



Faculty of Computer Science and Information Technology

CRADLESENSE: IOT-BASED SMART CRADLE SYSTEM FOR INFANTS

Sheana Kasih Benedict

Bachelor of Computer Science with Honours (Network Computing)

2025

CRADLESENSE: IOT-BASED SMART CRADLE FOR INFANTS

SHEANA KASIH BENEDICT

This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science with Honours (Network Computing)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2025

CRADLESENSE: BUAIAN PINTAR BERASASKAN IOT UNTUK BAYI

SHEANA KASIH BENEDICT

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dan Teknologi Maklumat
(Pengkomputeran Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE CRADLESENSE: AN IOT-BASED SMART CRADLE SYSTEM

ACADEMIC SESSION: 2024/2025

(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (✓)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
☒	UNRESTRICTED	

Sheanakk
(AUTHOR'S SIGNATURE)

Permanent Address

Lot 3959, Lorong 21, Jalan Kuching
Timur 3, Taman Tunku, 98000 Miri,
Sarawak

Date: 22 July 2025

Validated by 
Assoc. Prof. Dr. Jihari Abdullah
Faculty of Computer Science & Information Technology
Universiti Malaysia Sarawak
94300 Kota Samarahan
(SUPERVISOR'S SIGNATURE)

Date: 22 July 2025

Note * Thesis refers to PhD, Master, and Bachelor Degree

** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I, Sheana Kasih Benedict with the matric number of 78525, hereby solemnly declare that the report titled 'CradleSense: IoT-Based Smart Cradle System for Infants' is prepared and completed authentically by me in partial fulfilment of the requirements for the degree of Bachelor of Computer Science with Honours (Network Computing) is a record of an original work carried out by me under the guidance of Associate Professor Dr. Johari bin Abdullah. I further declare that the work reported in this project has not been previously or concurrently submitted for any other degree at UNIVERSITI MALAYSIA SARAWAK (UNIMAS) or any other institutions. I declare that I have not copied or plagiarize from any other sources except for quotations and citations which have been duly acknowledged.



SHEANA KASIH BENEDICT (78525)
Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to my Final Year Project supervisor, Associate Professor Dr. Johari bin Abdullah, who has supported and provided continuous guidance throughout the process of completing this project. His support and feedback have contributed greatly to the completion of this project. I would also like to thank Associate Professor Dr. Mohamad Nazim bin Jambli, who is the examiner of this project for giving useful feedback in improving the quality of this project. Additionally, I would like to extend my thanks to Professor Dr. Wang Yin Chai, for guiding us tirelessly with important information in regard to the Final Year Project contents, and to always have useful guidelines for student reference. Moreover, I am deeply grateful to all the respondents who were willing to contribute their time and insights by answering the questionnaire as part of my research. Not to forget, I wanted to express my gratitude to my fellow friends and coursemates who tirelessly accompanied, provided feedback and shared useful guidance in the completion of this project. Last but not least, I extend my deepest gratitude and appreciation to my family, for their continuous support and affirmation during my lowest moments throughout the completion of this project.

ABSTRACT

The challenges of infant care, particularly for single mothers, can lead to significant mental and physical strain, making it critical to develop solutions that support effective caregiving. The CradleSense project addresses this issue by proposing an IoT-based smart cradle system designed to automate aspects of infant care and monitoring. Infants are highly sensitive to environmental changes, and their needs often require immediate attention, which can be taxing for single mothers who balance multiple responsibilities. The proposed system integrates a Raspberry Pi controller with an array of sensors, including sound, motion, temperature and humidity. These sensors work together with actuators such as a servo-driven rocking mechanism and speakers, enabling the cradle to detect and respond to a variety of conditions, including crying and motion. The system's functionality is managed through a custom-built mobile application, which provides real-time alerts and remote control capabilities. This personalised feedback allows mothers to make informed decisions, ensuring better care. Developed using the Rapid Application Development (RAD) methodology, the system emphasizes iterative design and user-centric feedback to deliver a robust and efficient solution. Python scripting on a Raspbian OS drives the system's operation, while its modular design ensures adaptability to various caregiving scenarios. The CradleSense system aspires to alleviate the challenges of infant care by combining advanced IoT capabilities with practical applications for everyday use. Its multifunctional design promotes timely intervention and efficient caregiving, contributing significantly to the mental and physical well-being of both infants and their caregivers. While its innovative approach holds significant promise, current limitations include hardware constraints and the potential need for further refinement in environmental monitoring accuracy. Nonetheless, CradleSense represents a vital step towards integrating technology into infant care, addressing the pressing need for smarter, more accessible solutions in modern parenting.

ABSTRAK

Cabaran dalam penjagaan bayi, terutamanya bagi ibu tunggal, boleh menyebabkan tekanan mental dan fizikal yang signifikan, menjadikan penyelesaian yang menyokong penjagaan yang berkesan amat penting. Projek CradleSense menangani isu ini dengan mencadangkan sistem buaian pintar berasaskan IoT yang direka untuk mengautomasi aspek penjagaan dan pemantauan bayi. Bayi sangat sensitif terhadap perubahan persekitaran, dan keperluan mereka sering memerlukan perhatian segera, yang boleh membebankan ibu tunggal yang sering menguruskan pelbagai tanggungjawab. Sistem yang dicadangkan ini mengintegrasikan pengawal Raspberry Pi dengan pelbagai sensor, termasuk sensor bunyi, pergerakan, suhu, dan kelembapan. Sensor-sensor ini berfungsi bersama dengan penggerak seperti mekanisme buaian yang digerakkan oleh servo dan pembesar suara, membolehkan buaian mengesan dan memberi tindak balas terhadap pelbagai keadaan, termasuk tangisan dan pergerakan bayi. Fungsi sistem ini dikendalikan melalui aplikasi mudah alih yang direka khas, yang menyediakan amaran dalam masa nyata dan kawalan mudah alai. Maklum balas yang diperibadikan ini membolehkan ibu membuat keputusan yang lebih bermaklumat, memastikan penjagaan yang lebih baik. Dibangunkan menggunakan metodologi Rapid Application Development (RAD), sistem ini menekankan reka bentuk berulang dan maklum balas yang berpusatkan pengguna untuk menyampaikan penyelesaian yang kukuh dan cekap. Pengaturcaraan Python pada sistem operasi Raspbian menggerakkan operasi sistem ini, sementara reka bentuk modularnya memastikan kebolehsuaian kepada pelbagai senario penjagaan bayi. Sistem CradleSense bertujuan untuk meringankan cabaran penjagaan bayi dengan menggabungkan keupayaan IoT yang canggih dengan aplikasi praktikal untuk kegunaan harian. Reka bentuk multifungsinya mempromosikan intervensi tepat pada masanya dan penjagaan yang cekap, menyumbang secara signifikan kepada kesejahteraan mental dan fizikal bayi serta penjaganya. Walaupun pendekatan inovatif ini mempunyai potensi besar, batasan semasa termasuk kekangan perkakasan dan keperluan untuk penambahbaikan ketepatan pemantauan persekitaran. Namun begitu, CradleSense merupakan langkah penting ke arah mengintegrasikan teknologi dalam penjagaan bayi, memenuhi keperluan mendesak untuk penyelesaian yang lebih pintar dan mudah diakses dalam keibubapaan moden.

TABLE OF CONTENTS

DECLARATION.....	iv
ACKNOWLEDGEMNT	v
ABSTRACT.....	vi
ABSTRAK	vii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xvi
CHAPTER 1: INTRODUCTION.....	1
1.1 Project Title.....	1
1.2 Introduction/Background	1
1.3 Problem Statement	2
1.4 Project Objectives	5
1.5 Brief Methodology.....	6
1.6 Project Scopes	8
1.7 Significance of Project.....	11
1.8 Project Schedule.....	13
1.9 Expected Outcome	14
1.10 Project Description.....	15
1.11 Basic Skill Requirements.....	15
1.12 Project Outline	15
CHAPTER 2: LITERATURE REVIEW	18
2.1 Introduction.....	18
2.2 Review on Similar Existing Systems.....	18
2.2.1 SNOO Smart Sleeper Bassinet.....	19
2.2.2 Cradlewise Convertible Smart Bassinet and Crib.....	24
2.2.3 IoT-Based Baby Monitoring System for Smart Cradle (IoT-BBMS)	28
2.3 Features Comparison of the Reviewed Systems	30
2.4 Review on Tools and Technology	33
2.4.1 Software	33

2.4.1.1 Android Studio.....	33
2.4.1.2 Raspberry Pi OS (Previously known as Raspbian).....	38
2.4.2 Hardware.....	41
2.4.2.1 Raspberry Pi Microprocessor.....	41
2.4.2.2 Raspberry Pi Camera Module.....	44
2.4.2.3 LM35 Temperature Sensor Module.....	46
2.4.2.4 PIR Motion Sensor.....	47
2.4.2.5 Servo Motor	49
2.4.2.6 Speaker.....	51
2.4.2.7 MSI Modern 15 Laptop	52
2.4.2.8 Arduino Uno Microcontroller	53
2.5 Summary	55
CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN.....	57
3.1 Introduction.....	57
3.2 Rapid Application Development (RAD)	57
3.3 Requirements Planning	61
3.3.1 Analysis on Similar Existing Systems	62
3.3.2 Analysis on Proposed System.....	63
3.3.3 Questionnaire	65
3.3.4 Software Requirements.....	78
3.3.5 Hardware Requirements.....	79
3.4 User Design.....	83
3.4.1 Overall System Architecture Design	83
3.4.2 Flowchart Diagram	86
3.4.3 Context Diagram.....	89
3.4.4 Data Flow Diagram Level 1	91
3.4.5 Data Flow Diagram Level 2.....	94
3.4.6 Entity Relationship Diagram.....	99
3.4.7 Wireframes.....	100
3.5 Construction.....	107
3.5.1 System Development	108

3.5.2 System Refinement	109
3.6 Cutover.....	109
3.6.1 System Testing.....	110
3.6.2 System Deployment.....	110
3.7 Summary	111
CHAPTER 4: IMPLEMENTATION	112
4.1 Introduction.....	112
4.2 Feature Implementation	112
4.2.1 Installation and Setup of Android Studio.....	112
4.2.2 Installation and Setup of Raspberry Pi Imager	113
4.2.3 Installation and Setup of Arduino IDE	118
4.2.4 Installation and Setup of Flutter SDK in Android Studio.....	121
4.3 System Hierarchy.....	125
4.4 Workflow of the Proposed Application	127
4.4.1 Real-time Cry Detection Process	128
4.4.2 Motion Detection Process	129
4.4.3 Moisture Detection Process	129
4.4.4 Temperature Monitoring Process.....	129
4.4.5 Live Video Streaming Process.....	130
4.5 Coding of Proposed Algorithm and Function.....	130
4.5.1 Cry Detection Algorithm	131
4.5.2 Motion Detection Algorithm.....	132
4.5.3 Temperature Reading Algorithm	133
4.5.4 Wetness Detection Algorithm.....	135
4.5.5 Cradle Rocking Algorithm.....	136
4.6 Summary	140
CHAPTER 5: TESTING.....	141
5.1 Introduction.....	141
5.2 Functional Testing	141
5.2.1 Unit Testing	142
5.2.2 Integration Testing.....	151

5.3 Non-functional Testing	154
5.3.1 Reliability Testing.....	155
5.3.2 Usability Testing.....	156
5.4 Discussion on Obtained Results.....	162
5.5 Summary	165
CHAPTER 6: CONCLUSION AND FUTURE WORKS	166
6.1 Introduction.....	166
6.2 Objectives and Achievements.....	166
6.3 Project Limitations and Challenges	167
6.4 Future Works	168
6.5 Conclusion	169
REFERENCES.....	170
APPENDICES	173
Appendix A.....	173
Appendix B.....	175

LIST OF TABLES

Table 2.1: Comparative Table Analysis of the Three Systems.....	32
Table 2.2: Features and Justifications for Raspberry Pi OS	39
Table 2.3: Features and Justification for Choosing Raspberry Pi.....	42
Table 2.4: Features and Justification for Choosing MSI Modern 15 Laptop	52
Table 2.5 Features and Justification for Choosing Arduino Uno Microcontroller	54
Table 3.1: Key Features and Functionalities of CradleSense	64
Table 3.2: Software Requirements for CradleSense	78
Table 3.3: Hardware Requirements for CradleSense.....	79
Table 5.1 Unit Test Case for Sign Up.....	143
Table 5.2 Unit Test Case for Login	144
Table 5.3 Unit Test Case for User Profile.....	145
Table 5.4 Unit Test Case for Live Video Feed	148
Table 5.5 Unit Test Case for Soothing Sound Selection	148
Table 5.6 Unit Test Case for Rock Cradle.....	150
Table 5.7 Unit Test Case for Baby Alerts.....	151
Table 5.8 Integration Test Case for CradleSense App.....	152
Table 5.9 Test Cases for Reliability Testing.....	156

LIST OF FIGURES

Figure 1.1: Phases of Rapid Application Development (RAD).....	6
Figure 1.2: Phases of RAD methodology for CradleSense.....	7
Figure 1.3: Gantt Chart Task Table on Project Timeline for Phase 1 to Phase 9	13
Figure 2.1: SNOO Smart Sleeper Bassinet features (2023).....	20
Figure 2.2: Screenshot from SNOO mobile app (2023)	22
Figure 2.3: Built-in swaddle in the cradle (2023)	23
Figure 2.4: SNOO Smart Sleeper Bassinet (2023)	24
Figure 2.5: Cradlewise Convertible Smart Bassinet and Crib dimensions	25
Figure 2.6: Features of Cradlewise	26
Figure 2.7: Screenshot of the mobile app “Sleep Insights” page.....	27
Figure 2.8: Flowchart of the system (Jabbar et al., 2019).....	29
Figure 2.9: Mobile application interface (Jabbar et al., 2019).....	30
Figure 2.10: Android Studio Interface	34
Figure 2.11: Layout Editor of Android Studio.....	36
Figure 2.12: Emulator in Android Studio	36
Figure 2.13: Assistant Tool Window in Android Studio	37
Figure 2.14: Raspberry Pi OS Interface.....	39
Figure 2.15: Evolution of Raspberry Pi	41
Figure 2.16: Different Raspberry Pi camera modules	45
Figure 2.17: LM35 Temperature Sensor Module	46
Figure 2.18: Components of a PIR sensor	48
Figure 2.19: Servo motor	49
Figure 2.20: Mini 8 ohm 1W speaker	51
Figure 2.21 Arduino Uno Microcontroller.....	54
Figure 3.1: The 4 Phases of Rapid Application Development (RAD)	59
Figure 3.2: Age Range of Respondents	66
Figure 3.3: Respondents’ Current Work Arrangements	66
Figure 3.4: Age Range of Respondents’ Youngest Child.....	67
Figure 3.5: Types of Baby Monitoring Tools Used by Respondents	68
Figure 3.6: Frequency Respondents Use Cradle to Soothe Their Baby	69
Figure 3.7: Type of Cradle Used by Respondents	69
Figure 3.8: Challenges Faced by the Respondents with Their Current Cradle.....	70
Figure 3.9: Cradle Features that could Improve Respondents’ Cradle Experience	71
Figure 3.10: Basic Cradle and Spring Cot Users Consideration to use Smart Cradle	71
Figure 3.11: Semi-automated Cradle Limitations.....	72
Figure 3.12: Features that Would Improve Semi-automated Cradle User Experience.....	73
Figure 3.13: Alternative Methods Used to Soothe Baby without Cradle	74
Figure 3.14: Features Considered by Non-Cradle Users to Use a Cradle	74

Figure 3.15: Respondents' Concerns on Using Smart Devices for Infant Care	75
Figure 3.16: Maximum Amount Respondents are Willing to Pay for a Smart Cradle	76
Figure 3.17: Factors Affecting Respondents' Purchase of a Smart Cradle	77
Figure 3.18: Overall System Architecture Design	84
Figure 3.19: Flowchart of CradleSense	86
Figure 3.20: Context Diagram (Data Flow Diagram Level 0) for CradleSense	90
Figure 3.21: Data Flow Diagram Level 1 for CradleSense	92
Figure 3.22: Data Flow Diagram Level 2 for Process 1.0 of CradleSense	95
Figure 3.23: Data Flow Diagram Level 2 for Process 2.0 of CradleSense	95
Figure 3.24: Data Flow Diagram Level 2 for Process 3.0 of CradleSense	96
Figure 3.25: Data Flow Diagram Level 2 for Process 4.0 of CradleSense	97
Figure 3.26: Data Flow Diagram Level 2 for Process 5.0 of CradleSense	98
Figure 3.27: Data Flow Diagram Level 2 for Process 6.0 of CradleSense	99
Figure 3.28: Entity Relationship Diagram for CradleSense	100
Figure 3.29: Wireframe for CradleSense App (1)	102
Figure 3.30: Wireframe for CradleSense App (2)	104
Figure 3.31: Wireframe for CradleSense App (3)	105
Figure 3.32: Wireframe for CradleSense App (4)	107
Figure 4.1 Android Studio Installation Files.....	112
Figure 4.2 Raspberry Pi Imager Download Link.....	114
Figure 4.3 Raspberry Pi Imager Setup Wizard	114
Figure 4.4 Raspberry Pi Imager Wizard Setup Complete.....	115
Figure 4.5 Raspberry Pi Imager Application	116
Figure 4.6 Selecting Raspberry Pi Device Model.....	117
Figure 4.7 Selecting Raspberry Pi Operating System.....	118
Figure 4.8 Arduino IDE Installation Link.....	119
Figure 4.9 Arduino IDE License Agreement.....	119
Figure 4.10 Arduino IDE Wizard Setup Complete.....	120
Figure 4.11 Arduino IDE Boards Manager.....	121
Figure 4.12 Arduino IDE Board Selection	121
Figure 4.13 Flutter SDK Platform Selection.....	122
Figure 4.14 Flutter SDK Application Selection.....	123
Figure 4.15 Laptop Environment Variables Setting	124
Figure 4.16 Environment Variables Path Settings.....	124
Figure 4.17 Adding Flutter Path to Environment Variables	125
Figure 4.18 System Hierarchy of the Proposed System	126
Figure 4.19 Hardware Connection Between Arduino and Raspberry Pi	127
Figure 4.20 Code Snippet from Python in Raspberry Pi	128
Figure 4.21 Arduino IDE Pin Initialisation and Declaration	131
Figure 4.22 Serial Monitor Output in Arduino IDE	131

Figure 4.23 Raspberry Pi main.py Code Snippet to Communicate with Firebase	131
Figure 4.24 Information Stored In Firestore	132
Figure 4.25 Arduino Snippet for Cry Detection Algorithm.....	132
Figure 4.26 Code Snippet from main.py for Motion Sensing and Firebase Data Stored	133
Figure 4.27 Arduino Code Snippet for Motion Detection Algorithm	133
Figure 4.28 Code Snippet from main.py for Temperature Sensing and Data Stored in Firebase	134
Figure 4.29 Arduino Code Snippet for Temperature Monitoring Algorithm	135
Figure 4.30 Code Snippet from main.py for Wetness Sensing and Data Stored in Firebase	135
Figure 4.31 Arduino Code Snippet for Wetness Detection Algorithm.....	136
Figure 4.32 Flutter Code Snippet rock_cradle_controller.dart	136
Figure 4.33 Code Snippet from main.py for Rocking Cradle Manually Function	137
Figure 4.34 Arduino Code Snippet for Rocking Cradle Manually	139
Figure 4.35 Arduino Code Snippet for Accepting Code from Raspberry Pi	140
Figure 5.1 UAT Survey User Demographics.....	158
Figure 5.2 UAT Survey Features and Effectiveness on Cry Detection Feature	158
Figure 5.3 UAT Survey on Features and Effectiveness of CradleSense	159
Figure 5.4 UAT Survey on Features and Effectiveness of CradleSense for Soothing Sounds Feature.....	159
Figure 5.5 UAT Survey on Features Usefulness and Consistency	160
Figure 5.6 UAT Survey on Overall Functional Experience	161
Figure 5.7 UAT Survey on Application Appropriateness for Parents or Caregiver	162

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
DC	Direct Current
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
IDE	Integrated Development Environment
IoT	Internet of Things
IoT-BBMS	Internet of Things-Based Baby Monitoring System
MQTT	Message Queuing Telemetry Transport
MSI	Micro-Star International
OS	Operating System
PIR	Passive Infrared
PWM	Pulse Width Modulation
QTT	Quality Time Tradeoff
RAD	Rapid Application Development
SDLC	System Development Life Cycle
SIDS	Sudden Infant Death Syndrome
WHO	World Health Organisation
XML	Extensible Markup Language

CHAPTER 1: INTRODUCTION

1.1 Project Title

CradleSense: An IoT-Based Smart Cradle for Infants.

1.2 Introduction/Background

Early childhood is one of the most critical phases of a child's development. Aligned with the international policy standards, World Health Organization (WHO) defines early childhood as the development phase from prenatal to eight years of age. During this period of rapid growth and heightened vulnerability, children undergo significant physical, mental, and emotional changes. Consistent and attentive caregiving is essential for promoting healthy growth and supporting physical and emotional well-being. According to Meriem et al. (2020), the quality of early stimuli, which includes environmental influences, profoundly impacts future child development into adulthood. However, fulfilling these caregiving demands as a working single mother has proven to be challenging.

Single mothers are defined as women who are raising one or more children without the presence or support of a partner or spouse. While men and women initially enter the workforce on equal terms, their paths begin to differ significantly once they become parents. Men's career advancement is seldom impacted by having children, whereas women's progress is often more hindered. The increase in female participation in the workforce affects infant care in many families. Research underscores the strain of stress and fatigue these multitasking roles place on single mothers, many of whom struggle to maintain consistent caregiving amidst work and household duties (Yee Theng et al., 2022). As a result, the impact it has on both their mental and physical

well-being may affect the quality of care provided to their infants, particularly during sleep and rest periods (Bhat et al., 2024).

In recent years, the concept of smart cradles has emerged as a significant innovation in childcare, leveraging the Internet of Things (IoT) to enhance parental monitoring and infant care. According to the research conducted by Saude and Vardhini (2020), traditional cradles may not adequately support single mothers, particularly those who are employed full-time, due to the demands of constant cradle monitoring. The need for innovative solutions that provide real-time monitoring, and support has never been more critical. Existing technologies in this domain often lack comprehensive features that cater specifically to the needs of single mothers, leaving a gap that this project aims to fill.

The proposed IoT-based Smart Cradle utilises Raspberry Pi technology to offer functionalities such as remote monitoring, environmental sensing, and automated soothing mechanisms. The system will notify the caregivers on the infant's condition via a mobile application, followed by offering action suggestions according to the data collected by the sensors. By addressing these needs, this project not only aims to improve infant care, but also seeks to alleviate some of the mental burdens faced by single mothers, ultimately contributing to better work-life balance.

1.3 Problem Statement

The increasing number of working parents, especially single mothers, highlights the challenge of balancing professional responsibilities with the intensive demands of infant care. Without adequate support, many single mothers struggle to provide consistent, real-time monitoring for their infants,

resulting to delays in responding to their needs and a growing sense of stress and inadequacy. Studies indicate that single mothers experience heightened stress when balancing work and caregiving alone, with delayed responses to infant's needs potentially impacting infant development and parental well-being.

Many single mothers face the dual demands of caregiving and work, which can result in mental fatigue and emotional strain. The limited options for affordable and accessible childcare means that these mothers often rely on informal care arrangements. For instance, sending their infants to the grandparents' house or hiring a nanny. Not all households have the luxury of having the grandparents live nearby or being able to afford the latter. Opting for these arrangements results in the mothers' lack of consistent supervision or the capacity to provide specialised attention. Findings shows that without adequate support, the vigilance required to monitor an infant, especially during critical periods such as illness, teething and sleep, can significantly increase stress and anxiety. This can impact both maternal health and the quality of infant care provided.

Current conventional infant monitoring solutions, such as audio or video baby monitors, provide limited assistance, requiring constant parental supervision. Some parents even prefer to have a nursery room, where the infant sleeps in a separate room from them. These baby monitors are often paired with the regular and traditional baby cradles. Despite being able to monitor the baby's condition, parents still need to physically attend to the infant in the cradle. This can be challenging during resting times or when parents are occupied with fulfilling occupational demands. There are existing smart cradles on the market, however, the high cost of these automated systems is often inaccessible to single mothers due to financial constraints. These limitations underscore the need for an affordable and accessible automated system tailored to alleviate the

caregiving load for single mothers by providing real-time monitoring and intelligent responses to infant needs.

The proposed smart cradle, CradleSense, is an IoT-based infant monitoring system designed to support single mothers by incorporating real-time, data-driven responses to infant needs. This system utilises a Raspberry Pi and Arduino controller with multiple sensors, such as temperature, motion, audio, and camera, along with actuators like rocking mechanism and speaker. CradleSense will operate through a mobile app, which enables remote monitoring and control, real-time alerts, and data logging for sleep and activity patterns. By analysing motion and audio, CradleSense will then automatically soothe the baby when certain parameters are triggered and provides insights into the baby's well-being, offering a reliable and accessible tool for single mothers.

This project aims to reduce the mental and physical burden on single mothers, allowing them to monitor and respond to their infant's needs more affectively while balancing other responsibilities. By providing an intelligent solution, CradleSense supports single mothers in reducing stress, enhancing their caregiving capacity, and promoting infant well-being with minimal manual intervention.

In summary, CradleSense offers an innovative, automated solution for infant care that supports single mothers by providing personalised, data-driven monitoring and caregiving insights. This system promotes healthier caregiving habits, reduces stress, and enhances overall well-being of both mother and child, ultimately reducing the long-term challenges associated with single-parent infant care.

1.4 Project Objectives

The main objective of the project is to design and develop an IoT-based smart cradle which automates aspects of infant care to reduce the physical and emotional strain of infant care on single mothers. Other objectives include:

1. To implement an intelligent and responsive Raspberry Pi and Arduino system that analyses real-time data from integrated sensors, enabling the cradle to autonomously adjust and respond to the infant's need based on the detected parameters. The cradle's functions will be monitored and controlled via a mobile application which allows for caregiving with real-time notification, remote monitoring and direct control over cradle features.
2. To create and evaluate the algorithms for data-driven infant care analysis, using the collected sensor data to monitor sleep patterns, detect potential signs of discomfort, and log activity history. This will enable caregivers to understand their infant's needs better and make informed decisions regarding the baby's well-being.
3. To evaluate the performance of the proposed system and effectiveness of CradleSense in addressing the mental strain caused by balancing parental and work responsibilities on single mothers. This evaluation will consider user feedback and potential impacts on maternal mental health and infant development.

By achieving these objectives, the project aims to offer an intelligent and accessible caregiving solution that promotes infant safety and maternal well-being, while addressing the challenges faced by single mothers that are balancing work and caregiving roles.

1.5 Brief Methodology

The System Development Life Cycle (SDLC) provides a conceptual framework for managing the various phases involved in developing the CradleSense project. A system development methodology such as Rapid Application Development (RAD) structures, plans, and controls this process by focusing on quick iterations and user feedback to ensure proper alignment with the needs of single mothers caring for infants. Figure 1.1 below shows the phases involved in the RAD methodology.

Rapid Application Development (RAD)

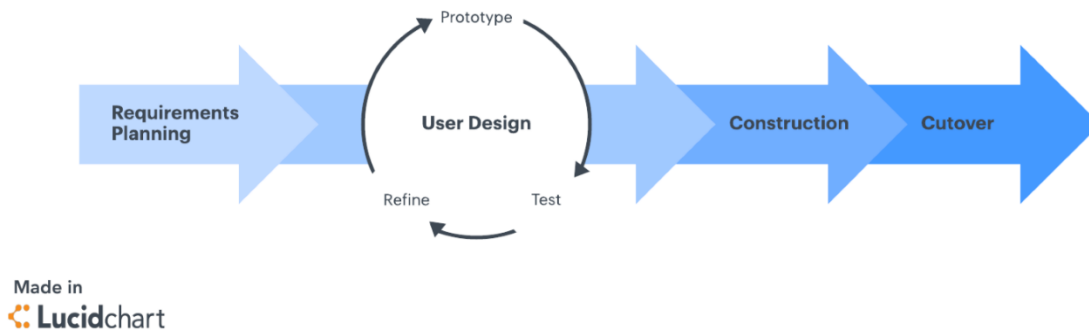


Figure 1.1: Phases of Rapid Application Development (RAD)

In this project, the RAD methodology is chosen as it enables the prototyping and iterative testing essential for developing a responsive and adaptable system. CradleSense requires real-time functionality for monitoring, alerting, and automating cradle operations. As shown in Figure 1.1, RAD's flexible approach facilitates swift adjustments based on user feedback and evolving project requirements. This method allows single mothers to provide early feedback, which can ensure the final product closely meets their needs for convenient and efficient infant monitoring and soothing.

Figure 1.2 below illustrates the phases of RAD methodology that will be implemented to guide the development of this project.

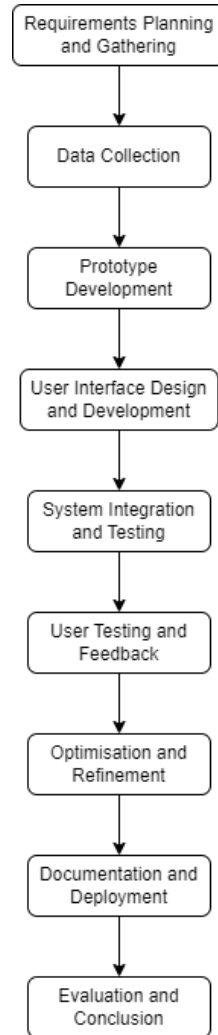


Figure 1.2: Phases of RAD methodology for CradleSense

Raspberry Pi and Arduino technology is combined with mobile app development tools to provide a seamless experience for single mothers in managing infant care. In this project, sensors such as, but not limited to, temperature (DHT11 Digital Temperature and Humidity Sensor), motion (PIR motion sensor) and audio (Sound Microphone input) are connected to a Raspberry Pi microprocessor and Arduino microcontroller, which acts as the main controller for collecting and

processing sensor data. The cradle will be equipped with actuators, including a rocking mechanism and a speaker, which will autonomously respond to the baby's stimuli, such as initiating soothing motions or playing lullabies based on the input from these sensors.

Flutter framework will be utilised for the mobile app development, primarily using Dart for defining the user interface, application logic, and functionality. The app will provide an intuitive interface for users to remotely monitor and control the cradle's functions. It enables real-time notifications and data visualisation on environmental conditions like temperature and noise levels around the cradle. The mobile app is designed to be accessible and user-friendly to ensure that single mothers can operate it efficiently.

The Firebase Realtime Database will be used for the cloud-hosted data storage, which allows real-time synchronisation between the cradle system and the mobile application. This database enables remote access, therefore allowing users to retrieve historical data on the baby's environmental conditions or sleeping patterns. This will help mothers make informed decisions when caring for their infants. Additionally, SQLite is used as a local database for storing essential data when internet connectivity is unavailable. This ensures a continuous functionality of the cradle's monitoring features.

1.6 Project Scopes

This section provides an overview of the project, outlining the key aspects of the smart cradle, CradleSense, for single mothers who are balancing work and infant care. The project scope is divided into three main aspects, which are target users, location of use, and tools and technologies.

1. Target Users

The primary users of CradleSense are single mothers who often face the challenge of managing professional responsibilities while providing continuous care for their infants. This cradle will support them by automating infant care functions, therefore reducing the strain of constant monitoring. In cases where single mothers rely on part-time caregivers, CradleSense can assist caregivers by monitoring infant needs, providing automated responses, and ensuring a high standard of care when parents are not available. This cradle is designed to support the developmental needs of infants, creating a comfortable and secure environment by adjusting based on data collected in real-time to respond to physical needs like movement, temperature, and soothing sounds.

2. Location of Use

The primary setting for this system is the home, ideally located in an infant's nursery or sleeping area where it can monitor the infant's activity and respond as needed. The cradle should be placed in a stable, well-ventilated area, with enough room for the sensors to function unobstructed. The mobile application provides remote monitoring capabilities, allowing mothers to check on the cradle's status from any part of the home or while away. Caregivers and mothers can use the mobile app to monitor the infant's status and control the cradle's functions remotely. This feature is especially useful when the mother is outside the home, allowing her to check in and receive notifications if the cradle detects any irregular activity or if the infant requires additional care.

3. Tools and Technologies

CradleSense will use a variety of sensors, including temperature, motion, and camera. These sensors will gather data on the infant's environment and conditions, enabling automated responses. For example, motion sensors can detect if the infant is awake, while audio sensors can trigger a soothing sound or song if the infant cries. The cradle's control system is built on Raspberry Pi and Arduino platforms to process sensor data and activate the cradle's functions. This system can control cradle rocking, play soothing sounds, and adjust other parameters based on the infant's needs. The mobile application, developed using Kotlin or Dart with Flutter and Android Studio, will allow caregivers to monitor real-time data, control the cradle remotely, and receive notifications of any significant changes. The app interface will be user-friendly, prioritising ease of use for busy mothers who need quick access to information.

CradleSense will log and analyse collected data to track the infant's sleep patterns, activity levels, and responses over time. This data can help mothers understand their child's needs and behaviours, potentially aiding in early detection of health issues or sleep irregularities. CradleSense aims to ease the childcare burden on single mothers by automating infant care functions and offering remote monitoring control through a mobile application. This system will provide personalised, data-driven care responses while supporting infant's comfort and safety. While the primary objective is to automate essential care functions and providing data on the infant's condition and environment, CradleSense does not include direct health interventions and is not a replacement for actual infant health care. The focus is on supporting everyday care and comfort for the child.

1.7 Significance of Project

The significance of this project lies in its implementation of IoT-driven automation to provide an intelligent infant care solution specifically designed for working single mothers. By leveraging on commonly available mobile devices and smart technology, CradleSense aims to assist single mothers in balancing their professional and parental responsibilities with minimal physical and emotional strain. This approach seeks to ensure the well-being of infants through real-time monitoring, while also alleviating the mental and physical burden single mothers who often face the challenges of caregiving alone.

CradleSense offers a practical and accessible solution by integrating sensors and automation for real-time responsiveness to the infant's needs. Using a mobile app interface, mothers can instantly monitor their child's condition, receive alerts, and adjust the cradle settings remotely. This system proactively addresses essential aspects of infant care, such as sleep patterns, movement, and comfort levels, to support a healthier environment for the child and provide peace of mind for the mother, even when she cannot be physically present.

By providing timely information on the infant's state, caregivers can make informed decisions and provide immediate care if required. Compared to traditional caregiving methods or cost specialised devices, the smart cradle system offers a more convenient, real-time solution through mobile accessibility. Traditional baby monitoring or rocking devices often lack the integration of multi-sensor data for automated response, requiring constant human intervention for manual adjustments. In contrast, CradleSense's IoT-driven design allows mothers to receive prompt alerts and control cradle functions remotely, resulting in saving time and reducing the need

for frequent hands-on supervision, which is especially beneficial during busy work hours or while handling other household tasks.

The system extends beyond basic monitoring by analysing the infant's sleep patterns and conditions over time, with options of added features like a soothing rocking mechanism or audio playback to calm the baby. These additional functionalities provide a more comprehensive support system for mothers, equipping them with the tools to enhance their child's comfort and safety, particularly during the critical early stages of development. Through automation, CradleSense minimises the need for continuous manual monitoring and adjustment, which can be physically and emotionally taxing for single mothers. By integrating sensor data with pre-programmed responses, the cradle operates objectively, adjusting itself according to real-time conditions without needed caregiver input. This intelligent system reduces subjective judgements and minimises caregiver errors that could occur from exhaustion or divided attention.

The project aims to be a cost-effective solution, designed to operate without costly subscriptions or additional fees, making it accessible to a broad range of single mothers. The integration of mobile devices for remote control and monitoring further reduces the need for expensive hardware, ensuring that essential infant care support remains available at an affordable price.

In summary, CradleSense provides an efficient and reliable approach to infant care that supports both the child's and the mother's well-being. By addressing the pressing needs of working single mothers, the system promotes better mental health by alleviating the constant worry and exhaustion often associated with solo caregiving. The project has the potential to contribute to

healthier developmental outcomes for infants and reduce the long-term stress and strain on single mothers, ultimately enhancing the quality of life for both caregiver and child.

1.8 Project Schedule

The project is scheduled using a Gantt Chart to plan the progress of the project in these ten months. Figure 1.3 shows a brief Gantt Chart task table on the overview of the project schedule according to the nine RAD, which covers the progress of Chapter 1 – Chapter 6.

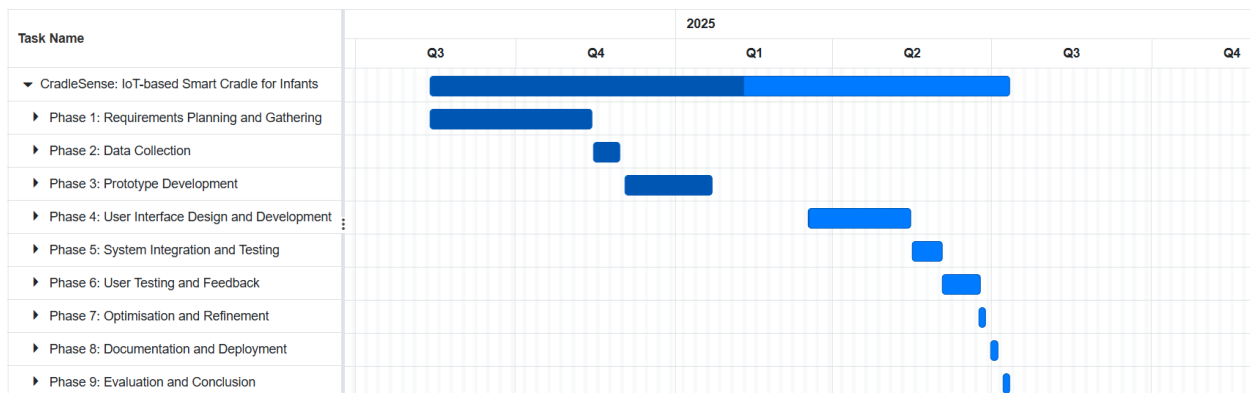


Figure 1.3: Gantt Chart Task Table on Project Timeline for Phase 1 to Phase 9

The project schedule as shown in Figure 1.3 above contains the list of project activities, milestones, followed by the start and finish date in which the project is expected to be completed by. The duration of this project is set for two semesters, which is roughly 10 months, as it spans from October 2024 to July 2025.

