

**Bölüm Adı: Elektrik ve Otomasyon**

**Program Adı: Mobil Teknolojileri**

**Ders Adı: İos Programlamaya Giriş**

**Proje Adı: Otel Uygulaması**

**19MY93004**

**Bilal Mercan**

İçindekiler

[Swift 4](#_Toc38015254)

[Swift Playgrounds 4](#_Toc38015255)

[Xcode 4](#_Toc38015256)

[Navigator Panel 8](#_Toc38015257)

[Standart Editor 9](#_Toc38015258)

[Utilities Panel 10](#_Toc38015259)

[ToolBar 11](#_Toc38015260)

[Sayfa oluşturma 12](#_Toc38015261)

[İçerik ekleme 13](#_Toc38015262)

[Buton Ekleme 15](#_Toc38015263)

[Constraint Ekleme 15](#_Toc38015264)

[Sayfa geçişleri 16](#_Toc38015265)

[Buton ile Sayfa kodunu birbirine bağlama 17](#_Toc38015266)

[İmage view 18](#_Toc38015267)

[İmage view attributes 19](#_Toc38015268)

[Text View 20](#_Toc38015269)

[Text view attributes 20](#_Toc38015270)

[Sayfa Swift Dosyası 21](#_Toc38015271)

[Sayfa geçiş türleri 22](#_Toc38015272)

[Label 23](#_Toc38015273)

[Label Tanımlama 23](#_Toc38015274)

[Switch 24](#_Toc38015275)

[Switch Tanımlama 24](#_Toc38015276)

[Stepper 25](#_Toc38015277)

[Stepper Tanımı ve Fonksiyonu 25](#_Toc38015278)

[TextField 26](#_Toc38015279)

[TextField Tanımı 26](#_Toc38015280)

[Segmented Control 26](#_Toc38015281)

[Segmented Control Tanımı 26](#_Toc38015282)

[İndicator View 27](#_Toc38015283)

[İndicator View Tanımı 27](#_Toc38015284)

[Sayfalar Arası Veri Alış-Verişi 28](#_Toc38015285)

[Veri Gönderme 28](#_Toc38015286)

[Veri Alma 29](#_Toc38015287)

[Uyarı Mesajı 30](#_Toc38015288)

[İkon 31](#_Toc38015289)

[Uygulama konusu 32](#_Toc38015290)

[Uygulama ekran görüntüleri 32](#_Toc38015291)

[Proje, Sunum, Rapor, Afiş Github Adresi 32](#_Toc38015292)

[Kaynakça 33](#_Toc38015293)

# Swift

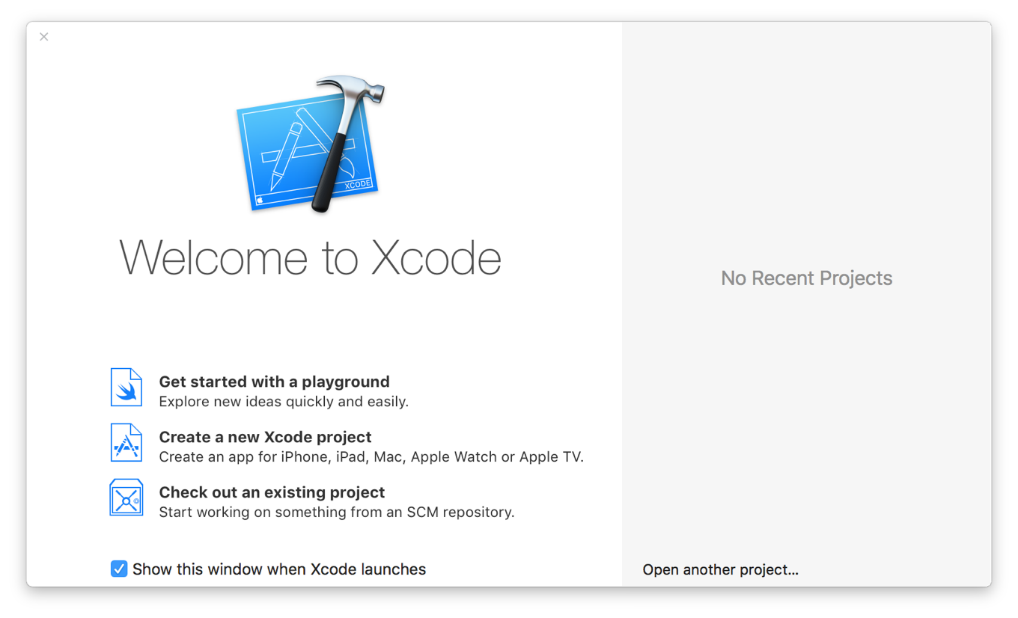
Swift, C ve Objective-C yazılım dilleri tabanlı oluşturulmuş ve C ve Objective-C dillerinin karmaşasından kurtarılarak hazırlanmış bir yazılım dilidir. Genel programlama yapısından kopmadan, esnek ve kolay kod yazılabilecek şekilde tasarlanmış mimariye sahiptir. Swift yazılım dilinde bellek yönetimi olarak Automatic Reference Counting (ARC) sistemi kullanılmaktadır. Swift dili Objective-C dilini tamamen kapsamakta ve Objective-C ile yazılan tüm kodları, kütüphaneleri destekleyerek daha modern ve geliştirilmesi kolay bir yapıya ulaşmıştır. Objective-C dilinin aksine Swift yazılım geliştirmeye yeni başlayanlar için oldukça basit ve anlaşılabilir yapıdadır.

# Swift Playgrounds

Apple Swift yazılım dili ile birlikte playgrounds sunumunu gerçekleştirdi. MacBook’unuza XCode kurduğunuzda otomatik olarak gelen playgrounds, iPad ile sunulan swift playgrounds uygulamasından biraz farklı. XCode ile birlikte sunulan playgrounds yazdığınız kodu anında çalıştırarak sonucu görmenizi sağlarken, iPad uygulaması biraz daha öğrencilere yönelik olarak Swift ile kod yazma mantığını öğretmeye çalışan hem oyun hem de bir uygulama.

# Xcode

Xcode bir IDE(integrated development environment) uygulamasıdır. Bünyesinde programlama editöründen test uygulamasına, tasarım oluşturmaktan hata ayıklama programlarına kadar pek çok özelliği içeren paketlere IDE diyoruz. Genellikle bu IDE uygulamaları kodlama editörlerinin ismini alır. Xcode uygulaması da ismini bu şekilde alır. Uygulama paketi sayesinde uygulamanızın yazılımını, tasarımını, hata ayıklamasını ve test edeceği simülasyon cihazlarını tek bir uygulama içerisinden yapabiliyor.

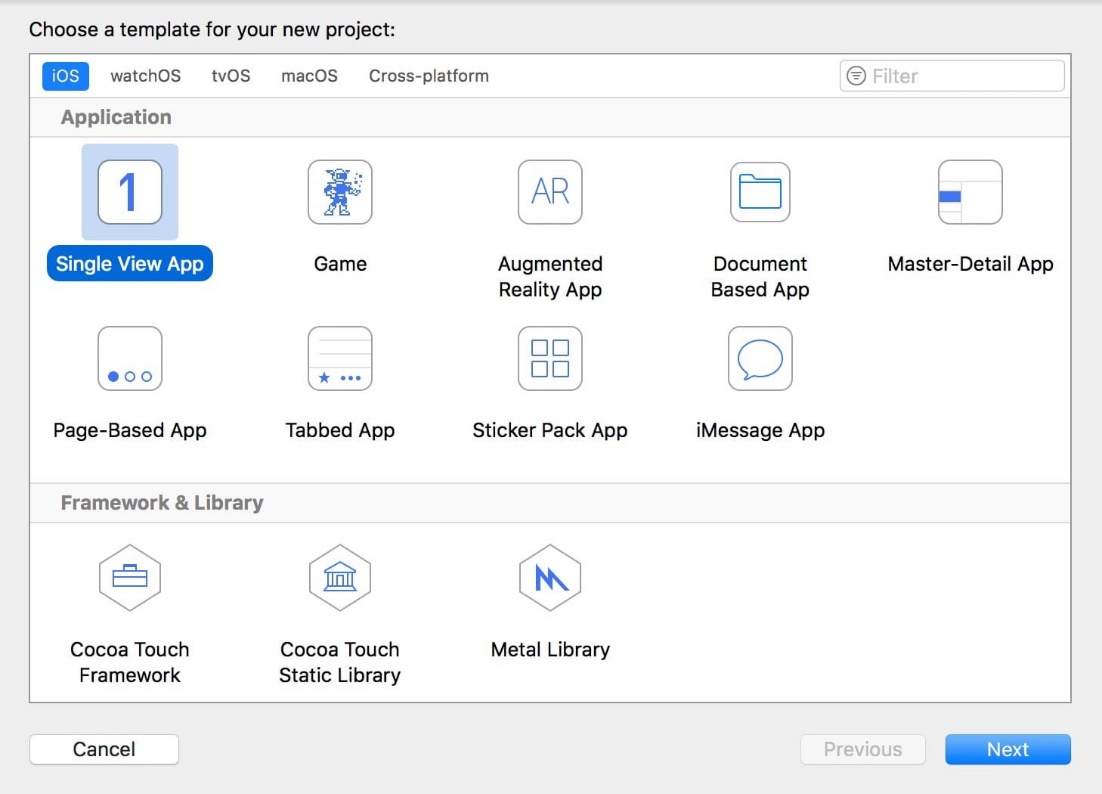


**Get started with a playground** ile sadece kod editörünü açıp burada Swift kodları yazabiliriz. Bu kısım normal bir proje yaratmadan farklı olarak sadece basit kodlar yazıp temel uygulama ekranları gösterebileceğiniz bir alan. Yeni başlayanların bu ekran üzerinde kod yazmaları ve WWDC (Worldwide Developers Conference) etkinliği için uygulama geliştiren öğrencilerin bu **playground** dosyası üzerinde çalışmaları tavsiye edilir.

