# CAPSTONE PROJECT WORK REPORT

**Phase II**

**IV ALERTX**

# DIVYA M

# Reg. No:2328B0025

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Science in Computer Science with Data Analytics**

Guide**: Mr. B. RAMESH KUMAR** M.Sc., MCA, M.Phil., NET, SET, (Ph.D.).,

Assistant Professor, Department of Computer Science with Data Analytics

## Department of Computer Science with Data Analytics

**KPR College of Arts Science and Research**

(Affiliated to Bharathiar University, Coimbatore) Avinashi

Road, Arasur, Coimbatore-641 407

**OCTOBER 2025**

# CAPSTONE PROJECT WORK REPORT

## Phase-II

**IV ALERTX**

Bonafide Work Done by

# DIVYA M

**Reg. No: 2328B0025**

Dissertation submitted in partial fulfillment of the requirements for the award of Bachelor of Science in Computer science with Data Analytics of Bharathiar University, Coimbatore-46.



**Signature of the Guide Signature of the HOD** [Mr. B. Ramesh Kumar] [Dr. G. Satyavathy]

Submitted for the Viva-Voce Examination held on

**Internal Examiner External Examiner**

**OCTOBER 2025**

**ACKNOWLEDGEMENT**

# ACKNOWLEDGEMENT

This acknowledgement to express my deepest gratitude for the support and guidance you have provided me during the course of my capstone project. I would like to convey my profound gratitude to Dr. K. P. Ramaswamy, Founder and Chairman of KPR College of Arts Science and Research and Dr. P. Geetha, Principal of KPR College of Arts Science and Research, for her endless support. I would also like to extend my sincere thanks to Dr. K. Pradeepa, Dean, School of Intelligent Systems and Data Science and our beloved Dr. G. Satyavathy, Professor & Head, Department of Computer Science with Data Analytics, for providing the moral support and motivation throughout the project.

I’m extremely grateful to my supervisor Mr. B. Ramesh Kumar, Assistant Professor, Department of Computer Science with Data Analytics for being such an incredible mentor throughout my academic journey. I feel privileged to had the opportunity to learn from him and I am grateful for the valuable lessons and experiences that I have gained under your guidance.

Divya M

Place : Coimbatore

Date :

**CONTENTS**

# 

|  |  |
| --- | --- |
| **CONTENTS** | **Pg. No.** |
| ACKNOWLEDGEMENT |  |
| SYNOPSIS |  |
| **1.INTRODUCTION** | **1** |
| 1.1AI and ML Technology | **2** |
| 1.2OrganizationProfile | **3** |
| 1.3System Specification | **4** |
| 1.3.1 Hardware Configuration | **4** |
| 1.3.2 Software Configuration | **4** |
| **2.SYSTEM STUDY** | **5** |
| 2.1 Existing System | **6** |
| 2.1.1 Drawbacks | **6** |
| 2.2 Proposed System | **7** |
| 2.2.1 Features | **7** |
| 2.2.2 Modules | **8** |
| **3.SYSTEM DESIGN** | **9** |
| 3.1 Form Design | **9** |
| 3.2 Input Design | **10** |
| 3.3 Output Design | **10** |
| 3.4 Database Design | **11** |
| **4.SOFTWARE TESTING & IMPLIMENTATION** | **11** |
| CONCLUSION | **15** |
| **BIBLIOGRAPHY** | **16** |
| **APPENDICES** | **17** |
| A. Data Flow Diagram | **17** |
| B. Flow chart | **18** |
| C. Sample Code | **19** |

**SYNOPSIS**

## SYNOPSIS

The IV Alertx is designed to track IV fluid levels in real time and notify medical staff when the fluid reaches a critical level. By ensuring timely replacement, the system helps prevent air embolism risks and enhances patient safety. This solution is ideal for hospitals, clinics, and home healthcare settings, where continuous IV fluid monitoring is essential.

The system consists of several key components that work together to provide accurate monitoring and alerts. It includes weight or ultrasonic sensors to measure fluid levels, a microcontroller (such as Arduino or Raspberry Pi) to process data, and a wireless communication module (Wi-Fi, Bluetooth, or IoT) to transmit alerts.

To ensure timely response, the system provides multiple types of alerts. These include visual indicators like LEDs, audible alarms through buzzers, and real-time notifications sent to nurses and hospital staff via mobile apps or hospital managements systems. This multi-layered alert system minimizes the risk of IV fluid depletion.

