An

**Industrial Oriented Mini Project Report**

on

**SMART BLOOD BANK MANAGEMENT SYSTEM USING IOT**

Submitted to

**Jawaharlal Nehru Technological University, Hyderabad**

*For the partial fulfillment of requirements for the award of the degree in*

**BACHELOR OF TECHNOLOGY**

in

## COMPUTER SCIENCE AND ENGINEERING

Submitted by

|  |  |
| --- | --- |
| **KONUKATI RAMYA** | **(21271A0533)** |
| **KHAMMAM ROSHINI** | **(21271A0536)** |
| **PANYALA SAI VINOD** | **(21271A0544)** |
| **KOMATIREDDY SAI RAHUL REDDY** | **(21271A0539)** |
|  |  |

Under the Esteemed guidance of

**G.SRIKANTH**

Assistant Professor

Dept of CSE



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**JYOTHISHMATHI INSTITUTE OF TECHNOLOGY AND SCIENCE**

**(Autonomous, NBA (CSE, ECE, EEE) and NAAC ‘A’ Grade)**

**(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)**

**Nustulapur, Karimnagar 505481, Telangana, India**

**2024-2025**

**CERTIFICATE**

This is to certify that the Industrial Oriented Mini Project Report entitled “**SMART BLOOD BANK MANAGEMENT SYSTEM USING IOT”** is being submitted by KONUKATI RAMYA (21271A0533), KHAMMAM ROSHINI (21271A0536), PANYALA SAI VINOD (21271A0544), KOMATIREDDY SAI RAHUL REDDY (21271A0539) in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering to the Jyothishmathi Institute of Technology & Science, Karimnagar, during academic year 2024-2025, is a Bonafide work carried out by them under my guidance and supervision.

The results presented in this Project Work have been verified and are found to be satisfactory. The results embodied in this Project Work have not been submitted to any other University for the award of any other degree or diploma.

**Project Guide Head of the Department**

## G.SRIKANTH Dr . R. JEGADEESAN

**Assistant Professor Professor & HOD**

**Dept of CSE Dept. of CSE**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our advisor**, G. SRIKANTH**, Assistant Professor, whose knowledge and guidance has motivated us to achieve goals we never thought possible. The time we have spent working under his supervision has truly been a pleasure.

The experience from this kind of work is great and will be useful to us in future. We thank

**Dr.R.JEGADEESAN**, Professor & HOD CSE Dept for his effort, kind cooperation, guidance

and encouraging us to do this work and also for providing the facilities to carry out this work.

It is a great pleasure to convey our thanks to **Dr. T. ANIL KUMAR**, Principal, Jyothishmathi Institute of Technology & Science and the College Management for permitting us to undertake this project and providing excellent facilities to carry out our project work.

We thank all the **faculty** members of the Department of Computer Science & Engineering for sharing their valuable knowledge with us We extend our thanks to the **Technical Staff** of the department for their valuable suggestions to technical problems Finally Special thanks to our parents for their support and encouragement throughout our life and this course Thanks to all our friends and well-wishers for their constant support.

# 

# 

**DECLARATION**

We hereby declare that the work which is being presented in this dissertation the entitled,

**“SMART BLOOD BANK MANAGEMENT SYSTEM USING IOT”** submitted towards the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science And Engineering, Jyothishmathi Institute of Technology & Science**, Karimnagar is an authentic record of our own work carried out under the supervision of **G.SRIKANTH**, Assistant Professor, Department of CSE, Jyothishmathi Institute of Technology and Science, Karimnagar.

To the best of our knowledge and belief, this project bears no resemblance with any report submitted to JNTUH or any other University for the award of any degree or diploma.

|  |  |
| --- | --- |
| **KONUKATI RAMYA** | **(21271A0533)** |
| **KHAMMAM ROSHINI** | **(21271A0536)** |
| **PANYALA SAI VINOD** | **(21271A0544)** |
| **KOMATIREDDY SAI RAHUL REDDY** | **(21271A0539)** |

Date:

Place: Karimnagar

# ABSTRACT

Blood bank is a place where blood is collected and stored before it is used for transfusions. The main objectives of the blood banks are providing blood to the patients with minimal blood transfusion error. The blood is very important medical supplies so it should be managed well. It consists of a number of manual steps, therefore it will become difficult for the blood banks to provide a high level of accuracy, reliability, automation in blood storage and transfusion process. The system proposed is divided into three segments, the first segment consists Temperature sensor, IR sensor nodes which is installed in rack of blood bank, and the GSM Module for sending request of blood to the donors and blood banks all these are interfaced with Arduino Mega. Second segment consists of wi-fi module for data transfer to the server and third segment is displaying the status of available blood stock. All the real time status relates to the available blood stock of the blood bank is are displayed on web page, so that the blood seeker can get the blood from their nearest blood bank.

ⅰ

# TABLE OF CONTENTS

**CHAPTER NO TITLE PAGE NO**

**ABSTRACT i**

**LIST OF FIGURES v**

**LIST OF ABBREVIATIONS vi**

**1 INTRODUCTION 1**

1.1 Project Overview 1

1.2 Project Purpose 2

1.3 Project Scope 2 1.4 Project Features3

**2 LITERATURE REVIEW 4 3 EXISTING & PROPOSED SYSTEM 6**

3.1 Existing System 6

3.2 Existing System Disadvantages 6

3.3 Problem Statement 7

3.4 Proposed System 7

3.5 Proposed System Advantages 8

**4 SYSTEM REQUIREMENTS 10**  ii

4.1 Software Requirements 10 4.2 Hardware Requirements10

**5 PROJECT DESCRIPTION 11**

5.1 Modules11

**6 SYSTEM DESIGN 14**

6.1 System Architecture 14

6.2 Data Flow Design 16

6.3 UML Design16

6.3.1 Use Case Diagram 18

6.3.2 Class Diagram 19

6.3.3 Activity Diagram 20 6.3.4 Sequence Diagram 21

**7 SOFTWARE SPECIFICATIONS 22**

7.1 python 22

**8 IMPLEMENTATION 23**

8.1 Installations of Arduino 23  8.2 ThingSpeak setup 28

8.3 Code 31

**9 SOFTWARE TESTING 34**

9.1 Unit Testing iii 34

9.2 Integration Testing 34

9.3 System Testing 35

9.4 Acceptance Testing 35

9.5 Performance Testing 36

9.6 Security Testing 36

9.7 Compatibility Testing 37

**10 RESULTS & DESCRIPTION 38**

**11 CONCLUSION 40**

**12 FUTURE SCOPE 41**

**REFERENCES 42**

iv

**LIST OF FIGURES**

**FIGURE NO NAME OF THE FIGURE PAGE NO**

6.1 System Architecture 15

6.2 Data Flow Diagram 16

6.3.1 Use Case Diagram 18

6.3.2 Class Diagram 19

6.3.3 Activity Diagram 20

6.3.4 Sequence Diagram 21

8.1.1 USB Port 23

8.1.2 Downloading of Arduino IDE Software 24

8.1.3 Launching Arduino 25

8.1.4 Creating New Project 25

8.1.5 Existing Project Example 26

8.1.6 Selecting Arduino 26

8.1.7 Selecting Port 27

8.1.8 Uploading the Program 27

8.2.1 Creating Account 28

8.2.2 New Channel Creation 29

8.2.3 Naming channel and fields 29

8.2.4 Saved Channel Status 40

8.2.5 API keys 40

v

**LIST OF ABBREVIATIONS**

**ABBREVIATION DESCRIPTION**

DHT11 Digital Humidity and Temperature

HX711 Analog to Digital converter

ESP32 System on chip

USB Universal Serial Bus

vi

**CHAPTER 1**

## INTRODUCTION

### PROJECT OVERVIEW

The Smart Blood Bank Management System using IoT is an innovative solution designed to streamline and enhance the efficiency of blood donation, storage, and distribution processes. This system addresses the critical challenges faced by traditional blood bank systems, such as mismanagement, wastage of blood, and delays in emergency situations. By leveraging the power of the Internet of Things (IoT), the system provides real-time monitoring of blood storage conditions, including temperature, humidity, and stock levels, ensuring optimal storage and reducing spoilage.

