Smart Hospital Bed

BS COMPUTER SCIENCE

Muhammad Hamza	BSCS17004	BSCS17
Ammar Bin Asim	BSCS17005	BSCS17
Aoun Ahmed	BSCS17008	BSCS17
Muhammad Junaid Ahmad	BSCS17012	BSCS17
Warda Ahmer	BSCS17027	BSCS17
Meram Zia	BSCS17038	BSCS17

DEPARTMENT OF COMPUTER SCIENCE

INFORMATION TECHNOLOGY UNIVERSITY

LAHORE, PAKISTAN

Smart Hospital Bed

A final year project submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

Muhammad Hamza	BSCS17004	BSCS17
Ammar Bin Asim	BSCS17005	BSCS17
Aoun Ahmed	BSCS17008	BSCS17
Muhammad Junaid Ahmad	BSCS17012	BSCS17
Warda Ahmer	BSCS17027	BSCS17
Meram Zia	BSCS17038	BSCS17

Supervisor: Mr. Belal Muhammad Hashmi

Co-Supervisor: Dr. Adnan Noor Mian

Final Year Project Committee

Mr. Belal Muhammad Hashmi

Dr. Adnan Noor Mian

Mr. Aftab Alam

DECLARATION

This project is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. I also declare that this work is the result of my own investigations, except where identified by references and free from plagiarism of the work of others.

Signature: ………….
Muhammad Hamza

Signature: ………….
Ammar Bin Asim

Signature: ………….
Aoun Ahmed

Signature: ………….
Muhammad Junaid Ahmed

Signature: ………….
Warda Ahmer

Signature: ………….
Meram Zia

Date: 19/07/2021

The undersigned hereby certify that they have read and recommend this project entitled "Smart Hospital Bed" by …………………………………………….. for the degree of Bachelor of Science in Computer Science.

____

Mr. Belal Muhammad Hashmi

____

Dr. Adnan Noor Mian

____

Mr. Belal Muhammad Hashmi, Final Year Project Committee Member

____

Dr. Adnan Noor Mian, Final Year Project Committee Member

____

Mr. Aftab Alam, Final Year Project Committee Member

____

Dr. Saeed ul Hassan, Head of the Department

ACKNOWLEDGEMENT

Firstly, we would like to thank Allah Almighty, without whose will, this project would not have been possible. Then we would like to express our sincere gratitude to several individuals for supporting us throughout our undergraduate study. First, we wish to express our sincere gratitude to our supervisor, Mr. Belal Muhammad Hashmi, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped us tremendously at all times in our project. His immense knowledge, profound experience and professional expertise in embedded systems as well as various other areas have enabled us to complete this project successfully. Without his support and guidance, this project would not have been possible. We could not have imagined having a better supervisor in our study. Also, we would like to pay our gratitude to our co-supervisor, Dr. Adnan Noor Mian, who supported us throughout the course of this project.

We also wish to express our sincere thanks to the Information Technology University for accepting us into the undergraduate program, and for providing us with a lab and relevant equipment required to carry out this project.

We are also grateful to the committee member, Mr. Aftab Alam, for his consistent support and assistance.

Thanks for all your encouragement!

DEDICATION

We dedicate this project to our beloved parents and teachers for their never ending moral support and prayers, which always acted as a catalyst in our academic life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Term**
I2C	Inter-Integrated Circuit
ReLU	Rectified Linear Unit
Conv2d	2D convolution
UI	User Interface
FPS	Frames Per Second
HSV	Hue, Saturation, and Value
ANC	Active Noise Cancellation
FFT	Fast Fourier Transform
MEMS	Micro Electronic Mechanical Systems
ADC	Analog-to-Digital Converter

ABSTRACT

Our smart hospital bed estimates a patient's temperature, posture, and weight without the use of any tethering or intrusive equipment. The bed aims to provide estimations that can be used to detect a patient's health's severity level as soon as he/she lies on the bed. The doctors' attention is directed towards the patient accordingly.

In the case of coma patients, the smart bed can detect whether the patient's body has been moved or turned periodically to prevent bedsores. If not, the nurses are alerted to turn the patient's body.

Likewise, the bed can also detect if a patient has not urinated in a specific time, and an alert is generated if 6 hours have surpassed. Moreover, the staff is also notified if the bag is about to become full and needs to be changed, or if it has been removed.

CHAPTER ONE

INTRODUCTION

Doctors and medical staff members are responsible for curing medical conditions, disorders, and diseases, which requires daily check-up with intensive care. Nurses have the primary duty to monitor patients' temperature, pulse, urination frequency etc. frequently to track their health and treat them accordingly.

The doctor to patient ratio in Pakistan is 1:1300, and nurse to patient ratio is 1:20. Sometimes tracking patients' health at regular intervals may go unhandled or get delayed due to uncertainty or human negligence. As a result, the patient's condition may worsen, or his/her life may be put in danger.

Seeking a solution to this problem, we have developed a smart hospital bed that allows for the patients' temperature, posture, and weight to be monitored in real-time, without any tethers or intrusive equipment. Due to this bed, doctors will have more time to attend to other patients who might be in a relatively critical condition. Moreover, it minimizes the need for nurses' continuous monitoring, especially in the case of disabled or coma patients.

CHAPTER TWO

COMPARISON WITH EXISTING SOLUTIONS

Smart hospital beds still aren’t a norm, and most hospitals in the world are still using either manual or electric beds. This is because not most companies manufacture smart hospital beds, and the ones that do it are very expensive to buy from. Let’s review a few popular smart hospital beds.

2.1 Stryker ProCuity

Stryker, one of the world's leading medical technology companies, has launched the industry’s first and only completely wireless hospital bed, ProCuity. This intelligent bed was designed to help reduce in-hospital patient falls at all acuity levels, improve nurse workflow efficiencies and safety, as well as help lower hospital costs. It is the only bed on the market today that can connect seamlessly to nurse call systems without the use of cables or wires.

2.2 Hillrom Centrella Bed

An integrated sensor in the bed frame detects heart rate and respiratory rate through the surface and updates the readings twice per second. This information is compiled into an algorithm to create a running trend. The bed then alerts when heart rate or respiratory rate exceeds a customizable threshold. When this happens, caregivers are notified via a SafeView+ System indicator light, GCI touchscreen alerts and audible alerts. Alerts are also sent from the bed to the nurse call system via dome lights, status board and mobile devices.

2.3 Midmark Protilt

ProTilt offers a smart and efficient way of continuous monitoring of patient body weight without need to transfer the patient from the bed. A digital display continually monitors a patient's weight with high accuracy of ±200 gms and provides instant reading for effective clinical management.

2.4 Our Smart Hospital Bed

Our smart hospital bed is a cost-effective alternative to these international smart hospital beds, and it can be deployed feasibly. The components and raw materials used for making our bed are much cheaper than the ones used in the other smart hospital beds. The pressure sensors are handmade, and the material used in them is readily available.

Our total cost to make one smart hospital bed has been estimated to be around Rs.50000 only. In comparison, the international smart hospital beds, which use state-of-the-art technology, but each offering only a very small subset of the features in our bed, cost around Rs.5 lac per bed.

Compared to the beds we reviewed earlier, it can be clearly seen that our smart hospital bed has about 4 to 6 times more features (stated in the next chapter) than those beds, yet having ten times less price.

