

Randevu App Project Requirements Document

1. Giriş (Introduction)

- **Proje Adı:** Randevu App.
- **Açıklama:** Kuaför salonlarının randevu ayarlamalarını kolaylaştırmak, müşterilerin daha kolay ve pratik şekilde randevu almasını sağlayan saas.
- **Hedefler:** Kuaförlerin randevu karmaşıklığının önlenmesi.
- **Kapsam (Scope):** Proje, kuaför salon sahiplerini, kuaförleri ve müşterilerini kapsayan bir saas projesi.

2. Paydaşlar (Stakeholders)

- **Müşteriler:** Kuaför salon sahipleri, kuaförler ve müşteriler.
- **İç Paydaşlar:** Proje ekibi(ben) ve yapay zeka(sen)
- **Dış Paydaşlar:** Sms Api, Auth Services(maybe), Servers

3. Kullanıcı Hikayeleri (User Stories)

Kullanıcı Roller:

- **Müşteri:** Randevu almak isteyen son kullanıcı.
- **Kuaför:** Salonda hizmet veren personel.
- **Salon Sahibi:** Salonu yöneten, ayarları yapan ve raporları inceleyen yönetici.

Genel / Sistem

- **Bir kullanıcı olarak,** uygulamayı ilk açtığımda kısa bir açılış ekranı (splash screen) görmek istiyorum, böylece uygulamanın yüklendiğini anlayabilirim.
- **Bir Kuaför/Salon Sahibi olarak,** uygulamaya güvenli bir şekilde giriş yapabilmek istiyorum, böylece sadece yetkili olduğum alanlara erişebilirim.

- **Bir Kuaför/Salon Sahibi olarak**, uygulamadan güvenli bir şekilde çıkış yapabilmek istiyorum, böylece oturumum açık kalmaz.
- **Bir Kuaför/Salon Sahibi olarak**, uygulama içinde farklı ekranlar arasında kolayca gezinebilmek istiyorum (örn. yan menü ile), böylece istediğim işlemleri hızlıca yapabilirim.
- **Bir kullanıcı olarak**, bir işlem başarısız olduğunda veya bir hata oluştuğunda bilgilendirme mesajı (snackbar vb.) görmek istiyorum, böylece ne olduğunu anlayabilirim.
- **Bir kullanıcı olarak**, uygulamayı farklı ekran boyutlarındaki (telefon, tablet) cihazlarda rahatça kullanabilmek istiyorum, böylece arayüz bozuk görünmez.

Müşteri Rolü

- **Bir müşteri olarak**, kişisel bilgilerimi (ad/soyad, telefon, e-posta) ve isteğe bağlı notumu girerek randevu alma işlemine başlayabilmek istiyorum, böylece salon beni tanıyabilir.
- **Bir müşteri olarak**, salonda sunulan hizmetleri (fiyat ve süre bilgisiyle) listeleyebilmek ve almak istediğim bir veya birden fazla hizmeti seçebilmek istiyorum, böylece ihtiyacıma uygun hizmetleri belirleyebilirim.
- **Bir müşteri olarak**, salondaki aktif kuaförleri listeleyebilmek ve randevu almak istediğim kuaförü seçebilmek istiyorum, böylece tercih ettiğim kişiden hizmet alabilirim.
- **Bir müşteri olarak**, bir takvim üzerinden randevu almak istediğim tarihi seçebilmek istiyorum, böylece uygun olduğum bir günü belirleyebilirim.
- **Bir müşteri olarak**, seçtiğim tarih için uygun olan saat dilimlerini görebilmek ve birini seçebilmek istiyorum, böylece randevu saatimi netleştirebilirim.
- **Bir müşteri olarak**, tüm adımları tamamladıktan sonra randevumu oluşturabilmek istiyorum, böylece randevum sisteme kaydedilir.
- **Bir müşteri olarak**, randevum başarıyla oluşturulduğunda bir onay ekranı görmek ve randevu detayları ile birlikte benzersiz bir randevu kodu almak istiyorum, böylece randevumu takip edebilirim.
- **Bir müşteri olarak**, sahip olduğum randevu kodunu girerek randevumun detaylarını (tarih, saat, kuaför, hizmetler, durum) görüntüleyebilmek istiyorum,

böylece randevu bilgilerimi kontrol edebilirim.

- **Bir müşteri olarak**, sahip olduğum randevu kodunu kullanarak, belirli koşullar altında (örn. henüz tamamlanmamış/iptal edilmemişse) randevumu iptal edebilmek istiyorum, böylece gelemeyeceğim randevuyu bildirebilirim.
- **Bir müşteri olarak**, ana randevu ekranından Kuaför/Salon Sahibi giriş ekranına geçiş yapabilmek için bir buton görmek istiyorum, böylece ilgili personel kolayca giriş yapabilir.

Kuaför Rolü

- **Bir kuaför olarak**, kullanıcı adı ve şifremle sisteme giriş yapabilmek istiyorum, böylece kendi panelime erişebilirim.
- **Bir kuaför olarak**, ana panelimde (dashboard) hoş geldin mesajı, bugünkü ve bekleyen randevu sayılarımı hızlıca görmek istiyorum, böylece güncel durum hakkında bilgi sahibi olabilirim.
- **Bir kuaför olarak**, onay bekleyen randevularımı ana panelimde listeleyebilmek ve bu randevuları onaylayabilmek veya reddedebilmek istiyorum, böylece yeni randevu taleplerini yönetebilirim.
- **Bir kuaför olarak**, bugünkü onaylanmış randevularımı ana panelimde listeleyebilmek ve tamamlananları işaretleyebilmek istiyorum, böylece günlük iş akışımı takip edebilirim.
- **Bir kuaför olarak**, tüm randevularımı bir liste veya takvim görünümünde görüntüleyebilmek istiyorum, böylece genel programımı görebilirim.
- **Bir kuaför olarak**, randevularımı durumlarına göre (bekleyen, onaylanmış, tamamlanmış, iptal/reddedilmiş) filtreleyebilmek istiyorum, böylece aradığım randevulara kolayca ulaşabilirim.
- **Bir kuaför olarak**, haftanın her günü için çalışma saatlerimi (başlangıç, bitiş) ayarlayabilmek ve çalışmayacağım günleri pasif hale getirebilmek istiyorum, böylece müşteriler sadece uygun olduğum zamanlarda randevu alabilir.
- **Bir kuaför olarak**, çalışmayacağım özel günleri (izin, tatil vb.) bir takvim üzerinden işaretleyebilmek istiyorum, böylece o günlere randevu alınmasını engelleyebilirim.

- **Bir kuaför olarak,** bana daha önce randevu almış müşterilerin listesini görebilmek istiyorum, böylece müşteri geçmişini takip edebilirim.
- **Bir kuaför olarak,** müşteri listesinde arama yapabilmek (isim veya telefonla) istiyorum, böylece belirli bir müşteriyi kolayca bulabilirim.
- **Bir kuaför olarak,** seçtiğim bir müşterinin randevu geçmişini ve eklediğim notları (varsa) görüntüleyebilmek istiyorum, böylece müşteriye özel hizmet sunabilirim.
- **Bir kuaför olarak,** müşteriler hakkında özel notlar ekleyebilmek/silebilmek istiyorum, böylece önemli bilgileri kaydedebilirim.
- **Bir kuaför olarak,** profil bilgilerimi (ad/soyad, e-posta, telefon) güncelleyebilmek istiyorum, böylece bilgilerim her zaman doğru olur.
- **Bir kuaför olarak,** giriş şifremi değiştirebilmek istiyorum, böylece hesabımın güvenliğini artırabilirim.

Salon Sahibi Rolü

- **Bir salon sahibi olarak,** kullanıcı adı ve şifremle sisteme giriş yapabilmek istiyorum, böylece salon yönetim paneline erişebilirim.
- **Bir salon sahibi olarak,** ana panelimde salonumun adını, genel istatistikleri (bugünkü randevu, toplam kuaför/hizmet sayısı vb.), hızlı erişim butonlarını ve bugünkü randevuların bir özetini görmek istiyorum, böylece salonun genel durumunu hızlıca anlayabilirim.
- **Bir salon sahibi olarak,** salonumda çalışan kuaförleri listeleyebilmek, arayabilmek ve durumlarına göre (aktif/pasif) filtreleyebilmek istiyorum, böylece personelimi yönetebilirim.
- **Bir salon sahibi olarak,** sisteme yeni bir kuaför ekleyebilmek (isim, iletişim bilgileri, kullanıcı adı, şifre) istiyorum, böylece yeni personeli kaydedebilirim.
- **Bir salon sahibi olarak,** mevcut bir kuaförün bilgilerini (isim, iletişim, durum, şifre) düzenleyebilmek istiyorum, böylece personel bilgilerini güncel tutabilirim.
- **Bir salon sahibi olarak,** bir kuaförü sistemden silebilmek (veya pasif yapabilmek) istiyorum, böylece ayrılan personeli yönetebilirim.