The system operates by continuously monitoring the IV fluid levels, with the microcontroller processing the data to detect when the fluid reaches a predefined low threshold. When this level is reached, it automatically triggers alerts, ensuring that medical staff can promptly replace the IV bag.

In addition to real-time monitoring, the system can integrate with hospital management software for automated data logging. This allows for better tracking of IV fluid usage, reducing manual monitoring efforts and improving overall workflow efficiency in healthcare settings.

## 1.INTRODUCTION

Intravenous (IV) therapy is a crucial aspect of modern healthcare, used for delivering fluids, medications, and nutrients directly into a patient’s bloodstream. However, manual monitoring of IV fluid levels can be time-consuming and prone to human error, leading to potential risks such as fluid depletion, air embolism, and delayed patient care.

To address these challenges, an IV Bag Monitoring and Alert System has been developed to automate the process of tracking IV fluid levels and notifying medical staff when intervention is required.

This system utilizes advanced sensor technology, microcontrollers, and wireless communication to continuously monitor the IV bag’s fluid level in real time. By integrating visual, audible and mobile notifications, it ensures that healthcare providers receive timely alerts when fluid levels approach a critical threshold, reducing the risk of complications and improving patient safety.

Designed for use in hospitals, clinics, and home healthcare settings, this system enhances efficiency by minimizing the need for manual checks. It allows medical professionals to focus on critical tasks while ensuring uninterrupted IV therapy.Additionally, integration with hospital management systems enables automated logging and data tracking, contributing to better healthcare management.

With features such as real-time monitoring, automated alerts, wireless connectivity, battery backup, and user-friendly interfaces, the IV Bag Monitoring and Alert System provides a reliable and efficient solution for healthcare providers. By reducing the chances of IV fluid depletion and improving workflow efficiency, this system represents a significant step toward smarter and safer patient care.

**1.1.AI and ML Technology:**

Artificial Intelligence (AI) and Machine Learning (ML) enhance the IV Bag Monitoring and Alert System by enabling predictive analytics, anomaly detection, and automated decision-making, improving patient safety and efficiency. ML algorithms analyze real-time IV fluid consumption patterns to predict depletion, allowing for proactive replacement alerts.

AI detects anomalies such as blockages, leakage, or patient movement, triggering instant alerts to prevent complications. Smart alert prioritization ensures critical alerts are sent immediately, while AI-driven infusion pumps adjust IV flow rates based on patient needs, integrating with Electronic Health Records (EHRs). Remote monitoring via AI-powered dashboards enables real-time tracking and hospital workflow optimization.

The ML model involves data collection (IV bag weight, flow rate, patient vitals), preprocessing (noise removal, normalization), model training (supervised learning for prediction, unsupervised learning for anomaly detection), and real-time deployment to monitor and minimize false alarms.

Benefits include automated IV monitoring, reduced human error, optimized nursing workflows, and enhanced patient safety. Future advancements may include AI-powered infusion pumps, health sensor integration, and AI chatbots for real-time monitoring, making IV fluid management more efficient and intelligent.The IV Bag Monitoring System form should be user-friendly, allowing healthcare professionals to efficiently input and track IV-related data.

The form begins with a Patient Information Section, where details such as patient name, hospital ID, age, gender, and room/bed number are recorded. Next, the IV Bag Details section includes fields for IV type (e.g., Saline, Dextrose, or Ringer’s Lactate), IV volume (in mL), flow rate, start time, estimated completion time, & IV bag expiry date.

To ensure accurate monitoring, the Monitoring Parameters section automatically updates real-time values for IV level percentage, flow rate, line pressure, and anomaly detection (e.g., blockages, leaks, or air bubbles) using sensors or AI-based vision detection. The Alert & Notification Settings allow medical staff to set thresholds for low IV level alerts, flow rate limits, and notification methods (such as SMS, email, mobile alerts, or nurse station notifications). It also includes fields for entering the responsible nurse or doctor’s contact details.

The form also provides essential Action & Control buttons such as Start IV Monitoring, Pause/Stop IV, Acknowledge Alerts, Update IV Details, and Generate Reports for easy data management. Additionally, AI-powered features such as IV depletion time prediction using ML algorithms, graphical visualization with D3.js, and voice alerts through text-to-speech can be integrated to enhance

Usability.

