



**SAKARYA**  
ÜNİVERSİTESİ

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ  
FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
NESNELERİN İNTERNETİ ve  
UYGULAMALARI  
PROJE ÖDEVİ**

**Mümtaz DANACI  
B211210370  
1. Öğretim B Grubu**

# DUMAN ALGILADIĞINDA KULLANICIYA BİLDİRİM GÖNDEREN PROJE

Çeşitli nedenler evde ya da kapalı alanlarda oluşan duman özellikle gece saatlerinde insanlara ciddi sağlık sorunları yaratabilmektedir. Ve duman aynı zamanda bir yangınında habercisi olabilir. Bu sebepler nedeni ile dumanın erkenden tespiti hayati önem taşımaktadır. Geliştirdim proje kapalı alanda oluşan duman tespit edilip belli bir seviyenin üzerine çıktıktan sonra kullanıcı ya da kullanıcının belirlediği başka bir kişiye hem telefonla arama hemde telefona bildirim gönderme yolu ile bilgilendirilecektir. Bu noktada telefonla arama önem arz etmektedir. Çünkü normal bildirimler her zaman kullanıcı tarafından farkedilir olmayabilir. Ancak bir telefon araması çok daha etkili bir yöntemdir. Bu proje sadece bir kişiyi aramakla kalmayıp birden fazla yere bildirim göndermeye evrilebilmeye çok müsaittir. Örneğin eğer kullanıcı isterse hem 112 gibi acil durumda aranacak yerlere proje yardımıyla otomatik olarak haber verebilir. Yada eğer isterse birden fazla kişi telefonla aranabilir.

Bu projede yangını tespit etmek için NodeMCU modulu ve duman sensörü donanım bileşenleri kullanıldı.

## GEREKLİ BİLEŞENLER

- NodeMCU ESP8266
- Direnç 1k
- Breadboard
- İletken kablolar
- Arduinio IDE
- Blynk Uygulaması
- IFTTT Uygulaması

- Yanıcı Gaz ve Duman Sensörü

## **NodemCU ESP8266**

ESP8266 Nodemcu açık kaynaklı bir IoT platformudur. Espressif Systems'den düşük maliyetli Wi-Fi özellikli ESP8266 Wi-Fi SoC ve ESP-12 modülünü temel alan donanım üzerinde çalışan bellekimi içerir. İletişim ve ona bağlı diğer çevre birimlerini kontrol etmek için GPIO, SPI, I2C, ADC, PWM VE UART pinlerine sahiptir. Gemide NodeMCU, USB'den TTL'ye işlev sağlayan CP2102 IC'ye sahiptir.

## **Yanıcı Gaz ve Duman Sensörü**

MQ-2 gaz sensörü 300 ile 10.000 ppm konsantrasyonlarda yanıcı gaz ve sigara dumanını algılayacak türden bir sensördür. MQ-2 gaz sensörü 300 ile 10.000 ppm konsantrasyonlarda yanıcı gaz ve sigara dumanını algılayacak türden bir sensördür. MQ2 yanıcı gaz ve duman sensörü, havadaki yanıcı gazları algılar ve çıktığı analog voltaj olarak verir. Sensör, 300 ile 10.000 ppm arasında konsantrasyona sahip yanıcı gazları ölçebilir, -20 dereceden 50 dereceye kadar olan sıcaklık aralığında çalışabilir ve tüketimi 5V'ta 150mA'den azdır.

## **Blynk Uygulaması**

Blynk uygulaması iOS ve Android'te mevcuttur. Blynk ile Arduino, Raspberry Pİ gibi benzer kartları kontrol etmenize olanak sağlayan bir platformdur.

Blynk nesnelerin interneti için tasarlanmıştır. Bağlı donanımları uzaktan kontrol edebilir, sensör verilerini okuyabilir, verileri görselleştirebilir, depolayabilir ve bir çok daha güzel şeyler yapabilir.

