A PROJECT REPORT ON

BABY MONITORING SMART CRADLE USING RASPBERRY PI AND IOT SENSORS

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY

BY

ASHWINI PHALKE	71714649K
IRAJ SHAIKH	71714742J
YUGANDHARA PAWAR	71714646E

Under the guidance of
DR. JAYASHREE V. BAGADE

DEPARTMENT OF INFORMATION TECHNOLOGY
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY
KONDHWA (BK), MAHARASHTRA, INDIA

2019-2020

CERTIFICATE

This is to certify that the project report entitled

BABY MONITORING SMART CRADLE
USING RASPBERRY PI AND IOT SENSORS

Submitted by

Ashwini Phalke	71714649K
Iraj Shaikh	71714742J
Yugandhara Pawar	71714646E

is a bonafide work carried out by them under the supervision of Professor Dr. Jayashree V. Bagade and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology).

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Dr. Jayashree V. Bagade	Dr. Pravin Futane
Internal Guide	Head of Department
Department of Information Technology	Department of Information Technology
	Dr. Vivek S. Deshpande
External Examiner	Director
Date:	Vishwakarma Institute of Information Technology

Place:
Date:

ACKNOWLEDGEMENT

We've put in considerable amount of time and efforts for successful completion of this project. However, it would have been unfathomably difficult to achieve this feat without incredible support and help of many people. We would like to thank to all of them. We also highly grateful to Dr. Jayashree V. Bagade for her constant guidance and supervision along the way in completing the project. Our thanks and appreciations also go to our classmates in developing the project and people who have willingly helped us out with their abilities.

Ashwini Phalke
Iraj Shaikh
Yugandhara Pawar

LIST OF FIGURES

2.1 Infrared radiation of human body 4

2.2 Working of PIR sensor 5

2.3 PIR sensor 6

2.4 DHT11 6

2.5 DHT11 components 7

2.6 Sound sensor 8

2.7 Raspberry Pi 3 Model B 10

2.8 Raspberry Pi Camera Board Module 13

2.9 Camera Setup 14

2.10 Raspberry Pi Software Configuration Tool 14

2.11 Working of JSP 18

3.1 Use case diagram 24

3.2 Class diagram 25

3.3 Activity diagram of register 25

3.4 Activity diagram of add phone number 26

3.5 Activity diagram of sound sensor 26

3.6 Activity diagram of moisture sensor 27

3.7 Activity diagram of PIR sensor 27

3.8 Sequence diagram of register 28
3.9 Sequence diagram of add phone number 28
3.10 Sequence diagram of record data 29
3.11 Sequence diagram of retrieve data 29

4.1 Architecture 30
4.2 Data flow diagram level 0 31
4.3 Data flow diagram level 1 31
4.4 Use case diagram 32
4.5 Class diagram 33
4.6 Activity diagram of register 34
4.7 Activity diagram of add phone number 34
4.8 Activity diagram of sound sensor 35
4.9 Activity diagram of moisture sensor 35
4.10 Activity diagram of PIR sensor 36
4.11 Sequence diagram of register 37
4.12 Sequence diagram of add phone number 37
4.13 Sequence diagram of record data 38
4.14 Sequence diagram of retrieve data 39
4.15 State Machine 39

5.1 Web server and web application project structure 44
5.2 Web application homepage 45

LIST OF TABLES

2.1 Technical Specification for PIR Sensor 4

2.2 Technical Specifications for Humidity and Temperature Sensor 7

2.3 Technical Specifications for sound sensor 8

2.4 Technical Specifications of Raspberry Pi 3 - Model B 10

2.5 Technical Specifications for Raspberry Pi Camera 16

4.1 Table of Sensor Data 32

Contents

1 INTRODUCTION **1**

1.1 MOTIVATION BEHIND PROJECT 1

1.2 AIM AND OBJECTIVES OF THE PROJECT 1

1.2.1 Aim 1

1.2.2 Objectives 2

2 BACKGROUND AND LITERATURE REVIEW **3**

2.1 EXISTING VS PROPOSED 3

2.1.1 Existing Cradles 3

2.1.2 Proposed 3

2.2 TOOLS, SOFTWARE USED 4

2.2.1 PIR Sensor 4

2.2.1.1 Introduction 4

2.2.1.2 Technical Specifications 4

2.2.1.3 Description 5

2.2.2 Temperature and Humidity Sensor (DHT11) 6

2.2.2.1 Introduction 6

2.2.2.2 Technical Specifications 7

2.2.2.3 Description 7

2.2.3 Sound Sensor 8
2.2.3.1 Introduction 8
2.2.3.2 Technical Specifications 8
2.2.3.3 Description 8
2.2.4 Raspberry Pi 3 - Model B 9
2.2.4.1 Introduction 9
2.2.4.2 Technical Specifications 10
2.2.4.3 Raspbian 11
2.2.4.4 Python 3 11
2.2.5 Raspberry Pi Camera 12
2.2.5.1 Introduction 12
2.2.5.2 Camera Details 12
2.2.5.3 Technical Specifications 16
2.2.6 Java 16
2.2.6.1 Android SDK 17
2.2.7 JSP 18
2.2.8 Heroku 18

3 REQUIREMENT AND ANALYSIS 20

3.1 PROBLEM STATEMENT 20
3.2 FUNCTIONAL REQUIREMENTS 20
3.2.1 Functions of the product 20

3.2.2 External Interface 21
3.2.2.1 HARDWARE INTERFACES 22
3.2.2.2 SOFTWARE INTERFACES 22
3.2.2.3 COMMUNICATION INTERFACE 22
3.3 NON-FUNCTIONAL REQUIREMENTS 22
3.3.1 PERFORMANCE REQUIREMENTS: 22
3.3.2 SAFETY REQUIREMENTS: 23
3.3.3 AVAILABILITY 23
3.3.4 CORRECTNESS: 23
3.3.5 USABILITY: 23
3.3.6 MAINTAINTABILITY: 23
3.3.7 PORTABILITY: 23
3.4 ANALYSIS 24
3.4.1 Use case diagram 24
3.4.2 Class diagram 25
3.4.3 Activity diagram 25
3.4.4 Sequence diagram 28

4 DESIGN **30**

4.1 ARCHITECTURE 30
4.2 DATA FLOW DIAGRAM 31
4.3 TABLE OF SENSOR DATA: 32

4.4 UML DIAGRAMS: 32

4.4.1 Use case diagram 32

4.4.2 Class diagram 33

4.4.3 Activity diagram 34

4.4.4 Sequence diagram 37

4.4.5 State Machine 39

5 IMPLEMENTATION **41**

5.1 MODULE 1: SENSORS 41

5.2 MODULE 2: RASPBERRY PI 41

5.3 MODULE 3: WEBSERVER, WEBSITE AND ANDROID APPLICATION 43

6 RESULTS AND EVALUATION **47**

6.1 TEST CASES AND RESULTS 47

6.1.1 Unit tests 47

6.1.2 Integration testing 50

7 CONCLUSION AND FUTURE WORK **52**

7.1 CONCLUSION 52

7.2 FUTURE WORK 53

ABSTRACT

Babies are unpredictable. They need constant supervision and monitoring. There are multiple devices, available in the market, which allow us to monitor the baby. But these devices lack features and are also very expensive. There is a need to develop a new low cost baby monitoring system. This report proposes an indigenous baby monitoring tool which allows the parents and guardians to easily monitor their babies using a website or any Android device. An Raspberry Pi microcontroller was used to collect all of the data from sensors. The moisture sensor detects if the baby has wet the bed, the sound sensor detects if the baby is crying. If the baby moves the motion sensor will detect it. If the parent left the baby sleeping, when the baby moves the parent will be alerted that baby has woken up. If the crying of the baby continues for more than stipulated amount of time, parents will be alerted by SMS and call. Parents can also view the current state of the baby through the live feed on their smartphone. This system aims to aid working parents / hospital nurses in taking care of infants.

CHAPTER 1: INTRODUCTION

In present world most of the parents are busy in their professional life. They do not get sufficient time to take care of their babies. It may be expensive to for a household to afford a nanny. For the purpose of safety something or someone is always required near the baby now technology helps parents by introducing a smart baby cradle. This application provides 24 hours security using cameras. The system is designed to help parents.

1.1 MOTIVATION BEHIND PROJECT

The project idea develops from the fact that a woman finds it difficult to concentrate on her child owing to her busy schedule of house life. The situation get worst when she has job or has some household business ,since she can neither compromise with her work nor she can ignore her child's needs. Many devices are available to ease her task and help her to balance between her work and the needs of her child. Our automated cradle proposes to be one of them. This system updates parent time to time so that they can do their work without taking tension of baby

1.2 AIM AND OBJECTIVES OF THE PROJECT

1.2.1 Aim

The project aims at following points:

1. Send notification to parents when baby starts crying.
2. When the baby's bed was wet then the message will be send to parent's mobile.
3. If the baby is not in proper position then also it informs parent that baby is not in proper position.
4. Camera for live monitoring.

1.2.2 Objectives

- To design the smart baby cradle, this can be monitor baby position, diaper-wet condition, baby cry , body temperature and live monitoring.
- To check whether a baby is safe or not PIR sensor is used to detect the position of baby.
- To check diaper-wet condition to keep baby in hygienic environment.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

This chapter introduce the survey of existing solutions,programs or applications similar to our project.It introduce the software, programming language, library code,frameworks and other tools. It also introduce background information and research.

2.1 EXISTING VS PROPOSED

2.1.1 Existing Cradles

- Need all time care and attention when the baby is in the cradle
- Does not alert parents when baby needs tending to
- Does not allow parents to remotely monitor the baby when they are not near the baby
- Even if smart cradles exist, they are very expensive.

2.1.2 Proposed

- A low cost, high functionality smart cradle
- This cradle will be aware of the baby as opposed to traditional cradles.
- It will take appropriate action depending on the state of the baby.
- It will alert the baby's parents/guardians if anything is not as usual.
- For example, if the baby has been crying for too long (more than 5 min), the cradle will send SMS and call parents.

2.2 TOOLS, SOFTWARE USED

2.2.1 PIR Sensor

2.2.1.1 Introduction

PIR sensor is mainly used in security system to detect the motion of human or position of human body. Infrared (IR) light is electromagnetic radiation with a wavelength between 0.7 and 300 micrometers . Human beings are the source of infrared radiation. It was found that the normal human body temperature radiate IR at wavelengths of 10 micrometer to 12 micrometer . PIR sensors are passive electronic devices which detect motion by sensing infrared fluctuations[1] . It has three pins (gate, drain and source).[After it has detected IR radiation difference, a high input is sent to the signal pin.[1]. A PIR based motion detector (usually called PID, for Passive Infrared Detector) uses this PIR sensor with some additional electronics circuitry for detecting motion. A typical PID sensor gives a logical zero when there is no motion or to the background IR level, and gives a logical one when it detects a hot body motion.

Figure 2.1: Infrared radiation of human body

Source:https://pdfs.semanticscholar.org/cc0d/856116d7b7cdee1a16fbb76

2.2.1.2 Technical Specifications

Item	Value
Operation voltage	4-12V
Operation Current	400µA at 3.3V
PIR Input	68dB
Output Pulse Width	0.5 sec min
Operation Temperature	-20 °C -50 °C

Table 2.1: Technical Specification for PIR Sensor

2.2.1.3 Description

Infrared (IR) radiation is a kind of electromagnetic radiation. Visible light has a lesser wavelength than Infrared light. The infrared has a wavelength of 750 nm to 100μm[2].Human can't see it by naked eye but we can feel it.

Infrared region can further be divided into sub-regions as follows:

- Near Infrared (NIR): 750nm to 1.5μm.
- Short Wavelength Infrared (SWIR): 1.5μm to 3μm.
- Mid Wavelength Infrared (MWIR): 3μm to 8μm.
- Long Wavelength Infrared (LWIR): 8μm to 15μm.
- Far Infrared (FIR): Longer than 15μm.

The MWIR (Mid Wavelength Infrared) and LWIR (Long Wavelength Infrared) is known as the thermal infrared. Black body radiation (thermal radiation) is emit by almost all object [3]. Due to the its temperature, this get emitted from surface of object. Human body at normal body temperature radiates IR approximately at wavelengths around 9.4μm.

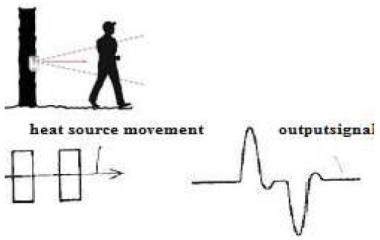


Figure 2.2: Working of PIR sensor

Source:https://pdfs.semanticscholar.org/cc0d/856116d7b7cdee1a16fbb76

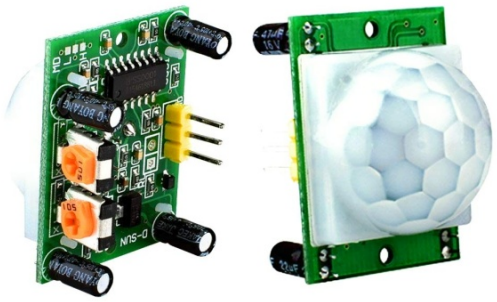


Figure 2.3: PIR sensor

Source:https://pdfs.semanticscholar.org/cc0d/856116d7b7cdee1a16fbb76

2.2.2 Temperature and Humidity Sensor (DHT11)

2.2.2.1 Introduction

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

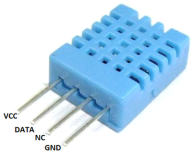


Figure 2.4: DHT11

Source:https://www.indiamart.com/proddetail/dht11-temperature-and-humidity-sensor-19237198597.html

2.2.2.2 Technical Specifications

Item	Value
Measurement Range	20-90%RH 0-50 °C
Humidity Accuracy	±5 RH
Temperature Accuracy	±2°C
Resolution	1
Package	4 Pin Single Row

Table 2.2: Technical Specifications for Humidity and Temperature Sensor

2.2.2.3 Description

DHT11 consists of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate changes or the resistance between these electrodes changes. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller.On the other hand, for measuring temperature these sensors use a NTC temperature sensor or a thermistor.

