HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY PEOPLE USING ESP-32

Submitted in partial fulfillment of the requirements for the award of a Bachelor of Technology in Biomedical Engineering

By

AAKESH U (39240001)

YASWANT RAJASEKARAN (39240041)

DEPARTMENT OF BIOMEDICAL ENGINEERING

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC I 12B Status by UGC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AAKESH U (39240001)** and **YASWANT RAJASEKARAN (39240041)** who carried out the project entitled **"HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY USING ESP-32"** under my supervision from **October 2022** to **April 2023**.

Internal Guide
Ms. A. SABARIVANI, M.Tech., (Ph.D.)

Head of the Department
Dr. T. SUDHAKAR, M.Sc., Ph.D.,

HEAD OF THE DEPARTMENT
DEPARTMENT OF BIOMEDICAL ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar Rajiv Gandhi Salai
Chennai - 600 119

Submitted for Viva-voce Examination held on 18.04.2023.

Internal Examiner **External Examiner**

DECLARATION

We **AAKESH U (39240001) and YASWANT RAJASEKARAN (39240041) hereby** declare that the Project Report entitled **"HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY USING ESP-32"** done by us under the guidance of **Ms. A. Sabarivani, M.Tech., (Ph.D.),** Department of Biomedical Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

1. Aakesh.U

2.

DATE: 18.04.2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sudhakar, M.Sc., Ph.D., Head of the Department, Department of Biomedical Engineering** for providing us with necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Ms. A. Sabarivani, M.Tech., (Ph.D.)**, **Department of Biomedical Engineering** for her valuable guidance, suggestions, and constant encouragement paving the way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the Project.

We would also like to express our gratitude to our parents for supporting us in the successful completion of our project.

ABSTRACT

Designing solutions for the elderly in a community with an aging population is a rising challenge. IoT is a groundbreaking phenomenon that is dramatically altering our way of life and aspires to convert existing healthcare to one that is more personalized, preventive, and all-inclusive.

This Project concentrates on a microcontroller-based technique for curing elderly living patients that can monitor and record critical information for patients in crisis and as well as guidelines for setting off alarms to combine these major concerns with the caregiver. The low-cost/wireless capabilities transform this method into a secure and convenient bracelet that works everywhere.

This review would help the innovators, scientists, and others to understand the aforementioned problem and would help them to create an impact on the lives of all elders and help them overcome the challenges of daily life on their own.

Keywords: Healthcare, IoT, Microcontroller

TABLE OF CONTENT

LIST OF ABBREVIATIONS

ABBREVIATION		EXPANSION
3D	-	3-Dimension
AAPT	-	Android Asset Packaging tool
AC	-	Alternative Current
AF	-	Atrial Fibrillation
APK	-	Android Application Package
BPM	-	Blood Pressure per minute
CAD	-	Computer-Aided Design
DC	-	Direct Current
ECG	-	Electrocardiogram
GPS	-	Global Positioning System
IDE	-	Integrated Development Environment
IDEA	-	Interactive Data Extraction Analysis
IOS	-	iPhone Operating System
IOT	-	Internet Of Things
IR	-	Infra Red
LAN	-	Local Area Network
LED	-	Light Emitting Diode
MCU	-	Microcontroller Unit
MHZ	-	Mega Hertz
NC	-	Normally Closed
NO	-	Normally Open

PLC	-	Polylactic acid
PPG	-	Photoplethysmography
RC	-	Resistor Capacitor
RTD	-	Resistance Temperature Detection
SOC	-	System On Chips
SOS	-	System Of System
SPI	-	Serial Peripheral interface
TV	-	Television
UART	-	Universal Asynchronous Receiver Transmitter
USB	-	Universal Serial Bus
USP	-	Unique Selling Point
UV	-	Ultra Violet

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Despite accounting for just 7.4% of the people at the turn of the century, India's senior population, as well as those over 60, is fast growing. Existing healthcare services are under strain as a result of growing healthcare costs and an aging population. Patients over the age of 65 with chronic conditions, in particular, require continual long-term monitoring to detect changes in their condition as soon as feasible. The majority of study efforts have gone toward creating or upgrading telemedicine systems, which deal with the distant delivery of healthcare services via telecommunications, etc. Preventive interventions, promoting home care, and enhancing autonomy are seen to be effective ways to reduce the high expenditures of this population's hospitalization or care in specialist facilities. Preventive interventions, promoting home care, and enhancing autonomy are seen to be effective ways to reduce the high expenditures of this population's hospitalization or care in specialist facilities. Personal emergency response systems, like alarm watches, can be used by the first group to lessen the fear of falling. This anxiety is justified because research indicates that as people age, become more disabled, and have functional impairment, both the frequency of falls and the severity of problems rise. To avoid the enormous expenses and increased risk of death linked to poor medication adherence, other technologies, including prescription reminders, are essential. As people's worldwide health improves, a variety of sensors, like blood pressure monitors and glucose meters, can help. While certain specialized solutions are already on the market, their uses are still somewhat restricted. Modern communication gadgets and mobile and personal monitoring together have the potential to produce better-integrated tools that could fundamentally alter circumstances. We are particularly interested in examining the potential of smartwatches to see whether senior folks can effectively use them as personal assistants. The current demographic shift's rising senior population puts significant strain on the healthcare sector. Strong presumptions exist that keeping these people in their homes as long as possible will lower costs and improve their quality of life

There is a wide range of environmental changes and assistive technology that can be used to maintain elderly people at home. Wristband ESP32 is an innovative wearable technology that combines the capabilities of ESP32 and Arduino Nano, providing a platform for developing healthcare monitoring and tracking solutions for the elderly population. Wristband ESP32 offers a wide range of features, including sensor integration, communication capabilities, and programmability, that can be utilized for healthcare monitoring and tracking.

Smartwatches have the potential to provide personalized, real-time, and convenient monitoring and tracking solutions for the elderly population, allowing them to maintain their independence and age in place. The aim of this research paper is to explore the potential of smartwatches, specifically wristband ESP32 using ESP32 and Arduino Nano, as personal assistants for senior folks, focusing on the development, implementation, and evaluation of healthcare monitoring and tracking capabilities.

CHAPTER 2

LITERATURE SURVEY

Inam Hazem Maraqa et al. (2021) proposed a Microcontroller-based smartwatch for seniors that uses sensors to assess the elderly person's vital indicators. A GPS is used to identify the location, and a button and piezo sounder are connected to the microcontroller. When an elderly person presses the button, an alert is sent to the caretaker and a sound alarm is made, alerting people nearby to the situation and allowing them to assist the elderly person. A program created with MIT App Inventor may show the actual data of the crucial characteristics of the elderly person that is measured and uploaded to Google Firebase.

Akash SD et al. (2021) developed an Internet-of-things wearable technology system for the elderly to monitor their whereabouts and condition of health. The key characteristics of the system include location data, motion tracking, accident prevention, GPS access, health management, and a panic button. The system contains a GPS module, a Node MCU ESP32, an accelerometer, a sound sensor, and a buzzer. When a person falls, the accelerometer sensor produces a peak value at the accumulated acceleration. To transmit an alert to the Blynk IoT App, the NodeMCU ESP32 is utilised as just a microcontroller and Wi-Fi module. Real-time tracking of the user is made possible via the GPS module, and a dedicated panic button can be useful in an emergency. In the event of an accident, the sound sensor will pick up any form of strange, loud sound.

Israa S. AL-Forati et al. (2021) proposed an innovative method for creating an electronic medical wristband using basic and low-cost electronic components to assess the wearer's health data, including temperature, blood pressure, and oxygen saturation. The medical bracelet contains two types of sensors: the GY 906 and MAX 3100, which measure both temperature and ambient temperature. In shaped manufacturing, all components are compiled and the AutoCAD program is used to be compliant with system requirements. The BLNK application is used to gain and transmit test results in real-time.

Sumathy B et al. (2021) use a wearable belt's embedded sensors to detect the patient's temperature, respiration rate, and pulse rate. The output is shown on the LCD for reference, and a buzzer will alert as an emergency alarm. The output data is sent to the patient's family members and doctors for emergency scenarios.

Bassam Al-Naami et al. (2021) revealed that when a device is turned on, it first tries to connect to a Wi-Fi network and if unsuccessful, it is shut off. The sensors (oximeter, pulse, temperature, gyro, accelerometer, and GPS) are then sent to the LCD and sent to the Firebase Internet Cloud through the microcontroller (ESP32). Every 10 seconds, all of the sensors' data will be updated in the Firebase internet cloud, where it may be shared with healthcare professionals. A smartphone application has been created for this gadget to view real-time data on an old person's vital signs and send messages to remind them to take their prescription at the appropriate times.

Kaveri Ramesh Dabhadeet al.(2020) explains that the sound of the heartbeat is made by the valves that direct the blood from one place to the next. Photoplethysmography (PPG) is a technological advancement that uses an optical force variety to measure the volumetric variations of blood flow. PPG sensors use an infrared light transmitting diode (IR- LED) or a green LED as their main light source. IR-LEDs are used to measure blood flow, which is more intensely accumulated in some areas of the body, and a photodetector is used to measure the light intensity reflected from the tissue. It is possible to evaluate the changes in blood volume based on the measurement of the distinct light (calculated). Certain body parts must be covered by wearable PPG sensors.

