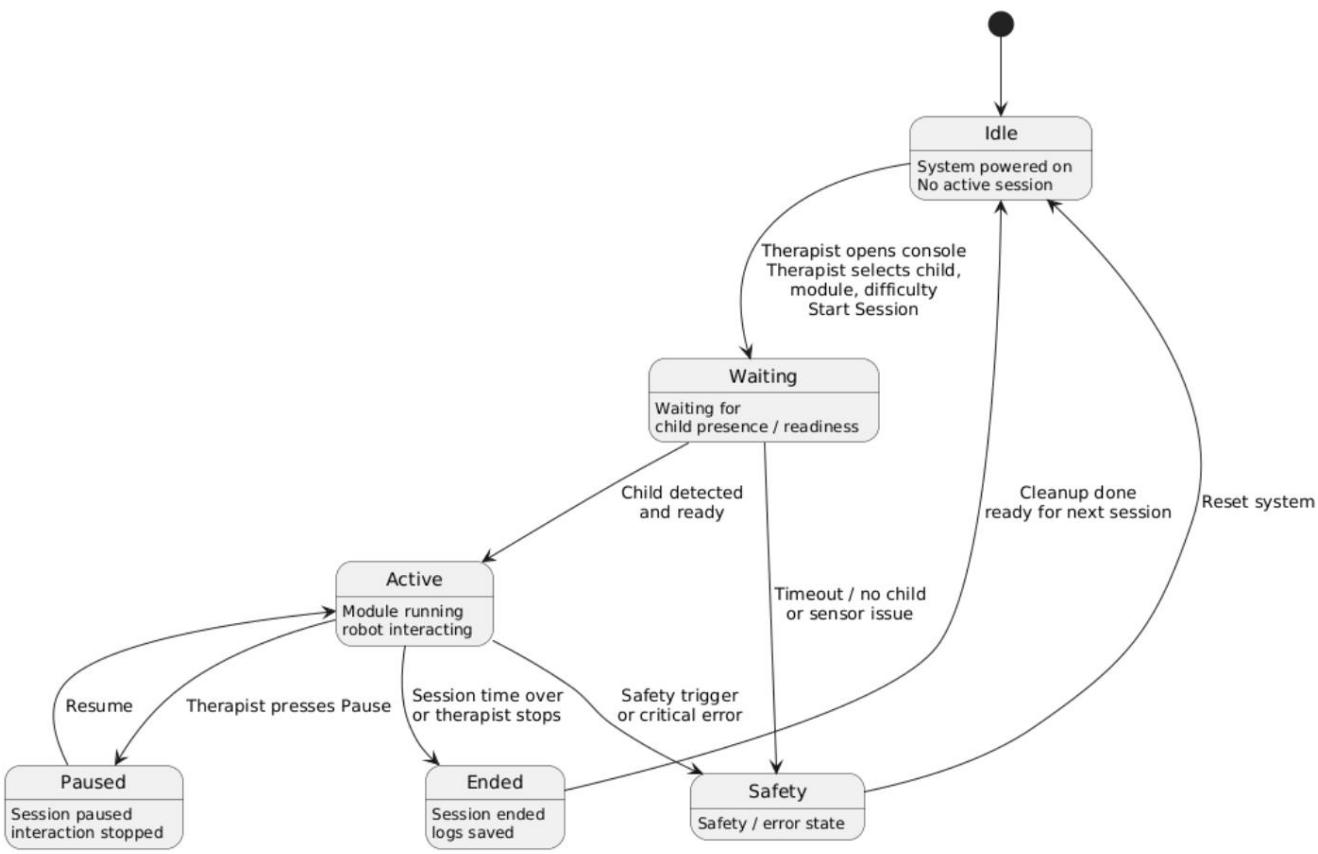


Figure 3: State Machine Diagram

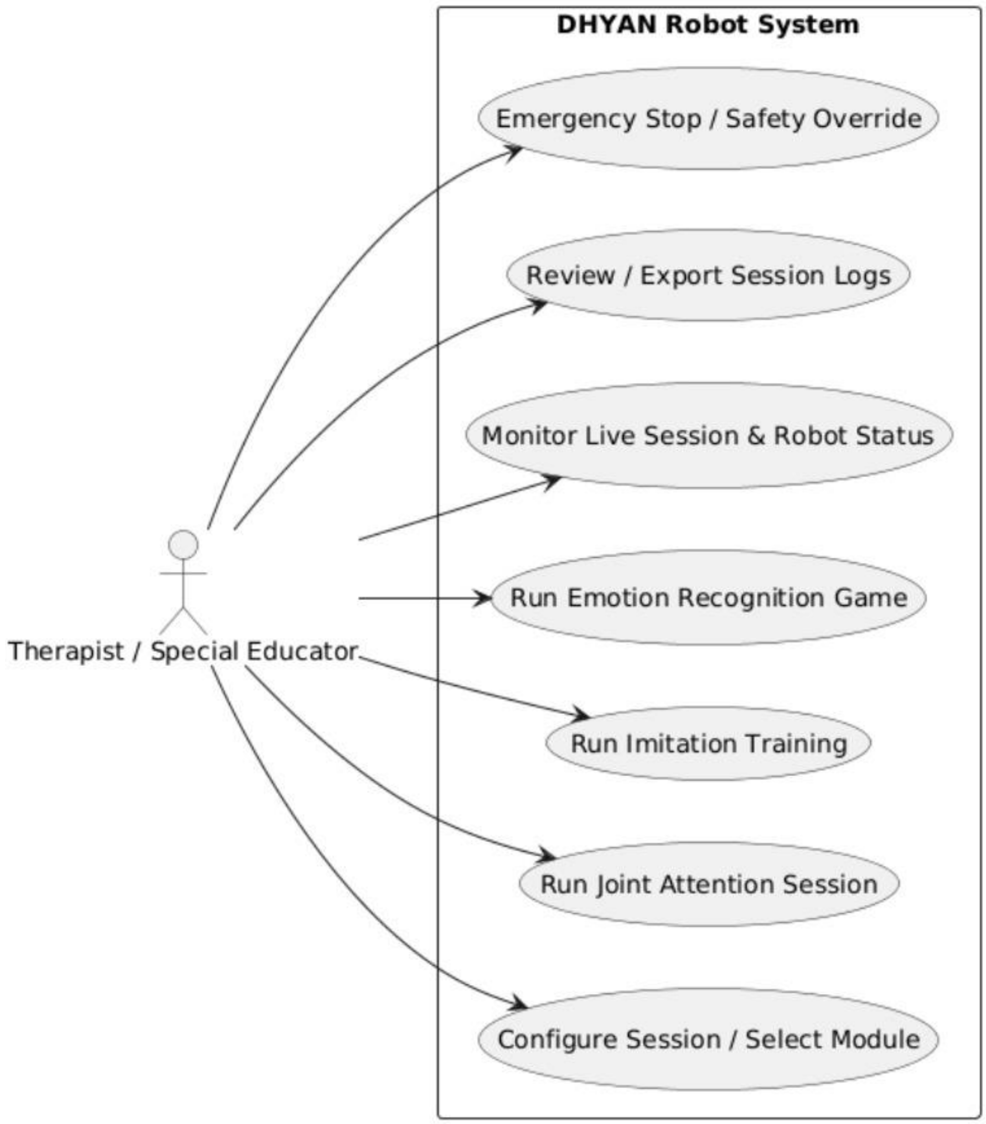