- **Bir salon sahibi olarak,** salonda sunulan hizmetleri listeleyebilmek, arayabilmek ve durumlarına göre (aktif/pasif) filtreleyebilmek istiyorum, böylece hizmet portföyümü yönetebilirim.
- **Bir salon sahibi olarak,** salona yeni bir hizmet ekleyebilmek (isim, fiyat, süre, açıklama) istiyorum, böylece yeni hizmetler sunabilirim.
- **Bir salon sahibi olarak,** mevcut bir hizmetin bilgilerini (isim, fiyat, süre, açıklama, durum) düzenleyebilmek istiyorum, böylece hizmet detaylarını güncelleyebilirim.
- **Bir salon sahibi olarak,** bir hizmeti silebilmek (veya pasif yapabilmek) istiyorum, böylece sunulmayan hizmetleri kaldırabilirim.
- **Bir salon sahibi olarak,** salondaki *tüm* randevuları bir liste veya takvim görünümünde görüntüleyebilmek istiyorum, böylece salonun genel yoğunluğunu görebilirim.
- **Bir salon sahibi olarak,** randevuları kuaföre veya durumuna göre filtreleyebilmek istiyorum, böylece belirli randevuları inceleyebilirim.
- **Bir salon sahibi olarak,** manuel olarak yeni bir randevu oluşturabilmek (müşteri bilgileri, kuaför, tarih/saat seçerek) istiyorum, böylece telefonda veya yüz yüze alınan randevuları sisteme girebilirim.
- **Bir salon sahibi olarak,** salondaki herhangi bir randevunun durumunu güncelleyebilmek (onaylama, reddetme, tamamlama, iptal etme) istiyorum, böylece randevu yönetimini sağlayabilirim.
- **Bir salon sahibi olarak,** belirli bir tarih aralığı seçerek detaylı raporlar alabilmek istiyorum, böylece salonumun performansını analiz edebilirim.
- **Bir salon sahibi olarak,** seçtiğim tarih aralığı için toplam geliri, kuaför bazında gelir dağılımını ve hizmet bazında gelir dağılımını görebilmek istiyorum (Gelir Raporu).
- **Bir salon sahibi olarak,** seçtiğim tarih aralığı için toplam randevu sayısını, randevu durumlarına göre dağılımı, kuaför bazında randevu sayısını ve saat/gün bazında randevu yoğunluğunu görebilmek istiyorum (Randevu Raporu).

- **Bir salon sahibi olarak,** seçtiğim tarih aralığı için toplam müşteri sayısını, yeni müşteri sayısını, en çok harcama yapanları ve en sık gelen müşterileri listeleyebilmek istiyorum (Müşteri Raporu).
- **Bir salon sahibi olarak,** seçtiğim tarih aralığı için en popüler hizmetleri ve hizmetlerin kullanım sıklığı dağılımını görebilmek istiyorum (Hizmet Raporu).
- **Bir salon sahibi olarak,** salonumun temel bilgilerini (isim, adres, telefon, e-posta, web sitesi) güncelleyebilmek istiyorum.
- **Bir salon sahibi olarak,** salonun genel randevu ayarlarını (online randevuya izin verme, varsayılan randevu süresi, iptal politikası vb.) yapılandırabilmek istiyorum.
- **Bir salon sahibi olarak,** SMS bildirim ayarlarını (servis aktifliği, API bilgileri, gönderici ID, mesaj şablonları) yapılandırabilmek istiyorum, böylece otomatik bildirimleri yönetebilirim.
- **Bir salon sahibi olarak,** kendi giriş şifremi değiştirebilmek istiyorum.

4. Fonksiyonel Gereksinimler (Functional Requirements)

Genel

- **FR-SYS-01:** Sistem, uygulama açıldığında bir açılış ekranı göstermelidir.
- **FR-SYS-02:** Sistem, kayıtlı Kuaför ve Salon Sahibi kullanıcılarının kullanıcı adı ve şifre ile kimliğini doğrulamalıdır.
- **FR-SYS-03:** Sistem, başarılı kimlik doğrulama sonrası kullanıcıya bir JWT token sağlamalıdır.
- **FR-SYS-04:** Sistem, sonraki yetki gerektiren isteklerde JWT token'ı doğrulamalıdır.
- **FR-SYS-05:** Sistem, Kuaför ve Salon Sahibi kullanıcılarının oturumlarını sonlandırabilmelerini sağlamalıdır (token'ı istemciden silerek).
- **FR-SYS-06:** Sistem, kullanıcı rollerine göre farklı ekranlara ve yetkilere yönlendirme yapmalıdır.

- **FR-SYS-07:** Sistem, hatalı giriş denemelerinde veya API hatalarında kullanıcıya anlaşılır hata mesajları göstermelidir.
- **FR-SYS-08:** Sistem arayüzü, farklı mobil cihaz ekran boyutlarına uyumlu olmalıdır.
- **FR-SYS-09:** Sistem, varsayılan dil olarak Türkçe'yi desteklemelidir.

Müşteri İşlevleri

- **FR-CUS-01:** Sistem, müşterinin ad/soyad, telefon ve e-posta bilgilerini girmesine izin vermelidir.
- **FR-CUS-02:** Sistem, müşterinin isteğe bağlı olarak randevu notu eklemesine izin vermelidir.
- **FR-CUS-03:** Sistem, salonun aktif hizmetlerini isim, fiyat ve süre bilgileriyle listelemelidir.
- **FR-CUS-04:** Sistem, müşterinin bir veya daha fazla hizmet seçmesine izin vermelidir.
- **FR-CUS-05:** Sistem, salonun aktif kuaförlerini listelemelidir.
- **FR-CUS-06:** Sistem, müşterinin bir kuaför seçmesine izin vermelidir.
- **FR-CUS-07:** Sistem, müşterinin ileriye dönük tarihlerden birini seçebileceği bir takvim sunmalıdır.
- **FR-CUS-08:** Sistem, seçilen tarih ve kuaför için (kuaförün çalışma saatleri, tatilleri ve mevcut randevuları dikkate alınarak) uygun saat dilimlerini göstermelidir.
- **FR-CUS-09:** Sistem, müşterinin uygun bir saat dilimi seçmesine izin vermelidir.
- **FR-CUS-10:** Sistem, girilen tüm bilgilerle yeni bir randevu kaydı oluşturmalı ve durumu "pending" (beklemede) olarak ayarlamalıdır.
- **FR-CUS-11:** Sistem, randevu oluşturulduktan sonra müşteriye benzersiz bir randevu kodu ve randevu özetini göstermelidir.
- **FR-CUS-12:** Sistem, müşterinin girdiği randevu koduna göre ilgili randevu detaylarını getirmelidir.

- **FR-CUS-13:** Sistem, müşterinin randevu kodu ile sorguladığı ve durumu "pending" veya "confirmed" olan randevuyu "cancelled" durumuna getirmesine izin vermelidir (salon ayarlarına bağlı iptal süresi kontrolü ile).
- **FR-CUS-14:** Sistem, müşteri randevu ekranında Kuaför/Salon Sahibi giriş ekranına yönlendiren bir buton sağlamalıdır.

Kuaför İşlevleri

- **FR-HDR-01:** Sistem, kuaförün kendi paneline erişimini sağlamalıdır.
- **FR-HDR-02:** Sistem, kuaför panelinde bugünkü ve onay bekleyen randevu sayılarını göstermelidir.
- **FR-HDR-03:** Sistem, kuaföre ait onay bekleyen randevuları listelemelidir.
- **FR-HDR-04:** Sistem, kuaförün onay bekleyen bir randevuyu "confirmed" veya "rejected" durumuna getirmesine izin vermelidir.
- **FR-HDR-05:** Sistem, kuaföre ait bugünkü "confirmed" durumdaki randevuları listelemelidir.
- **FR-HDR-06:** Sistem, kuaförün bugünkü "confirmed" bir randevuyu "completed" durumuna getirmesine izin vermelidir.
- **FR-HDR-07:** Sistem, kuaföre ait tüm randevuları liste veya takvim görünümünde sunmalıdır.
- **FR-HDR-08:** Sistem, kuaförün randevularını durumlarına göre filtrelemesine izin vermelidir.
- **FR-HDR-09:** Sistem, kuaförün haftanın her günü için çalışma durumunu (aktif/pasif) ve başlangıç/bitiş saatlerini ayarlamasına izin vermelidir.
- **FR-HDR-10:** Sistem, kuaförün belirli tarihleri tatil/izin günü olarak işaretlemesine izin vermelidir.
- **FR-HDR-11:** Sistem, kuaföre daha önce hizmet verdiği müşterilerin listesini (isim, telefon vb.) göstermelidir.
- **FR-HDR-12:** Sistem, kuaförün müşteri listesinde isim veya telefona göre arama yapmasına izin vermelidir.
- **FR-HDR-13:** Sistem, seçilen müşterinin randevu geçmişini (tarih, hizmet, durum) göstermelidir.

- **FR-HDR-14:** Sistem, kuaförün belirli bir müşteri için notlar eklemesine, görüntülemesine ve silmesine izin vermelidir.
- **FR-HDR-15:** Sistem, kuaförün kendi profil bilgilerini (ad/soyad, e-posta, telefon) güncellemesine izin vermelidir.
- **FR-HDR-16:** Sistem, kuaförün kendi şifresini değiştirmesine izin vermelidir.