A thermistor is actually a variable resistor that changes its resistance with change of the temperature. These sensors are made by sintering of semiconductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The term “NTC” means “Negative Temperature Coefficient”, which means that the resistance decreases with increase of the temperature.

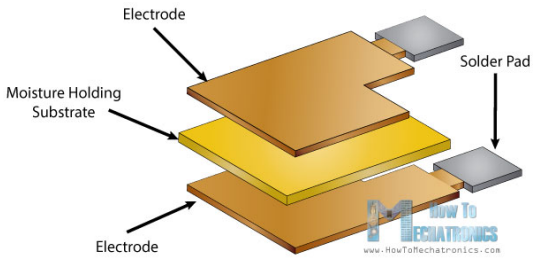


Figure 2.5: DHT11 components

Source:https://howtomechatronics.com/tutorials/RaspberryPi/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-Raspberry Pi/

2.2.3 Sound Sensor

2.2.3.1 Introduction

This module can be used for security, switch, and monitoring applications. Its accuracy can be easily adjusted for the convenience of usage. It uses a microphone which supplies the input to an amplifier, peak detector and buffer[4]. When the sensor detects a sound, it processes an output signal voltage which is sent to a microcontroller then performs necessary processing.

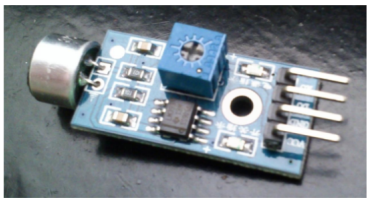


Figure 2.6: Sound sensor

Source:https://pdfs.semanticscholar.org/cc0d/856116d7b7cdee1a16fbb76

2.2.3.2 Technical Specifications

Item	Value
Operating voltage	3.3V-5V
Output model	0 and 1, high or low level
PCB size	3.4cm * 1.6cm
Mounting screw hole	Present

Table 2.3: Technical Specifications for sound sensor

2.2.3.3 Description

Sound sensor works similar to our Ears, Ears have diaphragm which convert vibrations into signals and in Sound sensor we are using Microphone, which will convert sound vibrations into Voltage or Current proportionals. It typically have a diaphragm built inside made up of magnets which are coiled by metal wire.

Whenever sound waves strikes the diaphragm, magnets vibrates and at the same time current is induced from the coil

2.2.4 Raspberry Pi 3 - Model B

2.2.4.1 Introduction

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education[5].

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core 1.4GHz CPU with 1GB RAM. The main price point for Raspberry Pi has always been $35 and all models have been $35 or less, including the Pi Zero, which costs just $5.

All over the world, people use Raspberry Pis to learn programming skills, build hardware projects, do home automation, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things (IoT).

Why Raspberry Pi? Raspberry Pi is the best choice because it provide inbuilt ethernet, WIFI, bluetooth and camera module. It also runs on linux OS so multitasking becomes easy. As it runs on OS we can choose from a plethora of languages like c, c++, python, java and more languages that are supporte in linux. It is inexpensive as it has most of the hardware built in.

Why Raspberry Pi 3 Model B? Currently, the latest models are version 3 model A+ and version 3 model B. All the older models have reached their end of life and no longer supported. Also, it is difficult to find older models in the market. Even if we find one, its not easy to get compatible OS packages to make it operational. Model B has built in wifi and bluetooth which model A doesn't.

2.2.4.2 Technical Specifications

Model	Pi 3 – Model B (Original)
Clock Speed	1.2GHz
Dimensions (mm) LxWxH	85 x 56 x 17
On Board Ports	10/100 BaseT Ethernet socket 15-pin MIPI Camera Serial Interface (CSI-2) 4 x USB 2.0 Connector 40-pin 2.54 mm (2×20 strip) GPIO Combined 3.5 mm audio jack and composite video Composite RCA (PAL and NTSC) HDMI (rev 1.3 & 1.4) Push/pull Micro SD I/O
Operating Power	5V, 2.5~3A
Processor	1.2GHz Quad-Core ARM Cortex-A53 (64Bit) Broadcom BCM2387 chipset.
RAM Memory	1GB LPDDR2
Micro-SD Card Slot	Yes
Weight (gm)	44

Table 2.4: Technical Specifications of Raspberry Pi 3 - Model B

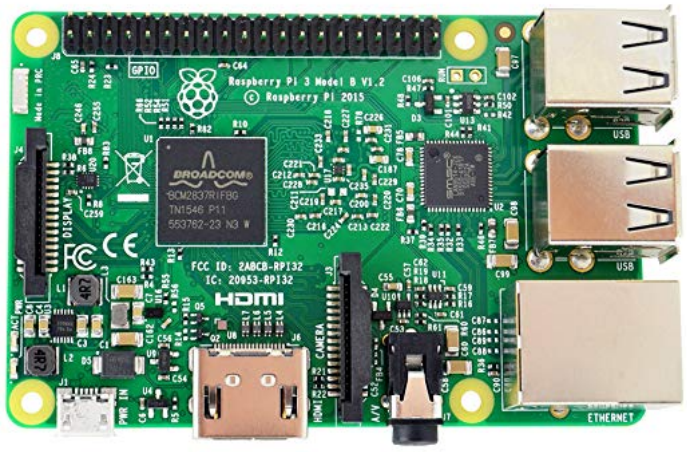


Figure 2.7: Raspberry Pi 3 Model B

Source:https://www.amazon.com/Raspberry-Pi-MS-004-00000024-Model-Board/dp/B01LPLPBS8

2.2.4.3 Raspbian

Raspberry supports multiple operating systems such as ARM-Based Linux Operating Systems, RISC OS Pi, Plan 9, FreeBSD, etc. We will be using Raspbian as it is specifically optimized for Raspberry Pi.

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

To install Raspbian OS, we need to download the NOOB (New Out of Box Software) ISO installer from the Raspberry website. First, we need to mount the ISO on SD card and plug this SD card into the Raspberry Pi slot. When Pi is connected to our screen, it will load install screen from where Raspbian can be installed.

2.2.4.4 Python 3

The Raspberry Pi Foundation recommends Python as a language for learners. Any language which will compile for ARMv6 can be used with the Raspberry Pi, though; so we are not limited to using Python. C, C++, Java, Scratch, and Ruby all come installed by default on the Raspberry Pi. So, we have chosen Python 3 as the language of development.

Python is an object oriented programming language like Java. Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of the Python. Python doesn’t convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So within python, compilation happens, but it’s just not into a machine language. It is into byte code and this byte code can’t be understood by CPU. So we need actually an interpreter called the python virtual machine. The python virtual machine executes the byte codes.

The Python interpreter performs following tasks to execute a Python program :

Step 1 : The interpreter reads a python code or instruction. Then it verifies that the instruction is well formatted, i.e. it checks the syntax of each line.If it encounters any error, it immediately halts the translation and shows an error message.

Step 2 : If there is no error, i.e. if the python instruction or code is well formatted then the interpreter translates it into its equivalent form in intermediate language called "Byte code".Thus, after successful execution of Python script or code, it is completely translated into Byte code.

Step 3 : Byte code is sent to the Python Virtual Machine(PVM).Here again the byte code is executed on PVM.If an error occurs during this execution then the execution is halted with an error message.

2.2.5 Raspberry Pi Camera

2.2.5.1 Introduction

Raspberry Pi camera module is a low cost hardware module that supports still picture and video recording and is the first official hardware add-on for the Raspberry Pi. We'll also look at the camera's features and specifications, how to set it up and examine the software that supports it.

2.2.5.2 Camera Details

Since 2012, the Raspberry Pi Foundation had been reporting that an official camera module was in development. In May 2013, an announcement was made by RS Components and Premier Farnell/Element 14, distribution partners of Raspberry Pi, that the camera module was available (it is also available from other sources) and sells for retail €30 or US$25. The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. The camera module is shown below:

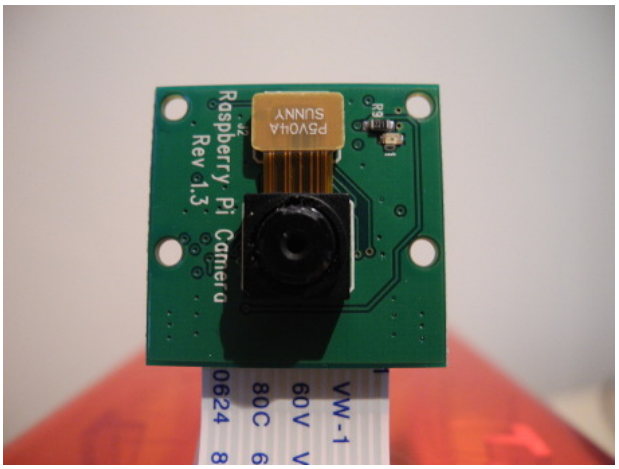


Figure 2.8: Raspberry Pi Camera Board Module

Source:https://www.ics.com/blog/raspberry-pi-camera-module

Installation involves connecting the ribbon cable to the CSI connector on the Raspberry Pi board. This can be a little tricky, but if you watch the videos that demonstrate how it is done, you shouldn't have any trouble.

When you purchase the camera, you will receive a small camera board and cable. You'll want to devise some method of supporting the camera in order to use it. Some camera stands and Raspberry Pi cases are now available. You can also rig up something simple yourself if you wish. I attached mine to a case using a small piece of plastic and double-sided tape, as shown below:

Once the hardware is set up, you can move on to configuring the software.

Software

Since its inception, the camera is supported in the latest version of Raspbian, the preferred operating system for Raspberry Pi. The instructions in this blog post assume you are running Raspbian. The first step is to get the latest Raspberry Pi firmware, which supports the camera. You can do that from a console by running:

sudo apt-get update

sudo apt-get upgrade

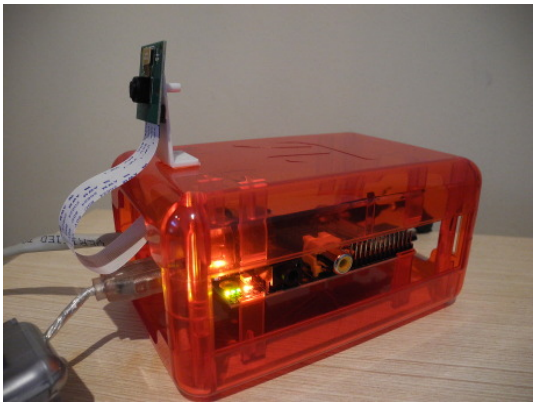


Figure 2.9: Camera Setup

Source:https://www.ics.com/blog/raspberry-pi-camera-module

First: Enable the Camera Begin by making sure you have connected your Raspberry Pi camera to the mini-computer. Next, boot the device, and log in (we're assuming you're using the default Raspberry Pi OS, Raspbian). At the command line, enter sudo raspi-config In the menu select Enable Camera.

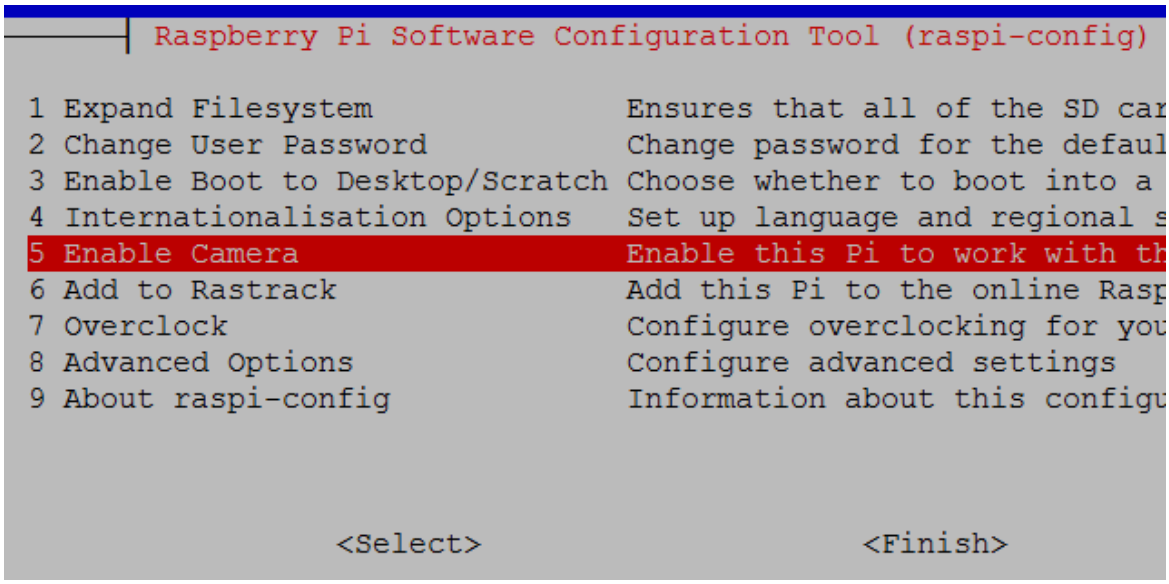


Figure 2.10: Raspberry Pi Software Configuration Tool

Source:https://www.ics.com/blog/raspberry-pi-camera-module

of the Raspberry Pi. should then reboot when prompted by the raspi-config program. The camera will be enabled on subsequent boots Several applications should From here, select Enable, then Finish and Yes to reboot.

Choose "camera" from the program and then select "Enable support for Raspberry Pi camera". You should then reboot when prompted by the raspi-config program. The camera will be enabled on subsequent boots of the Raspberry Pi.

Several applications should now be available for the camera: the rapistill program captures images, raspivid captures videos, and raspiyuv takes uncompressed YUV format images. These are command line program. They accept a number of options, which are documented if you run the commands without options. Several examples are given here.

