VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA – 590018

PROJECT REPORT ON

"SAFELDA: IoT-BASED FALL DETECTION AND PROTECTION FOR AGED PEOPLE"

Submitted by

DANVITHA D	**4DM21IS012**
PRADHVINI M P	**4DM21IS042**
RACHEL JACOB FERNANDES	**4DM21IS043**
THRIPADA P	**4DM21IS058**

UNDER THE GUIDANCE OF

Dr. PRADEEP UDUPA **M.Tech, PhD**

Associate Professor & HOD Dept. Of ISE

In the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

YENEPOYA INSTITUTE OF TECHNOLOGY

N.H.13,Thodar,Vidyanagar,Moodbidri,Mangalore,

Karnataka – 574225

2024-2025

YENEPOYA INSTITUTE OF TECHNOLOGY

N.H.13,Thodar,Vidyanagar,Moodbidri, Mangalore,Karnataka-574225

(Affiliated to Visvesvaraya Technological University)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Major Project Work entitled "SAFELDA IoT-BASED FALL DETECTION AND PROTECTION FOR AGED PEOPLE" carried by **MS. DANVITHA D (4DM21IS012), Ms. PRADHVINI M P (4DM21IS042), Ms. RACHEL JACOB FERNANDES (4DM21IS043), Ms. THRIPADA P (4DM21IS058),** bonafied student of **Yenepoya Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering in Information Science & Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2024-2025.It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in all report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

....	....	....
Dr. Pradeep Udupa	**Dr. Pradeep Udupa**	**Dr. Prabhakara B K**
Internal Guide	**Head of the Department**	**Principal**
Dept of ISE	**Dept of ISE**	**YIT**

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

The successful completion of any work would be incomplete without a mention of the people who made it possible, whose constant guidance and encouragement served as a beacon light and crowned our efforts with success. We owe our gratitude to many people who helped and supported us during our Major Project **"SAFELDA IoT-BASED FALL DETECTION AND PROTECTION FOR AGED PEOPLE"**.

My deepest thanks to our guide **Dr. Pradeep Udupa**, Associate Professor, Dept. of ISE, Yenepoya Institute of Technology for his constant support and encouragement and providing with the necessary advice's and help. We are highly indebted to his for taking keen interest in our work, monitoring and providing guidance throughout the course.

We sincerely express our gratitude to **Dr. Pradeep Udupa**, Associate professor, Dept. of ISE for his constant support and guidance as a project coordinator.

We sincerely express our gratitude to **Dr. Pradeep Udupa,** H.O.D, Dept. of ISE for his constant support and guidance.

We take immense pleasure in thanking my beloved Principal **Dr. Prabhakara B K ,** for his constant support.

We also thank our faculty members who were ready with a positive comment to help us all time, whether it was an off-hand comment to encourage us or a constructive piece of criticism.
At last but not the least we want to thank our classmates and friends who appreciated our work and motivated us.

DANVITHA D	**4DM21IS012**
PRADHVINI M P	**4DM21IS042**
RACHEL JACOB FERNANDES	**4DM21IS043**
THRIPADA P	**4DM21IS058**

DECLARATION

This is to certify that we have followed the guidelines provided by the University & Institute in preparing this Final Project report and whenever we have used materials (data, theoretical analysis, figures and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

DANVITHA D	**4DM21IS012**
PRADHVINI M P	**4DM21IS042**
RACHEL JACOB FERNANDES	**4DM21IS043**
THRIPADA P	**4DM21IS058**

ABSTRACT

For elderly people living alone or without constant supervision, a fall can lead to long periods of immobility, delayed medical attention, and even life-threatening consequences. In many cases, the fear of falling can also lead to reduced mobility and social isolation, impacting both their physical and emotional well-being. Caregivers and family members often worry about the safety of their elderly loved ones, but they cannot always be present to provide immediate help. Therefore, technology that can detect falls and alert caregivers in real-time is essential in ensuring the safety, independence, and quality of life for elderly individuals.

SAFELDA is an IoT-based wearable device designed to detect falls in elderly individuals and provide real-time alerts to caregivers, while also offering vital sign monitoring and GPS tracking. The system leverages machine learning to adapt to the user's movement patterns, reducing false alarms and improving detection accuracy. A unique feature of SAFELDA is its protective airbag mechanism, which deploys during a fall to cushion the impact and prevent serious injury. This project aims to enhance the safety and well-being of elderly individuals by providing a quick-response fall detection system integrated with advanced protective measures.

SAFELDA addresses the critical issue of fall-related injuries among elderly individuals by combining cutting-edge IoT technology with advanced safety features. The system’s wearable device continuously monitors the user’s movements, and in the event of a fall, it automatically triggers an alert that is sent to caregivers or family members, along with the GPS location of the incident. Through its combination of real-time monitoring, personalized detection, and physical protection, SAFELDA offers a comprehensive solution to one of the most pressing safety issues faced by the elderly population.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

The rapid growth of the elderly population globally has raised significant concerns regarding their well-being and safety, particularly in the context of falls, which are one of the leading causes of injury among older adults. Traditional methods of fall detection and monitoring often rely on manual reporting or delayed interventions, which can lead to serious health complications, especially when immediate help is not available. To address this critical issue, this project proposes the development of a Smart Fall Detection System for elderly individuals, incorporating cutting-edge technologies such as the Internet of Things (IoT), real-time monitoring, and machine learning algorithms to enhance both fall detection accuracy and timely response.

The system revolves around a wearable device equipped with advanced sensors that continuously monitor the user's movement patterns, enabling the detection of falls in real-time. Upon detection, the system instantly alerts caregivers or family members, providing them with the elderly person’s exact GPS location for immediate assistance. In addition to fall detection, the device monitors vital signs such as heart rate and body temperature, sending data to a cloud-based platform for ongoing health analysis and tracking. This combination of fall detection and health monitoring ensures comprehensive support for elderly individuals, helping caregivers stay informed of any potential health risks.

A key feature of this project is the integration of machine learning algorithms, which personalize the fall detection system based on each individual’s unique movement patterns. By analyzing historical data and adjusting to specific user behaviors, the system minimizes the occurrence of false alarms, offering a more reliable and accurate solution compared to traditional methods. Moreover, this project includes an innovative protective mechanism in the form of an airbag that deploys upon detecting a fall, cushioning the user and reducing the risk of injury. This airbag mechanism operates similarly to automotive airbags, providing an additional layer of safety by mitigating the impact of a fall.