Salon Sahibi İşlevleri

- **FR-OWN-01:** Sistem, salon sahibinin salon yönetim paneline erişimini sağlamalıdır.
- **FR-OWN-02:** Sistem, salon sahibi panelinde salon adı, genel istatistikler (toplam randevu, kuaför, hizmet), hızlı erişim linkleri, bugünkü randevu özeti ve kuaför listesi özeti göstermelidir.
- **FR-OWN-03:** Sistem, salondaki tüm kuaförleri listelemelidir.
- **FR-OWN-04:** Sistem, salon sahibinin kuaförleri isme göre aramasına ve aktif/pasif durumuna göre filtrelemesine izin vermelidir.
- **FR-OWN-05:** Sistem, salon sahibinin yeni kuaför kaydı oluşturmaya izin vermelidir (gerekli tüm bilgilerle).
- **FR-OWN-06:** Sistem, salon sahibinin mevcut kuaförlerin bilgilerini (iletişim, durum, şifre sıfırlama) düzenlemesine izin vermelidir.
- **FR-OWN-07:** Sistem, salon sahibinin bir kuaförü silmesine (veya pasif duruma getirmesine) izin vermelidir.
- **FR-OWN-08:** Sistem, salondaki tüm hizmetleri listelemelidir.
- **FR-OWN-09:** Sistem, salon sahibinin hizmetleri isme göre aramasına ve aktif/pasif durumuna göre filtrelemesine izin vermelidir.
- **FR-OWN-10:** Sistem, salon sahibinin yeni hizmet tanımlamasına izin vermelidir.
- **FR-OWN-11:** Sistem, salon sahibinin mevcut hizmetlerin bilgilerini (isim, fiyat, süre, açıklama, durum) düzenlemesine izin vermelidir.
- **FR-OWN-12:** Sistem, salon sahibinin bir hizmeti silmesine (veya pasif duruma getirmesine) izin vermelidir.

- **FR-OWN-13:** Sistem, salondaki *tüm* randevuları liste veya takvim görünümünde sunmalıdır.
 - **FR-OWN-14:** Sistem, salon sahibinin randevuları kuaföre veya duruma göre filtrelemesine izin vermelidir.
 - **FR-OWN-15:** Sistem, salon sahibinin müşteri, kuaför, tarih/saat ve hizmet seçerek manuel randevu oluşturmaya izin vermelidir.
 - **FR-OWN-16:** Sistem, salon sahibinin salondaki herhangi bir randevunun durumunu (pending, confirmed, completed, cancelled, rejected) değiştirmesine izin vermelidir.
 - **FR-OWN-17:** Sistem, salon sahibinin belirli bir tarih aralığı seçmesine izin vermelidir (Raporlama için).
 - **FR-OWN-18:** Sistem, seçilen tarih aralığı için toplam geliri, kuaför bazında gelirleri ve hizmet bazında gelirleri hesaplayıp göstermelidir.
 - **FR-OWN-19:** Sistem, seçilen tarih aralığı için toplam randevu sayısı, durum dağılımı, kuaför dağılımı ve zaman (saat/gün) dağılımını hesaplayıp göstermelidir.
 - **FR-OWN-20:** Sistem, seçilen tarih aralığı için toplam müşteri sayısı, yeni müşteri sayısı, en çok harcama yapanlar ve en sık gelen müşterileri hesaplayıp göstermelidir.
 - **FR-OWN-21:** Sistem, seçilen tarih aralığı için hizmetlerin kullanım sayılarını ve en popüler hizmetleri hesaplayıp göstermelidir.
 - **FR-OWN-22:** Sistem, salon sahibinin salon bilgilerini (isim, adres, iletişim vb.) düzenlemesine izin vermelidir.
 - **FR-OWN-23:** Sistem, salon sahibinin online randevu ayarları, varsayılan süre, iptal süresi gibi genel salon ayarlarını yapmasına izin vermelidir.
 - **FR-OWN-24:** Sistem, salon sahibinin SMS bildirim servisini aktif/pasif yapmasına, API bilgilerini girmesine ve mesaj şablonlarını düzenlemesine izin vermelidir.
 - **FR-OWN-25:** Sistem, salon sahibinin kendi şifresini değiştirmesine izin vermelidir.
-

Fonksiyonel Olmayan Gereksinimler (Örnekler):

- **Performans:** API yanıt süreleri X milisaniyenin altında olmalıdır. Randevu listeleme ve raporlama işlemleri Y saniyeden uzun sürmemelidir.
- **Güvenlik:** Şifreler güvenli bir şekilde hashlenmelidir. API endpoint'leri yetkilendirme ile korunmalıdır. Veri tabanına SQL injection gibi saldırılara karşı önlem alınmalıdır.
- **Kullanılabilirlik:** Arayüz anlaşılır ve kullanımı kolay olmalıdır. Farklı roller için arayüzler optimize edilmelidir.
- **Ölçeklenebilirlik:** Sistem, artan kullanıcı ve veri yükünü kaldırabilecek şekilde tasarlanmalıdır (örn. veritabanı optimizasyonu, yatay/dikey ölçekleme).
- **Güvenilirlik:** Sistem kesintisiz çalışmalı, veri kaybı yaşanmamalıdır (örn. veritabanı yedekleme).

5. Detaylı Uygulama Akış Dokümanı

1. Giriş

Bu doküman, "Randevu App" mobil uygulamasının farklı kullanıcı rolleri (Müşteri, Kuaför, Salon Sahibi) için temel kullanıcı akışlarını, ekranlar arası geçişleri, önemli kullanıcı etkileşimlerini ve sistem yanıtlarını detaylı bir şekilde açıklamaktadır.

Kullanıcı Roller:

- **Müşteri:** Randevu alan son kullanıcı.
- **Kuaför:** Hizmet veren personel.
- **Salon Sahibi:** Salonu yöneten yönetici.

2. Genel Akışlar

2.1. Uygulama Başlatma

1. **Kullanıcı:** Cihazında uygulama ikonuna dokunur.
2. **Sistem:** Uygulama başlatılır, `SplashScreen` ekranı belirir. Logo ve "Randevu App" başlığı ile birlikte bir yükleniyor göstergesi (CircularProgressIndicator) görüntülenir.

3. **Sistem:** Arka planda `AuthController` 'ı başlatır ve `_loadUserData` fonksiyonu ile cihazın yerel depolamasından (SharedPreferences) kaydedilmiş kullanıcı ID, rol, token gibi bilgileri okumaya çalışır.
4. **Sistem:** Yaklaşık 2 saniye sonra veya veri yükleme tamamlandığında (`_navigateToNextScreen`):
 - **Giriş Bilgisi Bulunursa (Token Geçerli Sayılırsa):** `AuthController` 'daki `userRole` bilgisine göre:
 - Rol "hairstresser" ise → `HairdresserDashboardScreen` 'e yönlendirilir.
 - Rol "salonOwner" ise → `SalonOwnerDashboardScreen` 'e yönlendirilir.
 - Rol "none" veya tanımsız ise (beklenmedik durum) → `CustomerAppointmentBookingScreen` 'e yönlendirilir.
 - **Giriş Bilgisi Bulunmazsa:** `CustomerAppointmentBookingScreen` 'e yönlendirilir.

2.2. Giriş (Kuaför / Salon Sahibi)

1. **Kullanıcı:** `CustomerAppointmentBookingScreen` 'in AppBar'ındaki "Giriş" ikonuna (`Icons.login`) dokunur.
2. **Sistem:** `LoginScreen` (Giriş Ekranı) yüklenir. Logo, başlık, kullanıcı adı ve şifre alanları, "Giriş Yap" butonu ve "Müşteri Randevu Sayfasına Dön" linki gösterilir.
3. **Kullanıcı:** "Kullanıcı Adı" ve "Şifre" alanlarına bilgilerini yazar. Şifre alanındaki göz ikonuna basarak şifrenin görünürlüğünü değiştirebilir.
4. **Kullanıcı:** "Giriş Yap" (`CustomButton`) butonuna dokunur.
5. **Sistem:** Buton üzerinde yükleniyor göstergesi belirir. `_formKey.currentState.validate()` ile alanların boş olup olmadığını kontrol eder.
 - **Doğrulama Başarısız:** İlgili alanın altında hata mesajı gösterir (örn. "Lütfen kullanıcı adınızı girin"). Yükleniyor göstergesi kaybolur.
 - **Doğrulama Başarılı:** `AuthController.login(username, password)` fonksiyonunu çağırır.
 - **API İsteği Başarılı ve Kullanıcı Doğrulandı:** `AuthController` kullanıcı bilgilerini (ID, rol, token, salon ID) günceller ve SharedPreferences'a kaydeder. `isLoading` false olur. Kullanıcı rolüne göre:

- Rol "hairstresser" → `Get.offAllNamed(AppRoutes.hairstresserDashboard)` ile Kuaför Ana Paneline yönlendirilir.
 - Rol "salonOwner" → `Get.offAllNamed(AppRoutes.salonOwnerDashboard)` ile Salon Sahibi Ana Paneline yönlendirilir.
 - Rol "none" veya tanımsız → Hata Snackbar'ı gösterir ("Giriş başarılı ancak rol belirlenemedi").
 - **API İsteği Başarısız veya Kullanıcı Doğrulanamadı:** `isLoading` false olur. Hata Snackbar'ı gösterir ("Kullanıcı adı veya şifre hatalı"). Kullanıcı `LoginScreen` 'de kalır.
6. **Kullanıcı (Alternatif):** "Müşteri Randevu Sayfasına Dön" (`TextButton`) linkine dokunur.
7. **Sistem:** `Get.offAllNamed(AppRoutes.customerAppointmentBooking)` ile Müşteri Randevu Alma Ekranına yönlendirir.

