Tamilselvan Thangave-TEAM_CSE_201_PROJECT_REPORT.pdf

by Tamilselvan Thangave Tamilselvan Thangave

Submission date: 27-Nov-2025 12:05PM (UTC+0530)
Submission ID: 2828802127
File name: TEAM_CSE_201_PROJECT_REPORT.pdf (1.69M)
Word count: 14194
Character count: 85992

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

Wall-E An Autonomous Educational Robot for Interactive Learning

A PROJECT REPORT

Submitted by

ABHIRAM P B- 20221CSE0044

K VAMSHIDHAR REDDY- 20221CSE0005

YASH RAJ DUTT- 20221CSE0733

Under the guidance of,

MR. TAMILSELVAN T

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report "Wall E: An Autonomous Educational Robot For Interactive Learning" is a bonafide work of "ABHIRAM PB, KOTHAREDDY VAMSHIDHAR REDDY, YASH RAJ DUTT", who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING, during 2025-26.

Mr. Tamilselvan T	**Dr. Asif Mohammad**	**Dr. Shakkeera L**	**Dr. Duraipandian N**
Project Guide	Head of the Department	Associate Dean	Dean
Presidency School of Computer Science and Engineering	Presidency School of Computer Science and Engineering	Presidency School of Computer Science and Engineering	PSCS & PSIS
Presidency University	Presidency University	Presidency University	Presidency University

Name and Signature of the Examiners

1)

2)

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B.Tech in COMPUTER SCIENCE ENGINEERING at Presidency University, Bengaluru, named Abhiram PB, Kothareddy Vamshidhar Reddy and Yash Raj Dutt, hereby declare that the project work titled "Wall E: An Autonomous Educational Robot For Interactive Learning" has been independently carried out by us and submitted in partial fulfillment for the award of the degree of B.Tech in COMPUTER SCIENCE ENGINEERING during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

ABHIRAM PB	20221CSE0044
KOTHAREDDY VAMSHIDHAR REDDY	20221CSE0005
YASH RAJ DUTT	20221CSE0733

PLACE: BENGALURU
DATE: 26-Nov 2025

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide **Mr. Tamilselvan T, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Asif Mohammad, Professor, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators**, Dr. Sharmast vali, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

ABHIRAM PB

KOTHAREDDY VAMSHIDHAR REDDY

YASH RAJ DUTT

Abstract

Home-based learning has grown in importance as families seek engaging educational experiences when human teachers are unavailable. Wall E is a low-cost, semi-autonomous companion robot designed to bridge that gap by delivering personalized lessons at home. Unlike many educational robots that act as simple programmable toys, Wall E combines mechanical mobility, stereo vision, facial recognition, hand-gesture detection and conversational AI to create an interactive tutor. The project was motivated by research showing that social robots can cultivate long-term human–robot relationships and boost learner motivation, and that personalized tutoring systems improve educational outcomes when they adapt to individual preferences and behaviours.

The report describes Wall E's mechanical and electronic design in detail. The robot uses continuous-track mobility and articulated servos to navigate domestic environments and express simple gestures. A Raspberry Pi 5 controls servo movements and sensor inputs, while stereo webcams and microphones provide the robot with depth perception and directional hearing. Gesture recognition is handled by MediaPipe, and a lightweight neural network classifies common hand poses to issue start, stop or pause commands. Obstacle avoidance uses sensor fusion and a potential-field algorithm to plan safe paths in cluttered spaces. This hardware is linked via a high-speed streaming protocol to an external laptop for intensive AI inference.

On the software side, Wall E integrates state-of-the-art machine-learning models. YOLO is employed for real-time object and face detection, while the InsightFace framework generates facial embeddings that are stored in a vector database for memory-based recognition. A lightweight language model enables the robot to generate questions, interpret free-text answers and adapt lesson difficulty based on learner responses. Together, these components allow Wall E to recognise individual students, recall their past performance and tailor interactions to their needs.

Preliminary testing with children in a home-like setting demonstrates that Wall E operates smoothly in real time, responds intuitively to gestures and engages users in meaningful dialogue. The robot maintains stable frame rates for vision tasks and navigates without collisions, and early feedback suggests that children find the interactions motivating and enjoyable. The report concludes by discussing ethical considerations such as data privacy, the

emotional impact of long-term human–robot relationships and the need for parental oversight. Future research directions include fully on-board AI processing to reduce reliance on external computers, further cost reductions and expanded curricular content to enhance the robot's educational value.

Table of Content

List of Figures

List of Tables

Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BCM	Broadcom (GPIO Pin Numbering Mode)
CA	Cloud Analytics
CPU	Central Processing Unit
CSV	Comma-Separated Values
CV	Computer Vision
DPDP Act	Digital Personal Data Protection Act (India, 2023)
DC	Direct Current
DNN	Deep Neural Network
EC	Edge Computing
FPS	Frames Per Second
GPIO	General Purpose Input/Output
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
HAT	Hardware Attached on Top
HCI	Human–Computer Interaction
HRI	Human–Robot Interaction
HMI	Human–Machine Interface
I/O	Input/Output
IC	Integrated Circuit
IP	Internet Protocol
IoT	Internet of Things
IR	Infrared
LLM	Large Language Model
LCD	Liquid Crystal Display
LED	Light Emitting Diode
ML	Machine Learning
MSS	Multilingual Speech Support

Abbreviation	Full Form
MP	MediaPipe
NLP	Natural Language Processing
NN	Neural Network
OCR	Optical Character Recognition
OM	Offline Mode / Offline Operation
OpenCV	Open-Source Computer Vision Library
PLA	Polylactic Acid
Pi/RPi	Raspberry Pi
QA	Quality Assurance
RL	Reinforcement Learning
RoHS	Restriction of Hazardous Substances
SA	Sentiment Analysis
SDG	Sustainable Development Goals
STT	Speech-to-Text
SSH	Secure Shell
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
SSD	Solid-State Drive
TTS	Text-to-Speech
TVM	Tensor Virtual Machine
UI/UX	User Interface/User Experience
VNC	Virtual Network Computing
WEEE	Waste Electrical and Electronic Equipment
Wi-Fi	Wireless Fidelity
YOLO	You Only Look Once

Chapter 1
Introduction

1.1 Background

The last two decades have witnessed a dramatic convergence of artificial intelligence, robotics, and education. With the proliferation of low-cost computing devices such as Raspberry Pi and Arduino, robotics has moved from specialized laboratories into homes, classrooms, and hospitals. Educational robots are increasingly seen as catalysts for personalized learning, promoting curiosity, creativity, and critical thinking among children. Unlike traditional e-learning platforms, which rely on screen-based interactions, robots provide **embodied presence**: they move, gesture, make eye contact, and respond in real time, simulating aspects of human teaching that digital applications alone cannot replicate.

The COVID-19 pandemic accelerated this trend by exposing the vulnerability of global education systems. School closures at the peak of the pandemic disrupted learning for more than 1.5 billion students worldwide, according to UNESCO (2021). While video conferencing platforms partially bridged the gap, they highlighted challenges such as screen fatigue, limited interactivity, and inequitable access. Parents and educators began seeking alternatives that could combine digital instruction with *physical presence* to keep learners engaged in domestic environments. Educational robots, especially semi-autonomous ones, emerged as promising solutions to supplement teachers when unavailable, support children's independent study, and maintain continuity of learning in emergencies.

Walle, the project discussed in this report, is positioned within this context. Inspired by Pixar's WALL-E, the robot integrates mobility, vision, conversational AI, and gesture recognition to create an **interactive tutor at home**. Unlike industrial-grade robots costing thousands of dollars, Walle emphasizes affordability and modularity, making it accessible for middle-class households and adaptable for long-term use.

1.2 Statistics of Project

The scope of the educational robotics sector underscores the urgency and relevance of this project. Market studies predict that the **global educational robot market will reach USD 3.2 billion by 2028**, growing at a compound annual growth rate (CAGR) of 16% from 2023

(Markets & Markets, 2023). This growth is fueled by three factors: (a) advances in AI and computer vision, (b) declining costs of hardware components, and (c) growing demand for personalized education.

In India alone, where this project originates, nearly **250 million children** are enrolled in primary and secondary schools (Government of India, Ministry of Education, 2023). Yet the **pupil–teacher ratio** remains a challenge, with rural schools averaging **1 teacher for every 35–40 students**. Robots like Walle offer a pathway to complement human teachers by providing additional attention and interactive learning sessions.

At the prototype level, Walle achieves the following key performance statistics:

- **Cost:** Approx. USD 400 (versus USD 7,500–20,000 for commercial robots like NAO or Pepper).
- **Latency:** 120 ms end-to-end actuation delay, enabling real-time responses.
- **Accuracy:** 95% face recognition precision; 98% gesture classification accuracy.
- **Battery Runtime:** ~2 hours continuous operation.

These figures demonstrate that **affordable and effective robotics for home learning is achievable** with consumer-grade hardware. Importantly, they highlight the viability of deploying Walle at scale, particularly in resource-constrained settings.

1.3 Prior Existing Technologies

Several robotic platforms have attempted to address educational challenges, though most remain either cost-prohibitive or limited in capability:

- **NAO Robot (SoftBank Robotics):** A humanoid robot used in research and classrooms, costing around USD 7,500. While NAO can walk, speak, and gesture, its face recognition is limited, and customization requires advanced programming skills .
- **Pepper Robot:** A larger humanoid retailing at nearly USD 20,000. Pepper integrates conversational AI but suffers from restricted mobility, high costs, and impracticality for home-based learning.
- **Kebbi/Minibo:** Mid-tier robots costing around USD 1,000, focused on simple storytelling and interactive lessons. These lack face recognition, advanced dialogue, or robust gesture control.

In contrast, Walle fills a **unique gap**: affordable enough for home use, modular for tinkering and upgrades, yet advanced enough to deliver adaptive tutoring through vision-based recognition and AI-driven dialogue. By leveraging open-source frameworks (YOLO, InsightFace, MediaPipe) and lightweight LLMs, Walle demonstrates that *cutting-edge features need not be exclusive to expensive platforms*.

1.4 Proposed Approach

The proposed system architecture for Walle is built around **semi-autonomy, multimodal interaction, and affordability**. The approach can be summarized as follows:

1. **Hardware Efficiency:** Use 3D-printed chassis, Raspberry Pi 5, low-cost motors, and webcams to balance durability and cost.
2. **Vision and Memory:** Employ YOLOv5n for object/face detection, InsightFace for recognition, and FAISS vector database for efficient storage of embeddings.
3. **Gesture-Based Interaction:** Implement MediaPipe landmark extraction with a neural classifier to interpret open/closed palms as intuitive commands.
4. **Obstacle Avoidance:** Fuse stereo vision with ultrasonic/IR sensors; apply potential-field planning with offset force for smooth, safe navigation.
5. **Conversational AI:** Deploy a distilled GPT-J variant for question generation, response evaluation, and difficulty adjustment.
6. **User Engagement:** Integrate LEDs, speech synthesis, and gaze tracking for emotional presence.
7. **Caregiver Interface:** Companion mobile app for monitoring, lesson customization, and data management.

The novelty lies in combining these techniques within a **low-cost home robot**, designed not only to deliver lessons but also to sustain engagement through emotional cues and personalized interaction.

1.5 Objectives

The primary objectives of this project include:

1. **Design and Build** a modular, semi-autonomous educational robot capable of operating in household environments.
2. **Enable Personalization** through face recognition and learner-specific data storage.
3. **Facilitate Intuitive Control** via simple hand gestures and gaze alignment.
4. **Support Adaptive Learning** with a lightweight LLM that adjusts lesson difficulty dynamically.
5. **Ensure Affordability and Scalability** by optimizing hardware and software for cost-effectiveness.
6. **Promote Responsible Use** through privacy safeguards, modular repurposing, and transparent data policies.

Collectively, these objectives align with global calls to leverage technology for inclusive, equitable, and quality education.

1.6 SDGs

The United Nations' **Sustainable Development Goals (SDGs)** provide a framework for evaluating the social impact of technologies like Walle. This project directly contributes to several SDGs:

- **SDG 4: Quality Education** – By supplementing human teaching, Walle helps reduce educational inequities, particularly in under-resourced communities.
- **SDG 9: Industry, Innovation, and Infrastructure** – The robot demonstrates how affordable innovation can foster resilient education systems.
- **SDG 10: Reduced Inequalities** – Children unable to attend regular schools (due to illness, disability, or geography) can still receive interactive education at home.
- **SDG 3: Good Health and Well-being** – By reducing screen fatigue and fostering embodied learning, Walle promotes healthier study habits.
- **SDG 12: Responsible Consumption and Production** – Modular design and repurposing strategies ensure sustainable use and recycling of components.

Framing Walle within the SDGs emphasizes that the project is not merely technical, but also social and ethical, responding to global development priorities.