## Çalışma ve Kod Açıklaması

Öncelikle gerekli kütüphaneleri eklendi ve Wi-Fi bağlantısını başlatmak için WiFi SSID ve parola değişkenlerini tanımlandı. SMTP sunucusuna bağlanması gerektiği için, sunucunun adresi ile birlikte bir değişken bildirildi. Blynk uygulamasından gelen token auth listesine atandı. Bu sayede blynkte ilgili uygulamaya bağlanabiliyoruz. Ve IFTTT uygulamasına erişmek içinde ilgili kod satırları mevcuttur.

```
#define BLYNK_PRINT Serial

char auth[] = "GQgx0nGCCCCJ68wPX10jPRSlq9YSOySKK";
char ssid[] = "FiberHGW_TPFC5E_2.4GHz";
char pass[] = "cbkpjpXL";

String MakerIFTTT_Key ;
;String MakerIFTTT_Event;
char *append_str(char *here, String s) { int i=0; while (*here++ = s[i]){i++;};return here-1;}
char *append_ul(char *here, unsigned long u) { char buf[20]; return append_str(here, ultoa(u, buf, 10));}
char post_rqst[256];char *p;char *content_length_here;char *json_start;int compi;

BlynkTimer timer;
WiFiClient client;

int GazSensoru = A0; // A0 pinine
int data = 0;
|
```

Setup ve loop fonksiyonları da şekildeki gibidir.

```
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(Yesilled, OUTPUT);
  pinMode(Kirmiziled, OUTPUT);
  timer.setInterval(1000L, getSendData);
}

void loop()
{
  timer.run();
  Blynk.run();
}
```

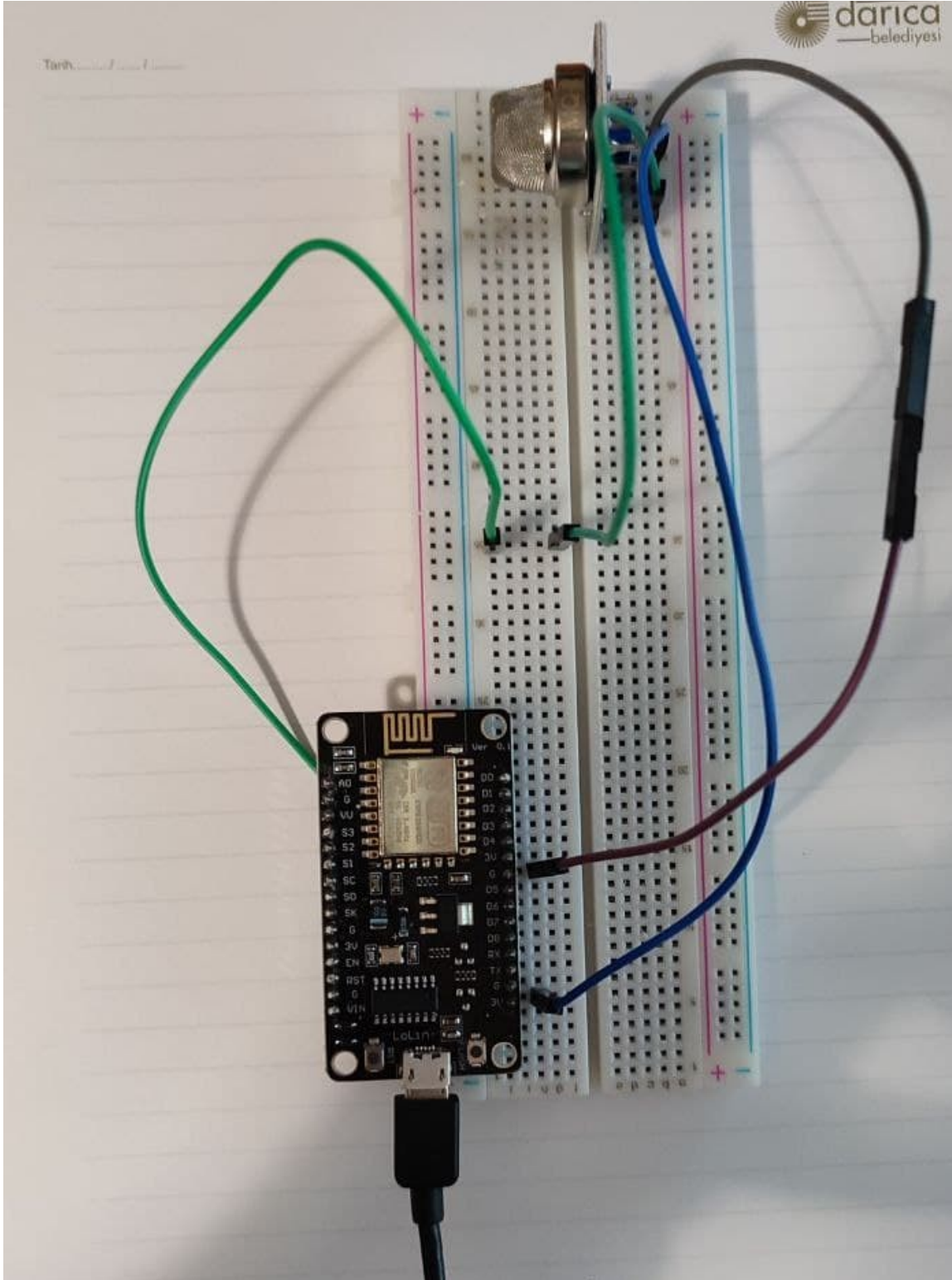
Ve son olarak verinin işlendiği ve kaydedildiği kısım sekildeki gibidir. İçerisinde IFTTT ve Blynk uygulamalarını tetikleyen kodlar bulunmaktadır.