CHAPTER THREE

FEATURES

Our smart hospital bed’s detailed features include;

- Detection of patient's presence on the bed
- Visualization of patient's body contour and position
- Patient's weight range estimation
- Alert generation if a disabled or coma patient is not turned periodically
- Patient’s turns pattern history graph
- Urine bag's filled percentage detection
- Alert generation if the patient has not urinated in 6 hours
- Alert generation if urine bag is more than 80% full
- Alert generation if urine bag has been removed
- Patient’s urination pattern history graph
- Patient’s body temperature prediction
- Body's heat map display

CHAPTER FOUR

ARCHITECTURE

4.1 Pressure Sensors

The sensors for body pressure detection, equivalent to flexiforce sensors, are handmade to make the product cost-effective. We are using a 6x4 grid of sensors connected to 2 MPR121s, a capacitive touch sensor controller which uses I2C for communication (a single MPR121 supports up to 12 electrodes). This 6x4 grid was preceded by a 4x3 grid of bigger sensors, but that had two disadvantages. Firstly, the bigger sensors experienced huge fluctuations in their capacitance values. Secondly, the 4x3 grid was not providing enough body contour resolution for accurate patient turn detection and even accurate body weight detection. The current 6x4 grid results in a fluid high resolution of the body contour, and a much better weight estimation.

Both the MPR121s are fixed under the bed, and are attached to an STM32 fixed on the left side of the bed. Figure 1 shows the pressure sensors attached on the bed.

Figure 1: Pressure sensors

We used thicker sensors with more layers on the middle of the bed, while the edges of the bed house thinner sensors with lesser layers. The architecture of the sensors in the 1st and 4th column of the grid is shown in figure 2, while the architecture of the sensors in the 2nd and 3rd column of the grid is shown in figure 3.

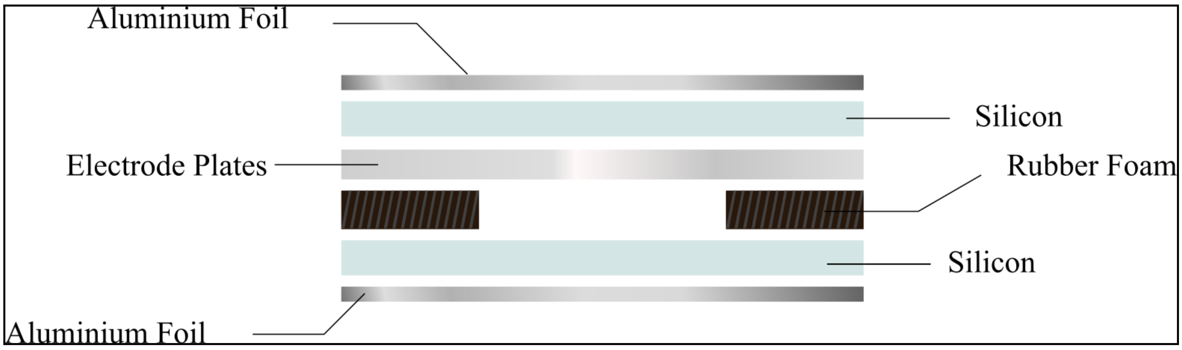

Figure 2: Outer column sensors

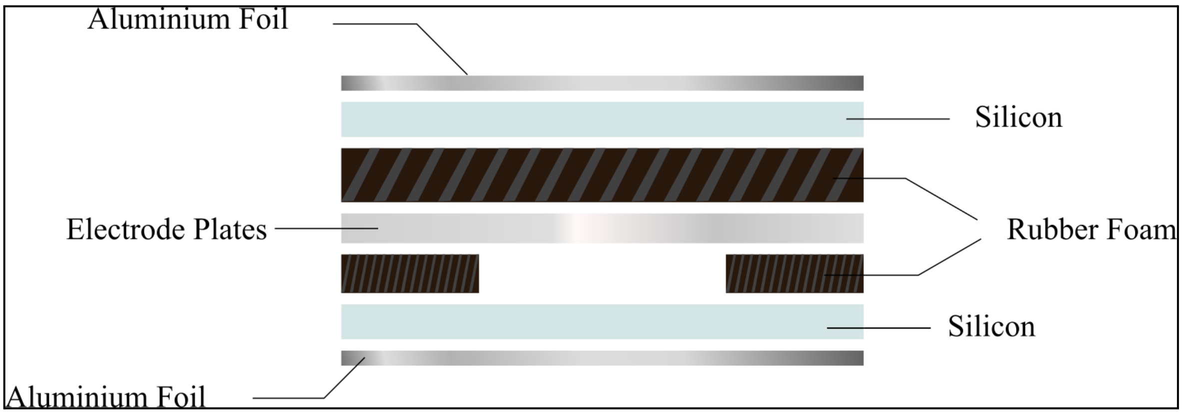

Figure 3: Inner column sensors

The aluminium foils of all the sensors are connected together to a common ground on the STM32, while the electrode plate of each sensor is connected to an electrode on one of the MPR121s.

4.2 Neural Networks

Both the following neural networks have been implemented in Python using Tensorflow Keras library.

4.2.1 Dataset

For use in both the following neural networks, we collected a dataset from 6 persons with 28 lying poses per person. 38 readings were captured when the bed was unoccupied. So a total of 206 calibrated capacitance differential values + actual body weights were collected. As we had a limited dataset, 95% of the dataset was used for training and only 5% was used for validation. This was preceded by an experiment where we also included sitting positions in the dataset, but that just confused the networks and provided lesser accuracy in both.

4.2.2 Body Detection

The body detection neural network is a classifier that detects whether a body is present or absent on the bed. Using a categorical cross entropy loss and an Adam optimizer, this network takes a 6x4 matrix of calibrated capacitance differential values from the 24 pressure sensors as input. This data then passes through 1 convolution layer, 1 flatten layer, and 4 dense layers, to eventually become an output of a one-hot encoded 2-vector. The last layer has a softmax activation, while all the other layers (except the flatten layer) have a ReLU activation. An output of 0 means that there is nobody on the bed, while an output of 1 means that there is someone on the bed. With the previous dataset, an output of 2 was also defined for sitting positions. Figure 4 shows the architecture of our body detection neural network model.

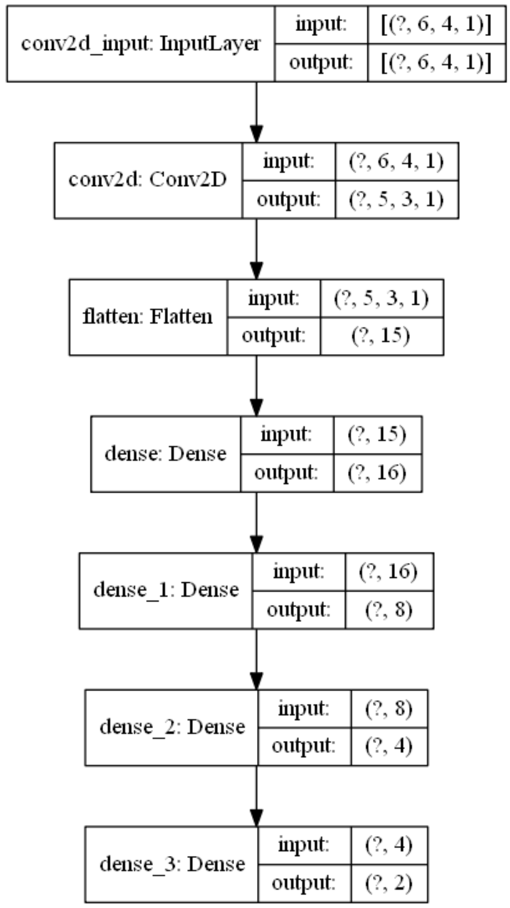

Figure 4: Body detection neural network

4.2.3 Weight Estimation

Weight estimation neural network is a regressor that estimates the weight range of the patient. Using a mean square error loss and an Adam optimizer, this network takes a 6x4 matrix of calibrated capacitance differential values from the 24 pressure sensors as input. This data then passes through 1 convolution layer, 1 flatten layer, and 5 dense layers, to eventually become a single value output of the estimated weight in kilograms. Then the lower value of the weight range is calculated by subtracting 5 from this value, while the upper value for the weight range is calculated by adding 5 to this value. All the layers (except the flatten layer) have a ReLU activation. With the previous dataset, the output of the body detection neural network (0, 1 or 2) was also concatenated with the data after it passed through the flatten layer. Figure 5 shows the architecture of our weight estimation neural network model.