1.9 Expected Outcome

The successful completion of this project will showcase the development of an IoT-based smart cradle system that automates aspects of infant care through a responsive Raspberry Pi and Arduino system. This cradle integrates real-time monitoring and responsive soothing mechanisms that responds to an infant's needs based on data from various sensors. The system will utilise a mobile application to provide instant notifications, remote monitoring and control, which allows caregivers to manage the cradle's functionalities efficiently.

The project aims to achieve key objectives such as ensuring automated responses from the cradle to comfort the baby, enabling caregivers to receive real-time updates on the baby's status, and assessing the cradle's overall effectiveness in reducing caregiver strain. By incorporating these functionalities, CradleSense will offer a reliable means of infant monitoring, which can ease the caregiving burden for single mothers who struggle with balancing work, parental responsibilities and other household demands.

A comprehensive documentation and validation process will follow the cradle's development, including user testing and feedback to ensure the system meets practical needs while operating accurately. This validation process will contribute to the credibility and reliability of the system, offering insights into its performance and potential areas for future improvement. The anticipated success of this project will address the challenges faced by single mothers in balancing infant care with professional commitments, alongside other household responsibilities.

By providing real-time and automated infant care assistance, this system promotes a supportive caregiving environment while reducing the stress of managing multiple responsibilities

at the same time. Ultimately, this will contribute positively to mothers' mental and emotional well-being, therefore not compromising their care towards the infant.

1.10 Project Description

This project aims to develop an IoT-based Smart Cradle designed to enhance infant care for parents, particularly single mothers balancing work and childcare. By utilising sensors, Raspberry Pi and Arduino technology, CradleSense provides automated soothing features which responds to the changes in the infant's condition, and real-time monitoring via mobile application. The mobile application also allows mothers to receive alerts and offers remote control capabilities. Personalised insights and recommendations based on collected data will help mothers make informed care decisions. By addressing the challenges of constant supervision and the need for reliable childcare solutions, this project seeks to improve the well-being of both infants and their caregivers, fostering a supportive environment for healthy development.

1.11 Basic Skill Requirements

Raspberry Pi, Python, Sensor Integration, Flutter, Dart/Kotlin, Bluetooth/Wi-Fi Communication, Motor Control, Firebase/AWS IoT, SQLite, Actuator Control, QTT, OpenCV

1.12 Project Outline

There are five chapters in this paper to discuss and describe the smart cradle system, which leverages on IoT technology, to automate aspects of infant care, which helps single mothers reduce the physical and mental strains of infant care while balancing multiple responsibilities.

Chapter 1: Introduction provides a review on the concept of the CradleSense, which is designed to support single mothers in monitoring their infants by providing real-time feedback on the baby's status and automating soothing actions. It explains the background and importance of such a system for single mothers. This chapter also includes the problem statement, objectives, brief methodology, project significance, project schedule, expected outcomes, and project outline.

Chapter 2: Literature Review provides an analysis of existing smart cradle systems and other related IoT-based baby monitoring devices. It examines similar systems' features, such as automated rocking mechanisms, sensor-based monitoring, and app-based interfaces, comparing them with the proposed system. This chapter also discusses the unique functionalities of the proposed Smart Cradle system, including its real-time monitoring capabilities and cloud integration.

Chapter 3: Requirement Analysis and Design outlines the requirements of the CradleSense system, including both user and system requirements. Diagrams and design components presented in this chapter may include Use Case Diagrams, Flowcharts, System Architecture, Context Diagrams, Data Flow Diagrams (DFDs), and Entity Relationship Diagrams (ERDs). Wireframes for the mobile application interface are also presented to demonstrate the intended user experience.

Chapter 4: Implementation and Testing details the implementation phase of the CradleSense system, describing the process of integrating the sensors, actuators, and mobile application. It includes a technical overview of the coding structure and the software-hardware integration process. This chapter also provides test plans, covering various functionalities like real-time monitoring, alert generation, and app-based controls. The testing results and performance evaluations are discussed in detail to ensure system reliability.

Chapter 5: Conclusion and Future Work summarizes the CradleSense project, highlighting the objectives achieved and the contributions made by the system. It discusses any limitations encountered during the development and deployment phases and provides suggestions for future enhancements, such as advanced machine learning for behaviour prediction or extended sensor capabilities for improved monitoring.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter provides an in-depth analysis of existing smart cradle systems, examining their features, functionalities, and key processes. The subsequent section offers a literature review of three comparable systems. A comparative table will be created to highlight the differences between the current and proposed smart cradle systems, showcasing their strength and weaknesses respectively.

Additionally, this chapter explores the selection of essential hardware and software components for the development of the proposed smart cradle system. The evaluation aims to identify the most appropriate software and hardware that aligns with the system's requirements. The unique features found in the reviewed systems will be integrated into the proposed system, followed by a detailed discussion of its distinctive features. In conclusion, this chapter combines the comparative analysis and provides a comprehensive summary of the insights and discussions.

2.2 Review on Similar Existing Systems

The three chosen systems to be compared are SNOO Smart Sleeper Bassinet, Cradlewise Convertible Smart Bassinet and Crib, and IoT-Based Baby Monitoring System for Smart Cradle (IoT-BBMS). These systems are chosen for study and review because they have unique yet innovative features that set them apart in the domain of smart cradles. For instance, the SNOO Smart Sleep Bassinet employs advanced responsive technology to address a baby's fussing or

crying. It features an in-built swaddle system to ensure safety by preventing rolling over and automatically adjusts the rocking intensity based on the baby's level of distress. Its companion app provides detailed sleep tracking and allows parents to control the cradle remotely. This system's focus on safety and adaptability makes it a compelling point of comparison.

Next, the Cradlewise Convertible Smart Bassinet and Crib stands out for its AI-powered sleep monitoring. It tracks the baby's sleep patterns and uses machine learning to predict wake-up times, automatically soothing the baby with gentle rocking before they wake fully. Additionally, it offers features like white noise playback and compatibility with a mobile app for remote monitoring and control, which align with some functionalities of the proposed system.

Finally, the IoT-Based Baby Monitoring System (IoT-BBMS) integrates IoT technologies to monitor and respond to a baby's needs. The IoT-BBMS uses a NodeMCU ESP8266 Wi-Fi Controller Board as the main microcontroller. It employs sensors to detect environmental conditions such as sound (crying), temperature, and humidity. Based on the sensor data, automated responses are triggered, such as automatically swinging the cradle when the baby cries or turning on a fan if the temperature is too high. Additionally, the IoT-BBMS provides real-time updates to parents through a mobile app, specifically the MQTT Dash app, which is connected to the Adafruit MQTT server.

2.2.1 SNOO Smart Sleeper Bassinet

The SNOO Smart Sleep Bassinet is designed and marketed to provide a safe and soothing sleep environment for infants through its responsive technology and unique features. SNOO uses gentle,

continuous micro-jiggling motions that mimic a parent's soothing movements. Unlike traditional rocking or bouncing, the motion is smooth and non-disruptive, designed to calm the baby quickly. The motion is set to start after the baby cries and stops once the time is up. Users also note that there is a subtle rocking sound. The intensity of the movement is automatically adjusted based on the baby's distress levels, escalating only when needed and reducing as the baby calms.

The bassinet is equipped with high-sensitivity microphones to detect crying and audio cues, accelerometers to monitor the baby's movement, and ambient sensors to track room temperature and noise levels. These sensors work in harmony to provide a comprehensive understanding of the baby's needs, ensuring an immediate and tailored response to maintain a calming environment. Figure 2.1 shows the features that are available on this smart cradle.



Figure 2.1: SNOO Smart Sleeper Bassinet features (2023)

SNOO also features a companion mobile application that offers several functionalities for parents, as shown in Figure 2.2. The *Tracker* on the app provides detailed analytics of the baby's sleep patterns, including duration and wake times, which help parents identify and address potential sleep issues. Through the app, parents can remotely control various cradle settings, such as adjusting the rocking intensity, white noise volume, and motion configurations. It also allows for customization of the cradle's responses based on the baby's preferences. Additionally, the app includes safety alerts, notifying parents if the baby is not properly swaddled or if the room's environmental conditions fall outside recommended ranges.

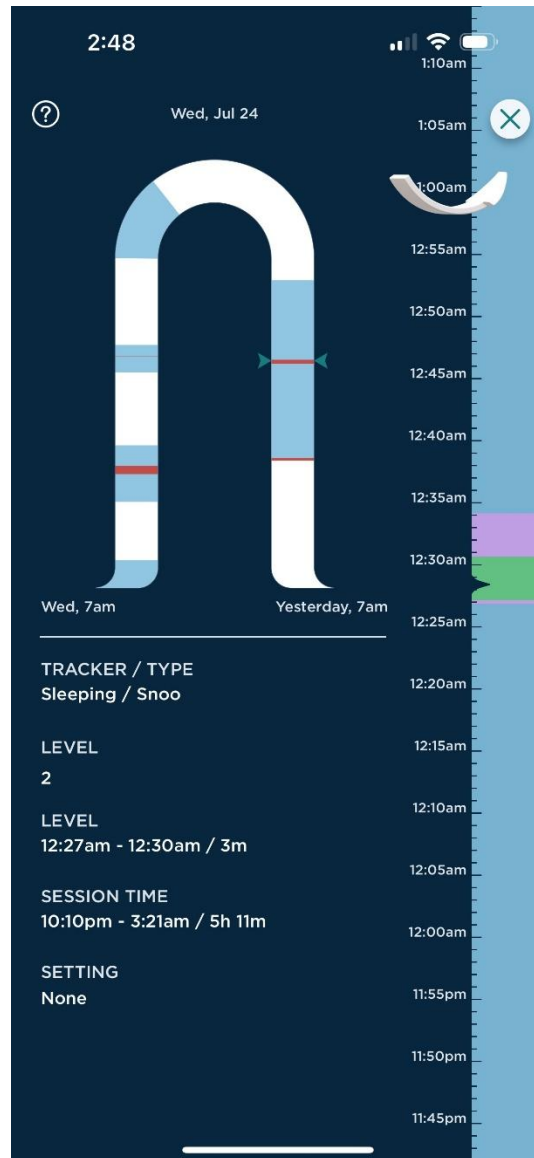


Figure 2.2: Screenshot from SNOO mobile app (2023)

One of the standout features of the SNOO is its unique swaddle system, which secures the baby safely on its back to reduce the risk of Sudden Infant Death Syndrome (SIDS). The swaddle, as shown in Figure 2.3, is designed for both comfort and safety, ensuring the baby remains securely positioned during sleep. Furthermore, SNOO employs an adaptive algorithm that adjusts its soothing patterns as the baby grows, ensuring consistent and age-appropriate care. For instance, it uses gentler motions for newborns and slightly more dynamic settings for older infants.



Figure 2.3: Built-in swaddle in the cradle (2023)

A notable downside to this product is the short product longevity. SNOO Smart Sleeper Bassinet was made for short-term use. According to multiple parents' review, SNOO Smart Sleeper Bassinet is no longer safe for infants aged six months and above, or those who can get on their hands and knees. This is due to the height of the cradle being at 81.9cm with the legs intact, meanwhile the height of a six month old infant, according to Chertoff (2018), ranges from 65.7 cm to 67.6 cm. Based on Figure 2.4, the height of the bassinet alone seems to be roughly half the total height, which gives an estimation of 40 cm. This can be easily reached by the infants, hence making it no longer safe to be used.



Figure 2.4: SNOO Smart Sleeper Bassinet (2023)

2.2.2 Cradlewise Convertible Smart Bassinet and Crib

The Cradlewise Convertible Smart Bassinet and Crib is an advanced infant care solution that combines AI technology and thoughtful design to ensure optimal sleep quality for babies. This system incorporates features that adapt to the needs of the baby as they grow, transitioning from a bassinet for newborns to a crib for toddlers, making it a long-term investment for parents. Figure 2.5 shows the dimensions of this crib once fully built.



Figure 2.5: Cradlewise Convertible Smart Bassinet and Crib dimensions

One of the key features of Cradlewise is its AI-powered sleep monitoring system, which tracks the baby's sleep patterns and uses machine learning algorithms to predict wake-up times. This predictive capability allows the cradle to initiate gentle bouncing motions, in which the cradle moves up and down instead of swinging side to side, before the baby fully wakes, therefore helping to soothe them back to sleep. The motion is designed to be rhythmic and calming, mimicking the natural movements experienced in a parent's arms. The bouncing motion is set to start before the baby cries and automatically stops when the baby is asleep. Users noted that Cradlewise silently bounces.

Cradlewise is equipped with a variety of sensors as shown in Figure 2.6 below to monitor the baby's environment and condition. These include microphones to detect crying, cameras to observe movement, and ambient sensors to measure temperature, humidity, and light levels. The

data collected by these sensors is used to adjust the cradle's responses in real-time, ensuring a comfortable and safe sleeping environment for the baby.



Figure 2.6: Features of Cradlewise

A companion mobile application enhances the usability of Cradlewise by offering a wide range of features for parents. Through the app, parents can remotely monitor their baby's sleep statistics, including total sleep duration, sleep quality, and wake-up times, as shown in Figure 2.7 below. The app also allows users to customize the cradle's settings, such as adjusting the intensity and duration of the bouncing motion or selecting white noise options. Additionally, the app provides notifications and alerts for significant events, such as when the baby is waking or if the environmental conditions need adjustment.



Figure 2.7: Screenshot of the mobile app “Sleep Insights” page

The cradle also incorporates features to aid sleep training, gradually reducing the bouncing motion as the baby’s sleep improves, helping them transition to self-soothing over time. Furthermore, Cradlewise is designed with safety in mind, ensuring the baby remains secure during use. Its sturdy construction and well-thought-out design minimize risks, providing parents with peace of mind.

By integrating AI-driven sleep monitoring, adaptive soothing mechanisms, and a comprehensive mobile app, the Cradlewise Convertible Smart Bassinet and Crib stands out as a holistic solution for infant care. Its innovative features and emphasis on long-term usability make it a strong system.

2.2.3 IoT-Based Baby Monitoring System for Smart Cradle (IoT-BBMS)

In the paper titled "IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle" by Jabbar et al. (2019), the authors propose an IoT-based system designed to monitor babies in a smart cradle. This system aims to offer real-time monitoring of the baby's environment and condition, providing a practical solution for working parents who are unable to constantly supervise their infants.

For the system mechanism, the smart cradle is equipped with various sensors such as sound, temperature and humidity, and actuators, which are all controlled by a NodeMCU microcontroller. The NodeMCU collects data from the sensors and transmits it to the Adafruit MQTT IoT server via Wi-Fi. The system's operation follows an algorithm to respond to different conditions. For instance, when the sound sensor detects the baby crying, the system automatically activates the cradle's rocking mechanism and the musical toy. If the temperature sensor detects high ambient temperatures, the system turns on a mini fan to cool the baby's surroundings. Parents can also manually control these functions through the MQTT interface. Additionally, a wireless camera provides real-time video monitoring of the baby, which can be accessed remotely via a mobile application. Figure 2.8 shows the flowchart of the system.

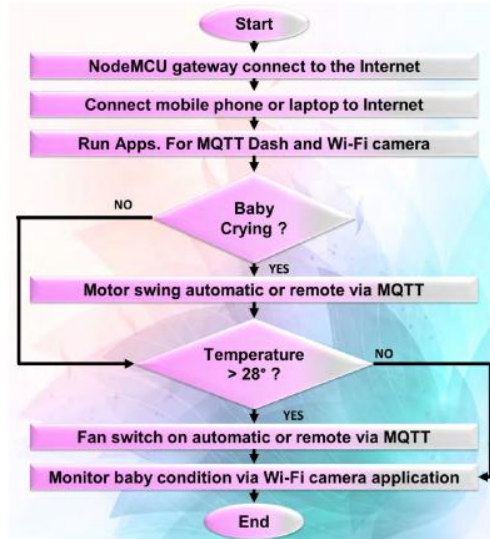


Figure 2.8: Flowchart of the system (Jabbar et al., 2019)

The advantages of this system include real-time monitoring, enabling parents to track their baby's condition and environment, such as temperature and sound levels. As shown in figure 2.9, The system also allows for remote control of the cradle's swing, fan, and musical toy. The wireless camera enhances the system by providing visual monitoring of the baby. Furthermore, the system is cost-effective, as it uses readily available components.

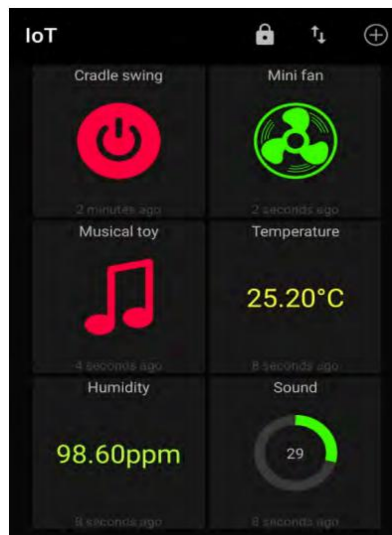


Figure 2.9: Mobile application interface (Jabbar et al., 2019)

However, according to Jabbar et al. (2019), there are some limitations to the system. The wireless camera currently requires a local network connection, restricting remote viewing to users within the same network. The GUI depends on external applications, which may have limited customization options. Additionally, data transmission may experience a slight delay, depending on Wi-Fi signal strength. The authors suggest that future improvements could include the use of an IP camera for enhanced remote monitoring, the development of a custom GUI, and the addition of more sensors to monitor a wider range of baby conditions.

2.3 Features Comparison of the Reviewed Systems

Based on the comparative analysis presented in Table 2.1, the implementation of SNOO Smart Sleeper Bassinet, Cradlewise Convertible Smart Bassinet and Crib, and IoT-Based Baby Monitoring System for Smart Cradle (IoT-BBMS) fails to fulfil the essential requirements for an effective smart cradle system designed to assist single mothers in automating aspects of childcare.

System Features	SNOO Smart Sleeper Bassinet	Cradlewise Convertible Smart Bassinet and Crib	IoT-Based Baby Monitoring System for Smart Cradle (IoT-BBMS)	CradleSense
Cry detection	✓	✓	✓	✓
Motion detection	X	✓	X	✓
Temperature monitoring	X	✓	✓	✓
Humidity monitoring	X	X	✓	✓
Automated rocking	✓	✓	✓	✓
Remote control via mobile app	✓	✓	✓	✓

Real-time camera monitoring	X	✓	✓	✓
Sound playback	✓	✓	✓	✓
Cloud integration	X	✓	✓	✓
Mobile app alerts	✓	✓	✓	✓
Price below RM1000	X	X	✓	✓

Table 2.1: Comparative Table Analysis of the Three Systems

Firstly, both SNOO Smart Sleeper Bassinet and IoT-BBMS does not offer motion detection in their system, as both of these systems' soothing motion is detected by sound. Moreover, SNOO Smart Sleeper Bassinet lacks temperature monitoring, which can be dangerous for households in warmer climate. Next, SNOO Smart Sleeper Bassinet and Cradlewise Convertible Smart Bassinet and Crib does not include humidity monitoring, making it less suitable for environments with varying moisture levels.

Thirdly, SNOO lacks the real-time camera monitoring and cloud integration. This creates a gap in reassurance, as mothers would not be able to visually monitor their baby from a distance

when they are currently occupied with other responsibilities. The lack of real-time surveillance is also dangerous as mothers would not be able to quickly assess urgent situations like choking, rolling over, or unusual movements that sensors might miss. Without cloud integration, mothers are unable to access real-time or historical data remotely, therefore limiting their ability to monitor and manage the cradle while being away.

Overall, despite IoT-BBMS fulfilling the majority of feature needs, it still lacks one of the crucial features – motion detection. CradleSense is the only system that fulfils all the requirements which includes motion detection, temperature monitoring, humidity monitoring, real-time camera monitoring, and cloud integration. It also aims to be affordable for more households, therefore being able to assist more single mothers and ease their caregiving worries.

2.4 Review on Tools and Technology

This section provides an in-depth analysis and discourse on the tools and technological prerequisites essential for the advancement of the proposed system. It delves into the explanation history, and features for both software and hardware requirements. This part can be referred to as the justification section, as it explains why the system will require specific software and hardware when there are several alternatives.

2.4.1 Software

This section will review the software used for the advancement of the proposed system.

2.4.1.1 Android Studio

Android Studio is an Integrated Development Environment (IDE) officially developed by Google for Android application development. Announced at the Google I/O conference on May 16, 2013,

it has since become the primary tool for creating Android applications. Android Studio is open-source software released under the Apache License 2.0, enabling developers to build applications for the Android platform. It is based on IntelliJ IDEA, designed specifically for Android development, and provides a comprehensive and user-friendly environment for developers to design, code, test, and deploy applications. Figure 2.10 illustrates the interface of Android Studio.

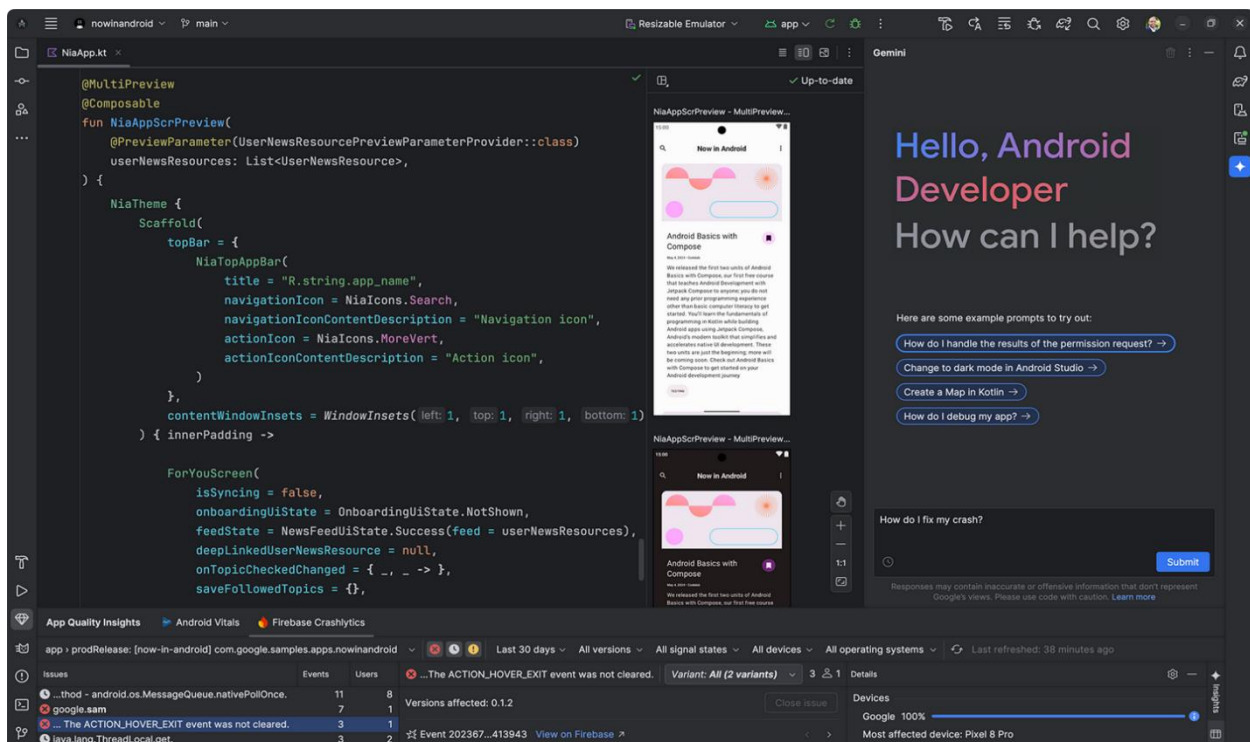


Figure 2.10: Android Studio Interface

Android Studio's first demo, version 0.1, was released in May 2013, followed by version 1.0 in December 2014. Kotlin, introduced as the recommended programming language for Android development by Google on May 7, 2019, is fully supported within the IDE. Besides Kotlin, Java remains an essential programming language available for developers in Android Studio.

Referring to Figure 2.10, Android Studio provides a central code editor panel, enabling developers to write code in Java or Kotlin. The IDE offers features such as code completion, syntax highlighting, code analysis, and refactoring, improving productivity by helping developers write cleaner and more efficient code. In May 2024, Gemini was integrated into Android Studio, similar to GitHub Copilot on Visual Studio Code. Gemini is an Artificial Intelligence (AI) powered chatbot which was developed by Google. This integration allows programmers to use Gemini as an assistive tool when developing their programs.

Additionally, Android Studio simplifies UI design through its Layout Editor, as shown in Figure 2.11. This tool allows developers to create XML layout files by dragging and dropping UI components. The ConstraintLayout Application Programming Interface (API) works seamlessly with the Layout Editor, enabling developers to design interfaces efficiently. Users can preview the layout in real-time while adjusting measurements, colours, fonts, and other properties.

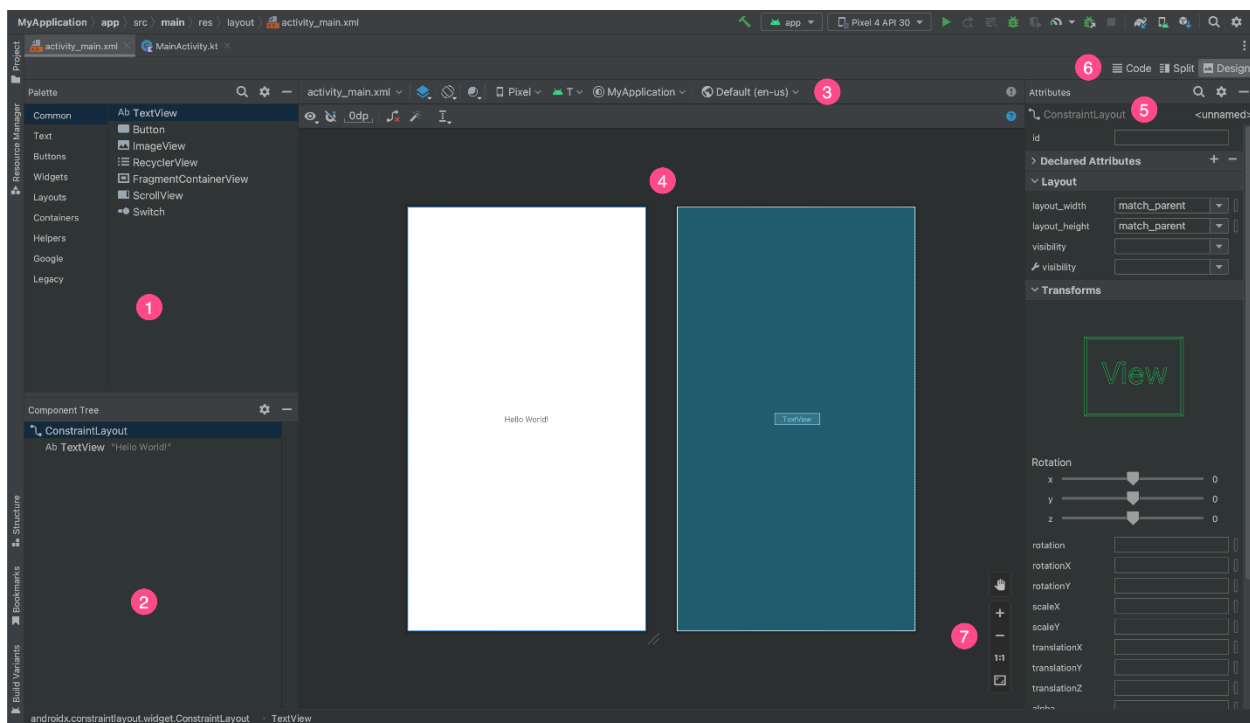


Figure 2.11: Layout Editor of Android Studio

Android Studio includes an emulator that allows developers to test applications on virtual devices with various API levels, screen sizes, and Android versions. The emulator replicates nearly all functionalities of a physical Android device, such as receiving notifications, simulating location, and rotating the screen. There is also a wide selection of Android devices available in the emulator.

Figure 2.12 shows the emulator in Android Studio.

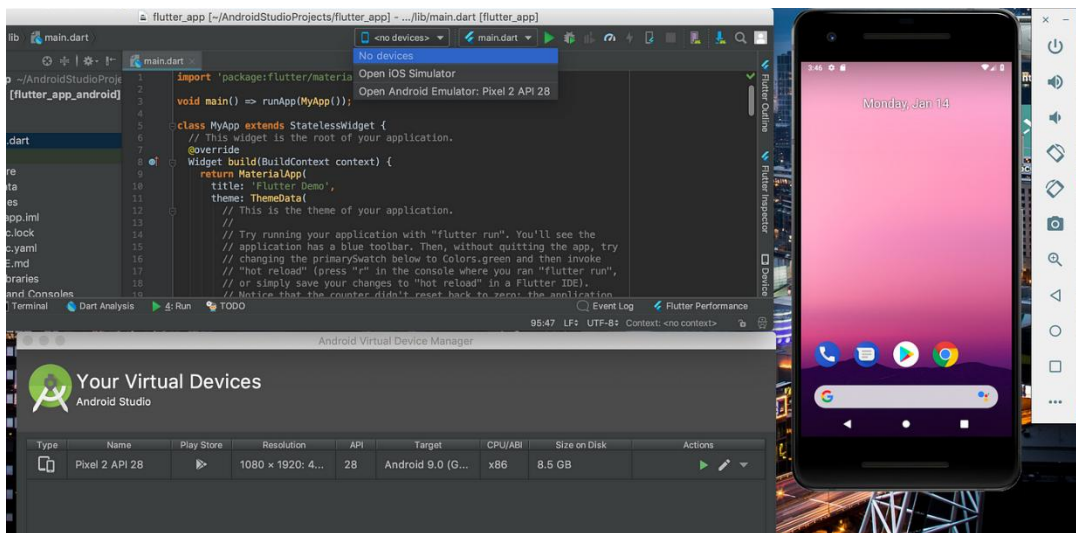


Figure 2.12: Emulator in Android Studio

The IDE supports integration with Firebase services through the Assistant tool window (Figure 2.13). Firebase provides functionalities such as real-time data synchronization, notifications, authentication, analytics, and more (Connect to Firebase, n.d.). Developers can connect their applications to Firebase directly within Android Studio, simplifying the integration of cloud services like Google Cloud Endpoints. Android Studio also supports creating Android App Bundles, a publishing format that reduces app size and improves installation times by delivering optimized APKs for various device configurations.

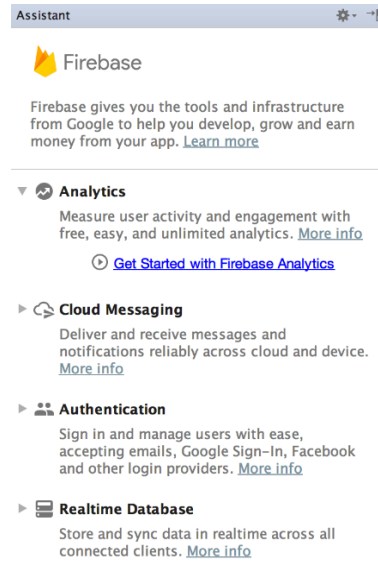


Figure 2.13: Assistant Tool Window in Android Studio

Android Studio offers templates for creating common Android app components such as activities, fragments, and services, significantly speeding up the development process (Das, 2023). It uses the Gradle build system, which simplifies the management of dependencies and the build process for Android projects. Furthermore, Android Studio includes performance profiling tools, real-time debugging, memory and performance profiling, and support for breakpoints. It integrates with version control systems like Git, enabling developers to track code changes and collaborate with others effectively.

In relation to CradleSense, Android Studio serves as a critical development platform for building a mobile application that enables parents to monitor and control the cradle remotely. The Layout Editor can be used to design a user-friendly interface for the app, allowing features like real-time alerts, live video feed, and remote control of cradle functions. The Firebase integration in Android Studio facilitates seamless real-time communication between the app and the cradle

system, while the emulator provides a controlled environment to test these functionalities thoroughly before deployment.

In summary, Android Studio provides a comprehensive development environment with tools and features that are essential for creating a mobile application. Its capabilities, such as the Layout Editor, Firebase integration, emulator make it an indispensable tool for developing robust applications for systems like CradleSense.

2.4.1.2 Raspberry Pi OS (Previously known as Raspbian)

Raspberry Pi OS is the official operating system for Raspberry Pi devices, developed and maintained by the Raspberry Pi Foundation. Formerly known as Raspbian, it is a Debian-based Linux distribution tailored specifically for the Raspberry Pi hardware (Raspberry Pi Foundation, 2023). Figure 2.14 shows the interface of Raspberry Pi OS.

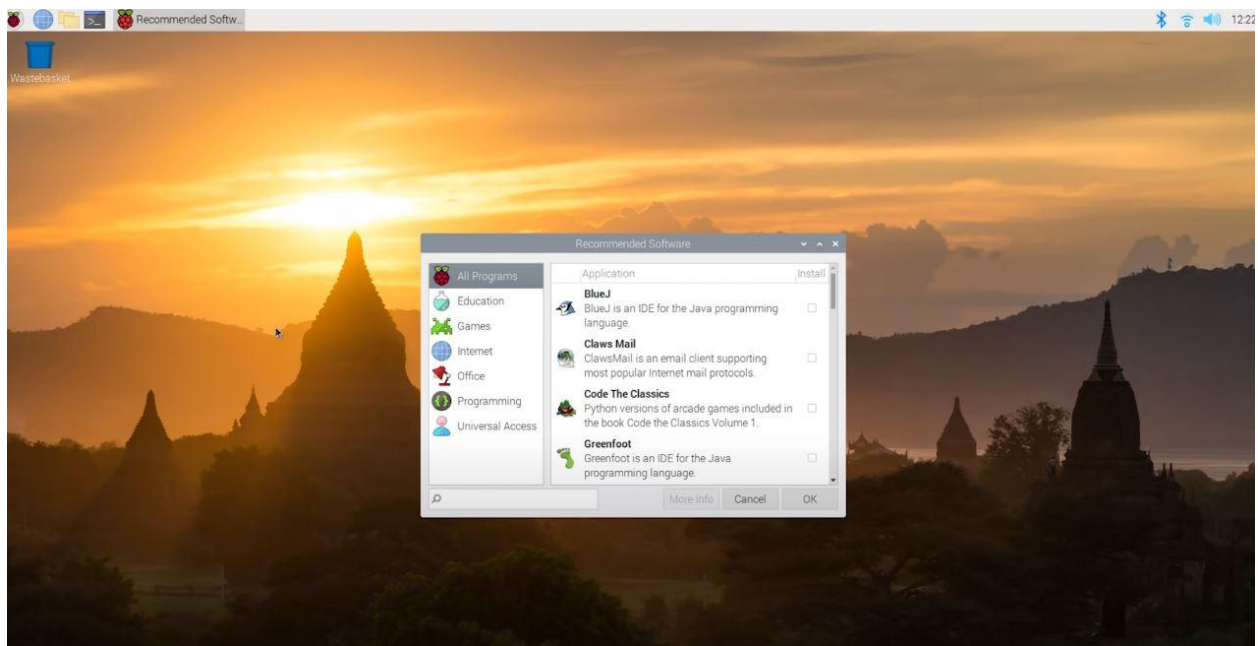


Figure 2.14: Raspberry Pi OS Interface

Raspberry Pi OS was first released in June 2012 and has undergone continuous updates to optimize its performance on Raspberry Pi hardware (Raspberry Pi Documentation, 2023). The operating system is lightweight and includes pre-installed tools and utilities designed to make it easy for beginners and experts to develop projects using the Raspberry Pi. It supports Python and other programming languages, making it versatile for a wide range of applications. Table 2.2 shows the justifications for using Raspberry Pi OS according to its key features.

Table 2.2: Features and Justifications for Raspberry Pi OS

Key Features	Justification
Lightweight and optimised	Raspberry Pi OS is specifically designed for the Raspberry Pi's ARM architecture to ensure efficient performance, even on older models. It comes with a minimal set of drivers and software to reduce system overhead, which is crucial for real-time applications like CradleSense.
GPIO Pin Support	This OS provides access to the GPIO pins through libraries like RPi.GPIO and GPIO Zero. These pins allow integration with external sensors and actuators, which are essential for detecting baby's movements, sounds, and environmental conditions in CradleSense.

Programming Tools	The OS comes pre-installed with Python and a variety of libraries for interfacing with sensors and hardware components. For instance, libraries like smbus can be used for I2C communication with sensors, meanwhile RPi.GPIO simplifies motor and LED control, which aligns with CradleSense's requirements to monitor and control cradle mechanisms.
Real-time Performance	Raspberry Pi OS is capable of handling time-sensitive tasks effectively when properly optimised. This makes it suitable for managing the sensors and actuators in CradleSense while ensuring a smooth user experience.
Network and Cloud Connectivity	Raspberry Pi OS supports Wi-Fi and Ethernet connections, which allows seamless integration with Firebase or other cloud platforms. This capability is important to enable real-time data updates, push notifications, and remote-control functionality in the proposed system.
Graphical User Interface (GUI)	This OS includes a GUI that helps to simplify tasks like testing sensors and actuators, setting up network connections, and monitoring system performance for development purposes.
Customisability	Additional software and libraries can be installed as needed. For instance, OpenCV can be installed for image processing, since CradleSense needs motion detection and live video streaming.

2.4.2 Hardware

2.4.2.1 Raspberry Pi Microprocessor

The Raspberry Pi is a compact, single-board computer developed by the Raspberry Pi Foundation, a UK-based charity focused on making computing accessible to all. Figure 2.15 shows the evolution of Raspberry Pi boards through the years. Raspberry Pi was initially launched in 2012 to enhance computer science education and has since become a versatile platform for a wide range of applications, from hobbyist projects to advanced IoT systems.

