**The Islamia University of Bahawalpur**

**Department of Software Engineering**

**Faculty of Computing**

****

**Software Design Documentation (SDD)**

**For**

##### **“IOT Based Real Time Patient Health Monitoring System with AI Prediction”**

**By**

**Student Name: SYED MUHAMMAD MAAZ**

**Roll No: F21BSEEN1M01128**

**Section: Morning (2M)**

**Session: Fall 2021 – 2025**

**Supervisor**

**Sunnia Ikram**

**Bachelor of Science in Software Engineering**

### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Dr. | Supervisor, **SENG- 01803** |  |
|  |  |  |  |
|  |  |  |  |

### 

**Supervised by**

**Sunnia Ikram**

**Signature** \_\_\_\_\_\_ \_\_\_\_\_

**Table of Contents**

**1. Introduction**………………………………………………………………………………….…....**5**  
*1.1. Purpose*……………………………………………………………………………………...........5 *1.2. Scope*……………………………………………………………………………………....….......5 *1.3. Overview*……………………………………………………………...…………………….….....6 *1.4. Reference Material*………………………………………………………………………….........6 *1.5. Definitions and Acronyms*………………………………………….……………………….........7

**2. System Overview**……………………………………………………………………………..........**8**  
*2.1. General System Overview*…………………………………………………………………….......8 *2.2. System Features and Objectives*…………………………………………………………..…........8 *2.3. Target Audience and Users*………………………………………………………………….........9

**3. System Architecture**………………………………………………………………………….......10  
*3.1. Overview of System Architecture*…………………………………………………...………........10 *3.2. High-Level Architecture Diagram*………………………………………………………….........11

**4. Components & Technologies**……………………………………………………......……...…...**12**  
*4.1. Hardware Components*………………………………………………………...…………..........12 *4.1.1. Node MCU*…………………………………………………………………………….…........12 *4.1.2. MAX30100 Pulse Oximeter and Heart-Rate Sensor*…………………………………….........14 *4.1.3. AD8232 ECG Sensor Module*…………………………………………………………...........16  
*4.1.4. DS18B20 Waterproof Temperature Sensor*………………………………….………...…......18 *4.1.5. LEDs*…………………………………………………………………...…………………......20

*4.1.6. LCD I2C Display*……………………………………………………...…………….……......22

*4.1.7. General PCB*…………………………………………………………….….…………..........22

**4.2. Software Tools**…………………………………………………………………………….......**22**  
*4.2.1. Arduino IDE*………………………………………………………………………...….….....23  
4.2.2. Ubidots IoT Platform…………………………………………………………...………........24

*4.3. Communication Protocols*………………………………………………………………..........25 *4.3.1. MQTT Protocol*………………………………………………………………………….......26

**5. System Design**…………………………………………………………………….……….........**27**  
*5.1. Design Definition*………………………………………………………………………….......27 *5.2. Design Characteristics*………………………………………………………….…………......27 *5.3. Alternatives for System Elements*………………………………………………………….......27

**5.4. AI Prediction Module Design**………………………………………………………..............28

5.4.1. Dataset Description and Auto-Labeling Rules……………………………………………...28

5.4.2. Preprocessing Workflow……………………………………………………………………29

5.4.3. CNN and Transformer Model Architectures……………………………………………….29

5.4.4. Evaluation Metrics and Graphs…………………………………………………………….30

5.4.5. Output Predictions and Sample Results…………………………………………………...31

**6. UML Diagrams and Workflow**…………………………………………………...……..…....**33**  
*6.1. Use Case Diagrams*……………………………………………………………...……..…......33 *6.1.1. Use Case Diagram for User Login*…………………………………………….……...….....33 *6.1.2. Use Case Diagram for System Operations*…………………………………….………........34

**6.2. Sequence Diagrams**………………………………………………………………………….……..**37**  
*6.2.1. Sequence Diagram for Login*……………………………………………………………………...37 *6.2.2. Sequence Diagram for System*….……………………………………………………………........38

*6.3. Flow Chart of System Workflow*………………………………………………………………...…..38

**7. Circuit Design**………………………………………………………………………………….……..**40**  
*7.1. Circuit Diagram*……………………………………………………………………………………..40 *7.2. Components Used in Circuit*……………………………………………………………….….……..41

**8. Block Diagram**……………………………………………………………………………….………**42**  
*8.1. Block Diagram Components*………………………………………………………………………..42 *8.2. Block Diagram*………………………………………………………………………………….…..42

**9. Additional Features and Code**………………………………………………………………………**44***9.1. Project Dashboard Overview, Features, & UI Design*……………………………………………..44 *9.2. Email Notification*……………………………………………………………………….…………..45 *9.3. System Visuals (with Predicted Results Table and Graph)* ……………………………….………..46

**10. Appendices**…………………………………………………………………………….…………....**48**  
*10.1. Glossary*…………………………………………………………………………………………...48 *10.2. References*………………………………………………………………………………..………..48 *10.3. Abbreviations*………………………………………………………………………….…………..48

**Chapter 1: Introduction**

* 1. **Purpose**

The purpose of this document is to outline the system design of the IoT-based Real-Time Patient Health Monitoring System. This system is designed to monitor and transmit critical health parameters (heart rate, body temperature, ECG, and SPO2) in real-time to healthcare providers for timely intervention. Unlike the SRS, which outlines the functional requirements and expectations of the system, this document delves into the technical design and architecture, including how the system is built and how various components work together.

**Features:**

* Integration of sensors (MAX30100 for heart rate and SPO2, DS18B20 for temperature, and AD8232 for ECG) to monitor vital health signs.
* Real-time data transmission to a cloud platform like Ubidots.
* Alert system to notify caregivers in case of abnormal readings.
* Chronic disease management for elderly patients, reducing hospital visits and improving quality of care.
  1. **Scope**

The scope of the system encompasses the following key components:

* **Health Monitoring**: The system provides continuous tracking of heart rate, ECG, body temperature, and SPO2, making it possible to monitor patients with critical health conditions.
* **Real-Time Alerts**: Immediate notifications are sent to caregivers or healthcare professionals if any vital signs fall outside normal ranges.
* **Remote Monitoring**: IoT integration ensures that patients can be monitored from any location, making the system ideal for home care, especially for elderly patients.
* **Cloud Integration**: Cloud-based platforms like Ubidots will securely store and analyze data, providing healthcare professionals with easy access from anywhere.
* **Home Care Application**: Simplified health monitoring for patients at home, reducing frequent hospital visits and increasing comfort for elderly or immobile individuals.
  1. **Overview**

This IoT-based Patient Health Monitoring System uses sensors like the MAX30100 (heart rate and SPO2), AD8232 (ECG), and DS18B20 (temperature) to continuously monitor vital health signs. The data is then sent to a cloud platform via a NodeMCU microcontroller using the MQTT protocol. This design ensures real-time health monitoring and helps healthcare professionals stay informed about their patient's condition from any location.

This system is built on an IoT architecture that integrates several components to create an effective health monitoring solution. The key components include:

1. **Sensors** (MAX30100, AD8232, DS18B20) for capturing health data.
2. **NodeMCU:** A Wi-Fi-enabled microcontroller used to collect sensor data and transmit it to the cloud.
3. **MQTT Protocol:** A lightweight messaging protocol used for efficient data transmission.
4. **Ubidots Cloud Platform:** For storing, processing, and visualizing health data.
5. **Alert System:** Automated notification system for caregivers and healthcare professionals based on abnormal readings.

The document will also describe the software layers (such as MQTT communication, data aggregation, etc.) and the hardware components involved in this system.

* 1. **Reference Material**

1. "Patient Health Monitoring System using IoT" by John Doe, published in the International Journal of Engineering Research, 2023.
2. Ubidots Documentation: <https://ubidots.com/docs>.
3. "NodeMCU and IoT Applications" by Jane Smith, IoT Innovators Press, 2022.
4. Research papers and datasheets related to the specific sensors used (MAX30100, AD8232, DS18B20).
   1. **Definitions and Acronyms**

* **IoT (Internet of Things)**: A network of interconnected devices that communicate data automatically without human intervention.
* **NodeMCU**: A microcontroller with Wi-Fi capabilities, designed for use in IoT applications.
* **ECG (Electrocardiogram)**: A medical tool used to measure the electrical activity of the heart.
* **SPO2 (Blood Oxygen Saturation)**: A measure of the oxygen level in the blood.
* **MQTT (Message Queuing Telemetry Transport)**: A lightweight protocol for real-time message transmission in IoT systems.
* **Ubidots**: A cloud platform that collects and visualizes data from IoT devices.
* **MAX30100**: A sensor for measuring heart rate and SPO2.
* **AD8232**: A sensor for monitoring ECG signals.
* **DS18B20**: A waterproof temperature sensor for tracking body temperature.
* **LED Indicators**: LEDs used to indicate health status (red for abnormal, green for normal).

**Chapter 2: System Overview**

**2.1 General System Overview**

The IoT-based Real-Time Patient Health Monitoring System is a comprehensive solution that continuously tracks the health status of patients, especially elderly or chronically ill individuals. The system leverages various sensors to monitor key health metrics such as heart rate, body temperature, ECG (electrocardiogram), and blood oxygen saturation (SPO2). These sensors send real-time data to a cloud platform (Ubidots) via the NodeMCU microcontroller, which is connected to the internet. The data is analyzed and visualized on the cloud, allowing healthcare professionals to access the patient’s health data remotely.

The system also includes a set of alert mechanisms that notify caregivers or healthcare staff immediately when abnormal readings are detected, ensuring that timely medical intervention can be provided. It integrates wireless communication protocols such as MQTT to ensure data is transmitted reliably and securely. The system is designed to be cost-effective, user-friendly, and scalable for various use cases, such as home care, chronic disease management, and remote healthcare monitoring.

**2.2 System Features and Objectives**

* **Real-Time Health Monitoring:** Continuous monitoring of heart rate, body temperature, ECG, and SPO2 for critical health tracking.
* **Cloud Integration:** The system transmits data to the cloud platform (Ubidots) using the MQTT protocol, where it is stored and analyzed in real-time for easy access by healthcare providers.
* **Alert Mechanism:** Instant alerts are generated and sent to caregivers or medical staff when any of the monitored health parameters fall outside the normal range, facilitating immediate intervention.
* **Remote Monitoring:** Enables healthcare professionals to remotely monitor patient health, which is especially useful for elderly patients or those with chronic conditions.

**2.3 Target Audience and Users**

* **Healthcare Professionals:** Doctors, nurses, and healthcare providers who need to monitor patients' health remotely. They will use the system to access real-time data, analyze trends, and receive alerts for any abnormal health readings.
* **Patients:** Primarily elderly individuals or those with chronic conditions such as heart disease, diabetes, or respiratory illnesses. These patients benefit from the ability to have their health monitored remotely, ensuring continuous care without frequent hospital visits.
* **Caregivers:** Family members or professional caregivers who need to monitor patients' health at home. The system will alert them to any significant health changes, allowing them to take appropriate action.
* **Home Care Agencies:** Organizations providing care for homebound patients can use the system to monitor their clients' health remotely, enhancing the quality of care provided in home settings.