The following shell command runs the video capture program with a preview showing all of the built-in camera effects and makes for an interesting demonstration:

for effect in none negative solarise sketch denoise emboss oilpaint colourswap washedout posterise colourpoint colourbalance cartoon

  do

  echo $effect

  raspivid -d -ifx$effect

  done hatch gpen pastel watercolour film blur saturation

2.2.5.3 Technical Specifications

Properties	Camera Module v1
Net price	$25
Size	Around 25 × 24 × 9 mm
Weight	3g
Still resolution	5 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90
Linux integration	V4L2 driver available
C programming API	OpenMAX IL and others available
Sensor	OmniVision OV5647
Sensor resolution	2592 × 1944 pixels
Sensor image area	3.76 × 2.74 mm
Pixel size	1.4 µm × 1.4 µm
Optical size	1/4"
Full-frame SLR lens equivalent	35 mm
S/N ratio	36 dB
Dynamic range	67 dB @ 8x gain
Sensitivity	680 mV/lux-sec
Dark current	16 mV/sec @ 60 C
Well capacity	4.3 Ke-
Fixed focus	1 m to infinity
Focal length	3.60 mm +/- 0.01
Horizontal field of view	53.50 +/- 0.13 degrees
Vertical field of view	41.41 +/- 0.11 degrees

Table 2.5: Technical Specifications for Raspberry Pi Camera

2.2.6 Java

We have selected Java as the development language for both Android application and web application.

Why Java?

1. Applications developed with Java are fairly small i.e. they create lighter APKs as compared to applications developed using any of the above languages.
2. Applications developed with Java are more efficient and without any bugs.

3. Crashes are few and far between on user‘s device.

4. Java is cross platform.

5. Java has the same behaviour on all types of software be it a browser, a phone or virtual machine. So, we can rest assured that behaviour of the program will be same irrespective of the platform used. This allows us to reuse code.

6. Android SDK itself consists of standard Java libraries. So, the integration is tighter and we can reap maximum benefits from it.

7. Learning curve is short and easy.

8. Java is secure. Everything gets executed inside the Java Virtual Machine (JVM).

We’ll be using Android SDK to develop Android application

2.2.6.1 Android SDK

The Android SDK includes nearly all development tools required for Android.

1. **Platform tools** The default platform tools version we get with Android SDK is the latest version of tools. Each version of platform tools is backward compatible i.e. it allows us to develop apps for current Android version as well as previous Android versions [6].

2. **Build tools** The Build tools were once part of the Platform tools but they now have been moved to separate package. Now, they can be updated and upgraded individually irrespective of platform tools version. As the name suggests, these are prerequisites to building an Android application. This includes a tool for e.g. which optimizes the app to use least amount of memory. It also includes a tool which signs the APK, this is similar generating md5 sum for files.

3. **ADB** Similar to gdb (GNU project debugger) which allows us to debug C and C++ programs. ADB is a tool that allows us to talk with any Android machine running our application. It is part of the Platform-tools package. It depends on Platform package for the to and fro communication with Android device. Android Debug Bridge can be used to reach command line tools such as logcat, to perform various operations.

4. **Android emulator** Android applications are rendered differently on different android devices depending on their screen sizes, dimensions and Android versions. The Android emulator makes it easy for us to test our app on wide spectrum of devices without necessarily having the device at hand[6]. It also lets us monitor the applications while running on the simulated devices. Android Virtual Device manager allows us to choose which flavour of Android we would like to emulate, with wide variety of options to select the device specifications including but not limited to screen size and performance, etc.

2.2.7 JSP

To build a web application, we will be using Java, JSP, HTML, JavaScript and CSS.

Why JSP?As it is browser independent, it is easy to maintain i.e. no need to change code even if new versions of browser are released or if we need to run the website on old browsers.Allows to make websites with dynamic content. As JSP internally uses Java, it allows us to create platform agnostic web pages.

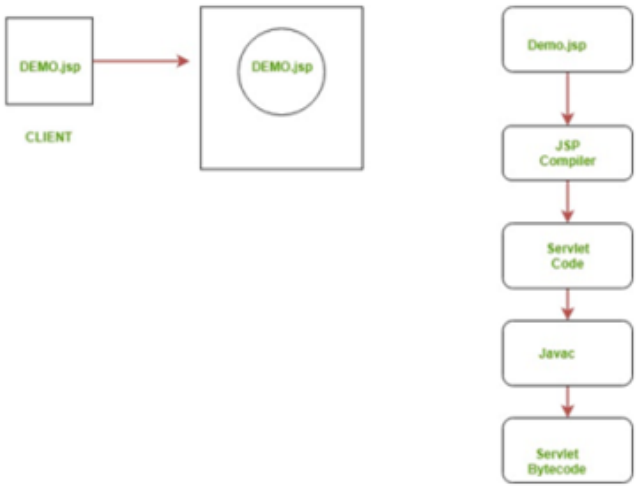


Figure 2.11: Working of JSP

2.2.8 Heroku

The website needs to be hosted on a server to be accessible by hitting the URL from any network connected device. Heroku will be used for this task.

Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. In Heroku, we can specify the number of dynos or virtual servers that our application will be served upon[7].

Why Heroku? Heroku's free account provides 550 free dyno hours without even adding credit card details as opposed to Google App Engine and Amazon AWS. Heroku CLI is easy to use and hides much of the complexity[8].

CHAPTER 3: REQUIREMENT AND ANALYSIS

The Requirement specifications determine specific feature expectation , resolution of conflict or ambiguity in requirements as demanded by various users. This chapter introduce the requirements of the system and also it shows how this requirements will be accomplished. Requirement analysis involves all tasks that are conducted to identify the needs of different users.

3.1 PROBLEM STATEMENT

To create a smart cradle which monitors babies' activities placed in it such as crying, diaper moisture,temperature and movement.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 Functions of the product

1. Function: Detect parameters of the baby
   Description: Sensors detect the state of the baby: humidity of diaper, sound of baby, temperature of the baby, baby's motion.
   Input: humidity, sound, temperature, motion.
   Source: Baby
   Action: Sensors detect the state of the baby

2. Function: Record parameters of the baby
   Description: Raspberry Pi records the state of the baby: humidity of diaper, sound of baby, temperature of the baby, baby's motion.
   Input: humidity, sound, temperature, motion.
   Source: PIR sensor, humidity sensor, sound sensor, temperature sensor LM350, CCTV.
   Action: Raspberry Pi records the state of the baby

3. Function: Analysis of parameters of the baby
   Description: Raspberry Pi records the state of the baby: humidity of diaper, sound of baby, temperature of the baby, baby's motion. Based on these, Raspberry Pi performs the following actions: Call and SMS if the baby continues crying for more than 5 min and if baby is not in cradle
   Input: humidity, sound, temperature, motion
   Source: PIR sensor, humidity sensor, sound sensor, temperature sensor LM350, CCTV
   Action: Raspberry Pi takes action based on the values received from sensors

4. Function: Send parameters of the baby to the web server
   Description: Raspberry Pi sends the state of the baby to webserver
   Input: humidity, sound, temperature, motion
   Source: Raspberry Pi
   Action: Raspberry Pi sends the state of the baby to webserver using REST APIs

5. Function: Get parameters of the baby from the web server
   Description: Android app and web application gets the state of the baby from web server
   Input: humidity, sound, temperature, motion
   Source: webserver
   Action: App gets data from web server using REST APIs

3.2.2 External Interface

The parent must download the app and generate a unique url by entering mobile number. The user can enter a maximum of 3 mobile numbers for a single cradle. The generated URL can also be accessed using any browser or the user can access the state of the baby using the app itself. The user can input data with the help of the keyboard or click with the mouse wherever necessary. The app provides pull down menus from which the user can select and links and icons to navigate among the web pages. In the user interface, the user should be able to view following parameters: a) Sound b) Motion c) Moisture d) Live video of the baby

3.2.2.1 HARDWARE INTERFACES

1. PIR sensor
2. Humidity Sensor
3. Sound sensor
4. CCTV
5. Raspberry Pi

3.2.2.2 SOFTWARE INTERFACES

We have chosen Windows operating system for its best support and user- friendliness. Java has the same behavior on all types of software be it a browser, a phone or virtual machine. So, we can rest assured that behavior of the program will be same irrespective of the platform used. This allows us to reuse code. As JSP internally uses Java, it allows us to create platform agnostic web pages. Heroku is being used to host web server. It is highly available and more reliable than other services.

3.2.2.3 COMMUNICATION INTERFACE

The communication between between Raspberry Pi and csv file, Android Application and csv file happens using REST apis. REST stands for Representational State Transfer. Rest specifies a set of rules that will be followed to communicate over http protocol between two systems connected to the internet.

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 PERFORMANCE REQUIREMENTS:

The data represented to the parents should not be older than 10 secs. The system should perform near instant updates to the web server.

3.3.2 SAFETY REQUIREMENTS:

- Quality of information

- Integrity of information

3.3.3 AVAILABILITY

Measure of how likely the system is available for use. Takes repair/restart time into account. Availability of 0.998 means software is available for 998 out of 1000 time units. Relevant for continuously running systems e.g. telephone switching systems. The system must be up 24 hours per day The state of the baby should always be available to parents. There should be no interruption in service.

3.3.4 CORRECTNESS:

The data sent to the parents should always be correct.

3.3.5 USABILITY:

The data should be easily accessible to both parents and all the guardians through phone or website

3.3.6 MAINTAINTABILITY:

Software maintenance should be done once every six months.

3.3.7 PORTABILITY:

Software is only compatible with Windows, Linux.