Jeni Paay et al. (2022) developed a memento tool that consists of a smartphone, a smartwatch, small transmitters (proximity beacons) that residents already wear, and extra beacons affixed to particular items within the care facility. The caregiver keeps their smartphone in their pocket constantly looking out for nearby residents and objects that might be connected to those seniors' life stories. When a resident approaches, the caregiver's smartphone shows the resident's name, address, and photo. The resident's life story database can be searched for any references or pertinent keywords related to an item using a wireless link to the staff computer. The process of registering an artifact entails photographing it, physically fastening a proximity beacon to it, and tying keywords to the signal identifying the beacon.

The battery-powered proximity beacons have a life expectancy of a year or more, depending on how frequently they are used. A matching system links nearby residents and one or more labeled objects with fragments of their lives using keywords. An application was created for Android to help with the database entry of residents' stories.

Po-Cheng Chang et al. (2022) conducted a study of consecutive patients who underwent ambulatory Holter ECG monitoring to identify atrial fibrillation (AF) or gauge the severity of AF. All participants signed written consent forms and were disqualified if they were pregnant, under the age of 19, having a cardiac implanted electric device, or unable to wear a wristwatch and an ambulatory ECG Holter monitor for 24 hours. Baseline clinical parameters were gathered on the study enrolment day and the Holter and smartwatch were synchronized.

Puchuan Tan et al. used Polylactic acid (PLA) printing materials and a three-dimensional printer (Raize 3D) to create moulds for the blood pressure prediction bracelet (BPPW). Using a commercial Bluetooth board, they collected and transmitted wireless data to maintain the operational cycle. A linear motor (LinMot E1100, Suzhou, China) was used to provide mechanical traction to the BPPW.

Erdong Chen et al. (2020) used a wristband to monitor Atrial Fibrillation (AF). The wristband synchronizes with a smartphone application periodically or actively using a Bluetooth connection, and a photoelectric sensor on the inside of the wristband automatically collects the PPG signals at a sample rate of 50 hertz. Metal sensors on the outside and inside of the bracelet are used to collect the ECG data. ECG and PPG data was transferred from a smartphone to an Internet server to make decisions using AI algorithms. Both inpatients and outpatients at Peking University First Hospital's Cardiovascular Department participated in this investigation. The wristband captured PPG signals for three minutes while the patients were seated quietly. To begin recording an ECG for 60 seconds after receiving PPG data and finishing the automatic judgment, patients had to actively activate the bracelet. Two ECG doctors independently reviewed each record once the patient's information was removed from the initial wristband ECG recordings.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

- TO DESIGN A SMARTWATCH,

    1. That monitors the health of the elderly
    2. That alerts the caregiver and those nearby to assist the elderly when their health parameters are unstable
    3. That monitors the current location of the old person so that the caregiver may observe the current location of the old person via an app to which the smartwatch will transmit all data.
    4. That also includes fall detection, accident detection, and medication alert system in it.
    5. That has an emergency (SOS) button in it.

3.2 SCOPE

Healthcare monitoring and tracking wristbands can be very useful for elderly people, especially those who live alone or have chronic health conditions. An ESP-32 microcontroller and an Arduino nano can be used to create such a wristband, as it is a powerful and versatile chip that can connect to various sensors and wireless networks.

Some possible features of a healthcare monitoring and tracking wristband using ESP-32 and Arduino Nano for elderly people are:

3.2.1 Heart rate monitoring

An optical heart rate sensor can be used to monitor the heart rate of the wearer continuously. This data can be sent to a mobile app or a web dashboard for analysis and tracking.

3.2.2 Fall detection

A fall detection sensor can be integrated into the wristband to detect if the wearer has fallen. If a fall is detected, an alert can be sent to a caregiver or emergency services.

3.2.3 Sleepwalk detection

An accelerometer and gyroscope can be used to track the physical activity of the wearer, including steps taken and calories burned. This can help the wearer to stay active and monitor their health.

3.2.4 GPS tracking

A GPS module can be used to track the location of the wearer in real-time. This can be useful for caregivers to keep track of the wearer's whereabouts and ensure their safety.

3.2.5 Reminders and notifications

The wristband can be programmed to send reminders to the wearer for taking medications and can also send notifications to the caregiver in case of any abnormalities in the wearer's health parameters.

CHAPTER 4

MATERIALS AND METHODS

4.1 HARDWARE USED

4.1.1 ESP-32

Figure 4.1 represents the ESP32 is a low-cost System on Chip (SoC) Microcontroller from Systems, the developers of the famous ESP8266 SoC. As shown .It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6Microprocessor with integrated Wi-Fi and Bluetooth.

Specifications of ESP32:

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to240Hz.
- 520KB of SRAM, 448 KB of ROM, and 16 KB of RTC SRAM.
- Supports 802.11 b/g/h Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8- bit DAC.
- Serial Connectivity includes 4 X SPI, 2 X I2C, 2 X I2 S, 3 X UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller

for SDIO/SPI.

- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.

Fig. 4.1: ESP32

4.1.2 Arduino NANO

Arduino Nano is small which is represented in Fig. 4.2 which is a breadboard-friendly development board based on the ATmega328P microcontroller. It is similar in functionality to the Arduino UNO, but is smaller in size and uses a Mini-B USB cable for programming instead of a full-sized USB cable. The Arduino Nano board has 14 digital input/output pins (of which 6 can be used as PWM outputs), 8 analog inputs, and a 16 MHz quartz crystal. It also has a micro USB port for power and serial communication and a reset button. The Arduino Nano board can be programmed using the Arduino Integrated Development. The board can also be programmed using other programming languages and software, such as C++ and Atmel Studio. The Arduino Nano board is commonly used in DIY electronics projects, robotics, and automation. Its small size and versatility make it a popular choice for hobbyists and professionals alike.

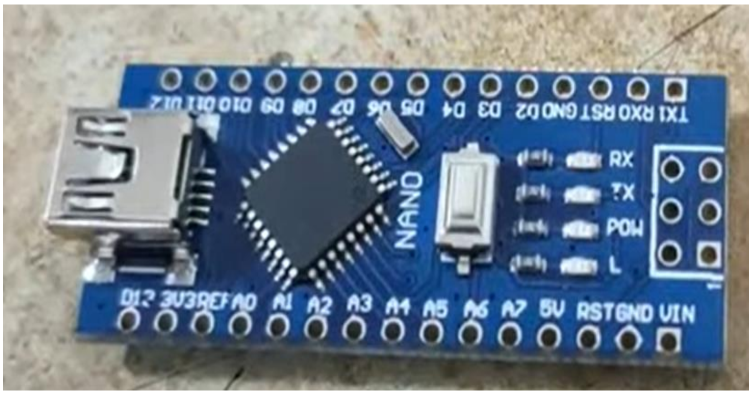

Fig. 4.2: Arduino nano

4.1.3 MPU6050

A three-axis accelerometer and a three-axis gyroscope are part of the MPU6050 Micro Electromechanical System (MEMS) that is denoted in Fig. 4.3 . It may be used to quantify velocity, direction, acceleration, displacement, and other motion-like qualities. The Digital Motion Processor (DMP) of the MPU6050 is capable of executing complex calculations. A 16-bit analog-to-digital converter is included in the MPU6050. It can catch three-dimensional motion at the same time because of this capability. This module offers several well-known and generally accessible functionalities; as a result, it may be used with a well-known microcontroller such as Arduino. The MPU6050 is a perfect alternative if you're looking for a sensor to regulate the motions of your drone, self-balancing robot, RC vehicle, or something similar. For connection with Arduino, this module employs the I2C module. The MPU6050 is less costly and has the advantage of being easy to integrate with an accelerometer and gyro.

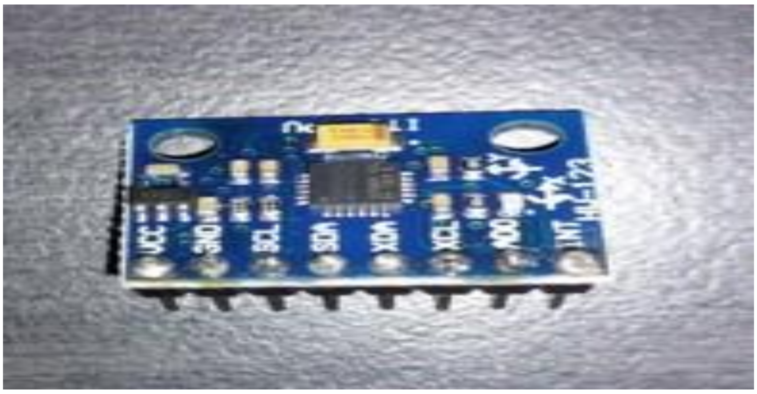

Fig. 4.3: MPU6050

4.1.4 Organic Light-Emitting Diode

OLED stands for Organic Light Emitting Diode. It is a type of display technology used in televisions, computer monitors, and mobile devices, among other applications. OLED displays i.e Fig. 4.4 are made up of organic compounds that emit light when an electric current is applied. Compared to traditional LCD displays, OLED displays offer better color accuracy, faster refresh rates, and deeper blacks because each pixel can be turned on and off individually. OLED displays also tend to be thinner and lighter because they do not require a backlight, as each pixel produces its own light. However, OLED displays can suffer from "burn-in," where static images displayed for a long period of time can cause permanent damage to the display. They can also be more expensive to manufacture than traditional LCD displays.