By combining IoT technology, machine learning, and a proactive safety measure, this project aims to create a robust and effective solution that addresses the pressing issue of fall-related injuries in the elderly. The goal is to provide both elderly individuals and their caregivers with peace of mind, knowing that the system will not only detect falls promptly but also help protect the user from further harm.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

A literature survey involves conducting a comprehensive review and analysis of existing published research, scholarly articles, books, patents, and other relevant sources related to the specific field or topic of the project. It aims to identify the current state of knowledge, gaps in research, methodologies used, and findings relevant to the project's objectives. The survey helps in understanding existing solutions, technologies, and approaches, guiding the project's direction, and ensuring it contributes novel insights or advancements in the field.

Smart Shield: An IoT-Based Fall Detection, GPS Tracking, and Health Monitoring System Using ESP8266"* (December 13, 2024): This study presents a wearable device designed to enhance personal safety and health monitoring. It incorporates a fall detection system powered by the MPU6050 sensor and a custom-built dataset generated from real-life fall and non-fall scenarios. The device detects falls accurately and mitigates false alarms by allowing users to cancel alerts within 20 seconds via an emergency button. In the event of a confirmed fall, the system triggers SMS alerts containing real-time GPS coordinates, ensuring continuous location tracking.

The Smart Belt is designed to detect falls accurately using the MPU6050 sensor and a machine learning model trained on a custom dataset. This dataset was created by collecting real-life fall and non-fall scenarios, ensuring high precision in distinguishing genuine falls from false alarms. A unique feature of the device is its buzzer alert system, which provides a 20-second window to cancel false detections via an emergency button.

The methodology involves multiple stages, starting with the implementation of a fall detection system using the MPU6050 accelerometer and gyroscope sensor, which continuously monitors motion and orientation changes. A custom dataset consisting of real-life fall and non-fall scenarios was created to train a machine learning model, which was then deployed on an ESP8266 microcontroller to enable real-time fall detection. To minimize false alarms, the system includes a buzzer alert that activates upon fall detection, allowing users to cancel the alert within 20 seconds using an emergency button if the detection is incorrect.

For emergency response, the system integrates a GPS module to track the user's real-time location. When a fall is confirmed, SMS alerts containing GPS coordinates are sent to caregivers via IFTTT integration, ensuring prompt assistance. Additionally, a geofencing feature is incorporated to notify caregivers if the user moves beyond a predefined safe zone. Alongside fall detection, the system also provides health monitoring capabilities through a MAX sensor, which measures SpO_2 (oxygen saturation) and heart rate in real time. These health parameters are displayed on an OLED screen when the user interacts with the sensor, allowing for continuous tracking of vital signs.

Additionally, the device incorporates a MAX sensor to monitor SpO_2 (oxygen saturation) and heart rate, with real-time readings displayed on an OLED screen for easy access. Designed for convenience, the Smart Shield is powered by a rechargeable lithium battery with USB Type-C charging, offering long-term usability in a lightweight and wearable form factor. Leveraging IoT and cloud integration, the system enables seamless data transmission via an ESP8266 microcontroller, allowing caregivers to monitor health trends remotely. This innovative combination of fall detection, health tracking, and emergency response makes Smart Shield a reliable and efficient safety solution for elderly individuals.

The IoT-Based System for Real-Time Fall Risk Assessment and Health Monitoring (December 2, 2024): This research introduces an IoT-based system designed to provide real-time fall risk assessment and monitor health parameters using wearable sensors. The system integrates the MPU6050 sensor with IoT technology for efficient data collection and analysis. A Random Forest algorithm is employed to process the complex health data, offering precise and reliable fall detection models. The study identifies the waist as the optimal sensor placement, achieving up to 97.9% accuracy in detecting mild falls while standing.

The research article titled “Wearable sensors for remote health monitoring” Mondal T, Majumder S, and Deen M[6] identify an issue caused by an aging population, underlining the need for novel approaches to meeting healthcare and social welfare demands. Faisal AI, Majumder S, Subramaniam S, and Deen MJ [7] describe a health monitoring system that builds on existing solutions and research.

Tracking the health of your feet, particularly irregularities in plantar pressure, activity, and gait, can aid in disease identification and rehabilitation. Visvanathan R. and Khow KSF [8] documented cases of falls among elderly people. Assesses current understanding of falls, with a focus on fast screening, evaluation, and community fall prevention activities[9].

Fievez D, Jojczyk L, Grenard R, Buisseret F, Catinus L, and Bar Vaux V [10] used wearable inertial sensors in timed up-and-go and six-minute walking tests to predict the fall risk of senior nursing home residents. Mun J, Kim B, Kim H,Heo H, Sim T, and Cates B [11] suggested a novel detection model and its optimum characteristics for categorizing falls in daily activities with low and high acceleration utilizing an insole sensor system control.

IoT-Based Safety Management for Elderly People (November 25, 2022): This project focuses on developing a smart wearable device using multiple IoT sensors to detect falls and monitor health conditions in elderly individuals. The system tracks heartbeat, temperature, and movement using sensors like accelerometers. If abnormal health values or a fall are detected, an alert call is made to the appropriate caretaker or family member using a GSM module.

The IoT-based safety management system for elderly people begins with the initialization and activation of IoT sensors embedded in a wearable device. A SIM card is inserted into the GSM module to establish a network connection with the telecom service provider. Once the device is worn, various sensors continuously monitor the health vitals of the user. The heartbeat sensor tracks the heart rate, while the DHT11 sensor records temperature and humidity levels. An infrared (IR) sensor is used to detect obstacles, and an accelerometer measures acceleration along the x, y, and z axes. Using these readings, the system calculates pitch and roll values with predefined formulas. If these values exceed a threshold of ±50, the system detects a fall event.

The decision-making process involves analyzing health parameters and acceleration values to identify abnormal conditions. If any vital signs deviate from normal levels or a fall is detected, an alert is triggered. The GSM module then sends an SMS or initiates a call to the designated caretaker or family member, ensuring immediate response.

CHAPTER 3

SYSTEM REQUIREMENT ANALYSIS AND SPECIFICATIONS

CHAPTER 3

SYSTEM REQUIREMENT ANALYSIS AND SPECIFICATIONS

3.1 Functional Requirements:

Functional requirements define the specific actions and behaviors a system must perform to meet user needs. They outline what the system should do, focusing on its core functionalities, without specifying implementation details.

3.1.1 Fall Detection:

- The system must accurately detect falls using sensors (e.g., accelerometer, gyroscope) embedded in a wearable device.
- The fall detection algorithm should identify sudden, abnormal movements that indicate a fall event, distinguishing between falls and other daily activities.
- The system should automatically trigger an alert to caregivers or family members once a fall is detected.

