**A PRELIMINARY PROJECT REPORT**

**ON**

**SAFE - OXI: A REMOTE BLOOD OXYGEN SATURATION AND TEMPERATURE MONITORING AND ANALYSIS SYSTEM OF A COVID PATIENT USING IOT**

Submitted to the Savitribai Phule Pune University in the partial fulfillment of the requirements for the award of degree

of

**BACHELOR OF COMPUTER ENGINEERING**

**BY**

**APOORV BEDMUTHA Roll no. 405A013 Exam seat no.**

**IRESH SAHANI Roll no. 405A064 Exam seat no. B150234360**

**RASIKA KOTKAR Roll no. 405B034 Exam seat no.**

**SHIVAM JHA Roll no. 405A061 Exam seat no.**

**PROJECT GROUP NO: 6**

## DEPARTMENT OF COMPUTER ENGINEERING

Accredited by NACC

## STES’

## SINHGAD COLLEGE OF ENGINEERING

**S.N. 44/1, Vadgaon Bk, Off Sinhgad Road**

**Pune – 411041**

**YEAR 2020-21**



## SINHGAD COLLEGE OF ENGINEERING

**S.N. 44/1, Vadgaon Bk, Off Sinhgad Road**

**Pune – 411041**

**CERTIFICATE**

This is to certify that the preliminary project report entitled

**“SAFE - OXI: A REMOTE BLOOD OXYGEN SATURATION AND TEMPERATURE MONITORING AND ANALYSIS SYSTEM OF A COVID PATIENT USING IOT”**

Submitted by

Apoorv Bedmutha Exam Seat No. B150

Iresh Sahani Exam Seat No. B150234360

Rasika Kotkar Exam Seat No.

Shivam Jha Exam Seat No.

is a bonafide work carried out and is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the

Degree of Bachelor of Computer Engineering

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

Place : Pune

Date :

Prof. Shweta Kambre Prof. M. P. Wankhade

Internal Guide Head

Department of Computer Engineering Department of Computer Engineering

Dr. S.D. Lokhande

Principal

SCOE, Pune

**ACKNOWLEDGEMENT**

It gives us great pleasure to present a mini project on **“SAFE-OXI: A REMOTE BLOOD OXYGEN SATURATION AND TEMPERATURE MONITORING AND ANALYSIS SYSTEM OF A COVID PATIENT USING IOT”** in partial fulfillment of requirement of Bachelor In **Computer Engineering**.

We take the opportunity to express our deep sense of gratitude and whole hearted thanks to our respected guide, **Prof Shweta Kambre**, Department of Computer Engineering. we are greatly in debt to you piloting us whenever we faced difficulties in our work.

We are also thankful to **Prof M. P. Wankhede** Head Of The Department of computer Engineering. **Principal Dr. S. D. Lokhande** for his overwhelming support in valuable guidance we are also thankful to our respected teachers and our colleagues for their support.

Apoorv Bedmutha

Iresh Sahani

Rasika Kotkar

Shivam Jha

**ABSTRACT**

The pandemic situation has been tough for everyone. Thousands of lives were lost. A major part of the affected and the casualties were the medical personnel’s and doctors who were responsible for treating Covid - 19 patients. In such times, the doctors and the medical staff are the most valuable asset for the country and its people. Hence, we provide a solution that may decrease the mortality of such saviors. "Safe-oxi" is a safe and contact free approach in comparison to traditional use of pulse oximeter for calculation of covid patient's blood oxygen saturation and other useful parameters with the use of IOT and Machine learning. Safe - oxi will also result in the overall efficiency of covid ward as the system will make it possible to monitor multiple patients at the same time.

Blood oxygen saturation is an important physiological parameter involved in respiration and circulation, which is also a critical indicator in the area of medical and health monitoring. Here we present a safe and contact free approach in comparison to traditional use of pulse oximeter for calculation of covid patient’s oxygen saturation and temperature. Safe-Oxi will bring the world of Internet Of Things and machine learning to the covid ward.

**LIST OF FIGURES**

2. 2 Agile Development Model..................................................................................................16

2.4.1 Time-Line Chart................................................................................................................19

3.1 Idea Matrix...........................................................................................................................21

3.4 Use Case Diagram................................................................................................................25

3.5 Activity Diagram..................................................................................................................26

3.6 Class Diagram......................................................................................................................27

3.7 Sequence Diagram...............................................................................................................28

3.8 State Transition Diagram.....................................................................................................29

3.9 Deployment Diagram...........................................................................................................30

**LIST OF TABLES**

1.2 Literature Survey ……………………………………………………………………………10

2.3.1 Cost Estimate Table ….........................................................................................................17

4.2 UI Testing ...............................................................................................................................31

4.3 Unit Testing ............................................................................................................................32

4.4 Integration Testing .................................................................................................................33

4.5 Acceptance Testing …............................................................................................................33

**ABBREVIATIONS**

IOT Internet Of Things

WHO World Health Organization

GUI Graphical User Interface

IMA Indian Medical Association

IDE Integrated Development Environment

SpO2 Oxygen Saturation

PPE Personal Protective Equipment

BPM Beats Per Minute

API Application Programming Interface

**HIGH LEVEL ORGANIZATION OF PROJECT REPORT**

1. **Introduction:** Background and problem definition of the undertaken project.
2. **Project Planning and Management:** Creation of SRS, choosing appropriate SDLC Model and creating a backlog for execution.
3. **Analysis and Design:** Design of Architecture at all levels and submodules with their integration. Creation of UML Diagrams.
4. **Testing:** Creation of test cases with their input and expected output, designing the test scenarios and stubs for testing.
5. **References:** Attached herewith the referred papers (IEEE or International conference) for literature survey and getting domain knowledge of the project space.