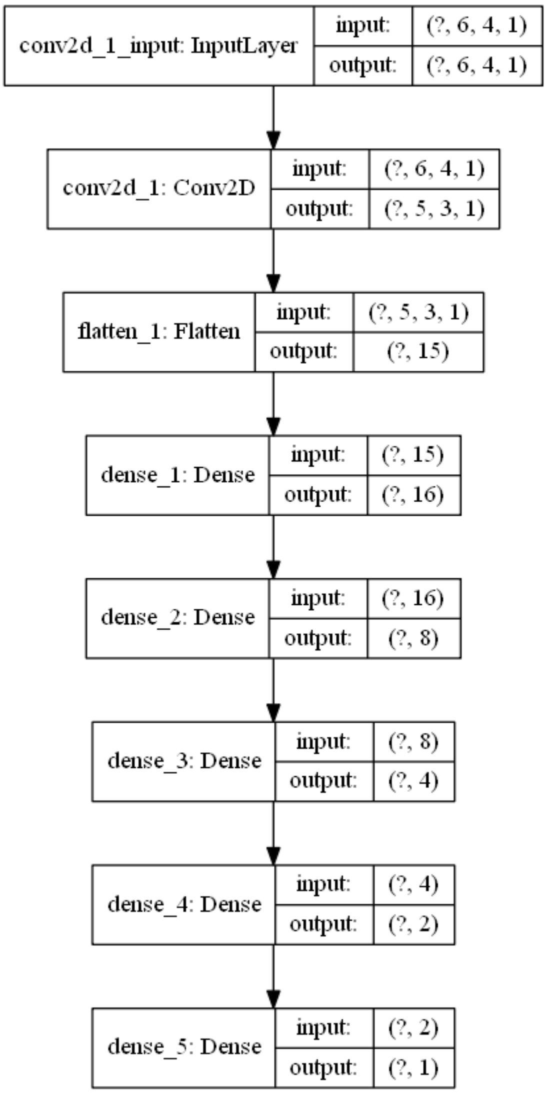

Figure 5: Weight estimation neural network

4.3 Infrared Sensor

The infrared sensor (MLX90641) is mounted on a servo motor attached at the end of a curved stand at the top of the bed in order to face the patient's face. Both of these are attached to an STM32, which is fixed on top of the stand. Figure 6 shows the temperature module.

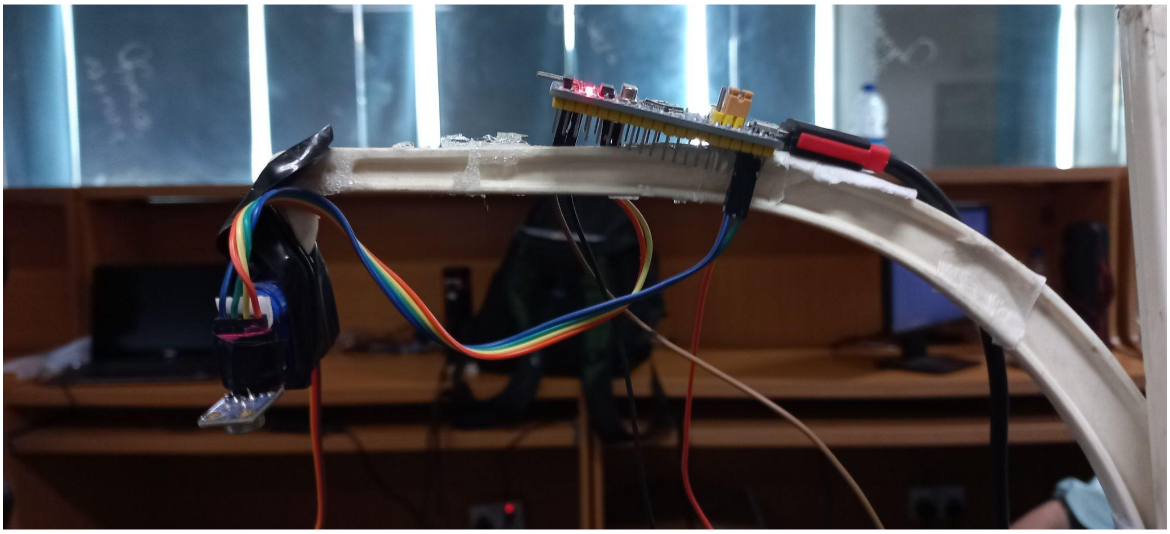

Figure 6: Temperature module

4.4 Load Cell

A 50 kg load cell has been mounted on the left side of the bed, used with HX711 (allows reading of weight readings from the load cell), connected to an STM32 fixed at its side. The urine bag is hung on the load cell between the two screws raised above it. Figure 7 shows the urine bag module.

Figure 7: Urine bag module

4.5 The Central Node

All the modules are attached to a central Raspberry Pi 3B, running Linux operating system. The working of each module has been implemented in the individual STM32s in C-language. The communication between the modules is also occurring directly over I2C, without any interference by the Raspberry Pi. The Raspberry Pi is being used to run the user interface, connect the modules to it, and run any required backend servers. However, during the development of the bed, a Windows laptop was used to run all this.

CHAPTER FIVE

IMPLEMENTATION

5.1 User Interface

Figure 8 shows the user interface of our smart hospital bed.

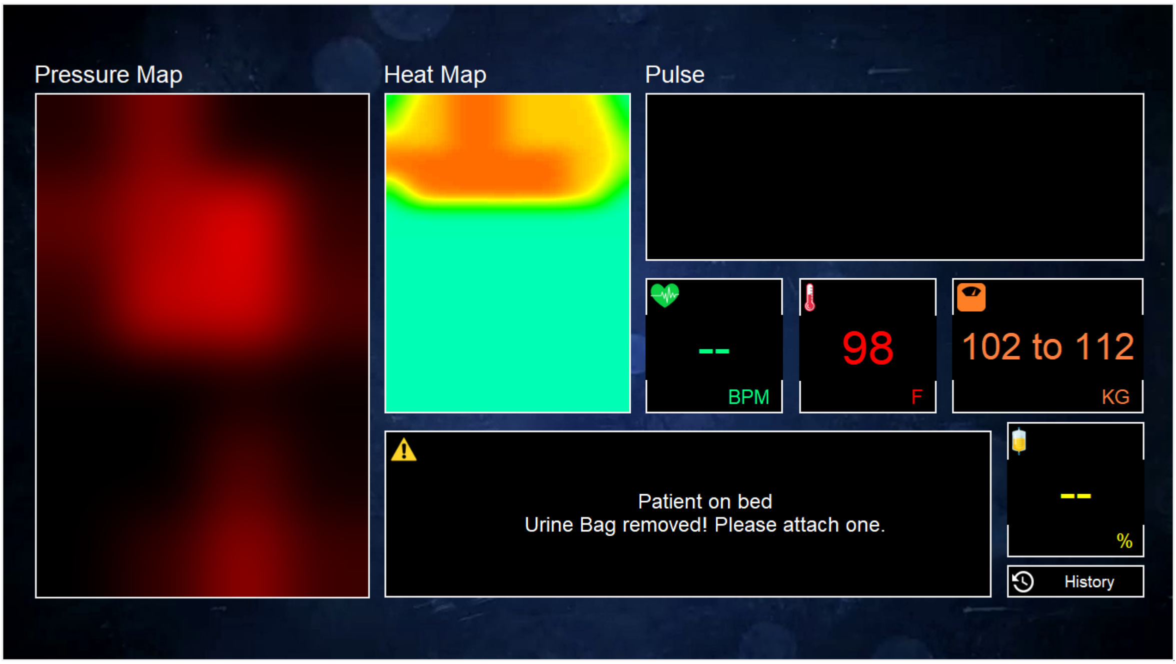

Figure 8: Smart hospital bed UI

The user interface has been implemented in Python using the Tkinter library. It communicates with all the modules through serial ports. The following information is displayed on the UI:

- Body contour visualization
- Body's heat map visualization
- Pulse visualization
- Heart rate (bpm)
- Temperature (°F)

- Weight (kg)
- Urine bag filled percentage
- Alerts from all the modules
- History button for Grafana graphs

The animated background of the UI turns red if an important alert is triggered, along with the sounding of a beeping alarm. The user interface also plays unique tones when it's launched, when the patient arrives on bed, as well as when the patient leaves the bed.

