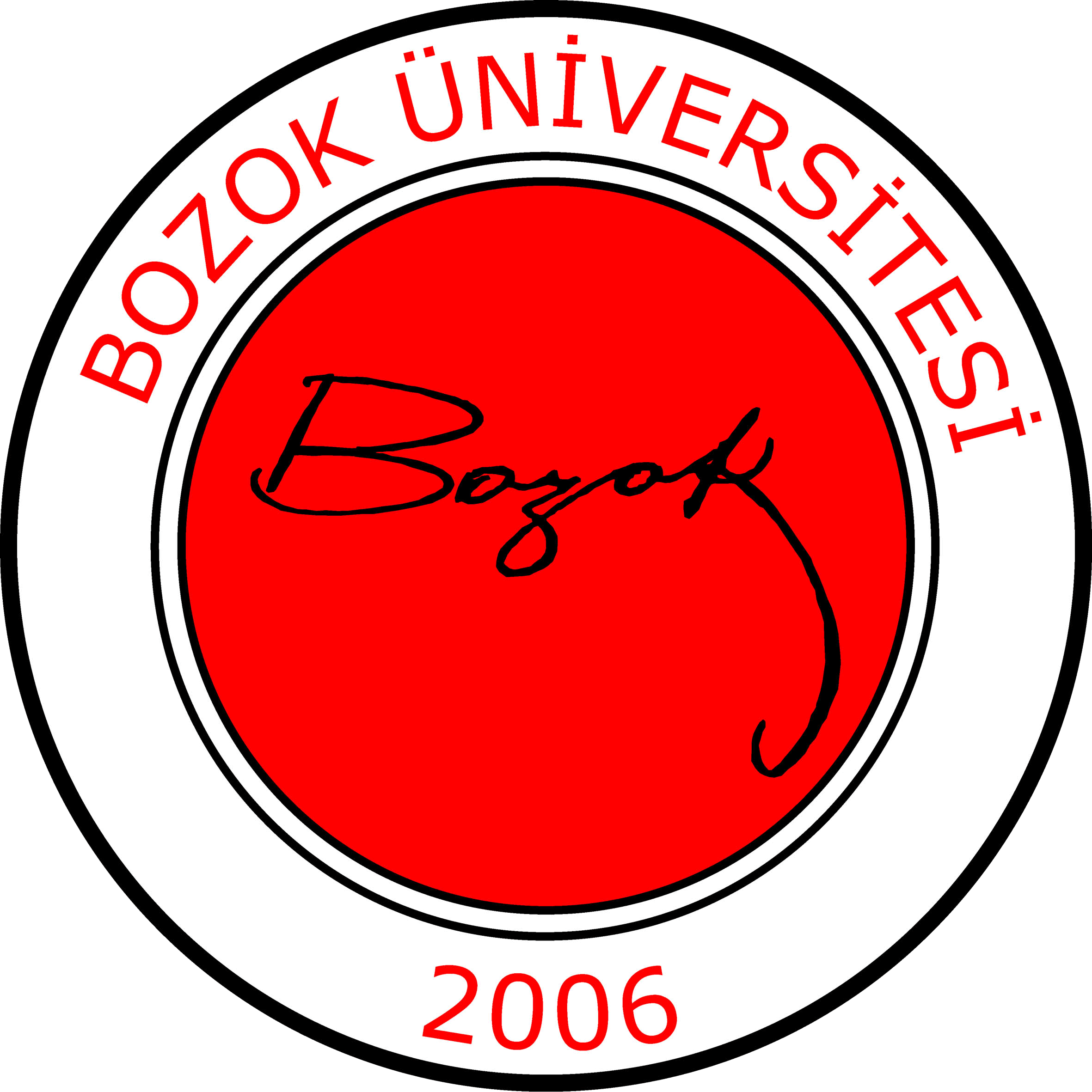
**BOZOK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

****

**JAVA İLE ANDROID TABANLI İNGİLİZCE SOHBET UYGULAMASI “PURPLE BEE”**

**BİTİRME ÖDEVİ**

**Hazırlayanlar**

**Üzeyir İlhan ÖZTÜRK**

**Hasan KURT**

**2019-2020 BAHAR DÖNEMİ**

**BOZOK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**JAVA İLE ANDROID TABANLI İNGİLİZCE SOHBET UYGULAMASI “PURPLE BEE”**

**BİTİRME ÖDEVİ**

**Hazırlayanlar**

**Üzeyir İlhan ÖZTÜRK**

**Hasan KURT**

**Danışman**

**Doç. Dr. Mehmet BAKIR**

**2019-2020 BAHAR DÖNEMİ**

# TEŞEKKÜRLER

Bu projede bizden desteğini esirgemeyen Sayın Doç. Dr. Mehmet BAKIR’ a teşekkürlerimizi ve sevgimizi sunuyoruz. Gösterdikleri anlayış ve desteklerinden dolayı ailelerimize müteşekkiriz.

İçindekiler

[TEŞEKKÜRLER I](#_Toc44332539)

[ŞEKİL LİSTESİ III](#_Toc44332540)

[KISALTMALAR IV](#_Toc44332541)

[ÖZET V](#_Toc44332542)

[1. GİRİŞ 1](#_Toc44332543)

[1.1. PROJENİN AMACI 2](#_Toc44332544)

[1.2. YAPILACAK ÇALIŞMALAR 2](#_Toc44332545)

[1.4. NEDEN ANDROİD STUDİO? 5](#_Toc44332546)

[1.5. JAVA NEDİR? 5](#_Toc44332547)

[1.6. JAVA TARİHÇESİ VE ÖZELLİKLERİ 6](#_Toc44332548)

[1.7. JAVA İLE NELER YAPILABİLİR? 9](#_Toc44332549)

[1.8. FİREBASE NEDİR? 9](#_Toc44332550)

[1.8.1. Google Firebase Ne İşe Yarar? 10](#_Toc44332551)

[1.8.2. Google Firebase Tarihi 10](#_Toc44332552)

[1.8.3. Google Firebase’in En Temel Özellikleri 11](#_Toc44332553)

[1.8.4. Firebase Ürünleri ve Hizmet Türleri 12](#_Toc44332554)

[1.8.5. Google Firebase Nasıl Kullanılır? 12](#_Toc44332555)

[1.9. ONE SİGNAL NEDİR? 13](#_Toc44332556)

[2. LİTARATÜR TARAMASI 15](#_Toc44332557)

[3. KULLANILAN ARAÇ VE YÖNTEMLER 17](#_Toc44332558)

[3.1. Android Geliştirme Platformu 17](#_Toc44332559)

[3.1.1. Android studio’ nun artı yönleri 18](#_Toc44332560)

[3.1.2. Android studio’ nun eksi yönleri 19](#_Toc44332561)

[3.1.3. Android Studio ile Yapılabilecek Projeler 19](#_Toc44332562)

[3.1.4. Farklı Platformlar 19](#_Toc44332563)

[3.2. Firebase 20](#_Toc44332564)

[3.2.1. Firebase Veritabanı Yapısı 20](#_Toc44332565)

[3.2.2 Push ID 22](#_Toc44332566)

[3.2.3 Child Parent Kavramı 23](#_Toc44332567)

[3.3. SDK Nedir? 24](#_Toc44332568)

[3.3.1. SDK vs API 25](#_Toc44332569)

[3.3.3. iOS SDK 26](#_Toc44332570)

[3.3.4. Android SDK 26](#_Toc44332571)

[4. TARTIŞMA VE SONUÇ 27](#_Toc44332572)

[5. KAYNAKÇA 34](#_Toc44332573)

[6. EKLER 35](#_Toc44332574)

[ÖZGEÇMİŞ 145](#_Toc44332575)

# ŞEKİL LİSTESİ

Şekil 3.1 Android Studio Arayüzü………………………………………………………….….18

Şekil 4.1 E-mail ile Programa Giriş Ekranı…………………………………………………....27

Şekil 4.2 Cep Telefonu İle Giriş………………………………………………………………..28

Şekil 4.3 Seçenekler Kısmı ve İçindeki Araçlar………………………………………………..28

Şekil 4.4 Oluşturulan Gruplar…………………………………………………………………29

Şekil 4.5 Arkadaş Arama………………………………………………………………………30

Şekil 4.6 Arkadaşların Çevrimiçi Görünmesi………………………………………………..30

**Şekil 4.7 OneSignal Ekranı……………………………………………………………………..31**

Şekil 4.8 Rekabetçi İngilizce Testi Çözme……………………………………………………..32

Şekil 4.9 İngilizce Quizleri………………………………………………………………………33

# KISALTMALAR

APK Android Package Kit

# ÖZET

Bu uygulama insanlarla İngilizce konuşma ve sosyalleşme platformu olup günümüz teknolojisinde faydalı bir içerik ve kullanım kolaylığı ile İngilizce öğrenmek isteyen insanlara yardımcı olacaktır. Ulusal dil olarak seçilen İngilizce tüm dünya tarafından öğrenilmesi gerekir biz bu süreci hızlandırmak istiyoruz. Farklı ülkelerden insanların sadece İngilizce dilini kullanarak hiç tanımadığı insanlarla sohbet etmesini sağlıyoruz. Öncelikle herkesin görebildiği bir grupta konuşarak arkadaş olabilirler dilerse özel sohbetten sadece ikisinin olduğu bir ortamda İngilizcelerini daha iyi geliştirebilmekteler. Ayrıca uygulamamızda testlerde bulunmaktadır kademeli olarak giderek kullanıcının bilgi seviyesinin ölçülmesi gerçekleştirilmiştir.

**SUMMARY**

This application is a platform for speaking and socializing with people in English and will help people who want to learn English with useful content and ease of use in today's technology. English, which is chosen as the national language, must be learned by the whole world. We enable people from different countries to chat with people they have never met using only the English language. First of all, if they want to be friends by talking in a group that everyone can see, they can improve their English better in an environment where only two of them have private conversation. Also, there are tests in our application, gradually increasing the measurement of the user's knowledge level.

# 1. GİRİŞ

İngilizce, ilk olarak 5. yüzyılda İngiltere'de ortaya çıkmış, modern zamanlarda ise küresel bir lingua franca haline gelmiş Batı Cermen dili. İngilizce günümüzde Birleşik Krallık, Amerika Birleşik Devletleri, Karayipler, Avustralya, İrlanda, Kanada ve Yeni Zelanda'da yaşayan kişilerin çoğunluğunun anadilini oluşturmaktadır. İkinci dil ve resmî dil olarak dünya genelinde, özellikle İngiliz Milletler Topluluğu üyeleri ve çok sayıda uluslararası örgüt tarafından kullanılmaktadır.

Modern İngilizce, anadili farklı olan insanların konuştuğu ilk "küresel ortak dil" olarak tanımlanmaktadır. İngilizce iletişim, bilim, ticaret, havacılık, eğlence, radyo ve diplomasi alanlarında egemen uluslararası yardımcı dildir. Dilin Britanya Adaları'ndan öteye yayılarak benimsenmesinde, Britanya İmparatorluğu ile Amerika Birleşik Devletleri'nin II. Dünya Savaşı sonrasında artan ekonomik ve kültürel etkisi rol oynamıştır.

İngilizceyi öğrenmenin en iyi yolu konuşmaktır çekinmeden karşıdaki insan ile sohbet alışverişi yapabilmek gerekir. Bu uygulamada da ister anonim olun ister olmayın rahatça farklı ülkelerden farklı kültürlerde ki insanlarla rahatça konuşabileceksiniz.

# 1.1. PROJENİN AMACI

Bu projede arkadaşlarımızla kendimizi özel hissedeceğimiz bir platform oluşturarak dünya genelinde bu platformda sosyalleşme ve iletişim uygulaması yaptık. Projemiz Google Play ortamında canlı olarak yayındadır. Verilerin daha fazla kişinin görebileceği bir veri tabanını kendi Firebase veri tabanımızı oluşturarak kişisel verilerin kontrolünü bir nebze olsun kendi elimizde olmasını sağladık. Kullanıcılara testler sunarak onların bir rekabet içinde olmasını sağladık böylece sıkılmadan sınav ortamı oluşturulacak hemde yeni insanlarla tanışıp sohbet etme imkanı sağlayacağız.

# 1.2. YAPILACAK ÇALIŞMALAR

Android studio ortamında java dili ile bir sohbet uygulaması yazdık ve yapılan çalışmada insanların İngilizce dillerini geliştirmesini amaçladık. Girilen veriler firebase veritabanına kaydedilecek ve istenilen zamanda çekilebilecek. Öncelikle firebase hesabı oluşturulması gerekiyor oluşturulan hesapla tüm veri akışını görebileceğiz. Bu çalışmayı insanlara sunmak için 25$ harcanarak Google Play hesabı açılacaktır.

**1.3.ANDROİD İŞLETİM SİSTEMİ**

Android; Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilmekte olan, Linux tabanlı özgür ve ücretsiz bir işletim sistemidir. Sistem açık kaynak kodlu olsa da, kodlarının ufak ama çok önemli bir kısmı Google tarafından kapalı tutulmaktadır. Google tarafından ücretsiz olmasının sebebi, sistemin daha hızlı ve çabuk gelişmesi, birçok popüler marka tarafından kullanılması ve bu sayede reklamlarını daha fazla kişiye ulaşmasını sağlamaktır. Google, Android sistemi üzerinde çalışan Google Play marketteki oyun ve uygulamalar üzerinde aldığı reklamları yayınlayarak para kazanmaktadır. Android'in desteklenen uygulama uzantısı ".apk"dır.

Android, aygıtların fonksiyonelliğini genişleten uygulamalar yazan geniş bir geliştirici grubuna sahiptir. Android için hali hazırda 1 milyondan fazla uygulama bulunmaktadır. Google Play Store ise, Android işletim sistemi uygulamalarının çeşitli sitelerden indirilebilmesinin yanı sıra, Google tarafından işletilen kurumsal uygulama mağazasıdır. Geliştiriciler, ilk olarak aygıtı, Google'ın Java kütüphanesi aracılığıyla kontrol ederek Java dilinde yazmışlardır.

Open Handset Alliance, 5 Kasım 2007'de Android'i kurduğunu duyurmuştur ve ardından 34 adet donanım, yazılım ve telekom şirketi, mobil cihazlar için telif hakkı olmayan bir işletim sisteminin teknolojinin gelişimi için yararlı olduğu konusunda hemfikir olmuşlardır.

Android, Linux çekirdeği üzerine inşa edilmiş bir mobil işletim sistemidir. Bu sistem ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu Java-uyumlu kütüphaneleri içine alan uygulama iskeleti üzerinden çalışmaktadır. Android, derlenmiş Java kodunu çalıştırmak için dinamik çevirmeli Android Runtime (ART) kullanır ve cihazların fonksiyonelliğini artıran uygulamaların geliştirilmesi için çalışan geniş bir programcı-geliştirici çevresine sahiptir. Google aynı zamanda işletim sistemindeki hataları bulan kullanıcıları para ödülü ile ödüllendirmektedir.**[1]**

**1.4. ANDROİD STUDİO TARİHÇESİ?**

Android Studio yetkili oldu entegre geliştirme ortamı (IDE) hakkında 'ın Android işletim sistemine üzerine inşa JetBrains ' IntelliJ IDEA yazılım ve için özel olarak tasarlanmış Android geliştirme. Windows , macOS ve Linux tabanlı işletim sistemlerinde indirilebiliR. O yerine geçer Eclipse Android Development Tools yerel Android uygulama geliştirme için birincil IDE .

Android Studio, 16 Mayıs 2013 tarihinde Google I / O konferansında açıklandı.. Mayıs 2013'te 0.1 sürümünden başlayarak erken erişim ön izleme aşamasındaydı, daha sonra Haziran 2014'te yayınlanan 0.8 sürümünden başlayarak beta aşamasına girdi. İlk kararlı yapı, Aralık 2014'te 1.0 sürümünden başlayarak piyasaya sürüldü.

7 Mayıs 2019'da Kotlin , Java'yı Google'ın Android uygulama geliştirme için tercih ettiği dil olarak değiştirdi.. Java hala C ++ gibi desteklenmektedir.