A centralized platform connects donors, blood banks, and hospitals, enabling seamless communication and collaboration. Donors can register and use a mobile or web application to track their donation history, receive alerts for urgent blood requirements, and schedule future donations. Blood banks benefit from IoT-enabled smart storage systems that monitor and maintain ideal conditions for blood preservation. Automated alerts notify stakeholders about critical situations, such as low stock levels, impending expirations, or storage anomalies.

The system also integrates an optimized distribution mechanism, utilizing location tracking to ensure timely delivery of blood to hospitals during emergencies. Additionally, it incorporates data analytics to forecast blood demand trends, optimize inventory management, and identify donor demographics. This predictive capability helps in planning blood drives and maintaining an adequate supply of all blood types.

The Smart Blood Bank Management System improves the overall efficiency, reliability, and safety of blood management. It minimizes wastage, enhances emergency response capabilities, and ensures that the right blood reaches the right place at the right time, saving lives and fostering a more connected and responsive healthcare ecosystem.

* 1. **PROJECT PURPOSE**

The purpose of a Smart Blood Bank Management System using IoT is to revolutionize the way blood banks operate by leveraging advanced technologies to address critical challenges in blood donation and distribution. Blood is a vital resource in healthcare, and its effective management is essential to save lives during emergencies and routine medical procedures. Traditional blood bank systems often face issues such as inefficient inventory management, lack of real-time data, and risk of blood wastage due to improper storage or expiration. The introduction of IoT-based solutions aims to overcome these limitations by providing a smarter, more integrated approach

Using IoT devices and cloud-based platforms, stakeholders can access real-time information on blood inventory levels, types, and availability. This not only improves the efficiency of blood distribution but also ensures that hospitals can quickly procure the required blood during emergencies. Additionally, the system can notify donors about nearby blood donation drives and encourage them to donate based on the specific needs of their community, thus maintaining a steady supply of blood.

In summary, the Smart Blood Bank Management System using IOT transforms traditional blood bank operations by introducing automation, real-time monitoring, and enhanced connectivity. It minimizes wastage, ensures the timely availability of blood, and ultimately contributes to saving lives while optimizing healthcare resources. This innovative solution addresses the growing demand for efficient and reliable blood management in the modern healthcare ecosystem

**1.3 PROJECT SCOPE**

A **Smart Blood Bank Management System** using IOT improves the way blood banks operate by making them more efficient and responsive. With IOT sensors, it can monitor things like temperature and humidity in blood storage, ensuring the blood stays safe. It also tracks blood types, quantities, and expiry dates in real-time, reducing waste. The system connects blood banks, hospitals, and collection centers , so blood can be quickly and safely transported where it's needed.

The system also helps manage donors by sending smart notifications about donation opportunities and using wearable devices to monitor donor health

Overall, this IOT-powered system ensures timely access to safe blood, lowers costs, and has a positive impact on public health.

### 1.4 PROJECT FEATURES

A **Smart Blood Bank Management System** using IOT comes with several features that enhance efficiency, reliability, and safety in blood collection, storage, and distribution. Some key features include:

* **Real-time Monitoring**: IOT sensors continuously monitor critical factors such as temperature, humidity, and light conditions in blood storage units to ensure that blood is stored under optimal conditions. If any parameter deviates from the required range, alerts are sent to staff for immediate action.
* **Inventory Management**: The system keeps track of blood types, quantities, and expiry dates in real time. This helps prevent stockouts and reduces wastage by ensuring that older blood units are used first.
* **Automated Alerts and Notifications**: Automated notifications are sent to donors for important events like blood expiry, stock shortages, or the need for blood donations. This improves overall communication and ensures timely interventions.
* **Emergency Response System**: In critical situations, the system can quickly identify available blood units based on blood type and urgency, facilitating rapid matching and ensuring timely delivery to hospitals or patients in need.

## 

## CHAPTER 2

## LITERATURE REVIEW

## 

## The demand for effective blood bank management systems has grown with the increasing need for timely, safe, and efficient blood donation and transfusion processes. Smart blood bank systems utilize modern technology to optimize these processes, addressing traditional challenges such as shortages, wastage, and logistical inefficiencies. This literature review explores the development and applications of smart blood bank systems, focusing on their components, benefits, challenges, and emerging trends.

## This paper Smart Blood Bank Management publised by Kanna Anil Kumar , gavini chaitanya Sai Kumar , Ajay varma Urge To Implement IoT For Monitoring and Preventing Blood Bank System Crisis Integrated Blood Bank Solution: The approach combines web-based technologies and IoT to enhance coordination in blood donation, storage, and distribution, addressing inefficiencies in current systems .Real-Time Monitoring and Management: It utilizes GSM modules and Arduino microcontrollers for real-time tracking of blood inventory levels, temperature, and location, ensuring optimal storage conditions.User-Friendly Online Platform:

## .This paper Block chain-Based Management of Blood Donation is published byIsmail, R.D.; Hussein, H.A.Salih, M.M.Ahmed, M. A .Hameed, Q.A.; Omar, M.B. The Use of Web Technology and IoT to Contribute to the Management of Blood Banks in Developing Countries. Appl. Syst. Published: 6 September 2022 Blockchain-Based Solution: The paper proposes a private Ethereum blockchain solution for blood donation management, addressing shortcomings in traceability, transparency, and security while eliminating the risks of centralization.Decentralized Storage: Non-critical and large data are stored off-chain using the InterPlanetary File System (IPFS), enhancing efficiency and scalability while maintaining privacy and security.Comprehensive System Design: The paper includes detailed system architecture, sequence diagrams, and algorithms, alongside performance evaluations and a publicly available smart contract code on GitHub, demonstrating the solution's effectiveness and security.

## This paper Use of Web Technology and IoT to Contribute to the Management of Blood Banks in Developing Countries published by Diana Hawashin,Dunia Amin j.Mahboobeh,khaled salah ,Raja JayaRaman. Integrated System Development: The study introduces a combined application using Web technology for data management and IoT sensors for real-time temperature monitoring of blood inventory, targeting management issues in blood banks, especially in developing countries. Addressing Key Challenges: It addresses critical challenges such as inadequate blood supply, unstable electricity, and reliance on error-prone manual systems, enhancing the reliability of blood bank operations . Positive User Evaluation: Feedback from 22 blood bank professionals revealed high satisfaction and effectiveness, indicating that the system significantly improves blood bank management and has strong potential for wider adoption in healthcare.

## 

## CHAPTER 3

## EXISTING & PROPOSED SYSTEM

### 3.1 EXISTING SYSTEM

### 

In the present scenario, many people were facing a lot of problems in getting blood for patients at right time.

The needy does not know where to get the blood or how to get access to the a required quantity of blood quickly in an emergency.

Traditional blood donation systems often rely on manual processes for donor registration, blood collection, testing, and distribution. While these systems have been effective to some extent, they face challenges such as inefficiency, lack of real-time data tracking, and limited accessibility for donors and blood bank

### 3.2 EXISTING SYSTEM DISADVANTAGES

* **Manual Data Collection:** Manual data collection in a blood bank involves recording and managing essential information about donors, recipients, blood units, and processes manually, without relying on automated systems.
* **Inefficient Search and Retrieval:** Inefficient search and retrieval are common challenges in manual data management systems. These inefficiencies arise due to various factors, such as unorganized record-keeping, lack of indexing, on outdated methods.
* **Limited Scalability:** The manual nature of the existing system imposes limitations on scalability, making it challenging to efficiently manage large volumes of event-related data as the scale of events increases.
* **Risk of Information Loss:** Manual processes increase the risk of information loss or miscommunication, particularly during data collection and dissemination.
* **Potential for Data Entry Errors:** Data entry errors are a common issue in manual systems,where accuracy is critical. These errors can occur during the recording, transcribing, or transferring of information and may lead to significant operational, safety, or compliance problems..
* **Reduced Accessibility:** Reduced accessibility in a blood bank's data management system—whether manual or digital.This can lead to delays in operations, inefficiencies, and even life-threatening situations in emergencies.