Figure 4: Use Case Diagram

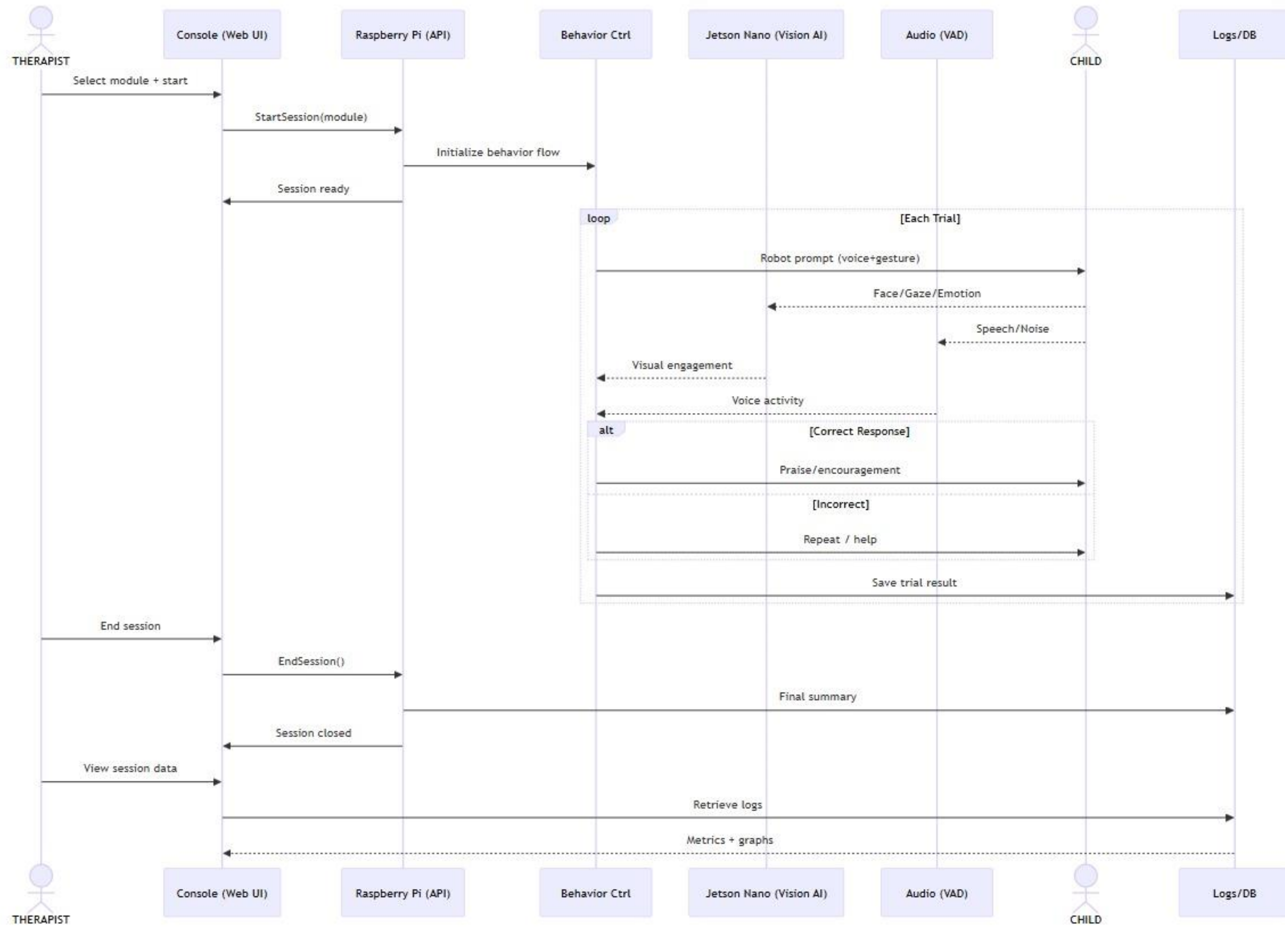


Figure 5: Sequence Diagram