**CONTENTS**

**TITLE**  **PAGE NO**

Certificate 2

Acknowledgement 3

Abstract 4

List of Figures 5

List of Tables 6

Abbreviations 7

High Level Organization of project report 7

**Chapter No.**  **Title**  **Page No.**

**1 INTRODUCTION**

1.1. Background and Basics....................................................9

1.2. Literature Survey..............................................................10

1.3. Project Undertaken...........................................................10

1.3.1 Problem Definition.............................................11

1.3.2 Scope Statement.................................................11

1.4. Organization of the Project Report...................................11

**2 PROJECT PLANNING AND MANAGEMENT**

2.1. Detail System Requirement Specification (SRS).............12

2.1.1: System Overview...............................................12

2.1.2: Functional Requirements...................................12

2.1.3: Non- Functional Requirements..........................13

2.1.4: Deployment Environment..................................14

2.1.5: External Interface Requirement.........................15

2.2. Project Process Modeling..................................................16

2.3. Cost & Efforts Estimates...................................................17

2.4. Project Scheduling.............................................................19

2.4.1 Time Line Chart...................................................19

**3 ANALYSIS & DESIGN**

3.1. IDEA matrix.......................................................................21

3.2. Mathematical Model..........................................................22

3.3. Feasibility Analysis............................................................24

3.4. Use-Case Diagram..............................................................25

3.5. Activity Diagram................................................................26

3.6. Class Diagrams...................................................................27

3.7. Sequence Diagrams............................................................28

3.8. State Transition Diagrams..................................................29

3.9. Deployment Diagrams........................................................30

**4**  **TESTING**

4.1. UI Testing...........................................................................31

4.2. Unit Testing........................................................................32

4.3. Integration Testing..............................................................33

4.4. Acceptance Testing.............................................................33

**REFERENCES**.....................................................................................34

**CHAPTER 1**

**INTRODUCTION**

**1.1 BACKGROUND AND BASICS**

Mortality rate of doctors has been a rising concern since the beginning of Pandemic. COVID-19 has caused a huge burden and loss to the world with doctors bearing the brunt of physical burnout, mental stress, occupational risk of infection with increased risk of morbidity and mortality, being the front-line workers. Especially in India, the mortality of these doctors has made a dent in an already compromised health care system due to poor doctor patient ratio. The Indian Medical Association (IMA) National COVID-19 registry data suggests more than 1000 doctors have been infected with Covid-19 virus, where 75% of them are above the age of 50 years. Concerns have been raised since nearly 200 doctors have succumbed to COVID-19 so far with a significant number of healthcare professionals affected as well. IMA has issued a ‘Red Alert’ and requested the health authorities to ensure adequate safety of all doctors along with support from state sponsored medical and life insurance facilities to all involved in the coronavirus containment efforts. Doctors account for 0.5% of the total deaths in India due to Covid-19.

But that's not it, since a doctor is different from any other patient. A doctor or any medical staff is a valuable resource since absence of each medical authority will affect the treatment of patients vastly. Due to poor doctor to patient ratio, the hospitals have already been overflowing with patients. Also, the shortage of PPE suits have also culminated into spreading covid among the medical staff responsible for the patients.

According to WHO, The virus that causes COVID-19 spreads primarily through droplets generated when an infected person coughs, sneezes or speaks. One can also be infected by touching a contaminated surface and then touching your eyes, nose or mouth before washing your hands. Hence, it was really necessary to minimize the contact between a covid affected and the doctor also to provide a system that will allow doctors to treat multiple patients at the same time.

**1.2 LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr no.** | **Title** | **Year** | **Author And Publisher** | **Summary** |
| 1. | Development of Wearable Pulse Oximeter Based on Internet of Things and Signal Processing Technique | 2017 | Yuyang Xie, Yongjie Gao, Yuchun Li, Yu Lu  IEEE | Wearable non-invasive blood oxygen saturation monitoring approaches overcomes the shortcomings of the traditional approaches, and reflects the tendency to change of breathing function in real time, which has great advantages. Thanks to the IOT and advanced signal processing algorithms, we have  developed wearable pulse oximeters. These devices are able to detect data such as *SpO*2, *PR*, and *PI*. These data can then be transmitted to an APP. Then, the data could be uploaded to a server to generate a report. |
| 2. | A patch-type wireless forehead pulse oximeter for SpO2 measurement | 2017 | A. Azhari, S. Yoshimoto, T. Nezu, H. Iida, H. Ota, Y. Noda, T. Araki, T. Umemura and T. Sekitani  IEEE | A patch-type wireless wearable pulse oximeter system has been developed to measure the heart beat rate and oxygen saturation of blood in reflective mode from a person’s forehead. The system uses light sources of two wavelengths in an optical sensor,  separates photodetector signal and transimpedance amplifier  output into (IR) and (PPG)  signals, and then digitizes and transmits data wirelessly via a  Bluetooth module to a remote PC in real time, where the SpO2  value is calculated. |
| 3. | COVID-19 and mortality in doctor - how covid has affected the people working in medical sector. | 2020 | Karthikeyan Iyengar, Pranav Ish, Gaurav Kumar Upadhyaya, Nipun Malhotra, Raju Vaishya, Vijay Jain  Springer | COVID-19 has been associated with an increased mortality in doctors and health care workers. Until an effective cure/vaccine is developed, risk assessments at work, mitigating confounding  factors, adequate supply of personal protective equipment (PPE) and enhanced protection against infection are necessary to protect health care professionals on the coronavirus frontline. Otherwise, this occupational risk can lead to further untimely mortality and become another unintended consequence of  the COVID-19 pandemic. |

**T1.2 Literature survey table**

**1.3 PROJECT UNDERTAKEN**

**1.3.1 PROJECT DEFINITION**

To provide a complete remote monitoring system for a covid patient that will significantly decrease the mortality and chances of contracting covid-19 virus amongst the medical authorities, the system should also solve the problem of low patient to doctor ratio by enabling doctors to monitor multiple patients at the same time. The system will also solve the problem of using different devices for body temperature and blood oxygen saturation and decrease overall chances of contact among the actors.

**1.3.2 SCOPE STATEMENT**

This project involves building of a web based as well as app-based system that once powered by IOT hardware can be used to get real time medical statistics of a covid patient, with good support of machine learning algorithms the system will be able to analyze each patient’s condition and further help doctor in understanding the patient. The system will eliminate the use of pulse oximeter and thermometer separately by providing a device that may sense both. Further a real time alert system will also be introduced so that in critical situations doctors and responsible staff will be notified. Using this infrastructure we hope to minimize the chances of spread of contagious diseases among the medical personnel.

**1.4 ORGANISATION OF THE PROJECT REPORT**

The overall report revolves around the objective of achieving remote covid patient monitoring and analysis system.

First chapter deals with introduction of pandemic situation, Problem definition and possible solution with use of IOT and Cloud support as well as machine learning. Literature survey and project undertaken is also described.

Second chapter comprises of project planning and management. In that we include system Requirements Specification (SRS), Project Process Modeling, Cost and Effort estimates and finally project scheduling.

Third chapter deals with analysis and design. In this we include Idea matrix and UML Diagrams.

Fourth chapter deals with the testing modules implemented on the project. In this we include unit testing, Integration testing, Acceptance testing and GUI testing.

The above defines the flow this report.

**Chapter 2**

**Project Planning and Management**

**2.1 Detailed System Requirements Specification (SRS)**

**2.1.1 SYSTEM OVERVIEW**

"Safe Oxi" integrates Internet Of Things with Disease Monitoring by creating a single consistent platform which caters to different medical needs for a hospital. The following are the main features of SAFE-OXI:

1. Cross Platform Support: Provides App support for Android, IOS as well as browser.

2. Real Time Data: Provides real time data with just 1 click.

3. Emergency Notification system: System that notifies the medical staff when the patient is moving in critical situation.

4. Single device, Multiple data: The same device monitors multiple measures such as BPM, Temperature and Oxygen Saturation.

5. Data Analytics and Prediction support: Complete analysis and statistics provided within the application.

6. Live Covid 19 stats: Doctors can get Live covid stats by use of API's.

7. Downloadable files: The data and Stats can be downloaded in PDF format.

* + 1. **Functional Requirements**

1. The data should be updated after every 1 minute.

2. Emergency notification must be sent as soon as data turns out to be critical

3. The data and statistics generated must be downloadable in pdf format

4. The data will be analyzed and be run through data mining algorithms to make predictions and statistics.

5. Hardware specifications: Node MCU(ESP8266) Wi-Fi module, MAX 30100 sensor and jumper cables.

6. The System developed must be cross platform and should be accessible from android, iOS as well as browser.

**2.1.3 Non-Functional Requirements**

**1. Performance Requirements**

When we scale our primary requirement will be near real-time performance of application. We will be fetching data and inferring from it which is critical and data changes every minute and it should impact our predictions up to a certain extent.

