IE2090
Professional Engineering Practice and Industrial Management

2^{nd} Year, 2^{nd} Semester

Final Report

SMART BACKPACK with IoT

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

18/06/2024

Declaration

We declare that the this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

Project Title	SMART BACKPACK with IoT
Project ID	PEP_02

Group Members

Reg. No	Name	Signature
IT22160652	A.M.R.P.Aththanayaka	
IT22199362	R.M.H.L.Rathnayake	
IT22089168	P.A.G.S.S.Senadheera	
IT22275660	R.M.A.J.D.Senacirathne	

Abstract

The "SMART BACKPACK" project is a hardware-based innovation aimed at addressing typical issues faced by customers, such as overpacking, lack of security, risk of losing belongings, and the absence of modern technological features in traditional backpacks. By integrating Internet of Things (IoT) functionalities, this backpack ensures enhanced security and comfort for customers.

Problem:Overpacking often causes discomfort and physical strain. Traditional backpacks lack sufficient protection, and losing a backpack can lead to the loss of priceless possessions.. Furthermore, traditional backpacks do not meet the modern consumer's demand for advanced technological features.

Solution:
The SMART BACKPACK uses a number of cutting-edge innovations to overcome these problems.

Weight Detection System: Keeps track of and regulates the weight of the backpack to avoid discomfort and overloading.
Automatic strap lifting : These straps make lifting simpler by offering help when the weight is greater than 3 kg.
Only authorized individuals are permitted access to the backpack's contents due to the fingerprint lock.
GPS Tracker: Allows track the backpack's whereabouts in real time.
Mobile apps: Gives users remote access to the backpack's functions, such as location and weight tracking.

Results:
The SMART BACKPACK combines functionality and security, providing a seamless and stress-free travel experience. The weight detection system prevents overloading by alerting users and limiting the lifting mechanism at a 5 kg maximum. The fingerprint sensor ensures only authorized access, enhancing security. The GPS tracker allows users to track the backpack's location in real-time, reducing the risk of losing valuable things. The mobile app enhances user experience by providing an interface for monitoring the backpack's features remotely.
After undergoing rigorous testing, the SMART BACKPACK has shown to be dependable, easy to use, and efficient, which makes it the perfect choice for contemporary users looking for cutting-edge, safe, and cozy travel gear.

Key words

- Fingerprint Reader sensor-biometric device that capture fingerprint
- Stepper motors and drivers - Components used in precise control of rotational motion in machines
- Load cell 5kg -A sensor that measures the force (weight) applied to i
- NodeMCU ESP8266- A compact and versatile microcontroller unit featuring the ESP8266 Wi-Fi module

Acknowledgement

We would like to express our sincere appreciation to everyone who helped us during the "SMART BACKPACK" project.

We extend our sincere thanks to Mrs. Narmada Gamage for providing us with the opportunity to undertake this project.

Special thanks to our supervisors and to all the instructors and lecturers who provided their expert guidance and valuable feedback.

We also acknowledge the hard work and dedication of our group members:

A.M.R.P.Aththanayaka
R.M.H.L.Rathnayake
P.A.G.S.S.Senadheera
R.M.A.J.D.Senacirathne

Finally, we are grateful to our friends and family for their unwavering support and encouragement.

Table of Contents

List of Figures

List of Acronyms and Abbreviations

- GPS -Global Positioning System
- IoT- Internet of things
- WiFi- Wireless Fidelity
- Arduino IDE-Integrated Development Environment

1. Introduction

1.1 Problem Statement

The "SMART BACKPACK" is a hardware-based project designed to improve both security and comfort for your everyday travels which is full with innovative IOT features such as ;

- Weight Detection System
- Automatic-Lifting Handles
- GPS Tracker
- Fingerprint Lock

We devised this concept as a comprehensive solution to address the shortcomings found in conventional backpacks.

Problems:

1. Discomfort and Tension from Overpacking
   - Conventional backpacks frequently lack systems to keep track of the weight of the contents. Users might overpack as a result, which could cause discomfort and physical pressure on the shoulders and back. Long-term health problems including back discomfort and poor posture may result from this.

2. Lack of Effective Security Features
   - Most traditional backpacks don't have sophisticated security measures. Usually, they are locked using basic locks or simple zippers that are readily manipulated or broken into. They become open to theft and unwanted access as a result.

3. Risk of Losing Valuable Belongings
   - Laptops, paperwork, and personal possessions are among the crucial and priceless goods that can disappear from a standard bag. A misplaced or stolen backpack is more likely to be permanently lost without a tracking system since it is hard to find.

4. Absence of Modern Features
   - Conventional backpacks lack contemporary technological capabilities that might improve ease and user experience. They lack integrated biometric locks, automated handles, GPS tracking, and weight detecting devices. When compared to their contemporary counterparts, this renders them less secure and less functional.

1.2 Product Scope

The full scope of the Smart Backpack with IoT includes the design, development, deployment, and maintenance of a system that combines numerous IoT devices to enhance the user's travel experience. Actuators, sensors, and secure locking mechanisms are examples of hardware covered in this.