Fig. 4.4: Organic Light Emitting Diode

4.1.5 Temperature sensor

Fig. 4.5 shows a temperature sensor which is a device that detects and measures temperature. It can be used to monitor the temperature of various objects or environments and can be found in a wide range of applications, such as in HVAC systems, industrial processes, medical equipment, and consumer electronics. There are several types of temperature sensors, including thermocouples, RTDs (resistance temperature detectors), thermistors, and infrared sensors. Each type has its own advantages and disadvantages in terms of accuracy, response time, and cost. Thermocouples work by measuring the voltage generated by the temperature difference between two dissimilar metals. RTDs, on the other hand, use the resistance of a metal wire to measure temperature, Ronak Hada, Prajjwol Thermistors are made of semiconducting material,s and their resistance changes with temperature, while infrared sensors measure temperature by detecting the infrared radiation emitted by an object. Temperature sensors can be analog or digital, and some sensors may require additional circuitry to convert their output into a usable signal. The accuracy of a temperature sensor depends on several factors, including the sensor type, calibration, and environmental factors such as ambient temperature and humidity.

Fig. 4.5: Temperature sensor

4.1.6 Piezoelectric sensor

A piezoelectric sensor which is shown in Fig .4.6 is a type of sensor that generates an electrical charge in response to mechanical stress or pressure. It is based on the piezoelectric effect, which is the ability of certain materials to generate an electric charge when subjected to mechanical stress. Piezoelectric sensors are commonly used in a wide range of applications, such as in pressure sensors, accelerometers, and vibration sensors. They are particularly useful in measuring dynamic events, such as rapid changes in pressure or acceleration, due to their fast response times. Piezoelectric sensors are typically made of materials such as quartz, ceramic, or certain polymers. These materials are able to generate an electric charge in response to mechanical stress due to their asymmetric crystal structure, which causes the positive and negative charges in the crystal lattice to separate when the crystal is compressed or stretched. When a piezoelectric sensor is subjected to mechanical stress, such as pressure or vibration, it generates an electrical charge proportional to the magnitude of the stress. This charge can be measured and used to determine the magnitude of the applied stress. Piezoelectric sensors can be designed to be sensitive to different types of stresses, such as compression or shear, and can be configured to measure a wide range of pressures, accelerations, and vibrations. They are widely used in industrial, automotive, and aerospace applications for monitoring and control purposes.

Fig .4.6: Piezoelectric sensor

4.1.7 Push button

A push button as shown in Fig .4.7 is a simple switch mechanism that is activated by pressing a button or a key. It is a common type of switch used in a wide variety of electronic devices and equipment, including keyboards, remote controls, and control panels. A typical push button switch consists of a button or keycap, a spring-loaded mechanism that returns the button to its original position after it is released, and a set of electrical contacts that are closed when the button is pressed. When the button is pressed, it completes a circuit and allows current to flow through the switch. When the button is released, the contacts open and the circuit is broken. Push buttons can be designed with a variety of shapes, sizes, and colors to suit different applications. Some push buttons (Fig 4.7) are illuminated with LEDs or other light sources to provide visual feedback when they are activated. Others are designed with different types of actuators, such as rocker switches, slide switches, or toggle switches, to provide different modes of operation. Push buttons can be either normally open (NO) or normally closed (NC), depending on their design. In a normally open switch, the contacts are open when the button is not pressed and close when the button is pressed. In a normally closed switch, the contacts are closed when the button is not pressed and open when the button is pressed. Push buttons are reliable, durable, and easy to use, and are a versatile switch mechanism that can be used in a wide range of electronic devices and equipment.

Fig. 4.7: Push button

4.1.8 Light Emitting Diode

When electricity travels through a light-emitting diode as shown in Fig .4.8 produces light. Photonsare produced when electrons in a semiconductor recombine with electron holes, releasing energy. The energy required for electrons to pass the semiconductor's bandgap determines the hue of light (equivalent to photon energy). Multiple semiconductors or a light-emitting phosphor layer on the semiconductor device are used to generate white light. LEDs originally debuted as working electrical components . In 1962, emitting low-intensity infrared (IR) light. Infrared LEDs are employed in remote-control circuits, which are used in a broad variety of consumer electronics. The first visible-light LEDs were low-intensity and only came in red. Early LEDs were extensively used as indicator lamps and in seven-segment displays, where they replaced small incandescent bulbs. LEDs in visible, ultraviolet (UV), and infrared wavelengths have lately been produced with high, low, or intermediate light output, such as white LEDs excellent for room and outdoor area illumination. LEDs have also given rise to new types of displays and sensors, and their fast switching rates are useful in advanced communications technology, with applications ranging from aviation lighting to fairy lights, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, horticultural grow lights, and medical devices. LEDs outperform incandescent light sources in several ways, including lower power consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs have electrical limitations to low voltage and generally DC (notAC) power, the inability to provide steady illumination from a pulsing DC or an AC electrical supply source, and lower maximum operating temperature and storage temperature in exchange for these generally favorable characteristics. Unlike LEDs, incandescent lamps may be constructed to operate at virtually any source voltage, can use AC or DC interchangeably, and can deliver steady illumination at frequencies as low as 50 Hz whether driven by AC or pulsating DC. While an incandescent bulb may and frequently does function directly from an unregulated DC or AC power supply, LEDs frequently require the usage of electrical support components.

Fig. 4.8: Light Emitting Diode

4.1.9 Heart rate sensor

A heart rate sensor is a device that measures the heart rate or pulse of a person. It can be used to monitor the heart rate which is similar to Fig. 4.9 during exercise or physical activity, to track heart rate variability for stress management, or to diagnose certain medical conditions such as arrhythmia or tachycardia. Heart rate sensors come in various forms, including chest straps, wrist-worn devices, and even smartphone apps that use the phone's camera to detect changes in skin color caused by blood flow. The most accurate heart rate sensors use chest straps that wrap around the chest and use electrical signals to detect the heart rate. Heart rate sensors are commonly used in fitness and sports to help people monitor their heart rate during exercise and ensure that they are working at the appropriate intensity for their fitness goals. They can also be used in medical settings to help diagnose and monitor heart conditions.

Fig .4.9: Heart rate sensor

4.1.10 GPS module

A GPS module that is shown in Fig .4.10 is a device that receives signals from GPS (Global Positioning System) satellites to determine the user's geographic location. It is commonly used in navigation systems, mobile devices, and other applications where accurate location information is required. GPS modules typically consist of a GPS receiver, an antenna, and a processor that processes the received signals and calculates the user's location. The GPS receiver is responsible for receiving signals from GPS satellites, while the antenna is used to capture and amplify the signals. The processor uses the received signals to calculate the user's latitude, longitude, altitude, and other location-related information. GPS modules can be used for a wide range of applications, including navigation, tracking, and surveying. In navigation, GPS modules can provide turn-by-turn directions to help users navigate to their destination. In tracking, GPS modules can be used to monitor the location of vehicles, assets, or people. In surveying, GPS modules can be used to create maps or measure the position of geographic features. GPS modules are available in various sizes and formats, including modules that can be integrated into other devices, such as smartphones or wearable devices. They are also available as standalone modules that can be connected to other devices using standard communication interfaces, such as UART or SPI.

Fig. 4.10: GPS module

4.2 SOFTWARE USED

4.2.1 Arduino idle

The Arduino IDE is a free and open-source tool allowing you to generate and upload code to ESP32 boards. The IDE application is compatible with a wide range of operating systems, including Windows, Mac OS X, and Linux. It is compatible with the programming languages C and C++. IDE is an acronym for Integrated Development Environment. Sketching is the process of developing a program or code in the Arduino IDE. To upload the sketch written in the Arduino IDE software, we must link the Genuino and Arduino board to the IDE. The drawing is saved with the. In extension. The IDE creates a Hex file when a user writes code and compiles it. (Hex files are Hexadecimal files that Arduino can comprehend) and then uploaded to the device via USB. Every Arduino board contains a microcontroller, which receives and executes the hex file.

4.2.1 React native

React Native is an open-source framework for building mobile applications. It was developed by Facebook and was released in 2015. React Native allows developers to build mobile applications using the same principles and concepts as React, a popular JavaScript library for building user interfaces. React Native enables developers to create cross-platform mobile applications using a single codebase. This means that developers can write code once and deploy it to multiple platforms, such as iOS and Android, without having to write platform-specific code. This approach saves time and resources and makes it easier to maintain and update the application. React Native uses a combination of JavaScript and native code to create mobile applications. The user interface is built using native components, which are rendered using the native platform's rendering engine. The logic and business logic are written in JavaScript and executed using the JavaScript runtime. React Native has a large and active community of developers, which means that there are many resources available to help developers learn and use the framework. Additionally, React Native supports many third-party libraries and plugins, which can help developers to add additional functionality to their applications. Overall, React Native is a popular choice for building mobile applications because it offers a fast and efficient way to create cross-platform applications that are high-performing and easy to maintain.