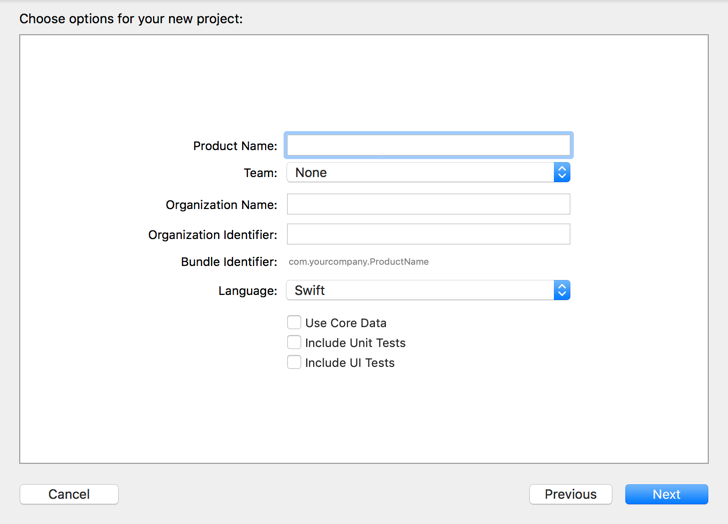
**Create a new Xcode project;** ile sıfırdan bir uygulama şablonu yaratırsınız. Xcode sizlere tüm arkaplan işleri bitmiş bir kalıp hazırlar. Böylece tek yapmanız gereken kodlamaya başlamak ve tasarımınızı hazırlamak. Uygulama geliştirmek isteyenlerin bu opsiyonu seçmesi önerilir.

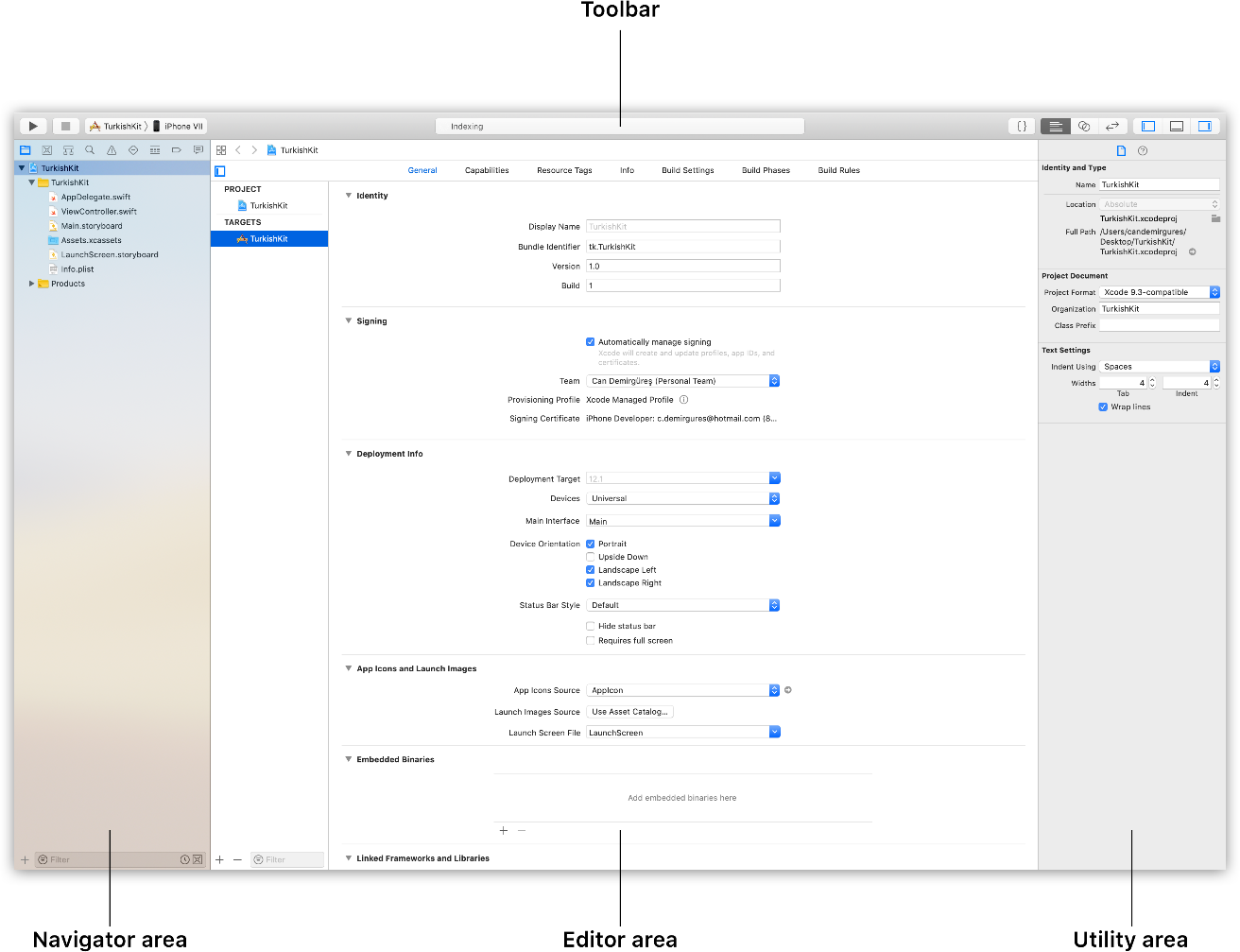
**Clone an existing project;** bizlere Github, Gitlab ve benzeri Git tabanlı projeleri internetten çekip onun üzerinde çalışmamızı sağlayan bir alan. Xcode proje yaratma kısmı ile aynı olan bu kısım bizlere internette hazır olan projeyi alabilme kolaylığı sağlıyor.

Açılış ekranının sağ tarafında daha önce başlanmış bir proje var ise bu sekmede görülür ve bu sekmeden ulaşılabilir. Get started with a playground seçeneğine tıklayıp yeni proje sayfasını açtıktan sonra karşımıza gelen ekran;



Burada farklı şablonlar bulunuyor. Eğer şablon kullanılması istenilmiyorsa Single View App seçeneği seçilerek uygulamayı şablonsuz olarak oluşturabiliriz.

Bu ekranda proje ile ilgili proje adı, takım (isteğe bağlı seçim), uygulama paket adı (AppStore Mağazasında Gösterilecek Şirket veya Organizasyon ismi), AppStore mağazasında uygulama geliştirici kimliğini ve Uygulama dili seçenekleri bulunuyor. Veri tabanı kullanılacaksa Use Core Data seçeneği işaretlenmelidir. Aynı şekilde uygulama test edilmek isteniliyorsa diğer iki seçenekte işaretlenmelidir.



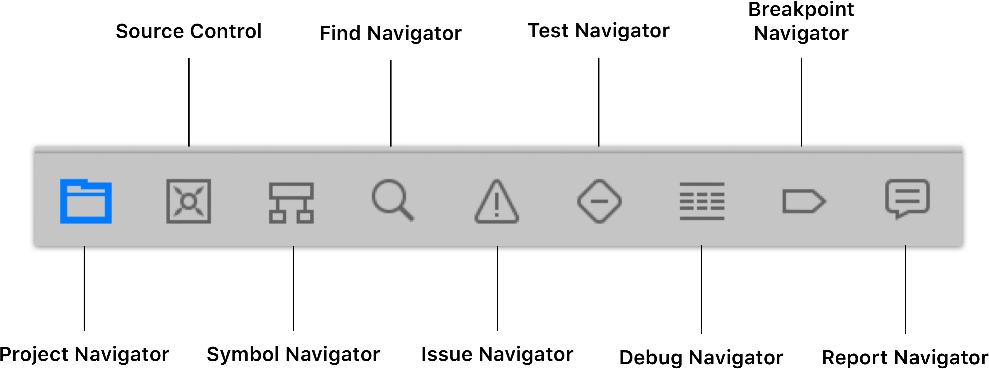
Xcode ekranı dört panelden oluşuyor. Bu panellere **Navigator Panel**, **Standart Editor, Utilities Panel** ve **Toolbar olarak dört kısma ayrılıyor. Öncelikle bu panelleri tanıyalım.**

# ****Navigator Panel****

**.swift** dosyaları yazdığımız kodları içerir ve burası uygulamanın arka yüzüdür. Uygulamanın çalıştıracağı tüm kodlar, hesaplayacağı işlemlerin kodları ve uygulamanın ana işlem merkezi burada saklanır.

**.xcassets** dosyası bir kategoridir. Renk, data ve resim kategorileri yer alır. Bu sayede sol panel resimlerle dolmaz ve uygulama içinde resim, renk ve data kategorilerini daha rahat kullanabiliriz.

**.storyboard** dosyası tasarım dosyasıdır ve uygulamanın ön yüzünü oluşturur. Burada kullanıcılar buton, resim vb. gibi dosyalarını sürükleyerek oluşturur, onlara şekil verir ve uygulamanın ana tasarımını oluştururlar.

**.plist** dosyası ayar dosyasıdır ve burada uygulamanın aldığı izinler, kullanıcıya sunulacak teknolojilere verilen izinler gibi ayarlar yer alır. Bu bölümün çoğu hazırdır ve genellikle sadece telefon, kamera izni gibi gerekli birkaç izin gerektiğinde burada oluşturulur.