### 3.3 PROBLEM STATEMENT

The blood bank faces significant challenges due to its reliance on manual data collection and management systems. The current approach involves paper-based records and fragmented storage, which leads to inefficiencies in search and retrieval of critical information. Staff frequently encounter delays in locating donor details, blood inventory data, and test results, particularly during emergencies. Additionally, manual processes are prone to data entry errors, such as transcription mistakes, incomplete records, and duplication, which compromise the accuracy and reliability of information. These errors increase the risk of issuing incompatible blood to recipients and disrupt operational workflows. Accessibility to data is also a concern, as physical storage systems and inconsistent filing practices limit timely access to essential records, hindering decision-making and service delivery during critical situations. Furthermore, the inability to provide accurate and organized documentation poses compliance risks during audits or inspections, potentially resulting in penalties and reputational damage. These issues underscore the urgent need for improved data management practices to enhance efficiency, ensure patient safety, and maintain regulatory compliance.

**3.4 PROPOSED SYSTEM**

The system proposed is divided into three segments.The first segment consists Temperature sensor, IR sensor nodes which is installed in rack of blood bank, and the GSM Module for sending request of blood to the donors and blood banks all these are interfaced with Arduino Mega. Second segment consists of wi-fi module for data transfer to the server.And third segment is displaying the status of available blood stock. All the real time status relates to the available blood stock of the blood bank is are displayed on web page, so that the blood seeker can get the blood from their nearest blood bank.

### 3.5 PROPOSED SYSTEM ADVANTAGES

### ****Real-Time Monitoring and Tracking:****

IOT-enabled devices can monitor and track blood storage conditions, such as temperature, humidity, and expiration dates, ensuring compliance with storage standards.Continuous monitoring reduces the risk of spoilage or damage to blood units.

Efficient Inventory Management**:**

IOT sensors can track blood unit availability in real-time, providing accurate inventory updates to minimize shortages or overstocking.Automatic alerts can notify staff of critical low stock levels or upcoming expiry dates.

### ****Improved Accessibility:****

### A smart system can integrate with a mobile or web application to provide real-time information on blood availability across multiple locations.

**Enhanced Transparency and Traceability**

Each blood unit can be tagged with IOT-enabled tracking devices (e.g., RFID or QR codes) to monitor its journey from donation to transfusion. This improves trust and reduces the risk of mismatched or counterfeit blood units.

**Automated Alerts and Notifications**

Notifications can be sent to hospital staff, donors, or administrators regarding critical updates, such as inventory depletion, donation reminders, or abnormalities in storage conditions.Timely alerts prevent wastage and ensure prompt action.

### ****Donor Engagement and Data Collection****

IOT devices can streamline the donor process by automating health checks (e.g., blood pressure, hemoglobin levels) before donation.

Data from donors can be collected and securely stored to improve future services and encourage regular donations through personalized reminders.

### ****Cost-Effectiveness****

Automated processes reduce manual effort and human errors, cutting down operational costs. Prevention of spoilage through real-time monitoring minimizes financial losses.

### ****Enhanced Emergency Response****

Smart systems can quickly identify and allocate required blood units during emergencies by analyzing real-time data. Faster response times save lives in critical situations.

### ****Data Analysis and Predictive Insights****

IOT systems collect data over time, which can be analyzed to predict trends, such as demand for specific blood types or seasonal donation patterns. Predictive Analytics aids in better planning and resource allocation.

### ****Improved Patient Safety****

IOT devices ensure proper storage and handling, reducing the risk of using compromised blood units.Detailed tracking minimizes the chance of transfusion-related errors.

By integrating IOT into blood bank systems, the healthcare sector can significantly enhance the efficiency, reliability, and accessibility of blood storage and distribution, ultimately saving more lives.

**CHAPTER 4**

### SYSTEM REQUIREMENTS

#### 4.1 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are the software requirements

* **Programming Language :** Python
* **Operating System :** Windows 11
* Arduino Ide and ThingSpeak

#### 4.2 HARDWARE REQUIREMENTS

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

### Arduino

### Hx711 Sensor

* DHT11 Sensor
* ESP01( Wifi module)

### CHAPTER 5

### PROJECT DESCRIPTION

The **Smart Blood Bank System using IOT** is an innovative solution designed to modernize traditional blood bank operations by leveraging Internet of Things (IOT) technology. This system integrates IoT-enabled sensors and devices to monitor blood storage conditions, such as temperature and humidity, ensuring that all units are stored under optimal conditions. It provides real-time inventory updates, allowing blood banks and hospitals to efficiently manage stock levels and reduce wastage. The system enhances transparency and traceability by using technologies to track blood units throughout their lifecycle, from donation to transfusion. A centralized database connects blood banks, hospitals, and donors, enabling seamless data sharing and faster responses during emergencies. Through mobile and web applications, users can access real-time information on blood availability and locate the nearest blood bank with the required type. The system also supports donor engagement by automating health checks and sending personalized donation reminders. By incorporating predictive analytics, the smart blood bank system can forecast demand patterns and optimize resource allocation. This project aims to minimize wastage, improve accessibility, and ensure the timely availability of safe blood, ultimately contributing to a more efficient and reliable healthcare ecosystem.

**5.1 MODULES**

1. **WiFi Connection Module**

* **Purpose:** Establishes a connection between the ESP8266 and the WiFi network.
* **Functions:**
* connectToWiFi(): Configures the ESP8266 module and connects to the specified WiFi network.
* **Key Components:** WiFi credentials (ssid and password).
* AT commands for ESP8266 configuration (AT+CWMODE, AT+CWJAP).

**2.Sensor Initialization Module**

* **Purpose**: Initializes the connected sensors (DHT11 and HX711 modules).
* **Functions :** Initializes the DHT11 sensor for temperature and humidity readings.
* Configures two HX711 modules for weight measurement with individual data pins**.**
* **Key Components:**DHT11 initialization (dht.begin()).
* HX711 initialization (scale1.begin(), scale2.begin()).

**3.Sensor Reading Module**

* **Purpose:** Reads data from the sensors and provides it for processing or transmission.
* **Functions:**
  + Reads temperature and humidity using the DHT11 sensor.
  + Measures weights using two HX711 modules**.**
* **Key Components:**
  + dht.readTemperature() and dht.readHumidity() for DHT11.
  + scale1.get\_units() and scale2.get\_units() for HX711.

**4.Data Transmission Module**

* **Purpose:** Sends collected data to the ThingSpeak IoT platform via ESP8266.
* **Functions:**
  + Prepares the data payload in ThingSpeak format.
  + Establishes a TCP connection to the ThingSpeak server.

### Sends the data using HTTP GET requests.

* + Closes the TCP connection after the data is sent.
* **Key Components:**
  + String formatting for ThingSpeak data (field1, field2, etc.).
  + AT commands for TCP communication (AT+CIPSTART, AT+CIPSEND, AT+CIPCLOSE).

**5.Feedback/Logging Module**

* **Purpose:** Provides real-time feedback to the user via the Serial Monitor.
* **Functions:**
  + Logs WiFi connection status.
  + Displays sensor readings (weights, temperature, humidity).
  + Confirms data transmission to ThingSpeak**.**
* **Key Components:**
  + Serial.print() and Serial.println() for detailed logs**.**

**6.Timing and Delay Module**

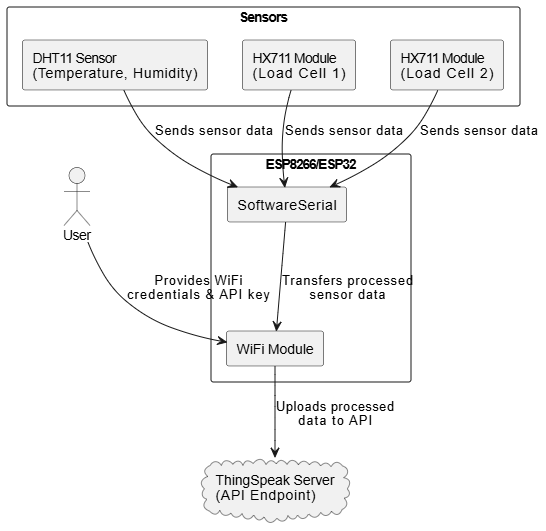
* **Purpose:** Manages delays and ensures compliance with ThingSpeak's 15-second update interval**.**
* **Functions:**
  + Maintains a delay between consecutive data transmissions to ThingSpeak.
* **Key Components:**
  + delay(15000) to adhere to ThingSpeak's update restrictions.