4.2.3 Android studio

Android Studio is an Integrated Development Environment (IDE) for developing Android applications. It is developed by Google and is based on the IntelliJ IDEA software. Android Studio provides a comprehensive environment for developing, testing, and deploying Android applications. Android Studio includes a range of features and tools to make the development process easier and more efficient. These include Layout Editor: A drag-and-drop tool that allows developers to create Android app layouts visually, Code Editor: A powerful code editor that supports syntax highlighting, code completion, and other productivity features, Gradle build system: A build system that automates the build process, including compiling code and packaging resources into APK Android Emulator: A virtual Android device that allows developers to test their applications without having to use a physical device. Debugger: A debugging tool that allows developers to find and fix bugs in their code. Profiler: A performance analysis tool that helps developers identify application performance issues. Android Asset Packaging Tool (AAPT): A tool that creates the APK file for distribution. Android Studio supports a wide range of programming languages, including Java, Kotlin, and C++. It also supports the development of Wear OS, Android TV, and Android Auto applications.

4.3 METHODOLOGY

4.3.1 Research and analysis

Conduct thorough research on the existing technologies, tools, and devices used in healthcare monitoring and tracking for elderly people. Analysing the available solutions to identify the gaps and limitations and opportunities to improve the technology.

4.3.2 Requirements gathering

Identifying the requirements of the healthcare monitoring and tracking wristband based on the needs and preferences of the elderly people, their caregivers, and healthcare professionals.

4.3.3 Hardware and software design

Designing the hardware and software components of the wristband, including selecting appropriate sensors, and wireless modules, and developing the code for data acquisition, processing, and communication.

4.3.4 Prototype development

Building the prototype of the wristband using the designed hardware and software components, testing and troubleshooting the functionality, and making necessary improvements.

4.3.5 User testing and evaluation

Conducting user testing and evaluation of the prototype with elderly people, caregivers, and healthcare professionals, collecting feedback and suggestions for improvements.

4.4 BLOCK DIAGRAM WITH DESCRPTION

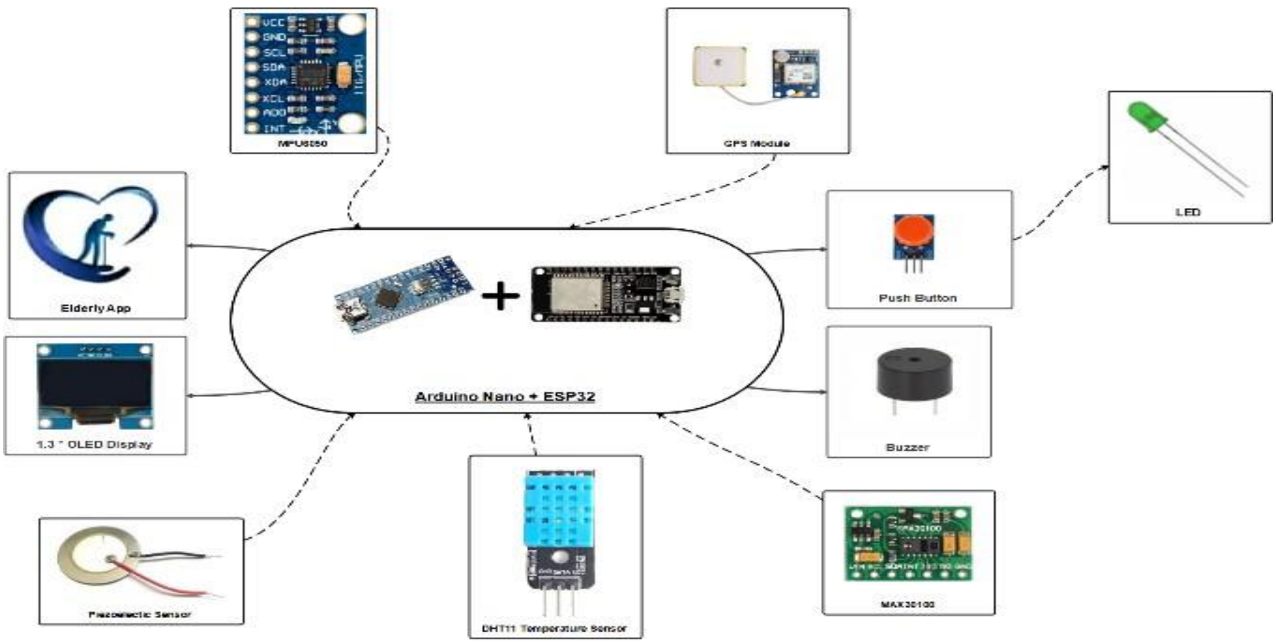

Fig .4.11: Block diagram of Wristband

The circuit consists of two microcontrollers, namely Arduino Nano and ESP-32, which serve as the central processing units of the system. The MPU6050 module, which comprises an accelerometer and a gyroscope, detects falls and sends the data to the microcontrollers. Upon receiving this data, the microcontrollers trigger the OLED display and buzzer to activate, and the fall detection status changes from "Normal" to "Abnormal" in the elderly app, accompanied by a notification. A push button, powered by the 3.3 V microcontrollers, controls an LED that lights up when pressed. The MAX30100 sensor measures heart rate and blood oxygenation, and the microcontrollers transmit this data to the OLED display and elderly app for display. The Neo 6M GPS module tracks the location of the elderly person in real-time, providing the caretaker with location information through the elderly app. The DHT11 temperature sensor monitors the body temperature of the elderly person, and the microcontrollers relay this data to the OLED display and elderly app for monitoring.A piezoelectric sensor, placed on the footwear, is connected to the microcontrollers via wires. When triggered by foot pressure, the microcontrollers prompt the elderly app to update the sleepwalk status from "Not Detected" to "Detected," and a notification is sent accordingly. The caretaker can set pill reminder times in the elderly app, which are then transmitted to the microcontrollers for display on the OLED screen of the elderly person's wristband. The microcontrollers establish a Wi-Fi connection with the elderly app, which is installed on the caretaker's smartphone.

CHAPTER 5
RESULTS AND DISCUSSION

5.1 RESULTS

Fig. 5.1: Prototype of wristband

The wristband system which is seen in Fig .5.1 is being developed here has the capacity to identify sleepwalking, indicating that the old person is not sleeping well. An external piezoelectric sensor that is inserted into the shoes and wired to the wristband is used to detect this. When an old person gets out of bed at night and places one foot on the floor, the piezoelectric sensor beneath that foot is activated by the pressure of the foot as shown in Fig. 5.3, which causes the buzzer to begin buzzing. Moreover, an alert message is sent to the elderly app. When the foot is not in contact with the piezoelectric sensor then the sleepwalk is not detected by this it can be inferred that the elderly person is sleeping well.

Fig. 5.2: When the heels are not in contact with the piezoelectric sensor

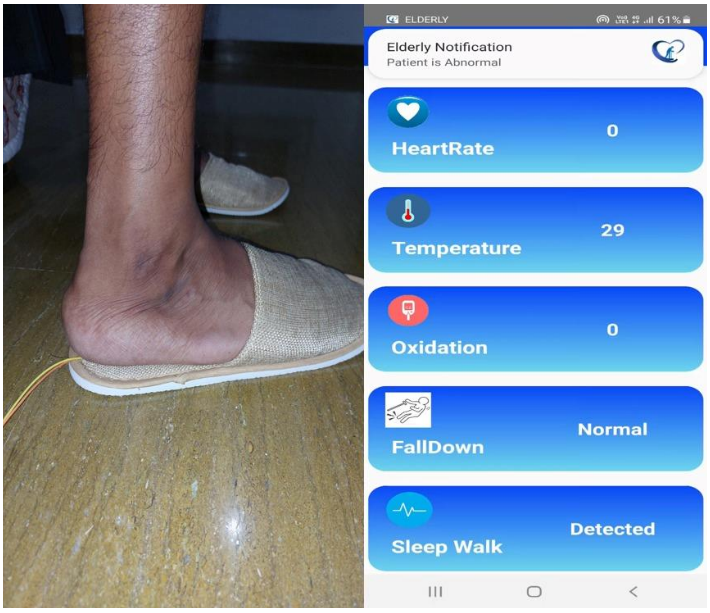

Fig .5.3: When the heels are in contact with the piezoelectric sensor sleepwalk is detected

The mechanism on the wristband also enables the elderly person to visually alert caretakers nearby by pressing a push button as shown in Fig.5.4 and Fig. 5.5 . (When the push button is pressed, the LED begins to illuminate, providing a visual alert.)

Fig .5.4: When the push button is pressed the LED starts glowing

Fig. 5.5: When the push button is not pressed the led doesn't glow

The wristband system that is denoted in Fig .5.6 and Fig. 5.7 also includes a GPS module that assists in displaying the elderly person's current location to the caretaker in the elderly app so that the caretaker can maintain track of the person's whereabouts and ensure the elderly person doesn't get lost.

