**RAMS Tasarım İlkelerine Göre Uygulama Tasarımı**

**RAMS** (Güvenilirlik - Reliability, Erişilebilirlik - Availability, Bakım Yapılabilirlik - Maintainability, Güvenlik - Safety) ilkeleri, uygulamanızın sağlam, sürekli erişilebilir, kolay yönetilebilir ve güvenli olmasını sağlamak için rehber niteliği taşır.

**1. Güvenilirlik (Reliability)**

**Amaç:** Uygulamanın kullanıcılar için her zaman sorunsuz çalışmasını sağlamak.

* **Kart Tarama Modülü**:
  + Kart tarama algoritmalarında hata toleransı ekleyin. Kart okunamıyorsa, kullanıcıya yeniden deneme seçeneği sunun.
  + Veri doğrulama mekanizmaları kullanarak yanlış veri girişlerini engelleyin.
* **Oyun Modülü**:
  + Görevler ve oyunlar sırasında hata oluşmasını önlemek için görev akışlarını kontrol eden bir mantık ekleyin.
* **Offline Mod**:
  + İnternet bağlantısı olmadığında yerel sunucu üzerinden tüm işlemlerin devam etmesini sağlayın.
  + Veri senkronizasyonu için yeniden bağlanıldığında otomatize bir sistem kullanın.

**2. Erişilebilirlik (Availability)**

**Amaç:** Kullanıcıların uygulamaya her an erişebilmesini sağlamak.

* **Çoklu Platform Desteği**:
  + Uygulamanın uçak içi ekranlarda, mobil cihazlarda ve web platformunda kullanılabilir olmasını sağlayın.
* **Sunucu Yapısı**:
  + Yerel sunucuyu yük dengeleme özellikleriyle yapılandırarak tüm cihazların performans kaybı olmadan bağlanmasını sağlayın.
  + Online modda bulut tabanlı yedekleme ve erişim sağlayarak sürekli kullanılabilirlik oluşturun.
* **Kesintisiz Çalışma**:
  + Sistem modüllerinde hata oluşursa diğer modüllerin etkilenmemesi için bağımsız süreçler oluşturun (örneğin, kart tarama başarısız olsa bile oyun modülü çalışmaya devam edebilir).

**3. Bakım Yapılabilirlik (Maintainability)**

**Amaç:** Sistem güncellemelerinin ve onarımlarının kolayca yapılabilmesini sağlamak.

* **Modüler Yapı**:
  + Kart tarama, oyun modülü, ödül yönetimi gibi bileşenleri bağımsız modüller olarak tasarlayın. Böylece bir modülde güncelleme yapılırken diğer modüller etkilenmez.
* **Loglama ve Hata İzleme**:
  + Hata ve kullanıcı davranışlarını kaydeden bir log sistemi entegre edin.
  + Hatalar için otomatik bildirim sistemi oluşturun.
* **Versiyon Kontrolü**:
  + Github gibi bir versiyon kontrol sistemi kullanarak kod tabanını düzenli olarak güncelleyin ve eski sürümlere kolayca erişim sağlayın.

**4. Güvenlik (Safety)**

**Amaç:** Kullanıcı verilerinin korunmasını ve güvenli işlem yapılmasını sağlamak.

* **Veri Şifreleme**:
  + Kart bilgileri ve kullanıcı verileri hem depolama sırasında hem de aktarım sırasında şifrelenmelidir (örneğin, HTTPS ve AES kullanımı).
* **Kullanıcı Yetkilendirme**:
  + Web sitesi girişinde kullanıcı doğrulama (e-posta ve şifre) ve çift faktörlü kimlik doğrulama ekleyin.
* **Erişim Kontrolleri**:
  + Havayolu personelinin erişebileceği yönetim ekranlarını diğer kullanıcıların erişemeyeceği şekilde kısıtlayın.
* **Geri Bildirim Modülü**:
  + Kullanıcıların geri bildirim formuna zararlı içerik eklemesini önlemek için içerik filtresi ekleyin.

**Tasarım Uygulama Planı**

1. **Başlangıç**:
   * Uygulama gereksinimlerini RAMS ilkelerine uygun şekilde analiz edin.
2. **Modül Geliştirme**:
   * Her modül için yukarıdaki ilkeler çerçevesinde tasarım ve geliştirme yapın.
3. **Test ve Optimizasyon**:
   * Her modül için güvenilirlik, erişilebilirlik ve güvenlik testleri gerçekleştirin.
   * Uygulamanın tüm modüllerinin birbirleriyle entegrasyonunu kontrol edin.
4. **Güncelleme ve Yönetim**:
   * Github ve bulut tabanlı sistemlerle düzenli güncellemeler yapın.
   * Hataları ve güncellemeleri yönetmek için sürekli izleme mekanizmaları kullanın.