A. Problem Statement

Mrs. Lee, at 80, struggles with mobility due to knee issues and often forgets to turn off the lights in her kitchen after using and balcony when it is daytime. Her daughter needs a hassle-free way to access their home without relying on Mrs. Lee to navigate the stairs. How might we create a system that enables smooth, key-free entry for her daughter while also ensuring the lights are used efficiently, enhancing safety, convenience, and energy conservation in their living spaces?

B. Solution Idea

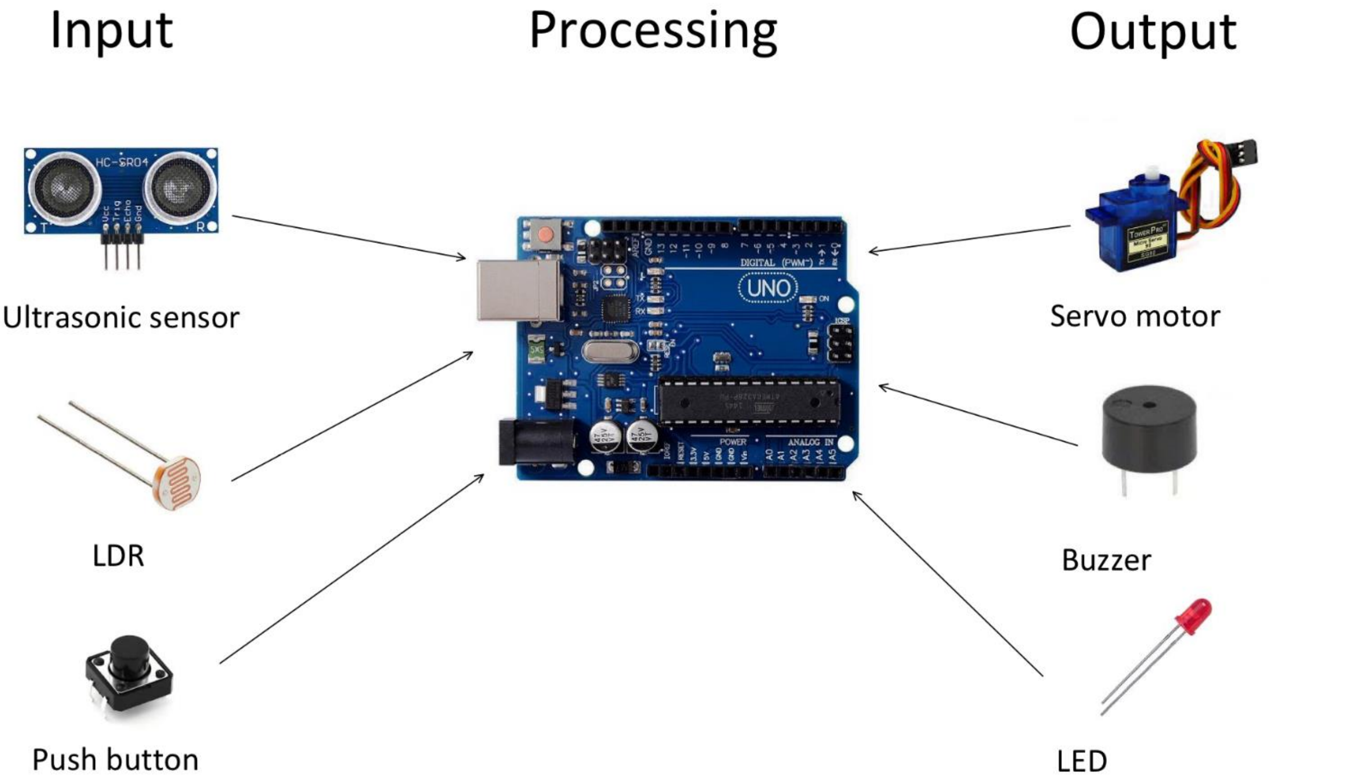
A smart and sustainable home is proposed to be the solution. It features a password-protected entry mechanism operated by a pushbutton and servo motor, removing the need for physical keys. The kitchen is equipped with an ultrasonic sensor-linked LED light that activates upon someone's entry and deactivates upon exit, conserving energy. Additionally, servo motor can be manually controlled by pushbuttons located both upstairs and downstairs, facilitating door operation without requiring Mrs. Lee to travels downstairs. An external doorbell with a buzzer ensures she is alerted to visitors. LDR is placed outside in the balcony to turn on the light only when it is nighttime.