In-Scope

1. Weight Detection System
   - Sensor integration to ensure comfort and reduce strain by monitoring and warning users when the backpack is overpacked.

2. Automatic-Lifting Handles
   - The creation of automatically lifting handles would improve convenience and lessen physical strain on the user when carrying the backpack.

3. GPS Tracker
   - putting in place a GPS tracking system that will let customers trace the whereabouts of their backpack in real time, boosting security and reducing anxiety.

4. Fingerprint Lock
   - installation of a biometric fingerprint lock to improve security by limiting access to the backpack's contents to authorized individuals only.

Out-of-Scope

1. Non-IoT Based Security Systems
   - Does not include non-IoT based security measures or conventional locking methods.

2. Medical Monitoring Systems
- Excludes sophisticated medical monitoring features like medical alert systems or health sensors.

3. Non-Personal Use Applications
- Is only intended for personal use; it is not intended for business or industrial use.

4. Custom Hardware Development
- The scope does not include custom hardware components that go beyond the basic sensors and actuators.

5. Long-Term Data Storage and Analysis
- Does not include comprehensive data analysis and storage options for long-term tracking and observation.

6. Direct Emergency Response Features
- Does not include capabilities that automatically contact emergency agencies or offer direct emergency response or interventions.

1.3 Project Report Structure

The Smart Backpack with IoT project report guarantees clarity and thoroughness throughout its journey by providing a detailed description of its creation, implementation, evaluation, and future prospects.

2. Methodology

2.1 Requirements and Analysis

"The Smart Backpack" is designed to improve daily travels and is outfitted with modern Internet of Things (IoT) technologies. It has an automated weight detecting mechanism that modifies itself to avoid overloading. During commutes, the automated lifting handles offer comfort, and security is guaranteed by the integrated fingerprint lock. While on the go, real-time GPS tracking keeps you connected. This smart backpack is perfect for exploring, traveling, or commuting.

2.1.1 Functional Requirements

Weight Detection System:

1. The weight of the backpack has to be continually monitored.
2. The system should prevent overloading by limiting the capacity to 5 kg.
3. The backpack should automatically adjust when the weight reaches 3 kg.

Automatic Strap Lifting:

1. The backpack should assist the user by automatically lifting the straps when the weight reaches a certain threshold (3 kg).
2. If the weight exceeds the backpack’s maximum capability (5 kg), the system should limit lifting assistance

Fingerprint Sensor:

1. The backpack must include a fingerprint sensor for secure access control.
2. Only authorized users should be able to access the backpack’s contents.

GPS Tracking:

1. The backpack should integrate a GPS tracker.
2. Users can track the backpack’s location in real-time through a linked mobile app.
3. Remote tracking and communication with the backpack should be possible via the app.

Mobile Application Interface:

The linked mobile app should allow users to:

- Check the backpack's weight.
- Monitor its location.
- Communicate with the backpack remotely.

Integration and Connectivity:

1. The backpack can function independently but should also integrate into larger systems (e.g., via Wi-Fi).
2. An overview of subsystems, linkages, and external interfaces would enhance understanding of the backpack's design and operation.

2.1.2 Non Functional Requirements

1. **Usability:**
    - The backpack interface should be intuitive and user-friendly.
    - Minimal training should be required for users to operate it effectively.
2. **Security:**
    - The fingerprint sensor must be reliable and resistant to unauthorized access.
    - Data transmission (e.g., GPS coordinates) should be encrypted for privacy.
3. **Reliability:**
    - The weight detection system and lifting mechanism should work consistently.
    - The GPS tracker should provide accurate location information.
4. **Performance:**
    - The system should respond quickly to weight changes and lifting requests.
    - The mobile app should load and display data promptly.

