Rojan Verma (1510110508
Manendra Sing (1510110507
Sarthak Mittal (1410110370

Arduino Code:

```
int temp_sensorPin = A0;
int temp_sensorValue = 0;

int button = 2;
int r1 = 3;
int r2 = 13;
int g1 = 7;
int y1 = 4;
int ldrPin = A3;
int sensorVal = 0;

int pirState = LOW;
volatile int sensor = -1;

int tempConvert(int s) {
  return (int)s*250.0/1024;
}

void switch_sensors() {
  // 0 = PIR
  // 1 = LDR

  if(sensor == -1)
    sensor = 0;
  else if(sensor == 0){
    sensor = 1;
  }
  else{
    // LDR sensing

    sensorVal = analogRead(ldrPin);
    sensor = 0;
  }

  Serial.print("Interrupt Occoured;" );
  Serial.print(sensor == 0? "PIR": sensor==1?"LDR":"Not started");
  Serial.println(" sensor active;");
  Serial.print("Sensor Value: ");
  Serial.print(sensorVal);
  if(sensorVal > 300){
   digitalWrite(g1, HIGH);
  }
  delay(3000);
  digitalWrite(g1,LOW);
  Serial.println(" sensor active;");
  Serial.println("Interrupt Successfully Handled - returning to main program");
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial) ; // Needed for Leonardo only
  pinMode(r1, OUTPUT);
  pinMode(y1, OUTPUT);

  attachInterrupt(digitalPinToInterrupt(button), switch_sensors, FALLING);


  delay(1000);
  //sensorValue = analogRead(sensorPin);
}

void blink(int pin1, int pin2, int times){
 for(int i = 0; i < times; i++){
    digitalWrite(pin1, HIGH);
    if(pin2) digitalWrite(pin2, HIGH);
    delay(100);
    digitalWrite(pin1, LOW);
    if(pin2) digitalWrite(pin2, LOW);
    delay(100);
  }
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  temp_sensorValue = analogRead(temp_sensorPin);
  int temp = tempConvert(temp_sensorValue);
  int r_blinks = 0, y_blinks = 0;

  if(temp < 30){
    r_blinks = 2;
    blink(r1, 0, 2);
  }

  else if(temp >= 30 && temp <= 40){
     blink(r1, y1, 4);
     r_blinks = 4;
     y_blinks = 4;
  }
  else if(temp > 40) {
    blink(y1, 0, 6);
    y_blinks = 6;
  }

  if(r_blinks) digitalWrite(r1, HIGH);
  if(y_blinks) digitalWrite(y1, HIGH);
  Serial.print("<");
    Serial.print(temp);
      Serial.print(" (c), ");
       Serial.print(r_blinks);
      Serial.print(", ");
      Serial.print(y_blinks);
      Serial.print(", ");
      Serial.print(y_blinks?"Y":"N");
      Serial.println("> ");

  delay(2000);


}
```

PRocessing

```
int rectX, rectY;      // Position of square button
int rectSize = 90;     // Diameter of rect
color rectColor, circleColor, baseColor;
color rectHighlight, circleHighlight;
color currentColor;
boolean rectOver = false;

// actual logic
String sensor_name;
String final_string = "";
int button_counter = 0;

// text
PFont f;

void update_final_string() {
  int s = second();  // Returns values from 0 to 59
  int m = minute();  // Returns values from 0 to 59
  int h = hour();    // Returns values from 0 to 23
  String time = "<" + h + ":" + m + ":" + s + ">";  // Prints the time to the console

  int d = day();
  int mon = month();
  int year = year();
  String date = "<" + d + " " + mon + " " + year + ">";

  if(button_counter%2 == 0) {
    sensor_name = "PIR ";
  }
  else {
    sensor_name = "LDR ";
  }

  final_string = sensor_name+"sensor was last accessed at "+time+" "+ date;
}


void setup() {
  size(640, 360);
  rectColor = color(0);
  rectHighlight = color(51);
  baseColor = color(255);
  currentColor = baseColor;
  rectX = width/2-rectSize-10;
  rectY = height/2-rectSize/2;
  ellipseMode(CENTER);


  //f = createFont("Arial",16,true);

}

void draw() {
  update(mouseX, mouseY);
  background(currentColor);

  if (rectOver) {
    fill(rectHighlight);
  } else {
    fill(rectColor);
  }
  stroke(255);
  rect(rectX, rectY, rectSize, rectSize);
  //textFont(f);
  textAlign(BOTTOM, CENTER);
  text(final_string, 315, 150);

}

void update(int x, int y) {

  if ( overRect(rectX, rectY, rectSize, rectSize) ) {
    rectOver = true;
  } else {
    rectOver = false;
  }
}

void mousePressed() {
  if (rectOver) {
    currentColor = color(255);
  }

  button_counter++;
  update_final_string();
  println(final_string);
  redraw();

}

boolean overRect(int x, int y, int width, int height)  {
  if (mouseX >= x && mouseX <= x+width &&
      mouseY >= y && mouseY <= y+height) {
    return true;
  } else {
    return false;
  }
}
```