Mevcut kararlı sürümde aşağıdaki özellikler sağlanır:

Gradle tabanlı yapı desteği

Android'e özel yeniden düzenleme ve hızlı düzeltmeler

Performans, kullanılabilirlik, sürüm uyumluluğu ve diğer sorunları yakalamak için tüy bırakmayan araçlar

ProGuard entegrasyonu ve uygulama imzalama özellikleri

Ortak Android tasarımları ve bileşenleri oluşturmak için şablon tabanlı sihirbazlar

Kullanıcıların UI bileşenlerini sürükleyip bırakmasına olanak tanıyan zengin bir düzen düzenleyici , birden çok ekran yapılandırmasında düzenleri önizleme seçeneği

Android Wear uygulamaları oluşturma desteği

Firebase Cloud Messaging (Daha önce 'Google Cloud Messaging') ve Google App Engine ile entegrasyon sağlayan Google Cloud Platform için yerleşik destek

Android stüdyosundaki uygulamaları çalıştırmak ve hatalarını ayıklamak için Android Sanal Aygıtı (Emulator).

Android Studio, IntelliJ (ve CLion ) ile aynı programlama dillerini destekler; örneğin Java , C ++ ve daha fazlası Go gibi uzantılarla; ve Android Studio 3.0 veya üstü, Kotlin ve "tüm Java 7 dil özelliklerini ve platform sürümüne göre değişen Java 8 dil özelliklerinin bir alt kümesini" destekler. Dış projeleri backport bazı Java 9 özellikleri. IntelliJ, Android Studio'nun yerleşik olarak yayınlanmış tüm Java sürümlerini ve Java 12'yi desteklediğini belirtirken, Android Studio'nun Java 12'ye kadar Java sürümlerini hangi düzeyde desteklediği açık değildir (belgeler kısmi Java 8 desteğinden bahseder). Android'de en azından Java 12'ye kadar bazı yeni dil özellikleri kullanılabilir**[2]**

# 1.4. NEDEN ANDROİD STUDİO?

Yazdığımız chat uygulaması telefonlarda kullanılacağı için ve sadece android telefonlara hitap edeceği için andriod studio kullandık.

# 1.5. JAVA NEDİR?

Java ™ platformu bilgisayar ağının varlığı da göz önüne alınarak uygulamaların/programların farklı işletim sistemleri üzerinde çalıştırılabilmesi düşüncesiyle geliştirilmiş bir teknolojidir. Java ™ platformu hem programlama dili, hem de bir ortam olarak düşünülebilir. Programlama dili olarak, açık kodlu, nesneye yönelik (object-oriented), güvenli, sağlam, İnternet için elverişli bir teknolojidir denilebilir

Java programlama dili ve bu dile bağlı alt teknolojiler, VB™ veya Borland Delphi™ gibi sadece belirli bir firma tarafından geliştirilmiş ürünler değillerdir. Java ve ona bağlı alt teknolojiler, Sun MicroSystems tarafından tanımlanmış belirtimlerden (specifications) oluşmaktadır. Bu belirtimlere sadık kalan her yazılım firması Java Sanal Makinası, kısaca JVM (Java Virtual Machine), veya Java programlama diline bağlı alt teknolojiler yazabilir (örneğin Application Server - Uygulama Sunucusu). Eğer bu belirtimlere sadık kalınmayıp standart dışı bir JVM veya Java programlama diline bağlı alt teknolojiler yazılmaya kalkışılırsa hukuki bir suç işlenmiş olur.

Peki belirtim (specifications) ne demektir? Sun MicroSystems, JVM veya Java programlama diline bağlı alt teknolojiler yazmak için belirli kurallar koymuştur; bu kurallar topluluğuna “belirtimler” denir. Örneğin çöp toplama sistemi (garbage collector).

Çöp toplama sistemi daha önceden oluşturulmuş, ancak kullanılmayan ve bellekte boşu boşuna yer işgal eden nesneleri belirleyerek otomatik olarak siler. Böylece Java programcısı “acaba oluşturduğum nesneyi bellekten silmiş miydim?” sorusunu sormaktan kurtulurlar, ki bu soru C++ programlama dillinde uygulama yazan kişilerin kendilerine sıkça sorması gereken bir sorudur. Şimdi bir yazılım firması hayal edelim, adının ABC yazılım firması olduğunu varsayalım. Bu firma, eğer bir JVM yazmak istiyorsa, bu çöp toplama sistemini, oluşturdukları JVM’in içerisine yerleştirmeleri gereklidir. Çünkü Sun MicroSystems’ın belirtimlerinde, çöp toplama sistemi koşuldur! Eğer ABC firması üşenip de çöp toplama sistemini, oluşturdukları JVM’in içerisine yerleştirmezse hukuki bir suç işlemiş olur.

Şu anda en yaygın kullanılan JVM’ler, IBM ve Sun Microsystems’ın üretilmiş olan JVM’lerdir; ayrıca, HP, Apple gibi bir çok firmanın üretmiş oldukları JVM’ler de bulunmaktadır.

# 1.6. JAVA TARİHÇESİ VE ÖZELLİKLERİ

Şu anda dünyadaki en popüler programlama dillerinden biri olan Java, 1990’lı yıllarda SUN bilgisayar şirketince akıllı elektronik ev araçları projesi çerçevesinde mikrodalga fırınları, buzdolapları, televizyonlar, uzaktan kumanda gibi cihazların birbiriyle haberleşmesini sağlamayı amaçlayan bir proje içerisinde geliştirilmeye başlandı. Orijinal ilk adı Oak’dı ve bu ad dilin yaratıcıları James Gosling, Patrick Naughton, Chis Wartdh, Ed Frank ve Mike Sheridan tarafından konulmuştu. Daha sonra bu isimde başka bir programlama dili olduğu anlaşılınca o anda bir kahvehanede kahve içen programlama gurubu tarafından kahve markasından esinlenerek Java olarak değiştirildi.

Akıllı elektronik ev araçları pazarı SUN gurubunun tahminlerinden çok daha yavaş bir gelişme gösteriyordu. Bu yüzden Java dili projesi ticari bir geliştirme projesi olarak büyük olasılıkla iptal edilecekti. 1993 Yılında "World Wide Web" büyük bir atılım göstererek bütün dünyaya yayılmaya başladı. Javanın Dinamik Web sayfaları hazırlamadaki büyük potansiyelini gören SUN şirketi projeyi bu tarafa yönlendirdi.

Mayıs 1995 de SUN javayı büyük bir konferansta tanıttı. Program iş dünyası tarafından derhal büyük bir ilgiyle karşılandı. Java modern bilgisayar dünyasının ses, grafik işlem, haberleşme gibi ihtiyaçlarına cevap verebilen ve ticari gayeler için hazırlanan bir Programlama dili olarak daha önceki bilgisayar dillerinin hiç birinin kapsayamadığı özellikleri içermekteydi. Bunun yanı sıra dil komut yapısı olarak C++ diline çok yakın olması da öğrenilmesini kolaylaştırıyordu. SUN Javayı "World Wide Web" de kullanmak isteyen herkese ücretsiz olarak sundu. Java internette yayınlanmasının ardından çok büyük bir patlama yaşadı. 1997 ye gelindiğinde dünyadaki bütün bilgisayar okullarında temel bilgisayar dili olarak gösterilmeye başlandı. Dünyada şu anda hala en çok kullanılan bilgisayar dili olan C++ dilinin, yapılan hataları tam olarak denetlememesi programın çalışma hızını arttırma yönünden iyi bir özellik olsa da profesyonel programcılar dışında kullanılmasını sınırlandırıcı bir etki yapıyordu. Java ise bütün hataları bildiren yapısı ve modern bilgisayarın bütün fonksiyonlarına ulaşabilen kütüphaneleriyle programcıların çok daha kolaylıkla öğrenebileceği bir dildir. Burada hemen şunu da belirtelim. C dili hızlı çalışma amacı birinci planda tutularak yaratılmış bir dildir. Javada ise emniyet ilk planda yer almıştır. Hız açısından düşünüldüğünde java, C (ve C++) diliyle rekabet edemez. Zaten program derleyicisi de C++ dilinde yazılmıştır.

Javanın diğer önemli bir temel özelliği Nesne kökenli (object oriented) bir dil olmasıdır. Nesne kökenli diller, nesneleri gerçek dünyadakine daha benzer bir yapıda tanımlayarak anlaşılmalarını kolaylaştırırlar. Nesneleri gerçek dünyadaki gibi masa, sandalye, bilgisayar gibi tanımlayarak programlamak insan beyninin anlaması açısından çok daha kolaydır. Bundan önceki tam nesnel kökenli programlama dillerinden hiçbiri çok yaygınlık kazanmamıştı. Bu tür dillere Smalltalk,u örnek verebiliriz. C++ nesnel kökenli programlama yapabilen bir dildir. Fakat yapısal (structural) bir programlama dili olan C dilinin bir uzantısı olarak geliştirildiğinden tam anlamıyla nesnel kökenli bir dil olduğu söylenemez.

Java dilini geleceğin dili yapan diğer bir özelliği de çok kullanımlı (multi-tasking) ve paralel kullanımlı (multi-treading) bir dil olmasıdır. Çok kullanımlılık birden fazla işlemin aynı anda yapılabilmesinin tanımıdır. Paralel kullanımlılık ise birden fazla programın aynı anda hafızayı beraber kullanılabilmesidir. Örnek olarak Word ve Excel programlarının Windows NT ortamında aynı anda kullanılmasıdır. ADA gibi bazı eski Program dillerinde çok kullanımlılık programlanabiliyordu. Paralel kullanım olanaklarını sunan ilk bilgisayar dili ise javadır. Paralel kullanım paralel programlama kavramından ayrıdır ve karıştırılmamalıdır. Paralel programlanmada birden fazla Bilgi işlem Ünitesine (CPU ) ayrı programlar veya bir programın ayrı parçaları gönderilir. Paralel kullanımda ise bir CPU nun kullanım zamanı küçük parçalara ayrılarak değişik Program veya Program parçacıkları bu zaman paketçiklerini paylaşarak kullanırlar.

Javayı önemli bir Program dili haline getiren en önemli özelliği ise kullanılan bilgisayardan bağımsız olmasıdır. Javada yazılan bir Program Unix, Machintosh, Windows 95 veya Windows NT veya herhangi bir 32 bit makinada hiç değiştirilmeden kullanılabilir. Java programlarının grafikleri "Wold Wide Web" sayfalarının belge işlem sistemi olan html (hypertext markup language) ile aktarılır. Bu yüzden html ve java programlarını birlikte kullanmak ve java programlarını gerçek zamanda www sayfalarında göstermek mümkündür

Javanın HTML diliyle kullanılabilmesi, bazı kişilerde htmlin bir parçası olduğu gibi bir kavram gelişmesine yol açmiştir. Programlamaya yeni başlayanlar sık sık html ile javayı birbirine karıştırırlar veya javanın HTML'nin bir uzantısı olduğunu düşünürler. Bu temelde oldukça yanlış bir varsayımdır. HTML, internet belgelerini birbirine bağlamak amacıyla geliştirilmiş bir belge işlem sistemidir ve bir programlama dili değildir. HTML ile Java’nın tek gerçek ilgisi HTML deki applet komutudur. Bu komut yardımıyla java dilinde yazılmış programların sonuçları HTML ortamına dinamik olarak aktarılabilir.

# 1.7. JAVA İLE NELER YAPILABİLİR?

Java diliyle projeler diğer programlama dillerine göre daha kolay, sağlıklı ve esnek şekilde yapılması mümkün olur. Java diliyle gerçekleştirilebilecek uygulamalardan bazıları şunlardır:

· GUI (Grafiksel Kullanıcı Arayüzü) uygulamaları, Applet’ler

· Veri tabanına erişimle ilgili uygulamalar

· Web tabanlı uygulamalar

· Dağınık bileşenler (Distributed components) (örneğin EJB, RMI, CORBA).

· Cep telefonları, Smart kartlar için uygulamalar.[3]

# 1.8. FİREBASE NEDİR?

Firebase, Google’un kullanıcılara armağan ettiği ücretsiz platformlardan birisi durumunda. Özellikle pek çok yeni özelliğin eklendiği Firebase, şu anda internet kullanan kişilerin hemen hemen tüm ihtiyaçlarını karşılıyor. Örneğin herhangi bir platform üzerinde uygulama geliştirdiniz. Bu tip durumlarda kontrol panellerine ya da veri depolama alanlarına ihtiyaç duyabilirsiniz. Bugün ortaya çıkarılan uygulamalar hemen her türlü cihazda ya da işletim sisteminde sorunsuz şekilde yer almak istiyor. Yönetim panellerinde genelde kayıt, oturum, veri analizi, uygulama testleri gibi işlemleri yapmak mümkün hale gelir.

İşte dünyanın en büyük arama motoru olan Google tam olarak bu ihtiyaçtan hareketle Firebase’i piyasaya sürdü. Firebase sisteminin ücretsiz sunulması ve yüksek performans ve verimlilik sunması, onu mucizevi kılıyor. Kullanıcılar Firebase teknolojisinden şimdilik oldukça memnun görünüyor. Normalde uygulama yönetimini ya da bildirim gönderme süreçlerini ayrı sunucular sayesinde yapmak mümkün hale geliyordu. Firebase teknolojisinde ise sunucu taraflı kod yazmaya gerek yoktur.

## 1.8.1. Google Firebase Ne İşe Yarar?

Google Firebase yalnızca mobil uygulamalarda işe yarayan bir teknoloji değildir; aynı zamanda bilgisayar gibi cihazlarda çalışan programlar için de yararlanılabilecek bir platformdur. Bugün üretilen projelerin kendilerine özgü yazılım dilleri olsa bile tüm pazara hitap edecek bir yapıya sahip olmak zorundadırlar. Arka plan verilerinin tutulması ve uygulamaların her türlü platformda kullanılabilmesi, Firebase’in sağladığı faydalar arasındadır. Bu anlamda pek çok uygulamanın ortak sorunlarına karşı çözüm üretebildiğini söyleyebiliriz.

Bugün yazılımcılar genelde uygulama süreçlerinde hangi backendt teknolojisinden yararlanmaları gerektiğini tartışırlar. Aynı zamanda hangi veri tabanından yararlanmaları gerektiği de kritik bir sorudur. Firebase sayesinde yazılım uzmanlarının işi bir hayli kolaylaştı, çünkü artık Firebase ile beraber daha hızlı geliştirme yapabiliyorlar. Firebase ile beraber yazılımcılar özellikle mobil uygulama konusunda daha büyük bir öz güvene sahip oldular ve bu durum niceliksel anlamda üretime doğrudan yansıdı.

## 1.8.2. Google Firebase Tarihi

Firebase platformu 2011 senesinde iki startup tarafından kuruldu. O dönemki ilk adı ise Envolve idi. İlk çıktığında asıl amaç, uygulama üreten yazılımcıların zahmetsiz şekilde chat uygulamasını ekleyebilmeleriydi. Bu uygulamayı API şeklinde tanımlayabiliriz. Chat uygulaması konusunda kolaylık sağlaması pek çok kişinin ilgisini çekti. Buna rağmen tam anlamıyla bu noktayla tatmin olmadılar. Çünkü asıl ihtiyaç olan şey chat uygulaması değildi, bunu fark etmiş oldular. Anlık verilerin senkronizasyonu chat uygulamasından çok daha acil bir uygulama olarak göründü. Dolayısıyla iki startup bu yönde gelişim sağlamaya karar verdiler. Anlık verilerin senkronuna örnek olarak multi oyunculu oyunlarda oyunlara ait verilerinin senkronize edilmesi gibi düşünülebilir. Bu nedenle hemen bir sene sonra Firebase isminde yepyeni bir firma kurdular ve burada ağırlıklı olarak data senkronu üzerine hizmet verdiler.

2014 senesinde Google, bu hizmeti iki girişimciden satın aldı. Asıl büyüme hikayesi de tam olarak bu noktada başladı. Google sürekli olarak yeni ürünler geliştirdi ve sistemi büyüttü. Bugün gelinen noktada yazılımla ilgilenen ve daha çok uygulama tasarlayan kişiler için olmazsa olmaz teknolojilerden biri haline gelmiştir. Sizin de kullanıcı girişleri bulunan ya da bazı verilerin depolandığı bir platformunuz varsa Google Firebase sisteminden dilediğiniz ölçüde yararlanabilirsiniz. Firebase bugün hem küçük hem de orta ölçekli pek çok uygulamanın ihtiyaçlarına yanıt verebiliyor. Bu nedenle de en başından bu yana tamamen ücretsiz olarak sunuldu.

## 1.8.3. ****Google Firebase’in En Temel Özellikleri****

Günümüzde uygulama yapanların kurtarıcısı olan Google Firebase, kendine has pek çok özelliğe sahiptir. Ayrıca bu özelliklerini her geçen sene daha da geliştirdiği için ayakta kalmayı başarılı şekilde sürdürüyor.