**Chapter 3: System Architecture**

**3.1 Overview of System Architecture**

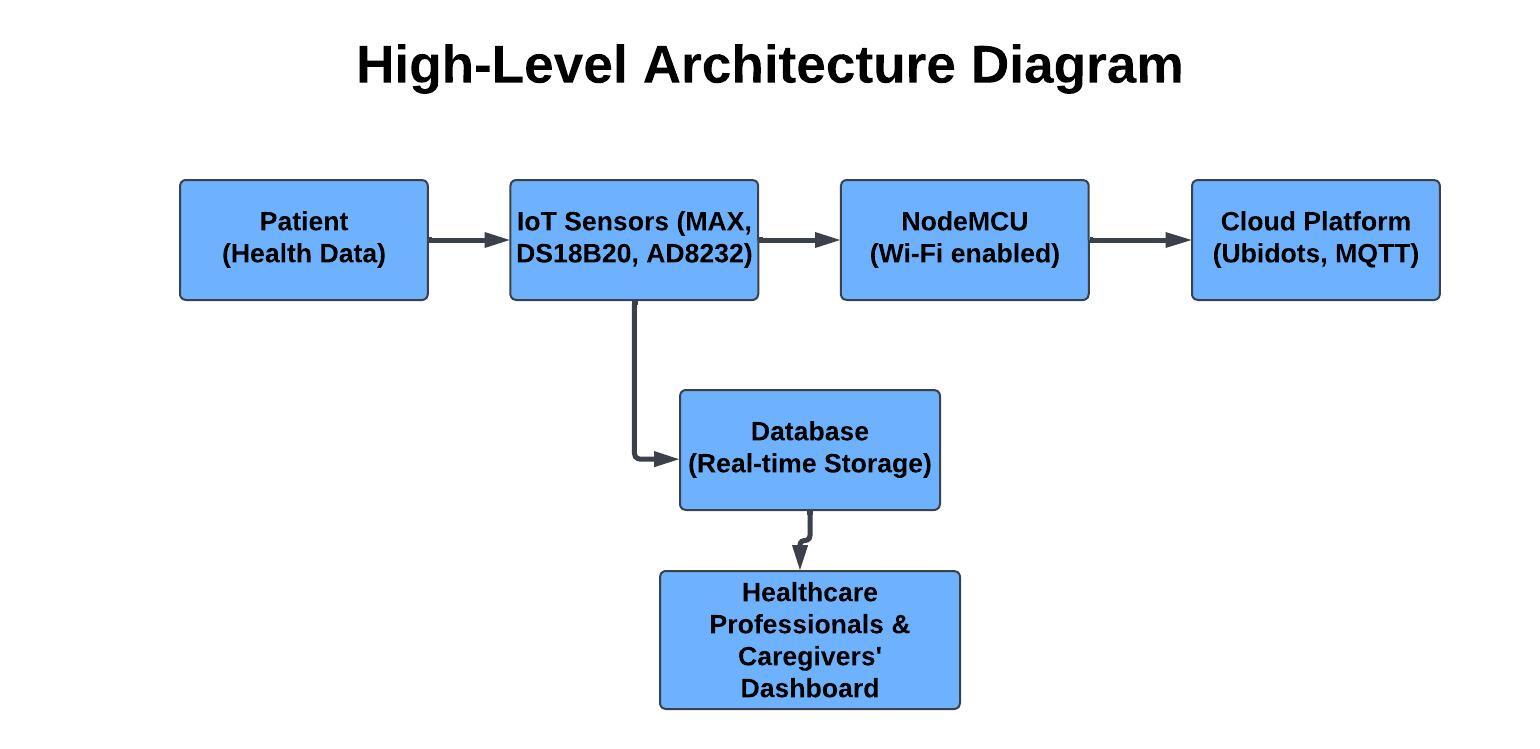
The architecture of the IoT-based Real-Time Patient Health Monitoring System is designed to ensure seamless integration between hardware components (sensors, NodeMCU), communication protocols, data transmission to the cloud, and real-time monitoring by healthcare professionals. The system is a client-server architecture where the client-side consists of sensors that monitor vital health parameters, and the server-side includes the cloud platform that collects, stores, and processes the data.

The core architecture components are as follows:

1. **IoT Sensors:** Sensors such as MAX30100 (for heart rate and SPO2), DS18B20 (for temperature), and AD8232 (for ECG) continuously collect data related to the patient's health status.
2. **Microcontroller (NodeMCU):** The NodeMCU microcontroller acts as the intermediary between the sensors and the cloud. It collects data from the sensors and transmits it wirelessly via Wi-Fi to the cloud platform using the MQTT protocol.
3. **Cloud Platform (Ubidots):** The data collected by the sensors is sent to a cloud platform like Ubidots, where it is stored, analyzed, and visualized. Healthcare professionals can access the data through the platform’s dashboard.
4. **Alert System:** When the data indicates any abnormal health condition, the alert system automatically triggers notifications (e.g., via email or SMS) to caregivers or medical staff.
5. **User Interface:** Healthcare professionals and caregivers can access the real-time data, graphs, and alerts through a user interface, typically a web-based dashboard that displays health metrics and provides control over alert settings.

**3.2 High-Level Architecture Diagram**

Graphical representation of the high-level architecture of the system:



*Figure 3.2: Health data flow from patient to cloud platform, including IoT sensors, Node*

*MCU, and real-time data storage.*

**Chapter 4: Components & Technologies**

**4.1 Hardware Components**

The hardware components used in the IoT-based Patient Health Monitoring System include a combination of sensors, microcontrollers, and other peripheral devices to capture, process, and transmit health data to the cloud platform. The system is designed to monitor vital health parameters, such as heart rate, oxygen levels, ECG signals, and temperature, and display this information in real-time.

**4.1.1 NodeMCU**

NodeMCU is a cost-effective, open-source IoT platform built around the ESP8266 Wi-Fi SoC, with later support for the ESP32 MCU. The ESP8266EX provides a complete Wi-Fi solution, capable of hosting applications or offloading Wi-Fi functions to another processor. It boots directly from external flash memory and includes an integrated cache to enhance performance. The NodeMCU ESP-12E development board can be powered using a 5V micro-USB connector or through the VIN pin. Its I/O pins operate at a maximum of 3.3V and are not 5V tolerant. For interfacing with 5V devices, level converters or resistor voltage dividers are required.

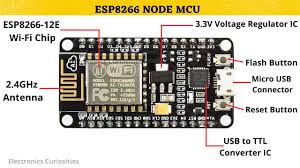
**Features:**

* **Microcontroller:** ESP8266, capable of running at 80 MHz and 160 MHz.
* **Wi-Fi Capabilities:** The NodeMCU can connect to a Wi-Fi network and send data to the cloud, making it a perfect choice for remote monitoring applications.
* **GPIO Pins:** Provides multiple General-Purpose Input/Output (GPIO) pins for connecting sensors.
* **Ease of Use**: Programmed using the Arduino IDE, which makes it user-friendly for developers.

**Function in the System:**

* The NodeMCU acts as the central controller of the system. It collects data from all the sensors (pulse oximeter, ECG, temperature sensor) and transmits the data to the Ubidots IoT platform using the MQTT protocol.
* It processes the data and sends real-time updates to the cloud for monitoring and alerting.

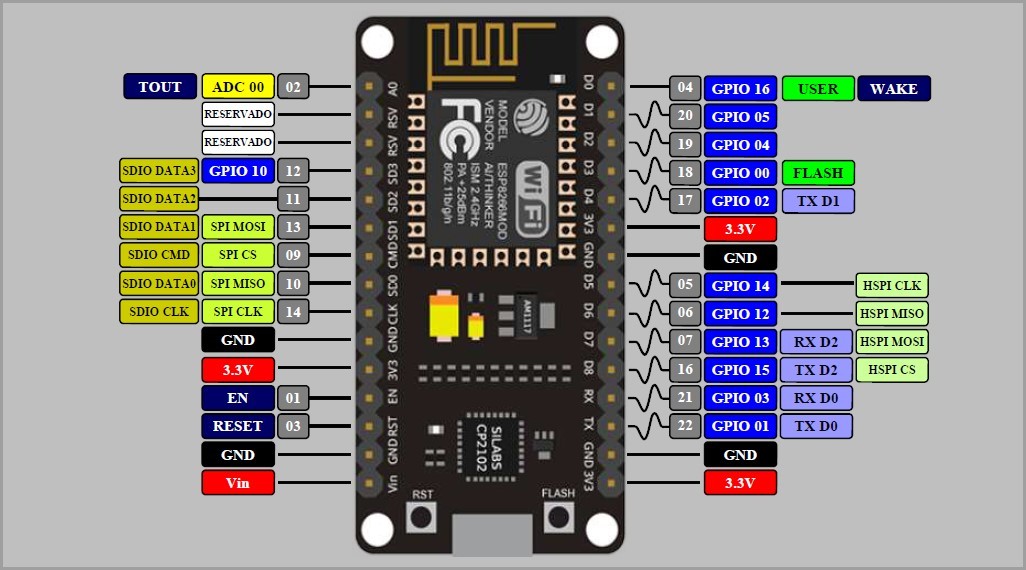
**Graphical Representation:**



*Figure 4.1: Node MCU ESP8266 development board, showcasing its design and*

*components for IoT applications.*

**Pin diagram**:



*Figure 4.2: Pin diagram of the Node MCU ESP8266, illustrating the arrangement of*

*input/output pins and their functions for IoT projects.*

**4.1.2 MAX30100 Pulse Oximeter and Heart-Rate Sensor Module**

The MAX30100 is an integrated pulse oximetry and heart rate monitor sensor solution. It combines two LEDs, a photo detector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. The MAX30100 operates from

1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

**Key Features:**

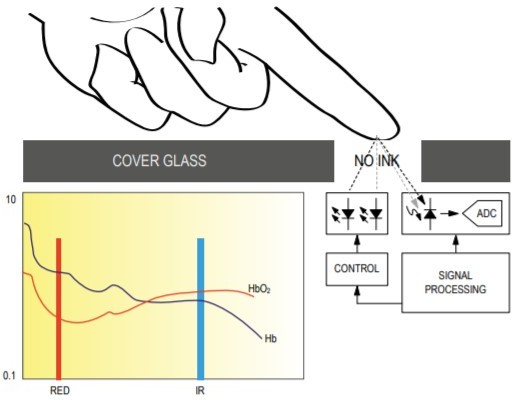
* **Pulse Oximeter:** Measures the oxygen saturation in the blood (SpO2).
* **Heart-Rate Sensor:** Measures the beats per minute (BPM) of the user’s heart rate.
* **Low Power Consumption:** Ideal for battery-powered applications.