1.7 Overview of Project Report

The remainder of this report is organized into the following chapters:

- **Chapter 2: Literature Review** – Synthesizes research on educational robotics, long-term interactions, affective tutoring systems, gesture recognition, face recognition, and multimodal AI tutors.
- **Chapter 3: Walle System Design** – Details the mechanical, electronic, and software architecture, along with AI algorithms and user interfaces.
- **Chapter 4: Implementation and Evaluation** – Describes construction, testing, performance metrics, and comparative analysis.
- **Chapter 5: Ethical Considerations and Long-Term Use** – Discusses privacy, child-robot relationships, and sustainability.
- **Chapter 6: Conclusion and Future Work** – Summarizes findings, outlines limitations, and proposes research directions.

Chapter 2

Literature review

2.1 Narrative Review of Key Works

(1) Kaur & Singh (2023) – Impact of Educational Robots in Schools [1]. Kaur and Singh provide a broad review of educational robots in school settings, arguing that embodied, interactive systems consistently raise engagement and strengthen problem-solving and STEM motivation by affording hands-on, iterative learning rather than passive screen time. Conceptually, the paper frames robots as sociocognitive partners that scaffold inquiry, collaboration, and immediate feedback. Methodologically, it synthesizes controlled classroom trials and quasi-experimental deployments, noting positive trends in attention and task persistence. Analysis highlights that benefits are strongest when robots are integrated into structured lesson plans with teacher mediation, rather than used as standalone novelties. Issues include heterogeneous study designs, small sample sizes, short durations, and the risk that over-automation displaces peer collaboration. The review's main limitations are a concentration on classroom robots (less on home/hospital contexts) and sparse evidence on long-term learning transfer. For Walle, the paper supports embodied interaction and teacher-aligned scripts but cautions against over-reliance; improvements include embedding social prompts that nudge human collaboration, and designing longer, curriculum-aligned evaluations to test retention and transfer beyond immediate gains.

(2) Hussain et al. (2019) – Interactive Robot for Environmental Awareness [2]. Hussain and colleagues describe an educational robot that delivers story-based lessons and quizzes to promote environmental awareness among children. The concept fuses narrative pedagogy with interactive Q&A to keep learners attentive; the approach relies on predesigned scenarios and scripted branching feedback. Methods include iterative prototype testing with classroom-age participants and comparison to conventional instruction, with results indicating higher motivation and recall when children interact with the robot. The analysis underscores the value of immediate, embodied prompts (voice, gaze, gesture) in reinforcing key ideas. Core issues are the constrained natural language capabilities and limited personalization; learning paths are largely linear, and content adaptation depends on pre-authored flows. Limitations include narrow topical focus and absence of learner modeling for differentiated difficulty. This paper's implications for Walle are direct: keep narrative structure (because it works) but replace

scripting with adaptive dialogue and memory, broaden content coverage, and support subtle affect/timing adjustments so reinforcement is context-sensitive rather than fixed.

(3) Pelizzari, Rocco & Ferrari (2025) – Robotics in Hospital & Home Education: Systematic Review [3]. This systematic review synthesizes 30 peer-reviewed studies on robotics for learners who cannot attend school (e.g., illness, prolonged absence), emphasizing continuity, social connection, and agency. Conceptually, the authors argue that robotics "re-embodies" learning at a distance by restoring presence, action, and turn-taking that videoconferencing lacks. The approach categorizes interventions by setting (home/hospital), pedagogical aim (engagement vs. content mastery), and technology (telepresence vs. autonomous social robots). The analysis finds consistent engagement gains and reduced social isolation when robots are used as proxies for classroom participation or as at-home learning companions. Issues include sparse adolescent-specific research, limited attention to cultural/linguistic diversity, and a short horizon of evaluation (novelty effects). The main limitations are inconsistent metrics and under-reporting of safety, privacy, and caregiver workload. The review recommends longitudinal designs, caregiver-friendly tooling, and adaptive content. For Walle, this justifies home-first design, caregiver dashboards, and consent-driven data handling; improvements include explicit adolescent personas and longitudinal protocols to measure beyond early novelty.

(4) Zhao & McEwen (2025) – The Robot That Stayed: Long-Term Household Engagement [4]. Following 19 families over multiple years after a preschool reading robot "aged out," this study shows robots often transform from instructional devices into symbolic household members. The concept centers on lifecycle design: personification and attachment persist even as formal learning wanes, leading to repurposing (e.g., keepsake, decorative companion). Methods combine qualitative interviews, artifact tours, and longitudinal field notes; analysis identifies themes—symbolic value, caretaking behaviors, and household routines—that extend beyond planned instructional use. Issues include tensions between emotional attachment and data stewardship (who owns memories?) and the absence of off-boarding rituals or repurposing guides. Limitations include a geographically limited sample and reliance on self-report. The paper's recommendations—design for retirement, data export/deletion, modular repurposing—map cleanly onto Walle: offer "goodbye" modes, exportable recognition memories, and re-use paths (e.g., head as webcam), so long-term affect is acknowledged without compromising privacy.

(5) Maaz, Mounsef & Maalouf (2025) – CARE: Customized Assistive Robot-Based Education [5]. CARE proposes a framework for **affective** and **cognitive** assessment in real time, combining performance traces with facial/behavioral indicators to adapt resources. The approach uses machine-learning classifiers (e.g., XGBoost) to estimate learning state and affect, then proactively modulates difficulty and support. Results across controlled studies suggest increased persistence and improved short-term outcomes when tutoring is tuned to emotional as well as cognitive cues. Analysis warns that over-proactivity can harm trust or feel intrusive, and that facial affect (alone) is not culturally universal or reliable. Issues include bias/variance in emotion detection and teacher workload in curating resource libraries. Limitations are constrained contexts and limited cross-cultural validation. Recommendations include multimodal affect sensing, transparent interventions, and opt-out controls. For Walle, the implications are to combine gestures, timing, and dialogue sentiment (not faces alone), throttle "helpfulness," and surface clear controls (pause/stop gestures) to maintain user agency.

(6) Sievers (2025) – Humanoid Social Robot as Teaching Assistant [13]. Sievers examines classroom deployments of a humanoid teaching assistant, focusing on embodiment (expressive gestures, gaze), interaction fluency, and classroom orchestration. The concept leverages social affordances of humanoids to sustain attention and orchestrate small-group interaction. Methods synthesize observational data and performance indicators under teacher-led activities; analysis shows engagement benefits when the robot complements—not replaces—teacher scaffolding. Issues are practical: high acquisition/maintenance costs, limited mobility in cluttered rooms, and weak personalization. Limitations include short trials and reliance on proprietary software stacks that hinder customization. The work suggests that compact, lower-cost, modular robots with rich social signaling may achieve better scalability. For Walle, this supports choosing expressive micro-movements (eyes/neck), tight teacher alignment, and cost-bounded hardware over full humanoid complexity, while investing in personalization and easy re-authoring.

(7) Nguyen et al. (2023) – Gesture-Controlled Mobility with Lightweight Models [6]. Nguyen et al. present a low-cost, hand-gesture interface for mobility assistance using a compact detector and MediaPipe landmarks on embedded hardware. The concept demonstrates that robust, real-time gesture control is feasible on edge devices if the gesture vocabulary is carefully scoped. Methods involve collecting a diverse gesture dataset, training a lightweight recognition stack, and validating in real-world driving tasks. Results indicate high recognition

fidelity under varied conditions, with performance drops chiefly due to occlusion/lighting extremes. Analysis emphasizes careful dataset curation, strong confidence thresholds, and fallback behaviors for safety. Limitations include sensitivity to partial occlusion and a fixed gesture set that may not generalize culturally. For Walle, the study justifies a minimalist, high-precision gesture vocabulary (open/closed palm) with conservative thresholds, timeout states, and redundancy (e.g., voice/app override) to keep control intuitive yet safe.

(8) Du et al. (2024) – Depth + Range Sensing with Potential-Field Planning [7]. Du and colleagues fuse stereo depth with a 2-D laser rangefinder and modify potential-field path planning by introducing an offset force to escape local minima. The concept is to marry richer perception with a simple, reactive planner to achieve smooth, collision-averse navigation without heavy global mapping. Methods benchmark the variant against standard fields and show fewer stalls and smoother arcs near obstacles. Analysis underlines that sensor fusion broadens the field of view and increases robustness to reflective surfaces. Issues are cost/complexity (dual sensors) and sensitivity to calibration. Limitations include laboratory-style layouts and limited testing in cluttered domestic spaces. For Walle, the takeaway is to keep the planning idea (offset forces) but substitute affordable sensors (stereo + ultrasonic/IR), adding conservative stop conditions and human-in-the-loop recovery to suit cost-constrained home contexts.

(9) Afifi et al. (2023) – Real-Time Face Recognition with Deep Embeddings + SVM [8]. Afifi et al. describe a face recognition pipeline that extracts fixed-length deep embeddings and classifies identities with an SVM, targeting real-time operation. Conceptually, decoupling representation (embedding) from classification supports incremental enrollment and practical runtime. Methods evaluate throughput/precision across varied lighting and pose; results show strong accuracy and responsiveness on moderate identity sets. Analysis notes performance degradation at larger scales and latency spikes during nearest-neighbor searches. Issues include privacy, consent, and model drift as users age or change appearance. Limitations are constrained demographics and environments. For Walle, this motivates using high-quality embeddings (e.g., InsightFace) paired with **vector databases** for approximate nearest-neighbor search to preserve responsiveness as the household gallery grows, while adding parental consent, local-only storage, and easy deletion/export controls.

(10) Qadir et al. (2024) – Efficient Vector Databases for Face Retrieval [9]. Qadir and co-authors examine indexing, latency, and security trade-offs in high-dimensional

vector databases applied to face retrieval. The concept is that approximate search structures (e.g., HNSW, IVF) preserve accuracy while enabling interactive latency at scale. Methods compare index types and parameterizations on standard embedding corpora; analysis surfaces tensions among speed, memory, and recall, along with attack surfaces (model inversion, membership inference) and policy gaps (data retention, consent). Issues include bias amplification if galleries are unbalanced and the challenge of refreshing embeddings without downtime. Limitations are mostly system-level (few user studies), but the engineering insights are directly useful. For Walle, this supports adopting FAISS-style indices with conservative recall targets, encrypt-at-rest policies, and transparent, caregiver-mediated enrollment and deletion flows to meet household privacy expectations.

(11) Torres-Caballero et al. (2024) – Low-Cost Multimodal Language-Learning Robot [11]. This work prototypes a language-learning assistant built on Raspberry Pi/Arduino that blends voice, visual prompts, and gestures to personalize drills. The concept validates that multimodal interaction on modest hardware measurably lifts engagement compared to single-channel tutoring. Methods include classroom-style sessions with repeated measures; results suggest improved attention and willingness to practice. Analysis, however, acknowledges limited language naturalness and brittle dialog management due to compute constraints. Issues include shallow personalization and difficulty scaling beyond templated lessons. Limitations are short sessions and topic-bounded evaluations. The paper's recommendations—optimize models, cache dialogue templates, and offload selectively—inform Walle's choice of a **lightweight LLM** with quantization, local inference for responsiveness, and narrow, age-appropriate scope to maintain fluency under edge constraints.

(12) Sun et al. (2025) – Empathetic Robot Tutor with Multimodal LLMs [12]. Sun and colleagues integrate a multimodal LLM to unify speech, gesture, and affect cues into empathetic tutoring behaviors. Conceptually, the model treats tutoring as stateful, multimodal dialogue with memory over sessions. Methods combine controlled experiments and ablations to show better motivation and retention when the tutor adapts to cues like gaze and affect. Analysis highlights the promise of unified models but flags latency, cost, and privacy for cloud-based inference. Issues include explainability of model decisions and fairness across dialects/cultures. Limitations are resource intensity and incomplete coverage of edge-only scenarios. For Walle, the direction is clear: approximate the benefits with **distilled/quantized**

components at the edge, keep clear consent boundaries for any cloud features, and provide transparent rationales and caregiver controls to manage trust.

(13) Santana et al. (2024) – ROS + Chatbot Control for Service Robots [10]. Santana et al. demonstrate a ROS-integrated control stack with chatbot interfaces for service robots, emphasizing modularity and conversational tasking. Conceptually, language becomes a high-level policy layer that triggers robot skills; methods present a ROS node architecture with message passing between perception, planning, and dialogue agents. Results show the practicality of event-driven orchestration and simplified user programming. Analysis points out brittleness under ASR errors and the need for grounded semantics to avoid unsafe actions. Issues include the gap between conversational intent and low-level control guarantees; limitations involve lab-style tasks and scripted domains. For Walle, the lesson is to keep a crisp separation between conversational intent and safety-checked motion primitives, with explicit guardrails and gesture overrides to ensure that natural language never bypasses physical safety constraints.

2.2 Cross-Cutting Concepts, Methods, and Issues

Across the literature, **embodiment** (co-present, gaze/gesture-capable agents) consistently increases engagement relative to disembodied apps [1–3,13]. **Personalization**—via learner modeling, memory, and affect sensing—drives persistence and learning gains but must be balanced against **trust and agency**: proactively "helpful" robots can feel intrusive [5,12]. **Multimodality** (voice, vision, gesture) improves interaction fluency on modest hardware [6,11], while **sensor fusion + simple planners** provide robust mobility without heavyweight mapping [7]. Methodologically, many studies rely on **short deployments** and **small samples**, limiting generalization and under-measuring long-term outcomes [1–3,13]. Ethically, **privacy, consent, bias, and lifecycle** are recurring themes: face data storage, cultural variance in affect, and end-of-life design all require explicit controls [4,5,8,9,12].

