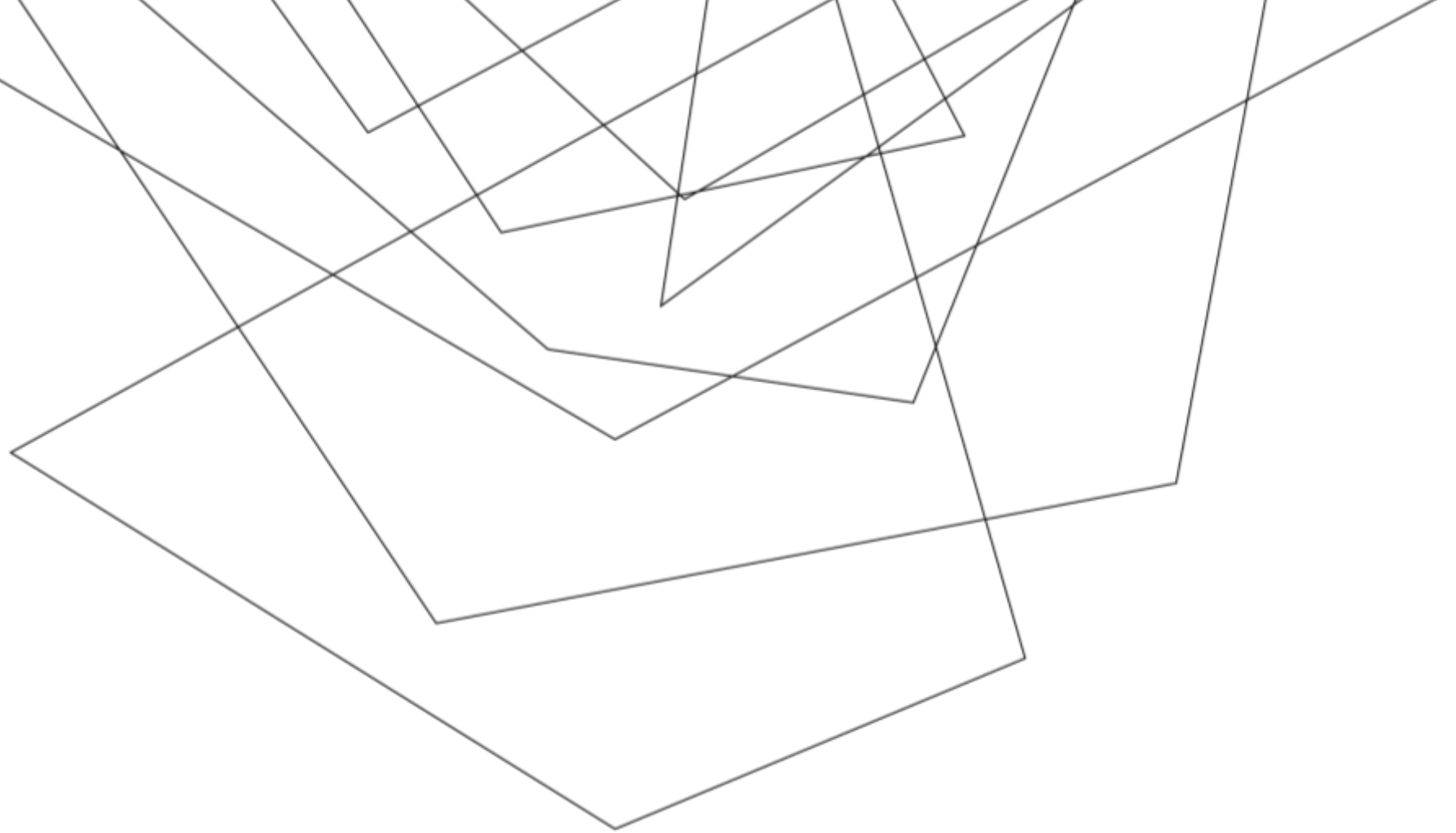