The body contour and the heat map outputs were each applied with a gaussian blur to smooth the blocky output. The body contour was originally made up of huge blocks in a 6x4 grid that corresponded to the 6x4 grid of sensors. So a large sigma value of 32 was used in the gaussian blur applied to it. The heat map, however, was made up of a grid of 80x36 small blocks, so a sigma value of only 2 was used in the gaussian blur applied to it.

The values for temperature, weight and urine bag filled percentage were not displayed directly as received from the relevant modules. Instead, as these modules are sending their data with the delays of 100 ms, 500 ms and 100 ms respectively, the values from each module received in a 3-second window were averaged and then displayed onto the UI to avoid rapid fluctuations and instead provide smoothly changing values. Also, the UI is updating itself every 42ms to keep a 24 FPS frame rate (particularly for the animated background).

The UI also shows helpful diagnostics if a module is disconnected from the Raspberry Pi. This may also be done intentionally if a certain module's readings are not required in the case of a particular patient. But mostly, this diagnostic information helps fix the errors of faulty USB cables in need of replacement or a human unplugging a module by mistake. For example, in figure 9, the temperature module has been unplugged from the USB port, which has been indicated in the alert box.

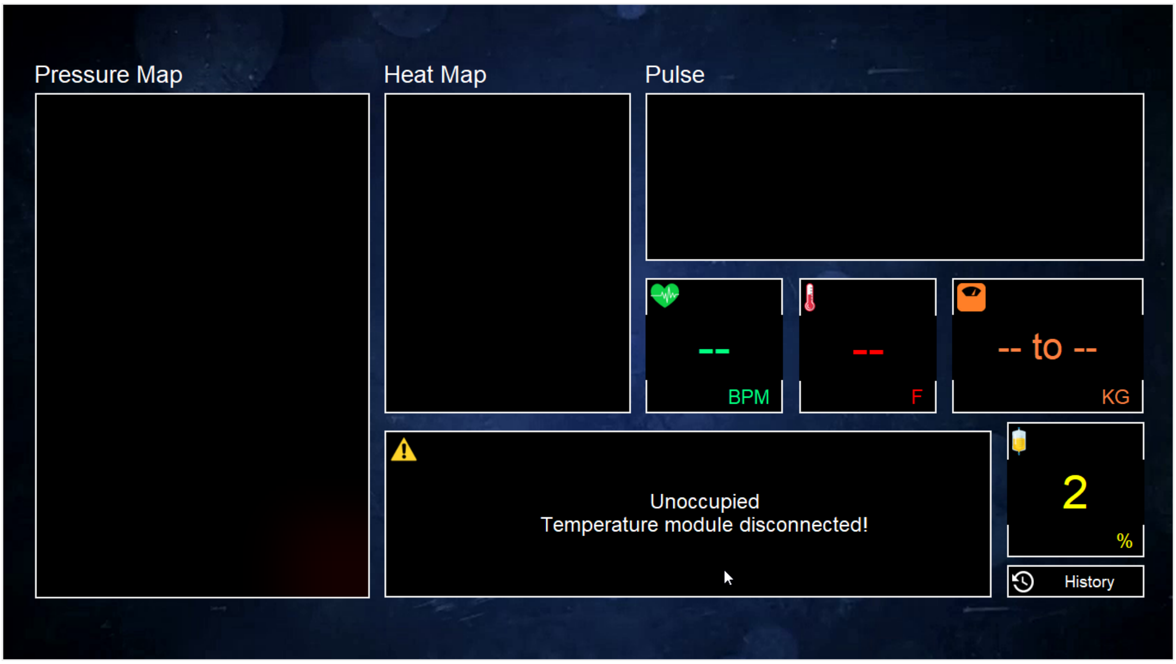

Figure 9: Diagnostic information

Apart from this, there is much more detailed information about the data being received from each module, being printed on the Python terminal running in the background of this user interface. Minimizing the UI shows that terminal, as shown in figure 10. The body detection module's information is sent to the UI in JSON format as it contains a lot of variables, whereas all the other modules send their information as single values. This information is in the background because it's only intended for the developers and testers of the smart hospital bed system.

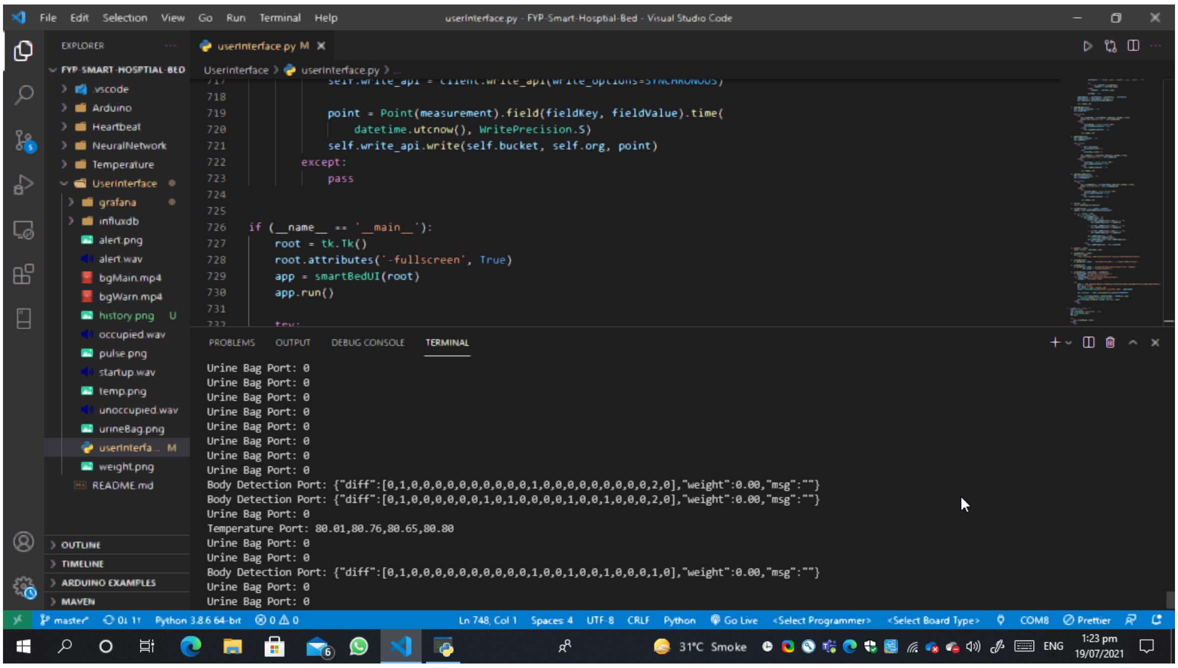

Figure 10: Background Python terminal

5.2 Body Detection Module

The body contour data from the pressure sensor is sent to two trained neural networks. The first one detects the presence of a body on the bed and if a patient is present, the second one estimates the weight range of the patient. If no patient is detected, all the modules except the urine bag filled capacity detection module, are turned off, and the UI displays "Unoccupied". As soon as a patient lays on the bed, all the modules become active and the UI says "Patient on bed". Moreover, we have developed a turn detection algorithm for coma patients. This checks if the activated sensors shift from left to right or vice versa. If the number of activated sensors that shift are greater than a threshold, a turn is detected. If a turn is not detected for many hours, an alert is generated saying "Patient hasn't been repositioned. Please do it." Figure 11 shows the UI when no patient is present on the bed, while figure 12 shows it when the patient hasn’t been turned in a while.