2.1.3 Use case diagram

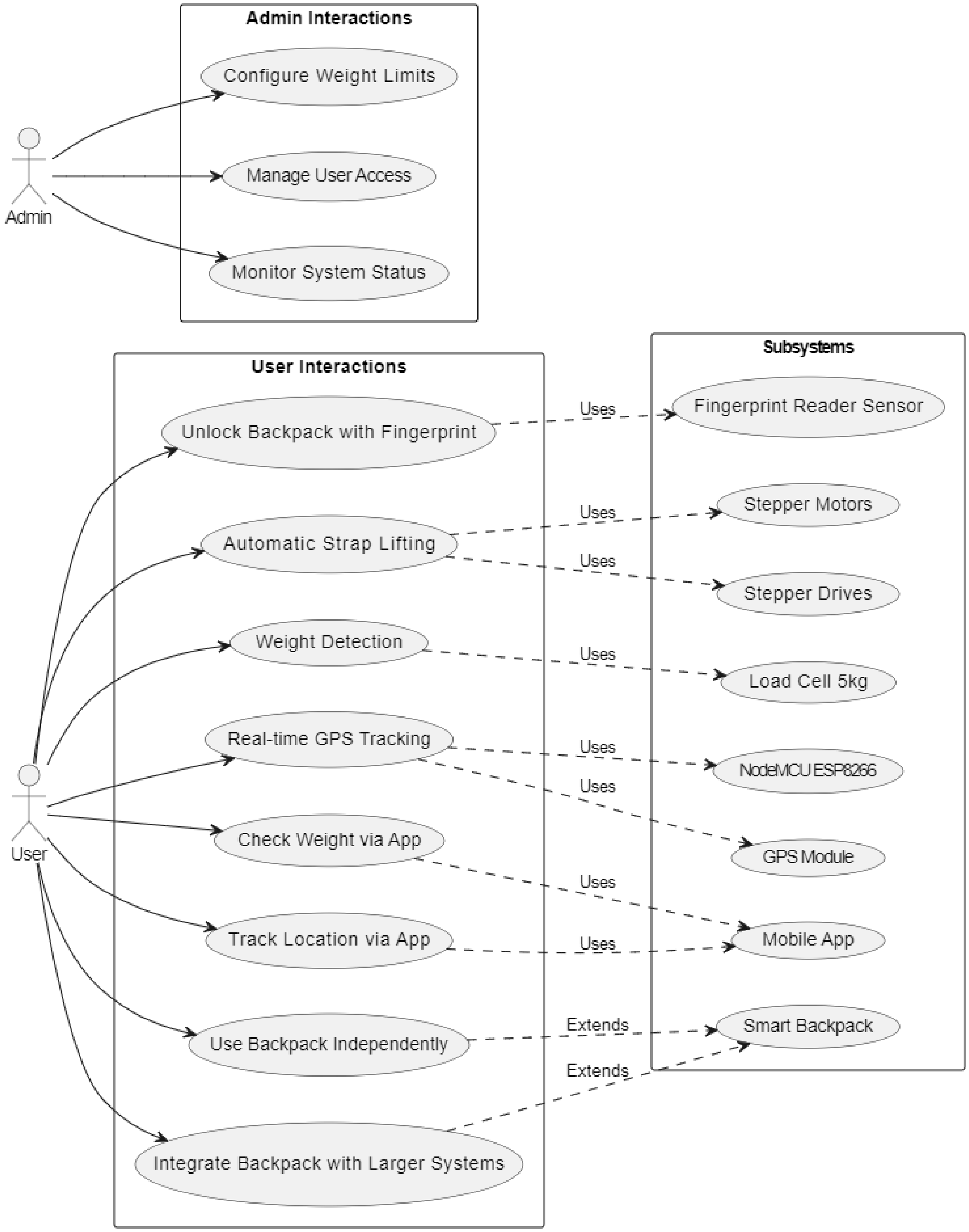


Figure 2.1.3: Use case Diagram

2.2. Design

2.2.1 Activity Diagram

The SMART BACKPAK activity diagram illustrates user-friendly features: fingerprint unlocking, weight monitoring with automatic handle lifting for loads up to 5kg, GPS tracking via NodeMCU ESP8266, and mobile app integration for remote monitoring. It highlights convenience and connectivity through WiFi for enhanced functionality.

