

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÇALIŞMASI

SON RAPOR

PROJENİN ADI

KANINLA KURTAR

YAZANLAR

Ahmet Bedirhan SAĞIR (030414026)

Mohammad ALRAWI (030114083)

DANIŞMAN

Doç. Dr. Fatih KOÇAN

ÖZET

Günümüzde insanlar birçok durumda kullanılmak üzere acil kan ihtiyacı duymaktadırlar. Ancak, bu ihtiyacın sadece çok azı karşılanabilmekte ve bu yüzden ölümler meydana gelebilmektedir.

Bu uygulama android iletim sistemi kullanan tüm akıllı telefonlarda, insanların kan ihtiyaçlarını en hızlı biçimde karşılanması için yapılmaktadır.

ABSTRACT

Nowadays, people need urgent blood to be used in many cases. However, only very less of these needs can be met and deaths can occur.

This app has been made for meeting the needs of people's blood in the fastest way on all smartphones, which uses the android mobile operating system.

ANAHTAR KELİMELER/ KEYWORDS

Kan, ihtiyaç, acil, kan bağışı, blood, urgent state, blood donation

İÇİNDEKİLER

ÖZET	iii
FİGÜR LİSTESİ.....	V
GİRİŞ	6
LİTERATÜR TARAMASI.....	7
PROJE HAKKINDA BİLGİLER.....	9
PROJE NEDİR?.....	10
FIREBASE HAKKINDA BİLGİLER	11
YAPILAN İŞLER	12
METOT, UYGULAMA VE TESTLER	15
METOT	15
UYGULAMA.....	18
TESTLER.....	20
GELECEKTE YAPILACAK İŞLER	21
KAN BAĞIŞI HAKKINDA BAZI BİLGİLER	22
KAYNAKÇA	30
EKLER	31

FIGÜR LİSTESİ

Resim 1: Google Firebase Kanınla Kurtar Sunucu Paneli.....	11
Resim 2: İş Paketi.....	13
Resim 3: Gantt Şeması.....	14
Resim 4: Lejand.....	14
Resim 5: Çevik Yazılım Geliştirme Döngüsü.....	16
Resim 6: Kan Grupları Tablosu.....	25

GİRİŞ

Kan; kırmızı renkli olan, içinde hayat için önemli olan birtakım maddeleri (oksijen, karbondioksit, akyuvar, alyuvar, trombosit, hormon vb.) bulunduran yaşam için zorunlu bir sıvıdır. “Tek kaynağının insan olması ve ihtiyaç duyulduğunda yerine konabilecek bir yedeğinin olmaması nedeniyle yaşam kurtarıcı bir sıvıdır” (Kan Verme 2019). İnsanlar sürekli olarak kana ihtiyaç duymaktadır. Ülkemizdeki kan stokları oldukça yetersizdir ve yapılan kan bağışları ihtiyaçları sağlayabilecek oranda değildir. Yılda çok sayıda hasta kan ihtiyacının karşılanamaması nedeniyle hayatını kaybetmektedir. Ayrıca kan ihtiyacı olduğunda bu ihtiyacın duyurulmasında da ciddi sıkıntılar vardır. Acil ve afet durumlarında zirve noktasına çıkan kan ihtiyacının karşılanmasında yaşanan bu ciddi problem, projemizin yapılmasının sebebini de oluşturmaktadır.

Problem için çözümümüz; herkesin kullanabileceği bir akıllı telefon uygulaması yapmaktır. Dünyada ve ülkemizde en çok kullanılan akıllı telefon işletim sisteminin Android® (Türkiyede her akıllı telefonun %81.6 sı (Stat Counter Türkiye Telefon Kullanım Bilgileri 2019) ve dünyadaki her akıllı telefonun %74.15’inin (Stat Counter Worldwide Smartphone infos 2019)) Android® işletim sistemine sahip olması sebebiyle, uygulamamız bir android uygulaması olacaktır.

Projemiz; kullanıcıların kan veya kan ürünü ihtiyacı olması durumunda hızlı bir şekilde bu ihtiyacın duyurulmasını sağlayacak, diğer kullanıcılara çağrı yaparak, kana/kan ürününe ihtiyacı olan kişiyle temasa geçmelerinde ve ihtiyacın sağlanmasında yardımcı olacaktır.