**2. Security Requirements**

Data Management and privacy protection will be very critical because of the use of patient's personal medical records. We will be using a login system to prioritize access and might be implementing encryption while transmission of data.

**3. Software Quality Attributes**

1. Continues Contact of device and patient:

2. The device needs to be connected to the patient at all times, since the system can be used efficiently if data is continuously updated on cloud

**4. Basic Internet Connection**

The app and device are both using cloud services and hence need a constant internet connection.

**2.1.4 Deployment Environment**

**1. Hardware Requirement**

1. Stable Internet Connection

2. Node MCU micro-controller (ESP8266)

3. MAX30105 pulse oxi sensor

4. Jumper cables

5. APK or browser for accessing output

**2. Software Requirement**

1. Arduino IDE for writing and uploading code to ESP8266.

2. Wire.h library for connection and port recognition between ESP8266 and sensors.

3. MAX30105.h library for using and controlling pulse oxi sensor.

4. heartRate.h library for calculating heart rate from the sensor readings.

5. ThingSpeak.h library for connecting the micro controller to Thingspeak cloud.

6. ESP8266WiFi.h library for using the Wi-Fi module of the micro controller.

7. Axios JavaScript Library for fetching data and making a get request on thingspeak cloud.

8. FireBase Web Api for hosting and Management of Safe-Oxi Website.

9. GoNative.io for Cross Platform app generation.

**3. External Interface Requirements**

**3.1 User Interfaces**

1. Login Page to only allow authorized users to access the platform.

2. Dashboard for actual representation of data and statistics.

3. Profile page for owner's / hospital's information

**3.2 Communication Interfaces**

1. Web Browser with stable internet connection. eg: Mozilla Firefox, Google Chrome, Opera Mini, etc.

2. Stable network with a minimum internet speed of 400-500 kbps for uploading as well as downloading data to and fro from cloud.

**4. Other requirement**

**4.1 Completeness**

All external libraries including their respective license will be documented.

**4.2 Usability**

1. Web applications have grown to expect easy way of presenting data, graphs and statistics.

2. Given Help annotations for new users for better understanding of features and its capabilities.

**4.3 Maintainability**

Application code will be cohesive and have easily recognizable functionality. Classes will be abstract enough to facilitate data structure changes. Class and function modularity will be implemented to avoid the need for major refactoring.

**2.2 Project Process Modeling**

Here we have used agile process modeling technique. This allows versatility in our project. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The Agile model was primarily designed to help a project to adapt to change requests quickly. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task agility is required. Agility is achieved by fitting the process to the project, removing activities that may not be essential for a specific project.



**Figure** **Fig 2.2 Agile Process Model**

In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed within a couple of weeks only. At a time one iteration is planned, developed and deployed to the customers. Long-term plans are not made.

**2.3 Cost and Efforts Estimates**

Cost of the project will be the cost of hardware plus the cost of the work that is put in. Basic COCOMO:

* **Project class:**

We have determined our project is the characteristics of semi-detached mode as project is college level and requirements are rigid and less than rigid.

* **Number Of Code Lines:**

We estimate our project will have 10000 delivered source instructions

So the basic COCOMO model equations are as follows:

Effort Applied (E) = a*b* (*KLOC*) b*b* [man months]

Development time (D) = c*b* (*E*)d*b* [months]

People Required (P) = *E/D* [count]

Where:

KLOC (kilo lines of code) is the estimated number of delivered lines (in thousands) of code for project.

E is the effort applied per person per month.

D is the development time in consecutive months.

The coefficient a*b*,b*b*,c*b*,d*b*  are predetermined according to project class given in the following table:

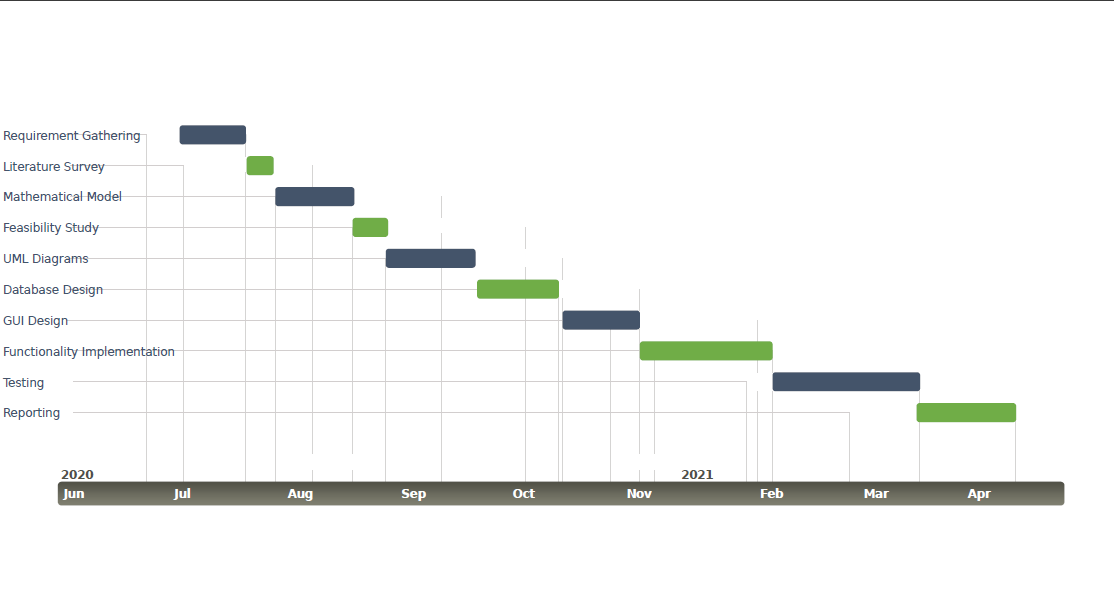
Calculations: So, this project comes under the semi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software Project Class** | a*b* | b*b* | c*b* | d*b* |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

**T2.3 Cost Estimated Table**

**2.4 Project Scheduling**

**2.4.1 Time Line Chart**

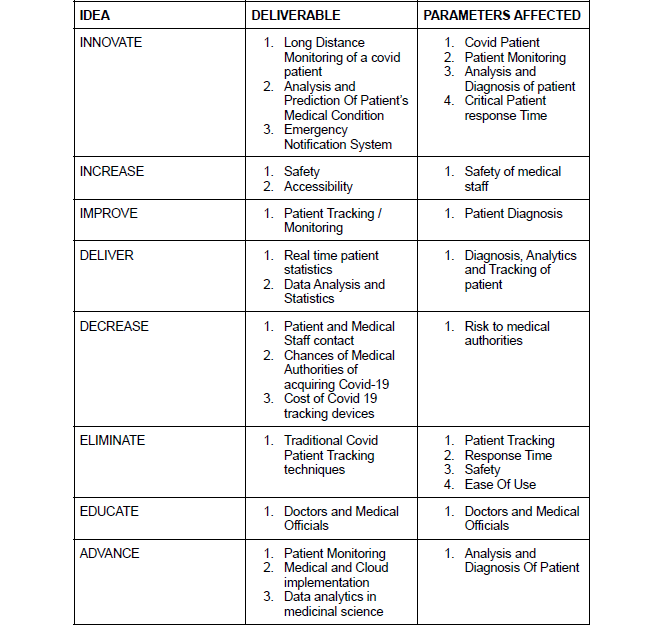
**Figure 2.4.1: Time-Line Chart**

**Chapter 3**

**Analysis and Design**

**3.1 Idea Matrix**

An IDEA matrix is a concept that evaluates various effects that the idea has. This tells us almost everything about the project.