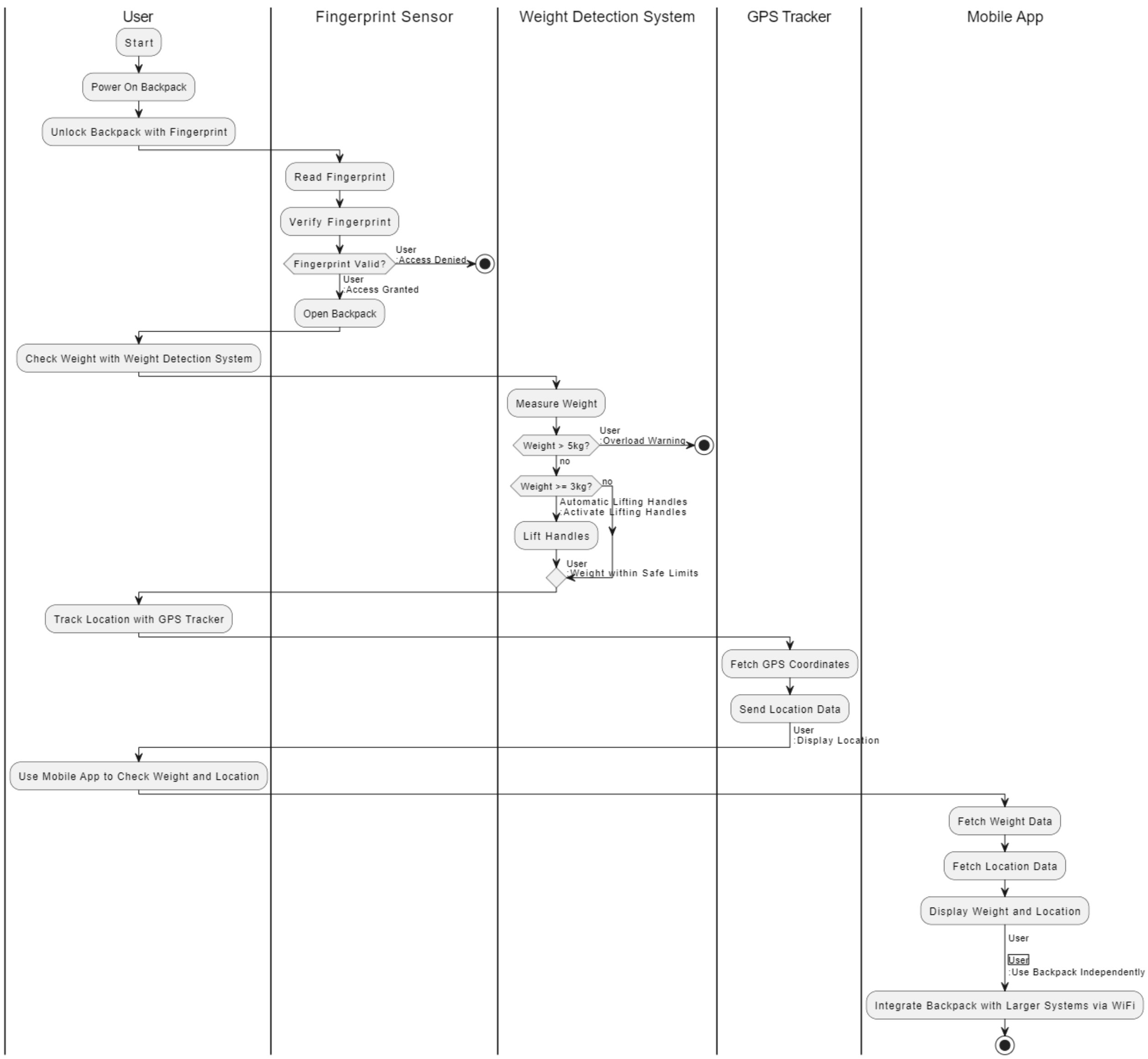


Figure 2.2.1: Activity Diagram

2.2.2 Circuit Diagram & Photos

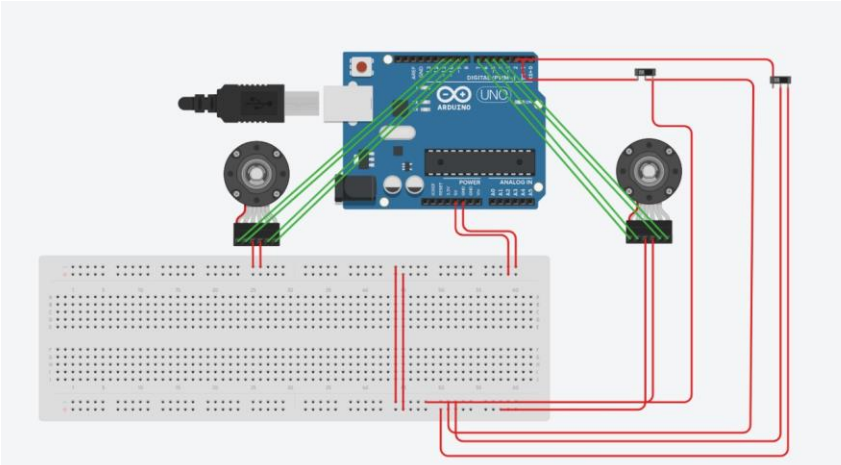


Figure 2.2.2.1: Automatic Strap Lifting

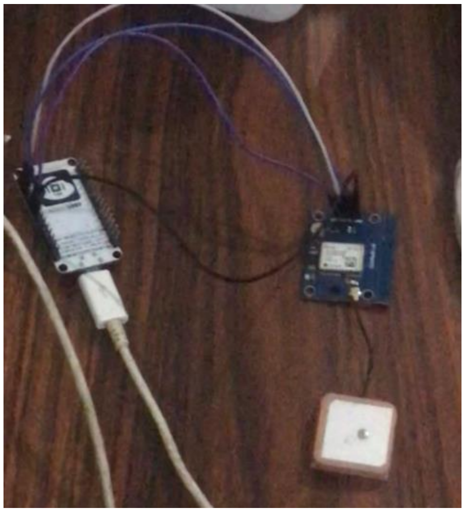


Figure 2.2.2.2: GPS Tracker

Figure 2.2.2.3: Fingerprint Sensor

2.3. Implementation

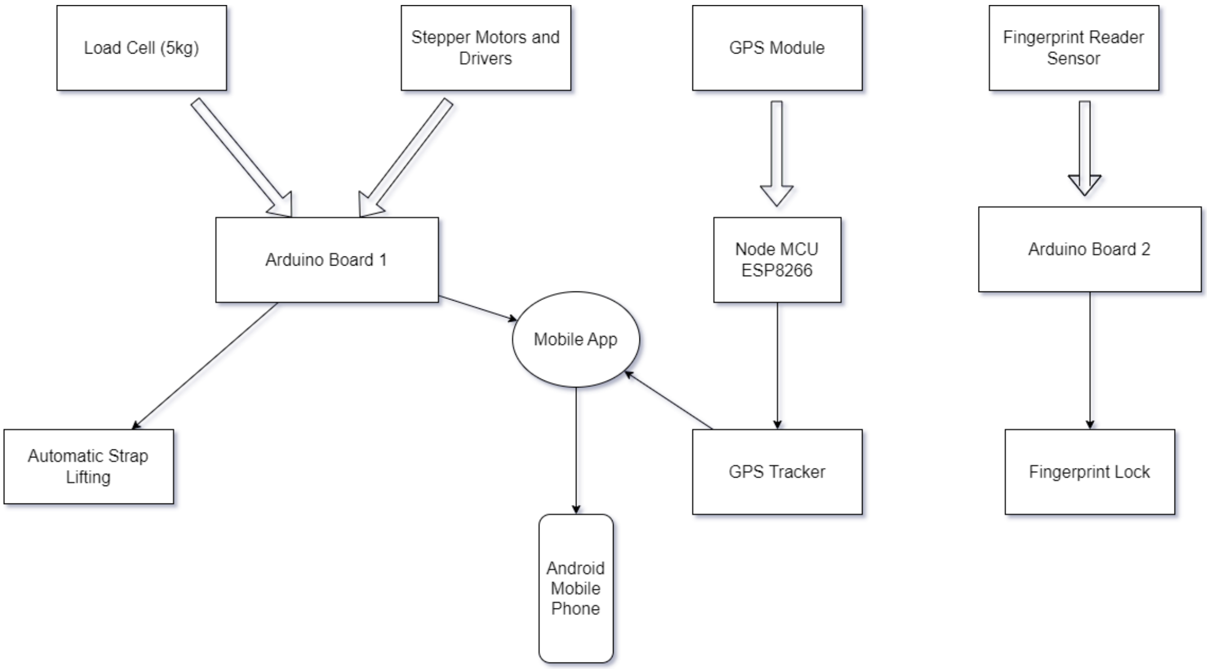


Figure 2.3:System Diagram

The foundational modules for the SMART BACKPAK with IoT project are developed during the implementation phase to meet the functional requirements outlined in the analysis section. Here is a detailed overview of the project's modules, development tools, programming languages, and unique algorithms.

Weight Detection System:-

Description: This module utilizes a load cell to measure the weight of the backpack, preventing it from exceeding safe limits. It also activates the automatic lifting handles when the weight exceeds a predefined threshold.

Implementation: This module, which was created with the Arduino IDE and the C programming language, analyses load cell data to track the weight of the backpack. The lifting mechanism is activated upon the achievement of predefined weight criteria.

Automated Handle Lifting Module:

Description: Using stepper motors and drivers, this module raises the backpack's handles automatically to help users when the weight of the backpack exceeds a certain threshold.

Implementation: This module uses a switch to regulate the stepper motors and drivers, which raises the handles when the load cell senses a weight of three kilograms or greater. The Arduino microcontroller is in charge of this module.

GPS Tracking Module:

Description: Using a GPS module and NodeMCU ESP8266 for wireless communication, this module tracks the location of the backpack in real time.

