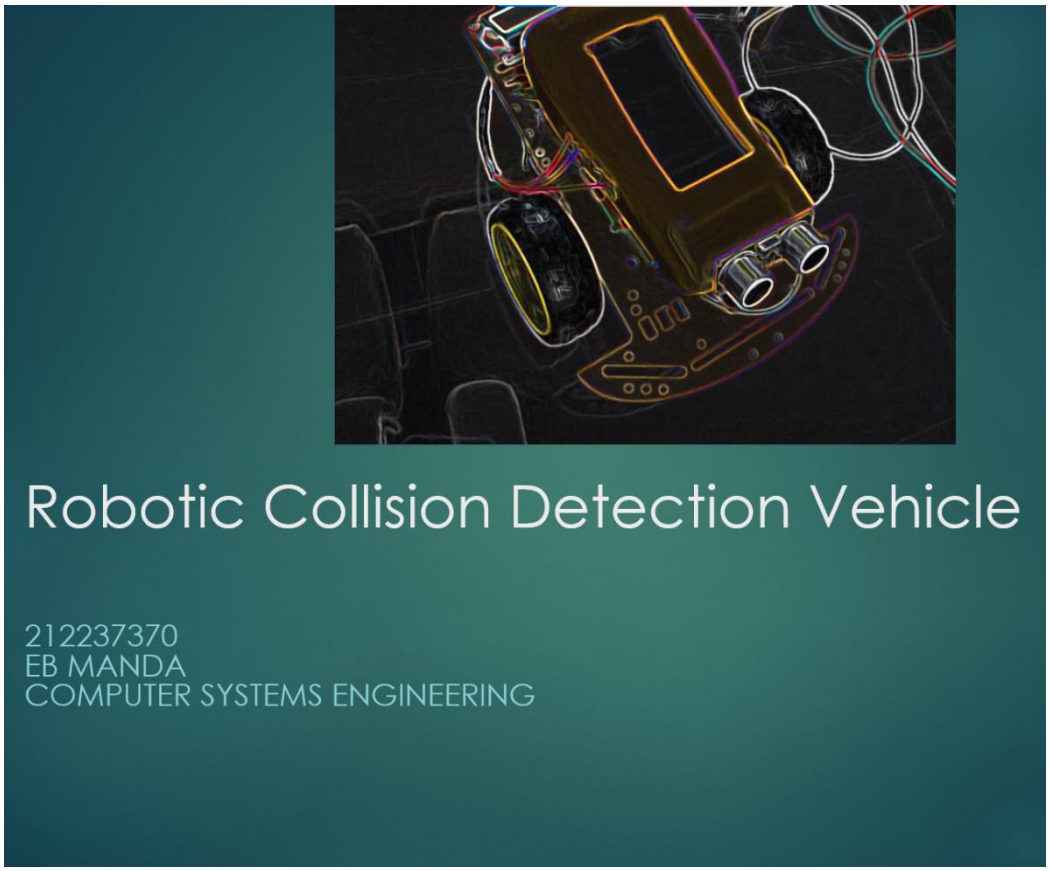
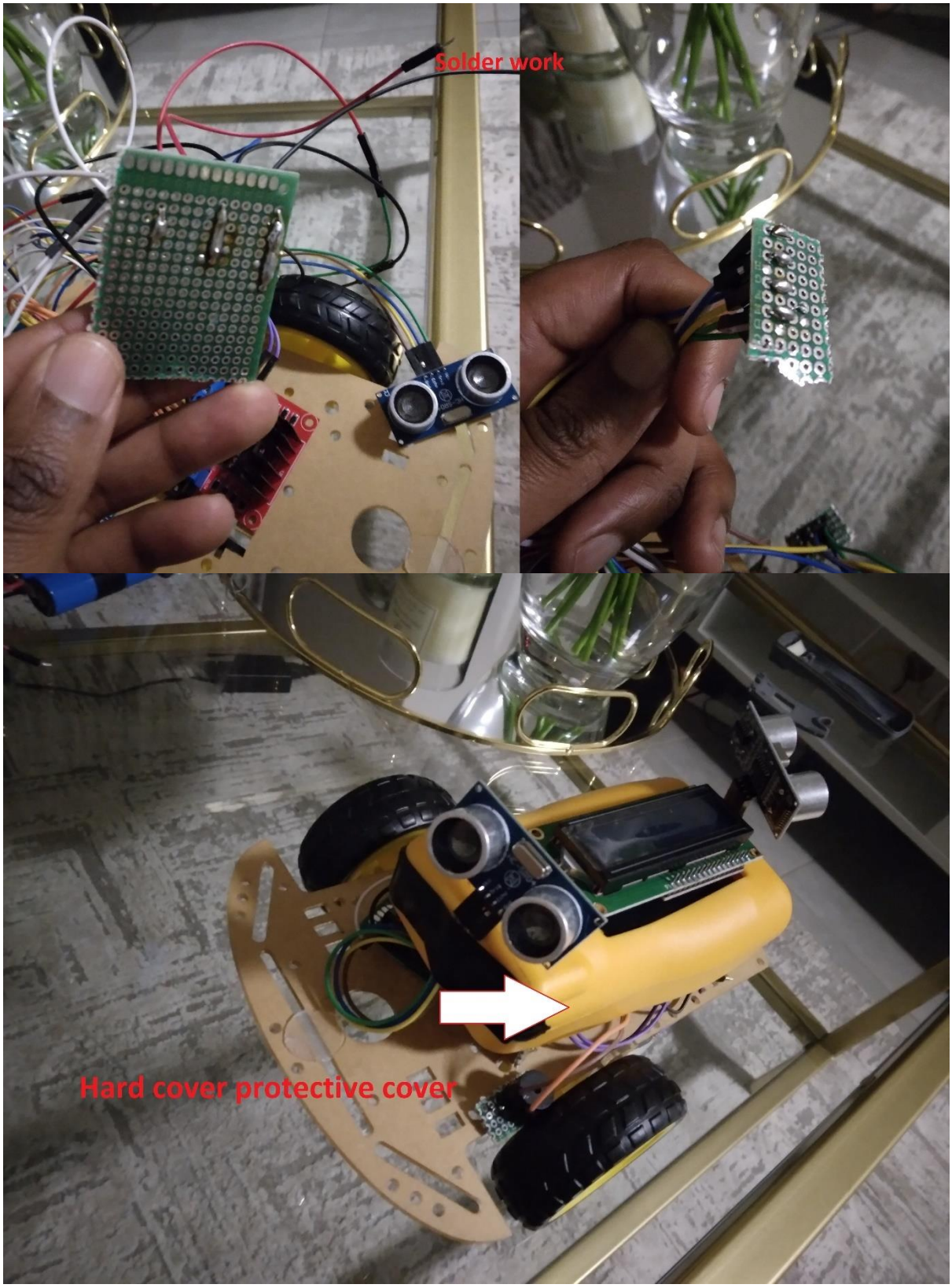
