### 1. ****Gelişmiş Hava Trafiği Kontrol Sistemi****

1. **UAV İzleme Modülü:**
   * **Sensörler ve Kameralar**:
     1. Uçakların iniş, kalkış ve taksi süreçlerini takip edecek kameralar (örneğin ESP32 CAM) ve ultrasonik sensörler gibi uzaklık ölçen cihazlar kullanılacak.
     2. **Veri İşleme**: Kameralar ve sensörlerden gelen veriler, düşük gecikmeli bir veri işleme sistemi (ör. ESP32) tarafından işlenir ve merkeze iletilir.
2. **Trafik Veri Toplama:**
   * **Veri Depolama**: Toplanan tüm veriler bir merkezi sunucuda (ör. Raspberry Pi veya bulut tabanlı bir çözüm) depolanarak analiz için kullanılacak.
   * **Veri Entegrasyonu**: Hava trafiği yönetim modülü, uçakların hava trafiğindeki pozisyonunu ve taksi sırasında konumlarını gerçek zamanlı olarak izleyip yönetebilecek.
3. **API Entegrasyonu:**
   * **Hava Trafiği Kontrol API**: Mevcut hava trafiği kontrol sistemlerinden (örn. ADS-B verileri) uçuş bilgilerini almak ve UAV'ların karar verme süreçlerinde kullanmak için açık API'ler kullanılacak.
   * **İki Yönlü İletişim**: UAV'lar ile merkezi sistem arasındaki veri paylaşımını sağlamak için RESTful API'ler veya MQTT tabanlı hafif haberleşme protokolleri kullanılabilir.
4. **Hızlı İletişim Ağı:**
   * **nRF24L01 Modülleri**: Düşük gecikmeli ve uzun menzilli iletişim için nRF24L01 modülleri kullanılacak. Bu modüller, UAV'lar arasında ve UAV’lar ile kontrol merkezi arasında yüksek hızlı veri aktarımı sağlayacak.
   * **Ağ Yapısı**: Merkezi bir haberleşme ağı (mesh network) kurulacak. Böylece UAV'lar birbirleriyle ve yer istasyonlarıyla sorunsuz bir şekilde iletişim kurabilecek.
   * **Gerçek Zamanlı İletişim**: Anlık pozisyon ve trafik verileri, iletişim ağında yüksek bant genişliği ve düşük gecikme ile iletilir.

### 2. ****Otokontrol Yöntemleriyle Taksi ve Park Sistemi****

1. **Otomatik Kalkış ve İniş Sistemi:**
   * **Optimizasyon Algoritmaları**: Taksi yollarını ve park yerlerini optimize eden, en kısa ve güvenli yolu seçen algoritmalar geliştirilecek. Bu algoritmalar, uçakların konumunu, mevcut taksi yollarını ve trafik yoğunluğunu hesaba katarak UAV'lara park yeri atayacak.
   * **Otonom Hareket**: UAV'lar, park yerlerine doğru otonom hareket edebilmek için LIDAR (veya daha uygun maliyetli bir alternatif olan ultrasonik sensörler) ile taksi yollarını algılar ve yollarını çizer.
2. **Sensör Entegrasyonu:**
   * **ESP32 CAM**: Pist ve park alanlarını görsel olarak tanımak için ESP32 CAM modülleri kullanılır. Bu modüller, park yerinin dolu olup olmadığını veya pisti izleyerek UAV’ın taksi rotasını yönetir.
   * **Ultrasonik Sensörler**: UAV’ların yakındaki objeleri ve pist yüzeyini algılamasını sağlayacak ultrasonik sensörler kullanılarak çarpışma önlenir ve UAV’ın hareket güvenliği sağlanır.
3. **Motor ve Kontrol Sistemi:**
   * **Servo Motorlar**: Hassas hareket için yönlendirme ve hız ayarlaması yapabilen servo motorlar kullanılır. UAV'lar bu motorlar sayesinde kalkış, iniş ve park esnasında tam kontrol sağlayabilir.
   * **Motor Kontrolörleri**: Fırçasız motorlar ve PWM motor kontrol devreleri ile UAV'ların hızları ve yönleri hassas şekilde kontrol edilir.