Implementation: The GPS module is implemented using the Arduino IDE to gather location data, which is then sent to a mobile app using the NodeMCU ESP8266 to allow users to track the whereabouts of the backpack.

Fingerprint Authentication Module:

Description: This module uses a fingerprint scanner to authenticate users, hence securing access to the contents of the backpack.

Implementation: Only authorized users are allowed access to the fingerprint reader, which is controlled by an Arduino microcontroller and compares the user's fingerprint to recorded data.

Mobile Application Integration:

Description: This module offers a user-friendly interface via a mobile application that lets users examine the backpack's weight and position in real time.

Implementation: For user convenience, the mobile app displays current weight and position data after receiving data from the microcontrollers over WiFi.

Development Tools and Reusable Code:

Arduino IDE: Employed for programming the Arduino boards to integrate sensors and control hardware components.

Programming Languages: Microcontrollers (Arduino boards) are programmed in the C programming language to control hardware interactions and communicate with sensors.

Special Algorithms:

Algorithms for real-time processing of sensor data, such as noise reduction and sensor reading interpretation, are known as sensor data processing algorithms.

Control algorithms: These are the programs that regulate how sensors (such as load cells, GPS units, and fingerprint readers), actuators (such as stepper motors), and the Internet of Things system interact to provide precise and responsive system behaviour.

