College of Computing & Informatics (CCIT)

SENIOR PROJECT-I REPORT

<Baby Monitoring App Based on IoT & AI>

Author(s):

S170305172 Reem Fahad Alsaqabi

S170332304 Sarah Khalid Bin Jadeed

S170041370 Hanan Ahmed Alghamdi

Project Supervisor:

Dr Ashwaq Alhargan

Baby Monitoring App Based on IoT & AI

By:

Reem Fahad Alsaqabi

Sarah Khalid Bin Jadeed

Hanan Ahmed Alghamdi

Thesis/Project submitted to:

College of Computing & Informatics, Saudi Electronic University, Riyadh, Saudi Arabia.

In partial fulfilment of the requirements for the degree of:

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Dr Ashwaq Alhargan

Project Supervisor

Project Committee Chair

ABSTRACT

This project provides a child monitoring system for parents so that they can ensure proper care and safety for their children.

With the circumstances of life today, many parents face difficulties in monitoring their children, even inside the house.

In this system, we provide parents with the ability to monitor their children through a video camera and different sensors to monitor external conditions such as increased body temperature, crying and send alerts and notifications to the app.

Furthermore, this system helps parents with vision and hearing disabilities by providing them with some additional features that enable them to interact with the application interface.

Our professional growth has improved by this project and it helped us learn modern technologies such as creating and programming applications by Android Studio and linking them with IoT and AI. We applied an approach in which we linked the application with a microcontroller connected to a camera and sensors to attain the required functionalities and utilize them to serve our project.

The ease of use and merging of different technologies and harnessing them to serve people of various occupational and health conditions is the most important characteristic of our system.

DEDICATION

We dedicate this work to everyone who helped, supported, and encouraged us to complete this project.

And we would like to thank Dr Ashwaq Alhargan for her guidance and supervision in the fulfilment of the project. She helped our group to apply what we had learned and to learn new technologies that would help us develop our project. We truly appreciate her support and her valuable feedback and critics of our work. We are grateful for the motivation received by our family and friends towards the completion of our project.

PREFACE

The Internet is the technology that changed our entire world, and it allowed us to communicate with each other.
What is happening today is that the number of mobile devices and computers is about 50 billion devices connected to the Internet.
And these devices are not just computers, they are things, and to know what we will say that all wearable things can interact with humans and others we will talk about them.
In 1999, Kevin Ashton coined the word "Internet of Things", during a presentation he gave in a company he was working for and arrived at the time to the identification card technology using radio waves, or RFID, by which you can know the location and send a signal from the body carrying the card To any computer by sending a special signal, which is the first step on the road to the Internet of Things.
The Internet of Things is simply giving the ability to connect to the Internet for various devices and things, in addition to the ability to communicate with each other through the network and cloud services, to exchange information so that each part can perform its task and the function it performs.
Devices and things include everything you can think of, from televisions, refrigerators, surveillance cameras to clothes, glasses, and shoes, all the way to the body parts! Imagine a world with all these things connected to the Internet and exchanging different information. So, the important part is the sensors that we will provide to be able to collect information from its surroundings, whether it is the physical environment or the human body.
The embedded sensors allow the mobile user to gather more accurate data rapidly, monitor changes in the ambient environment, and visualize phenomena of the real world. Thus, accelerate processes and increase productivity.
Artificial Intelligence of Things (AIoT): is the use of artificial intelligence (AI) technologies with the infrastructure of the Internet of Things (IoT) to achieve more efficient Internet operations and improve data management and analysis. Artificial intelligence can be used to transform IoT data into useful information to enhance decision-making processes and improve human-machine interactions.
In this project, we will apply these modern technologies and use artificial intelligence with the Internet of things in an effective way with smart device application for monitoring the baby's condition.

REVISION HISTORY

Name	Date	Reason for Changes	Version
Hanan Alghamdi	01/10/2020	Spelling and grammatical mistakes	V1
Sarah Bin Jadeed	19/10/2020	Verify system requirements in the use cases.	V2
Reem Alsaqabi	12/11/2020	Review the whole file and correct mistakes.	V3

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Project Background/Overview:

This is an application to monitor the child and help parents take care of their children because it is connected to a camera in the child's room

We found many applications related to this topic, but these applications lack some features that may be important for some people with disabilities, as they need additional features that help them take care of their children.

The application helps parents pay attention to their children during their work hours or when they are busy at home by sending alerts during abnormal situations.

1.2 Problem Description:

It is difficult for parents to take care of the child and work at the same time inside the house with a feeling of comfort and lack of anxiety, and it is difficult to monitor the child all the time and monitor his temperature, we found a way to help parents using modern techniques to make them work comfortably and more reassuring about their child by observing the child and quickly discovering abnormal situations.

1.3 Project Scope:

After its implementation, the application will be available in the application stores for smartphones, and the person only needs to buy the device.

1. Developing software to help parents monitor the child's condition.
2. Send alerts based on the child's condition.
3. Sending alerts to the application in case the child cries or has a high temperature.
4. The application provides solutions for people with disabilities.

1.4 Project Objectives:

The goal of the project is to create a child monitoring system in a way that links a mobile application with a hardware device by using IoT (Internet of Things) capabilities. Via

- Connecting the microcontroller to the application
- Using many sensors that give us a readout of certain things, such as the child's temperature and abnormal positions
- Discovering the child's abnormal conditions and sending notices to the parents to know these conditions.
- Using a video camera to watch and observe the child through the application
- There will be an ease of use of the app for specific users.
- Ease of access and use of the project.

1.5 Project Structure/Plan:

First, after the problem has been identified and described, the scope of the project and the objectives of the project have been determined, we will create a structure or plan for all activities that will be implemented in the project as illustrated in Table 1.1, then the similar ideas will be identified and evaluated, after which the appropriate methodology for the implementation of the project will be determined and then the system is analyzed by presenting the product's features, functional and non-functional requirements, and then We will make analysis models, and after the analysis stage, the system will be designed by many diagrams, then at the end of the design stage, we will discuss all the previous activities and write the conclusion and mention the references. Figure1.1 shows the Gantt chart for the project plan. The Arrow Diagramming Method (ADM) and the Precedence Diagraming Method (PDM) for the project plan are shown in Figure1-2 and Figure1-3 respectively.

	Task Name	Duration	Start Date	End date	Predecessor	Assigned To
1	1: INTRODUCTION	13 days	06/09/2020	22/09/2020		All team members
2	1.1ProjectBackground/Overview:	5 days	06/09/2020	10/09/2020		Reem
3	1.2Problem Description:	1 day	13/09/2020	13/09/2020	2	Reem
4	1.3Project Scope:	1 day	14/09/2020	14/09/2020	2	Hanan
5	1.4Project Objectives:	1 day	15/09/2020	15/09/2020	4	Hanan
6	1.5Project Structure/Plan	5 days	16/09/2020	22/09/2020	4-5	Sarah
7	2: LITERATURE REVIEW	7 days	23/09/2020	01/10/2020	6	All team members
8	3: METHODOLOGY	3 days	04/10/2020	06/10/2020	6	All team members
9	4: SYSTEM ANALYSIS	9 days	07/10/2020	19/10/2020		All team members
10	4.1Product Features	1 day	07/10/2020	07/10/2020	8	Reem
11	4.2Functional Requirements	3 days	8/10/2020	12/10/2020	8	Reem
12	4.3Non-functional Requirements	1 day	13/10/2020	13/10/2020	8	Sarah
13	4.4Analysis Models	4 days	14/10/2020	19/10/2020	8	Hanan
14	5: SYSTEM DESIGN	11 days	20/10/2020	03/11/2020		All team members
15	5.1 Component Diagram	2 days	20/10/2020	21/10/2020	13	Reem
16	5.2 Deployment Diagram	2 days	22/10/2020	25/10/2020	13	Hanan
17	5.3 Design Level Sequence Diagram	2 days	26/10/2020	27/10/2020	13	Sarah
18	5.4 Class Diagram	2 days	28/10/2020	29/10/2020	13	Reem
19	5.5 Entity-Relationship Diagram	3 days	01/11/2020	03/11/2020	13	Hanan
20	6: DISCUSSION & CONCLUSION	3 days	04/11/2020	08/11/2020		All team members
21	6.1Discussion	2 days	04/11/2020	05/11/2020	19	Sarah-Hanan
22	6.2Conclusion	1 day	08/11/2020	08/11/2020	21	Reem
23	7: REFERENCES	2 days	09/11/2020	10/11/2020	7	All team members
24	8: GLOSSARY	2 days	11/11/2020	12/11/2020	22	All team members

Table 1.1 Project Plan

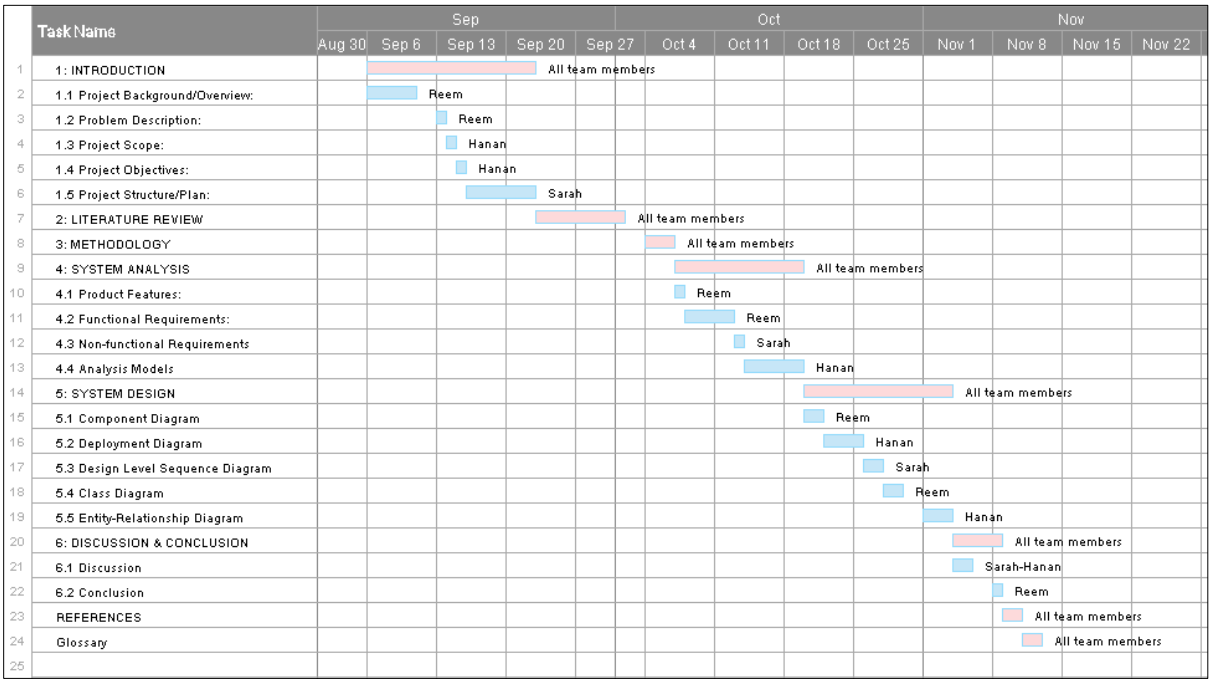

Figure 1-1Gantt chart

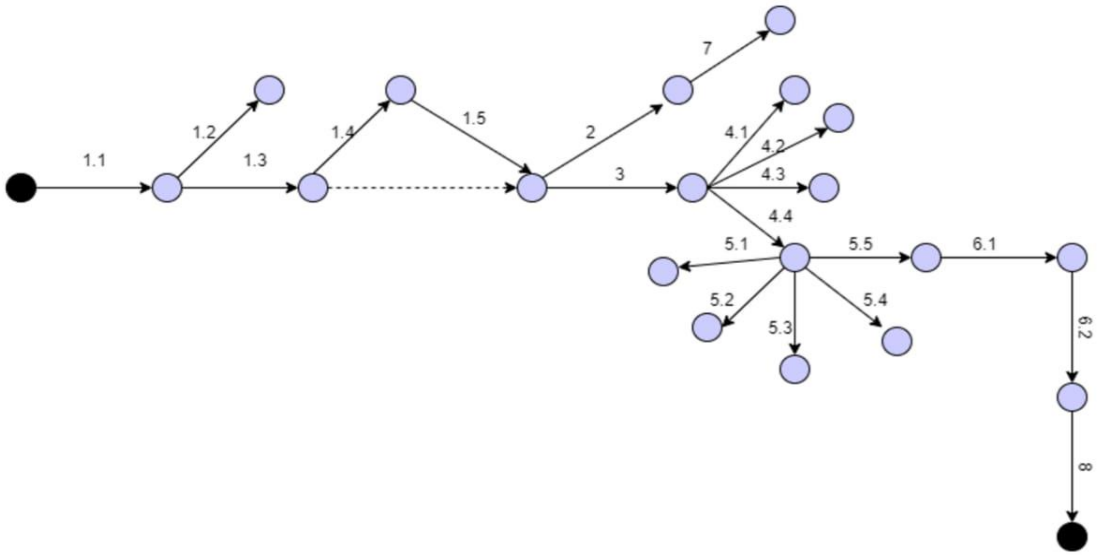

Figure 1-2 ADM Diagram.