**Project Navigator** varsayılan olarak seçili ve tüm dosyaları gösteriyor.

**Source Control**uygulamamızı Git tabanına bağlayan ve oradaki ayarları yapmamızı sağlayan kısım olarak işe yarıyor.

**Symbol Navigator**kodlarda kullandığımız tüm sınıf ve objeleri görebilir, koddaki yerini bulunabilir.

**Find Navigator** ile kod içindeki aranılan şeyler bulunabilir.

**Issue Navigator** bize uygulamanın oluşturulma ve çalışma sırasındaki hatalarını gösterir.

**Test Navigator** ise bize uygulamadaki tasarım dosyalarını test edip hata aranıldığı zaman işe yarar.

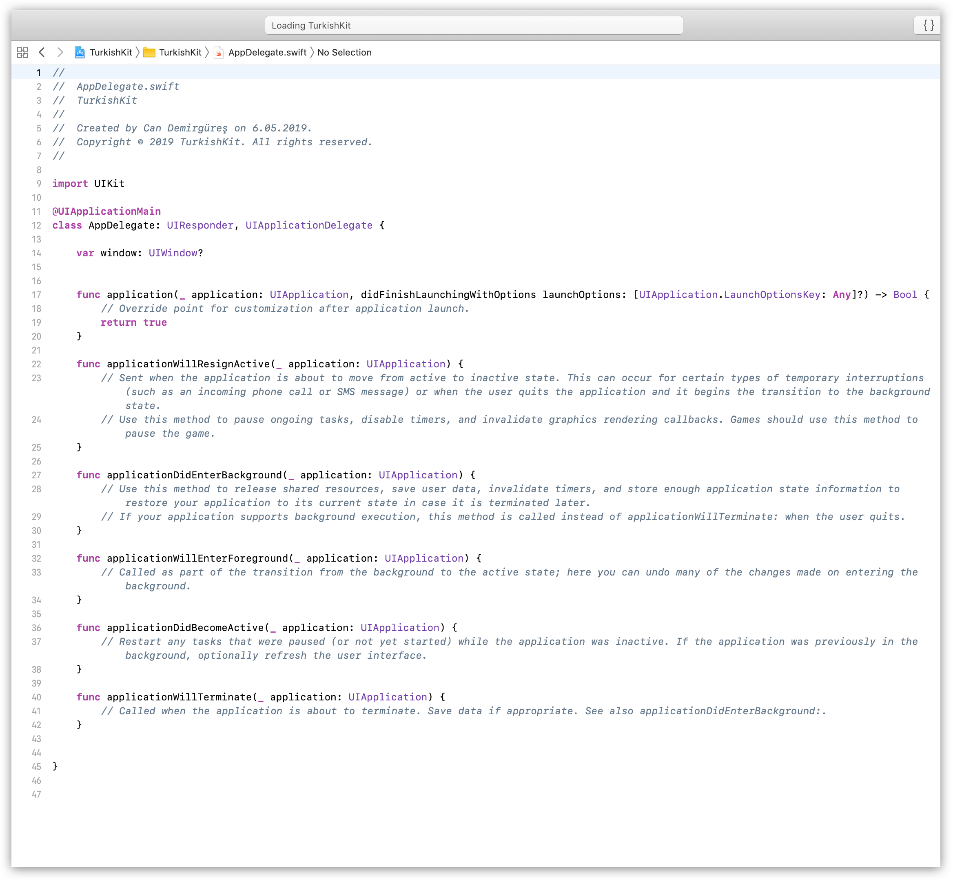
**Debug Navigator** uygulamayı çalıştırırken test etmek için koyulan hata noktalarını ve burada yaratılan detayları gösteriyor.

**Breakpoint Navigator** kod içine koyulan sınırları gösteriyor ve uygulamanın ilgili fonksiyonu, kodun çalıştırıldığından emin olunmasını ve bu kodun nasıl çıktı verdiğini anlık görülmesini sağlıyor.

**Report Navigator** uygulamanın bütün oluşturduğu kayıtları veriyor ve bir hatadan çalışma sorununa kadar çıkabilecek her türlü durumun detayını görülmesine olanak sağlıyor.

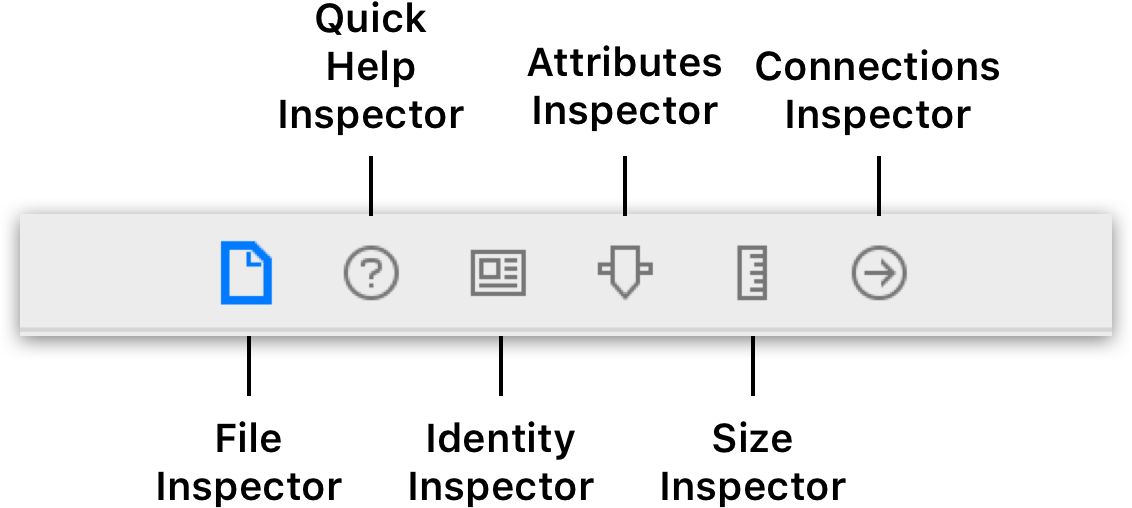
# ****Standart Editor****

Bu panel kodları ve tasarımları gösteriyor. İçerisinde birçok özellik bulunduran bu panel sayesinde kodları görebilir, tasarımını oluşturabilir, tasarım sırasında cihaz görünümlerini görebilir ve bu ikisini yan yana getirerek birbirleri ile bağlantıları kurulabilir.



# ****Utilities Panel****

Bu panel pek çok kısayol sekmesi sunuyor.



**File Inspector** dosya adı ve dosyanın türünü, nerede kaydedildiğini ve tasarım aracının hangi Xcode ve Swift sürümü ile kurulduğunu gösteriyor.

**Quick Help** **Inspector** kod ile uğraşırken bir sınıfı seçildiğinde o sınıfın dokümantasyonunu ve örnek kodunu göstererek yardımcı oluyor.

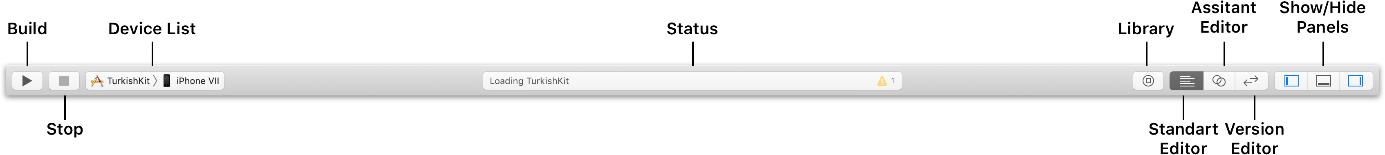
**Identity Inspector** seçilen objenin hangi sınıfa bağlı olduğunu görüp eğer oluşturulduysa özel sınıfa bağlanılmasını sağlıyor. Ayrıca buradan Storyboard dosyasına bir id yazarak kod içerisinden çağırılabilir.

**Attributes Inspector** seçilen objenin tüm düzenlenebilir ayarlarını sunuyor ve buradan font, boy, renk, arkaplan gibi pek çok özelliğini değiştirilebilmesine olanak sağlıyor.

**Size Inspector** çeşitli arayüz objelerinin boyutlarını değiştirilmesini sağlıyor.

**Connection Inspector** ise kod ile tasarımın arasındaki bağlantıları gösteriyor. Bu sayede eğer uygulama daha önce silinen bir objenin referansını içeriyorsa o referansı buradan silinebilir.