2.4Testing

Test Scenario ID		Weight Detection -1		Test Case ID	WD-01
Test Case Description		Verify the functionality of the Weight Detection System to prevent overloading.		Test Priority	High
Pre-Requisite		Load cell, Arduino board, stepper motors, and drivers are powered and connected.		Post-Requisite	Ensure the system correctly identifies safe weight limits and prevents overloading..
Test Execution Steps:					
S.No	Action	Inputs	Expected Output	Actual Output	Test Result
01.	Simulate backpack weight equal to 3kg	N/A	No overload warning, system does not activate handles	No overload warning, system does not activate handles	fail
02.	Simulate backpack weight between 3kg and 5kg	N/A	No overload warning, system does not activate handles	No overload warning, system does not activate handles	fail

Test Scenario ID		Automatic strap lifting -1		Test Case ID	ASL-01
Test Case Description		Verify the functionality of the Automatic Lifting Handles to activate manually using a switch.		Test Priority	High
Pre-Requisite		Stepper motors, drivers, and Arduino board are powered and connected. Switch for manual activation is installed and functional..		Post-Requisite	Ensure the lifting handles activate when the switch is toggled
Test Execution Steps:					
S.No	Action	Inputs	Expected Output	Actual Output	Test Result
01.	Toggle the manual activation switch	N/A	Automatic lifting handles activate	Automatic lifting handles activate	pass
02.	Toggle the switch again	N/A	Automatic lifting handles deactivate	Automatic lifting handles deactivate	pass

Test Scenario ID	GPS Traking -1	Test Case ID	GT-01
Test Case Description	Ensure GPS tracking provides accurate location data and updates in real-time.	Test Priority	High
Pre-Requisite	GPS module and NodeMCU ESP8266 are powered and connected to WiFi.	Post-Requisite	Location data link from NodeMCU displays current location of the backpack accurately.

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
01.	Power on GPS module and NodeMCU	N/A	GPS module initializes, NodeMCU connects to WiFi	GPS module initialized, NodeMCU connected to WiFi	pass
02.	Fetch current GPS coordinates	N/A	NodeMCU retrieves GPS coordinates	GPS coordinates available via provided link	pass

Test Scenario ID	Fingerprint Lock -1	Test Case ID	FL-01
Test Case Description	Verify the functionality of the Fingerprint Sensor to grant access to the backpack's contents.	Test Priority	High
Pre-Requisite	Fingerprint reader sensor and Arduino board are powered and connected. Authorized fingerprint(s) are enrolled in the system..	Post-Requisite	Ensure the fingerprint sensor grants access upon recognizing an authorized fingerprint.

Test Execution Steps:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
01.	Place an authorized fingerprint on the sensor	Authorized fingerprint data	Access granted, backpack unlocks	Access granted, backpack unlocks	pass
02.	Place an unauthorized fingerprint on the sensor	Unauthorized fingerprint data	Access denied	Access denied	pass

3. Conclusion

3.1. Assessment of the Project Results

Smart Backpack Project
By incorporating IoT technologies effectively, the Smart Backpack project has improved user convenience and security with the following major features:

GPS Tracking:

Outcome: Implemented using NodeMCU technology.

Performance: Ensures security and peace of mind for users by providing accurate real-time location data that can be accessed through a link provided.

Fingerprint Authentication:

Outcome: Implemented to enhance security and ease of access.
Performance: Performance: Only those with permission can access the contents of the backpack because the system authenticates users with reliability.

Weight Detection (Not Implemented**)**:

Reason: Decision was made to forego implementation due to potential technical challenges and to avoid impact on other functionalities.

Impact: The strap lifting function operates effectively using a switch mechanism, maintaining usability and reliability without weight-based detection.

Strap Lifting:
Outcome: Implemented using a switch mechanism for automated lifting.

Performance: Provides user-friendly operation without the need for weight detection, supporting ease of use and functionality.

The Smart Backpack project effectively incorporates Internet of Things technology, such as GPS tracking and fingerprint verification, to improve user convenience and security. By leveraging NodeMCU technology, GPS tracking provides precise real-time position data that can be accessed over a connection, establishing the foundation for upcoming location-based applications. Robust access control is ensured via fingerprint authentication, effectively protecting contents. By omitting weight sensing, the strap lifting mechanism operates smoothly via a switch for automatic lifting, prioritizing system dependability and user-friendliness. By standardizing the integration of IoT into everyday items, this initiative improves user experience and functionality.

3.2. Lessons Learned

1. Importance of Clear Requirements:

The Smart Backpack project highlighted the critical importance of having well-defined and precise requirements from the outset. Clear requirements ensure that development efforts are focused and aligned with the project goals. In our case, specifying the functionalities of GPS tracking, fingerprint authentication, and strap lifting with clarity helped streamline development and avoid scope creep.

2. Iterative Development Approach:

Having an iterative development process was a great way to handle the complexity of integrating Internet of Things features. With this method, improvements and modifications might be made gradually in response to feedback and iterative testing. It made sure that every function, including fingerprint authentication and GPS tracking, could be improved and maximized before to being put into use.

Using an iterative development process was a great way to handle the complexity of integrating Internet of Things features. With this method, improvements and modifications might be made gradually in response to feedback and iterative testing. It made sure that every function, including fingerprint authentication and GPS tracking, could be improved and maximized before to being put into use.