**CHAPTER 6**

**SYSTEM DESIGN**

#### 6.1 SYSTEM ARCHITECTURE

The Smart Blood Bank Management System leverages the power of IoT to streamline and monitor blood bank operations. At the core of the system, sensors such as the DHT11, HX711 (Load Cell 1), and HX711 (Load Cell 2) play vital roles. The DHT11 sensor monitors the temperature and humidity inside the blood storage units, ensuring that the blood is kept at the required temperature range (usually 2°C to 6°C) to prevent spoilage. The HX711 load cells are used to measure the weight of the blood bags, which helps in tracking the quantity of blood available for donation or transfusion. These sensors are connected to an ESP microcontroller (ESP8266/ESP32), which acts as the central hub. The ESP device collects real-time data from the sensors and processes it. It then transmits this data over Wi-Fi to a cloud-based platform, such as ThingSpeak . ThingSpeak provides an intuitive way to visualize the data, offering dashboards for authorized personnel to monitor key metrics like temperature, humidity, and blood stock levels. Alerts can be set up within ThingSpeak to notify users of abnormal conditions, such as temperature fluctuations or low stock levels. This architecture ensures that the blood bank’s conditions are continuously monitored, and the blood inventory is efficiently managed, making the system an essential tool for maintaining a safe and well-organized blood donation and storage process.



**Figure 6.1** : System Architecture

**6.2 DATA FLOW DESIGN**

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system or process. It is a modeling technique that shows how data moves through processes and how it is stored, transformed, or exchanged within a system. DFDs are often used in system analysis and design to visualize and understand the data processing and flow of information in a structured manner.

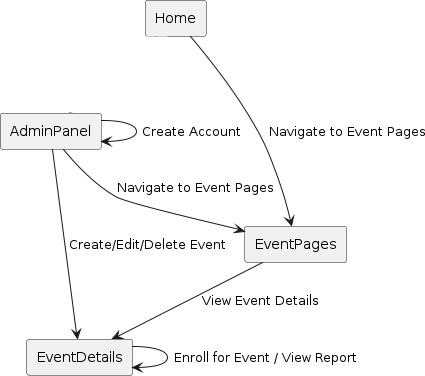


Figure 6.2 : Data Flow Diagram

#### 6.3 UML DESIGN

Unified Modeling Language (UML) is a standardized modeling language in the field of software engineering. It provides a way to visualize a system's design through a set of diagrams. UML diagrams help software developers and system architects communicate and understand the structure and behavior of a system.In UML, the diagrams can be broadly categorized into two main types: structural diagrams and behavioral diagrams.

##### 1. Structural Diagram

Structural diagrams in UML focus on representing the static structure of a system. They showcase the components that make up the system and their relationships.The following are different structural diagrams:

1. Class diagram
2. Object diagram
3. Component diagram
4. Deployment diagram

##### 2. Behavioral Diagram

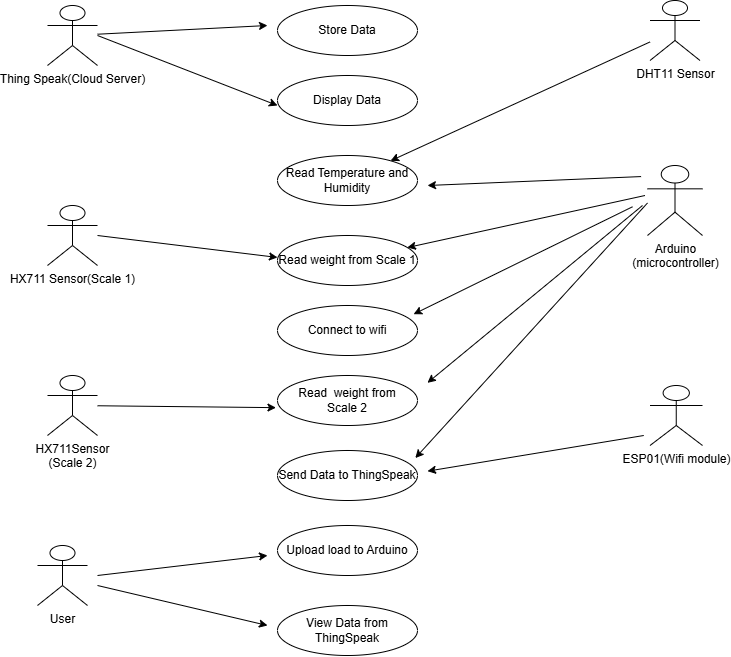
Behavioral diagrams primarily capture the dynamic facet of the system. Dynamic aspects can be more delineated because of the dynamical or moving elements of a system.UML has the subsequent5 varieties of behavioral diagrams.

They are

1. Use case diagram
2. Sequence diagram
3. Collaboration diagram
4. State chart diagram
5. Activity diagram

**6.3.1 Use Case Diagram**

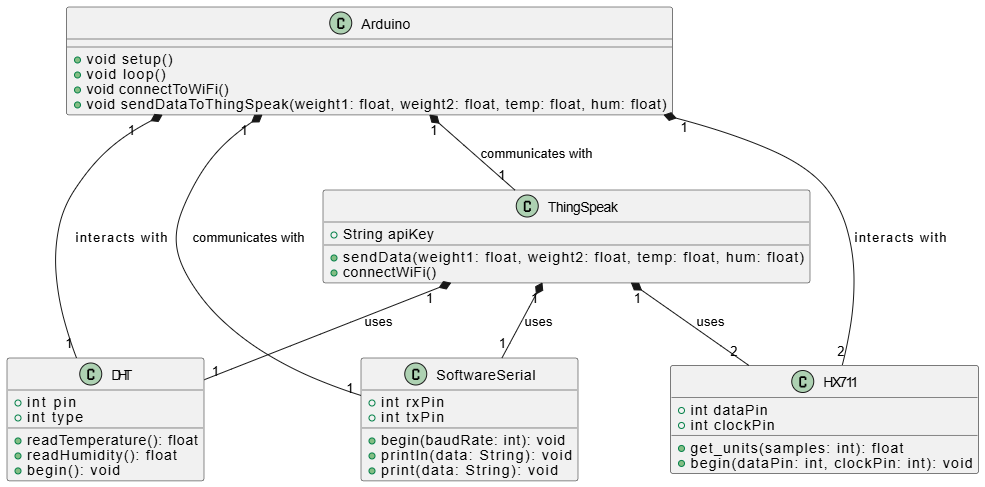
Use Case Diagrams in UML describe interactions between a system and external entities known as actors. Use cases represent specific functionalities or scenarios that the system provides to its users.



**Figure 6.3.1** : Use Case Diagram

**6.3.2 Class Diagram**

The Class Diagram in UML illustrates the static structure of a system, detailing classes, attributes, methods, and their relationships. Rectangles represent classes, and lines indicate associations, dependencies, and inheritances.

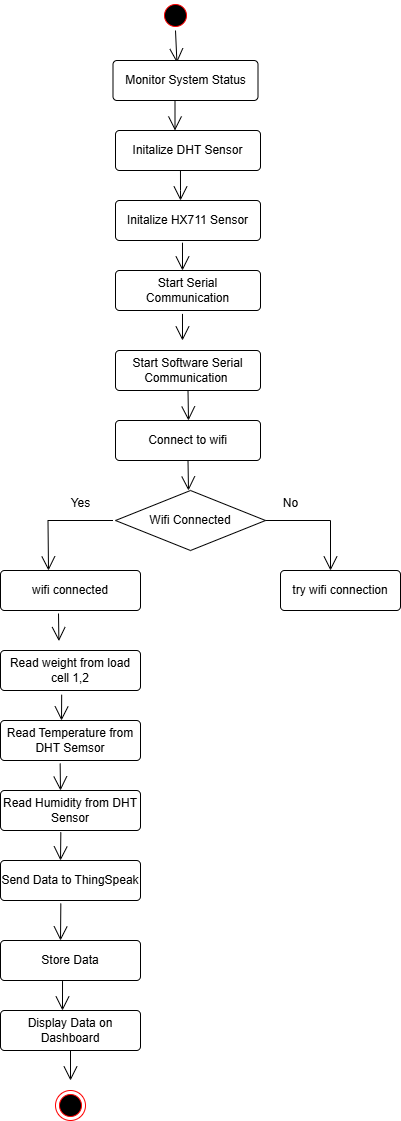


**Figure 6.3.2** : Class Diagram

**6.3.3 Activity Diagram**

##### 

Activity Diagrams represent the flow of activities within a process or workflow. They focus on actions, decisions, and control flows, providing a high-level view of the dynamic aspects of a system.