2.3. Çıkış (Kuaför / Salon Sahibi)

1. **Kullanıcı:** İlgili Dashboard ekranında yan menüyü (Drawer) açar veya `HairdresserSettingsScreen` / `SalonOwnerSettingsScreen` 'e gider.
2. **Kullanıcı:** "Çıkış Yap" (`ListTile`) seçeneğine dokunur.
3. **Sistem:** Onay iletişim kutusu (AlertDialog: "Çıkış yapmak istediğinize emin misiniz?") gösterir.
4. **Kullanıcı:** "Vazgeç" veya "Çıkış Yap" butonuna dokunur.
5. **Sistem:**
 - **Vazgeç:** İletişim kutusunu kapatır.
 - **Çıkış Yap:** `AuthController.logout` fonksiyonunu çağırır. SharedPreferences'daki kullanıcı verileri silinir, Controller'daki durum sıfırlanır, API token'ı temizlenir.
 - **Sistem:** `Get.offAllNamed(AppRoutes.customerAppointmentBooking)` ile Müşteri Randevu Alma Ekranına yönlendirir (veya `AppRoutes.login` 'e de yönlendirebilir, koddaki son duruma göre `customerAppointmentBooking` 'e gidiyor).

3. Müşteri Akışları

3.1. Randevu Alma (Detaylı)

1. **Ekrani:** `CustomerAppointmentBookingScreen`

2. **Sistem:** Ekran yüklenirken `_loadInitialData` ile aktif kuaför ve hizmet listelerini API'den çeker (`HairdresserController` , `ServiceController`). Yüklenirken ilgili listelerin olduğu yerlerde yükleniyor göstergesi belirebilir. Adım göstergesi (StepIndicator) Adım 1'i aktif gösterir.

3. Adım 1: Müşteri Bilgileri

- **Kullanıcı:** Ad/Soyad, Telefon, E-posta alanlarını (`CustomTextField`) doldurur. (Opsiyonel) Not alanını doldurur.
- **Sistem:** Girdi sırasında formatlama/validasyon ipuçları verebilir.
- **Kullanıcı:** "İleri" (`CustomButton`) butonuna dokunur.
- **Sistem:** Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hatalı alanların altında mesajlar gösterir.
 - **Hata Yoksa:** Bilgileri `AppointmentController` 'a kaydeder. Adım 1 göstergesini tamamlandı yapar, Adım 2'yi aktif hale getirir. Ekranda Adım 2 içeriği (`_buildServiceSelectionStep`) gösterilir.

4. Adım 2: Hizmet Seçimi

- **Sistem:** Yüklenen aktif hizmetleri kartlar (`Card`) halinde listeler. Her kartta hizmet adı, açıklama (varsa), fiyat ve süre bilgisi bulunur. Başlangıçta hiçbir hizmet seçili değildir.
- **Kullanıcı:** İsteddiği hizmet kartlarına dokunur.
- **Sistem:** Dokunulan kartın kenarlık rengini değiştirir ve/veya bir 'check' ikonu gösterir. İlgili hizmeti `AppointmentController.selectedServices` listesine ekler veya çıkarır.
- **Kullanıcı:** Seçimini tamamlayınca "İleri" butonuna dokunur.
- **Sistem:** `AppointmentController.selectedServices` listesinin boş olup olmadığını kontrol eder.
 - **Boşsa:** Snackbar ile uyarı verir ("Lütfen en az bir hizmet seçin").

- **Doluysa:** Adım 2 göstergesini tamamlandı yapar, Adım 3'ü aktif hale getirir. Ekranda Adım 3 içeriği (`_buildHairdresserSelectionStep`) gösterilir.

5. Adım 3: Kuaför Seçimi

- **Sistem:** Yüklenen aktif kuaförleri kartlar halinde listeler (resim/placeholder, isim, e-posta). Başlangıçta kuaför seçili değildir.
- **Kullanıcı:** İsteddiği kuaför kartına dokunur.
- **Sistem:** Dokunulan kartın kenarlık rengini değiştirir ve/veya bir 'check' ikonu gösterir. Seçilen kuaförü `AppointmentController.selectedHairdresser.value` 'a atar.
- **Kullanıcı:** "İleri" butonuna dokunur.
- **Sistem:** `AppointmentController.selectedHairdresser.value` 'nun null olup olmadığını kontrol eder.
 - **Null ise:** Snackbar ile uyarı verir ("Lütfen bir kuaför seçin").
 - **Null Değilse:** Adım 3 göstergesini tamamlandı yapar, Adım 4'ü aktif hale getirir. Ekranda Adım 4 içeriği (`_buildDateTimeSelectionStep`) gösterilir.

6. Adım 4: Tarih ve Saat Seçimi

- **Sistem:** `TableCalendar` widget'ını gösterir. Geçerli günü ve seçili günü (`AppointmentController.selectedDate.value` , başlangıçta bugündür) vurgular. Uygun saat dilimlerini (`availableTimeSlots` - şimdilik statik, idealde seçilen kuaför ve güne göre API'den gelmeli) butonlar/chip'ler halinde listeler.
- **Kullanıcı:** Takvimden farklı bir güne dokunur.
- **Sistem:** Seçilen günü günceller, takvimde vurgular. Saat seçimini sıfırlar ve o güne ait uygun saatleri (varsa) gösterir.
- **Kullanıcı:** Listelenen saat dilimlerinden birine dokunur.
- **Sistem:** Seçilen saat dilimini (`selectedTimeSlot.value`) günceller ve görsel olarak vurgular.
- **Kullanıcı:** "Tamamla" butonuna dokunur.
- **Sistem:** `selectedTimeSlot.value` 'nun boş olup olmadığını kontrol eder.
 - **Boşsa:** Snackbar ile uyarı verir ("Lütfen bir saat seçin").

- **Doluysa:** Adım 4 göstergesini tamamladı yapar. `_completeBooking` fonksiyonunu tetikler.
 - Saat bilgisini `DateTime` nesnesine çevirir ve `AppointmentController.selectedTime.value` 'a atar.
 - "Tamamla" butonunda yükleniyor göstergesi belirir.
 - `AppointmentController.createAppointment` çağrılır. API'ye randevu oluşturma isteği (müşteri bilgileri, kuaför ID, hizmet ID'leri, tarih/saat) gönderilir.
 - **API Başarılı Yanıt Verirse:** Yükleniyor göstergesi kaybolur. `AppointmentController` 'daki form verileri sıfırlanır. Dönen randevu bilgileri (özellikle `appointment_code`) ile `Get.toNamed(AppRoutes.customerAppointmentConfirmation, arguments: appointmentData)` çağrılır.
 - **API Başarısız Yanıt Verirse:** Yükleniyor göstergesi kaybolur. Snackbar ile API'den gelen veya genel bir hata mesajı gösterilir. Kullanıcı Adım 4'te kalır.

7. Ekran: `CustomerAppointmentConfirmationScreen`

- **Sistem:** Başarı ikonu, onay mesajı, randevu kodu ve randevu özet bilgilerini (Müşteri, Kuaför, Tarih, Saat, Hizmetler) gösterir.
- **Kullanıcı:** Randevu koduna dokunur.
- **Sistem:** Kodu panoya kopyalar ve Snackbar ile bilgi verir ("Randevu kodu panoya kopyalandı").
- **Kullanıcı:** "Randevu Yönetimi" butonuna dokunur.
- **Sistem:** `Get.offAllNamed(AppRoutes.customerAppointmentManage)` ile Randevu Yönetim Ekranına yönlendirir.
- **Kullanıcı:** "Bitir" butonuna dokunur.
- **Sistem:** `Get.offAllNamed(AppRoutes.customerAppointmentBooking)` ile Müşteri Randevu Alma Ekranının ilk adımına yönlendirir.

3.2. Randevu Yönetme (Kod ile - Detaylı)

1. Ekran: `CustomerAppointmentBookingScreen`

2. **Kullanıcı:** AppBar'daki "Randevu Düzenle" (`Icons.edit_calendar`) ikonuna dokunur.
3. **Sistem:** `Get.toNamed(AppRoutes.customerAppointmentManage)` ile `CustomerAppointmentManageScreen` 'e yönlendirir. Ekranda randevu kodu giriş alanı ve "Randevuyu Sorgula" butonu gösterilir.
4. **Kullanıcı:** Aldığı randevu kodunu ilgili alana (`_codeController`) yazar.
5. **Kullanıcı:** "Randevuyu Sorgula" (`CustomButton`) butonuna dokunur.
6. **Sistem:** Butonda yükleniyor göstergesi belirir. `_formKey.currentState.validate()` ile alanın boş olup olmadığını kontrol eder. Boş değilse, `AppointmentController.loadAppointmentByCode(code)` fonksiyonunu çağırır. API'ye istek gönderilir.
 - **API Başarılı Yanıt Verirse (Kod Geçerli):** Yükleniyor göstergesi kaybolur. `isCodeSubmitted.value` true olur. Kod giriş bölümü gizlenir, `_buildAppointmentDetailsSection` ile randevu detayları (Durum, Kod, Tarih, Saat, Kuaför, Hizmetler, Not) gösterilir. Randevunun durumu "pending" veya "confirmed" ise "Randevuyu İptal Et" butonu görünür olur.
 - **API Başarısız Yanıt Verirse (Kod Geçersiz veya Hata):** Yükleniyor göstergesi kaybolur. Snackbar ile hata mesajı gösterilir ("Randevu bulunamadı..."). Kullanıcı kod giriş bölümünde kalır.
7. **Kullanıcı (Randevu Detayları Gösteriliyorsa):**
 - Bilgileri inceler.
 - (Opsiyonel) "Randevuyu İptal Et" (`CustomButton` , type: outlined) butonuna dokunur.
8. **Sistem (İptal Tıklandıysa):**
 - Onay iletişim kutusu (AlertDialog: "Randevunuzu iptal etmek istediğinize emin misiniz?") gösterir.
 - **Kullanıcı:** "Vazgeç" veya "İptal Et" butonuna dokunur.
 - **Sistem (İptal Et Seçildiyse):** İletişim kutusunu kapatır. "Randevuyu İptal Et" butonunda yükleniyor göstergesi belirir. `AppointmentController.updateAppointmentStatus(appointment.id!, AppointmentStatus.cancelled)` fonksiyonunu çağırır. API isteği gönderilir.