3.4 ANALYSIS

3.4.1 Use case diagram

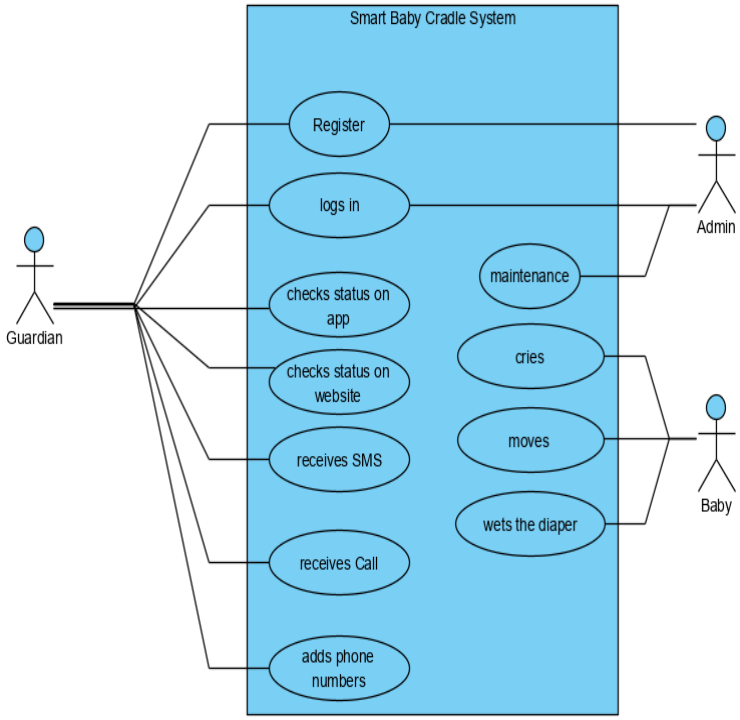


Figure 3.1: Use case diagram

3.4.2 Class diagram

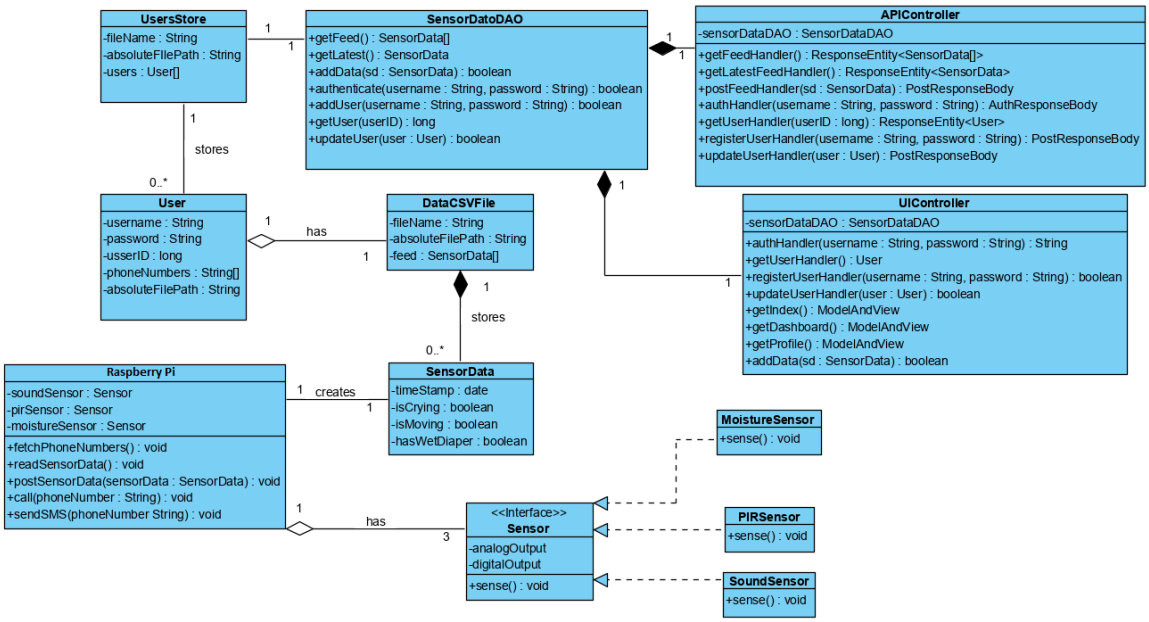


Figure 3.2: Class diagram

3.4.3 Activity diagram

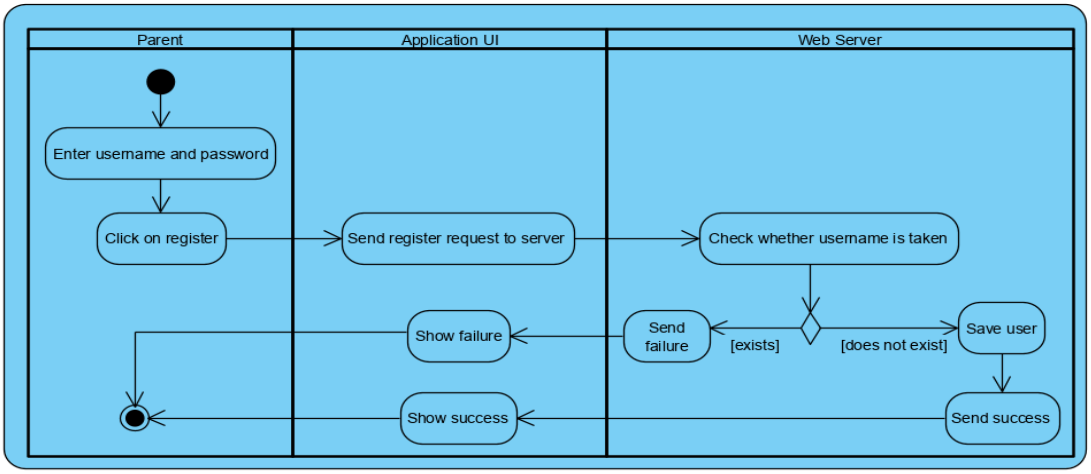


Figure 3.3: Activity diagram of register

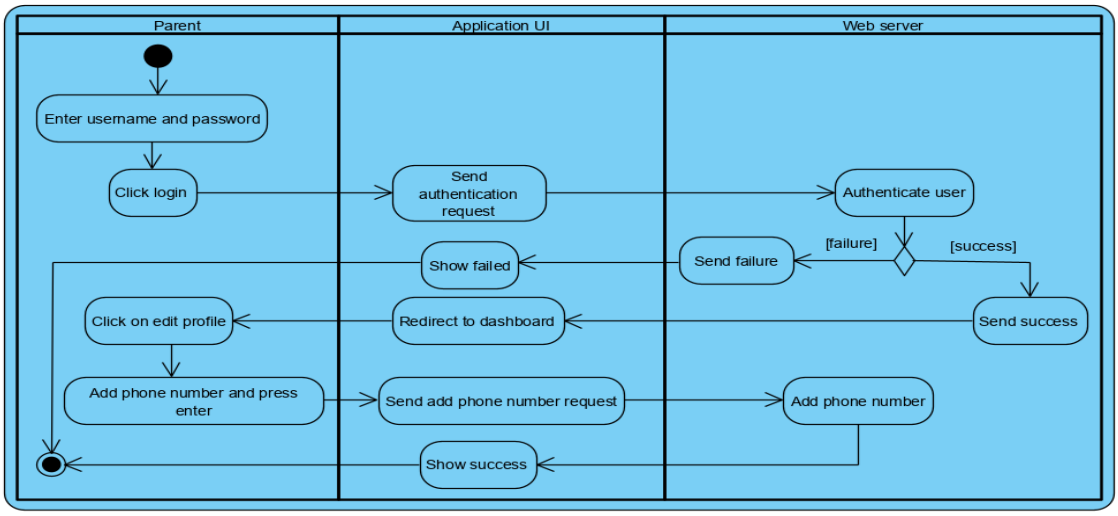


Figure 3.4: Activity diagram of add phone number

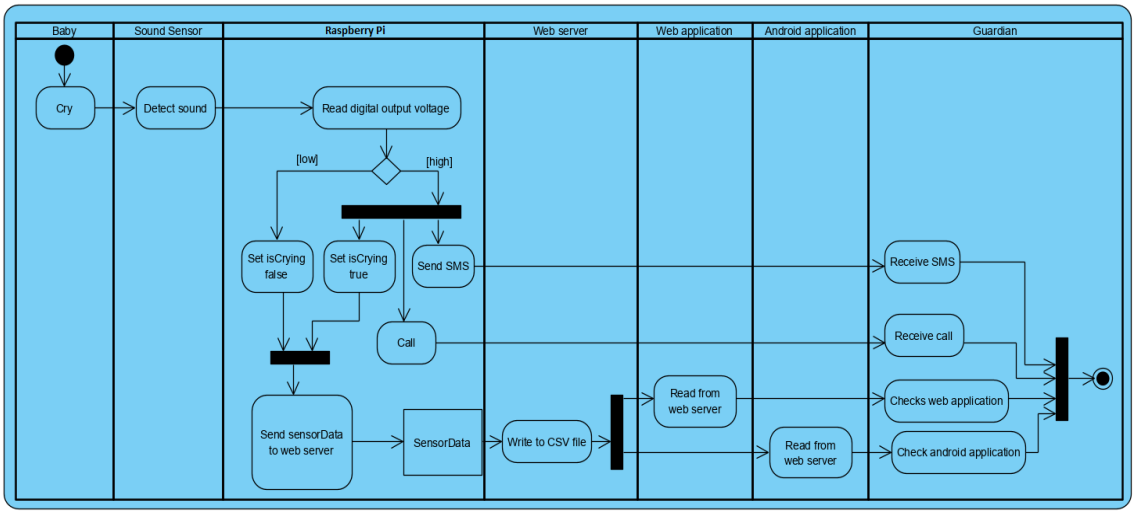


Figure 3.5: Activity diagram of sound sensor

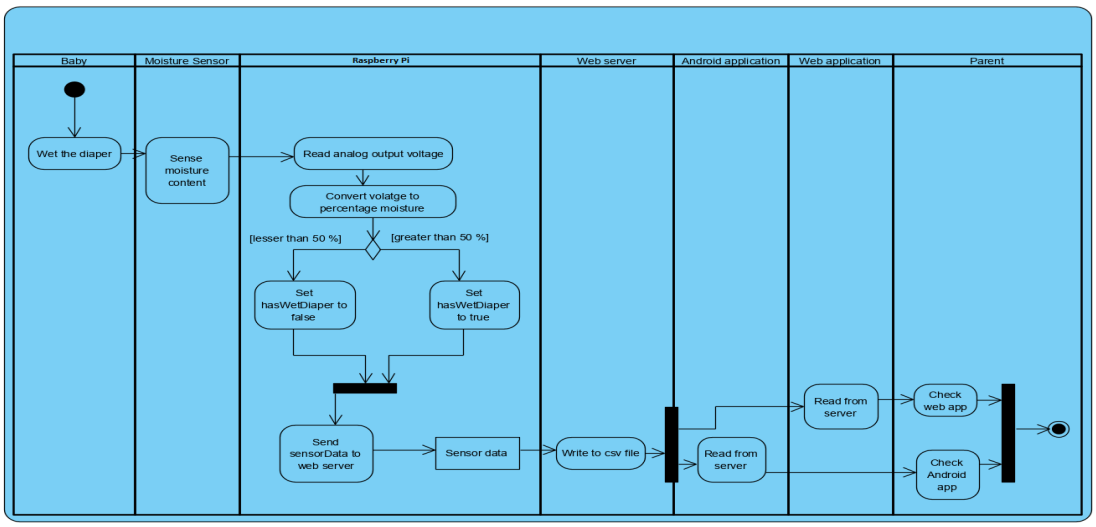


Figure 3.6: Activity diagram of moisture sensor

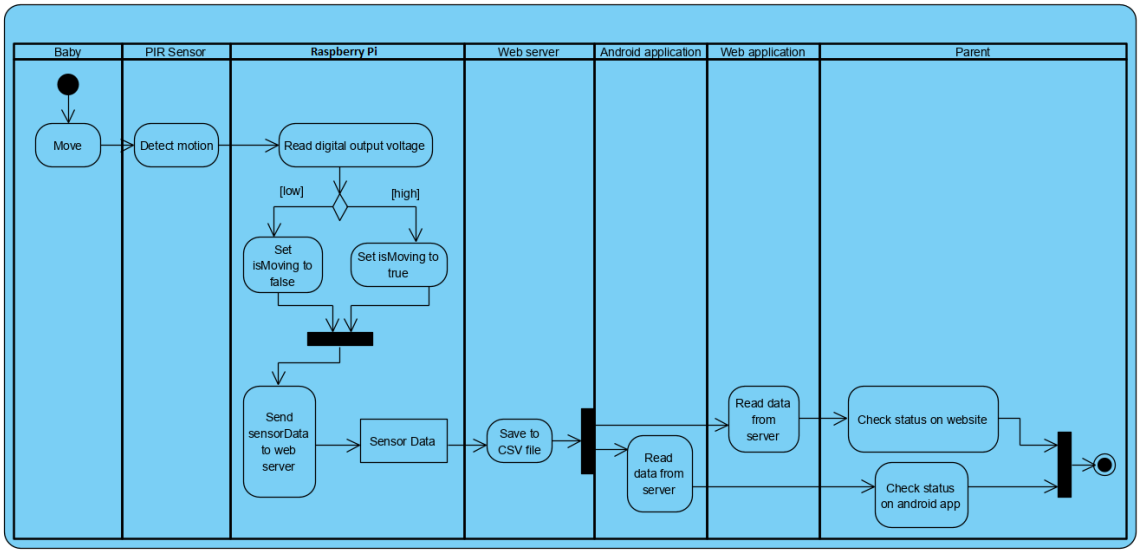


Figure 3.7: Activity diagram of PIR sensor

3.4.4 Sequence diagram

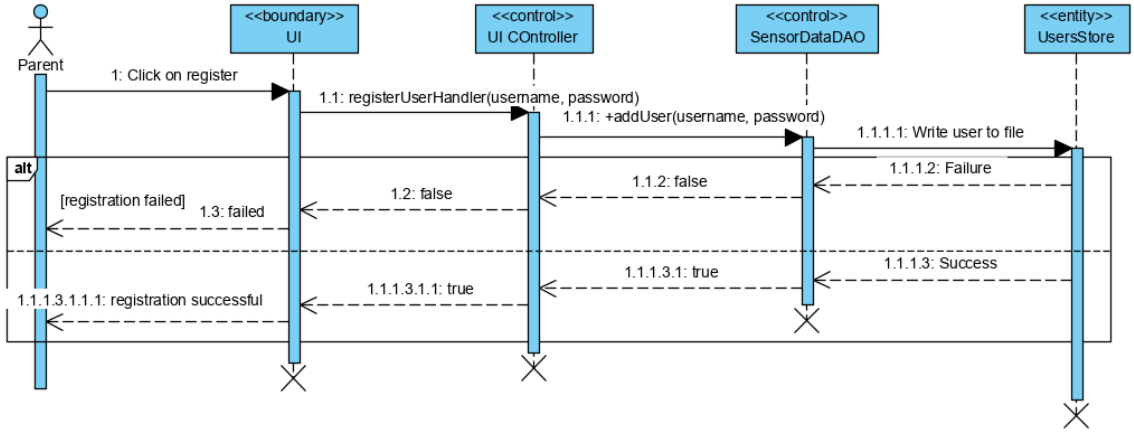


Figure 3.8: Sequence diagram of register

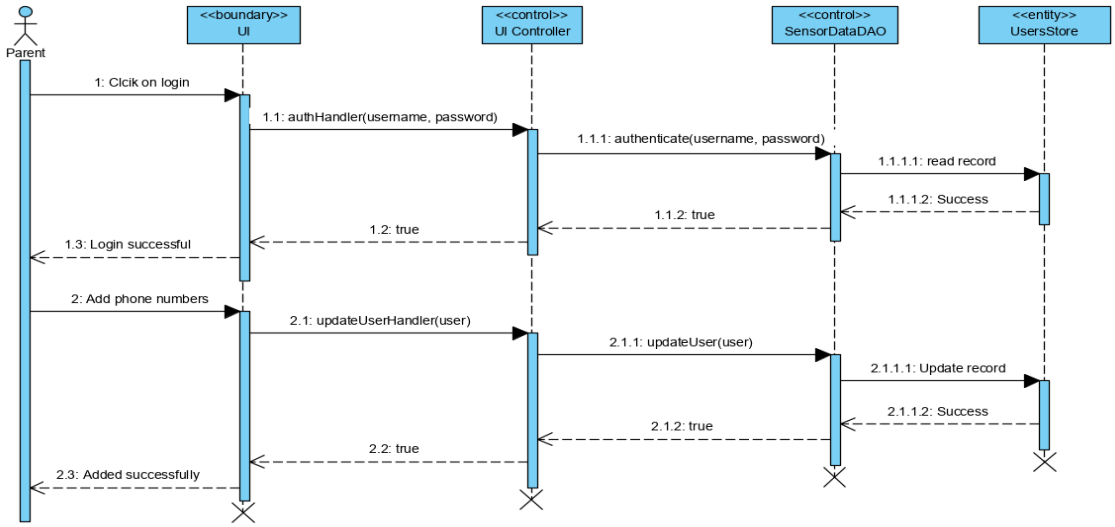


Figure 3.9: Sequence diagram of add phone number

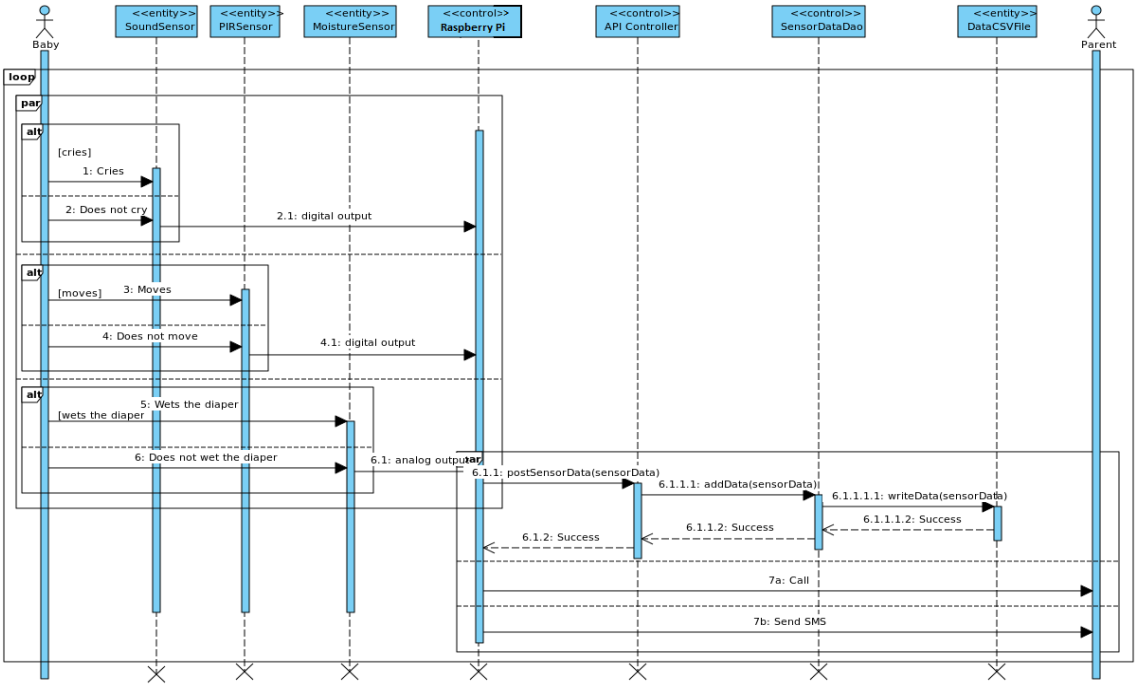


Figure 3.10: Sequence diagram of record data

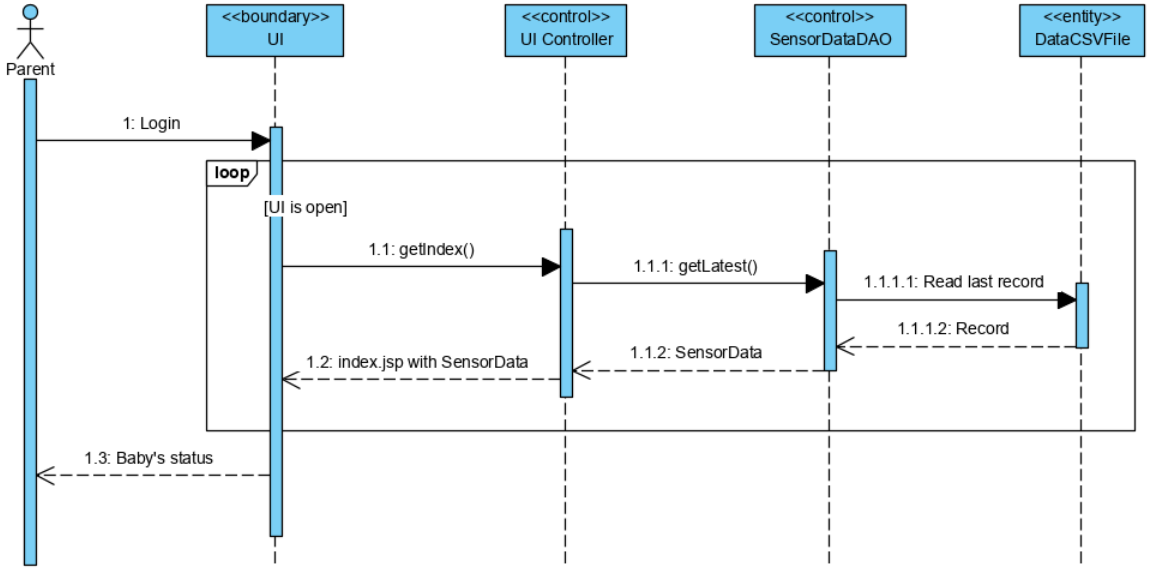


Figure 3.11: Sequence diagram of retrieve data

CHAPTER 4: DESIGN

This chapter introduce the basic architecture of our project and describe all the components of it. And also it shows the necessary DFDs and UML diagrams with proper explanation. It gives the high level overview of our project.

4.1 ARCHITECTURE

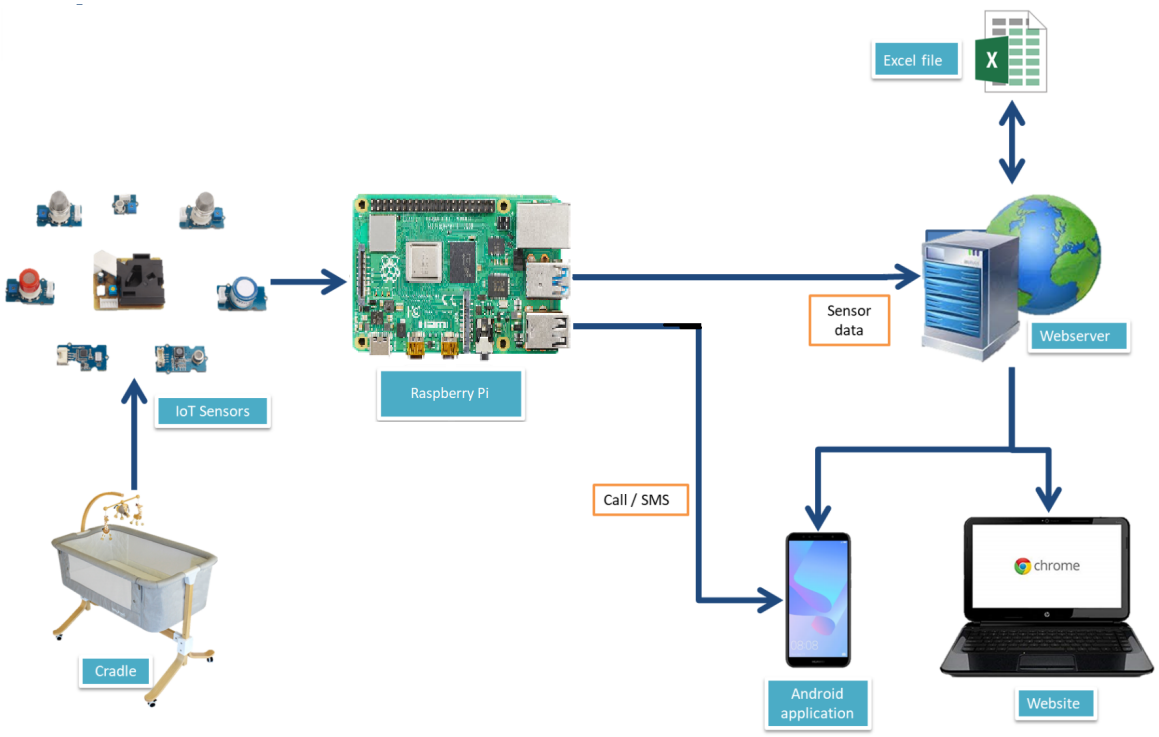


Figure 4.1: Architecture

4.2 DATA FLOW DIAGRAM

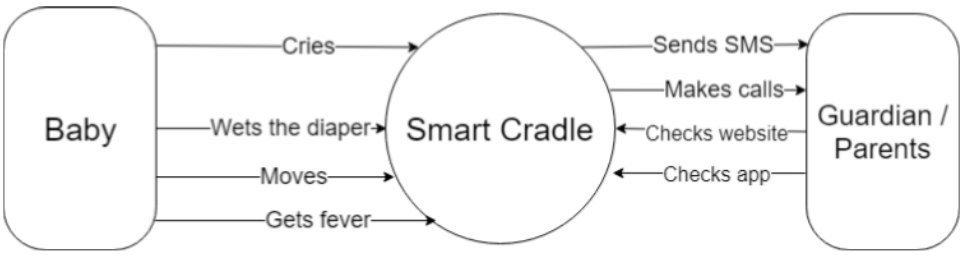


Figure 4.2: Data flow diagram level 0

DFD Level-0 : It only contains one process node (Process 0) that generalizes the function of the entire system in relationship to external entities.

