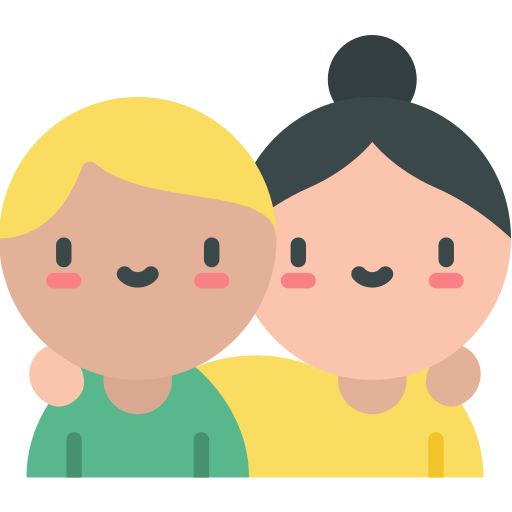
KARADENİZ TEKNİK ÜNİVERSİTESİ YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

MOBİL UYGULAMA GELİŞTİRME RAPORU

Raporu Hazırlayanlar:

404688 - Fatmanur Erbaş

404720 - Fatma Yazıcı



Bu proje bir mesajlaşma uygulamasıdır. Bu uygulama kayıt ekranı, giriş ekranı, ayarlar ekranı, sohbet ekranı, grup sohbet ekranı içermektedir. Bu uygulama kullanıcılara çoklu giriş seçeneği sunar (Google, facebook, telefon ile giriş yapma). Kullanıcılar sohbeti hem tekli hem de çoklu olarak kullanabilmektedirler. Kullanıcılar sohbet ekranının sağ üstünde yer alan menü seçeneğinden ayarlar, grup sohbeti ve çıkış yap seçeneklerinden istediğini seçerek uygulayabilir. Aynı zamanda kullanıcılar ayarlar sekmesinden profil fotoğraflarını, kullanıcı adlarını ve durumlarını değiştirebilmektedirler.

Projemizde ilk olarak kayıt ekranını tasarladık. Projemize uygulamamızın iconunu ekledik. Daha sonra kullanıcılarımızın giriş yapabilmesi için kullanıcı adı , email , şifre textlerini ekledik. Kullanıcılar istedikleri kayıt seçeneği ile uygulamamıza kayıt olurlar. Bu seçenekler Google, facebook ve telefon ile giriş yap seçenekleridir. Eğer kullanıcının hesabı varsa kullanıcı giriş ekranına yönlendirilir. Giriş ekranına gittikten sonra kullanıcı hangi seçenek ile giriş yapmak istiyorsa bilgilerini girip giriş yapar. Eğer kullanıcı kayıt olmadıysa kayıt ekranına geri dönebilir. Sohbet ekranında sohbet edilen kişiler görülür. İstediğimiz kişi ile özel sohbet edebiliriz ya da menü seçeneğinden grup sohbetini seçerek grup mesajlaşması yapabiliriz.

FIREBASE

İlk önce projeye firebase eklemekle işe başladık.Tools sekmesinden authentication ve realtime database eklemeleri yaptık.Projenin ilerleyen kısımlarında cloud storage for firebase eklemesi yaptık.Bu eklemeler ile kullanıcı bilgilerimiz ve mesaj bilgileri database’de kayıt altına alınmış oldu.

KAYIT OLMA EKRANI

activity\_sign\_up.xml

Öncelikle kayıt ekranımızı tasarlamakla başladık.Kayıt ekranımızın arka plan resmini ve iconumuzu belirledik.Sonrasında bu resimlerimizi drawable klasörümüze kaydettik.İconumuzun boyutunu ve konumunu ayarladık.Veri girilecek sütunlarımızı ekledik.Bu sütunlar kullanıcı adı, email, şifre.Bu sütunların konumlarını,boyutlarını,yazı renklerini ayarladık.Verilerimizi girdikten sonra bizi yönlendirecek olan butonlarımızı ekledik.Birinci giriş yap butonu.Giriş yap butonumuzun rengini ve boyutunu ayarladık.Diğer butonlarımızda kullanıcılara giriş için farklı seçenekler sunduk.İlki Google aracılığı ile giriş.Diğeri ise facebook aracılığı ile giriş.Eğer ki kullanıcının hesabı varsa kullanıcı giriş yapma ekranına yönlendirilir.Bunu “hesabın var mı” seçeneği ile sağladık.