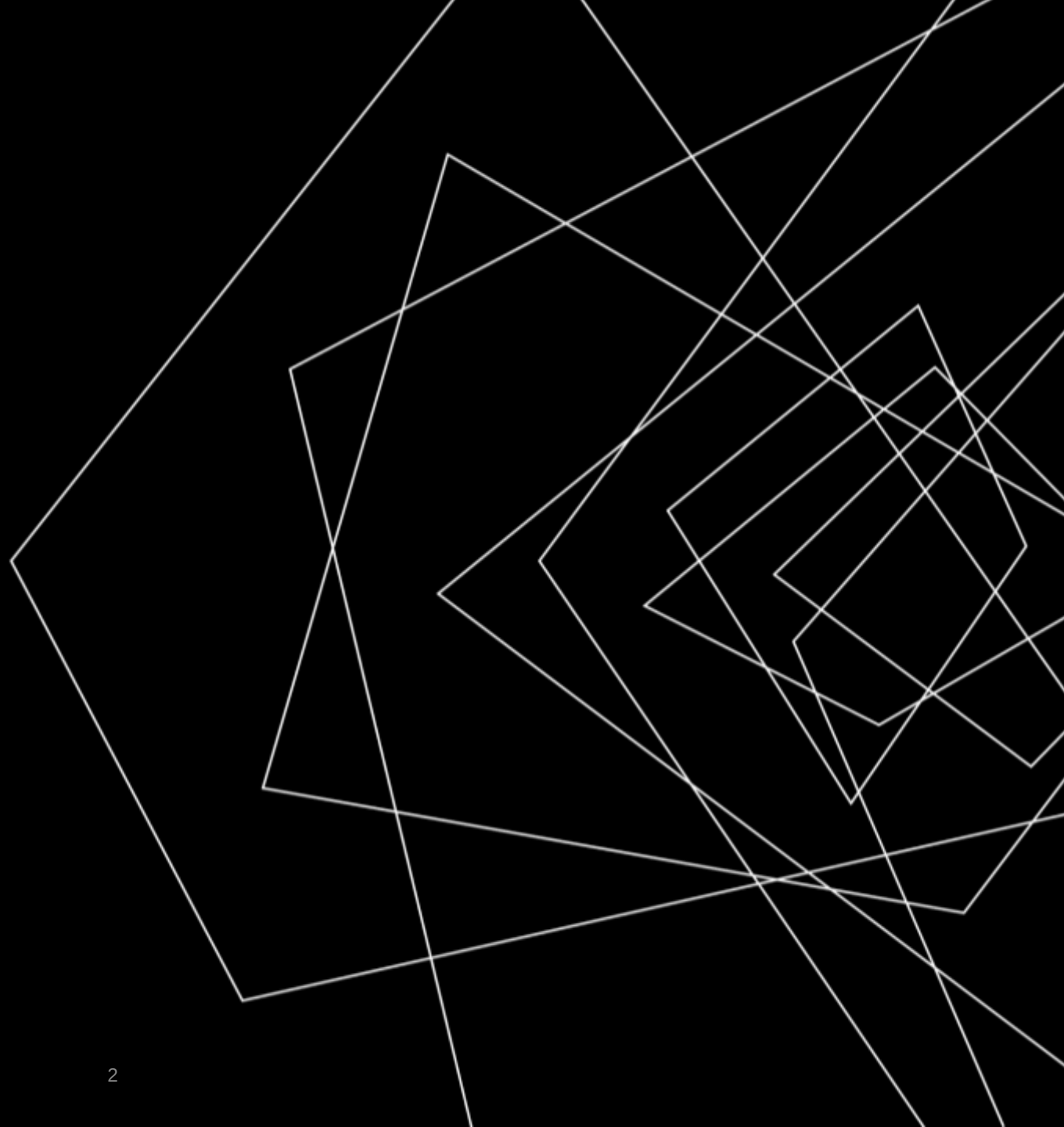