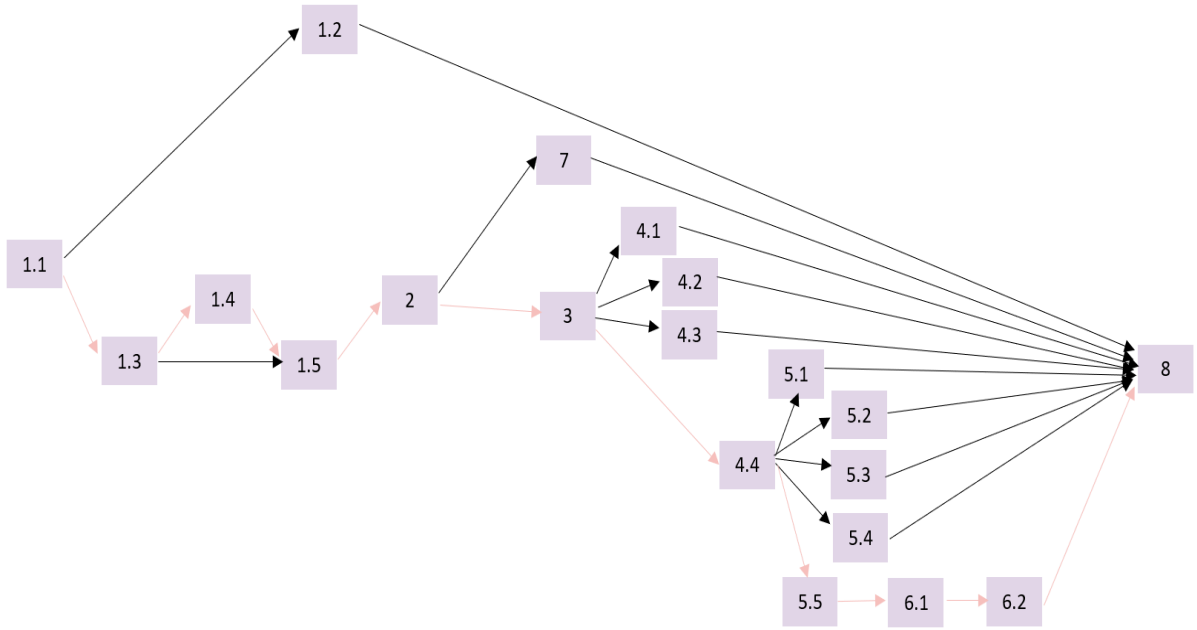

Figure 1-3PDM Diagram

CHAPTER 2: LITERATURE REVIEW

Baby monitoring projects are widespread, especially recently, and there are many works of literature related to this topic and other topics also related to our project, such as artificial intelligence, the Internet of things and others. In this project, we propose a system to monitor the vital signs of babies and their overall state to enable parents and caregivers to intervene quickly and take appropriate and timely action. By tracking vital signs and early intervention, child and newborn mortality will decrease dramatically to 50 per cent [1].

In one of the researches that talk about the suffocation of newborns while sleeping, technology and physical and biological standards were used for immediate measurements and alerts, by using sensors for the positions of the child. It reduces as much as possible the sudden death of the infant. These devices are electronic wearable products and a surveillance camera connected to a smartphone [2].

Too often, disabled parents are excluded from research agendas that emphasize the social and environmental context in which parenting occurs. The World Health Organization (WHO) estimates that the number of people with visual disability is at least 1 billion globally [3]. And around 466 million people have disabling hearing loss worldwide estimated that over 900 million people will have disabling hearing loss by 2050 [4].

In this project, we give significant importance to parents with disabilities. Unlike many of the existing monitoring applications, this application provides features to help disabled parents doing their parenting role. The feature of detecting the child's crying has become available and among the companies that provided this feature is the Samsung Galaxy S5, where when the child cries, the parents are alerted by shaking the mobile phone or by the flashlight available in the mobile phone when a sound is detected baby crying [5].

We availed of some technologies to accomplish this project such as artificial intelligence. Artificial Intelligent and ML will dramatically improve therapy, medication, screening & prediction, forecasting, contact tracing, and the production of drugs/vaccines, and decrease human intervention in medical practice [6].

What is artificial intelligence? This science has spread a lot, especially in recent period, we must know at the beginning that artificial intelligence is the intelligence of machines and not like the intelligence of humans and animals, but rather a simulation of it, it is the behaviour and certain characteristics of electronics and computer programs. These behaviour and characteristics mimic Human mental capacity. Computer science defines artificial intelligence as an intelligent device that studies the environment around us and takes measures that increase its chances of successfully achieving its goals. It is also defined as the system's ability to correctly interpret all external data and learn from it and use the knowledge to achieve specific goals [7].

Artificial Intelligence (AI) advances combine intelligent decision-making and reasoning algorithms with data from various sources that allow rapid data processing and complex data analysis [8].

There are many limitations instead of just developing the next generation of artificial intelligence technology. Its goals are to develop a new concept for general-purpose intelligent cognition technology called "Beyond Artificial Intelligence". Specifically, they are also planning to develop an intelligent educational model called "brain intelligence (BI)" that generates new ideas about events without experiencing them using artificial life with visualization function [9]. In this project, we used one of the applications of AI which is IBM WASTON. This is a cognitive computing system capable of answering questions in natural languages. Human knowledge can greatly understand this and has many properties that make it also able to answer related questions more effectively to unleash Watson's full power. We need to train their peers in many processed question-and-answer pairs [10]. There have been a lot of developments and progress in this site as the company announced a new development that happened. Today, IBM announces many exciting changes and additions to IBM Watson Discovery. Within the Watson Discovery Premium plan, users can now experience a new user interface, a heuristic experience to help users quickly start using Watson Discovery for their use case, and many new features including content exploration and ongoing developments [11].

We must define the Internet of things, which is a phrase: many of the physical things that we can reach through the Internet, the Internet of things appeared as a wave from the development of the Internet and it is the third wave [12]. In 2022, the global Internet of Things (IoT) spending is expected to hit 1.1 trillion U.S. dollars [13]. Numerous objects, such as sensors, cars, houses, and appliances, can be connected to the Internet through the Internet of Things (IoT), which enables users to exchange information, data, and resources [14]. In the IoT model, the term 'things' includes the virtual environment (entities, cyber activities, cyber events, and services) as well as the physical world (objects, behaviours, tendencies, and physical events) [15]

The large amounts of data generated by IoT devices give us information about many things such as the temperature in the motors or the reading from a smart meter. All these IoT data must be collected and stored, and then this evidence is analyzed and utilized in several aspects. Also, artificial intelligence takes this data and enters it into artificial intelligence systems that will benefit from this data and use it to make different predictions [16].

Physiological signals have been detected, measured, and processed using various technological formats, such as wearable sensors, with the assistance of advanced technologies. However, they often involve physical contact with the skin surface [17].
We found a project that is using a device to monitor the temperature of the child's body, but it is not comfortable for children to use, and we also found research specialized in monitoring the temperature of the infant talking in the design of a device that is placed on the child's foot and giving a warning by using a wireless connection with the parents, the Blynk application was used with him. We think that this research is good, but a little annoying for the child, as the child may be upset that something is touching him [18]. Therefore, we used a non-contact temperature sensor to provide more comfort and healthy sleep for the child.

Human face recognition technology is a technique like other technologies that change the world and the way we live in this method. This method helped a lot in many things that benefit the human being, for example. Facial recognition was used in monitoring, for example, monitoring children, and also systems that track criminals and identify fugitives. It has also been used in very many matters. All of them are in the interest of people and their safety [19].
Face recognition and facial expression libraries are available, and they are simple offices. These libraries were developed using deep learning, and their accuracy is often great. We found a project (CCBeBe) that used these two features in its project to monitor the child so that it sends notifications to parents based on artificial intelligence when crying and recorded some videos [20]. So far, the AI specialists did not find a very precise facial recognition API for the infant.
One of the researches was on developing a special system to automatically detect pain in a child, by analyzing images using many algorithms to analyze the child's facial features. Using machine learning to determine whether he is crying, or if the crying is caused by pain. This is done through a baby monitor. The algorithms used have a high correctness rate, despite the presence of the error rate [21]. We believe that if high-quality hardware were also added, the results would be better.
Microcontrollers were used in the project "Intelligent Baby Behavior Monitoring using Embedded Vision in IoT for Smart Healthcare Centers", in which they used the Internet of Things, and they moved away from wearable devices. In this project, the technique of behaviour analysis was used. When any abnormal behaviour is noticed, notifications are sent to the device

associated with it. There are risks in this project in terms of that wrong notifications may be sent because the behaviour of the child, especially the infant, is sometimes incomprehensible [22].
There are many applications in the apps stores that specialize in monitoring children, such as:
1- DORMI is an application that relies on monitoring a child through the camera of another smartphone device. This application provides a live broadcast for child and child monitoring. One of the features that we liked about this application is the encryption that it uses. It is good encryption as all the data that is exchanged is encrypted using encryption algorithms [23]. But it may be that one of the most prominent disadvantages of this application is that it depends on another device for monitoring, and the efficiency of this method is less than the camera that includes sensors and standards for the superiority of the phone. We must also look at the comparison in the battery, and the night vision ability, which even if it is available, is by using libraries, but in cameras, it is using infrared.
2- Luna -baby monitor app provides a feature to monitor babies and turn the iPhone or iPad into a monitoring device in an unlimited range. This app sends an alert to parents when the baby is crying. It offers many benefits, including listening to podcasts or watching live videos. Mobile phone battery consumes much less than similar applications [24].
*3- Baby Monitor 3G*is a baby monitor application that provides many features, including detection of all sounds in the child's room and provides a night light that makes parents can see their children all night. Baby Monitor 3G is already in your pocket at a tenth of the cost of a traditional monitoring system and it works everywhere [25].

After reviewing these applications and many others, we are looking forward to adding many features that are not available in these applications and obtain an optimal application that provides facilities for parents especially those with disabilities.

CHAPTER 3: METHODOLOGY

- **3.1 Software Development Model**

For the Baby Monitoring Application, the proper methodology would be the ***Incremental*** process model. However, since the whole project is extended on two courses where the implementation phase will be conducted later on next course "senior project 2, " the team have to follow the approach defined in the template and delay the implementation and the iterative releases or repeat planning and design later after the first increment that has the core functionality.

An ***Incremental*** process model is an approach that combines features from the waterfall methodology and agile, as illustrated in Figure 3-1. It is in middle between the upfront planning and the systematic nature of the waterfall, and the flexibility of agile. In the ***Incremental*** process, the waterfall phases are repeated iteratively for every function of the software one at an increment.

The project team has chosen the ***Incremental*** approach due to the numerous functionalities and the different APIs used in the Baby Monitoring application. The chosen approach allows adding and validating these functionalities one by one. It also enables generating the core software and get the user's feedback early in the SDLC which saves much time and effort.

The ***Incremental*** is the most suitable approach for this project since some of the other approaches do not permit flexibility and not adaptable for continuous updates. Moreover, implementing (coding) all functionalities of the software at one need a very high-level technical experience and more time.

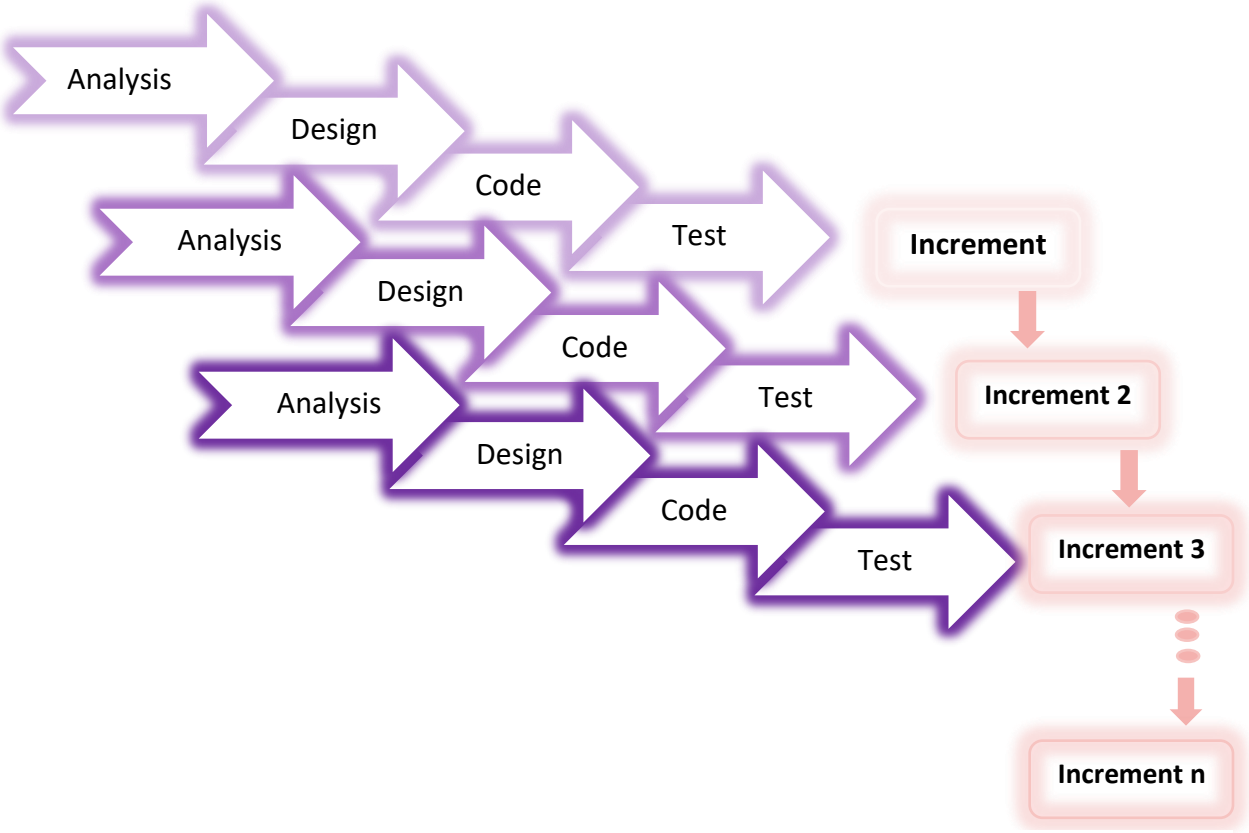

Figure 3-1Incremental Model Processes

- **3.2 Development Environment**

    - **3.2.1 Software Android, SDK**

Android Studio is a completely new and integrated development environment, recently launched by Google for the Android operating system. It is designed to provide new tools for application development and to provide an alternative to Eclipse, the IDE most in use today. It can provide a unified environment where you can build apps for Android phones and tablets, Android Wear, Android TV and Android Auto. Structured code modules allow you to divide your project into functional units that you can create, test and debug independently. We choose this program in our projects because we gained background about it at the university and we believe that we can develop an application for our project using Android Studio. Android Studio has several versions, and the latest version was released on September 23, 2008

Its latest version is Android 10 which was released on September 3, 2019. Our project works on all modern versions.