# ToolBar

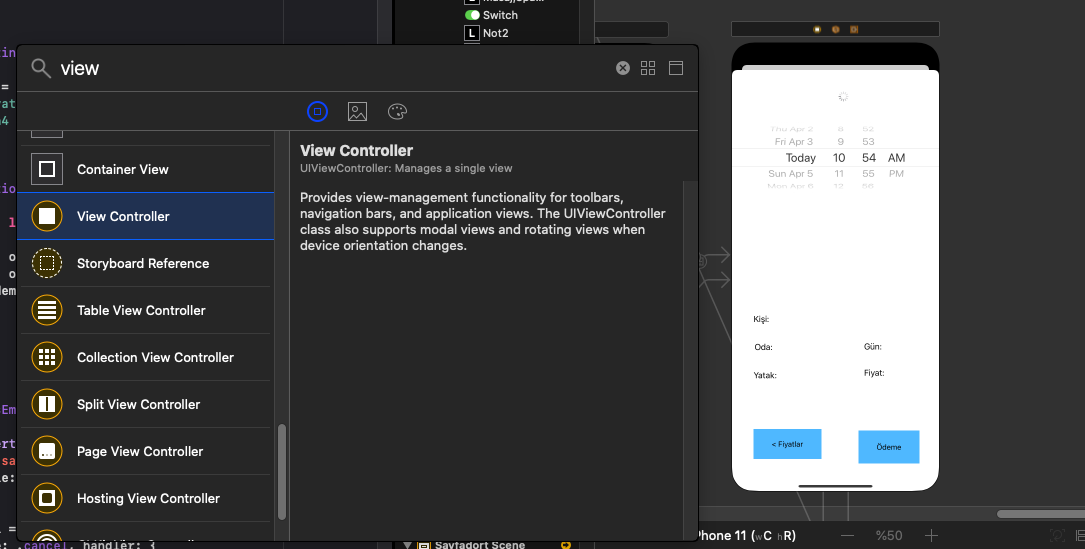


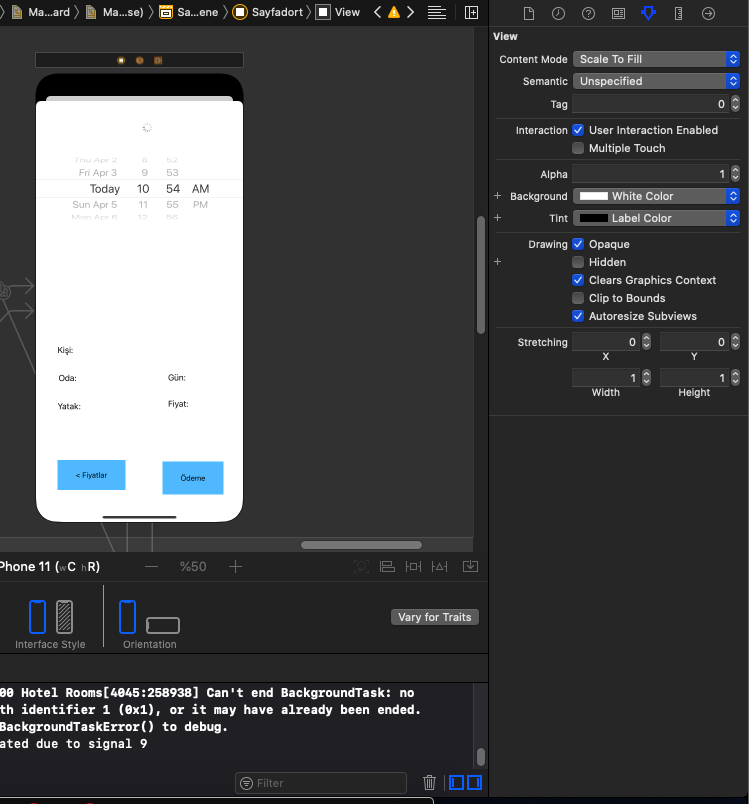
Burada uygulamaya ait ana aksiyonlar yer alıyor. Sol tarafta uygulamayı başlatma ve durdurma butonları yer alıyor. Ayrıca uygulamayı çalıştırılacak cihazı da burada seçilebiliyor.

Sağ köşede ise kütüphane butonu UI elemanları ve hazır kod bloklarını içeren bir liste sunuyor. Sürükle bırak ile seçilebilir. Hemen yanında yer alan 3 seçenek ise kod editörünü, kodun tasarıma bağlanma editörünü ve kodun eski hallerini görülmesini sağlayan versiyon editörünü içeriyor. Son olarak en sağda yer alan 3 seçenek ise gerektiğinde panelleri gizleyip gösterilebilmesine olanak sağlıyor.

# Sayfa oluşturma

Uygulama içerisinde sayfa oluşturmak için öncelikle XCode içerisinde sağ üst kısımda bulunan library (kütüphane) seçeneğine tıklayıp açılan sekme içerisinde View Controller yazıp karşımıza çıkan objeyi ekrana doğru sürüklüyoruz bu sayede ekran oluşturulmuş oluyor.



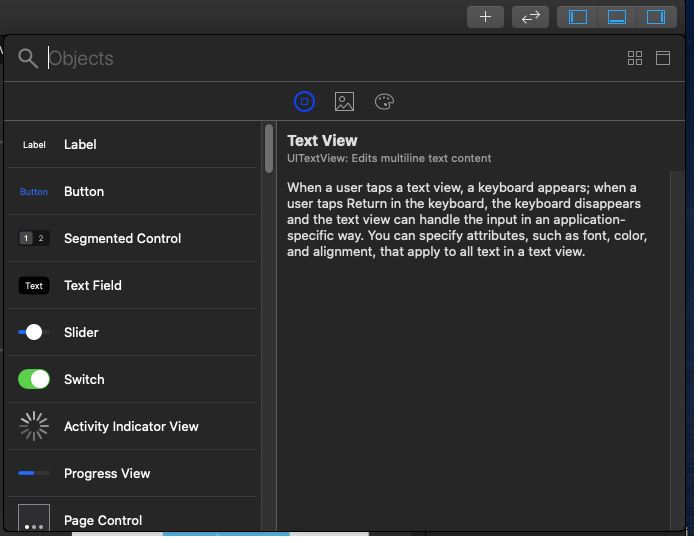


Eklenen ekrana tıkladığımızda ise yan sekmede eklenen ekran ile ilgili seçenekler bulunuyor. Bu seçenekler ile ekranı özelleştirebiliriz.

Bu işlemler arasında arka plan değiştirme, eklenen içeriklerin ekran yerleşimi ayarlanılabilir.

# İçerik ekleme

Uygulama içerisinde kullanılacak olan öğeler aşağıda görülen sekme içerisinde bulunur. Eklenecek olan içerik ekrana sürükleyerek aktarılır.



Uygulamada Kullanılan İçerikler

- Exit Button

-Activity İndicator

-Stepper

-Label

-Switch

-Datepicker

-Segmented Control

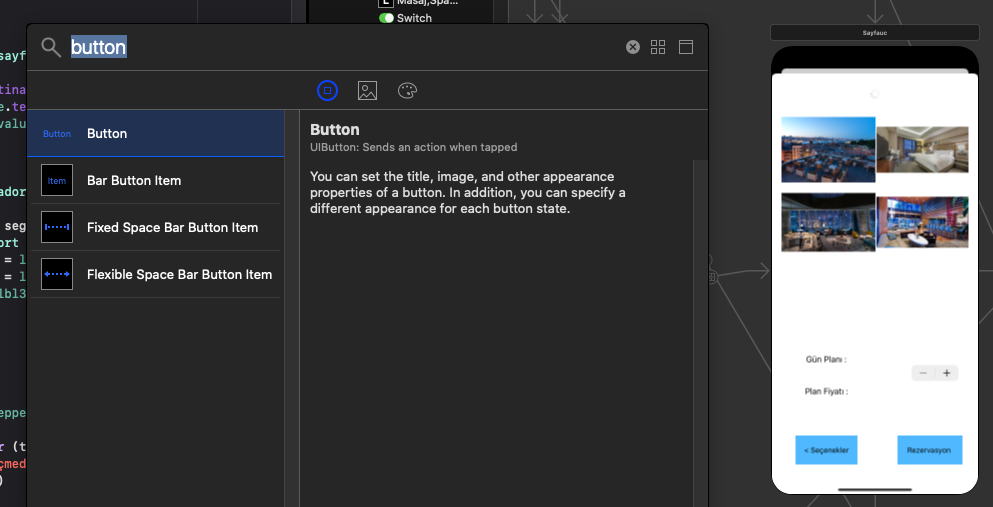
-Button

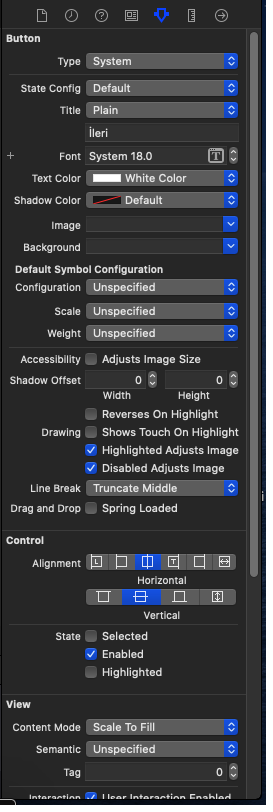
-İmage View

-Text View

# Buton Ekleme

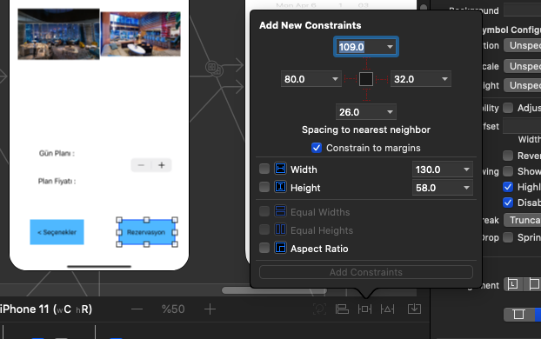
Buton eklemek için öncelikle Object menüsünden aratılır ve ekrana doğru sürükleyerek eklenir. Sonrasında ise buton boyutu kenarlarındaki işaretler ile sürüklenerek ayarlanır. Son adımda ise Constraint layout’ları ayarlanır.



Button attributes

Bu sekmeden buton ile ilgili ayarlamalar yapılabilir.