SWITCHEASE

Experiential Learning- Group 5

PROBLEM STATEMENT:

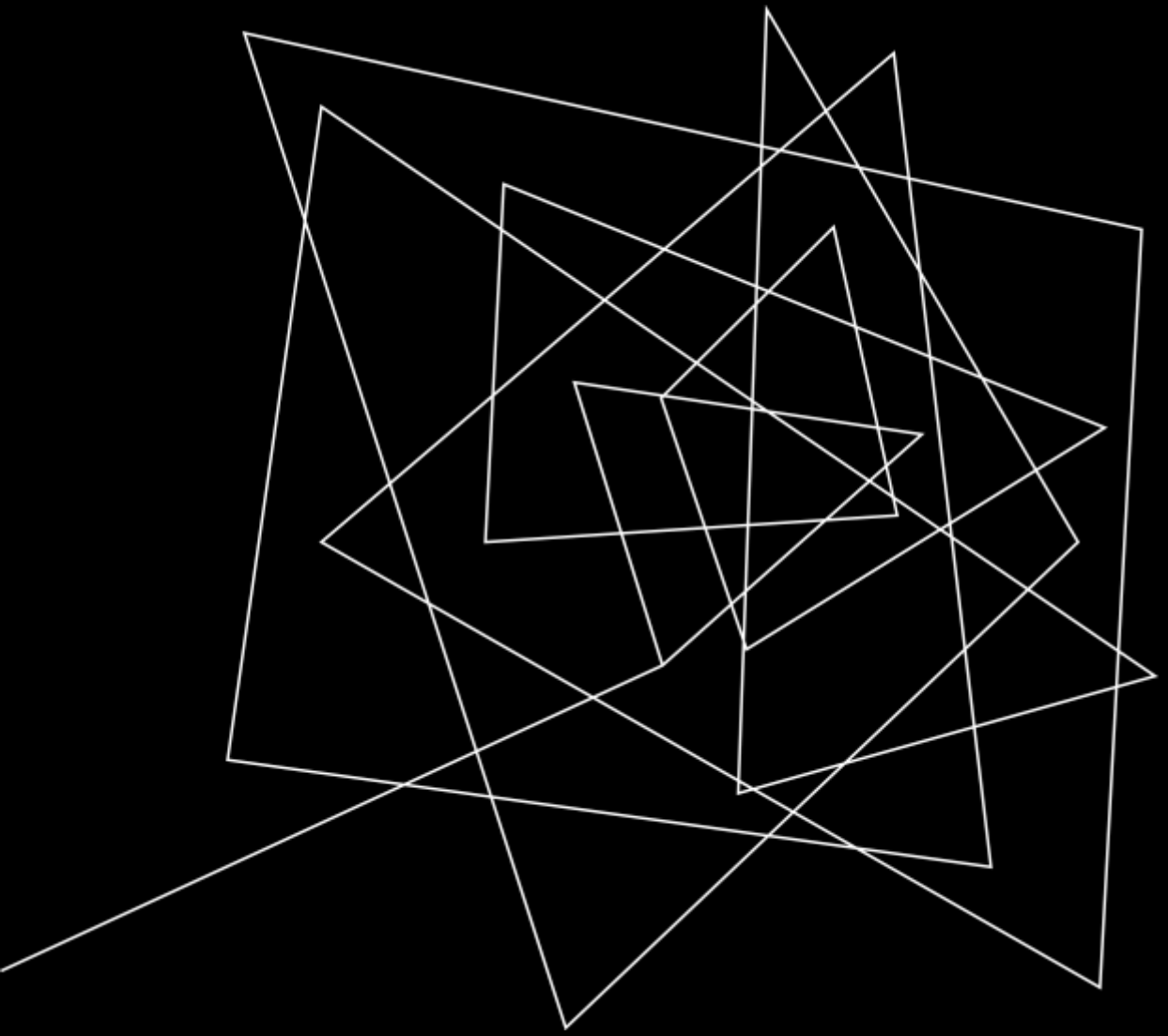
Creating a switch board that takes remote commands on a phone and uses an Arduino to switch the status of a switch.



INTRODUCTION

We begin by analyzing all the components that the construction of this device can be broken into.

These parts include a mobile application interface, an Arduino board in combination with a breadboard, and the actual switch enclosure that would house the switch-set and the motor-gear mechanism



SUBTASKS:

The project was broken into the following subparts:

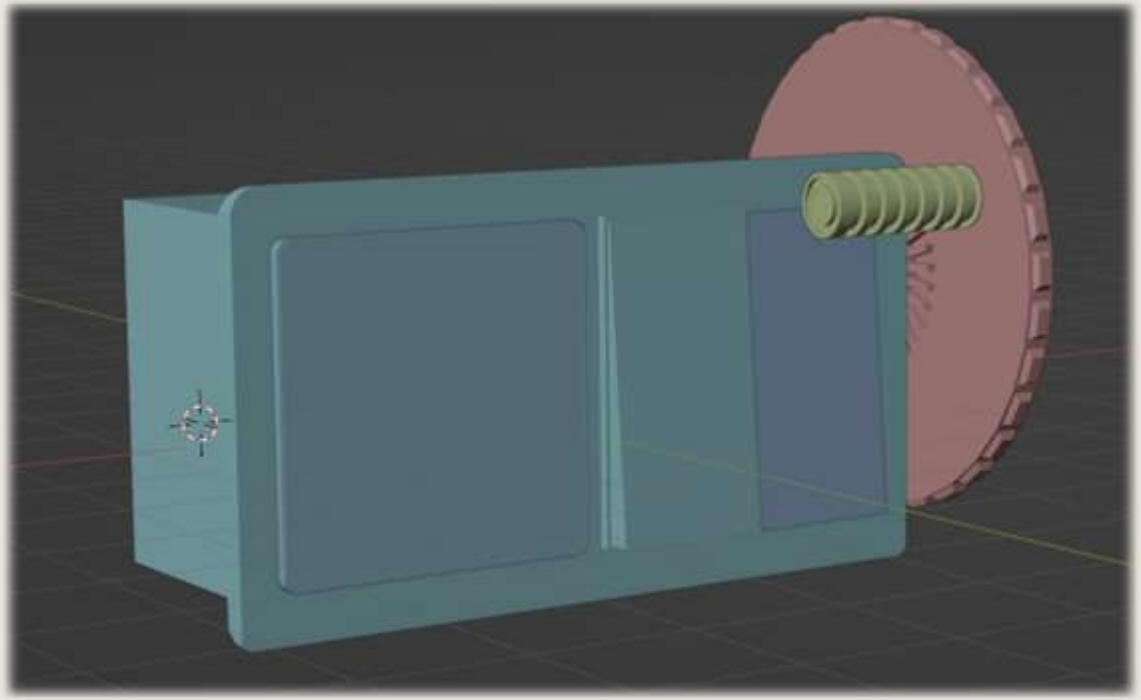
1. Design
2. Arduino and circuitry establishment
3. Mobile interface design
4. Coding

DESIGN

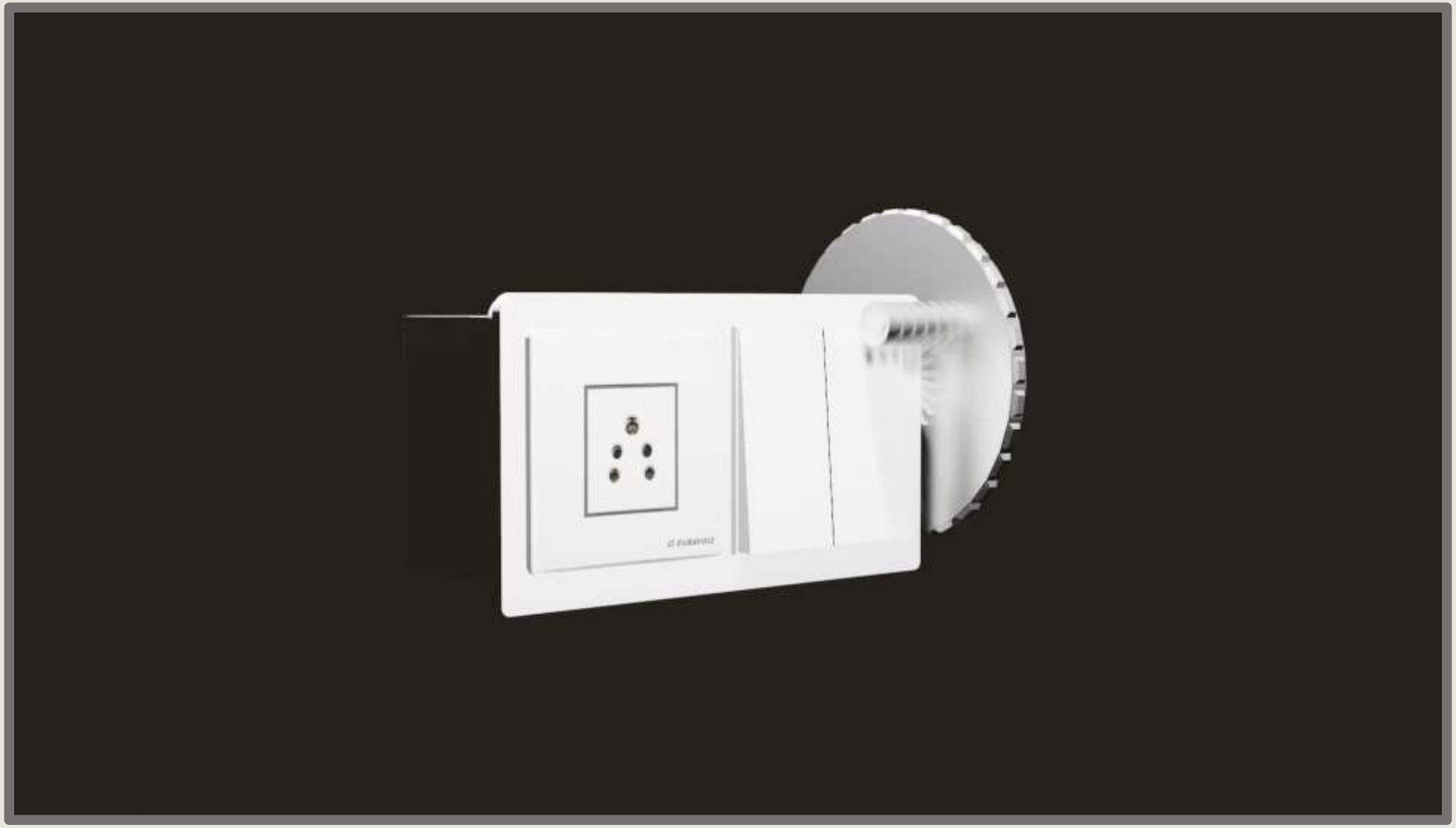
Blender and AutoCAD are used to realise the design of the product.