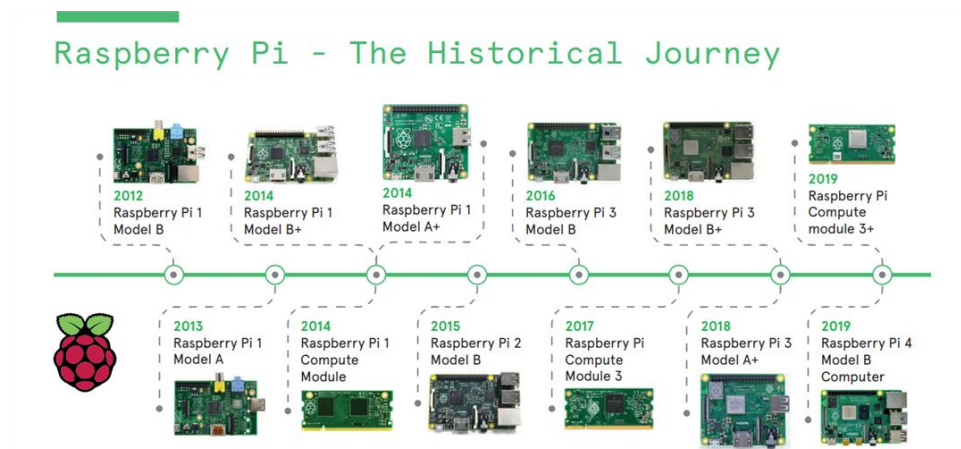


Figure 2.15: Evolution of Raspberry Pi

Its affordability, robust community support, and compatibility with various software and hardware make it particularly beneficial for projects like CradleSense. Its small form factor and adaptability allow developers to create innovative solutions, blending functionality and cost-

effectiveness. Table 2.3 explains the reason behind choosing Raspberry Pi Microprocessor for this proposed system.

Table 2.3: Features and Justification for Choosing Raspberry Pi

Features	Justification
Compact and Energy Efficient Design	The small size of a Raspberry Pi and low power consumption makes it an ideal choice for portable and energy-efficient projects. This ensures that CradleSense can operate efficiently without requiring a large or power-intensive setup.
Broad Hardware Compatibility	Raspberry Pi supports various add-on modules and accessories, such as cameras, sensors, and actuators, which enables seamless integration with the components of CradleSense. For instance, the GPIO pins enables the connection and control for sensors like motion detectors, sound sensors, and temperature sensors.
High Processing Power	Modern Raspberry Pi models, such as the Raspberry Pi 4, come equipped with quad-core processors, ample RAM which ranges from 2GB to 8GB, and a GPU, making them capable of handling computationally intensive tasks such as real-time sensor data processing and camera image analysis. This processing power is crucial for implementing features like cry detection or live video streaming in CradleSense.

Integrated Connectivity Options	Raspberry Pi boards typically include built-in Wi-Fi and Bluetooth for wireless communication. This is critical for integrating CradleSense with Firebase for real-time notifications and remote monitoring via the mobile app. Additionally, Ethernet ports provide a reliable wired connection when needed.
Expandable Storage	The Raspberry Pi uses microSD cards as its primary storage medium, which can be easily replaced or expanded. This flexibility enables storing log, images, or application data locally on the device, complementing Firebase's cloud storage.
Camera Interface	A dedicated camera interface (CSI) port enables direct connection to a Raspberry Pi Camera Module or other compatible cameras. This is particularly useful for implementing live video monitoring or motion detection in CradleSense.
GPIO Pins	The GPIO pins on the Raspberry Pi provides an interface to connect sensors and actuators. These pins support various communication protocols, such as I2C, SPI, and UART, making them versatile for connecting multiple hardware components when setting up CradleSense.
Multiple USB Ports	The multiple number of USB ports on the Raspberry Pi allows to connect peripherals like keyboards, mice, or additional sensors, which can further extend its functionality.

The Raspberry Pi will serve as the central processing unit (CPU) of CradleSense, where it will manage inputs from sensors and executing actions via actuators. Its processing capabilities enable real-time analysis of environmental data, while its connectivity options facilitate remote control and monitoring via the mobile app.

In addition, the Raspberry Pi's GPIO pins will interface with the sensors, such as sound and temperature sensors, and actuators, such as a servo motor for the rocking mechanism. Its camera interface can support live video streaming, and its wireless connectivity ensures seamless communication with Firebase for push notifications and data updates. The expandable storage provides a backup for locally saving important data, such as logs of the baby's activity and environmental conditions.

In conclusion, the Raspberry Pi's hardware features, versatility, and ease of integration make it a vital component for CradleSense, providing the computational power and connectivity necessary to deliver a robust and user-friendly solution.

2.4.2.2 Raspberry Pi Camera Module

The Raspberry Pi Camera Module is a high-quality imaging accessory that enhances the capabilities of Raspberry Pi projects by adding the ability to capture still images and video. The module is designed to integrate seamlessly with the Raspberry Pi through the Camera Serial Interface (CSI) port. It supports high-speed data transfer, which ensures efficient performance for real-time applications. The module's compact size and lightweight design make it ideal for embedded systems like CradleSense, where space and weight considerations are critical.



Figure 2.16: Different Raspberry Pi camera modules

As shown in Figure 2.16, the camera modules are available in several models, including the Camera Module 2 and the High-Quality (HQ) Camera, which offers flexibility in resolution and lens options to suit various project requirements. For example, the Camera Module 2 features an 8-megapixel sensor, which is suitable for capturing detailed visuals, while the HQ Camera supports interchangeable lenses, enabling customization based on specific needs. The module's low power consumption is another key advantage, whereby it aligns with energy-efficient designs.

The Raspberry Pi Camera Module also benefits from extensive software support. It is compatible with libraries like Picamera and OpenCV, which enables advanced image processing tasks such as motion detection, cry detection, and even facial recognition. These capabilities align seamlessly with the CradleSense's objectives, allowing for enhanced functionality like real-time video monitoring and activity alerts.

For CradleSense, the Raspberry Pi Camera Module plays an important role in providing visual monitoring capabilities. It allows parents to access live video feeds of their baby via a mobile app, ensuring peace of mind and convenience. Furthermore, its imaging capabilities can be

leveraged for motion detection, providing alerts when the baby moves or requires attention. By integrating the camera module with the cradle's software system, the project delivers a comprehensive solution that enhances safety and user experience.

Overall, the Raspberry Pi Camera Module's compact design, high-resolution imaging, and compatibility with advanced software make it a critical component in creating a robust and innovative CradleSense system.

2.4.2.3 LM35 Temperature Sensor Module

The LM35 Temperature Sensor, as shown in Figure 2.17 below, is a precise, analog temperature sensor widely used in embedded systems and IoT applications for monitoring ambient temperature. Unlike digital sensors, the LM35 provides a continuous analog voltage output that is linearly proportional to the temperature in Celsius, making it highly suitable for real-time monitoring of environmental conditions. Its compact size, low power consumption, and ease of calibration make it a cost-effective solution for temperature-sensitive applications.



Figure 2.17: LM35 Temperature Sensor Module

One of the key advantages of the LM35 sensor is its high accuracy and simplicity of interfacing with microcontrollers such as the Arduino. It requires only a single analog input pin to read the voltage output, which corresponds directly to the ambient temperature (e.g., 10 mV/°C).

This straightforward relationship eliminates the need for complex calculations or digital communication protocols, allowing for efficient integration into systems such as CradleSense.

The LM35 operates within a temperature range of -55°C to 150°C with a typical accuracy of $\pm 0.5^{\circ}\text{C}$ at room temperature. For the CradleSense system, it is configured to monitor the temperature around the baby's environment, with logic implemented to classify the reading into color-coded zones: blue for too cold, red for too hot, and black for optimal temperature. These values are transmitted to the Raspberry Pi, logged in Firebase Firestore, and displayed on the Flutter mobile app in real-time.

In the CradleSense project, the LM35 plays an essential role in maintaining infant comfort and safety. It works alongside other sensors such as the sound sensor, PIR motion detector, and moisture sensor to create a holistic monitoring environment. When temperature thresholds are breached, alerts are triggered to notify parents immediately through the mobile application.

Overall, the LM35 is a reliable, precise, and energy-efficient temperature sensor that aligns with the objectives of CradleSense. Its analog simplicity and dependable performance make it an ideal choice for projects requiring continuous temperature surveillance and real-time decision-making.

2.4.2.4 PIR Motion Sensor

A Passive Infrared (PIR) Motion Sensor is an electronic device that detects changes in infrared (IR) radiation within its field of view, typically caused by the movement of objects emitting heat, such as humans. PIR sensors are widely used in applications like security systems, automatic lighting, and motion-activated devices due to their simplicity, energy efficiency, and reliability. In

the context of CradleSense, the PIR motion sensor plays a vital role in monitoring the baby's activity within the cradle, enhancing safety and responsiveness. Figure 2.18 shows the components in a PIR sensor.

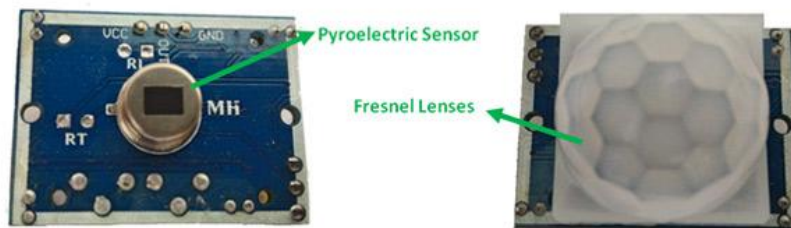


Figure 2.18: Components of a PIR sensor

The PIR sensor works by utilizing a pyroelectric sensor, which generates a small electrical signal when exposed to varying levels of infrared radiation. Since humans naturally emit heat energy in the form of infrared radiation, the sensor detects changes in these emissions when a person moves within its range. The term "passive" refers to the sensor not emitting any energy itself; it merely detects the energy radiated by other objects.

The sensor's design incorporates a dome-shaped Fresnel lens, which focuses and amplifies the infrared signals onto the pyroelectric sensor. This ensures the sensor is sensitive to even subtle movements while covering a broader field of detection. While the PIR sensor is highly effective at detecting motion, it cannot identify specific objects or individuals; it merely detects the presence of movement.

In the context of CradleSense, the PIR motion sensor is essential for monitoring the baby's movements. For example, it can detect if the baby is awake, shifting positions, or restless,

triggering appropriate actions such as sending a notification to the parent's mobile app or activating the cradle's rocking mechanism. This capability ensures that parents are promptly informed of the baby's activity, allowing for timely intervention when needed.

The PIR motion sensor's passive operation and low power consumption make it an ideal choice for CradleSense. Its integration into the system enhances the cradle's ability to monitor the baby's behaviour while maintaining energy efficiency and reliability. By combining the PIR sensor with other components, such as sound and environmental sensors, CradleSense provides a comprehensive monitoring solution, ensuring the safety and comfort of the baby at all times.

2.4.2.5 Servo Motor

A servo motor, as shown in Figure 2.19 below, is a rotary actuator or linear actuator designed to provide precise control of angular or linear position, velocity, and acceleration. It is widely used in applications requiring accurate motion control, such as robotics, automated systems, and embedded devices. For CradleSense, the servo motor plays a crucial role in driving the cradle's rocking mechanism, delivering smooth and controlled motion to soothe the baby.



Figure 2.19: Servo motor

The servo motor operates using a control signal, typically a Pulse Width Modulation (PWM) signal, to determine its position. Unlike standard motors, servo motors are equipped with an internal feedback mechanism that constantly monitors the motor's position and adjusts its movement to match the desired angle. This feedback loop ensures high precision and stability, making servo motors ideal for applications that demand accurate and repeatable movements.

Servo motors consist of three main components, which are a DC motor, a gearbox for torque amplification, and a control circuit for interpreting the input signals. This integrated design allows the servo motor to achieve the desired position with minimal delay and minimal overshooting. Additionally, most servo motors are compact and energy-efficient, making them suitable for projects where size and power consumption are critical considerations.

In terms of CradleSense, the servo motor is responsible for generating the gentle rocking motion that mimics a parent's natural soothing actions. This motion helps to calm and comfort the baby, encouraging better sleep and reducing stress. The servo motor's ability to execute precise, programmable movements ensures that the rocking intensity and duration can be customized according to the baby's needs. For instance, a gentle motion can be activated when the baby is sleeping, while a more pronounced motion might be triggered in response to the baby's cries or restlessness.

The servo motor's compatibility with microcontrollers, such as the Raspberry Pi, makes it easy to integrate into the CradleSense's system. Using PWM signals, the Raspberry Pi can control the motor's position and speed, therefore enabling dynamic adjustments in real time. This capability is especially beneficial when combined with inputs from sensors, such as sound or motion detectors, to create a responsive and intelligent cradle system.

Overall, the servo motor's precision, reliability, and adaptability make it an indispensable component of CradleSense. By providing controlled and consistent rocking motion, it enhances the cradle's functionality and contributes to the overall comfort and well-being of the baby.

2.4.2.6 Speaker

A speaker is an important component for CradleSense as it provides auditory feedback that enhances the user experience. It is responsible for playing soothing sounds, lullabies, or parental voice recordings, which help calm the baby. Figure 2.20 shows the size of the speaker.



Figure 2.20: Mini 8 ohm 1W speaker

The speaker used in this project is compact and energy-efficient, ensuring seamless integration into the cradle's design without excessive power consumption. It can be connected to the Raspberry Pi via GPIO pins or a 3.5mm audio jack, depending on the model selected. This flexibility allows for straightforward installation and compatibility with various audio formats.

2.4.2.7 MSI Modern 15 Laptop

The MSI Modern 15 laptop is a versatile and high-performance device, integral to the development and operation of CradleSense. Manufactured by Micro-Star International (MSI), a leading Taiwanese multinational company, MSI is renowned for its high-quality computer hardware, particularly gaming laptops and workstations. Known for its innovative designs and advanced performance capabilities, MSI has become a trusted brand among professionals and tech enthusiasts worldwide. The Modern 15 laptop embodies this reputation, offering a blend of portability, powerful performance, and durability that makes it an excellent choice for both development and operational purposes in IoT projects such as CradleSense. Table 2.4 explains the justification on using MSI Modern 15 Laptop.

Table 2.4: Features and Justification for Choosing MSI Modern 15 Laptop

Features	Justification
Processing Power	MSI Modern 15 uses the 11 th Gen Intel Core i7 processor, which ensures a smooth performance for Android Studio, Flutter development, Firebase integration, and to code program for Raspberry Pi.
RAM	This laptop has 8GB of RAM, which will enable smooth multitasking when running development tools such as IDEs and debugging tools.
Storage	MSI Modern 15 has 512GB SSD that can provide fast read/write speeds for storing project files, datasets, and software tools.

Portability	The laptop weighs at 1.6KG and is easily portable, which allows for on-site testing at different locations or for the purpose of presenting CradleSense.
Port Compatibility	MSI Modern 15 has both USB-C and USB-A ports that can support compatibility with various peripherals, including the Raspberry Pi.

The MSI Modern 15 laptop is crucial for developing, testing, and maintaining the CradleSense's system. Its processing power and ample memory allow efficient use of Android Studio and Flutter for creating the mobile app interface. Additionally, the laptop serves as a reliable power source for the Raspberry Pi during initial testing phases. The high storage capacity enables secure archiving of project data, while its connectivity options facilitate seamless integration with Raspberry Pi hardware and cloud-based tools like Firebase.

2.4.2.8 Arduino Uno Microcontroller

The Arduino Uno is an open-source microcontroller board based on the ATmega328P. Developed by the Arduino Project, it is widely used for prototyping and educational purposes due to its user-friendly design and flexibility. Figure 2.21 shows the standard layout of an Arduino Uno board, including its I/O pins, USB interface, and power ports. Introduced as a tool for designers, engineers, and students, the Arduino Uno has since become a staple in embedded systems, robotics, and IoT projects.



Figure 2.21 Arduino Uno Microcontroller

The Arduino Uno is especially suitable for sensor-based control systems like CradleSense. With its analog and digital input/output pins, built-in ADC (Analog-to-Digital Converter), and compatibility with numerous sensor and actuator modules, it serves as a reliable hardware interface for direct interaction with the physical environment. Table 2.4 presents the features and rationale behind choosing the Arduino Uno for this project.

Table 2.5 Features and Justification for Choosing Arduino Uno Microcontroller

Features	Justification
Real-Time Hardware Interaction	The Arduino Uno is well-suited for handling real-time input and output operations. This is essential for reading sensors like the sound

	sensor, motion sensor, LM35 temperature sensor, and water sensor, as well as controlling the servo motor for rocking.
Simple Programming Environment	Arduino IDE provides an intuitive platform for coding, uploading, and testing, which accelerates development and debugging processes in CradleSense.
Analog and Digital I/O Support	With 14 digital I/O pins and 6 analog input pins, the board supports multiple sensor connections simultaneously, enabling efficient monitoring of baby conditions.
Low Power Consumption	The board operates on a low voltage (5V), making it energy-efficient for continuous operation within the cradle without excessive battery drain.
Serial Communication Capability	The Arduino communicates effectively with the Raspberry Pi through serial UART. This allows data such as sensor readings and trigger alerts to be relayed reliably.

2.5 Summary

To conclude, Chapter 2 provided an in-depth review of multiple comparable systems related to IoT-based infant monitoring solutions. This included a detailed examination of the strengths and limitations of existing systems, followed by a comparative analysis against the smart cradle system. The discussion was structured around two core components: software and hardware. As outlined,

no existing system fully integrates advanced IoT capabilities with real-time monitoring features such as cry detection, motion detection, and environmental condition tracking within a single, cohesive framework.

The proposed smart cradle system stands out for its innovative use of Raspberry Pi, versatile sensors, and actuators to address multiple infant care challenges simultaneously. By leveraging IoT technologies and integrating multiple features like live video and audio monitoring, automated rocking, and real-time notifications, the system represents a significant advancement in the field. This chapter underscores the novelty and practicality of the proposed system in combining essential functionalities into a singular, intelligent solution for busy parents.

CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

Chapter 3 will provide a detailed explanation of the chosen methodology for developing the proposed application. According to (Bahishti, 2022), methodology refers to the structured approach used in planning, designing, and executing a project. It encompasses a range of techniques and processes, such as design planning, data collection, analysis, and evaluation, to ensure that the project is carried out systematically and rigorously. A well-defined methodology provides a framework that guides the entire process, which helps to produce reliable and valid results. In terms of system development, this approach ensures consistency, quality, and scalability which makes it essential for the success and completion of complex projects. Out of all the available methodologies, Rapid Application Development (RAD) has been chosen as the most effective approach for efficiently developing and delivering high-quality application with a reduced development timeframe. RAD is type of software development methodology that prioritizes flexibility and adaptability by enabling teams to develop software solutions through rapid prototyping and iterative feedback (Chien, 2020), replacing traditional plan-driven methods with a process focused on quick adaptation to changes, informed by data-driven decisions and continuous insights from evolving requirements (Beres, 2022). Therefore, the RAD methodology has been selected and will be fully utilised as the framework for planning, analysing, developing, testing, and implementing the proposed application and system.

3.2 Rapid Application Development (RAD)

Rapid Application Development (RAD) is a development life cycle methodology aimed at achieving an accelerated system development and enhanced quality in comparison to traditional models. According to the Digital Policy Office of Hong Kong (2024), The primary objectives of

RAD are high speed, high quality, and low cost. This approach is both people-centred and incremental, therefore emphasizing on active user involvement and fostering collaboration and cooperation among all stakeholders. An integral part of the RAD process is continuous testing, as this ensures that both developers and users are able to review and validate the system incrementally throughout the development life cycle.

In RAD, prototypes or mock-ups of the software's user interface and functionality are created in the early development process. These prototypes serve as tools to gather feedback from users and stakeholders, which helps refine the application's design. RAD promotes an iterative approach, where developers work on smaller, more manageable sections of the application where it is refined and expanded through multiple cycles or iterations. Additionally, RAD encourages close collaboration between developers, users, and other stakeholders. Regular feedback and communication are vital to ensure that the software meets users' needs and expectations. To further accelerate the development, RAD often utilises pre-existing software components or libraries, including third-party APIs, frameworks, or modules that offer common functionalities.

Moreover, RAD typically relies on tools and technologies that support quick prototyping and development, such as visual development environments and low-code or no-code platforms. It also allows multiple teams or individuals to work on different parts of the application simultaneously, therefore speeding up the project timeline. The methodology is adaptable to changing requirements, hence able to accommodate alterations or additions to the project scope without significant delays.

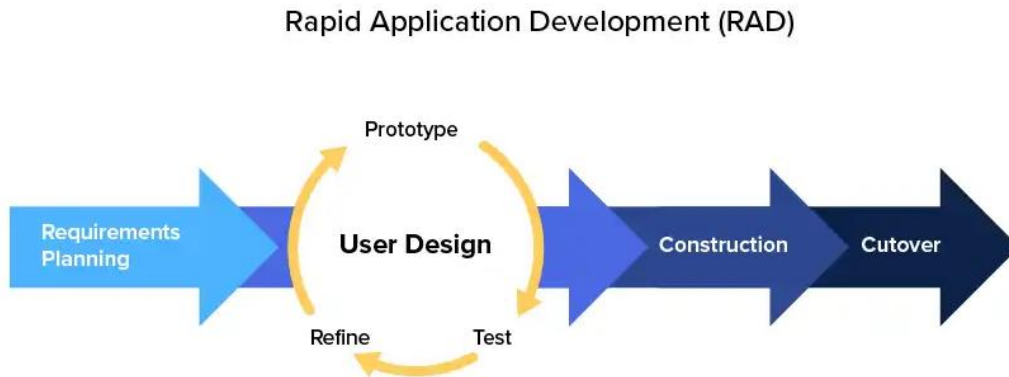


Figure 3.1: The 4 Phases of Rapid Application Development (RAD)

RAD consistently follows the four key phases as shown in Figure 3.1 above, despite being subdivided into several phases using various approaches. The phases are often referred to as the “RAD cycle” or the “RAD framework”. These four phases are:

1. Requirements Planning

In developing a system, the Requirements Planning phase is a crucial phase for establishing the foundational architecture of a system. During this phase, the development team works closely with stakeholders to identify the key functionalities and system requirements. This includes understanding the technical environment, defining the data architecture, and setting up the basic framework that will guide the system’s development. Since RAD focuses on rapid iterations, this phase avoids the extensive upfront documentation. Instead, it captures enough detail to start the development process quickly while allowing flexibility for changes. This phase lays down the system’s goals, the primary modules, and integration points with other systems or databases, therefore ensuring that the project aligns with business objectives from the start.

2. User Design

A system development becomes more concrete through the creation of prototypes and user interfaces in the User Design phase. Developers and users collaborate to build functional prototypes that represent different system components. These prototypes are iterative, with each version improving based on user feedback. During this phase, developers focus on user interface design, system navigation, and workflow processes. This will ensure they are intuitive and meet the user expectations. This phase often uses tools like UI/UX design software and interactive prototyping platforms. By focusing on iterative designs, developers can identify the potential system flaws or areas of improvement early on. As a result, the need for major changes later in the development process can be reduced.

3. Construction

The Construction phase is where the system is built, drawing from the prototypes and user feedback obtained in the previous phase. This phase involves coding the system's components, integrating various modules, and ensuring that the system architecture supports scalability and performance. Developers use rapid development tools, frameworks, and reusable components to speed up coding. The system's core functionalities are developed, and continuous testing is carried out to ensure each module works correctly and integrates seamlessly with the others. This phase also involves setting up databases, creating APIs, and configuring servers. The emphasis on quick development cycles allows for continuous integration and deployment of system updates as new requirements emerge.

4. Cutover

In the Cutover phase, the system is moved from the development environment to the production environment. This involves several critical tasks in system development, including system testing, performance tuning, data migration, and deployment. System testing ensures that all components work together correctly, and any bugs or issues identified are fixed. Performance tuning is done to optimise the system for real-world use to ensure it can handle the expected load. Data migration involves transferring existing data into the new system, ensuring data integrity and consistency. Finally, deployment makes the system available to users, often starting with a soft launch or phased rollout to manage any unexpected issues. This phase also includes user training and documentation to ensure that users can effectively use the new system. Post-deployment support is provided to address any problems that arise and to make further refinements based on user feedback.

3.3 Requirements Planning

Requirements Planning is the foundational phase in the Rapid Application Development (RAD) methodology and involves identifying the core problem and defining specific objectives for CradleSense. This phase includes activities such as brainstorming sessions and data collection to formulate an initial project plan and establish a shared understanding between stakeholders and developers. An essential aspect of this phase is gathering system requirements from end users, which are primarily mothers, to ensure the proposed Smart Cradle genuinely aligns with their needs and expectations.

This phase also involves an in-depth examination of existing solutions and technologies, which provided valuable insights for planning both the functional and non-functional aspects of

the system. These insights helped determine the core features of the proposed system, such as automated soothing mechanisms, environmental monitoring, and remote-control capabilities. Before progressing to the user design phase, finalizing the requirements with stakeholders was essential to mitigate the risk of miscommunication. To achieve this, a targeted questionnaire was distributed to mothers to gather their perspectives on current challenges and expectations. The responses were thoroughly analysed to inform the system's design and ensure it effectively addressed their needs. This structured and collaborative approach established a strong foundation for subsequent development stages, aligning the project with user expectations and technical feasibility.

3.3.1 Analysis on Similar Existing Systems

Existing smart cradle systems often lack the integration of essential sensors such as sound detection, motion detection, and moisture sensors, along with live video feed, which results in limited functionality. Many systems are designed for singular tasks like rocking or playing soothing lullabies, leaving parents to rely on multiple devices for comprehensive care, which also leads to increase in cost and complexity for infant care. Another key limitation is the absence of real-time responses and actionable insights.

Additionally, user interfaces in current systems can be not parent-friendly, with complicated setups, limited remote control options, and inadequate mobile app integration. The lack of cloud connectivity and real-time notifications further diminishes convenience for parents who are away their baby's cradle. These limitations highlight the need for a smart cradle system that integrates advanced sensors, provides real-time responses, delivers actionable insights, and offers a user-friendly experience tailored to modern parenting needs.

3.3.2 Analysis on Proposed System

The proposed Smart Cradle system has been developed as an innovative solution to address the challenges of current baby monitoring systems. After evaluating the existing systems and their limitations, this system was designed to provide a comprehensive monitoring for infants while assisting parents, specifically single mothers with automated responses for soothing and caring for their babies. The system incorporates IoT technologies to detect the infant's physical and surrounding conditions, such as a crying detection, motion detection and even temperature sensor, which will then trigger appropriate actions such as automated rocking, playing lullabies, and sending alerts to parents.

The main objective of CradleSense is to assist single mothers with infant care by automating aspects of infant care and as a result, it can alleviate the mental strain on managing infant care while balancing other responsibilities. This can be achieved by improving parental control over infant care by integrating sensor data and actuators, which will then provide real-time monitoring and feedback through a mobile application. This mobile application enables parents to receive notifications regarding their infant's condition, control cradle settings remotely, and track sleep patterns. The system utilises a Raspberry Pi with multiple sensors, which includes but is not limited to, temperature sensors, motion sensors, audio sensors, combined with actuators such as servo motor and camera module. These components all together allows mothers to assess the infant's condition, and the sensor data will trigger soothing mechanisms.

Furthermore, the proposed system provides a cloud-based platform for data storage, allowing mothers to review infant sleep patterns and activities over time. By using cloud integration, mothers can access their infant's historical data, therefore providing them with insights

into their baby’s development and sleep habits. Table 3.1 below shows the key features and functionalities of CradleSense’s proposed system.

Table 3.1: Key Features and Functionalities of CradleSense

Mothers (Mobile Application)	Cradle (Hardware)	Cloud Platform
• Login to the app • Receive notifications based on triggered parameters (e.g., crying, motion) • View live audio and video feed • Adjust cradle settings remotely • View sleep insights	• Detect infant crying • Monitor temperature and humidity • Trigger rocking mechanism • Play lullabies/white noise to soothe infant	• Store activity data • Analyse infant sleep patterns • Generate reports/visual analysis

The proposed system offers real-time feedback and automated infant soothing mechanisms based on detected sensor data. This eliminates the need for constant parental supervision, therefore offering mothers a peace of mind while still allowing control over the baby’s care. It also helps mothers to track and improve their infant’s sleep patterns and overall well-being by providing easy-to-read data through a user friendly mobile interface. Additionally, by using the Raspberry

Pi's sensors and actuators, the system provides real-time and objective responses to the infant's needs. In return, the system is able to support better sleep and comfort for the infant.

3.3.3 Questionnaire

A questionnaire was distributed to several mothers and received 30 respondents in total, with the aim of gathering insights. This questionnaire was conducted to not only get a sense on mothers' opinion on current existing solutions, but also to determine the expectation of potential users and whether the proposed system meets the requirements of probable consumers. There are six sections in total, featuring both open-ended and close-ended questions that require responses from the intended participants. Pie charts and column bar charts were selected to visually represent certain key questions. These questions were specifically highlighted due to their critical role in determining the development process. An in-depth analysis of the participants' responses was carried out with the findings summarised in the following sub-sections.

3.3.3.1 Demographics

The demographic section comprises of seven questions inquiring about the respondents' age, employment status, occupation, current work arrangement, marital status, number of children in their household and the age range of their youngest child.

What is your age?
30 responses

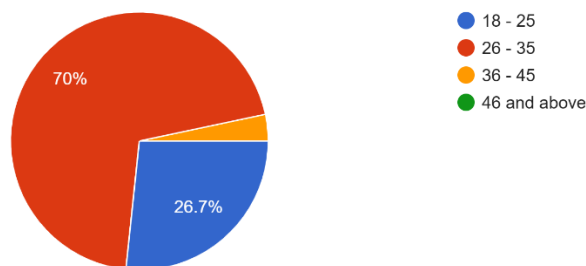


Figure 3.2: Age Range of Respondents

In Figure 3.2 above, we observe a total of 30 respondents, all of whom are mothers. The reason for targeting mothers, instead of purely single mothers as the respondents, is that the system is designed for single mothers with infants. Single parent or not, mothers are often the primary caregiver for a child. Therefore, the input of a non-single mother is still relevant to this data collection. Based on Figure 3.1, 21 (70%) of the respondents ranges between age 26 to 35 years old, 8 (26.7%) of respondents fall under the age range of between 18 to 25 years old, meanwhile the remaining 1 (3.3%) range between the ages 36 to 45 years old. Within this respondent group, 19 (63.3%) are full-time employees, 10 (33.3%) are currently not employed, and 1 (3.3%) respondent is a full-time employee and part-time student.

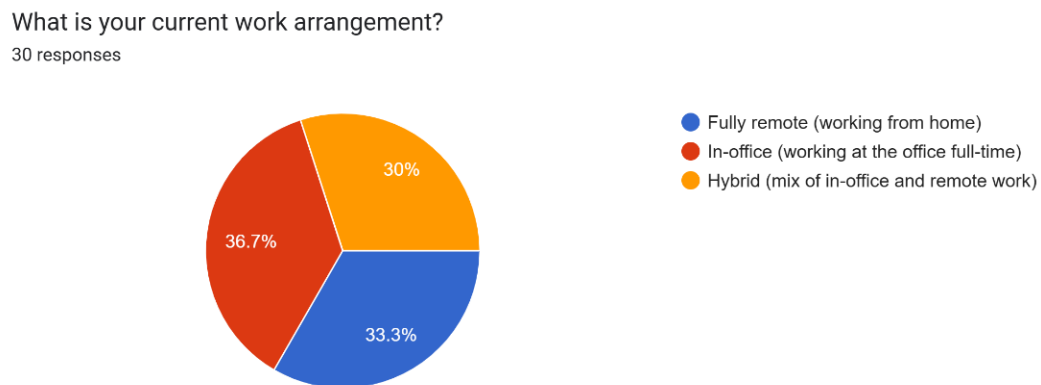


Figure 3.3: Respondents' Current Work Arrangements

Out of the 30 respondents, 9 (30%) reports to be a Housewife, meanwhile the remaining 21 (70%) respondents are employees in various work fields. In addition, as shown in Figure 3.3, the majority of the respondents, 11 (36.7%) of them are working in the office full-time, 10 (33.3%) of them are working fully remote from home, while 9 (30%) have a hybrid working arrangement, where they occasionally work from home, and occasionally need to be in the office.

What is the age range of your youngest child?

30 responses

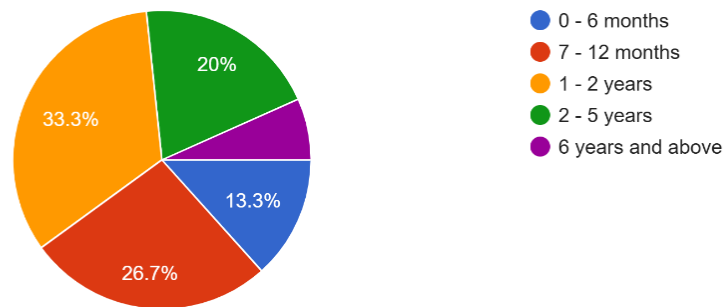


Figure 3.4: Age Range of Respondents' Youngest Child

The majority of the respondents, 25 (83.3%) of them are married, 4 (13.3%) of them are single parents, while the remaining 1 (3.3%) respondent reports to be single. Out of the 30, 22 (73.3%) of the respondents have one child in their household, 7 (23.3%) of which has between two to three children, and 1 (3.3%) of the respondents have four children and above in their household. Based on Figure 3.4, the majority of the respondents' children, 10 (33.3%) of them reports that their youngest child is between ages 1 to 2 years old. 8 (26.7%) respondents' youngest child ages between 7 to 12 months old, 6 (20%) respondents have their youngest child at ages between 2 to 5 years, 4 (13.3%) of the respondents have children ages 0 to 6 months youngest, while the remaining 2 (6.7%) respondents' youngest child ages report to be 6 years old and above.

3.3.3.2 Current Solutions

This section is to gather insights on the current solutions the respondents use to assist in infant care. In this section, the majority of the respondents, 26 (86.7%) of them documents that they never used any baby monitoring tools, 2 (6.7%) respondents claim that they use baby monitoring tools a few times a week, 1 (3.3%) respondent documents they use the baby-monitoring tool daily while the remaining 1 (3.3%) respondent rarely uses the baby-monitoring tool.

What baby-monitoring tools or methods do currently use?

30 responses

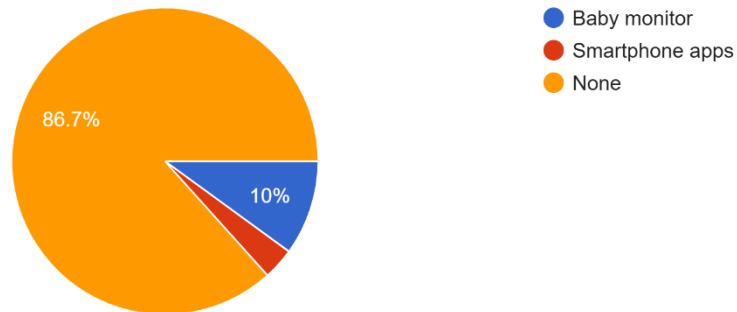


Figure 3.5: Types of Baby Monitoring Tools Used by Respondents

According to Figure 3. 5 above, 26 (86.7%) of the respondents does not use any form of baby monitoring tool, 3 (10%) of the respondents documented to have used baby monitors while the remaining 1 (3.3%) respondent documents to have used smartphone apps as a monitoring tool. The follow-up question for this inquires the respondents to document the smartphone or baby monitoring tool used. Two documented answers were received, which are WhatsApp and the Xiaomi XiaoVV webcam. It is implied that the respondent uses WhatsApp as a method to monitor their baby via frequent communication or video calls with the caregiver while the respondent is at work, while the Xiaomi XiaoVV webcam serves as a DIY (Do It Yourself) baby monitor.

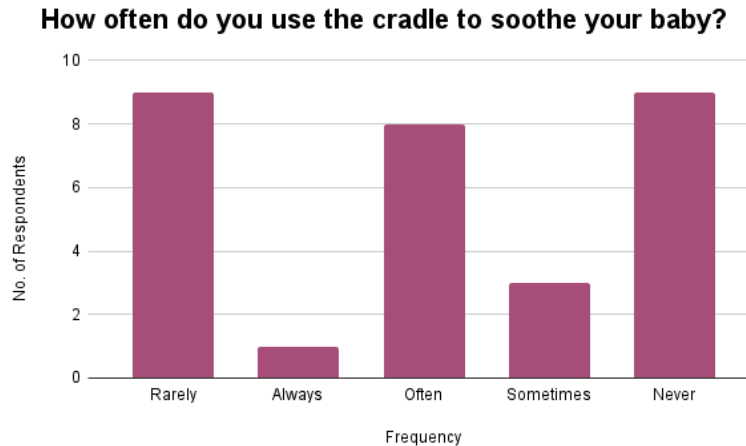


Figure 3.6: Frequency Respondents Use Cradle to Soothe Their Baby

As depicted in the column graph of Figure 3.6, the questionnaire records that the majority of the respondents, 18 (60%) records that they rarely to never use a cradle to soothe their baby. 8 (26.7%) respondents records that they often use their cradle, 3 (10%) of the respondents document that they sometimes use their cradle to soothe their baby while the remaining 1 (3.3%) respondent always uses their cradle. Figure 3.7 below shows the type of cradles the responders use. 11 (36.7%) of the responders does not use any cradle, 6 (20%) respondents use a semi-automated cradle, another 6 (20%) uses a spring cot, while the remaining 7 (23.3%) uses a basic cradle.

What type of cradle are you currently or have experience in using?

30 responses

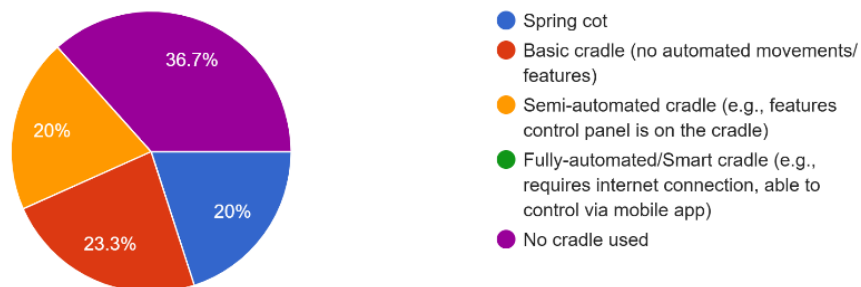


Figure 3.7: Type of Cradle Used by Respondents

3.3.3.2.1 Spring Cot and Basic Cradle Users

Based on the respondents' answer in Figure 3.6 above, a series of follow-up questions are presented according to the respective cradle types used. Respondents were asked to select all the challenges with their current cradle that applied. Out of the 13 respondents who are using a basic cradle and spring cot, the main challenge noted by this group of respondents with their current cradle is that it requires constant supervision. Another notable challenge with using basic cradle and spring cot is they lack additional features and is time consuming to operate manually. The votes casted by this group of respondents is illustrated as in Figure 3.8 below.