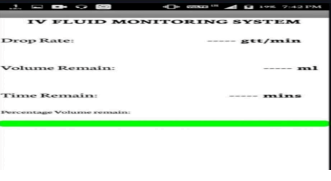


Fig.1.Fluid Monitoring

**1.3. SYSTEM SPECIFICATION**

### 1.3.1 Hardware Configuration

|  |  |
| --- | --- |
| DEVICE NAME | ARDUINO UNO |
| PROCESSOR | 16 MHZ ATmega328p |
| Device ID | 787E2198-A71D-440B  A9E2DB7E84754FC1 |
| Product ID | 00356-24659-86620-AAOEM |
| System Type | 64-bit operating system, x64-based processor |

#### 1.3.2 Software Configuration

|  |  |
| --- | --- |
| IDE | Visual Studio |
| Language Support | C,C++ |
| Platform | Web-based Dashboard |
| Browser | Google Chrome Version 101.0.4951.67 |

## 2. SYSTEM STUDY

### 2.1 EXISTING SYSTEM

Several existing IV bag monitoring and alert systems are designed to improve patient safety, reduce manual monitoring, and enhance hospital efficiency. These systems use different technologies, such as weight-based monitoring, which relies on load sensors to measure the IV bag’s weight and alert when fluid levels are low, and flow rate monitoring, which tracks IV drip rates using flow sensors to detect any irregularities. Smart IV pumps, like the Baxter Smart IV Pumps and BD Alaris System, come with builtin monitoring and alarms to ensure precise medication delivery.

Additionally, IoT-based monitoring solutions, such as IVision, provide real-time alerts to healthcare staff via mobile apps, while optical and camera-based monitoring systems use image processing to track liquid levels inside IV bags. Advanced infusion systems like the Ivenix Infusion System incorporate cloud based monitoring for enhanced efficiency, and AI-powered solutions like SafeDose IV Monitoring help reduce medication errors.

These innovations contribute significantly to patient safety and hospital workflow optimization. IV systems go a step further by using predictive analytics to anticipate when a refill is needed and even automating flow rate adjustments based on patient needs. Examples of existing smart IV monitoring solutions include Baxter Smart IV Pumps, which regulate flow and detect occlusions; the BD Alaris System, which provides alarm notifications; and the Ivenix Infusion System, which features a touch-screen interface and wireless alerts.

Existing IV bag monitoring and alert systems range from manual methods to fully automated solutions. Traditional systems rely on nurses or caregivers manually checking IV fluid levels, which can lead to human error and delayed intervention, especially in busy hospital settings. Some semi-automated systems use weight based or volume sensors to estimate the remaining fluid level, while others feature

simple visual indicators like mechanical floats or LED warnings.

Advanced fully automated systems integrate various technologies such as weight sensors, optical sensors, and flow rate sensors to provide real-time monitoring. Many modern solutions leverage IoT technology to connect IV monitoring devices to hospital networks, enabling remote tracking through mobile apps or dashboards. Some systems also incorporate data visualization tools like D3.js to present IV fluid levels, flow rates, and alert statuses effectively.

AI-powered smart IV systems go a step further by using predictive analytics to anticipate when a refill is needed and even automating flow rate adjustments based on patient needs. Examples of existing smart IV monitoring solutions include Baxter Smart IV Pumps, which regulate flow and detect occlusions; the BD Alaris System, which provides alarm notifications; and the Ivenix Infusion System, which features a touch-screen interface and wireless alerts.

#### 2.1.1 DRAW BACKS

One major concern is the high cost, as advanced systems like smart infusion pumps and IoT-based solutions can be expensive to implement, making them less accessible for smaller hospitals and clinics. Additionally, these systems require proper calibration, maintenance, and periodic software updates, which can be challenging for healthcare facilities with limited technical expertise.

False alarms are another issue, as some systems generate frequent alerts due to sensor sensitivity, leading to alarm fatigue among healthcare staff and reducing overall efficiency.

Connectivity issues also pose a challenge, as IoT-based and cloud-connected systems depend on stable internet or network infrastructure, and any failure can result in delayed or missed alerts.

Moreover, many monitoring systemsrely on continuous power or battery backups, making them vulnerable to power failures or battery depletion. Integration challenges further complicate their adoption, as some systems are not easily

compatible with existing hospital management software, leading to additional costs for integration and modifications. Sensor limitations can also affect accuracy, as weight and opticalsensors may not provide precise readings if the IV bag is moved or obstructed.