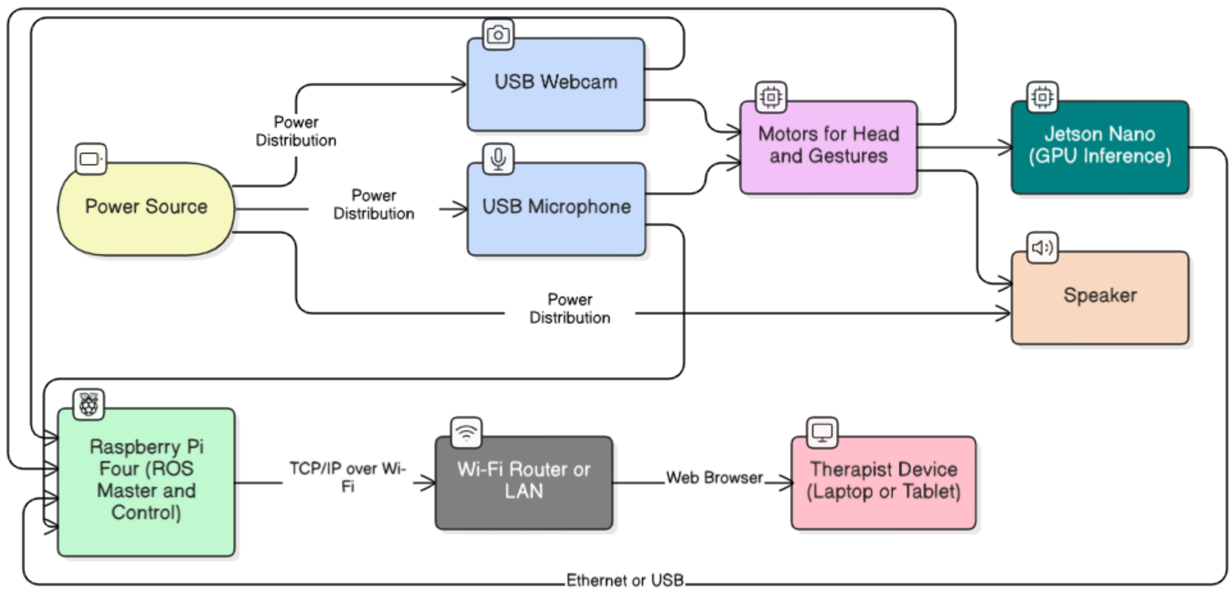


Figure 6: Deployment Diagram

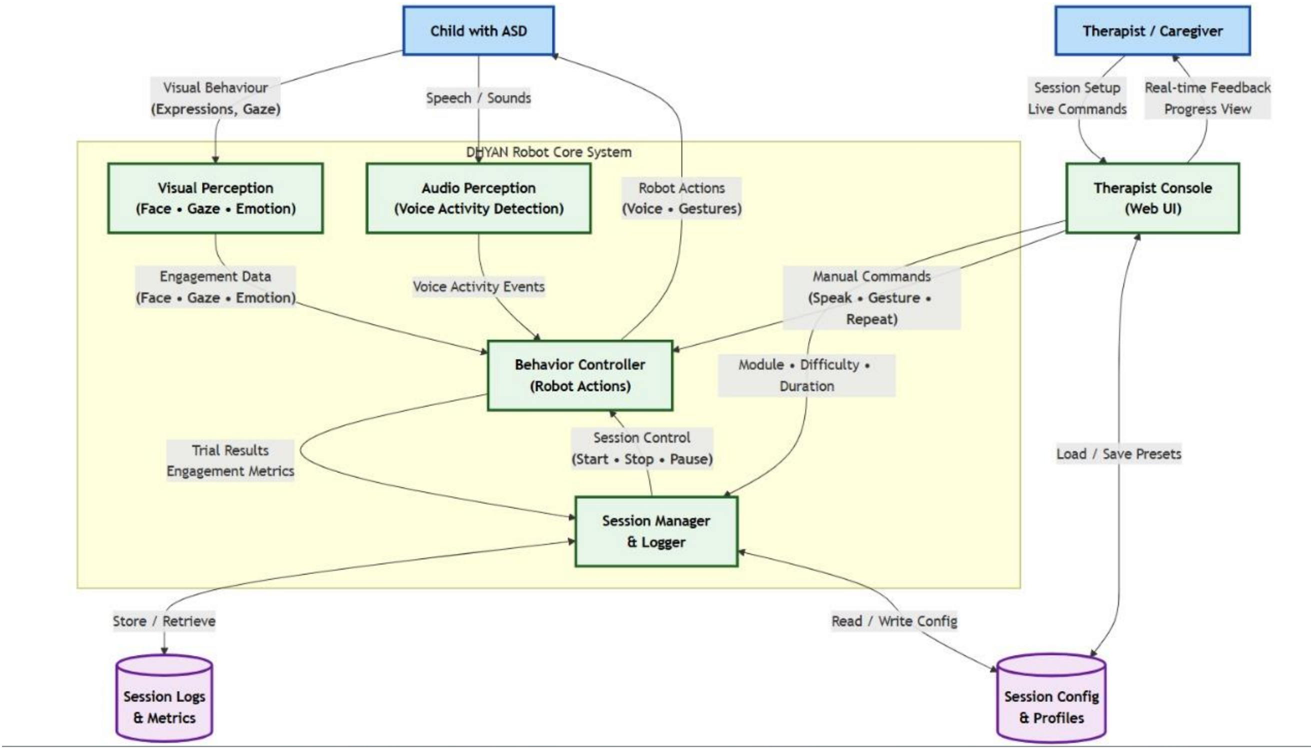
