System Design Document

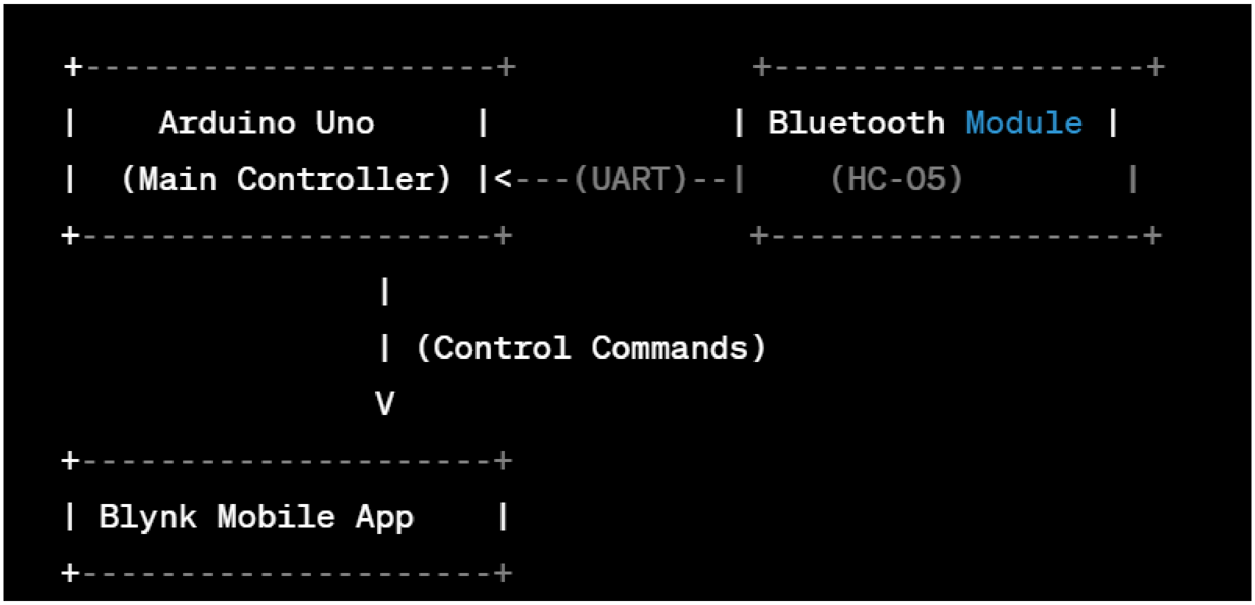
1. Design Options:

For the Capstone project, we evaluated two main system design options for the Smart Home Automation System using Arduino. Each design option focuses on different hardware components and communication protocols to achieve the project's objectives.

Option 1: Arduino Uno with Bluetooth Module

Description: This design option involves using Arduino Uno as the main controller for home automation, paired with a Bluetooth module for communication. The Bluetooth module facilitates the interaction between the Arduino and the mobile application.

Block Diagram:

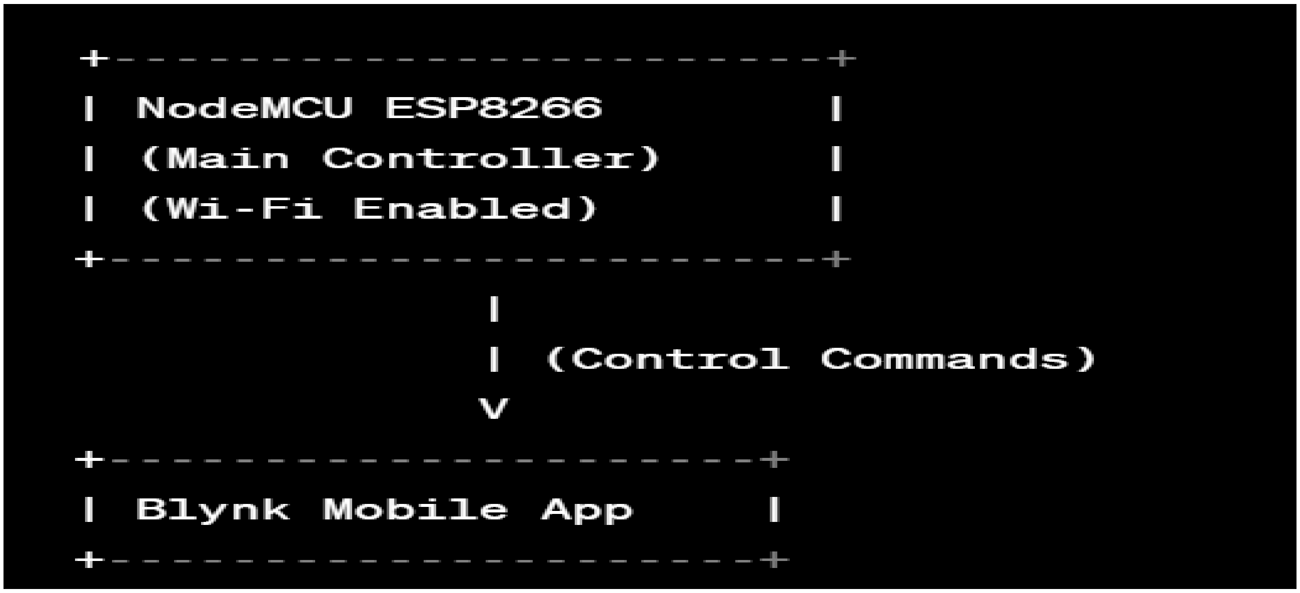


```
+--------------------+           +-------------------+
|    Arduino Uno     |           | Bluetooth Module  |
|  (Main Controller) |<---(UART)--|    (HC-05)        |
+--------------------+           +-------------------+
               |
               | (Control Commands)
               V
+--------------------+
| Blynk Mobile App   |
+--------------------+
```

Option 2: NodeMCU ESP8266 with Wi-Fi

Description: This design option utilizes NodeMCU ESP8266 as the main controller, eliminating the need for a separate Bluetooth module. The NodeMCU ESP8266 connects directly to Wi-Fi, enabling communication with the Blynk mobile app.