**Figure 3.1: Idea Matrix**

**3.2 Mathematical Model**

In this section we will construct a mathematical model that describes the functions of each of our modules in terms of set relations, input output tuples and domain to which they belong. We will discuss these domains first and then after construct their mapping.

**3.2.1 Input Domain**

We have an Audio Video input so input varies with time. Consider that is input stream, of two time-varying tensors,

Where, is the audio stream,

and is the r-g-b video stream.

Where are bytes.

**3.2.2 Output Domain**

The output is also a stream of tuples which belong to following output domain,

The Output is composed of following entries:

1. t : time :time at which a reading was taken w.r.t video.
2. df : Boolean : 1 if deepfake frame else 0.
3. gs : float : genuineness score between 0 to 1.
4. pe : emotion ID : prominent expression found at time t.

In set theory,

Let op be output tuple and O is output domain,

Hence op is 4-tuple as follows,

**3.2.3 Functions or Modules**

Our project or application can be viewed as a function F that takes in input ip and gives output op.

Where mapping

3.2.3.1: Module 1: DeepFake Detection

DeepFake detection can be defined as a function that takes a video frame at time t and gives binary result 1 if deepfake else 0.

Where mapping

Furthermore, DF is composed of 2 functions:

1. : Face detection, grayscale conversion and Cropping
2. : DeepFake Detection

3.2.3.2: Module 2: Face Expression Detection:

Face Expression detection can be defined as a function that takes a video frame at time t and gives one of the expression IDs in E.

Where mapping

Furthermore, FE is composed of 2 functions:

1. : Face detection, grayscale conversion and Cropping
2. : Face Expression Detection

3.2.3.3: Module 3: Sound Expression Detection:

Sound Expression detection can be defined as a function that takes an audio frame at time t and gives one of the expression IDs in E.

Where mapping

Furthermore, SE is composed of 2 functions:

1. : Mel-Frequency Cepstral Coefficients features extraction where F is features set.
2. : Sound Expression Detection.

3.2.3.4: Module 4: Aggregator:

Aggregator is a function that takes in deepfake reading df, face expression ID ‘fe’, and sound expression ID ‘se’. Let’s say that we define AGG as aggregator function then,

Its mapping is

Aggregator has a function to calculate the genuineness score, GS which takes two arguments,

1. fea : array : array of face expressions for t = t - 5 to t = t
2. se : E : sound expression generated at t = t recorded every 5 sec.

Genuineness score can be defined as:

Further the Aggregator function can be defined as,

Here prominent expression .

Hence the final project function F can be defined as,

Which completes our mathematical model.

**3.3 Feasibility Analysis: Algorithmic Feasibility**

This subsection discusses on how the algorithms or structures used in our modules are feasible and can be implemented mathematically. Here we will analyze the computational modules we will be using in our implementation w.r.t. their space and time complexity whether it is P or NP complete.

For this we need to first understand the fundamental unit of computation we will be using in all of our modules involved in audio-video processing, the Convolutional Neural Network which is in turn based on Neural Networks. So, any inferences we make on neural networks will implicitly apply on Convolutional Neural networks and in turn on the higher order models used. And thus, on the architecture we will be using.

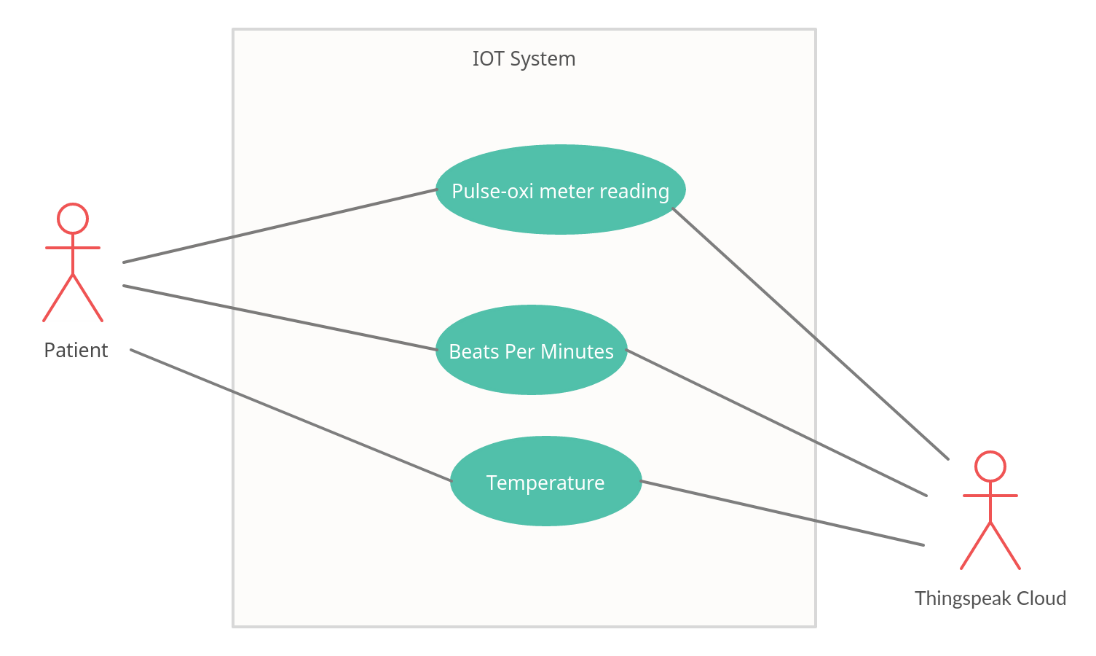
The Neural Network has two main functionalities, forward propagation and backward propagation. Every operation in neural network is a matrix multiplication, addition or function mapping out of which addition and mapping of m by n matrix has time complexity and multiplication of m by n and n by p matrix has when applied one element at a time also these operations will produce a matrix which has finite space complexity i.e. . As Neural network is composed of matrices and its operations, a Neural network is P complete and hence can also be effectively further reduced in terms of time by using parallel computation.

We are using pytorch library to achieve this which has functionalities of creating matrices and tensors which are parallelly computed in dedicated Graphic Processing Units (GPUs). These processors compute all rows and all columns at constant time so any operation for matrix which was taking time will take time with same space complexity. So, addition and mapping will take time and multiplication will take time, which is also of polynomial order. It turns out that Neural Networks, hence convolutional Neural Networks and thus our models are P-complete.

From above statements we conclude that our architecture is feasible with respect to space and time mathematically.