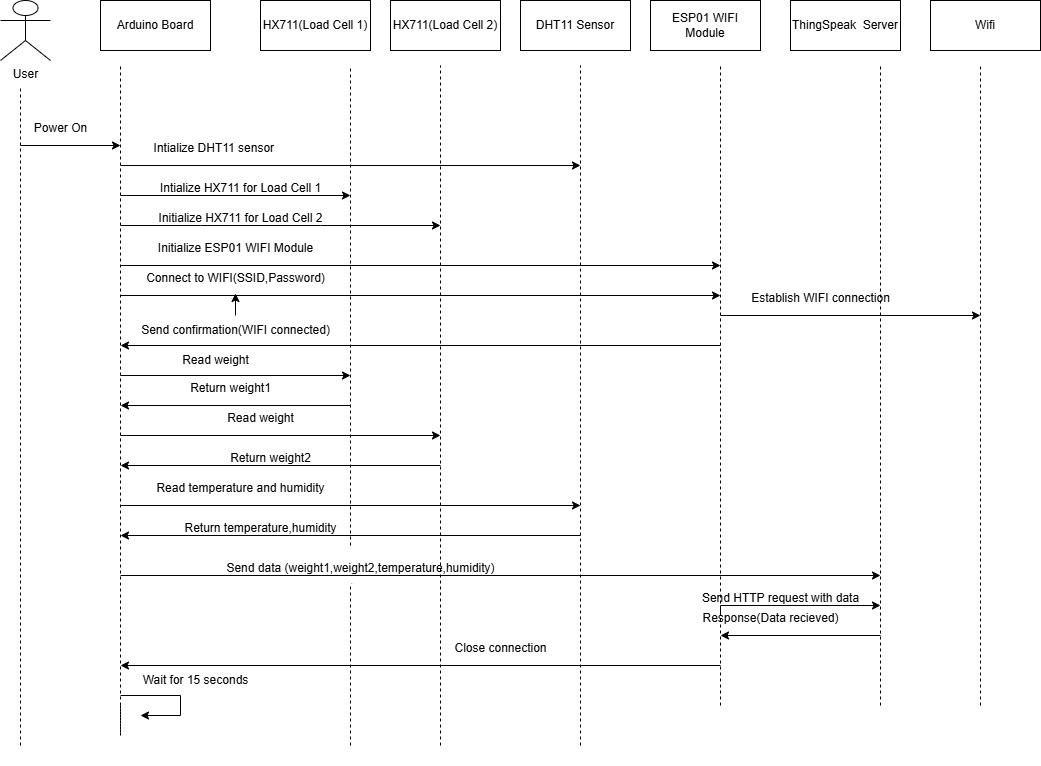


**Figure 6.3.3 :** Activity Diagram

**6.3.4 Sequence Diagram**

A sequence diagram merely depicts interaction between objects in an exceedingly serial order i.e., the order during which these interactions turn up. We will conjointly use terms event diagrams or event eventualities to consult with a sequence diagram. These diagrams are widely employed by businessmen and software package developers to document and perceive needs for brand new and existing systems.

.



**Figure 6.3.4** : Sequence Diagram

**CHAPTER 7**

### SOFTWARE SPECIFICATIONS

#### 7.1 PYTHON

Python, a versatile and widely-used programming language, served as the primary language for the development of this project.Python was chosen as the cornerstone of the college event management system website for its distinctive characteristics and key features.

##### Key Features

* **Versatility:** Python's versatility allowed for seamless integration across different components of the website, from backend development to data processing and automation scripts.
* **Readability:** Python's clean and readable syntax facilitated rapid development and collaboration among team members, enhancing the maintainability of the codebase.
* **Extensive Ecosystem:** Leveraged Python's extensive ecosystem of libraries and frameworks, including Django, to enhance functionality and streamline development.
* **Community Support:** Benefited from the vibrant Python community, ensuring access to a wealth of resources, documentation, and support for overcoming challenges encountered during the development process.

### CHAPTER 8

### IMPLEMENTATION

The implementation of a **Smart Blood Bank System using IOT** involves deploying IOT-enabled hardware such as sensors for monitoring storage conditions, RFID/QR code scanners for tracking blood units, and IOT gateways for data transmission. A communication network using protocols like MQTT and technologies like Wi-Fi or 4G/5G connects these devices to a centralized cloud platform, which manages data storage, real-time analytics, and notifications. Software applications, including a web-based admin dashboard, a donor mobile app, and a hospital interface, facilitate user interactions and system management. Security measures like encryption, role-based access control, and backups ensure data integrity and privacy. Following comprehensive testing and a pilot deployment, the system is launched, with ongoing maintenance and updates to address issues, improve functionality, and scale to support additional blood banks and hospitals.

**8.1 INSTALLATIONS OF ARDUINO:**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a

text editor for writing code, a message area, a text console, a toolbar with buttons for common

functions and a series of menus. It connects to the Arduino hardware to upload programs and

communicate with them.

**Step 1** − First you must have your Arduino board (you can choose your favorite board) and a

USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560,

or Diecimila, you will need a standard USB



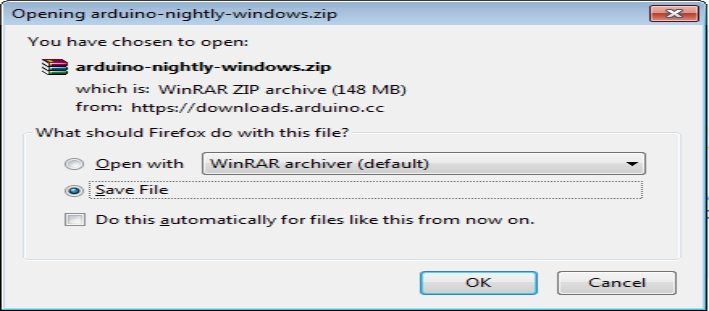
**Fig.8.1.1.** USB Port

**Step 2 − Download Arduino IDE Software.**

You can get different versions of Arduino IDE from the Download page on the Arduino

Official website. You must select your software, which is compatible with your operating

system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



**Fig.8.1.2.** Downloading of Arduino IDE Software

**Step 3 − Power up your board**

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from

either,the USB connection to the computer or an external power supply. If you are using an the

Arduino Diecimila, you have to make sure that the board is configured to draw power from the

USB connection. The power source is selected with a jumper, a small piece of plastic that fits

onto two of the three pins between the USB and power jacks. Check that it is on the two pins

closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED

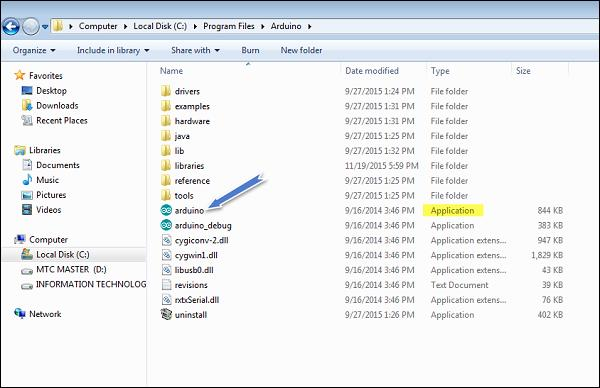
(labeled PWR) should glow.

**Step 4 − Launch Arduino IDE.**

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder,

you can find the application icon with an infinity label ( application.exe). Double-click the icon

to start the IDE.



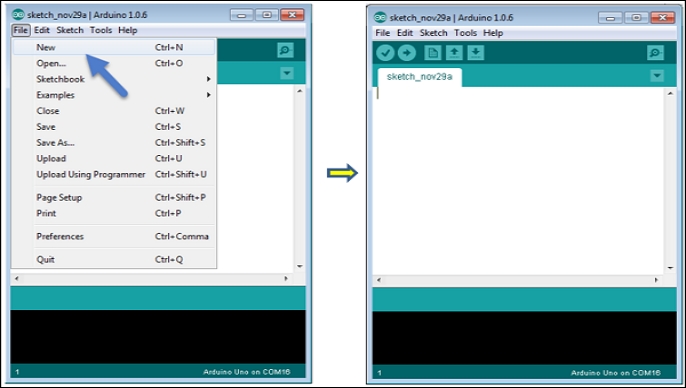
**Fig.8.1.4.** Launching Arduino IDE

**Step 5 − Open your first project.**

Once the software starts, you have two options −

* Create a new project.
* Open an existing project example.

