

İçindekiler

1.Giriş	1
2.Sayfalar.....	2
2.2.App.js.....	2
2.3.Login.js	2
2.4.Register.js.....	2
2.5.Home.js	2
2.5.1. İş Ekle.....	2
2.5.2. İş Sil.....	2
2.5.3. İş Tamamla	3
2.6. Calendar.js.....	3
2.6.1. Notes.js.....	3
2.7. Memory.js	3
2.7.1. AddMemory.js.....	3
3.Ekran Görüntüleri.....	4
Kaynakça	12

1.Giriş

Projede hem ajanda hem de anı defteri olarak kullanılabilecek bir uygulama tasarlanması amaçlanmaktadır. Uygulamada günlük yapılacaklar listesi, sonraki tarihler için takvimli bir not bölümü ve fotoğraflarla beraber fotoğrafa ait anıların tutulabileceği bir sayfa bulunmaktadır.

İlk olarak App.js içerisinde StackNavigator ve TabNavigator oluşturulmuştur. StackNavigator sayfalar arasında geçiş yapılabilmesi için kullanılırken TabNavigator ise uygulamanın alt kısımlarındaki, üç ana sayfa (Home, Calendar, Memory) arasındaki geçişleri sağlamak için kullanılmaktadır. screens klasörü içerisinde bulunan tüm sayfalar StackNav.Navigator içerisine tanımlanmıştır. Firebase ile olan ana bağlantı da yine App.js içerisinde gerçekleştirilmektedir.

Uygulama LoadingScreen.js sayfası ile başlamaktadır. Eğer kullanıcı daha önceden giriş yapmış ise doğrudan Ana sayfaya yönlendirilmektedir. Daha önceden login olunmamışsa program kullanıcıyı giriş yapması için Login sayfasına yönlendirmektedir.

2.Sayfalar

2.2.App.js

Uygulamanın çalışması için gerekli dosyadır. Bu dosyada sayfalar arası geçiş yapabilmeyi sağlayan Stack Navigation^[2], Tab Navigation ve Drawer Navigation^[4] bulunmaktadır. Uygulamada aynı zamanda Firebase veritabanı^[4] ve Firebase authentication işlemler, bu dosya üzerinden yapılmaktadır.

2.3.Login.js

Login.js sayfası kullanıcının giriş yapabilmesi için oluşturulmuştur. Kullanıcıdan alınan email ve şifre bilgileri firebase veritabanına gönderilmektedir. Eğer bu email ve şifre ile oluşturulmuş bir kullanıcı var ise bu durumda kullanıcı Home sayfasına yönlendirilmektedir. Eğer kullanıcı hatalı bir giriş yapmış ise ilgili mesajlar Alert componenti ile kullanıcıya bildirmektedir. Eğer kullanıcı şifresini unutmuş ise girmiş olduğu email adresine bir şifre sıfırlama maili gönderilmektedir. Bu mail yardımıyla kullanıcı şifresini değiştirebilmektedir. Eğer kullanıcı daha önce kaydolmamış ise bu durumda Kayıt ol tuşuna basılmasıyla kayıt olma penceresine yönlendirilerek kullanıcının kayıt olması sağlanmaktadır.

2.4.Register.js

Register.js sayfası yeni kullanıcılar oluşturmak için tasarlanmış bir sayfadır. Bu sayfada kullanıcıdan email ve 2 kez şifre bilgisi alınmaktadır. Eğer kullanıcı bu şifre değerlerini farklı girerse ekrana Alert componenti yardımıyla bir uyarı mesajı çıkarılmaktadır. Eğer kullanıcı daha önce giriş yapmış olan bir mail adresi ile giriş yapmayı denerse bu durumda da Alert componenti yardımıyla uyarı mesajı gösterilmektedir. Kullanıcının yapabileceği bazı hatalar için önlemler alınmıştır.

2.5.Home.js

Home.js sayfası kullanıcının o günün içinde yapacakları işleri ekranda göstermektedir. Ekranın sağ aşağı köşesinde bir tane Action button^[6] bulunmaktadır. Kullanıcı bu butona basarak istediği zaman iş ekleyebilmektedir. İsterse İş silebilmekte veya yapmış olduğu işi tamamlayabilmektedir. İlgili ekran [Ekran Görüntüsü 2](#) ve [Ekran Görüntüsü 3](#) 'de gösterilmektedir.

2.5.1. İş Ekle

Kullanıcı ekranın sağ aşağı köşesindeki Action button a tıkladıktan sonra 3 tane button kullanıcıya gösterilmektedir. Bunlardan birisi İş ekle butonudur. Kullanıcı bu butona tıklayarak bulunduğu güne iş ekleyebilmektedir. Eklemiş olduğu iş kullanıcının ajandasında güncellenmektedir. [Ekran Görüntüsü 1](#) 'de ilgili ekran gösterilmektedir.

2.5.2. İş Sil

Kullanıcı ekranın sağ aşağı köşesindeki Action button a tıkladıktan sonra gösterilen buttonlardan biri de İş sil butonudur. Kullanıcı bu butona basarak istediği bir işi silebilmektedir. Bu yapabilmek için butona tıkladıktan sonra silinmesi istenen işe tıklanması yeterlidir. Kullanıcının yanlış bir işe tıklamasının önlenmesi için ekrana Alert componenti yardımıyla bir uyarı mesajı gösterilmektedir. Eğer kullanıcı bu işi silmek istiyor ise açılan Dialogda Evet tuşuna basarak seçmiş olduğu işi silebilmektedir.