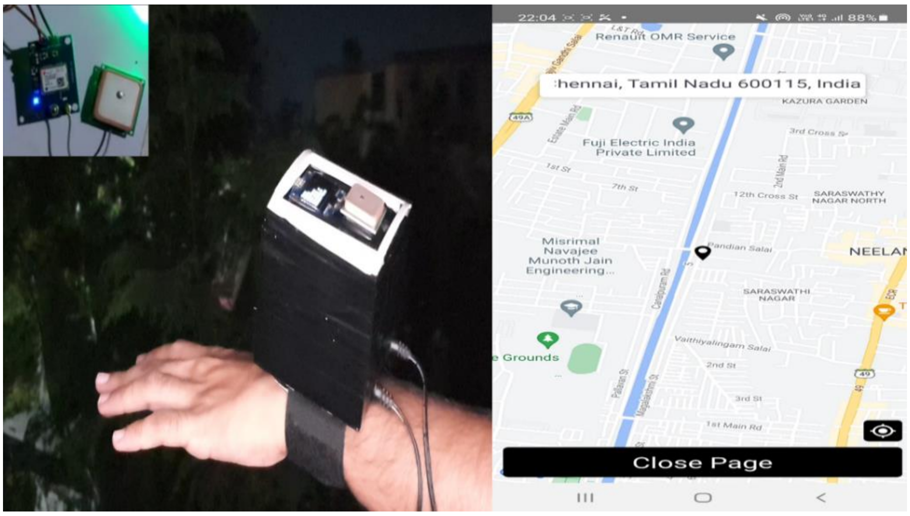

Fig .5.6: Current location of the elderly

The wristband system also includes a temperature sensor that, when in touch as shown in Fig. 5.7 with an old person's wrist, may detect that person's body temperature. The measured body temperature is then shown on the OLED screen and in the elderly app (Fig .5.8).

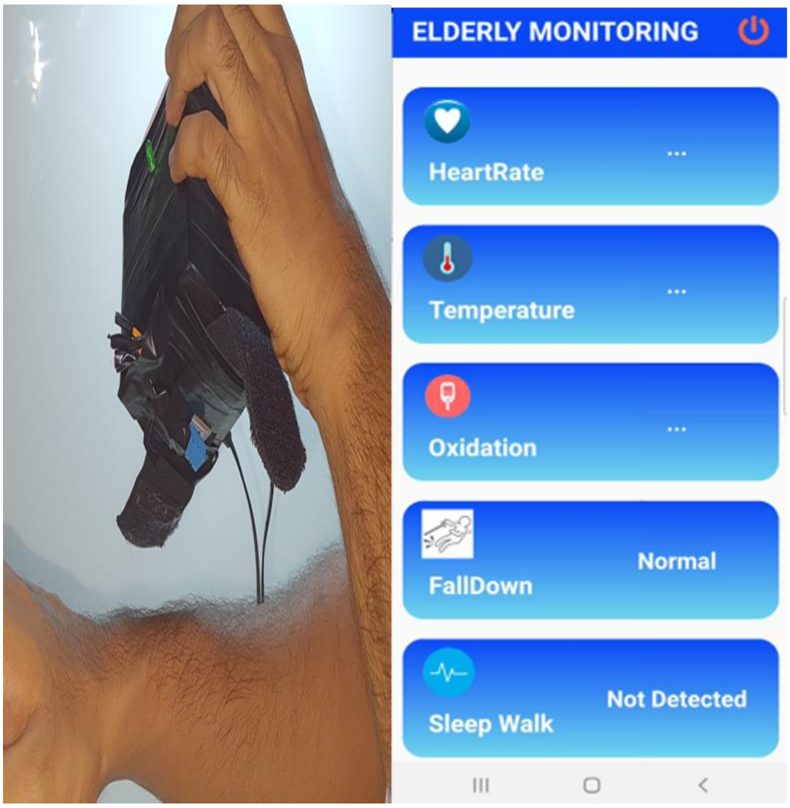

Fig. 5.7: Temperature is not shown when the wrist is placed away from the sensor

Fig. 5.8: Detection of body temperature when the wrist is in touch with the sensor

The wristband's system also includes an accelerometer (present in MPU6050), which is used to detect falls that is shown in Fig.5.9. When an old person falls unexpectedly as in Fig. 5.10, the accelerometer will sense it, activate the buzzer, and send an alert to the caretaker via the elderly app so that the caretaker may go over and assist the elderly person as he fell.

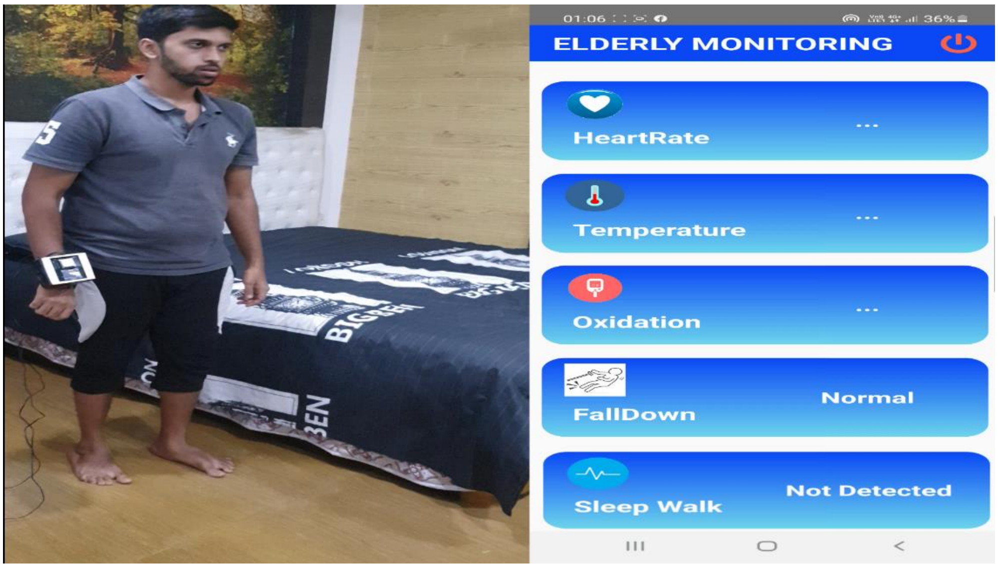

Fig .5.9: When a fall doesn't occur, the application shows "normal"

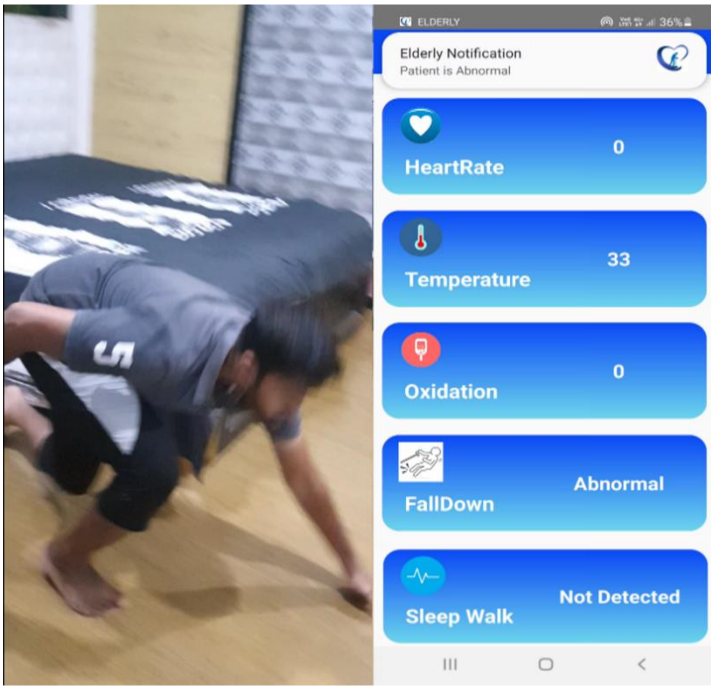

Fig. 5.10: When there is a fall a notification pops up as "abnormal"

When the caretaker enters the specific time at which the medications must be taken in an input box provided in the elderly app and pushes the "submit button," a specific time for taking medications is shown in Fig.5.11 on the OLED screen of the wristband's system

Fig. 5.11: When the caretaker sends a reminder message for taking pills at a particular time.

The smart wristband system also includes a heart rate and oxidation detecting sensor (MAX30100), which is worn on a finger and wires are linked externally to the smart wristband. So, this sensor measures the heart rate and oxidation levels in the senior person's body, and then it displays the information on the OLED display and in the elderly app as in Fig. 5.12

Fig .5.12: When the finger is placed on the sensor the oxidation and the heart rate are detected

CHAPTER 6
SUMMARY AND CONCLUSION

6.1 SUMMARY

The proposed smartwatch system can provide valuable assistance in monitoring the health and safety of elderly individuals, especially those living alone or with chronic conditions. The watch can monitor heart rate, track sleepwalks, detect falls, and send reminders for medication. It can also alert caregivers and nearby individuals in case of any health emergency, providing real-time GPS tracking through an app. The use of ESP32 and Arduino Nano microcontrollers provides a powerful and versatile platform for integrating multiple sensors and wireless networks.