- **API Başarılı Yanıt Verirse:** Yükleniyor göstergesi kaybolur. Snackbar ile başarı mesajı gösterir ("Randevunuz iptal edildi").

`Get.offAllNamed(AppRoutes.customerAppointmentBooking)` ile ana randevu alma ekranına yönlendirir.

- **API Başarısız Yanıt Verirse:** Yükleniyor göstergesi kaybolur. Snackbar ile hata mesajı gösterir ("Randevu iptal edilirken bir hata oluştu"). Kullanıcı randevu detayları ekranında kalır.

- **Sistem (Vazgeç Seçildiyse):** İletişim kutusunu kapatır.

4. Kuaför Akışları (Detaylı)

4.1. Giriş ve Ana Panel (Kuaför)

1. **Kullanıcı:** Uygulamayı açar, `LoginScreen` 'e ulaşır (Bkz. Genel Akış 2.2).
2. **Kullanıcı:** Kendi kuaför kullanıcı adı ve şifresini girer, "Giriş Yap" butonuna dokunur.
3. **Sistem:** `AuthController.login` ile giriş işlemini yapar.
 - **Başarılı:** Kullanıcı rolü "hairdresser" olarak belirlenir.
`Get.offAllNamed(AppRoutes.hairdresserDashboard)` ile Kuaför Ana Paneline yönlendirilir. Token ve kullanıcı bilgileri kaydedilir.
 - **Başarısız:** Hata mesajı gösterilir.
4. **Ekran:** `HairdresserDashboardScreen` (Kuaför Ana Panel)
5. **Sistem:** Ekran yüklenirken `_loadInitialData` fonksiyonu çalışır:
 - `AppointmentController.loadAppointments` ile kuaföre ait randevuları API'den çeker.
 - `HairdresserController.loadCurrentHairdresser` ile giriş yapan kuaförün detaylı bilgilerini (isim vb.) API'den çeker.
 - Yüklenirken ekranın ilgili bölümlerinde yükleniyor göstergeleri belirebilir.
6. **Sistem:** Veriler yüklendikten sonra:
 - Hoş geldin mesajını kuaförün adıyla gösterir.
 - İstatistik kartlarını günceller (Bugünkü randevu sayısı, Bekleyen randevu sayısı).

- "Onay Bekleyen Randevular" bölümünde `AppointmentController.pendingAppointments` listesini kartlar halinde (`_buildAppointmentCard`) gösterir. Liste boşsa "Bekleyen randevu bulunmuyor" mesajı gösterir.
- "Bugünkü Randevular" bölümünde `AppointmentController.confirmedAppointments` listesinden sadece o güne ait olanları filtreleyerek (`isSameDay`) kartlar halinde gösterir. Liste boşsa "Bugün için randevu bulunmuyor" mesajı gösterir.

7. Kullanıcı:

- Ekranı aşağı kaydırarak listeleri yenileyebilir (RefreshIndicator).
- Sol üstteki menü ikonuna (veya kenardan kaydırarak) dokunarak yan menüyü (Drawer - `_buildDrawer`) açar.
- Yan menüden "Randevularım", "Çalışma Saatlerim", "Müşterilerim", "Ayarlar" veya "Çıkış Yap" seçeneklerine dokunarak ilgili ekranlara veya işlemlere gider.
- Sağ üstteki Ayarlar ikonuna (`Icons.settings`) dokunarak `HairdresserSettingsScreen` 'e gider.
- Onay bekleyen bir randevu kartındaki "Onayla" veya "Reddet" butonuna dokunur (Bkz. Akış 4.2).
- Bugünkü bir randevu kartındaki "Tamamla" butonuna dokunur (Bkz. Akış 4.2).

4.2. Randevu Yönetimi (Kuaför)

1. **Kullanıcı:** Yan menüden "Randevularım" seçeneğine dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.hairdresserAppointments)` ile `HairdresserAppointmentsScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadAppointments` ile güncel randevu listesini çeker (eğer zaten güncel değilse). Varsayılan olarak "Liste" tabı ve "Tümü" filtresi seçilidir.

4. Liste Görünümü:

- **Kullanıcı:** Farklı durum filtrelerine (Bekleyen, Onaylanan vb. - `_buildFilterChip`) dokunur.

- **Sistem:** Seçilen filtreye göre `_getFilteredAppointments` ile randevu listesini (`_buildAppointmentCard`) günceller. Liste boşsa "Randevu bulunamadı" mesajı gösterir.
- **Kullanıcı:** Bir randevu kartındaki ilgili butonlara dokunur:
 - **Bekleyen:** "Onayla" veya "Reddet".
 - **Onaylanmış:** "Tamamla".
- **Sistem:** İlgili fonksiyonu (`_confirmAppointment` , `_rejectAppointment` , `_completeAppointment`) çağırır.
 - Onay iletişim kutusu gösterir (Reddetme durumunda).
 - Kullanıcı onaylarsa, `AppointmentController.updateAppointmentStatus` ile API'ye istek gönderir. Butonda yükleniyor göstergesi belirebilir.
 - **API Başarılı:** Başarı mesajı (Snackbar) gösterir ve randevu listesini günceller.
 - **API Başarısız:** Hata mesajı (Snackbar) gösterir.

5. Takvim Görünümü:

- **Kullanıcı:** "Takvim" tabına dokunur.
- **Sistem:** `TableCalendar` widget'ını gösterir. Randevusu olan günler altında işaretleyici (marker) gösterir. Varsayılan olarak bugünün tarihi seçilidir (`_selectedDay`). Seçili güne ait randevuları aşağıda listeler.
- **Kullanıcı:** Takvimde farklı bir güne dokunur.
- **Sistem:** Seçilen günü (`_selectedDay`) günceller ve takvimde vurgular. Aşağıdaki listeyi o güne ait randevularla günceller. Liste boşsa "Bu tarihte randevu bulunmuyor" mesajı gösterir.
- **Kullanıcı:** Listelenen randevu kartlarındaki butonlara dokunarak işlem yapabilir (Liste görünümündeki gibi).

6. **Kullanıcı:** Ekranı aşağı kaydırarak randevu listesini yenileyebilir (RefreshIndicator).

7. **Kullanıcı:** Sağ alttaki Yenileme butonuna (FloatingActionButton) dokunur.

8. **Sistem:** `_loadAppointments` ile randevu listesini yeniden yükler.

4.3. Çalışma Takvimi Yönetimi

1. **Kullanıcı:** Yan menüden "Çalışma Saatlerim" seçeneğine dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.hairdresserSchedule)` ile `HairdresserScheduleScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadHairdresserData` ile kuaförün mevcut çalışma saatlerini ve tatil günlerini yükler. Varsayılan olarak "Çalışma Saatleri" tabı seçilidir.

4. Çalışma Saatleri Tabı:

- **Sistem:** Haftanın 7 günü için ayrı kartlar (`_buildDayScheduleCard`) gösterir. Her kartta gün adı, aktif/pasif anahtarı (Switch) ve aktifse başlangıç/bitiş saatleri gösterilir.
- **Kullanıcı:** Bir günün anahtarına dokunarak o günü aktif/pasif yapar.
- **Sistem:** Anahtarın durumu değişir. Pasif yapılırsa saat seçme alanları gizlenir.
- **Kullanıcı:** Aktif bir günün başlangıç veya bitiş saatine dokunur.
- **Sistem:** `_selectTime` fonksiyonunu çağırır ve cihazın standart saat seçme arayüzünü (TimePicker) gösterir.
- **Kullanıcı:** Yeni bir saat seçer ve onaylar.
- **Sistem:** Seçilen saati ilgili alanda günceller (`_workingSchedule[day]`).
- **Kullanıcı:** Tüm değişiklikleri yaptıktan sonra "Çalışma Saatlerini Kaydet" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. `_updateWorkingHours` fonksiyonunu çağırır. `HairdresserController.updateWorkingHours` ile API'ye istek gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Başarı mesajı (Snackbar) gösterir.
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir.