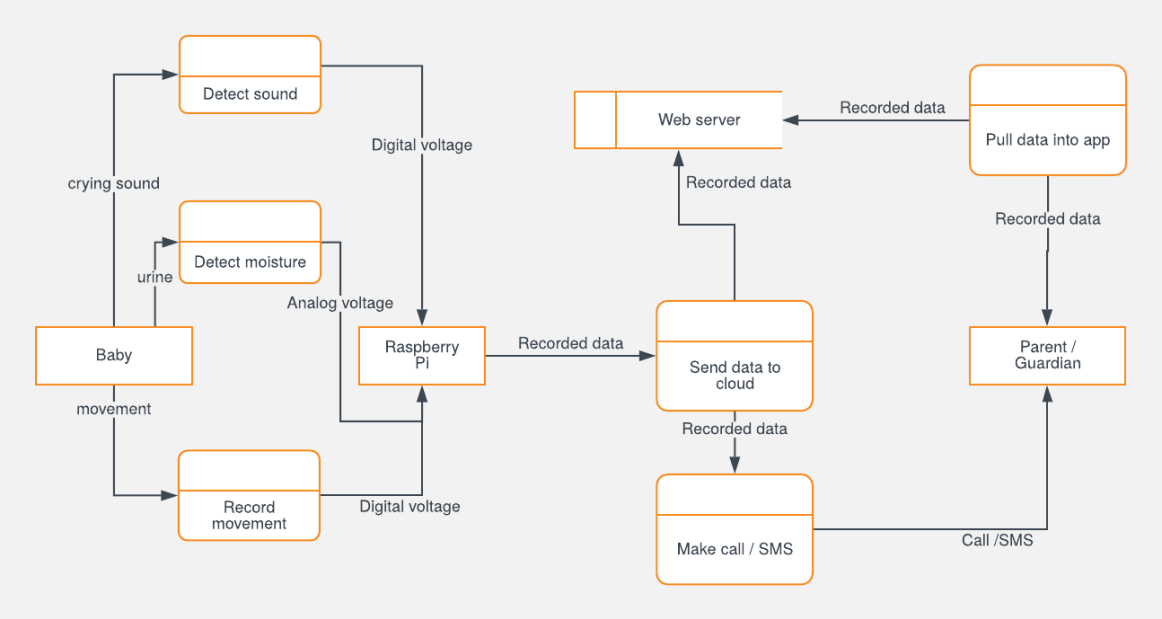


Figure 4.3: Data flow diagram level 1

DFD Level-1: DFD level 1 diagram expands the DFD 0 and shows the detailed flow of the proposed system.

4.3 TABLE OF SENSOR DATA:

Web server stores the csv file of sensor data. It also provides REST endpoints for Raspberry Pi and mobile application to send and receive data respectively. It is basic tomcat server which will be hosted on heroku. Below is the sample sensorData file:

IsMoving	HasWetDiaper	IsCrying
FALSE	FALSE	FALSE
TRUE	FALSE	FALSE
FALSE	FALSE	TRUE

Table 4.1: Table of Sensor Data

4.4 UML DIAGRAMS:

4.4.1 Use case diagram

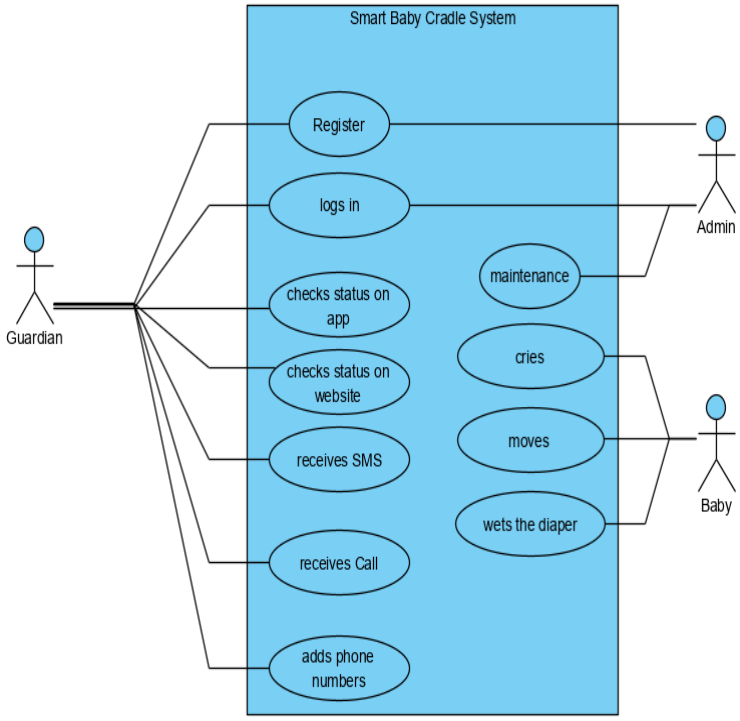


Figure 4.4: Use case diagram

Above use case diagram figure 4.4 is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.Guardian can interact with the system using login credentials.Guardian can check status of baby on app or website.crying, moving and wetting the diaper these are activities perform by baby. Admin is responsible for the maintenance of the system.

4.4.2 Class diagram

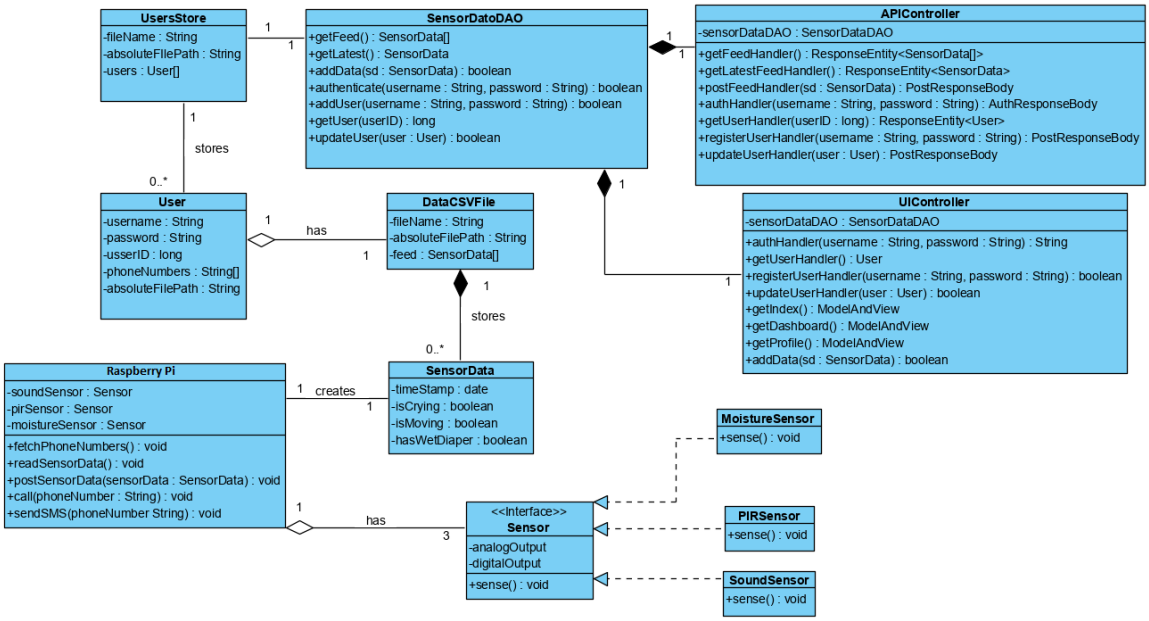


Figure 4.5: Class diagram

Class diagram gives the static view of an application . A class diagram describe the type of the object in the system and the different types of relationship that exist among them.The data csv file stores the sensor data but it can only be accessed by the data access layer (DAO). So, when Raspberry Pi wants to write data to the file it does this through the API controller which in turn talks to DAO.To show the data in UI , UI controller reads the data from the file by talking to DAO.