C. Block Diagram and Brief Description

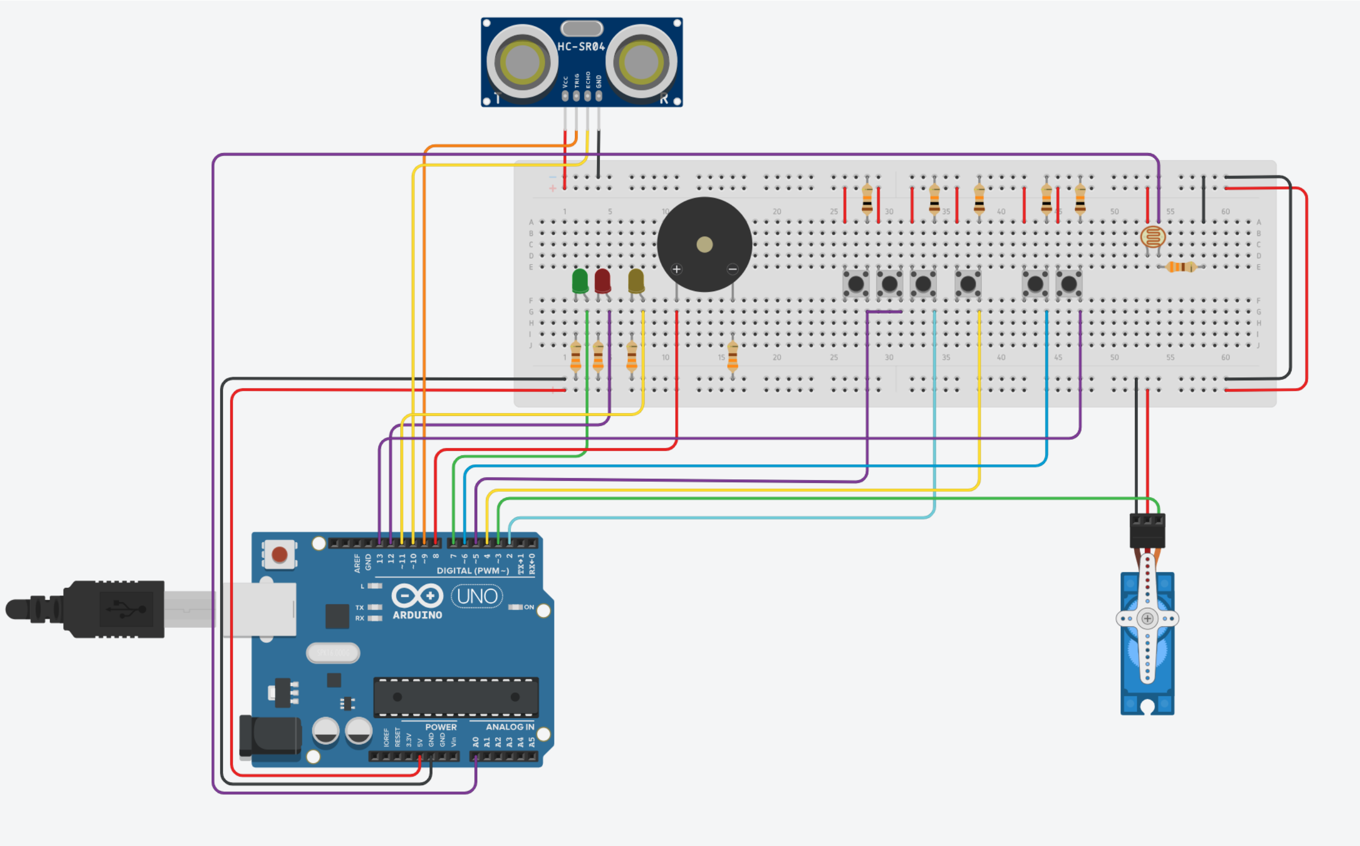
- An ultrasonic sensor in the kitchen that detects movement, automatically switching the LED lights on when someone enters the room and off when not in used, saving electricity.

- A Light-Dependent Resistor (LDR) on the balcony that detects low-light conditions, triggering the LED to light up the area during nighttime without manual intervention.

