### **Test Case #1 – Ses Kaydı ve Dosya Oluşturma**

* **Test Adı**: Ses Kaydı Fonksiyonu
* **Test ID**: TC01
* **Ön Koşullar**:
  + Mikrofon bilgisayara bağlı ve çalışır durumda olmalı.
  + Uygulama GUI (Tkinter) açılmış olmalı.
* **Adımlar**:
  + Uygulamadaki “Konuş ve Tahmin Et” butonuna tıkla.
  + Ekrana gelen mesaj kutusunda (MessageBox) “Kayıt başlıyor…” uyarısını onayla.
  + 7 saniye boyunca konuş ve kayıt süresinin bitmesini bekle.
  + “Kayıt tamamlandı” mesaj kutusunu onayla.
* **Beklenen Sonuç**:
  + realtime\_audio.wav adında bir dosya oluşur.
  + Dosya boyutu 0’dan büyük olmalıdır (içinde ses verisi olmalı).
  + Uygulama kapanmadan devam etmeli.
* **Gerçek Sonuç**: (Test esnasında doldurulacak.)

### **Test Case #2 – Kişi Tanıma Modülü**

* **Test Adı**: Kişi Tanıma İşlevi
* **Test ID**: TC02
* **Ön Koşullar**:
  + person\_identifier.pkl dosyası var ve doğru dizinde (exe veya aynı klasör) bulunmalı.
  + realtime\_audio.wav içinde kayıt edilmiş bir kullanıcının sesi (veritabanında tanımlı olması gerekir).
* **Adımlar**:
  + “Konuş ve Tahmin Et” butonuna basarak yeni kayıt al ya da daha önce oluşturulmuş realtime\_audio.wav dosyasını kullan.
  + Uygulama, kişi tanıma fonksiyonunu (identify\_person) otomatik çağıracak.
  + Kişi tanıma sonucu arayüzde “Tanımlanan Kişi: ...” şeklinde görünecek.
* **Beklenen Sonuç**:
  + person\_label üzerinde “Tanımlanan Kişi: <beklenen isim>” yazmalıdır.
  + Tanımlanan kişinin ismi, eğitim setindeki modele uygun şekilde olmalıdır (ör. “Laura”, “Liam” vb.).
  + Uygulama hata vermeden devam etmeli.
* **Gerçek Sonuç**: (Test esnasında doldurulacak.)

### **Test Case #3 – Ses Duygu Analizi ve Metin Duygu Analizi**

* **Test Adı**: Duygu Analizi (Audio + Metin)
* **Test ID**: TC03
* **Ön Koşullar**:
  + emotion\_classifier.pkl dosyası var ve doğru dizinde bulunmalı.
  + Uygulamada “ses duygu analizi” ve “metin duygu analizi” fonksiyonları eklenmiş olmalı.
* **Adımlar**:
  + Uygulamadaki “Konuş ve Tahmin Et” butonuna bas.
  + 7 saniye konuşurken belirli bir duygu ifade etmeye çalış (örn. mutlu bir tonlama).
  + Kayıt bittiğinde, ses duygu analizi devreye girecek ve arayüzde “Tahmin Edilen Ses Duygusu: ...” sonucunu göstermesi beklenir.
  + Konuşulan metin Google Speech API aracılığıyla metne dönüştürülür ve “Tahmin Edilen Metin Duygusu: ...” alanında bir duygu çıkacaktır.
* **Beklenen Sonuç**:
  + Ses duygu analizi: “happy” (örnek).
  + Metin duygu analizi: Konuşulan kelimeler “mutlu, harika, mükemmel” vb. içeriyorsa, “happy” veya “neutral” olabilir.
  + Duygu sonuçları arayüzde görünecek, ayrıca gTTS ile sesli olarak söylenecek.
  + Hata veya “model bulunamadı” gibi bir sorun oluşmamalı.
* **Gerçek Sonuç**: (Test esnasında doldurulacak.)

### **Test Case #4 – Konuşmayı Metne Dönüştürme ve Konu Analizi**

* **Test Adı**: Speech-to-Text ve Topic Analysis
* **Test ID**: TC04
* **Ön Koşullar**:
  + İnternet bağlantısı olmalı (Google Speech API kullanıldığı varsayılıyor).
  + nltk kütüphanesi, stopwords paketleri vb. doğru kurulmuş olmalı.
  + Konu analizi (topic analysis) fonksiyonunda TF-IDF ve nltk.tokenize.ToktokTokenizer doğru çalışır durumda olmalı.
* **Adımlar**:
  + “Konuş ve Tahmin Et” butonuna basarak 7 saniye boyunca **teknoloji** odaklı bir cümle söyle (ör. “Bugün yapay zekâ ve yazılım teknolojileri üzerine çalışıyoruz.”).
  + Ses kaydı tamamlanınca konuşma Google Speech API ile metne dönüştürülecek.
  + Uygulama “Konu Analizi” işlevini çağırarak anahtar kelimeleri ve konuyu belirleyecek (görevde “topic\_label”/“topic\_result\_label” vb. alanlar varsa orada görünecek).
* **Beklenen Sonuç**:
  + Metin alanında: “bugün yapay zekâ ve yazılım teknolojileri üzerine çalışıyoruz” benzeri bir yazı.
  + Anahtar kelimeler: “yapay, zekâ, yazılım, teknolojileri, …” vb.
  + Konu: “Bilim ve Teknoloji” veya “Genel Konu” şeklinde tahmin.
  + Uygulama kapanmaz, hata vermez.
* **Gerçek Sonuç**: (Test esnasında doldurulacak.)

### **Test Case #5 – Grafik Oluşturma (ACC-FM)**

* **Test Adı**: Plot ACC-FM Fonksiyonu
* **Test ID**: TC05
* **Ön Koşullar**:
  + Matplotlib kütüphanesi kurulu olmalı.
  + Uygulamadaki “plot\_acc\_and\_fm” fonksiyonu aktif ve hata vermeden çalışabilir olmalı.
* **Adımlar**:
  + “Konuş ve Tahmin Et” butonuna basarak bir ses kaydı al.
  + Uygulama otomatik olarak duygu analizi, kişi tanıma vb. aşamalardan sonra grafikleri oluşturacak.
  + ACC (dalga formu) ve FM (frekans modülasyonu) penceresi otomatik olarak açılacak.
* **Beklenen Sonuç**:
  + İki alt grafik içeren bir Matplotlib penceresi açılmalı:
    - Dalga formu (Zaman x Amplitüd).
    - Frekans spektrumu (Frekans x Genlik).
  + Grafikler kapatılmadıkça uygulama devam etmeli, kapatıldığında da uygulama normal çalışmaya devam etmeli.
* **Gerçek Sonuç**: (Test esnasında doldurulacak.)

## **3. Test Ortamı ve Gereksinimler**

* **İşletim Sistemi**: Windows 10 veya üzeri
* **Python Sürümü**: 3.9 veya 3.10 (proje ortamına göre)
* **Ek Kütüphaneler**: numpy, matplotlib, librosa, speech\_recognition, gTTS, nltk, sklearn, joblib, vb.
* **Donanım**: Mikrofon, hoparlör veya kulaklık (gTTS sesini duymak için), İnternet bağlantısı (Google Speech API)

## **4. Beklenen Sonuçlar**

* Tüm testler başarıyla geçildiğinde; kişi tanıma, ses/metin duygu analizi, konu analizi, metin çıktısı ve grafik fonksiyonlarının sorunsuz çalıştığı doğrulanacak.
* Hata durumlarında, uygulama kullanıcıya anlamlı mesajlar gösterecek ve kapanmadan devam edecek.