3.1.2 Real-Time GPS Location Tracking:

- Upon detecting a fall, the system must send the user's real-time GPS coordinates to the caregivers or designated emergency contacts.
- The GPS system should be accurate and reliable for outdoor usage, with a minimum accuracy of 5 meters.

3.1.3 Vital Signs Monitoring:

- The wearable device should continuously monitor and transmit vital signs such as heart rate, body temperature, and possibly blood oxygen levels.
- The system should notify caregivers if the monitored vital signs deviate from predefined safe thresholds, indicating potential health issues.

3.1.4 Alerting System:

- When a fall is detected, caregivers or family members should receive instant notifications via a mobile app, SMS, or email.

3.2 Non-Functional Requirements:

Non-functional requirements (NFRs) specify how a system should perform rather than what it should do. They encompass aspects such as reliability, performance, usability, security, scalability, maintainability, availability, portability, interoperability, and compliance. These requirements are essential for ensuring the overall quality and effectiveness of a system, guiding decisions throughout the development lifecycle.

.

3.2.1 Reliability and Availability:

The system must operate reliably in real-time, with a high level of accuracy in detecting falls and monitoring vital signs.

3.2.2 Performance:

The system must operate with minimal latency, especially for fall detection and alert generation, ensuring that notifications are sent in real-time.

3.2.3 Scalability:

The system should be scalable to accommodate multiple users, allowing caregivers to monitor several elderly individuals at once.

3.2.4 User Experience:

The mobile app should be intuitive, with large, easy-to-read buttons and clear alert notifications to ensure caregivers can respond promptly.

3.4 Hardware Requirements:

3.3.1 Minimum Configuration:

Processor: Intel Core i5

RAM: 8GB

Storage: 100GB HDD

Storage: SSD for faster performance

GPU: NVIDIA RTX 2060+ (for AI-based voice processing)

3.3.2 Recommended Configuration:

Processor: Intel Core i7

RAM: 16GB

Storage: SSD for faster performance

GPU: NVIDIA RTX 2060+ (for AI-based voice processing)

3.3.3 ESP32 Microcontroller:

- **Model:** ESP32-WROOM-32 (or similar)
- **Memory:** 4MB Flash, 520KB SRAM.
- **I/O Pins:** At least 10 GPIOs (for interfacing with QR scanner and motor**)**.

3.3.4 Accelerometer (ADXL345 or MPU6050):

- Detects rapid changes in movement, orientation, and acceleration to determine falls.

3.3.5 Gyroscope (MPU6050 or LSM6DS3):

- Helps differentiate between regular movements and sudden falls by tracking angular velocity

3.3.6 Heart Rate Sensor (MAX30100 or MAX30102):

- Monitors the user’s heart rate before and after a fall to detect possible health risks.

3.3.7 Mini Airbag System (Custom-made):

- Deploys automatically when a fall is detected.
- Uses a micro gas inflator (CO_2 cartridge) for rapid inflation.
- Controlled via a **MOSFET circuit** triggered by the microcontroller.

3.5 Software Requirements:

3.4.1 Mobile Application (For Caregivers and Family Members):

A mobile application is required for monitoring real-time fall alerts, viewing GPS location, and checking health status.

Features:

- **Fall Alerts:** Receive instant notifications.
- **GPS Tracking:** View real-time location of the elderly person.
- **Health Monitoring:** Display heart rate, body temperature, and activity data.
- **User Authentication:** Secure login for caregivers.
- **Settings Panel:** Adjust fall detection sensitivity and notification preferences.

3.4.2 Cloud Platform and Backend Development:

- Node.js with Express.js (for handling API requests).
- Python with Flask or FastAPI (alternative for data processing)

3.4.3 Cloud Database:

- **Service**: Firebase Firestore or AWS DynamoDB.

CHAPTER 4

PROPOSED SYSTEM

CHAPTER 4
PROPOSED SYSTEM

The IoT-Based Fall Detection and Assistance System is designed to provide a real-time, intelligent, and proactive safety solution for elderly individuals, ensuring quick emergency response and injury prevention. The system consists of a smart wearable device equipped with advanced motion sensors, a GPS tracking module, a health monitoring unit, and an airbag protection mechanism. By utilizing machine learning algorithms, the system adapts to the user's unique movement patterns, significantly reducing false alarms while accurately detecting falls. Upon detecting a fall, the system triggers an automatic alert to caregivers or emergency contacts via a mobile application, along with the real-time GPS location of the elderly individual. Additionally, the system monitors vital health parameters such as heart rate and body temperature to assess post-fall conditions and potential health risks.

The wearable device integrates an accelerometer and gyroscope to continuously track body movements and detect sudden falls based on impact force and orientation changes. To enhance safety, the system includes a mini airbag mechanism, which deploys instantly upon detecting a fall, cushioning the impact and minimizing injuries. The device is powered by an ESP32 microcontroller, which enables seamless Wi-Fi and Bluetooth communication, allowing real-time data transmission to a cloud-based server for remote monitoring. A machine learning model processes motion data and differentiates between accidental falls and normal activities like sitting or bending, improving detection accuracy.

A dedicated mobile application enables caregivers and family members to receive instant alerts when a fall is detected. Through Google Maps integration, caregivers can track the elderly person's precise location and assess their condition via the health monitoring dashboard. The application also allows users to configure system settings, such as adjusting fall detection sensitivity, emergency contact preferences, and alert modes (e.g., push notifications, SMS, or automated calls). The backend system, developed using Node.js and Firebase, ensures secure data storage, real-time synchronization, and encrypted communication between the wearable device and the caregiver’s mobile app.

Incorporating AI-driven predictive analysis, the system can identify early warning signs of potential falls based on irregular movement patterns and health fluctuations, offering preventive measures before an incident occurs. Furthermore, the device is designed for lightweight, comfortable, and long-lasting use, featuring a rechargeable Li-ion battery with power optimization techniques to ensure extended battery life. With a user-friendly interface, automated alerts, and intelligent fall detection, this IoT-enabled solution revolutionizes elderly care by enhancing safety, reducing response time, and preventing fall-related injuries.

4.1 Key Features and Functionality:

- **Real-Time Fall Detection**

  The system uses an accelerometer and gyroscope to detect sudden falls. Machine learning algorithms help distinguish between actual falls and normal activities, reducing false alarms.

- **Instant Emergency Alerts**

  When a fall is detected, an automatic alert is sent to caregivers through SMS, mobile notifications, or calls. The alert includes real-time GPS location data to ensure quick assistance.