## Constraint Ekleme

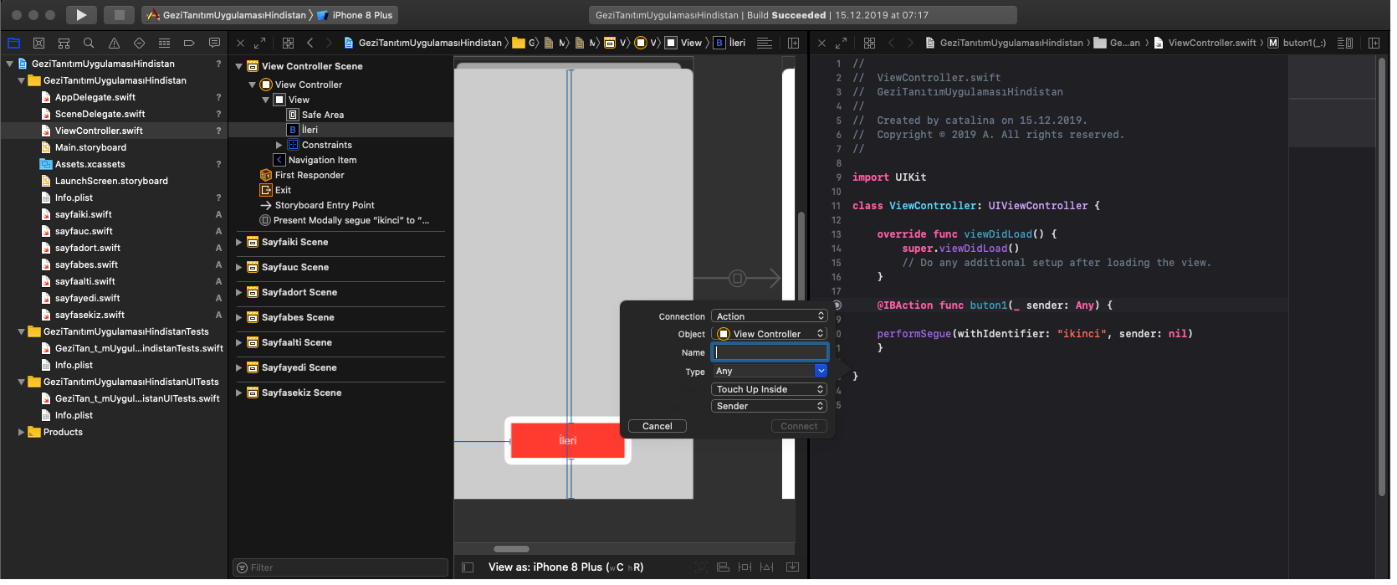
Seçilen içeriğe tıkladıktan sonra sağ alt barda bulunan üçgen simgeye tıklayarak Add Missing Constraint seçeneğine tıklayıp veya manuel olarak değerleri verebiliriz.

# Sayfa geçişleri

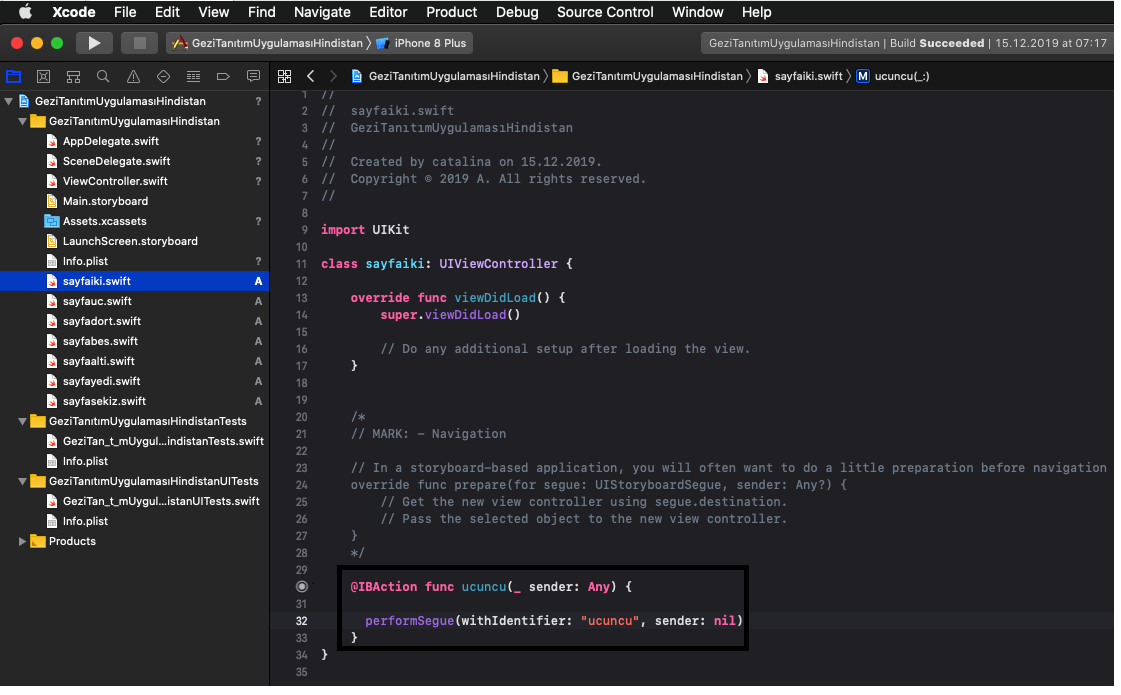
Sayfa geçişlerini ayarlamak için öncelikle butona tıkladıktan sonra cmd veya sanal makinada alt tuşu ile diğer sayfaya sürüklenir ve açılan pencereden Show seçeneği seçilir ve sayfalar birbirine bağlanmış olur.



## Buton ile Sayfa kodunu birbirine bağlama



Buton ile butonun bulunduğu sayfanın swift sayfasını açıp butonu sayfa geçişinde olduğu gibi swift dosyasına sürüklüyoruz ve açılan pencereden Üst kısımdaki seçenekten action seçeneğini seçiyoruz ve ismini belirliyoruz. Bu işlemi her sayfa geçişinde uygulamamız gerekiyor.

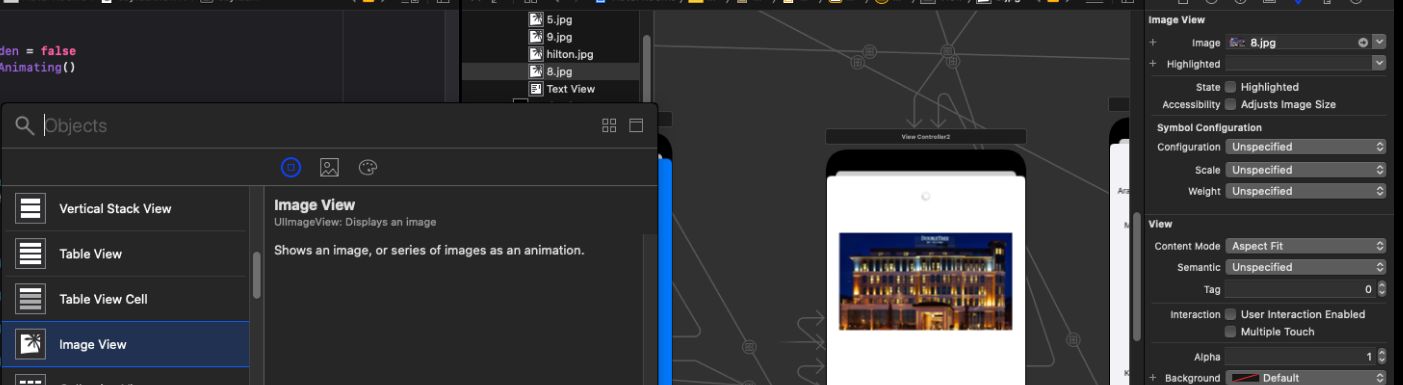


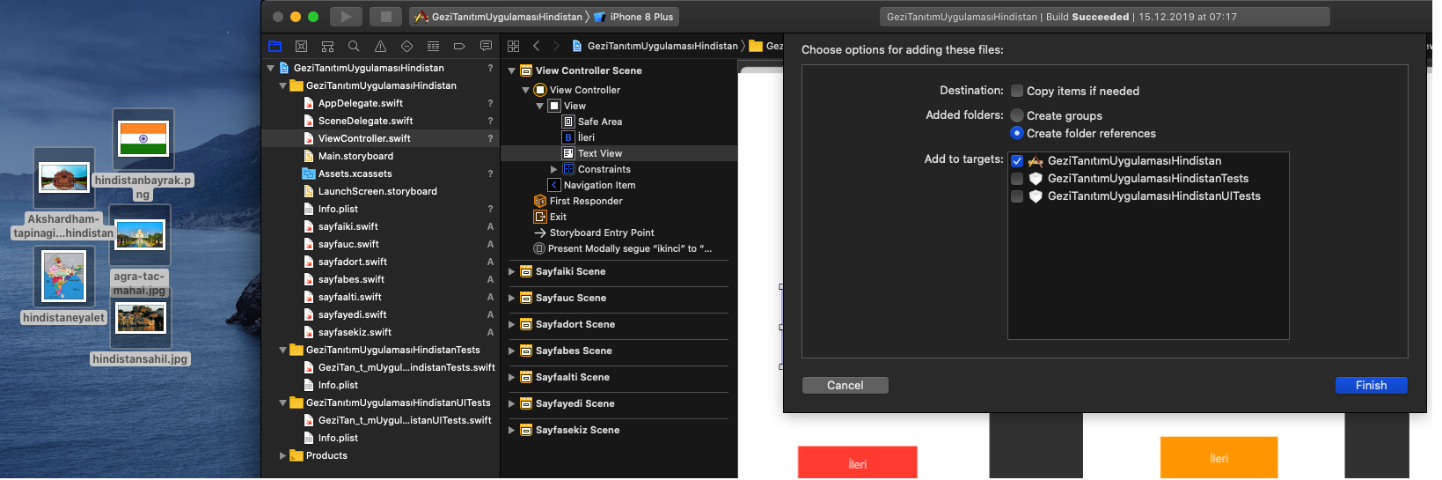
Ekranda görülen dikdörtgen içerisindeki kodu her sayfa için belirlenen identifier’ları girerek yazıyoruz.