3. Challenges with System Integration:

The Smart Backpack project's integration of several sensors and technologies highlighted the need of a strong system architecture and meticulous planning. Careful integration design was necessary to ensure that GPS modules, fingerprint sensors, and automated strap lifting mechanisms interacted seamlessly and remained reliable.

4. User-Centric Design strategy:

The Smart Backpack's functionality was optimized through the use of a user-centric design strategy. Through an emphasis on user demands and behaviours, including the requirement for secure and frictionless access, the project made sure that features like automatic strap lifting and fingerprint verification were not only technically sound but also simple to use and intuitive. This method improved usability and overall happiness by enabling a more customized user experience.

3.3. Future Work

Enhanced GPS Functionality:
Future development will concentrate on improving the GPS tracking features to deliver more accurate and comprehensive position data. This entails investigating modern GPS modules and perfecting location data transmission to guarantee accurate and real-time tracking.

Enhancement of Verification by Fingerprints:
The objective of future research and development is to improve the fingerprint authentication system's dependability and security. To stop unwanted access, this entails improving fingerprint recognition algorithms and making sure authentication procedures are strong.
Integrating Environmental Sensors:
Future development on the Smart Backpack will integrate environmental sensors to increase its usefulness. This might have temperature sensors to regulate the backpack's environment automatically, making sure that delicate equipment or goods are stored in the best possible circumstances.

Creation of a Complementary Mobile Application:
Next up on the agenda is the creation of a companion smartphone application. Through this app, users will be able to change settings like fingerprint access control and GPS tracking preferences, as well as remotely check the state of the backpack and get notifications.

Exploring Extra Features:
The usefulness and appeal of the Smart Backpack might be improved by investigating other features like proximity sensors for theft prevention, smart storage compartments with automatic opening/closing mechanisms, and interaction with health monitoring devices as part of ongoing improvement.

Integration of Smart Storage Solutions:
The Smart Backpack's integration of smart storage solutions will be the main focus of future advancements. This involves putting in place automatic systems for compartment opening and shutting in response to orders or user proximity. These kinds of advancements will make things more convenient and easy to use. They will also make sure that stored materials are secure and protected from unwanted access.

4. References

[1] DegrawSt, "AUTODESKInstructables," 06 sep 2017. Arduino Scale With 5kg Load Cell and HX711 Amplifier [Online]. Available: https://www.instructables.com/Arduino-Scale-With-5kg-Load-Cell-and-HX711-Amplifi/. [Accessed 25 may 2024].

[2]. How to Electronics, @HowtoElectronics](2020,sep 6),DIY IoT Weighing Scale using Load Cell HX711 & ESP8266 for Remote Weight Monitoring | Blynk,[You tube]. https://www.youtube.com/watch?v=nodSjsUoWVA&t=34s [Accessed 2024,Mar 20]

[3]. ENGINEER THIS ELECTRONICS,@ENGINEERTHIS1],(2020, Mar 22).How to Connect GPS module with Arduino || HINDI. [YouTube], https://www.youtube.com/watch?v=XwCtzGK9bmE&t=267s [Accessed 2024,Mar 20]

[4]. Dejan, "How To Mechatronics," 23 Aug 2022. Stepper Motors and Arduino – The Ultimate Guide [Online]. Available: https://howtomechatronics.com/tutorials/arduino/stepper-motors-and-arduino-the-ultimate-guide/
[Accessed 25 may 2024].

[5]. T. S. A. U. A. J. Shazia Talib, "Ijraset Journal For Research in Applied Science and Engineering Technology," IoT-based Smart Bag Using Arduino, vol. III, no. 12, p. 50, 2022.