4.4.3 Activity diagram

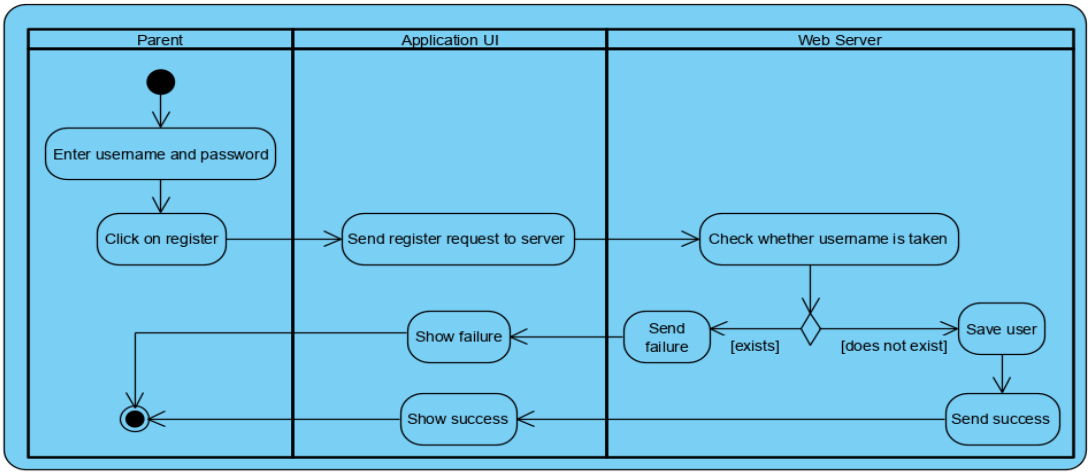


Figure 4.6: Activity diagram of register

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. Above activity diagram describes the flow of activities to register. Parent has to enter the username and password then click on the register button.

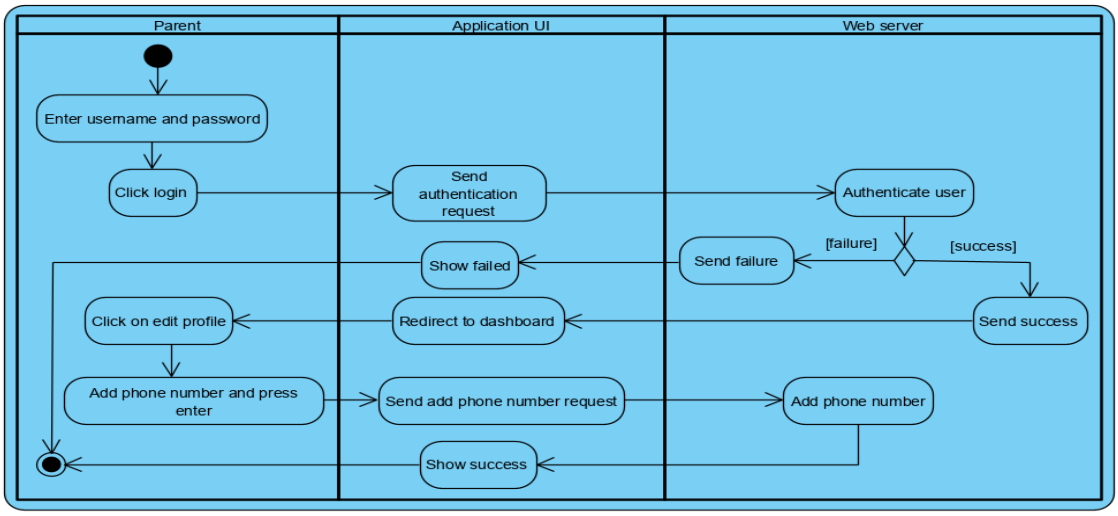


Figure 4.7: Activity diagram of add phone number

An Activity diagram visually presents a series of actions or flow of control in a system similar to flowchart or data flow diagram. Above activity diagram represents flow of activities to add phone number in the system. Parent has to login using username and password. If the username and password are correct then parent has to click on the edit profile and then he/she will be able to add phone number.

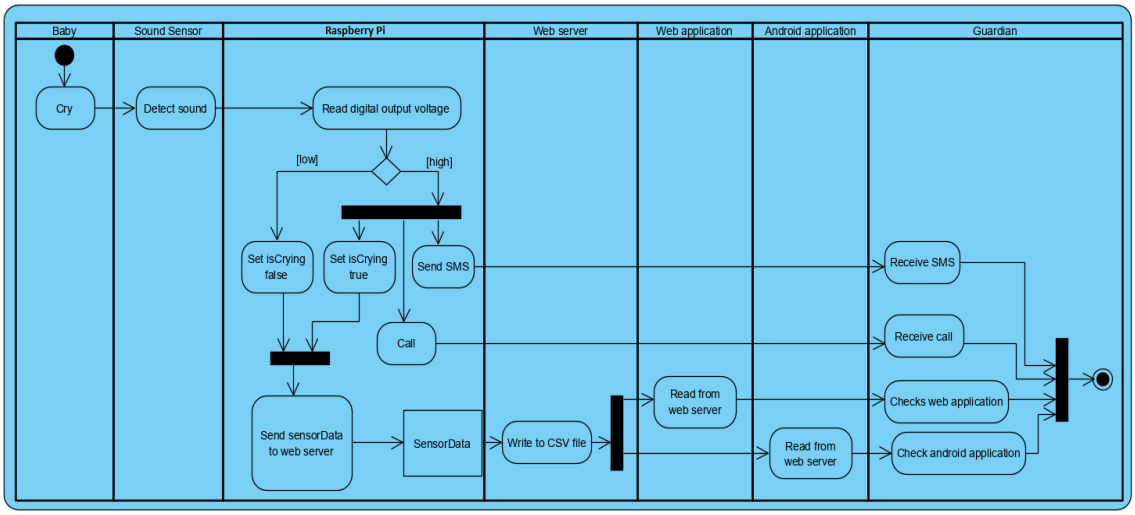


Figure 4.8: Activity diagram of sound sensor

Above diagram figure 4.8 represents the activity flow for sending message to parent that baby is crying.When sound sensor detects sound, it gives digital high voltage at output pin. Raspberry Pi reads this digital high voltage and sends it to webserver.

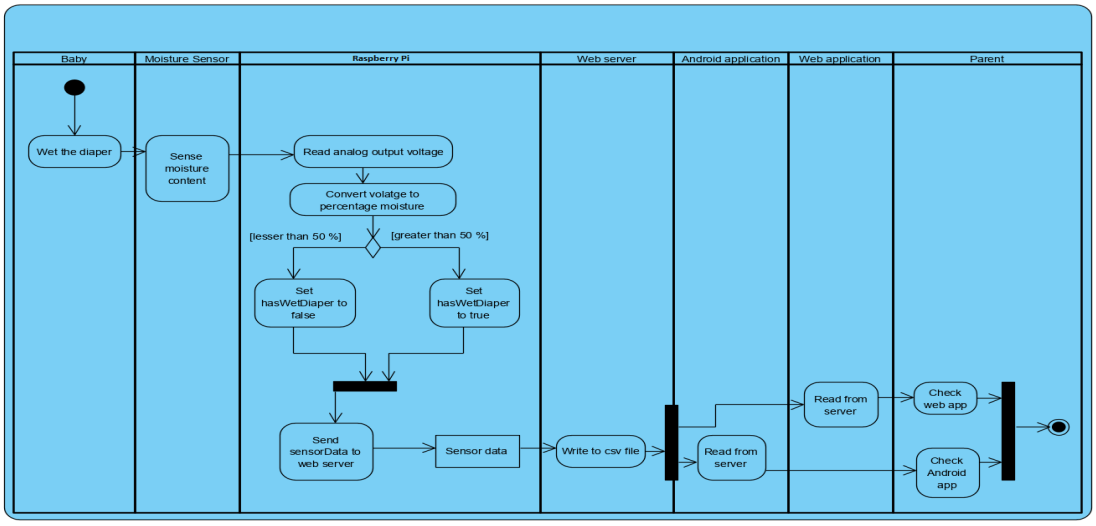


Figure 4.9: Activity diagram of moisture sensor

When moisture sensor calculates moisture content, it outputs analog voltage between 0 and 1023 on its output pin. Raspberry Pi reads this voltage and converts it to percentage using formula: (Recorded/ 1024)*100. If this is greater than 50, that means the diaper is wet.If this is less than 50, that means the diaper is not wet. The Raspberry Pi then transfers this data to web server.

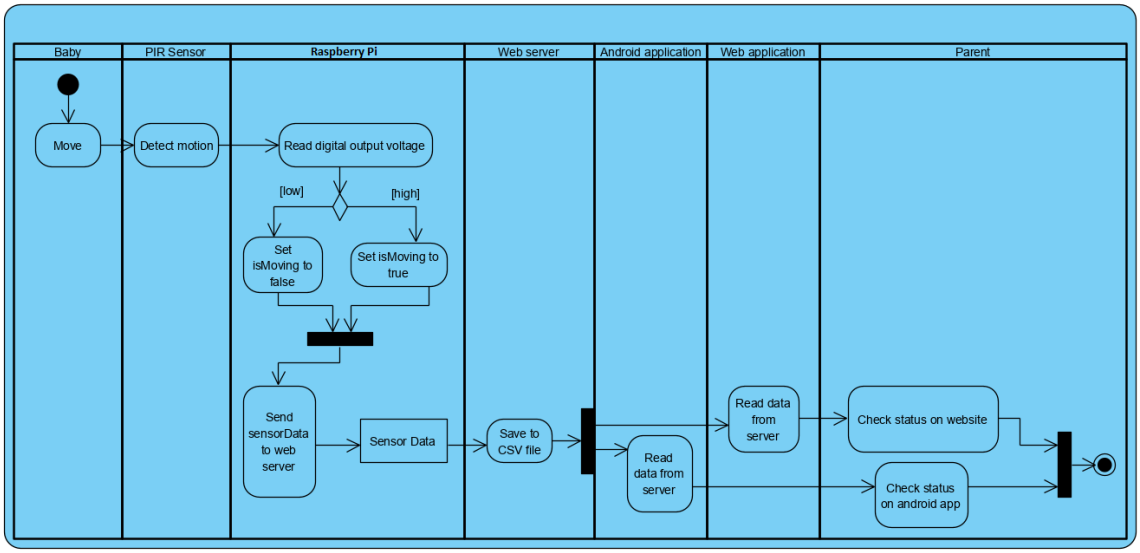


Figure 4.10: Activity diagram of PIR sensor

It outputs digital voltage.When voltage is high, that means baby is moving. If voltage is low it means baby is not moving.

4.4.4 Sequence diagram

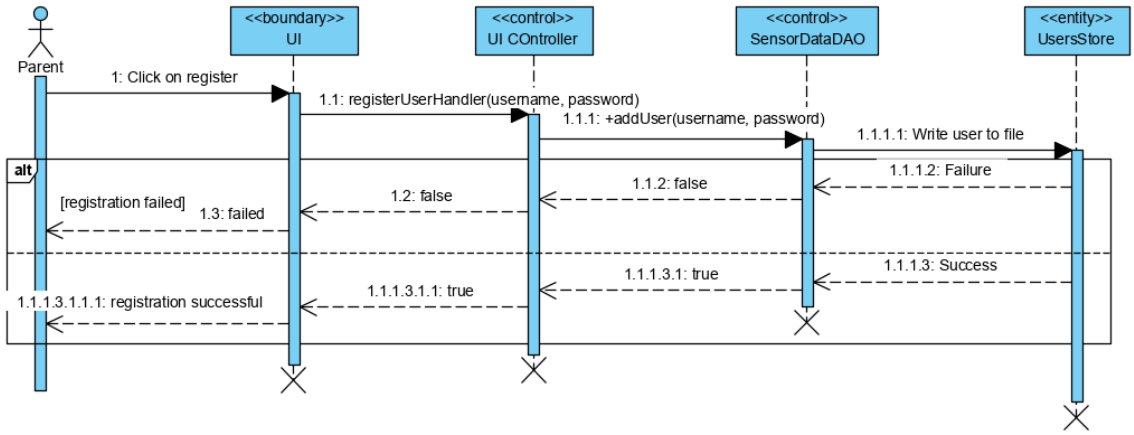


Figure 4.11: Sequence diagram of register

A sequence diagram simply depicts interaction between objects in a sequential order i.e order in which these interaction takes place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagram describe how and in what order the objects in a system function. Above sequence diagram resembles how different objects like parent,UI,UIcontroller,sensorDataDAO and UsersStore interacts with each other.