Additionally, IoT-based IV monitoring systems are susceptible to cybersecurity threats, including hacking and data breaches, which could compromise patient safety and privacy.Despite these drawbacks, continuous advancements in technology aim to address these challenges and enhance the reliability of IV bag monitoring and alert systems.

Patient comfort and mobility can also be affected, as wearable or wired IV monitoring devices may cause discomfort or restrict movement. Additionally, if the sensors are not securely attached, they may shift, leading to inaccurate readings.

Maintenance and reliability also present challenges, as the system requires periodic software updates, hardware servicing, and continuous staff training to ensure its effective use. Despite these drawbacks, an IV bag monitoring and alert system remains a valuable tool in improving patient care, provided that these challenges are addressed effectively.

**2.2 PROPOSED SYSTEM**

To address the limitations of existing IV bag monitoring systems, we propose a smart, real-time IV bag monitoring and alert system that integrates advanced sensors, IoT connectivity, and AI-driven analytics for enhanced accuracy, reliability, and efficiency. The system will utilize high-precision weight sensors to continuously monitor IV fluid levels and flow rate sensors to detect any irregularities in the infusion process.

AI algorithms will analyze sensor data to minimize false alarms, ensuring that only critical alerts are sent, while also predicting IV depletion time to notify nurses in advance. With IoT and cloud connectivity, IV status will be wirelessly transmitted to a centralized hospital monitoring system, allowing real-time remote monitoring via mobile and web based applications.

Automated alerts will be sent through mobile apps, nurse station displays, and wearable devices, using a multi-level alert system to prevent alarm fatigue. The system is designed to be power-efficient, incorporating low-power sensors and rechargeable battery backups for uninterrupted operation, along with energy efficient wireless communication for long-term use.

Additionally, it will seamlessly integrate with hospital management software and Electronic Health Records (EHR), ensuring secure data encryption and cloud storage to maintain patient privacy. The user-friendly interface will include a graphical dashboard for easy monitoring and customizable settings for alarm

thresholds.

This proposed system offers significant advantages, such as reducing human error by automating IV monitoring, improving patient safety with timely alerts, enhancing efficiency by allowing nurses to monitor multiple patients remotely, and lowering false alarms through AI-based alert optimization.

In hospitals, intravenous (IV) therapy is essential for patient care, but manual monitoring of IV fluid levels can be inefficient and prone to human error. Nurses must frequently check IV bags to prevent depletion, which increases their workload and the risk of delays in refilling. If an IV bag runs empty unnoticed, it can lead to serious health complications, such as air embolism or improper medication delivery. To solve this issue, an automated IV Bag Monitoring and Alert System is proposed, ensuring continuous tracking of IV fluid levels with real-time notifications.

The proposed system integrates sensor technology, microcontrollers, and a web based dashboard to monitor IV bag levels and alert healthcare professionals when a refill is needed. A fluid level sensor is attached to the IV bag, which continuously measures the remaining liquid and transmits the data to a microcontroller (such

an Arduino or ESP32). The microcontroller processes the information and sends it to a web dashboard, where D3.js visualizations display the real-time IV fluid status.

When the IV level falls below a predefined threshold (e.g., 20%), an alert is triggered, notifying the nurse or doctor via visual indicators, sound alarms, or SMS/email notifications.

One of the key advantages of this system is that it reduces human intervention and improves efficiency in patient care. Instead of manually checking each IV bag, nurses can monitor multiple patients remotely through the web dashboard, significantly saving time and effort. The automated alert mechanism ensures that IV bags are replaced on time, preventing risks associated with fluid depletion.

Additionally, the system can log historical data, helping hospitals analyze IV usage patterns and optimize inventory management.From a technological perspective, the system is designed to be cost-effective and scalable for hospitals of all sizes. It leverages D3.js for real-time data visualization, Firebase/MySQL for data storage, and Node.js/Python for backend processing.

The use of wireless communication allows easy integration with hospital networks, enabling remote monitoring. Moreover, this system can be expanded to support multiple IV bags simultaneously, making it suitable for ICU environments and large healthcare facilities.

In conclusion, the IV Bag Monitoring and Alert System enhances patient safety by eliminating manual errors and ensuring timely IV refills. By automating the monitoring process, it not only improves hospital efficiency but also reduces the workload on healthcare professionals, allowing them to focus on critical patient care. The integration of sensors, microcontrollers, and web-based dashboards creates a smart healthcare solution that can be further developed for broader medical applications. This system represents a significant step toward digital transformation in hospitals, ensuring better healthcare outcomes through technology.