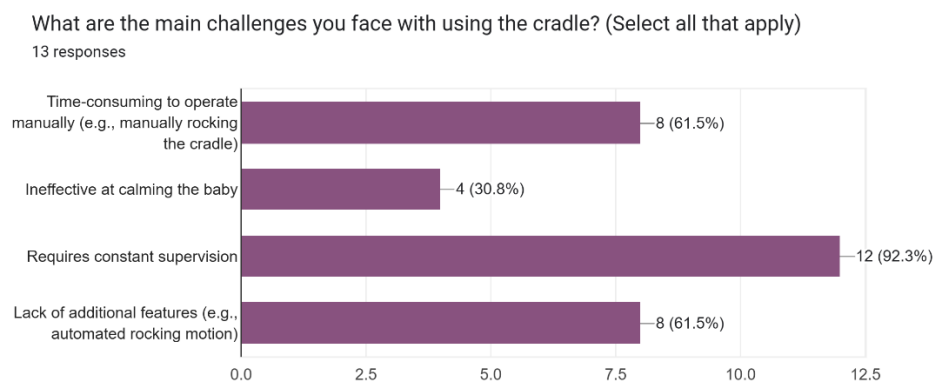


Figure 3.8: Challenges Faced by the Respondents with Their Current Cradle

The 13 respondents were then asked to select the features that they believe would improve their current cradle experience. The bar graph in Figure 3.9 below shows the findings. The three main features that highly contributes to a better cradle experience according to this group of participants are real-time alerts on the baby's condition, automated rocking and soothing, and remote control abilities via mobile app.

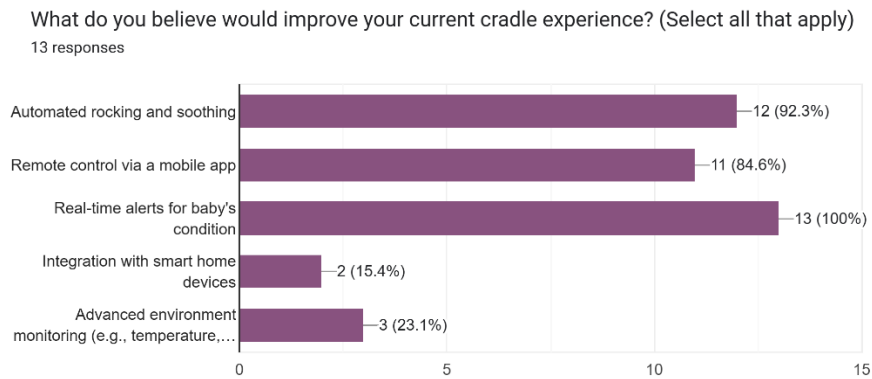


Figure 3.9: Cradle Features that could Improve Respondents' Cradle Experience

Respondents were also asked if they would consider upgrading to a fully smart cradle if it addressed their current challenges. Based on the documented results, 7 (53.8%) out of the 13 respondents said yes, while the remaining 6 (46.2%) said maybe. This finding is reflected in the pie chart below as shown in Figure 3.10.

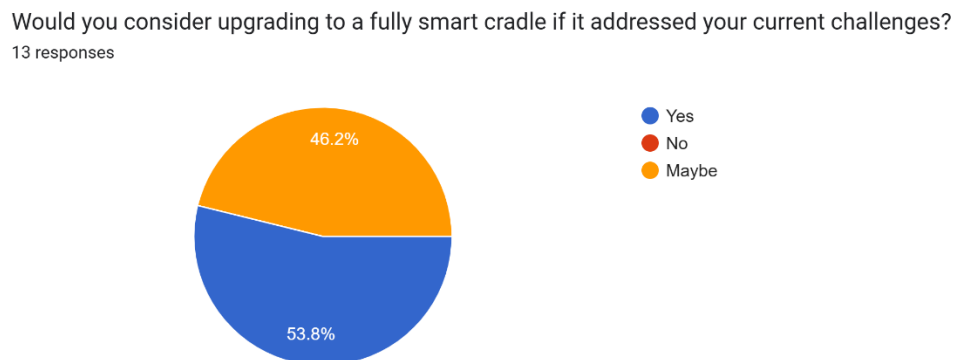


Figure 3.10: Basic Cradle and Spring Cot Users Consideration to use Smart Cradle

3.3.3.2.2 Semi-automated Cradle Users

The 6 respondents who are using a semi-automated cradle were asked a series of follow-up question to further understand their user experience. The two top features that were offered by

their semi-automated cradle are automated rocking mechanism and soothing sounds for lullabies. The 6 respondents also reported limitation that they experience with their cradles. The limitations are as shown in Figure 3.11 below.

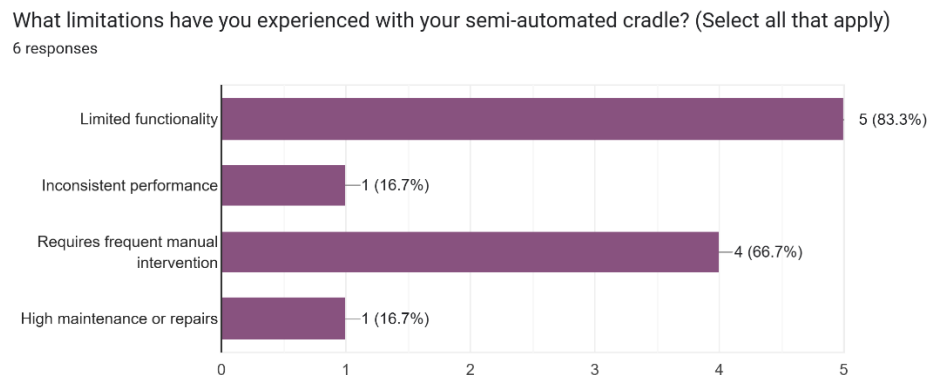


Figure 3.11: Semi-automated Cradle Limitations

Based on Figure 3.11, it was reported that the main limitations are limited functionality and the frequent requirement for manual intervention. Despite the limitations, the respondents documented that they liked their semi-automated cradle due to the automated rocking feature and the ability to play soothing sounds. As shown in Figure 3.12 below, the features that they deem would improve their current cradle experience are real-time alerts for their baby's condition and the ability to remotely control the cradle via a mobile app.

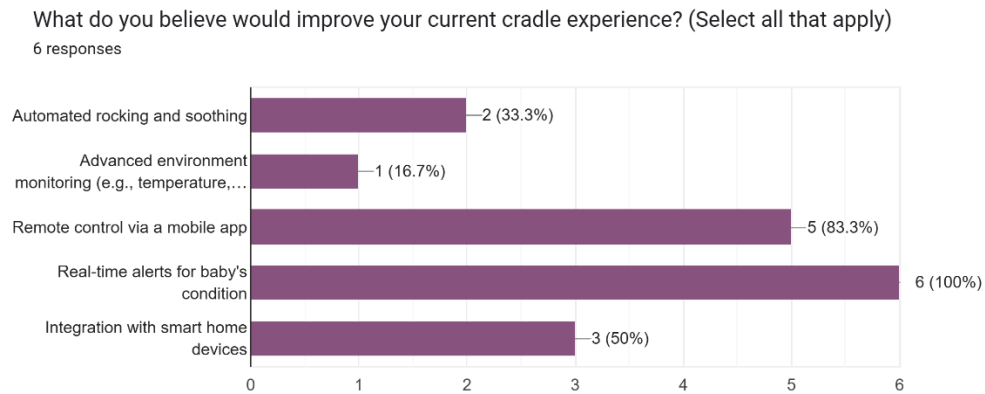


Figure 3.12: Features that Would Improve Semi-automated Cradle User Experience

3.3.3.2.3 Smart Cradle Users

Although there was 1 respondent under the Smart Cradle user section, however this is a faulty result as there was an issue with the form section links during the early stages of the form distribution.

3.3.3.2.4 Respondents Who Does Not Use Cradle

According to figure 3.6, the majority of the respondent recorded that they do not use a cradle for their baby. Out of the 12 respondents, 10 (83.3%) respondents agree that one of the challenges in soothing their baby without a cradle is it can be physically exhausting and takes a lot of time and effort. The main method recorded that the respondents use to soothe their baby is by holding and rocking them manually, aside from pacing while holding the baby and playing soothing music. This result can be seen in Figure 3.13 below.

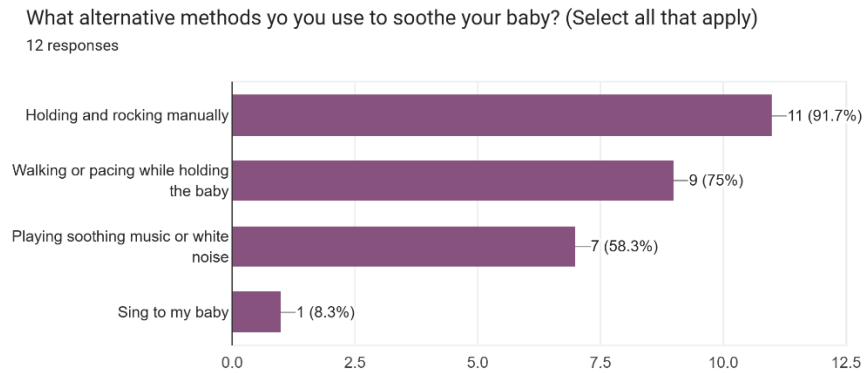


Figure 3.13: Alternative Methods Used to Soothe Baby without Cradle

This survey also shows that 6 (50%) of the respondents refused to use a cradle due to safety concerns, while the remaining 6 (50%) of the respondents refused to use a cradle mostly due to cost concerns and limited space at home. 6 (50%) of the respondents also recorded that they have considered using a cradle but ultimately decided against it. Figure 3.14 below shows the features that would persuade the respondents to use a cradle.

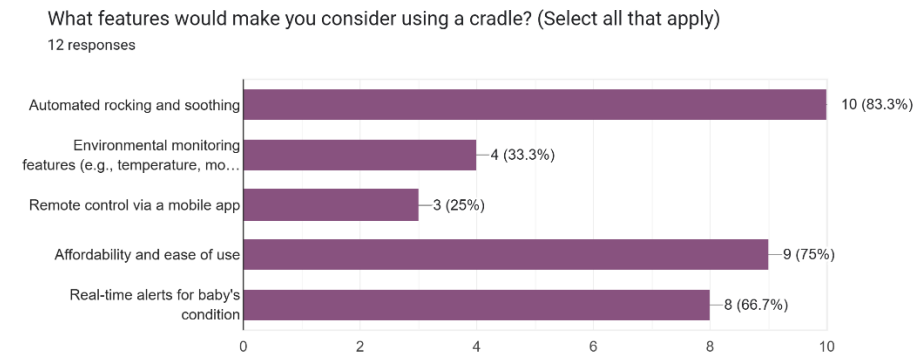


Figure 3.14: Features Considered by Non-Cradle Users to Use a Cradle

3.3.3.3 Technology Awareness and Preferences

In terms of technology preferences, 24 (80%) of the respondents documented that they feel comfortable in using mobile apps for baby monitoring, and all 30 (100%) of the respondents prefer to receive push notifications on their phone when using baby-monitoring devices. The three main concerns documented by the respondents in regard to using smart devices for infant care are cost, complexity and synchronisation issues. This can be observed in Figure 3.15 below.

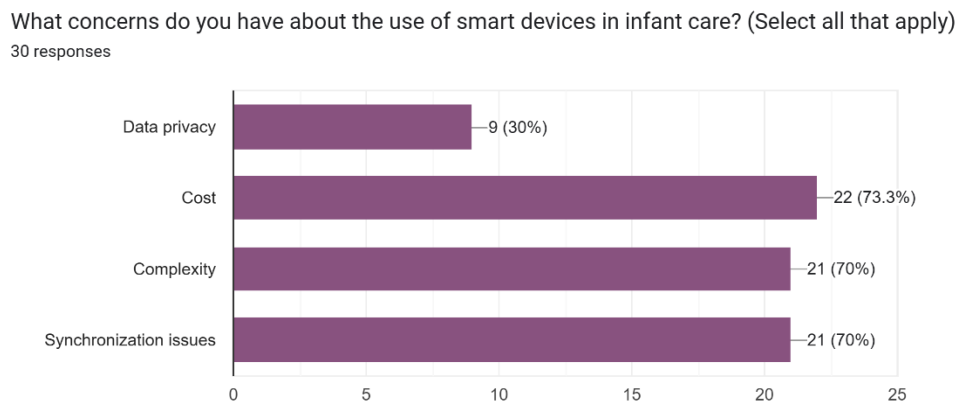


Figure 3.15: Respondents' Concerns on Using Smart Devices for Infant Care

3.3.3.4 Feature Preferences

A list of features was listed and respondents had to rank the importance of each feature from very important to not important. The findings are as follows:

- Cry/Sound detections: 29 (96.7%) agreed it is very important.
- Motion detections: 26 (86.7%) agreed it is very important.
- Temperature and humidity monitoring: 16 (53.3%) agreed it is important, while 12 (40%) respondents have neutral opinions on this feature.
- Automated rocking mechanism: 26 (86.6%) agreed it is very important.

- Wet diaper detection: 28 (96.7%) agreed it is very important.
- Lullaby/White noise playback: 23 (76.7%) agreed it is very important.
- Remote control via mobile app: 28 (93.3%) agreed it is very important.
- Camera for live monitoring: 29 (96.7%) agreed it is very important.
- Real-time alerts via mobile application: 24 (80%) agreed it is very important, while 6 (20%) has neutral opinions on this feature.

3.3.3.5 Budget and Purchase Intent

This section of the questionnaire was conducted to identify the respondent's preferred cost for a smart cradle as this will affect whether or not users will end up using one. In terms of budget, 13 (43.3%) respondents are willing to pay between RM1000 to RM1500 for a smart cradle, 11 (36.7%) respondents are willing to pay between RM500 to RM1000, while the remaining 6 (16.7%) is willing to pay below RM500 for a smart cradle. This result is shown in Figure 3.16 below.

What is the maximum amount you are willing to spend on a smart cradle?
30 responses

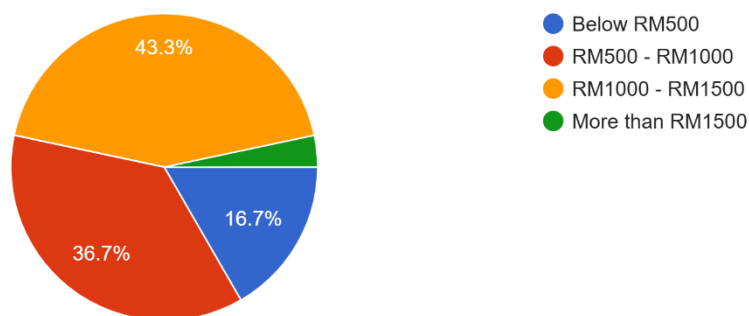


Figure 3.16: Maximum Amount Respondents are Willing to Pay for a Smart Cradle

Figure 3.17 below shows that the main factor that influences respondents to purchase a smart cradle is the features of the cradle, in which obtained 14 (46.7%) votes from out respondents, 10 (33.3%) of them prioritises the product operational lifespan, 5 (16.7%) respondents prioritised cost, while the remaining 1 (3.3%) respondent documented that all of the above answers played a factor in their decision to buy a smart cradle.

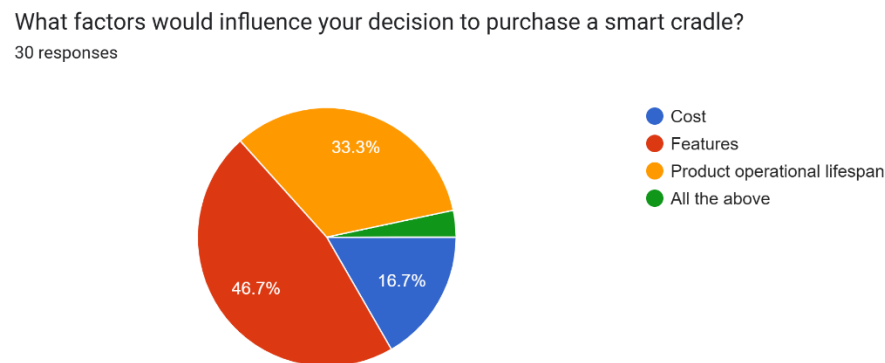


Figure 3.17: Factors Affecting Respondents' Purchase of a Smart Cradle

3.3.3.6 Feedback and Current Suggestions

This section of the questionnaire was to identify if there were any additional feedback from the respondents in regard to the smart cradle. Respondents have documented that a specific concern about using smart cradles is its safety and comfort towards their child and the reliability of the smart cradle to actually assist in automating aspects of infant care.

They also included additional suggestions that were not mentioned as an answer choice in the questionnaire. The main suggestion was to make the cradle travel friendly or include flexibility so that their baby can use it for a longer time, instead of just during the infant phase.

3.3.4 Software Requirements

The software requirements for the proposed Smart Cradle system, CradleSense, are detailed in this subsection, as summarised in Table 3.2.

Table 3.2: Software Requirements for CradleSense

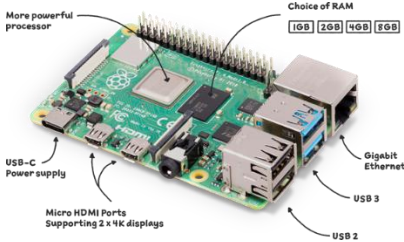
Software	Descriptions
Raspbian OS	The official operating system for the Raspberry Pi to provide the environment for executing Python scripts and managing sensors.
Android Studio	An integrated development environment (IDE) used to develop the mobile application for CradleSense. It provides tools for designing, testing, and deploying the app.
Visual Studio Code	A source-code editor used for managing Python scripts and any additional app-related configurations.
Firebase	Firebase is used as a cloud-based backend platform to log and store sensor data, such as temperature and humidity readings. It enables real-time data synchronisation between the IoT device and the mobile application.

Figma	A UI/UX design tool for creating the interface mock-up of the mobile application, also known as the wireframe, and to visualise the user flow.
-------	--

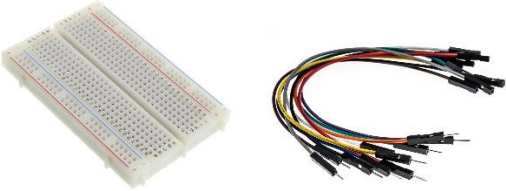
3.3.5 Hardware Requirements

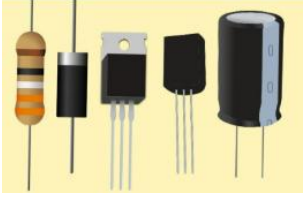
The hardware requirements for the proposed Smart Cradle system are specified below, as shown in Table 3.3. These components are essential for the functionality of the cradle, including sensor data collection, actuating the cradle, and ensuring effective monitoring and control.

Table 3.3: Hardware Requirements for CradleSense

Hardware	Description
Raspberry Pi 4 Model B	The main controller for the system. It will handle all processing tasks such as reading sensor data, running the system’s logic and sending control commands. 

Raspberry Pi Camera Module	A camera module to monitor the baby's surrounding environment, observe motion, and enable live video streaming for real-time monitoring. 
LM35 sensor	An analog temperature sensor used to monitor the baby's surrounding environment to ensure optimal comfort. 
PIR Motion Sensor	A passive infrared sensor to detect movement or the baby's position in the cradle. It triggers responses such as the rocking mechanism. 

Servo Motor	Used to control the rocking mechanism of the cradle, providing automatic soothing based on the triggered parameters. 
Speaker	For playing soothing sounds to calm the baby, such as lullabies or white noise. 
Breadboard and Jumper Wires	To prototype and connect sensors with the Raspberry Pi. This allows for easy wiring of the components before the final build. 
Resistors and Capacitors	For setting up proper sensor connections and to ensure stable signal processing for components such as the LM35 sensor.

	
11th Gen Intel(R) Core(TM) i7-1195G7 CPU	The system and mobile app will be developed using the MSI Modern 15 laptop with the mentioned processing power. This processor ensures smooth operation for the development tools such as Android Studio, Flutter, and the Raspberry Pi OS.
8GB of RAM	The laptop provides sufficient memory for seamless multitasking, therefore enabling the efficient execution of multiple development tools like IDEs and debugging environments simultaneously.
512GB SSD	The laptop provides a sufficient amount of storage to facilitate the efficient handling of project files, datasets, and software tools required during the development and testing phases.

3.4 User Design

Once the requirements for the Smart Cradle system were established, the User Design phase can begin. This phase focuses on developing prototypes and mock-ups for the system's user interface (UI) and core functionalities. The prototypes are designed to demonstrate essential features and emphasizing functionality over visual refinement to ensure that the system aligns with the identified user needs.

This phase aims to provide a clear visual and functional representation of the system, including wireframes for the mobile app, workflows for automated responses, and data flow processes. Diagrams such as flowcharts, Data Flow Diagrams (DFDs), and Entity Relationship Diagrams (ERDs) outlined system architecture and data management. Prototypes were refined through repeated testing to address any issues and ensure usability. The next development stage proceeded only after user approval, ensuring that the system met the expectations and provided an intuitive solution for infant monitoring and soothing.

3.4.1 Overall System Architecture Design

System architecture refers to the high-level design that outlines how various components of a system are structured and interact with each other to achieve the desired functionalities. It acts as a conceptual blueprint which guides the implementation and integration of the system's modules while ensuring a clear understanding of its operational workflow. System architecture also demonstrates how the system interfaces with external systems, detailing the interconnection of its components, and facilitates a coherent development process.

Figure 3.18 below demonstrates the general system architecture of the proposed smart cradle system. In this proposed system, the system architecture is designed to detect and respond

to a baby's needs through automated processes. The architecture integrates key components such as, but not limited to, sensors, a Raspberry Pi, a DC motor, and a mobile application, which ensures a seamless interaction and real-time response. The sensors, for instance as shown in Figure 3.17, audio detection and the camera module will detect the baby's cries and movements. The collected sensor data will then be transmitted to the Raspberry Pi for processing and sent to Google Firebase, a cloud platform, for storage and analysis. These sensor data will also send a trigger notification on the parent's mobile app, in which mothers will be able to monitor their baby in real-time and perform remote control when needed. Simultaneously, the system triggers the DC motor to rock the cradle, soothing the baby automatically.

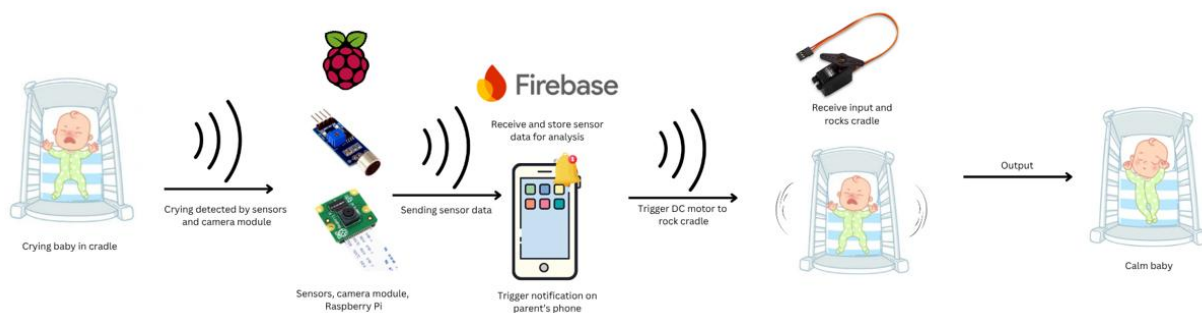


Figure 3.18: Overall System Architecture Design

The detected signals, for example, the baby's cry, motion, or environmental conditions, will be processed by the Raspberry Pi, which analyses the input data to determine the appropriate soothing response. The generated response will include triggering the cradle's rocking mechanism, playing calming music or white noise, and sending real-time notifications to the parent's mobile application. Notifications include details such as the detected event, timestamp, environmental conditions, and a short description of the response initiated by the system.

The proposed system operates in a sequence of steps, starting from detecting the baby's condition to executing the appropriate soothing actions:

1. Detect baby's cries using an audio sensor.
2. Analyse movement through motion detection sensors.
3. Monitor surrounding environmental conditions such as the temperature and humidity.
4. Activate the rocking cradle and or autoplay lullabies or soothing sounds if the baby is crying or restless.
5. Notify parents via the mobile application with the event details and current status.
6. Parents manually control cradle actions via mobile app if necessary.

The system begins with the audio sensor capturing sound signals to detect the baby's cry. If no crying is detected, the system continues monitoring without triggering any response. Simultaneously, the motion sensor detects excessive baby movements that might indicate restlessness. If either condition is met, the Raspberry Pi processes the data and sends an instruction to the DC motor to rock the cradle gently. Environmental sensors continuously monitor the cradle's temperature and humidity levels, ensuring a comfortable environment for the baby.

At the same time, the system uploads the sensor data to Firebase for analysis and data logging, providing parents with insights into their baby's behaviour and environmental conditions over time. Notifications are sent to the parent's mobile application, allowing them to monitor the cradle's activities and manually adjust settings if necessary. The mobile application also provides access to real-time data and historical records, enhancing user control and engagement. This automated process ensures a seamless and efficient response to the baby's needs, reducing the parents' burden while prioritising the baby's comfort and safety.

3.4.2 Flowchart Diagram

Figure 3.19 shows the general flow of the proposed system from the beginning to the end of system process.

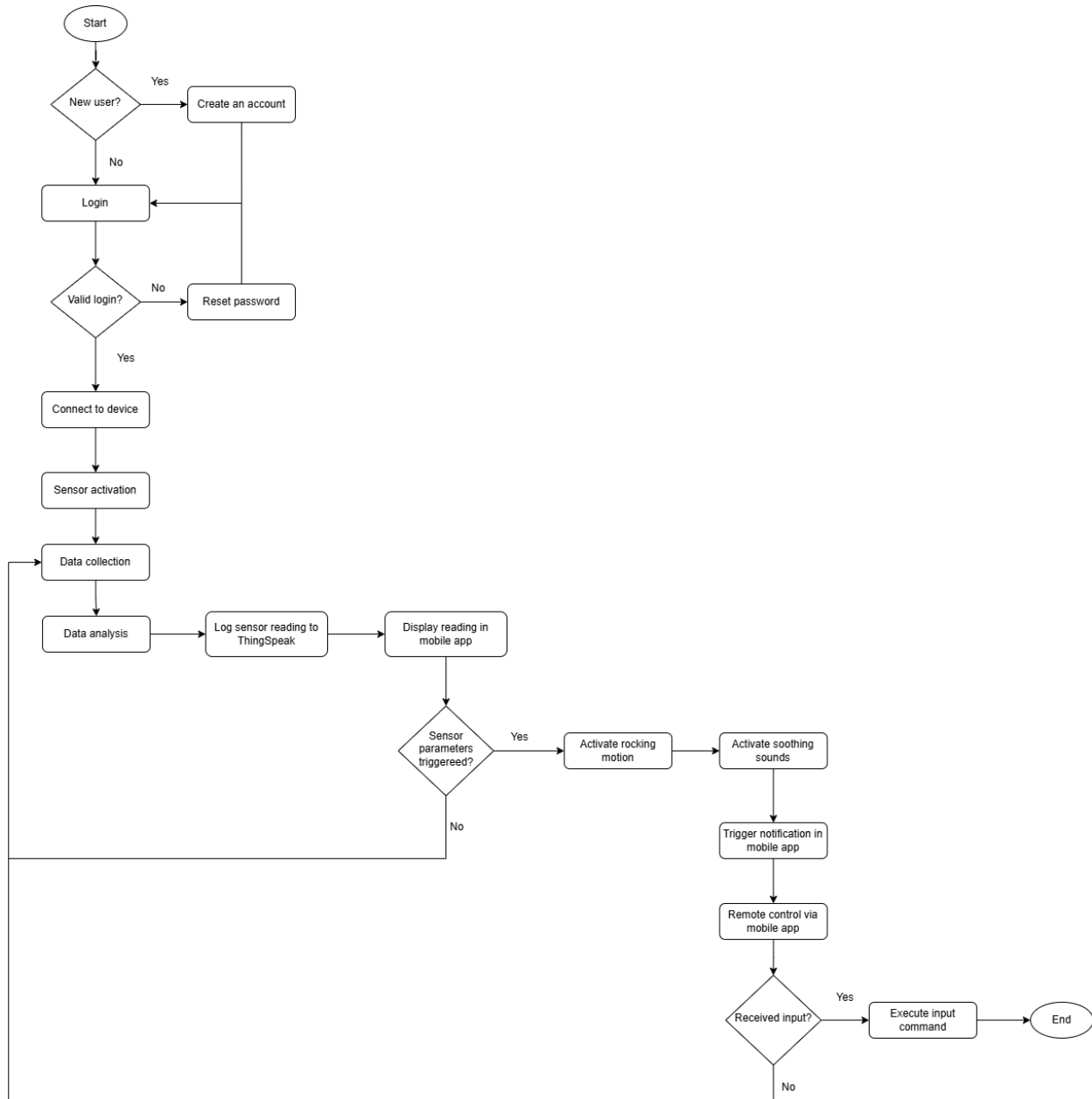


Figure 3.19: Flowchart of CradleSense

The processes in the flowchart of the proposed system are explained in detail as follows:

1. Create an account

If the user is new to the system, they will need to create an account. Otherwise, they can proceed to log in using their existing credentials.

2. Login validity and Reset Password

If the login is invalid, the user will be prompted to reset their password and log in again. Once logged in successfully, they can proceed to the next step.

3. Connect to the Device

The user's phone will establish a connection with the Smart Cradle device, enabling communication between the mobile app and the cradle.

4. Sensor Activation

After the connection is established, the system will activate the sensors in the Smart Cradle to begin monitoring the baby's environment. Sensors include temperature and humidity sensor, motion sensor, sound sensor, along with the live video feed.

5. Data Collection

The sensors will collect relevant data such as temperature, motion, and other surrounding environmental factors. The collected data will then be analysed to assess the baby's condition and determine if soothing actions are required.

6. Log Sensor Reading to Firebase

The system will log the sensor readings to Firebase for further analysis and monitoring.

7. Display Reading in Mobile App

The sensor readings will be displayed in real-time on the mobile app for the user to monitor the baby's condition.

8. Sensor Parameters

If none of the predefined sensor parameters such as abnormal temperature, crying sound, or excessive baby movements are triggered, the system will return to data collection. If a trigger is detected, the system will proceed to the next step.

9. Activate Soothing Mechanisms

If a sensor triggers such as baby crying detected or baby restlessness detected, the rocking mechanism of the Smart Cradle will be activated to soothe the baby. In addition to the rocking mechanism, soothing sounds will be activated to calm the baby. Soothing sounds include, but is not limited to, lullabies and white noise.

10. Trigger Notification in Mobile App

The system will send a notification to the parent's mobile app to alert them about the baby's current condition and the actions being taken.

11. Remote Control via Mobile App

The user can remotely control the Smart Cradle through the mobile app, allowing them to adjust settings such as rocking speed, sound preferences, or even the rocking duration.

12. Execute Input Command

If no input is received from the mobile app, the system will return to data collection.

Otherwise, the system will execute the input command from the user, such as adjusting the cradle's rocking mechanism or changing the soothing sound.

3.4.3 Context Diagram

The context diagram, also known as a level 0 DFD, represents the highest level in a data flow diagram. It includes a single process that represents the entire Smart Cradle system, defining its context and boundaries. Creating a context diagram is the first step in developing a series of Data Flow Diagrams (DFDs). This diagram helps identify the information flows between the system and external entities. It provides a clear and simple illustration of the relationships between the system and its environment.

The context diagram below shows the data flow between the proposed Smart Cradle system and the external entities. The central process represents the Smart Cradle system, while parents and caregivers are the external entities that interact with the system. The data flows connect these entities to the central process, illustrating how information moves between them. This can be seen in Figure 3.20 below.

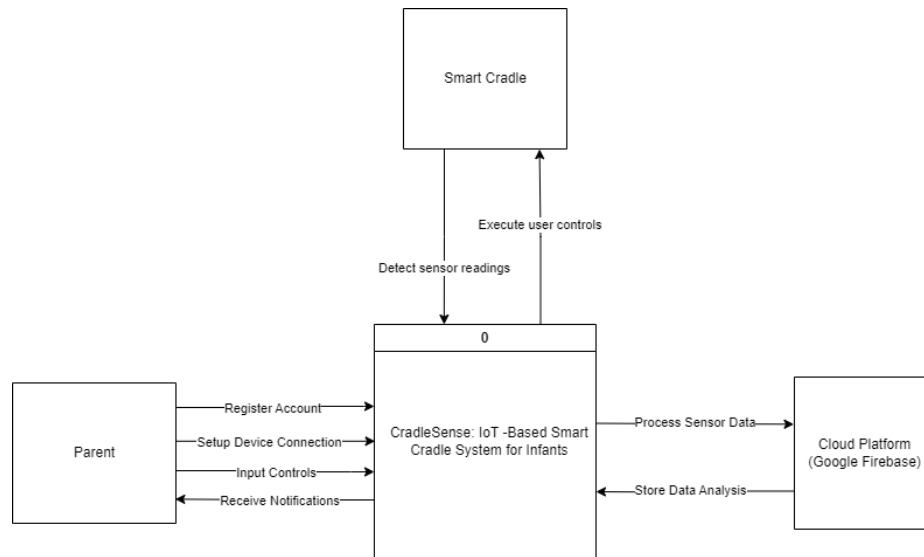


Figure 3.20: Context Diagram (Data Flow Diagram Level 0) for CradleSense

There are 4 data flow movements between the system and parents:

- Parents register account to the system.
- Parents set up device connection with the system.
- Parents input cradle controls to the system via mobile app.
- Parents receive notification from the system.

There are 2 data flow between the system and the smart cradle:

- Smart cradle detects sensor readings and sends it to the system.
- Smart cradle executes the cradle controls input by the user into the system.

There are 2 data flow movements between the system and cloud platform:

- System sends the collected sensor data from the smart cradle to the cloud platform for processing.
- System stores the data analysis performed by the cloud platform.

3.4.4 Data Flow Diagram Level 1

The Data Flow Diagram (DFD) Level 1 offers a more comprehensive view of the system's core functions compared to a context diagram. In a Level 1 DFD, the single process node from the context diagram is decomposed into sub-processes. As these sub-processes are introduced, new data flows and data stores are required to connect them. This increased level of detail allows for a deeper understanding of the system's operations and the flow of information. Figure 3.21 below illustrates the Level 1 DFD of the proposed system, which includes six processes, all of which are Register User, Login User, Sensor Data Collection, Mobile App Controls, Control Cradle Functions, and Trigger Notifications. Additionally, there are two data stores, which are Account Details and Sensor Data. The external actors in the system are parents, cloud platform, and the smart cradle itself.

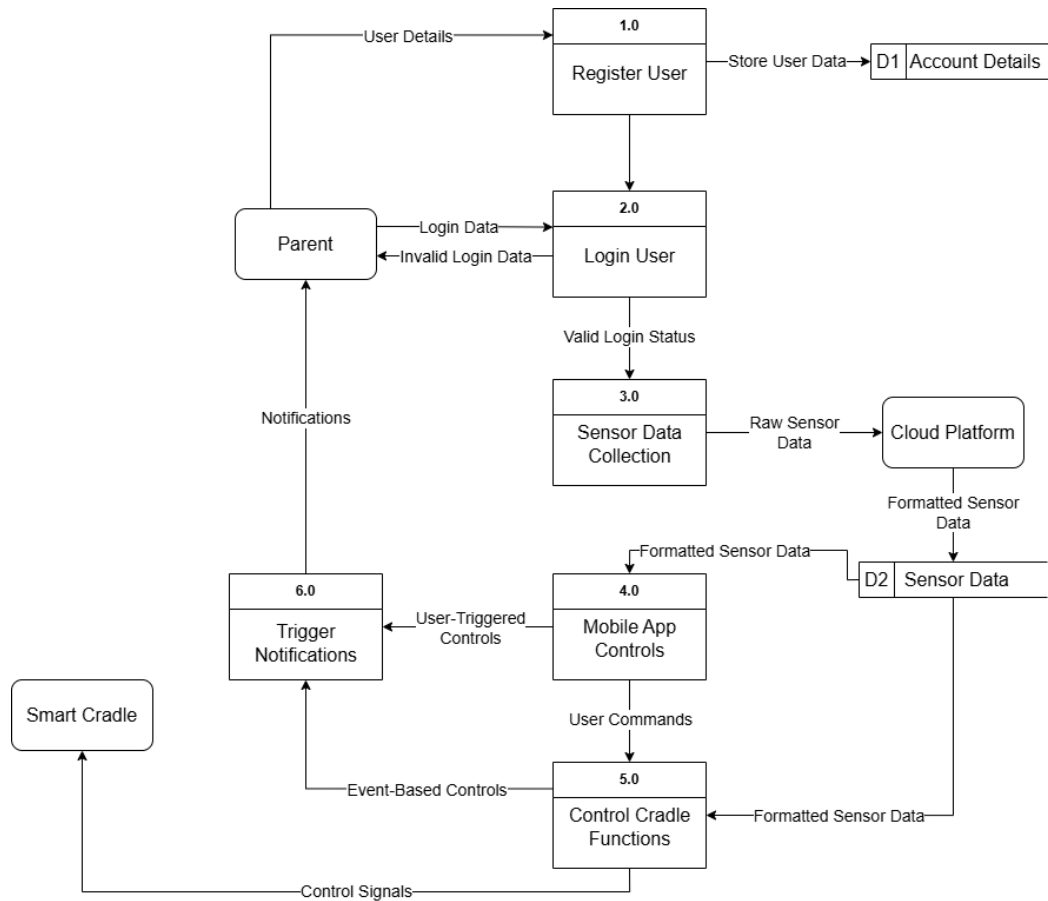


Figure 3.21: Data Flow Diagram Level 1 for CradleSense

The processes are explained in detail as follows:

Process 1.0: Register User

To begin using the Smart Cradle system, parents must create an account by providing their details, including email and password. The system collects this data and formats it before storing it in the Account Details database. If any invalid details, such as an incorrect email format, are detected, the system prompts the parent to re-enter the information. Once the system validates the information, the parent can proceed to log in to the system.