Eşzamanlı veritabanı

Kullanıcı girişlerinde yetkilendirme

Depolama

Makine öğrenme kiti

Performans test ortamı

Platform arası ortak biçimde uygulama analizleri

Ortak fonksiyon

Reklam materyalleri

Bildirim gönderimi

Genel uygulama yönetimi

Google Firebase ile alakalı daha kapsamlı özellikleri incelemek adına Firebase Documentation isimli sayfayı ziyaret edebilirsiniz. Bu arada Google Firebase ile alakalı olarak genelde bazı konularda ciddi bir bilgi yanlışlığı söz konusu olabiliyor. Örneğin Google Firebase sanıldığı gibi bir framework değildir. Aynı zamanda yapısal bir veritabanına sahip değildir. Hele hele bir yazılım dili hiç değildir. Diğer yandan API yaratmak adına ideal bir servis olduğunu da söyleyemeyiz.

## 1.8.4. Firebase Ürünleri ve Hizmet Türleri

Google Firebase uygulama geliştirme noktasında çok sayıda ürünü ve hizmeti bir arada sunuyor. Bu hizmetlerden ve ürünlerden bazıları şu şekilde:

***Realtime Database*:** Veri tutmaya yarayan büyük bir veritabanı uygulaması diyebiliriz. Temel amacı anlık olarak verileri senkronize etmektir.

***Authentication*:** Bu serviste işlem yapmak adına giriş yapmanız şarttır.

***Test Lab***: Ürettiğiniz ya da tasarım aşamasında olduğunuz mobil uygulamalarla ilgili size ön bilgiler sunar. Uygulamada denediğiniz her şeyi hızlıca test edebilirsiniz.

***Crashlytic***: Anlık olarak loglama yapmanızı sağlar. Ayrıca hatalarınızı anında deşifre eder ve bu sayede güncellemeler yapmanıza olanak sağlar.

***Firestore****:* Anlık senkronizasyonun daha ölçeklenebilir halidir. Realtime Database ile sık sık karşılaştırılır. Günümüzde beta sürümü olarak hizmet verir.

## 1.8.5. Google Firebase Nasıl Kullanılır?

Başlangıç düzeyinde dahi kolay bir şekilde Google Firebase’in olanaklarından yararlanmak mümkündür. Google Firebase kullanmak adına Google’a ait pek çok serviste olduğu gibi Google kimliğine sahip olmak şarttır. Örneğin Google hesabınıza aşt olan bir mail adresi bu noktada işinizi görecektir. Daha sonra Google Firebase’e ait resmi web sayfasına gidip doğrudan kontrol paneline ulaşabilirsiniz. Console bağlantısını tıklamak ya da Get Started bölümünü tıklamak, ilerlemenizi sağlayacaktır. Bu noktadan sonra siz de özgür bir şekilde yeni projelerinizi oluşturmaya başlayabilirsiniz. Gerek Android gerek İOS gerekse de bilgisayar uygulamalarında çeşitli özellikleri aktif biçime getirebilirsiniz. Burada kolayca görebileceğiniz Proje Oluştur butonunu kullanarak söz konusu sürece dahil olabilirsiniz.

Proje oluşturma aşamasından sonra çeşitli işletim sistemlerine entegre olması sağlanmalıdır. Zaten proje sayfalarında bu konuyla alakalı brieflere detaylı olarak ulaşabilirsiniz. İOS entegrasyonu, Android uygulama entegrasyonu ve web uygulaması entegrasyonu ayrı ayrı entegre edilir. Bugün pek çok şirket Google Firebase sayesinde önemli ölçüde yol kat edilebiliyor. Google Firebase aslında bir tür bulut alt yapı teknolojisidir. Projenizi oluştururken backend işlemleriyle kafanızı karıştırmaya gerek kalmayacaktır. Tüm veriler sistem içinde birer nesne olarak yer tutar.[4]

## 1.9. ONE SİGNAL NEDİR?

**One Signal** hem mobil cihazlar ve tabletler için hem de bilgisayarlar için özel olarak tasarlanmış basit ve güvenilir ücretsiz (bireysel kullanım için) bir **Push Notification** (bildirim alma) sistemidir. Uyumlu olduğu plaformlar  iOS, Android, Chrome, Unity 3D, Amazon, Windows Phone, Phonegap, Marmalade, Corona şeklinde uzamaktadır.

Websitenizde yaptığınız kampanyaları, indirimleri veya yeni içeriklerinizi daha önceden sitenize gelmiş kişilere bildirmek için kullanılabilecek en güvenilir sistemlerden bir tanesi push servisleridir. Bildirim sistemlerinin de en iyilerinden bir tanesi OneSignal’dir. Çünkü neredeyse her platformda sorunsuz bir şekilde çalışmaktadır.

Mesaj şablonları hazırlayarak daha sonra kullanmak istediğiniz bildirim mesajlarını kaydedebilir, zamanlama planlayarak istediğiniz saat ve dakikada bildiriminizi kullanıcılarınıza gönderebilirsiniz. Gönderdiğiniz bildirimlerin analizlerini onesignal üzerinden detaylı bir şekilde yapabilirsiniz.

Segmentlere ayırarak kullanıcılarınıza özel bildirimler yollayabilirsiniz. Örneğin sitenizi 5 kereden az ziyaret etmiş kullanıcılara farklı bir bildirim, sitenizi aktif olarak ziyaret eden kullanıcılara daha farklı bir bildirim gönderebilirsiniz. Segmentlerde yapılacak özelleştirmeleri kendiniz ayarlayabildiğiniz için çok kullanışlı bir sistemdir.

Ayrıca pek bilinmeyen bir özellik olan **“intelligent delivery”** özelliği sayesinde diğer push servislerinden farkını ortaya koyuyor. Türkçe karşılığı akıllı teslimat olan bu özellik sayesinde kullanıcıların aktif olduğu saatlere göre özel bildirim hedefleri koyuluyor. Kullanıcının aktif olduğu saatler Onesignal’in kendi analizleri sonucu çıkıyor sürekli gelişim aşamasında olduğu için gün geçtikçe daha da akıllı bir hal alıyor bu özellik.

Biraz daha detaylı anlatmak gerekirse; örneğin bir kullanıcı gündüz saat 11 gibi telefonda veya bilgisayarda aktif oluyor ve internette gezme eğilimi bu saatlerde olarak görülüyor, o kişiye saat 11 olmadan hemen önce bir bildirim gönderiyor. Başka bir kullanıcı ise akşam 7’den sonra müsait olabiliyor ve telefonla ilgilenebiliyor o kişiye ise saat 7’de aynı bildirimi gönderiyor. Böylelikle istenmeyen bildirimlerin önüne geçilmiş oluyor. Sitenize nitelikli trafik çekebilmeniz için kullanışlı bir özellik olduğu için bildirimlerinizde akıllı teslimatı öneriyoruz.[5]

# 2. LİTARATÜR TARAMASI

**DALDAL(1998), “*Yerel bilgisayar ağlarında görüntü iletimine ilişkin yazılım tasarımı”***

Bu proje kapsamında Windows 3.11 işletim sistemli kişisel bilgisayarların bağlı olduğu Novell Netware ağ işletim sistemi kurulu 10 Mbps lik Ethernet kartı kullanılan yerel bilgisayar ağlarında kullanıcılar arasında gerçek zamanlı görüntülü iletişimi sağlayacak bir program yazıldı. Görüntülü iletişim konusunda yapılan çalışmalar ve çoklu ortam iletişimine getirilen standartlar araştırıldı. Çoklu ortam iletişimi dünyanın her yerindeki kullanıcılarla ses, görüntü ve veri alışverişine, gerçek zaman görüntülü bağlantılara olanak sağlar. Bu konuda yapılan çalışmalar görüntü bilgisinin kurulu ağlara ek yük getirmeden ortak bir dile sahip olarak nasıl daha hızlı iletileceği konusunda yoğunlaşmaktadır. Bu çalışmada geliştirilen uygulama programına görüntü sağlamak için kişisel bilgisayara paralel porttan bağlanan dijital bir masaüstü kameradan yararlanıldı. İletişimi gerçekleştirecek olan program Windows 3.11 ortamında Visual Basic proglamlama dilinde yazıldı. "Visual Chat" isimli proje yazılımı, standart bir Pc' ye masaüstü kameradan başka ek donanım gerektirmemektedir. Görüşme yapmak isteyen her iki kullanıcı birbirlerinin terminalinin yerel ağdaki adını programın kullanıcı arabiriminde yazar ve bağlantı sağlanınca kameralarını çalıştırırlar. Görüntü bilgisi iletişim halindeki diğer bilgisayara Windows işletim sistemininin DDE (Dynamic Data Exchange) teknolojisinin açtığı kanal yardımıyla kare kare gönderilir. Çok görevli Windows ortamında aktif pencere her saniye değiştirilerek kameranın kendi programı ve "Visual Chat" programının beraber çalışması sağlanır. Şu anda kullanılan 10 Mbps lik Ethernet kartlı yerel ağlarda görüntü iletişimine çözüm getirmek amacıyla geliştirilmiş olan bu program veri trafiğinin çok yoğun olduğu ortamlarda yavaş kalacaktır. Kullanıcı sayısı az ve veri iletişim trafiği yoğun olmayan ortamlarda kullanıldığında ise malî açıdan ekonomik olacaktır.

**ÇALIŞKAN(2019),** ***“Android akıllı telefon ve tabletlerde kulanıcı aktivitelerini izleme”*** Teknolojinin ilerlemesiyle birlikte mobil cihaz kullanımında önemli bir artışın olduğu görülmektedir. Literatür taramaları sonucunda elde edilen bilgi ve sonuçlar bu yargının kesinliğini göstermektedir. Yapılan tez çalışmasında, Android işletim sistemine sahip akıllı telefon ve tablet üzerinde cihaz kullanıcılarının aktivitelerini izleyen bir uygulama geliştirilmiştir. Android işletim sisteminin açık kaynak kodlu olması, kullanılan cihaz ve kullanıcı sayısının yüksek oranlara sahip olması gibi sebeplerden dolayı, tez çalışmasında Android tercih edilmiştir. Geliştirilen uygulama sayesinde, kullanıcı güvenliği amacıyla örneğin ebeveynler çocuk takibini, şirketler ve işletmeler ise personel denetimini sağlayabilecekler, kullanıcı aktivitelerini vb. izleyebileceklerdir. Geliştirilen uygulama, yönetici ve izlenecek uygulama olmak üzere 2 ayrı akışta çalışacaktır. Android Studio, Visual Studio, SQL Server Management Studio ve Firebase teknolojileri kullanılarak uygulama geliştirilmiştir. Yönetici cihazında bulunan uygulama ile izlenen kullanıcının cihazında bulunan uygulama listesi, kullanıcının konumu ve cihazın sistem sürümü gibi cihazın temel bilgileri alınabilmektedir. Ayrıca arama geçmişi, rehberde kayıtlı numaralar, gelen giden SMS takibi, tuş vuruşları (keylogger) vb. izlemelerin yönetici tarafında bulunan Android mobil uygulama ile gözlemlenmesi sağlanmıştır. Tez çalışması kapsamında geliştirilen uygulama etik durumlar dikkate alınarak geliştirilmiş bir tür casus yazılım (spyware) uygulamasıdır. Kaynak araştırmalarında, gerçekleştirilen bu tez çalışmasının amacını destekler nitelikte sonuçlar elde edilmiştir ve öneriler sunulduğu görülmüştür. Araştırmalarda mobil cihaz kullanımının artmasıyla birlikte bazı güvenlik risklerinin de bulunabileceği, bu güvenlik risklerine karşı önlem alınması ve kullanıcı takibinin yapılması gerekliliği ifade edilmektedir. Uygulama hem sanal cihazlarda hem de gerçek cihazlar üzerinde test edilmiştir ve başarıyla çalıştığı görülmüştür.[6]

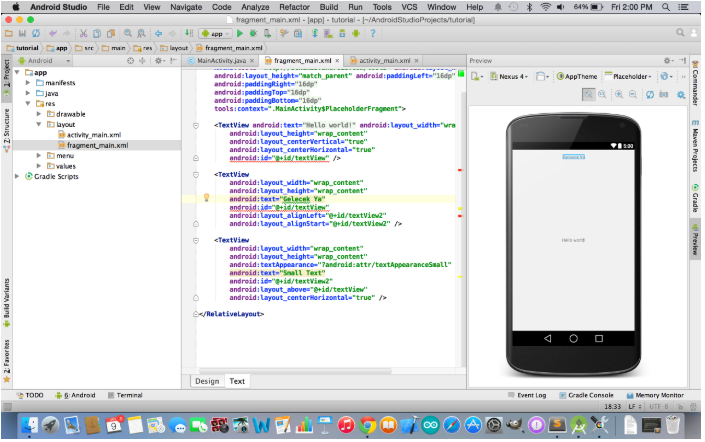
# 3. KULLANILAN ARAÇ VE YÖNTEMLER

## 3.1. Android Geliştirme Platformu

\*Android Studio

\*Java Dili

Android Studio, Android uygulama geliştirme alanında geliştiricilerin işini kolaylaştırmak amacıyla, üst seviye bir çok özelliğie sahip ve Google tarafından önerilen bir programlama aracıdır. **IntelliJ IDEA** tabanlı olarak geliştirilen özel bir geliştirme ortamıdır. En önemli özelliklerinden birisi sanal cihaz (emulator)’lardır.



Şekil 3.1 Android Studio Arayüzü

### 3.1.1. Android studio’ nun artı yönleri

1-Google Cloud için yerleşik destek sağlar.

2-Kendine ait tüm cihaz ve android sürümleri için emülatör bulundurur.

3-Telefon hariç diğer tüm cihazlarda (tablet, televizyon, saat vb.) bir arayüzü bulundurur.

4-Github entegrasyonu ve hazır şablonlar barındırır.

5-UML diyagramı desteği sağlar.

6-Kendi sanal cihazına (emulatör) sahiptir.

7-Vektör imajlar sağlar.

8-Tamamen ücretsizdir.

9-Canlı kod değişikliği desteği sağlar.

### 3.1.2. Android studio’ nun eksi yönleri

1- Kısıtlanmış dil desteği sunar.

2-Diğer IDE’ler ile karşılaştırıldığında RAM kullanımının yüksektir.

3-Hedef işletim sisteminin sadece Android' tir.[androidstudio3.bölüm]

### 3.1.3. Android Studio ile Yapılabilecek Projeler

1. Medikal Uygulamalar

2. Robotik Uygulamalar

3. Mekatronik Uygulamalar

4. Mobil Uygulamalar

5. Giyilebilir Uygulamalar

6. Kablosuz Haberlesme Uygulamaları

7. Algoritmik Uygulamalar

8. RFID Uygulaması

9. Ev Otomasyonu

### 3.1.4. Farklı Platformlar

1.Visual Studio(Xamarin

2.Cordova(PhoneGap)

3.Appcelarator Titanium

4.Qt

5.Eclipse

6.Intellıj IDEA

7.NetBeans

8.Komodo

9.AIDE **[7]**

## 3.2. Firebase

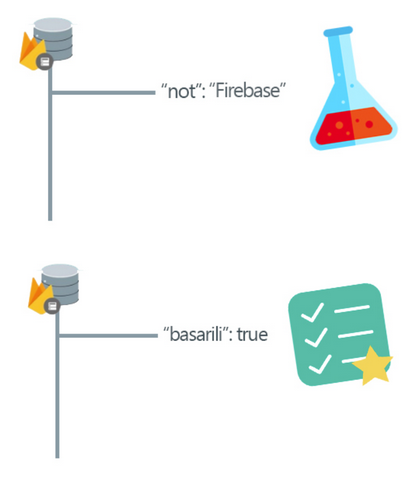
Firebase gerçek zamanlı veri tabanı, bulut tabanlı NoSql(Not Only Sql) bir veri tabanı sistemidir. Hiç bir sql sorgusuna gerek duymadan json parametreleri ile yönetebilirler. Veri depolamanın yanı sıra asenkron çalışması ile veri değişimlerinin anlık olarak takip edilmesine olanak sağlar.