To create a new project, select File → **New**.



**Fig.8.1.5.** Creating New Project

To open an existing project example, select File → Example → Basics → Blink.

#### 

#### Fig. 8.1.6. Existing Project Example

#### 

#### Here, we are selecting just one of the examples with the name Blink. It turns the LED on and

#### off with some time delay. You can select any other example from the list.

#### Step 6 − Select your Arduino board.

#### To avoid any error while uploading your program to the board, you must select the correct

#### Arduino board name, which matches with the board connected to your computer.

#### Go to Tools → Board and select your board.

#### 

#### 

#### Fig.8.1.7 : Selecting Arduino Board

#### 

#### Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

#### Step 7 − Select your serial port.

#### Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely

#### to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To

#### find out, you can disconnect your Arduino board and re-open the menu, the entry that

#### disappears should be of the Arduino board. Reconnect the board and select that serial port.

#### 

#### Fig.8.1.8 : Selecting Port

#### Step 8 − Upload the program to your board.

#### Before explaining how we can upload our program to the board, we must demonstrate the

#### function of each symbol appearing in the Arduino IDE toolbar.

#### 

#### Fig.8.1.9 : Uploading the Program

#### A − Used to check if there is any compilation error.

#### B − Used to upload a program to the Arduino board.

#### C − Shortcut used to create a new sketch.

#### D − Used to directly open one of the example sketch.

#### E − Used to save your sketch.

#### F − Serial monitor used to receive serial data from the board and send the serial data to the

#### board.

#### Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see

#### the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done “.

#### 8.2 THINGSPEAK SETUP

#### Steps to set up thingspeak

#### Step1: Open iot analytics thingspeak website and click on the profile.

#### Step2: Click on create one and enter your mail id & other details.

#### 

#### 

#### 

#### Fig 8.2.1 : Creating account

#### Step 3: Click on the new channel for creating your own channel.

#### Step 4: Give the name for channel and set the fields then click on save channel

#### Fig 8.2.2: New Channel Creation

#### 

#### 

#### 

#### 

#### Fig 8.2.3: Naming channel and fields

#### 

#### Here in the below we created a channel with needed fields.

#### 

#### 

#### 

#### Fig 8.2.4: Saved Channel Status

#### 

#### Step 5: Inorder to link the fields with our devices use the api keys provided by server in the corresponding code.

#### 

#### 

#### 

#### Fig 8.2.5. API keys

#### 8.3 CODE

#### 

#### #include <HX711.h>

#### #include <DHT.h>

#### #include <SoftwareSerial.h>

#### 

#### // ThingSpeak and WiFi credentials

#### const char\* ssid = "YourWiFiSSID";

#### const char\* password = "YourWiFiPassword";

#### const char\* server = "api.thingspeak.com";

#### String apiKey = "YourAPIKey";

#### 

#### // Pin definitions

#### #define DHTPIN 7

#### #define DHTTYPE DHT11

#### #define HX711\_DT1 3

#### #define HX711\_DT2 4

#### #define HX711\_SCK 2

#### 

#### // Initialize sensors

#### DHT dht(DHTPIN, DHTTYPE);

#### HX711 scale1, scale2;

#### SoftwareSerial espSerial(10, 11); // RX, TX

#### 

#### void setup() {

#### Serial.begin(9600);

#### espSerial.begin(115200); // ESP01 baud rate

#### 

#### // Initialize DHT sensor

#### dht.begin();

#### 

#### // Initialize HX711 modules

#### scale1.begin(HX711\_DT1, HX711\_SCK);

#### scale2.begin(HX711\_DT2, HX711\_SCK);

#### 

#### // Connect to WiFi

#### connectToWiFi();

#### }

#### 

#### void loop() {

#### // Read Load Cell 1

#### float weight1 = scale1.get\_units(10);

#### Serial.print("Weight 1: ");

#### Serial.print(weight1);

#### Serial.println(" g");

#### 

#### // Read Load Cell 2

#### float weight2 = scale2.get\_units(10);

#### Serial.print("Weight 2: ");

#### Serial.print(weight2);

#### Serial.println(" g");

#### 

#### // Read DHT11

#### float temperature = dht.readTemperature();

#### float humidity = dht.readHumidity();

#### Serial.print("Temperature: ");

#### Serial.print(temperature);

#### Serial.println(" °C");

#### Serial.print("Humidity: ");

#### Serial.print(humidity);

#### Serial.println(" %");

#### 

#### // Send data to ThingSpeak

#### sendDataToThingSpeak(weight1, weight2, temperature, humidity);

#### 

#### delay(15000); // ThingSpeak requires a 15-second delay between updates

#### }

#### 

#### void connectToWiFi() {

#### espSerial.println("AT");

#### delay(1000);

#### espSerial.println("AT+CWMODE=1");

#### delay(1000);

#### espSerial.print("AT+CWJAP=\"");

#### espSerial.print(ssid);

#### espSerial.print("\",\"");

#### espSerial.print(password);

#### espSerial.println("\"");

#### delay(5000);

#### Serial.println("Connected to WiFi.");

#### }

#### 

#### void sendDataToThingSpeak(float weight1, float weight2, float temp, float hum) {

#### String data = "field1=" + String(weight1) + "&field2=" + String(weight2) +

#### "&field3=" + String(temp) + "&field4=" + String(hum);

#### 

#### espSerial.println("AT+CIPSTART=\"TCP\",\"" + String(server) + "\",80");

#### delay(2000);

#### espSerial.println("AT+CIPSEND=" + String(data.length() + 16 + apiKey.length()));

#### delay(2000);

#### espSerial.print("GET /update?api\_key=");

#### espSerial.print(apiKey);

#### espSerial.print("&");

#### espSerial.println(data);

#### delay(2000);

#### espSerial.println("AT+CIPCLOSE");

#### Serial.println("Data sent to ThingSpeak”)

}

#### CHAPTER 9

#### SOFTWARE TESTING

**Software testing** is crucial in ensuring that a system operates effectively, securely, and reliably. In the context of the **Smart** **Blood Bank Management System Using IoT**, testing is essential to verify that the system meets user requirements, works under real-world conditions, and integrates well with IoT devices and platforms like ThingSpeak for real-time monitoring and control. Below is an outline of the testing phases adapted for the Blood Bank Management System with IoT integration:

**9.1 UNIT TESTING**

Unit testing focuses on testing individual components or functions of the Blood Bank Management System in isolation. Each unit is tested to ensure it performs its intended functionality. For example:

* **IoT sensor readings:** Testing the functionality of sensors that measure temperature, humidity, and blood stock levels to ensure they provide accurate data.
* **Database operations:** Verifying that operations like adding new donors, registering blood types, and updating stock levels work independently.
* **User Interface (UI) interactions:** Testing user authentication, account creation, and accessing blood bank inventory information.

Unit testing helps ensure that each part of the system works correctly before integrating it with other components

**9.2 INTEGRATION TESTING**

* Integration testing checks the interaction between different modules and components within the Blood Bank Management System. This ensures that IoT devices, the database, and the application interface work together seamlessly. For example:
* **IoT data integration:** Verifying that sensor data (such as temperature readings from IoT sensors) correctly flows into the ThingSpeak platform and is retrieved by the system for monitoring.
* **Database and IoT synchronization:** Ensuring that data like blood availability and storage conditions are updated in real-time based on sensor input and user interactions.
* **APIs and communication protocols:** Ensuring APIs communicate between IoT devices and the system effectively, allowing real-time updates and notifications

**9.3 SYSTEM TESTING**

System testing verifies that the complete Blood Bank Management System works as expected, with all components integrated. This phase involves testing the full functionality of the system, from registering new donors to tracking blood inventory. Specific tests include:

* **Blood donation and request flow:** Verifying that users can donate blood, and the system correctly updates the available stock, while donors are registered appropriately.
* **Real-time monitoring and alerts:** Ensuring the system provides real-time temperature and blood inventory alerts via ThingSpeak integration.
* **End-to-end user interactions:** Testing the complete user experience, including registration, blood donation tracking, requests for blood, and status updates.