**Function in the System:**

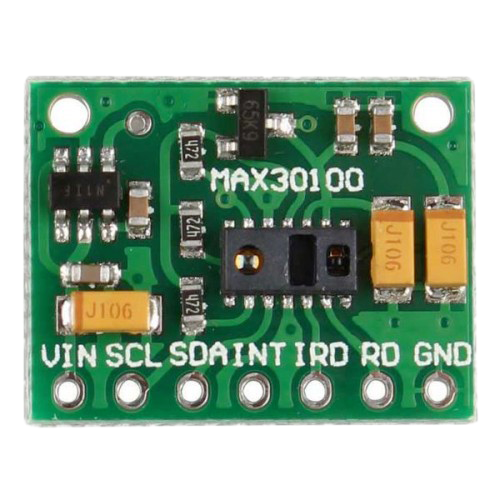
* The MAX30100 sensor is responsible for monitoring the heart rate and SpO2 levels of the patient. It sends the data to the NodeMCU, which processes and transmits it to the cloud platform.

### Applications:

* Wearable Devices
* Fitness Assistant Devices
* Medical Monitoring Devices



**Graphical Representation:**

****

**a) b)**

*Figure 4.1.2: MAX30100 Pulse Oximeter and Heart-Rate Sensor Module for*

*real-time SpO2 and heart rate monitoring*

**4.1.3 AD8232 ECG Sensor Module**

The AD8232 ECG Sensor is a specialized sensor used for capturing ECG (Electrocardiogram) signals. It is designed to measure the electrical activity of the heart, providing valuable information about heart function and potential irregularities.

The AD8232 SparkFun Single Lead Heart Rate Monitor is an affordable device used to measure the heart's electrical activity, which can be displayed as an ECG or analog signal. Since ECGs can be noisy, the AD8232 includes an integrated signal conditioning block to amplify, filter, and extract clear biopotential signals even in noisy conditions like motion or remote electrode placement.

This board provides nine connection points: SDN, LO+, LO-, OUTPUT, 3.3V, GND, and RA (Right Arm), LA (Left Arm), and RL (Right Leg) for custom sensors. It includes an LED indicator that pulses in rhythm with the heartbeat. To operate, it requires biomedical sensor pads and cables. These features make it ideal for ECG and other biopotential measurement applications.

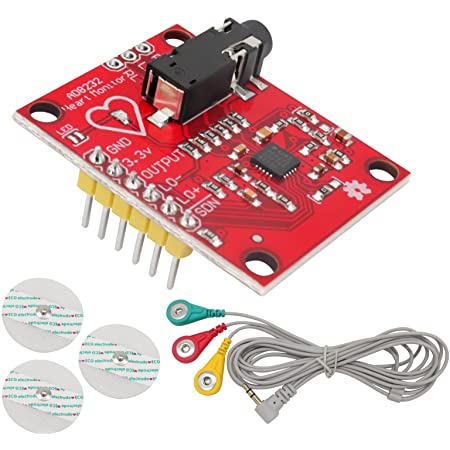
**Key Features:**

* **ECG Signal Detection:** Detects electrical signals generated by the heart.
* **Low Power Consumption:** Designed for portable and wearable devices.
* **High Accuracy:** Provides accurate and real-time ECG readings.
* **Amplification**: The sensor amplifies the ECG signal for better measurement.

**Function in the System:**

* The AD8232 sensor captures the ECG signals, which are critical for monitoring heart conditions. The data is sent to the NodeMCU, processed, and uploaded to the cloud for analysis and visualization.

**Graphical Representation:**

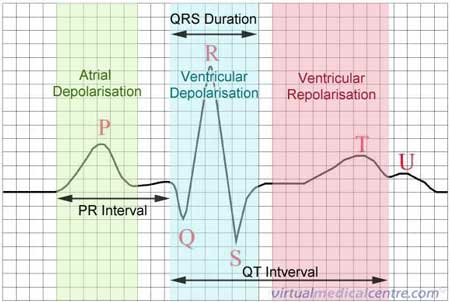
****

**a)  b)**

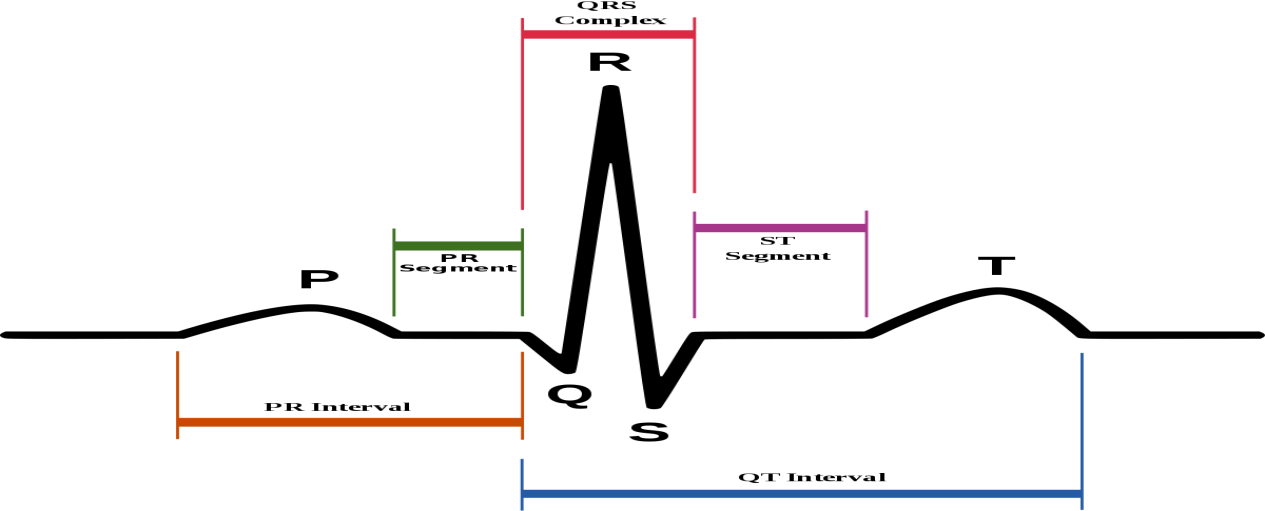
*Figure 4.1.3: AD8232 ECG Sensor Module for capturing electrocardiogram signals*

*from the patient for heart health monitoring.*

**ECG - GRAPH**



**a)**

**b)**

*Figure 4.1.3: ECG graph generated by the AD8232 ECG Sensor Module, showing*

*the heart's electrical activity.*

**4.1.4 DS18B20 Waterproof Temperature Sensor**

The DS18B20 is a digital temperature sensor known for its accuracy and waterproof design. It is commonly used for environmental and body temperature monitoring in IoT projects.

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area.

**Key Features:**

* **Digital Output:** Provides digital temperature readings, making it easy to interface with microcontrollers like the NodeMCU.
* **Waterproof Design:** Perfect for use in health applications where the sensor needs to be exposed to moisture or be in contact with the human body.
* **High Accuracy:** Temperature measurement accuracy of ±0.5°C.
* **Multiple Sensors:** It supports multiple DS18B20 sensors connected to the same data line.

**Function in the System:**

* The DS18B20 sensor measures the body temperature of the patient. It sends the temperature data to the NodeMCU, which then transmits the readings to the cloud for real-time monitoring.

### Applications:

* Thermostatic Controls
* Industrial Systems
* Consumer Products
* Thermometers
* Thermally Sensitive Systems

**Graphical Representation:**

****

*Figure 4.1.4: DS18B20 Waterproof Temperature Sensor used for measuring*

*temperature in the system.*

**4.1.5 LEDs**

LEDs (Light Emitting Diodes) are used in the system as simple indicators for the health status of the monitored parameters. These visual indicators help healthcare providers or users quickly assess the health condition.

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light- emitting phosphor on the semiconductor device.

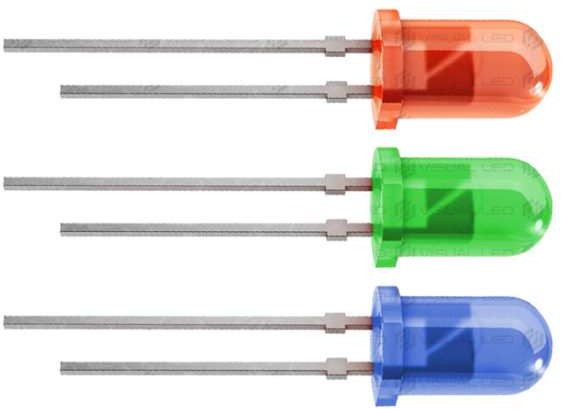
**Key Features:**

* **Red and Green LEDs:** Typically used to indicate abnormal (red) or normal (green) conditions.
* **Low Power Consumption:** LEDs consume minimal power, making them suitable for portable devices.
* **Visual Feedback:** Provides an easy-to-understand indication of system status without requiring detailed interpretation of sensor data.

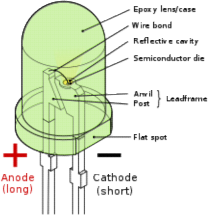
**Function in the System:**

* LEDs are used to indicate whether the monitored health parameters are within a safe range. For example:
  + **Red LED:** If the heart rate, temperature, or ECG signals are abnormal or outside the safe range.
  + **Green LED:** If all the monitored parameters are within normal range.
  + **Blue LED:** Turn ON when Wifi is connected and turn off when wifi connection is lost.

**4.1.5 Graphical Representation:**

**a)**

*Figure 4.1.6:(a) Diagram illustrating individual LED diodes used for visual feedback in the system.*



**b)**

*Figure 4.1.6:(b) LED components used in the system for visual indicators of health data or system*

*status.*

**4.1.6. LCD I2C Display:**

The **LCD I2C Display** simplifies connections using only two pins (SDA, SCL) for communication. It supports 16x2 or 20x4 layouts, adjustable contrast, and optional backlighting. Ideal for IoT projects, it displays real-time data like sensor readings or system statuses.

## 4.1.7 General Purpose Zero PCB:

*Figure 4.1.7: General-purpose zero PCB used for assembling & connecting various system*

*components.*

As Name suggests General purpose PCB Means Designer is free to make any circuit and place any component anywhere. It is used to test new ideas and in small scale electronics device production. PCB provides a means to hold all the solderable components together in one place as a single unit, but it does not provide the connection between components as provided by a specific purpose PCB using tracks. So, the users must make the connections using wires and solder joints.

General purpose PCBs have holes all over it in a grid like pattern so, one can place components anywhere as required.

**4.2 Software Tools**

The following software tools are essential for programming, configuring, and monitoring the system's hardware and cloud-based components. These tools ensure seamless integration and real-time data processing.