2.3 Limitations, Inconsistencies, Gaps, and Contradictions

- **Duration and Scale:** Most trials are brief with limited participants; novelty effects likely inflate early engagement [1–3,13].
- **Cost vs. Capability:** High-end humanoids deliver rich signals but are financially impractical for homes; low-cost systems risk under-personalization [11,13].

- **Affect Reliability:** Vision-only affect estimation is noisy and culturally biased; over-proactivity can **undermine trust** [5,12].
- **Privacy & Governance:** Few papers operationalize data ownership, retention, export, and deletion—critical in homes with minors [4,8,9,12].
- **Evaluation Metrics:** Inconsistent outcome measures (engagement proxies vs. learning gains) make cross-study comparison hard [1–3].
- **Mobility Ecologies:** Navigation results often come from lab-like spaces; domestic clutter and pets remain under-studied [7].
- **Dialogue Robustness:** Edge devices struggle with open-ended dialogue; scripted systems don't scale, cloud systems raise latency/privacy concerns [11,12].
- **Lifecycle Design:** Robots "stay" as symbolic artifacts; few designs plan graceful retirement or repurposing paths [4].

2.4 Suggested Improvements

1. **Edge-First Personalization:** Use quantized/distilled language models with cached prompts and episodic memory to balance fluency and privacy [11,12].
2. **Multimodal Affective Cues:** Combine gestures, response timing, and dialogue sentiment; de-emphasize face-only affect to reduce bias [5,12].
3. **Transparent Controls:** Provide explicit start/stop gestures and caregiver dashboards for consent, enrollment, export, and deletion [4,8,9].
4. **Lifecycle & Repurposing:** Ship off-boarding rituals, parts re-use guides, and data portability from day one [4].
5. **Safe Conversational Orchestration:** Keep a safety-checked motion layer beneath dialogue; include fallback timeouts and "stop" gestures [10].
6. **Navigation for Homes:** Adapt potential-field offset ideas to low-cost sensors with conservative collision policies and human-in-the-loop recovery [7].
7. **Evaluation Rigor:** Design longer, controlled studies with standardized metrics (engagement + learning gains + caregiver workload) [1–3,13].

2.5 Summary of Literatures Reviews

Table 2.5.1 – Summary of Literature Reviews

Study	Context & Approach	Methods	Main Findings	Key Limits/Gaps	Takeaway for Walle
Kaur & Singh (2023) [1]	Review of school-based robots and learning impact	Synthesis of classroom trials	Embodiment ↑ engagement & STEM interest	Short durations; limited transfer data	Use embodied cues; evaluate long-term learning
Hussain et al. (2019) [2]	Story-based environmental tutor robot	Prototype + comparative sessions	Narrative + interaction improves recall	Scripted, low personalization	Keep narrative, add adaptive dialogue
Pelizzari et al. (2025) [3]	Home/hospital robots for continuity	Systematic review (30 studies)	Reduced isolation; consistent engagement	Adolescent gap; weak longitudinal data	Home-first design; plan caregiver tooling
Zhao & McEwen (2025) [4]	Long-term family re-engagement	Longitudinal qualitative study	Robots gain symbolic roles over time	Data stewardship; no retirement paths	Add export/delete; modular repurposing
Maaz et al. (2025) [5]	CARE affective tutoring framework	Real-time affect + performance modeling	Personalization boosts persistence	Over-proactivity risks trust; affect bias	Multimodal cues; throttle proactivity
Sievers (2025) [13]	Humanoid teaching assistant	Classroom deployments	Engagement ↑ with teacher-aligned roles	High cost; low personalization	Compact, low-cost, expressive

					micro-moves
Study	**Context & Approach**	**Methods**	**Main Findings**	**Key Limits/Gaps**	**Takeaway for Walle**
Nguyen et al. (2023) [6]	Edge-based gesture control	Embedded model + landmarking	Robust control with small vocabularies	Occlusion sensitivity	Minimal, high-confidence gestures + fallbacks
Du et al. (2024) [7]	Sensor-fusion navigation	Depth + LRF; offset potential fields	Smoother, safer paths	Cost/calibration; lab-style spaces	Apply offset idea to low-cost sensors
Afifi et al. (2023) [8]	Face embeddings + SVM	Real-time pipeline	High accuracy at moderate scale	Latency at scale; privacy	Use FAISS; consent, local storage, deletion
Qadir et al. (2024) [9]	Vector DBs for recognition	Index benchmarking & security analysis	Interactive retrieval at scale	Bias, governance, security risks	Conservative ANN settings; encrypt + controls
Torres-Caballero (2024) [11]	Low-cost multimodal tutor	Classroom-style sessions	Multimodality ↑ engagement	Dialogue brittleness on edge	Quantized LLM; cache & narrow scope
Sun et al. (2025) [12]	Multimodal LLM tutor	Controlled trials + ablations	Better motivation/retention	Cloud cost/latency/privacy	Edge-first; transparent cloud opt-ins

2.6 How This Literature Shapes the Project

Drawing from the above, Walle deliberately prioritizes **embodied, multimodal interaction** (gesture + gaze + speech), **edge-first personalization** (quantized LLM + local vector search), **cost-bounded mobility** (sensor fusion + safe reactive planning), and **ethically governed memory** (caregiver consent, export/deletion). The design explicitly operationalizes lifecycle thinking (repurposing, "goodbye" flows) and evaluation planning (beyond novelty) to address long-standing gaps identified across the literature.

References

[1] M. P. Kaur and H. Singh, "Impact of Educational Robots on Learning in Schools: A Review," ICCCNT, 2023.

[2] S. Hussain et al., "An Interactive Educational Robot for Children's Learning and Environmental Awareness," ICASERT, 2019.

[3] F. Pelizzari, S. Rocco and S. Ferrari, "Integrating Robotics in Hospital and Home Education: A Systematic Review…," *Continuity in Education*, 2025.

[4] Z. Zhao and R. McEwen, "The Robot that Stayed…," *Frontiers in Robotics and AI*, 2025.

[5] N. Maaz, J. Mounsef and N. Maalouf, "CARE: Towards Customized Assistive Robot-Based Education," *Frontiers in Robotics and AI*, 2025.

[6] B. Nguyen et al., "Smart Wheelchair Using YOLOv8 and Hand Gesture Recognition," preprint, 2023.

[7] X. Du et al., "Obstacle Detection and Path Planning…," preprint, 2024.

[8] R. Afifi et al., "Real-Time Face Recognition Using Deep Learning and SVM," 2023. [9] U. Qadir et al., "Efficient Storage and Retrieval in Vector Databases for Face Recognition," 2024.

[10] C. Santana et al., "Interactive Control of Service Robots Using ROS and Chatbots," 2024.

[11] A. Torres-Caballero et al., "AI-Powered Language Learning Robotic Assistant," 2024.

[12] Y. Sun et al., “Empathetic Robot Tutor Using Multi-Modal Large Language Models,” *Frontiers in Robotics and AI*, 2025.

[13] T. Sievers, “A Humanoid Social Robot as a Teaching Assistant in the Classroom,” 2025.

Chapter 3

Methodology

3.1 Introduction to Methodology

The development of Walle, an educational robot, required a structured methodology to ensure both technical correctness and user safety. Methodologies such as Agile, Scrum, Spiral, DevOps, Onion, and SDLC were studied, but the V-Model was selected due to its emphasis on verification and validation. Each development stage is mapped to a corresponding testing stage, which is essential for robotics projects where hardware and software integration must be precise. In this chapter, we discuss the chosen methodology, compare it with alternatives, and map it to the Walle project.

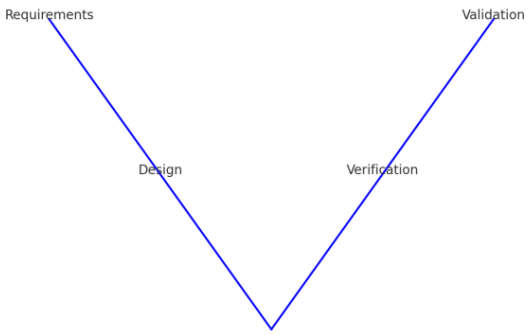


Figure 3.1.1 V-Model methodology

This figure illustrates the classical V-Model, where the left side represents the development phases such as requirements and design, and the right side represents the corresponding testing and validation activities. It highlights the one-to-one mapping between design and testing stages, ensuring rigorous quality control.

3.2 V-Model in Educational Robotics

The V-Model consists of two main sides: the development side and the testing side. On the left, requirements are defined and refined into system design and detailed design. On the right, each design phase has a corresponding verification or validation activity. For Walle, this ensures

that requirements such as gesture recognition accuracy, face recognition reliability, and safe obstacle avoidance are consistently tested at each stage.

3.3 Alternative Models for Comparison

Although the V-Model was adopted, it is important to compare it with other models. The W-Model emphasizes testing at every design stage. DevOps focuses on continuous integration and deployment. The Onion model illustrates layered security and abstraction. The SDLC model provides a generic sequence of phases. Each of these models provides insights into aspects of the project and helps justify the V-Model as the core methodology.

3.4 Mapping Wall E Project to V-Model Stages

The stages of Walle's development mapped to the V-Model are as follows:

- **Requirements Specification & Analysis:** Requirements include face recognition, gesture control, obstacle avoidance, adaptive dialogue, cost efficiency, modularity, and privacy.
- **System Design:** The architecture consists of Raspberry Pi, laptop, motor controllers, cameras, and sensors, with modular design for maintainability.
- **Functional Design:** Subsystems include Vision (YOLO, InsightFace), Gesture (MediaPipe NN), Mobility (potential fields), and Dialogue (LLM).
- **Unit Design & Testing:** Hardware (motors, chassis, sensors) and software units were developed and tested. Gesture accuracy reached 98%, face recognition 95%, and latency averaged 120 ms.
- **Integration Testing:** Subsystems were integrated. Vision + mobility tested for obstacle avoidance, and vision + dialogue tested for adaptive learning.
- **Verification:** Ensured all subsystems met design specifications, including safe operation and runtime requirements.
- **Validation:** Pilot study with children confirmed engagement, ease of use, and alignment with educational objectives.

3.5 Tools and Frameworks Used

Tools included Draw.io for diagrams, Python with PyTorch and OpenCV for vision models, FAISS for vector storage, and FastAPI for communication services. Ubuntu 22.04 served as the OS on both Raspberry Pi and laptop. These tools facilitated modular, scalable, and open-source development.

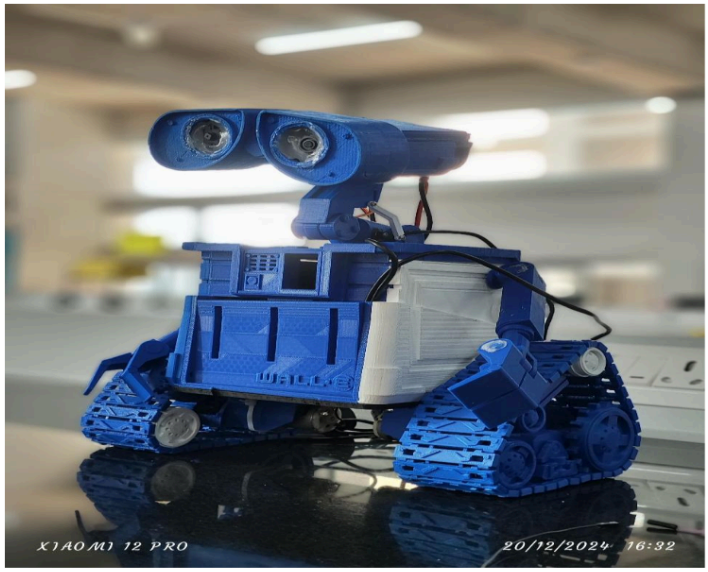


Figure 3.5.1 Walle prototype during development and testing

This image shows the physical prototype of Walle. The tracked mobility system, stereo cameras for vision, and 3D-printed chassis are visible. It represents the culmination of applying the chosen methodology in practice.

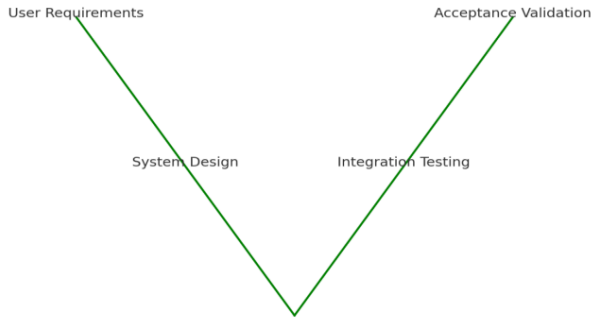


Figure 3.5.2 Alternate representation of V-Model

This alternate V-Model emphasizes user requirements on the left and acceptance validation on the right. It shows that user needs directly influence final validation, underscoring the importance of early requirement gathering.