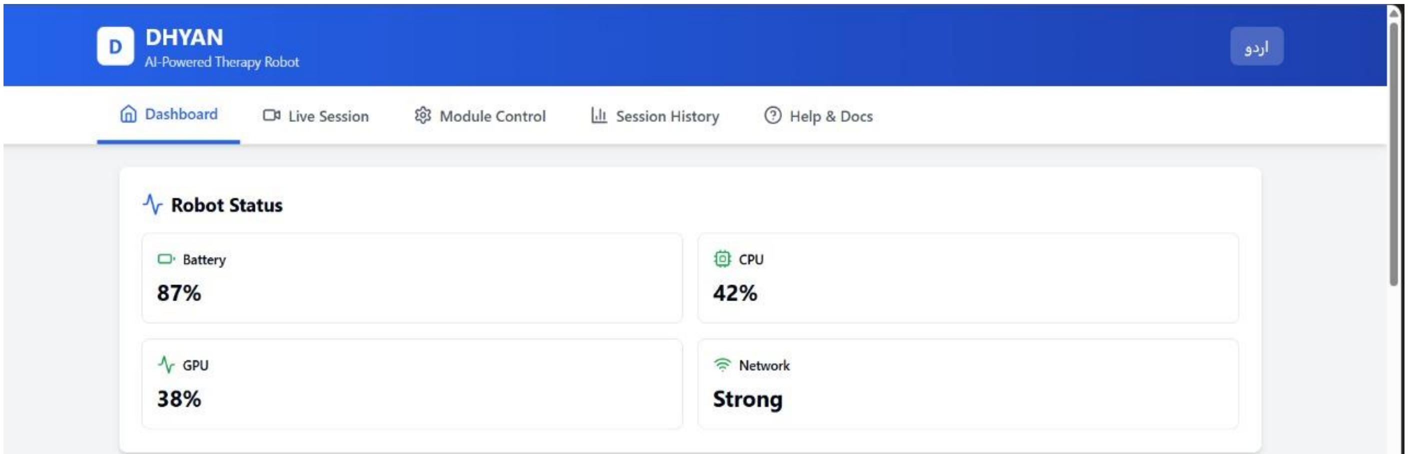
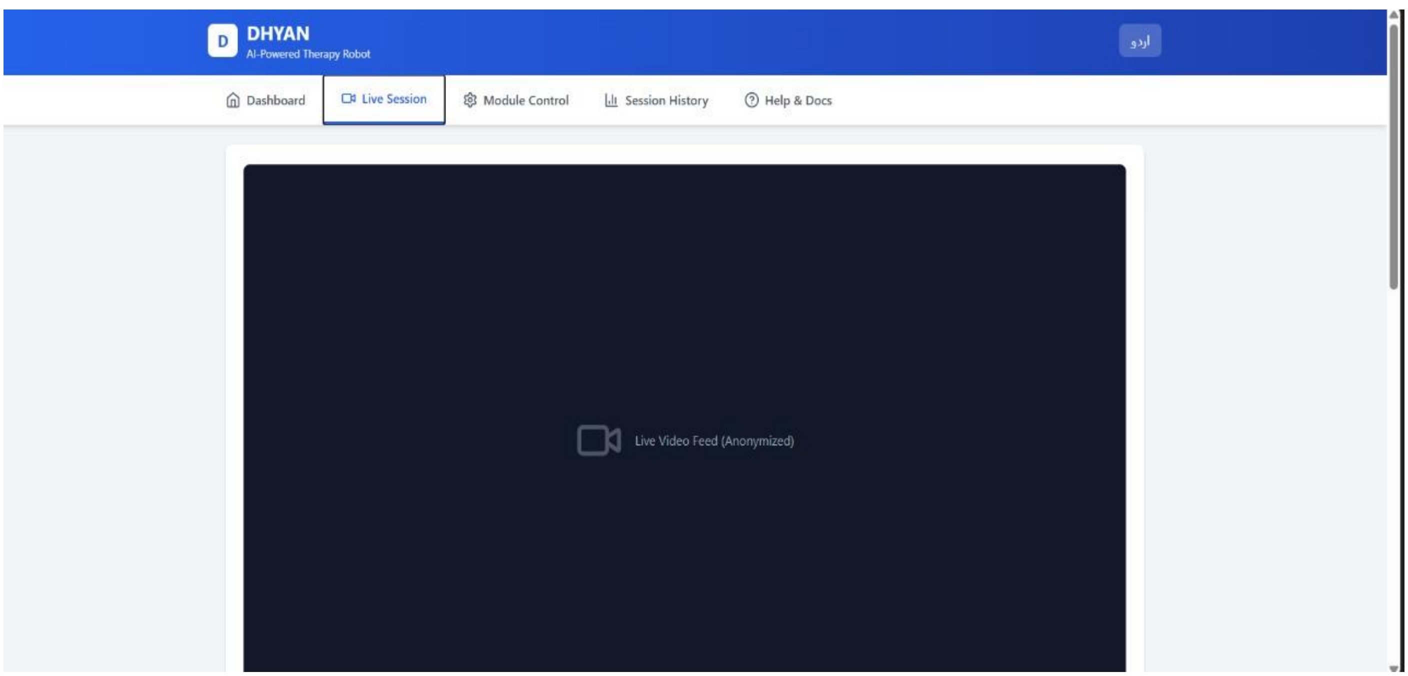


