# ***Client’s Security System:***

***Project Summary:***

This project is a basic home security system that detects motion and alerts the user. It uses a small device called an **Arduino Uno** to control the system. The setup includes a **motion sensor**, **LED light**, **buzzer**, and **wires** all connected through a **breadboard**. It will be explained in deeper detail below.

**When the motion sensor detects movement (like someone walking past for example), it will:**

* **Turns on an LED light**
* **Plays a sound through a buzzer**
* **Displays a message saying "Motion detected!"** on a computer screen
* When the motion stops, everything turns off and the message changes to **"Motion stopped!"**

# **Initial Design:**

Now moving onto the design aspect. Here below I have included a scan of the initial design which was drawn out. As shown, I have included: An Arduino Uno, Buzzer, Motion Sensor, Led, and Breadboard. This was accompanied with some wiring too. It is key to keep in mind that these connections regarding the wiring are not accurate and are only for design purposes. Diagram

Description automatically generated

Following this, I have explained all the components below.

# **Components:**

**Arduino Uno R3**

The Arduino Uno R3 is a Microcontroller which is based off its board. This microcontroller is easily to use and simple for those who may have never used a microcontroller before. This is due to how its functionalities and features allow for many possibilities which can all be started with a basic set up. Connecting the Arduino Uno R3 to a computer through a USB cable, or even through an AC-DC adapter, you can start immediately. Aside from this, the Arduino Uno R3 contains 14 Digital Input/output Pins, 6 Analog Inputs, a 16MHz ceramic resonator, a USB Connection, a Power jack, and ICSP header, and a Reset Button (UNO R3, 2011). These are just some of the technical specifications which help towards making this microcontroller create endless creations.

**Piezo**

A Piezo, Piezo Buzzer to be more specific, is an additional tool which can be used in sync with the Arduino. It acts as a small speaker which can be connected directly and can produce sound if triggered through the correct pins and electrical signals. For example, certain actions linked to the Arduino, potentially a code behind it, may cause the Piezo to activate and output a sound as a response. In addition to this, this sound produced can also be customised by the user. This implies the frequency, and the length of the sound can be altered.

**Breadboard**

A Breadboard is yet another tool which is used alongside the Arduino when making complex circuits. It is yet another board which acts as the construction base for an electronic circuits. Wires will enter specific sections through its pin and can be connected back to the Arduino. As well as this, it is solderless which implies that there is no need for any precise and stressful soldering, the user can simply connect wires with a firm fitting, holding the connection.

**PIR Sensor**

A PIR Sensor, or Passive Infrared Sensor, is a type of sensor which can be used with many circuits, including the Arduino Uno R3. This sensor can detect any infrared which may be radiated from any objects, given that they emit heat. To gain a better understanding, these can also be referred to a “Motion-Based” Detectors. This implies that when activated this tool will search for any infrared wavelengths and will trigger any output linked to it.

**Wiring**

Wiring is fairly straightforward and quite self-explanatory. In this case, they are small wires which have either: Male to Female, Male to Male, or Female to Female connection points. This is helpful when using and creating circuits which may use the Arduino as well as the Breadboard.

**LED:**

Similarly, to the Wiring, this is simple and easy to understand. The LED used in situations like this are a small bulb with two connection points beneath it. These connector points can be attached to any of the circuit boards and then can be lit if coded correctly.

# **Virtual simulation of project:**

As this is a more of a physical project, it can be inconvenient to transport it , so I have created a virtual simulation on tinkercad to give an idea of how the project appears

Diagram, schematic

Description automatically generated

A screenshot of a computer

AI-generated content may be incorrect.

Once motion is detected over the PIR sensor, this will trigger the LED will turn on, a sound to be played from the piezo buzzer and a message will be displayed (“Motion detected!”), as soon as the motion is no longer detected over the sensor, the sound from the buzzer will no longer be made, the LED will turn off and the message ("motion stopped!") will then be displayed.

# **Code:**

const int MOTION\_SENSOR\_PIN = 7;   // Arduino pin connected to the OUTPUT pin of motion sensor

const int BUZZER\_PIN        = 3;   // Arduino pin connected to Buzzer's pin

const int LED\_PIN =5; //Arduino pin connected to LED

int motionStateCurrent      = LOW; // current  state of motion sensor's pin

int motionStatePrevious     = LOW; // previous state of motion sensor's pin

void setup() {

  Serial.begin(9600);                // initialize serial

  pinMode(MOTION\_SENSOR\_PIN, INPUT); // set arduino pin to input mode

  pinMode(BUZZER\_PIN, OUTPUT);          // set arduino pin to output mode

  pinMode(LED\_PIN, OUTPUT);          // set arduino pin to output mode

}

void loop() {

  motionStatePrevious = motionStateCurrent;            // store old state

  motionStateCurrent  = digitalRead(MOTION\_SENSOR\_PIN); // read new state

  if (motionStatePrevious == LOW && motionStateCurrent == HIGH) { // pin state change: LOW -> HIGH

    Serial.println("Motion detected!");

    digitalWrite(BUZZER\_PIN, HIGH); // turn on

    digitalWrite(LED\_PIN, HIGH); // turn on

  }

  else

  if (motionStatePrevious == HIGH && motionStateCurrent == LOW) { // pin state change: HIGH -> LOW

    Serial.println("Motion stopped!");

    digitalWrite(BUZZER\_PIN, LOW);  // turn off

    digitalWrite(LED\_PIN, LOW);  // turn off

  }

}