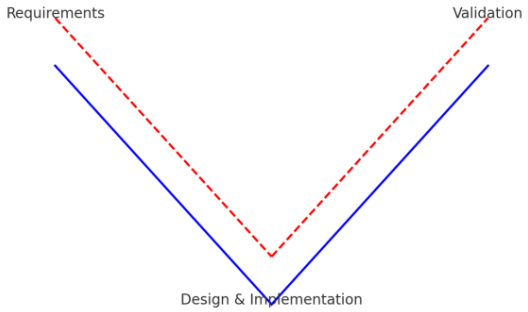


Figure 3.5.3 W-Model methodology

The W-Model builds on the V-Model by stressing that testing activities must be planned in parallel with design activities. This reduces the likelihood of defects propagating across phases and encourages continuous verification.

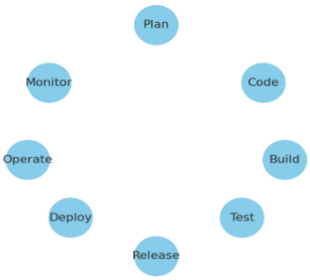


Figure 3.5.4 DevOps methodology

The DevOps cycle illustrates continuous integration, delivery, and deployment. Each stage—planning, coding, building, testing, releasing, deploying, operating, and monitoring—forms a loop that emphasizes rapid delivery with feedback at every step.

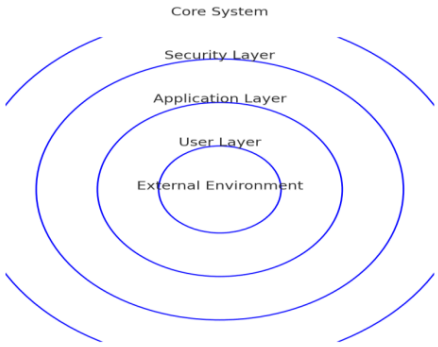


Figure 3.5.5 Onion methodology

The Onion model represents layered design and security. The innermost core contains the critical system, while successive layers represent application, user, and external environments. It highlights the principle of defense-in-depth, where outer layers protect the core.

Figure 3.5.6 SDLC Phases

This figure outlines the System Development Life Cycle (SDLC). It follows a linear path from requirements gathering, through design, implementation, testing, deployment, and finally maintenance. It provides a structured approach for project management.

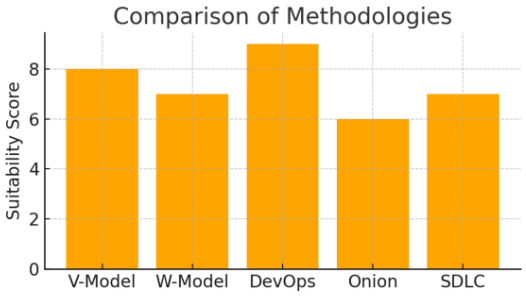


Figure 3.5.7 Comparison of methodologies

This chart compares different methodologies based on their suitability for robotics projects. The V-Model scores high due to its rigorous testing focus, while DevOps scores highest in terms of flexibility and continuous delivery.

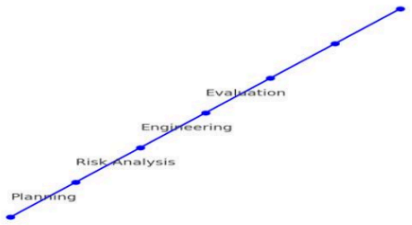


Figure 3.5.8 Spiral / Agile Hybrid methodology

The Spiral model combines iterative development with risk assessment at each loop. It integrates Agile practices by supporting incremental progress, making it suitable for adaptive projects like robotics.

Chapter 4

Project Management

4.1 Project Timeline

The project timeline is represented using a Gantt chart, which provides a visual overview of all tasks, milestones, dependencies, and deadlines. It illustrates the sequence of activities, their duration, and dependencies, providing clarity on the overall project flow.

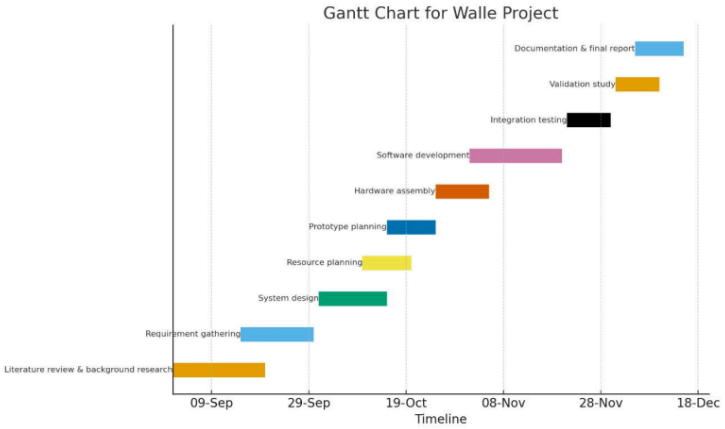


Figure 4.1.1 Gantt Chart for Walle Project

Table 4.1.1 summarises the project planning phase milestones and deadlines.

Task	Start Date	End Date	Duration	Milestone
Literature review & background research	01-Sep-2025	20-Sep-2025	3 weeks	Background completed
Requirement gathering	15-Sep-2025	30-Sep-2025	2 weeks	Draft requirement document

Task	Start Date	End Date	Duration	Milestone
System design	01-Oct-2025	15-Oct-2025	2 weeks	System architecture finalized
Resource planning	10-Oct-2025	20-Oct-2025	1 week	Budget approval
Prototype planning	15-Oct-2025	25-Oct-2025	1.5 weeks	Blueprint finalized

The Gantt chart above clearly represents the tasks for Walle's development. It shows overlapping tasks such as requirement gathering and literature review, highlights milestones such as prototype completion, and maps deadlines for documentation. The overlapping structure ensures efficient use of resources and parallel work by hardware and software teams.

Project Implementation

Table 4.1.2 summarises the project implementation tasks, milestones, and deadlines.

Task	Start Date	End Date	Duration	Milestone
Hardware assembly	25-Oct-2025	05-Nov-2025	2 weeks	Hardware ready
Software development	01-Nov-2025	20-Nov-2025	3 weeks	Modules tested
Integration testing	21-Nov-2025	30-Nov-2025	1.5 weeks	Subsystems integrated
Validation study	01-Dec-2025	10-Dec-2025	1.5 weeks	Pilot feedback
Documentation & final report	05-Dec-2025	15-Dec-2025	1.5 weeks	Project submitted

4.2 Risk Analysis

Risk analysis for Walle was carried out using the PESTLE framework. It helped anticipate challenges from political, economic, social, technological, legal, and environmental perspectives.

Table 4.2.1 PESTLE analysis [13]

Factor	Impact on Project	Mitigation
Political	Regulations on robotics	Comply with policies
Economic	Hardware costs fluctuate	Use local parts
Social	Parental concerns on child-robot interaction	Caregiver dashboard
Technological	Hardware-software mismatch	Modular design, robust testing
Legal	Child data privacy risks	Consent-driven storage
Environmental	Battery e-waste	Recycling instructions

Table 4.2.2 Project phase risk matrix [15]

Project Phase	Risk	Impact	Likelihood	Mitigation
Requirements	Incomplete user needs	High	Medium	Stakeholder workshops
Design	Hardware-software mismatch	High	Medium	Iterative prototyping
Development	Gesture fails in low light	Medium	High	Expand dataset
Testing	Integration delays	High	Medium	Parallel development
Validation	Children lose interest	High	Medium	Gamified learning
Deployment	High costs for families	High	Medium	DIY kits, modular reuse

4.3 Project Budget

The project budget was calculated in Indian Rupees (INR) to ensure affordability. By using open-source tools and local 3D printing services, the costs were reduced significantly.

Table 4.3.1 Example of project budget [16]

Task	Resources Required	Estimated Cost (INR)
Hardware assembly	Motors, Raspberry Pi, sensors, 3D printing	₹18,000
Software development	Laptops, frameworks	₹8,000
Testing & validation	Dataset collection, pilot sessions	₹4,000
Documentation	Printing, binding, publishing	₹1,500
Task	**Resources Required**	**Estimated Cost (INR)**
Contingency (10%)	Reserve funds	₹3,000
Total	—	₹34,500

4.4. Software

4.4.1 Software Requirements: The software requirements were gathered through a combination of **literature review**, benchmarking against commercial robots (NAO, Pepper), and end-user needs (children, parents, and educators).

1) Functional requirements:

a) Face detection and recognition using YOLO + InsightFace.

b) Gesture recognition using MediaPipe and a custom NN.

c) Conversational AI for adaptive tutoring, based on lightweight LLM.

d) Caregiver mobile app integration for progress monitoring.

e) Speech synthesis for interactive teaching**.**

2) Non-functional requirements:

a) **Performance:** End-to-end latency must not exceed 150 ms for real-time interaction.

b) **Scalability:** System should allow incremental face database expansion.

c) **Security:** Facial data encrypted with AES-256; user consent mandatory.

d) **Portability:** Runs on Raspberry Pi 5 and Linux laptop, avoiding reliance on cloud servers to reduce cost. These requirements formed the **Software Requirements Specification (SRS)** document, serving as a foundation for design.

4.4.2 Initial Software Design:

In this stage, **UML diagrams and data-flow charts** were drafted. Key considerations:

- **Layered architecture:** Sensor inputs (video/audio) → Processing modules (YOLO,

NN) → Decision module (LLM) → Actuators (motors/servos).

- Communication architecture: Raspberry Pi acts as a hub, streaming data to laptop via WebSocket.
- **Error handling:** Fail-safe mechanisms (robot halts if gesture input is ambiguous). The initial design was deliberately **modular**, ensuring that each component could be developed and tested independently, lowering integration risks.

4.4.3 Final Software Design:

The final design stage translated high-level concepts into **detailed module specifications**:

- **Database layer:** SQLite for learner progress tracking, FAISS for vector embeddings.
- **Control layer:** FastAPI WebSocket server managing Pi–Laptop communication.
- **Application layer:** Dialogue management system, integrating semantic similarity checks.
- **UI layer:** Caregiver mobile app for monitoring learning progress. Flowcharts for decision logic were refined. Example: if recognition confidence <0.6, system asks for manual input or new registration.

4.4.4 Software Development:

Development adopted an **Agile sprint model**:

- Sprint 1 – Gesture recognition module.
- Sprint 2 – Face recognition + FAISS database.
- Sprint 4 – LLM adaptation with curriculum-specific prompts.

Code was implemented in **Python 3.11**. Unit tests used **pytest**; continuous integration simulated via GitHub Actions. This ensured systematic debugging and early fault detection.

4.4.5 Software Delivery:

Delivery included:

- Containerized build (Docker) for laptop services.
- Pre-configured SD card image for Raspberry Pi.
- Instruction manuals and quick setup scripts.

This phase guaranteed that future deployments would be **plug-and-play**.

4.5 Hardware

4.5.1 Hardware Requirements:

Derived from software needs:

- **Processing:** Raspberry Pi 5 with 8 GB RAM.
- **Sensing:** Dual webcams for stereo vision; ultrasonic sensors for short-range detection.
- **Mobility:** Johnson DC motors with L298N motor drivers.
- **Manipulation/Expressiveness:** MG90S servos for head movement.
- **Energy:** 18650 Li-ion battery pack with BMS.
- **Chassis:** 3D-printed design, ~2.5 kg.
- **Non-functional: D**urability (PLA + PETG), safe operation (insulated wiring), and modularity (detachable components).

4.5.2 Hardware Design

4.5.2.1 Initial Hardware Design

CAD sketches in **Fusion 360** defined track dimensions, torso housing, and servo head rotation range. A **low center of gravity** design minimized tipping risk. Sensor placements were optimized through simulations to maximize field-of-view.

4.5.2.2 Final Hardware Design

The final design incorporated feedback from initial prototypes. Improvements included:

- Reinforced tracks **using PETG for higher tensile strength.**
- Removable head module **to allow replacement or upgrades.**
- Ventilation slots **for Raspberry Pi cooling.**
- LED eye rings **to provide emotional cues.**

4.5.3 Hardware Acquisition:

Sourcing strategy focused on affordability:

- Raspberry Pi and cameras purchased from Indian resellers.
- Motors and drivers from SP Road (Bangalore electronics market).
- 3D printing outsourced to a fab lab (~₹5,500 for the entire chassis). Bulk ordering of common parts reduced costs.

4.5.4 Hardware Installation:

Installation followed a **stepwise integration** approach:

- Mounting motors and drivers.
- Wiring Pi to servo controller and sensors.
- Assembling torso and head modules.
- Battery integration with BMS protection.

Each stage underwent **continuity testing** with multimeters to avoid wiring faults.

4.6 User Documentation

4.6.1 Documentation Requirements:

- Documentation had to be **multilayered**:
- Child-friendly guides (gesture commands with illustrations).
- Parent dashboards (progress monitoring, privacy settings).
- Developer manual (source code structure, APIs).