6.2 CONCLUSION

The proposed design can significantly improve the quality of life for elderly individuals and reduce healthcare costs by promoting preventive care and home-based care. However, further research and development are required to address the limitations and challenges of developing such a smartwatch. These challenges include the design of user-friendly interfaces, addressing privacy concerns related to data collection and sharing, and ensuring the reliability and accuracy of the monitoring and alert systems. Nevertheless, the potential benefits of this technology for the growing senior population make it a worthwhile investment for healthcare providers and technology companies alike. In conclusion, this proposed smartwatch system can make a significant contribution to improving the lives of elderly individuals and promoting better healthcare outcomes, especially as the world's senior population continues to grow.

REFERENCES

[1] Akash SD, Anusha Upadhya, Harshitha BR, Hemanth, Mr. Shivarudraiah B. Smart Wearable System for The Elderly Using IOT. International Research Journal of Modernization in Engineering Technology and Science. e-ISSN: 2582-5208. Volume:04/Issue:07/July-2022. Impact Factor- 6.752. Page No: 2799 to 2806.

[2] Bassam Al-Naami, Hamza Abu Owida, Feras Al-Naimat, Moh'd Agha, Mohammed Abu Mallouh, Abdel-Razzak Al-Hinnawi. A New Prototype of Smart Wearable Monitoring System Solution for Alzheimer's Patients. Medical Devices: Evidence and Research 2021 (This work is published and licensed by Dove Medical Press Limited). Medical Devices: Evidence and Research 2021:14 pg no :423–433.

[3] Carolyn Weeks-Levy, Gautam Sadarangani, Carlo Menon. Perceptions of senior citizens on the use and desired features of a wristband for maintaining, strengthening, and regaining hand and finger function. Taylor & Francis Publications (Cogent Engineering). Weeks-Levy et al., Cogent Engineering (2020), 7: 1719572. DOI: https://doi.org/10.1080/23311916.2020.1719572. Page No: 1 to 13.

[4] Dr.Ruchika Singh, Rupesh Yadav, Sanidhya Gupta, Shitanshu Rai. Smartwatch For Senior/Elderly using a Microcontroller. EasyChair Publications April 12, 2022. Page No: 1 to 3.

[5] Erdong Chen, MD; Jie Jiang, MD; Meng Gao, MB; Jing Zhou, MD; Yong Huo, MD; Rui Su, MM; Sainan Zhu, MD. A new smart wristband equipped with an artificial intelligence algorithm to detect atrial fibrillation. Elsevier publications (Heart Rhythm Journal). Heart Rhythm, Vol 17, No 5PB, May 2020. DOI: https://doi.org/10.1016/j.hrthm.2020.01.034. Page No: 847 to 853.

[6] Francisco Javier González-Cañete, Eduardo Casilari. A Feasibility Study of the Use of Smartwatches in Wearable Fall Detection Systems. MDPI publications Sensors 2021, 21, 2254. DOI: https://doi.org/10.3390/s21062254. Page.No: 18 to 21.

[7] Frederic Ehrler, Christian Lovis. Supporting Elderly Homecare with Smartwatches: Advantages and Drawbacks. Studies in Health Technology and Informatics, 2014;205:667-71. DOI: 10.3233/978-1-61499-432-9-667. Page.No: 667 to 671.

[8] Inam Hazem Maraqa, Tasnim Yousef Al-Karaki, Rawand Iyad Yasin, Elayan Abu Gharbyeh. Smart Watch For Elderly Using Microcontroller. Palestine Polytechnic University Journal 2021.Page.No: 1 to 54.

[9] Israa S. AL-Forati, Bayadir A. Issa, Hayder M. Amer, Alla'a.I.Hussein. Developing a wristband to monitor Heartbeat and Temperature Using Internet of Things (IoT). International Journal of Advances in Engineering and Management (IJAEM). Volume 3, Issue 3 Mar. 2021, pp: 1111-1120. ISSN: 2395-5252. DOI: 10.35629/5252-030311111120 Impact Factor value 7.429.

[10] Jeni Paay, Jesper Kjeldskov, Ivan Aaen, Mette Bank. Usercentered iterative design of a smartwatch system supporting spontaneous reminiscence therapy for people living with dementia. SAGE publications (Health informatics Journal). DOI: 10.1177/14604582221106002. Health Informatics Journal 1–22.

[11] Kaveri Ramesh Dabhade, Kiran Suresh Mulik, Himani Jerath. IoT-based Wearable Smart Health Band Assistance. International Journal of Engineering Research & Technology (IJERT). ISSN: 2278-0181 Vol. 9 Issue 11, November-2020. IJERTV9IS1. Page.No: 100 to 106.

[12] Luis A. Durán-Vega, Pedro C. Santana-Mancilla, Raymundo Buenrostro-MariscaL, Juan Contreras-Castillo, Luis E. AnidoRifón, Miguel A. García-Ruiz, Sval A. Montesinos-López, Fermín Estrada-González. An IoT System for Remote Health Monitoring in Elderly Adults through a Wearable Device and Mobile Application. MDPI(Geriatrics (Basel). 2019 Jun; 4(2): 34). Published online 2019 May 7. DOI: 10.3390/geriatrics4020034. PMCID: PMC6631618. PMID: 31067819. Page.No: 1 to 13.

[13] Luís Correia, Daniel Fuentes, José Ribeiro, Nuno Costa, Arsénio Reis, Carlos Rabadão, João Barroso, António Pereira. Usability of Smart bands by the Elderly Population in the Context of Ambient Assisted Living Applications. MPDI

publications, Electronics 2021, 10(14), 1617; DOI: https://doi.org/10.3390/electronics10141617. Page.No: 1 to 18.

[14] c, MD; Ming-Shien Wen, MD; Chung-Chuan Chou, MD; Chun-Chieh Wang, MD; Kuo-Chun Hung, MD. Atrial fibrillation detection using ambulatory smartwatch photoplethysmography and validation with simultaneous Holter recording. Elsevier publications (American Heart Journal volume 247). DOI : https://doi.org/10.1016/j.ahj.2022.02.002. Page.No: 55 to 62.

[15] Puchuan Tan, Yuan Xi, Shengyu Chao, Dongjie Jiang, Zhuo Liu, Yubo Fan, Zhou Li. An Artificial Intelligence-Enhanced Blood Pressure Monitor Wristband Based on Piezoelectric Nanogenerator. MDPI publications (Biosensors Journal 2022,12,234). DOI: . https://doi.org/10.3390/bios12040234. Page: 1 to 11.

[16] Ranjeeth Kumar, M. Kaleel Rahman, E. Derrick Gilchrist, R. Lakshmi Pooja, C. Sruthi. Smart Band For Elderly Fall Detection Using Machine Learning. Natural volatiles and essential OILS journal 2021; Volume:8(Issue:5).Page.No::8269-8285.

[17] Rob Powers, Maryam Etezadi-Amoli, Edith M. Arnold, Sara Kianian, Irida Mance, Maxsim Gibiansky, Dan Trietsch, Alexander Singh Alvarado, James D. Kretlow, Todd M. Herrington, Salima Brillman, Nengchun Huang, Peter T. Lin, Hung A. Pham , Adeeti V. Ullal. Smartwatch inertial sensors continuously monitor real-world motor fluctuations in Parkinson’s disease. American Association for the Advancement of Science publication (Science Translational Medicine Journal) 2021 Feb 3;13(579):eabd7865.PMID: 33536284. DOI: 10.1126/scitranslmed.abd7865. Page.No: 29 to 31.

[18] Ronak Hada, Prajjwol Shrestha, Rohan Shrestha, Nabin Dangi, Rupesh Pun. Health Monitoring Band. KECConference2019, Kantipur Engineering College, Dhapakhel Lalitpur. Page.No: 104 to 107.

[19] Sathish Kumar.R , Nivedha.K , Anitha.K , Jayaprakash.D. An IOT-based health care system for elderly people. European Journal of Molecular & Clinical Medicine. ISSN 2515-8260. Volume 07, Issue 09, 2020.DOI: 10.31838/ejmcm.07.09.172. Page.No: 1568 to 1580

[20] Sumathy B, .Kavimullai, .Shushmithaa, .Sai Anusha. Wearable Non-invasive Health Monitoring Device for Elderly using IOT. RIACT2020 IOP Conference Series:

Materials Science and Engineering. 1012 (2021) 012011. doi:10.1088/1757-899X/1012/1/012011. Page.No: 1 to 4.

[21] Zhihua Wang, Zhaochu Yang, Tao Dong. A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities, and Monitor Vital Signs in Real-Time. MPDI publications, Sensors 2017, 17(2), 341; DOI: https://doi.org/10.3390/s17020341. Page.No: 1 to 7.

[22] Zhu Zhu, Yingying Ren, Pei Duan. Modeling of Smart Watch and System Construction Method for the Elderly Based on Big Data. Hindawi (Mathematical Problems in Engineering), Volume 2022,ArticleID 2606781.DOI:https://doi.org/10.1155/2022/2606781. Page.No: 1 to 9.

PROOF OF ACCEPTANCE

5th International Conference on
Smart Systems and Inventive Technology
(ICSSIT 2023)

23-25, January 2023 | Tirunelveli, India

http://icssit.com/2023/ | icssit.info@gmail.com

ACCEPTANCE LETTER

Manuscript ID : ICSSIT – 858

Manuscript Title: REVIEW ON HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY PEOPLE USING ESP-32

Author's: Aakesh U, Yaswant Rajasekaran, Sabarivani, Sudhakar T,

Sathyabama Institute of Science and technology.