#### 9.4 ACCEPTANCE TESTING

Acceptance testing ensures the system meets the end user's requirements and expectations. For the Blood Bank Management System, this involves verifying that it meets the critical needs of both donors and administrators. For instance:

* **User access and account creation:** Ensuring that users can easily register, log in, and perform actions such as donating blood and receiving blood requests.
* **Administrator interface:** Verifying that administrators can efficiently manage donor data, monitor blood stock, and receive alerts when inventory reaches critical levels.
* **Real-time updates and reporting:** Ensuring that the system provides accurate and timely reports on blood availability and condition, leveraging ThingSpeak’s real-time monitoring.

#### 9.5 PERFORMANCE TESTING

Performance testing evaluates how well the Blood Bank Management System handles various load conditions, especially when integrating with IoT devices. This includes:

* **Scalability:** Ensuring the system can handle a growing number of users and sensor data without performance degradation.
* **Stress testing:** Simulating high traffic, like many simultaneous blood donations or requests, and evaluating how the system performs under pressure.
* **Response time and throughput:** Testing how quickly the system can respond to queries,update stock levels,and push real-time notifications to administrators

#### 9.6 SECURITY TESTING

Security testing focuses on ensuring that the Blood Bank Management System is secure and protects sensitive data, such as donor information and blood inventory details. This includes:

* **IoT device security:** Ensuring that data from IoT sensors (e.g., blood temperature, humidity) is encrypted to protect against tampering.
* **Authentication and authorization:** Verifying that users, administrators, and devices have the appropriate permissions to access or modify data.
* **Data encryption:** Ensuring that sensitive donor information and blood bank records are encrypted in transit and storage.

**Protection against vulnerabilities:** Testing for common threats like SQL injection, cross-site scripting (XSS), and unauthorized access.

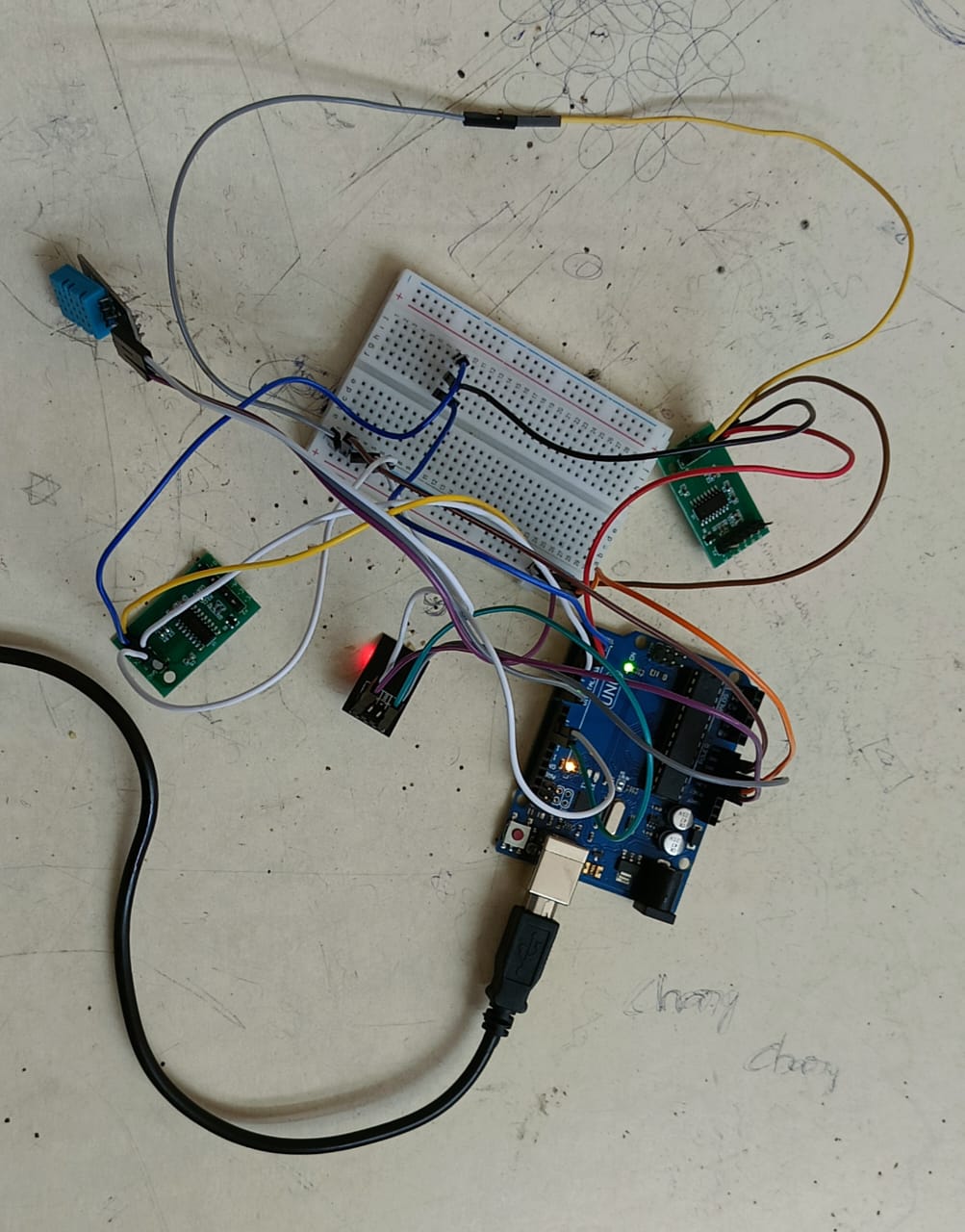
### 9.7 ****COMPATIBILITY TESTING****

Compatibility testing ensures the Blood Bank Management System works across different devices, browsers, and operating systems. It is crucial to verify that the system provides a consistent user experience across various platforms:

* **Browser compatibility:** Ensuring that the system functions correctly on popular browsers such as Chrome, Firefox, Safari, and Edge.
* **Device compatibility:** Testing the system on different devices, including desktop computers, tablets, and mobile phones, to ensure responsiveness and ease of use.
* **Operating system compatibility:** Ensuring that the system operates smoothly across multiple operating systems (e.g., Windows, macOS, Linux, iOS, Android).
* **ThingSpeak integration on mobile devices:**Thingspeak platform and its API work across various mobile platform.