- **GPS Tracking for Location Monitoring**

  The wearable device is equipped with a GPS module that provides real-time location updates of the elderly individual. Caregivers can track the user's position using a mobile application with Google Maps integration.

- **Protective Airbag Mechanism**

  The system features a mini airbag that deploys automatically upon detecting a fall, reducing impact injuries. This mechanism works similarly to vehicle airbags, ensuring enhanced safety for the user.

4.2 Benefits and Implications:

- The system provides real-time fall detection and emergency response, reducing the risk of severe injuries and ensuring timely assistance. The protective airbag mechanism further minimizes impact injuries, improving overall safety.

- Elderly individuals can move freely without fear of being unattended in case of a fall. The system enhances their confidence, allowing them to maintain an active lifestyle while ensuring they are monitored for safety.

- Caregivers receive real-time updates about the elderly person's health and location, allowing them to provide better care and respond efficiently in emergencies. Remote monitoring reduces the burden of constant physical supervision.

- Continuous monitoring of vital signs helps in detecting early signs of health deterioration. Historical data stored in the cloud allows medical professionals to analyze trends and take preventive actions to avoid future falls.

CHAPTER 5

SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

System design is the process of defining the structure, components, interfaces, and data of a system to meet specified requirements. It involves creating a blueprint that outlines how the system will be structured and how its components will interact.

5.1 System Architecture:

5.1.1 Wearable Device Layer (Edge Layer):

The wearable device consists of an ESP32 microcontroller integrated with motion sensors such as an accelerometer and gyroscope to detect sudden falls. It also includes heart rate and temperature sensors for monitoring vital signs. The airbag deployment mechanism is embedded within the device to cushion the impact of a fall. The device continuously collects data from sensors, processes it locally, and transmits alerts when a fall is detected.

5.1.2 Communication Layer:

The system uses Wi-Fi or Bluetooth for seamless data transmission between the wearable device and the cloud server. The ESP32 module ensures real-time connectivity, allowing motion and health data to be sent instantly when a fall occurs. The communication layer plays a crucial role in ensuring that alerts and GPS location data reach caregivers without delay.

5.1.3 Cloud-Based Processing and Storage Layer:

A cloud server, such as Firebase or a Node.js backend, is responsible for securely storing all collected data. It processes fall detection signals and vital sign readings while using machine learning algorithms to differentiate between actual falls and false alarms. This layer also maintains historical data logs for further medical analysis and predictive healthcare insights.

5.1.4 Application Layer (User Interface):

The mobile application serves as the main interface for caregivers, displaying real-time fall alerts, GPS tracking, and health status updates. It allows caregivers to customize alert settings, view past incidents, and monitor trends in the elderly person’s activity and health conditions. Designed for ease of use, the app ensures quick response times in emergencies.

5.1.5 Cloud Database:

Data such as expiry dates, food item types, and sorting status is stored securely in a cloud database. This allows for easy retrieval, updating, and synchronization of data across different devices. Popular cloud platforms such as AWS, Google Cloud, or Firebase may be used for secure data storage

5.1.6 Backend API:

The backend is responsible for managing the interactions between the app, cloud database, and hardware components. It is developed using technologies like Node.js, Python (Flask/Django), or similar backend frameworks to provide the following

5.1.7 Notification System:

he notification system sends alerts to users via the mobile app when food items are nearing their expiry date. It may use services like Firebase Cloud Messaging (FCM) or Apple's Push Notification Service (APNS) to deliver timely reminders.

CHAPTER 6

SYSTEM IMPLEMENTATION

CHAPTER 6

SYSTEM IMPLEMENTATION

The **SAFELDA** involves a series of well-defined steps for system implementation, ensuring seamless integration between hardware components, cloud storage, and the mobile application. Below is a detailed implementation guide outlining the steps involved in developing the system.

6.1 Hardware Setup:

The system begins with assembling the wearable device, which includes an ESP32 microcontroller, an accelerometer, a gyroscope, heart rate and temperature sensors, a GPS module, and an airbag deployment mechanism. These components are integrated onto a compact PCB to ensure efficiency and portability. The sensors continuously collect real-time motion and health data, which is processed locally by the ESP32 before transmitting it to the cloud.

- **Sensor Calibration and Data Processing:** To ensure accurate fall detection, the accelerometer and gyroscope are calibrated to recognize patterns of movement associated with falls. The system uses predefined thresholds for acceleration and angular velocity to detect abrupt changes in motion. Additionally, vital sign sensors are tested to ensure they provide reliable health data after a fall.
- **Machine Learning Model Training and Deployment:** A machine learning model is trained using motion datasets containing fall and non-fall movements. The model learns individual movement patterns, reducing false alarms by distinguishing normal activities from actual falls. The trained model is deployed in the cloud and integrated with the wearable device for real-time processing.
- **Cloud Integration and Data Management:** The system utilizes cloud storage, such as Firebase or AWS, to store real-time fall detection alerts, GPS locations, and health data. The cloud server processes incoming data and maintains a historical log for analysis. A secure API ensures seamless data transfer between the wearable device, mobile application, and caregiver dashboard.
- **Mobile Application Development:** A user-friendly mobile application is developed for caregivers, providing real-time notifications, GPS tracking, and health monitoring. The app is built using React Native or Flutter for cross-platform compatibility. Features include emergency alerts, fall history logs, and adjustable sensitivity settings for personalized fall detection.

- **Emergency Alert System Implementation:** When a fall is detected, the system immediately triggers an alert via SMS, mobile notifications, or automated voice calls to registered caregivers. The alert includes real-time GPS location and health status updates, ensuring prompt assistance. Additionally, integration with emergency services is considered for critical situations.

6.2 Software Development:

The mobile app allows users to interact with the system, receive notifications

- **Emergency Alert System Integration**: The emergency alert system is integrated with third-party services (SMS gateways, voice call APIs, etc.) to send notifications to caregivers when a fall is detected. The system also includes an automated call feature, ensuring caregivers receive immediate alerts in case of emergencies. Integration with emergency services or healthcare providers can be added to facilitate quick medical intervention when necessary.
- **Cloud Server and Backend Development**: The cloud server is developed using platforms such as Firebase, AWS, or a custom Node.js backend. It handles real-time communication between the wearable device and mobile application. The server processes fall alerts, manages health data, and stores historical records securely. It also supports real-time data analytics and updates the mobile application with the latest information, including GPS tracking and health monitoring.

6.3 Integration and System Synchronization:

- **Real-Time Data Synchronization Between Cloud and Mobile Application**: The mobile application regularly syncs with the cloud server to display real-time fall alerts, health statistics, and GPS data. Changes in the system, such as sensitivity adjustments or alert configurations, are synchronized between the mobile app and the wearable device to ensure consistency.