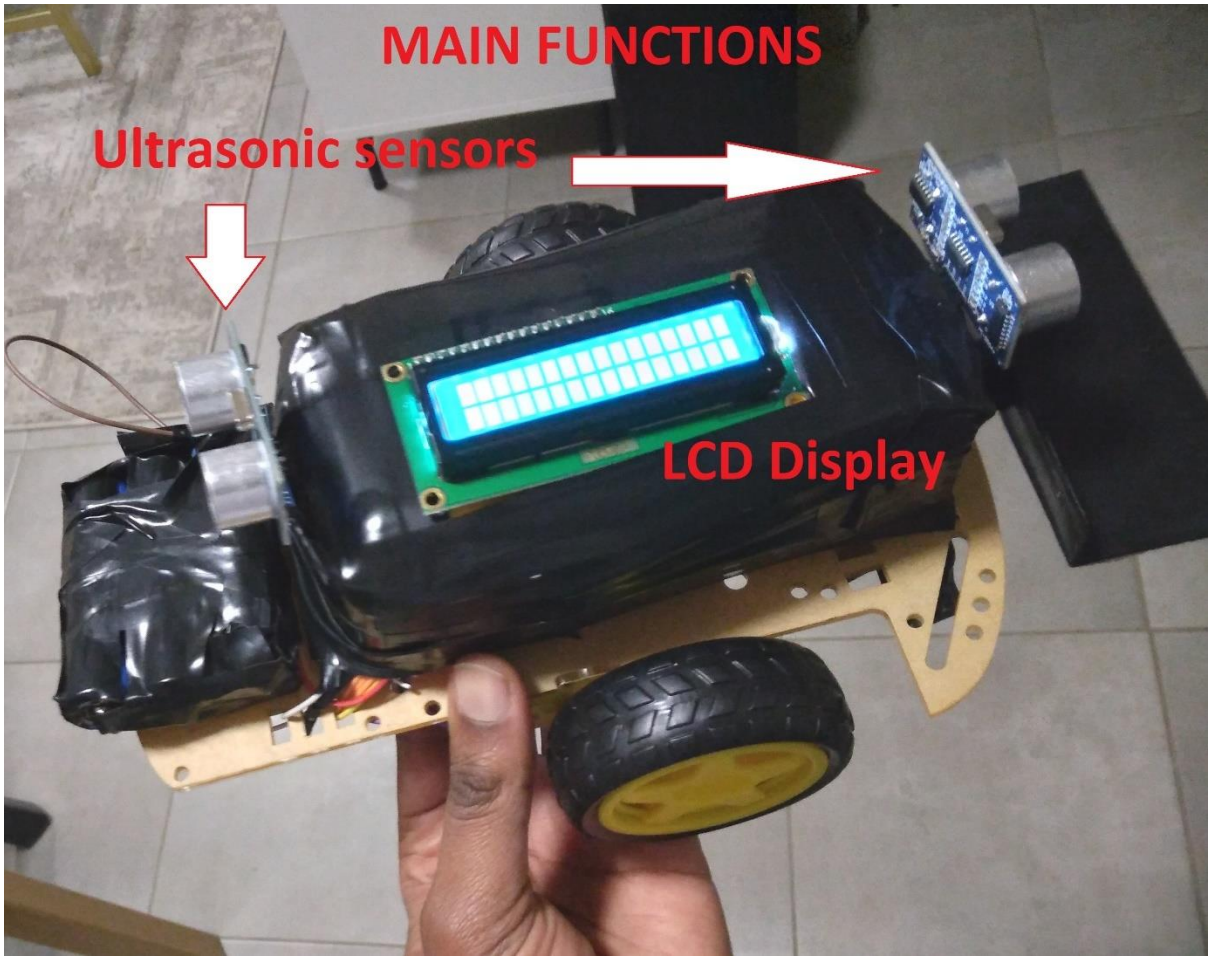