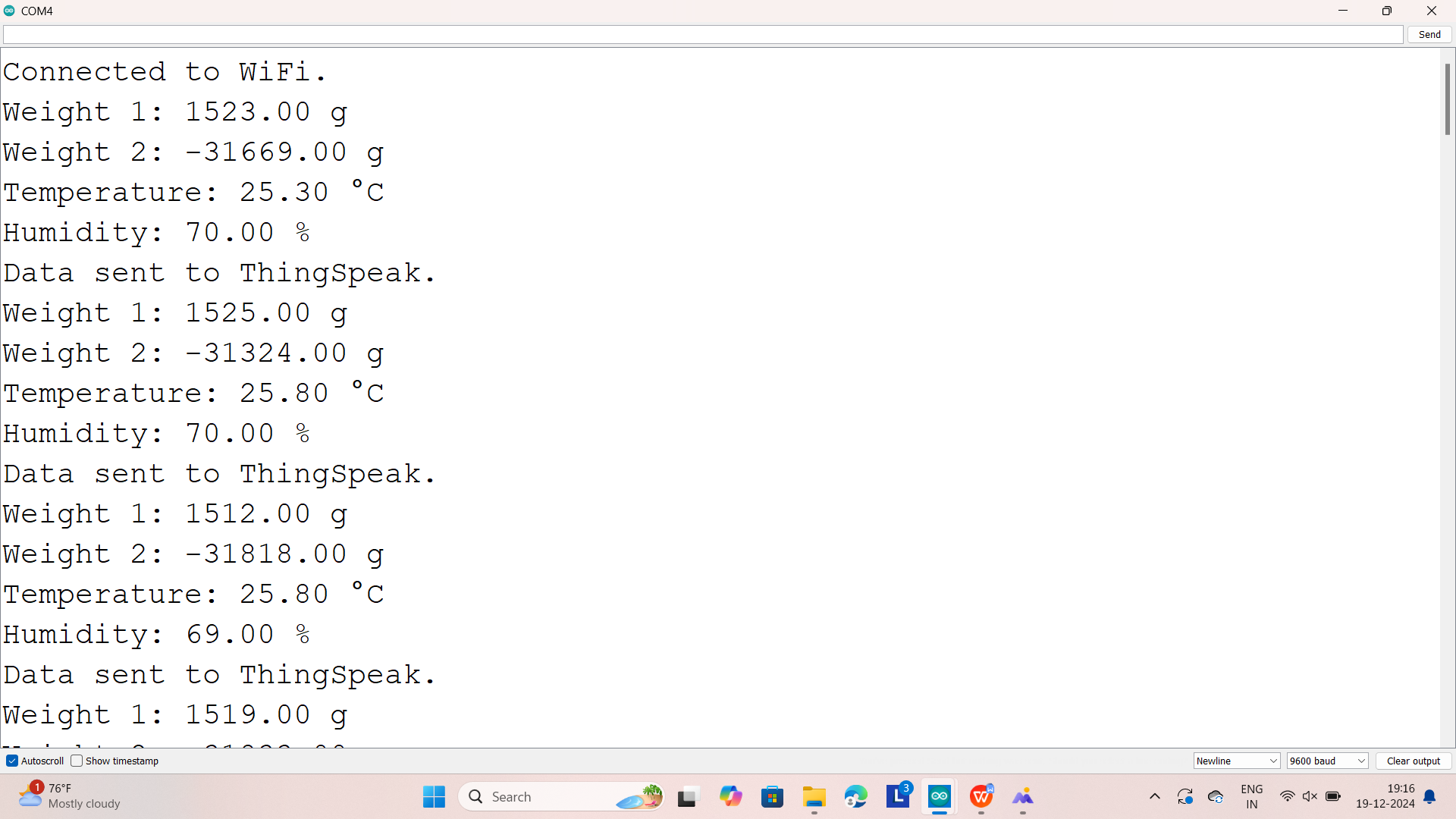
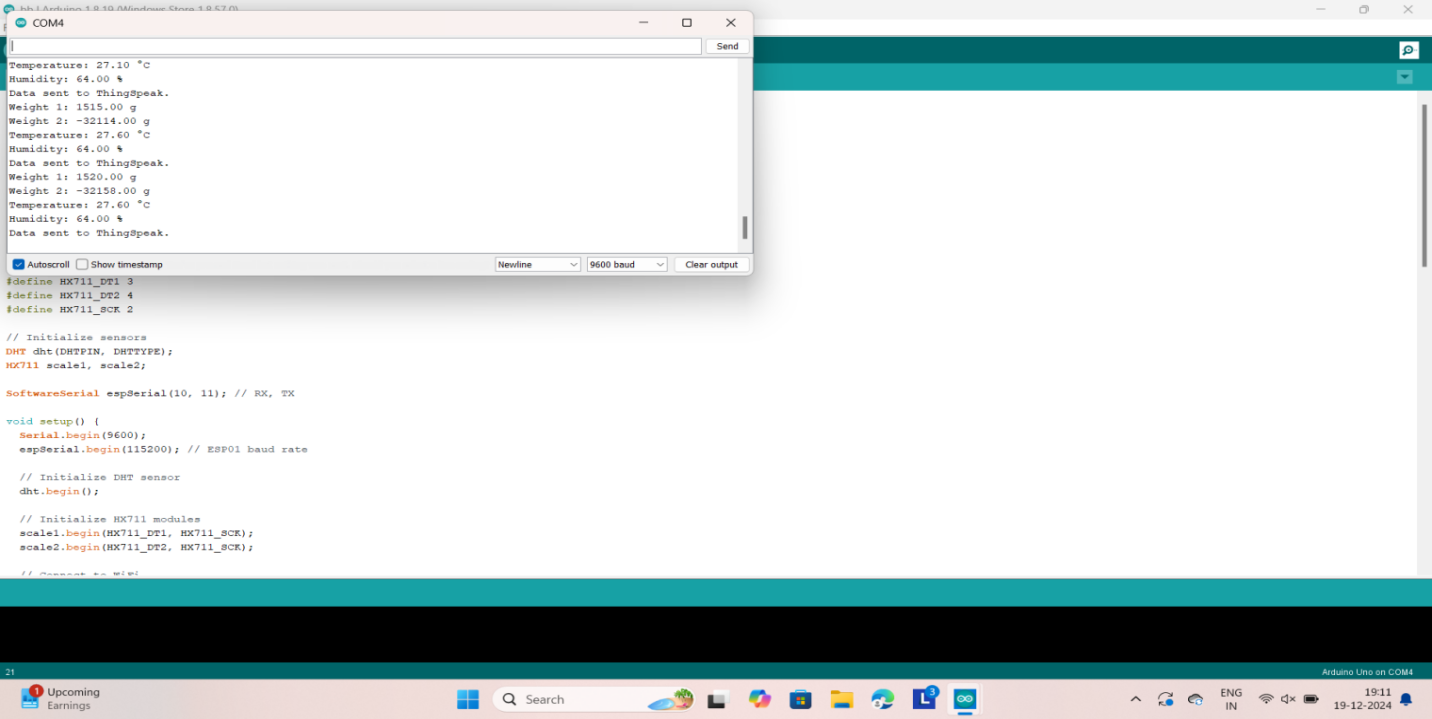
# CHAPTER 10

# RESULTS AND DISCUSSIONS



In the Smart Blood Bank Management System using IoT, various components are integrated to ensure efficient monitoring and management of blood donations. The system utilizes two \*HX711 Load Cells\* (Load Cell 1 and Load Cell 2) to measure the weight of blood bags. These load cells the HX711 modules to amplify and read the weight data. The load cells' \*DT\* (Data) and \*SCK\* (Clock) pins are connected to specific analog and digital pins on the Arduino for proper data transfer and processing. Additionally, the \*DHT11 sensor\* is incorporated to measure the temperature and humidity in the blood storage area, ensuring the blood bags are maintained in an optimal environment. The data from the DHT11 is read by the Arduino, which monitors the conditions in real time.

### RESULTS



**CHAPTER 11**

### CONCLUSION

The Blood Bank System represents a step toward improving blood donation and management processes. Throughout the development of this system, several challenges were addressed, and significant milestones were achieved. The system successfully implemented real-time monitoring of blood stock levels, donor information management, and IOT - enabled data collection for efficient inventory tracking.

This innovative solution provides a centralized platform for healthcare providers and blood banks, streamlining the management of blood resources. Real-time data visualization and analytics on Thing\_Speak enable stakeholders to monitor trends, anticipate demand, and make informed decisions. By leveraging IOT technology, the system ensures transparency, efficiency, and reliability in blood storage and distribution.

The implementation of this system lays the groundwork for enhancing the availability and accessibility of blood resources, saving lives through timely action, and fostering a more efficient healthcare ecosystem. As the system continues to evolve and incorporate advanced features, it has the potential to revolutionize blood donation management and adapt to the ever-changing needs of the healthcare industry

**CHAPTER 12**

### FUTURE SCOPE

While the Blood Bank System using IOT with ThingSpeak meets the defined scope and requirements, there are several areas for future enhancements, including:

* **Integration with Emergency Notification Systems.**
* **Advanced Data Analytics and Reporting.**
* **Predictive Stock Management.**
* **Mobile Application Integration.**
* **Interoperability with National Databases.**
* **IoT Device Integration.**
* **User Feedback Mechanism.**
* **Geo-location-based Features.**
* **Real-time Notifications.**
* **Blockchain for Data Security.**

### REFERENCES

[1] Radha Mahalle, S. S. Thorat, "Smart Blood Bank Based On IoT”,<https://www.irjet.net/archives/V5/i1/IRJET-V5I1101.pdf>[2]Shelley Khosla and Dr. Anu Sahi, "Smart Blood Bank System Based on IOT" <https://indusedu.org/pdfs/IJREISS/IJREISS_4277_94721.pdf>

[2] E. Muthu Kumaran, K. Velmurugan, P. Venkumar, D. Amutha Guka, V. Divya, "Artificial Intelligence-Enabled IoT-Based Smart Blood Banking System” <https://link.springer.com/chapter/10.1007/978-981-16-6332-1_12>

[3] G. Muddu Krishna, S. Nagaraju,"Design and Implementation of Short Message Service (SMS) Based Blood Bank"

[4] Shelley Khosla and Dr. Anu Sahi, "Smart Blood Bank System Based on IOT" <https://indusedu.org/pdfs/IJREISS/IJREISS_4277_94721.pdf>