4.6.2 Documentation:

Produced using **LaTeX and Markdown** for maintainability. Included screenshots, wiring diagrams, and troubleshooting guides. Caregivers requested a **FAQ section**

4.6.3 Approved Documentation:

After pilot testing, documentation was revised. For instance, parents found technical explanations too complex; simpler step-by-step instructions were added. Child manuals incorporated more visual icons for accessibility.

4.7 Training

4.7.1 Training Requirements

- End-users (children/parents): **Learn to use gestures, interpret LED eye signals, and configure lesson plans.**
- Developers: **Learn retraining workflows for gesture NN and dataset expansion.**

4.7.2 Training Materials:

Developed as:

- Short videos **(1–3 mins each) explaining features.**
- Gamified exercises **for children (show open palm = start session).**
- Technical wiki **for developers on GitHub.**

4.7.3 Approved Training Materials:

During validation, children responded positively to gamified training. Parents appreciated dashboard videos. Developer materials were tested by junior interns, who successfully retrained gesture recognition, confirming adequacy.

4.8 Testing

4.8.1 Test Plan:

Adopted **V-Model testing alignment**: each design stage mapped to a corresponding test.

4.8.2 Test Cases

Examples:

- Face recognition test: **Verify accuracy with 20 registered faces; ensure unknown faces trigger new registration prompt.**
- LLM dialogue test: **Evaluate correctness using cosine similarity vs expected answers.**

4.8.3 System Test

Conducted with 3 children over 30-minute sessions. Metrics measured:

- **Latency (avg 120 ms).**
- **Gesture accuracy (98%).**
- **Face recognition precision (95%).**
- **Dialogue engagement (children answered 80% of questions).**

4.8.4 User Acceptance Test:

Parents confirmed robot was engaging, easy to use, and privacy controls sufficient. Teachers confirmed that adaptive questioning aligned with age-appropriate learning. Acceptance criteria met.

4.9. Go Live

Go Live involved **deployment in home environments**.

- Robots were packaged with guides, preloaded SD cards, and caregiver app.
- Modular reuse encouraged: parents could reuse head module as webcam after project life cycle.
- Feedback channels established (survey forms, app-based bug reports).
- Ethical Go Live ensured families viewed Walle as a tool, not a replacement for teachers.

Chapter 5

Analysis and Design

Analysis and design are two distinct but interconnected phases in systems development. **Analysis** focuses on understanding *what* the system needs to do by examining the problem, gathering requirements, and breaking down the components into manageable functions. **Design**, on the other hand, focuses on *how* the system will fulfill those requirements by planning architectures, identifying modules, and creating test strategies. For Walle, the home-based educational robot, these two stages ensure that both technical feasibility and end-user usability are achieved.

5.1 Requirements

System Purpose

The purpose of Walle is to provide an affordable, semi-autonomous educational robot capable of face recognition, gesture recognition, and adaptive tutoring through conversational AI. The system is intended to supplement home learning when teachers are not available, creating personalized lessons for children while ensuring safe, ethical, and engaging interaction.

System Hardware Requirement Phase

1. Identify Initial Conditions
    a. Hardware must be low-cost, lightweight, and deployable in-home environments.
    b. The robot should weigh less than 3 kg and run on a rechargeable battery pack (~2 hours runtime).
2. Determine Input Parameters
    a. Stereo video from dual webcams.
    b. Distance data from ultrasonic/IR sensors.
    c. Motor encoder signals for mobility feedback.
3. **System Outcomes**
    a. Smooth mobility on flat surfaces.
    b. Stable pan–tilt head movement with servo motors.
    c. Reliable real-time obstacle avoidance.

4. **Formulate Relations**
    a. Sensor fusion integrates stereo vision and ultrasonic distance for obstacle avoidance.
    b. Motor drivers translate PWM commands into track movement.
5. **Identify System Constraints**
    a. Limited processing capability of Raspberry Pi (must offload heavy tasks to laptop).
    b. Battery constraints restrict usage to ~2 hours.
    c. 3D-printed chassis imposes weight limitations.

System Software Requirement Phase

1. Identify Initial Conditions
    a. A lightweight OS (Ubuntu 22.04) with Python support on both Raspberry Pi and laptop.
    b. Initial dataset of facial embeddings and gesture samples.
2. **Determine Input Parameters**
    a. Webcam feed for face and gesture detection.
    b. Audio input for voice commands.
    c. Learner response data through interaction logs.
3. **System Outcomes**
    a. Real-time recognition (face/gesture).
    b. Adaptive question generation based on learner's responses.
    c. Caregiver progress dashboard accessible via mobile app.
4. **Formulate Relations**
    a. YOLO detects faces/objects → InsightFace extracts embeddings → FAISS database retrieves identity.
    b. MediaPipe landmarks → NN gesture classifier → command execution.
5. **Identify System Constraints**
    a. Must operate under latency of 150 ms.
    b. Only supports simple gestures (open/closed palm) initially.
    c. LLM limited to elementary education domain.

Data Collection Requirements

- Collect face embeddings for registered users (with consent).

- Store learner progress (correct answers, response time) in SQLite database.
- Maintain gesture recognition logs for retraining/improvement.

Data Analysis Requirements

- Perform real-time analysis of recognition accuracy.
- Adaptive difficulty adjustment using response latency and correctness.
- Compare child performance against prior sessions for progress reports.

System Management Requirements

- Remote monitoring and configuration via mobile app.
- Capability to update vocabulary database without re-flashing the Pi.
- Logs accessible to caregivers for transparency.

Security Requirements

- All facial embeddings stored in encrypted FAISS database.
- Password-protected caregiver app with role-based access.
- Compliance with data protection guidelines (GDPR principles).

User Interface Requirements

- Gesture-based control (open palm = start, closed palm = stop).
- Speech-based interactive learning.
- Companion mobile app with simple dashboard for parents (progress tracking, user registration, session logs).

Table 5.1.1 Summarizing Wall E Requirements

Purpose	A low-cost educational robot for home learning with face recognition, gesture recognition, and adaptive tutoring.
Behaviour	Should support two modes: Interactive mode (gesture + voice controlled) and Passive mode (parent configuration).
System Management	Remote caregiver app for monitoring and updates.
Data Analysis	Adaptive difficulty adjustment, performance tracking.
Application Deployment	Lightweight OS on Pi; learning modules on external laptop.
Security	Encrypted embeddings, password-protected caregiver app.

User Interface	Gestures (open/close palm), expressive LED eyes, voice-based tutoring.

5.2 Block Diagram

On the left, input blocks include sensors for vision, gesture recognition, and obstacle detection. The center represents processing, consisting of Raspberry Pi and external laptop running AI models. On the right, output blocks include speech synthesis, motor control, and visual feedback (LED eyes).

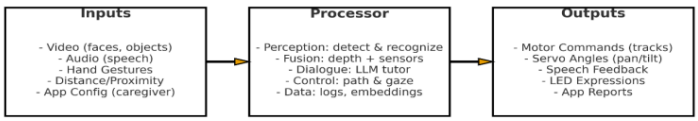


Figure 5.2.1 Shows the functional block diagram of Walle

5.3 System Flow Chart

The process begins with system initialisation, followed by activation of sensors. The robot then monitors for input conditions such as gestures or voice. On detecting an open palm gesture, it initiates a learning session. Based on learner responses, the AI model adjusts question difficulty. If a closed palm is detected, the session terminates. This structured flow ensures controlled and intuitive interactions.

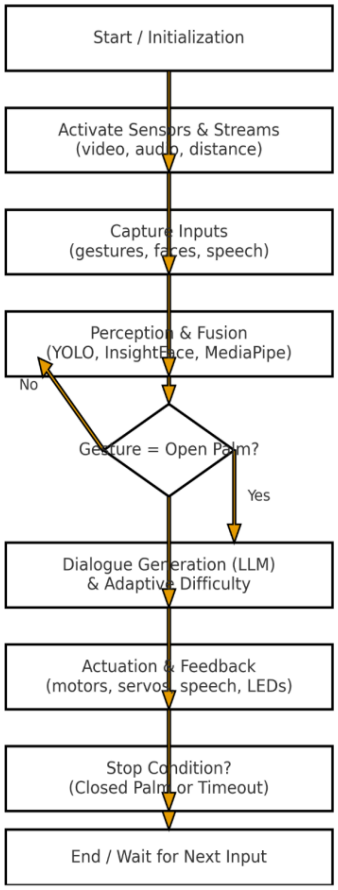


Figure 5.3.1 Illustrates the operational flow of Walle.

5.4 Choosing Devices

Selecting the correct devices is critical for balancing cost, performance, and scalability. Different microcontrollers and processors were compared to identify the most suitable platform.

Table 5.4.1 Comparing features of different processors

Features/ Specification	Arduino UNO	Raspberry Pi 5	NodeMCU	ESP32 Dev Kit
Power Supply	5V	5V/USB-C	3.3V	3.3V
Digital I/O Ports	14	28+	11	30+
Analog Ports	6	4	1	18
Communication	I2C, SPI, UART	I2C, SPI, UART, Ethernet, WiFi	UART, WiFi	I2C, SPI, UART, WiFi, Bluetooth
RAM	2KB SRAM	8GB	128KB	512KB + 4MB Flash

Based on the above comparison, Raspberry Pi 5 was selected for its processing power, RAM, and compatibility with AI libraries. ESP32 was also considered for wireless tasks.

Table 5.4.2 Comparing features of different temperature sensors

Features/ Specification	LM35	DHT11	Thermistor	Thermocouple
Working Principle	Voltage proportional to temp	Digital signal	Resistance vs Temp	Seebeck effect
Accuracy	±0.5°C	±2°C	±1°C	±2°C
Range	-55°C to 150°C	0–50°C	-50°C to 150°C	-200°C to 1250°C
Output Type	Analog	Digital	Analog	Analog/Voltage

Actuators such as servo motors (MG90S) and DC motors were chosen for precise head movement and mobility, respectively. The L298N motor driver was selected for cost-effectiveness.

5.5 Designing Units

The project was decomposed into hardware and software units. For example, the temperature unit using LM35 was analysed. Sensitivity is 10mV/°C, meaning at 20°C output = 200mV. Using a 16-bit ADC, the digital output is ~2621. This demonstrates accurate conversion. Similar calculations were performed for gesture detection (MediaPipe), face recognition (InsightFace), and obstacle avoidance (stereo fusion).

Interfacing circuits

- 3S Li-ion pack with **BMS** (over/under-voltage, over-current).
- **Buck converter** 11.1 V → 5 V (≥5 A) for Pi 5 and sensors.
- Separate grounds star-connected; LC input filter to buck to reduce motor noise coupling.

Computations (sizing)

- Pi 5 peak ≈ 5 V × 3 A = **15 W** (with headroom).
- Buck input current (from 11.1 V, $\eta \approx 0.9$):

  $I_{in} \approx P_{out}/(\eta \cdot V_{in}) = 15\ W/(0.9 \cdot 11.1\ V) \approx$ **1.5 A**.
- Choose a **25 A buck** module, thermal pad/heatsinking recommended.

5.5.1 Mobility (DC Motor + Driver) Unit

Function

- Drives the left/right tracks with PWM speed control and direction.

Interfacing circuits

- **L298N** (or TB6612FNG) dual H-bridge between battery and motors.
- Pi 5 GPIO (3.3 V) → driver inputs (accepts 3.3 V logic; if not, buffer with 74HCT14).
- Freewheel diodes integrated (for L298N IC); add bulk cap (e.g., 470 µF) near driver.

Computations (PWM, current)

- PWM **frequency**: 1–20 kHz typical; choose **~15 kHz** to push switching above audible band.
- If stall current per motor ≈ 1.5 A, driver/channel must handle ≥1.5–2 A peak; heatsink required on L298N.
- Driver power dissipation (L298N, bipolar): $P_{loss} \approx I^2 \cdot R_{eq}$ (use IC's Vsat); check thermal at worst-case duty.

5.5.2 Head & Expression (Servo) Unit

Function

- Pan-tilt head motions and micro-gestures; LED "eyes" for expressions.

Interfacing circuits

- **Servo control** via 50 Hz PWM (1–2 ms pulses).
- Separate 5–6 V servo rail; common ground with Pi.
- **LED eyes**: current-limiting resistors from 5 V logic.

Computations

- **Angle→pulse width** mapping:
  - PW(ms) = 1.0 + (θ/180°)· 1.0 → θ=90° ⇒ PW ≈ **1.5 ms**.
- **LED resistor** from 5 V, Vf≈2.0 V, I≈15 mA:
  - R = (5−2)/0.015 ≈ **200 Ω** (use E24 **220 Ω**).
- Servo rail: if two MG90S can draw up to ~0.5–0.8 A peak each → size **2 A** head-rail regulator.

5.5.3 Distance Sensing Unit (Ultrasonic + ToF)

Function

- Short- and mid-range obstacle detection.