- **Event Handling and Notification Synchronization**: When a fall is detected, the wearable device triggers the event and sends the data to the cloud, which then propagates to the mobile app in real time. The system ensures the synchronization of fall events, ensuring caregivers are alerted simultaneously via multiple channels (SMS, app notification, or voice call).

6.4 Code:

```
#include <Wire.h>
#include <WiFi.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM6DS33.h>  // For Accelerometer/Gyroscope
#include <HTTPClient.h>
// Wi-Fi Credentials
const char* ssid = "your_wifi_ssid";
const char* password = "your_wifi_password";

// API URL for cloud (Firebase, Node.js, etc.)
const String cloudUrl = "https://your-cloud-endpoint.com/alert";

// Accelerometer and Gyroscope initialization
Adafruit_LSM6DS33 lsm6ds33;  //

// Variables
float ax, ay, az, gx, gy, gz;
int fallThreshold = 2;

void setup() {
  Serial.begin(115200);

  // Initialize Wi-Fi connection
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");

  // Initialize accelerometer and gyroscope
  if (!lsm6ds33.begin_I2C()) {
    Serial.println("Failed to initialize sensor");
    while (1);
  }
}

void loop() {
  // Read accelerometer and gyroscope data
  sensors_event_t event;
  lsm6ds33.getEvent(&event);
  ax = event.acceleration.x;
  ay = event.acceleration.y;
  az = event.acceleration.z;
  gx = event.gyro.x;
  gy = event.gyro.y;
  gz = event.gyro.z;

  // Fall detection logic based on acceleration
  float acceleration = sqrt(ax*ax + ay*ay + az*az);
  if (acceleration > fallThreshold) {
  Serial.println("Fall detected!");

    // Send fall alert to the cloud
    sendAlertToCloud();
```

```
  }

  delay(1000);  // Delay to reduce load on the device
}

void sendAlertToCloud() {
  HTTPClient http;
  http.begin(cloudUrl);  // Specify the cloud endpoint URL
  http.addHeader("Content-Type", "application/json");

  // Prepare the payload (send data like GPS, timestamp, etc.)
  String payload = "{\"fall\": \"true\", \"location\": \"GPS_data\", \"timestamp\":
\"timestamp_here\"}";

  // Send the POST request
  int httpCode = http.POST(payload);

  if (httpCode > 0) {
    Serial.println("Alert sent successfully");
  } else {
    Serial.println("Failed to send alert");
  }

  http.end();
}

const express = require('express');
const bodyParser = require('body-parser');
```

```
const app = express();
const port = 3000;

// Middleware
app.use(bodyParser.json());

// Route to handle incoming fall alerts
app.post('/alert', (req, res) => {
  const fallData = req.body;

  console.log("Fall detected:", fallData);
  // Handle the data (e.g., send an SMS, email, etc.)
  sendNotificationToCaregiver(fallData);

  res.status(200).send("Alert received");
});

// Simulate sending a notification to the caregiver
function sendNotificationToCaregiver(data) {
  // This would be integrated with an SMS or email service
  console.log(`Sending alert to caregiver: Fall detected at ${data.timestamp}`);
}

// Start the server
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});

import React, { useState, useEffect } from 'react';
```

```
import { View, Text, Button } from 'react-native';
import { Notifications } from 'react-native-notifications';

export default function App() {
  const [alert, setAlert] = useState('');

  useEffect(() => {
    // Listen for incoming alerts (this could be via WebSockets or HTTP polling)
    Notifications.events().registerNotificationReceivedForeground((notification) => {
    setAlert(`Fall detected at ${notification.payload.timestamp}`);
    });
  }, []);

  const simulateFallAlert = () => {
    // Simulate receiving a fall alert
    setAlert('Fall detected at 2025-01-30 14:25:00');
  };

  return (
    <View style={{ padding: 20 }}>
      <Text>{alert || 'Waiting for alerts...'}</Text>
      <Button title="Simulate Fall Alert" onPress={simulateFallAlert} />
    </View>
  );
}
```

6.5 Circuit Diagram:

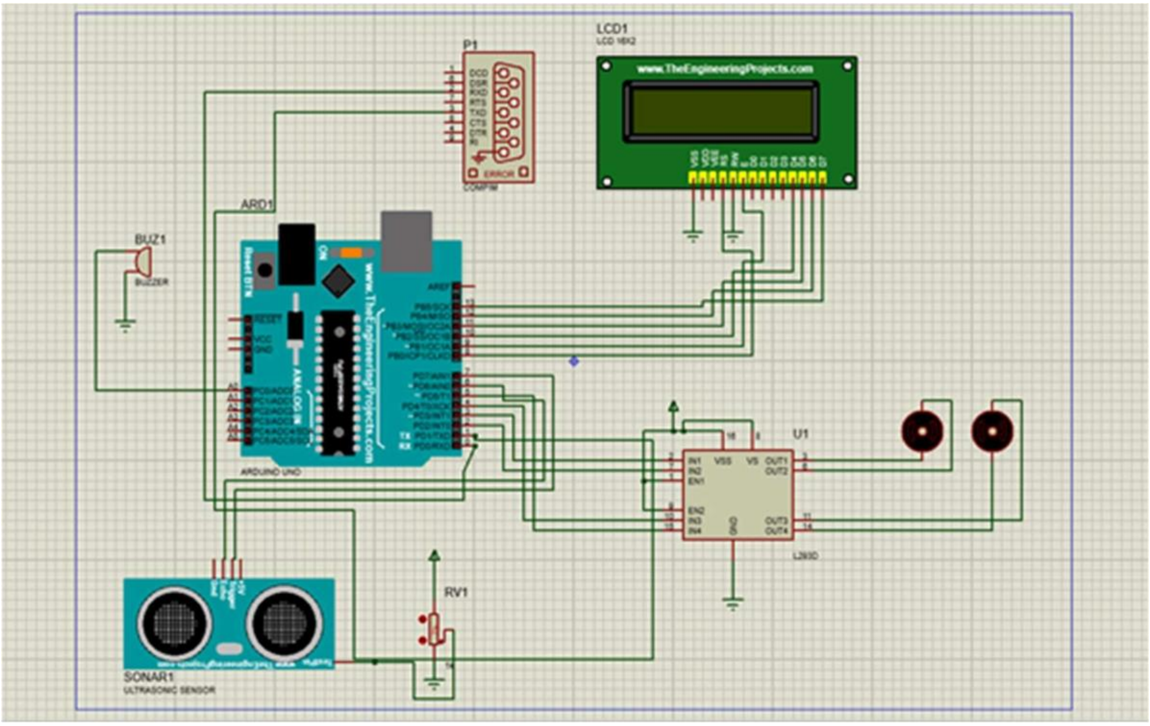

Fig 6.5: Circuit Diagram of SAFELDA Band

CHAPTER 7

SOFTWARE TESTING

CHAPTER 7

SOFTWARE TESTING

Software testing isn't just about finding bugs – it's a comprehensive examination to guarantee a software application or system functions flawlessly. By running the software through its paces, testers aim to uncover issues and verify it behaves as expected in various situations. The ultimate Objective is to identify anything that might hinder the software's functionality, reliability, performance, and, of course, how users experience it.