#### 2.2.1 FEATURES

An IV Bag Monitoring and Alert System should have key features that ensure real-time tracking of fluid levels, patient safety, and ease of use.

Here are some essential features:

**Real-Time Monitoring**

* Fluid Level Detection: Sensors measure the remaining IV fluid and update data in real time.
* Flow Rate Monitoring: Tracks the IV drip rate to ensure proper administration.
* Weight-Based Measurement: Uses load sensors to determine the IV bag’s weight for accuracy.

**Alert System**

* Low Fluid Alerts: Sends notifications when the IV fluid is running low.
* Drip Rate Alerts: Alerts if the flow rate is too fast or too slow.
* Bag Empty Alerts: Notifies nurses or caregivers when the bag is empty.
* Disconnection Alerts: Detects if the IV tube is disconnected or blocked.

**Dashboard & Visualization (D3.js Integration)**

* Graphical Representation: Shows real-time IV fluid levels in a user-friendly format.
* Historical Data Analysis: Tracks past IV usage trends for better treatment decisions.
* Interactive Charts: Uses D3.js for dynamic and responsive visualizations.

**Connectivity & Integration**

* Wireless Connectivity: Uses Wi-Fi, Bluetooth, or IoT-based communication.
* Mobile & Web Alerts: Sends SMS, email, or app notifications.
* Integration with Hospital Systems: Can be linked with electronic health records (EHR)

**Security & Access Control**

* User Authentication: Only authorized personnel can access or modify data.
* Data Encryption: Ensures patient data security and compliance with regulations

**Power Management**

* Battery Backup: Ensures continuous monitoring during power failures.

**MODULES**

**Display Module**

* LCD/OLED Display – Shows real-time IV fluid levels and alerts.
* Touchscreen Interface – For user interaction in advanced systems.

**Wireless Communication Modules**

* Wi-Fi Module (ESP8266, ESP32) – Sends data to a hospital server or mobile app.
* Bluetooth Module (HC-05, BLE) – Connects with smartphones for monitoring.
* GSM/GPRS Module (SIM800L, SIM900) –Sends SMS alert caregivers.

**Alarm & Notification System**

* Buzzer – Alerts when IV fluid is low.
* LED Indicator – Visual warning for fluid depletion.

**3.SYSTEM DESIGN**

**3.1. FORM DESIGN**

The IV Bag Monitoring and Alert System form captures patient details, IV monitoring settings, and alert preferences efficiently. It includes Patient Information (ID, Name, Age, Gender, Room Number, Assigned Doctor/Nurse) and IV Bag Details (Bag ID, Fluid Type, Volume, Infusion Rate, Start & End Time).

Monitoring & Alert Settings allow users to set minimum fluid level, maximum flow rate alerts, and notification preferences (SMS, Email, App, Nurse Station). System Status & Logs provide real-time IV levels, infusion status, last alert time, and a nurse action log. Control Buttons include Start, Pause, Reset, and Stop Monitoring. The form ensures accuracy, integration with hospital systems, real-time updates, and role based security, improving patient safety and hospital

efficiency.

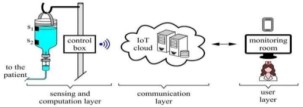


Fig.2 Form Design

**3.2. INPUT DESIGN**

An IV Bag Monitoring and Alert System helps track IV fluid levels in real time, ensuring patients receive the right amount of fluids without issues. It includes sensors to monitor key factors: weight sensors measure remaining fluid, flow sensors track movement speed, pressure sensors detect blockages or air bubbles, optical sensors check fluid levels inside the bag, and temperature sensors ensure the fluid stays at the correct temperature.

Medical staff can enter patient details, IV type, and flow rate using a touchscreen,

mobile app, barcode, or RFID scanner, with voice commands for hands-free control and a keypad as a backup.

A small computer, like an Arduino, Raspberry Pi, or ESP32, processes the data and uses AI to predict issues such as low fluid levels, flow rate changes, or air bubbles. If a problem occurs, the system sends alerts through sound alarms, blinking lights, mobile notifications, or updates to the nurse’s station.