LİTERATÜR TARAMASI

Projemizin konusuyla alakalı gerek Google® Play da çok sayıda uygulama bulunmakla beraber biz sadece bu uygulamalardan 3 tanesi hakkında bilgi verip, kendi projemizin bunlardan farkını açıklayacağız.

Bunlardan ilki Kan Lazım (Kan Lazım 2017) adlı Android uygulamasıdır. Bu uygulamada kullanıcı sisteme kendisine ait tüm gerekli bilgileri girerek üye olur. Daha sonrasında ise kullanıcının yakınındaki kan ihtiyaçları kullanıcıya bildirim olarak gönderilir. Bu uygulamada ayrıca bir de anlaşmalı hastanelerde bulunan bir arayüz (Kan Lazım Hastane Arayüzü) vardır. Kan duyurusu sadece bu arayüz uygulaması aracılığı ile yapılabilmektedir. İlgili arayüz ise az sayıda hastanede bulunmaktadır.

Literatür taraması yaptığımız uygulamalardan ikincisi ise Bir Damla Umut (Bir Damla Umut 2017) adlı Android uygulamasıdır. Bu uygulamada kullanıcı sisteme kayıt olur, telefon numarası ve parola ile sisteme giriş yaptıktan sonra hem kan ihtiyacı ilanı verebilir hem de ihtiyaç sahibiyle iletişime geçebilir. Bu uygulamada kan ihtiyacı ilanı verildiğinde bu ilan tüm üyelere uyarı mesajı olarak gider. Bu uygulamadaki en büyük eksiklik kan ihtiyaç ilanlarının konuma göre listelenmemesidir. Örnek vermek gerekirse, İstanbul Bakırköy de ikamet eden bir üyeye Adana Seyhan'daki bir kan bağış ihtiyacı mesajı gönderilmektedir. Bu durum istenen bir durum olmamakla beraber, konuma göre kan ihtiyacının listelenmediğini bizlere göstermektedir.

Literatür taraması kapsamında tarama yaptığımız son uygulama ise Kan Bankasıdır. (Kan Bankası 2016) Bu uygulamada kullanıcı kayıt olur ve sisteme giriş yapar. Daha sonra ise ilanlar menüsündeki “+” işaretine basılarak yeni ilan verebilir. İlanları görebilir. Lakin system ihtiyaç sahibiyle bağışçı arasında iletişim kurmaz. (ihtiyaç sahibinin telefon numarası paylaşılır ama iletişime geçmeyle ilgili bir komut bu uygulamada bulunmaz) Ayrıca bu uygulamada da konuma göre ilan listeleme yoktur. Örnek vermek gerekirse, Ankara Çankayada ikamet eden bir üye, İzmir Tiredeki bir üyeye ait bir kan bağış ihtiyacı ilanını görebilmektedir. Bu durum istenen bir durum olmamakla beraber, konuma göre kan ihtiyacının listelenmediğini bizlere göstermektedir.

Projemizin bu projelerden farkı ve üstün olacağı noktalar; bulunulan il'e göre duyuruları görebilme ve kolay bir şekilde ilan verebilme imkanının sağlanabilmesi olacaktır.

PROJE HAKKINDA BİLGİLER

Projemiz; kan ihtiyacı ilanı vermeyi kolaylaştıran bir android uygulamasıdır. Bu uygulamada sunucu ihtiyacı Google Firebase, veritabanı ihtiyacı için Realtime Database, kullanıcı yetkilendirme ihtiyacı için Firebase Auth kullanıldı.

Kullanıcı ilan verebilmek için sisteme üye olmalı ve e-posta ve parolasıyla sisteme giriş yapmalıdır.

Üyelik için T.C. kimlik numarası, ad soyad, telefon, yaşadığı il, kan grubu, e-posta ve parola girişi gereklidir. Ayrıca bir e-posta ile sadece bir kullanıcı kaydolabilir.

Üye girişi e-posta ve parola ile yapılmaktadır.

Üye girişi yapan bir üye; ilan verebilir, verdiği ilanları görebilir, diğer ilanları görebilir, bilgilerini güncelleyebilir ve oturumu kapatabilir.

İlan vermek için birtakım bilgilerin girilmesi şarttır. Bu bilgiler; ilan verenin adı soyadı, ilan ili, hastane adı, iletişim numarası ve ilan bilgilerinin bulunduğu ilan detayıdır.