SignUpActivity.java

Tasarımımızdaki butonlarımıza işlevsellik kazandırmak için binding ile verilerimizi bağladık.Firebase bağlantısı için değişkenler tanımladık.inflater ile xml dosyamızı uygulama kaynak kodumuzda kullanabilmek için bir araç sağladık.Hesap oluşturulurken kullanıcılara bilgilendirmek için progress dialog nesnesini kullandık.Ardından giriş yapma butonuna tıkladığımızda yapılmasını istediklerimiz kodladık.Eğer kullanıcı adı,email ve şifre boş değilse kullanıcıya progress dialog nesnesi ile kayıt olunduğu bilgisi verilir.Yapılan iş başarılıysa kullanıcı kullanıcı firebase’e kaydedilir.Ardından Toast ile giriş başarılı mesajı verilir.Başarılı değilse hata döndürüyor.Eğer girilen bilgiler eksik veya hatalı ise “kimlik bilgilerinizi giriniz” mesajı verir.Eğer kullanıcı zaten kayıtlı ise “hesabın var mı” seçeneği ile giriş sayfasına intent ile yönlendirilir.

GİRİŞ EKRANI

activity\_sign\_in.xml

Kayıt ekranımızı tasarladıktan sonra giriş ekranımızı tasarladık.Giriş ekranımızın aynı arka plan resmini ve iconumuzu ekledik.İconumuzun boyutunu ve konumunu ayarladık.Veri girilecek sütunlarımızı ekledik.Bu sütunlar email ve şifre.Bu sütunların konumlarını,boyutlarını,yazı renklerini ayarladık.Verilerimizi girdikten sonra bizi yönlendirecek olan butonlarımızı ekledik.Birinci giriş yap butonu.Giriş yap butonumuzun rengini ve boyutunu ayarladık.Diğer butonlarımızda kullanıcılara giriş için farklı seçenekler sunduk.İlki Google aracılığı ile giriş.Diğeri ise facebook aracılığı ile giriş.Eğer ki kullanıcının hesabı yoksa kullanıcı kayıt olma ekranına yönlendirilir.Bunu “kaydolmak için tıklayın” seçeneği ile sağladık.

SignInActivity.java

Tasarımımızdaki butonlarımıza işlevsellik kazandırmak için binding ile verilerimizi bağladık.Firebase bağlantısı için değişkenler tanımladık.inflater ile xml dosyamızı uygulama kaynak kodumuzda kullanabilmek için bir araç sağladık.Öncelikle Google ile giriş yapmayı kodladık. Eğer email ve şifre boş değilse kullanıcıya progress dialog nesnesi ile giriş yapıldığı bilgisi verilir. Yapılan iş başarılıysa kullanıcı kullanıcı firebase’e kaydedilir ve kullanıcı intent kullanılarak ana ekrana yönlendirilir.Başarılı değilse toast ile uyarı mesajı verilir. Eğer girilen bilgiler eksik veya hatalı ise “kimlik bilgilerinizi giriniz” mesajı verir. Eğer ki kullanıcının hesabı yoksa kullanıcı kayıt olma ekranına yönlendirilir.Bunu “kaydolmak için tıklayın” seçeneği ile sağladık.Google ile giriş yapma başarılı olursa veriler database’e kaydedilir ve Toast ile bilgi mesajı verilir.Eğer giriş başarılı değilse hata döndürülür.

SOHBET EKRANI

Sample\_show\_user.xml

Tasarımımıza sohbet edilen kişilerin gösterildiği ekranla devam ettik.Ekranın içerisinde profil fotoğrafı kısmı ekledik.Profil fotoğrafı iconunu drawable klasörüne ekledik.Bu iconun rengini ve boyutunu belirledik.Sohbet ettiğimiz kişinin isminı ve son mesajını içeren kısmı profil fotoğrafının yanına ekledik.