Process 2.0: Login User

After registering, parents must log in by entering their email and password. The system checks the login details against the data stored in the Account Details database. If the credentials match, the parent gains access to the system. If the credentials are incorrect, the parent is asked to re-enter the login details.

Process 3.0: Sensor Data Collection

Once logged in, the Smart Cradle begins to collect sensor data, such as temperature, motion, and audio. This data is gathered continuously to monitor the baby's status. The collected sensor data is then formatted and stored in the Sensor Data database for further analysis. This ensures that the system can track the baby's condition in real-time.

Process 4.0: Mobile App Controls

The parent can control the Smart Cradle through the mobile app. This includes features like adjusting the rocking mechanism, activating soothing sounds, or turning on the camera. The app sends commands to the Smart Cradle, which are processed and executed.

Process 5.0: Control Cradle Functions

The system allows the parent to control the cradle's functions, such as rocking or playing soothing sounds. The mobile app sends control signals to the Smart Cradle, which processes them and adjusts its functions accordingly. These actions are monitored in real-time, ensuring that the system responds accurately to the parent's commands.

Process 6.0: Trigger Notifications

The system triggers notifications to alert the parent if any unusual behaviour is detected, such as the baby crying or the cradle having excessive movements for long periods. These notifications

are sent via the mobile app. The notifications ensure that parents are promptly informed about their baby's status and can take necessary actions.

3.4.5 Data Flow Diagram Level 2

The Data Flow Diagram Level 2 (DFD Level 2) further decomposes processes from the Level 1 DFD into more detailed sub-processes. Extending DFD Levels beyond Level 3 is rarely done as Level 3 DFDs typically provide enough detail for analysis. The goal of a Level 2 DFD is to break down processes from the Level 1 DFD into specific tasks, with each process focusing on a single function. It's important to note that not all processes in the Level 1 DFD require further decomposition. Only those that are complex and require a deeper understanding are broken down into Level 2 DFDs.

3.4.5.1 DFD Level 2 for Process 1.0

Figure 3.22 illustrates the breakdown of Process 1, which is responsible for handling new user registration. It shows that the process begins with the user providing their details. These details are then processed in two steps. First, the system checks the validity of the email and password. If the details are invalid, an error message is triggered. However, if the details are valid, the system proceeds to the next step where it confirms the account creation and generates formatted user details. These details are then stored in the Account Details data store, completing the user registration process.

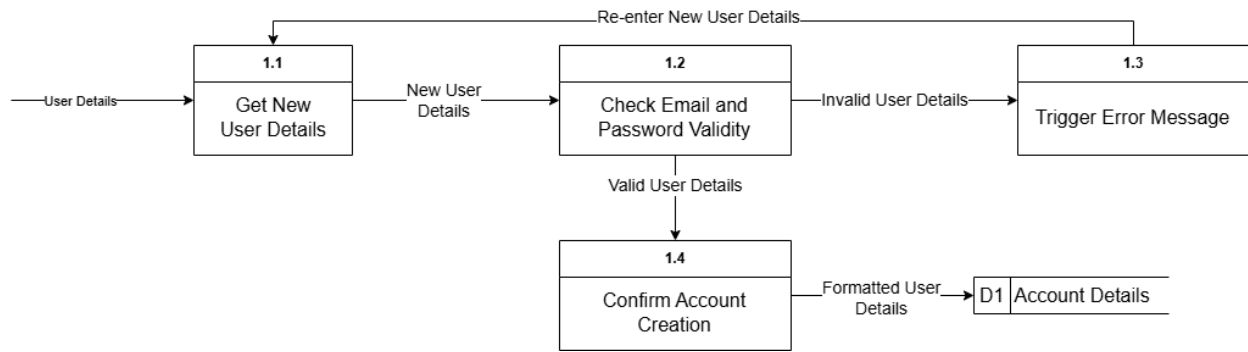


Figure 3.22: Data Flow Diagram Level 2 for Process 1.0 of CradleSense

3.4.5.2 DFD Level 2 for Process 2.0

Figure 3.23 illustrates the breakdown of Process 2, which is responsible for handling user login. It begins with the user entering their login details, in this case is their email and password. The system then verifies these details against the stored account information. If the login details are invalid, an error message is triggered. However, if the details are valid, the user is granted access to the system, and the login status is updated.

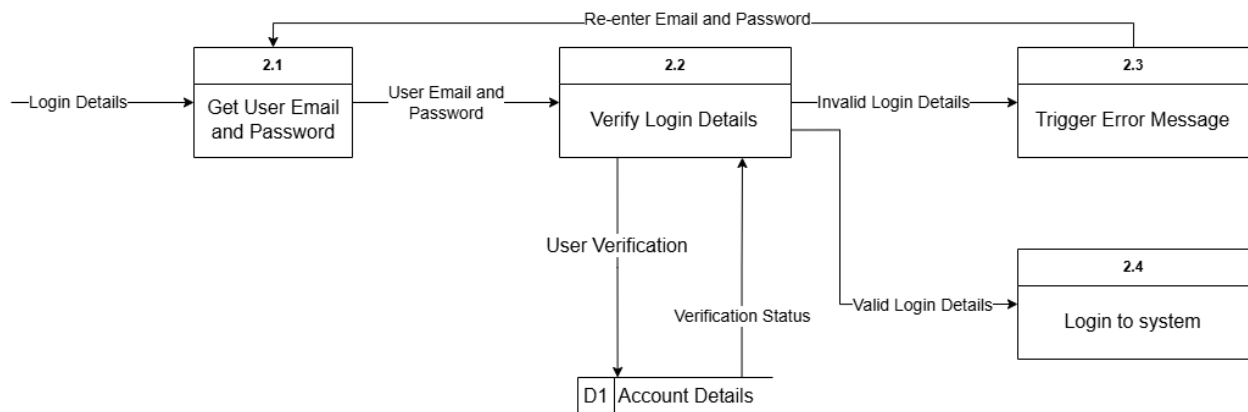


Figure 3.23: Data Flow Diagram Level 2 for Process 2.0 of CradleSense

3.4.5.3 DFD Level 2 for Process 3.0

Figure 3.24 breaks down the sensor data handling process into four key stages. Initially, raw sensor data is read and collected. This data then undergoes processing, to ensure its accuracy and consistency. The processed data is subsequently stored in a designated data store for future reference and analysis. Finally, the stored sensor data is subjected to in-depth analysis using various techniques such as statistical analysis, machine learning, and data visualisation. This analysis aims to extract valuable insights, identify patterns, and make informed decisions based on the sensor data.

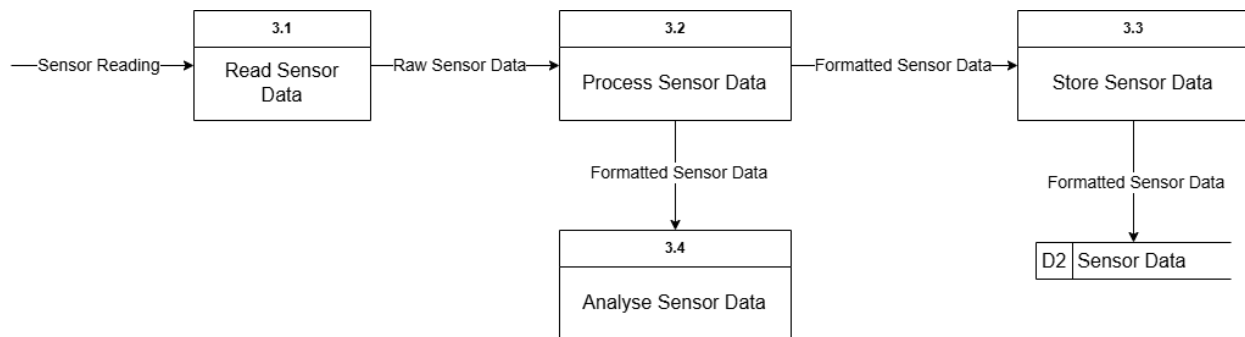


Figure 3.24: Data Flow Diagram Level 2 for Process 3.0 of CradleSense

3.4.5.4 DFD Level 2 for Process 4.0

The DFD Level 2 diagram is illustrated in Figure 3.35, which describes the breakdown of Process 4, which is responsible for handling user commands. It starts with the system receiving user commands. These commands are then validated to ensure they are valid and executable. If the commands are invalid, an error message is triggered. However, if the commands are valid, the system accepts and executes them. Finally, the status of the executed commands is reported, indicating whether the commands were completed successfully.

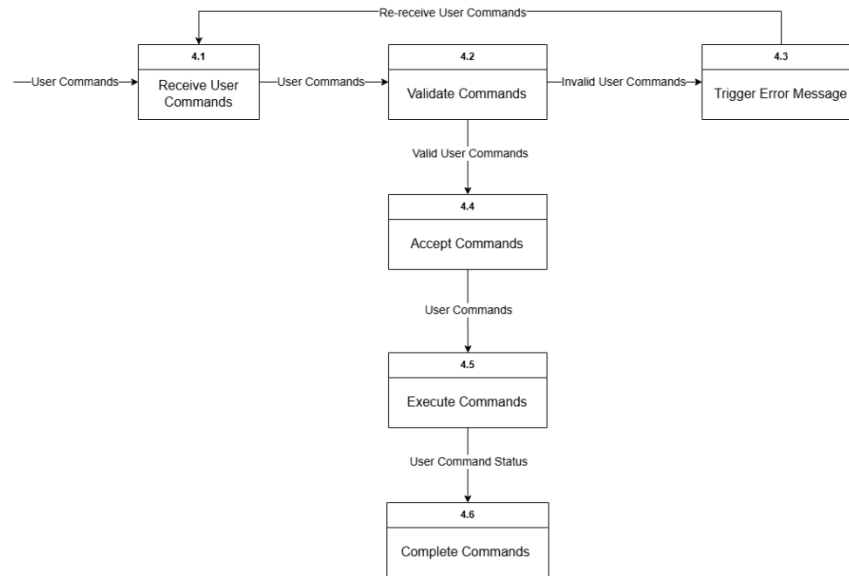


Figure 3.25: Data Flow Diagram Level 2 for Process 4.0 of CradleSense

3.4.5.5 DFD Level 2 for Process 5.0

Figure 3.26 below provides a detailed view of the cradle control and monitoring system. It begins with user commands being received and interpreted into control signals. These control signals are then used to execute specific actions within the cradle system. Simultaneously, the system continuously monitors the cradle's state by utilizing sensor data stored in the Sensor Data store. This monitoring process allows for real-time tracking of the cradle's status, which is then used to generate feedback data. This feedback data can be used to inform users about the cradle's current state, adjust control signals, or trigger alarms if necessary.

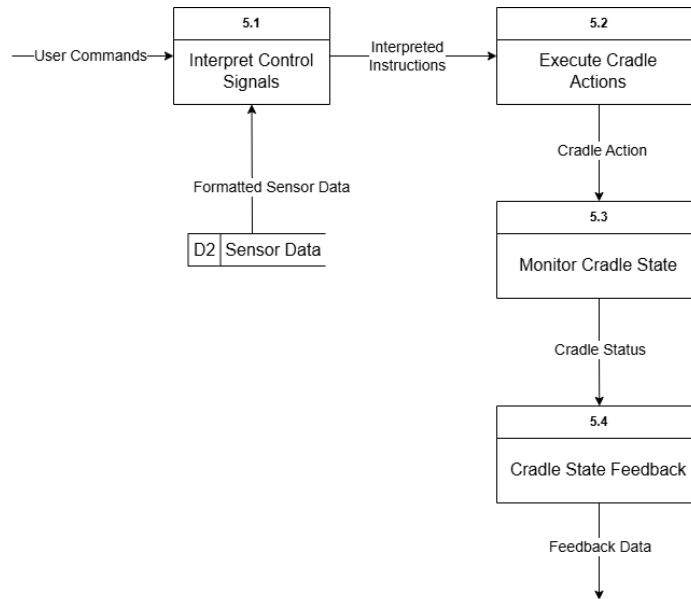


Figure 3.26: Data Flow Diagram Level 2 for Process 5.0 of CradleSense

3.4.5.6 DFD Level 2 for Process 6.0

The DFD Level 2 diagram for Process 6, as shown in Figure 3.27 below, delves into the notification system for the cradle. It starts by receiving feedback data from the cradle system. This data is continuously monitored for any events that necessitate a notification. These events, or triggers, could include things like abnormal sensor readings, system errors, or user-defined thresholds being crossed. Once a trigger is identified, the system generates the appropriate notification content. This content might include details about the event, its severity, and any relevant timestamps. Finally, the generated notification content is delivered to the designated recipients, in this case would be the mothers. This notification mechanism ensures that critical events are promptly communicated and addressed, allowing for timely intervention and maintenance of the cradle system.

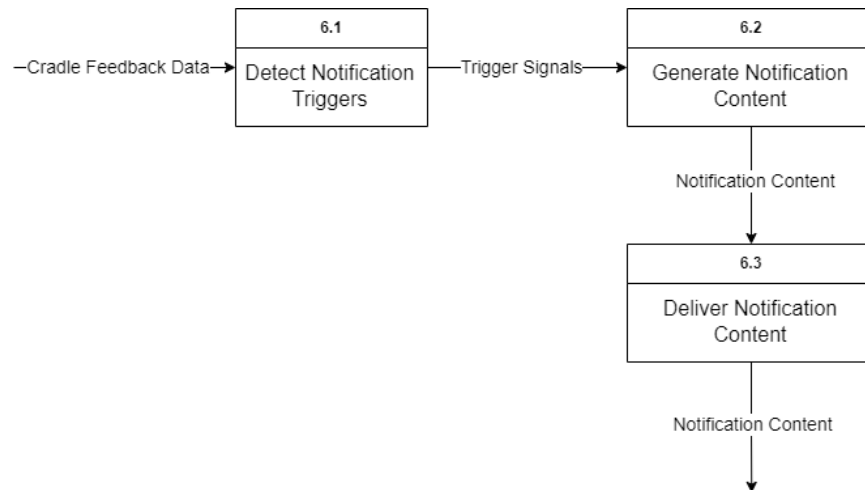


Figure 3.27: Data Flow Diagram Level 2 for Process 6.0 of CradleSense

3.4.6 Entity Relationship Diagram

Entity Relationship Diagram (ERD) is a diagram that is commonly used in database design as a visual representation of the various entities within a system and their interrelationships. Figure 3.28 below shows the ERD for CradleSense to illustrate the relationship between the entities of the proposed system using Crow's Foot notation. This ERD illustrates a database schema for a system that monitors smart cradles. It consists of three main entities, which are Parent, SmartCradle, and SensorData. The Parent entity stores information about the users of the system, including their name, email, password, and contact number. The SmartCradle entity stores information about the smart cradles, such as their ID, status, and model. The SensorData entity stores sensor readings collected from the smart cradles, including temperature, humidity, motion status, and timestamps.

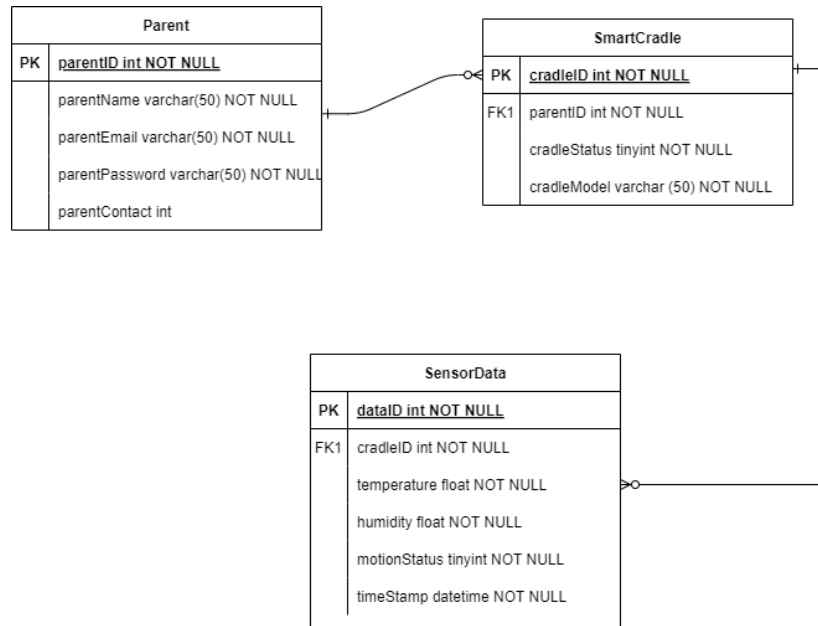


Figure 3.28: Entity Relationship Diagram for CradleSense

3.4.7 Wireframes

A wireframe is a preliminary graphical representation of a proposed application. It serves as a prototype to gather feedback and insights from target users before finalizing the designs. Figure 3.29 shows three separate screens, which are the Splash Screen, Login Page and Login Error Page. The splash screen serves as the initial point of contact for users, setting the stage for the CradleSense application. The prominent visual of a baby mobile instantly establishes a connection to the app's core function, which is for monitoring and safeguarding infants, creating a positive and reassuring visual association. The text "Safe and Sound with CradleSense" reinforces the app's value proposition, emphasizing the sense of security and well-being it aims to provide to parents. The clear "Get Started" call to action encourages users to proceed to the next step, either logging

in or creating an account. This streamlined approach ensures a smooth and welcoming user onboarding experience.

Upon clicking on the “Get Started” button, users will be redirected to the Login Page. The login page presents a clean and user-friendly interface for existing users to access the CradleSense app. The input fields for "Email Address" and "Password" are clearly labelled, making it intuitive for users to understand the information required. The "Login" button provides a clear and immediate action for users to initiate the authentication process. For new users, the inclusion of a “Create Account” button ensures easy access to the registration process.

For existing users trying to login, if the credentials input by the user is wrong, they will be redirected to the Login Error Page. The use of a red exclamation mark and the bold "Invalid Input!" message immediately alerts the user to the error. The accompanying text provides specific feedback about the error: "The email or password you entered is not found in the system." This detailed message helps users understand the reason for the login failure and assists them in correcting their input. The "Try again" button allows users to quickly attempt to log in again with the correct information.

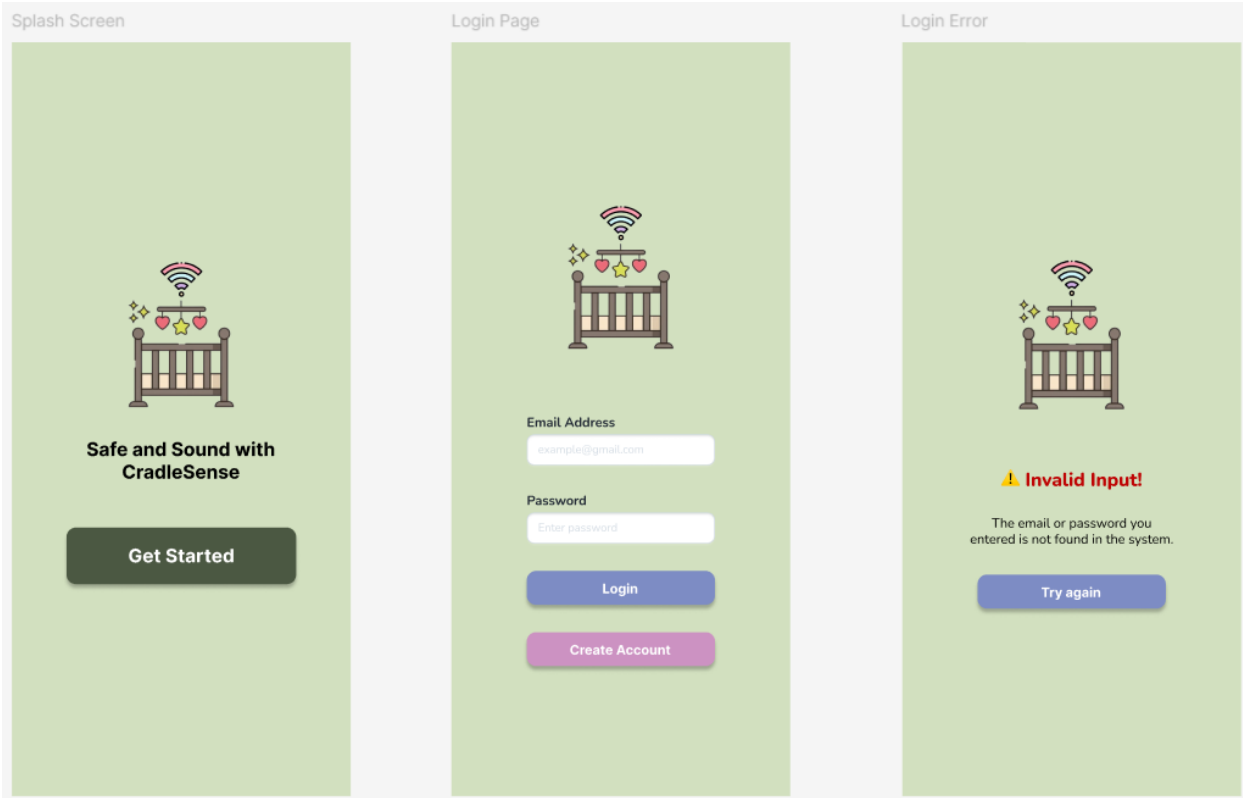


Figure 3.29: Wireframe for CradleSense App (1)

Figure 3.30 shows another three screens available in the application, which are the Create Account screen for new users, and the Registration Error screen for error handling of new user registrations, and the Home Page. The "Create Account" screen presents a user-friendly interface for registering new users in the CradleSense application. The screen is visually appealing, incorporating the baby mobile graphic that reinforces the app's brand identity. It includes clear input fields for essential user information such as Full Name, Email Address, Password, and Phone Number. The "Create" button allows users to initiate the account creation process. This screen is designed to be intuitive and easy to navigate, guiding users through the account registration process smoothly.

The “Registration Error” screen displays an error message when the information provided during the account creation process is invalid. A red exclamation mark and the bold "Invalid Input!" message immediately alert the user to the issue. The accompanying text provides a specific and informative error message: "The email or password you entered already exist in the system!" This clear and concise message helps users understand the reason for the error and allows them to quickly identify and correct the issue. The "Try Again" button provides a convenient way for users to re-enter their information and attempt to create an account again, minimizing frustration and improving the overall user experience.

The Home Page serves as the central hub of the CradleSense application. The screen prominently displays real-time information about the cradle, including its status, temperature, humidity, and sound decibel. Actionable buttons for key functionalities like View Live Video Feed, Audio Playback, Start Rocking Cradle, and Alerts are readily accessible. A persistent bottom navigation bar provides easy access for users to switch back and forth between sections. This design ensures a user-friendly and informative experience for parents.

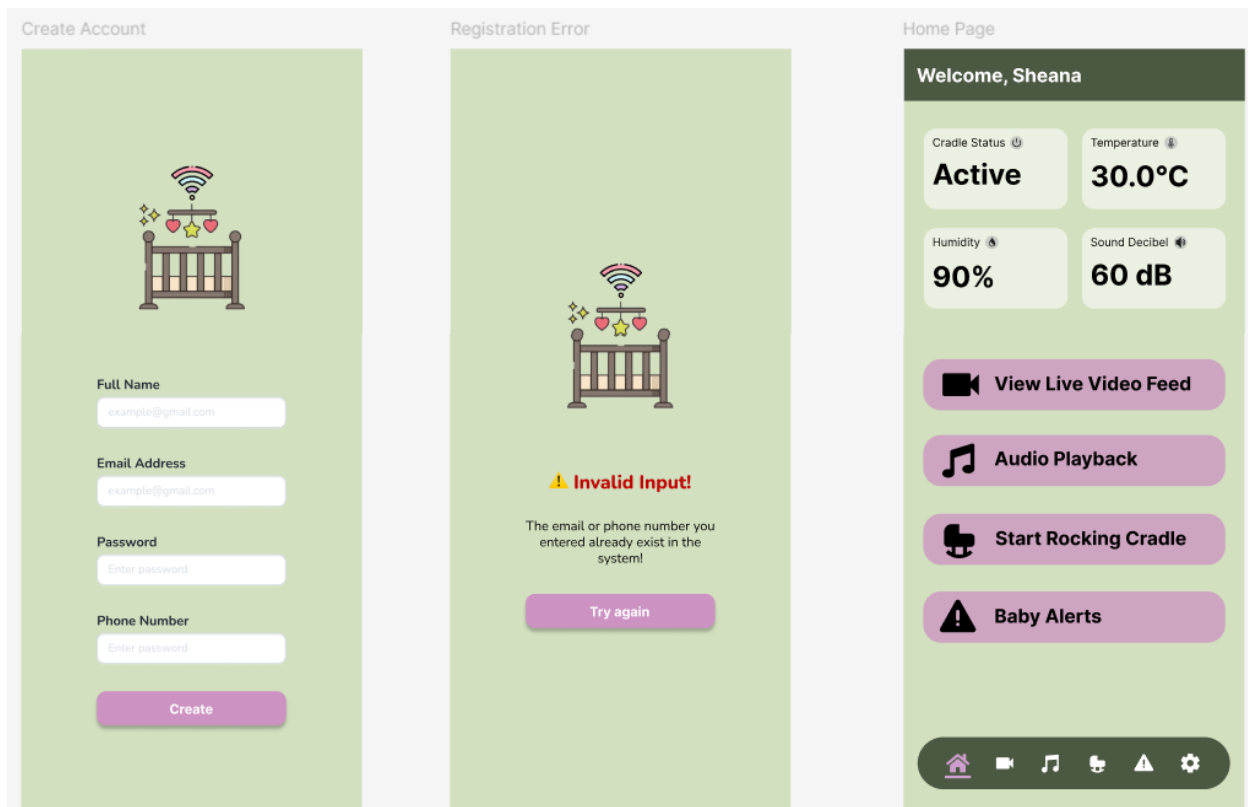


Figure 3.30: Wireframe for CradleSense App (2)

Next, Figure 3.31 below shows the Live Video Feed Page, Audio Playback Page, and Start Rocking Cradle Page. The Live Video Feed screen provides parents with a real-time visual connection to their baby. It displays a live video stream from the cradle, allowing them to monitor their baby's movements and activities remotely. The device name and current time are displayed above the feed for reference.

The Audio Playback Page is designed to provide parents with soothing audio options when their baby is crying. The page features a list of available songs, allowing parents to choose from a selection of calming melodies. Song details such as duration are provided for reference. The page also includes controls like a timer to set the playback duration, a "Play Songs" button to initiate playback, an "Add" button for expanding the song library, and a "Delete" button to remove

selected songs from the song library. This well-organized interface provides parents with a convenient and user-friendly solution for soothing their crying baby.

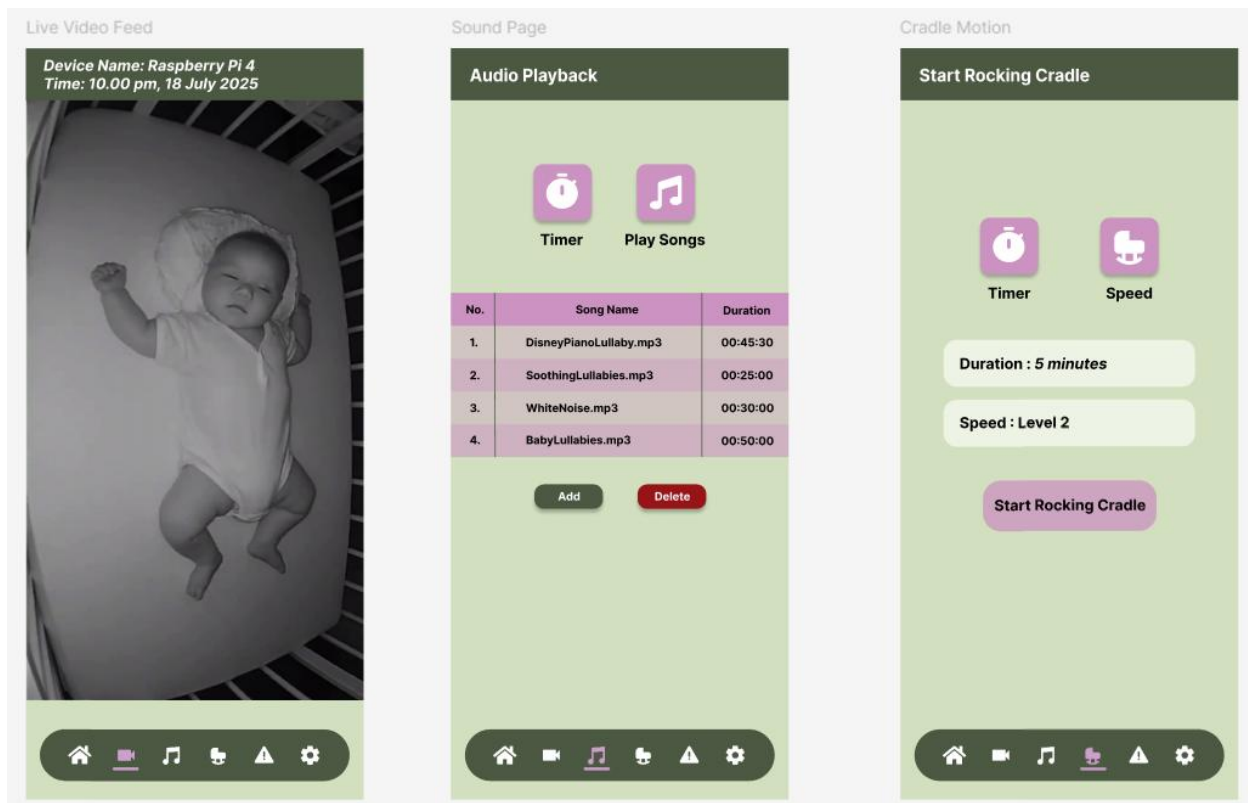


Figure 3.31: Wireframe for CradleSense App (3)

The Start Rocking Cradle screen offers a simple yet effective way to comfort the baby when they are crying. This page comes with two major features, which are the “Timer” button and the “Speed” button. Similar to the Audio Playback page, the “Timer” button allows users to set the duration of this soothing intervention to be executed. Once confirmed, the set timer duration will be displayed. The "Speed" button allows users to determine the speed of the cradle’s rocking motion. Similarly, once selected, the speed level will be displayed. In addition, the “Start Rocking Cradle” button allows parents to remotely activate the cradle's rocking motion, providing gentle rhythmic movement to comfort the baby.

Next, Figure 3.32 below shows the wireframe for the “Alerts” Page and the “Settings” Page. The Alert Page provides parents with a comprehensive view of recent events related to their baby's well-being. A detailed table presents a history of alerts or events, including the timestamp, date, a description of the event, and the action taken by the system in response. This information allows mothers to understand their baby's needs and provides valuable insights into their sleeping patterns and behaviour. The persistent bottom navigation bar ensures easy access to other sections of the app, allowing parents to navigate seamlessly between different functionalities.

The Settings screen provides users with access to account and device management options. It is organised with clear section headings, making it easy to navigate and find the information they need. The "Account" section allows users to view or edit their profile information, such as their email and password, while the “Devices” section enables them to manage their connected cradles. The “Logout” option provides a clear and convenient way for users to log out of the app. The bottom navigation bar, consistent with the Home Page, ensures easy navigation between different sections of the app.

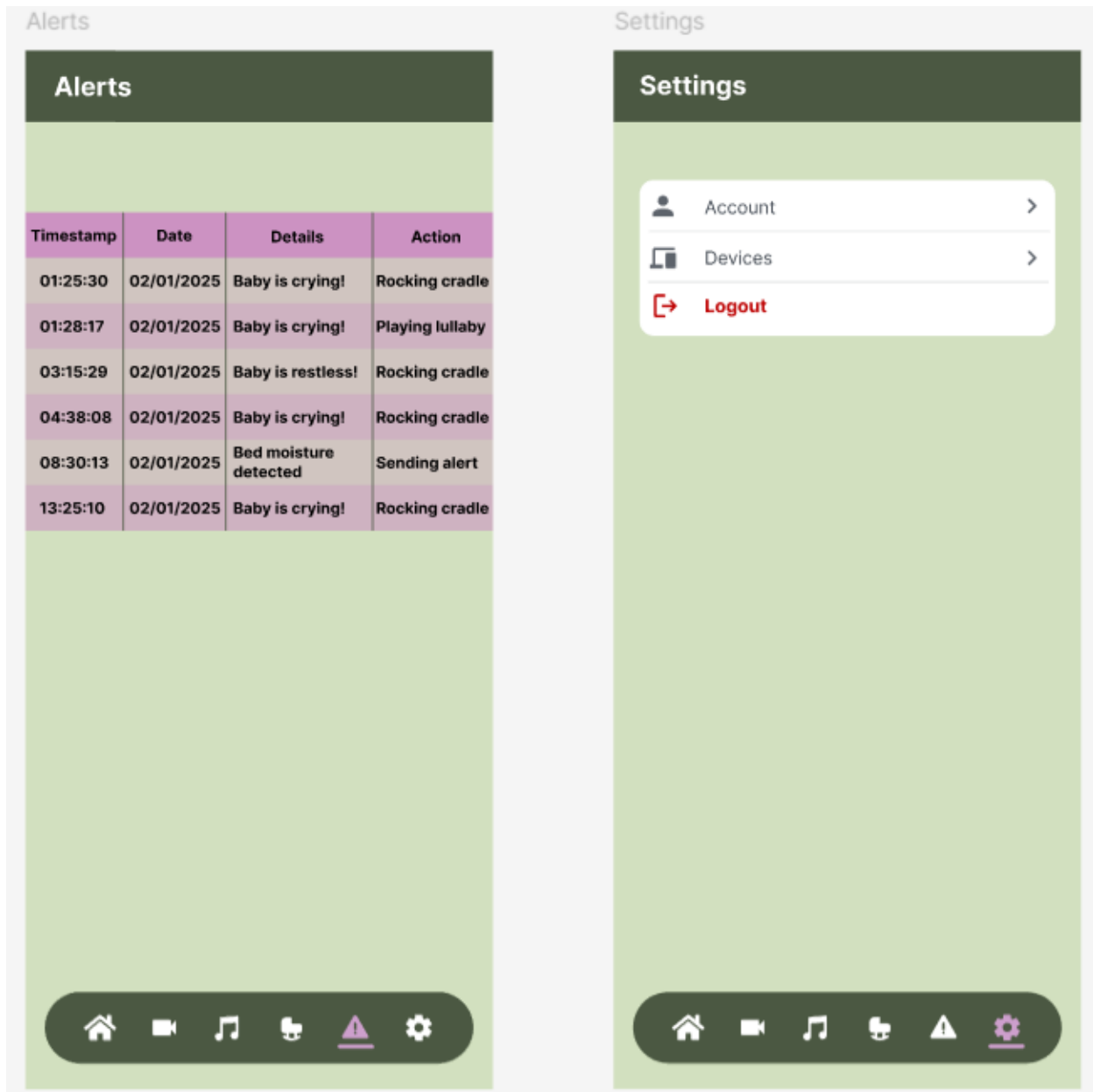


Figure 3.32: Wireframe for CradleSense App (4)

3.5 Construction

The construction phase of RAD is where the system's actual development and coding take place. This phase builds on the detailed prototypes and beta systems created earlier, refining them into a fully operational model. It is inherently iterative, allowing for the addition of new elements and

adjustments as needed. The iterative prototyping process is crucial in resolving user concerns and meeting client demands, making it possible to finalise a functional system more effectively than with traditional development approaches. During this phase, the software undergoes extensive testing to ensure it meets the outlined goals and client expectations. Developers collaborate closely with clients and end users, incorporating their feedback on both functionality and design to make ongoing improvements. Clients play an active role by providing insights, recommending changes, and suggesting creative solutions to tackle emerging issues. The emphasis remains on rapid iteration and development to deliver a working software version as quickly as possible.

3.5.1 System Development

During this development stage of CradleSense, the prototype for the Smart Cradle system and its associated mobile app is rapidly created, adhering to the requirements established in the analysis phase. This prototype incorporates both hardware and software components, leveraging tools such as the Raspberry Pi for system control and Python for coding the back-end functionalities. For the mobile app's interface, Flutter and Android Studio are employed to design and develop user-friendly layouts. Real-time visualization of UI changes is facilitated through physical device testing and Android emulators. The mock-up for the UI is created using Figma.

Development begins with creating the app's screen layouts, ensuring clarity and accessibility. Core functionalities, including menus, notifications, and real-time controls for the cradle's sensors and actuators, are then integrated. Data management for the system is structured using SQLite or similar databases to handle sleep patterns, alert logs, and user settings efficiently. The design of the mobile app adheres to principles such as simplicity, intuitive navigation, responsive feedback, error handling, and scalability to ensure an optimal user experience for busy parents. This phase emphasizes rapid prototyping while maintaining high standards for

functionality and user-focused design, preparing the system for further refinement and client feedback.

3.5.2 System Refinement

If the demonstrated prototype does not meet client expectations, it moves into a new iteration phase. This stage focuses on enhancing the user interface, refining functionalities, and improving concepts from the previous prototype based on the valuable feedback provided by the clients. After implementing these refinements, the updated prototype is presented again to evaluate its alignment with client requirements. This iterative process continues until the clients are fully satisfied that the final prototype integrates all necessary functionalities and achieves the desired objectives. Once this consensus is achieved, the approved mobile application advances to the testing phase.

3.6 Cutover

The cutover phase of the RAD methodology marks the transition of the software from development to operational use. This stage involves a variety of critical tasks, including data migration, user training, issue tracking, technical documentation, thorough testing, final customization, and system deployment. Developers focus on integrating backend data seamlessly and determining which components should be moved to the production environment. By breaking the project into smaller segments, such as initial prototypes, the RAD approach allows for iterative adjustments based on user feedback. Prototypes are presented to clients ensure the application's features and functionality meet user needs as it nears the launch date. With the software now in active use, ongoing maintenance and updates are essential to resolve issues, introduce new features, and maintain its effectiveness. This phase carries higher stakes than earlier stages, requiring rigorous testing to ensure a smooth and error-free deployment. The cutover phase concludes the RAD cycle with the successful launch and delivery of the final product.