# İmage view

İmage view eklemek için buton örneğinde olduğu gibi ekrana sürükleyerek sayfaya ekliyoruz.

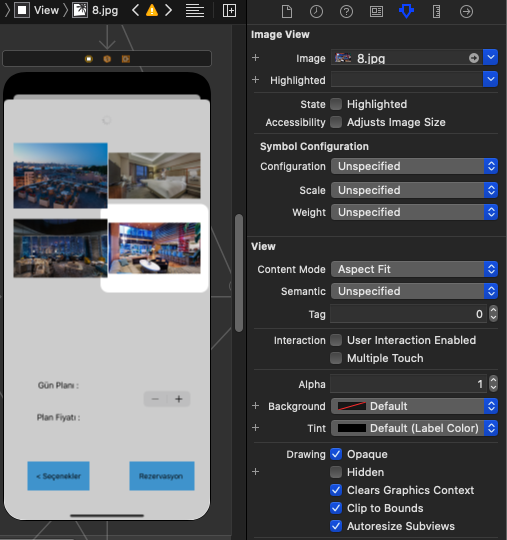
Yine buton örneğindeki constraint ayarlamalarını yapıyoruz ve eklenecek içeriği Xcode ekliyoruz.





Finish e tıkladıktan sonra resim, fotoğraf dosyalarımız yükleniyor.

## İmage view attributes

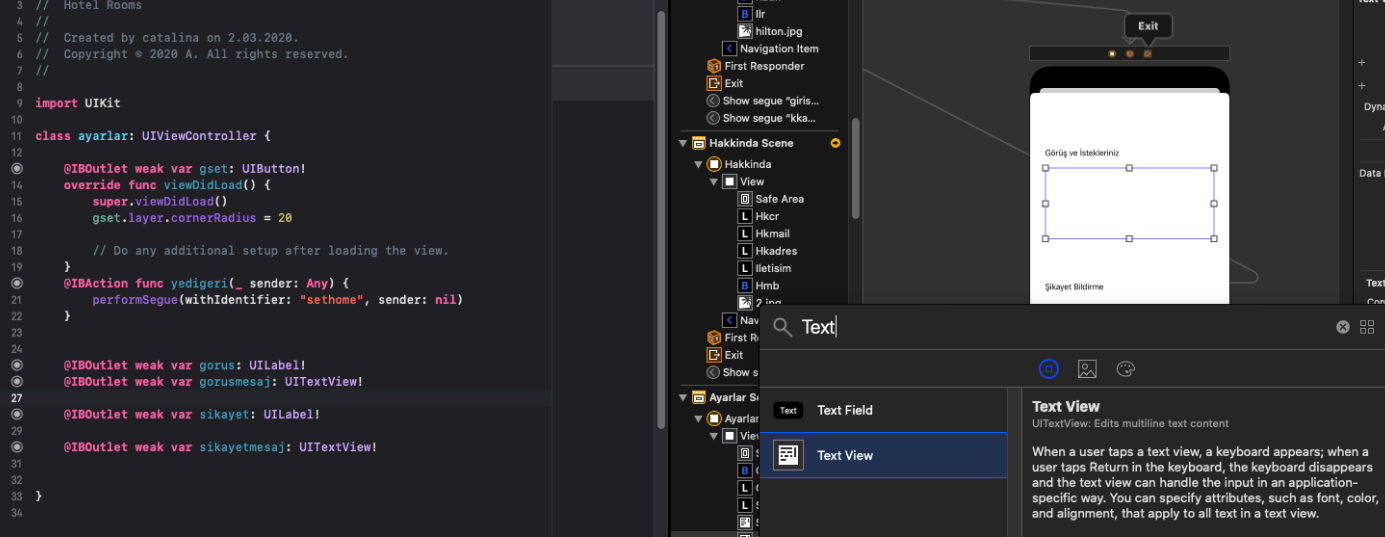


Yüklenen fotoğrafları seçmek için eklenen image view dosyasına tıklayıp sağ sekmede görülen image seçeneğinden yüklenen fotoğrafları seçerek uygulamaya eklenmiş oluyor.

Aynı zamanda yüklenen içerik ile ilgili birçok ayarı bu sekme içerisinden yapılabiliyor.

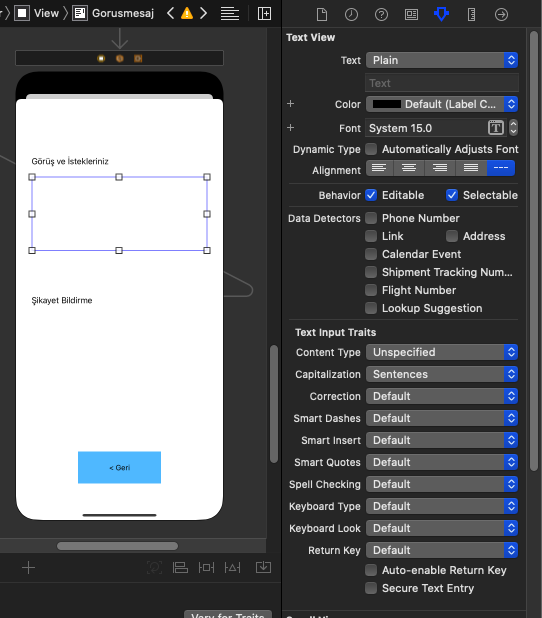
# Text View

Text view eklemek için image view işleminde yapılan işlemlerin aynısını text view için de yapıyoruz. Constraint layout ayarlamalarını da aynı şekilde ayarladıktan sonra text dosyasını düzenlememiz gerekiyor.

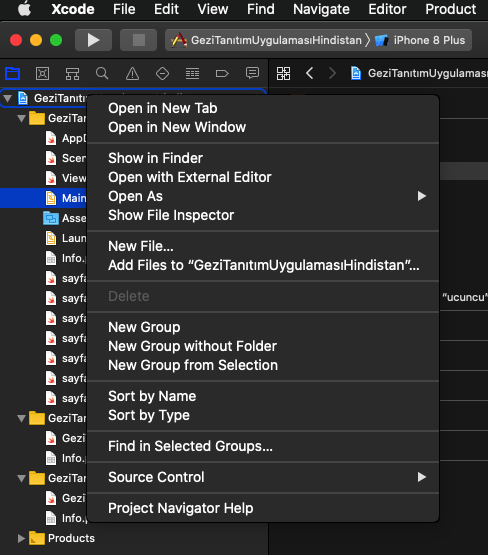


## Text view attributes

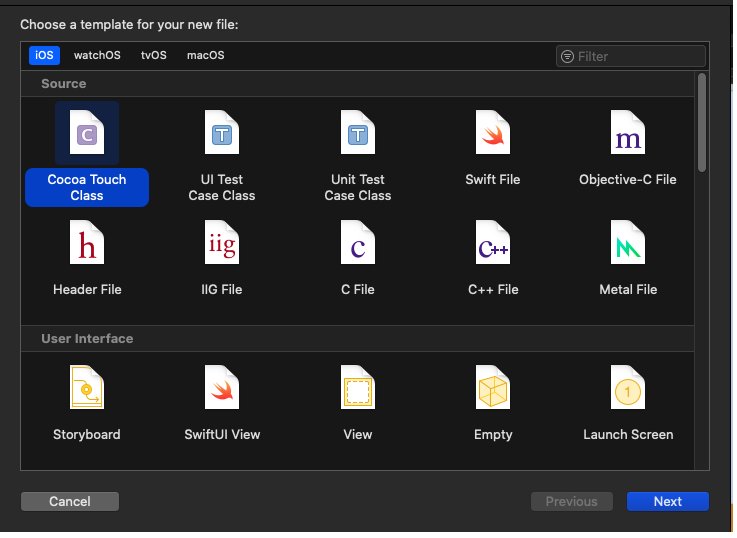
Text view attributes sekmesinden metin, yazı boyutu, yazı tipi ve text view ile ilgili birçok ayarlamaları bu sekmeden gerçekleştiriyoruz.

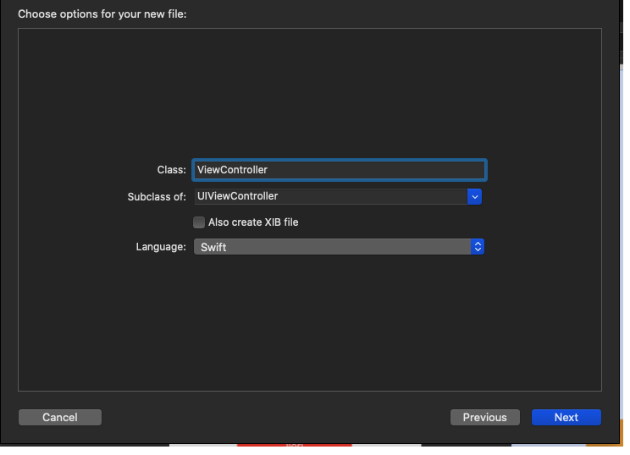


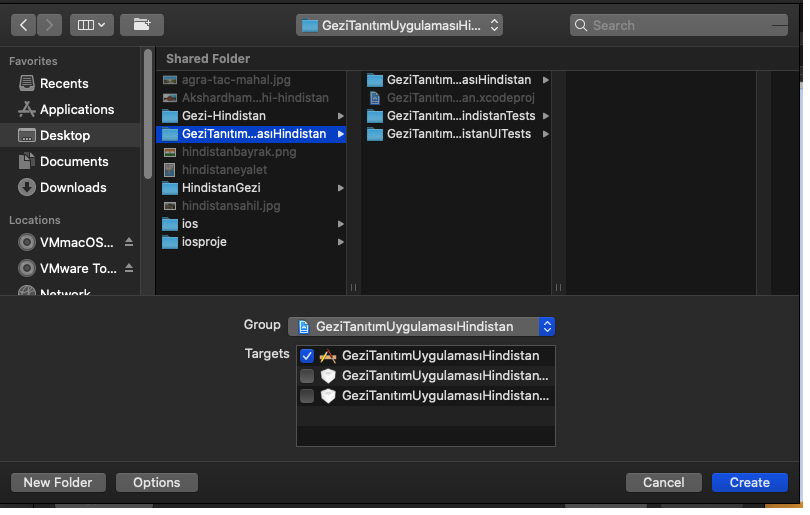
# Sayfa Swift Dosyası



Eklenen view controller ‘lar için sayfa oluşturuyoruz. Proje dosyasına sağ tık ile New file seçeneğini seçerek ilerliyoruz.