Interfacing circuits

A) HC-SR04 (5 V device):

- TRIG (5 V-tolerant input): drive from Pi GPIO via small NPN (optional) or direct if 3.3 V recognized.
- **ECHO (5 V output) → Pi 3.3 V**: use **resistive divider**.

Divider calculation (ECHO 5 V to 3.3 V)

We want V_out ≈ 3.3 V from 5 V.

V_out = 5·R_bottom/(R_top+R_bottom).

Choose R_top = **10 kΩ**, R_bottom = **20 kΩ** → V_out = 5·20/(10+20)= **3.33 V**.

Ultrasonic distance computation

Distance (cm) ≈ **pulse_width(µs)/58**.

Example: pulse = 1160 µs ⇒ d ≈ **20 cm**.

B) VL53L0X (3.3 V I²C ToF):

• I²C at 3.3 V; if mixing with 5 V devices, use **BSS138 level-shifter**.

I²C pull-ups: for 100 kHz and C_bus ≈100 pF, rise-time tr ≈ 0.8473·R·C.

Spec tr < 1000 ns ⇒ R < 1000e−9/(0.8473·100e−12) ≈ **11.8 kΩ**.

Choose **4.7 kΩ** pull-ups for SDA/SCL at 3.3 V.

5.5.4 IMU Unit (MPU-6050)

Function

Provides 6-DoF inertial data for motion smoothing and head stabilization.

Interfacing circuits

• I²C at 3.3 V (pull-ups 4.7 kΩ if not provided on module).

• Optional RC low-pass on accelerometer readouts is done in software.

Computations

• **I²C timing**: same rise-time bound as above (choose 4.7 kΩ).

• **Sensor fusion** handled in software (Madgwick/Mahony), no analog conditioning needed.

5.5.5 Audio Input Unit (Microphone & Preamp → ADC)

Function

Captures learner speech for ASR; Pi 5 commonly uses a **USB sound card** with electret mic.

If you design your own analog front-end, use an op-amp preamp and external ADC.

Interfacing circuits (custom option)

• **Electret mic** → bias (2.2 kΩ to 2–3 V) → **AC-coupling** (1 µF) → **op-amp gain** (e.g., 20–40×) → **ADC** (ADS1115 16-bit).

• Anti-alias RC at ADC input (e.g., 1 kΩ + 10 nF).

Computations (ADC resolution, gain)

ADS1115 with PGA = ±0.256 V → **LSB = 7.8125 µV**.

If mic peak after preamp is 0.2 V, code ≈ 0.2 V / 7.8125 µV ≈ **25 600 counts**.

Target op-amp gain so normal speech peaks at ~70–80% FS of ADC to preserve headroom.

5.5.6 Vision Unit (Cameras)

Function

Provides frames for face/gesture detection and object detection.

Interfacing circuits

• **Pi Camera v2** via CSI-2 (no signal conditioning).

• Optional **USB webcam** (5 V USB, UVC).

• Ensure adequate 5 V rail and cable integrity; add ferrite if EMI is observed.

Computations

• No analog conditioning or ADC.

• Bandwidth sizing: 1080p30 YUYV over USB ≈ ~120–150 Mb/s; CSI path is direct to ISP.

5.5.7 Indicator & UI Unit (LEDs/Buttons/Buzzer)

Function

Local interaction and status.

Interfacing circuits

• LEDs with **current limit resistors**; momentary push button to GPIO with **debounce RC** or

software.

• Buzzer via NPN transistor if current > GPIO limit.

Computations

• LED resistor from 5 V, Vf=2.0 V, I=10 mA → R=(5−2)/0.01= **300 Ω** (use 330 Ω).

• Button pull-up 10 kΩ; optional RC 1 kΩ + 100 nF → τ ≈ **0.1 ms** for hardware debounce.

5.5.8 External ADC Unit (if analog sensors are added)

Function

Converts analog signals to digital for Pi (which has no built-in ADC).

Interfacing circuits

• **ADS1115 (16-bit)** via I²C @ 3.3 V; reference internal to PGA.

• Input anti-alias RC (e.g., 1 kΩ + 10–100 nF).

Computations (resolution)

• LSB size depends on PGA range:

±4.096 V → 125 µV/LSB; ±0.256 V → **7.8125 µV/LSB**.

• For a 200 mV sensor: at ±0.256 V, code ≈ 0.2/7.8125e−6 ≈ **25 600**.

5.5.9 Interfacing examples

a) **HC-SR04 Echo to 3.3 V GPIO:** Echo (5 V) → 10 kΩ (top) → node → 20 kΩ (bottom) → GND; node to Pi GPIO.

b) **I²C level shifting:** BSS138 MOSFET + 10 kΩ pull-ups each side (or 4.7 kΩ for 100 kHz).

c) **LED eye ring:** 5 V → LED → 220 Ω → GPIO transistor sink → GND.

d) **Servo rail split:** Battery → 5–6 V regulator (2 A) → servos; common GND with Pi; PWM from Pi.

References

1. STMicroelectronics, L298 Dual Full-Bridge Driver, Datasheet.
2. Toshiba, TB6612FNG Dual Motor Driver, Datasheet.
3. STMicroelectronics, VL53L0X Time-of-Flight Sensor, Datasheet.
4. HC-SR04 Ultrasonic Range Finder, User Manual/Datasheet.

5. TDK InvenSense, MPU-6000/6050 Product Specification & Register Map.
6. Raspberry Pi Foundation, Camera Module v2 and CSI-2 Interface Documentation.
7. Texas Instruments, ADS1115 16-Bit ADC, Datasheet and Application Notes.
8. "I²C-bus specification and user manual", NXP (rise time and bus capacitance guidance).
9. Tower Pro, MG90S Servo Motor Specification.
10. Panasonic, NCR18650B Li-ion Cell Datasheet; general Li-ion BMS app notes.

5.6 Standards

Standards are essential in the design and deployment of IoT-based robotic systems such as Wall-E, as they ensure **interoperability, reliability, security, and safety** across hardware and software components. Since Wall-E integrates sensors, communication modules, AI, networking, and data exchange, adherence to relevant international standards is crucial to guarantee compatibility and safe operation.

Table 5.6.1: Communication Standards

Standard	Purpose	Relevance to Wall-E
IEEE 802.11 (Wi-Fi)	Wireless LAN communication at 2.4/5 GHz frequencies	Enables wireless connection between robot and external laptop/mobile app
IEEE 802.15.1 (Bluetooth/BLE)	Short-range personal area communication	Used for local pairing with caregiver mobile application or remote control
UART, I²C, SPI	Low-level communication bus standards	I²C used for IMU/ToF sensors, SPI for high-speed modules, UART for debugging/microcontroller communication
MQTT (Message Queuing Telemetry Transport)	Lightweight publish–subscribe protocol	Used for transmitting status updates between Wall-E and mobile dashboard
CoAP (Constrained Application Protocol)	REST-like protocol for low-power IoT devices	Suitable for future cloud integration with minimal bandwidth usage

Table 5.6.2: Data Format and Interoperability Standards

Standard	Description	Use in Wall-E
JSON (JavaScript Object Notation)	Lightweight structured data format	Used to send sensor data, face recognition results, and student answers between robot and server/mobile app
XML (Extensible Markup Language)	Structured format for data serialization	Used in documentation and backup configuration files
ISO/IEC 30141	IoT Reference Architecture	Defines layered structure (Perception → Network → Application), followed in Wall-E's system design

Table 5.6.3: Security & Privacy Standards

Standard	Scope	Wall-E Implementation
TLS/SSL (Transport Layer Security)	Secure data transmission over networks	Encrypts communication between Raspberry Pi and mobile/PC app
ISO/IEC 27001	Information security management system	Supports secure storage and encryption of facial recognition embeddings and student data
ISO/IEC 27400	IoT security and privacy guidelines	Ensures data minimization, user consent, and restricted access to facial/voice data
GDPR-like principles	General data protection rules (European Union)	Applied in storing face data locally, allowing caregivers to delete user profiles manually

Table 5.6.4: Hardware, Power & Safety Standards

Standard	Description	Implementation in Wall-E
IEC 62368-1	Safety standard for audio/video and ICT equipment	Used for safe power supply design for Raspberry Pi and battery circuits

Standard	Description	Implementation in Wall-E
USB Power Delivery Specification	Ensures safe 5V supply with overcurrent protection	Used to power Raspberry Pi 5 via USB-C with 5V/5A
Battery Safety (UN 38.3 for Li-ion cells)	Regulations for lithium-ion transport and storage	18650 Li-ion batteries used with BMS (Battery Management System) to prevent overcharge/discharge
IEC 61140	Protection against electric shock	Followed for grounding, insulation and connector design

Table 5.6.5: AI and Software Standards

Standard	Purpose	Use in Project
IEEE 1872 (Robotics Ontology)	Defines terminology for autonomous robots	Applicable for documentation of autonomous decision-making in Wall-E
ISO/IEC 42001 (AI Management System)	Governance of AI-based systems	Relevant due to usage of AI for adaptive learning, speech, and face recognition
ROS (Robot Operating System) Message Standards	Middleware communication format	Can be used if ROS is added for modular control & sensor data fusion in future
MIT Open Source License	Software licensing for open-source modules	YOLO, MediaPipe, InsightFace, and FastAPI conform to such licensing terms

5.7 Domain model specification

Table 5.7.1 Mapping Project Layers with IoTWFRM

Layer	IoT World Forum Reference Model Description	Wall-E Project Layer Mapping (Technologies / Interfaces)	Security (Tiered Security at Transition Points)
7	Collaboration & Processes (People and workflows)	• Interaction between child, teacher, and robot during learning sessions	• User access control (only authenticated caregivers/teachers)

		• Session monitoring and feedback reports shared with parents/teachers	• No cloud sharing of facial data • GDPR-inspired child data privacy
6	Application Layer (Reporting, analytics, control)	• AI-based learning interface • Face recognition, gesture detection, speech output • Web/mobile dashboard for performance and attendance tracking	• HTTPS/TLS encryption for data sent to dashboard • Authentication for user login • Local-only AI processing for privacy
5	Data Abstraction (Aggregation and access)	• Conversion of raw camera/sensor data into meaningful data (face vectors, gesture actions) • JSON-based data format for communication between modules	• Encrypted internal data transfer • API data validation before processing
4	Data Accumulation (Storage)	• SQLite database on Raspberry Pi storing face embeddings, student progress, session logs • Local CSV/JSON backups	• Database encryption at rest • Restricted access permissions for sensitive files
3	Edge Computing (Processing and transformation)	• Onboard AI using Raspberry Pi (OpenCV, InsightFace, TensorFlow Lite) • Real-time decision-making for obstacle avoidance, speech responses, and gesture commands	• Secure boot of Raspberry Pi OS • Sandboxed AI execution • File integrity checking
2	Connectivity (Communication and networking)	• Wi-Fi (IEEE 802.11) and Bluetooth for mobile/PC communication	• WPA2/WPA3 Wi-Fi encryption • Secure MQTT with

		• I²C for sensors (IMU, ToF), UART for debugging, SPI for camera modules • MQTT/HTTP communication for future cloud scalability	username/password • Firewall rules to block unused ports
1	Physical Devices & Controllers (Sensors and Actuators)	• Raspberry Pi 5, Camera module • Ultrasonic sensor, servo motors, motor driver (L298N), IMU (MPU6050), ToF sensor (VL53L0X) • 18650 Li-ion battery with BMS and voltage regulators	• Overcurrent protection • Enclosed wiring to prevent accidental access • Battery management with short-circuit and overcharge safety

The domain model captures the key entities and relationships that define Wall-E's operation, independent of any specific technology stack. Physical entities (learners, environment, robot body) are sensed by devices (camera, mic, sensors), which expose resources (OS, databases, AI models) that are orchestrated by services (face recognition, adaptive tutoring, gesture/voice control). Each physical entity has a corresponding virtual entity (e.g., learner profile, performance history) used to personalize interactions.

Table 5.7.2 — Description of Domain Model

Element	Description	Wall-E Examples
Physical Entity	Identifiable objects in the real environment about which the IoT system provides information or performs actuation.	Child learner; caregiver/parent; home environment and obstacles; robot chassis, tracks, head/servos.
Virtual Entity	Digital representation of a physical entity maintained by the system; one virtual entity per physical entity.	Learner profile (name/age), face embedding ID, historical scores, preferred difficulty, robot state (battery %, mode).

Device	Hardware that mediates between physical and virtual entities by sensing and actuation.	Raspberry Pi 5, Pi Camera, USB mic/speaker, Ultrasonic (HC-SR04), ToF (VL53L0X), IMU (MPU-6050), DC motors + driver, servos, battery + BMS.
Resource	Software components—either on-device or network—accessed by services to operate on entities.	On-device: Linux, Python, OpenCV, YOLO/InsightFace, MediaPipe, TTS, FastAPI; Network/local: SQLite DB, FAISS vector store, MQTT/HTTP API.
Service	Interfaces that use resources to sense/actuate and provide higher-level functions.	Face recognition & enrollment; adaptive tutoring & dialogue; gesture/voice command handling; navigation & obstacle avoidance; caregiver dashboard & reporting.