3.6.1 System Testing

In the testing phase, the application undergoes a thorough assessment to ensure all requirements are met and its functionality is validated. This process involves four key stages, which are unit testing, integration testing, system testing, and acceptance testing. The first stage, unit testing, focuses on individually evaluating each function in the app's codebase to confirm proper operation. Known as white-box testing, this stage exposes the app's internal structure to testers for detailed analysis. Next comes integration testing, where individual functions are combined and tested together to verify seamless interaction between the user interface and underlying functions.

Following this, system testing examines the entire application as a whole, ensuring it meets all specified requirements and adheres to high-quality standards. Finally, acceptance testing is conducted with a group of potential users evaluating the application's readiness for release. This phase employs black-box testing, where the application is tested without knowledge of its internal workings. After completing these testing rounds, a test summary report is prepared to document feedback and observations, addressing aspects such as what was tested and any limitations in the features.

3.6.2 System Deployment

In the deployment stage, the application reaches its finalised form, with production-ready versions being released. At the same time, the latest version should be published on the Google Play Store to ensure broader accessibility. This phase marks the transition from using test data to processing real-world data, offering a more realistic assessment of the application's performance. Comprehensive technical and user documentation is also provided to equip both developers and users with the knowledge needed to effectively understand and use the software. Maintenance begins during this stage, highlighting the importance of regular updates and improvements to keep

the application functioning optimally. Additionally, end users are thoroughly trained to operate the application in real-world scenarios, addressing any uncertainties and ensuring smooth usage. This approach maximizes the application's benefits while minimizing potential operational challenges.

3.7 Summary

In conclusion, Chapter 3 outlines the methodology selected for the development of the proposed system, CradleSense, with a focus on the Rapid Application Development (RAD) approach. This chapter highlights the significance of gathering user feedback through questionnaires to better understand the needs and challenges faced by busy parents. The design phase, which includes brainstorming and exploring potential solutions, is followed by the creation of system architecture, flowchart, context diagram, DFDs, ERD, and wireframes the mobile applications. Additionally, the chapter discusses the testing, deployment, and review steps that will be carried out after the development phase to ensure the system effectively meets its objectives.

CHAPTER 4: IMPLEMENTATION

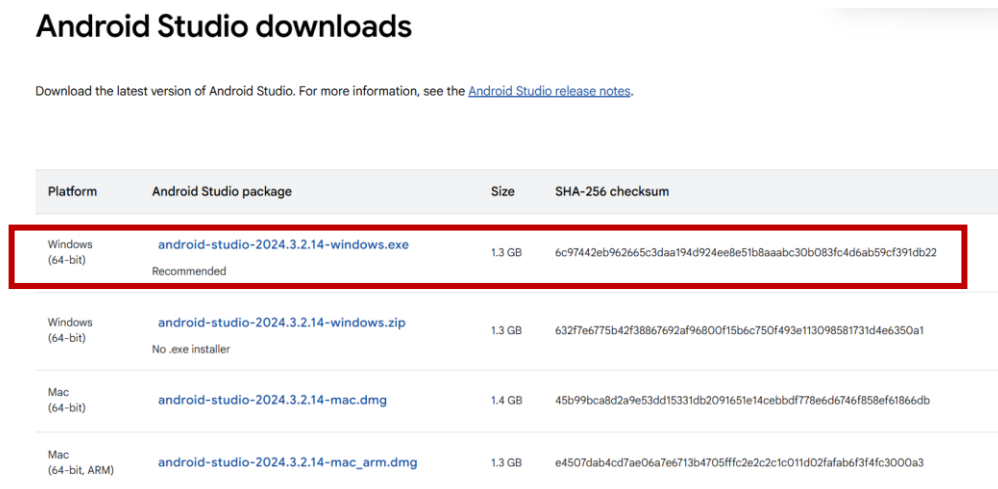
4.1 Introduction

This chapter focuses on the specifics of implementing the proposed system based on the requirement analysis and design in Chapter 3, by using the software and hardware described in previous chapters. The implementation falls within the prototype cycle, which is an iterative step that includes demonstration, refinement, and development. This is a repeated step in which the functioning prototype is built and upgraded to satisfy the specifications. It is the process of developing and preparing the system for delivery. Additionally, the chapter highlights the front-end and back-end development processes involved in implementing the proposed system.

4.2 Feature Implementation

4.2.1 Installation and Setup of Android Studio

Installation of Android Studio



Android Studio downloads			
Download the latest version of Android Studio. For more information, see the Android Studio release notes .			
Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-2024.3.2.14-windows.exe Recommended	1.3 GB	6c97442eb962665c3daa194d924ee8e51b8aaabc30b083fc4d6ab59cf391db22
Windows (64-bit)	android-studio-2024.3.2.14-windows.zip No .exe installer	1.3 GB	632f7e6775b42f38867692af96800f15b6c750f493e113098581731d4e6350a1
Mac (64-bit)	android-studio-2024.3.2.14-mac.dmg	1.4 GB	45b99bca8d2a9e53dd15331db2091651e14cebbdf778e6d6746f858ef6186ddb
Mac (64-bit, ARM)	android-studio-2024.3.2.14-mac_arm.dmg	1.3 GB	e4507dab4cd7ae06a7e6713b4705fffc2e2c2c1c011d02fafab6f3f4fc3000a3

Figure 4.1 Android Studio Installation Files

By default, the Android Studio version available for download is intended for Windows 64-bit operating systems. Users may also select from other versions based on their system compatibility

and preferences. Although the latest version available on the official website is Android Studio Meerkat, the proposed system utilises the Koala version, as illustrated in Figure 4.1. After clicking the download button, a terms and conditions dialog box will appear. Once accepted, users must scroll down, check the agreement box, and proceed with the download. Launch the setup file and follow the installation steps using the default settings. After installation is complete, Android Studio will open automatically. Users can navigate through the Android Studio Setup Wizard using custom settings by clicking Next, and finalize the setup by clicking Finish, after which the necessary components will begin downloading.

4.2.2 Installation and Setup of Raspberry Pi Imager

Installation of Raspberry Pi Imager

Raspberry Pi Imager is an official tool developed by the Raspberry Pi Foundation that allows users to easily install operating systems onto SD cards for Raspberry Pi devices. It features a simple and user-friendly interface, enabling users to select an OS, choose the storage device, and write the image with just a few clicks. Raspberry Pi Imager supports a variety of operating systems, including Raspberry Pi OS and other Linux-based distributions, making it an essential tool for setting up and configuring Raspberry Pi projects. The official Raspberry Pi Imager can be installed at <https://www.raspberrypi.com/software/> as shown in Figure 4.2 below .

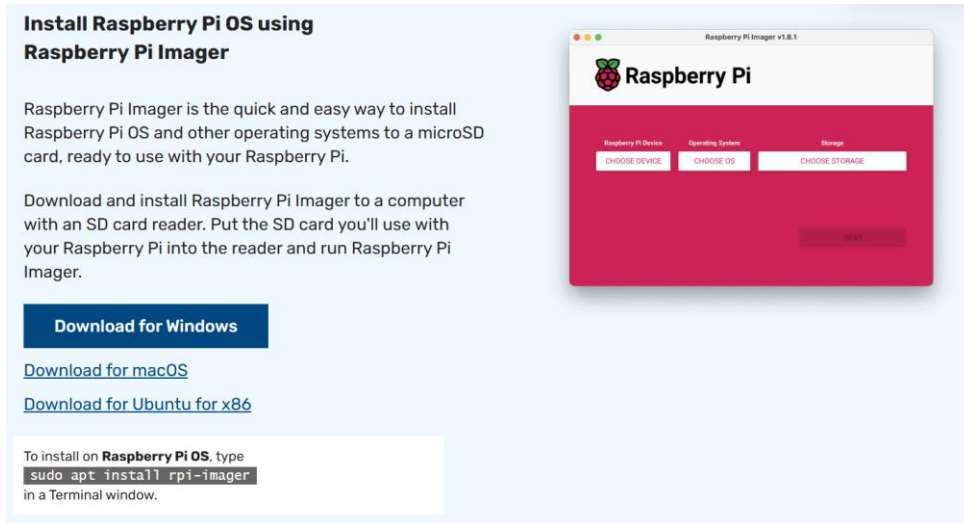


Figure 4.2 Raspberry Pi Imager Download Link

Upon downloading the Raspberry Pi Imager, run the file and the installation wizard will appear. When Figure 4.3 below is shown, click 'Install' to begin the installation process.

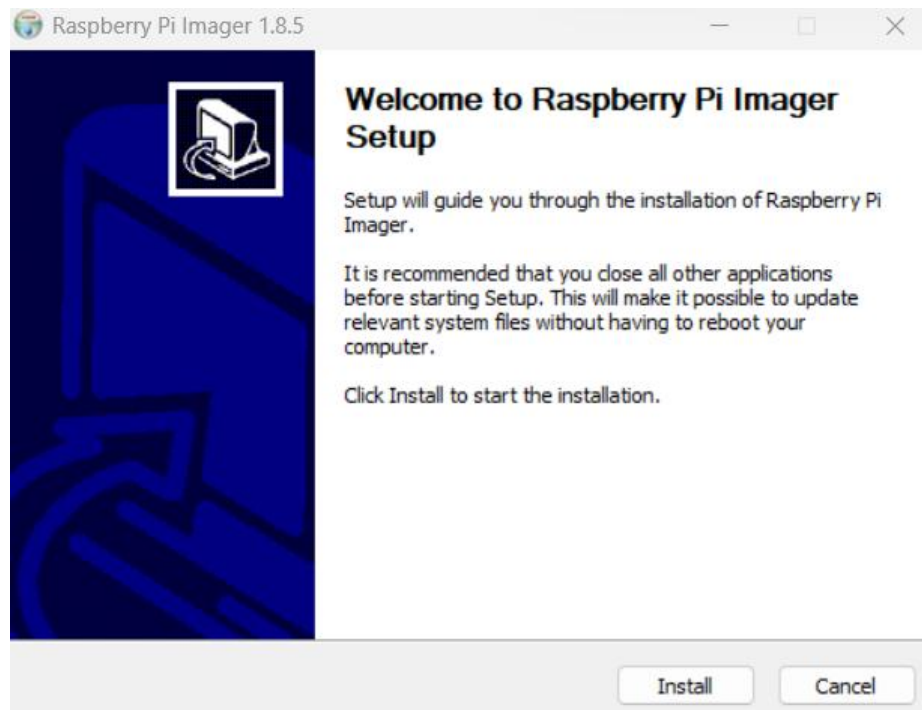


Figure 4.3 Raspberry Pi Imager Setup Wizard

Once the installation is complete, Figure 4.4 below will appear. Then, click ‘Finish’ to complete the installation and run the Raspberry Pi Imager.

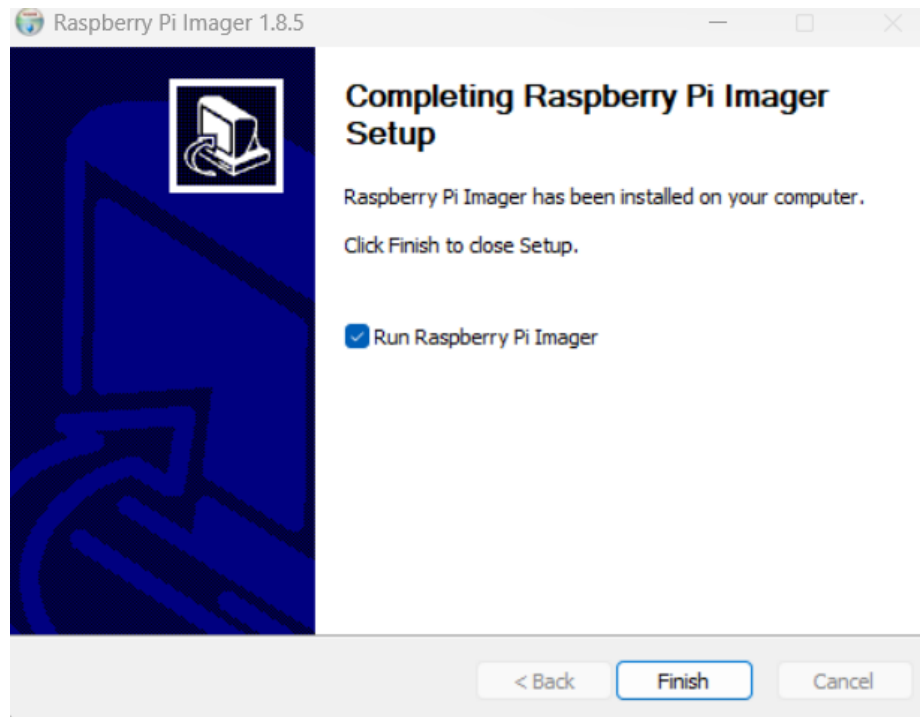


Figure 4.4 Raspberry Pi Imager Wizard Setup Complete

Setup Raspberry Pi Imager

When users first open the Raspberry Pi Imager application, they are prompted to choose three key components: the Raspberry Pi device, the operating system, and the SD card for installation. Selecting the correct board ensures compatibility, while choosing the appropriate OS and SD card size is essential for optimal performance. This simple step-by-step interface makes the setup process quick and beginner-friendly, as shown in Figure 4.5 below.

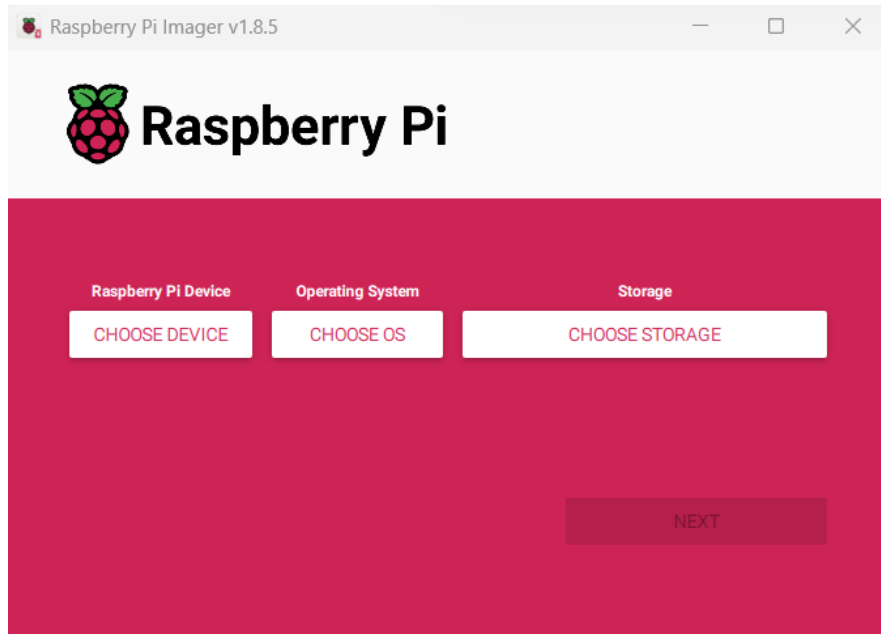


Figure 4.5 Raspberry Pi Imager Application

Raspberry Pi Imager allows users to choose from various Raspberry Pi models to ensure compatibility with the selected operating system. Since this project uses a Raspberry Pi 4, the Raspberry Pi 4 board is selected during the setup process. Choosing the correct board ensures that the operating system is optimized for the hardware's performance and features. Figure 4.6 below shows the available, but not limited to, boards that users can choose from.

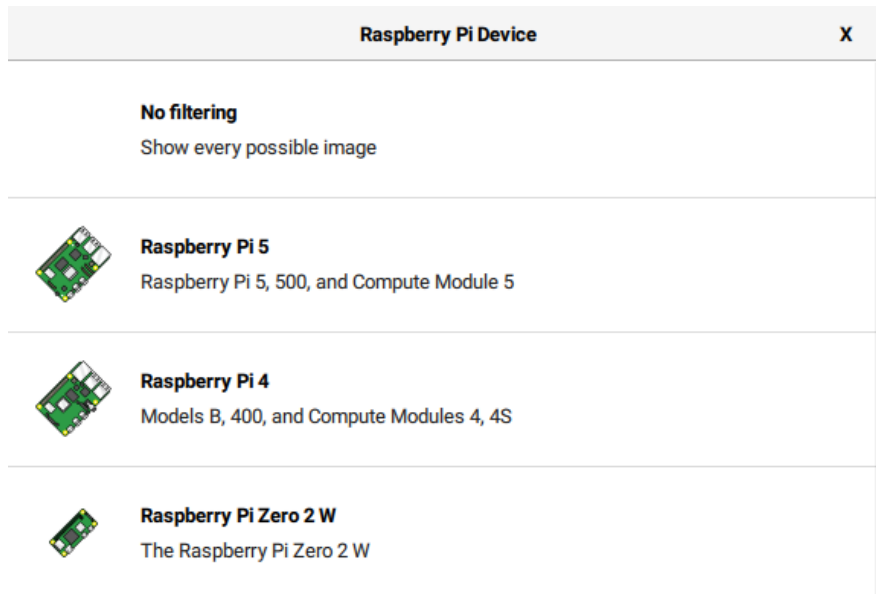


Figure 4.6 Selecting Raspberry Pi Device Model

Raspberry Pi Imager offers several versions of Raspberry Pi OS, as shown in Figure 4.7 below, each designed for different needs. The 64-bit version is suitable for newer Raspberry Pi models and can handle more memory and modern applications. The 32-bit version is more compatible with a wider range of software and works well for general tasks. The Legacy (32-bit) version is based on an older version of Debian and supports older applications or hardware. For most users and projects, especially those using Raspberry Pi 4 or newer, the Raspberry Pi OS (64-bit) is recommended for better performance and future compatibility.

Operating System		X
	Raspberry Pi OS (64-bit) A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended) (Recommended) Released: 2025-05-06 Online - 1.1 GB download	
	Raspberry Pi OS (32-bit) A port of Debian Bookworm with the Raspberry Pi Desktop Released: 2025-05-06 Online - 1.1 GB download	
	Raspberry Pi OS (Legacy, 32-bit) A port of Debian Bullseye with security updates and desktop environment Released: 2025-05-06 Online - 0.9 GB download	

Figure 4.7 Selecting Raspberry Pi Operating System

4.2.3 Installation and Setup of Arduino IDE

Installation of Arduino IDE

The Arduino IDE is an open-source software used to write and upload code to Arduino boards. It has a simple interface and supports C/C++, making it ideal for developing electronics and IoT projects. The latest version of Arduino IDE can be downloaded at <https://www.arduino.cc/en/software/> as demonstrated in Figure 4.8.

Downloads


Arduino IDE 2.3.6

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (x86-64)

Linux ZIP file 64 bits (x86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits

macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

Release Notes

Figure 4.8 Arduino IDE Installation Link

Once the Arduino IDE file is downloaded, running the executable will launch the Setup Wizard. Users must accept the License Agreement as shown in Figure 4.9 to proceed with the installation. After that, the installation process will begin automatically.

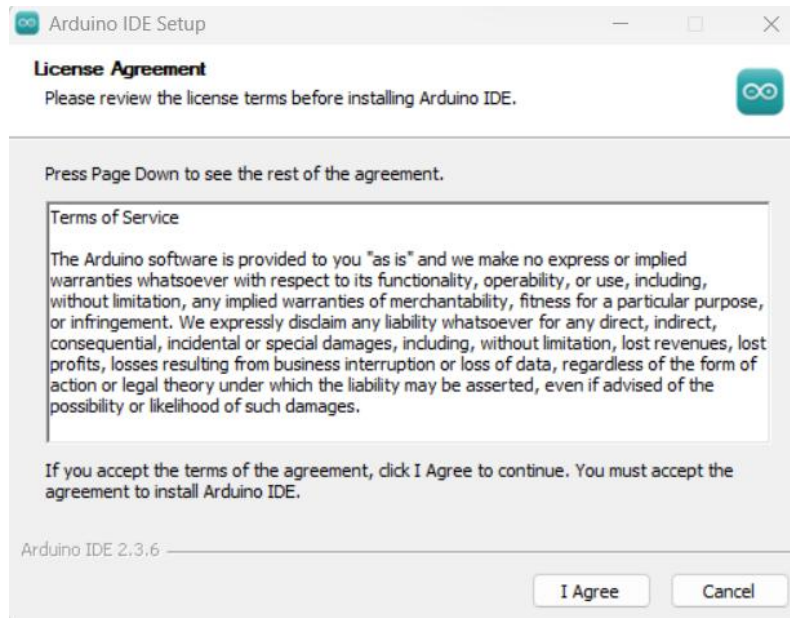


Figure 4.9 Arduino IDE License Agreement

Once the installation is complete, as shown in Figure 4.10, users can choose to launch the Arduino IDE immediately or close the installer and run it later at their convenience.

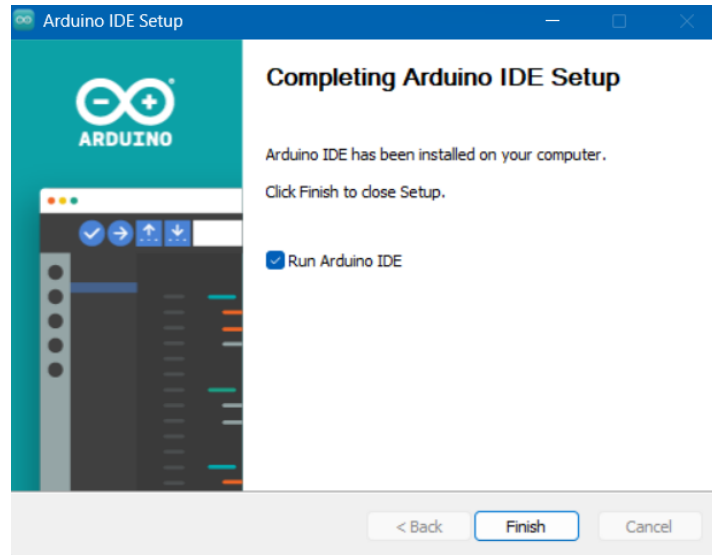


Figure 4.10 Arduino IDE Wizard Setup Complete

Setup of Arduino IDE

Upon launching the IDE, it's important to ensure the correct board manager is installed so that the IDE can detect the connected Arduino UNO board. To do this, navigate to *Tools > Board > Board Manager* as shown in Figure 4.11 below, search for *Arduino AVR Boards*, and click Install. This package includes support for a wide range of Arduino boards, including the Arduino UNO. Figure 4.12 below shows the path to select the Arduino UNO board.

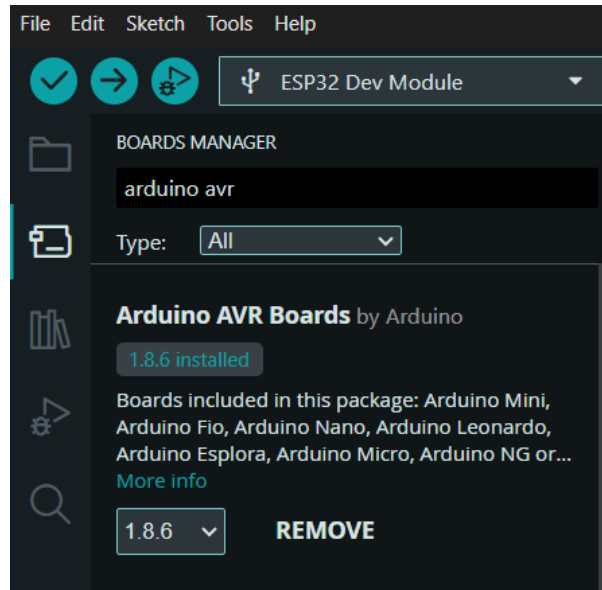


Figure 4.11 Arduino IDE Boards Manager

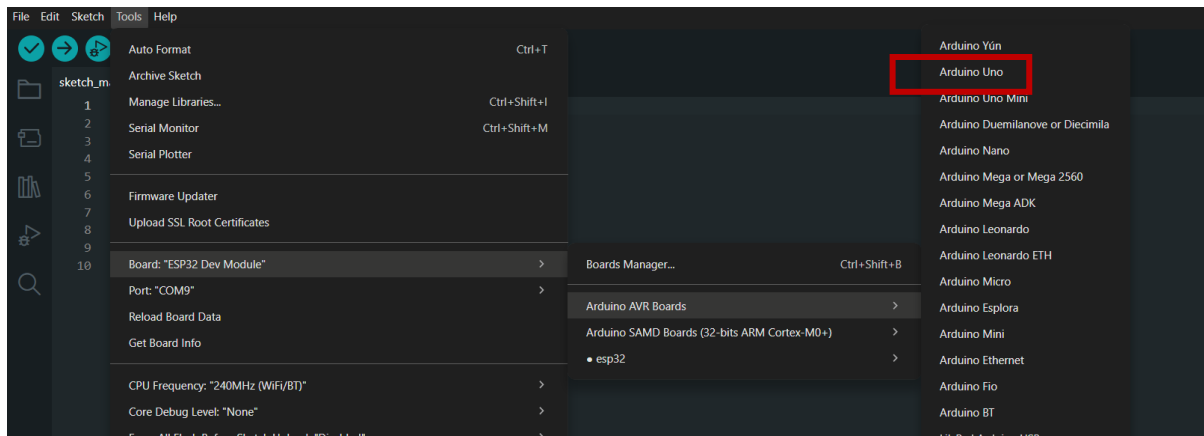


Figure 4.12 Arduino IDE Board Selection

4.2.4 Installation and Setup of Flutter SDK in Android Studio

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and provides a rich set of pre-designed widgets, enabling fast and expressive user interface development.

For this project, Flutter is used to develop the CradleSense mobile application, which serves as the main interface for parents. The app allows users to monitor their baby in real-time, receive alerts, and control cradle functions remotely. Flutter's cross-platform capability and seamless integration with Firebase make it an ideal choice for building a responsive and interactive baby monitoring app.

Installing Flutter SDK

The latest Flutter SDK can be downloaded from the official website at <https://docs.flutter.dev/get-started/install/windows/mobile>. This page provides detailed instructions and lists the system requirements needed to set up the development environment, allowing developers to ensure their devices meet the necessary conditions, as shown in Figure 4.14. Once the ZIP file is downloaded, extract its contents which include the Dart SDK, Flutter command-line tools, and other essential files, and place the extracted folder in a recommended directory such as `C:\src\flutter` for proper configuration.

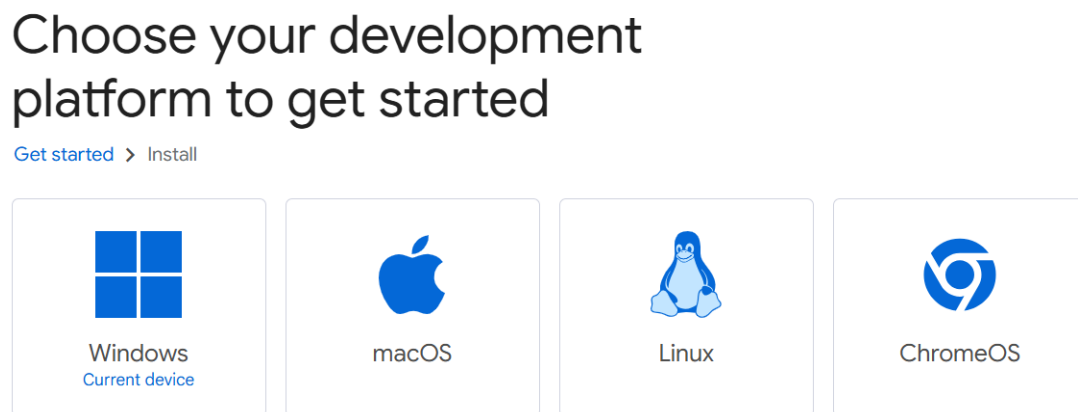
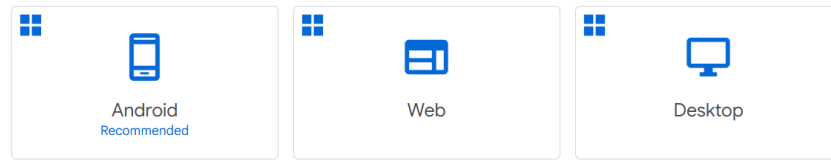


Figure 4.13 Flutter SDK Platform Selection

Choose your first type of app

[Get started](#) > [Install](#) > [Windows](#)



Your choice informs which parts of Flutter tooling you configure to run your first Flutter app. You can set up additional platforms later. If you don't have a preference, choose [Android](#).

Figure 4.14 Flutter SDK Application Selection

Setup Flutter SDK

After installing Flutter, it is necessary to add Flutter to the Windows PATH environment variable to enable running Flutter commands in the Windows console. To modify the system environment variables, first open the System Properties window as shown in Figure 4.15 below, and navigate to the 'Advanced' tab. Then, click the 'Environment Variables' button to open the 'Environment Variables' window.

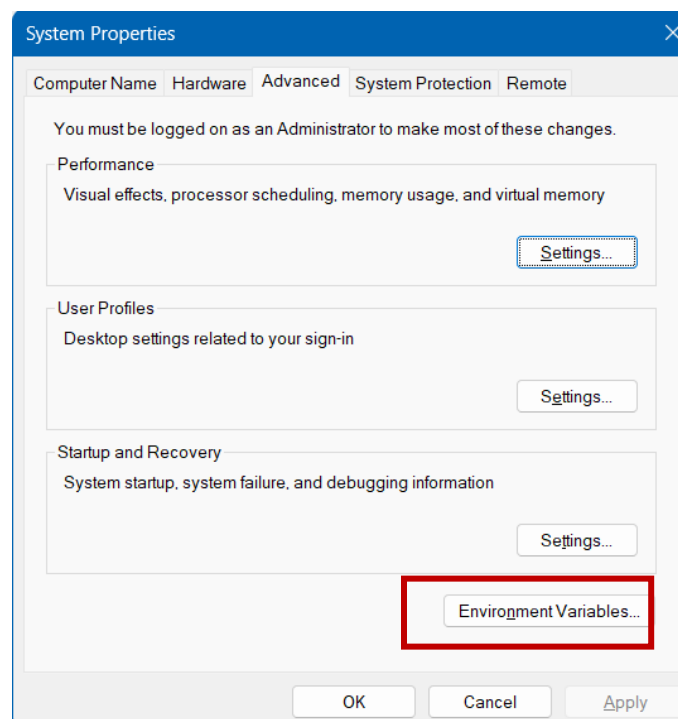


Figure 4.15 Laptop Environment Variables Setting

Once the ‘Environment Variables’ window is open, locate ‘Path’ under ‘Environment Variables’ and click the ‘Edit’ button to access the ‘Edit environment variables’ window.

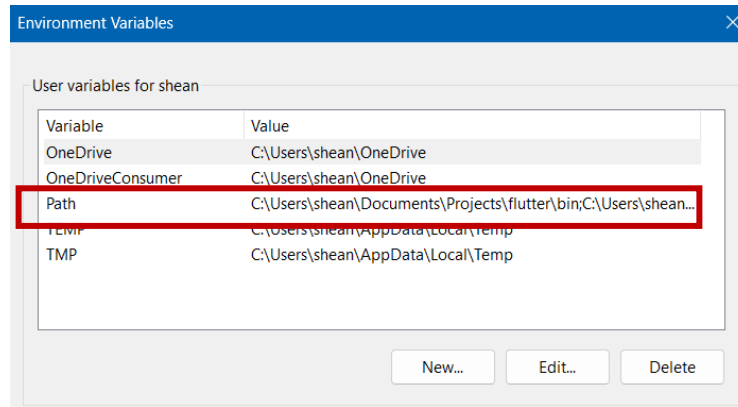


Figure 4.16 Environment Variables Path Settings

Next, navigate to the ‘bin’ folder within the installed Flutter SDK and copy its path. Add this copied location as a new environment variable, then click the ‘OK’ button at the bottom to save the changes, as shown in Figure 4.17 below.

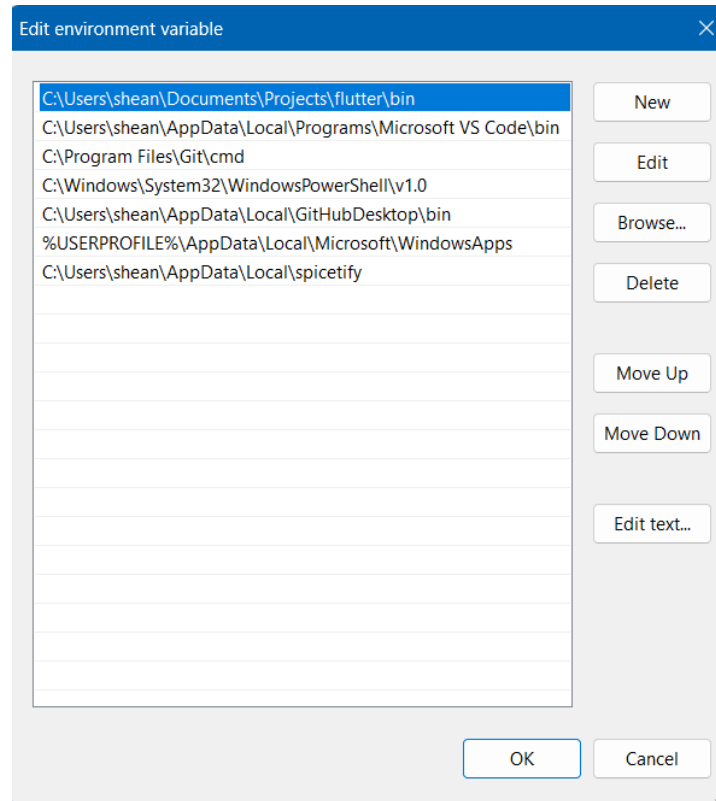


Figure 4.17 Adding Flutter Path to Environment Variables

4.3 System Hierarchy

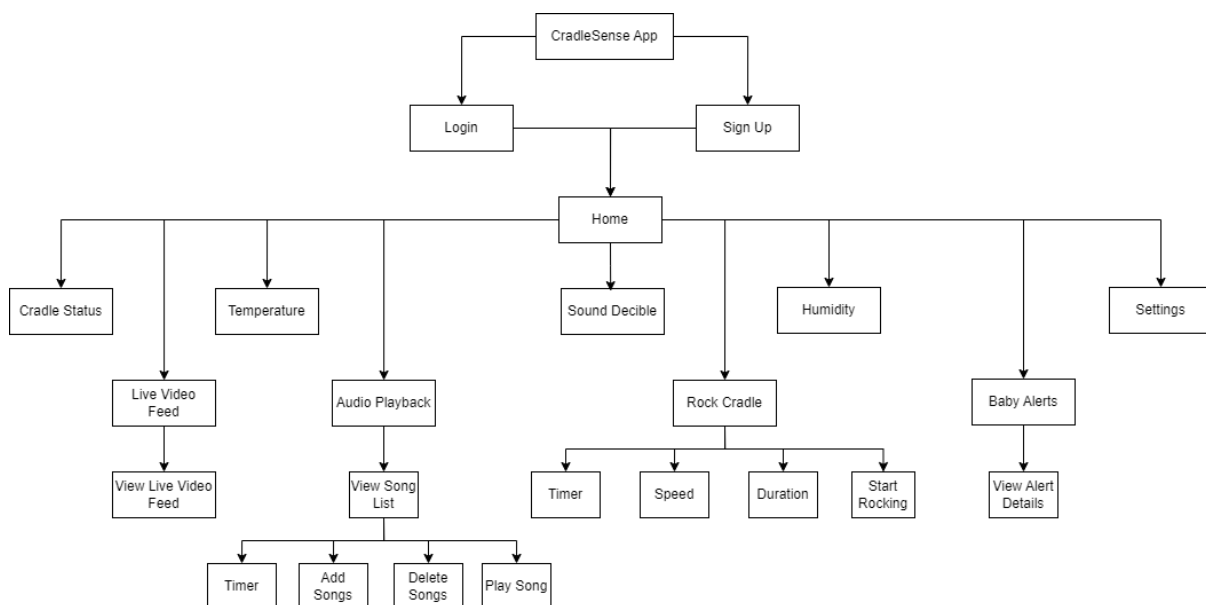


Figure 4.18 System Hierarchy of the Proposed System

Figure 4.18 shows the system hierarchy of CradleSense's system and application. At the topmost level of the CradleSense system is the mobile app, which serves as the central hub for all baby monitoring and cradle control functions. This app is divided into several main components: Login and Sign Up, which handle user authentication and account creation, respectively. Once users successfully log in or sign up, they are directed to the Home module, where they can access all available features.

The Home module is further divided into specific sub-systems, each of which are designed to monitor and control different aspects of the Smart Cradle. These include Cradle Status, which provides live updates on the cradle's condition; Temperature and Humidity, showing the current environmental conditions; Sound Decibel, which tracks noise levels in the room; Rock Cradle, allowing users to control the cradle's motion; Baby Alerts, handling notifications for any changes or concerns; and Settings, where users can edit or view account information, manage connected devices, and logout from their account.

Digging deeper, the Cradle Status module includes a Live Video Feed, enabling users to watch the baby through a connected camera. The CradleSense app also offers Audio Playback, with options for managing a Song List, adding or removing songs, setting a Timer for playback, and playing music to soothe the baby. The Rock Cradle module allows users to adjust the cradle's movement by setting a Timer, Speed, and Duration, followed by a Start Rocking function. The Baby Alerts module enables users to view detailed notifications triggered by the system.

This organisation ensures a smooth, user-friendly experience, allowing parents to easily monitor and control various aspects of their baby's well-being while using the CradleSense app.

The modular structure integrates these features seamlessly, creating an intuitive interface for managing the Smart Cradle system.

4.4 Workflow of the Proposed Application

The soothing response process is triggered based on the detection of specific baby-related conditions, such as crying sounds or motion. In other words, the CradleSense's automated actions, such as rocking or playing lullabies, are only activated when the system detects that the baby is crying or moving, ensuring that the response is contextually relevant and helps comfort the infant when needed.