Figure 11: Bed unoccupied

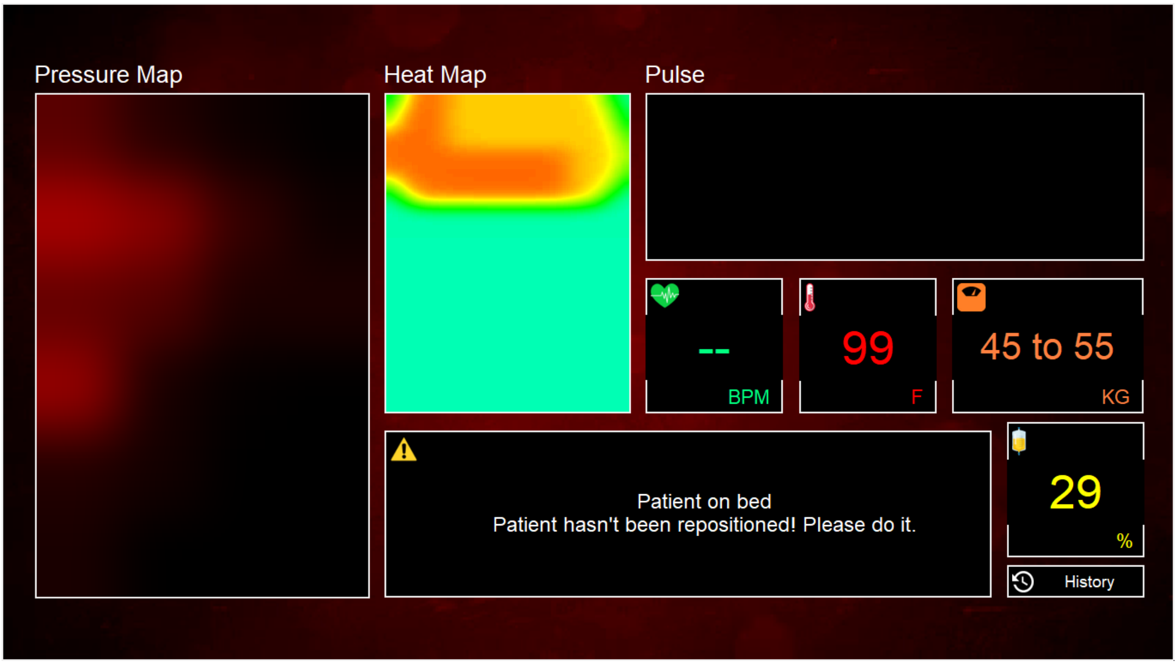

Figure 12: Patient not turned in a while

5.3 Temperature Module:

The infrared sensor (MLX90641) has a 90-degree field of view which gives an average temperature over a given area. To get an estimated temperature profile, our temperature estimation algorithm uses the temperatures recorded at three different angles by rotating the infrared sensor using a servo motor. Moreover, it uses the data from the pressure sensors, received over I2C from the body detection module's STM32 to the temperature module's STM32, to get the ratio of the patient's area to the total area. Also, when no patient is present on the bed, the UI stops sending a flag value of 1 to the temperature module's STM32 through the serial port, so it stops rotating the servo motor, records the bed's temperature and stores it as the temperature reading without the patient. Then the algorithm calculates the estimated temperature using the following formulas;

$$Ratio = \frac{BodyArea}{TotalArea}$$

$$SensorReading = (Ratio)(Temperature) + (1 - Ratio)(ReadingW/OPatient)$$

$$\downarrow$$

$$BodyTemperature = \frac{SensorReading - (1 - Ratio)ReadingW/OPatient}{Ratio}$$

The previous version of this module didn't have a servo motor, and instead had an ultrasonic sensor to calculate the distance to the patient's body. But as that version read temperature from only a single angle, it estimated the temperature with much lesser accuracy, with even worse results if the patient laid a little lower than the optimal position. Placing the sensor on the servo motor allows it to move and capture temperature readings covering the upper ⅓ area of the bed, leading to more accurate results. Also, due to the sensor now rotating, the ultrasonic sensor needed to be removed as it won't work at non-perpendicular directions to the bed. The average distance to the body was then hard-coded for each angle, with values that work for most patients, but would cause minor differences in the result for some patients. In both the previous version and the current version, the distance to the bed's mattress at each angle is always hard-coded as that never changes.

The body's heat map is also estimated by plotting each circular area from which the temperature was captured at each rotation of the servo motor, and averaging the temperatures in the areas where these circles overlap. These circular areas (shown in figure 13) are actually ellipses when captured at angles non-perpendicular to the bed, but we roughly considered them to be circles to be able to apply relevant calculations to them.

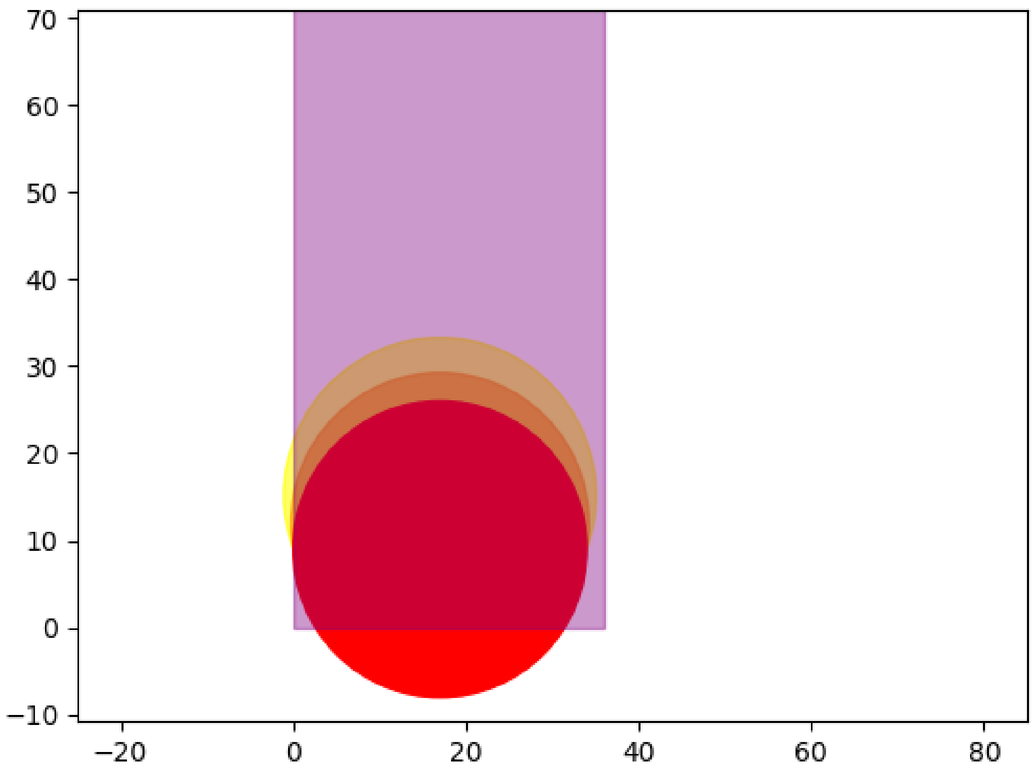

Figure 13: Circular areas on bed, covered by IR sensor

The heat map calculation also takes into account the data from the pressure sensors to display the calculated temperature only in areas where the body is present, and display the bed's temperature where the body isn't present. The temperatures in the heat map are displayed using the HSV color model. The hotter an area, the more red it is. The colder the area, the more greenish-blue it is. The heat map displayed on the UI can be seen in figure 8.