### ****3.2.1. Firebase Veritabanı Yapısı****

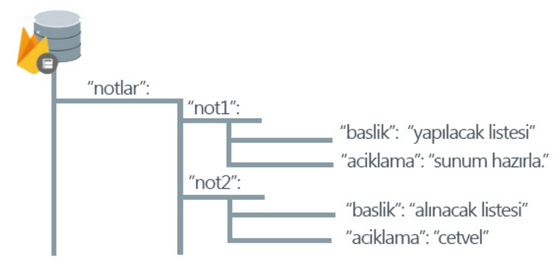
Firebase Realtime Database,bütün verileri JSON(JavaScript Object Notation) objesi olarak tutar.Firebase Realtime yapısını genel olarak bulut tabanlı bir json ağaç yapısı olarak düşünebilirsiniz. Json ağacına veri eklediğinizde  veriniz JSON formatında bir node(düğüm) haline gelir.



Bu düğümler anahtar-değer (key-value) yapısına sahiptir. Kullanıcı ID veya semantik isimler gibi kendi anahtarlarınızı oluşturabilirsiniz. JSON yapısı içerisinde anahtar değerler String tipinde ve eşsizdirler. Anahtarın aldığı değerler **String,Integer,Boolean** gibi çeşitli tiplerde olabilirler.



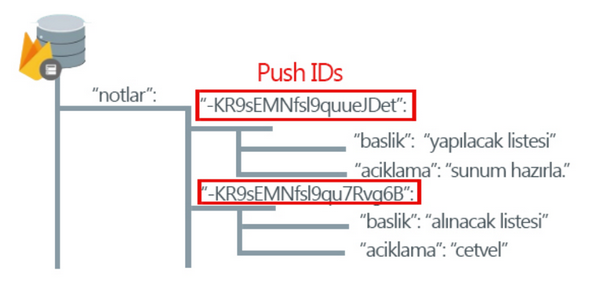
Aşağıdaki Json yapısında **“notlar”** isimli bir ana düğüm ve bu düğüm altında **“not1”, ”not2”** alt düğümleri bulunmaktadır. Bu düğümler ile veriler arasında gezerek arama,listeleme işlemleri gerçekleştirilir.



### 3.2.2 Push ID

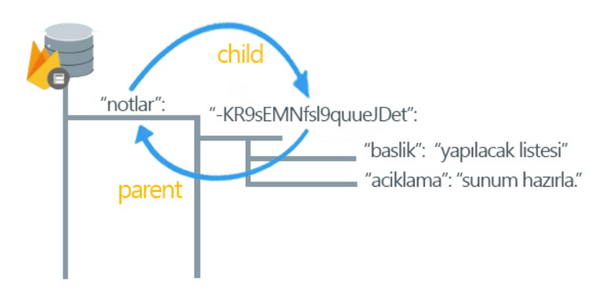
Bu yapıda “not1” ve “not2” isminde iki adet numaralandırılmamış statik düğüm bulunuyor.  
Peki iki kullanıcı eş zamanlı olarak mesaj eklemek isterse bir sonraki mesaj hangi kullanıcı tarafından eklenir?

Bu sorunun gerçekleşmemesi için eşsiz bir not ID oluşturuyoruz. Bu eşsiz ID Firebase üzerinde Push ID olarak ifade edebiliriz.Push ID metodu ile çoklu kullanıcılarda gerçekleşen işlemlerde çakışma yaşamamış oluyoruz.

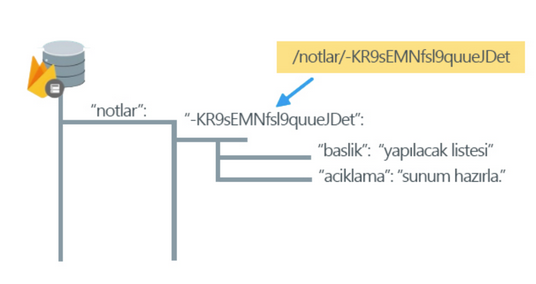


### 3.2.3 Child Parent Kavramı

Json yapınız üzerinde **child** ve **parent** olmak üzere iki ana kavram vardır. Aşağıdaki yapıda “not1” ve “not2” düğümleri  “notlar” düğümünün child düğümleri yani alt düğümleri olarak ifade edilir.Child ifadesi ana düğümün verileri arasında  okuma yazma gibi işlemlerinizi gerçekleştirebilmenize olanak sağlar.



Child düğümüne erişebilmek için  web üzerinde Url formatına benzer bir path yapısı kullanılır. Örneğin “notlar”  düğümüne ‘/notlar/-KR9sEMNfsl9quueJDet’ ile erişebiliriz.



Json formatında ağaç yapısı array tipini desteklemektedir.İşlemlerinizde verileriniz arasında dolaşabilmek için child by child yapısını kullanabilirsiniz. **[8]**

## 3.3. SDK Nedir?

Bir **SDK’yı** Ikea mobilyası olarak düşünün. Ikea mobilyalarını kendi başına monte etme işlemi zordur, ancak bu belirsiz araçlar veya talimatlar olmadan yapmayı hayal edin. İmkansıza yakın, değil mi? Bir şey inşa etmek için parçaların kendisinden daha fazlasına ihtiyacınız var.

**SDK (software development kit)** veya **yazılım geliştirme kiti**, geliştiricilerin belirli işletim sistemleri için uygulamalar oluşturmak için kullandıkları indirilebilir bir araç setidir. Genel olarak, bir SDK bir uygulama içinde belirli bir modül oluşturmak için ihtiyacınız olan her şeyden oluşur ve kütüphaneler, araçlar, örnek kod, ilgili belgeler ve çoğu zaman API’ler içerebilir.

Uygulamanızda oluşturmaya çalıştığınız işlevi veya özelliğe bağlı olarak çeşitli yazılım araçları içeren farklı SDK’lar vardır. Artı, iOS ve Android SDK’lar gibi her platform / işletim sistemi için özelleştirilmiş SDK’lar vardır.

### 3.3.1. SDK vs API

Yukarıda kısaca bahsettiğimiz gibi, bir SDK genellikle bir API veya uygulama programlama arayüzü ile dokümantasyon ve kütüphaneler içerir.

Bir API, yazılımın diğer yazılımlarla iletişim kurmasını sağlar. Karşılıklı anlama için esas olarak komut setlerini farklı yazılım programlarından çeviren uygulamaların etkileşiminden sorumludurlar.

Belgeler, API’yi fiilen entegre etmek için nasıl yapılır kılavuzudur. Kütüphaneler, “arayabileceğiniz” ve kodunuza ekleyebileceğiniz örnek ve programlama dilidir.

**3.3.2 Neden Bir SDK Yapısı Kullanmalısınız?**

Yazılım geliştirme kitleri, uygulamanıza işlevsellik eklemenize yardımcı olur. SDK’lar, uygulamanızın oldukça standart bileşenlerini daha hızlı ve daha kolay oluşturmanızı sağlar. Bir konum özelliği veya veri depolama için tekerleği yeniden icat etmeniz gerekmez. SDK’lar, geliştiricilerin her kod satırını sıfırdan yazmadan uygulamalar oluşturmasına olanak tanır.

Evet, teknik olarak bir geliştirici SDK’ları birlikte bırakmaya karar verebilir, ancak bu çok sıkıcı ve zaman alan bir kodlama süreci anlamına gelir. Bir dakikalığına Ikea’ya geri dönelim. SDK kullanmadan bir uygulama için kod yazmak, kitaplık oluşturmak için kendi ağaçlarınızı kesmek, artı aradaki her adımı kullanmak gibi olur.

Genellikle, yazılım geliştirme kitleri, oturum açma, konum servisleri, mobil ödemeler ve daha fazlası gibi basit işlevler için kullanılır. Ancak sanal ve artırılmış gerçeklik gibi, daha karmaşık uygulama özellikleri için de bazı şeyler var.

### 3.3.3. iOS SDK

Facebook giriş SDK’si, kullanıcıların Facebook hesap bilgilerinizle uygulamanıza giriş yapmalarını sağlar. Kendi giriş özelliğini oluşturmak yerine, sadece Facebook’un girişini kullanabilirsin.

Kullanıcılarınıza her iki seçeneği de sunarsanız (uygulamanızla bir hesap oluşturmak veya Facebook ile giriş yapmak için), büyük olasılıkla Facebook girişini seçecektir, çünkü bilgilerin tekrar girilmesi son derece elverişli değildir. Kullanıcılardan, uygulamanıza Facebook’a erişmesi için yetki vermeleri istenir.

Bu SDK, iOS paketi için Facebook giriş bilgilerinin nasıl uygulanacağına ilişkin adım adım talimatları içerir, ayrıca geliştiriciye yardımcı olabilecek diğer dokümantasyon kaynaklarını da sağlar.

### 3.3.4. Android SDK

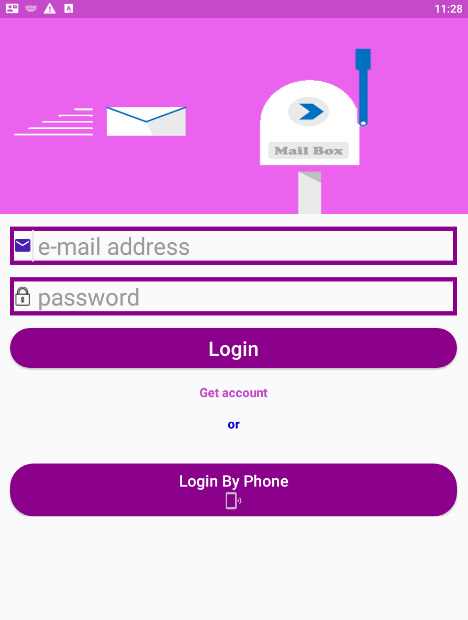
Geliştiriciler, bu yazılım geliştirme kiti ile uygulamalarında Google Haritalar’ın gücünü kullanabilirler.

Android için Google Haritalar SDK’sı, uygulamanıza nasıl ekleyeceğinizle ilgili tam bir eğitim sunar. Yine, kendi dünya haritanızı oluşturabilirsiniz, ancak bunu sizin için bulmak için neden Google’a bırakmıyorsunuz?

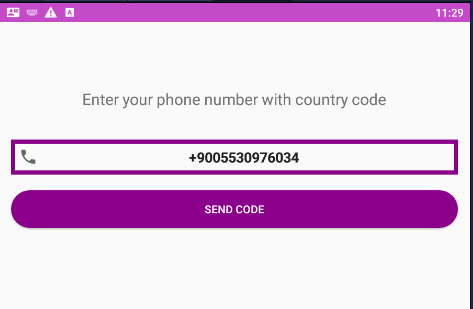
Eklemek istediğiniz harita özelliğine özgü farklı paketler var. “Geçerli Yeri Seç”, geliştiricilerin, cihazın geçerli konumunu bulabilen bir işlev oluşturmasını sağlar.**[9]**

# 4. TARTIŞMA VE SONUÇ

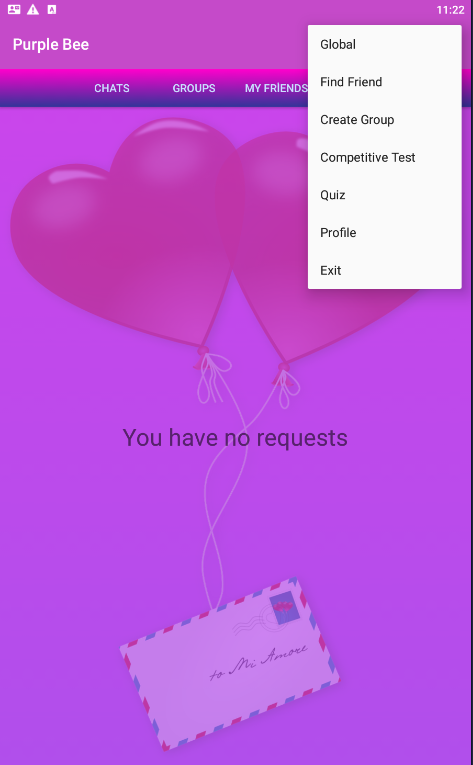
Yazmış olduğumuz Java dilindeki yazılımımız android telefonlarda ve tabletlerde çalıştırıldı. Programın bölümleri ve bölümlerinde neler olduğu ekran alıntılarıyla gösterildi.



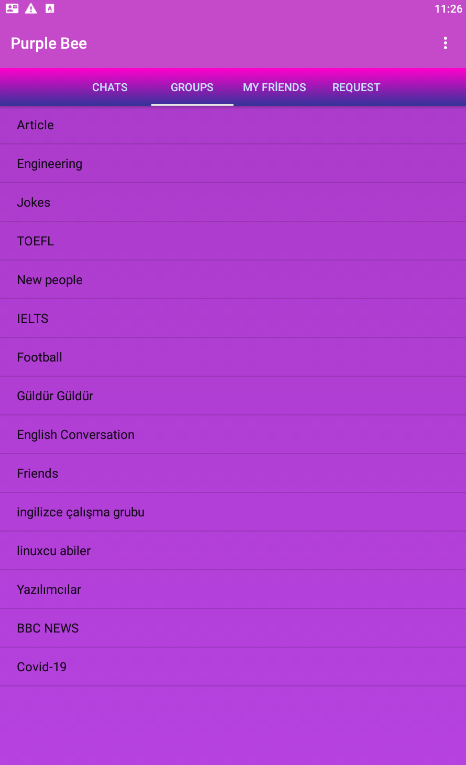
Şekil 4.1 E-mail ile Programa Giriş Ekranı



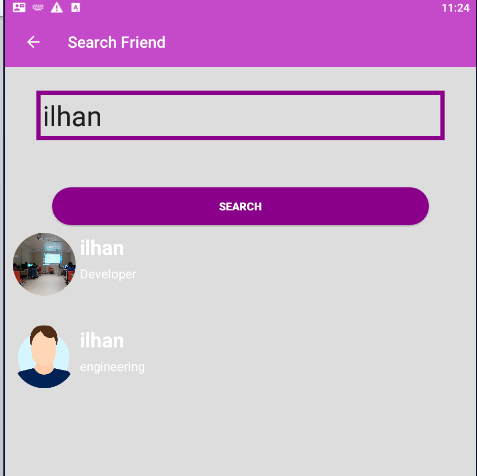
Şekil 4.2 Cep Telefonu İle Giriş



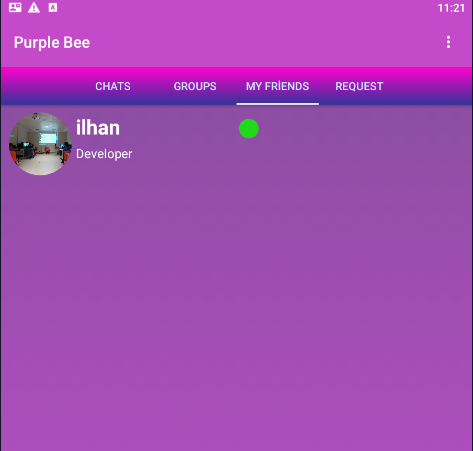
Şekil 4.3 Seçenekler Kısmı ve İçindeki Araçlar



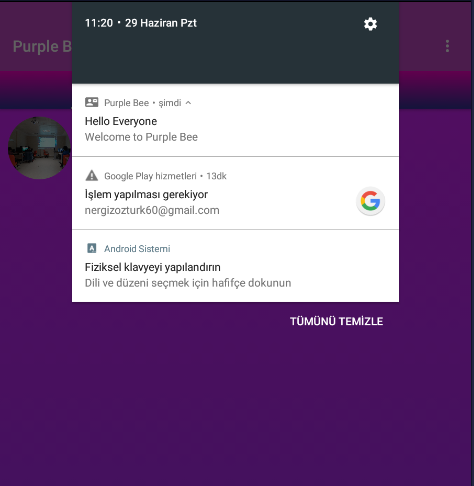
Şekil 4.4 Oluşturulan Gruplar



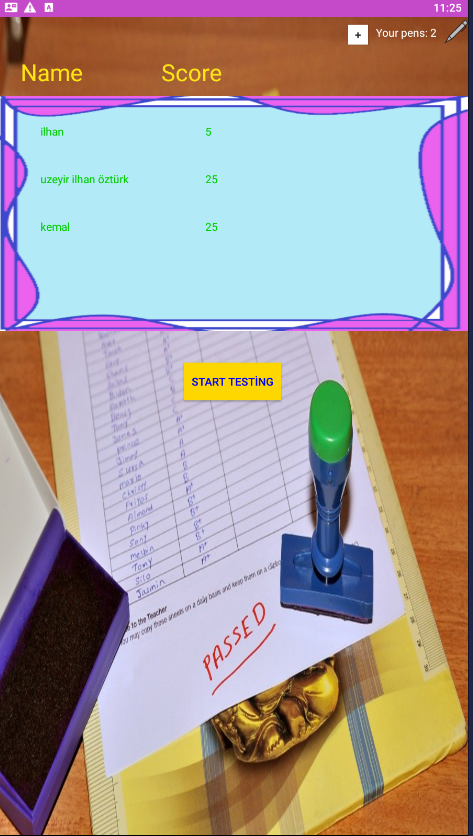
Şekil 4.5 Arkadaş Arama



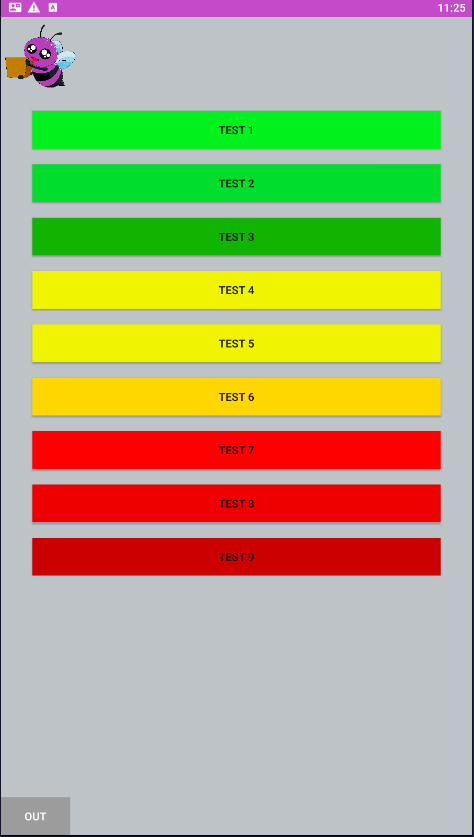
Şekil 4.6 Arkadaşların Çevrimiçi Görünmesi



Şekil 4.7 OneSignal Ekranı



Şekil 4.8 Rekabetçi İngilizce Testi Çözme [10]



Şekil 4.9 İngilizce Quizleri

# 5. KAYNAKÇA

[1]<https://tr.wikipedia.org/wiki/Android_(işletim_sistemi)#:~:text=Temmuz%202005%27te%20Google%2C%20Android,Sears%20ve%20Chris%20White%27dır>.