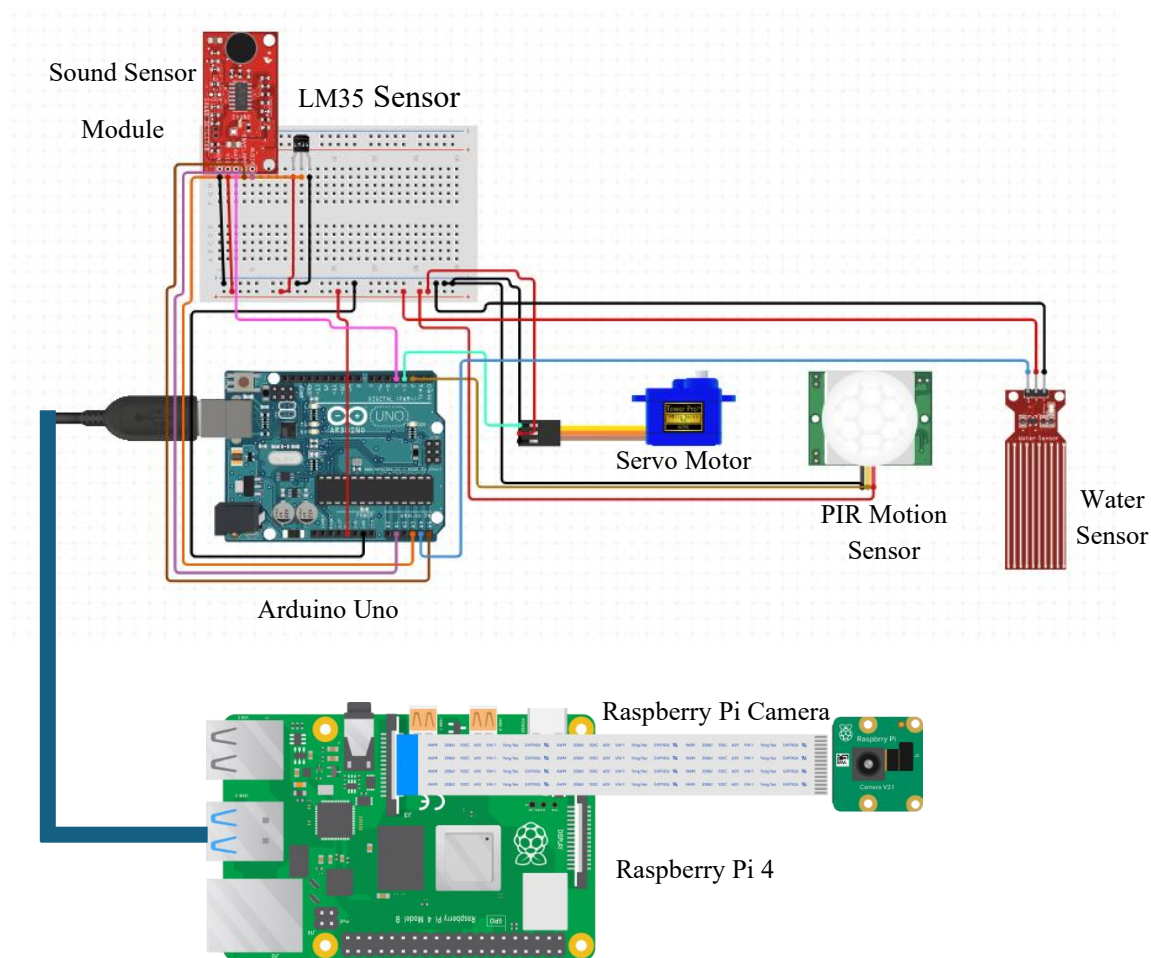


Figure 4.19 Hardware Connection Between Arduino and Raspberry Pi

As shown in Figure 4.19 above, the code for the Arduino to control the sensors will be uploaded to the board beforehand. Once uploaded, the Arduino will then be plugged into the Raspberry Pi, as the Pi will act as a gateway for the hardware to run headless. This means, essentially, no computer or monitor needs to be constantly present to run the system.

```
# Connect to Arduino via USB

arduino = serial.Serial('/dev/ttyACM0',
                        9600, timeout=1)

time.sleep(2) # Wait for Arduino to reboot
```

Figure 4.20 Code Snippet from Python in Raspberry Pi

This code snippet represents the connection between the Arduino and the Raspberry Pi. When the Arduino is plugged into the Pi, the Pi detects it as `/dev/ttyACM0`, this code snippet is to ensure that the Pi is communicating with the correct Arduino.

4.4.1 Real-time Cry Detection Process

The real-time cry detection process is central to the CradleSense system's responsiveness. Using a sound sensor connected to the Arduino Uno, the system continuously listens for audio input in the cradle's environment. When the sensor detects a sound that matches the frequency and volume range typical of a baby's cry, the system identifies it as a cry event. This triggers a series of automated soothing actions, such as activating the rocking mechanism or playing a pre-programmed lullaby. The real-time nature of this process ensures that the system can respond promptly to signs of distress, minimising delays in providing comfort and reducing the burden on parents, especially during nighttime or when they are multitasking.

4.4.2 Motion Detection Process

The motion detection process monitors the physical movement of the baby within the cradle using a motion sensor, such as a PIR motion sensor. Sudden or continuous movements may indicate discomfort, restlessness, or potential risk of falling. When such movement is detected, the system interprets it as a need for attention and activates the soothing functions, including gentle rocking and or sound playback. This process enhances safety by allowing the system to recognise abnormal motion patterns and ensures the baby remains calm and secure. It is particularly useful for recognising when the baby is transitioning from sleep to wakefulness or experiencing mild agitation.

4.4.3 Moisture Detection Process

The moisture detection process is designed to identify wet conditions that may result from a soiled diaper or accidental spills. A moisture sensor placed beneath or in contact with the baby's bedding to detect the presence of liquid. Once a threshold level of moisture is reached, the sensor sends a signal to the Raspberry Pi, which in turn sends a notification to the parent's mobile app. This allows caregivers to promptly address the issue, maintaining hygiene and preventing skin irritation. This process reduces the need for constant manual checks and contributes to the overall comfort and health of the baby.

4.4.4 Temperature Monitoring Process

Maintaining a comfortable ambient temperature is essential for infant care. The temperature monitoring process uses an LM35 analog temperature sensor to continuously measure the surrounding temperature within the cradle area. The Raspberry Pi interprets these readings and compares them against predefined safe thresholds. If the temperature falls below or exceeds the

acceptable range, an alert is sent to the parent's mobile application. This enables timely adjustments to the room's climate, such as turning on a fan or heater. By integrating this process, CradleSense helps prevent overheating or chilling, both of which can disrupt sleep and pose health risks to infants.

4.4.5 Live Video Streaming Process

To enhance remote monitoring capabilities, the live video streaming process uses a Raspberry Pi Camera Module to transmit real-time footage and sound from the cradle to the parent's mobile application. This feature allows parents to visually check on their baby from anywhere, offering peace of mind and reducing the need for physical presence. This process complements the sensor-based detection features by giving parents direct access to real-time context and reassurance.

4.5 Coding of Proposed Algorithm and Function

Each of the algorithm's sensor readings will be printed out in the Serial Monitor of Arduino IDE at serial baud 9600. Figure 4.19 shows the declaration and initialisation of the variables and pins of the connected sensors. The serial baud rate is set at 9600, therefore each serial print out can be seen when the serial monitor is set to 9600, as shown in Figure 4.20.

<pre>#include <Servo.h> const int soundPin = A0; const int wetnessPin = A1; const int tempPin = A2; const int pirPin = 2; const int servoPin = 9; const int soundThreshold = 600; const int wetnessThreshold = 300; const int stopValue = 89;</pre>	<pre>Servo cradleServo; bool rocking = false; unsigned long rockStartTime = 0; unsigned long rockDuration = 0; int rockingDelay = 800; void setup() { Serial.begin(9600); cradleServo.attach(servoPin); cradleServo.write(stopValue); pinMode(pirPin, INPUT); }</pre>
---	---

Figure 4.21 Arduino IDE Pin Initialisation and Declaration

<pre>21:27:58.554 -> [ALERT]:motion 21:28:00.279 -> [TEMP]:38.61,red 21:28:05.367 -> [ALERT]:wetness 21:28:08.351 -> [TEMP]:28.35,black 21:28:10.076 -> [ALERT]:wetness 21:28:14.794 -> [ALERT]:wetness 21:28:17.773 -> [TEMP]:28.35,black 21:28:19.505 -> [ALERT]:wetness</pre>
--

Figure 4.22 Serial Monitor Output in Arduino IDE

4.5.1 Cry Detection Algorithm

Software

The Raspberry Pi listens to the serial port for this alert and logs the event in the Firebase Firestore under the alerts collection with a timestamp. In the Flutter application, the Baby Alerts page listens to Firestore changes and updates in real time to inform parents.

<pre>if line.startswith("[ALERT]:cry"): log_alert("cry")</pre>
--

Figure 4.23 Raspberry Pi main.py Code Snippet to Communicate with Firebase

```
{  
  "type": "cry",  
  "timestamp": <SERVER_TIMESTAMP>  
}
```

Figure 4.24 Information Stored In Firestore

Hardware

When the system detects a high sound level from the sound sensor connected to the Arduino's analog pin A0, it assumes the baby is crying. If the detected value exceeds the defined threshold and the cradle is not already rocking, the Arduino sends a serial message [ALERT]:cry to the Raspberry Pi. Simultaneously, the Arduino begins a 3-minute low-speed rocking motion.

```
// === 1. CRY DETECTION ===  
int soundLevel = analogRead(soundPin);  
if (soundLevel > soundThreshold && !rocking) {  
  Serial.println("[ALERT]:cry");  
  startRocking(180000, "low"); // Default to 3 min, low speed  
}
```

Figure 4.25 Arduino Snippet for Cry Detection Algorithm

4.5.2 Motion Detection Algorithm

Software

The Raspberry Pi logs the event in Firestore. The Baby Alerts page in Flutter captures this and displays the alert instantly.

```

        if line.startswith("[ALERT]:motion"):
            log_alert("motion")

    {
        "type": "motion",
        "timestamp": <SERVER_TIMESTAMP>
    }

```

Figure 4.26 Code Snippet from main.py for Motion Sensing and Firebase Data Stored

Hardware

A PIR motion sensor is connected to the Arduino to monitor the baby's movement. If motion is detected (PIR reads HIGH) and the cradle is not already rocking, the Arduino sends [ALERT]:motion over serial and begins low-speed rocking for 3 minutes.

```

// === 2. MOTION DETECTION ===
if (digitalRead(pirPin) == HIGH && !rocking) {
    Serial.println("[ALERT]:motion");
    startRocking(180000, "low");
}

```

Figure 4.27 Arduino Code Snippet for Motion Detection Algorithm

4.5.3 Temperature Reading Algorithm

Software

The sensor readings are sent over serial to the Raspberry Pi in the format [TEMP]:25.1,black. The Raspberry Pi writes this to Firestore, and the Flutter app displays the temperature with the corresponding text colour.

```
if line.startswith("[TEMP]:"):
    temp, color = line[7:].split(',')
    db.collection('temperature').add({
        'value': float(temp),
        'color': color,
        'timestamp': firestore.SERVER_TIMESTAMP
    })})
```

```
{
  "value": 25.1,
  "color": "black",
  "timestamp": <SERVER_TIMESTAMP>
}
```

Figure 4.28 Code Snippet from main.py for Temperature Sensing and Data Stored in Firebase

Hardware

The LM35 sensor outputs a voltage that the Arduino reads to calculate ambient temperature in Celsius. The system converts this voltage and determines a colour code, where red for hot, blue for cold, and black for ideal temperature.

```
// === 4. TEMPERATURE READING (Every 5s) ===
static unsigned long lastTempTime = 0;
if (millis() - lastTempTime > 5000) {
  lastTempTime = millis();
  int analogValue = analogRead(tempPin);
  float voltage = analogValue * (5.0 / 1023.0);
  float temperature = voltage * 100.0;

  String color;
  if (temperature < 20) color = "blue";
  else if (temperature > 30) color = "red";
  else color = "black";

  Serial.print("[TEMP]:");
  Serial.print(temperature);
  Serial.print(",");
  Serial.println(color);
}
```

Figure 4.29 Arduino Code Snippet for Temperature Monitoring Algorithm

4.5.4 Wetness Detection Algorithm

Software

The Raspberry Pi logs this alert into the Firestore under the alerts collection. In the Flutter app, the Baby Alerts section listens for and displays this alert immediately.

```
if line.startsWith("[ALERT]:wetness"):
    log_alert("wetness")    })

{
  "type": "wetness",
  "timestamp": <SERVER_TIMESTAMP>
}
```

Figure 4.30 Code Snippet from main.py for Wetness Sensing and Data Stored in Firebase

Hardware

A moisture sensor connected to the Arduino detects if the baby's diaper is wet. When the moisture reading falls below a certain threshold, the Arduino sends [ALERT]:wetness via serial to the Raspberry Pi.

```
// === 3. WETNESS DETECTION ===
int wetValue = analogRead(wetnessPin);
if (wetValue < wetnessThreshold) {
  Serial.println("[ALERT]:wetness");
  delay(3000); // debounce
}
```

Figure 4.31 Arduino Code Snippet for Wetness Detection Algorithm

4.5.5 Cradle Rocking Algorithm

Software

This section of the code can be found in *rock_cradle_controller.dart* whereby this controller method writes the selected speed and duration from the Flutter UI to Firestore under the document control/rock. The Raspberry Pi listens to this document and forwards commands to the Arduino to control cradle movement.

```
import 'package:cloud_firestore/cloud_firestore.dart';

class CradleController {
  Future<void> rockCradle({
    required String speed,
    required int durationInSeconds,
  }) async {
    await FirebaseFirestore.instance
      .collection('control')
      .doc('rock')
      .set({
        'status': 'start',
        'speed': speed.toLowerCase(),
        'duration': durationInSeconds,
      });
  }

  Future<void> stopCradle() async {
    await FirebaseFirestore.instance
      .collection('control')
      .doc('rock')
      .update({'status': 'stop'});
  }
}
```

Figure 4.32 Flutter Code Snippet *rock_cradle_controller.dart*

This method is called when the user taps the “Start Rocking” button. It sends the selected values to the controller, starts a countdown timer in the app, and stops the cradle when the timer finishes, or the user cancels it.

```
void _startRocking() async {
  if (selectedDurationLabel == null || selectedSpeedLabel == null) return;

  await CradleController().rockCradle(
    speed: selectedSpeedLabel!,
    durationInSeconds: durationOptions[selectedDurationLabel!]!.inSeconds,
  );
  remainingTime = durationOptions[selectedDurationLabel!]!.inSeconds;
  countdownTimer?.cancel();
  countdownTimer = Timer.periodic(const Duration(seconds: 1), (_) {
    if (remainingTime.inSeconds <= 1) {
      countdownTimer?.cancel();
      CradleController().stopCradle();
      setState(() {
        isRocking = false;
        selectedDurationLabel = null;
        selectedSpeedLabel = null;
      });
    } else {
      setState(() => remainingTime -= const Duration(seconds: 1));
    }
  });
  setState(() => isRocking = true);
}
```

Figure 4.33 Code Snippet from main.py for Rocking Cradle Manually Function

This segment of the Raspberry Pi Python script is responsible for listening to cradle control instructions issued from the Flutter mobile application through Firebase Firestore. When the status field in the control/rock document is set to "start", the script dynamically retrieves the selected speed and duration values submitted by the user. If any of these values are missing, it defaults to "low" speed and 180 seconds duration to ensure system continuity. This command is sent to the Arduino board via the serial port using `ser.write()`, where it is interpreted and used to drive the servo motor for cradle movement accordingly. Finally, the status field in Firestore is reset to "idle" to prevent repeated execution of the same command and to prepare the system for future inputs.

```
if status == 'start':  
    speed = data.get('speed', 'low')  
    duration = data.get('duration', 180)  
    command = f"rock:start,{speed},{duration}\n"  
    ser.write(command.encode())
```

Hardware

This section of the Arduino sketch is responsible for both executing the rocking motion of the cradle based on a timer, and processing manual commands sent from the Raspberry Pi via serial communication.

The first block checks if the rocking flag is active and whether the elapsed time since rocking started is still within the specified `rockDuration`. If so, the servo motor oscillates between two positions (60° and 120°) with a delay interval defined by `rockingDelay`, which corresponds to the user-selected speed. This mimics a left-right rocking motion. After each full cycle, the servo is returned to its calibrated center (`stopValue`) to stabilize the cradle.

Once the specified duration elapses, the system disables the rocking mode by setting rocking = false, stops the servo, and prints a serial message indicating that the rocking has ended.

```
// === 5. ROCKING TIMER ===  
if (rocking && millis() - rockStartTime < rockDuration) {  
    cradleServo.write(60); delay(rockingDelay);  
    cradleServo.write(120); delay(rockingDelay);  
    cradleServo.write(stopValue);  
} else if (rocking && millis() - rockStartTime >= rockDuration) {  
    rocking = false;  
    cradleServo.write(stopValue);  
    Serial.println("ROCKING ENDED");  
}
```

Figure 4.34 Arduino Code Snippet for Rocking Cradle Manually

The second block listens for incoming serial commands from the Raspberry Pi. If the message starts with "rock:start", the code parses the speed and duration values embedded in the string using `String.indexOf()` and `substring()` functions. These values are then passed to a helper function `startRocking()`, which initializes the timer and speed for the rocking operation. If the received command is "rock:stop", the system immediately halts any ongoing rocking activity, resets the servo to its idle position, and prints a confirmation message over serial.

```

// === 6. COMMAND FROM PI ===
if (Serial.available()) {
  String cmd = Serial.readStringUntil('\n');
  cmd.trim();

  if (cmd.startsWith("rock:start")) {
    int speedIndex = cmd.indexOf(',') + 1;
    int durIndex = cmd.lastIndexOf(',') + 1;

    String speed = cmd.substring(speedIndex, durIndex - 1);
    int duration = cmd.substring(durIndex).toInt();

    startRocking(duration * 1000UL, speed);
    Serial.println("MANUAL ROCK STARTED");
  }

  if (cmd == "rock:stop") {
    rocking = false;
    cradleServo.write(stopValue);
    Serial.println("MANUAL ROCK STOPPED");
  }
}

delay(100);

```

Figure 4.35 Arduino Code Snippet for Accepting Code from Raspberry Pi

4.6 Summary

In Chapter 4, the implementation phase of the CradleSense Smart Cradle system is comprehensively discussed, beginning with an overview of the development tools and technologies used. The chapter outlines the installation and configuration processes for required software such as Flutter, Android Studio, Raspberry Pi Imager and Arduino IDE, ensuring the development environment is properly set up. The system hierarchy is then broken down to show the structure and interaction between the mobile application, Raspberry Pi gateway, and Arduino-based cradle system. This is followed by an explanation of the system's workflow, illustrating how data is collected, processed, and acted upon in real time. Although the detailed coding implementation is still in progress, key components and integration points have been defined. This chapter establishes the groundwork for system testing and refinement in the upcoming phases.

CHAPTER 5: TESTING

5.1 Introduction

Testing is a critical phase in the development of the CradleSense system, ensuring that the final product functions reliably and meets the expectations of end users, particularly parents and caregivers. Rigorous testing was conducted to validate the effectiveness, stability, and usability of the Smart Cradle under different operating conditions. This chapter outlines the testing strategies implemented throughout the project, which was developed using the Rapid Application Development (RAD) methodology. The testing process is categorised into two major types, which are functional testing and non-functional testing. Functional testing concentrates on validating the software components, including unit testing and integration testing. These ensure that each module of CradleSense, such as cry detection, temperature and humidity monitoring, cradle rocking, and mobile app controls, performs as intended both individually and when integrated with other components. On the other hand, non-functional testing evaluates the system's overall behaviour and performance. This includes reliability testing to assess the system's stability over time, performance testing to monitor response times and resource usage, usability testing to ensure an intuitive and smooth user experience, and metrics-based evaluations to assess the efficiency of key features like live video streaming and alert notifications. Each testing methodology is discussed in the subsequent sections, detailing its purpose, process, and the results obtained, providing a comprehensive view of CradleSense's readiness for real-world deployment.

5.2 Functional Testing

Functional testing is a form of software testing that ensures the system behaves according to the predefined functional requirements. It focuses on validating each component of the CradleSense system, as well as their interactions, to confirm they operate as expected and fulfil user needs. The

primary aim is to ensure that all features such as cry detection, cradle rocking, environmental monitoring, and mobile app commands perform accurately and reliably.

Two types of functional testing were employed in this project, which are unit testing and integration testing. Unit testing involves evaluating the functionality of individual modules in isolation. This allows for early detection of errors or bugs within specific components, which are then corrected before proceeding further. Once all individual units are verified to be error-free, integration testing is conducted to examine how these modules interact with each other. This includes verifying data flow, such as ensuring sensor input is correctly processed and stored in the database, and that it can be retrieved seamlessly by the mobile application.

During integration testing, each module is supplied with relevant input values to validate whether the output aligns with expected behaviour defined in the test cases. The testing process was carried out manually by comparing actual results to expected outcomes, thereby confirming that the CradleSense system functions as intended when deployed as a whole.

5.2.1 Unit Testing

Unit testing refers to the process of verifying the functionality of individual components or modules of the CradleSense system in isolation. Each unit, whether a sensor function, control logic, or a specific feature in the mobile app, is tested independently to ensure it operates as intended. The main objective of unit testing in this project is to detect and resolve errors early in the development process, thereby improving the reliability and efficiency of the system.

By isolating and testing specific functions such as cry detection logic, temperature reading from the LM35 sensor, or command execution for rocking the cradle, developers can confirm the correctness of each module before integrating it with the rest of the system. This approach helps prevent the propagation of bugs into more complex interactions during later stages.

Unit testing in CradleSense also supports better code understanding and maintenance, allowing quick identification and correction of faulty logic. A white-box testing approach was applied, focusing on the internal logic of the code to ensure each part of the system behaves as expected. These tests provided a strong foundation for building a reliable and responsive smart cradle system.

Test cases for Proposed Mobile Application

Table 5.1 to Table 5.7 shows the various unit test cases for the modules available in the CradleSense Application.

Table 5.1 Unit Test Case for Sign Up

ID	2	Module Name	Sign Up			
Objective	To test the register functionality					
Name	Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass
No	Prerequisites					
1	The application is installed correctly and launched successfully.					
2	User must have an active internet connection.					
Test Scenario	User wants to create a new account.					
ID	Test Case	Input Data	Expected Result	Actual Result	Status	
1	Open the application and navigate to the Sign Up screen.	Click on the “Sign Up” from Welcome screen and “Not a Member? Sign Up” from the Login Screen.	The Sign-Up screen is displayed.	As Expected	Pass	
2	Enter a valid full name, email address, phone number and password.	- Email: djohn@gmail.com - Full Name: John Doe - Phone Number: 0112223456 - Password: Johndoe1*	User account is created, and a message “Success! Account created successfully.” is displayed. User is directed to Dashboard screen.	As Expected	Pass	

3	Attempt to sign up with an invalid email format.	Email: djohn@gmailcom	Error message indicating “Please enter a valid email” at the email text field.	As Expected	Pass
4	Attempt to sign up with an existing registered email address.	Email: djohn@gmail.com	Error message indicating “Registration failed. Email already exists. Please sign in.”	As Expected	Pass
5	Attempt to sign up with an invalid full name format.	Full Name: John Doe 007	Error message indicating “Name must contain only letters and spaces.” at the full name text field.	As Expected	Pass
6	Attempt to sign up with an invalid phone number format	Phone Number: 0112	Error message indicating “Phone number must be 10 or 11 digits.” at the phone number text field.	As Expected	Pass
7	Attempt to sign up with an invalid password format.	Password: johndoe	Error message indicating “Password must be 8+ chars, include uppercase, lowercase, number and special char.” at the password text field.	As Expected	Pass

Table 5.2 Unit Test Case for Login

ID	2	Module Name	Login				
Objective	To test the login functionality						
<i>Name</i>	<i>Sheana Kasih Benedict</i>			<i>Date</i>	<i>4th June 2025</i>	<i>Status</i>	<i>Pass</i>
No	Prerequisites						
1	The application is installed correctly and launched successfully.						
2	User must have an active internet connection.						
3	User must have an existing registered account.						

Test Scenario		User wants to login to their account.			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Open the application and navigate to the Login screen.	Click on the “Login” from Welcome screen and “Already Have an Account? Login” from Sign Up Screen.	The Login screen is displayed.	As Expected	Pass
2	Enter a valid email and password.	- Email: djohn@gmail.com - Password: Johndoe1*	User is logged in successfully. User is directed to Dashboard screen.	As Expected	Pass
3	Attempt to sign up with an invalid email format.	Email: djohn@gmailcom	Error message indicating “Please enter a valid email” at the email text field.	As Expected	Pass
4	Attempt to sign up with an invalid password.	- Email: djohn@gmail.com - Password: john1	Error message indicating “Login failed! Incorrect email or password.”	As Expected	Pass
5	Attempt to sign up with an empty email and/or password.	Leave email and/or password fields empty	Error message indicating “Please enter your email” and “Please enter your password” at the email and password text field.	As Expected	Pass
6	Validate session persistence after closing and reopening the app.	Log in successfully, close the app, and reopen the app.	User remains logged in and is navigated directly to the dashboard screen after pressing on “Login” on the welcome screen.	As Expected	Pass

Table 5.3 Unit Test Case for User Profile

ID	3	Module Name	User Profile				
Objective	To test the functionalities of viewing, updating, and logging out from the user profile.						
Name	Sheana Kasih Benedict		Date	4 th June 2025		Status	Pass
No	Prerequisites						
1	The application is installed correctly and launched successfully.						
2	User must have an active internet connection.						
3	User must be logged in to access the profile.						

Test Scenario		View Account Settings				
ID	Test Case	Input Data	Expected Result	Actual Result	Status	
1	Navigate to the Account Settings screen	Tap on the Profile icon in the home screen, then click on the “Account Settings” in the Settings Page	Account Settings screen is displayed with the current user details and option to edit them.	As Expected	Pass	

Test Scenario		Update User Profile				
ID	Test Case	Input Data	Expected Result	Actual Result	Status	
2	Update full name and phone number with valid data.	- Full Name: Jane Doe - Phone Number: 0105556789	Profile is updated, and changes are reflected in the profile.	As Expected	Pass	
3	Update email with valid data.	Email: janedoe@gmail.com	Verification email is sent to new email. Once verified, changes are reflected in the profile after re-login.	As Expected	Pass	
4	Update password with valid data.	- Current Password: Johndoe1* - New Password: Janedoe123*	Profile is updated, and users can login with new password upon next login.	As Expected	Pass	
5	Attempt to update email with an invalid email format.	Email: janegmail.com	Error message indicating “Please enter a valid email” at the email text field.	As Expected	Pass	
6	Attempt to update full name with invalid name format.	Full Name: Jane Doe 123	Error message indicating “Name must contain only letters and spaces.” at the full name text field.	As Expected	Pass	

7	Update profile with special characters in the full name.	Full Name: Jane O'Malley	Error message indicating "Name must contain only letters and spaces." at the full name text field.	As Expected	Pass
8	Attempt to update phone number with invalid phone number format	Phone Number: 012345	Error message indicating "Phone number must be 10 or 11 digits." at the phone number text field.	As Expected	Pass
9	Attempt to update password with invalid credentials.	- Current Password: john - New Password: Janedoe123*	Error message indicating "Error! Wrong credentials."	As Expected	Pass
10	Attempt to update password with invalid password format.	- Current Password: Johndoe1* - New Password: jane	Error message indicating "Password must be 8+ chars, include uppercase, lowercase, number and special char." at the password text field.	As Expected	Pass
11	Delete user account.	- Tap on "Delete Account" from the Manage Account Screen and input password. - Password: Johndoe1*	Success message "Account Deleted! Your account has been permanently removed." Is displayed and user is navigated back to the Welcome Screen.	As Expected	Pass

Test Scenario		Logout			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
12	Log out from the application.	- Tap on "Logout" from the Profile Icon in the Dashboard Screen. - Tap on "Logout" in the Settings Screen.	User is logged out and navigated to the Login screen.	As Expected	Pass

13	Verify user is still logged in when no has been selected during logout confirmation.	Tap on “Logout” button in User Profile screen, followed by “No” during confirmation.	User is not logged out.	As Expected	Pass
14	Verify session persistence after logout and closing the app.	Log out, close the app, and reopen it.	User is navigated to the Login screen, indicating that the session has not persisted.	As Expected	Pass

Table 5.4 Unit Test Case for Live Video Feed

ID	4	Module Name	Live Video Feed				
Objective	To test the functionalities of viewing real-time footage.						
Name	Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass	
No	Prerequisites						
1	The application is installed correctly and launched successfully.						
2	User must have an active internet connection.						
3	User must be logged in to an existing account.						
Test Scenario	User wants to view live video footage.						
ID	Test Case		Input Data		Expected Result	Actual Result	Status
1	Navigate to the Live Video screen.		Tap on the “Live Video” button in the Dashboard Screen.		User is navigated to the “Live Video” screen and the footage is displayed.	As Expected	Pass

Table 5.5 Unit Test Case for Soothing Sound Selection

ID	5	Module Name	Soothing Sound Selection					
Objective		To test the functionalities of selecting, adding, and deleting songs to be played.						
Name		Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass	
No	Prerequisites							
1	The application is installed correctly and launched successfully.							
3	User must have an active internet connection.							
4	User must be logged in to an existing account.							
Test Scenario		User wants to perform CRUD functions to the Soothing Sounds page.						
ID	Test Case		Input Data		Expected Result		Actual Result	Status

1	Navigate to the Soothing Sounds screen.	Tap on the “Soothing Sounds” button in the Dashboard screen.	Users are navigated to the Soothing Sounds screen.	As Expected	Pass
2	Play songs for a certain duration.	Select songs and duration from the available song selection, then tap “Play Song” button.	Song starts playing and timer begins countdown.	As Expected	Pass
3	Add songs to the song selection list.	Tap on the “Add Songs” button in the Soothing Sounds screen.	User is able to select mp3 audio files from their phone files.	As Expected	Pass
4	Delete songs from the song selection list.	Tap on the “Delete Songs” button in the Soothing Sounds screen.	User is able to delete songs from the available song list.	As Expected	Pass
5	Stop song from playing before the timer is done.	Tap on the “Stop Song” button on the Soothing Sounds screen after song and timer begins.	Song stops playing and timer resets.	As Expected	Pass
6	Verify song continues playing when “cancel” is selected during “Stop Song” confirmation.	Tap on the “Cancel” button when “Stop Song” button is selected.	Song and timer continue playing.	As Expected	Pass
7	Attempt to start timer without selecting any song.	• Timer: 3 mins • Song selection: none.	Error message indicating “Missing info. Please select both song and timer duration.”	As Expected	Pass
8	Attempt to play songs without selecting timer duration	• Song: Hollow.mp3 • Timer duration: -	Error message indicating “Missing info. Please select both song and timer duration.”	As Expected	Pass
9	Attempt to select songs from an empty list.	Tap on “Select Songs” in the Soothing Sounds screen.	Error message indicating “No songs. Please add songs first.”	As Expected	Pass
10	Attempt to delete songs from an empty list.	Tap on “Delete Songs” in the Soothing Sounds screen.	Error message indicating “No songs to delete”.	As Expected	Pass

Table 5.6 Unit Test Case for Rock Cradle

ID	6	Module Name	Rock Cradle					
Objective		To test the functionalities of rocking the cradle.						
Name		Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass	
No	Prerequisites							
1	The application is installed correctly and launched successfully.							
2	User must have an active internet connection.							
3	User must be logged in to an existing account.							
Test Scenario		User wants to test cradle rocking controls.						
ID	Test Case		Input Data		Expected Result		Actual Result	Status
1	Navigate to the Rock Cradle screen.		Tap on the “Rock Cradle” button in the Dashboard Screen.		User is navigated to the “Rock Cradle” screen.		As Expected	Pass
2	Set cradle rocking duration and speed.		• Select desired timer duration. • Timer: 2 mins • Speed: Medium		Cradle will start rocking based on the selected speed and duration, timer will begin countdown.		As Expected	Pass
3	Attempt to start rocking the cradle with insufficient instructions.		• Duration: 3 mins. • Speed: -		Error message indicating “Please select both duration and speed.”		As Expected	Pass
4	Attempt to stop rocking the cradle before timer stops.		• Duration 3 mins • Speed: Low		Cradle stops rocking, timer and speed resets.		As Expected	Pass
5	Verify cradle continues rocking when “Cancel” is selected during “Stop Rocking” confirmation.		Tap on “cancel” when “Stop Rocking” button is selected.		Cradle continues rocking at the selected speed until the timer stops.		As Expected	Pass

Table 5.7 Unit Test Case for Baby Alerts

ID	7	Module Name	Baby Alerts					
Objective		To test the functionalities of receiving real-time alerts.						
Name		Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass	
No	Prerequisites							
1	The application is installed correctly and launched successfully.							
2	User must have an active internet connection.							
3	User must be logged in to an existing account.							
Test Scenario		User wants to test receiving real-time alerts.						
ID	Test Case		Input Data		Expected Result		Actual Result	Status
1	Navigate to the Baby Alerts screen.		Tap on the “Baby Alerts” button in the Dashboard Screen.		User is navigated to the “Baby Alerts” screen.		As Expected	Pass

5.2.2 Integration Testing

Integration testing is a vital phase within functional testing that involves combining individually tested modules and evaluating them as a unified system. While unit testing ensures that each component functions correctly in isolation, integration testing focuses on the interactions between these components once they are integrated. The primary objective is to verify that different modules within the CradleSense system such as sensors, microcontrollers, cloud services, and the mobile application work together seamlessly and exchange data accurately.

This testing phase is crucial for detecting issues related to data flow, interface mismatches, and unexpected behaviour that may not be apparent when modules are tested independently. By validating the communication and coordination between components, integration testing ensures the overall system operates reliably, efficiently, and as intended, providing a stable and cohesive experience for end users. Table 5.8 below shows the integration test case for the CradleSense application.

Table 5.8 Integration Test Case for CradleSense App

ID	8	Application Name	CradleSense				
Objective		Ensure the app functions smoothly and each modules interact correctly					
<i>Name</i>	<i>Sheana Kasih Benedict</i>		<i>Date</i>	<i>4th June 2025</i>		<i>Status</i>	<i>Pass</i>

Test Scenario		Integration Testing for Sign Up and Login			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify Sign Up and Login integration.	New user details.	User signs up successfully can is able to login with new credentials.	As Expected	Pass
2	Verify error handling during Sign Up and Login.	Invalid user details.	User receives error messages accordingly.	As Expected	Pass
3	Verify Sign Up and Login flow redirects to Dashboard screen correctly	Valid user credentials.	User is redirected to Dashboard screen after successful sign up and login.	As Expected	Pass
4	Verify failed Login does not redirect to Dashboard screen.	Invalid user credentials.	User remains on Login screen with appropriate error message.	As Expected	Pass

Test Scenario		Integration Testing for Logout and Login			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify Logout redirects to Welcome screen.	Tap on “Logout” from the Profile Icon in the Dashboard screen or tap on “Logout” from the Settings Screen.	User is redirected to Welcome screen after logging out.	As Expected	Pass
2	Verify user can log back in after successful log out.	Valid user credentials.	User logs back in successfully.		

Test Scenario		Integration Testing for Arduino, Raspberry Pi and Firebase			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify LM53 temperature data is transmitted.	Analog temperature signal from Arduino.	Firebase shows the correct temperature value.	As Expected	Pass
2	Verify PIR sensor detects motion and updates firebase.	Motion detected by PIR motion sensor.	Motion alert is pushed to Firebase.	As Expected	Pass
3	Verify water sensor sends wet/dry state to Firebase	Water sensor reading.	Firebase field updates with "Wet/Dry".	As Expected	Pass

Test Scenario		Integration Testing for Raspberry Pi Camera and CradleSense App			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify live video feed loads.	Tap on "Live Video" button on the Dashboard screen.	Live video displays immediately.	As Expected	Pass

Test Scenario		Integration Testing for Firebase and CradleSense App			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify wet bed notification is shown in Baby Alerts screen.	Moisture level = Wet	Baby Alerts screen show "Wet bed detected!" notification.	As Expected	Pass
2	Verify cry alert is triggered by sound sensor.	Sound level above 600.	Baby Alerts screen show "Loud sound detected! Baby might be crying" notification.	As Expected	Pass
3	Verify restlessness alert is triggered by	Motion detected by PIR motion sensor.	Baby Alerts screen show "Baby is restless!" notification.	As Expected	Pass

	PIR motion sensor.				
--	--------------------	--	--	--	--

Test Scenario		Integration Testing for Dashboard and All Other Modules			
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Verify navigation from Dashboard to Live Video.	Tap on “Live Video” in Dashboard.	User can navigate to Live Video from Dashboard	As Expected	Pass
2	Verify navigation from Dashboard to Soothing Sounds.	Tap on “Soothing Sounds” in Dashboard.	User can navigate to Soothing Sounds from Dashboard	As Expected	Pass
3	Verify navigation from Dashboard to Rock Cradle.	Tap on “Rock Cradle” in Dashboard.	User can navigate to Rock Cradle from Dashboard	As Expected	Pass
4	Verify navigation from Dashboard to Baby Alerts.	Tap on “Baby Alerts” in Dashboard.	User can navigate to Baby Alerts from Dashboard	As Expected	Pass
5	Verify navigation from Dashboard to Settings.	Tap on “Settings” in in Profile Icon in Dashboard.	User can navigate to Settings from Dashboard	As Expected	Pass

5.3 Non-functional Testing

Non-functional testing refers to a category of software testing that assesses the overall operational characteristics of the system rather than specific functional behaviours. This type of testing focuses on evaluating aspects such as performance, reliability, usability, and user satisfaction. The main objective is to ensure that the CradleSense system not only functions correctly but also operate efficiently, remains stable under varying conditions, and provides a positive user experience. Non-functional testing verifies that the system can handle normal and peak loads, respond appropriately

to invalid or unexpected inputs, and meet user expectations in terms of ease of use and responsiveness.

5.3.1 Reliability Testing

Reliability testing is an essential part of non-functional testing in the CradleSense system, aimed at assessing the system's ability to operate consistently and correctly under various real-world and edge-case conditions. The main purpose of this testing is to ensure that both the hardware (such as sensors and actuators) and the mobile application can maintain stable performance even when faced with invalid, unexpected, or inconsistent inputs.