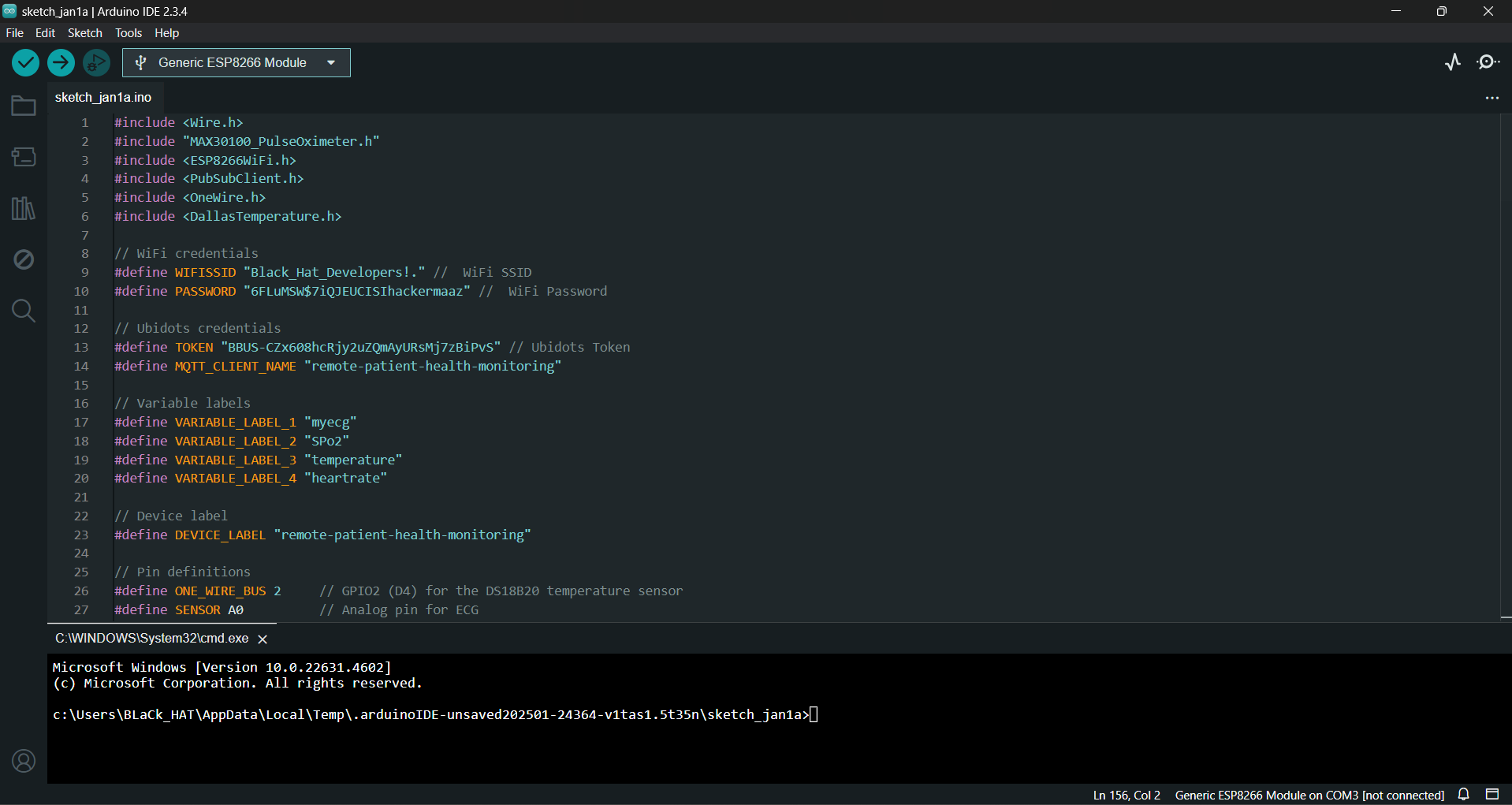
**4.2.1 Arduino IDE**

* The Arduino IDE is the primary development environment used to program the NodeMCU and other connected sensors. It is open-source and provides an easy-to-use interface for writing and uploading code.
* The Arduino IDE is used to write the code for the NodeMCU. This code collects data from the sensors, processes it, and sends it to the Ubidots platform for monitoring and visualization.

**Key Features:**

* **C/C++ Programming:** It supports C/C++ programming languages that are used to control the hardware components.
* **Library Integration**: It offers a vast library repository, allowing developers to interface easily with various sensors like MAX30100, AD8232, DS18B20, and more.
* **Cross-Platform Compatibility:** Works on Windows, macOS, and Linux.
* **Ease of Use:** Suitable for both beginners and advanced developers.

**Graphical Representation:**



*Figure 11: Arduino IDE used for programming the NodeMCU and microcontrollers.*

**4.2.2 Ubidots IoT Platform**

* Ubidots is an IoT platform that allows users to store, manage, and visualize real-time sensor data in the cloud. Ubidots collects and stores the data sent by the NodeMCU. It provides real-time visualization and alerts when health parameters go beyond safe thresholds. Healthcare providers can view the data on customized dashboards.

**Key Features:**

* **Cloud Integration:** Supports easy integration with IoT devices for sending and receiving data.
* **Data Visualization:** Provides dashboards that visualize health metrics like heart rate, SpO2, ECG, and temperature.
* **Real-Time Monitoring:** Enables continuous tracking of health parameters.
* **Alerting Mechanism:** Sends alerts for abnormal readings (e.g., high heart rate or temperature).

**Live View:**

**A screenshot of a computer

AI-generated content may be incorrect.**

*Figure 4.2.2: Live view of the Ubidots dashboard displaying real-time data from IoT*

*sensors.*

**4.3 Communication Protocols**

Effective communication protocols are essential for transmitting data from the sensors to the cloud. MQTT is used in this project to ensure efficient and reliable data transmission.

**4.3.1 MQTT Protocol**

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol used for efficient communication in IoT applications.

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol that was developed by IBM and first released in 1999. It uses the pub/sub pattern and translates messages between devices, servers, and applications.The MQTT protocol was initially created in order to link sensors on oil pipelines with communications satellites, with an emphasis on minimal battery loss and bandwidth consumption.

**Key Features:**

* **Publish/Subscribe Model:** Devices (like NodeMCU) "publish" messages to specific topics, while other devices (like Ubidots) "subscribe" to those topics to receive updates.
* **Lightweight:** MQTT messages are small in size, making it ideal for low-bandwidth applications.
* **Low Power:** MQTT operates efficiently on low-power devices, making it suitable for battery-operated IoT systems.
* **Quality of Service (QoS):** MQTT supports different levels of message delivery, ensuring reliable data transmission.

**Chapter 5: System Design**

**5. Introduction to System Design**

It can be described as the process for defining fundamentals of system such as modules, interfaces to interact, architecture, and flow of data in the system and the components on which the system is based. It is the development of system that fulfills specific requirements of an organization or business. System analysis is done to define that how these components will cooperate to accomplish the business requirements.

The system design describes the data and information about system so that implementation of system would be consistent in terms of architecture with relative to modules.

Following is the list of tasks that we performed during system design:

**5.1. Design Definition**

The IoT-based Real-Time Patient Health Monitoring System is designed with the goal of continuously monitoring key health parameters of patients, such as heart rate, body temperature, ECG (Electrocardiogram), and blood oxygen saturation (SPO2). The system is intended to provide healthcare professionals and caregivers with real-time insights into the patient's health, enabling quick intervention when needed.

The system is composed of several interconnected components that function together to monitor and transmit health data to a cloud-based platform (Ubidots) for analysis and visualization. The main components of the system include IoT sensors, NodeMCU (for data transmission), cloud platform integration, real-time alerting mechanism, and a user interface for healthcare professionals and caregivers.

**5.2. Design Characteristics**

The design characteristics of the system are essential for ensuring that the system meets the requirements of reliability, efficiency, and ease of use.

**5.3. Alternatives for System Elements**

While designing the system, there were several alternative technologies and components that could be used to achieve similar functionality. Below are the alternatives considered for each major element:

1. **Microcontroller (NodeMCU)**
   * **ESP32:** ESP32 which has more processing power and additional features (e.g., Bluetooth). It could be a good choice for more complex applications, but it might be overkill for a simple health monitoring system.
2. **Heart Rate and SPO2 Sensor (MAX30100)**
   * **MAX30102:** The MAX30102 is an upgraded version of the MAX30100 sensor with better accuracy and more features, such as better light sensitivity and fewer interferences. However, it may come at a slightly higher cost.
3. **ECG Sensor (AD8232)**
   * **ECG Sensor (MCP602):** MCP602 a different type of ECG sensor that could provide similar functionality for heart signal monitoring.
4. **Temperature Sensor (DS18B20)**
   * **LM35 Temperature Sensor:** The LM35 is another commonly used analog temperature sensor. It offers good accuracy but requires more components to interface with the microcontroller.

**5.4 AI Prediction Module Design**

The AI module adds intelligent decision-making capability to the existing IoT-based system. By analyzing real-time sensor data, it predicts whether a patient's condition is normal or abnormal, thus enhancing the effectiveness of the system and reducing manual monitoring workload.

**5.4.1 Dataset Description and Auto-Labeling Rules**

Two datasets were used in the development of the AI prediction engine:

* **Training Dataset**: Patient\_Training\_Dataset.xlsx  
  Contains fields like temperature, heart\_rate, spo₂, and a 200-sample ecg signal. This dataset was synthetically labeled and cleaned.
* **Test Dataset**: data\_device\_4\_variables\_3O2Y83P1SO.csv  
  Collected via the deployed IoT system, it was auto-labeled based on defined medical thresholds:

1. Temperature < 35.2°C or > 40.1°C → Abnormal
2. Heart Rate < 60 or > 100 bpm → Abnormal
3. SpO₂ < 93% → Abnormal

If any condition was met, the label = 1 (Abnormal); else, label = 0 (Normal).

**5.4.2 Preprocessing Workflow**

The AI pipeline required structured transformation of raw input:

1. **Auto-Labeling**: Binary labels assigned using threshold logic described above.
2. **ECG Segmentation**: Each 200-sample ECG signal was divided into chunks of **20 samples**.
3. **Feature Matrix Creation**: Each chunk was paired with static vitals (temp, HR, SpO₂), forming a **(20 × 4)** time-series matrix.
4. **Normalization**: StandardScaler from scikit-learn was used to scale the entire dataset for consistent training.

This processed data was then split into training and test sets and used for training both models.

**5.4.3 CNN and Transformer Model Architectures**

Two different deep learning models were implemented using TensorFlow and Keras:

**A. CNN (Convolutional Neural Network)**

* **Architecture**: Conv1D → MaxPooling → Flatten → Dense → Output
* **Input Shape**: (20, 4)
* **Loss**: binary\_crossentropy
* **Optimizer**: Adam
* This model showed better generalization and learning ability for small IoT health datasets.