[2][http://ssryapim.blogspot.com/p/android-studio-nedir-android- studio.html](http://ssryapim.blogspot.com/p/android-studio-nedir-android-%20%20%20%20%20%20%20%20studio.html)

[3]<http://selcukoglu.blogspot.com/2011/02/java-nin-tarihcesi-ve-ozellikleri.html>

[4] <https://www.mediaclick.com.tr/blog/firebase-nedir>

[5] <https://blog.112dijital.com/web-push-notification-bildirim-kurulumu-detayli-anlatim/#:~:text=One%20Signal%20hem%20mobil%20cihazlar,Notification%20(bildirim%20alma)%20sistemidir>.

[6] <https://www.ulusaltezmerkezi.net/>

[7]<https://www.argenova.com.tr/en-iyi-10-android-gelistirme-ortami#:~:text=Android%20Studio%2C%20Android%20uygulama%20geliştirme,geliştirilen%20özel%20bir%20geliştirme%20ortamıdır>.

[8]<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-401/firebase-realtime-database> 2019

[9]<https://medium.com/heapjet-t%C3%BCrkiye/sdk-nedir-api-ile-sdk-aras%C4%B1ndaki-fark-nedir-4c5aaa07ca6c>

[10] <https://pixabay.com/>

# 6. EKLER

**EK-A** **MainActivity**

public class MainActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ViewPager myViewPager;

public TabLayout myTablaLayout;

private SekmeErisimAdapter mySekmeErisimAdapter;

private int i=0;

// private static int SPLASH\_TIME\_OUT = 4000;

private AdView mAdView;

//firebase

private FirebaseAuth mYetki;

private DatabaseReference kullanicilarReference,grupreferans,mesajyolu,databaseReference;

private FirebaseDatabase database;

private InterstitialAd mInterstitialAd;

private String aktifKullaniciId,IdMesajAlici,IdmesajGonderen;

private String alici="0";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Intent iin = getIntent();

Bundle bundle = iin.getExtras();

if(bundle!=null)

{

alici = (String)bundle.get("intro");

}

//intro

if(alici.equals("intro"))

{

Intent intent2 = new Intent(getApplicationContext(),IntroActivity.class);

startActivity(intent2);

}

//animasyon

RelativeLayout relativeLayout = findViewById(R.id.arka\_plan);

AnimationDrawable animationDrawable = (AnimationDrawable) relativeLayout.getBackground();

animationDrawable.setEnterFadeDuration(2000);

animationDrawable.setExitFadeDuration(4000);

animationDrawable.start();

//admob

mAdView = findViewById(R.id.adView);

AdRequest adRequest = new AdRequest.Builder().build();

mAdView.loadAd(adRequest);

mYetki=FirebaseAuth.getInstance();

// OneSignal Initialization

kullanicilarReference=FirebaseDatabase.getInstance().getReference();

databaseReference=FirebaseDatabase.getInstance().getReference();

// OneSignal Initialization

mToolbar=findViewById(R.id.anasayfa\_toolbar);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle("Purple Bee");

myViewPager = findViewById(R.id.ana\_sekmeler\_pager);

mySekmeErisimAdapter = new SekmeErisimAdapter (getSupportFragmentManager());

myViewPager.setAdapter(mySekmeErisimAdapter);

myTablaLayout=findViewById(R.id.anasekmeler);

myTablaLayout.setupWithViewPager(myViewPager);

//firebase

grupreferans=FirebaseDatabase.getInstance().getReference("Gruplar");

}

@Override

protected void onStart() {

super.onStart();

try {

FirebaseUser mevcutKullanici=mYetki.getCurrentUser();

if(mevcutKullanici==null)

{

KullaniciyiLoginActivityeGonder();

}

else

{

kullaniciDurumuGuncelle("çevrimiçi");

kullanicininVarliginiDogrula();

}

}catch (Exception e){

e.printStackTrace();

}

}

@Override

protected void onStop() {

super.onStop();

FirebaseUser mevcutKullanici=mYetki.getCurrentUser();

if(mevcutKullanici != null)

{

kullaniciDurumuGuncelle("çevrimdışı");

}

}

@Override

protected void onDestroy() {

super.onDestroy();

FirebaseUser mevcutKullanici=mYetki.getCurrentUser();

if(mevcutKullanici != null)

{

kullaniciDurumuGuncelle("çevrimdışı");

}

}

private void kullanicininVarliginiDogrula() {

String mevcutkullaniciId = mYetki.getCurrentUser().getUid();

kullanicilarReference.child("Kullanicilar").child(mevcutkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if((dataSnapshot.child("ad").exists()))

{

OneSignal.startInit(MainActivity.this)

.inFocusDisplaying(OneSignal.OSInFocusDisplayOption.Notification)

.unsubscribeWhenNotificationsAreDisabled(true)

.init();

OneSignal.idsAvailable(new OneSignal.IdsAvailableHandler() {

@Override

public void idsAvailable(final String userId, String registrationId) {

UUID uuid = UUID.randomUUID();

final String uuidString = uuid.toString();

DatabaseReference sorguDatabaseReference = FirebaseDatabase.getInstance().getReference("PlayerIds");

sorguDatabaseReference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

ArrayList<String> playerIDsFromserver = new ArrayList<>();

for(DataSnapshot ds: dataSnapshot.getChildren())

{

HashMap<String,String> hashMap = (HashMap<String, String>) ds.getValue();

String currentPlayerId = hashMap.get("playerID");

playerIDsFromserver.add(currentPlayerId);

//Toast.makeText(MainActivity.this, currentPlayerId, Toast.LENGTH\_SHORT).show();

}

if(!playerIDsFromserver.contains(userId))

{

databaseReference.child("PlayerIds").child(uuidString).child("playerID").setValue(userId);

String mevcutId = mYetki.getCurrentUser().getUid();

kullanicilarReference.child("Kullanicilar").child(mevcutId).child("BildirimID").setValue(userId);

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

});

}

else

{

Intent ayarlar = new Intent(MainActivity.this,AyarlarActivity.class);

//Geri tuşunu engelleme

ayarlar.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(ayarlar);

finish();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void KullaniciyiLoginActivityeGonder() {

Intent loginIntent = new Intent(MainActivity.this,LoginActivity.class);

loginIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(loginIntent);

finish();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

super.onCreateOptionsMenu(menu);

getMenuInflater().inflate(R.menu.secenekler\_menu,menu);

return true;

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

super.onOptionsItemSelected(item);

if(item.getItemId()==R.id.ana\_arkadasBul)

{

Intent arkadaBul = new Intent(MainActivity.this,ArkadasBulActivity.class);

startActivity(arkadaBul);

}

if(item.getItemId()==R.id.ana\_serach)

{

Intent arama = new Intent(MainActivity.this,SearchActivity.class);

startActivity(arama);

}

if(item.getItemId()==R.id.rekabetci)

{

Intent rekabet = new Intent(MainActivity.this,RekabetciActivity.class);

startActivity(rekabet);

}

if(item.getItemId()==R.id.quiz)

{

Intent quiz = new Intent(MainActivity.this,QuizActivity.class);

startActivity(quiz);

}

if(item.getItemId()==R.id.ayarlar)

{

Intent ayarla = new Intent(MainActivity.this,AyarlarActivity.class);

startActivity(ayarla);

}

if(item.getItemId()==R.id.cikisYap)

{

kullaniciDurumuGuncelle("Offline");

mYetki.signOut();

Intent giris = new Intent(MainActivity.this,LoginActivity.class);

startActivity(giris);

}

if(item.getItemId()==R.id.ana\_grup\_olustur)

{

//admob geçiş

MobileAds.initialize(this,"ca-app-pub-2724540026642482~8500905698");

final InterstitialAd interstitialAd = new InterstitialAd(this);

interstitialAd.setAdUnitId("ca-app-pub-2724540026642482/5661073643");

interstitialAd.loadAd(new AdRequest.Builder().build());

interstitialAd.setAdListener(new AdListener(){

@Override

public void onAdLoaded() {

super.onAdLoaded();

interstitialAd.show();

}

});

yeniGrupTalebi();

}

return true;

}

//Grup Oluşturma

private void yeniGrupTalebi() {

AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this,R.style.AlertDialog);

builder.setTitle("Enter group name:");

final EditText grupalani = new EditText(MainActivity.this);

grupalani.setHint("Sample: Music Group");

grupalani.setMaxLines(2);

builder.setView(grupalani);

builder.setPositiveButton("Create", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

String grupadi = grupalani.getText().toString();

if(TextUtils.isEmpty(grupadi))

{

Toast.makeText(MainActivity.this, "Enter group name", Toast.LENGTH\_LONG).show();

}

else if(grupalani.length()>50)

{

Toast.makeText(MainActivity.this, "too long!", Toast.LENGTH\_LONG).show();

}

else

{

YeniGrupolustur(grupadi);

}

}

});

builder.setNegativeButton("İptal", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

dialogInterface.cancel();

}

});

builder.show();

}

private void YeniGrupolustur(final String grupadi) {

grupreferans.child(grupadi).setValue("")

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if(task.isSuccessful())

{

Toast.makeText(MainActivity.this, grupadi+" group created.", Toast.LENGTH\_LONG).show();

}

}

});

}

public void kullaniciDurumuGuncelle(String durum)

{

String kaydedilenAktifZaman,kaydedilenAktifTarih;

Calendar calendar = Calendar.getInstance();

//Tarih formatı

SimpleDateFormat aktifTarih = new SimpleDateFormat("MMM dd, yyyy");

kaydedilenAktifTarih = aktifTarih.format(calendar.getTime());

// saat formatı

SimpleDateFormat aktifZaman = new SimpleDateFormat("hh:mm a");

kaydedilenAktifZaman = aktifZaman.format(calendar.getTime());

HashMap<String,Object> cevrimiciDurumu = new HashMap<>();

cevrimiciDurumu.put("zaman",kaydedilenAktifZaman);

cevrimiciDurumu.put("tarih",kaydedilenAktifTarih);

cevrimiciDurumu.put("durum",durum);

aktifKullaniciId=mYetki.getCurrentUser().getUid();

kullanicilarReference.child("Kullanicilar").child(aktifKullaniciId).child("kullaniciDurumu").updateChildren(cevrimiciDurumu);

}

}

**EK-B** **LoginActivity**

public class LoginActivity extends AppCompatActivity {

private Button girisButton, telefonGirisButton;

private EditText KullaniciMail, Kullanicisifre;

private TextView YenihesapAlma;

//Firebase

private FirebaseAuth mYetki;

ProgressDialog girisDialog;

private DatabaseReference kullaniciYolu;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

//kontrol tanımlamalari

girisButton = findViewById(R.id.giris\_button);

telefonGirisButton= findViewById(R.id.telefonla\_giris);

KullaniciMail=findViewById(R.id.giris\_email);

Kullanicisifre=findViewById(R.id.giris\_password);

YenihesapAlma=findViewById(R.id.yeni\_hesapalma);

//progressdialog

girisDialog= new ProgressDialog(this);

//Firebase

mYetki=FirebaseAuth.getInstance();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

YenihesapAlma.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent kayitaktivityIntent = new Intent(LoginActivity.this,KayitActivity.class);

startActivity(kayitaktivityIntent);

}

});

girisButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

KullaniciyaGirisizniver();

}

});

telefonGirisButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent telefonlaOturum = new Intent(LoginActivity.this,TelefonlaAcmaActivity.class);

startActivity(telefonlaOturum);

}

});

}

private void KullaniciyaGirisizniver() {

String email = KullaniciMail.getText().toString();

String sifre = Kullanicisifre.getText().toString();

if(TextUtils.isEmpty(email))

{

Toast.makeText(this, "Please enter e-mail", Toast.LENGTH\_SHORT).show();

}

if(TextUtils.isEmpty(sifre))

{

Toast.makeText(this, "Please enter password", Toast.LENGTH\_SHORT).show();

}

else

{

//progress

girisDialog.setTitle("Logging in");

girisDialog.setMessage("Please Wait");

girisDialog.setCanceledOnTouchOutside(true);

girisDialog.show();

//giris

mYetki.signInWithEmailAndPassword(email,sifre)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful())

{

String aktifKullaniciId = mYetki.getCurrentUser().getUid();

// Benzeriz bir ID bildrim için

String cihatToken = FirebaseInstanceId.getInstance().getToken();

//Tokken numarasını kullanıcı bilgilerine ekleme

kullaniciYolu.child(aktifKullaniciId).child("cihaz\_token").setValue(cihatToken)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if(task.isSuccessful())

{

Intent anasayfa = new Intent(LoginActivity.this,MainActivity.class);

anasayfa.putExtra("intro","intro");

anasayfa.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(anasayfa);

finish();

Toast.makeText(LoginActivity.this, "Login successful", Toast.LENGTH\_SHORT).show();

girisDialog.dismiss();

}

}

});

}

else

{

String mesaj = task.getException().toString();

Toast.makeText(LoginActivity.this, "Error: "+ mesaj +"Check Your Information", Toast.LENGTH\_LONG).show();

girisDialog.dismiss();

}

}

});

}

}

private void KullaniciyiAnaAktivitiyeGonder() {

Intent AnaAktiviteIntent = new Intent(LoginActivity.this,MainActivity.class);

startActivity(AnaAktiviteIntent);

}

}

**EK-C Kayıt Activity**

public class KayitActivity extends AppCompatActivity {

private Button KayitolusturmaButton;

private EditText KullaniciEmail, Kullanicisifre;

private TextView ZatenHesabimVar;

//firebase

private DatabaseReference kokReference;

private FirebaseAuth mYetki;

private ProgressDialog yukleniyorDialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_kayit);

//firebase tanımlaması

mYetki=FirebaseAuth.getInstance();

kokReference = FirebaseDatabase.getInstance().getReference();

//kayitoluşturma buttonu

KayitolusturmaButton=findViewById(R.id.kayit\_button);

KullaniciEmail=findViewById(R.id.kayit\_email);

Kullanicisifre=findViewById(R.id.kayit\_sifre1);

ZatenHesabimVar=findViewById(R.id.var\_olanhesap);

yukleniyorDialog=new ProgressDialog(this);

ZatenHesabimVar.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent loginActivityIntent = new Intent(KayitActivity.this,LoginActivity.class);

startActivity(loginActivityIntent);

}

});

KayitolusturmaButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

YeniHesapolustur();

}

private void YeniHesapolustur() {

String email = KullaniciEmail.getText().toString();

String sifre = Kullanicisifre.getText().toString();

if(TextUtils.isEmpty(email))