Verilen ilanlar Firebase sunucusundaki veritabanında ilanlar adlı tabloda tutulur.

İlan verilen il ile kullanıcının ili aynı ise, kullanıcıya notification (bildirim) gider. Kullanıcı bildirim basarak ilanı görebilir. Eğer bağışçı olacaksa da “BEN DE VARIM” butonuna basar. Bu butona basıldıktan sonra, kan verme şartlarını sağlayıp sağlamadığına dair bir sorgulama ekranını görür. Sorgulama ekranındaki şartlar sağlanırsa, bir sonraki aşamaya geçer. Bu aşama iletişime geçme aşamasıdır. Bu aşamada ilan gösterilir ve altında “KİŞİYİ ARA” butonu bulunur. Bu butona basılarak kullanıcı, ilan sahibiyle iletişime geçer.

PROJE NEDİR?

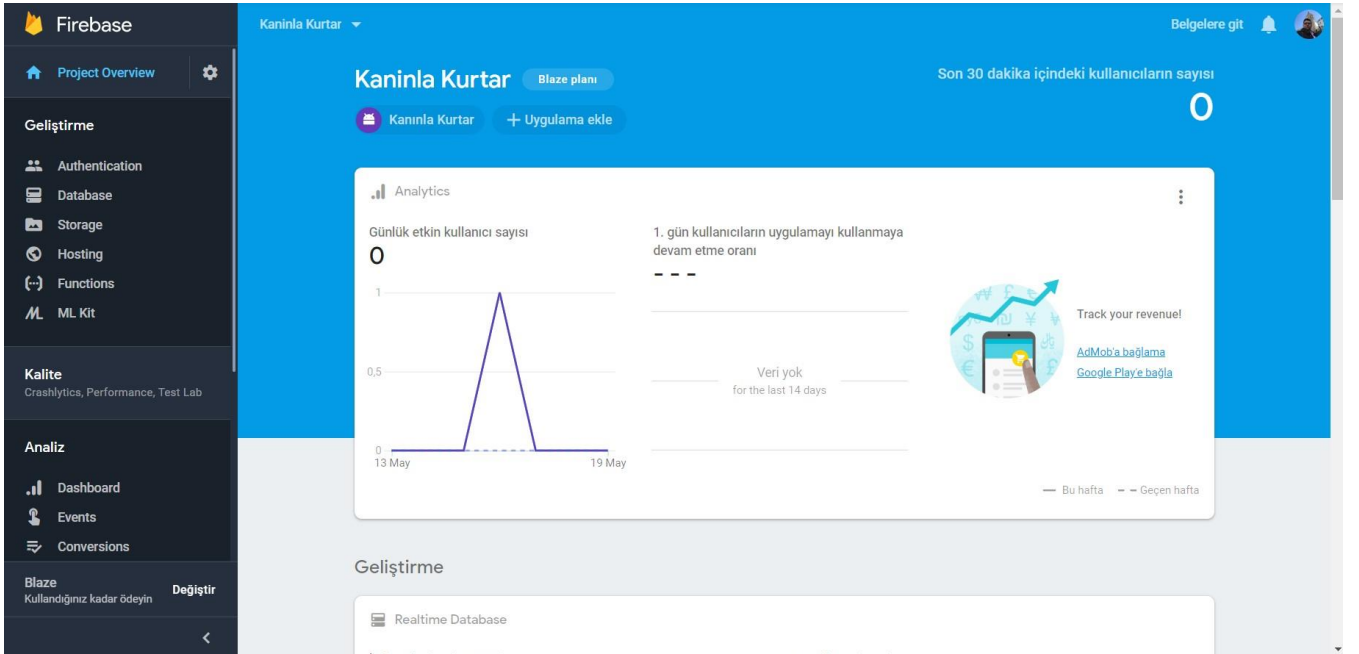
Proje insanların ihtiyaç duydukları kan ve kan ürünlerinin temininin sağlanması için oluşturulmuş olan mobil bir uygulamadır. Kan ve kan ürünleri bağışının hızlı bir biçimde sağlanabilmesi için, ihtiyaç duyurusunun sağlanması gerekmektedir. Her şeyin mobilize olduğu günümüz dünyasında, kan bağışı ilanlarının da mobil uygulamasını da olması gerekmektedir.