IEEE XPLORE COMPLIANT ISBN 978-1-6654-7467-2

FRANCIS XAVIER ENGINEERING COLLEGE AN AUTONOMOUS INSTITUTION

Certificate of Presentation

This certificate is awarded to

Aakesh U

have successfully presented the paper entitled

REVIEW ON HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY PEOPLE USING ESP-32

at the
5th International Conference on
Smart Systems and Inventive Technology (ICSSIT 2023)
organized by Francis Xavier Engineering College, Tirunelveli, India
on 23-25, January 2023.

Session Chair

Conference Chair
Dr. G. Rajakumar

Principal
Dr. V. Velmurugan

IEEE XPLORE COMPLIANT ISBN 978-1-6654-7467-2

FRANCIS XAVIER ENGINEERING COLLEGE AN AUTONOMOUS INSTITUTION

Certificate of Presentation

This certificate is awarded to

Yaswant Rajasekaran

have successfully presented the paper entitled

REVIEW ON HEALTHCARE MONITORING AND TRACKING WRISTBAND FOR ELDERLY PEOPLE USING ESP-32

at the
5th International Conference on
Smart Systems and Inventive Technology (ICSSIT 2023)
organized by Francis Xavier Engineering College, Tirunelveli, India
on 23-25, January 2023.

Session Chair

Conference Chair
Dr. G. Rajakumar

Principal
Dr. V. Velmurugan

APPENDIX

```
#include "WiFi.h"
#include <PubSubClient.h> // Allows us to connect to, and publish to the MQTT broker
#include <DHT.h>
#include <Ticker.h>  //Ticker Library
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include "MAX30100_PulseOximeter.h"
#define REPORTING_PERIOD_MS     1000
PulseOximeter pox;
uint32_t tsLastReport = 0;
unsigned int heart = 0, oxidation = 0;
int data_timer = 0, servo_timer = 0, servo_flag = 0;
/* Assign a unique ID to this sensor at the same time */
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
Ticker timer;
uint8_t DHTPIN = 5;   // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
// if you require to change the pin number, Edit the pin with your arduino pin.
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "project";
const char* wifi_password = "project@123";
// MQTT
// Make sure to update this for your own MQTT Broker!
const char* mqtt_server = "broker.hivemq.com";
//const char* mqtt_topic = "energy_meter/power_cut";
const char* mqtt_topic = "sathya/time";
const char* mqtt_topic1 = "sathya/elder";
const char* clientID = "elder";
```

```
// Initialise the WiFi and MQTT Client objects
WiFiClient wifiClient;
PubSubClient client(mqtt_server, 1883, wifiClient); // 1883 is the listener port for the Broker
int humidity = 0, temperature = 0, relay = 4;
//const int xpin = 32; // x-axis of the accelerometer
//const int ypin = 35; // y-axis
//const int zpin = 39; // z-axis
//const int analogInPin = 33;  // Analog input pin that the potentiometer is attached to
//const int analogInPin1 = 34;  // Analog input pin that the potentiometer is attached to
//const int analogInPin2 = 26;  // Analog input pin that the potentiometer is attached to
//const int analogInPin3 = 36;  // Analog input pin that the potentiometer is attached to
//unsigned int angle = 0, angle_previous = 0, difference = 0, temp_value = 0, vibration = 0, sensorValue = 0, vib_difference = 0, vibration_previous = 0;        // value read from the pot
//float voltage;
//int angle_change = 0, data_received = 0, angle_change_timer = 0;
//float average_current;
String total_data, temp_string, heart_string, oxidation_string, data_string; //, vibration_string, voltage_string, current_string, humidity_string;
int data_received=0;
char char_data[50];
//int vib_count=0, vibration_value =0, get_current = 0;
int timer_count=0;
String partial_data;
char time1;
void onBeatDetected()
{
 //    Serial.println("Beat!");
}
void ReceivedMessage(char* topic, byte* payload, unsigned int length) {
 // Output the first character of the message to serial (debug)
 //  Serial.print(length);
 char input;
```

```
  if (length == 1)
  {
    input = ((char)payload[0]);
    time1 = input;
  }
}
bool Connect() {
  // Connect to MQTT Server and subscribe to the topic
  if (client.connect(clientID)) {
    client.subscribe(mqtt_topic);
    return true;
  }
  else {
    return false;
  }
}
void setup()
{
  Serial.begin(9600);
    WiFi.mode(WIFI_STA); //init wifi mode

  WiFi.begin(ssid, wifi_password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  dht.begin();
  delay(1000);
```

```
  Serial.println();
  Serial.print("Connecting to network: ");
  Serial.println(ssid);
//  WiFi.disconnect(true);  //disconnect form wifi to set new wifi connection

  client.setCallback(ReceivedMessage);
  if (client.connect(clientID)) {
    Serial.println("Connected to MQTT Broker!");
  }
  else {
    Serial.println("Connection to MQTT Broker failed...");
  }
  if (Connect()) {
    Serial.println("Connected Successfully to MQTT Broker!");
  }
  else {
    Serial.println("Connection Failed!");
  }
  if (!pox.begin()) {
    Serial.println("FAILED");
    for (;;);
  } else {
    Serial.println("SUCCESS");
  }
  pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
  // Register a callback for the beat detection
  pox.setOnBeatDetectedCallback(onBeatDetected);
    //Initialize Ticker every 0.5s
  timer.attach_ms(1000, calculate_data); //Use <strong>attach_ms</strong> if you need
time in ms
```

```
}

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length() - 1;

  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
void loop()
{
  // Make sure to call update as fast as possible
  pox.update();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
//          Serial.print("Heart rate:");
//          Serial.print(pox.getHeartRate());

    heart = pox.getHeartRate();
    heart_string = String(heart);
//          Serial.println(heart);

//          Serial.print("bpm / SpO2:");
//          Serial.println(pox.getSpO2());
```

```
    oxidation = pox.getSpO2();
    oxidation_string = String(oxidation);

//            Serial.println("%");
sLastReport = millis();

  }

}
void calculate_data()
{
    if(Serial.available() > 0)
  {

    data_string = Serial.readStringUntil('\0');
//    Serial.println(data_string);
    data_received = 1;
  }
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);
    if (isnan(h) || isnan(t) || isnan(f)) {
      Serial.println(F("Failed to read from DHT sensor!"));
      return;
    }
    humidity = h;
    temperature = t;
//    Serial.print("temp = ");
    //  Serial.println(humidity);
```

```
    temp_string = String(temperature);
//    Serial.println(temp_string);
    delay(100);


  if (!client.connected()) {
    Connect();
  }
  // client.loop() just tells the MQTT client code to do what it needs to do itself (i.e. check for
messages, etc.)
  client.loop();


  if (data_received == 1)
  {
//    partial_data = temp_string + "," + heart_string + "," + oxidation_string;
    total_data = temp_string + "," + heart_string + "," + oxidation_string + "," + data_string;
    //Serial.println("total_data:");
//    Serial.println(total_data);
    total_data.toCharArray(char_data, 50);
    client.connect(clientID);
    //delay(100); // This                              delay ensures that client.publish doesn't clash with
the client.connect call
    client.publish(mqtt_topic1, char_data);
   // Serial.println("Published data");
    data_received = 0;
    data_string = " ";
  }
//  delay(1000);
 Serial.print('2');
 Serial.print(',');
 Serial.print(temp_string);
 Serial.print(',');
 Serial.print(heart_string);
```

```
 Serial.print(',');
 Serial.print(oxidation_string);
 Serial.print(',');
 Serial.print(time1);
 Serial.println(',');
}
```