There's a whole toolbox of testing methods available, each with a specific purpose within the software development lifecycle. Here are some of the most common ones:

7.1 Unit Testing: Zooming in on the Building Blocks:

- This method focuses on testing individual software components or modules in isolation.
- Usually performed by developers during coding, with the help of testing frameworks.
- It helps catch defects early on and ensures each component functions as intended.

7.2 Integration Testing: Making Sure Everything Plays Nice:

- Here, the focus shifts to how different software modules or components interact.
- The goal is to verify that integrated components work together seamlessly and communicate properly.
- This helps identify problems related to data flow, compatibility between interfaces, and overall system integration.

7.3 System Testing: The Big Picture Examination:

- This method evaluates the entire software system as a whole, ensuring it meets all the specified requirements.
- System testing assesses the software's functionality, performance, reliability, and security in scenarios that mimic real-world use.
- It helps validate that the software meets user expectations and performs as intended in its designated environment.

7.4 Acceptance Testing: The User's Seal of Approval:

- This step confirms whether the software meets user requirements and is ready for deployment.
- It involves testing the software with end-users or stakeholders to ensure it fulfills their needs and expectations.
- This often includes User Acceptance Testing (UAT), where users directly interact with the software to verify its usability, functionality, and suitability for their tasks.

7.5 Regression Testing: Keeping a Watchful Eye on Progress:

- This method ensures that changes or updates to the software don't introduce new problems or bring back old ones (regressions).
- Testers re-run previously executed test cases to guarantee that existing functionality remains unaffected by code modifications.
- Regression testing helps maintain software quality and stability across multiple releases or iterations.

7.6 Performance Testing:

- This method evaluates the software's performance under various conditions, including heavy workloads, stress situations, and scalability.
- Performance testing measures response times, throughput, and resource utilization to identify bottlenecks and optimize system efficiency.
- It helps ensure the software can handle expected user loads and maintain acceptable performance levels.

7.7 Security Testing: Building a Digital Fortress:

- This method identifies vulnerabilities and weaknesses in the software that could be exploited by attackers.
- Security testing focuses on finding vulnerabilities such as injection attacks, cross-site scripting (XSS), and authentication bypasses.
- It helps protect sensitive data and ensure compliance with security standards and regulations.

7.8 Software Testing For The Model:

Test Type	Test Description	Expected Outcome	Tools/Methods
Unit Testing	Test individual functions or modules in isolation (e.g., fall detection algorithm, notification function).	Functions should execute as expected and produce correct outputs.	Jest, Mocha (for JavaScript), Unit testing libraries
Integration Testing	Test interaction between modules (e.g., wearable device with cloud server, mobile app with backend).	Components should interact correctly without errors or data loss.	Postman, Insomnia, Integration testing libraries
System Testing	Test the entire system (wearable, cloud, mobile app) for end-to-end functionality.	The entire system should work as intended, with no failures in any component.	Selenium, Appium, Manual testing
Performance Testing	Test the system under load (e.g., multiple fall alerts or high sensor data rate).	The system should handle high traffic or data loads without crashing.	Apache JMeter, LoadRunner
Security Testing	Test the system for vulnerabilities (e.g., unauthorized access, data breaches).	Sensitive data should be encrypted, and there should be no security flaws.	OWASP ZAP, Burp Suite, Manual penetration testing
Usability Testing	Test the mobile app interface for ease of use by caregivers.	The mobile app should be intuitive, and caregivers should easily understand the alerts.	User testing, Focus group sessions
Acceptance Testing	Test the system's functionality based on user requirements.	The system should meet the specified requirements and user expectations.	Manual testing, Client feedback

Table 7.8.1: Software testing.

Test Type	Test Description	Expected Outcome	Tools/Methods
Sensor Calibration	Test accelerometer, gyroscope, and heart rate sensors for correct measurements and calibration.	Sensors should return accurate and reliable data within defined ranges.	Multimeter, Calibration software
Power Consumption	Test battery life under different conditions (e.g., normal use, high power consumption during fall detection).	The system should have an optimal battery life to last throughout the day.	Power meter, Battery performance tests
Airbag Deployment	Test the airbag mechanism for correct deployment during a simulated fall.	The airbag should deploy immediately and fully when a fall is detected.	Test rig, Manual fall simulations
Connectivity Testing	Test Wi-Fi or Bluetooth connection stability between the wearable device and the cloud/mobile app.	Device should maintain stable connectivity with minimal drops.	Wi-Fi analyzer, Bluetooth tester
Temperature Testing	Test the wearable device under various temperature conditions to ensure reliability.	The device should function normally across a wide temperature range.	Temperature chamber, Thermocouple
Drop Test	Test the device's physical durability by simulating a drop or impact.	The device should survive drops without damage or failure in function.	Drop test apparatus, Visual inspection
Waterproof Testing	Test the device's resistance to water or moisture (if applicable).	The device should function properly when exposed to moisture or water.	Water immersion test, Spray test
Sensor Accuracy	Test sensors (accelerometer, gyroscope, heart rate) for their accuracy in real-world scenarios.	Sensors should accurately detect falls, movements, and vital signs.	Test environment, Data logging tools