5. Tatil Günleri Tabı:

- **Kullanıcı:** "Tatil Günleri" tabına dokunur.
- **Sistem:** `TableCalendar` widget'ını gösterir. Kuaförün önceden kaydettiği tatil günleri takvimde seçili olarak işaretlenmiştir (`_selectedDays`).
- **Kullanıcı:** Takvimde bir güne dokunur.
- **Sistem:** Eğer gün zaten seçiliyse seçimi kaldırır, seçili değilse seçili hale getirir ve `_selectedDays` setini günceller. Seçili gün sayısını günceller.
- **Kullanıcı:** Seçimlerini tamamladıktan sonra "Tatil Günlerini Kaydet" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi (`_isHolidayUpdating`) belirir. `_updateHolidayDates` fonksiyonunu çağırır. `HairdresserController.updateHolidayDates` ile API'ye istek gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Başarı mesajı (Snackbar) gösterir.
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir.

4.4. Müşteri Görüntüleme (Kuaför)

1. **Kullanıcı:** Yan menüden "Müşterilerim" seçeneğine dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.hairdresserCustomers)` ile `HairdresserCustomersScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadCustomers` (aslında `AppointmentController.loadAppointments`) ile randevuları çeker ve `_groupCustomers` ile müşterileri gruplar.
4. **Sistem:** Müşterileri kartlar (`_buildCustomerCard`) halinde listeler. Her kartta müşteri adı, iletişim bilgileri, toplam/tamamlanan randevu sayısı ve son randevu tarihi gösterilir.
5. **Kullanıcı:** Arama çubuğuna (`_searchController`) müşteri adı veya telefon numarası yazar.
6. **Sistem:** Yazılan sorguya göre `_getFilteredCustomers` ile listeyi anlık olarak filtreler ve günceller. Arama çubuğundaki 'X' ikonuna basılırsa filtre temizlenir.
7. **Kullanıcı:** Bir müşteri kartındaki "Detayları Göster" (`TextButton`) butonuna dokunur.

8. **Sistem:** `_showCustomerDetails` fonksiyonunu çağırarak ekranın altından bir modal sayfa (BottomSheet) açar. Bu modalda "Randevu Geçmişi" ve "Müşteri Notları" tablaları bulunur.

9. Randevu Geçmişi Tabı:

- **Sistem:** Seçilen müşteriye ait tüm randevuları tarihe göre (en yeniden eskiye) sıralayarak listeler (`_buildAppointmentHistoryItem`). Her öğede tarih, saat, durum ve hizmetler gösterilir.

10. Müşteri Notları Tabı:

- **Sistem:** Müşteri için daha önce kaydedilmiş notları (tarih ve içerik bilgisiyle) listeler. "Yeni Not Ekle" bölümünü gösterir.
- **Kullanıcı:** Yeni not alanına (`TextField`) bir not yazar.
- **Kullanıcı:** "Not Ekle" (`ElevatedButton`) butonuna dokunur.
- **Sistem:** (Backend entegrasyonu varsayılarak) Notu API'ye gönderir. Başarılı olursa not listesini günceller, not alanını temizler ve başarı mesajı (Snackbar) gösterir. Başarısız olursa hata mesajı gösterir.
- **Kullanıcı:** Mevcut bir notun yanındaki silme ikonuna (`IconButton`) dokunur.
- **Sistem:** (Backend entegrasyonu varsayılarak) Silme onayı isteyebilir. Onaylanırsa notu API aracılığıyla siler, listeyi günceller ve mesaj gösterir.

11. **Kullanıcı:** Modal sayfadaki kapatma ikonuna (`IconButton`) dokunur.

12. **Sistem:** Modal sayfayı (BottomSheet) kapatır. Kullanıcı `HairdresserCustomersScreen` 'e döner.

4.5. Kuaför Ayarları

1. **Kullanıcı:** Yan menüden "Ayarlar" seçeneğine veya Ana Paneldeki Ayarlar ikonuna dokunur.

2. **Sistem:** `Get.toNamed(AppRoutes.hairdresserSettings)` ile `HairdresserSettingsScreen` 'e yönlendirir.

3. **Sistem:** Ekran yüklenirken `_loadHairdresserData` ile kuaförün mevcut profil bilgilerini yükler ve ilgili alanlara (`_nameController` , `_emailController` , `_phoneController`) yerleştirir.

4. Profil Bilgileri Güncelleme:

- **Kullanıcı:** Ad/Soyad, E-posta, Telefon alanlarındaki bilgileri değiştirir.

- **Kullanıcı:** "Profili Güncelle" (`CustomButton`) butonuna dokunur.
- **Sistem:** `_formKey.currentState.validate()` ile validasyon yapar.
 - **Hata Varsa:** Hatalı alanların altında mesaj gösterir.
 - **Hata Yoksa:** `_updateProfile` fonksiyonunu çağırır.
`HairdresserController.updateHairdresser` ile API'ye güncellenmiş bilgileri gönderir. Butonda yükleniyor göstergesi belirir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Başarı mesajı (Snackbar) gösterir.
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir.

5. Şifre Değiştirme:

- **Kullanıcı:** "Şifre Değiştir" yazısına dokunur.
- **Sistem:** Yeni şifre ve şifre tekrarı alanlarını (`CustomTextField`) görünür yapar (`_showChangePassword.value = true`).
- **Kullanıcı:** Yeni şifreyi ve tekrarını girer. Şifre görünürlük ikonlarını kullanabilir.
- **Kullanıcı:** "Profili Güncelle" butonuna dokunur.
- **Sistem:** Form validasyonunu (şifre alanları dahil) yapar.
 - **Hata Yoksa:** `_updateProfile` fonksiyonu içinde şifre alanlarının dolu olup olmadığını kontrol eder. Doluysa, güncellenecek verilere `password` alanını ekler ve API'ye gönderir.
 - **API Başarılı:** Başarı mesajı gösterir, şifre alanlarını temizler ve şifre değiştirme bölümünü gizler.
 - **API Başarısız:** Hata mesajı gösterir.

6. Diğer Ayarlar:

- **Kullanıcı:** "Çalışma Saatleri" veya "Müşterilerim" `ListTile` öğelerine dokunur.
- **Sistem:** İlgili ekranlara (`HairdresserScheduleScreen` veya `HairdresserCustomersScreen`) yönlendirir.

- **Kullanıcı:** "Çıkış Yap" `ListTile` öğesine dokunur (Bkz. Genel Akış 2.3).

5. Salon Sahibi Akışları (Detaylı)

5.1. Giriş ve Ana Panel (Salon Sahibi)

1. **Kullanıcı:** Uygulamayı açar, `LoginScreen` 'e ulaşır (Bkz. Genel Akış 2.2).
2. **Kullanıcı:** Kendi salon sahibi kullanıcı adı ve şifresini girer, "Giriş Yap" butonuna dokunur.
3. **Sistem:** `AuthController.login` ile giriş işlemini yapar.
 - **Başarılı:** Kullanıcı rolü "salonOwner" olarak belirlenir. `Get.offAllNamed(AppRoutes.salonOwnerDashboard)` ile Salon Sahibi Ana Paneline yönlendirilir. Token ve kullanıcı bilgileri kaydedilir.
 - **Başarısız:** Hata mesajı gösterilir.
4. **Ekrani:** `SalonOwnerDashboardScreen` (Salon Sahibi Ana Panel)
5. **Sistem:** Ekran yüklenirken `_loadInitialData` fonksiyonu çalışır:
 - `SalonController.loadSalon` ile salonun genel bilgilerini API'den çeker.
 - `HairdresserController.loadHairdressers` ile salondaki tüm kuaförleri çeker.
 - `ServiceController.loadServices` ile salondaki tüm hizmetleri çeker.
 - `AppointmentController.loadAppointments` ile salondaki tüm randevuları çeker.
 - Yüklenirken ekranın ilgili bölümlerinde yükleniyor göstergeleri belirebilir.
6. **Sistem:** Veriler yüklendikten sonra:
 - Hoş geldin mesajını salon adıyla gösterir (`_salonController.salon.value.name`).
 - İstatistik kartlarını günceller (`_buildStatisticsCards` - Bugünkü Randevu, Toplam Randevu, Aktif Kuaför, Hizmetler).
 - Hızlı Erişim kartlarını gösterir (`_buildQuickAccessCards`).
 - "Bugünkü Randevular" bölümünde o güne ait randevuların özetini (ilk 5) listeler (`_buildTodayAppointments`). Liste boşsa bilgi mesajı gösterir.
 - "Kuaförler" bölümünde kuaförlerin özetini (ilk 3) listeler (`_buildHairdressers`). Liste boşsa bilgi mesajı gösterir.

7. Kullanıcı:

- Ekranı aşağı kaydırarak verileri yenileyebilir (RefreshIndicator).
- Sol üstteki menü ikonuna (veya kenardan kaydırarak) dokunarak yan menüyü (Drawer - `_buildDrawer`) açar.
- Yan menüden "Kuaförler", "Hizmetler", "Randevular", "Raporlar", "Ayarlar" veya "Çıkış Yap" seçeneklerine dokunarak ilgili ekranlara veya işlemlere gider.
- Sağ üstteki Ayarlar ikonuna (`Icons.settings`) dokunarak `SalonOwnerSettingsScreen` 'e gider.
- Hızlı Erişim kartlarına dokunarak ilgili yönetim ekranlarına gider.
- Bugünkü randevu veya kuaför listesindeki "Tümünü Gör" (varsa) linklerine dokunarak ilgili detay ekranlarına gider.