Açılan sekmede Cocoa Touch Class dosyasını seçiyoruz ve ilerliyoruz.

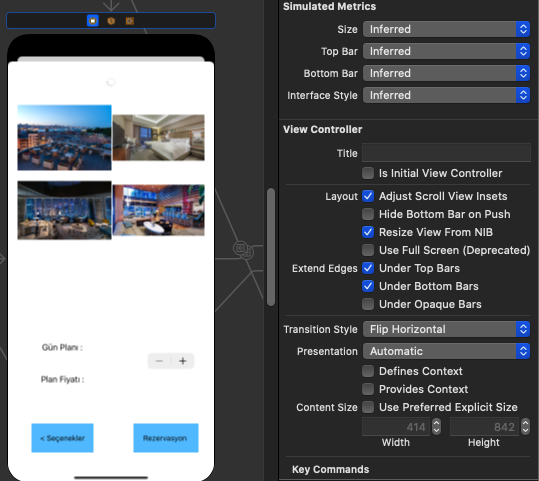
 Burada sayfa ismini belirliyoruz ve altındaki seçeneklerden UIViewController seçeneğini seçiyoruz ve ilerliyoruz.

 Burada istenilen klasör seçildikten sonra dosya oluşturuluyor.

# Sayfa geçiş türleri

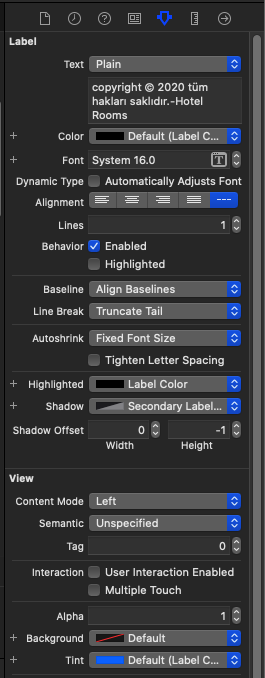
Sayfa geçiş türünü seçmek için öncelikle ekranda görülen Storyboard Segue kısmından

Kind kısmından Present Modally seçeneğini seçip ardından Transition kısmından istenilen geçiş türü seçilir.



# Label

Label daha önceden hazırlanmış metinleri ekranda göstermeye yarar. TextView’ dan farkı textview ile metni düzenleyebiliyorken label’ da kullanıcı düzenleyemez.



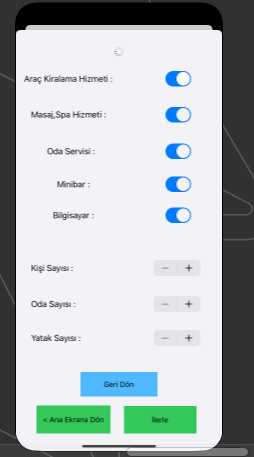
## Label Tanımlama

Eklenen label ilgili swift sayfasına seçilerek sürükleyerek outlet tanımları oluşturulur. Böylece fonksiyon ve bir çok yerde kullanılabilir.



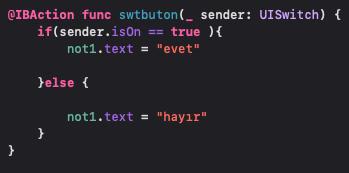
# Switch

Switch butonu evet hayır gibi ikili seçeneklerde veya on off gibi kullanıma uygundur.



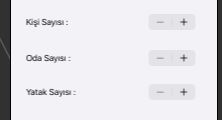
## Switch Tanımlama

Buton örneğinde olduğu gibi action bağlantı oluşturuluyor. Projede kullanılma amacı seçildiğinde evet kapandığında hayır yazısını vermesini sağlamak. Önceden tanımlanan label ile if else kontrol yapısı ile tetiklenecek olayı belirtiyoruz sender.İsOn == true seçildiğinde label’ da evet kapalı ise hayır gözüküyor.



# Stepper

Stepper ekle çıkar görevini görür. Miktar belirlemede ve sayı işlemlerinde sayaç görevi yapılabilir.

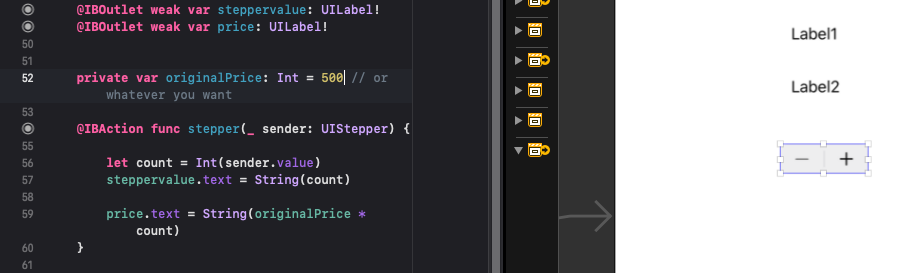


## Stepper Tanımı ve Fonksiyonu

Oluşturulan fonksiyonda count adında değişken oluşturuyoruz. Int değer veriliyor.

Önceden oluşturulan label ile String olarak count değişkenini eşitliyoruz.

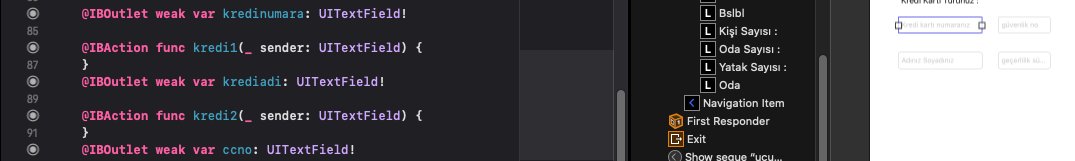
Fonksiyon dışında tanımlanan örnek değişkeni price adındaki label ile String türünde eşitliyoruz.



# TextField

Kullanıcı tarafından düzenlenebilen veri girişinde bulunmak için kullanılabilir.

## TextField Tanımı



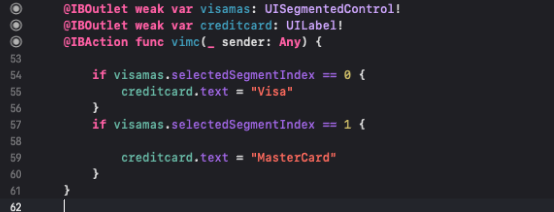
# Segmented Control

Segmented Control; ikili seçenek olarak veya daha da fazla seçenek olarak düzenlenerek kullanılabilir. Switch ile kullanım açısından benzerdir.

## Segmented Control Tanımı

Switch örneğindeki gibi kullanılır. İf ile önceden tanımlanan label içerisine yazı yazdırılır.

SelectedSegmentIndex ile hangi index (seçeneklerden herhangi biri) seçilirse ona göre mesaj verilir.



# İndicator View

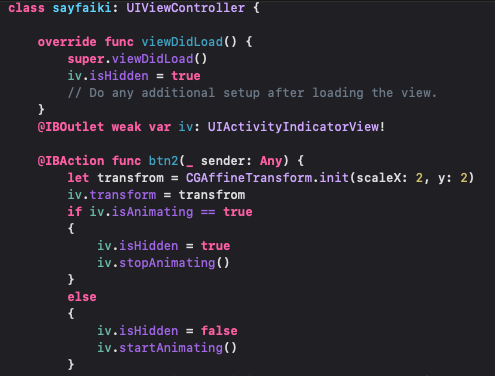
İndicatorView yükleniyor imgesi olarak kullanılır. Uygulama içerisinde işlem sırasında, internet bağlantısı kullanılırken ve buna benzer durumlarda kullanılır.

## İndicator View Tanımı

Outlet olarak tanımlanır. İv olarak tanımlanan indicator override fonksiyonu içerisine

iv.isHidden == true yani görünür olması için eklenir.

Butona tıklandığında tetiklenmesi için buton içerisine if else yapısı ile butona tıklandığında animasyon görünür olarak butona tıklanmadan önce ise görünmez olarak tanımlanıyor.



# Sayfalar Arası Veri Alış-Verişi

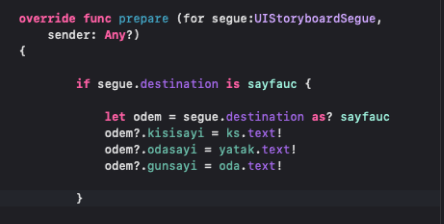
Projede bulunan oda sayısı, kişi sayısı, fiyat gibi verileri diğer sayfalara taşımak için kullanıyorum.

## Veri Gönderme

Stepper ile arttırılan sayı miktarını label’ da tutuluyor.