Android Studio contains many advanced features that made us decide to use it in our project to create an application on it, including that when you start creating a project on Android, the overall structure of the project will appear with all the files in its directory. This provides flexibility and smoothness in the creation process. The advantage is that it allows us to monitor and watch any visual changes we make to the application in real-time, and it also allows us to know how our application looks on different Android devices. Android Studio also features a direct and powerful development environment that makes many programmers work on it, especially as it allows performance testing on various devices It also features an editor that has many features that speed up the development of your application.

    - **3.2.2 Arduino IDE**

Open-source Arduino Software (IDE) is the program that will be used to program the microcontrollers. Among its advantages is the ease of writing programming commands through it, and after that, we can download it to the microcontroller that we will use.

The Arduino IDE contains a text editor for writing code, message area, command button bar and a series of menus, which are connected to a microcontroller to raise the program and communicate between the program and the device. We will be working on ARDUINO version 1.8.13.

- **3.2.3 Hardwar**

In this project, we will use ESP32-CAM that shown in Figure 3-2, it is a consumption camera module (OV2640 camera) he is a small size, Cheap price, and low power

ESP32-CAM specification

Processor: ESP32-D0WD

WIFI module: ESP-32S

Built-in Flash: 32Mbit

RAM: Internal 512KB + External 4M PSRAM

Security: WPA/WPA2/WPA2-Enterprise/WPS

ESP32-CAM Features:

ESP32-S module supports WiFi + Bluetooth

OV2640 camera with flash

Supports WiFi video monitoring and WiFi image upload

ESP32-CAM Uses or applications:

ESP32-CAM is used for IoT applications for example:

We can through it wireless monitoring

Wireless QR recognition

Face recognition

And also smart agriculture

And many other applications

Figure 3-2ESP32-CAM

We will also use many sensors. We may not be accurate in our choice now, because many developers and designers face difficulties or challenges in the implementation stage, but so far, we have found that the most suitable for our project:

MLX90614-DCI non-contact Temperature Sensor

The temperature sensor as shown in Figure 3-3is a non-contact infrared temperature sensor unit, and it is characterized by being long-range so we can use it to measure the child's temperature from afar. Its accuracy is higher the closer we are to the surface of the object to be measured, and its range may reach 50 cm.

Figure 3-3 MLX90614-DCI non-contact Temperature Sensor

PIR motion sensor:

We will use the PIR motion sensor as shown in Figure 3-4 to detect the child's movement, this sensor is used in smart homes and many robots.

Figure 3-4 PIR motion sensor

Digital Sound Detector Sensor Module:

We will also use a sound sensor as shown in Figure 3-5 to detect the sounds of the child and we may define the characteristics of the child's crying particularly as well.

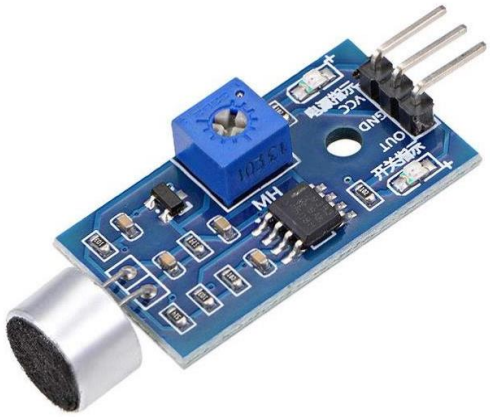

Figure 3-5 Sound Detector Sensor

CHAPTER 4: SYSTEM ANALYSIS

4.1 Product Features:

1. The product is software linked to the smart microcontroller camera.
2. If a parent wants to monitor their babies, they can place a camera near them, download the app, and monitor them and watch the real-time condition of their babies.
3. It can also monitor the baby's temperature, motion, and monitor whether they are crying or not.
4. The application provides settings to make the application accessible to those who have hearing disabilities by making the alerts flash or vibrate.
5. Use a screen reader to serves blind parents by reading the baby's condition and temperature for them.
6. Parents can request receiving a report of their baby's condition by e-mail.

4.2 Functional Requirements:

Functional requirements (concerning the production features identified in the section above) are expressed as use-cases as follows:

- UC-1 **Sign-in** illustrated in Table 4-1.
- UC-1.1 **forget password** illustrated in Table 4-2.
- UC-2 **Sign-up** illustrated in Table 4-3.
- UC-3 **Check baby condition** illustrated in Table 4-4.
- UC- 3.1 **Start (Chatbot)** illustrated in Table 4-5.
- UC- 3.2 **Camera** illustrated in Table 4-6.
- UC-4 **Modify app's settings** illustrated in Table 4-7.
- UC-4.1 **Modify Notifications Setting** illustrated in Table 4-8.
- UC-4.2 **Chang Password** illustrated in Table 4-9.
- UC-4.3 **Log-out** illustrated in Table 4-10.

Table 4-1 Sign-in Use-Case

Identifier	UC-1	
Purpose	To Sing-in to the system.	
Priority	High	
Pre-conditions	Existing user (Already have an account).	
Post-conditions	The user will be signed into the system.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Enter Username	
2	Enter password	
4	Click Ok	check if match
5		Open the system
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click sign up	Move from UC-1 to UC-2
2	Click forget password	Move from UC-1 to UC-1.1
3	The user entered an invalid Username or password.	Error message.

Table 4-2 Change Password Use-Case

Identifier	UC-1.1	
Purpose	To provide a new password	
Priority	High	
Pre-conditions	Existing user(Already have an account)	
Post-conditions	A new password is received	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Enter device number	Check if correct or match
2	Enter Email	Check if correct or match
3	Click Ok	Send a new password to the user's email
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	The user entered an invalid device number or Email.	Error message

Table 4-3 Sign-Up Use-Case

Identifier	UC-2	
Purpose	To register the user and save his/her data to the Users database.	
Priority	High	
Pre-conditions	Select (sign-up) button from the **sign-in** screen.	
Post-conditions	Registration form submitted and user profile created	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Fill in the registration form.	Validate values.
2	Submit form	Create a new profile.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	The user entered Invalid values or information.	The system requires valid values or information.

Table 4-4 Check baby's condition Use-Case

Identifier	UC-3	
Purpose	To access the home page and check the baby's condition.	
Priority	High	
Pre-conditions	Log in and configure the hardware.	
Post-conditions	Get access to the home page.	
Typical Course of Action		
Actor Action		**System Response**
The user signs in and gets access to the home page.		Open a live video from the connected camera.
The user hits the (Read Temperature) button from the **Options** menu.		Read current temperature from the connected sensor
The user hits (Condition Report) button from the **Options** menu.		Read the baby's condition from the connected hardware
The user hits the (Settings) button from the **Options** menu.		Open Settings page (UC-4).
The user hits the (Ask me) button from the **Options** menu.		Start chatbot API (UC- 3.1)
The user closes the (Ask me) window.		Back to (Home) page.
The user hits the (camera) button.		Open the camera page (UC- 3.2)
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No Alternate	No Alternate

Table 4-5 Start (Chatbot) Use-Case

Identifier	UC-3.1	
Purpose	To answer user's inquiries	
Priority	Medium	
Pre-conditions	Click the chatbot button from the **Options** menu.	
Post-conditions	The app is linked to IBM Watson to process the data and answer the user's inquiries.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click the chatbot button from the **Options** menu.	The app is linked to Watson to process the data.
2	The user asks questions.	The IBM Watson responds.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Unexpected question.	Exception.

Table 4-6 Camera Use-Case

Identifier	UC-3.2	
Purpose	To start video streaming	
Priority	High	
Pre-conditions	Click the camera button from the Home page.	
Post-conditions	Start video streaming.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user clicks the **start stream** button.	The app is linked to ESP32cam and start streaming.
2	The user clicks the **stop stream** button.	The app is linked to ESP32cam and stop streaming.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No Alternate	No Alternate

Table 4-7 Modify app's settings Use-Case

Identifier	UC-4	
Purpose	To modify the app's settings as preferred.	
Priority	Medium	
Pre-conditions	Get access to the HOME page after signing in.	
Post-conditions	The desired pages are accessed.	
Typical Course of Action		
Actor Action		**System Response**
The user hits the (Notifications) button from the menu.		Open Notifications page (UC-4.1).
The user hits the (Change Password) button from the menu.		Open Change Password page (UC-4.2).
The user hits (Privacy and Security Policy) button from the menu.		Display the Privacy and Security Policy on a new screen.
The user closes the Privacy and Security Policy screen.		Back to (Settings) page.
The user hits the (Log-out) button from the menu.		Open Log-out page (UC-4.3).
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No Alternate	No Alternate

Table 4-8 Modify Notifications Setting Use-Case

Identifier	UC-4.1	
Purpose	To modify the app's notifications settings as preferred.	
Priority	Medium	
Pre-conditions	Select the (Notifications) button from the **settings** menu.	
Post-conditions	Modifications applied and saved to the user profile in the users' database.	
Typical Course of Action		
Actor Action		**System Response**
The user switches the toggle button to enable or disable **Sound Notifications**.		apply changes and save them to the database.
The user switches the toggle button to enable or disable **Text Reading Mode**		apply changes and save them to the database.
The user switches the toggle button to enable or disable **Flash Notifications**		apply changes and save them to the database.
The user will schedule receiving reports.		save the option to the user profile in the users' database.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No Alternate	No Alternate

Table 4-9 Change Password Use-Case

Identifier	UC-4.2	
Purpose	TO change password	
Priority	Medium	
Pre-conditions	Login	
Post-conditions	User password is changed	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user enters the old password	Match password
2	The user enters a new password	
3	The user confirms the new password	Check if the new password &the confirmation matching.
4	The user Clicks Ok button	User password is changed
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	The user enters the old password	Message: Your old password is incorrect
2	The user entered an invalid confirmation.	Please enter the same value

Table 4-10 Log-out Use-Case

Identifier	UC-4.3	
Purpose	To log out from the system.	
Priority	Medium	
Pre-conditions	Login	
Post-conditions	User will logout from the system.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user clicks the **Logout** button.	Show a message to confirm the logout
2	The user clicks Ok.	Logout from the system
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	The user clicks the C**ancel** button.	Back to the **setting** screen (UC-4)

4.3 Nonfunctional Requirements

Performance Requirements

- The application must launch within 2 seconds.
- Load time for each screen should not exceed one second.
- The video streaming should be transmitted at least at 3.5 Mbps
- The image transfer rate of the connected camera must be at least 15 fps.
- Users can access the camera and all the application's features 99% of the time without failure.
- The average database response must not exceed 6 msec.
- The database server should process an average of 1000 query per second at worst.

Safety Requirements

- The system must preserve backups of all updates to the database.
- Back-ups shall be physically isolated from each other.

Security Requirements

- Any failed attempt by a user to access a data item must be registered on the audit trail.
- Constraints must be imposed on passwords (must be at least eight characters long, contain lower case, uppercase, numbers and symbol characters).
- Passwords must be invisible at entry time and at any other time.
- Only the data administrator of the system can modify the access permissions for system data.

Software Quality Attributes

Adaptability: The system can be connected and adapted with other devices using IoT technology.

Availability: The application is available 24 hour

Correctness: Through artificial intelligence and sensors, we can read the exact temperature and distinguish the sound of crying.

Flexibility: Adapting to future changes in requirements, application capacity and hardware to accommodate new features

Interoperability: Any change or upgrade to the interface between the system and the Warehouse system must be installed at the same time by both systems.

Maintainability:

- Customer service call centre analyzes 95% of problem reports within 2 hours.
- Items classified as "Urgent" is fixed within 3 working days in 98% of reported cases**.**
- The update made by the developer does not affect the application nor the customer's use of the application, so the new update is on a separate version.

Portability: The application should work on most Android devices of the same quality.

Reliability: error rate must be very rare.

Reusability: The programming language used in the application must be one of the most popular languages.

Robustness: Reducing defects and errors in hardware to a maximum of 3.4 products per million.

Usability: the user interfaces display in a large font without truncating displayed values or the text read aloud information displayed by Using the screen reader.

Other Requirements:

- The connected hardware should be lightweight, and the outer shell is designed without sharp edges to prevent any harm to the baby.
- **Communications Interfaces requirements**: Microcontrollers allows timely monitoring and sends notifications to the application through an advanced connection interface using the data centre structure.

- **The hardware needed to develop the system**

  Laptop, desktop.
  1. Processor i7.
  2. RAM: 32GB.
  3. HDD: 200 GB.

  Internet Access.

- **The software needed to develop the system**
  1. Android Studio.
  2. Android SDK
  3. Arduino IDE.
  4. Photoshop.

- **Legal requirements:**
  Obtaining the necessary licenses to sell the product, to ensure that no legal issues are exposed.
  The system must meet the policies agreed upon in the online marketplace for applications such as: " Play store" for Android applications.

- **Technical requirements:**
  We must have a professional design, development and maintenance team that are knowledgeable of the needed disciplines.

4.4 Analysis Models

The following use case diagram illustrated in ***Figure 0-1***, represents the user's interaction with the system that shows the relationship between the user and the different activities in which the user is involved.