Appendix C: Selected Code Listings

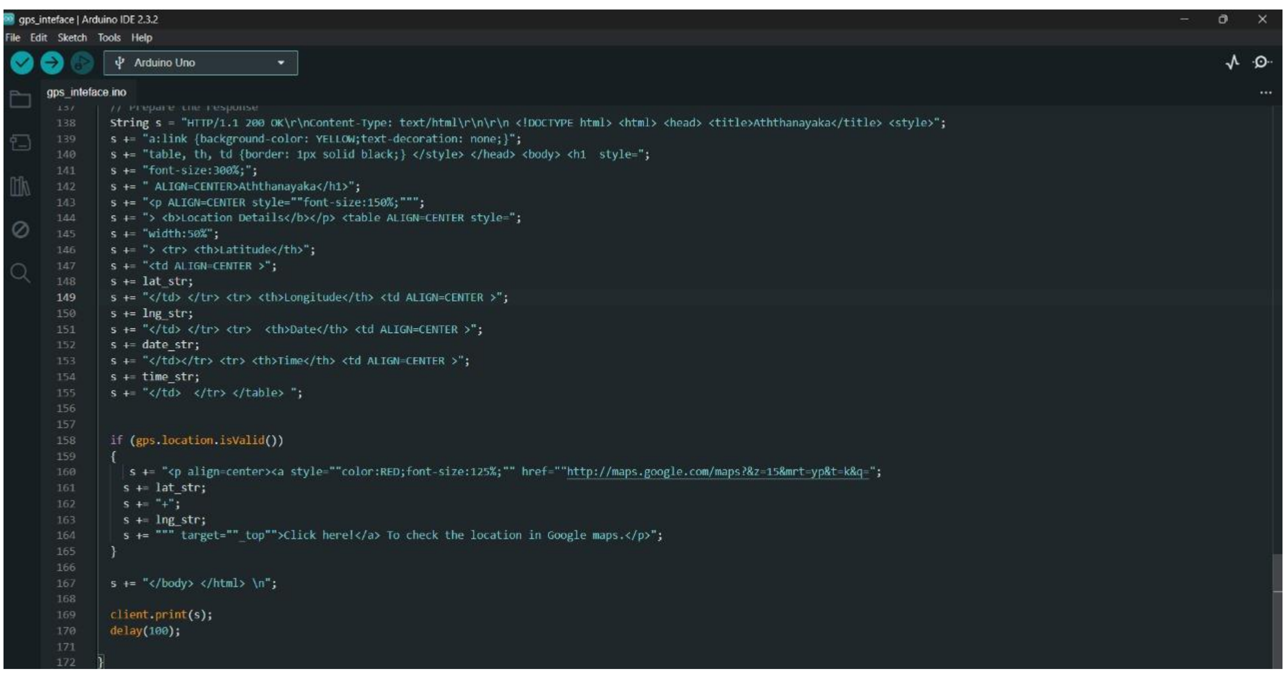
Automatic strap lifting code implementation

```
int motorSpeed = 15;
Stepper stepper1(2048, 8,  10, 9, 11);
Stepper stepper2(2048, 4, 5, 6, 7);

const int ledPin = 3;    // Pin where the LED is connected
const int buttonPin1 = 2;  // Pin where the push button is connected
const int buttonPin2 = 1;

int buttonState = 0;      // Variable to store the button state

void setup() {
  pinMode(ledPin, OUTPUT);      // Set the LED pin as an output
  pinMode(buttonPin1, INPUT);    // Set the button pin as an input
  pinMode(buttonPin2, INPUT);
  stepper1.setSpeed(motorSpeed);
  stepper2.setSpeed(motorSpeed);
}

void loop() {
 // buttonState1 = digitalRead(buttonPin1);  // Read the state of the push button

  if (digitalRead(buttonPin1) == HIGH) {  // If the button is pressed
    digitalWrite(ledPin, HIGH);  // Turn the LED on
    stepper1.step(-100);
  } else {
    digitalWrite(ledPin, LOW);   // Turn the LED off
  }

  //buttonState2 = digitalRead(buttonPin2);  // Read the state of the push button

  if (digitalRead(buttonPin2) == HIGH) {  // If the button is pressed
    digitalWrite(ledPin, HIGH);  // Turn the LED on
    stepper1.step(100);
```

GPS Tracker variable declaration

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>

TinyGPSPlus gps;

SoftwareSerial ss(4, 5);

const char* ssid = "Dialog 4G Rasara";
const char* password = "Rasara24011@";

float latitude , longitude;
int year , month , date, hour , minute , second;
String date_str , time_str , lat_str , lng_str;
int pm;
```

GPS tracker's HTML page creation



```
// Prepare the response
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n <!DOCTYPE html> <html> <head> <title>Aththanayaka</title> <style>";
s += "a:link {background-color: YELLOW;text-decoration: none;}";
s += "table, th, td {border: 1px solid black;} </style> </head> <body> <h1  style=";
s += "font-size:300%;";
s += " ALIGN=CENTER>Aththanayaka</h1>";
s += "<p ALIGN=CENTER style=""font-size:150%;""";
s += "> <b>Location Details</b></p> <table ALIGN=CENTER style=";
s += "width:50%";
s += "> <tr> <th>Latitude</th>";
s += "<td ALIGN=CENTER >";
s += lat_str;
s += "</td> </tr> <tr> <th>Longitude</th> <td ALIGN=CENTER >";
s += lng_str;
s += "</td> </tr> <tr>  <th>Date</th> <td ALIGN=CENTER >";
s += date_str;
s += "</td></tr> <tr> <th>Time</th> <td ALIGN=CENTER >";
s += time_str;
s += "</td>  </tr> </table> ";


if (gps.location.isValid())
{
  s += "<p align=center><a style=""color:RED;font-size:125%;"" href=""http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=";
  s += lat_str;
  s += "+";
  s += lng_str;
  s += """ target=""_top"">Click here!</a> To check the location in Google maps.</p>";
}

s += "</body> </html> \n";

client.print(s);
delay(100);

}
```

Figure 6.1: HTML page creation in GPS

Fingerprint sensor

```
/ returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p!=2){
    Serial.println(p);
  }
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p!=2){
    Serial.println(p);
  }
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -2;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
```