{

Toast.makeText(KayitActivity.this, "Please enter the e-mail", Toast.LENGTH\_SHORT).show();

}

if(TextUtils.isEmpty(sifre))

{

Toast.makeText(KayitActivity.this, "Please enter the password", Toast.LENGTH\_SHORT).show();

}

else

{

yukleniyorDialog.setTitle("Creating a new account");

yukleniyorDialog.setMessage("Please Wait");

yukleniyorDialog.setCanceledOnTouchOutside(true);

yukleniyorDialog.show();

mYetki.createUserWithEmailAndPassword(email,sifre)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful())

{

//Benzersiz bildirim ID

String cihazToken = FirebaseInstanceId.getInstance().getToken();

String mevcutKullaniciid=mYetki.getCurrentUser().getUid();

kokReference.child("Kullanicilar").child(mevcutKullaniciid).setValue("");

kokReference.child("Kullanicilar").child(mevcutKullaniciid).child("cihaz\_token")

.setValue(cihazToken);

Intent anasayfa= new Intent(KayitActivity.this,MainActivity.class);

anasayfa.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

finish();

// sql ---------------------------------------------------------------

try {

SQLiteDatabase sqLiteDatabase = KayitActivity.this.openOrCreateDatabase("Haklar",MODE\_PRIVATE,null);

sqLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS Haklar (hak INT(2))");

sqLiteDatabase.execSQL("INSERT INTO Haklar (hak) VALUES (2)");

}catch (Exception e){

e.printStackTrace();

}

Toast.makeText(KayitActivity.this, "New Account Successfully Created", Toast.LENGTH\_SHORT).show();

startActivity(anasayfa);

yukleniyorDialog.dismiss();

}

else {

String mesaj = task.getException().toString();

Toast.makeText(KayitActivity.this, "Error: " + mesaj + "Check Your Information", Toast.LENGTH\_SHORT).show();

yukleniyorDialog.dismiss();

}

}

});

}

}

});

}

}

**EK- D Chat Fragment**

public class ChatsFragment extends Fragment {

private View OzelsohbetlerView;

private RecyclerView sohbetlerListesi;

//Firebase

private DatabaseReference sohbetyolu,kullaniciYolu;

private FirebaseAuth mYetki;

private String aktifKullaniciId;

public ChatsFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

OzelsohbetlerView= inflater.inflate(R.layout.fragment\_chats, container, false);

//Firebase

mYetki=FirebaseAuth.getInstance();

aktifKullaniciId=mYetki.getCurrentUser().getUid();

sohbetyolu= FirebaseDatabase.getInstance().getReference().child("Sohbetler").child(aktifKullaniciId);

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

sohbetlerListesi=OzelsohbetlerView.findViewById(R.id.sohbetler\_listesi);

sohbetlerListesi.setLayoutManager(new LinearLayoutManager(getContext()));

return OzelsohbetlerView;

}

@Override

public void onStart() {

super.onStart();

FirebaseRecyclerOptions<Kisiler> secenekler = new FirebaseRecyclerOptions.Builder<Kisiler>()

.setQuery(sohbetyolu,Kisiler.class)

.build();

FirebaseRecyclerAdapter<Kisiler,SohbetlerViewHolder> adapter = new FirebaseRecyclerAdapter<Kisiler, SohbetlerViewHolder>(secenekler) {

@Override

protected void onBindViewHolder(@NonNull final SohbetlerViewHolder sohbetlerViewHolder, int i, @NonNull Kisiler kisiler) {

final String kullaniciIdleri = getRef(i).getKey();

final String[] resimal = {"Varsayılan Resim"};

//Veritabanından veri çağırma

kullaniciYolu.child(kullaniciIdleri).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

if(dataSnapshot.hasChild("resim"))

{

resimal[0] = dataSnapshot.child("resim").getValue().toString();

//Veri tabanından gelen veriyi kontrole aktarma

Picasso.get().load(resimal[0]).into(sohbetlerViewHolder.profilresmi);

}

// Veri tabanından kullanıcı durumuna yönelik verileri çekme

final String adAll= dataSnapshot.child("ad").getValue().toString();

final String durumAl= dataSnapshot.child("durum").getValue().toString();

sohbetlerViewHolder.kullaniciAdi.setText(adAll);

if(dataSnapshot.child("kullaniciDurumu").hasChild("durum"))

{

String durum = dataSnapshot.child("kullaniciDurumu").child("durum").getValue().toString();

String tarih = dataSnapshot.child("kullaniciDurumu").child("tarih").getValue().toString();

String zaman = dataSnapshot.child("kullaniciDurumu").child("zaman").getValue().toString();

if(durum.equals("çevrimiçi"))

{

sohbetlerViewHolder.kullaniciDurumu.setText("Online");

}

else if(durum.equals("çevrimdışı"))

{

sohbetlerViewHolder.kullaniciDurumu.setText("Last Seen: "+tarih + " "+zaman);

}

}

//her satıra tıklandığında ne yapsın

sohbetlerViewHolder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

//chat aktivitisine git

Intent chatAktivite = new Intent(getContext(),ChatActivity.class);

chatAktivite.putExtra("kullanici\_id\_ziyaret",kullaniciIdleri);

chatAktivite.putExtra("kullanici\_adi\_ziyaret",adAll);

chatAktivite.putExtra("resim\_ziyaret", resimal[0]);

startActivity(chatAktivite);

}

});

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

@NonNull

@Override

public SohbetlerViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.kullanici\_gosterme\_layout,parent,false);

return new SohbetlerViewHolder(view);

}

};

sohbetlerListesi.setAdapter(adapter);

adapter.notifyDataSetChanged();

adapter.startListening();

}

public static class SohbetlerViewHolder extends RecyclerView.ViewHolder {

//Kontroller

CircleImageView profilresmi;

TextView kullaniciAdi,kullaniciDurumu;

public SohbetlerViewHolder(@NonNull View itemView) {

super(itemView);

kullaniciAdi=itemView.findViewById(R.id.kullanici\_profil\_adi);

kullaniciDurumu=itemView.findViewById(R.id.kullanici\_durumu);

profilresmi=itemView.findViewById(R.id.kullanicilar\_profil\_resmi);

}

}

}

**EK –E Kisiler Fragment**

public class KisilerFragment extends Fragment {

private View KisilerView;

private RecyclerView kisilerListem;

//Firebase

private DatabaseReference SohbetlerYolu,KullanicilarYolu;

private FirebaseAuth mYetki;

private String aktifKullaniciId;

public KisilerFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

KisilerView= inflater.inflate(R.layout.fragment\_kisiler, container, false);

kisilerListem=KisilerView.findViewById(R.id.kisiler\_listesi);

kisilerListem.setLayoutManager(new LinearLayoutManager(getContext()));

//Firebase

mYetki=FirebaseAuth.getInstance();

aktifKullaniciId=mYetki.getCurrentUser().getUid();

SohbetlerYolu= FirebaseDatabase.getInstance().getReference().child("Sohbetler").child(aktifKullaniciId);

KullanicilarYolu = FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

return KisilerView;

}

@Override

public void onStart() {

super.onStart();

FirebaseRecyclerOptions secenekler = new FirebaseRecyclerOptions.Builder<Kisiler>()

.setQuery(SohbetlerYolu,Kisiler.class)

.build();

FirebaseRecyclerAdapter<Kisiler,KisilerViewHolder>adapter = new FirebaseRecyclerAdapter<Kisiler, KisilerViewHolder>(secenekler) {

@Override

protected void onBindViewHolder(@NonNull final KisilerViewHolder kisilerViewHolder, int i, @NonNull Kisiler kisiler) {

String tiklanansatirkullaniciId = getRef(i).getKey();

KullanicilarYolu.child(tiklanansatirkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

String durumAna = dataSnapshot.child("durum").getValue().toString();

kisilerViewHolder.kullaniciDurumu.setText(durumAna);

if(dataSnapshot.child("kullaniciDurumu").hasChild("durum"))

{

String durum = dataSnapshot.child("kullaniciDurumu").child("durum").getValue().toString();

String tarih = dataSnapshot.child("kullaniciDurumu").child("tarih").getValue().toString();

String zaman = dataSnapshot.child("kullaniciDurumu").child("zaman").getValue().toString();

if(durum.equals("çevrimiçi"))

{

kisilerViewHolder.cevrimiciIconu.setVisibility(View.VISIBLE);

}

else

{

kisilerViewHolder.cevrimiciIconu.setVisibility(View.INVISIBLE);

}

}

else

{

kisilerViewHolder.cevrimiciIconu.setVisibility(View.INVISIBLE);

kisilerViewHolder.kullaniciDurumu.setText("Offline");

}

if(dataSnapshot.hasChild("resim"))

{

String profilresmi = dataSnapshot.child("resim").getValue().toString();

String kullaniciAdi = dataSnapshot.child("ad").getValue().toString();

// String kullaniciDurumu = dataSnapshot.child("durum").getValue().toString();

kisilerViewHolder.kullaniciAdi.setText(kullaniciAdi);

// kisilerViewHolder.kullaniciDurumu.setText(kullaniciDurumu);

Picasso.get().load(profilresmi).placeholder(R.drawable.yeni\_person).into(kisilerViewHolder.profilresmi);

}

else

{

//resim olmadan işlemlere devam etme

String kullaniciAdi = dataSnapshot.child("ad").getValue().toString();

String kullaniciDurumu = dataSnapshot.child("durum").getValue().toString();

kisilerViewHolder.kullaniciAdi.setText(kullaniciAdi);

kisilerViewHolder.kullaniciDurumu.setText(kullaniciDurumu);

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

@NonNull

@Override

public KisilerViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.kullanici\_gosterme\_layout,parent,false);

KisilerViewHolder viewHolder = new KisilerViewHolder(view);

return viewHolder;

}

};

kisilerListem.setAdapter(adapter);

adapter.notifyDataSetChanged();

adapter.startListening();

}

public static class KisilerViewHolder extends RecyclerView.ViewHolder {

TextView kullaniciAdi,kullaniciDurumu;

CircleImageView profilresmi;

ImageView cevrimiciIconu;

public KisilerViewHolder(@NonNull View itemView) {

super(itemView);

kullaniciAdi=itemView.findViewById(R.id.kullanici\_profil\_adi);

kullaniciDurumu=itemView.findViewById(R.id.kullanici\_durumu);

profilresmi=itemView.findViewById(R.id.kullanicilar\_profil\_resmi);

cevrimiciIconu=itemView.findViewById(R.id.kullanici\_cemrimici\_olma\_durumu);

}

}

}

**EK –F Grup Chat Activity**

public class GrupChatsActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ImageButton Mesajgondermebuttonu;

private EditText kullaniciGirdisi;

private ScrollView mScrollView;

private TextView metinmesajlarinigoster;

private DatabaseReference kullaniciYolu,grupadiYolu,grupmesajanahtariyolu;

//Firebase

private FirebaseAuth mYetki;

private String mevcutgrupadi,aktifkullaniciId,aktifkullaniciADi,aktifTarih,aktifzaman;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_grup\_chats);

//intenti al

mevcutgrupadi=getIntent().getExtras().get("grupAdi").toString();

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

aktifkullaniciId=mYetki.getCurrentUser().getUid();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

grupadiYolu= FirebaseDatabase.getInstance().getReference().child("Gruplar").child(mevcutgrupadi);

// tanımlamalar

mToolbar = findViewById(R.id.grup\_chat\_bar\_layout);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle(mevcutgrupadi);

Mesajgondermebuttonu=findViewById(R.id.msaj\_gonderme\_buttonu);

kullaniciGirdisi=findViewById(R.id.grup\_mesaj\_girdisi);

metinmesajlarinigoster = findViewById(R.id.grup\_chat\_metni\_gosterme);

mScrollView=findViewById(R.id.grup\_scrollview);

Typeface typeface = Typeface.createFromAsset(getAssets(),"fonts/opensans.ttf");

metinmesajlarinigoster.setTypeface(typeface);

kullanicibilgisial();

//mesajııı göndermeeeee

Mesajgondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

MesajVeritabaninaKaydet();

kullaniciGirdisi.setText("");

}

});

}

@Override

protected void onStart() {

super.onStart();

grupadiYolu.addChildEventListener(new ChildEventListener() {

@Override

public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

}

@Override

public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void mesajlariGoster(DataSnapshot dataSnapshot) {

Iterator iterator = dataSnapshot.getChildren().iterator();

while (iterator.hasNext())

{

String sohbetAdi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetmesaji = (String)((DataSnapshot)iterator.next()).getValue();

String sohbettarihi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetZamani = (String)((DataSnapshot)iterator.next()).getValue();

metinmesajlarinigoster.append(sohbetZamani +" "+ sohbettarihi +"\n" + sohbetAdi+": "+sohbetmesaji + "\n" +"\n\n\n" );

// Toast.makeText(this, sohbetmesaji, Toast.LENGTH\_SHORT).show();

}

}

private void MesajVeritabaninaKaydet() {

String mesaj= kullaniciGirdisi.getText().toString();

String mesajAnahtari= grupadiYolu.push().getKey();

if(TextUtils.isEmpty(mesaj))

{

Toast.makeText(this, "Enter message!", Toast.LENGTH\_SHORT).show();

}

else

{

Calendar Tarihicintakvim=Calendar.getInstance();

SimpleDateFormat aktifTarihiformat = new SimpleDateFormat("MMM dd, yyyy");

aktifTarih=aktifTarihiformat.format(Tarihicintakvim.getTime());

Calendar zamanicinTak = Calendar.getInstance();

SimpleDateFormat aktifzamanformati = new SimpleDateFormat("hh:mm:ss a");

aktifzaman = aktifzamanformati.format(zamanicinTak.getTime());

HashMap<String,Object>grupMesajAnahtari = new HashMap<>();

grupadiYolu.updateChildren(grupMesajAnahtari);

grupmesajanahtariyolu = grupadiYolu.child(mesajAnahtari);

HashMap<String,Object> mesajbilgisimap = new HashMap<>();

mesajbilgisimap.put("ad",aktifkullaniciADi);

mesajbilgisimap.put("mesaj",mesaj);

mesajbilgisimap.put("tarih",aktifTarih);

mesajbilgisimap.put("zaman",aktifzaman);

grupmesajanahtariyolu.updateChildren(mesajbilgisimap);

}

}

private void kullanicibilgisial() {

kullaniciYolu.child(aktifkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

aktifkullaniciADi=dataSnapshot.child("ad").getValue().toString();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

}

**EK-G Profil Aktivity**

public class GrupChatsActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ImageButton Mesajgondermebuttonu;

private EditText kullaniciGirdisi;

private ScrollView mScrollView;

private TextView metinmesajlarinigoster;

private DatabaseReference kullaniciYolu,grupadiYolu,grupmesajanahtariyolu;

//Firebase

private FirebaseAuth mYetki;

private String mevcutgrupadi,aktifkullaniciId,aktifkullaniciADi,aktifTarih,aktifzaman;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_grup\_chats);

//intenti al

mevcutgrupadi=getIntent().getExtras().get("grupAdi").toString();

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

aktifkullaniciId=mYetki.getCurrentUser().getUid();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

grupadiYolu= FirebaseDatabase.getInstance().getReference().child("Gruplar").child(mevcutgrupadi);

// tanımlamalar

mToolbar = findViewById(R.id.grup\_chat\_bar\_layout);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle(mevcutgrupadi);

Mesajgondermebuttonu=findViewById(R.id.msaj\_gonderme\_buttonu);

kullaniciGirdisi=findViewById(R.id.grup\_mesaj\_girdisi);

metinmesajlarinigoster = findViewById(R.id.grup\_chat\_metni\_gosterme);

mScrollView=findViewById(R.id.grup\_scrollview);

Typeface typeface = Typeface.createFromAsset(getAssets(),"fonts/opensans.ttf");

metinmesajlarinigoster.setTypeface(typeface);

kullanicibilgisial();

//mesajııı göndermeeeee

Mesajgondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

MesajVeritabaninaKaydet();

kullaniciGirdisi.setText("");

}

});

}

@Override

protected void onStart() {

super.onStart();

grupadiYolu.addChildEventListener(new ChildEventListener() {

@Override

public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

}

@Override

public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void mesajlariGoster(DataSnapshot dataSnapshot) {

Iterator iterator = dataSnapshot.getChildren().iterator();

while (iterator.hasNext())

{

String sohbetAdi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetmesaji = (String)((DataSnapshot)iterator.next()).getValue();