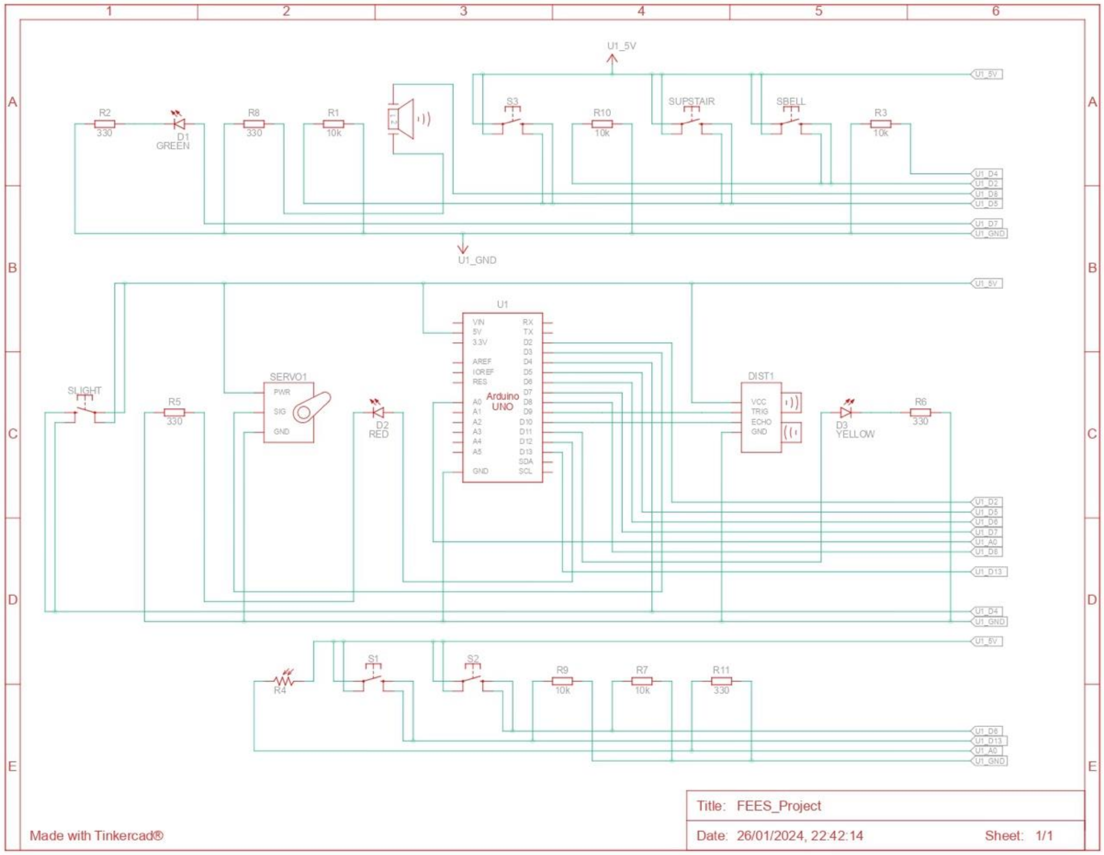
- Pushbuttons are strategically placed for multiple functionalities:
  - To sound a buzzer, promptly notifying residents of arriving guests.
  - To operate a servo motor that opens or closes the door.
  - To manually toggle the LED lights in the living room, providing convenient control over the lighting environment.
  - To input a pre-set password that, when entered correctly, prompts the servo motor to unlock the door for secure access.