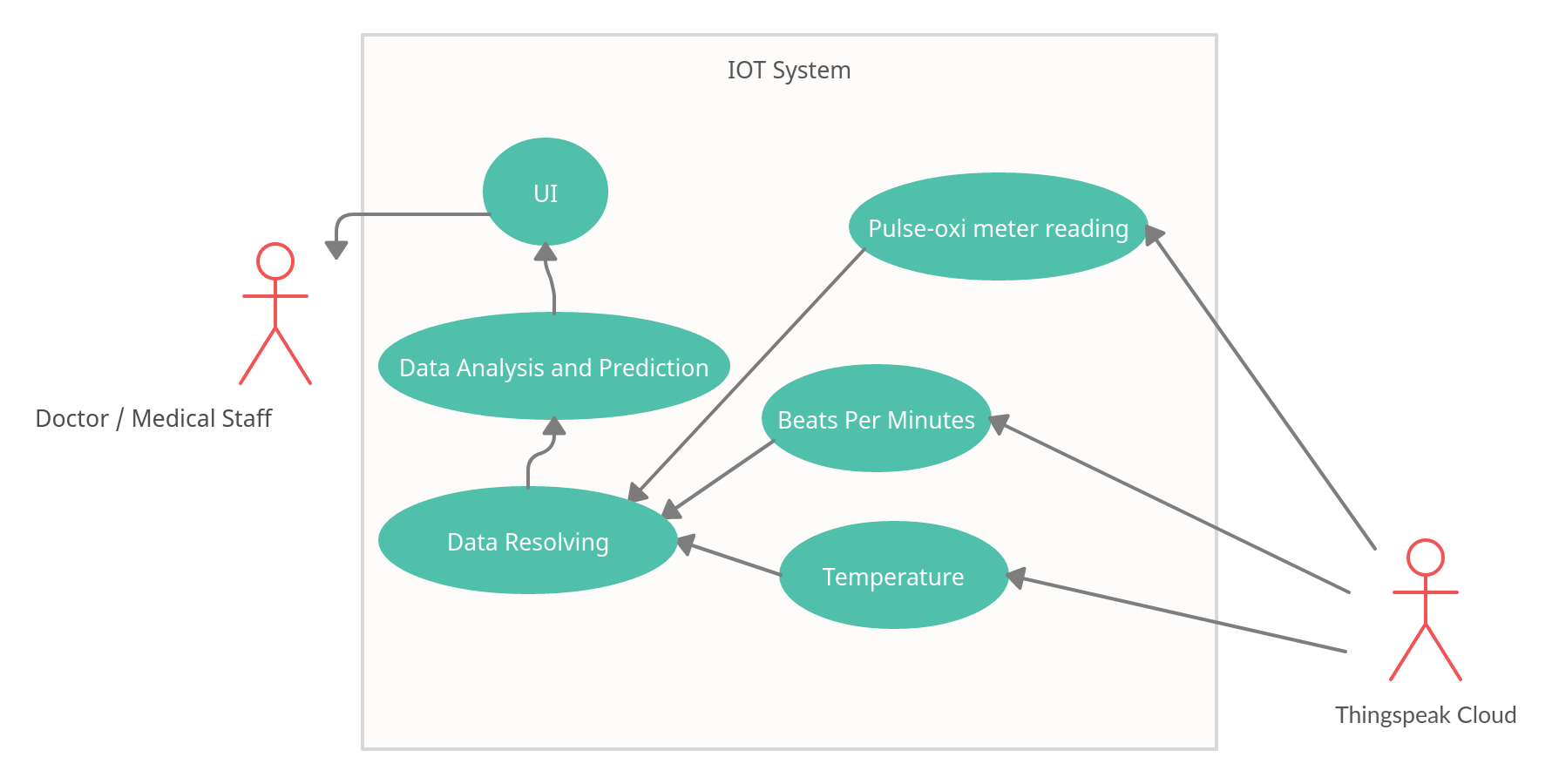
**3.4 Use-Case Diagrams**

**3.4.1 Use-Case diagram 1**



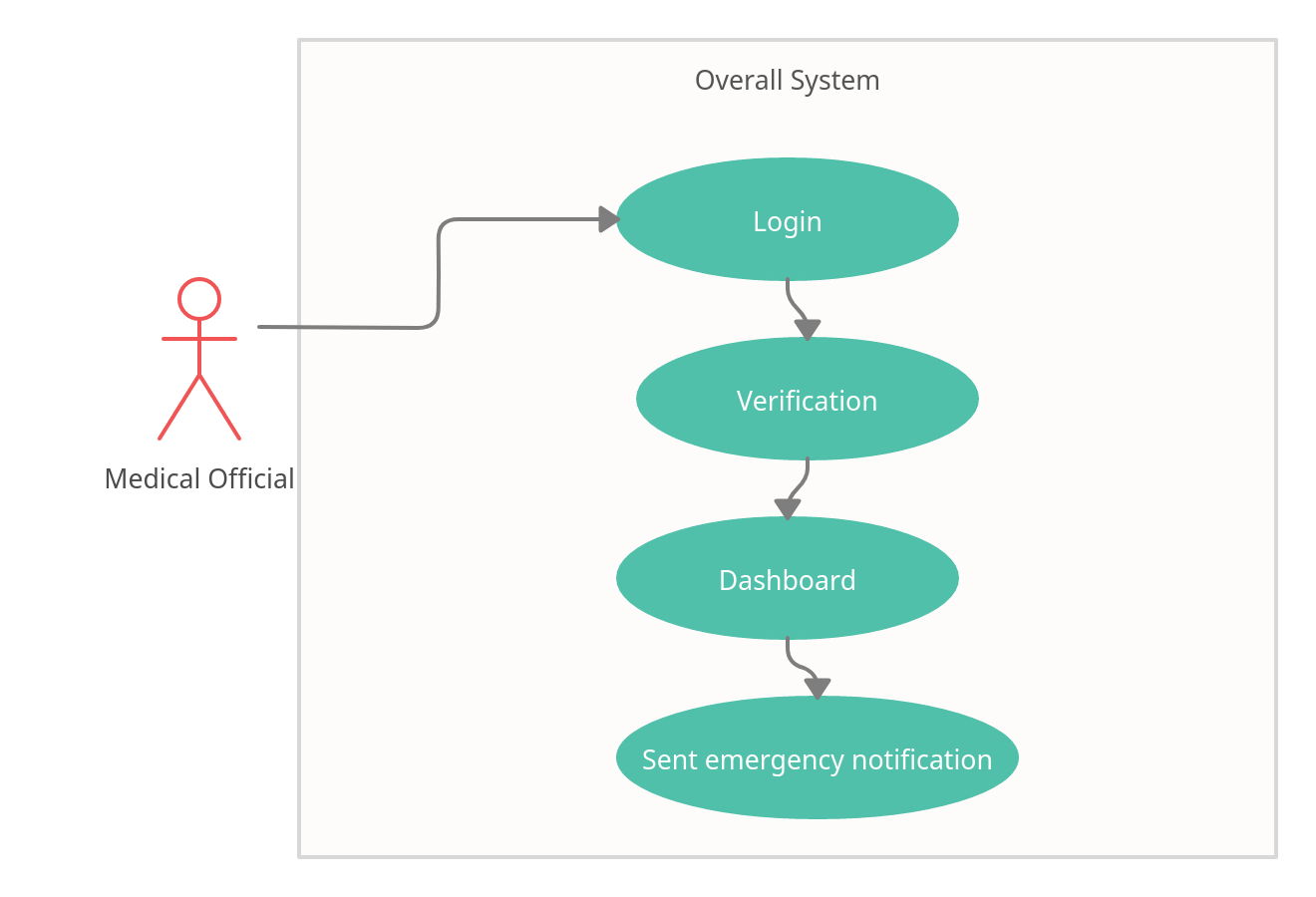
**Figure 3.4.1 : Use Case