Bu verileri segue ile gönderimini sağlıyorum. Override fonksiyonu içerisinde

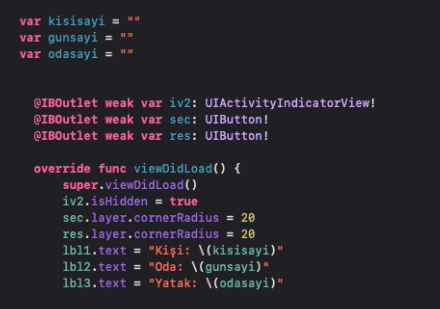
İf ile segue destinasyonunu belirleniyor. Hangi sayfaya iletilmesi isteniliyorsa ilgili segue seçiliyor. odem adında değişken oluşturuluyor. Segue destinasyonunu belirleyerek eşitleniyor. Projede kişi, oda ve yatak sayılarını gönderilen sayfadaki label ile eşitleniyor. Böylece verilerimiz gönderilmiş oluyor.



## Veri Alma

Gönderilen verileri almak için gönderilen sayfaya boş değişken tanımlanması gerekmektedir.

Veri gönderilen sayfada gönderilen verileri göstermesi için label’lar oluşturuluyor. Oluşturulan label’ lar tanımlanıyor. Tanımlanan label’ ları override fonksiyonu içerisinde oluşturulan değişkenler ile eşitlenerek gönderilen verilerin alımı gerçekleştiriliyor.



# Uyarı Mesajı

Uygulama içerisinde kullanılan içeriklerde eksik veya yanlış kullanımları engellemek amacıyla uyarı mesajı kullanılması tercih ediliyor.

Projede Doldurulması gerekilen alanların boş bırakılması durumunda kullanılıyor.

Veri alışverişinde kullanılan storyboardsegue fonksiyonu içerisinde

İf yapısı ile eğer ks, oda ve yatak adındaki label’ lar boş bırakılırsa (Stepper ile arttırılmaz ise)

Uyari değişkenini oluşturup alertcontroller metodu kullanılıyor. Mesaj başlığı “Hata”,

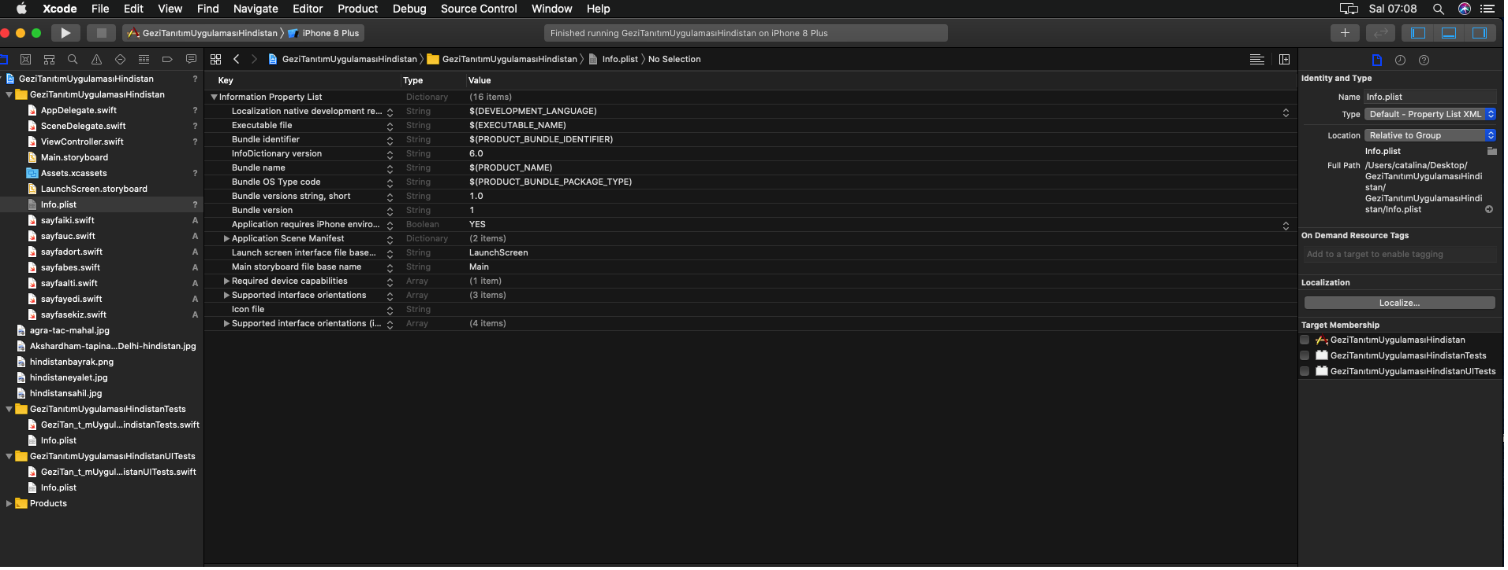
Mesaj: bilgilerinizi girmediniz olarak ayarlanıyor. İptal adında bir değişken daha oluşturuluyor. Bu değişkende ise AlertAction ile iptal tuşuna basıldığında hatanın kapanması sağlanıyor.

Self.present ile de uyari değişkeni yani Hata mesajı ekranı tanımlanıyor



# İkon

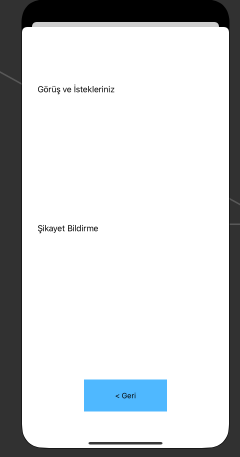
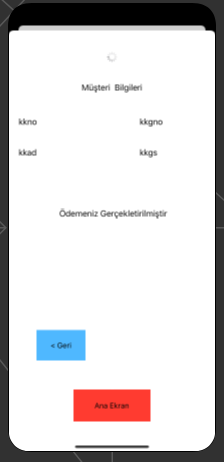
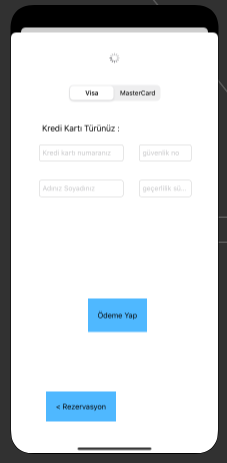
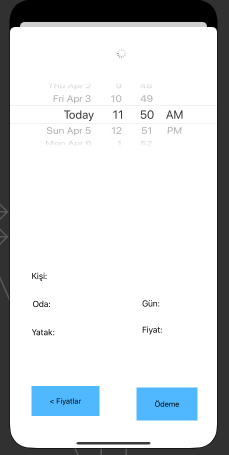
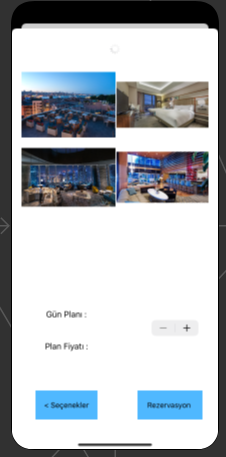
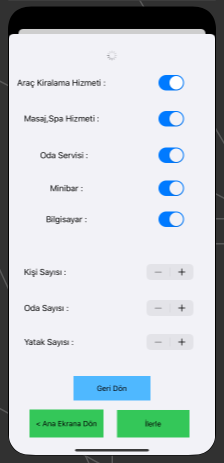
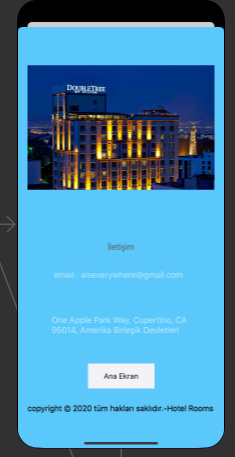
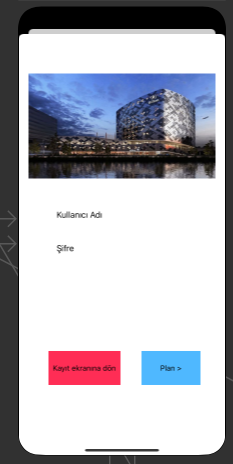
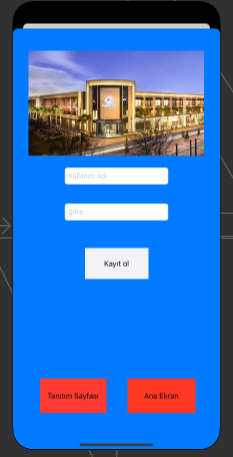
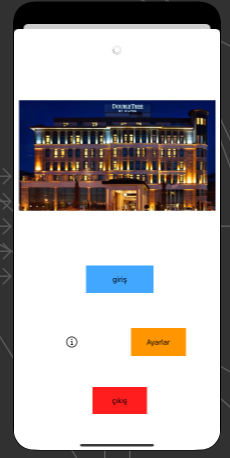
İkon eklemek için öncelikle ikon dosyasının image view de olduğu gibi Xcode programına yükledikten sonra İnfo.plist dosyası açılır ve sağ tık yapılarak Add Row seçeneğini seçtikten sonra Icon File seçeneği seçilir ve ikon dosyasının adı ve uzantısı birebir aynı yazılmalıdır.



# Uygulama konusu

Uygulama otel uygulaması otel tanıtımı, otel rezervasyonu, otel hizmetleri seçimi, ödeme seçenekleri bulunmakta.

## Uygulama ekran görüntüleri



# Proje, Sunum, Rapor, Afiş Github Adresi

https://github.com/BilalMercan/Swift\_Otel\_Proje.git

# Kaynakça

<https://kod5.org/>

<https://www.mobilhanem.com/>

<https://learnappmaking.com/>

<https://stackoverflow.com/>

<https://medium.com/>

<https://www.youtube.com/>

<https://kenanatmaca.com/>

<https://www.it-swarm.net/>

<https://medium.com/>

<https://kod5.org/>

<https://codewithchris.com/>