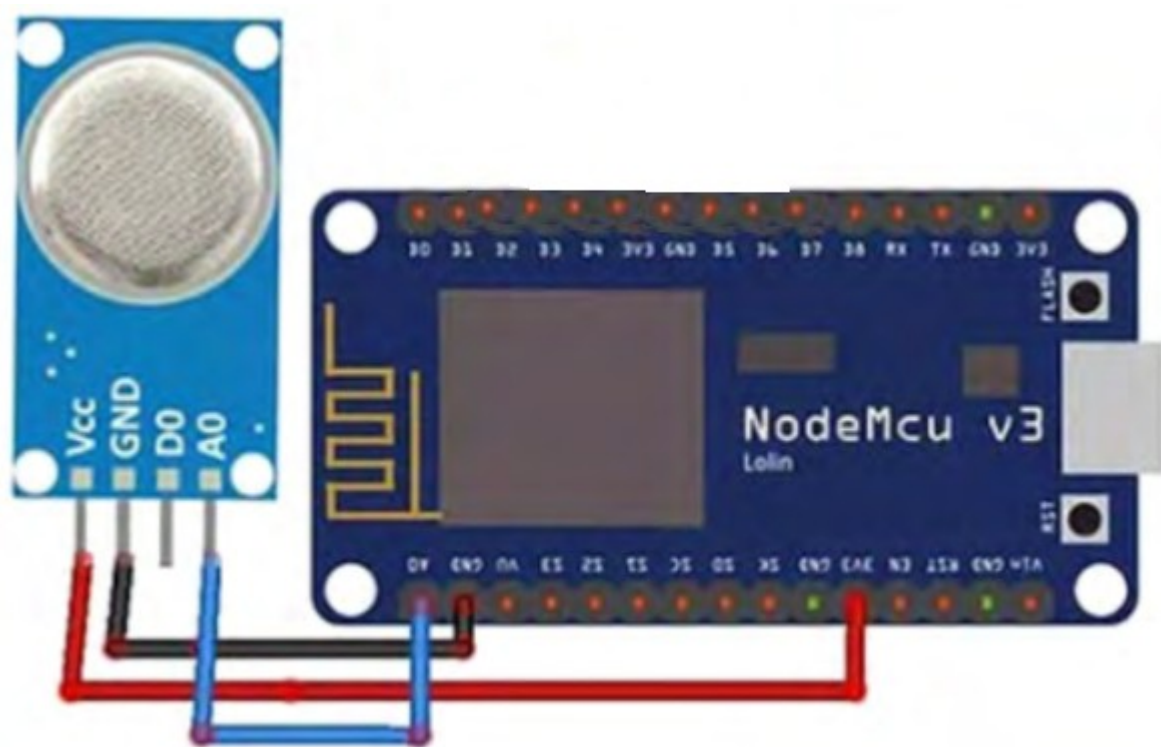
```
void getSendData()
{
    data = analogRead(GazSensoru);
    Blynk.virtualWrite(V2, data);

    if (data > 490)
    {
        Blynk.notify("Dikkat !!! Gaz Sensörünüz Çalıştı");
        digitalWrite(Yesilled, LOW);
        digitalWrite(Kirmiziled, HIGH);
        tone(buzzer, 1000, 200);
        delay(6000);
        if (client.connect("maker.ifttt.com",80)) {
            MakerIFTTT_Key ="he8OcYltwZAkb8tl3ALv0";
            MakerIFTTT_Event ="ESP_CALL";
            p = post_rqst;
            p = append_str(p, "POST /trigger/");
            p = append_str(p, MakerIFTTT_Event);
            p = append_str(p, "/with/key/");
            p = append_str(p, MakerIFTTT_Key);
            p = append_str(p, " HTTP/1.1\r\n");
            p = append_str(p, "Host: maker.ifttt.com\r\n");