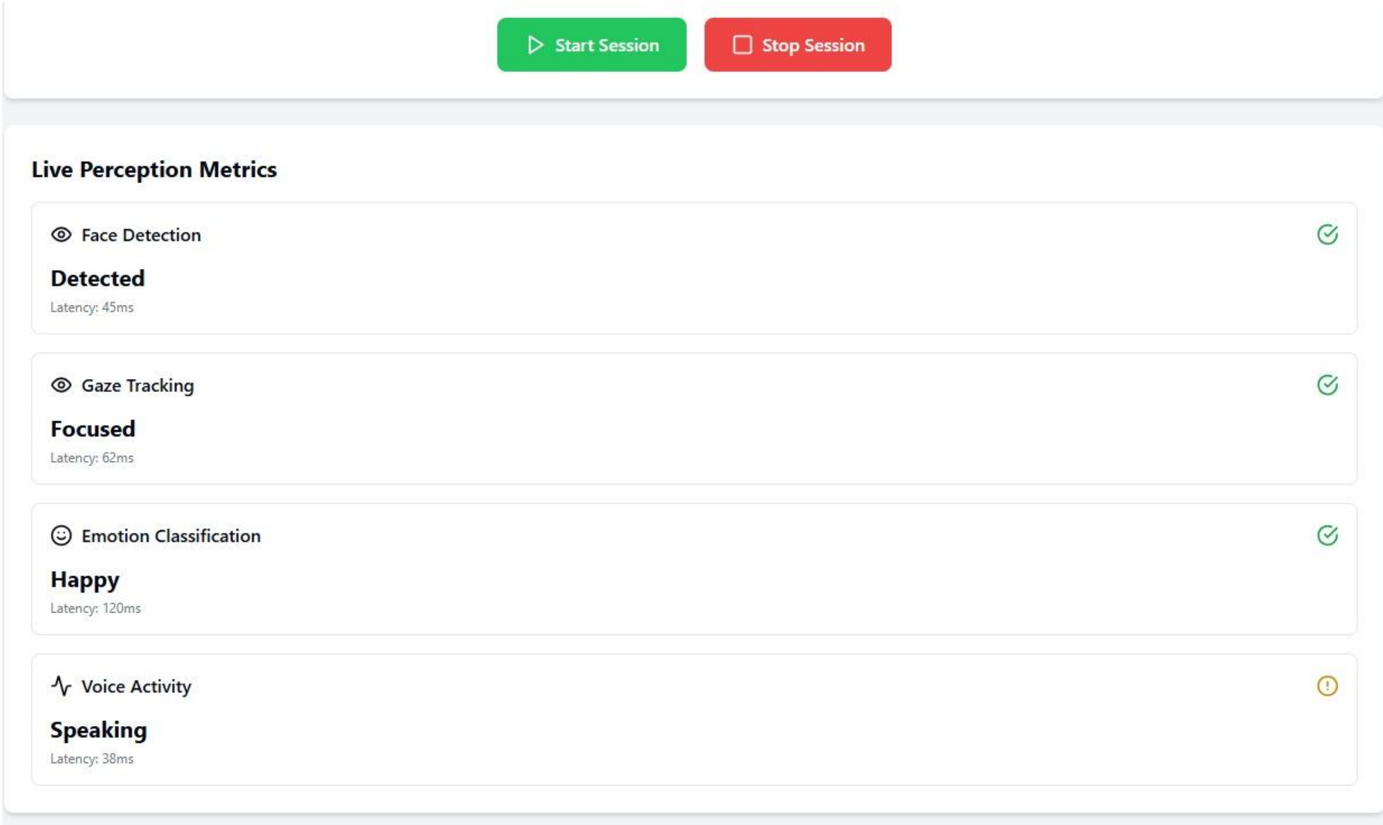
Figure 7: Dataflow Diagram

Figure 8: Entity Relationship Diagram

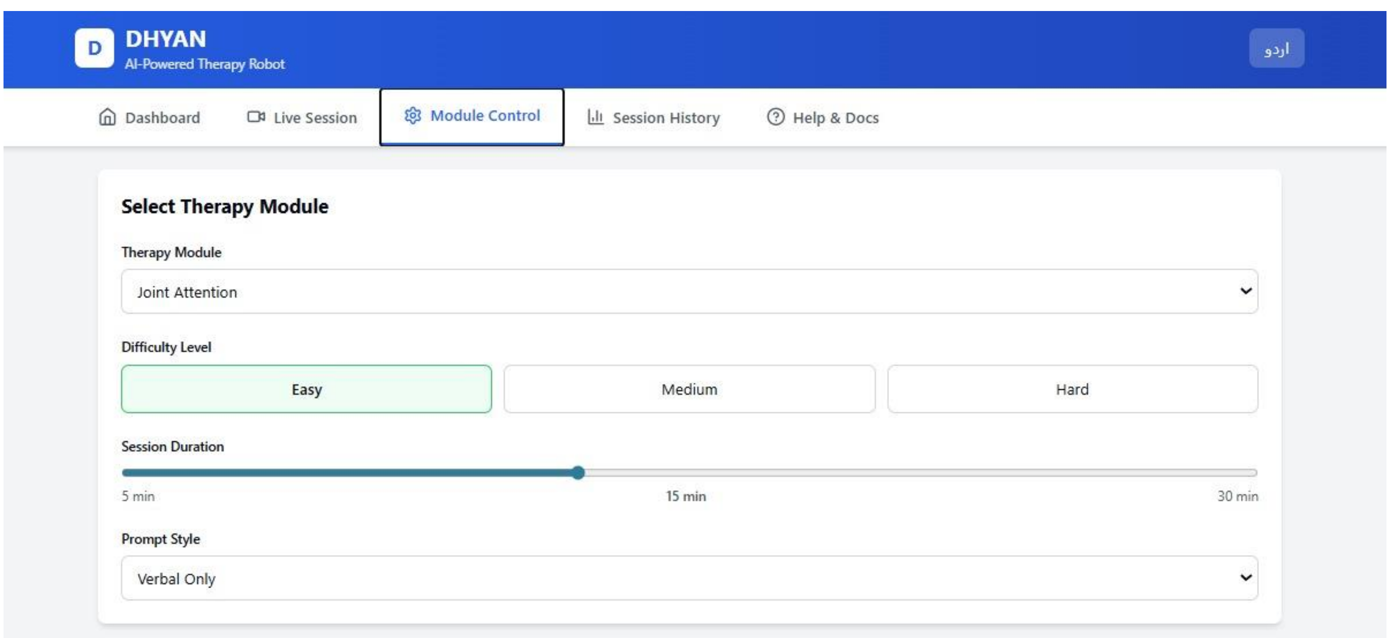
7. UI Design


D
DHYAN
AI-Powered Therapy Robot
اردو
Dashboard
Live Session
Module Control
Session History
Help & Docs
Robot Status
Battery
87%
CPU
42%
GPU
38%
Network
Strong