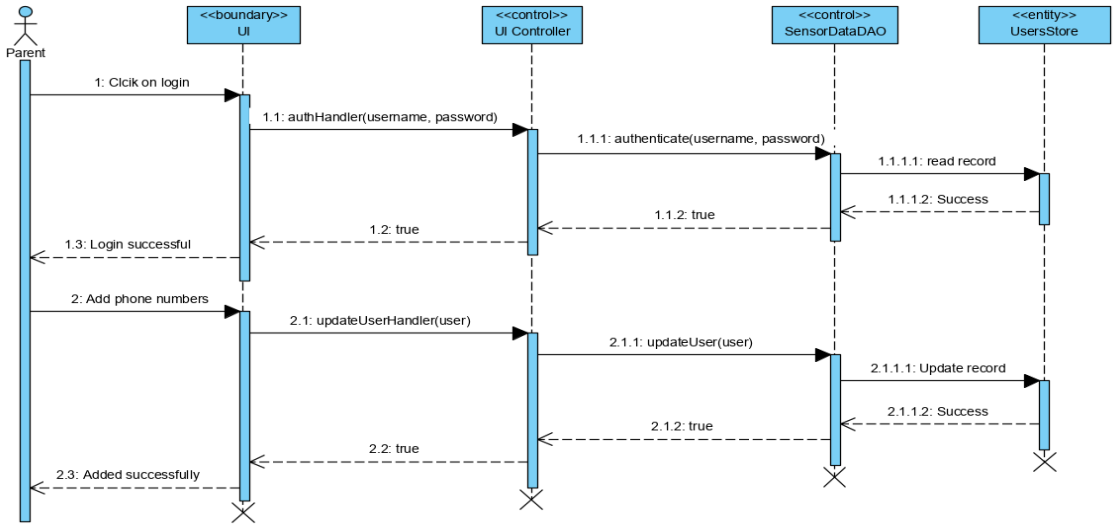


Figure 4.12: Sequence diagram of add phone number

A Sequence diagram is an interaction diagram that shows how processes operate with one

another and in what order. Sequence diagram are sometimes called event diagrams or event scenarios.The sequence diagram for the proposed system shows the interaction in between different components.Above sequence diagram represents the sequential flow to add phone number.

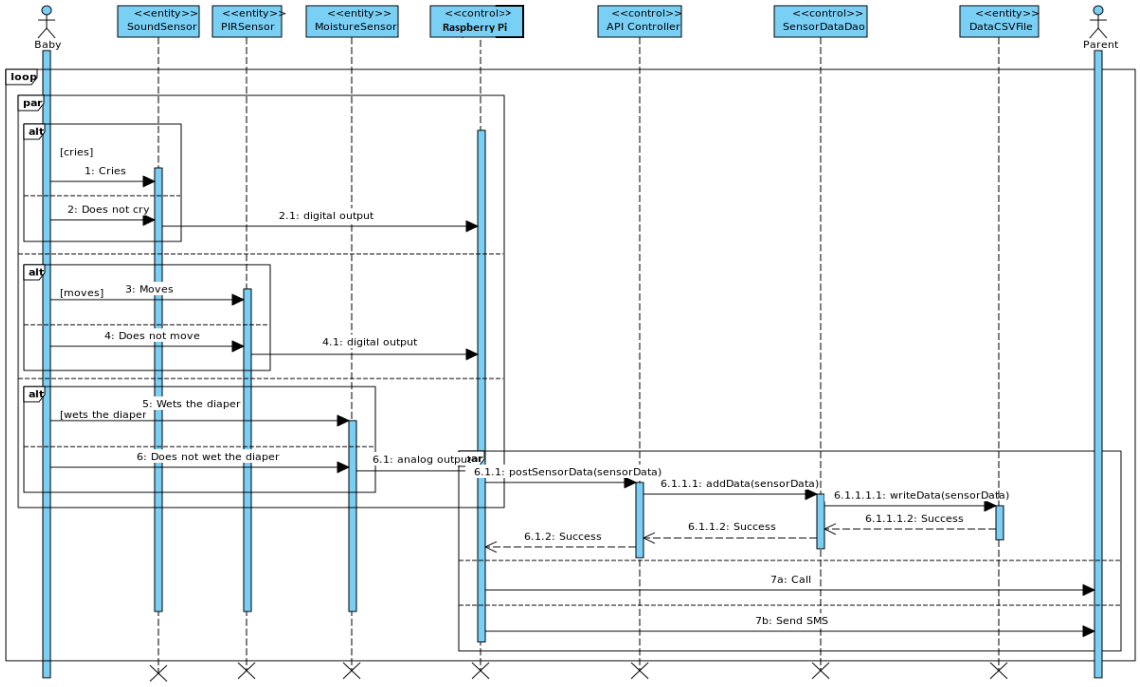


Figure 4.13: Sequence diagram of record data

While recording data, the following entities come into play: 1) Raspberry Pi 2) API controller 3) Sensordata DAO. Every 15 seconds, the sensors capture the state of the baby. Moisture, sound and motion are the parameters that are recorded.Raspberry Pi then forms HTTP POST request with url of web server and the body as baby parameters in json format.This request is sent to web server using gsm gprs module.API controller in web server serves as handler for this request. It reads the json and converts to string format and sends it to dao layer.DAO layer then writes this data to csv file.Before writing it checks whether file exists.If it does not exist, it creates one.

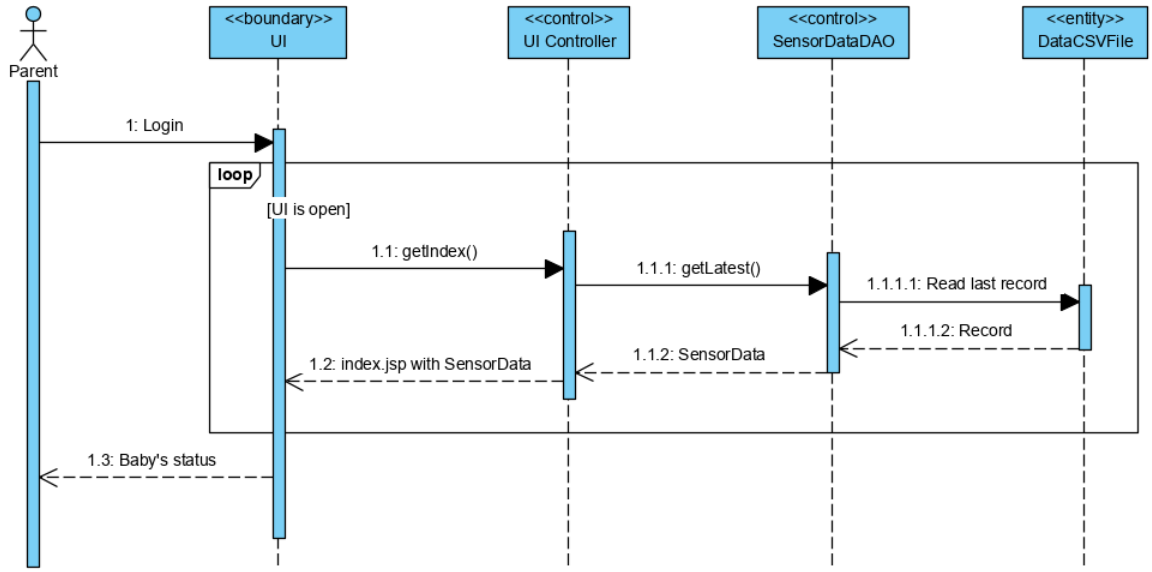


Figure 4.14: Sequence diagram of retrieve data

Sequence diagram of retrieve data: for the data to be shown in Android application. It first needs to be fetched from csv file. Android application creates a REST http request with URL as web server url of method GET. API controller in web server serves as handler for this request. API controller on receiving this request invokes dao layer. DAO layer in turn fetches data from csv file and returns the data in response body.If the file does not exist HTTP status code 500 if returned 500 signifies internal server error.

4.4.5 State Machine

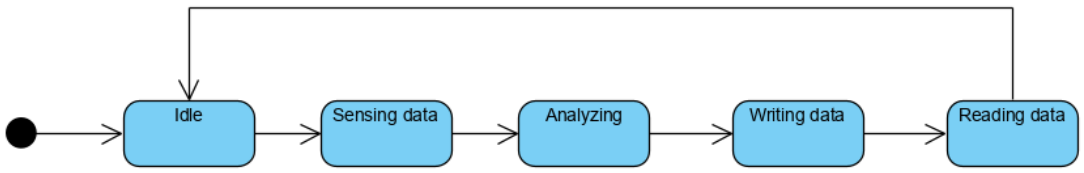


Figure 4.15: State Machine

State machine diagram is a behavior diagram which shows discrete behavior of a system through finite state transaction.State machine diagram is also called as state chart.These

diagram are used to capture the behavior of a protocol.It represents how the state of system changes concerning the event.It also represents corresponding changes in the system.

CHAPTER 5: IMPLEMENTATION

The system consists of 3 main parts: 1) Sensors, 2) Raspberry Pi and 3) Web sever. Raspberry Pi periodically captures data from sensors and sends a POST rest API call over http. Pi also makes call and sends sms when it detects baby crying for more than 5 min. The website and the android app fetch the data from the server using GET API call and display it. All the data sent to the server gets stored in an excel sheet.

5.1 MODULE 1: SENSORS

The following sensors:

1. Picam
2. Temparature Sensor (DHT11)
3. Sound sensor
4. Soil moisture sensor

have been connected to Raspberry PI General input output pins

5.2 MODULE 2: RASPBERRY PI

Raspberry records video and captures mages from picam sensor

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview(fullscreen=False, window = (550, 200, 640, 480))
#camera.start_preview()
sleep(5)
camera.capture('/home/pi/Desktop/image.jpg')
#camera.stop_preview()
```

Raspberry Pi records temperature every 15 seconds from DHT11 sensor

```
import sys import Adafruit_DHT
while True:
    humidity, temperature = Adafruit_DHT.read_retry(11, 4)
    print ('Temp: {} C  Humidity: {} %'.format(temperature, humidity))
```

It also records sound and deducts that baby is crying if sound continues for 30s or more

```
import RPi.GPIO as gpio
import time
sound_sensor = 21
gpio.setmode(gpio.BCM)
gpio.setup(sound_sensor,gpio.IN)
count=0
startTime = time.time()
while(True):
        value = gpio.input(sound_sensor)
        crtTime = time.time()
        nTime = crtTime-startTime
        if( value == 1):
                while(nTime < 10):
                        crtTime = time.time()
                        nTime = crtTime-startTime
                        value = gpio.input(sound_sensor)
                        if( value == 1):
                                count += 1
                                print(count)
                                if(count == 30):
                                        print('Baby is crying')
                                        count = 0
                startTime = crtTime
                count = 0
                print("Time out")
```

It records moisture from soil moisture sensor

```
import RPi.GPIO as gpio
import time
moist_sensor = 20
gpio.setmode(gpio.BCM)
gpio.setup(moist_sensor, gpio.IN)
while(True):
        value = gpio.input(moist_sensor)
        print(value)
        if (value == 0) :
                print("Baby's diaper is wet")
        else :
                print("Baby's diaper is not wet")
        time.sleep(2)
```

Raspberry then finally sends this data to web server by sending below REST API request

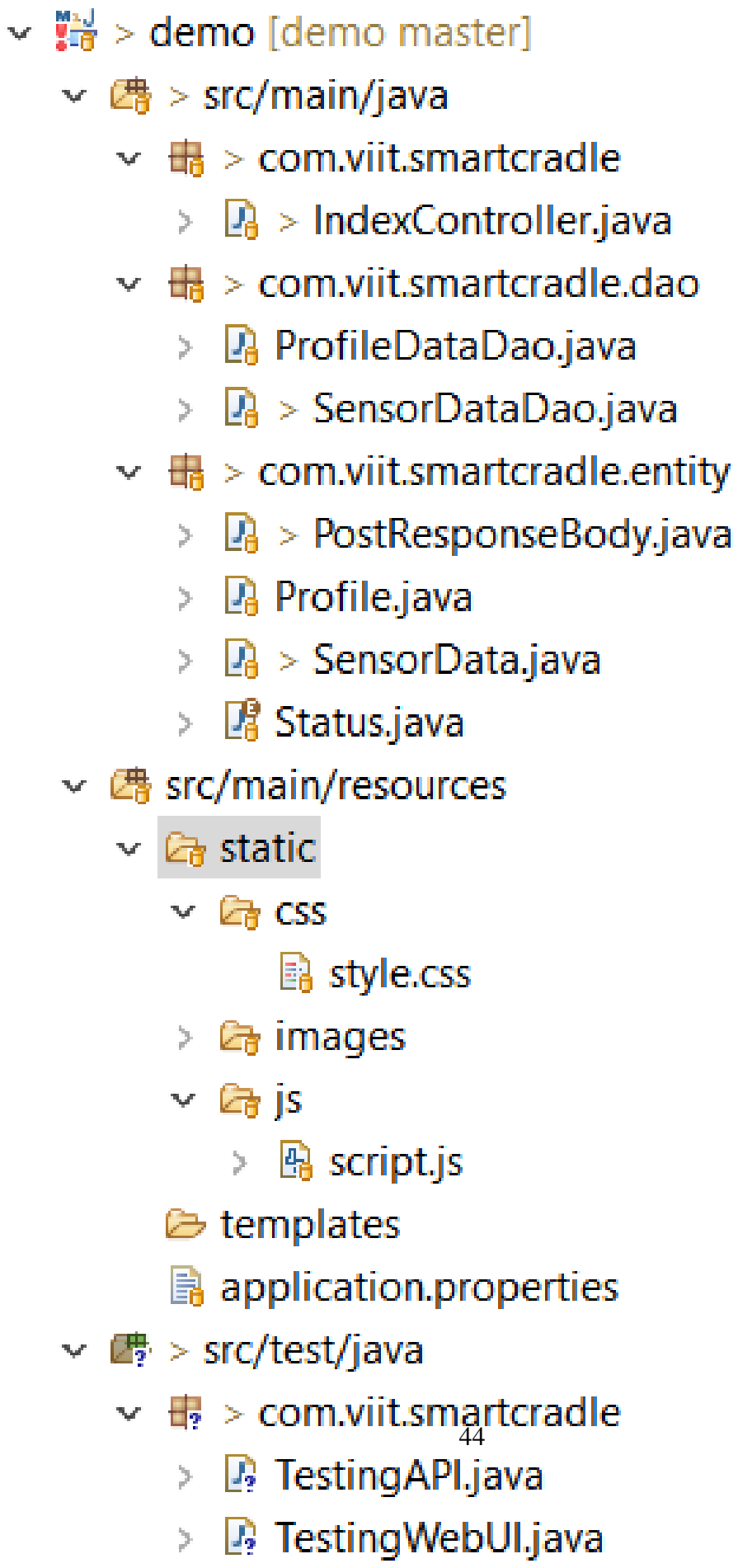
POST on http://smartcradle20.herokuapp.com/api/feed with sample body of request as

{ "temperature": 40.0, "isMoving": false, "hasWetDiaper": true, "isCrying": true }

5.3 MODULE 3: WEBSERVER, WEBSITE AND ANDROID APPLICATION

We have selected Java as the development language for both Android application and web application. The website needs to be hosted on a server to be accessible by hitting the URL from any network connected device. Heroku will be used for this task. Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. In Heroku, we can specify the number of dynos or virtual servers that our application will be served upon. [8] Why Heroku? Heroku's free account provides 550 free dyno hours without even adding credit card details as opposed to Google App Engine and Amazon AWS. Heroku CLI is easy to use and hides much of the complexity.

The web server is implemented using Spring boot framework. It follows model view architecture:


demo [demo master]
src/main/java
com.viit.smartcradle
IndexController.java
com.viit.smartcradle.dao
ProfileDataDao.java
SensorDataDao.java
com.viit.smartcradle.entity
PostResponseBody.java
Profile.java
SensorData.java
Status.java
src/main/resources
static
css
style.css
images
js
script.js
templates
application.properties
src/test/java
com.viit.smartcradle
TestingAPI.java
TestingWebUI.java

MVC model:

Here, Sensordata is the model which structures the values of sensors. It needs to be serialize as it will be written to csv file using CSVWriter java package.