**B. Transformer Model**

* **Architecture**:
  + Input projection to 64 dimensions
  + MultiHeadAttention (2 heads)
  + LayerNormalization and Dropout
  + Dense layers for binary classification
* Though conceptually stronger in capturing temporal dependencies, this model underperformed due to limited data and lack of diversity.

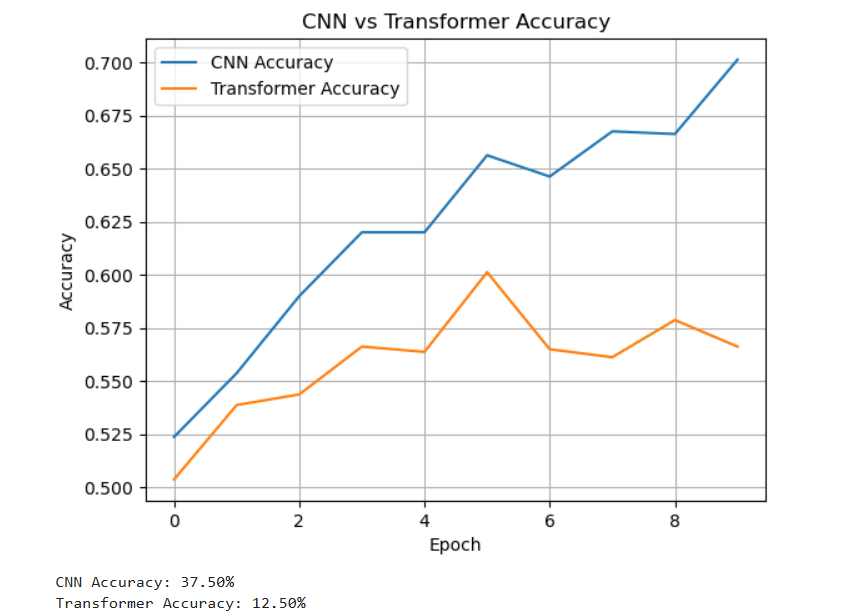
**5.4.4 Evaluation Metrics and Graphs**

The models were trained over 10 epochs. Key performance observations include:

* **CNN Accuracy**: 37.5% on test set
* **Transformer Accuracy**: 12.5% on test set

A line graph (as shown below) plots the training accuracy over epochs:

**Graph:**

****

*Figure 5.4.4: CNN vs Transformer Accuracy*

**Classification Report (Extract):**

* **CNN Recall for Abnormal**: 0.36
* **CNN Precision for Abnormal**: 0.83
* **Transformer Model**: Predicted only one class, resulting in poor metrics

**Accuracy-Report:**

**A screenshot of a computer

AI-generated content may be incorrect.**

*Figure 4.2.2: Classification Accuracy Report*

**5.4.5 Output Predictions and Sample Results**

The CNN model was used to predict the test set and results were compared with actual labels.

**Predicted Results:** Below is a sample of the prediction output

**A screenshot of a computer

AI-generated content may be incorrect.**

*Figure 5.4.5: Predicted Result Report*

**Chapter 6: UML Diagrams and Workflow**

**6.1. Use Case Diagrams**

Use Case diagrams are a type of behavioral diagram in UML (Unified Modeling Language) that provides a visual representation of the functionality provided by a system. It shows the interactions between actors (users or external systems) and the system itself, illustrating the system’s behavior under various conditions.

**6.1.1. Use Case Diagram for User Login**

The User Login Use Case diagram represents the interaction between the user and the system during the login process. It helps in visualizing the authentication process required for accessing the system.

**Actors:**

* **User (Healthcare Professional or Caregiver)**: The person who interacts with the system to log in.
* **System**: The platform that validates the user's credentials.

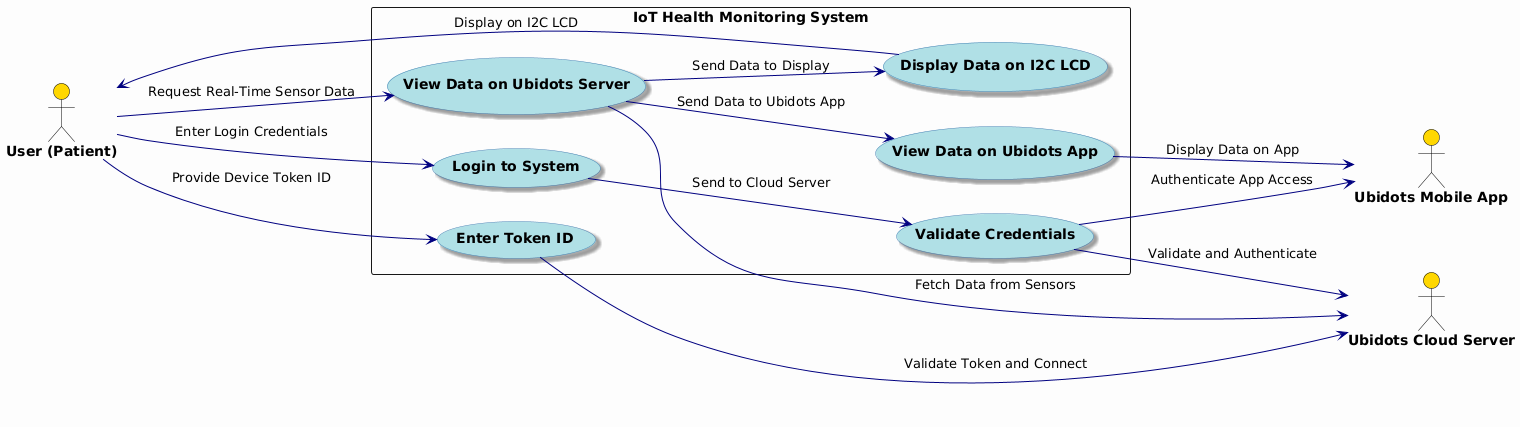
**Use Cases:**

1. **Enter Login Credentials**: The user enters their username and password.
2. **Authenticate User**: The system verifies the username and password.
3. **Access Dashboard**: Once authenticated, the user gains access to their personal dashboard.

##### 

##### **Use Case Diagram for Login:**

Figure shown below is a Use Case diagram for Login.



*Figure: Use Case Diagram of Login*

**6.1.2. Use Case Diagram for System Operations**

The System Operations Use Case diagram illustrates the various functionalities of the IoT-based Real-Time Patient Health Monitoring System. This involves patient data collection, transmission, monitoring, alerting, and displaying data on a dashboard for healthcare professionals and caregivers.

**Actors:**

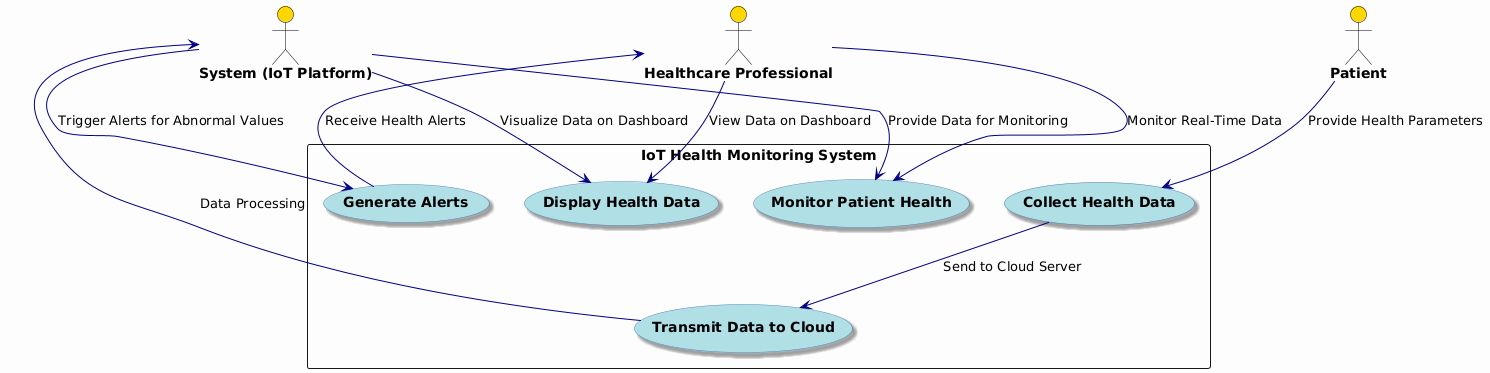
* **Patient**: The individual whose health data is being monitored.
* **Healthcare Professional**: The users who interact with the system to monitor patient health.
* **System**: The platform that manages the entire health data collection and processing.

**Use Cases:**

1. **Collect Health Data**: The system collects health data from the sensors (heart rate, ECG, temperature, SPO2).
2. **Transmit Data to Cloud**: The system transmits the collected data to the cloud platform.
3. **Monitor Patient Health**: Healthcare professionals or caregivers monitor real-time health data on the dashboard.
4. **Generate Alerts**: The system triggers an alert if any health parameters fall outside the normal range (e.g., high heart rate, low oxygen levels).
5. **Display Health Data**: The system provides real-time health data visualization through the dashboard interface.

##### **Use Case for System:**

Figure shown below is a Use Case diagram for System.



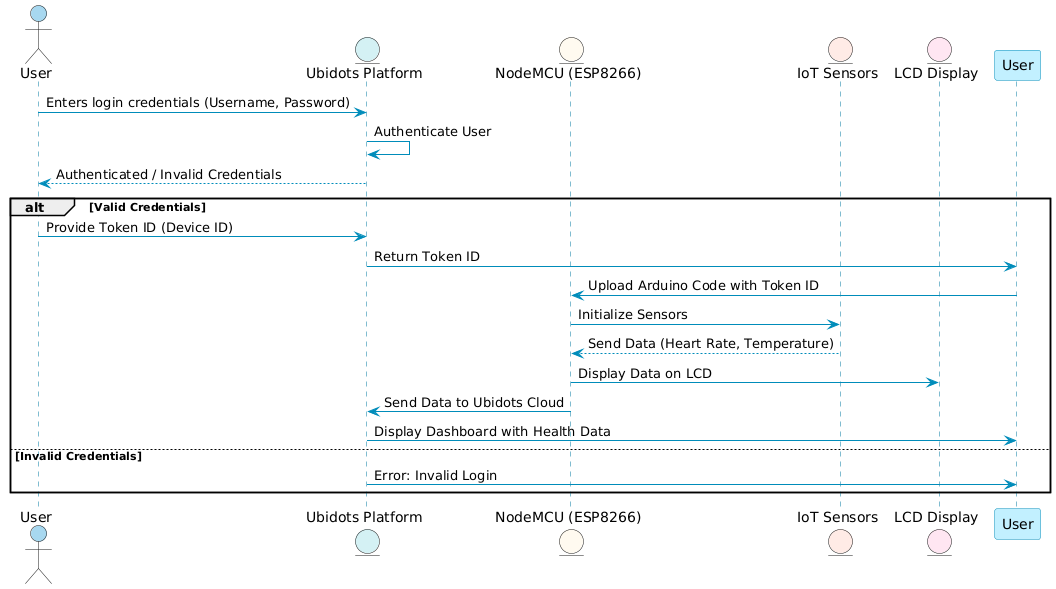
*Figure: Use Case Diagram of System*

##### **6.2 Sequence Diagram**

It is used to represent interaction between different objects in sequential manner. It shows the order in which each object will interact with other one. Event diagrams can also be used to refer sequence diagram. These diagrams are used in businesses and software developments for requirement identification to express order of objects.