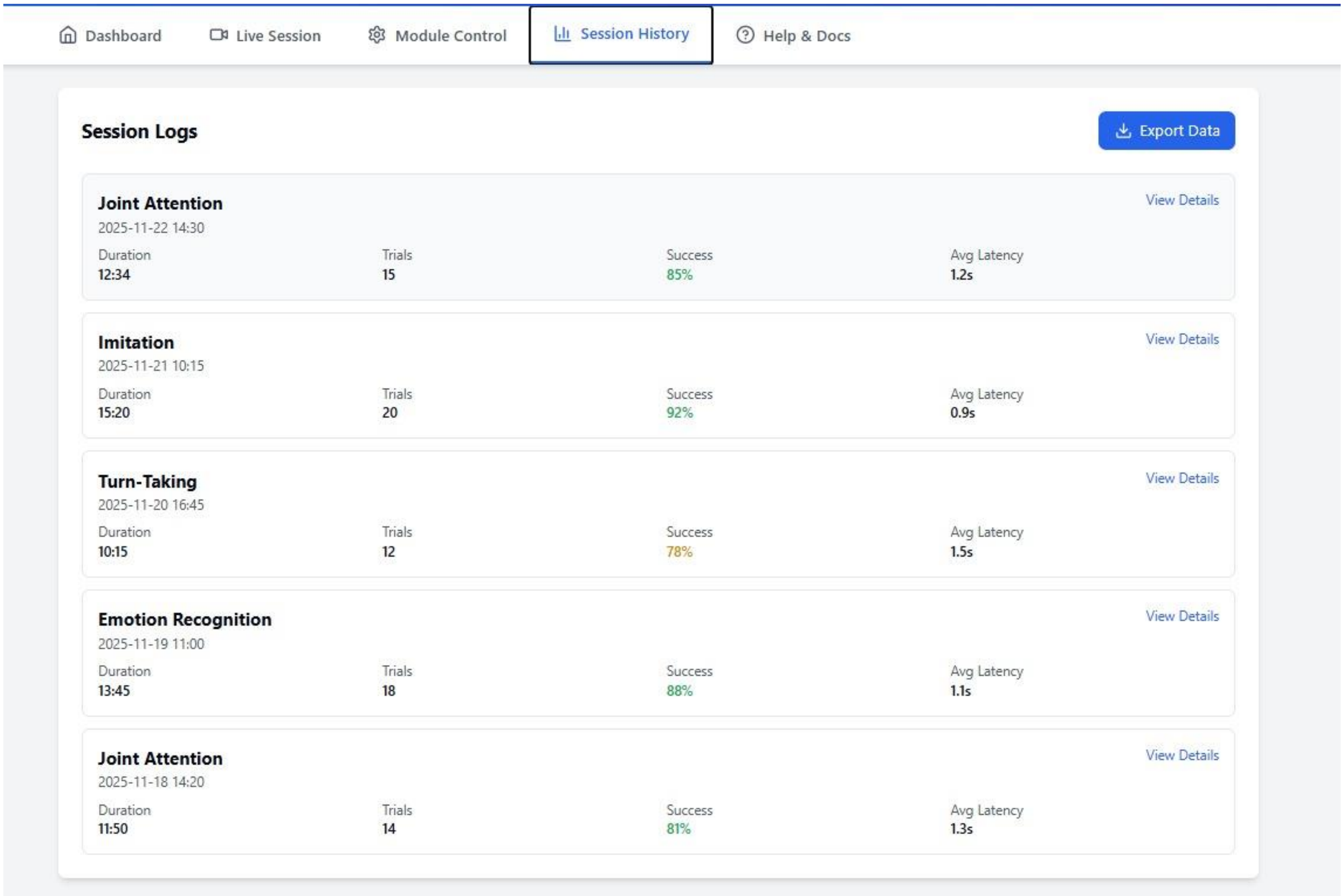

D
DHYAN
AI-Powered Therapy Robot
اردو
Dashboard
Live Session
Module Control
Session History
Help & Docs
Live Video Feed (Anonymized)