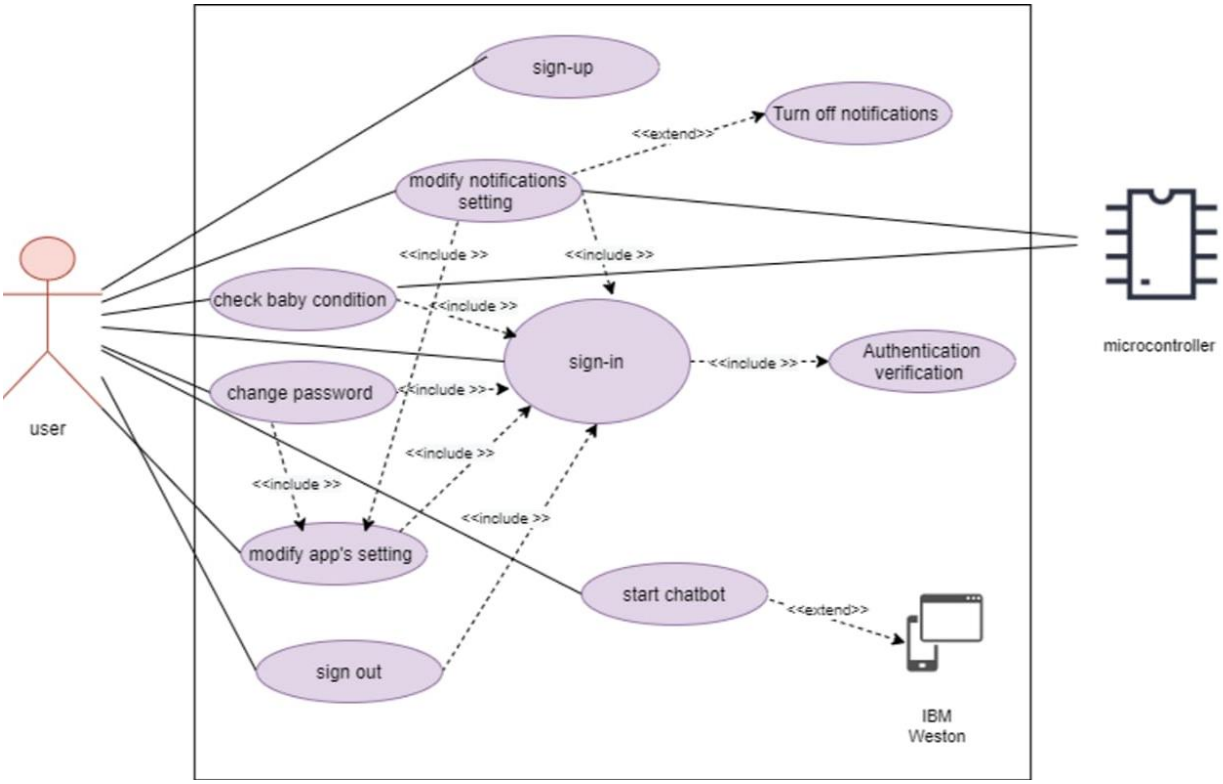

Figure 4-1 Use Case Diagram

CHAPTER 5: SYSTEM DESIGN

The **Entity-Relationship Diagram** (**ERD**) illustrated in Figure 5-1, represents the different entities within the system and how they relate to each other.

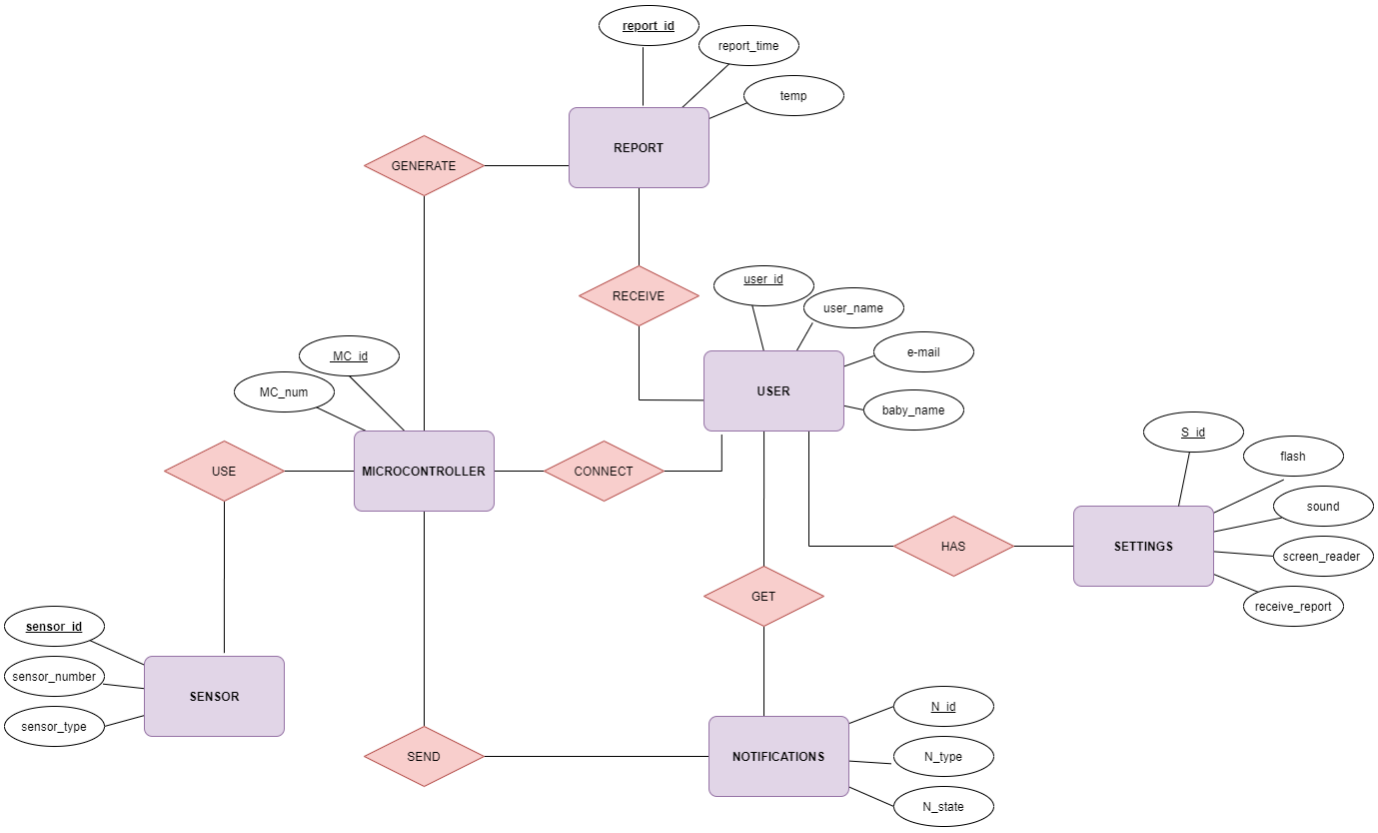

Figure 5-1 ERD in Chen notation

The **Class diagram** illustrated in Figure 5-2, shows the constant classes of the system, their attributes, methods, and relationships among classes.

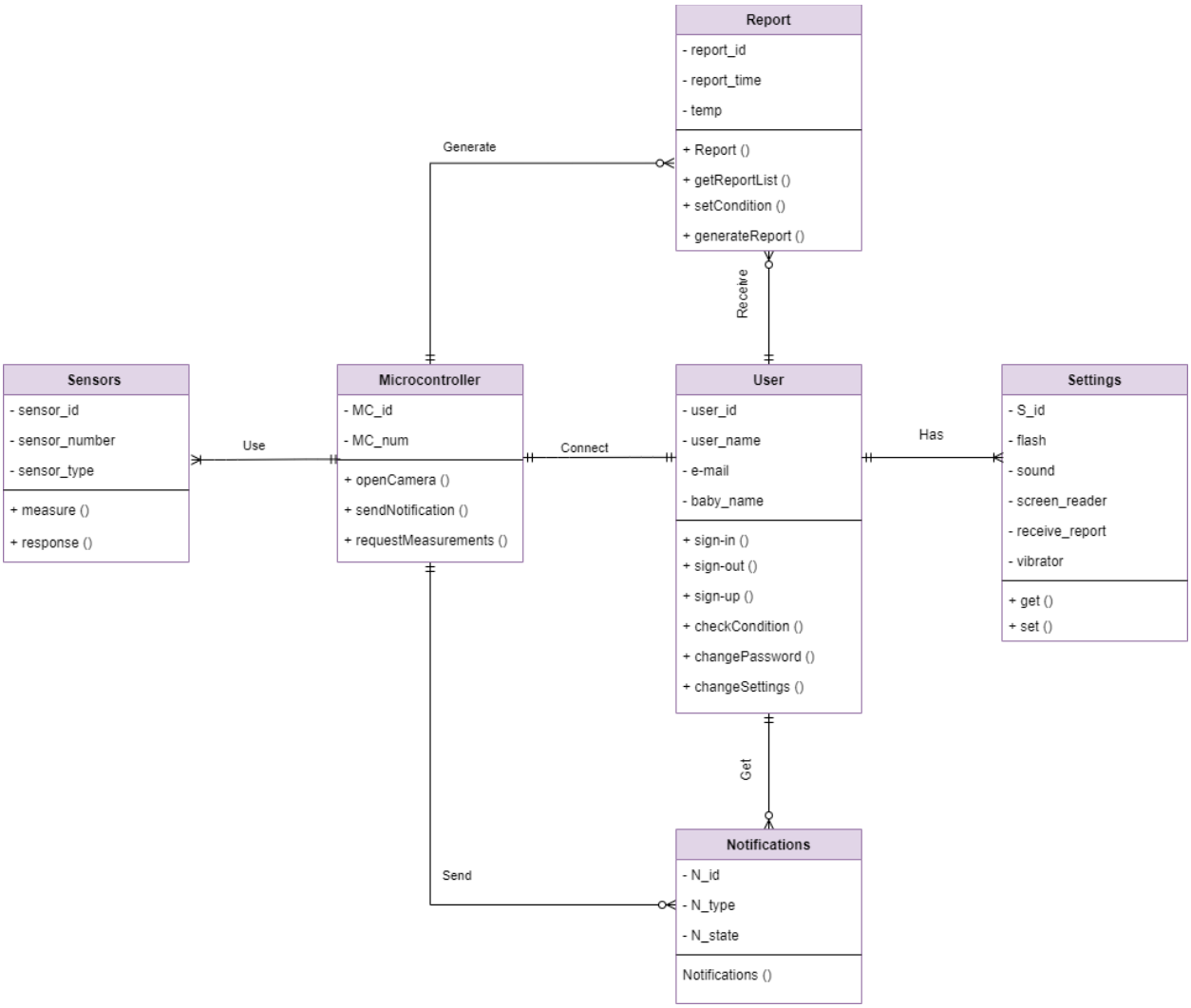

Figure 5-2 Class Diagram

The **sequence diagram** illustrated in Figure 5-3, displays the interactions of events ordered in a time sequence. It displays the objects that are involved in the scenario and the sequence of messages exchanged between the objects required to execute the scenario functionality.

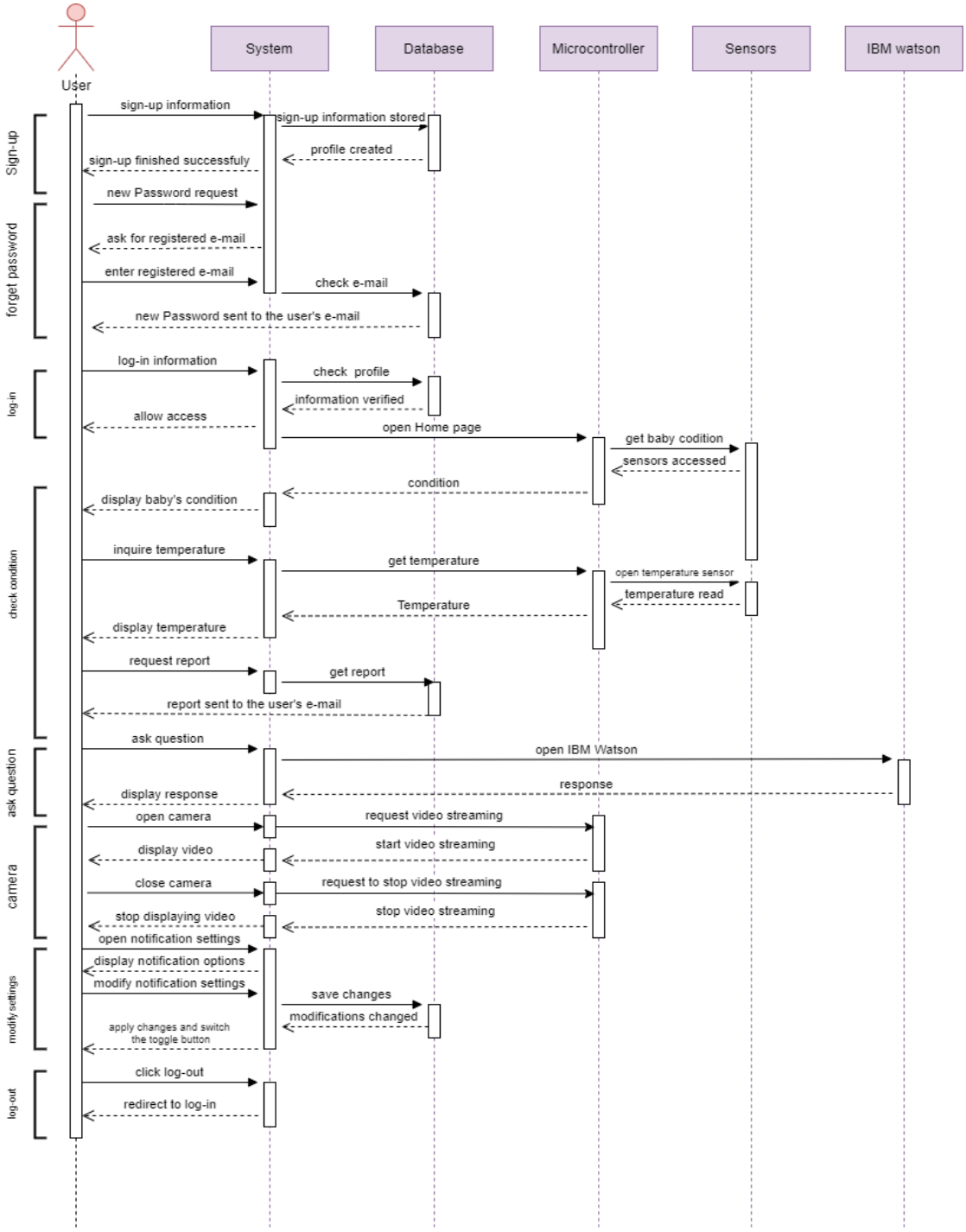

Figure 5-3 Sequence Diagram

The **component diagram** represented in Figure 5-4, illustrates how components are connected to shape the software system and explain how the system is organized.