The different components are as following:

1. Switch board enclosure
2. Gear mechanism
3. Probe arm
4. Motor



A VIDEO RENDER OF THE SWITCH MECHANISM



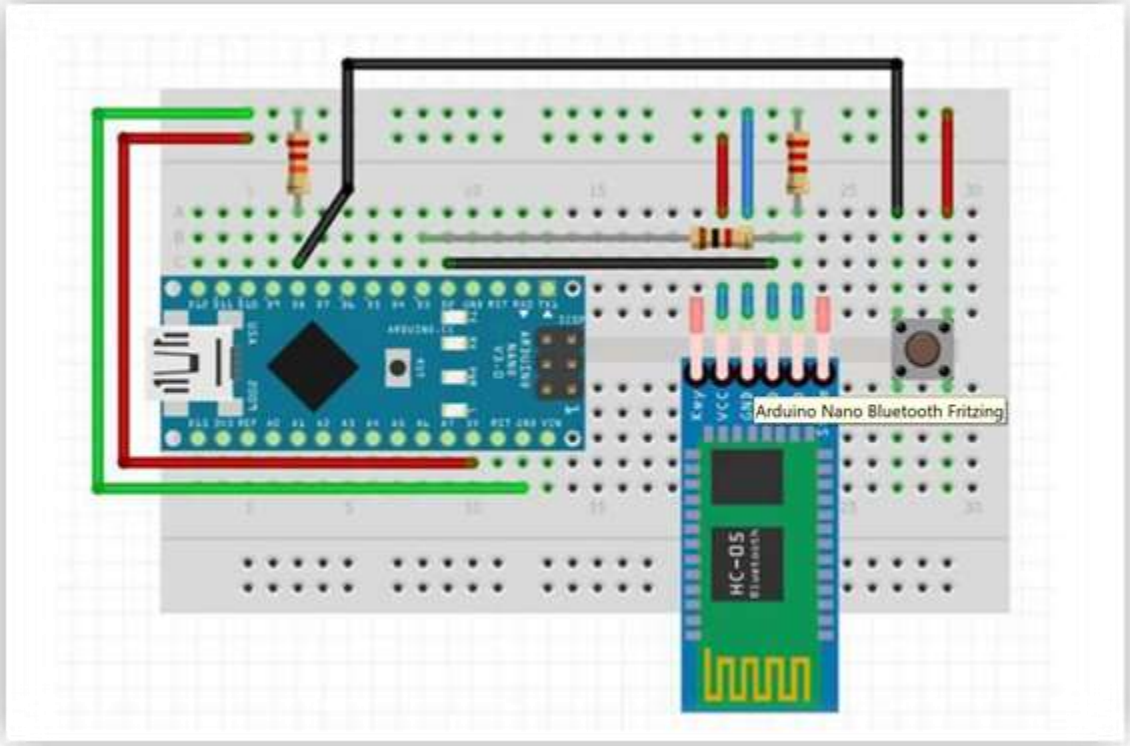
ARDUINO CIRCUITRY AND BLUETOOTH MODULE

The Arduino circuitry is composed of the following components:

1. Arduino Nano
2. HC-05 Bluetooth module

Connection is established through the following steps:

1. Connection of the motor to the breadboard
2. Connection of the Arduino to the breadboard
3. Programming the Arduino
4. Testing the established connection

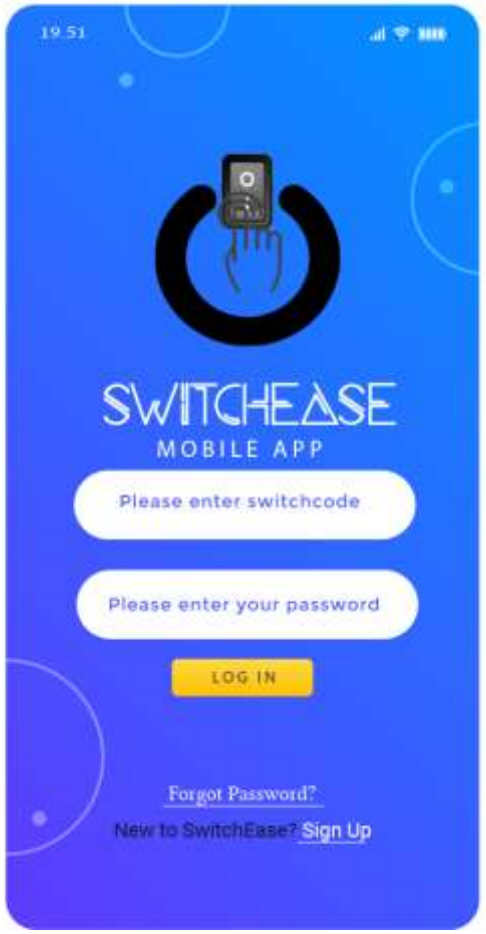


MOBILE INTERFACE

There are two components to the design of the mobile interface:

- Login page: Verifies user credentials and grants control to the switch device accordingly.

- Control page: Provides controls for the different switches available on the switchboard



ARDUINO CODE FOR MOTOR:

```
void setup() {
  // set digital pin 9 as an output
  pinMode(9, OUTPUT);
}

void loop() {
  // turn the motor on
  digitalWrite(9, HIGH);
  delay(1000); // wait for 1 second

  // set the speed of the motor to half
  analogWrite(9, 128);
  delay(2000); // wait for 2 seconds

  // turn the motor off
  digitalWrite(9, LOW);
  delay(1000); // wait for 1 second
}
```

JAVA CODE SAMLPLE FOR MOBILE APP:

```
/*
This is the most important piece of code. When "deviceName" is
found
the code will call a new thread to create a bluetooth connection
to the
selected device (see the thread code below)

BluetoothAdapter bluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
createConnectThread = new
CreateConnectThread(bluetoothAdapter,deviceAddress);
createConnectThread.start();
}
/*
Second most important piece of code. GUI Handler
*/
handler = new Handler(Looper.getMainLooper()) {
@Override
public void handleMessage(Message msg){
switch (msg.what){
case CONNECTING_STATUS:
switch(msg.arg1){
case 1:
toolbar.setSubtitle("Connected to " + deviceName);
progressBar.setVisibility(View.GONE);
buttonConnect.setEnabled(true);
buttonToggle.setEnabled(true);
break;
case -1:
toolbar.setSubtitle("Device fails to connect");
progressBar.setVisibility(View.GONE);
buttonConnect.setEnabled(true);
break;
}
break;}}};
```

OUR TEAM

- Aiden Correya
- Ayushman Gupta
- Charu Ramnani
- Kartik Takroo
- Kumar Piyush (L)
- Nabhya Sharma
- Parth Khaitan
- Rehan Sawarn
- Ritvik Singh
- Sheersh Bansal
- Vardan Agarwal

THANK YOU