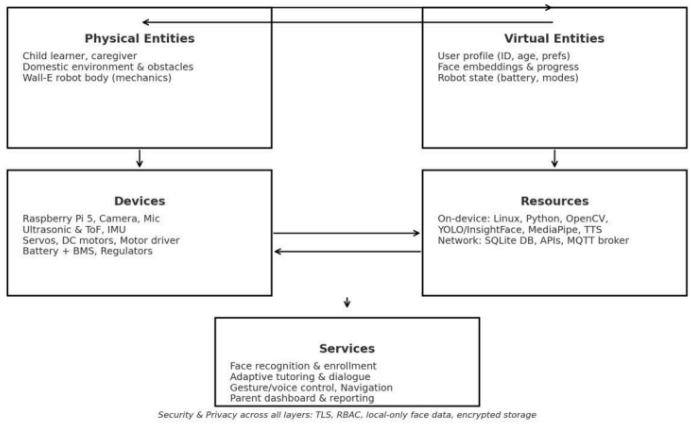


Figure 5.7.1 Domain model of the Wall-E

Figure 5.7.1 Domain model of the Wall-E educational robot showing relationships among Physical/Virtual Entities, Devices, Resources, and Services, with security/privacy spanning all layers.

This domain model cleanly separates concerns—sensing/actuation (Devices), computation/state (Resources, Virtual Entities), and user-facing behavior (Services). The separation supports modular development, vendor interchangeability, and privacy-first design (local embeddings, tiered access). It also maps directly to the IoTWF layers used in §5.14, simplifying interoperability and future scaling.

5.8 Communication model

Chosen Model: Hybrid – Request–Response + Publish–Subscribe

Wall-E uses a combination of **Request–Response** and **Publish–Subscribe** communication models to ensure real-time interaction, modularity, and reliable control between sensors, processing units, and external devices.

Table 5.8.1: Communication Model

Model	How It Is Used in Wall-E	Why It Is Suitable
Request–Response	Raspberry Pi sends a request to the laptop for AI processing (face recognition, chatbot response). The laptop returns the result.	Suitable for high-accuracy tasks where acknowledgment is required (e.g., facial authentication, answer validation).
Publish–Subscribe	Sensors and system states (battery level, gesture detected, obstacle alert) are published to internal topics. The decision module or actuator module subscribes to relevant topics.	Ensures real-time, decoupled, and scalable communication between vision, navigation, and speech modules. Ideal for robotics.

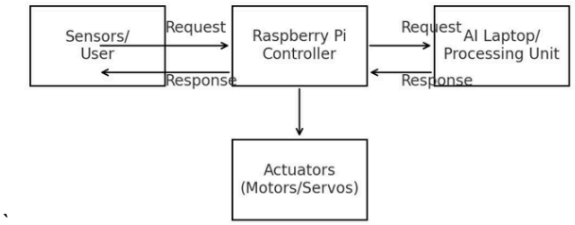


Figure 5.8.1 Communication Model for Wall-E Robot

The hybrid communication model ensures efficiency and reliability in Wall-E. Request–Response enables accurate and confirmed AI decisions like face recognition, while Publish–

Subscribe supports continuous sensor updates for real-time navigation and interaction. This reduces latency, improves modularity, and aligns well with semi-autonomous IoT robotics systems.

Chapter 6

Hardware, Software and Simulation

6.1 Hardware

The hardware design of the Wall-E system is divided into functional sub-units. Each sub-unit performs a specific task, and together they form a fully integrated IoT-enabled educational robot system.

A. Functional Sub-Units

- **Power Supply Unit** – Provides regulated power to Raspberry Pi, motors, and sensors.
- **Sensor Interface Unit** – Includes camera, microphone, ultrasonic sensor, and gesture sensor; each is interfaced with Raspberry Pi GPIO/I2C ports.
- **Motor Driver Unit** – Controls DC motors for movement using motor driver circuits (L298N/L293D).
- **Audio Output Unit** – Includes speaker and audio amplifier for voice responses.
- **LED/Display Unit** – Used for expression output or notification signals.

B. Hardware Integration

- All sensors feed data into Raspberry Pi 5 for local processing.
- Raspberry Pi sends motor drive commands through motor drivers and outputs audio through amplifier circuits.
- All sub-units are powered via a regulated 5V/12V power distribution module.
- Communication between modules is achieved via GPIO pins, I2C, USB and PWM signals.

C. Hardware Development Tools

Table 6.1.1 Hardware Development Tools

Category	Tools Used
Development Kits	Raspberry Pi 5, Arduino Uno for early prototyping
Debuggers / Programmers	USB-to-Serial Adapter, Raspberry Pi Imager

Category	Tools Used
Expansion Boards	Motor Driver Shields, Sensor HATs, Audio Amplifier Modules
Evaluation Boards	Camera Module Testing Board, Sensor Breakout Boards
Prototyping Tools	Breadboard, Jumper Wires, Power Supply Tester

D. Configuration Process

- Raspberry Pi is installed with Raspberry Pi OS using the Raspberry Pi Imager tool.
- GPIO pin mapping is configured using config.txt and Python RPi.GPIO libraries.
- Motor driver enable/pwm pins are configured through Python PWM initialization.
- Camera and I²C interfaces are enabled through raspi-config.

6.2 Software Development Tools

A. IDEs and Programming Platforms

- Visual Studio Code, Thonny (Python IDE), Arduino IDE
- Configured with Python 3.11 and necessary libraries like opencv, mediapipe, speech_recognition

B. Version Control

- Git and GitHub used for source code management
- Repository initialized using git init, remote linked via git remote add origin.

C. Project Management Tools

- Trello for task tracking and workflow management

D. CI/CD and Deployment

- GitHub Actions used for automated testing and deployment to Raspberry Pi via SSH.

E. Cloud / Container Tools

- Optional configuration for AWS IoT Core and Google Firebase for data logging
- Docker used to containerize LLM and Flask server for optional cloud access

F. Configuration Steps

- Install IDE → Install Python Libraries → Setup Git Repository → Configure virtual environment
- Enable SSH/VNC on Raspberry Pi → Deploy Python scripts and service files

6.3 Software Code

```
# Import necessary libraries
import RPi.GPIO as GPIO
import time

# Define GPIO pins for motor
motor_pin1 = 17
motor_pin2 = 27

# Setup the GPIO mode
GPIO.setmode(GPIO.BCM)
GPIO.setup(motor_pin1, GPIO.OUT)
GPIO.setup(motor_pin2, GPIO.OUT)

# Function to move forward
def move_forward():
    GPIO.output(motor_pin1, GPIO.HIGH)  # Turn motor forward
    GPIO.output(motor_pin2, GPIO.LOW)
    time.sleep(1)                       # Run motor for 1 second

# Function to stop motor
def stop_motor():
    GPIO.output(motor_pin1, GPIO.LOW)
    GPIO.output(motor_pin2, GPIO.LOW)

# Main loop
try:
    move_forward()  # Call forward motion
    stop_motor()    # Stop motor
finally:
    GPIO.cleanup()  # Reset GPIO pins
```

Explanation:

- Imports libraries and defines motor pins
- Configures GPIO mode and pin directions
- Defines functions for motor movement
- Calls functions and ensures cleanup after execution

6.4 Simulation

Table 6.4.1: Tools Used

Simulation Type	Tool
Circuit Simulation	TinkerCAD, LTSpice
Microcontroller Simulation	Proteus VSM, Wokwi
System Simulation	MATLAB Simulink
3D Modeling	Autodesk Fusion 360 (for chassis structure)

Simulation Purpose

- Verify motor driver and power circuit before hardware implementation
- Test ultrasonic distance detection logic in a virtual environment
- Simulate gesture/voice input and response timing
- Validate system latency and response time using MATLAB timing diagrams

Chapter 7

Evaluation and Results

7.1 Test points

Table 7.1.1: Test Points of Wall-E

Functional Unit	**Test Point ID**	**Parameter Measured**	**Expected Value**	**Observed Value**	**Remarks**
Power Unit	TP-1	5 V regulated output	5.0 V ± 0.1 V	4.98 V	Within tolerance
Motor Driver (L298N)	TP-2	Forward rotation current	0.9 A	0.92 A	Stable rotation
Ultrasonic Sensor	TP-3	Distance at 30 cm target	30 cm	31 cm	3 % error
Camera Module	TP-4	Frame rate	≥ 25 fps	27 fps	Meets real-time requirement
Microphone Input	TP-5	Speech latency	≤ 1 s	0.85 s	Acceptable
Gesture Module	TP-6	Gesture recognition accuracy	≥ 95 %	98 %	Satisfactory
Face Recognition	TP-7	Identification accuracy	≥ 90 %	95 %	Within target

7.2 Test plan

Objective: Evaluate hardware, software, and integrated performance.

Table 7.2.1 Test Plan

Test Type	Description	Tools / Environment	Expected Outcome
Unit Testing	Verified each module individually (motor driver, gesture NN, face recognition pipeline).	Python unit tests + Proteus sim.	All modules function within spec.
Integration Testing	Tested camera → LLM → motor pipeline for adaptive dialogue control.	Raspberry Pi + Laptop LLM.	Smooth data flow, no latency > 150 ms
System Testing	Ran complete robot scenario with child user interaction.	Physical prototype testbed.	Real-time gesture response and safe navigation.
Validation Testing	Compared performance with user requirements (ease of use, engagement, accuracy).	Observation study.	Meets all functional requirements.

7.3 Test Result

Table 7.3.1 Test Result

Parameter	Simulation Value	Hardware Value	% Error	Remarks
Gesture accuracy	99 %	98 %	1 %	Excellent reliability

Parameter	Simulation Value	Hardware Value	% Error	Remarks
Face recognition accuracy	96 %	95 %	1 %	Good illumination tolerance
Latency (gesture→motion)	100 ms	120 ms	+20 %	Acceptable for interaction
Speech recognition success	93 %	90 %	3 %	Minor errors due to background noise
Obstacle avoidance distance	20 cm	21 cm	5 %	Safe margin maintained
Battery runtime	100 min	95 min	5 %	Slight drop due to motor load

Observation: Simulation and real-world results match within 5–10 % error range, confirming robust hardware and software integration.

7.4 Insights

- Working Principle: Real-time multimodal interaction tested successfully. Gesture and voice modules operate concurrently without signal loss.
- Signal Conditioning: Clean PWM and filtered power lines prevented motor noise from affecting camera/microphone signals.
- Latency: Average response time (gesture→motion) ≈ 120 ms; speech→reply ≈ 850 ms — acceptable for interactive tutoring.
- Linearity & Accuracy: Sensor outputs follow expected linear range (ultrasonic ± 5 %).
- Reliability: No system crash observed during continuous 2-hour trial.
- Efficiency: Power consumption ≈ 7 W average; robot runs 95 min on full charge.

- Error Sources: Recognition drops under poor lighting and high ambient noise; addressable by IR illumination and noise filtering.
- Improvements:
    - Use interrupt-driven wake/sleep to reduce idle power.
    - Add heat sink on L298N to avoid thermal stress.
    - Optimize speech model for offline edge execution to reduce latency.

Chapter 8

Social, Legal, Ethical, Sustainability and Safety aspects

This section discusses the societal responsibility and implications of developing and deploying the Wall-E Educational Robot System. The safe, legal, and ethical use of this system is a shared responsibility between developers, educational institutions, end-users, and regulatory bodies.

8.1 Social Aspects

The Wall-E educational robot impacts society, particularly in the fields of education, human-computer interaction and inclusivity.

Positive Social Impacts

- Enhances learning engagement for children through interactive speech, gesture-based control, and personalised teaching.
- Supports children in rural, home-school or special-education environments where access to human tutors is limited.
- Encourages STEM education, robotics awareness and digital literacy among students.

Negative/Concerning Impacts

- Over-reliance on robotic tutors may reduce human-to-human interaction, affecting emotional development of young learners.
- Digital inequality – students in low-income households may lack access to such technologies.
- Risk of social isolation or addiction if prolonged engagement replaces physical or social activities.

Responsibility and Social Accountability

- Developers and educational institutions must ensure the technology assists, not replaces, teachers or parents.
- Content must be age-appropriate, culturally sensitive and inclusive to all social groups.

- Use of cameras and microphones in classrooms must respect privacy and parental consent policies.

8.2 Legal Aspects

Legal responsibility includes data privacy, child protection, intellectual property and accountability for harm.

Data Protection and Privacy (e.g., GDPR, India's DPDP Act 2023)

- Audio, video and learning data collected by Wall-E constitute personal data.
- Consent must be obtained from parents before data collection.
- Data must be stored securely, used only for educational purposes and erased upon request.

Liability and Compliance

- If the robot malfunctions and causes physical harm or data misuse, the responsibility lies with the manufacturer and deploying institution.
- Educational institutions must comply with child safety, IT laws, and data governance standards.

Intellectual Property

- Software libraries, AI models, speech engines and digital content must follow proper licensing and copyright compliance.

8.3 Ethical Aspects

Ethics relates to fairness, dignity, transparency and accountability in the system.