5.4 Urine Bag Module:

To calculate the urine bag's filled percentage, our algorithm uses the weight readings provided by the load cell and converts them to a percentage of the urine bag's total weight capacity. If a change in this percentage is not detected for 6 hours, it generates an alert saying "The patient has not urinated in the last 6 hours!". Similarly, if the bag is about to reach full capacity, it also generates an alert saying "Urine Bag almost full! Please drain/change it." These alerts are only generated if a patient is present on the bed. This presence is communicated to the urine bag module's STM32 from the body detection module's STM32 over I2C. If no urine bag is hung on the load cell, a notification is also shown with a single beep, saying "Urine bag removed! Please attach one." Figure 14 shows the UI when the patient has not urinated in the last 6 hours, while figure 15 shows the UI when the urine bag has been removed.

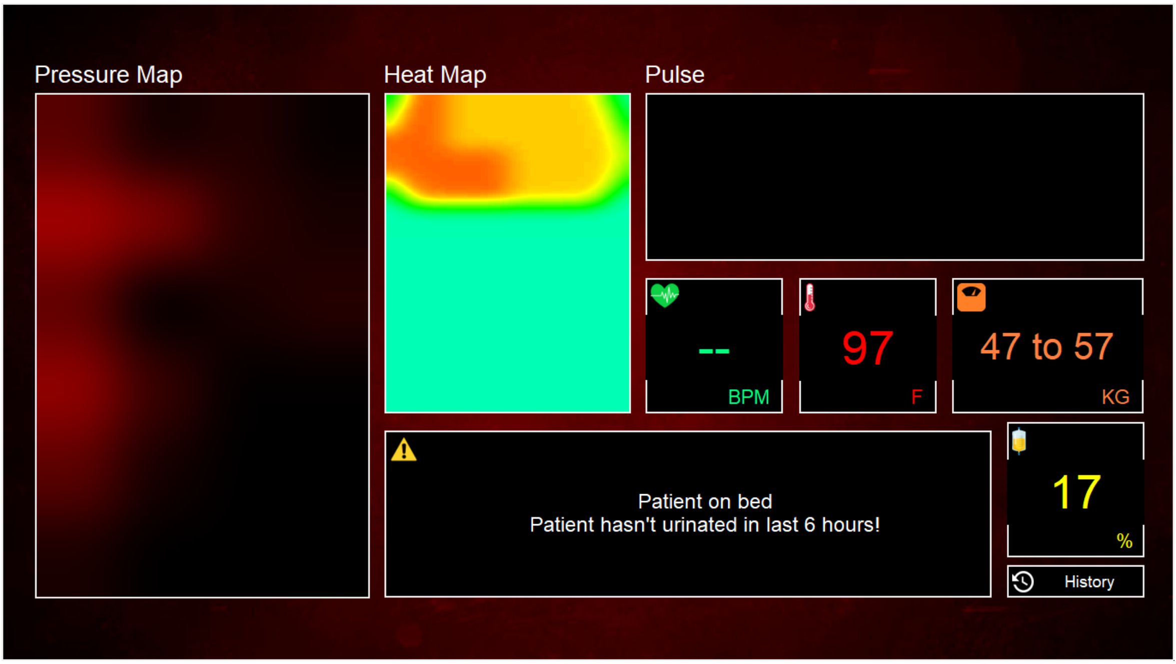

Figure 14: Patient didn't urinate in 6 hours

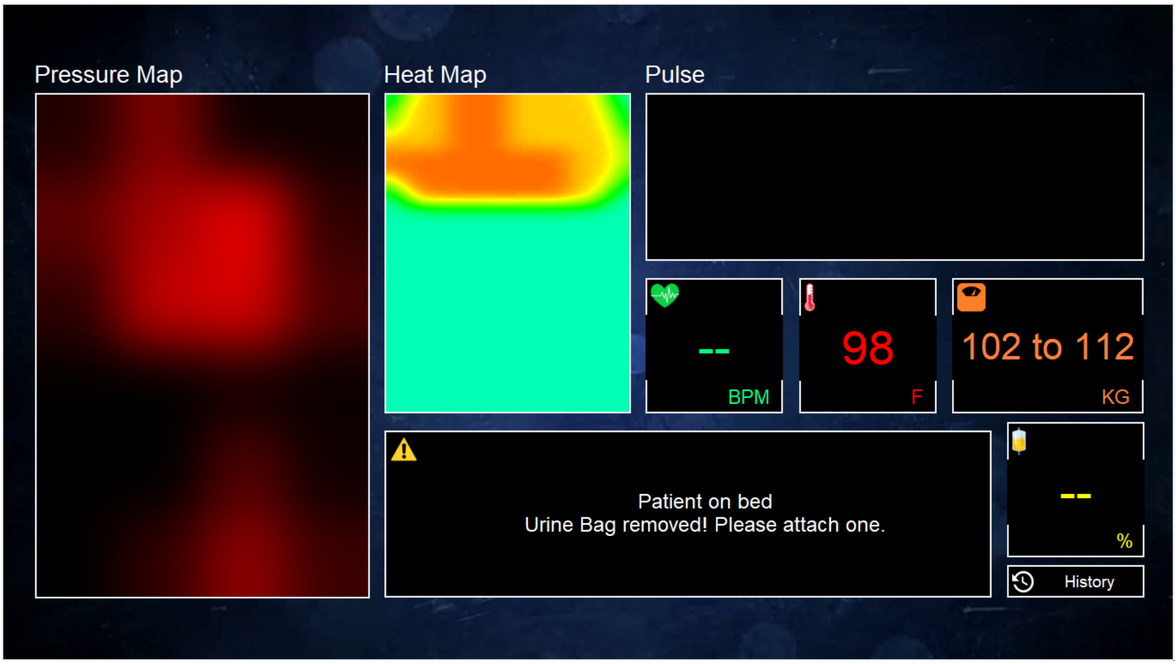

Figure 15: Urine bag removed

5.5 Pulse Detection Experiments

The goal of detecting the patient's pulse from the back side by embedding microphones into the mattress has shown a significant constraint. The heartbeats get muffled/distorted when recording them from the back side, while other noises like the lungs' sounds also get added. So far, we have only been able to detect the heartbeat in bits and pieces, with mostly no detection at all. Following are the experiments we performed in order to get the patient's pulse.

5.5.1 Experiment 1

We used a MAX9814 microphone array connected to an STM32, and captured 8000 samples per second with a centre mic placed directly above the heart and the surrounding two mics recording environmental noise, which was then cancelled from the main mic's signal to apply ANC. To further improve the results, a butter filter was then applied, followed by a FFT, which converted the signal into frequency domain. In this form, all the unneeded frequencies were dropped, and an inverse FFT was applied to project the signal back into the time domain. Under a controlled and orchestrated environment, we were able to detect significant sounds of heartbeat from the front side of the body, but didn't show considerable results from the back side. However, the noises could still not be completely eliminated, and the results were not satisfactory. Figure 16 shows the heartbeat signal that we were able to capture in experiment 1.