Diagram 1**

**3.4.2 Use-Case Diagram 2**

****

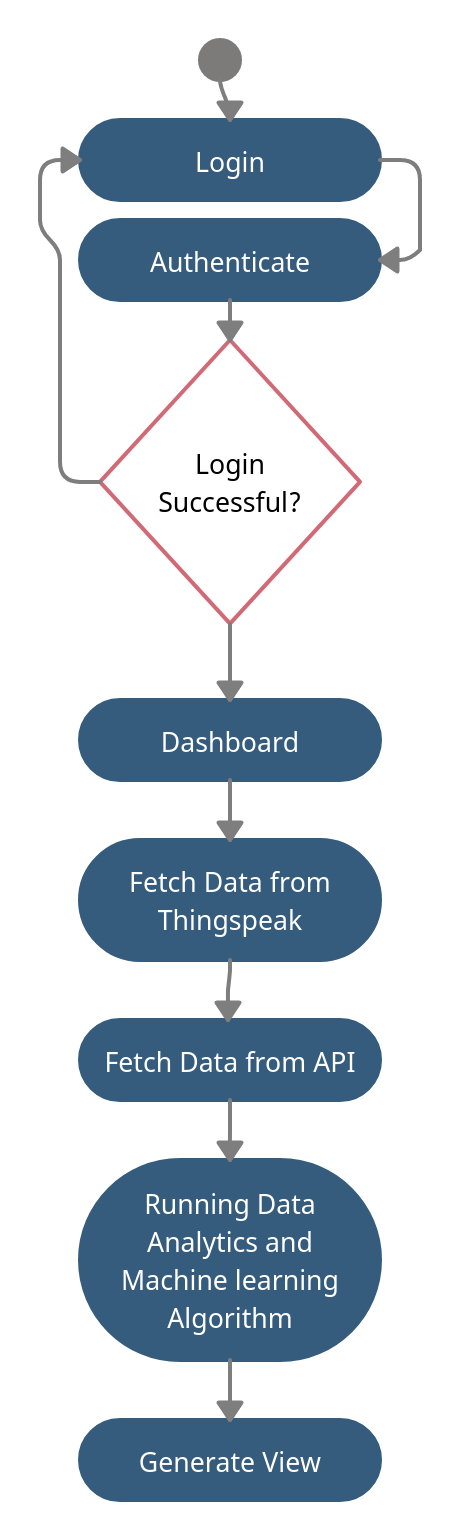
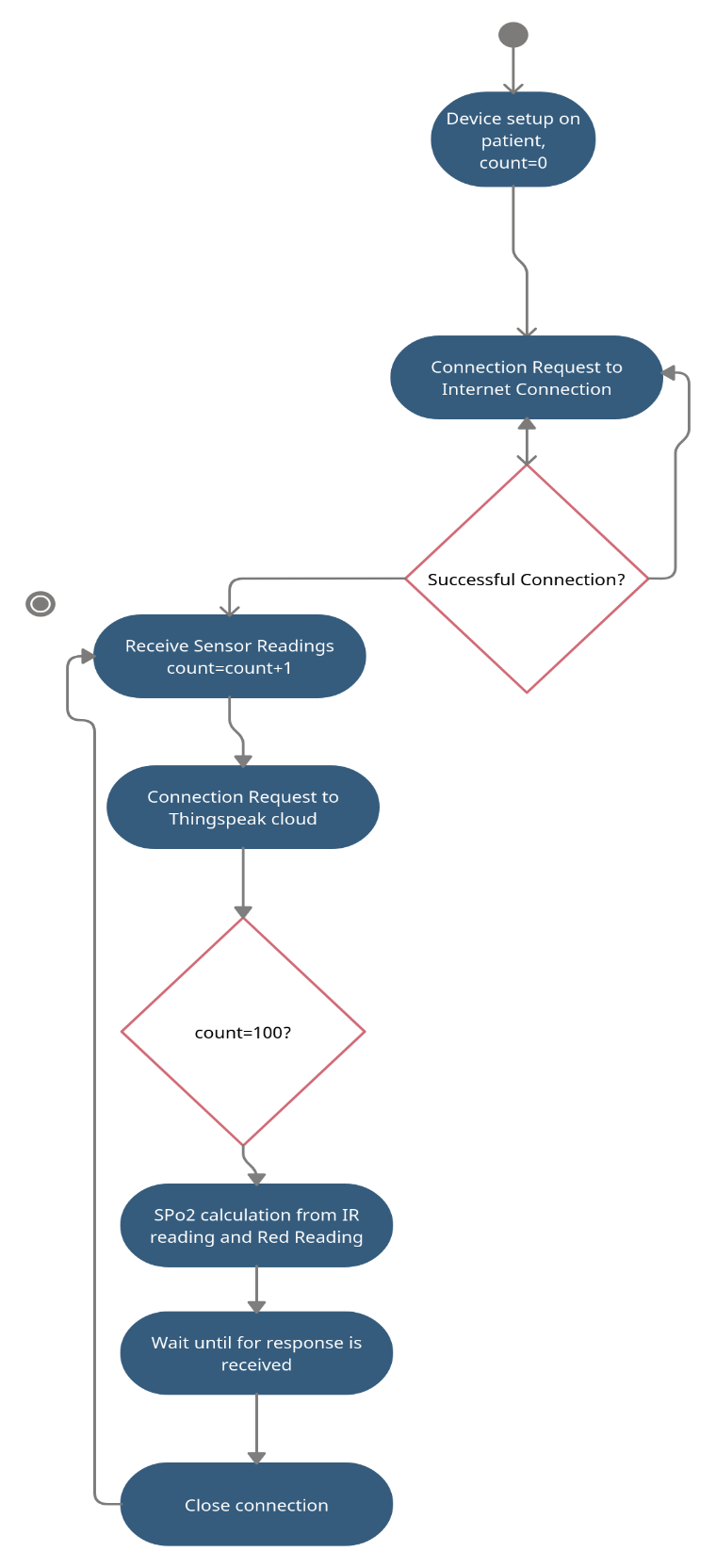
**Figure 3.4.2 Use Case Diagram**