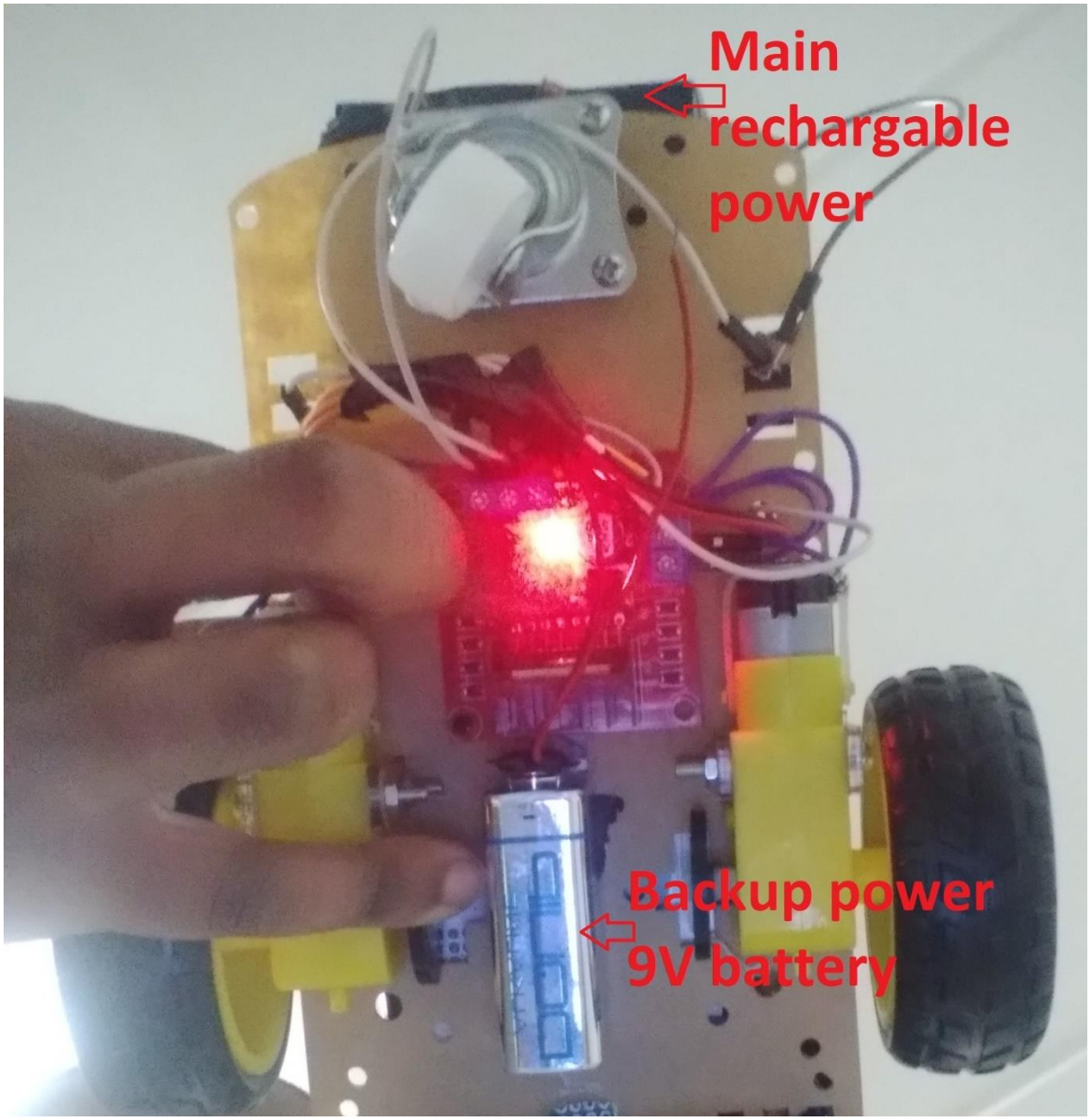
Brief Description

Safety and security are two lacking aspects in shopping wholesalers if we are talking about operating vehicles in the presence of shoppers. This system fills that need. Only licensed lifting vehicle operators are allowed to use the system. They will login with their individual credentials and be able to operate the vehicle. The vehicle will be able to detect shoppers in the way and automatically stop. After it has detected too many obstacles, it becomes inoperable. The operator will have to take steps in order to register a particular aisle onto a database where the obstacles are. This information will be available for other operators.

The vehicle




MAIN FUNCTIONS
Ultrasonic sensors
LCD Display



Code to run the vehicle

```
//By Edwin Manda

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

/* define section */

// Motor driver setup L298N
const int IN1 = 7;
const int IN2 = 6;
const int IN3 = 5;
const int IN4 = 4;
```