Table 7.8.2: Hardware testing

CHAPTER 8

RESULT AND IMPLEMENTATION

CHAPTER 8

RESULTS AND SNAPSHOTS

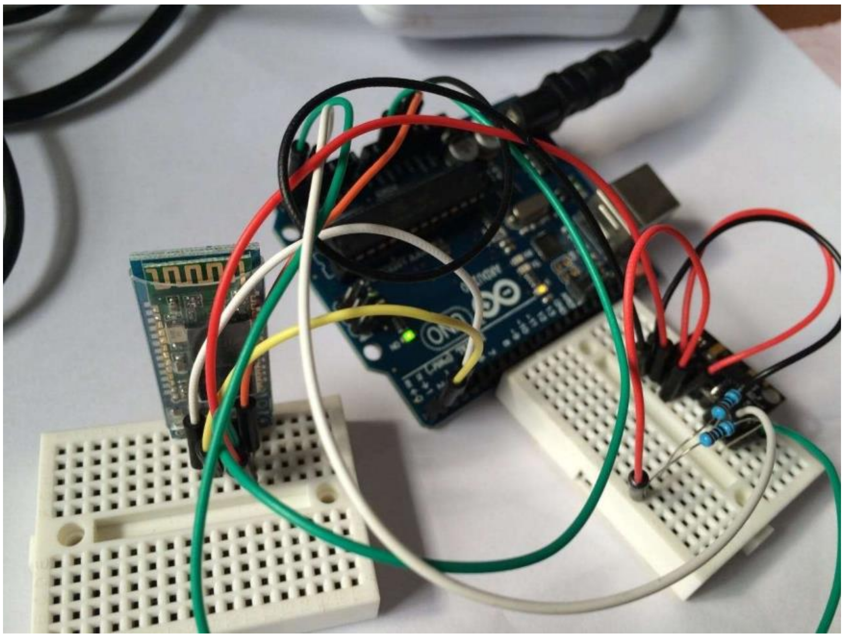

Fig 8.1: SAFELDA Connections.

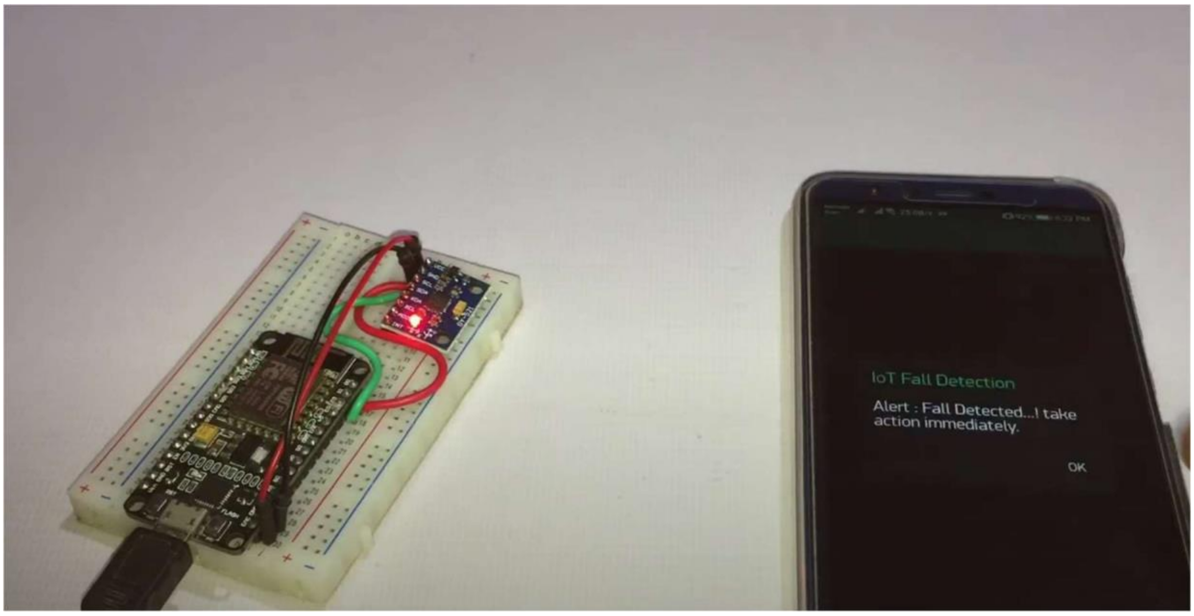

Fig 8.2: Fall detected notification.

The Fig 8. 1 and Fig 8.2 above shows the Design and implementation of Safelda: Iot-Based Fall Detection and Protection For Aged People

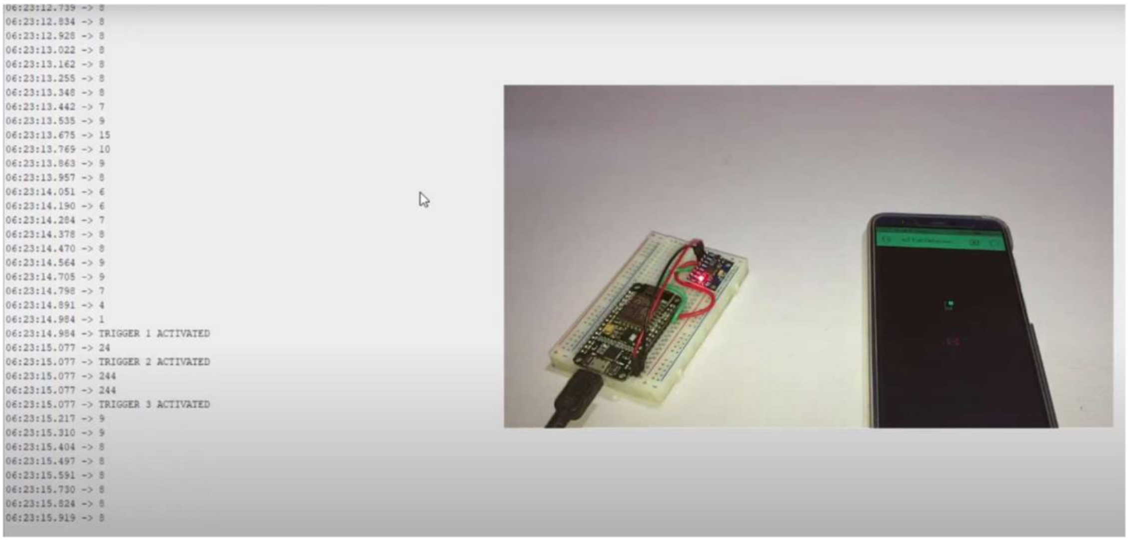

Fig 8.3: Software trigger activation.