CODING FOR HOMWE PAGE

```
import React, {Component, useState, useEffect} from 'react';
import {
  View,
  Text,
  StyleSheet,
  Image,
  Dimensions,
  TouchableOpacity,
  Alert,
  ScrollView,
  TextInput
} from 'react-native';
import LinearGradient from 'react-native-linear-gradient';
const {width, height} = Dimensions.get('screen');
import MQTT from 'react-native-mqtt-new';
import Icon2 from 'react-native-vector-icons/FontAwesome5';
import RNRestart from 'react-native-restart'; // Import package from node modules
import AsyncStorage from '@react-native-async-storage/async-storage';
import Address from './Address';
import axios from 'axios';
export default function PatientHealthStatus(props) {
  const [Temperature, setTemperature] = useState('...');
```

```
const [Oxidation, setOxidation] = useState('...');
const [HeartRate, setHeartRate] = useState('...');
const [Accelerometer, setAccelerometer] = useState('...');
const [Sleep_Walk, setSleep_Walk] = useState('...');
const [lat, setlat] = useState(12.9494);
const [lon, setlon] = useState(80.2488);
const [screen, setscreen] = useState(0);
const [publicdata, setpublicdata] = useState("");
useEffect(() => {
 // navigation.addListener('focus', () => {
 MQTT.createClient({
   uri: 'mqtt://broker.hivemq.com:1883',
   clientId: 'your_client_id',
 })
   .then(function (client) {
     client.on('closed', function () {
       // console.log('mqtt.event.closed');
     });
     client.on('error', function (msg) {
       // console.log('mqtt.event.error', msg);
     });
     client.on('message', async function (msg) {
       var mqttvalue = msg.data;
       var splits = mqttvalue.split(',');
       console.log(splits)
       setHeartRate(splits[0]);
       setTemperature(splits[1]);
       setOxidation(splits[2]);
       setAccelerometer(splits[3]);
       setSleep_Walk(splits[4]);
       setlat(splits[5]);
```

```
setlon(splits[6]);
if (splits[7] == '1') {
console.log("notificaytion called")
const options = {
url: 'https://fcm.googleapis.com/fcm/send',
method: 'POST',
headers: {
'Content-Type': 'application/json',
Authorization: [
'key',
'AAAA9myXIrY:APA91bHsrnPMm8TUAk7IfPCdXzdTZdz0riWCQIaoTpWVTCGMiWakwW
WEoAFdERvk6LQ8esGb-rRJvFTTY9NRTcVc-
O9WrNS_MaE3GNmoJ7tbRrqt46RRXIzIPVO4NBo21LFEA8IRMtSD',
].join('='),
},
data: JSON.stringify({
notification: {
title: 'Elderly Notification',
body: 'Patient Health is Abnormal',
sound: 'my_sound',
android_channel_id: 'sound_channel',
},
to: props.notification_mobile_value,
priority: 'high',
}),
};
axios(options)
.then((response) => {})
.catch((error) => {
console.log(error);
});
        }
```

```
});
client.on('connect', function () {
// console.log('connected');
client.subscribe('satels4', 0);
});
client.connect();
})
.catch(function (err) {
// console.log(err);
});
// })
});
function publishData() {
MQTT.createClient({
uri: 'mqtt://broker.hivemq.com:1883',
clientId: 'your_client_id'
}).then(function (client) {
client.on('closed', function () {
console.log('mqtt.event.closed');
});
client.on('error', function (msg) {
console.log('mqtt.event.error', msg);
});
client.on('message', function (msg) {
console.log('mqtt.event.message', msg);
});
client.on('connect', function () {
console.log('connected');

client.publish("satelp4", publicdata, 1, false);
// client.disconnect();
```

```
});
client.connect();
}).catch(function (err) {
console.log(err);
});
}
if (screen == 0) {
return (
<View style={{flex: 1}}>
<View
style={{
flexDirection: 'row',
justifyContent: 'space-around',
alignItems: 'center',
backgroundColor: '#0a49f7',
}}>
{/* <Header style={{ backgroundColor: '#1187ba' }}> */}
{/* <Body style={{ flexDirection: 'row',justifyContent:'space-between' }}> */}
<Text
style={{
fontSize: 24,
color: 'white',
padding: 10,
fontWeight: 'bold',
}}>
ELDERLY MONITORING
</Text>
<TouchableOpacity
style={{padding: 10}}
      onPress={() => {
Alert.alert('Logout', 'Do you want to Logout ?', [
```

```
{
text: 'CANCEL',
onPress: () => console.log('canceled'),
style: 'cancel',
},
{
text: 'OK',
onPress: () => {
// AsyncStorage.clear()
AsyncStorage.setItem('login', '0');
RNRestart.Restart();
},
},
]);
}}>
<Icon2 name="power-off" color="#FF6347" size={30} />
</TouchableOpacity>
</View>

<ScrollView
style={{flex: 1, flexDirection: 'column', padding: 10, marginTop: 20}}
showsVerticalScrollIndicator={false}>
<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
style={styles.linearGradient}>
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
        alignItems: 'center',
```

```
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'flex-start',
}}>
<Image
style={{width: width * 0.2, height: 50, right: 10}}
source={require('./heart.png')}
resizeMode={'contain'}
/>

<Text style={styles.buttonText}>HeartRate</Text>
</View>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'center',
}}>
<Text
style={{
fontSize: 21,
fontWeight: 'bold',
color: 'white',
textAlign: 'center',
fontFamily: 'Gill Sans',
}}>
{HeartRate}
</Text>
```

```
</View>
</View>
</LinearGradient>

<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
style={styles.linearGradient}>
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
alignItems: 'center',
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'flex-start',
}}>
<Image
style={{width: width * 0.2, height: 50, right: 10}}
source={require('./temperature.jpeg')}
resizeMode={'contain'}
/>

<Text style={styles.buttonText}>Temperature</Text>
</View>
<View
style={{
flex: 1,
```

```
justifyContent: 'center',
alignItems: 'center',
}}>
<Text
style={{
fontSize: 21,
fontWeight: 'bold',
color: 'white',
textAlign: 'center',
fontFamily: 'Gill Sans',
}}>
{Temperature}
</Text>
</View>
</View>
</LinearGradient>
<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
style={styles.linearGradient}>
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
alignItems: 'center',
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'flex-start',
```

```
}}>
<Image
style={{width: width * 0.2, height: 50, right: 10}}
source={require('./oxidation.png')}
resizeMode={'contain'}
/>

<Text style={styles.buttonText}>Oxidation</Text>
</View>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'center',
}}>
<Text
style={{
fontSize: 21,
fontWeight: 'bold',
color: 'white',
textAlign: 'center',
fontFamily: 'Gill Sans',
}}>
{Oxidation}
</Text>
</View>
</View>
</LinearGradient>

<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
```

```
style={styles.linearGradient}>
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
alignItems: 'center',
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'flex-start',
}}>
<Image
style={{width: width * 0.2, height: 50, right: 10}}
source={require('./acc.jpeg')}
resizeMode={'contain'}
/>

<Text style={styles.buttonText}>Accelerometer</Text>
</View>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'center',
}}>
<Text
style={{
fontSize: 21,
```

```
fontWeight: 'bold',
color: 'white',
textAlign: 'center',
fontFamily: 'Gill Sans',
}}>
{Accelerometer == '0' ? 'Normal' : 'Abnormal'}
</Text>
</View>
</View>
</LinearGradient>
<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
style={styles.linearGradient}>
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
alignItems: 'center',
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'flex-start',
}}>
<Image
style={{width: width * 0.2, height: 50, right: 9}}
source={require('./ecg.png')}
resizeMode={'contain'}
/>
```

```
<Text style={styles.buttonText}>Sleep Walk</Text>
</View>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'center',
}}>
<Text
style={{
fontSize: 21,
fontWeight: 'bold',
color: 'white',
textAlign: 'center',
fontFamily: 'Gill Sans',
}}>
{Sleep_Walk == '0' ? 'Detected' : 'Not Detected'}
</Text>
</View>
</View>
</LinearGradient>

<TouchableOpacity
style={{padding: 10}}
onPress={() => {
setscreen(1);
}}>
<LinearGradient
colors={['#0a49f7', '#6dd5ed']}
style={styles.linearGradient}>
```

```
<View
style={{
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
alignItems: 'center',
}}>
<View
style={{
flex: 1,
justifyContent: 'center',
alignItems: 'center',
}}>
<Image
style={{width: width * 0.2, height: 50, right: 10}}
source={require('./current.png')}
resizeMode={'contain'}
/>

<Text style={styles.buttonText}>Location</Text>
</View>
</View>
</LinearGradient>
<View style={{flexDirection:'row',justifyContent:'space-between'}}>
<View style={styles.inputView}>
<TextInput
style={styles.inputText}
placeholder="Enter Input"
placeholderTextColor="grey"
onChangeText={(text) => setpublicdata(text)}
/>
```

```
</View>
<TouchableOpacity style={styles.loginBtn} onPress={publishData}>
<Text style={styles.loginText}>Submit</Text>
</TouchableOpacity>
</View>

</TouchableOpacity>
</ScrollView>
</View>
);
} else {
return <Address callback={() => setscreen(0)} lat={lat} lng={lon} />;
}
}
var styles = StyleSheet.create({
linearGradient: {
flex: 1,
paddingLeft: 15,
paddingRight: 15,
borderRadius: 5,
marginBottom: 20,
borderRadius: 20,
alignItems: 'center',
justifyContent: 'center',
height: height / 5 - 50,
},

buttonText: {
fontSize: 22,
fontFamily: 'Gill Sans',
textAlign: 'center',
```

```
margin: 10,
color: '#ffffff',
backgroundColor: 'transparent',
fontWeight: 'bold',
},
logo: {
fontWeight: 'bold',
fontSize: 27,
color: 'white',
marginBottom: 50,
},
buttonText2: {
fontSize: 35,
fontFamily: 'Gill Sans',
textAlign: 'center',
margin: 10,
color: '#ffffff',
backgroundColor: 'transparent',
},
inputView: {
width: '49%',
backgroundColor: '#ebf2ed',
borderRadius: 10,
height: 50,
marginBottom: 20,
justifyContent: 'center',
padding: 20,
},
inputText: {
height: 50,
color: 'black',
```

```
fontSize: 18,
},
loginBtn: {
width: '49%',
backgroundColor: '#ed3d1a',
borderRadius: 10,
height: 50,
alignItems: 'center',
justifyContent: 'center',
// marginTop: 20,
marginBottom: 10,
},
loginText: {
color: 'white',
fontWeight: 'bold',
fontSize: 20
```