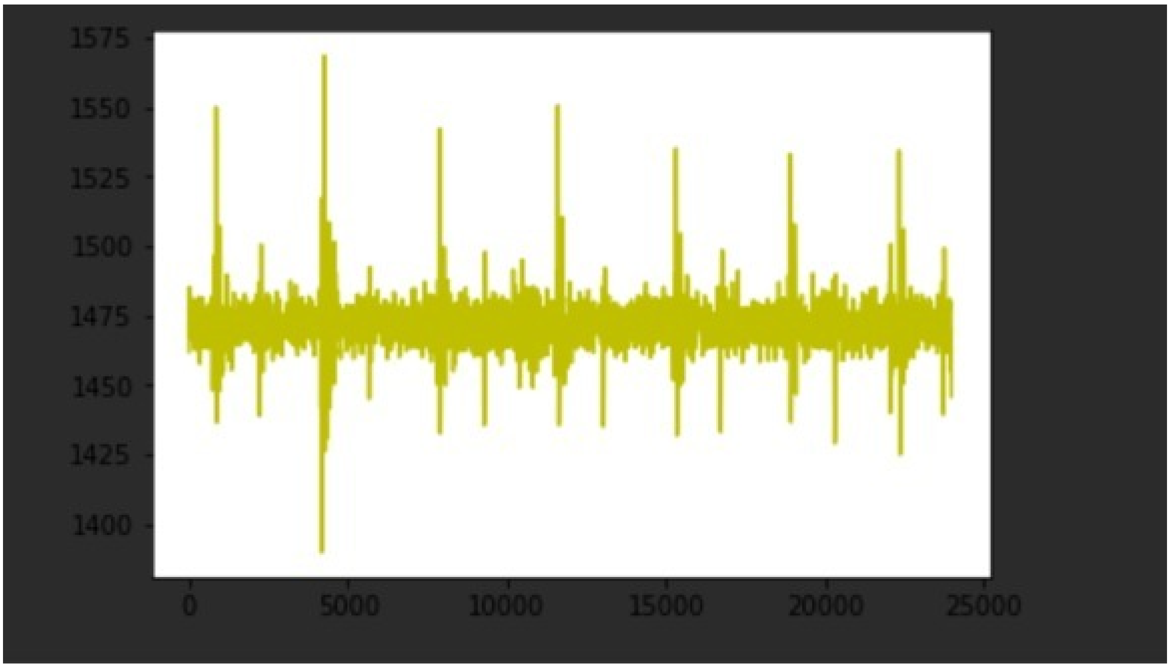

Figure 16: Experiment 1 heartbeat signal

5.5.2 Experiment 2

We used a stethoscope and fitted a 4 mm MEMS microphone embedded in its chest-piece (shown in figure 17), connected to the Raspberry Pi via ADC.

Figure 17: Stethoscope fitted with mic

The signal was then enhanced by Audacity to remove any ambient noise. With this device, we were able to detect clearer heartbeats from the front side without any noise reduction / cancellation. However, from the back side, we were able to detect some heartbeats, but they were never consecutive, and the beats in between were missed. Figures 18 and 19 show the waveform and the spectrogram of the captured heartbeats respectively.

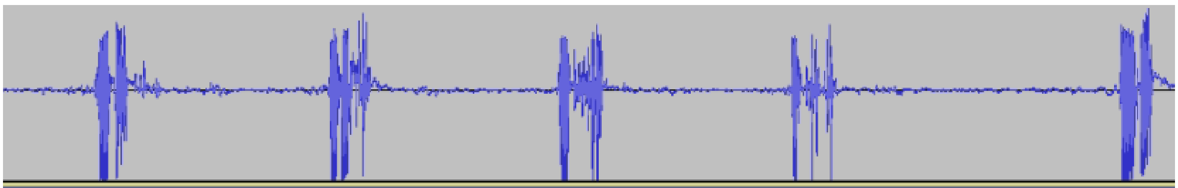

Figure 18: Experiment 2 heartbeat waveform

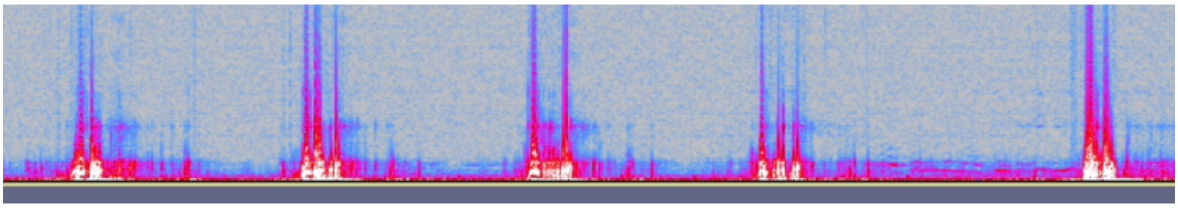

Figure 19: Experiment 2 heartbeat spectrogram

5.5.3 Experiment 3

We took a steel bowl, drilled a hole in it to embed a MAX90614 microphone in it, then sealed the hole using a glue gun. We also sealed its open side with a thin styrofoam sheet, making it a large stethoscope chest-piece (shown in figure 20). Then this was attached to an amplifier and used to detect heartbeats. The mic had a 60dB amplification, while the amplifier's board provided four amplifiers of 10dB each. This resulted in such a huge sound amplification that we were able to hear people walking in other rooms if this stethoscope was placed on the floor. So we were able to detect clear heartbeats from the front side of the body, without any noise cancellation. However, even with this much amplification, no significant results were achieved from the back side. This proved the word of the doctors we had consulted, who had told us that it is medically not possible to capture heartbeats from the back side.

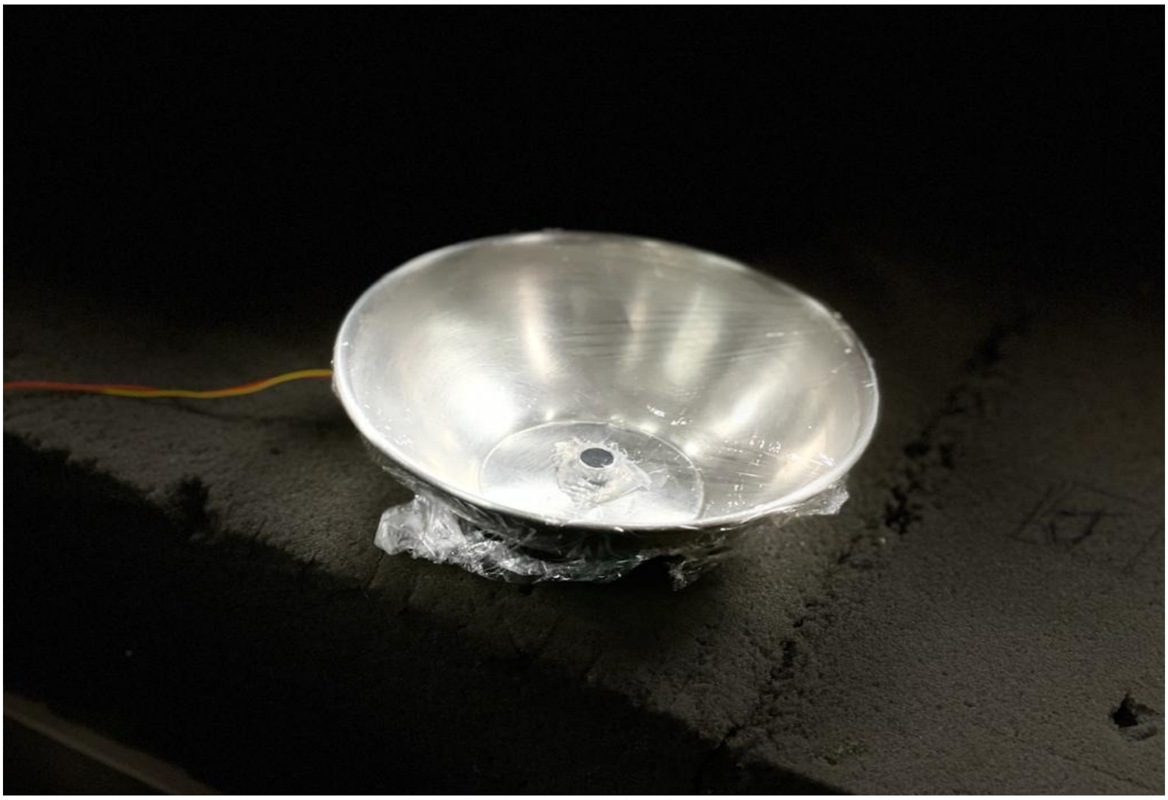

Figure 20: Experiment 3 large stethoscope

All in all, due to continuous failed results from the back side, we decided to currently not integrate pulse detection into our smart hospital bed.