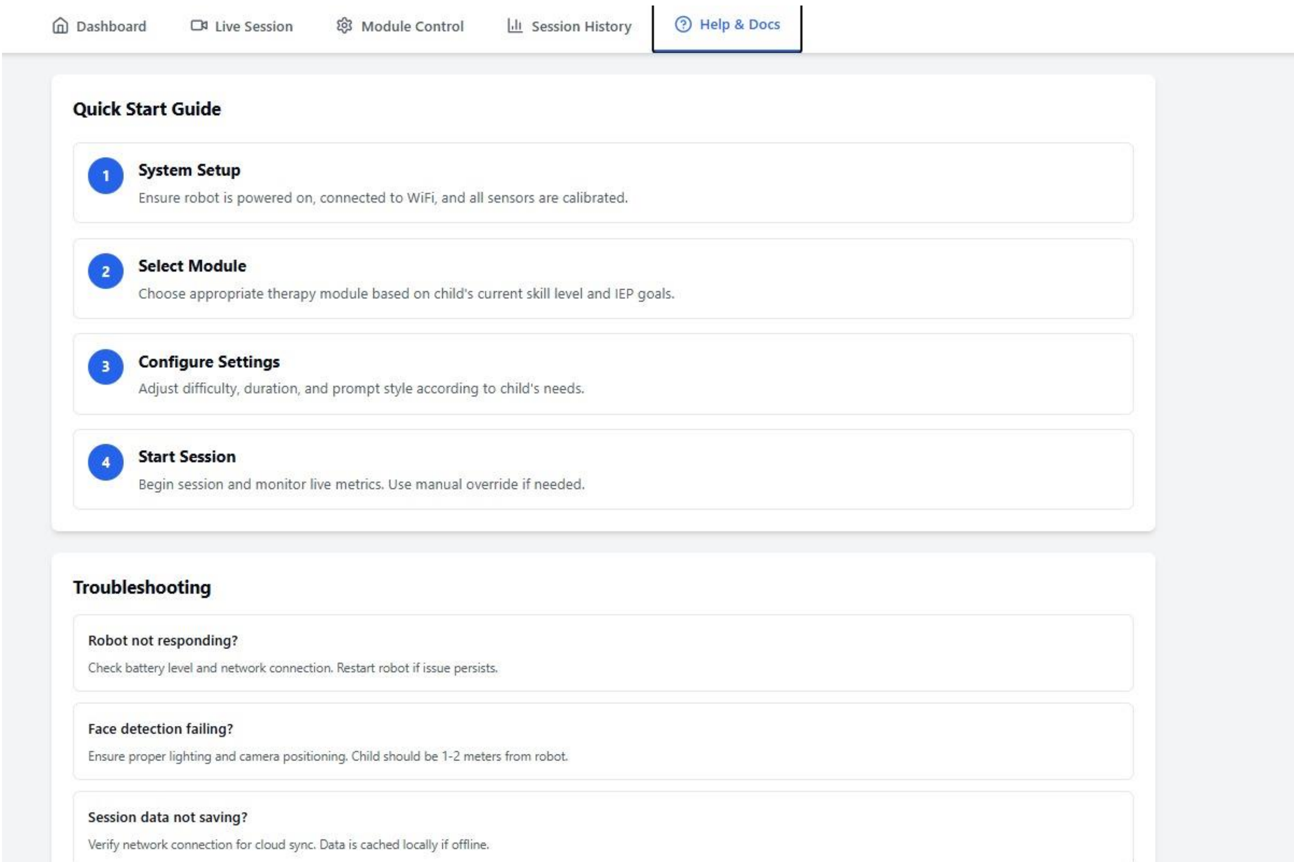


Start Session
Stop Session
Live Perception Metrics
Face Detection
Detected
Latency: 45ms
Gaze Tracking
Focused
Latency: 62ms
Emotion Classification
Happy
Latency: 120ms
Voice Activity
Speaking
Latency: 38ms



D
DHYAN
AI-Powered Therapy Robot
اردو
Dashboard
Live Session
Module Control
Session History
Help & Docs
Select Therapy Module
Therapy Module
Joint Attention
Difficulty Level
Easy
Medium
Hard
Session Duration
5 min
15 min
30 min
Prompt Style
Verbal Only


Dashboard
Live Session
Module Control
Session History
Help & Docs
Session Logs
Export Data
Joint Attention
2025-11-22 14:30
View Details
Duration
12:34
Trials
15
Success
85%
Avg Latency
1.2s
Imitation
2025-11-21 10:15
View Details
Duration
15:20
Trials
20
Success
92%
Avg Latency
0.9s
Turn-Taking
2025-11-20 16:45
View Details
Duration
10:15
Trials
12
Success
78%
Avg Latency
1.5s
Emotion Recognition
2025-11-19 11:00
View Details
Duration
13:45
Trials
18
Success
88%
Avg Latency
1.1s
Joint Attention
2025-11-18 14:20
View Details
Duration
11:50
Trials
14
Success
81%
Avg Latency
1.3s


Dashboard
Live Session
Module Control
Session History
Help & Docs
Quick Start Guide
1
System Setup
Ensure robot is powered on, connected to WiFi, and all sensors are calibrated.
2
Select Module
Choose appropriate therapy module based on child's current skill level and IEP goals.
3
Configure Settings
Adjust difficulty, duration, and prompt style according to child's needs.
4
Start Session
Begin session and monitor live metrics. Use manual override if needed.
Troubleshooting
Robot not responding?
Check battery level and network connection. Restart robot if issue persists.
Face detection failing?
Ensure proper lighting and camera positioning. Child should be 1-2 meters from robot.
Session data not saving?
Verify network connection for cloud sync. Data is cached locally if offline.

Appendix A: Glossary

This glossary defines key technical and domain-specific terms used throughout the DHYAN SRS.

- **ASD (autism spectrum disorder):** A neurodevelopmental condition characterized by challenges in social interaction, communication, and behaviour.