            p = append_str(p, "Content-Type: application/json\r\n");
            p = append_str(p, "Content-Length: ");
            content_length_here = p;
            p = append_str(p, "NN\r\n");
            p = append_str(p, "\r\n");
            json_start = p;
            p = append_str(p, "{\"value1\":\"");
            p = append_str(p, "DUMAN DUMAN UYANN DUMANN VARR");
            p = append_str(p, "\",\"value2\":\"");
            p = append_str(p, "");
            p = append_str(p, "\",\"value3\":\"");
            p = append_str(p, "");
            p = append_str(p, "\"}");

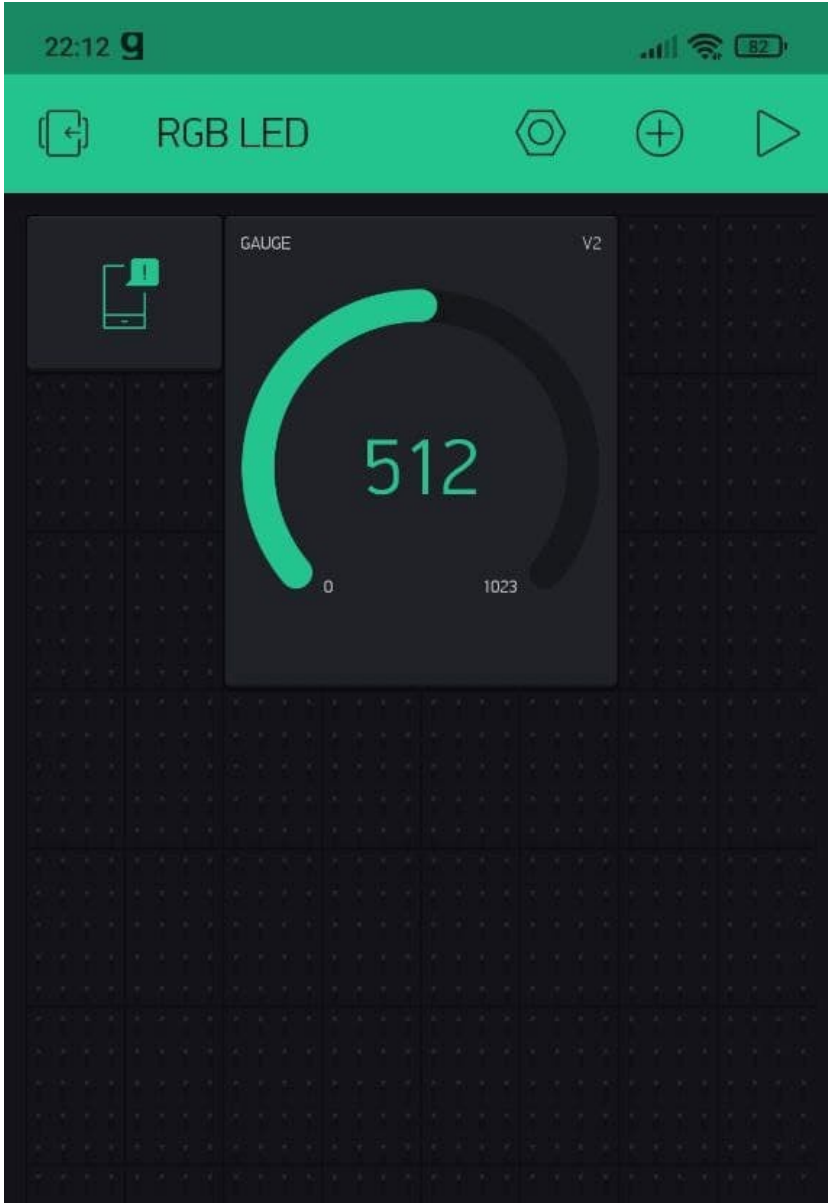
            compi= strlen(json_start);
            content_length_here[0] = '0' + (compi/10);
            content_length_here[1] = '0' + (compi%10);
            client.print(post_rqst);
        }
    }
}
```