### 3. ****Hava Durumu İzleme Modülü****

1. **SDR Anten ve API Kullanımı:**
   * **SDR Anten (Software Defined Radio)**: Yerel hava durumu verilerini toplamak için SDR antenleri kullanılacak. Bu antenler, yerel meteorolojik verileri toplayacak ve analiz edecek.
   * **API Entegrasyonu**: İnternetten gelen hava durumu verileri, Meteoroloji Genel Müdürlüğü veya OpenWeatherMap API'leri ile entegrasyon sağlanarak UAV’ların planlamasına entegre edilecek.
2. **Veri Toplama ve Analiz:**
   * **Sensör Verileri**: SDR antenle toplanan hava durumu verileri (sıcaklık, basınç, rüzgar hızı vb.) bir veri işleme modülü (ör. Raspberry Pi) üzerinde işlenerek uçuş planlama modüllerine aktarılır.
   * **Veri Entegrasyonu**: Hava durumu verileri, uçuş operasyonlarına ve taksi rotalarına gerçek zamanlı olarak entegre edilir.
3. **Anlık Uyarı Sistemi:**
   * **Hava Durumu Değişiklikleri**: Anlık hava durumu değişikliklerinde (ör. ani rüzgar artışı, yağış) UAV'lar ve kontrol merkezi uyarılır. Bu sistem, hava durumu verilerini sürekli izler ve kritik durumlarda alarm üretir.

### 4. ****Veri Toplama ve Analiz****

1. **Veritabanı:**
   * **Bulut veya Yerel Veritabanı**: Hava trafiği, hava durumu ve güvenlik verileri bulut tabanlı bir veritabanında (örn. AWS RDS) veya yerel bir sunucuda (örn. MySQL veya MongoDB) depolanacak.
   * **Veri Yedekleme**: Kritik verilerin yedeklenmesi için bulut tabanlı depolama veya RAID konfigürasyonları kullanılabilir.
2. **Veri Analizi Yazılımı:**
   * **Büyük Veri Analitiği**: Hava trafiği ve hava durumu verilerinin analizi için Python veya R tabanlı büyük veri analiz araçları kullanılacak. Örneğin, makine öğrenimi algoritmaları, gelecekteki hava durumu tahminlerini ve trafiği optimize etmek için kullanılabilir.
   * **Gerçek Zamanlı Analiz**: Sistem gerçek zamanlı olarak veri analizi yapar ve bu verileri kontrolörlere raporlar.
3. **Raporlama Sistemi:**
   * **Grafik ve Tablo Modülleri**: Verilerin görselleştirilmesi için matplotlib, Tableau veya PowerBI gibi araçlar kullanılarak grafikler ve tablolar oluşturulacak.
   * **Rapor Üretimi**: Günlük, haftalık ve aylık raporlar oluşturacak bir modül geliştirilecek. Bu raporlar, hava trafiği yönetimini iyileştirmek için veri sağlayacak.

### 5. ****Pilot Proje Uygulaması****

1. **Simülasyon Yazılımı:**
   * **Simülasyon Platformu**: UAV'ların ve hava trafiği kontrol sistemlerinin test edilebileceği bir simülasyon platformu kurulacak. Örneğin, MATLAB Simulink veya Gazebo gibi araçlar kullanılabilir.
   * **Gerçek Zamanlı Testler**: Simülasyonlar, gerçek dünya koşullarına benzer senaryoları taklit edecek ve sistemin tepkilerini ölçecek.
2. **Test Modülleri:**
   * **Senaryo Bazlı Testler**: Simülasyon yazılımı üzerinde farklı senaryolar test edilecek (örneğin, yoğun trafik durumu, kötü hava koşulları). Test modülleri, sistemin güvenilirliğini ve etkinliğini değerlendirecek.
3. **Veri Toplama ve Analiz Aracı:**
   * **Test Verileri**: Pilot proje sırasında toplanan veriler analiz edilecek ve iyileştirme önerileri sunulacak. Bu veri analizi, gelecekteki operasyonlar için iyileştirme fırsatlarını belirlemek amacıyla kullanılacak.
4. **Hata Yönetimi Sistemi:**
   * **Otomatik Hata Kaydı**: Sistem hata meydana geldiğinde, hata kayıt altına alınır ve analiz edilir.
   * **Çözüm Önerileri**: Hataların analizine dayalı olarak, sistem otomatik çözüm önerileri veya iyileştirme raporları sunacak.