2.5.3. İş Tamamla

Kullanıcı ekranın sağ aşağı köşesindeki Action button a tıkladıktan sonra gösterilen son button da İş tamamla butonudur. Kullanıcının bu butona tıkladıktan sonra tamamlamak istediği işin üzerine tıklaması gerekmektedir. Tıklandıktan sonra ilgili iş alanının renginin değiştirilmesiyle o işin bitmiş olduğunu kullanıcıya bildirilmektedir.

2.6. Calendar.js

Kullanıcı ekranın alt köşesindeki Tab navigationu kullanarak Home, Calendar ve Memory sayfaları arasında geçiş yapabilmektedir. Geçiş yapılabilen sayfalardan biri de Calendar.js sayfasıdır. Bu sayfada kullanıcının ekranında bir ajanda gösterilmektedir. Bu ajanda da kullanıcı istediği takdirde kendi için not veya bildirim ekleyebilmektedir. Kullanıcı istediği güne tıklayarak o gün için not ekleyebilir. Bunu yapabilmesi için kullanıcı Notes.js sayfasına yönlendirilmektedir. [Ekran Görüntüsü 4](#) 'te takvim ekranı gösterilmektedir.

2.6.1. Notes.js

Bu sayfada kullanıcının ajandada seçilen güne veri eklemesi için tasarlandı. Kullanıcı ilgili alanları doldurduktan sonra KAYDET butonuna tıklayarak o gün için not kaydedebilmektedir. Eğer içerik boş ise kullanıcının ekranında Alert componenti yardımıyla uyarı metni gösterilmektedir.

2.7. Memory.js

Kullanıcı ekranın alt köşesindeki Tab navigationu kullanarak Home Calendar ve Memory sayfaları arasında geçiş yapabilmektedir. Geçiş yapılabilen sayfalardan biri de Memory.js sayfasıdır. Bu sayfa kullanıcının anı defteri gibi tasarlanmıştır. Bu sayfada kullanıcı bir fotoğraf ekleyip^[5] o fotoğrafın altına yazı yazarak anı oluşturabilir. Bu anıları oluşturmak için kullanıcının ekranın sağ aşağı köşesindeki Action button a tıklaması gerekmektedir. Butona tıklandıktan sonra kullanıcı Anı ekle butonunu görebileceklerdir. Kullanıcı Anı ekle butonunu tıkladığında AddMemory.js sayfasına yönlendirilmektedir. Anı ekleme ile ilgili ekran görüntüsü [Ekran Görüntüsü 5](#) ,[Ekran Görüntüsü 6](#) ve [Ekran Görüntüsü 7](#) 'de verilmiştir.

2.7.1. AddMemory.js

Bu sayfada kullanıcı kendisi için bir anı oluşturabilir. Bu sayfada bir tane Image iki tane TextInput ve bir tane de TouchableOpacity bulunmaktadır. Kullanıcı ilgili alanları doldurduktan sonra Kaydet butonuna basarak Oluşturmuş olduğu anıyı kaydedebilir. Kullanıcının kaydetmiş olduğu veriler memory sayfasında görüntülenmektedir.

3.Ekran Görüntüleri



Ekran Görüntüsü 1 - İş notu ekleme ekranı

21:01



100 %

Your Journal

Çıkış Yap

Bugün Yapılacak İşler

1. Ekmek Al

2. Okula Git

3. Pizza Ye

4. Ödevi Bitir



Ana Sayfa



Takvim



Anılar

Ekran Görüntüsü 2 - Günlük işlerin listelendiği ekran

21:01



100 %

Your Journal

Çıkış Yap

Bugün Yapılacak İşler

1. Ekmek Al

2. Okula Git

3. Pizza Ye

4. Ödevi Bitir

Yeni İş



İş Sil



Tamamla



Ana Sayfa



Takvim



Anılar

Ekran Görüntüsü 3 - İş ekleme silme düzenleme işlemlerinin yapıldığı menü

Sun

Mon

Tue

Wed

Thu

Fri

Sat

29

30

1.

2.

3.

4.

5 •

1

Tue

Ali

Hasan

2

Wed

Selam

3

Thu

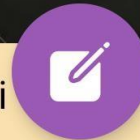
Uyu

4

Fri

Yemek Ye





21:12



100 %



Anı Ekle



Dumanın Doğum Günü

Bugün Duman tam bir yaşında.

Kaydet

Kaynakça

- [1] - Reactnavigation. (tarih yok). *Drawer navigation*. reactnavigation.org: <https://reactnavigation.org/docs/drawer-based-navigation/> adresinden alındı
- [2] - Reactnavigation. (tarih yok). *createStackNavigator*. reactnavigation.org: <https://reactnavigation.org/docs/stack-navigator/> adresinden alındı
- [3] - Nativebase. (tarih yok). *Introduction*. nativebase.io: <https://docs.nativebase.io/> adresinden alındı
- [4] - Expo. (tarih yok). *Using Firebase*. expo.io: <https://docs.expo.io/guides/using-firebase/> adresinden alındı
- [5] - Aditya Sharma. (tarih yok). *How to pick images from Camera & Gallery in React Native app*. enappd.com: <https://enappd.com/blog/pick-images-from-camera-gallery-in-react-nativeapp/78/> adresinden alındı
- [6] - Sibeliu. (tarih yok). *react-native-action-button*. github.com: <https://github.com/mastermoo/react-native-action-button> adresinden alındı