Connectivity is maintained through Wi-Fi or Bluetooth, while large hospitals can use LoRa or Zigbee for long-range communication, and cloud storage enables remote monitoring and data analysis. The process begins with scanning the IV bag, followed by continuous monitoring. If a problem is detected, alerts are sent, and data is stored for future use, improving patient care and hospital efficiency

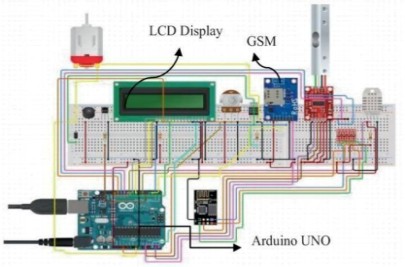


Fig 3. Input Design

**3.3. OUTPUT DESIGN**

The output design of the IV Bag Monitoring and Alert System ensures real time alerts, notifications, and data logging to enhance patient care. It includes audible alerts via buzzers for low fluid levels, flow irregularities, or air bubbles, and visual alerts using LED indicators and LCD displays to show system status. A mobile app sends SMS, email, or push notifications to nurses and doctors, while a centralized nurse station dashboard allows remote monitoring of multiple

patients.

The system updates electronic health records (HER) automatically and generates daily and weekly reports on IV fluid usage, alert history, and performance trends, including graphical data visualization. Connectivity options such as Wi-Fi, Bluetooth, LoRa, and Zigbee enable seamless communication, with cloud integration for secure data storage and remote access.

When the fluid level drops or an issue is detected, alarms are triggered, nurses receive notifications, and all events are logged for analysis, optimizing IV fluid management and improving hospital efficiency.



Fig 4. Output Design

**3.4. DATABASE DESIGN**

The IV Bag Monitoring System requires a well-structured database to store and manage information related to patients, IV bags, sensor data, alerts, and medical staff. The Patient Table stores patient details and links to the IV Bag Table, which contains information about IV fluid type, volume, expiry date, and status. The Sensor Data Table records real-time monitoring parameters such as weight, flow rate, pressure, and temperature, linking to the IV Bag Table since each IV bag has multiple sensor readings.

The Alert Table logs system alerts and notifications for critical issues like low fluid levels, blockages, or air bubbles, linking both to the IV Bag Table and the Patient Table, as each IV bag can trigger multiple alerts for a patient.

The User Table manages medical staff details, including doctors and nurses, and links to the Patient Table, ensuring that each patient is assigned a responsible doctor and nurse.

Additionally, the Log Table keeps track of system activities and historical records, capturing user interactions such as adjustments to IV flow rates or alert resolutions. The database design incorporates foreign keys to establish relationships, ensuring structured data retrieval, while indexes and constraints maintain data integrity, security, and efficiency. Additionally, AI-driven analytics can be integrated to provide predictive alerts and optimize IV fluid management.

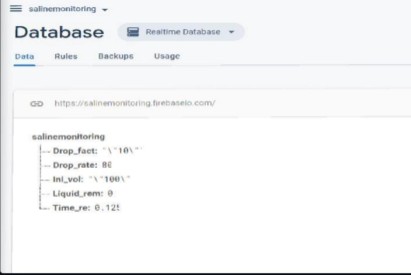


Fig 5. Database Design

**4. SOFTWARE TESTING & IMPLIMENTATION**

The software testing and implementation process for the IV Bag Monitoring and Alert System is meticulously designed to ensure it meets the highest standards of safety, reliability, and clinical efficiency. From the early development stages, the system undergoes extensive testing, beginning with unit testing to verify the functionality of individual components such as fluid level detection algorithms and alert generation modules.

This is followed by integration testing, where the interaction between hardware sensors and the software platform is validated to ensure accurate data capture and response. System-level testing is then conducted to evaluate the complete setup in simulated real- world conditions, ensuring that it meets all functional, performance, and safety requirements. Crucially, user acceptance testing (UAT) is carried out with nurses, clinical staff, and biomedical engineers in hospital-like environments to confirm that the interface is intuitive, the alert timing is appropriate, and the system aligns with existing workflows.

Throughout this process, advanced tools like automated testing scripts, sensor emulators, and bug tracking systems are used to identify and resolve potential issues, while continuous regression testing safeguards the stability of the software after updates or improvements. Compliance with international healthcare and medical device standards is a cornerstone of our QA process. The software aligns with IEC 62304 standards for the software lifecycle of medical devices, ISO 13485 for medical quality management systems, and HIPAA regulations where applicable, ensuring that both functionality and data privacy are upheld.