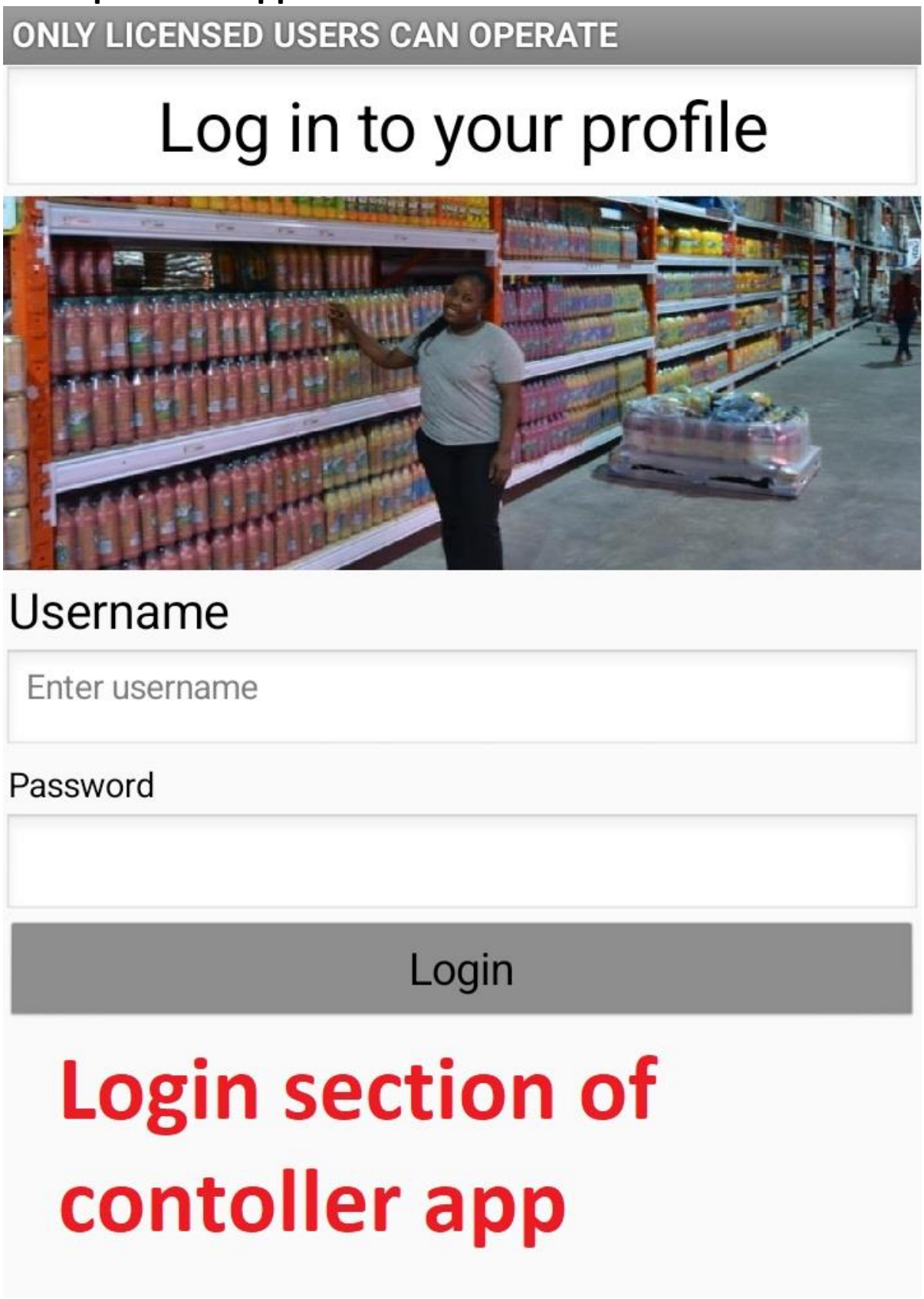
```
const int ENA = 9;
const int ENB = 3;

const int trig = 11;
const int echo = 12;

const int trig2 = 8;
const int echo2 = 13;

int buzzer = 10;
int command = 0;

long duration;
int dis;
int safe;

// New feature: obstacle count
int obstacleCount = 0;

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

/* Setup section */

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);

  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(trig2, OUTPUT);
  pinMode(echo2, INPUT);

  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.backlight();
  lcd.backlight();
}

/* Loop section */

void loop() {
  if (Serial.available() > 0) {
```

```
  command = Serial.read();
  executeCommand(command);
}

// First sensor
digitalWrite(trig, LOW);
delayMicroseconds(2);
digitalWrite(trig, HIGH);
delayMicroseconds(10);
digitalWrite(trig, LOW);
duration = pulseIn(echo, HIGH);
dis = (duration * 0.034) / 2;
safe = dis;

if (safe <= 10) {
  stopMotors();
  digitalWrite(buzzer, HIGH);
  displayLCD();
  obstacleCount++; // Increment obstacle count
} else {
  digitalWrite(buzzer, LOW);
  lcd.clear();
}

// Second sensor
digitalWrite(trig2, LOW);
delayMicroseconds(2);
digitalWrite(trig2, HIGH);
delayMicroseconds(10);
digitalWrite(trig2, LOW);
duration = pulseIn(echo2, HIGH);
dis = (duration * 0.034) / 2;
safe = dis;

if (safe <= 10) {
  stopMotors();