##### **6.2.1 Sequence Diagram for Login**

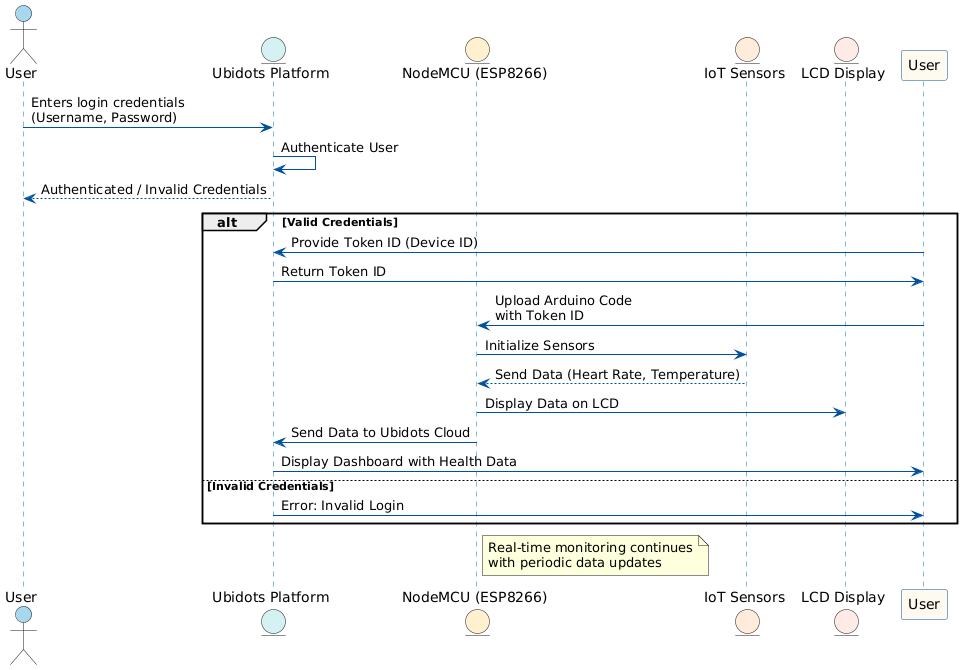
Figure shown below is a Sequence diagram for Login.



*Figure: Sequence Diagram Login*

##### **6.2.2 Sequence Diagram for System**

Figure shown below is a Sequence diagram for System.

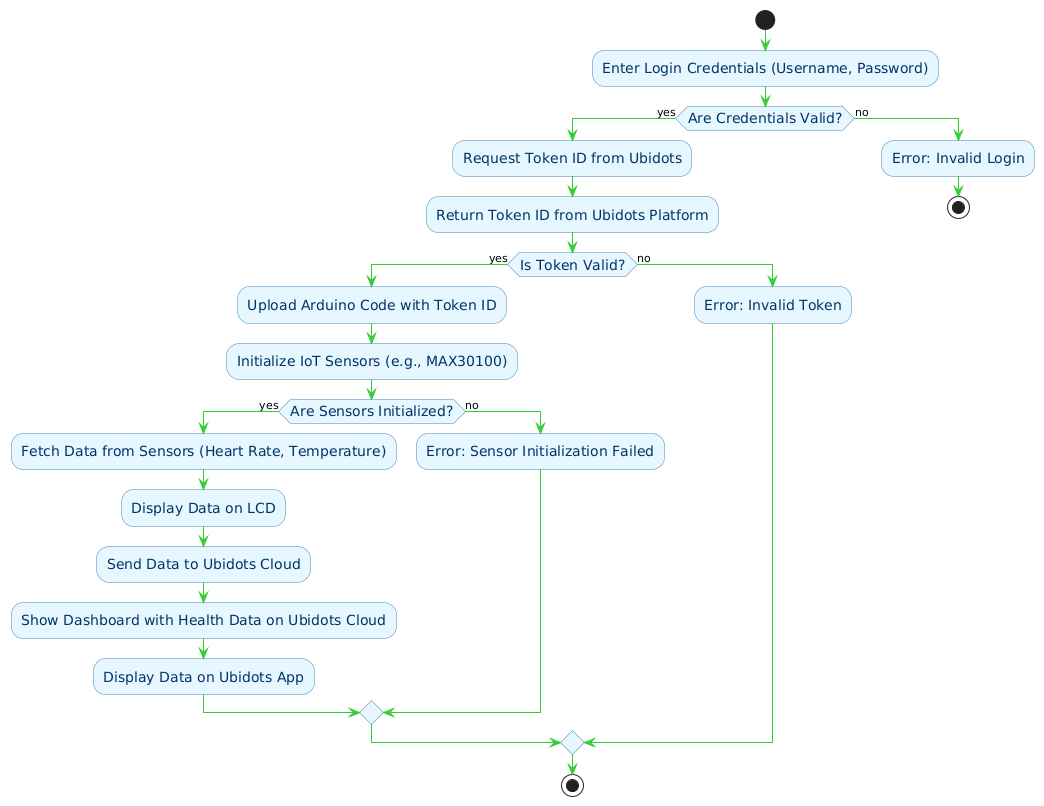


*Figure: Sequence Diagram for System*

##### **Flow Chart of the System (System Workflow)**

Figure show below is a flow chart of the system which tells us how things will work in our system. It is basically a flow of the login screen and if login credentials are not correct then it will show error and will not show the the main screen off Android Application.

**Flow Chart**



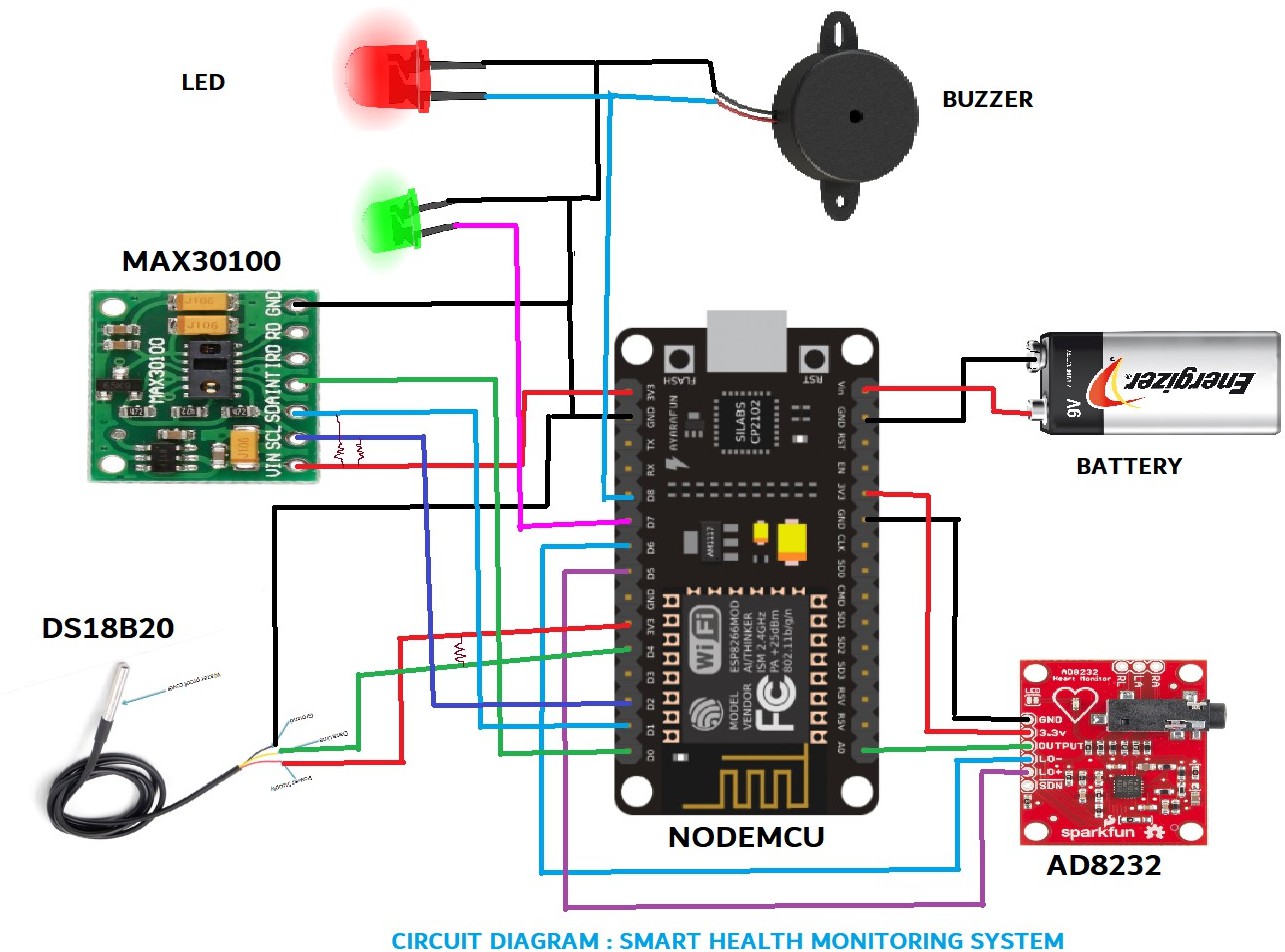
*Figure: Flow Chart of the system*

**Chapter 7: Circuit Design**

**7. Circuit Design**

The circuit design of the IoT-based Smart Health Monitoring System integrates the hardware components into a functional system. The components are connected as per the specified configuration to ensure seamless data collection and transmission.

**7.1 Circuit Diagram:**

****

*Figure 7.1: Circuit diagram of the real-time IoT-based patient health monitoring*

*system*

**7.2 Components in the Circuit:**

* **NodeMCU (ESP8266):** Acts as the central controller and is connected to all sensors via GPIO pins.
* **MAX30100 Pulse Oximeter & Heart-Rate Sensor:** The sensor is connected to the NodeMCU’s I2C pins (SCL, SDA).
* **AD8232 ECG Sensor:** The ECG sensor’s output is connected to an analog input pin (A0) of the NodeMCU.
* **DS18B20 Waterproof Temperature Sensor:** This sensor uses a 1-wire interface, connected to a digital GPIO pin of the NodeMCU.
* **LEDs:** Used to indicate the status (red for abnormal, green for normal) and connected to digital pins on the NodeMCU.