Once the testing phase is complete, the system enters a phased implementation process, starting with a detailed site assessment to understand the facility’s layout, network infrastructure, and patient care protocols. Based on this analysis, the system is configured and customized to meet the specific needs of the healthcare facility.

A pilot deployment is then executed in one or more units, allowing for live testing, staff feedback, and iterative improvements. Upon successful validation, a full-scale rollout is conducted across all relevant departments, accompanied by hands-on training sessions for nurses, IT staff, and administrators.

After implementation, our support team provides ongoing technical support, periodic updates, remote monitoring, and performance analytics to ensure the system continues to function at peak efficiency.

This end-to-end process guarantees a smooth transition from manual to automated IV monitoring, ultimately improving patient safety, reducing nursing workload, and minimizing the risk of IV-related complice.

**Parameters**

* Fluid Level – Measures the remaining IV fluid in the bag.
* Threshold Value – The minimum level that triggers an alert.
* Response Time – Time taken by the system to detect and notify.
* Flow Rate – Speed of fluid passing through the IV line.
* Network Latency – Delay between sensor data and cloud update.
* Power Consumption – Energy usage of the NodeMCU and sensors.
* Update Interval – Time gap between each data reading (e.g., 5 seconds).
* Accuracy – Precision of the load cell sensor in detecting fluid changes.
* Alert Sensitivity – Defines how quickly alerts are triggered.
* System Uptime – Duration for which the device runs without failure.

**Tools Used**

**1**. **Hardware Tools**

* NodeMCU ESP8266 (Microcontroller)
* HX711 Load Cell Module
* Load Cell Sensor (for weight measurement)
* I2C 16x2 LCD Display
* Buzzer and LED Indicators
* Breadboard and Connecting Wires

**2. Software Tools**

* Arduino IDE – For programming and uploading code
* Blynk IoT App – For real-time monitoring and alerts
* Google Chrome Browser – For accessing the web-based dashboard

**3. Programming Languages**

* C / C++ – For Arduino firmware
* JavaScript (D3.js) – For real-time dashboard visualization

**4. Testing & Debugging Tools**

* Arduino Serial Monitor
* Blynk Console Debugger
* Multimeter (for voltage and connection

**Software Testing**

The IV ALERTX – IoT-Based IV Bag Monitoring and Alert System underwent multiple stages of testing to ensure it functions correctly, safely, and efficiently in clinical environments.

**Testing Types:**

* **Unit Testing:**

Each component (sensor data reading, LCD display, Wi-Fi connectivity, and Blynk communication) was individually tested using Arduino IDE’s serial monitor to verify accuracy and response.

* **Integration Testing:**

The interaction between the ESP8266, HX711 load cell module, LCD, and Blynk IoT platform was validated to confirm that sensor data correctly flows through the system and alerts are triggered when thresholds are reached.

* **System Testing:**

A complete test was performed in a simulated hospital environment. The system was connected to an IV bag setup to ensure accurate detection of weight reduction and timely alert generation on both the LCD and mobile dashboard.

* **User Acceptance Testing (UAT):**

Nurses and healthcare staff tested the prototype and confirmed that alert notifications (visual, sound, and mobile) were clear, timely, and helpful during mock patient monitoring sessions.

* **Performance Testing:**

Verified system speed, network stability, and data refresh rate on the web dashboard using tools such as Blynk Cloud performance metrics and serial debugging logs.

**Implementation**

The system was implemented in three stages:

1. **Hardware Setup:**

* ESP8266 NodeMCU connected to HX711 load cell for weight measurement.
* 16x2 I2C LCD for displaying real-time readings.
* Buzzer and LED for local alerts.
* Blynk IoT app for cloud alerts and data visualization.

**2. Software Setup:**

* Code uploaded using Arduino IDE.
* Blynk platform used to create virtual dashboards.
* Calibrated the load cell with a known IV bag weight for accurate readings.

**3. Deployment:**

* Integrated with hospital Wi-Fi network.
* Live tested with sample saline IV bags to validate alert timing and notification reliability.

**CONCLUSION**

The IV Bag Monitoring System is designed to enhance patient safety and hospital efficiency by integrating real-time tracking, accurate fluid level detection, and automated alerts. By utilizing a well-structured relational database, the system effectively manages patients, IV bags, sensor data, alerts, and medical staff information.