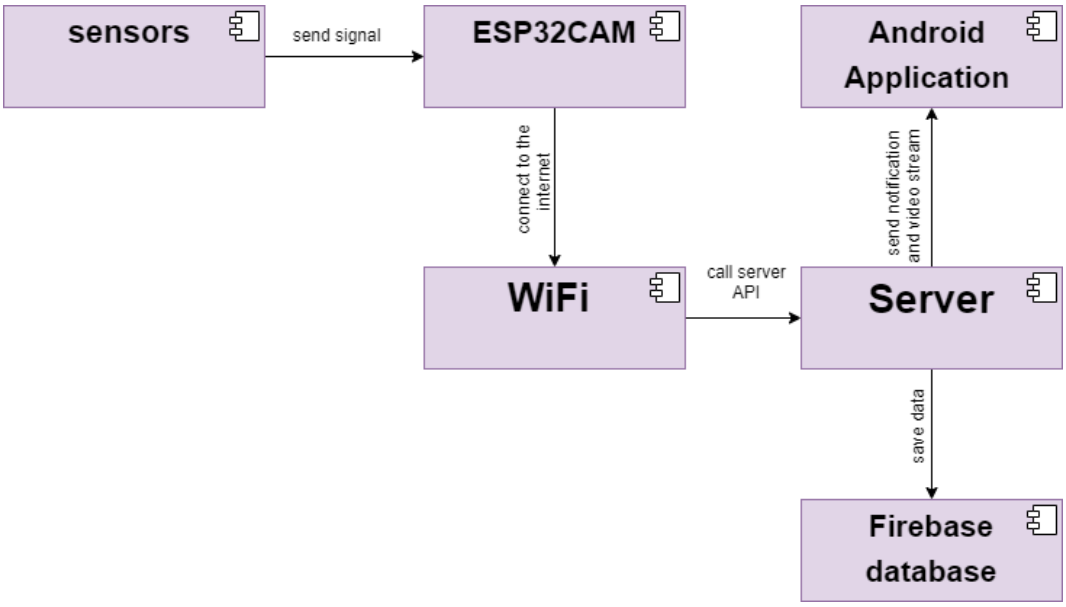

Figure 5-4 Component Diagram

The **Deployment diagram** shown in figure 5-5, Indicates how software functionality and subsystems will be allocated within the physical computing environment

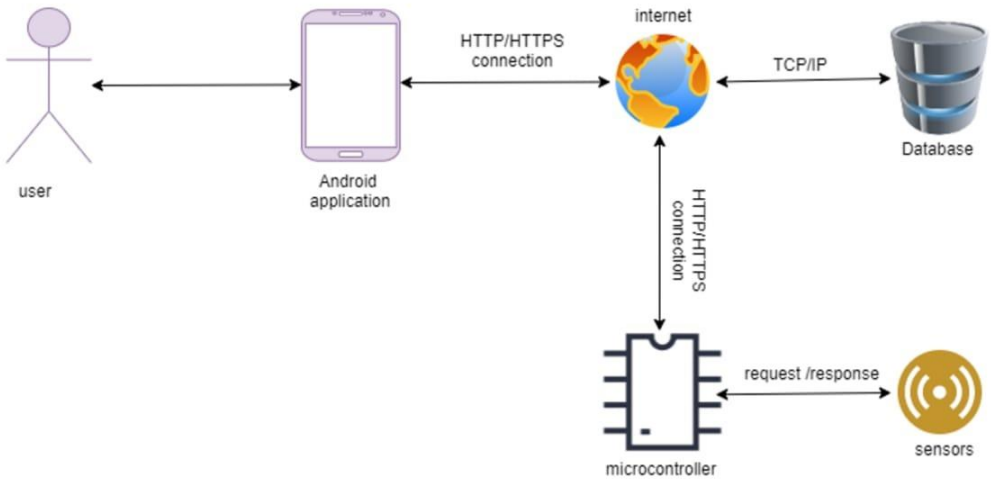

Figure 5-5 Deployment Diagram

CHAPTER 6: SYSTEM IMPLEMENTATION

6.1 Software Algorithm

The diagram shown in figure 6-1, illustrates the implemented data structures with their control flow from the start to the end and the potential branching of the process.

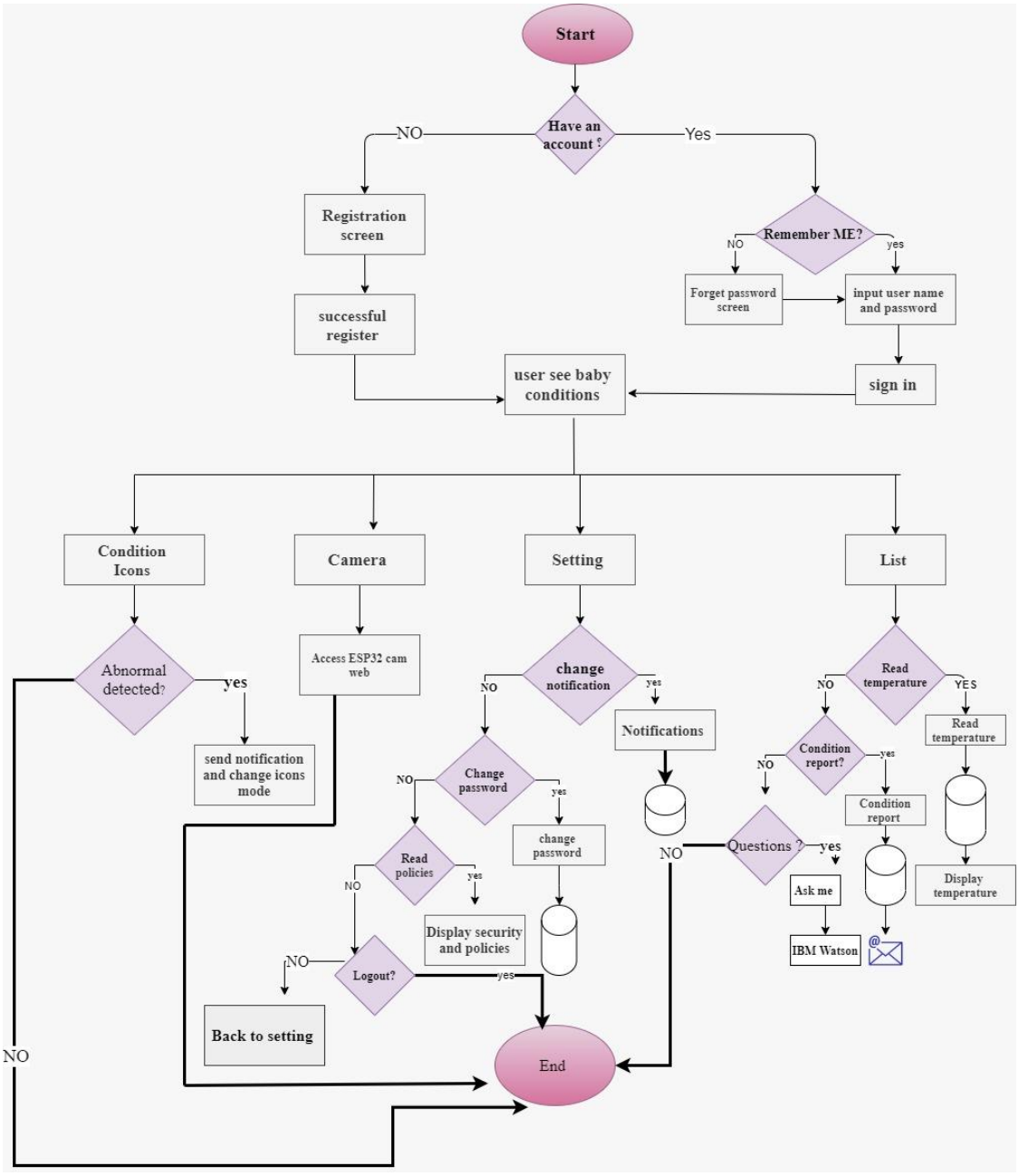

Figure 6-1 Control Flow Diagram

6.2 Software Implementation Code

Splash activity code:
This code shown in figure 6-2 is used to show the app's logo for 2 seconds.

```
lifecycleScope.launch(Dispatchers.Default) {  this: CoroutineScope
    delay( timeMillis: 2000)
    this@SplashActivity.openActivity(decideNavigation())
    overridePendingTransition(R.anim.fade_in_fast, R.anim.fade_out_fast)
    this@SplashActivity.finish()
}
```

Figure 6-2 Splash activity code

Registration code:
This code shown in figure 6-3, is used for registering new user to the app. It checks the existence of the username before creating a new user.

```
private fun registerWithUsername(userAccount: UserAccount) {
    firestoreAccountsCollection
        .whereEqualTo( field: "username", userAccount.username)
        .get()
        .addOnSuccessListener { result ->
            if (result.isEmpty) {
                val password = binding.passwordInputField.text.toString()
                createFirebaseUser(userAccount, password)
            } else {
                binding.loadingLayout.root.visibility = View.GONE
                "This username is already exist!"
                    .let { Snackbar.make(binding.root, it, Snackbar.LENGTH_LONG).show() }
            }
        }.addOnFailureListener {  it: Exception
            binding.loadingLayout.root.visibility = View.GONE
            Log.w( tag: "Registration",  msg: "Error getting documents: ", it)
        }
}
```

Figure 6-3 Registration code

Log-in code:
This code shown in figure 6-4, is used to log in to the app.

```
private fun performLogin(username: String, password: String) {
    firestoreAccountsCollection
        .whereEqualTo( field: "username", username)
        .get()
        .addOnSuccessListener { result ->
            if (result.isEmpty) {
                binding.loadingLayout.root.visibility = View.GONE
                "This username is not exist!"
                    .let { Snackbar.make(binding.root, it, Snackbar.LENGTH_LONG).show() }
            } else {
                val userAccount = result.documents
                    .first().toObject(UserAccount::class.java)
                userAccount?.let {  it: UserAccount
                    firebaseLogin(it.userEmail, password)
                }
            }
        }
        .addOnFailureListener {  it: Exception
            binding.loadingLayout.root.visibility = View.GONE
            Log.w( tag: "Login",  msg: "Error getting documents: ", it)
        }
}
```

Figure 6-4 log-in code

Temperature code:
This code shown in figure 6-5, gets the temperature degree and sends an alert to the user when it is higher than 37.

```
private fun updateTemperatureUi(temperature: Float) {
    tempValue = temperature
    binding.tempIndicator.setValueAnimated(tempValue)
    if (tempValue > 37) {
        addAlarmMedia(R.raw.alarm_temp)
        fireNotification(
            R.drawable.ic_heat,
            title: "Temperature is high",
            content: "Baby's temperature is gonna high!",
            id: 101
        )
    } else
        cancelNotification( notificationId: 101)
}
```

Figure 6-5 Temperature code

Sound code:

The code is shown in figure 6-6, gest the sound and sends an alert to the user when the sensor detects the baby crying.

```
private fun updateSoundUi(soundStatus: Boolean) {
    soundOn = soundStatus
    if (soundOn) {
        addAlarmMedia(R.raw.alarm_sound)
        binding.soundImage.alpha = 1f
        val iconRes = R.drawable.ic_megaphone
        binding.soundImage.setImageResource(iconRes)
        fireNotification(
            iconRes,
            title: "Sound detected",
            content: "There a sound detected around the baby",
            id: 102
        )
    } else {
        cancelNotification( notificationId: 102)
        binding.soundImage.alpha = 0.6f
        binding.soundImage.setImageResource(R.drawable.ic_megaphone_off)
    }
}
```

Figure 6-6 Sound code

Motion code:

This code shown in figuer6-7, gets the motions around the baby and sends an alert to the user when the sensor detects any motion.

```
private fun updateMotionUi(motionStatus: Boolean) {
    motionOn = motionStatus
    if (motionOn) {
        addAlarmMedia(R.raw.alarm_motion)
        binding.motionImage.alpha = 1f
        val iconRes = R.drawable.ic_motion_sensor
        binding.motionImage.setImageResource(iconRes)
        fireNotification(
            iconRes,
            title: "Motion detected",
            content: "There a motion detected around the baby",
            id: 103
        )
    } else {
        cancelNotification( notificationId: 103)
        binding.motionImage.alpha = 0.6f
        binding.motionImage.setImageResource(R.drawable.ic_motion_sensor_off)
    }
}
```

Figure 6-7 Motion code

ESP32cam code:
This code shown in figure 6-8, is used to link the app with the ESP32cam website.

```
class WebActivity : BaseActivity() {

    companion object {
        const val KEY_URL = "CAMP_IP_KEY"
    }

    private lateinit var binding: ActivityWebBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityWebBinding.inflate(layoutInflater)
        setContentView(binding.root)

        intent.extras?.getString(KEY_URL)?.let { url ->
            binding.webViewLayout.apply {  this: WebView
                settings.javaScriptEnabled = true
                webViewClient = WebViewClient()
                loadUrl(url)
            }
        }

    }
```

Figure 6-8 ESP32cam code

Change Password code:
This code shown in figure 6-9, is used to change the user's password.

```
binding.navigationLayout.okBtn.setOnClickListener {  it: View!
    val deviceNumber = binding.deviceNameInputField.text.toString()
    val oldPassword = binding.passwordInputField.text.toString()
    val newPassword = binding.newPasswordInputField.text.toString()
    val confirmPassword = binding.confirmPasswordInputField.text.toString()

    when {
        deviceNumber.isBlank() ->
            binding.deviceNumInputLayout.error = "Enter the device number!"
        oldPassword.length < 6 ->
            binding.passwordInputLayout.error = "Password is not correct!"
        newPassword.length < 6 ->
            binding.newPasswordInputLayout.error = "Password is not correct!"
        !ValidationUtils.isPasswordMatching(newPassword, confirmPassword) ->
            binding.confirmPasswordInputLayout.error =
                "Passwords are not match"
        else -> {
            binding.loadingLayout.root.visibility = View.VISIBLE
            validateDeviceNumber(deviceNumber)
        }
    }
```

Figure 6-9 Change Password code

Reset Password code:
This code shown in figure 6-10, is used to reset the password when the user forgets it and send a new password to the user's email.

```
private fun checkExistingUser(deviceNumber: String, email: String) {
    firestoreAccountsCollection
        .whereEqualTo( field: "deviceNumber", deviceNumber)
        .whereEqualTo( field: "userEmail", email)
        .get()
        .addOnSuccessListener { result ->
            binding.loadingLayout.root.visibility = View.GONE
            if (result.isEmpty) {
                "Device number or email are not correct!"
                    .let { Snackbar.make(binding.root, it, Snackbar.LENGTH_LONG).show() }
            } else {
                firebaseAuth
                    .sendPasswordResetEmail(email)
                    .addOnSuccessListener { it: Void!
                        "Reset link has been sent to your email address.".let { it: String
                            Snackbar.make(binding.root, it, Snackbar.LENGTH_LONG).show()
                        }
                    }
                    .addOnFailureListener { it: Exception
                        Log.w( tag: "Reset password", msg: "Reset password:failure", it)
                        it.localizedMessage?.let { msg ->
                            Snackbar.make(binding.root, msg, Snackbar.LENGTH_LONG).show()
                        }
                    }
            }
        }
```

Figure 6-10 Reset Password code

Logout code:
This code shown in figuer6-11, used to log out from the app.

```
private fun syncLogout() {
    binding.logout.visibility = View.VISIBLE
    binding.logout.setOnClickListener { it: View!
        FirebaseAuth.getInstance().signOut()
        val intent = Intent(applicationContext, LoginActivity::class.java).apply { this: Intent
            addFlags( flags: Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)
        }
        startActivity(intent)
    }
}
```

Figure 6-11 logout code

6.3 Database Design and Implementation

6.3.1 Database Management System

We used the Firebase tool provided by Google the backend as a service (BaaS) platform. We have chosen the Firebase backend platform due to its robustness and the wide range of features it provides which helps to develop agile mobile applications and reduce the need for server management and API development. It is a NoSQL database platform that follows the JSON protocol for data storage.