The Fig 8.3 above shows the fall detection triggering in software and sending notification

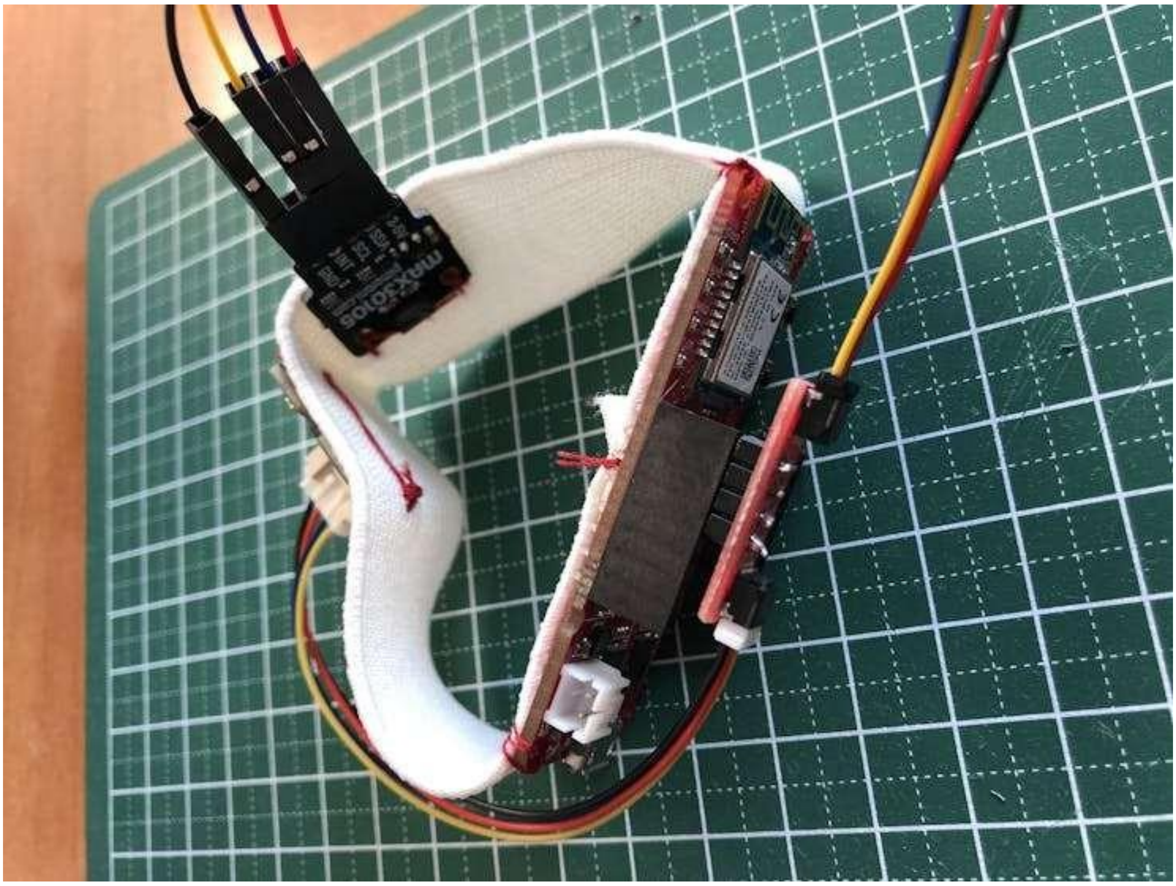

Fig 8.4: SAFELDA Band.

The Fig 8.4 above shows how the system has been implemented to wrist band or arm band

CHAPTER 9

CONCLUSION AND FUTURE WORK

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion:

In conclusion, the IoT-based Fall Detection System for elderly people represents a significant step forward in enhancing the safety and well-being of elderly individuals. By integrating advanced technologies such as wearable devices, real-time monitoring, machine learning algorithms, and proactive safety measures like airbag deployment, the system aims to reduce the risk of injury from falls and ensure timely assistance. The combination of sensors, cloud communication, and caregiver notifications provides a comprehensive solution to address the challenges of elderly care. With thorough software and hardware testing to ensure reliability and performance, this system has the potential to greatly improve the quality of life for elderly individuals while providing peace of mind to their families and caregivers. The successful implementation of this project could pave the way for further innovations in healthcare technology, especially in the context of aging populations.

9.2 Future Scope:

Looking ahead, there are several potential enhancements and avenues for further development in the SAFELDA.

- **Integration with Smart Home Systems:** The system can be integrated with existing smart home technology, such as smart lights, thermostats, and voice assistants like Amazon Alexa or Google Assistant. This would allow for better interaction and automation, such as turning on lights or adjusting the thermostat when the system detects a fall or an emergency.
- **Battery Life and Charging Efficiency:** Advances in battery technology can be explored to enhance the battery life of the wearable device. For instance, wireless charging or solar-powered devices could be developed to reduce the need for frequent charging, making the system more convenient and user-friendly.
- **Global Positioning System (GPS) Accuracy Improvement:** Future developments could improve the GPS accuracy, particularly in remote or urban environments where GPS signals may be weaker. Enhanced positioning systems could provide more precise location data, ensuring that caregivers or emergency services can respond more quickly.

REFERENCES

[1] Aslan, M., Sengur, A., Xiao, Y., Wang, H., Ince, M. C., and Ma, X. (2015). Shape feature encoding via fisher vector for efficient fall detection in depth-videos. Applied Soft. Comput. 37, 1023–1028. doi: 10.1016/j.asoc.2014.12.035

[2] Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., and Meunier, J. (2011). Fall detection with multiple cameras: an occlusion-resistant method based on 3-D silhouette vertical distribution. IEEE Trans. Inform. Technol. Biomed. 15, 290–300. doi: 10.1109/TITB.2010.2087385

[3] Aziz, O., Musngi, M., Park, E. J., Mori, G., and Robinovitch, S. N. (2017). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. Med. Biol. Eng. Comput. 55, 45–55. doi: 10.1007/s11517-016-1504-y

[4] Namba, T., and Yamada, Y. (2018b). Risks of deep reinforcement learning applied to fall prevention assist by autonomous mobile robots in the hospital. Big Data Cogn. Comput. 2:13. doi: 10.3390/bdcc2020013

[5] Stone, E. E., and Skubic, M. (2015). Fall detection in homes of older adults using the Microsoft Kinect. IEEE J. Biomed. Health Inform. 19, 290–301. doi: 10.1109/JBHI.2014.2312180

[6] Sucerquia, A., López, J., and Vargas-Bonilla, J. (2018). Real-life/real-time elderly fall detection with a triaxial accelerometer. Sensors 18:1101. doi: 10.3390/s18041101

[7] Thilo, F. J., Hahn, S., Halfens, R. J., and Schols, J. M. (2019). Usability of a wearable fall detection prototype from the perspective of older people-a real field testing approach. J. Clin. Nurs. 28, 310–320. doi: 10.1111/jocn.14599

[8] https://www.ijariit.com/manuscript/smart-shield-an-iot-based-fall-detection-gps-tracking-and-health-monitoring-system-using-esp8266/

[9] https://www.ijert.org/iot-based-safety-management-for-elderly-people

[10] https://ojs.wiserpub.com/index.php/JEEE/article/view/5472