**3.4.3 Use-Case Diagram 3**



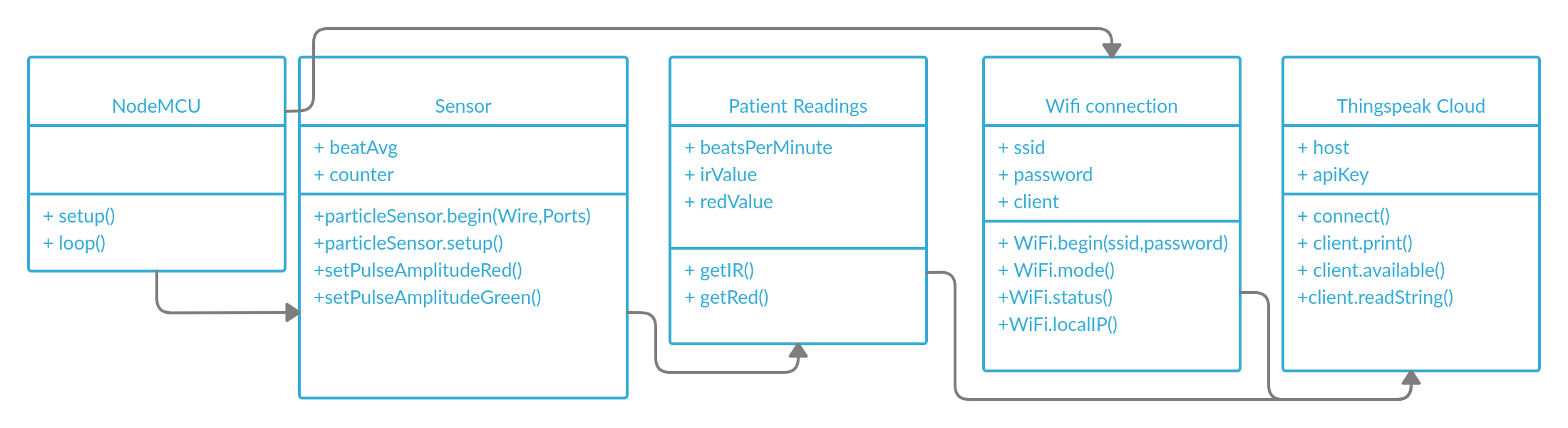
**Figure 3.4.3 Use Case Diagram 3**

**3.5 Activity Diagram**



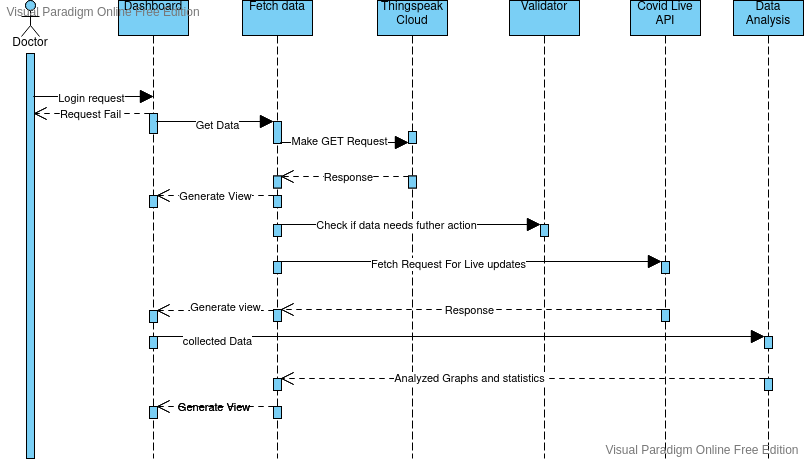
**Figure 3.5: Activity Diagram**

**3.6 Class Diagram**



**Figure 3.7: Class Diagram**

**3.7 Sequence Diagram**



**Figure 3.8: Sequence Diagram**

**3.8 State Transition Diagram**

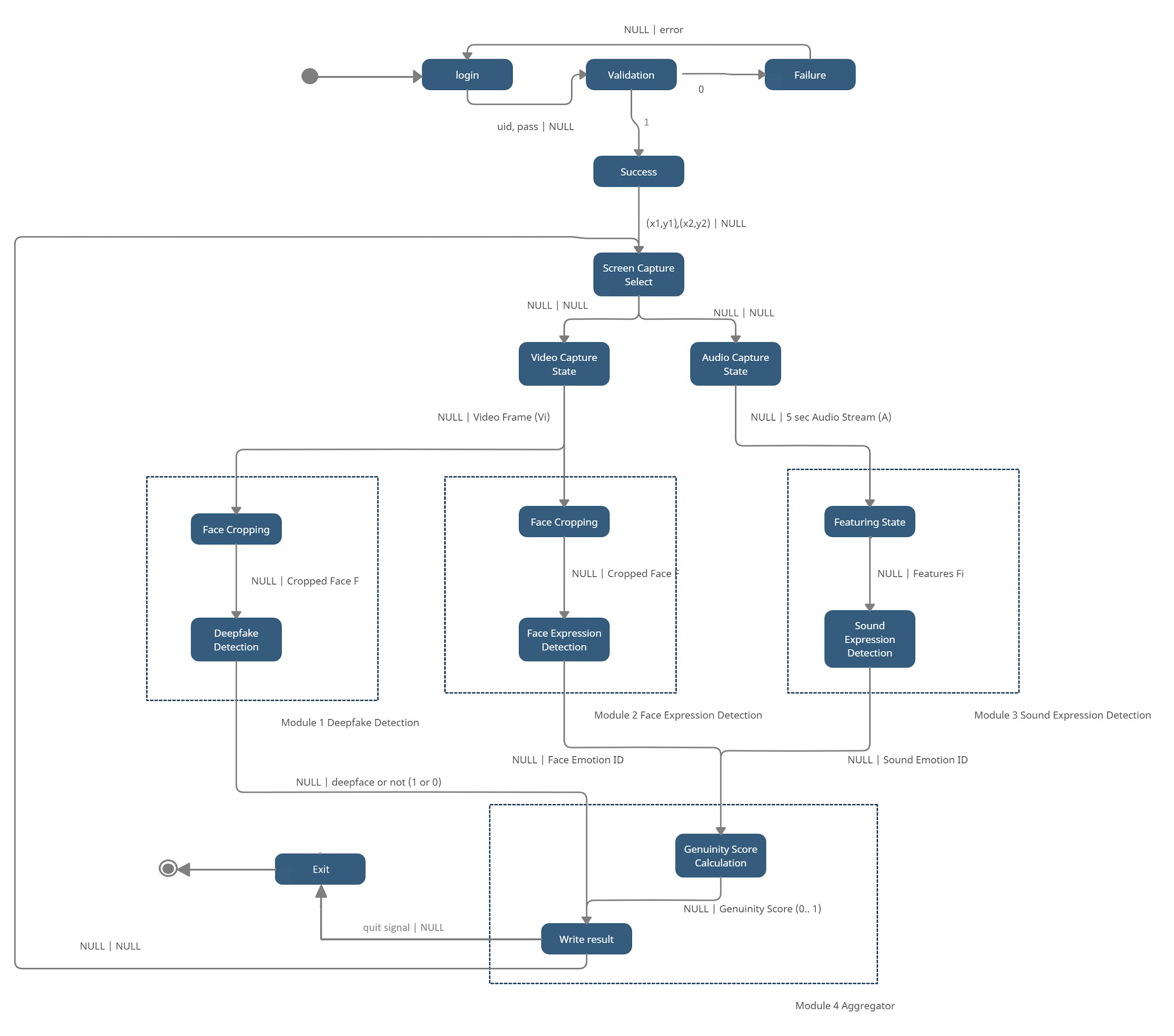


Figure 3.9 : State Transition Diagram

**3.9 Deployment Diagram**

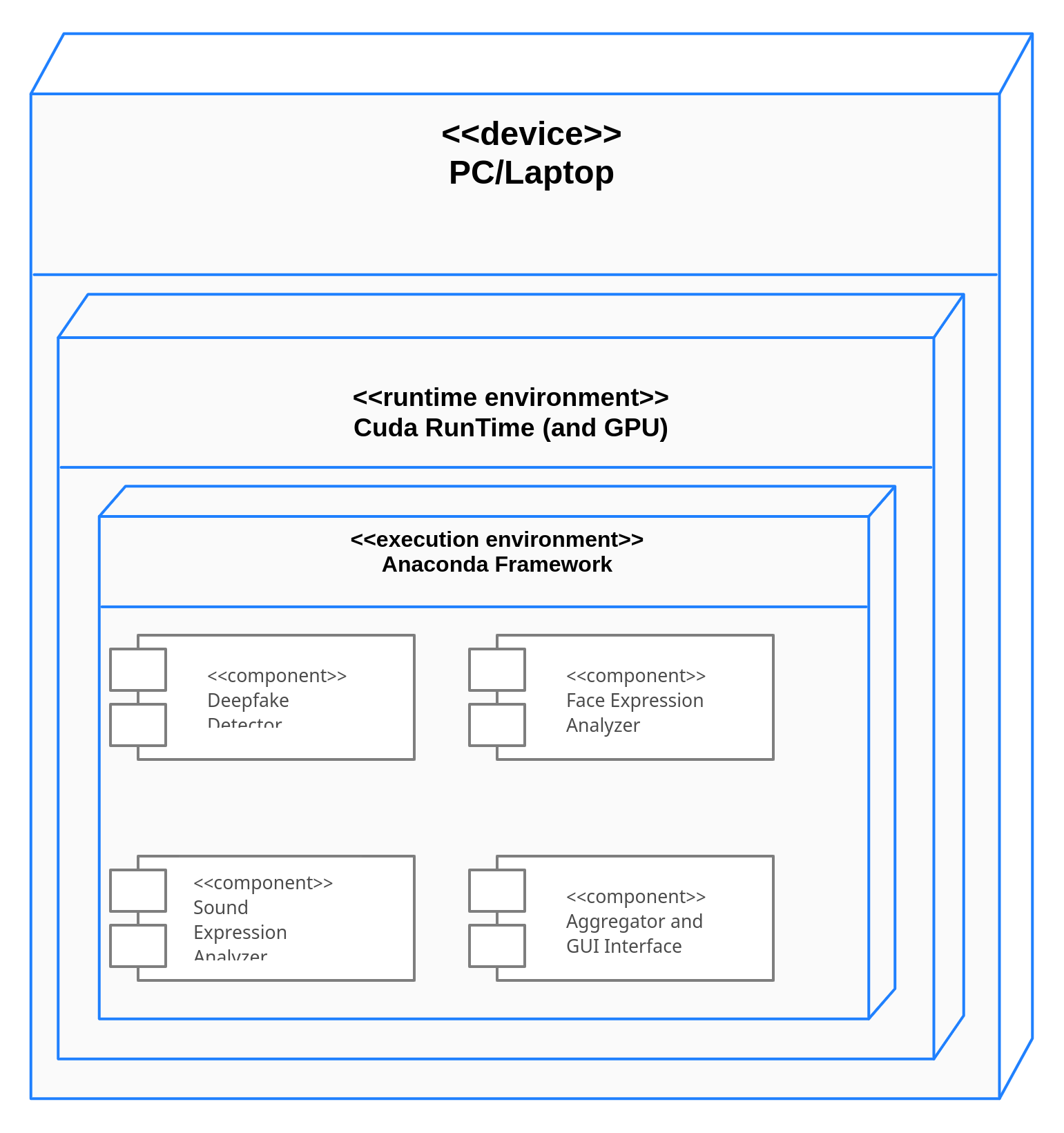


Figure 3.10 : Deployment Diagram

**Chapter 4**

**Testing**

This Chapter covers the testing approach used and the test cases .This Chapter covers the following testing approaches

1. UI Testing

2. Unit Testing

3. Acceptance Testing

4. Integration Testing

**4.1 UI Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Input | Actual Output | Expected Output |
| 1 | Login | Valid Username, Valid Password Registration | Login Successful | **-** |
| 2 | Login | Invalid Username, Valid Password | Login Failed | **-** |
| 3 | Login | Invalid Password | Login Failed | **-** |
| 4 | Registration | New Username, New Password | Registration Successful | **-** |
| 5 | Registration | Special Characters in Username | Invalid | **-** |
| 6 | Registration | Password according to Regex | Registration Successful | **-** |
| 7 | Start/Stop | On Start Press | Start Capturing after Loading | **-** |
| 8 | Start/Stop | On Stop Press | 1) Stop Capture (a+v)  2) Save Video | **-** |

Table 4.2: UI Testing

**4.2 Unit Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Input | Actual Output | Expected Output |
| 1 | Face Expression | Face with some Expression | 1) Face cropped at boundaries  2) Correct Expression predicted | **-** |
| 2 | Face Expression | Face Absent | No Cropped Face should be present | **-** |
| 3 | Sound Expression | Input Mic Capture of 5 sec | Correct Sound Expression predicted | **-** |
| 4 | Sound Expression | No Mic Input | Expression can be neutral but its Non-Deterministic | **-** |
| 5 | Deepfake Detection | Video captured with Actual Webcam | Fake % < 50% | **-** |
| 6 | Deepfake Detection | Video created by Avatarify\* | Fake % > 50% | **-** |