To connect our application to the database we performed the following steps:

1- Creating the database on Firebase **Realtime Database.**

2- Selecting Test mode as a starting mode for Firebase Security Rules.

3- Choosing a region for the database.

4- Adding the Realtime Database SDK to the app using the Firebase Android BoM.

5- Configuring Realtime Database Rules.

6- Writing to the database using the `getInstance()` method as shown in figure 6-12.

```
class LoginActivity : BaseActivity() {

    private lateinit var binding: ActivityLoginBinding

    private lateinit var firebaseAuth: FirebaseAuth
    private lateinit var firestoreAccountsCollection: CollectionReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)
        syncLanguageLabel()

        firebaseAuth = FirebaseAuth.getInstance()
        firestoreAccountsCollection = FirebaseFirestore
            .getInstance().collection( collectionPath: "accounts")

        binding.newUserLabel.setOnClickListener {  it: View!
            openActivity(RegisterActivity::class.java)
            useSlideEnterTransition()
        }

        binding.forgetPasswordLabel.setOnClickListener {  it: View!
            openActivity(ResetPasswordActivity::class.java)
            useSlideEnterTransition()
        }
```

Figure 6-12 Writing to the database

7- Reading from the database by adding a `ValueEventListener` to the referenced location, then, the onDataChange() method shown in figure 6-13, will be triggered when the listener is attached and every time the data changes.

```
testModeDataListener = object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        Log.i( tag: "FirebaseDB",  msg: "Got snapshot ${snapshot.value}")
        camIP = snapshot.child( path: "IP").getValue(String::class.java)
        val remoteTemp = snapshot.child( path: "temp").getValue(Float::class.java) ?: 0f
        val remoteSound = snapshot.child( path: "sound").getValue(Boolean::class.java) ?: false
        val remoteMotion = snapshot.child( path: "motion").getValue(Boolean::class.java) ?: false

        if (remoteTemp != tempValue)
            updateTemperatureUi(remoteTemp)
        if (remoteSound != soundOn)
            updateSoundUi(remoteSound)
        if (remoteMotion != motionOn)
            updateMotionUi(remoteMotion)

        voiceReadingAlarm()
    }

    override fun onCancelled(error: DatabaseError) {
        Log.e( tag: "FirebaseDB",  msg: "loadPost:onCancelled", error.toException())
    }
```

Figure 6-13 Reading from the database

This code shown in figure 6-14, is used to connect the app with the database.

```
package com.parentseyes.ui.home

import ...

class HomeActivity : BaseActivity(), NavigationView.OnNavigationItemSelectedListener {

    companion object {
        private const val CHANNEL_ID = "PARENT_EYES_ALERT_CHANNEL_01"
    }

    private lateinit var binding: ActivityHomeBinding

    private val realtimeDB by lazy { Firebase.database }
    private val testModeDataRef by lazy {
        realtimeDB.reference.child( pathString: "dev7")
    }
    private lateinit var testModeDataListener: ValueEventListener

    private var readingMode: Boolean = false

    private val playingMedia by lazy { mutableSetOf<Int>() }

    private var camIP: String? = null
    private var tempValue: Float = 0f
    private var soundOn: Boolean = false
    private var motionOn: Boolean = false
```

Figure 6-14 connecting to the database

6.3.2 Database Schema

Database Schema: this schema is written using SQL, however, we used Firebase for our project which is a NoSQL database that uses collections and documents to build the database.

```
Create table Report(
   report_id varchar(15) primary key,
   report_time  datetime,
   temp double not null,
);

Create table Notifications(
   N_id varchar(15) primary key,
   N_state  varchar(15),
   N_type varchar(15) not null
);

Create table Microcontrollers(
   MC_id varchar(15) primary key,
   MC_num  integer not null
);

Create table Accounts(
   accountId varchar(15) primary key,
   babyName  varchar(15) not null,
   deviceNumber  varchar(15) not null,
   userEmail  varchar(30) not null,
   username  varchar(15) not null
);

Create table Sensors(
   sensor_id varchar(15) primary key,
   sensor_number  integer not null,
   sensor_type varchar(15) not null
);

Create table Settings(
   S_id varchar(15) primary key,
   flash  BIT not null,
   sound  BIT not null,
   screen_reader  BIT not null,
   receive_report  BIT not null,
   vibrator  BIT not null
);
```

The table6-1, shows all tables and their relationships with nature i.e., one-to-one, one-to-many, many-to-many etc.

Relationship	From	To	**Relationship from**	**Relationship to**
Has	user	settings	(1-1)	(1-N)
Get	user	notification	(1-1)	(0-N)
Receive	user	Report	(1-1)	(0-N)
Connect	User	Microcontroller	(1-1)	(1-1)
Use	Microcontroller	Sensors	(1-1)	(1-N)
Send	Microcontroller	notification	(1-1)	(0-N)
generate	Microcontroller	Report	(1-1)	(0-N)

Table 6-1 database tables and relationships

6.3.3 Database Physical Model

The following tables describe the details of each table with all its entities, data styles, primary keys, and constraints:

Report

	Data style	Primary key	Unique	Not Null
report_id	String	-	-	-
report_time	Timestamp			
temp	Double			-

Table 6-2 Report table

Notifications

	Data style	Primary key	Unique	Not Null
N_id	String	-	-	-
N_state	String			
N_type	String			-

Table 6-3 Notification table

Microcontrollers

	Data style	Primary key	Unique	Not Null
MC_id	String	-	-	-
MC_num	Integer		-	-

Table 6-4 Microcontrollers table

Accounts

	Data style	Primary key	Unique	Not Null
accountId	String	-		-
babyName	String			-
deviceNumber	String		-	-
userEmail	String		-	-
username	String			-

Table 6-5 Accounts table

Sensors

	Data style	Primary key	Unique	Not Null
sensor_id	String	-		-
sensor_number	Integer			-
sensor_type	String		-	-

Table 6-6 Sensors table

6.3.4 Completed Database Screenshots

The following figures show representative filled data tables used in the system with real data values.

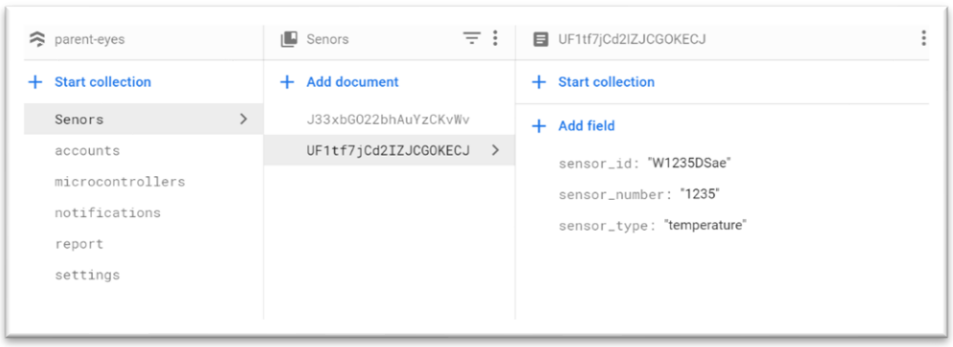

Figure 6-15 sensors table's screenshot

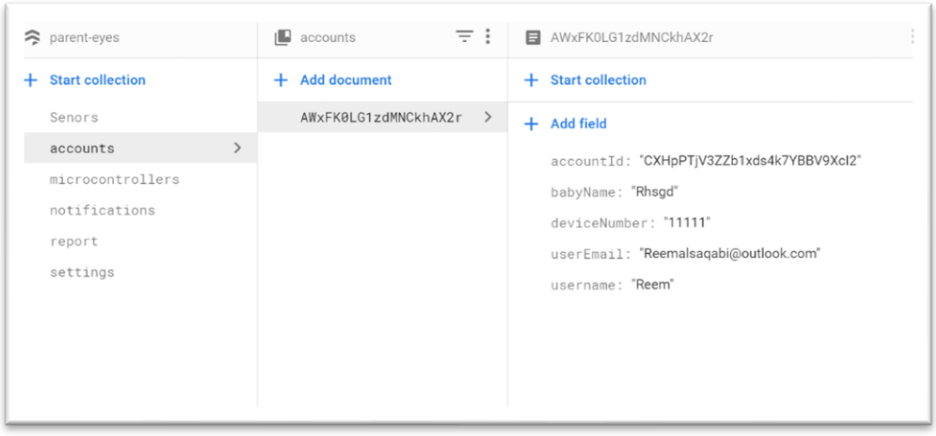

Figure 6-16 accounts table's screenshot

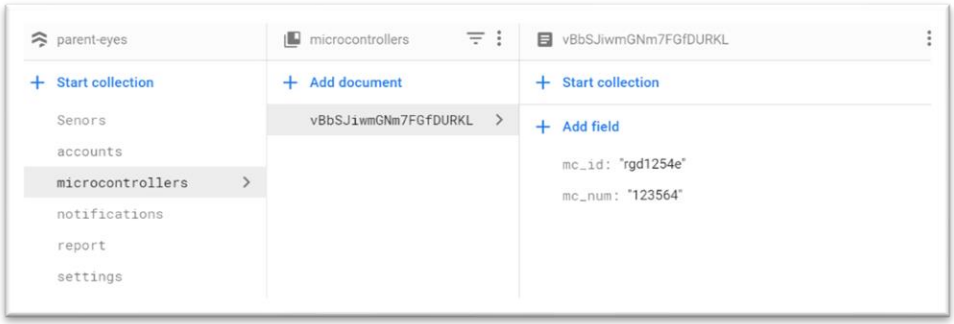

Figure 6-17 microcontroller table's screenshot

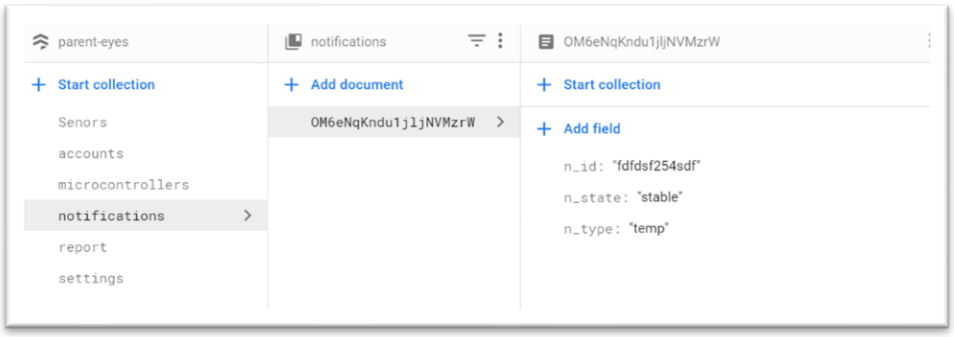

Figure 6-18 notifications table's screenshot

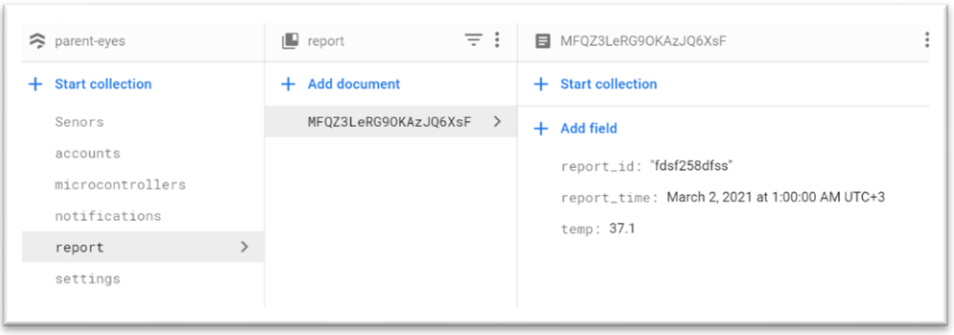

Figure 6-19 report table's screenshot

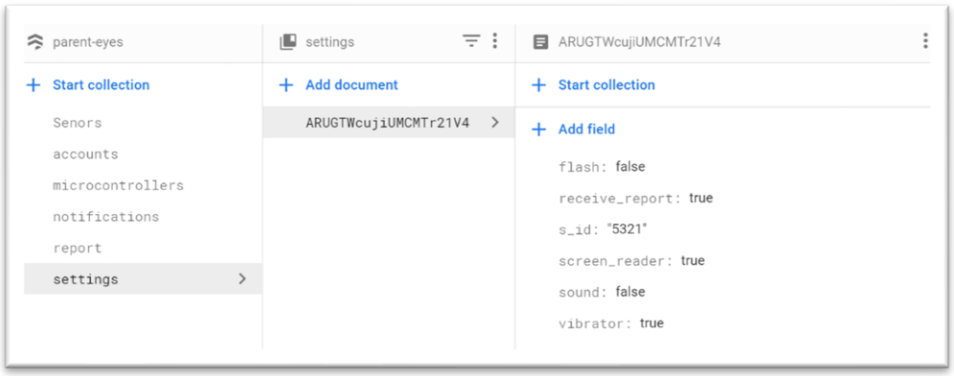

Figure 6-20 settings table's screenshot

6.4 Software Interfaces

The following figures show the software application's graphical user interface:

The splash activity shown in figure 6-21, will appear when the user presses the application button. It contains the application's logo and appears for a few seconds, after which we go directly to the log in.