For CradleSense, reliability testing focuses on verifying that the system:

- Continues to function without crashing when sensors produce abnormal or fluctuating data,
- Gracefully handles errors such as disconnection from Firebase or delayed data from Arduino or Raspberry Pi,
- Maintains consistent app behaviour during long usage sessions or intermittent network conditions,
- Identifies and resolves any faults that could result in incorrect alerts, failed data uploads, or UI malfunctions.

This testing helps improve the stability, fault tolerance, and overall robustness of the smart cradle system, ensuring it can perform reliably in dynamic caregiving environments without causing unnecessary disruption or failure. Table 5.9 below shows the test case for reliability testing of the CradleSense application.

Table 5.9 Test Cases for Reliability Testing

ID	9	Application Name	CradleSense					
Objective		To test the reliability and stability of the mobile application.						
Name		Sheana Kasih Benedict		Date	4 th June 2025	Status	Pass	
No	Prerequisites							
1	The application is installed correctly and launched successfully.							
2	User must have an active internet connection.							
3	User must be logged in to an existing account.							
Test Scenario		Verify the reliability and stability of the mobile app under various conditions.						
ID	Test Case		Input Data		Expected Result		Actual Result	Status
1	Continuous usage test.		Use app continuously for 2 hours.		Runs smoothly without crashes or performance issues.		As Expected	Pass
2	Network connectivity test.		Disconnect and reconnect from the network.		Application resumes correctly.		As Expected	Pass
3	Firebase sync after network loss.		Disconnect and reconnect to the network.		Application resumes data sync correctly.		As Expected	Pass
4	Install/Uninst all test.		Install and uninstall application multiple times.		Installs and uninstalls without issue each time.		As Expected	Pass

5.3.2 Usability Testing

User Acceptance Testing (UAT) is a vital phase of non-functional testing where real users evaluate the system to ensure it meets their expectations and usage requirements. The core objective of UAT is to validate that the CradleSense application performs effectively from the user's perspective and provides a satisfactory experience in terms of functionality, responsiveness, and ease of use.

This process helps verify that the system satisfies both user and functional requirements, as documented during the planning and design phases. Furthermore, UAT allows for the identification of usability challenges or functional issues that may not have been detected in earlier

technical testing stages. By collecting feedback from real end-users, developers can refine the system before final deployment.

To conduct UAT, a group of 15 volunteer parents and caregivers with infants or toddlers were selected. Each tester received access to the CradleSense APK via Google Drive, along with a user manual and a structured Google Form questionnaire. Participants installed the app on their Android smartphones (minimum Android 5.0 or later), followed setup instructions, and then interacted with all key features of the application, including Live Video, Rock Cradle, Soothing Sounds, Environmental Monitoring, and Baby Alerts.

Following this, participants completed the questionnaire designed to assess various aspects of the application's usability and performance. The results from the survey were then analysed to determine overall satisfaction, identify any usability concerns, and gather suggestions for future improvements.

5.3.3.1 Analysis of User Acceptance Survey

The user acceptance survey was structured into four sections, namely User Demographics, Features Accuracy and Effectiveness, Features Usefulness and Consistency, and Overall Functional Experience. The feedback from 15 respondents provided meaningful insights into the real-world applicability and user-friendliness of the CradleSense system.

User Demographics

A majority (66.7%) were aged between 25–35, followed by 26.7% between 35–44, and the remaining 6.7% above between 18-24. This demographic aligns with the system's intended user base, which are caregivers responsible for infant care. Most of them does not have experience using a smart cradle system, making their feedback valuable in evaluating first-time usability.



Figure 5.1 UAT Survey User Demographics

Features Accuracy and Effectiveness

The majority of the participants (73.3%) reported that CradleSense's cry detection feature was accurate. Similarly, 60% of the participants agreed that the temperature and water sensor readings are somewhat accurately reflecting the actual environment, while the remaining 40% agreed the readings are accurate. For the live video feed, 73.3% of the participants agreed that the live video feed was clear during testing, while the remaining 26.7% fully agreed that it was consistently clear throughout the testing.

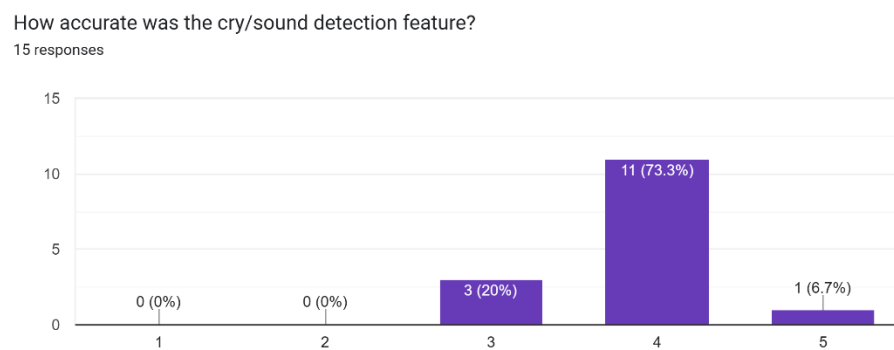
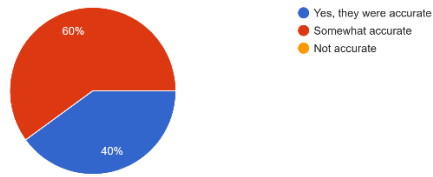


Figure 5.2 UAT Survey Features and Effectiveness on Cry Detection Feature

Did the temperature and water sensor readings reflect the actual environment?
15 responses



Was the live video feed consistently clear during testing?
15 responses

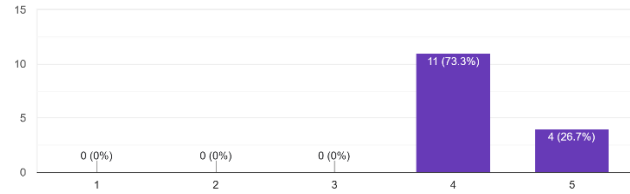


Figure 5.3 UAT Survey on Features and Effectiveness of CradleSense

One observation from this UAT is that there seems to be issues with the soothing sounds playing correctly. 73.3% of the participants agreed that in terms of accuracy, sometimes the songs selected will play longer or shorter than the intended time. Meanwhile, the remaining 26.7% had issues whereby the songs did not play at all.

Did the soothing sounds play correctly for the selected duration?
15 responses

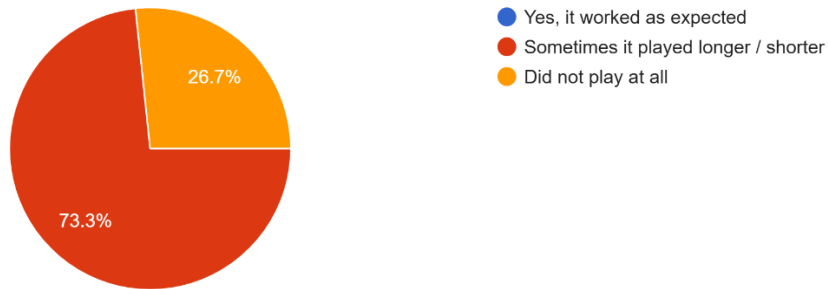


Figure 5.4 UAT Survey on Features and Effectiveness of CradleSense for Soothing Sounds Feature

Features Usefulness and Consistency

Feedback on core features was highly positive. 86.7% of the participants agreed that the sound detection alert was useful, while the remaining 13.3% strongly agreed on the usefulness of this alert.

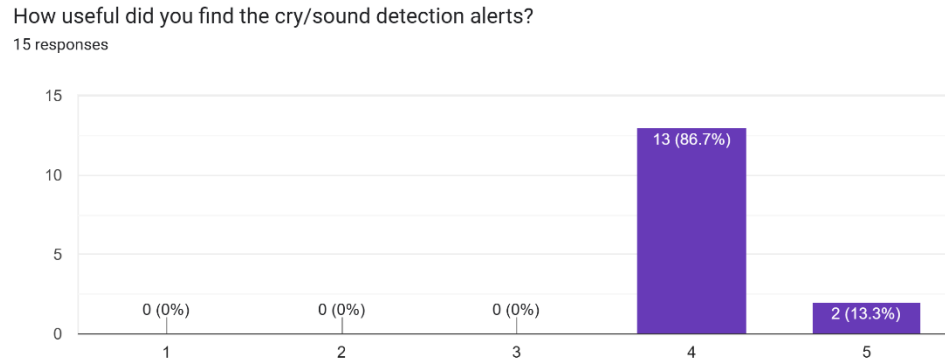
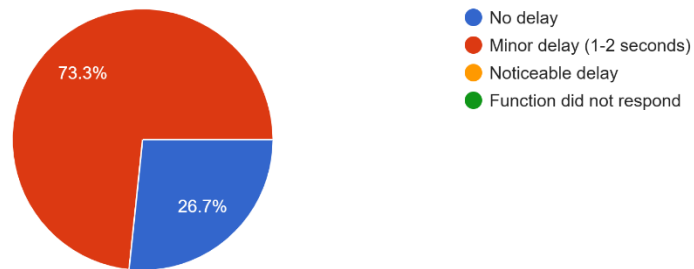


Figure 5.5 UAT Survey on Features Usefulness and Consistency

The majority of the participants (73.3%) noticed a minor delay of 1 to 2 seconds between the app command and hardware response. Meanwhile, the remaining 26.7% stated that they did not experience any delays.

Was there any delay or lag between the app command and hardware response? (e.g., rocking or playing music)?

15 responses



Overall Functional Experience

According to the participants, 73.3% agreed while the remaining 26.7% strongly agreed that the CradleSense app is effective in performing its intended functions. The same participants agreed on the same level that the app is useful for a parent or caregiver.

Overall, how effective was the CradleSense app in performing its intended functions?

15 responses

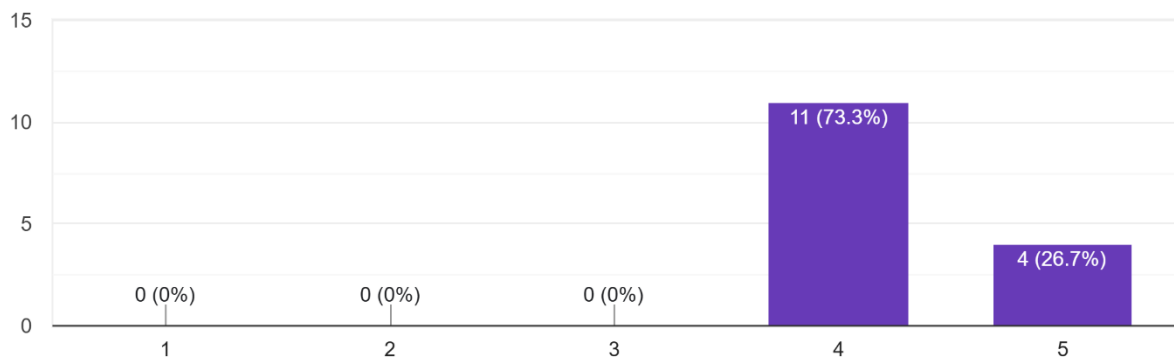


Figure 5.6 UAT Survey on Overall Functional Experience

How would you rate the usefulness of the app for a parent or caregiver?

15 responses

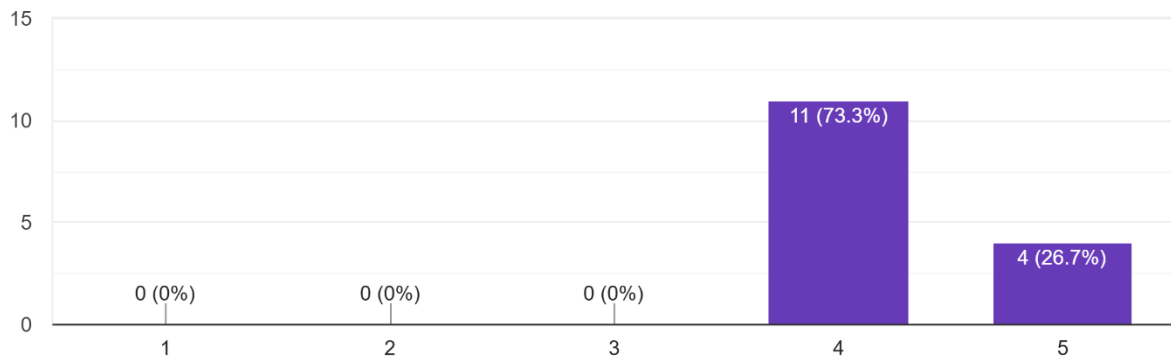


Figure 5.7 UAT Survey on Application Appropriateness for Parents or Caregiver

Overall, the usability testing confirmed that CradleSense is effective, user-friendly, and reliable. The application successfully delivers a seamless interface for real-time monitoring and remote cradle control. Positive responses from all users underscore the practicality of the system in supporting day-to-day infant care, especially for working parents or caregivers managing multiple responsibilities.

5.4 Discussion on Obtained Results

The CradleSense Smart Cradle system successfully fulfilled the majority of its defined objectives, particularly in terms of core functionality and overall system integration. The system's ability to monitor environmental conditions in terms of temperature and cradle wetness, detect baby motion and sound activity, provide soothing responses such as cradle rocking and lullaby playback, and stream live video through the Raspberry Pi was achieved through a well-orchestrated integration of hardware (Arduino Uno, sensors, actuators), Raspberry Pi (as gateway and cloud agent), and the Flutter-based mobile application connected to Firebase.

Despite these accomplishments, CradleSense is not yet fully optimised for real-world deployment or commercial scaling. A number of limitations and technical constraints emerged during system development and testing phases, particularly related to hardware limitations, responsiveness under various network conditions, and sensor reliability.

Sensor-based results were generally accurate in isolated unit tests, however, real-world performance varied based on several environmental factors. For example, the water sensor reliably detected moisture under controlled conditions, but its sensitivity to environmental humidity and material interference may lead to false positives or delayed detection in prolonged use. The LM35 temperature sensor provided consistent ambient readings, yet the inability to detect humidity, due to the absence or failure of the DHT11 module, limited the environmental monitoring scope originally intended.

In terms of interaction between hardware and cloud, the system performed effectively when pushing data from Arduino (via serial to Raspberry Pi) to Firebase. However, delays were occasionally observed in sensor data updates and actuation commands due to Firebase write latency and Raspberry Pi processing bottlenecks. These issues were more prominent during simultaneous operations such as live streaming and cradle rocking, suggesting the need for further performance optimisation or multi-threaded execution management on the Raspberry Pi.

The mobile application, developed using Flutter and integrated with Firebase services, offered a smooth user experience and allowed seamless navigation across modules including device management, account settings, live feed, and sensor status. Validation and error-handling mechanisms ensured data integrity during sign-up, login, and update operations. Nevertheless, email update processes were dependent on Firebase's verification protocol, which required user action and re-authentication, potentially disrupting user flow. Additionally, some updates (e.g.,

email address) required listeners to detect backend changes before reflecting them on the front-end, necessitating asynchronous logic and post-verification synchronization to maintain data consistency between Firebase Authentication and Firestore.

Key functional test cases such as motion detection triggering alerts, moisture detection updating cradle status, and successful playback of soothing sounds validated the individual components' operations. Integration tests further confirmed that system modules, Arduino hardware, Raspberry Pi gateway, Firebase backend, and Flutter front-end, worked cohesively as a distributed IoT ecosystem. However, real-time testing revealed network dependency as a critical factor, especially for users in low-connectivity environments. Any disruption in internet availability directly affects real-time data streaming, remote control functions, and cloud sync.

To summarise, the CradleSense system demonstrates high potential as a real-world baby monitoring solution, especially for busy or single parents. The project showcases the feasibility of integrating low-cost sensors and microcontrollers with modern cloud platforms and cross-platform mobile applications. However, the following limitations were observed:

- **Sensor Hardware Limitations:** Sensitivity of the water sensor and PIR sensor may vary with environmental conditions, requiring calibration and shielding.
- **Limited Edge Intelligence:** The system lacks onboard processing or AI-based anomaly detection (e.g., crying classification, movement recognition).
- **Firebase Delay & Sync Challenges:** Slight delays and inconsistencies in cloud syncing, especially during high activity periods.
- **Authentication Dependencies:** Email update flows require verification and re-login, which may confuse non-technical users.

- **Connectivity-Dependent:** Internet disruption leads to loss of remote monitoring/control, as system is not yet offline-tolerant.
- **Raspberry Pi Video Load:** Running real-time camera streaming and sensor communication concurrently may cause frame drops or latency.

In conclusion, CradleSense met the core expectations of a smart baby cradle monitoring prototype. While functional under ideal and stable conditions, enhancements are needed in areas such as sensor robustness, real-time performance, offline resiliency, and intelligent automation before the system is ready for full production or commercial deployment.

5.5 Summary

Testing plays a vital role in the system development lifecycle of the CradleSense Smart Cradle, ensuring that the final product functions accurately and meets the intended user needs. The testing process was categorised into two main types, which are functional testing and non-functional testing. Functional testing involved unit testing and integration testing, which focused on verifying that each individual component such as sensor modules, Firebase operations, and Flutter app interfaces operates as expected in isolation, and that all components interact seamlessly across the system. Non-functional testing was conducted to assess the system's reliability, performance, and responsiveness under real-world conditions. These tests evaluated how well the system maintains consistent behaviour during prolonged usage, responds to sensor inputs and commands in real time, and delivers a smooth user experience through the mobile app. Together, these tests ensured that CradleSense is not only functionally sound but also robust, user-friendly, and ready for real-world deployment with further refinement.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Introduction

This chapter presents the final reflection on the CradleSense Smart Cradle project. It discusses the key objectives achieved, the limitations and challenges encountered and proposes enhancements for future development. CradleSense was designed with the aim of addressing the emotional and logistical difficulties faced by parents, particularly single mothers, in caring for infants. Through the integration of Internet of Things (IoT) technologies and a mobile application, the project sought to provide a responsive, automated cradle system that enhances infant care by offering real-time monitoring and soothing mechanisms.

6.2 Objectives and Achievements

The CradleSense project set out to achieve several objectives, all of which were successfully met during the course of development. The first objective was to design a smart cradle capable of monitoring the baby's environment and condition. This was accomplished by integrating key sensors including a LM35 analog sensor for temperature, a sound sensor for cry detection, and a water sensor for wetness detection. These components provided real-time feedback on the baby's comfort and environmental status.

The second objective focused on implementing automated soothing responses based on sensor input. This was achieved using a servo motor to gently rock the cradle and a speaker system to play calming lullabies when crying was detected. The system was programmed to respond automatically, ensuring that the baby would be soothed even in the absence of immediate parental attention.

The third objective involved developing a mobile application to allow parents to monitor and control the cradle remotely. This was successfully completed using the Flutter framework and integrated with Firebase, enabling real-time notifications, status updates, and direct control of rocking and sound functions. The final major objective aimed to include a live video feed. This feature was implemented using the Raspberry Pi Camera Module, enabling parents to visually monitor their baby at any time.

These achievements collectively demonstrate that CradleSense fulfilled its goal of creating a responsive, supportive system for infant monitoring and care, using accessible and cost-effective technology.

6.3 Project Limitations and Challenges

Despite the successful implementation of the system, several limitations and challenges were encountered throughout the development process. One of the primary technical challenges involved the accuracy and reliability of sensor readings. For example, the moisture sensor occasionally produced inconsistent readings, especially when detecting wetness through fabric layers such as diapers. This limitation highlighted the need for either better sensor placement or the use of more sophisticated sensing technology.

Another significant constraint was the system's dependency on a constant power supply. Since the Raspberry Pi and connected components require uninterrupted power, the cradle is not fully portable in its current form. This restricts its usability in areas with unstable power sources or in outdoor scenarios. Additionally, the system's reliance on Wi-Fi for real-time updates and video streaming occasionally resulted in connectivity issues, such as delays or temporary disconnections, particularly in environments with weak or unstable internet coverage.

Hardware limitations were also encountered. The Raspberry Pi has a limited number of GPIO pins and processing power, which made it challenging to manage multiple sensors and a camera module simultaneously. Furthermore, the development of both the hardware and software components was conducted by a single individual, which posed constraints in terms of time, testing, and troubleshooting. Balancing these tasks individually extended the development cycle and increased the overall workload.

These challenges, while limiting certain functionalities of the prototype, provided critical insights into system design, hardware integration, and project management, which are valuable for future development and scalability.

6.4 Future Works

There is considerable potential for extending and enhancing the CradleSense system in future iterations. One significant improvement would be the integration of machine learning algorithms to classify different types of infant cries, such as cries caused by hunger, discomfort, or pain. This would enable the system to respond more appropriately and intelligently to the baby's needs.

Another key area for enhancement is power management. Incorporating a rechargeable battery system and optimizing the device's energy consumption would significantly increase its portability and usability in different settings. In terms of accessibility, expanding the mobile application to support iOS devices and integrating voice assistant capabilities such as Google Assistant or Siri could improve user convenience and inclusivity.

Health monitoring can also be extended by including additional sensors to measure vital signs such as heart rate or movement, thereby offering a more comprehensive safety system for infants. Moreover, implementing a cloud-based data dashboard would enable long-term storage

and visualization of baby activity, such as sleep patterns, environmental changes, and cry frequencies. This data could provide parents with deeper insights into their child's development and help health professionals identify trends if needed.

Lastly, improving offline functionality by incorporating Bluetooth or local fallback mechanisms would ensure that the cradle remains operational even when internet connectivity is lost. This would improve system reliability and allow parents to retain control during network outages.

6.5 Conclusion

In conclusion, the CradleSense Smart Cradle project has demonstrated the viability of using IoT and mobile technologies to automate and enhance infant care. The system successfully integrated multiple sensors and actuators with a user-friendly Flutter-based mobile application, offering parents the ability to monitor, soothe, and interact with their baby remotely. All core objectives were achieved, including real-time cry detection, cradle rocking, audio playback, environmental monitoring, and live video streaming with night vision. Although the project encountered several technical and resource-related limitations, these were managed effectively within the scope of a single-developer Final Year Project. The challenges faced offered valuable lessons in system design, time management, and hardware-software integration. CradleSense serves as a strong foundation for future research and development, and with further improvements, it has the potential to evolve into a reliable and scalable solution that supports the well-being of infants and the peace of mind of their caregivers.

REFERENCES

- Bahishti, A. A. (2022). The Vital Role of Research Methodology in Addressing the Research Questions. *International Journal of Methodology*, 1(1), 1–1.
<https://doi.org/10.21467/ijm.1.1.2016.29>
- Beres, J. (2022, September 6). *What Is Rapid Application Development (RAD Model)*. App Builder. <https://www.appbuilder.dev/blog/rapid-application-development>
- Bhat, B. S., Senthil, K., Jayashree, J., & Vasanth, K. (2024). SmartNest: An IoT-Based Smart Cradle System. *2024 Second International Conference on Advances in Information Technology (ICAIT)*, 1–7. <https://doi.org/10.1109/icait61638.2024.10690625>
- Bratsch-Hines, M. E., Carr, R., Zgourou, E., Vernon-Feagans, L., & Willoughby, M. (2020). Infant and Toddler Child-Care Quality and Stability in Relation to Proximal and Distal Academic and Social Outcomes. *Child Development*, 91(6), 1854–1864.
<https://doi.org/10.1111/cdev.13389>
- Chertoff, J. (2018, July 20). *What's the Average Baby Length by Month?* Healthline; Healthline Media. <https://www.healthline.com/health/parenting/average-baby-length>
- Chien, C. (2020, February 4). *What Is Rapid Application Development (RAD)?* Codebots.
<https://codebots.com/app-development/what-is-rapid-application-development-rad>
- Digital Policy Office of Hong Kong. (2024). *AN INTRODUCTION TO RAPID APPLICATION DEVELOPMENT (RAD)*.
https://www.digitalpolicy.gov.hk/en/our_work/digital_infrastructure/methodology/system_development/past_documents/rad/doc/g47a_pub.pdf
- Fauzi, M. A., Tribiakto, H., Moniva, A., Ilyas, I. K., & Utami, E. (2023). Systematic Literature Reviews on Rapid Application Development Information System. *Bulletin of Computer Science and Electrical Engineering*, 4(1), 57–64. <https://doi.org/10.25008/bcsee>

- Giddings, C. (2023, February 21). *Is the \$1,700 Snoo Smart Sleeper Bassinet Worth It? What to Know Before You Buy*. Wirecutter: Reviews for the Real World; Wirecutter.
<https://www.nytimes.com/wirecutter/reviews/snoo-smart-sleeper-what-to-know/>
- Hotchkiss, K. (2022, April 14). *Is the Snoo Worth the Money? A Sleep Expert's Review*. Sleep Wise Consulting. <https://sleepwiseconsulting.com/a-snoo-review/>
- Jabbar, W. A., Shang, H. K., Hamid, S. N. I. S., Almohammed, A. A., Ramli, R. M., & Ali, M. A. H. (2019). IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle. *IEEE Access*, 7, 93791–93805. <https://doi.org/10.1109/access.2019.2928481>
- Kamarul Ariffin, A. S., & Markom, A. M. (2024). Development of an IoT-Enabled Smart Baby Cradle for Enhanced Infant Monitoring and Parental Control. *Journal of Computing Research and Innovation (JCRINN)*, 9(2), 72–84.
<https://ir.uitm.edu.my/id/eprint/102792/1/102792.pdf>
- Kene, S. G., & Nishad, M. (2021). Impact of Working Mothers on their Children's Development. *International Research Journal of Education and Technology*, 1(5), 38–42.
- Meriem, C., Khaoula, M., Ghizlane, C., Asmaa, M. A., & Ahmed, A. O. T. (2020). Early Childhood Development (0 - 6 Years Old) from Healthy to Pathologic: A Review of the Literature. *Open Journal of Medical Psychology*, 09(03), 100–122.
<https://doi.org/10.4236/ojmp.2020.93009>
- Saude, N., & Vardhini, P. A. H. (2020). IoT based Smart Baby Cradle System using Raspberry Pi B+. *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing*, 273–278.
<https://doi.org/10.1109/ICSIDEMPC49020.2020.9299602>

Yee Theng, L., Abdul Rahman, R., & Ismail, S. B. (2022). Determinants of Depression among Malay Single Mothers Living in Community in Perak, Malaysia. *Malaysian Journal of Medical Sciences*, 29(2), 80–93. <https://doi.org/10.21315/mjms2022.29.2.8>

APPENDICES

Appendix A





Appendix B

Final Year Project - Survey on Smart Cradles for Infant Care

Greetings!

My name is Sheana Kasih Benedict (78525) and I am a fourth year Network Computing student at Univerisiti Malaysia Sarawak (UNIMAS), Kota Samarahan. I am currently working on my Final Year Project (FYP) to develop a **Smart Cradle designed to support single mothers in managing the challenges and automating aspects of infant care.**

The purpose of this survey is to better understand your experiences, challenges, and preferences related to infant care and existing cradle solutions. Your responses will provide valuable insights to guide the development of a cradle that addresses real-world needs.

The survey will take approximately **5-10 minutes** to complete. Your participation is completely voluntary, and your responses will remain **confidential and anonymous**, and will only be used strictly for the development of my FYP.

If you are ready to begin, click "Next" below. Your input is highly appreciated!

For any concerns regarding this survey, please reach out to me via:

- Email: 78525@siswa.unimas.my or sheanakb@gmail.com
- WhatsApp: 017 973 1309

* Indicates required question

Section 1: Demographics

This section is to analyse and understand demographics alongside the household background of the respondents.

1. What is your age? *

Mark only one oval.

- ☐ 18 - 25
- ☐ 26 - 35
- ☐ 36 - 45
- ☐ 46 and above

2. Which of the following best describes your current status? *

Mark only one oval.

- ☐ Currently not employed
- ☐ Part-time student
- ☐ Part-time employee
- ☐ Full-time employee
- ☐ Full-time employee & part-time student

3. What is your occupation? *

Mark only one oval.

- ☐ Housewife
- ☐ Teacher/Educator
- ☐ Entrepreneur/Business Owner
- ☐ Healthcare Worker
- ☐ Medical Professional
- ☐ Engineer
- ☐ IT Professional
- ☐ Freelancer
- ☐ Student
- ☐ Other: _____

4. What is your current work arrangement? *

Mark only one oval.

- ☐ Fully remote (working from home)
- ☐ In-office (working at the office full-time)
- ☐ Hybrid (mix of in-office and remote work)

5. What is your marital status? *

Mark only one oval.

- ☐ Single
- ☐ Married
- ☐ Single parent
- ☐ Other: _____

6. How many children are in your household? *

Mark only one oval.

- ☐ None
- ☐ 1
- ☐ 2 - 3
- ☐ 4 and above

7. What is the age range of your youngest child? *

Mark only one oval.

- ☐ 0 - 6 months
- ☐ 7 - 12 months
- ☐ 1 - 2 years
- ☐ 2 - 5 years
- ☐ 6 years and above

Section 2: Current Solutions

This section is to identify the current infant care solutions used by the respondents.

8. How often do you use baby-monitoring tools? *

Mark only one oval.

- ☐ Never
☐ Rarely
☐ A few times a week
☐ Daily

9. What baby-monitoring tools or methods do currently use? *

Mark only one oval.

- ☐ Baby monitor
☐ Smartphone apps
☐ None
☐ Other: _____

10. If you are using smartphone or baby-monitoring apps, please specify the app used:

11. Rate your satisfaction with these tools or methods *

Mark only one oval.

- 1 2 3 4 5
Ver: ☐ ☐ ☐ ☐ ☐ Very satisfied

12. How often do you use the cradle to soothe your baby? *

Mark only one oval.

1 2 3 4 5

Nev ☐ ☐ ☐ ☐ ☐ Always

13. What type of cradle are you currently or have experience in using? *

Mark only one oval.

- ☐ Spring cot *Skip to question 14*
- ☐ Basic cradle (no automated movements/features)
Skip to question 14
- ☐ Semi-automated cradle (e.g., features control panel is on the cradle)
Skip to question 20
- ☐ Fully-automated/Smart cradle (e.g., requires internet connection, able to control via mobile app) *Skip to question 26*
- ☐ No cradle used *Skip to question 31*

Thoughts on Current Solution : Basic Cradle and Spring Cot Users

This section is to gather further information on respondents who uses traditional and or spring cradles.

14. What are the main challenges you face with using the cradle? (Select all that apply)

Check all that apply.

- ☐ Time-consuming to operate manually (e.g., manually rocking the cradle)
- ☐ Ineffective at calming the baby
- ☐ Requires constant supervision
- ☐ Lack of additional features (e.g., automated rocking motion)
- ☐ Other: _____

15. Have you made any modifications to your traditional cradle for additional functionality? *

Mark only one oval.

☐ Yes

☐ No

16. If you answered **yes** to the previous question, please specify your modifications:

17. How satisfied are you with your current cradle? *

Mark only one oval.

1 2 3 4 5

Ver: ☐ ☐ ☐ ☐ ☐ Very satisfied

18. What do you believe would improve your current cradle experience? *
(Select all that apply)

Check all that apply.

☐ Automated rocking and soothing

☐ Remote control via a mobile app

☐ Real-time alerts for baby's condition

☐ Integration with smart home devices

☐ Advanced environment monitoring (e.g., temperature, moisture sensor)

☐ Other: _____

19. Would you consider upgrading to a fully smart cradle if it addressed your current challenges? *

Mark only one oval.

- ☐ Yes
☐ No
☐ Maybe

Skip to question 37

Thoughts on Current Solution : Semi-automated Cradle Users

This section is to gather further information on respondents who uses semi-automated cradles.

20. What features does your current cradle offer? (Select all that apply) *

Check all that apply.

- ☐ Automated rocking mechanism
☐ Soothing sounds or lullabies
☐ Timer for rocking duration
☐ Other: _____

21. What limitations have you experienced with your semi-automated cradle? (Select all that apply) *

Check all that apply.

- ☐ Limited functionality
☐ Inconsistent performance
☐ Requires frequent manual intervention
☐ High maintenance or repairs
☐ Other: _____

22. How satisfied are you with the features of your current cradle? *

Mark only one oval.

1 2 3 4 5

Ver: ☐ ☐ ☐ ☐ ☐ Very satisfied

23. What do you like most about your semi-automated cradle? *

24. What do you believe would improve your current cradle experience? *
(Select all that apply)

Check all that apply.

- ☐ Automated rocking and soothing
☐ Advanced environment monitoring (e.g., temperature, moisture sensor)
☐ Remote control via a mobile app
☐ Real-time alerts for baby's condition
☐ Integration with smart home devices
☐ Other: _____

25. Would you consider upgrading to a fully smart cradle if it addressed your current challenges? *

Mark only one oval.

- ☐ Yes Skip to question 37
☐ No Skip to question 37
☐ Maybe Skip to question 37

Skip to question 37

Thoughts on Current Solution : Fully-automated/Smart Cradle Users

This section is to gather further information on respondents who uses fully-automated or smart cradles.

26. What features does your current cradle offer? (Select all that apply) *

Check all that apply.

- ☐ Automated rocking mechanism
- ☐ Soothing sounds or lullabies
- ☐ Timer for rocking mechanism
- ☐ Environment monitoring (e.g., temperature and humidity, moisture sensor, motion sensor, sound sensor)
- ☐ Remote control via a mobile app
- ☐ Personalisation via a mobile app (e.g., able to adjust speed, duration, change songs)
- ☐ Other: _____

27. What limitations have you experienced with your smart cradle? (Select all that apply) *

Check all that apply.

- ☐ Inconsistent performance
- ☐ High maintenance or repairs
- ☐ Connectivity issues
- ☐ Unreliable sensor readings
- ☐ Inconsistent synchronisation with the mobile app
- ☐ Other: _____

28. What do you like most about your smart cradle? *

29. How satisfied are you with the features of your current smart cradle? *

Mark only one oval.

1 2 3 4 5

Ver: ☐ ☐ ☐ ☐ ☐ Very satisfied

30. What do you believe would improve your current smart cradle experience? *

Skip to question 37

Thoughts on Current Solution : Caregiving Without a Cradle

This section is to gather further information on respondents who does not use any cradle.

31. Do you face any challenges in soothing your baby without a cradle? *

Mark only one oval.

- ☐ Yes, it can be physically exhausting
- ☐ Yes, it takes a lot of time and effort
- ☐ No, I am satisfied with my current methods
- ☐ Other: _____

32. What alternative methods do you use to soothe your baby? (Select all that apply) *

Check all that apply.

- ☐ Holding and rocking manually
- ☐ Walking or pacing while holding the baby
- ☐ Playing soothing music or white noise
- ☐ Other: _____

33. How do you monitor your baby's safety and comfort without a cradle? *

34. Why have you chosen not to use a cradle? *

Mark only one oval.

- ☐ Prefer traditional methods of soothing
- ☐ Concern about safety of cradles
- ☐ Limited space at home
- ☐ Cost concerns
- ☐ Other: _____

35. Have you ever considered using a cradle for your baby? *

Mark only one oval.

- ☐ Yes, but decided against it
- ☐ No, I prefer other methods
- ☐ Maybe, if it met certain criteria
- ☐ No, I have never thought about it

36. What features would make you consider using a cradle? (Select all that apply) *

Check all that apply.

- ☐ Automated rocking and soothing
- ☐ Environmental monitoring features (e.g., temperature, motion, cry detection)
- ☐ Remote control via a mobile app
- ☐ Affordability and ease of use
- ☐ Real-time alerts for baby's condition
- ☐ Other: _____

Skip to question 37

Section 3: Technology Awareness and Preferences

This section is to understand the respondent's preferences and comfortability in using smart devices for infant care.

37. How comfortable are you in using mobile apps for baby monitoring? *

Mark only one oval.

- 1 2 3 4 5
-
- Not ☐ ☐ ☐ ☐ ☐ Very Comfortable

38. What type of notifications do you prefer for baby-monitoring devices? (Select all that apply) *

Check all that apply.

- ☐ Push notifications on phone
☐ Email alerts
☐ SMS alerts
☐ Other: _____

39. What concerns do you have about the use of smart devices in infant care? (Select all that apply) *

Check all that apply.

- ☐ Data privacy
☐ Cost
☐ Complexity
☐ Synchronization issues
☐ Other: _____

40. What devices do you own that can connect to smart systems? (Select all that apply) *

Check all that apply.

- ☐ Smartphone
☐ Tablet
☐ Laptop
☐ Other: _____

Section 4: Feature Preferences

This section is to understand the respondent's preferences when it comes to choosing smart cradles.

Rank the importance of the following features in a smart cradle:

41. Cry/Sound detection *

Mark only one oval.

	1	2	3	4	5	
Not	☐	☐	☐	☐	☐	Very important

42. Motion detection *

Mark only one oval.

	1	2	3	4	5	
Not	☐	☐	☐	☐	☐	Very important

43. Temperature and humidity monitoring *

Mark only one oval.

	1	2	3	4	5	
Not	☐	☐	☐	☐	☐	Very important

44. Automated rocking mechanism *

Mark only one oval.

	1	2	3	4	5	
Not	☐	☐	☐	☐	☐	Very important

45. Wet diaper detection *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

46. Lullaby/White noise playback *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

47. Remote control for smart cradle features via a mobile app *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

48. Camera for live monitoring *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

49. Real-time alerts via mobile application *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

50. Cloud integration to store data for tracking baby's habits over time *

Mark only one oval.

1 2 3 4 5

Not ☐ ☐ ☐ ☐ ☐ Very important

51. What customisation options would you prefer for the smart cradle? *

Section 5: Budget and Purchase Intent

This section is to understand the preferred cost of a smart cradle and the factor affecting a successful smart cradle purchase.

52. What is the maximum amount you are willing to spend on a smart cradle? *

Mark only one oval.

- ☐ Below RM500
- ☐ RM500 - RM1000
- ☐ RM1000 - RM1500
- ☐ More than RM1500

53. Would you consider a subscription-based service for advanced features? (e.g., cloud storage, AI analytics)? *

Mark only one oval.

- ☐ Yes
- ☐ No
- ☐ Maybe

54. What factors would influence your decision to purchase a smart cradle? *

Mark only one oval.

- ☐ Cost
- ☐ Features
- ☐ Product operational lifespan
- ☐ Other: _____

Section 6: Feedback and Suggestions

This section is for respondents to provide feedback regarding smart cradles.

55. What specific concerns do you have about smart cradles? *

56. What additional features or improvements would you suggest for a smart cradle? *

This content is neither created nor endorsed by Google.

Google Forms