String sohbettarihi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetZamani = (String)((DataSnapshot)iterator.next()).getValue();

metinmesajlarinigoster.append(sohbetZamani +" "+ sohbettarihi +"\n" + sohbetAdi+": "+sohbetmesaji + "\n" +"\n\n\n" );

// Toast.makeText(this, sohbetmesaji, Toast.LENGTH\_SHORT).show();

}

}

private void MesajVeritabaninaKaydet() {

String mesaj= kullaniciGirdisi.getText().toString();

String mesajAnahtari= grupadiYolu.push().getKey();

if(TextUtils.isEmpty(mesaj))

{

Toast.makeText(this, "Enter message!", Toast.LENGTH\_SHORT).show();

}

else

{

Calendar Tarihicintakvim=Calendar.getInstance();

SimpleDateFormat aktifTarihiformat = new SimpleDateFormat("MMM dd, yyyy");

aktifTarih=aktifTarihiformat.format(Tarihicintakvim.getTime());

Calendar zamanicinTak = Calendar.getInstance();

SimpleDateFormat aktifzamanformati = new SimpleDateFormat("hh:mm:ss a");

aktifzaman = aktifzamanformati.format(zamanicinTak.getTime());

HashMap<String,Object>grupMesajAnahtari = new HashMap<>();

grupadiYolu.updateChildren(grupMesajAnahtari);

grupmesajanahtariyolu = grupadiYolu.child(mesajAnahtari);

HashMap<String,Object> mesajbilgisimap = new HashMap<>();

mesajbilgisimap.put("ad",aktifkullaniciADi);

mesajbilgisimap.put("mesaj",mesaj);

mesajbilgisimap.put("tarih",aktifTarih);

mesajbilgisimap.put("zaman",aktifzaman);

grupmesajanahtariyolu.updateChildren(mesajbilgisimap);

}

}

private void kullanicibilgisial() {

kullaniciYolu.child(aktifkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

aktifkullaniciADi=dataSnapshot.child("ad").getValue().toString();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

}

**EK-H Talepler Fragment**

public class GrupChatsActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ImageButton Mesajgondermebuttonu;

private EditText kullaniciGirdisi;

private ScrollView mScrollView;

private TextView metinmesajlarinigoster;

private DatabaseReference kullaniciYolu,grupadiYolu,grupmesajanahtariyolu;

//Firebase

private FirebaseAuth mYetki;

private String mevcutgrupadi,aktifkullaniciId,aktifkullaniciADi,aktifTarih,aktifzaman;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_grup\_chats);

//intenti al

mevcutgrupadi=getIntent().getExtras().get("grupAdi").toString();

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

aktifkullaniciId=mYetki.getCurrentUser().getUid();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

grupadiYolu= FirebaseDatabase.getInstance().getReference().child("Gruplar").child(mevcutgrupadi);

// tanımlamalar

mToolbar = findViewById(R.id.grup\_chat\_bar\_layout);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle(mevcutgrupadi);

Mesajgondermebuttonu=findViewById(R.id.msaj\_gonderme\_buttonu);

kullaniciGirdisi=findViewById(R.id.grup\_mesaj\_girdisi);

metinmesajlarinigoster = findViewById(R.id.grup\_chat\_metni\_gosterme);

mScrollView=findViewById(R.id.grup\_scrollview);

Typeface typeface = Typeface.createFromAsset(getAssets(),"fonts/opensans.ttf");

metinmesajlarinigoster.setTypeface(typeface);

kullanicibilgisial();

//mesajııı göndermeeeee

Mesajgondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

MesajVeritabaninaKaydet();

kullaniciGirdisi.setText("");

}

});

}

@Override

protected void onStart() {

super.onStart();

grupadiYolu.addChildEventListener(new ChildEventListener() {

@Override

public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

}

@Override

public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void mesajlariGoster(DataSnapshot dataSnapshot) {

Iterator iterator = dataSnapshot.getChildren().iterator();

while (iterator.hasNext())

{

String sohbetAdi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetmesaji = (String)((DataSnapshot)iterator.next()).getValue();

String sohbettarihi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetZamani = (String)((DataSnapshot)iterator.next()).getValue();

metinmesajlarinigoster.append(sohbetZamani +" "+ sohbettarihi +"\n" + sohbetAdi+": "+sohbetmesaji + "\n" +"\n\n\n" );

// Toast.makeText(this, sohbetmesaji, Toast.LENGTH\_SHORT).show();

}

}

private void MesajVeritabaninaKaydet() {

String mesaj= kullaniciGirdisi.getText().toString();

String mesajAnahtari= grupadiYolu.push().getKey();

if(TextUtils.isEmpty(mesaj))

{

Toast.makeText(this, "Enter message!", Toast.LENGTH\_SHORT).show();

}

else

{

Calendar Tarihicintakvim=Calendar.getInstance();

SimpleDateFormat aktifTarihiformat = new SimpleDateFormat("MMM dd, yyyy");

aktifTarih=aktifTarihiformat.format(Tarihicintakvim.getTime());

Calendar zamanicinTak = Calendar.getInstance();

SimpleDateFormat aktifzamanformati = new SimpleDateFormat("hh:mm:ss a");

aktifzaman = aktifzamanformati.format(zamanicinTak.getTime());

HashMap<String,Object>grupMesajAnahtari = new HashMap<>();

grupadiYolu.updateChildren(grupMesajAnahtari);

grupmesajanahtariyolu = grupadiYolu.child(mesajAnahtari);

HashMap<String,Object> mesajbilgisimap = new HashMap<>();

mesajbilgisimap.put("ad",aktifkullaniciADi);

mesajbilgisimap.put("mesaj",mesaj);

mesajbilgisimap.put("tarih",aktifTarih);

mesajbilgisimap.put("zaman",aktifzaman);

grupmesajanahtariyolu.updateChildren(mesajbilgisimap);

}

}

private void kullanicibilgisial() {

kullaniciYolu.child(aktifkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

aktifkullaniciADi=dataSnapshot.child("ad").getValue().toString();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

}

**EK-I Search Aktivity**

public class GrupChatsActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ImageButton Mesajgondermebuttonu;

private EditText kullaniciGirdisi;

private ScrollView mScrollView;

private TextView metinmesajlarinigoster;

private DatabaseReference kullaniciYolu,grupadiYolu,grupmesajanahtariyolu;

//Firebase

private FirebaseAuth mYetki;

private String mevcutgrupadi,aktifkullaniciId,aktifkullaniciADi,aktifTarih,aktifzaman;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_grup\_chats);

//intenti al

mevcutgrupadi=getIntent().getExtras().get("grupAdi").toString();

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

aktifkullaniciId=mYetki.getCurrentUser().getUid();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

grupadiYolu= FirebaseDatabase.getInstance().getReference().child("Gruplar").child(mevcutgrupadi);

// tanımlamalar

mToolbar = findViewById(R.id.grup\_chat\_bar\_layout);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle(mevcutgrupadi);

Mesajgondermebuttonu=findViewById(R.id.msaj\_gonderme\_buttonu);

kullaniciGirdisi=findViewById(R.id.grup\_mesaj\_girdisi);

metinmesajlarinigoster = findViewById(R.id.grup\_chat\_metni\_gosterme);

mScrollView=findViewById(R.id.grup\_scrollview);

Typeface typeface = Typeface.createFromAsset(getAssets(),"fonts/opensans.ttf");

metinmesajlarinigoster.setTypeface(typeface);

kullanicibilgisial();

//mesajııı göndermeeeee

Mesajgondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

MesajVeritabaninaKaydet();

kullaniciGirdisi.setText("");

}

});

}

@Override

protected void onStart() {

super.onStart();

grupadiYolu.addChildEventListener(new ChildEventListener() {

@Override

public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

}

@Override

public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void mesajlariGoster(DataSnapshot dataSnapshot) {

Iterator iterator = dataSnapshot.getChildren().iterator();

while (iterator.hasNext())

{

String sohbetAdi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetmesaji = (String)((DataSnapshot)iterator.next()).getValue();

String sohbettarihi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetZamani = (String)((DataSnapshot)iterator.next()).getValue();

metinmesajlarinigoster.append(sohbetZamani +" "+ sohbettarihi +"\n" + sohbetAdi+": "+sohbetmesaji + "\n" +"\n\n\n" );

// Toast.makeText(this, sohbetmesaji, Toast.LENGTH\_SHORT).show();

}

}

private void MesajVeritabaninaKaydet() {

String mesaj= kullaniciGirdisi.getText().toString();

String mesajAnahtari= grupadiYolu.push().getKey();

if(TextUtils.isEmpty(mesaj))

{

Toast.makeText(this, "Enter message!", Toast.LENGTH\_SHORT).show();

}

else

{

Calendar Tarihicintakvim=Calendar.getInstance();

SimpleDateFormat aktifTarihiformat = new SimpleDateFormat("MMM dd, yyyy");

aktifTarih=aktifTarihiformat.format(Tarihicintakvim.getTime());

Calendar zamanicinTak = Calendar.getInstance();

SimpleDateFormat aktifzamanformati = new SimpleDateFormat("hh:mm:ss a");

aktifzaman = aktifzamanformati.format(zamanicinTak.getTime());

HashMap<String,Object>grupMesajAnahtari = new HashMap<>();

grupadiYolu.updateChildren(grupMesajAnahtari);

grupmesajanahtariyolu = grupadiYolu.child(mesajAnahtari);

HashMap<String,Object> mesajbilgisimap = new HashMap<>();

mesajbilgisimap.put("ad",aktifkullaniciADi);

mesajbilgisimap.put("mesaj",mesaj);

mesajbilgisimap.put("tarih",aktifTarih);

mesajbilgisimap.put("zaman",aktifzaman);

grupmesajanahtariyolu.updateChildren(mesajbilgisimap);

}

}

private void kullanicibilgisial() {

kullaniciYolu.child(aktifkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

aktifkullaniciADi=dataSnapshot.child("ad").getValue().toString();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

}

**EK-J Arkadaş Bul Activity**

public class GrupChatsActivity extends AppCompatActivity {

private Toolbar mToolbar;

private ImageButton Mesajgondermebuttonu;

private EditText kullaniciGirdisi;

private ScrollView mScrollView;

private TextView metinmesajlarinigoster;

private DatabaseReference kullaniciYolu,grupadiYolu,grupmesajanahtariyolu;

//Firebase

private FirebaseAuth mYetki;

private String mevcutgrupadi,aktifkullaniciId,aktifkullaniciADi,aktifTarih,aktifzaman;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_grup\_chats);

//intenti al

mevcutgrupadi=getIntent().getExtras().get("grupAdi").toString();

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

aktifkullaniciId=mYetki.getCurrentUser().getUid();

kullaniciYolu= FirebaseDatabase.getInstance().getReference().child("Kullanicilar");

grupadiYolu= FirebaseDatabase.getInstance().getReference().child("Gruplar").child(mevcutgrupadi);

// tanımlamalar

mToolbar = findViewById(R.id.grup\_chat\_bar\_layout);

setSupportActionBar(mToolbar);

getSupportActionBar().setTitle(mevcutgrupadi);

Mesajgondermebuttonu=findViewById(R.id.msaj\_gonderme\_buttonu);

kullaniciGirdisi=findViewById(R.id.grup\_mesaj\_girdisi);

metinmesajlarinigoster = findViewById(R.id.grup\_chat\_metni\_gosterme);

mScrollView=findViewById(R.id.grup\_scrollview);

Typeface typeface = Typeface.createFromAsset(getAssets(),"fonts/opensans.ttf");

metinmesajlarinigoster.setTypeface(typeface);

kullanicibilgisial();

//mesajııı göndermeeeee

Mesajgondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

MesajVeritabaninaKaydet();

kullaniciGirdisi.setText("");

}

});

}

@Override

protected void onStart() {

super.onStart();

grupadiYolu.addChildEventListener(new ChildEventListener() {

@Override

public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

if(dataSnapshot.exists())

{

mesajlariGoster(dataSnapshot);

}

}

@Override

public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

}

@Override

public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

private void mesajlariGoster(DataSnapshot dataSnapshot) {

Iterator iterator = dataSnapshot.getChildren().iterator();

while (iterator.hasNext())

{

String sohbetAdi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetmesaji = (String)((DataSnapshot)iterator.next()).getValue();

String sohbettarihi = (String)((DataSnapshot)iterator.next()).getValue();

String sohbetZamani = (String)((DataSnapshot)iterator.next()).getValue();

metinmesajlarinigoster.append(sohbetZamani +" "+ sohbettarihi +"\n" + sohbetAdi+": "+sohbetmesaji + "\n" +"\n\n\n" );

// Toast.makeText(this, sohbetmesaji, Toast.LENGTH\_SHORT).show();

}

}

private void MesajVeritabaninaKaydet() {

String mesaj= kullaniciGirdisi.getText().toString();

String mesajAnahtari= grupadiYolu.push().getKey();

if(TextUtils.isEmpty(mesaj))

{

Toast.makeText(this, "Enter message!", Toast.LENGTH\_SHORT).show();

}

else

{

Calendar Tarihicintakvim=Calendar.getInstance();

SimpleDateFormat aktifTarihiformat = new SimpleDateFormat("MMM dd, yyyy");

aktifTarih=aktifTarihiformat.format(Tarihicintakvim.getTime());

Calendar zamanicinTak = Calendar.getInstance();

SimpleDateFormat aktifzamanformati = new SimpleDateFormat("hh:mm:ss a");

aktifzaman = aktifzamanformati.format(zamanicinTak.getTime());

HashMap<String,Object>grupMesajAnahtari = new HashMap<>();

grupadiYolu.updateChildren(grupMesajAnahtari);

grupmesajanahtariyolu = grupadiYolu.child(mesajAnahtari);

HashMap<String,Object> mesajbilgisimap = new HashMap<>();

mesajbilgisimap.put("ad",aktifkullaniciADi);

mesajbilgisimap.put("mesaj",mesaj);

mesajbilgisimap.put("tarih",aktifTarih);

mesajbilgisimap.put("zaman",aktifzaman);

grupmesajanahtariyolu.updateChildren(mesajbilgisimap);

}

}

private void kullanicibilgisial() {

kullaniciYolu.child(aktifkullaniciId).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.exists())

{

aktifkullaniciADi=dataSnapshot.child("ad").getValue().toString();

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

}

}

**EK-K Telefonla Açma Activity**

public class TelefonlaAcmaActivity extends AppCompatActivity {

private Button dogrulamaKodugondermebuttonu,dogrulamabuttonu;

private EditText telefonnumarasiGirdisi,dogrulamakodugirdisi;

private PhoneAuthProvider.OnVerificationStateChangedCallbacks callbacks;

private String mDogrulamaID;

private PhoneAuthProvider.ForceResendingToken mResendToken;

//Firebase

FirebaseAuth mYetki;

//yükleniyor penceresi

private ProgressDialog yuklemebar;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_telefonla\_acma);

//Tanımlamalar

dogrulamaKodugondermebuttonu=findViewById(R.id.dogrulama\_kodu\_gonder\_buttonu);

dogrulamabuttonu=findViewById(R.id.dogrulama\_buttonu);

//Progress dialog tanımlama

yuklemebar=new ProgressDialog(this);

//Firebase tanımlama

mYetki=FirebaseAuth.getInstance();

telefonnumarasiGirdisi=findViewById(R.id.telefon\_numarasi\_girdi);

dogrulamakodugirdisi=findViewById(R.id.dogrulama\_kodunu\_yazın);

dogrulamaKodugondermebuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String telefonNumarasi= telefonnumarasiGirdisi.getText().toString();

if(TextUtils.isEmpty(telefonNumarasi))

{

Toast.makeText(TelefonlaAcmaActivity.this, "Enter the telephone number!", Toast.LENGTH\_SHORT).show();

}

else

{

//Yükleniyor penceresi

yuklemebar.setTitle("Phone Verification");

yuklemebar.setMessage("Please Wait");

yuklemebar.setCanceledOnTouchOutside(false);

yuklemebar.show();

PhoneAuthProvider.getInstance().verifyPhoneNumber(

telefonNumarasi, // Phone number to verify

60, // Timeout duration

TimeUnit.SECONDS, // Unit of timeout

TelefonlaAcmaActivity.this, // Activity (for callback binding)

callbacks);

}

}

});

dogrulamabuttonu.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

dogrulamaKodugondermebuttonu.setVisibility(View.INVISIBLE);

telefonnumarasiGirdisi.setVisibility(View.INVISIBLE);

String dogrulamakodu = dogrulamakodugirdisi.getText().toString();

if(TextUtils.isEmpty(dogrulamakodu))

{

Toast.makeText(TelefonlaAcmaActivity.this, "Enter verification code!", Toast.LENGTH\_SHORT).show();

}

else

{

//Yükleniyor penceresi

yuklemebar.setTitle("Code Verification");

yuklemebar.setMessage("Please Wait");

yuklemebar.setCanceledOnTouchOutside(false);

yuklemebar.show();

PhoneAuthCredential credential = PhoneAuthProvider.getCredential(mDogrulamaID, dogrulamakodu);

telefonlagirisyap(credential);

}

}

});

callbacks= new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

@Override

public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential) {

telefonlagirisyap(phoneAuthCredential);

}

@Override

public void onVerificationFailed(FirebaseException e)

{

//Yukleme penceresi

yuklemebar.dismiss();

Toast.makeText(TelefonlaAcmaActivity.this, "Invalid telephone number\n" +

" Be sure to enter with Country Code!", Toast.LENGTH\_LONG).show();

//Görüntü ayarları

dogrulamaKodugondermebuttonu.setVisibility(View.VISIBLE);

dogrulamabuttonu.setVisibility(View.INVISIBLE);

telefonnumarasiGirdisi.setVisibility(View.VISIBLE);

dogrulamakodugirdisi.setVisibility(View.INVISIBLE);

}

@Override

public void onCodeSent(@NonNull String verificationId,

@NonNull PhoneAuthProvider.ForceResendingToken token) {

// Save verification ID and resending token so we can use them later

mDogrulamaID= verificationId;

mResendToken = token;

//yukleme penceresi

yuklemebar.dismiss();

Toast.makeText(TelefonlaAcmaActivity.this, "Code Sent. Please wait a little", Toast.LENGTH\_LONG).show();

//Görünüm ayarları

dogrulamaKodugondermebuttonu.setVisibility(View.INVISIBLE);

dogrulamabuttonu.setVisibility(View.VISIBLE);

telefonnumarasiGirdisi.setVisibility(View.INVISIBLE);

dogrulamakodugirdisi.setVisibility(View.VISIBLE);

// ...