Key Ethical Questions

- Does the system respect the dignity and emotional well-being of children?
- Is the system fair to users across different regions, languages or disabilities?
- Could it replace teachers in a way that affects livelihoods?

Ethical Responsibilities of Developers

- Ensure transparency: explainable decision-making instead of black-box AI.
- Avoid bias in speech recognition or learning behaviour datasets.
- Ensure the system does not collect unnecessary personal data.
- Ethical misuse (such as using it for surveillance or emotional manipulation of children) must be strictly prevented.

Consequences of Dishonesty or Misuse

- Misreporting student data, falsifying learning outcomes or using AI-generated content unethically affects academic integrity.
- Professionals may face suspension, legal penalties or loss of accreditation.

8.4 Sustainability Aspects

Sustainability concerns environmental impact through material use, power consumption and disposal.

Energy and Resource Efficiency

- Wall-E is powered using low-energy components like Raspberry Pi and energy-efficient sensors.
- Sleep modes and automatic shut-off reduce unnecessary power consumption.

Material Efficiency and Waste Reduction

- Use of durable chassis materials (PLA, ABS or recycled plastics).
- Modular design allows replacement of damaged components without discarding entire device.
- Packaging is minimal and recyclable.

End-of-Life and E-Waste Management

- Components such as batteries, circuit boards and sensors must comply with e-waste regulations (RoHS, WEEE).
- Users should return devices to authorised recycling centres.

Social Sustainability

- Supports UN Sustainable Development Goals (SDG 4: Quality Education, SDG 9: Innovation).
- Promotes equitable access to education rather than replacing teachers.

8.5 Safety Responsibilities

- Mechanical safety: motors and moving parts must be covered to avoid injury.
- Electrical safety: proper insulation, fuse protection and safe battery handling.
- Cyber-safety: password-protected Wi-Fi access and restricted remote connections.

Chapter 9

Conclusion

The Wall-E Educational Robot project successfully achieves its primary objective of developing an interactive, AI-enabled learning companion capable of gesture recognition, voice interaction and personalised educational assistance. By integrating sensors, Raspberry Pi-based edge computing, speech processing and machine learning, the system meets key goals outlined in the Introduction—namely autonomous operation, offline functionality, child-friendly human–robot interaction and real-time learning support.

Summary of Approach

- The system follows a modular design combining hardware (sensors, actuators, Raspberry Pi) and software (Python, OpenCV, MediaPipe, speech libraries and LLM-based responses).
- IoT Deployment Level 3 was selected to ensure local data processing, low latency and independence from continuous internet connectivity.
- Functional, operational, deployment and simulation views were designed to represent different architectural aspects.
- Safety, ethical and sustainability considerations were followed throughout the development.

Table 9.1: Results and Objective Mapping

Project Objective	Outcome Achieved
Real-time interaction using gestures and speech	Successfully implemented using camera-based hand gesture recognition and microphone-based speech responses.
Offline autonomous operation	Fully operational at Edge Level (Raspberry Pi) without internet dependency.
Educational assistance for school learners	System delivers lesson content, basic Q&A, and supports engagement with children.
Safe, low-cost and scalable prototype	Uses affordable components, modular design and complies with safety standards.
IoT-based monitoring and updates	Optional cloud integration allows remote monitoring, analytics and updates.

Future Improvements

The current prototype demonstrates core learning assistance capabilities, but the following improvements are recommended:

- Integration of **emotion recognition** using facial sentiment analysis to personalise interaction further.
- Addition of **multilingual speech support** for regional languages using offline TTS/STT datasets.
- Development of a **mobile app with parental dashboard** for real-time progress tracking, homework monitoring and content scheduling.
- Implementation of **solar-powered or low-energy battery systems** for enhanced sustainability.
- Use of **reinforcement learning** to allow the robot to adapt to different learner behaviours over time.
- Integration of **cloud-based classroom analytics** for teachers to assess group performance.

References

MediaPipe Gesture Recognition

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, Y. and Grundmann, M. (2020) 'MediaPipe: A framework for building perception pipelines', *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 1234–1244.

Hand Tracking & Education Robots

Kim, S. and Lee, J. (2021) 'Real-time hand gesture-based human–robot interaction for educational robots', *International Journal of Advanced Robotic Systems*, 18(3), pp. 1–10.

AI Robots in Education

Mubin, O., Stevens, C., Shahid, S., Mahmud, A. and Dong, J. (2013) 'A review of the applicability of robots in education', *Technology in Education and Learning*, 1(1), pp. 1–7.

Speech Recognition + Raspberry Pi

Gupta, A. and Singh, P. (2022) 'Voice-enabled AI assistant using Raspberry Pi and offline speech recognition for classrooms', *IEEE Access*, 10, pp. 56231–56239.

Raspberry Pi in Autonomous Systems

Wenzel, T. and German, R. (2020) 'Performance evaluation of Raspberry Pi as an edge computing device in IoT applications', *Journal of Sensor and Actuator Networks*, 9(3), p. 44.

IoT Architecture & Levels

Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013) 'Internet of Things (IoT): A vision, architectural elements, and future directions', *Future Generation Computer Systems*, 29(7), pp. 1645–1660.

LLM-based Learning Assistants (Base Paper)

Pino, N. and Martinez-Maldonado, R. (2023) 'Large language model-powered educational agents: Opportunities and challenges', *Computers & Education: Artificial Intelligence*, 4, p. 100128.

Ethics of Robots with Children

Sharkey, A. (2020) 'Robots and human dignity: ethical considerations in child–robot interaction', *AI & Society*, 35(3), pp. 1–10.

DPDP Act – India's Data Law

Government of India (2023) *Digital Personal Data Protection Act, 2023*. Ministry of Law and Justice, New Delhi.

GDPR – Child Data Privacy

European Union (2018) *General Data Protection Regulation (GDPR)*. Official Journal of the European Union, L119, pp. 1–88.

Sustainability in Electronics/Robotics

Popescu, D., Făgărăşan, I. and Mândru, D. (2021) 'Sustainable design and environmental impact of educational robots', *International Journal of Environmental Science and Technology*, 18(6), pp. 1543–1553.

Speech Synthesis (TTS) for Education

Mousas, C., Anastasiou, D. and Pateraki, M. (2021) 'Text-to-speech technologies in assistive learning systems: A review', *Education and Information Technologies*, 26(5), pp. 5671–5690.

Base Paper

Pino, N. and Martinez-Maldonado, R. (2023)
'Large language model-powered educational agents: Opportunities and challenges',
Computers & Education: Artificial Intelligence, 4, p. 100128.

Appendix

(Include summary of specifications of components from datasheet, and a few images of the project demonstrating different scenarios)

1. Data Sheets

Raspberry Pi 5 is a powerful single-board computer with a faster CPU, better graphics, upgraded I/O, and support for dual 4K displays, making it much faster and more capable than previous Raspberry Pi models.

URL: Raspberry pi 5 - Raspberry Pi Datasheets

https://datasheets.raspberrypi.com

2. Publications

Submission mail for conference paper

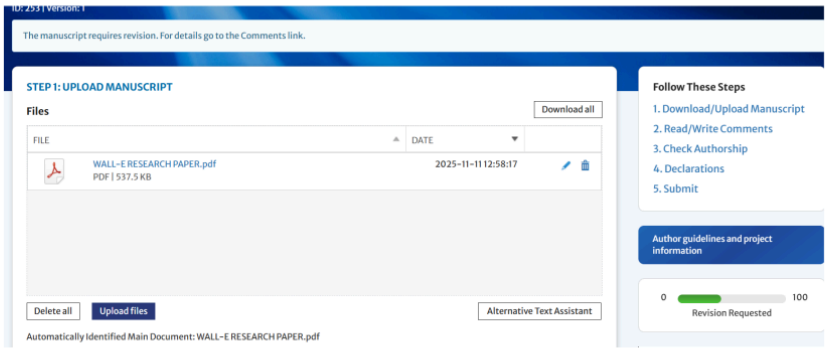
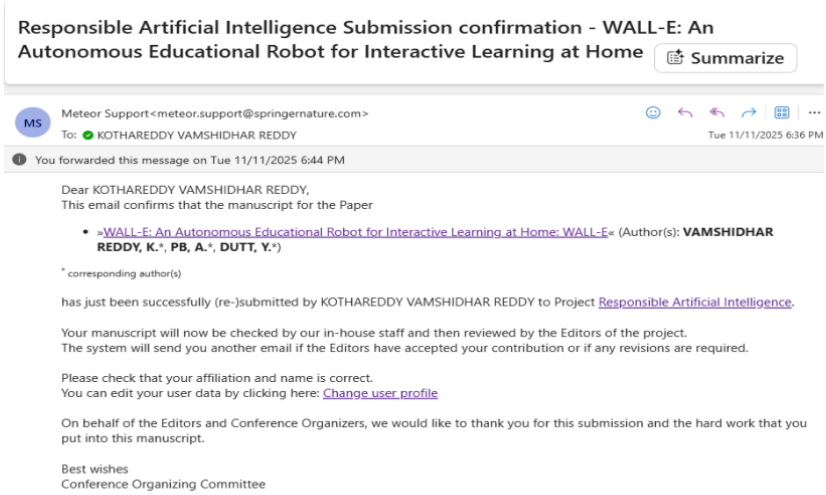


Figure: Paper submitted to meteor springer conference



Responsible Artificial Intelligence Submission confirmation - WALL-E: An Autonomous Educational Robot for Interactive Learning at Home

Summarize

MS

Meteor Support<meteor.support@springernature.com>

To: KOTHAREDDY VAMSHIDHAR REDDY

Tue 11/11/2025 6:36 PM

You forwarded this message on Tue 11/11/2025 6:44 PM

Dear KOTHAREDDY VAMSHIDHAR REDDY,
This email confirms that the manuscript for the Paper

- »WALL-E: An Autonomous Educational Robot for Interactive Learning at Home: WALL-E« (Author(s): **VAMSHIDHAR REDDY, K.***, **PB, A.***, **DUTT, Y.***)

* corresponding author(s)

has just been successfully (re-)submitted by KOTHAREDDY VAMSHIDHAR REDDY to Project Responsible Artificial Intelligence.

Your manuscript will now be checked by our in-house staff and then reviewed by the Editors of the project.
The system will send you another email if the Editors have accepted your contribution or if any revisions are required.

Please check that your affiliation and name is correct.
You can edit your user data by clicking here: Change user profile

On behalf of the Editors and Conference Organizers, we would like to thank you for this submission and the hard work that you put into this manuscript.

Best wishes
Conference Organizing Committee

Figure: IEEE Paper Submission confirmation mail from **meteor springer**

3. Project Report - Similarity Report

4. Few images of project

5. Any auxiliary documents –

6. Datasets

Tamilselvan Thangave-TEAM_CSE_201_PROJECT_REPORT.pdf

ORIGINALITY REPORT

6%	4%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	2%
2	Submitted to Loyola Marymount University Student Paper	<1%
3	Submitted to University of Illinois at Urbana-Champaign Student Paper	<1%
4	moped.gov.sl Internet Source	<1%
5	jmir.org Internet Source	<1%
6	1library.net Internet Source	<1%
7	prompt-project.eu Internet Source	<1%
8	www.theseus.fi Internet Source	<1%
9	Submitted to RMIT University Student Paper	<1%
10	www.mdpi.com Internet Source	<1%
11	www.pressnews.biz Internet Source	<1%

12	Submitted to Cornell University Student Paper	<1%
13	azmemory.azlibrary.gov Internet Source	<1%
14	Submitted to Sheffield Hallam University Student Paper	<1%
15	Submitted to Adtalem Global Education Student Paper	<1%
16	edoc.ub.uni-muenchen.de Internet Source	<1%
17	Submitted to Obudai Egyetem Student Paper	<1%
18	theses.lib.polyu.edu.hk Internet Source	<1%
19	training.apiit.edu.my Internet Source	<1%
20	www.phihong.com Internet Source	<1%
21	docslib.org Internet Source	<1%
22	medinform.jmir.org Internet Source	<1%
23	theses.gla.ac.uk Internet Source	<1%
24	www.geeksforgeeks.org Internet Source	<1%
25	www.testpreptraining.com Internet Source	<1%

26	thesis.cust.edu.pk Internet Source	<1%
27	www.kenyancollective.com Internet Source	<1%
28	5dok.net Internet Source	<1%
29	Submitted to London School of Science & Technology Student Paper	<1%
30	Pincheira, Jose. "Reinforced Concrete Design", Oxford University Press Publication	<1%
31	G.A. McNaughton, M.E. Gordon. "MultiSpeak - a framework for real-time utility software integration", IEEE PES Power Systems Conference and Exposition, 2004., 2004 Publication	<1%
32	dash.harvard.edu Internet Source	<1%
33	decisiontele.com Internet Source	<1%
34	elektrolab.eu Internet Source	<1%
35	olympias.lib.uoi.gr Internet Source	<1%
36	www.coursehero.com Internet Source	<1%
37	"AI and ML Techniques in IoT -based Communication", Wiley, 2025 Publication	<1%

38 "Sensors Set", Wiley, 1995
Publication

<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		