Figure 6-21 splash activity screenshot

Users can fill the form to log in to a profile, by entering the user's name and password in the log-in user face shown in the figure6-22.

Figure 6-22 log-in screenshot

The registration screen shown in figure 6-23, is for a new user in which the user enters the device number, the baby's name, the username, the password, and confirms the password and the user's email, then the user will have an account on the application database and can log in directly to the app.

Figure 6-23 registration screenshot

Change password interface shown in figure 6-24, appears if the user requested to change his password, through which he/she can change the password and obtain a new password.

Figure 6-24 change password screenshot

Reset password interface shown in figure 6-25, appears when the user forgets his/her password, the user must enter the device number and email to obtain a new password on his/her email

Figure 6-25 Reset password screenshot

The home page interface shown in figure 6-26, contains four icons indicating the sensors such as temperature, sound detection, motion detection, and the camera. At the top of the screen on the right has the Settings button and on the left side has a list that contains many functions.

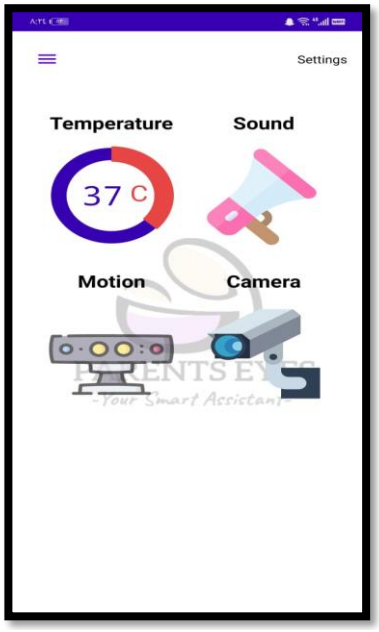

Figure 6-26 Home page screenshot

The user can access the list shown in figure 6-27 from the homepage. It display screen contains three options, the first one is the Read temperature to display the baby current temperature. The second one is the Condition report on the e-mail with the child's temperature. The third one is asking me is for questions related to the application by using IBM Watson.

Figure 6-27 list screenshot

The notification setting interface shown in figure 6-28, enables the user to control the notifications and the way they are displayed. He/she has several options such as sending notifications or reading the written speech which is useful for blind users or flash to alert the deaf users.

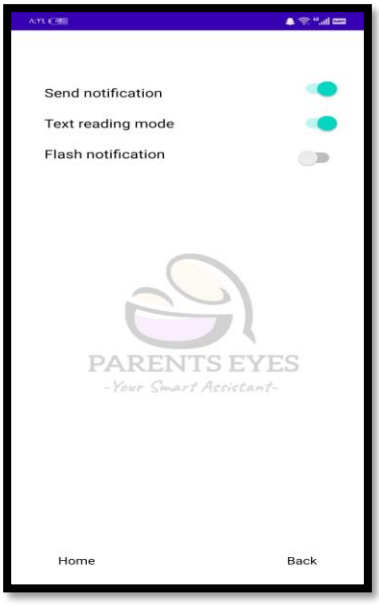

Figure 6-28 notification settings screenshot

The settings interface shown in figure 6-29, enables the user to change the notifications' settings, and request to change the password also view the application's privacy policy, and he/she can log out of the application

Figure 6-29 settings screenshot

The Ask-me interface shown in figure 6-30 enables the user to ask questions and get answers from the IBM Watson chatbot.

Figure 6-30 ask me screenshot

The security policy interface shown in figure 6-31 enables the user to read the security policy of the app.

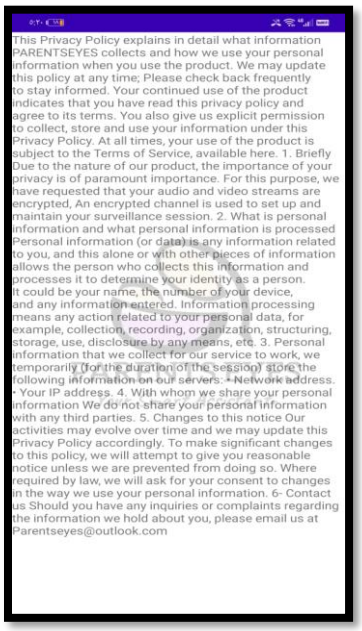

Figure 6-31 security policy screenshot

CHAPTER 7: SOFTWARE TESTING AND EVALUATION

7.1 Software Testing

Table 7-1 illustrates the case of singing-in to the system.

Table 7-1: TC-1

Identifier	TC-1
Priority	High
Related requirements(s)	UC-1
Short description	To Sing-in to the system.
Pre-condition(s)	Existing user (Already have an account).
Input data	Username and Password
Detailed steps	1. the user enters Username 2. the user enters the password 3. the user clicks Ok
Expected result(s)	Successful sign-in.
Post-condition(s)	Authentication response.

Table 7-2 illustrates the case of forgetting the password.

Table 7-2: TC-2

Identifier	TC-2
Priority	Medium
Related requirements(s)	UC-1.1
Short description	When the user forgets the password, he/she can have a new one throw the Emile
Pre-condition(s)	Login
Input data	Username and Emile
Detailed steps	1. The user enters the old password 2. The user enters a new password 3. The user confirms the new password 4. The user Clicks Ok button
Expected result(s)	Successfully change the password
Post-condition(s)	User password is changed

Table 7-3 illustrates the case of registering a new user.

Table 7-3: TC-3

Identifier	TC-3
Priority	High
Related requirements(s)	UC-2
Short description	To register the user and save his/her data to the Users database.
Pre-condition(s)	Select (sign-up) button from the **sign-in** screen.
Input data	User personal information
Detailed steps	1. Fill in the registration form 2. Submit form
Expected result(s)	Successfully registration
Post-condition(s)	Registration form submitted and user profile created

Table 7-4 illustrates the case of accessing the home page.

Table 7-4: TC-4

Identifier	TC-4
Priority	High
Related requirements(s)	UC-3
Short description	To access the home page functions
Pre-condition(s)	Log in and configure the hardware.
Input data	No input data
Detailed steps	1. The user signed in and access the options menu from the home page then open the camera 2. The user hits the (Read Temperature) button from the **Options** menu. 3. The user hits (Condition Report) button from the **Options** menu. 4. The user hits the (Settings) button from the **Options** menu. 5. The user hits the (Ask me) button from the **Options** menu. 6. The user closes the (Ask me) window. 7. The user hits the (camera) button.
Expected result(s)	The system response successfully to all the requests unless the Condition Report button.
Post-condition(s)	Get access to all the functions on the home page

Table 7-5 illustrates the case of answering user's inquiries via IBM Watson.

Table 7-5: TC-5

Identifier	TC-5
Priority	Medium
Related requirements(s)	UC-3.1
Short description	To answer user's inquiries
Pre-condition(s)	Click the chatbot button from the **Options** menu.
Input data	User's questions.
Detailed steps	1. The user clicks the chatbot button from the **Options** menu. 2. The user asks questions.
Expected result(s)	The app is linked to The IBM Weston, gets responses and throws an exception in case of unexpected questions.
Post-condition(s)	The app is linked to IBM Watson, process the data, and answer the user's inquiries.

Table 7-6 illustrates the case of starting the video stream.

Table 7-6: TC-6

Identifier	TC-6
Priority	High
Related requirements(s)	UC-3.2
Short description	To start video streaming
Pre-condition(s)	Click the camera button from the Home page.
Input data	No input data.
Detailed steps	1. The user clicks the **start stream** button. 2. The user clicks the **stop stream** button.
Expected result(s)	The app linked successfully to ESP32cam and start/stop streaming.
Post-condition(s)	The app is linked to ESP32cam and start/stop streaming.

Table 7-7 illustrates the case of modifying the app's settings.

Table 7-7: TC-7

Identifier	TC-7
Priority	Medium
Related requirements(s)	UC-4
Short description	To modify the app's settings as preferred.
Pre-condition(s)	Get access to the Home page after signing in.
Input data	No input data.
Detailed steps	• The user hits the (Notifications) button from the menu. • The user hits the (Change Password) button from the menu. • The user hits (Privacy and Security Policy) button from the menu. • The user closes the Privacy and Security Policy screen. • The user hits the (Log-out) button from the menu.
Expected result(s)	The desired pages are successfully accessed.
Post-condition(s)	The desired pages are accessed.

Table 7-8 illustrates the case of modifying the app's notifications settings.

Table 7-8: TC-8

Identifier	TC-8
Priority	Medium
Related requirements(s)	UC-4.1
Short description	To modify the app′s notifications settings as preferred.
Pre-condition(s)	Select the (Notifications) button from the **Settings** menu.
Input data	No input data.
Detailed steps	• The user switches the toggle button to enable or disable **Sound Notifications**. • The user switches the toggle button to enable or disable **Text Reading Mode** • The user switches the toggle button to enable or disable **Flash Notifications**
Expected result(s)	Notifications modified successfully except the **Flash Notifications.**
Post-condition(s)	Modifications applied and saved to the user profile in the users′ database.

Table 7-9 illustrates the case of changing the password.

Table 7-9: TC-9

Identifier	TC-9
Priority	Medium
Related requirements(s)	UC-4.2
Short description	To change password
Pre-condition(s)	Login
Input data	Old password, new password, new password confirmation
Detailed steps	1. The user enters the old password 2. The user enters a new password 3. The user confirms the new password 4. The user clicks the Ok button
Expected result(s)	password is changed successfully
Post-condition(s)	User password is changed

Table 7-10 illustrates the case of logging out from the system.

Table 7-10: TC-10

Identifier	TC-10
Priority	Medium
Related requirements(s)	UC-4.3
Short description	To log out from the system.
Pre-condition(s)	Login
Input data	No input data
Detailed steps	1. The user clicks the **Logout** button. 2. The user clicks Ok.
Expected result(s)	Successful logout
Post-condition(s)	The user will log out from the system.

7.2 Software Verification

The following table 7-11 lists all the defects found in the project during testing and their current status.

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1	Disable reading of the text by an automatic reader when notifications appear	Testing	Fixed	1	30
2	When the user tries to change the password, changes are not applied.	Testing	Fixed	5	45

Table 7-11: List of non-trivial defects

CHAPTER 8: HARDWARE IMPLEMENTATION

8.1 Hardware Connecting Diagram

The following diagram figure 8-1, illustrates how the sensors are linked with the ESP32cam.

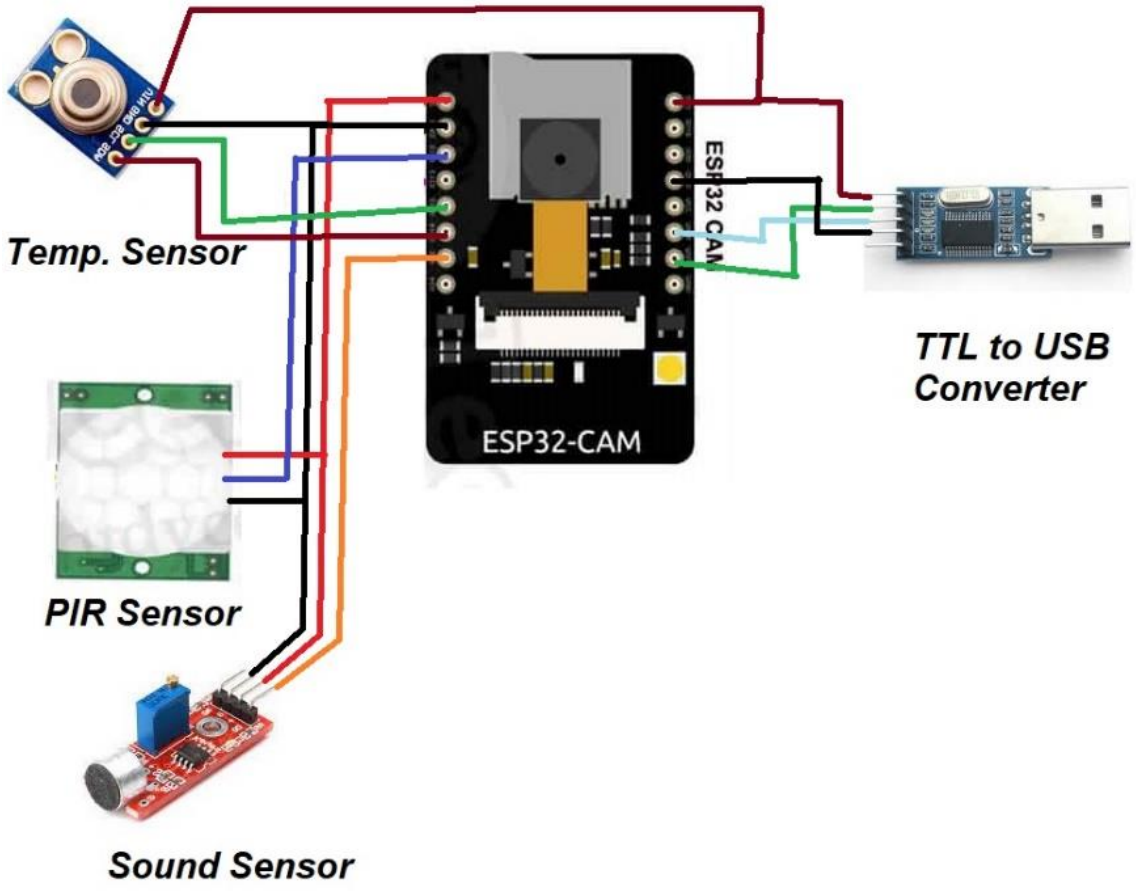

Figure 8-1 Hardware linkage

8.2 Hardware Codes

The microcontroller is programmed using Arduino (IDE). The screenshots below show the codes for the different sensors.