}

};

}

private void telefonlagirisyap (PhoneAuthCredential credential) {

mYetki.signInWithCredential(credential)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

yuklemebar.dismiss();

Toast.makeText(TelefonlaAcmaActivity.this, "Congratulations Logged in", Toast.LENGTH\_SHORT).show();

// sql ------------------------------

try {

SQLiteDatabase sqLiteDatabase = TelefonlaAcmaActivity.this.openOrCreateDatabase("Haklar",MODE\_PRIVATE,null);

sqLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS Haklar (hak INT(2))");

sqLiteDatabase.execSQL("INSERT INTO Haklar (hak) VALUES (2)");

}catch (Exception e){

e.printStackTrace();

}

KullaniciyiAnasayfayaGonder();

}

else

{

String hatamesaji = task.getException().toString();

Toast.makeText(TelefonlaAcmaActivity.this, "Error:" + hatamesaji , Toast.LENGTH\_LONG).show();

}

}

});

}

private void KullaniciyiAnasayfayaGonder() {

Intent anasayfa = new Intent(TelefonlaAcmaActivity.this,MainActivity.class);

anasayfa.putExtra("intro","intro");

anasayfa.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(anasayfa);

finish();

}

}

**EK-L** **Mesaj Adaptor**

public class MesajAdaptor extends RecyclerView.Adapter<MesajAdaptor.MesajlarViewHolder> {

private List<Mesajlar> kullanicimesajlarListesi;

//firebase

private FirebaseAuth mYetki;

private DatabaseReference kullanicilarYolu;

public MesajAdaptor (List<Mesajlar> kullanicimesajlarListesi)

{

this.kullanicimesajlarListesi = kullanicimesajlarListesi;

}

//View Holder

public class MesajlarViewHolder extends RecyclerView.ViewHolder

{

//özel mesajdaki kontroller

public CircleImageView aliciProfilResmi;

public TextView gondrenMesajMetni,aliciMesajMetni;

public ImageView mesajgondrenResim,mesajalanResim;

public MesajlarViewHolder(@NonNull View itemView) {

super(itemView);

aliciMesajMetni=itemView.findViewById(R.id.alici\_mesaj\_metni);

gondrenMesajMetni=itemView.findViewById(R.id.gonderen\_mesaj\_metni);

aliciProfilResmi=itemView.findViewById(R.id.mesaj\_profil\_resmi);

mesajgondrenResim=itemView.findViewById(R.id.mesaj\_gonderen\_image\_view);

mesajalanResim=itemView.findViewById(R.id.mesaj\_alan\_image\_view);

}

}

@NonNull

@Override

public MesajlarViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.ozel\_mesajlar\_layout,parent,false);

//firebaseTanımlama

mYetki=FirebaseAuth.getInstance();

return new MesajlarViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull final MesajlarViewHolder holder, final int position) {

String mesajGonderenId = mYetki.getCurrentUser().getUid();

//model tanımlama

Mesajlar mesajlar = kullanicimesajlarListesi.get(position);

final String kimdenKullaniciId = mesajlar.getKimden();

String kimdenmesajturu = mesajlar.getTur();

//Veri tabanı yolu

kullanicilarYolu = FirebaseDatabase.getInstance().getReference().child("Kullanicilar").child(kimdenKullaniciId);

//FireBaseden veri çekme

kullanicilarYolu.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

if(dataSnapshot.hasChild("resim"))

{

// Veritabanındaki resmi aktarma

String resmiAlici = dataSnapshot.child("resim").getValue().toString();

//kontrolle resmi aktarma

Picasso.get().load(resmiAlici).placeholder(R.drawable.ic\_person1).into(holder.aliciProfilResmi);

}

else

{

}

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

}

});

//Görünmez Yapma

holder.aliciMesajMetni.setVisibility(View.GONE);

holder.aliciProfilResmi.setVisibility(View.GONE);

holder.gondrenMesajMetni.setVisibility(View.GONE);

holder.mesajalanResim.setVisibility(View.GONE);

holder.mesajgondrenResim.setVisibility(View.GONE);

if(kimdenmesajturu.equals("metin"))

{

if(kimdenKullaniciId.equals(mesajGonderenId))

{

holder.aliciProfilResmi.setVisibility(View.INVISIBLE);

holder.gondrenMesajMetni.setVisibility(View.VISIBLE);

holder.gondrenMesajMetni.setBackgroundResource(R.drawable.gonderen\_mesajlar\_layout);

holder.gondrenMesajMetni.setTextColor(Color.BLACK);

holder.gondrenMesajMetni.setText(mesajlar.getZaman()+" "+mesajlar.getTarih()+"\n\n"+mesajlar.getMesaj());

}

else

{

//görünür yapma

holder.aliciProfilResmi.setVisibility(View.VISIBLE);

holder.aliciMesajMetni.setVisibility(View.VISIBLE);

holder.aliciMesajMetni.setBackgroundResource(R.drawable.alici\_mesajlari\_layout);

holder.aliciMesajMetni.setTextColor(Color.BLACK);

holder.aliciMesajMetni.setText(mesajlar.getZaman()+" "+mesajlar.getTarih()+"\n\n"+mesajlar.getMesaj());

}

}

else if (kimdenmesajturu.equals("resim"))

{

if(kimdenKullaniciId.equals(mesajGonderenId))

{

holder.mesajgondrenResim.setVisibility(View.VISIBLE);

Picasso.get().load(mesajlar.getMesaj()).into(holder.mesajgondrenResim);

}

else

{

holder.aliciProfilResmi.setVisibility(View.VISIBLE);

holder.mesajalanResim.setVisibility(View.VISIBLE);

Picasso.get().load(mesajlar.getMesaj()).into(holder.mesajalanResim);

}

}

else if(kimdenmesajturu.equals("pdf")||kimdenmesajturu.equals("docx"))

{

if(kimdenKullaniciId.equals(mesajGonderenId))

{

holder.mesajgondrenResim.setVisibility(View.VISIBLE);

Picasso.get().load("https://firebasestorage.googleapis.com/v0/b/talktofriends-923c2.appspot.com/o/Resim%20Dosyalar%C4%B1%2Fdocument.png?alt=media&token=f5cdb221-9959-4ae5-95ee-6a51dfb05041")

.into(holder.mesajgondrenResim);

}

else

{

holder.aliciProfilResmi.setVisibility(View.VISIBLE);

holder.mesajalanResim.setVisibility(View.VISIBLE);

Picasso.get().load("https://firebasestorage.googleapis.com/v0/b/talktofriends-923c2.appspot.com/o/Resim%20Dosyalar%C4%B1%2Fdocument.png?alt=media&token=f5cdb221-9959-4ae5-95ee-6a51dfb05041")

.into(holder.mesajalanResim);

}

}

if(kimdenKullaniciId.equals(mesajGonderenId))

{

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(kullanicimesajlarListesi.get(position).getTur().equals("pdf")||kullanicimesajlarListesi.get(position).getTur().equals("docx"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"Bu belgeyi indir ve görüntüle",

"İptal",

"Herkesten sil"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

gonderilenmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

if(i == 1)

{

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(kullanicimesajlarListesi.get(position).getMesaj()));

holder.itemView.getContext().startActivity(intent);

}

if(i == 2)

{

//iptal butonu

}

if(i == 3)

{

mesajiHerkestenSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

}

});

builder.show();

}

else if(kullanicimesajlarListesi.get(position).getTur().equals("metin"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"Herkesten sil"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

gonderilenmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

else if(i == 1)

{

mesajiHerkestenSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

}

});

builder.show();

}

else if(kullanicimesajlarListesi.get(position).getTur().equals("resim"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"Bu Resmi Görüntüle",

"İPTAL",

"Herkesten sil"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

gonderilenmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

else if(i == 1)

{

Intent intent = new Intent(holder.itemView.getContext(), ResimGostermeActivity.class);

intent.putExtra("url",kullanicimesajlarListesi.get(position).getMesaj());

holder.itemView.getContext().startActivity(intent);

}

else if(i == 3)

{

mesajiHerkestenSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

}

});

builder.show();

}

}

});

}

else

{

//Alıcı kısmı

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(kullanicimesajlarListesi.get(position).getTur().equals("pdf")||kullanicimesajlarListesi.get(position).getTur().equals("docx"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"Bu belgeyi indir ve görüntüle",

"İptal"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

alinanmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

if(i == 1)

{

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(kullanicimesajlarListesi.get(position).getMesaj()));

holder.itemView.getContext().startActivity(intent);

}

if(i == 2)

{

}

}

});

builder.show();

}

else if(kullanicimesajlarListesi.get(position).getTur().equals("metin"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"İptal"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

alinanmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

}

});

builder.show();

}

else if(kullanicimesajlarListesi.get(position).getTur().equals("resim"))

{

CharSequence secenekler [] = new CharSequence[]

{

"Benden Sil",

"Bu Resmi Görüntüle",

"İptal"

};

AlertDialog.Builder builder = new AlertDialog.Builder(holder.itemView.getContext());

builder.setTitle("Mesaj Silinsin mi?");

builder.setItems(secenekler, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

if(i == 0)

{

alinanmesajiSil(position,holder);

Intent intent = new Intent(holder.itemView.getContext(), MainActivity.class);

holder.itemView.getContext().startActivity(intent);

}

if(i == 1)

{

Intent intent = new Intent(holder.itemView.getContext(), ResimGostermeActivity.class);

intent.putExtra("url",kullanicimesajlarListesi.get(position).getMesaj());

holder.itemView.getContext().startActivity(intent);

}

}

});

builder.show();

}

}

});

}

}

@Override

public int getItemCount() {

return kullanicimesajlarListesi.size();

}

private void gonderilenmesajiSil(final int pozisyon, final MesajlarViewHolder holder )

{

final DatabaseReference mesajyolu = FirebaseDatabase.getInstance().getReference();

mesajyolu.child("Mesajlar")

.child(kullanicimesajlarListesi.get(pozisyon).getKimden())

.child(kullanicimesajlarListesi.get(pozisyon).getKime())

.child(kullanicimesajlarListesi.get(pozisyon).getMesajID())

.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful())

{

mesajyolu.child("Mesajlar")

.child(kullanicimesajlarListesi.get(pozisyon).getKimden())

.child(kullanicimesajlarListesi.get(pozisyon).getKime())

.child(kullanicimesajlarListesi.get(pozisyon).getMesajID())

.removeValue();

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

else

{

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

}

});

}

private void alinanmesajiSil(final int pozisyon, final MesajlarViewHolder holder )

{

DatabaseReference mesajyolu = FirebaseDatabase.getInstance().getReference();

mesajyolu.child("Mesajlar")

.child(kullanicimesajlarListesi.get(pozisyon).getKime())

.child(kullanicimesajlarListesi.get(pozisyon).getKimden())

.child(kullanicimesajlarListesi.get(pozisyon).getMesajID())

.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful())

{

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

else

{

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

}

});

}

private void mesajiHerkestenSil(final int pozisyon, final MesajlarViewHolder holder )

{

final DatabaseReference mesajyolu = FirebaseDatabase.getInstance().getReference();

mesajyolu.child("Mesajlar")

.child(kullanicimesajlarListesi.get(pozisyon).getKime())

.child(kullanicimesajlarListesi.get(pozisyon).getKimden())

.child(kullanicimesajlarListesi.get(pozisyon).getMesajID())

.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful())

{

mesajyolu.child("Mesajlar")

.child(kullanicimesajlarListesi.get(pozisyon).getKimden())

.child(kullanicimesajlarListesi.get(pozisyon).getKime())

.child(kullanicimesajlarListesi.get(pozisyon).getMesajID())

.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if(task.isSuccessful())

{

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

}

});

}

else

{

Toast.makeText(holder.itemView.getContext(), "Silme işlemi başarılı", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

**EK-M Kisiler Class**

public class Kisiler {

String ad,durum,resim;

public Kisiler() {

}

public Kisiler(String ad, String durum, String resim) {

this.ad = ad;

this.durum = durum;

this.resim = resim;

}

public String getAd() {

return ad;

}

public void setAd(String ad) {

this.ad = ad;

}

public String getDurum() {

return durum;

}

public void setDurum(String durum) {

this.durum = durum;

}

public String getResim() {

return resim;

}

public void setResim(String resim) {

this.resim = resim;

}

}

**EK- N Mesajlar Class**

public class Mesajlar {

private String kimden, mesaj, tur,kime,mesajID,zaman,tarih,ad;

public Mesajlar() {

}

public Mesajlar(String kimden, String mesaj, String tur, String kime, String mesajID, String zaman, String tarih, String ad) {

this.kimden = kimden;

this.mesaj = mesaj;

this.tur = tur;

this.kime = kime;

this.mesajID = mesajID;

this.zaman = zaman;

this.tarih = tarih;

this.ad = ad;

}

public String getKimden() {

return kimden;

}

public void setKimden(String kimden) {

this.kimden = kimden;

}

public String getMesaj() {

return mesaj;

}

public void setMesaj(String mesaj) {

this.mesaj = mesaj;

}

public String getTur() {

return tur;

}

public void setTur(String tur) {

this.tur = tur;

}

public String getKime() {

return kime;

}

public void setKime(String kime) {

this.kime = kime;

}

public String getMesajID() {

return mesajID;

}

public void setMesajID(String mesajID) {

this.mesajID = mesajID;

}

public String getZaman() {

return zaman;

}

public void setZaman(String zaman) {

this.zaman = zaman;

}

public String getTarih() {

return tarih;

}

public void setTarih(String tarih) {

this.tarih = tarih;

}

public String getAd() {

return ad;

}

public void setAd(String ad) {

this.ad = ad;

}

}

# ÖZGEÇMİŞ

Adı Soyadı: Üzeyir İlhan ÖZTÜRK

Doğum Yeri ve Tarih: İstanbul 05.03.1997

Yabancı Dili: İngilizce

İletişim (Telefon/e-posta):0553 097 6034 ilhan.ozturk@itsafe.com.tr

Eğitim Durumu (Kurum ve Yıl)

Lise: Atilla Uras Lisesi(2011-2015)

Lisans: Bozok Üniversitesi Bilgisayar Mühendisliği (2016-2020),

Adı Soyadı: Hasan KURT

Doğum Yeri ve Tarihi: İzmir 10.10.1998

Yabancı Dili: İngilizce

İletişim: (Telefon/e-posta):0531 374 2493 hasan.kurt.mh@gmail.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Güvercinlik Endüstri Meslek Lisesi, (2012-2016)

Lisans : Bozok Üniversitesi Bilgisayar Mühendisliği(2016-2020)