The relationships between tables ensure seamless data retrieval, enabling efficient monitoring and timely intervention. With AI-driven analytics, predictive alerts help prevent issues such as low fluid levels, blockages, and air bubbles, improving overall patient care. Additionally, the use of foreign keys, indexes, and constraints maintains data integrity and security.

**BIBLIOGRAPHY**

[1].Brown, T., & Smith, J. (2020). Advancements in Medical IoT: Real Time Monitoring Systems. Springer.

[2].Gupta, R., & Patel, M. (2019). Smart Healthcare Technologies:

Applications and Innovations. Elsevier.

[3].World Health Organization. (2021). Intravenous Fluid Therapy in Adults and Children. WHO Publications.

[4].Johnson, L., & Kim, S. (2022). “IoT-Based Intravenous Fluid Monitoring System for Healthcare Applications.” Journal of Medical Devices and Technology, 45(3), 102-118.

[5].Ahmed, Z., & Lee, C. (2021). “AI-Powered Smart IV Drip Monitoring: Reducing Human Errors in Healthcare.” International Journal of Biomedical Engineering, 38(2), 56-72.

[6].Mayo Clinic. (2023). “Intravenous Therapy: Benefits, Risks, and Monitoring.” Retrieved from https://www.mayoclinic.org

[7].National Institutes of Health (NIH). (2022). “Smart IV Bag

Monitoring Systems: A Review.” Retrieved from https://www.nih.gov

[8].Arduino Official Website. (2023). “Using Arduino for Medical

Applications.” Retrieved from https://www.arduino.cc

**APPENDICES**

1. **BLOCK DIAGRAM**

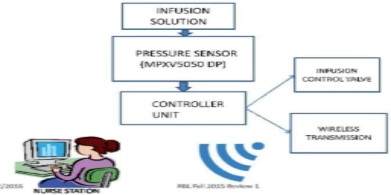


Fig 6. Block Diagram

1. **FLOW CHART**

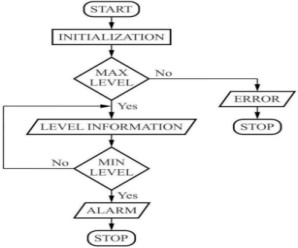


Fig 7. Flow chart

**C. Sample Code**

// ---------- BLYNK TEMPLATE DETAILS (Must be on top) ----------

#define BLYNK\_TEMPLATE\_ID "TMPL3urO08C3U"

#define BLYNK\_TEMPLATE\_NAME "IOT Based IV Alerts" #define

BLYNK\_AUTH\_TOKEN "FN3hiP86uIvYtrCrJ\_Ufm6w2ntPTT5qn"

// ---------- INCLUDE LIBRARIES ----------

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h> // Ensure ESP8266-compatible version // ---------- WI-FI CREDENTIALS ---------- char ssid[] = "Iv\_Wifi"; char pass[] = "divyamurugan02";

// ---------- LCD SETUP ----------

LiquidCrystal\_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16x2 display

// ---------- BLYNK TIMER ----------

BlynkTimer timer;

// ---------- SENSOR SETUP ---------- int sensorPin = A0; // Analog pin connected to fluid level sensor int fluidLevel = 0; int lowThreshold = 200; // Adjust based on your sensor calibration

// ---------- SENSOR DATA FUNCTION ---------- void sendSensorData() { fluidLevel = analogRead(sensorPin); Serial.print("Fluid Level: ");

Serial.println(fluidLevel);

// Send to Blynk (V1 - chart or display widget)

Blynk.virtualWrite(V1, fluidLevel);

// Display on LCD lcd.setCursor(0, 0); lcd.print("Fluid Level: "); lcd.setCursor(0, 1);

lcd.print("Value: "); lcd.print(fluidLevel); lcd.print(" ");

// If below threshold, send alert (V2 - webhook) if (fluidLevel < lowThreshold) {

Blynk.virtualWrite(V2, 1);

Serial.println("Alert: Fluid level

LOW!"); }

}

// ---------- SETUP ---------- void setup() {

Serial.begin(115200);

Serial.println("Serial started");

// Start LCD lcd.begin(); lcd.backlight(); lcd.print("IV Monitor Init");

Serial.println("LCD Initialized");

// Connect to Wi-Fi and Blynk

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

Serial.println("Blynk Initialized");

// Set timer to send data every 5 seconds timer.setInterval(5000L, sendSensorData);

Serial.println("Timer set for sensor data");

}

// ---------- LOOP ---------- void loop() { Blynk.run(); timer.run();

}