SOHBETLER , DURUM VE ARAMALAR EKRANLARI

fragment\_status.xml – fragment\_calls.xml

Toolbar tasarımımızda sohbetler kısmının yanına durum ve aramalar kısmı ekledik.

MODELS

MessageModel.java

Model uygulamamaızın verilerini tutar. MessageModel’de bu veriler id , mesaj , mesaj id ve zamandır. Bu verilerin constructor, getter ve setter’larını yazdık.

Users.java

Users’da tuttuğumuz veriler profil fotoğrafı, kullanıcı adı , mail, şifre, kullanıcı id, son mesaj ve durumdur. Bu verilerin constructor, getter ve setter’larını yazdık.

ADAPTERS

chatAdapter.java

Öncelikle sample\_sender.xml dosyamızı inflate metoduyla View nesnesi oluşturarak birbirine bağladık ve bunu geri dönderdik. Eğer kullanıcı alıcı ise inflater metodu kullanarak bu metodu geri dönderdik. Butona uzun süre basılı tuttuğumuzda uyarı mesajı verildikten sonra mesaj silinir. Mesajı gönderdiğimizde date nesnesi ile gönderilme tarihi belirlenir. Aynı şekilde saatte belirlenir. Bu işlemler hem gönderici hem de alıcı için ayrı ayrı yapılır. RecieverViewHolder ve SenderViewHolder ile receiverMsg ve receiverTime özelliklerine gerçek değerlerini atıyoruz.

FragmentsAdapter.java

Bu sınıfta fragmentlarımızın dönüş değerleri yazılır. Chats , Status , Calls sırasıyla dönüş değerleri 0, 1 ve 2 dir. Default olarak Chats döndürülür.

UsersAdapter.java

Son mesajı database ayarlamak için metod içerir.

FRAGMENTS

ChatFragment.java

Verileri UsersAdapter aracılığı ile databasedeki users kısmında gösterdik. Google ile giriş yapan kullanıcılar listelenir fakat farklı şekilde kaydedilir.

CallsFragment.java

StatusFragment.java

SOHBET İÇ EKRANI

activity\_chat\_detail.xml

Sohbet içeriğine girdiğimizde toolbarımızın ilk iconu geri tuşu.Geri tuşu iconumuzu drawable klasörümüze ekledikten sonra boyutunu belirledik.Ardından profil resmini drawable klasörüne kaydettikten sonra boyutunu ve diğer özelliklerini belirledik.Sonrasında sohbet ettiğimiz kişinin kullanıcı adını profil resminin yanına ekledik ve bu textview’in özelliklerini belirledik.Kullanıcı adını belirledikten sonra toolbarımızın sağ kısmına sırasıyla telefon iconu, görüntülü konuşma iconu ve menü iconu ekledik.

ChatDetailActivity.java

Tasarımımızdaki butonlarımıza işlevsellik kazandırmak için binding ile verilerimizi bağladık.Firebase bağlantısı için değişkenler tanımladık.inflater ile xml dosyamızı uygulama kaynak kodumuzda kullanabilmek için bir araç sağladık.Profil resimlerini yüklemek ve işlemek için Picasso kütüphanesini kullandık.Intent kullanarak geri ok tuşuna işlevsellik kazandırdık.MessageModel.java classından bir array oluşturduk.Database’e chats kısmı oluşturduk.Bu kısımda iletilen mesajlar görüntülenir. Sonrasında adaptera verinin değiştiğini bildiriyoruz. Ardından send tuşuna işlevsellik kazandırdık. Öncelikle MessageModel sınıfından bir nesne üretiyoruz ve bu nesneye senderId , message parametreleri gönderiyoruz. Şuanın tarihini model nesnesine gönderiyoruz. Yapılan mesajlaşmalar realtime database içerisinde chats bloğu altında senderRoom ve receiverRoom childları altına kayıt edilir.