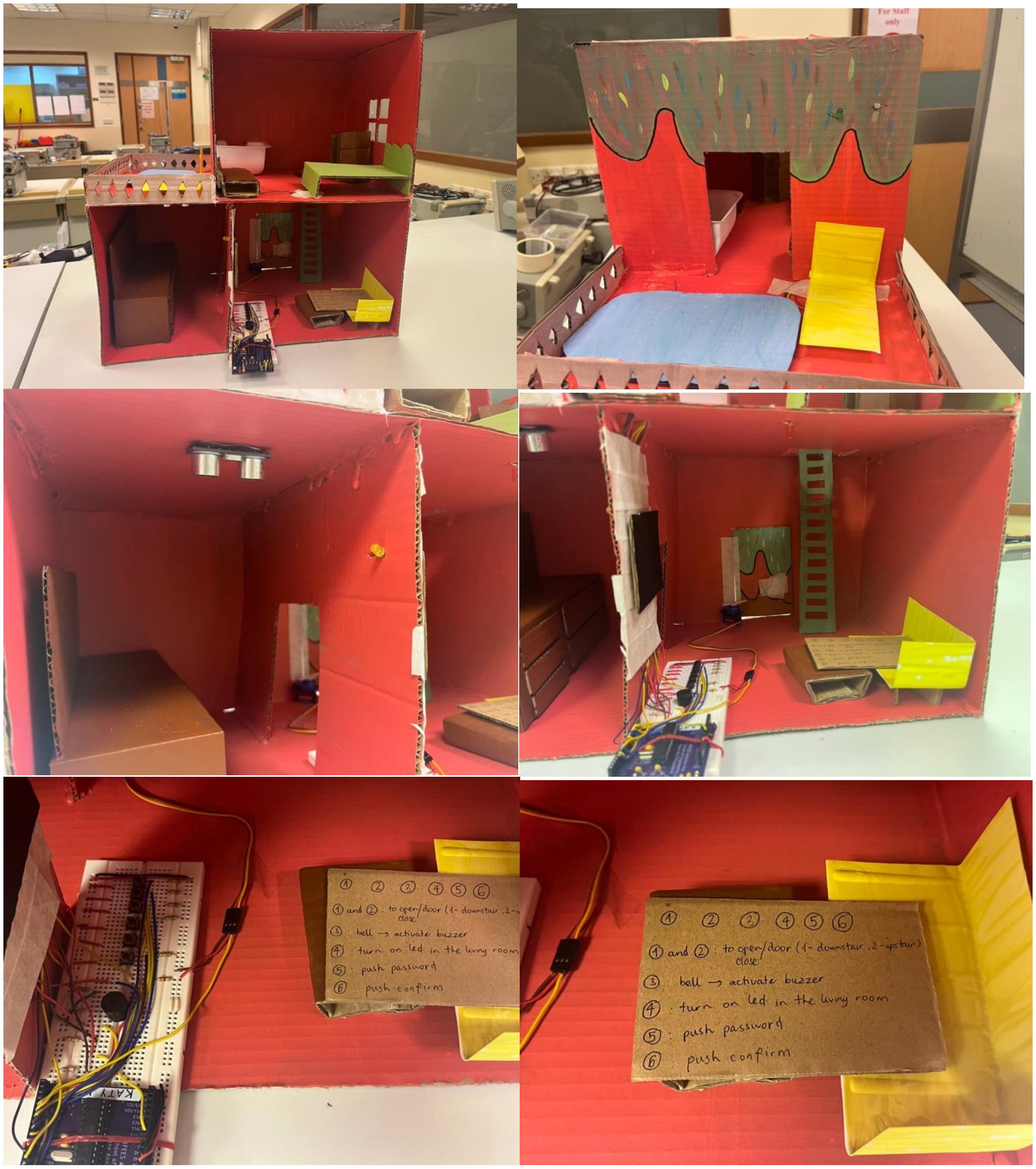
D. Tinkercad Diagram



E. Schematic/Wiring Diagram



F. Application/Prototype Photo


For Staff only
① ② ③ ④ ⑤ ⑥
① and ② : to open/close door (1- downstair, 2-upstair)
③ : bell → activate buzzer
④ : turn on led in the living room
⑤ : push password
⑥ : push confirm

G. Reflection

What did you learn from this project? Briefly elaborate in terms of both technical & civic learning.

(**Civic learning** refers to the process of acquiring knowledge, skills and values related to civic engagement, and developing an understanding of responsibilities of individuals in a society and contributing to the well-being of the community at large.)

From the technical view, I've immersed myself in learning and working with Arduino and Tinkercad, deepening my understanding of electronics and physical computing. I've developed competencies in Arduino programming, intricate electronic circuitry, and virtual simulations, along with practical troubleshooting techniques. Building a model house from cardboard allowed me to translate these skills into a tangible context. These troubleshooting strategies are invaluable; they sharpen my diagnostic abilities, enabling me to identify issues, document my approach, and implement solutions effectively. This knowledge is a springboard for future endeavors, equipping me with a robust toolset for enhancing upcoming ventures.

From the civic learning point, the project deepened my understanding of empathetic design in engineering. It showed me the importance of comprehending diverse user needs and how technology can be tailored to address specific challenges. As an engineering student tasked with crafting future technologies, it's essential for me to grasp the varied contexts in which innovations are used. Recognizing this, I've learned to approach problem-solving with a mindset that prioritizes inclusivity and practicality. This way, the solutions I develop are not just technically sound but also widely beneficial and adaptable to different segments of society. My education is not just about building things; it's about building things that make a difference in people's daily lives and contribute positively to the community.