Block Diagram:



```
+-------------------------+
| NodeMCU ESP8266         |
| (Main Controller)       |
| (Wi-Fi Enabled)         |
+-------------------------+
            |
            | (Control Commands)
            V
+---------------------+
| Blynk Mobile App    |
+---------------------+
```

2. Option Evaluation:

Option 1 Evaluation: Bluetooth-Based System with Arduino Uno

- *Pros:*
  - Established and widely used technology.
  - Cost-effective solution.
- *Cons:*
  - Limited range due to Bluetooth technology.
  - Potential interference in crowded environments.
- *Metrics Considered:*
  - Cost
  - Range
  - Interference
  - Availability

Option 2 Evaluation: Wi-Fi-Based System with NodeMCU ESP8266

- *Pros:*
  - Higher communication range due to Wi-Fi.
  - No additional cost for a separate Bluetooth module.
  - More stable communication in crowded environments.
- *Cons:*
  - Slightly higher cost compared to Option 1.
  - Slightly more complex setup due to Wi-Fi configuration.

- *Metrics Considered:*
    - Cost
    - Range
    - Complexity
    - Availability

3. Option Choice:

The decision to select Option 2, the Wi-Fi-Based System with NodeMCU ESP8266, was based on the evaluation metrics. While Option 1 was cost-effective, the enhanced communication range and stability offered by Wi-Fi in Option 2 outweighed the marginal increase in cost and complexity. Additionally, the availability and support for NodeMCU ESP8266 made it a suitable choice for achieving the project's goals. Therefore, Option 2 was chosen as the preferred system design for the Smart Home Automation System.

```
#define BLYNK_TEMPLATE_ID "TMPL6sTVx2PsK"
#define BLYNK_TEMPLATE_NAME "Home Automation System"
#define BLYNK_AUTH_TOKEN "wZD5yH4k8z3I9-3xIMEA6fOVE6vquafj"

#define BLYNK_PRINT Serial
#include <gpio.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleDHT.h>

int pinDHT11 = D0;
int Kfpin = D1;
int Kled = D2;
int Rfpin = D3;
int Rled1 = D4;
int Rled2 = D5;
int TVled1 = D6;
int TVled2 = D7;
int Wled = D8;


SimpleDHT11 dht11(pinDHT11);


char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "HASEEB"; // Your Wifi Name
char pass[] = "karachi00201"; // Your Wifi Password

BLYNK_WRITE(V2){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(Rled1,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V2 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V3){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(Rled2,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V3 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V4){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(Kled,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V4 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V5){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(TVled1,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V5 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V6){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(TVled2,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V6 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V9){
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  analogWrite(Wled,pinValue);
  //Blynk.virtualWrite(V0, pinValue);
  Serial.print("V9 Slider value is: ");
  Serial.println(pinValue);

}
BLYNK_WRITE(V7){     // room fan   relay
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a
variable
  Serial.print("V7 Slider value is: ");
  Serial.println(pinValue);
  if(pinValue == 1)
  {
    digitalWrite(Rfpin, LOW);
    Serial.println("Room Fan On, relay1");
  }
  if(pinValue == 0)
  {
    digitalWrite(Rfpin, HIGH);
    Serial.println("Room Fan Off, relay1");
  }
```

```
}
BLYNK_WRITE(V8){    //kitchen fan relay
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  Serial.print("V8 Slider value is: ");
  Serial.println(pinValue);
  if(pinValue == 1)
  {
    digitalWrite(Kfpin, LOW);
    Serial.println("Kitchen Fan On, relay2");
  }
  if(pinValue == 0)
  {
    digitalWrite(Kfpin, HIGH);
    Serial.println("kitchen Fan Off, relay2");
  }

}



void setup(){
  // Debug console
  Serial.begin(9600);
  pinMode(Kfpin,OUTPUT);
  pinMode(Kled,OUTPUT);
  pinMode(Rfpin,OUTPUT);
  pinMode(Rled1,OUTPUT);
  pinMode(Rled2,OUTPUT);
  pinMode(TVled1,OUTPUT);
  pinMode(TVled2,OUTPUT);
  pinMode(Wled,OUTPUT);
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  delay(500);
  Serial.println("DHT11 Humidity &    Temperature Sensor\n\n");
  delay(1000);

}

void loop(){
  byte temperature = 0;
  byte humidity = 0;
  dht11.read(pinDHT11, &temperature, &humidity, NULL);
  Serial.print(" Temperature & Humidity : ");
  Serial.print((int)temperature);
  Serial.print(" *C  ");
  Serial.print((int)humidity);
  Serial.println(" % H");
  // SEND the sensor data to blynk app
  Blynk.virtualWrite(V0,temperature);
  Blynk.virtualWrite(V1,humidity);


   if(temperature > 30 & humidity > 60 ){
     //   digitalWrite(Kfpin,LOW);
        Serial.println("on kitchen fan");
   }else{
       //s digitalWrite(Kfpin,HIGH);
        Serial.println("off kitchen fan");
   }

  delay(2000);

  Blynk.run();

}
```

Conclusion:

The selection of Option 2 aligns with the project's objectives of providing a reliable and scalable smart home solution. The higher upfront cost and minimal complexity associated with Wi-Fi connectivity are justified by the improved performance and user experience offered by this design option.