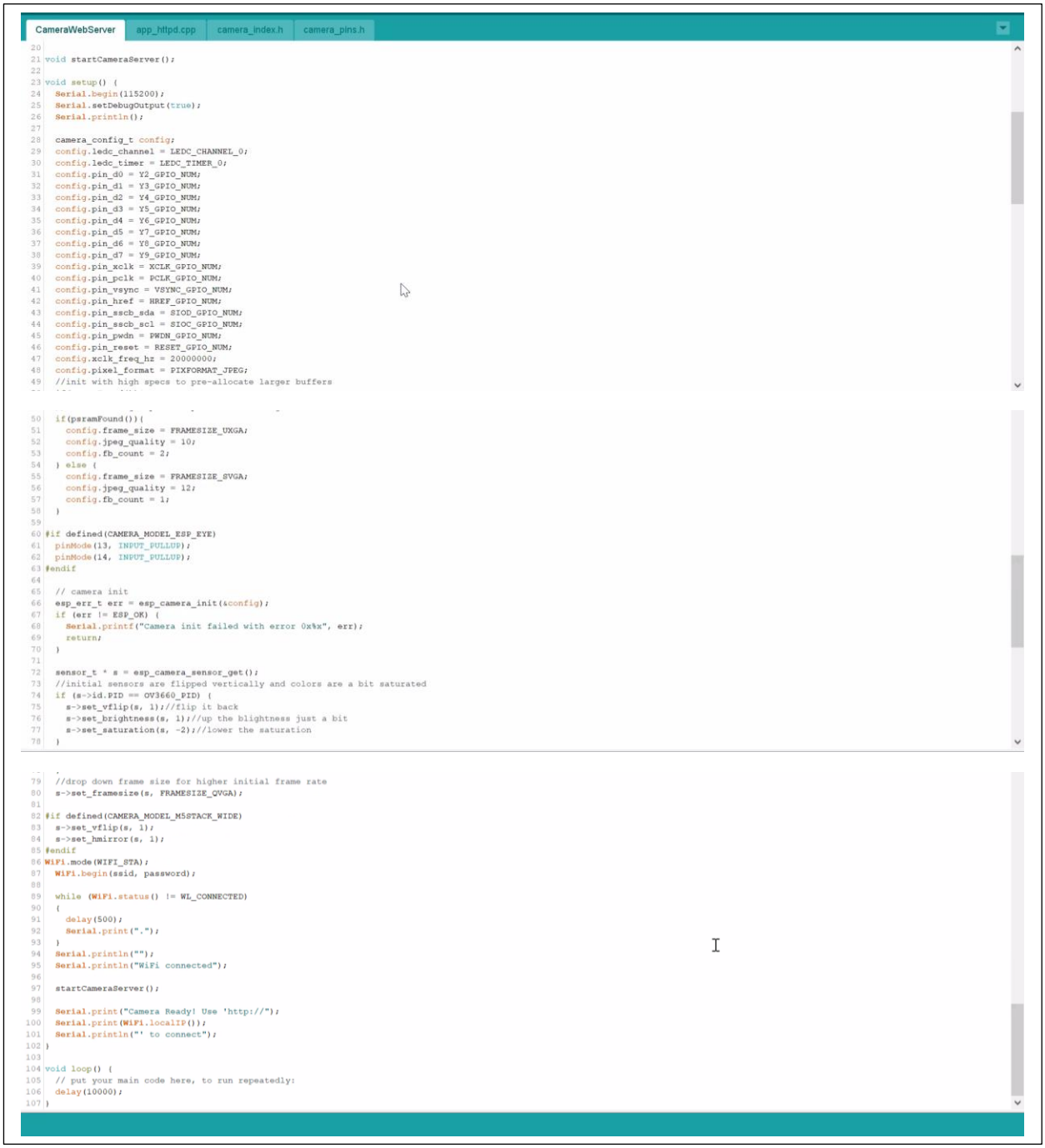

Figure 8-1 ESP32cam Code

```
18
19 #include <Wire.h> // i2c protocol
20 #include <Adafruit_MLX90614.h> // librariy for temp
21
22 Adafruit_MLX90614 mlx = Adafruit_MLX90614();
23
24 void setup() {
25   Serial.begin(115200);
26
27   Serial.println("Adafruit MLX90614 test");
28
29   mlx.begin();
30 }
31
32 void loop() {
33   Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());  الحراره المحيطه//
34   Serial.print("*C\tObject = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C"); حراره الجسم //
35   Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempF());
36   Serial.print("*F\tObject = "); Serial.print(mlx.readObjectTempF()); Serial.println("*F"); درجه الحراره بالفهرهايت //
37
38   Serial.println();
39   delay(500);
40 }
```

Figure 8-2 Temperature sensor Code

pir_code_ino

```
1  int LED = 2;                // the pin that the LED is atteched to
2  int PIR = 4;                // the pin that the sensor is atteched to
3
4  void setup() {
5    pinMode(LED, OUTPUT);     // initalize LED as an output
6    pinMode(PIR, INPUT);      // initialize sensor as an input
7    Serial.begin(115200);       // initialize serial
8  }
9
10 void loop(){
11   if (digitalRead(PIR) == HIGH) { // check if the sensor is HIGH
12     digitalWrite(LED, HIGH);      // turn LED ON
13     Serial.println("Motion detected!");
14     delay(250);                   // delay 100 milliseconds
15   }
16   else {
17     digitalWrite(LED, LOW);       // turn LED OFF
18     Serial.println("Motion stopped!");
19     delay(250);                   // delay 100 milliseconds
20   }
21 }
```

Figure 8-3 PIR motion sensor Code

sound_densor_part1 §

```
1
2
3  const int OUT_PIN = 8;
4
5  void setup() {
6    Serial.begin(9600);
7  }
8
9  void loop() {
10   Serial.println(digitalRead(OUT_PIN));
11 }
```

Figure 8-4 Sound Detector Sensor Code part1

sound_densor_part2

```
12 */
13
14 const int OUT_PIN = 8;
15 const int SAMPLE_TIME = 10;
16 unsigned long millisCurrent;
17 unsigned long millisLast = 0;
18 unsigned long millisElapsed = 0;
19
20 int sampleBufferValue = 0;
21
22 void setup() {
23   Serial.begin(9600);
24 }
25
26 void loop() {
27
28   millisCurrent = millis();
29   millisElapsed = millisCurrent - millisLast;
30
31   if (digitalRead(OUT_PIN) == LOW) {
32     sampleBufferValue++;
33   }
34
35   if (millisElapsed > SAMPLE_TIME) {
36     Serial.println(sampleBufferValue);
37     sampleBufferValue = 0;
38     millisLast = millisCurrent;
39   }
40
41 }
```

Figure 8-5 Sound Detector Sensor Code part2

CHAPTER 9: DISCUSSION & CONCLUSION

9.1 Discussion

The baby monitoring system approach depends on input acquisition from the ESP32 smart camera and other sensors connected to the microcontroller to stream live video of the baby that can be viewed through the app in smart devices and send the abnormality notifications in a wireless network.

Our goal of this application in the future is to develop more features, making it one of the best applications in the field of child monitoring. We strive to make this system affordable and can be owned by all parents. We are looking forward to integrating our project with other devices within the Internet of Things.

Future reports would contain more information about the child's condition. We are also looking forward to adding a face expression detection to capture the mood of the child from his facial expressions. We also hope that our project will be desirable by all parties specialized in the field of children in addition to that it supports the IOS operating system.

This app will revolutionize the field of baby monitor apps.

During the implementation, we faced the following challenges:

1- One of the challenges that we faced in the temperature sensor, as the project required a sensor of a certain type that measured from a distance of a half-meter and this sensor was ordered, but unfortunately it was not shipped, which forced us to use a sensor of the same type but with fewer capabilities as it measures the temperature from a shorter distance. It must be noted that the code and connections are all identical in both sensors.

2- We were not able to link sending the report to the e-mail because there were fees and time issues for this feature.

3- There were fees and time issues also for the flash and vibration feature, however, the buttons for them were added, and the feature would be activated if this project were implemented commercially.

4- It is important to change the microcontrollers in future releases because the nature of the ESP32cam requires its connection to the web and the appearance of the entire menu, and this menu cannot be hidden.

5- After starting with the hardware part, we found that the sound sensor used has a 5Volt, and this does not comply with the nature of the microcontrollers used, which made us use another sound sensor with specifications less than previously specified.

6- Also, after the arrival of the required parts, we found that this microcontroller did not have a USB port, which forced us to order a TTL to USB converter to connect it to the computer and upload the codes to it.

9.2 Conclusion

Taking care of babies could be difficult especially for working parents. In this project, we introduce a system for monitoring babies in homes and nurseries and report the status of the baby in integration with IoT.

We aimed to facilitate babies caring for parents and caregivers so that they can work and monitor the child at the same time and provide features to enable parent with vision and hearing disabilities. We intend to measure further physiological signals and add more features to the system to serve hospitals and healthcare centres.

REFERENCES

[1]. Ali, M. A., Tahir, N. M., & Ali, A. I. (2018, December). Monitoring Healthcare System for Infants: A Review. In *2018 IEEE Conference on Systems, Process and Control (ICSPC)* (pp. 44-47). IEEE.

[2]. Ul Hasan, N., & Negulescu, I. (2020, July). Wearable Technology for Baby Monitoring: A Review.

[3]. World Health Organization (WHO), October 2020 https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

[4]. World Health Organization (WHO), March 2020 https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[5]. https://www.samsung.com/in/support/mobile-devices/what-is-baby-crying-detector-in-samsung-galaxy-s5/.

[6]. Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals*, 110059.

[7].Artificial intelligence. (2020, October 20). Retrieved October 23, 2020, from https://en.m.wikipedia.org/wiki/Artificial_intelligence.

[8]. Adhikary, T., Jana, A. D., Chakrabarty, A., & Jana, S. K. (2019, January). The internet of things (IoT) augmentation in healthcare: An application analytics. In *International Conference on Intelligent Computing and Communication Technologies* (pp. 576-583). Springer, Singapore.

[9]. Lu, H., Li, Y., Chen, M. *et al.* Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Netw Appl* **23,** 368–375 (2018). https://doi.org/10.1007/s11036-017-0932-8.

[10].Lee, J., Kim, G., Yoo, J., Jung, C., Kim, M., & Yoon, S. (2016). Training IBM Watson using automatically generated question-answer pairs. *arXiv preprint arXiv:1611.03932*.

[11]. Anderson P. (2020, June 30). Introducing the new Watson Discovery Premium Plan. IBM https://www.ibm.com/blogs/watson/2020/06/introducing-the-new-watson-discovery-premium-plan/

[12]. Banafa, A. (2018). Secure and Smart Internet of Things (IoT): Using Blockchain and Artificial. Denmark, Gistrup: Rever.

[13]. Tankovska, H. (2020). Internet of Things (IoT) spending worldwide 2023, Statista.com.

[14]. Wu, F., Wu, T., & Yuce, M. R. (2019). An internet-of-things (IoT) network system for connected safety and health monitoring applications. *Sensors*, *19*(1), 21.

[15]. Ali, B., & Awad, A. I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. *sensors*, *18*(3), 817.

[16]. Ranger, Steve (2020 February 3) What is the IoT? Everything you need to know about the Internet of Things right now. Retrieved from https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/.

[17]. Yang, B., Li, X., Hou, Y., Meier, A., Cheng, X., Choi, J. H., ... & Li, A. (2020). Non-invasive (non-contact) measurements of human thermal physiology signals and thermal comfort/discomfort poses-A review. *Energy and Buildings*, 110261.

[18]. Zakaria, N. A., & Abdulrazak, M. A. (2018, July). IoT (Internet of Things) Based Infant Body Temperature Monitoring.

[19]. Zeng, Y., Lu, E., Sun, Y., & Tian, R. (2019). Responsible facial recognition and beyond. *arXiv preprint arXiv:1909.12935*.

[20]. Choi, S., Yun, S., & Ahn, B. (2020). Implementation of Automated Baby Monitoring: CCBeBe.

[21]. Martínez, A., A. Pujol, F., & Mora, D. (2020). Application of Texture Descriptors to Facial Emotion Recognition in Infants.

[22]. Hussain, T., Muhammad, K., Khan, S., Ullah, A., M., & Baik, S. W. (2019). Intelligent Baby Behavior Monitoring using Embedded Vision in IoT for Smart Healthcare Centers.

[23]. (Dormi - Baby Monitor) Application. http://dormi.sleekbit.com/features.html

[24]. (Luna - Baby Monitor with Video) Application. https://babyphone.app/

[25]. (Baby Monitor 3G) Application. https://www.babymonitor3g.com/

APPENDIX: Glossary

The Internet of Things (IoT)

A network of physical objects containing embedded technology to communicate with their inner states or the surrounding environment and to sense or interact with them.

Artificial intelligence (AI)

Sophisticated analysis and logic-based methods, like machine learning that are used to interpret events, support, and automate decisions and take action.

An application programming interface (API)

An interface that enables programmatic access within an application or a database to the service functionality and data.

Chatbot

A domain-specific conversational interface for interactions that uses an app, social network, messaging platform, or chat solution.

Artificial Intelligence of Things (AIoT):

Artificial Intelligence of Things (AIoT): is to use artificial intelligence (AI) technologies with the Internet of Things (IoT) to achieve more efficient operations and improve data management and analysis.

Hypertext Transport Protocol (HTTP)

It is a system for transmitting Internet data over the World Wide Web (the Web), and it is the main and most widespread method for transmitting data on the Web (www).

Hypertext Transport Protocol Secure (HTTPS):

It is a combination of hypertext transfer protocol with TLS / SSL / protocol to provide encrypted communications and define a secure web server network.

TCP /IP

Transmission Control Protocol and Internet Protocol called TCP / IP is an important network protocol that is used to transmit data over networks. In the world of networks, a protocol is a set of rules and procedures that control how the data transfer is carried out. Anyone around the world does the same thing, regardless of the location, programs, or devices used.

ESP32-CAM

ESP32-CAM is a microcontroller, a low-cost development board with a WiFi camera. It allows us to create IP camera projects to stream video with different resolution**.**

Arrow Diagramming Method (ADM)

It is one of the types of network diagrams made by representing activities with arrows, and these stocks are linked to each other through nodes.

Precedence Diagramming Method (PDM)

It is one of the types of network diagrams and activities are represented by nodes, nodes are linked to each other by shares. These arrows show the relationships between activities.

Structured Query Language (SQL)

The language used to perform operations on databases, including adding, updating, or deleting data from the database, or for modifying the database itself.

Android

Android system is defined as an operating system developed by Google.

IBM Watson

It is an artificial intelligence computer system it can answer questions posed in natural language, developed by the DeepQA Project at IBM.

SDK

Stands for a software development kit, or devkit for short. It is a suite of software tools and software that developers use to create applications for specific platforms.