## Devre Şeması



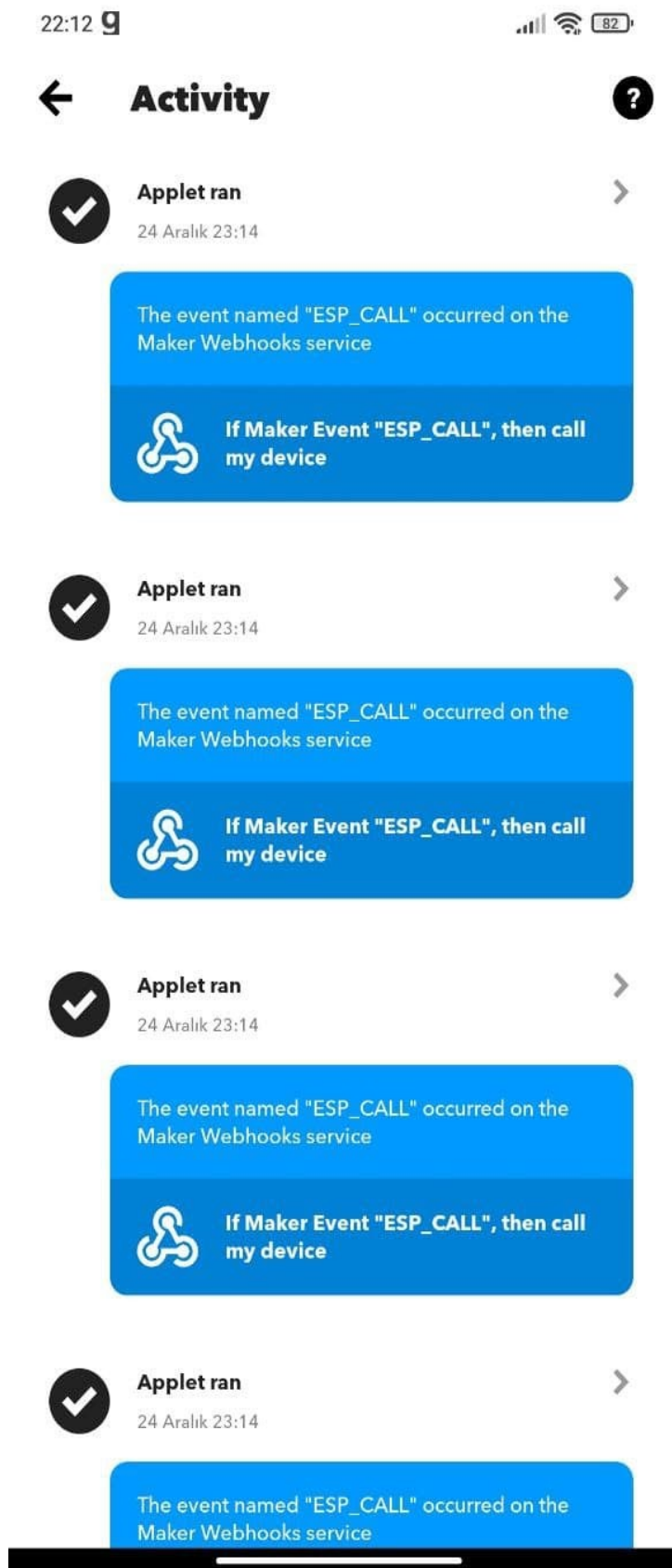


## Blynk Arayüzü

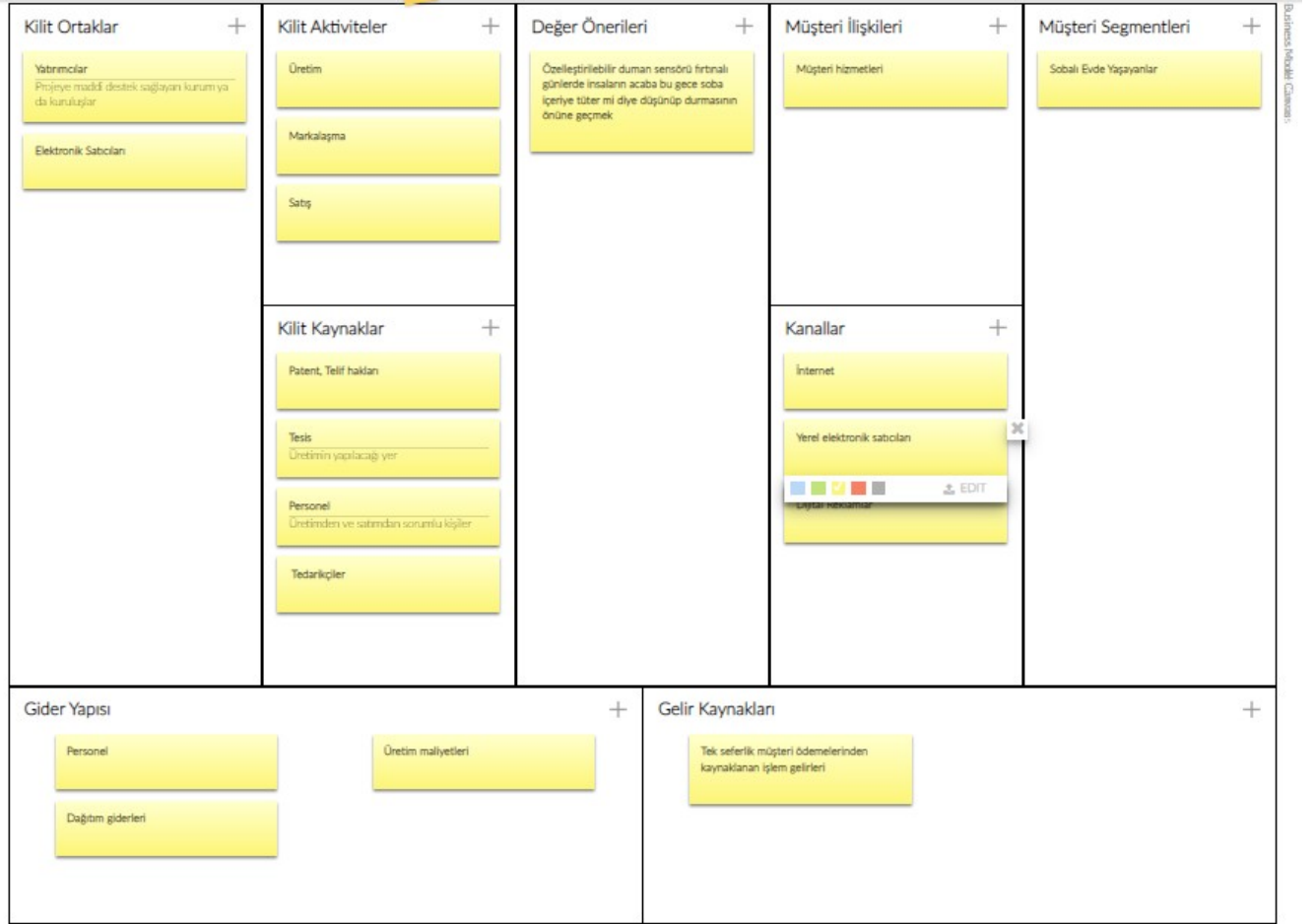




## IFTTT



# İş Kanvası



Link: <https://next.canvanizer.com/canvas/rrTQC0ycFT0SR>

## UML Diagram