H. Arduino Program Code

```
#include <Servo.h>

typedef bool boolean;
using namespace std;
const int pushDoor = 5;
boolean toggle = true;
int doorbell = 2;
int bell =  8;
int bellState = 0;
Servo servo_3;
const int ldrPin = A0;
const int ledPin = 12;
const int controlLight = 4;
const int ledBalcony = 7;
int ledState = 0;
int controlOld = 0;
int controlNew = 1;
int val = 0;
const int ledRoom = 11;
int a = 0;
int b = 0;
int c = 0;
const int pushPassword = 6;
const int pushConfirm = 13;
int stage = 1;
bool readyForNextStage = true;
const unsigned long debounceDelay = 200;
unsigned long lastPasswordPress = 0;
unsigned long lastConfirmPress = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  Serial.begin(9600);
  pinMode(pushDoor, INPUT);
  servo_3.attach(3, 500, 2500);

  pinMode(doorbell, INPUT);
  pinMode(bell, OUTPUT);

  pinMode(ldrPin, INPUT);
  pinMode(ledBalcony, OUTPUT);

  pinMode(controlLight, INPUT);
  pinMode(ledPin, OUTPUT);

  pinMode(ledRoom, OUTPUT);

  pinMode(pushPassword, INPUT);
  pinMode(pushConfirm, INPUT);
}

void loop()
{
  int pushUpstair = digitalRead(pushDoor);
  if (pushUpstair == HIGH)
  {
           if (toggle)
           {
      servo_3.write(0);
      toggle =! toggle;
    }
    else
    {
       servo_3.write(90);
              toggle =! toggle;
    }
  }
  delay(100);

  bellState = digitalRead(doorbell);
   if (bellState == HIGH)
   {
     for (int i = 0; i < 3; i++)
     {
      tone(bell, 1648);
      delay(400);
      tone(bell, 1960);
      delay(400);
    tone(bell, 1308);
    delay(400);
    noTone(bell);
    delay(1000);
    }
}

int ldrStatus = analogRead(ldrPin);
/*Serial.print("Light intensity = ");
Serial.println(ldrStatus);
delay(100);*/
if (ldrStatus <= 100)
{
  digitalWrite(ledBalcony, HIGH);
}
else
{
  digitalWrite(ledBalcony, LOW);
}

int controlNew = digitalRead(controlLight);
if(controlOld == 0 && controlNew == 1)
{
  if (ledState == 0)
  {
    digitalWrite(ledPin, HIGH);
    ledState = 1;
  }
  else
  {
    digitalWrite(ledPin, LOW);
    ledState = 0;
  }
}
controlOld = controlNew;
delay(100);

val = 0.01723 * readUltrasonicDistance(9, 10);
/*Serial.print("Distance = ");
Serial.println(val);
delay(100);*/
if (val < 10)
{
  digitalWrite(ledRoom, HIGH);
}
else
```

```
  {
    digitalWrite(ledRoom, LOW);
  }
  delay(10);

  int passwordPin = digitalRead(pushPassword);
  int confirmPin = digitalRead(pushConfirm);
  unsigned long currentMillis = millis();

  if (passwordPin == HIGH && currentMillis - lastPasswordPress > debounceDelay) {
    lastPasswordPress = currentMillis;
    incrementCounter();
  }

  if (confirmPin == HIGH && currentMillis - lastConfirmPress > debounceDelay) {
    lastConfirmPress = currentMillis;
    handleStageTransition();
  }
}

void incrementCounter()
{
  if (stage == 1)
  {
    a++;
    Serial.print("a: "); Serial.println(a);
  }
  else if (stage == 2)
  {
    b++;
    Serial.print("b: "); Serial.println(b);
  }
  else if (stage == 3)
  {
    c++;
    Serial.print("c: "); Serial.println(c);
  }
}

void handleStageTransition()
{
  Serial.print("Current Stage: "); Serial.println(stage);
  Serial.print("Counters - a: "); Serial.print(a);
  Serial.print(", b: "); Serial.print(b);
  Serial.print(", c: "); Serial.println(c);
  if (stage == 1 && a > 0)
  {
    stage = 2;
    Serial.println("Moving to Stage 2");
  }
  else if (stage == 2 && b > 0)
  {
    stage = 3;
    Serial.println("Moving to Stage 3");
  }
  else if (stage == 3)
  {
    checkSequence();
  }
}

void checkSequence()
{
  Serial.print("Checking Sequence: a="); Serial.print(a);
  Serial.print(", b="); Serial.print(b);
  Serial.print(", c="); Serial.println(c);

  if (a == 1 && b == 2 && c == 3)
  {
    Serial.println("Sequence Correct. Activating Servo.");
    servo_3.write(0);
    delay(1000);
  }
  else
  {
    Serial.println("Sequence Incorrect.");
  }
  a = b = c = 0;
  stage = 1;
}
```