**Chapter 8: Block Diagram**

**8. Block Diagram**

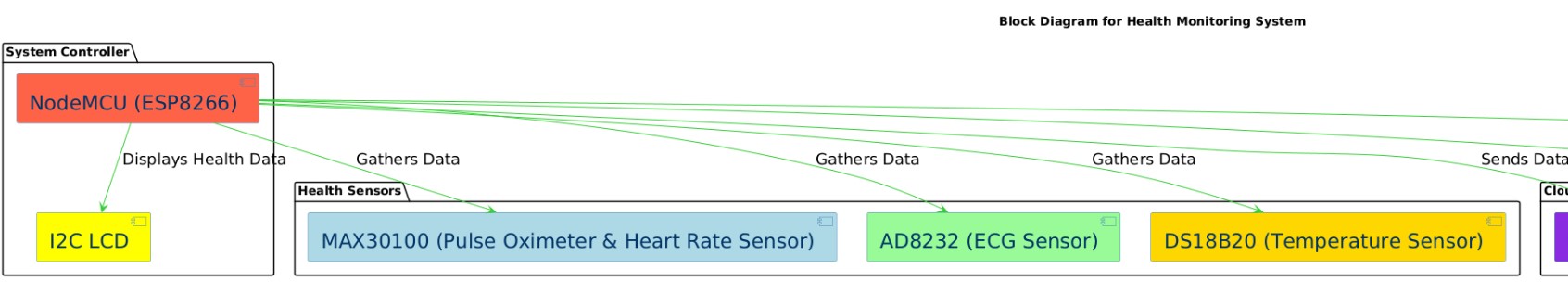
The block diagram represents the overall architecture of the system, showing the various subsystems and how they interact with each other.

**8.1 Block Diagram Components:**

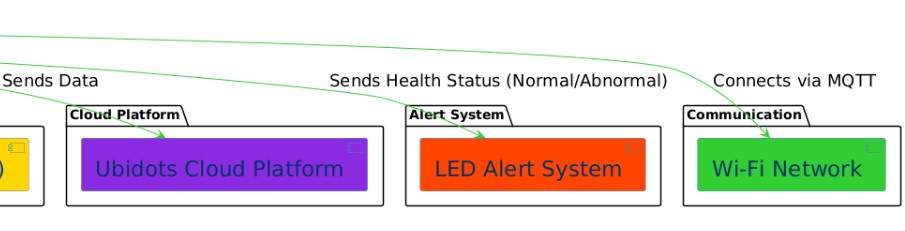
* **Health Sensors:**
  + MAX30100 (Pulse Oximeter & Heart Rate Sensor)
  + AD8232 (ECG Sensor)
  + DS18B20 (Temperature Sensor)
* **NodeMCU (ESP8266):** Acts as the central controller that gathers data from the sensors.
* **Cloud Platform (Ubidots):** Data is transmitted to the cloud for visualization and real-time monitoring.
* **Alert System:** Based on the readings from the sensors, LEDs will indicate abnormal (red) or normal (green) health parameters.
* **Wi-Fi Network:** Connects NodeMCU to the cloud via MQTT protocol.

##### **8.2: Block Diagram:**

The flow in the block diagram will show how the NodeMCU gathers data from sensors, processes it, and transmits it to Ubidots for monitoring. It will also show how the health status is displayed via LEDs and how data is communicated via the Wi-Fi network.

**a)**

**b)**

 *Figure 8.2: Block diagram of the IoT-based patient health monitoring system, showing the*

*interaction between patient sensors, Node MCU, cloud platform, and the healthcare dashboard.*

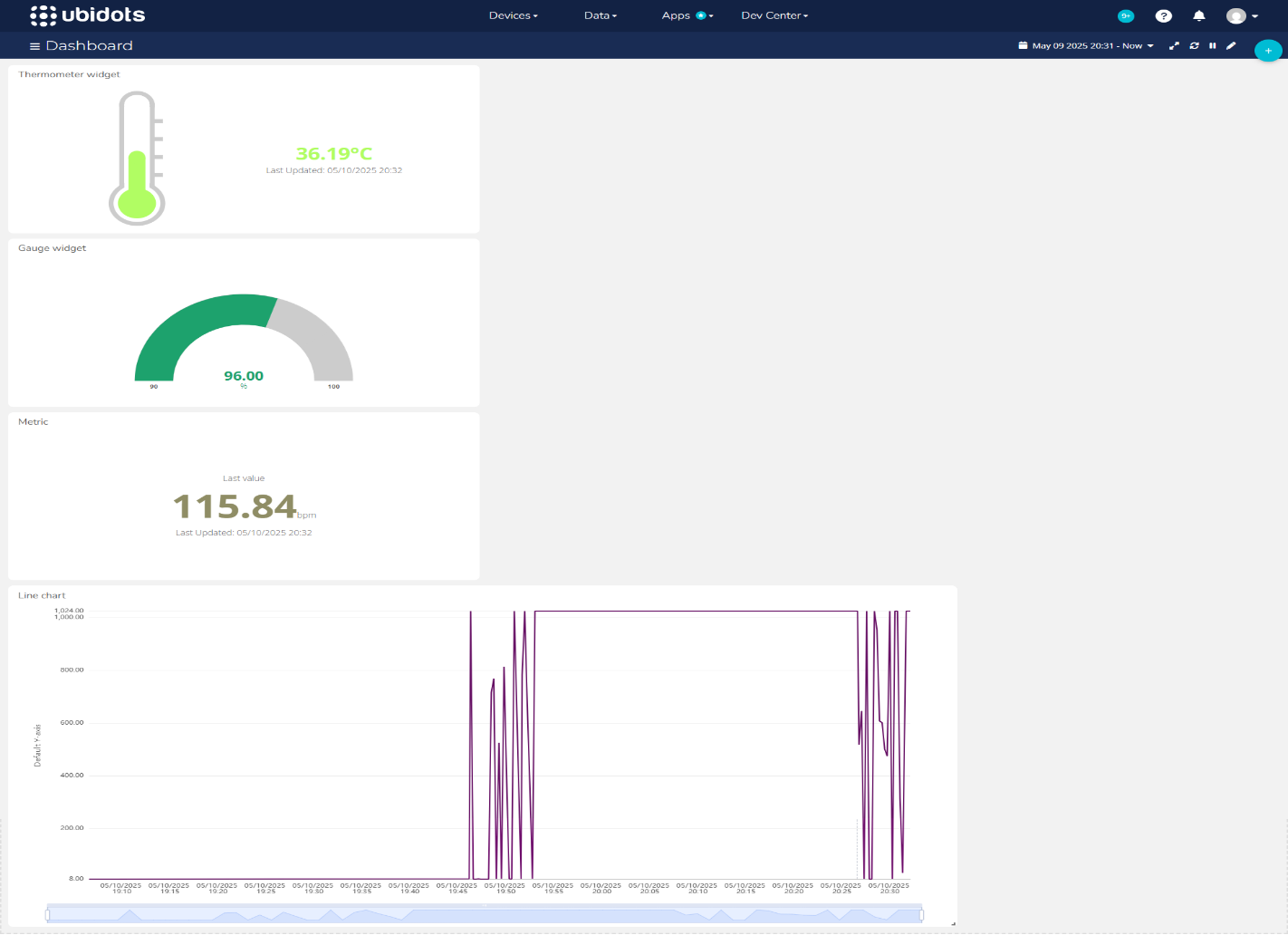
**Chapter 9: Additional Features & Code Snippets**

**9.1 Project Dashboard**

The Project Dashboard serves as the control and monitoring interface for the health monitoring system. It displays real-time data from various sensors and allows users to manage and track health metrics easily. Key features include:

* **Real-Time Data:** Heart rate, oxygen levels, temperature, and ECG are displayed in a user-friendly format.
* **Alerts & Notifications:** Alerts are triggered when any health parameter exceeds a defined threshold, providing immediate notifications.
* **Graphical Visualization:** Graphs and charts visualize the trends of key health metrics over time for better decision-making.

**Dashboard Screen:**

****

*Figure 9.1: Screenshot of the live dashboard view displaying real-time health data from IoT sensors, including heart rate, oxygen levels, temperature, and ECG, for patient monitoring.*

**9.2 Email Notification**

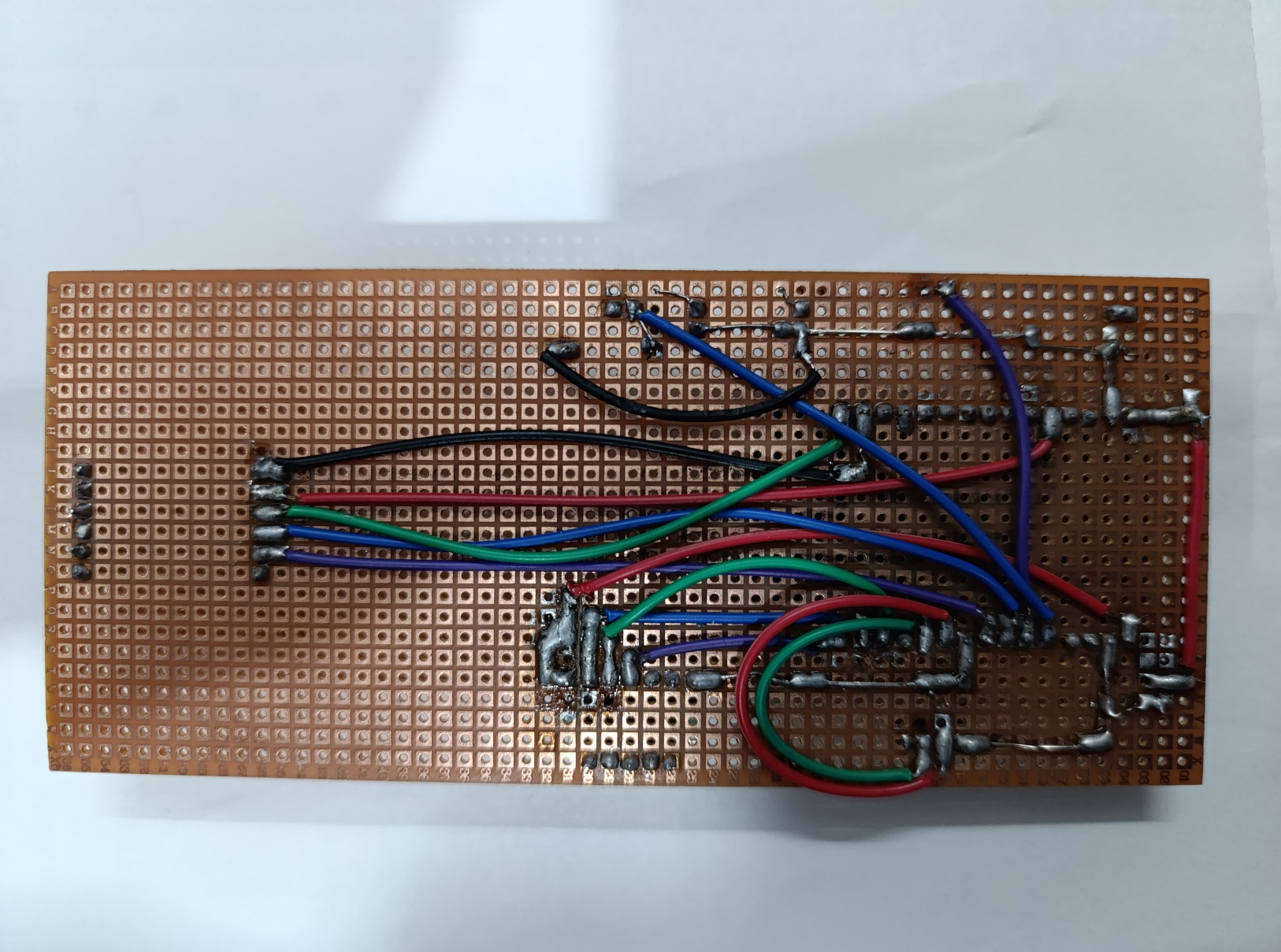
The Email Notification feature sends alerts via email when sensor readings breach critical thresholds.