5.2. Kuaför Yönetimi (Salon Sahibi)

1. **Kullanıcı:** Yan menüden "Kuaförler" seçeneğine veya Ana Paneldeki ilgili Hızlı Erişim kartına dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.salonOwnerHairdressers)` ile `SalonOwnerHairdressersScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadData` ile güncel kuaför listesini API'den çeker. Varsayılan olarak "Sadece aktif kuaförleri göster" filtresi seçilidir. Kuaförleri kartlar (`_buildHairdresserCard`) halinde listeler.
4. **Listeleme ve Filtreleme:**
 - **Kullanıcı:** Arama çubuğuna (`_searchController`) kuaför adı yazar.
 - **Sistem:** Yazılan sorguya göre listeyi anlık olarak filtreler.
 - **Kullanıcı:** "Sadece aktif kuaförleri göster" (`Checkbox`) seçeneğinin işaretini değiştirir.
 - **Sistem:** Seçime göre listeyi aktif veya tüm kuaförleri gösterecek şekilde günceller.
5. **Yeni Kuaför Ekleme:**
 - **Kullanıcı:** Sağ alttaki Ekle butonuna (FloatingActionButton `+`) dokunur.

- **Sistem:** `_showAddHairdresserModal` fonksiyonunu çağırarak ekranın altından modal sayfa (BottomSheet) açar. Gerekli alanlar (Ad/Soyad, E-posta, Telefon, Kullanıcı Adı, Şifre) gösterilir.
- **Kullanıcı:** Form alanlarını doldurur.
- **Kullanıcı:** "Ekle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hatalı alanların altında mesajlar gösterir. Yükleniyor göstergesi kaybolur.
 - **Hata Yoksa:** `HairdresserController.createHairdresser` ile API'ye yeni kuaför oluşturma isteği gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Modal sayfayı kapatır. Başarı mesajı (Snackbar) gösterir. Kuaför listesini yeniler (`_loadData`).
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir. Kullanıcı modal sayfada kalır.
- **Kullanıcı (Alternatif):** "İptal" butonuna dokunur.
- **Sistem:** Modal sayfayı kapatır.

6. Kuaför Düzenleme:

- **Kullanıcı:** Listedeki bir kuaförün kartındaki "Düzenle" (`ElevatedButton`) butonuna dokunur.
- **Sistem:** `_showEditHairdresserModal` fonksiyonunu çağırarak seçilen kuaförün bilgileriyle doldurulmuş modal sayfayı açar (Ad/Soyad, E-posta, Telefon, Yeni Şifre [opsiyonel], Aktif/Pasif Durum anahtarı).
- **Kullanıcı:** Bilgileri (veya durumu) değiştirir. (Opsiyonel) Yeni şifre girer.
- **Kullanıcı:** "Güncelle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hatalı alanların altında mesajlar gösterir. Yükleniyor göstergesi kaybolur.

- **Hata Yoksa:** `HairdresserController.updateHairdresser` ile API'ye kuaför güncelleme isteği gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Modal sayfayı kapatır. Başarı mesajı (Snackbar) gösterir. Kuaför listesini yeniler.
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir. Kullanıcı modal sayfada kalır.
- **Kullanıcı (Alternatif):** "İptal" butonuna dokunur.
- **Sistem:** Modal sayfayı kapatır.

7. Kuaför Silme:

- **Kullanıcı:** Listedden bir kuaförün kartındaki "Sil" (`TextButton`) butonuna dokunur.
- **Sistem:** Onay iletişim kutusu (AlertDialog: "...silme istediğinize emin misiniz?") gösterir.
- **Kullanıcı:** "İptal" veya "Sil" butonuna dokunur.
- **Sistem (Sil Seçildiyse):** İletişim kutusunu kapatır. `_deleteHairdresser` fonksiyonunu çağırır. `HairdresserController.deleteHairdresser` ile API'ye silme isteği gönderir. Yükleniyor göstergesi belirebilir.
 - **API Başarılı:** Başarı mesajı (Snackbar) gösterir. Kuaför listesini yeniler.
 - **API Başarısız:** Hata mesajı (Snackbar) gösterir.
- **Sistem (İptal Seçildiyse):** İletişim kutusunu kapatır.

5.3. Hizmet Yönetimi

1. **Kullanıcı:** Yan menüden "Hizmetler" seçeneğine veya Ana Paneldeki ilgili Hızlı Erişim kartına dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.salonOwnerServices)` ile `SalonOwnerServicesScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadData` ile güncel hizmet listesini API'den çeker. Varsayılan olarak "Sadece aktif hizmetleri göster" filtresi seçilidir. Hizmetleri kartlar (`_buildServiceCard`) halinde listeler.
4. **Listeleme ve Filtreleme:**

- **Kullanıcı:** Arama çubuğuna (`_searchController`) hizmet adı yazar.
- **Sistem:** Yazılan sorguya göre listeyi anlık olarak filtreler.
- **Kullanıcı:** "Sadece aktif hizmetleri göster" (`Checkbox`) seçeneğinin işaretini değiştirir.
- **Sistem:** Seçime göre listeyi aktif veya tüm hizmetleri gösterecek şekilde günceller.

5. Yeni Hizmet Ekleme:

- **Kullanıcı:** Sağ alttaki Ekle butonuna (FloatingActionButton `+`) dokunur.
- **Sistem:** `_showAddServiceModal` fonksiyonunu çağırarak modal sayfayı açar (Ad, Fiyat, Süre, Açıklama [opsiyonel]).
- **Kullanıcı:** Form alanlarını doldurur.
- **Kullanıcı:** "Ekle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hatalı alanların altında mesajlar gösterir. Yükleniyor göstergesi kaybolur.
 - **Hata Yoksa:** `ServiceController.createService` ile API'ye istek gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Modal sayfayı kapatır. Başarı mesajı (Snackbar) gösterir. Hizmet listesini yeniler (`_loadData`).
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir.
- **Kullanıcı (Alternatif):** "İptal" butonuna dokunur.
- **Sistem:** Modal sayfayı kapatır.

6. Hizmet Düzenleme:

- **Kullanıcı:** Listedden bir hizmetin kartındaki "Düzenle" (`ElevatedButton`) butonuna dokunur.

- **Sistem:** `_showEditServiceModal` fonksiyonunu çağırarak seçilen hizmetin bilgileriyle doldurulmuş modal sayfayı açar (Ad, Fiyat, Süre, Açıklama, Aktif/Pasif Durum anahtarı).
- **Kullanıcı:** Bilgileri (veya durumu) değiştirir.
- **Kullanıcı:** "Güncelle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hatalı alanların altında mesajlar gösterir. Yükleniyor göstergesi kaybolur.
 - **Hata Yoksa:** `ServiceController.updateService` ile API'ye güncelleme isteği gönderir.
 - **API Başarılı:** Yükleniyor göstergesi kaybolur. Modal sayfayı kapatır. Başarı mesajı (Snackbar) gösterir. Hizmet listesini yeniler.
 - **API Başarısız:** Yükleniyor göstergesi kaybolur. Hata mesajı (Snackbar) gösterir.
- **Kullanıcı (Alternatif):** "İptal" butonuna dokunur.
- **Sistem:** Modal sayfayı kapatır.

7. Hizmet Silme:

- **Kullanıcı:** Listedeki bir hizmetin kartındaki "Sil" (`TextButton`) butonuna dokunur.
- **Sistem:** Onay iletişim kutusu (AlertDialog) gösterir.
- **Kullanıcı:** "İptal" veya "Sil" butonuna dokunur.
- **Sistem (Sil Seçildiyse):** İletişim kutusunu kapatır. `_deleteService` fonksiyonunu çağırır. `ServiceController.deleteService` ile API'ye silme isteği gönderir. Yükleniyor göstergesi belirebilir.
 - **API Başarılı:** Başarı mesajı (Snackbar) gösterir. Hizmet listesini yeniler.
 - **API Başarısız:** Hata mesajı (Snackbar) gösterir.
- **Sistem (İptal Seçildiyse):** İletişim kutusunu kapatır.

5.4. Tüm Randevuları Yönetme (Salon Sahibi)

1. **Kullanıcı:** Yan menüden "Randevular" seçeneğine veya Ana Paneldeki ilgili Hızlı Erişim kartına dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.salonOwnerAppointments)` ile `SalonOwnerAppointmentsScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadInitialData` ile tüm randevuları ve kuaförleri çeker. Varsayılan olarak "Liste" tabı ve "Tümü" durumu filtresi seçilidir.