  digitalWrite(buzzer, HIGH);
  displayLCD();
  obstacleCount++; // Increment obstacle count
} else {
  digitalWrite(buzzer, LOW);
  lcd.clear();
}

// Check if obstacle count reaches 5
if (obstacleCount >= 4) {
  stopMotors();
  lcd.clear();
```

```
    lcd.setCursor(0, 0);
    lcd.print("Not safe to");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("operate vehicle");
    delay(8000);

  }
}

void executeCommand(int command) {
  switch (command) {
    case '0':
      moveForward();
      break;
    case '1':
      moveBackward();
      break;
    case '2':
      moveLeft();
      break;
    case '3':
      moveRight();
      break;
    case '4':
      stopMotors();
      break;
    default:
      break;
  }
}

void moveForward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, 60);
  analogWrite(ENB, 60);
}

void moveBackward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 60);
  analogWrite(ENB, 60);
```

```
}

void moveLeft() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, 60);
  analogWrite(ENB, 60);
}

void moveRight() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 60);
  analogWrite(ENB, 60);
}

void stopMotors() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}

void displayLCD() {
  lcd.setCursor(0, 0);
  lcd.print("SHOPPER DETECTED!");
  delay(1000);
  lcd.setCursor(0, 1);
  lcd.print("Kindly wait...");
  delay(8000);
}
```