| 7 | Aggregator | Deepfake ID is Non Zero in shm | 1) If 0 Real Count increased else if 1 Fake Count increased.  2) shm is again set to -1. | **-** |
| 8 | Aggregator | Sound Expression ID is Non Zero in shm | 1) Corresponding Sound Expression string extracted and drawn on face rectangle if exists.  2) At this moment the result of that sound clip along with aggregated genuiness score and deepfake reading dumped in SRT file writer. | **-** |
| 9 | Aggregator | Face Expression ID is Non Zero in shm | 1) corresponding face id extracted and drawn on face rect.  2) shm set to -1 again | **-** |

Table 4.3: Unit Testing

**4.3 Integration Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Input | Actual Output | Expected Output |
| 1 | Deepfake + Face | Every some ms we get Non-Zero values in shm | freq(non zero shm f + df) > freq(non zero shm s) | **-** |
| 2 | Sound | Every some 5 sec we get Non-Zero value in shm | delay(s1, s2) ~< 5 sec | **-** |

Table 4.4: Integration Testing

**4.4 Acceptance Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Input | Actual Output | Expected Output |
| 1 | A-V input | 1) Screen capture the Video.  2) Record PC / Mic Audio. | 1) Output Video file with Audio Recording.  2) Subtitles file displaying the forgery by deepfake %. Behavioral profile by correlation and genuineness score. | **-** |

Table 4.5: Acceptance Testing

**References**

1. FaceForensics++: Learning to detect Manipulated Facial Images, Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Nießner , New York, United States, 2019, arXiv.org

2. Human Emotion Recognition using Convolutional Neural Network in Real Time, Rohit Pathar; Abhishek Adivarekar;Arti Mishra; Anushree Deshmukh, Chennai, India, 2019, ICIICT

3. Emotions Understanding Model from Spoken Language using Deep Neural Networks and Mel-Frequency Cepstral Coefficients, Marco Giuseppe de Pinto; Marco Polignano; Pasquale Lops; Giovanni Semeraro, Bari, Italy, 2020, EAIS

4. Xcception: Deep Learning with Depthwise Seperable Convolutions, François Chollet, New York, United States, 2017, arXiv.org

5. Study on CNN in the Recognition of Emotion in Audio and Images, Bin Zhang; Changqin Quan; Fuji Ren, Okayama, Japan, 2016, ICIS