4. Liste Görünümü:


- **Kullanıcı:** Arama çubuğuna (`_searchController`) müşteri adı/telefon/e-posta yazar.
- **Sistem:** Listeyi anlık olarak filtreler.
- **Kullanıcı:** Kuaför filtresi (`DropDownButton`) içinden bir kuaför seçer veya "Tüm Kuaförler"i seçer.
- **Sistem:** Listeyi seçilen kuaföre göre filtreler.
- **Kullanıcı:** Durum filtrelerine (Tümü, Bekleyen vb. - `_buildFilterChip`) dokunur.
- **Sistem:** Seçilen filtreye göre `_getFilteredAppointments` ile randevu listesini (`_buildAppointmentCard`) günceller. Liste boşsa "Randevu bulunamadı" mesajı gösterir.
- **Kullanıcı:** Bir randevu kartındaki ilgili butonlara dokunur (Onayla, Reddet, Tamamla, İptal).
- **Sistem:** `_updateAppointmentStatus` fonksiyonunu çağırır. Onay alır (gerekirse), API isteği gönderir ve sonucu mesajla bildirir, listeyi günceller.

5. Takvim Görünümü:

- **Kullanıcı:** "Takvim" tabına dokunur.
- **Sistem:** `TableCalendar` widget'ını gösterir. Randevusu olan günler altında işaretleyici gösterir. Seçili güne ait randevuları aşağıda listeler.
- **Kullanıcı:** Takvimde farklı bir güne dokunur.
- **Sistem:** Seçilen günü günceller ve o güne ait randevuları listeler. Liste boşsa bilgi mesajı gösterir.

- **Kullanıcı:** Listelenen randevu kartlarındaki butonlara dokunarak işlem yapabilir.

6. Yeni Randevu Ekleme (Manuel):

- **Kullanıcı:** Sağ alttaki Ekle butonuna (FloatingActionButton ) dokunur.
- **Sistem:** `_showAddAppointmentModal` fonksiyonunu çağırarak modal sayfayı açar (Müşteri bilgileri, Kuaför seçimi, Tarih/Saat seçimi - Hizmet seçimi bu modalda eksik görünüyor, eklenmeli).
- **Kullanıcı:** Gerekli tüm bilgileri doldurur/seçer.
- **Kullanıcı:** "Randevu Oluştur" butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır.
 - **Hata Varsa:** Hata mesajı gösterir.
 - **Hata Yoksa:** `AppointmentController.createAppointment` ile (hizmetler de eklenerek) API'ye istek gönderir.
 - **API Başarılı:** Modal sayfayı kapatır. Başarı mesajı (Snackbar) gösterir. Randevu listesini yeniler.
 - **API Başarısız:** Hata mesajı (Snackbar) gösterir.
- **Kullanıcı (Alternatif):** "İptal" butonuna dokunur.
- **Sistem:** Modal sayfayı kapatır.

5.5. Raporları Görüntüleme

1. **Kullanıcı:** Yan menüden "Raporlar" seçeneğine veya Ana Paneldeki ilgili Hızlı Erişim kartına dokunur.
 2. **Sistem:** `Get.toNamed(AppRoutes.salonOwnerReports)` ile `SalonOwnerReportsScreen` 'e yönlendirir.
 3. **Sistem:** Ekran yüklenirken `_loadData` ile varsayılan tarih aralığı (örn. son 30 gün) için randevu, kuaför ve hizmet verilerini çeker. Varsayılan olarak "Gelir Raporu" tabı seçilidir.
- #### 4. Tarih Filtreleme:
- **Kullanıcı:** "Başlangıç Tarihi" veya "Bitiş Tarihi" alanlarına dokunur.

- **Sistem:** Tarih seçme arayüzünü (DatePicker) gösterir.
- **Kullanıcı:** Tarihleri seçer.
- **Kullanıcı:** "Filtrele" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. `_loadData` ile seçilen yeni tarih aralığı için verileri tekrar çeker ve raporları günceller.

5. Rapor Görüntüleme:

- **Kullanıcı:** "Gelir Raporu", "Randevu Raporu", "Müşteri Raporu" veya "Hizmet Raporu" tablarından birine dokunur.
- **Sistem:** Seçilen tab'a göre ilgili verileri (hesaplanmış istatistikler, listeler, grafik benzeri gösterimler) ekranda gösterir.
- **Gelir:** Toplam gelir, kuaför/hizmet bazında dağılım ve yüzdeler.
- **Randevu:** Toplam sayı, durum dağılımı (progress bar ile), kuaför dağılımı, saat/gün yoğunluğu (bar grafik benzeri).
- **Müşteri:** Toplam/yeni müşteri sayısı, en çok harcayanlar listesi, en sık gelenler listesi.
- **Hizmet:** En popüler hizmetler listesi (kullanım sayısı ve yüzde ile), hizmet dağılımı görselleştirmesi.

5.6. Salon Ayarlarını Yönetme

1. **Kullanıcı:** Yan menüden "Ayarlar" seçeneğine veya Ana Paneldeki Ayarlar ikonuna dokunur.
2. **Sistem:** `Get.toNamed(AppRoutes.salonOwnerSettings)` ile `SalonOwnerSettingsScreen` 'e yönlendirir.
3. **Sistem:** Ekran yüklenirken `_loadSalonData` ile mevcut salon bilgilerini ve ayarlarını yükler. Varsayılan olarak "Salon Bilgileri" tabı seçilidir.

4. Salon Bilgileri Tabı:

- **Kullanıcı:** Salon adı, adres, telefon, e-posta, web sitesi alanlarını (`CustomTextField`) düzenler.
- **Kullanıcı:** "Salon Bilgilerini Güncelle" (`CustomButton`) butonuna dokunur.

- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır. Hata yoksa `_updateSalonInfo` → `SalonController.updateSalon` ile API'ye istek gönderir. Sonucu Snackbar ile bildirir.

5. Salon Ayarları Tabı:

- **Kullanıcı:** "Online Randevu Alma" (`Switch`), "Varsayılan Randevu Süresi" (`DropDownButtonFormField`), "Minimum Bildirim Süresi", "İptal Süre Sınırı", "SMS ile Hatırlatma", "Hatırlatma Zamanı", "E-posta Zorunlu" gibi ayarları değiştirir.
- **Kullanıcı:** "Salon Ayarlarını Güncelle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. `_updateSalonSettings` → `SalonController.updateSalonSettings` ile API'ye istek gönderir. Sonucu Snackbar ile bildirir.

6. SMS Ayarları Tabı:

- **Kullanıcı:** "SMS Hizmeti Aktif" (`Switch`) anahtarını değiştirir. Aktifse, API Key, Gönderici ID ve mesaj şablonlarını (`CustomTextField`) düzenler.
- **Kullanıcı:** "SMS Ayarlarını Güncelle" (`CustomButton`) butonuna dokunur.
- **Sistem:** Butonda yükleniyor göstergesi belirir. Form validasyonunu çalıştırır (API Key vb. zorunlu olabilir). Hata yoksa `_updateSmsSettings` → `SalonController.updateSmsSettings` ile API'ye istek gönderir. Sonucu Snackbar ile bildirir.

7. Şifre Değiştirme (Alt Bar):

- **Kullanıcı:** Ekranın altındaki "Şifre Değiştir" (`CustomButton` , type: outlined) butonuna dokunur.
- **Sistem:** `_showChangePasswordModal` ile modal sayfayı açar.
- **Kullanıcı:** Mevcut şifre, yeni şifre ve yeni şifre tekrarını girer.
- **Kullanıcı:** Modaldaki "Şifre Değiştir" (`CustomButton`) butonuna dokunur.
- **Sistem:** Form validasyonunu çalıştırır. Şifrelerin eşleşip eşleşmediğini kontrol eder.
 - **Hata Varsa:** Hata mesajı (Snackbar veya modal içinde) gösterir.

- **Hata Yoksa:** `_changePassword` fonksiyonunu çağırır. (Backend'de ilgili API çağırısı yapılmalı). Başarılı olursa modalı kapatır ve Snackbar ile bilgi verir. Başarısız olursa hata mesajı gösterir.

- **Kullanıcı (Alternatif):** Modaldaki "İptal" butonuna dokunur.
- **Sistem:** Modalı kapatır.

8. Çıkış Yap (Alt Bar):

- **Kullanıcı:** Ekranın altındaki "Çıkış Yap" (`CustomButton`) butonuna dokunur (Bkz. Genel Akış 2.3).

Bu detaylı akışlar, uygulamanın tüm roller için nasıl çalışacağına dair daha kapsamlı bir görünüm sunmaktadır.

6. Teknik Gereksinimler (Technical Requirements)

- Frontend için flutter
- Backend için python flask
- Database için postgresql

7. Kullanıcı Arayüzü ve Deneyimi (UI/UX Requirements)

- Arayüz minimalist bir tasarıma sahip olmalı ve tüm tasarımlar belirgin olmalı
- Mobil uyumlu olmalı
- Responsive bir tasarım olmalı

8. Güvenlik ve Yetkilendirme (Security & Authentication)

- Kuaförler ve salon sahipleri işlem yapabilmesi için oturum yapabilmeli

9. Kısıtlamalar ve Varsayımlar (Constraints & Assumptions)

10. Başarı Kriterleri ve Kabul Şartları (Success Criteria & Acceptance Criteria)

- Projenin başarıyla tamamlanmış sayılması için hangi kriterlerin karşılanması gerekiyor?

“Proje, müşteri randevu aldığı anda, bu randevu kuaförün takvimine yansiyorsa ve kuaför tarafından onaylandığında anlık olarak müşteri ekranında güncelleniyorsa başarılı kabul edilir. Aynı zamanda, başka bir müşteri aynı saat için randevu almaya çalıştığında uygun saatlerin doğru şekilde güncellenmesi gerekmektedir.”