GÖNDERİCİ VE ALICI EKRANI

sample\_show\_user.xml

Sohbet ettiğimiz kişilerin göründüğü ekrana girdiğimizde en solda drawable klasörümüze eklediğimiz avatar gözükür. Bu avatarın genişlik ve boy gibi özelliklerini belirledik. Avatarımızın yanına bir LinearLayout ekledik. Bu layout içerisindeki textviewler mesajlaştığımız kişinin kullanıcı adı ve son mesajını gösterir. Daha sonra bu textviewlerin özelliklerini belirledik.

sample\_reciever.xml

Bu xml dosyası gelen mesajların tasarımını içerir. İçerisinde gelen mesaj ve bu mesajın gönderildiği saat yer almaktadır.

sample\_sender.xml

Bu xml dosyası gönderdiğimiz mesajların tasarımını içerir. İçerisinde gönderilen mesaj ve bu mesajın gönderildiği saat yer almaktadır.

MENU

menu.xml

Uygulamamızın toolbarının sağ kısmında menü yer almaktadır. Menu içerisinde sırasıyla ayarlar, grup sohbeti, çıkış yap sekmeleri yer almaktadır.

GROUP CHAT EKRANI

activity\_group\_chat.xml

Öncelikle mesaj gönderim kısmını tasarladık. Bu kısımda mesaj yazma texti ve mesaj gönderme butonu yer almaktadır. Bu kısımların özelliklerini belirledik. Daha sonra toolbarımızı tasarladık. Toolbar içerisinde geri ok tuşu, profil fotoğrafı, kullanıcı adı, arama iconu, görüntülü arama iconu ve menü iconu yer almaktadır. Bu kısımların özelliklerini belirledik.

GroupChatActivity.java

Geri ok tuşuna işlevsellik kazandırdık. Toolbarda görünen kullanıcı adı kısmına group chat yazdırdık. Adapter ile messsageModelsi çağırıyoruz. Snapshot aracılığı ile messageModelden aldığımız mesajları database ekler ve adaptera verilerin değiştiğini haber verir. Ardından send tuşuna işlevsellik kazandırdık. Öncelikle MessageModel sınıfından bir nesne üretiyoruz ve bu nesneye senderId , message parametreleri gönderiyoruz. Şuanın tarihini model nesnesine gönderiyoruz. Yapılan mesajlaşmalar realtime database içerisinde group chat bloğu altına kaydedilir ve eğer mesaj gönderimi başarılı ise toast ile bilgi mesajı verilir.

AYARLAR EKRANI

activity\_settings.xml

Tasarımımızın üst sağ köşesinde geri ok tuşu bulunur. Altında LinearLayout yer alır. Bu layoutun içerisinde profil fotoğrafı iconu yer alır. Bu iconun sağ alt köşesinde artı iconu bulunur. Layoutun devamında kullanıcı adı ve hakkımızda textleri bulunur ve layoutun en son kısmında kaydet butonu bulunur. Bu textlerin ve butonların özelliklerini belirleriz. Tasarımın alt kısmında farklı bir LinearLayout vardır. Bu layotun içerisinde beş adet text vardır. Bunlar gizlilik politikası , hakkımızda, arkadaşlarını davet et, bildirim, yardım. Bu textlere özelliklerini verdik.

SettingsActivity.java

Öncelikle geri ok tuşuna işlevsellik kazandırdık. Ardından kaydet butonuna basıldığında yapılacakları kodladık. Eğer kullanıcı adı ve hakkımda kısmı boş değilse HashMap ile kullanıcı adı ve durum arasında anahtar değer ilişkisi kurduk. Ardından databasede güncelleme yaptık ve toast ile bilgilendirme mesajı verdik. Eğer kullanıcı adı ve durum boş ise toast ile “lütfen kullanıcı adınızı ve durumunuzu giriniz” mesajı verdik. Artı işaretine işlevsellik kazandırıyoruz. Bu işaret profil fotoğrafı eklememizi sağlar. Database’e eklediğimiz profil fotoğrafını kaydediyoruz.

ANA EKRAN

activity\_main.xml

Ana ekranımızın özelliklerini belirledik.

MainActivity.java

Üç noktalı menüye inflate ile işlevsellik kazandırdık. Menü içeriklerine basıldığında ekrana gelecek sınıfları belirledik. Settings’e basıldığında intent ile ayarlar sekmesine yönlendirdik. GroupChat’e basıldığında intent ile grup sohbeti sekmesine yönlendirdik. Logout’a basıldığında intent ile giriş sekmesine yönlendirdik ve kullanıcı bu sayede çıkış yapmış oldu.