The Operator's app



ONLY LICENSED USERS CAN OPERATE

Log in to your profile

Username

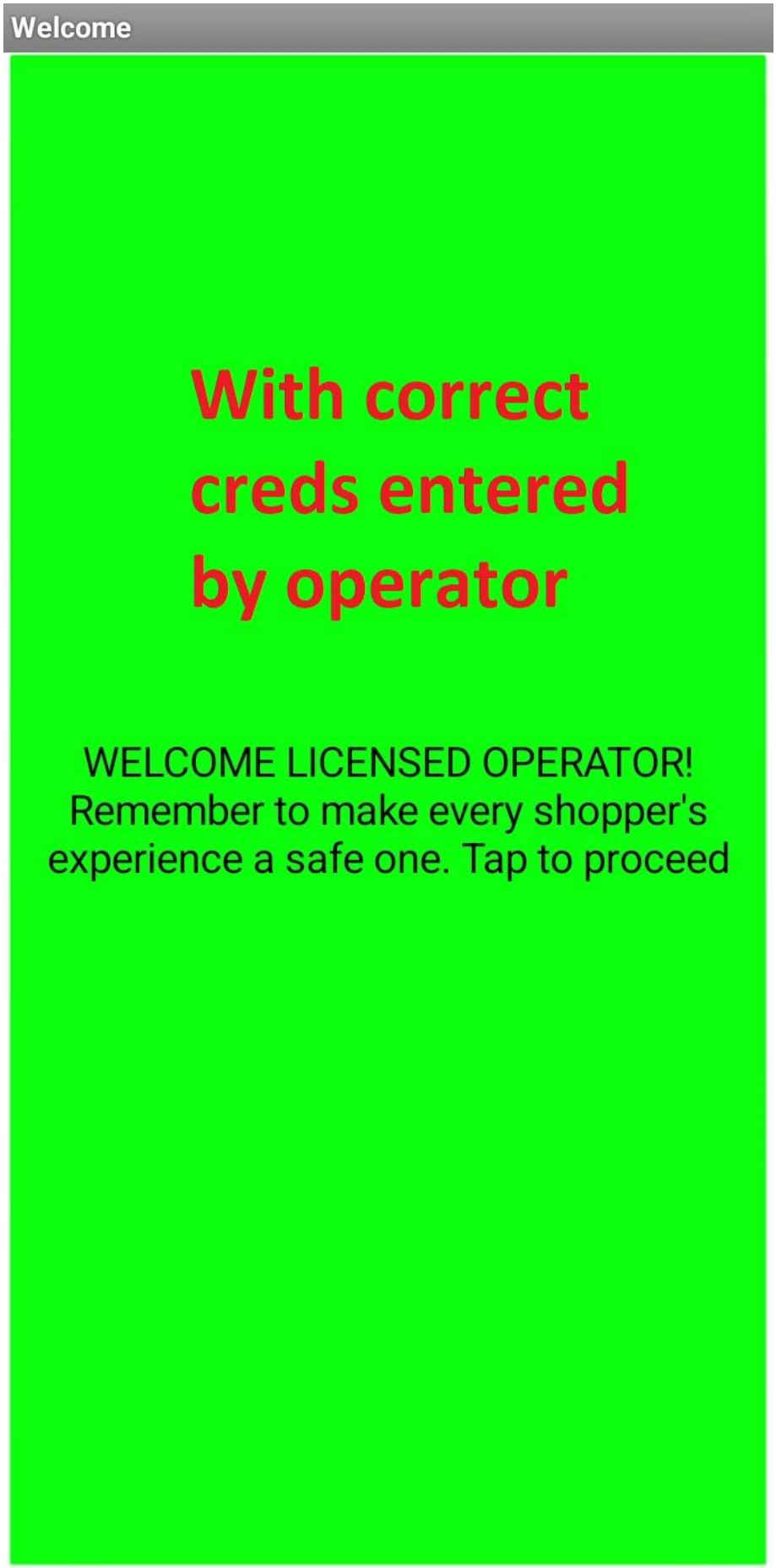
licensed_Operator1

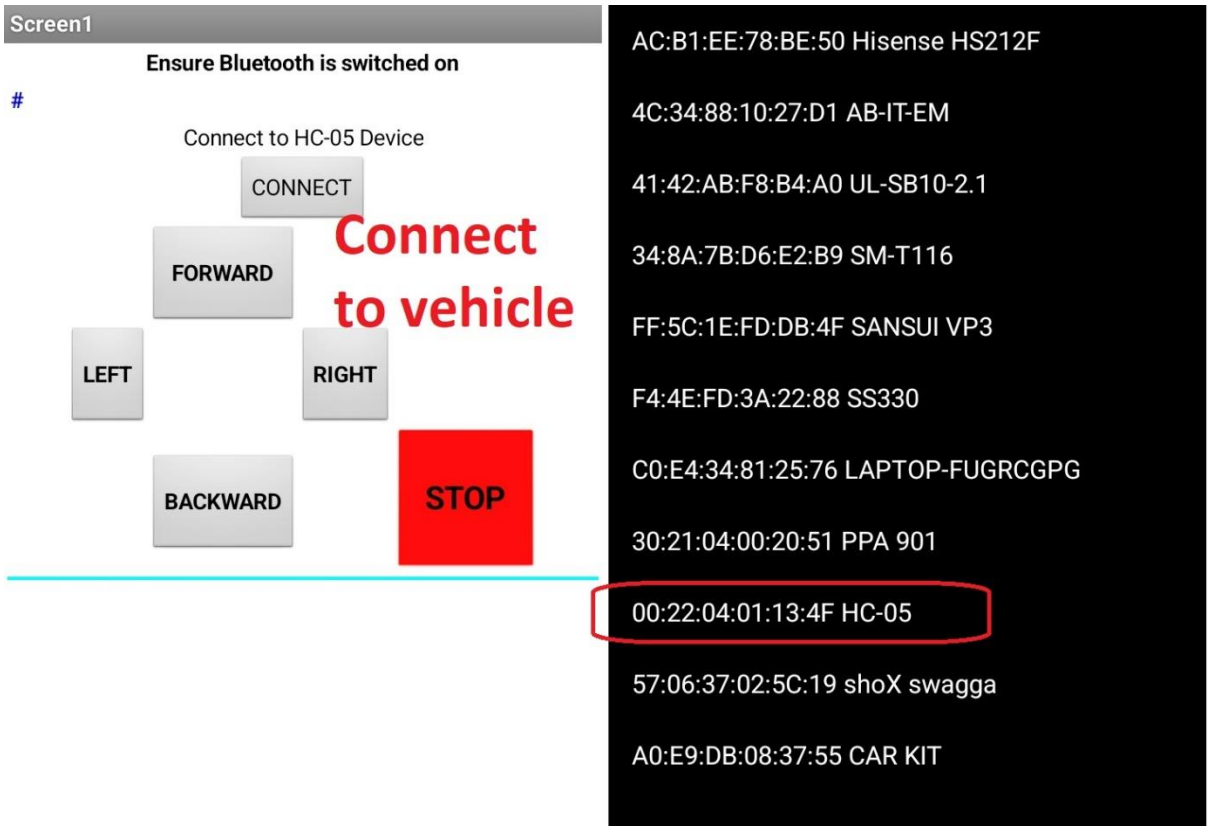
Password

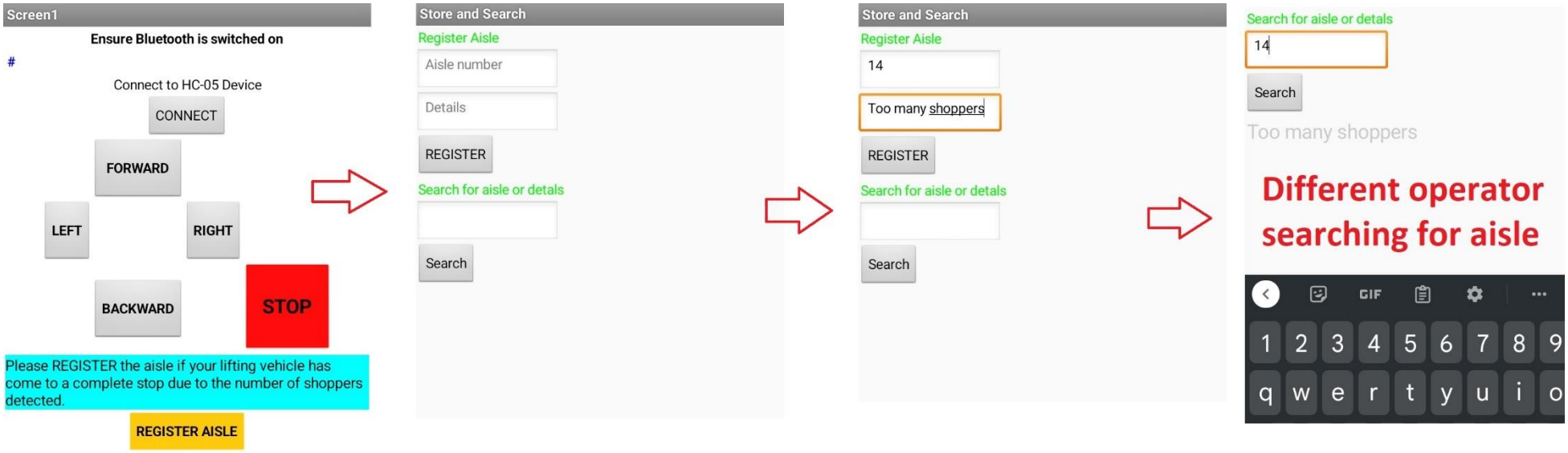
Incorrect creds entered

Login

Incorrect Credentials


Welcome
With correct creds entered by operator
WELCOME LICENSED OPERATOR! Remember to make every shopper's experience a safe one. Tap to proceed


Screen1
Ensure Bluetooth is switched on
#
Connect to HC-05 Device
CONNECT
FORWARD
LEFT
RIGHT
BACKWARD
STOP
Connect
to vehicle
AC:B1:EE:78:BE:50 Hisense HS212F
4C:34:88:10:27:D1 AB-IT-EM
41:42:AB:F8:B4:A0 UL-SB10-2.1
34:8A:7B:D6:E2:B9 SM-T116
FF:5C:1E:FD:DB:4F SANSUI VP3
F4:4E:FD:3A:22:88 SS330
C0:E4:34:81:25:76 LAPTOP-FUGRCGPG
30:21:04:00:20:51 PPA 901
00:22:04:01:13:4F HC-05
57:06:37:02:5C:19 shoX swagga
A0:E9:DB:08:37:55 CAR KIT



Once vehicle has come to a complete stop due to the number of obstacles/shoppers detected, operator needs to register the aisle on the database. Other operators with the same app will be able to search for the aisle and see what the issue is with the aisle.