```
public class SensorData implements Serializable {
        private static final long serialVersionUID = 1L;
        private double temperature;
        private boolean hasWetDiaper;
        private boolean isCrying;
}
```

MVC view:

Here, we have only one view that the index.jsp page:

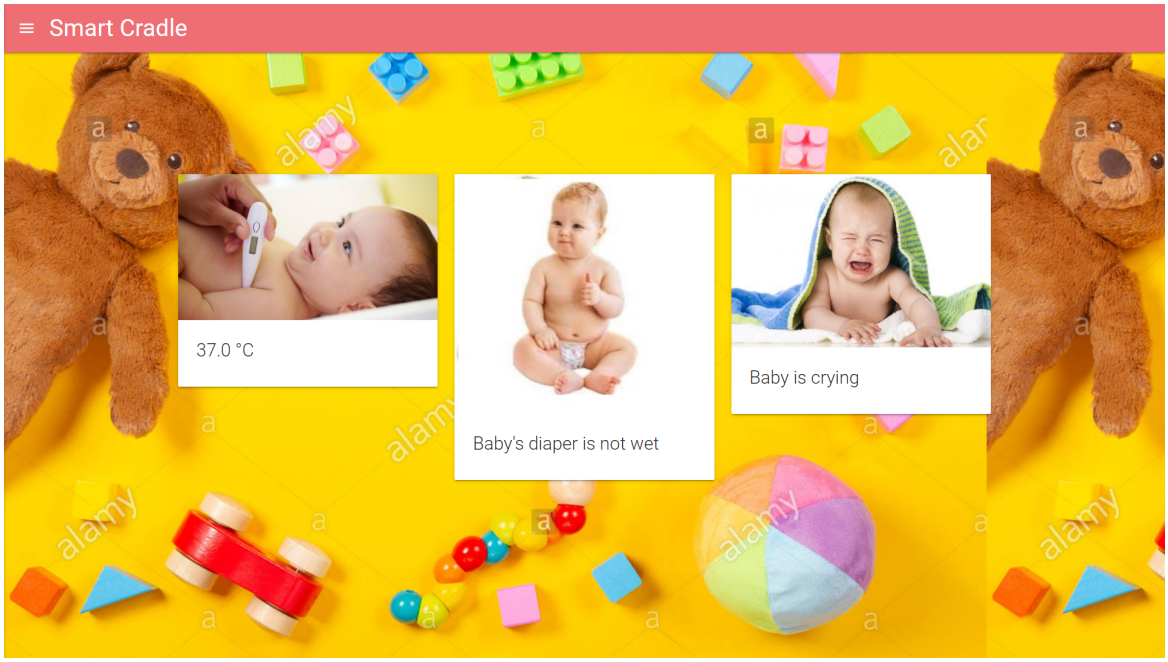


Figure 5.2: Web application homepage

MVC controller:

Here, we have IndexController which handles the following requests

1. GET on /: This is for the browser requests. It returns the index.jsp view with data from the DAO layer to get the latest sensor data from the CSV file

2. GET on /api/feed: This is for the Android application. It lets the app pull the latest data from CSV file

3. POST on /api/feed: This is used by Raspberry PI to post new data to the application

CHAPTER 6: RESULTS AND EVALUATION

6.1 TEST CASES AND RESULTS

6.1.1 Unit tests

Following test case have been added for API testing:

1. getLatestShouldReturnSensorData checks whether the API returns latest data returned by DAO layes
2. getLatestShouldReturnInternalServerErrorOnException checks whether the API return HTTP status code as 500 when server runs into exception because of file not found
3. addDataShouldReturnStatusSuccess tests whether POST on /api/feed returns status success when data is added to file
4. addDataShouldReturnBadRequestWhenTempIsNotDouble: This is for checking validation of fields i.e. it checks whether API returns HTTP Status code bad request if the temperature is not a number

Following test case have been added for web ui testing:

1. shouldReturnDashboardAsHomePage: This ensures that index.jsp view is returned when we hit "/" URL.

```
@WebMvcTest(IndexController.class) public class TestingAPI {
        @Autowired
        private MockMvc mockMvc;
        @MockBean
        private SensorDataDao sensorDataDao;
        @MockBean
        private ProfileDataDao profileDataDao;
```

```
    @Test
    public void getLatestShouldReturnInternalServerErrorOnException()
            throws Exception {
            when(sensorDataDao.get()).thenThrow(new IOException());
            this.mockMvc.perform(get("/api/feed"))
            .andExpect(status().is5xxServerError());
    }

    @Test
    public void getLatestShouldReturnSensorData() throws Exception {
            SensorData sensorData = new SensorData("35", "true", "false");
            this.mockMvc.perform(get("/api/feed"))
            .andExpect(status().isOk())
            .andExpect(content()
            .json(
            "{\"temperature\":35.0,\"hasWetDiaper\":true,\"isCrying\":false}"));

    }

    @Test
    public void addDataShouldReturnStatusSuccess() throws Exception {
            SensorData sensorData = new SensorData("35", "true", "false");
            .contentType(MediaType.APPLICATION_JSON)
            .content(
            "{\"temperature\":\"35\",\"hasWetDiaper\":true,\"isCrying\":false}"))
            .andExpect(status().isOk());
    }

    @Test
    public void addDataShouldReturnBadRequestWhenTempIsNotDouble()
            throws Exception {
            SensorData sensorData = new SensorData("35", "true", "false");
            this.mockMvc.perform(
            MockMvcRequestBuilders.post("/api/feed")
            .contentType(MediaType.APPLICATION_JSON)
            .content(
            "{\"temperature\":\"35s\",\"hasWetDiaper\":true,\"isCrying\":false}")
            .andExpect(status().is4xxClientError());
    }

}
```

```
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
public class TestingWebUI {
        @LocalServerPort
        private int port;
        @Autowired
        private TestRestTemplate restTemplate;
        @Test
        public void shouldReturnDashboardAsHomePage()
                throws Exception {
                assertThat(
                        this.restTemplate
                        .getForObject("http://localhost:" + port + "/",
                                String.class)).contains("temp.jpg");
        }

}
```

Following test case have been added for raspberry pi testing:

For Raspberry Pi unit testing, the sensor data is mocked using unittest.mock package

1. testWhetherIsCryingIsTrueWhenSoundDetected
2. testWhetherIsCryingIsTrueWhenSoundNotDetected
3. testWhetherSMSSentWHenTemperatureGreaterThan37C
4. testWhetherHasWetDiaperIsTrueWhenMoistureDetected
5. testWhetherHasWetDiaperIsFalseWhenMoistureNotDetected

RESULTS:

Maven generates reports using surefire:

Test set: com.viit.smartcradle.TestingWebUI

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6.013 s - in com.viit.smartcradle.TestingWebU

Test set: com.viit.smartcradle.TestingWebUI

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6.013 s - in com.viit.smartcradle.TestingAPI

6.1.2 Integration testing

The following scenarios have been covered in integration testing:

1. Crying baby sound, wet diaper, temperature < 35
    - The web UI should display the corresponding values correctly
    - SMS should not be sent
2. Crying baby sound, dry diaper, temperature < 35
    - The web UI should display the corresponding values correctly
    - SMS should not be sent
3. Crying baby sound, wet diaper, temperature > 35
    - The web UI should display the corresponding values correctly
    - SMS should be sent
4. Crying baby sound, dry diaper, temperature > 35
    - The web UI should display the corresponding values correctly
    - SMS should be sent
5. No sound, wet diaper, temperature < 35
    - The web UI should display the corresponding values correctly

- SMS should not be sent

6. No sound, dry diaper, temperature < 35
    - The web UI should display the corresponding values correctly
    - SMS should not be sent
7. No sound, wet diaper, temperature > 35
    - The web UI should display the corresponding values correctly
    - SMS should be sent
8. No sound, dry diaper, temperature > 35
    - The web UI should display the corresponding values correctly
    - SMS should be sent

CHAPTER 7: CONCLUSION AND FUTURE WORK

This chapter gives the information what we improve in the existing system .It gives overall summary of what our project has achieved. It conclude what we have done until now and what will future scope of our system.

7.1 CONCLUSION

Our project is Smart baby cradle .It is used for taking care of new born babies smart baby cradle is designed. The smart baby cradle is the best answer to today's parent's needs. In smart baby cradle different types of sensors are used. The PIR sensor is use to detect the position of the baby, sound sensor is use to detect the cry of the baby, humidity sensor is used to detect the wetness of diaper and the camera is used to monitor the baby. This system allows working moms and dads to do household chores along with caring for the baby .The automatic baby cradle can be used in home. It is very useful for working parents to take care of babies. It is economical and user friendly.

Technology has been developed in a great way that it makes human work simpler. So, in that aspect to convenient the baby care smart baby cradle has been designed. The automatic electronic baby cradle is the finest solution for today's parents who cannot find the sufficient time for their babies. This automatic baby cradle would let the working mother to do household works besides taking care of baby at the same time. It is economical and user friendly. The Smart baby cradle can be used in home. It is very useful for working parents .The cradle is capable of detecting the movement ,moisture of diaper,temperature of body, humidity,and also use for live monitoring. The device is use to minimize the work of parents.

We have understood how to build, run and deploy android application and the web application.We studied about Raspberry Pi hardware and software in this report. Now,we can easy implement Raspberry Pi in smart baby cradle system.

This study gave us the prerequisites of getting hands on experience handling IoT data.

7.2 FUTURE WORK

Currently, we support only Android OS, we will have app for IoS as well later We are not taking any advantage of the data collected. In the future, we will draw conclusions from the data and make predictions We are not physically moving the cradle now, later we will add this functionality.

REFERENCES

[1] G. Nepal, R. Biswa, D. Adhikari, P. Chodon, S. Gyeltshen, and Chencho, "Passive Infrared (PIR) Sensor Based Security System," *International Journal of Electrical, Electronics and Computer Systems*, vol. 14, pp. 772–778, 06 2013.

[2] B. Toreyin, E. Soyer, O. Urfalioglu, and A. Cetin, "Flame detection using PIR sensors," pp. 1 – 4, 05 2008.

[3] B. Song, H. Choi, and H. S. Lee, "Surveillance Tracking System Using Passive Infrared Motion Sensors in Wireless Sensor Network," in *2008 International Conference on Information Networking*, pp. 1–5, Jan 2008.

[4] M. H. M. Blea, "Automatically rocking baby cradle," 1972.

[5] "What is a Raspberry Pi?." https://opensource.com/resources/raspberry-pi (accessed Oct. 25, 2019).

[6] "What is Android?," (accessed Feb. 15, 2019). https://developer.android.com/about (accessed Feb. 15, 2019).

[7] R. Degges, "The heroku hacker's guide," 2012.

[8] J. Solheim, "Web apps in the computer science curriculum: a guide using heroku, java servlets, and postgres," *Journal of Computing Sciences in Colleges*, vol. 30, no. 5, pp. 126–133, 2015.

[9] P. Jamieson, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?."

[10] H. S. Dishart, "Baby cradle-like carrier," August 1966.

[11] M.D.Bass, "Analysis and Design of Structures," *Journal of Department of Electrical and Electronic Engineering*, 2016 (accessed 18 March 2019).

[12] M. P. Thuillard, "Method for analyzing the signals of a danger alarm system and danger alarm system for implementing said method," Jan. 2000.

[13] *Sound Sensor How to Use it?* http://invent.module143.com/sound-sensor-how-to-use-it/ (accessed Feb. 15, 2019).

[14] A. Zahariev, *Google app engine*. Helsinki University of Technology, 2009.

[15] "Raspberry Pi 4 Tech Specs." https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/ (accessed Oct. 25, 2019).

[16] X. L. Sheikh Ferdoush, "Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications," *Procedia Computer Science*, vol. 34, pp. 103–110, 2014.

[17] J. Holton and T. Fratangelo, "Raspberry pi architecture," *Raspberry Pi Foundation, London, UK*, 2012.