5.6 Grafana History

Grafana is the open source analytics & monitoring solution for every database. The Grafana server is started in the background as soon as the smart hospital bed's user interface launches. We are using InfluxDB as our database to periodically store the data for our graphs. The InfluxDB server is also started when the UI launches. This server communicates with the Grafana server to enable the Grafana server to update its dashboard. The "History" button on the UI launches this Grafana dashboard, which shows two graphs: "Patient Turn Intervals" and "Urine Bag Filled Percentage" graph. The first one shows the number of seconds for which the patient has not turned w.r.t time, while the second one shows the percentage of the urine bag filled w.r.t. time. Figure 21 shows this Grafana dashboard.

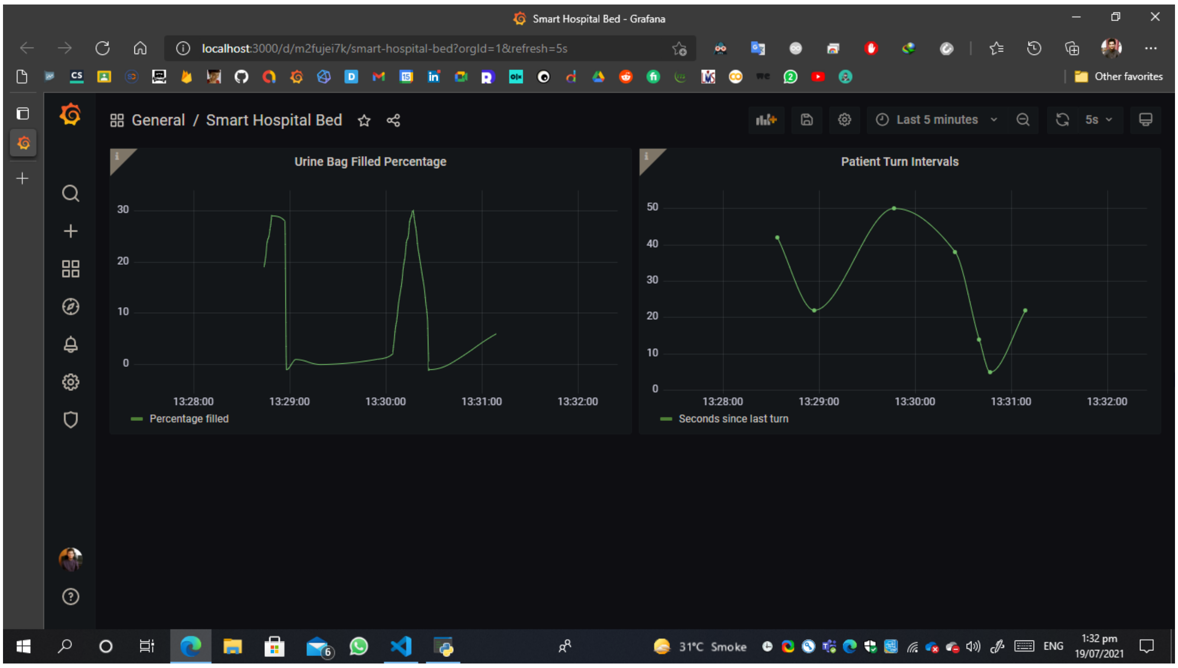

Figure 21: Grafana dashboard for smart hospital bed

If this Grafana dashboard is hosted on a server that the other devices can access, either through the local network or the internet, the doctors and nurses would be able to access this dashboard remotely on their PCs and even phones. This can help keep an eye on the patients without being physically present near the smart bed.

CHAPTER SIX

ANALYSIS

6.1 Body's Contour Visualization

The body contour displayed by our body contour visualization algorithm is mostly an accurate enough display of the actual body contour of the patient. Although, due to the pressure sensors being handmade, some of the pressure sensor readings may show huge random fluctuations sometimes. Also, if the mattress used on the bed is harder, then there would be unnecessarily more weight distribution on the sensors, leading to bad results. So a soft mattress is recommended for use with the bed.

6.2 Neural Networks

The body detection neural network as well as the weight estimation neural network both achieved a hundred percent training accuracy. The validation accuracy for the body detection neural network was also 100%, while the validation accuracy for the weight prediction neural network was 98%.

During testing, the body detection neural network always immediately detected that the patient is now on the bed or that the bed is now unoccupied. The weight estimation neural network, on the other hand, estimates a 10 kg range for the patient's weight, giving the hospital staff an idea of the patient's weight category. The patient's actual weight usually lies in this 10 kg range, but may sometimes be a little outside this range. However, as mentioned earlier, sometimes when a pressure sensor's reading shows huge fluctuations, a weight value may be calculated that is way off than the actual value.

6.3 Patient's Turns Detection

The patient's turns are always immediately detected by our turn detection algorithm. Although, sometimes if the patient turns really slowly, it may not detect the turn. But such slow turns are unnatural for the patients, and won't really occur in practical scenarios.

6.4 Temperature Prediction

The temperature reading predicted by our temperature prediction algorithm is within ±1 °F of the actual reading most of the time. A very few of the times, it may also go to ±2 °F accuracy. But the sometimes highly fluctuating value problem of a pressure sensor may disturb this calculation by providing an invalid ratio of the body's area to the total area, and the temperature calculated may become way off than the actual value.

6.5 Body's Heat Map Visualization

The fluctuation problem of the pressure sensors may also cause part of the heat map to flicker if the fluctuation was big enough to activate/deactivate a pressure sensor.

6.6 Urine Bag Filled Percentage Detection

The urine bag filled percentage displayed by our urine bag algorithm is always a 100% accurate one. Although, when the urine bag is removed, it may sometimes show a 0% value instead of "Urine Bag removed! Please attach one."

CHAPTER SEVEN

CONCLUSION

In short, the smart hospital bed is a cost-effective solution to the issues faced by patients and the medical staff. It aims to aid the medical staff in providing care to the patients efficiently. With the patients being monitored without the staff's frequent intervention, the latter will be able to look after the patients who are in a more critical condition, and that can save lives.

We think that with some more refinement, a greater level of accuracy can be achieved for each module, making the bed robust enough to be deployed in the real world hospitals as a cheap alternative to the internationally available smart hospital beds.

We would appreciate any future work to improve our smart hospital bed project. All our project's work and experiments, successes and failures, strengths and weaknesses, have been described in this report in immense detail. The software components developed for use in this project can be found in the following private GitHub repository:

mhamzai/FYP-Smart-Hosptial-Bed

However, to be able to access this repository, you will need to contact us at ammarasim40@live.com, and we will be happy to add you as a collaborator of this repository.