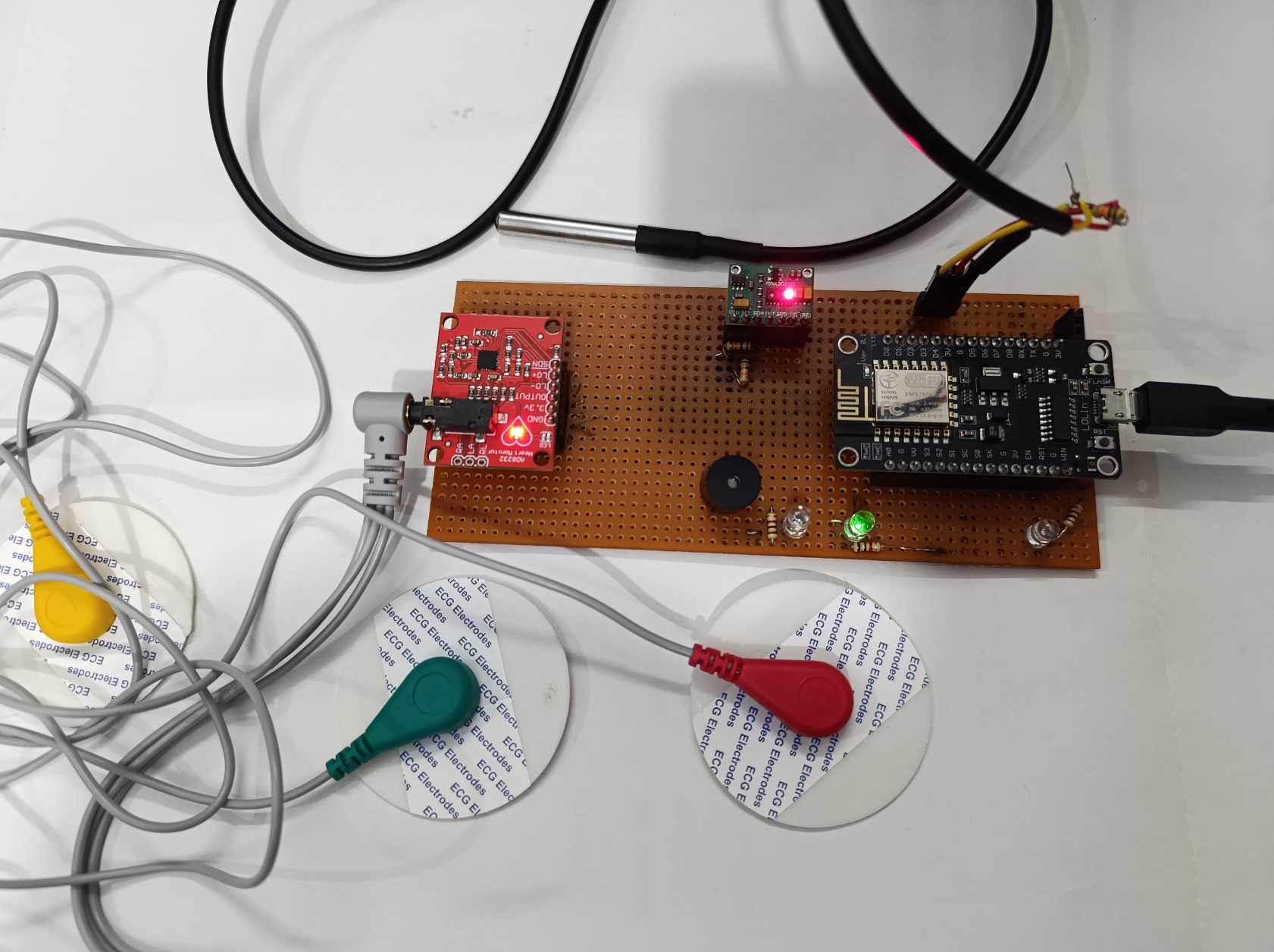
**A screenshot of a computer

AI-generated content may be incorrect.**

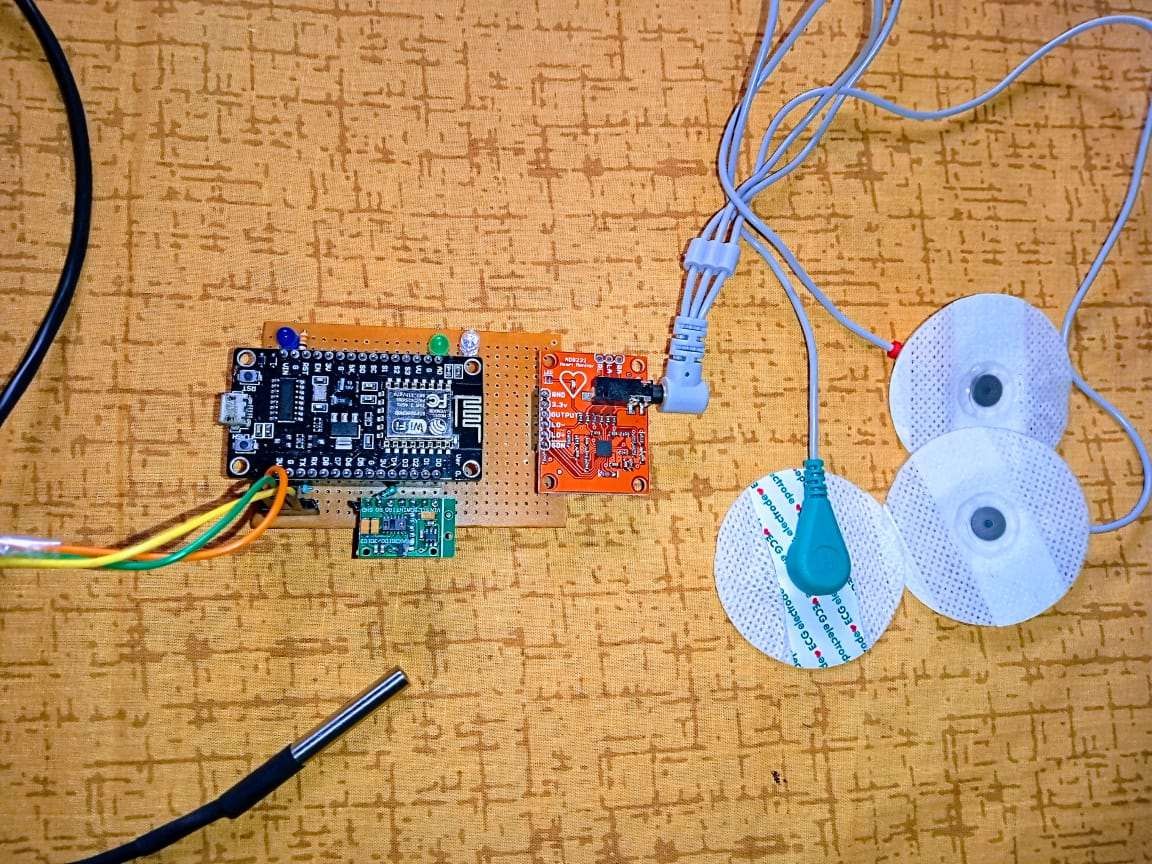
*Figure 9.2: Workflow of the email notification system for patient health updates.*

**9.3 IoT System Visuals / Project Visuals**

****

****

*Figure 9.3: Visuals of the IoT system setup, including sensor modules and hardware components used in the patient health monitoring system.*



A circuit board with wires and a display

AI-generated content may be incorrect.

**Chapter 10: Appendices**

**10.1 Glossary**

* **IoT (Internet of Things):** A network of physical objects embedded with sensors and connectivity to exchange data.
* **NodeMCU:** An open-source IoT platform based on the ESP8266 Wi-Fi SoC.
* **MQTT (Message Queuing Telemetry Transport):** A lightweight messaging protocol for small sensors and mobile devices.
* **Ubidots:** A cloud-based IoT platform for data visualization and remote monitoring.
* **MAX30100:** An integrated pulse oximeter and heart-rate sensor module.
* **AD8232:** An ECG sensor module for monitoring heart electrical activity.
* **DS18B20:** A digital temperature sensor with a waterproof design for precise temperature readings.
* **PCB (Printed Circuit Board):** A board used to mechanically support and electrically connect electronic components.

**10.2 References**

1. Ubidots IoT Documentation: <https://ubidots.com>
2. NodeMCU Documentation: <https://nodemcu.readthedocs.io>
3. Arduino IDE User Guide: <https://www.arduino.cc>
4. MAX30100 Datasheet: <https://datasheets.maximintegrated.com>
5. AD8232 ECG Sensor Datasheet: <https://www.analog.com>
6. DS18B20 Temperature Sensor Datasheet: <https://www.maximintegrated.com>

**10.3 Abbreviations**

* **IoT**: Internet of Things
* **MQTT**: Message Queuing Telemetry Transport
* **PCB**: Printed Circuit Board
* **Wi-Fi**: Wireless Fidelity
* **IDE**: Integrated Development Environment
* **ECG**: Electrocardiogram
* **LED (Diodes)**: Light Emitting Diode
* **LCD**: Liquid Crystal Display. (I2C)