- **SAR (Socially Assistive Robot):** A robot designed to provide assistance through social interaction rather than physical manipulation.
- **ROS (Robot Operating System):** A middleware framework enabling modular communication between software components in robotic systems.
- **Jetson Nano:** An NVIDIA embedded platform for GPU-accelerated deep learning inference.
- **Raspberry Pi:** A compact single-board computer used for behaviour control, sensor communication, and ROS node execution.
- **Face Detection:** Identifying and locating human faces in video input.
- **Gaze Estimation:** Determining where a person is looking using head pose and eye direction cues.
- **VAD (Voice Activity Detection):** Detecting whether a human voice is present in the audio stream.
- **TTS (Text-to-Speech):** Converting textual prompts into audible speech.
- **PWM (Pulse Width Modulation):** A technique used to position servo motors precisely.
- **Therapist Console:** A web-based application for configuring and supervising therapy sessions.
- **Session Log:** A recorded dataset containing timestamps, responses, and performance metrics for each session.

Appendix B: Analysis Models

B.1 System Architecture Diagram Describes the interaction between Raspberry Pi, Jetson Nano, therapist console.
B.2 Activity Diagram Shows the workflow of therapy sessions from initialization to completion.
B.3 State Machine Diagram Describes system states such as Idle, Waiting, Active, Paused, Fault, and Ended.
B.4 Use Case Diagram Represents interactions between therapist across therapy modules.
B.5 Sequence Diagram Illustrates the chronological message flow between robot components and external actors during a therapy task.
B.6 Deployment Diagram Details the hardware/software deployment across Raspberry Pi, Jetson Nano and therapist console.
B.7 Data Flow Diagram (DFD) Shows the flow of data from visual/audio input to behavior controllers, and logging modules.
B.8 Entity-Relationship Diagram (ERD) Models the database structure linking children, therapists, modules and sessions.