İşte bu yüzden, “KANINLA KURTAR” uygulaması yapılmıştır. Bu uygulama ile; kullanıcılar kolayca sisteme üye olabilir, sisteme giriş yaparak ilan vererek, diğer kullanıcılardan kan ve kan ürünü bağışı talebinde bulunabilirler veya kendileri de var olan bir talebe, -şartlar uygunsa- olumlu dönüş yaparak kan veya kan ürünleri bağışçısı olabilirler.

Kan ve kan ürünleri bağışı ile ilgili bilgileri, sonraki sayfalarda bulabilirsiniz.

FIREBASE HAKKINDA BİLGİLER

Google Firebase®; Google® firmasının geliştirmiş olduğu bir sunucu çözümüdür. Google Firebase ile uygulamalarınızın sunucu kullanımı ihtiyaçlarını karşılayabilirsiniz. Google Firebase, içerdiği birçok uygulama ile, uygulamalara veritabanı, depolama, kullanıcı girişi v.b. konularda yardımcı olmaktadır. Google Firebase ile uygulamanızın etkileşimi oldukça kolaydır. Google Firebase sayesinde uygulama-sunucu iletişimi için ayrıca bir php kodlama yapmanıza gerek kalmaz, çünkü Firebase'e projemizin package'ını girdiğinizde, Firebase ilgili json dosyasını sizin için otomatik olarak oluşturur. Siz de bu dosyayı uygulamanızın olduğu klasöre atabilirsiniz. Ayrıca, uygulama-sunucu bağlantısı için bu dosyanın klasöre atılması yeterli değildir. Firebase'in hangi hizmetini kullanmak istiyorsanız, o hizmetin implementation kodunu, gradle(proje) dosyasındaki dependencies bölümüne yapıştırmanız gerekir. Firebase'e uygulama ekleme komutu verdiğinizde, sistem size rehberlik ederek, uygulamanızın sunucu ile bağlantısını kolayca yaptırmaktadır. Bu yönüyle Google Firebase, uygulama geliştiriciler için oldukça uygun bir sunucu çözümüdür. Firebase ile sadece Android uygulamalar değil, İOS ve internet uygulamalarınıza da sunucu hizmeti verebilirsiniz.



Resim 1: Google Firebase Kanınla Kurtar Sunucu Paneli

YAPILAN İŞLER

Uygulamanın yapılması kapsamında; öncelikle sistem tanımı yapılmıştır. Bu kapsamda kan bağışı hakkında araştırma yapıldı. Kan verme şartları öğrenildi. Kan ve kan ürünleri bağışı ile ilgili bilgi sahibi olundu.

İhtiyaç analizi kapsamında programımızın kabiliyetleri belirlenmiş ve uygulamamızda nelerin bulunması gerektiği belirlenmiştir. Bu gereklilikler; kullanıcı kaydı, kullanıcı bilgilerinin ve ilanların veritabanında saklanması, kullanıcı ve ilan iline göre ilan gösterme, ilan ekleme ve iletişime geçmedir.

Tasarım aşamasında ise projemizin hangi programlama diliyle, hangi geliştirme ortamı (IDE) kullanılacağı, hangi programların hangi amaçlarla kullanılacağı belirlenmiş ve kullanıcı durum şeması (Usecase) şeması çıkarılmıştır. Programımız Android Studio uygulamasında, android diliyle geliştirilecek Veritabanı ve sunucu ihtiyacı Google Firebase üzerinden sağlanmış ve görsel üretimi içinse Paint 3D programı kullanılmıştır. Usecase ve diğer diyagramların tasarımı ise IBM Rational Rose Enterprise programında yapılmıştır.

Programın Firebase bağlantısı yapılmıştır. Böylelikle; kullanıcılar firebase sunucusuna eklendi. Program kullanıcıları programa kayıt olabiliyor, programa giriş-çıkış yapabilmektedirler. Ve ayrıca ilan verebilmektedirler. ilan görme modüllerinin oluşturulmasına devam edilmektedir. Bu modülde il tabanlı kayıt ve il'e göre ilan gösterme özellikleri bulunduğundan, bu kısım oldukça zaman almış olup projemiz zamanında bitmemiştir.

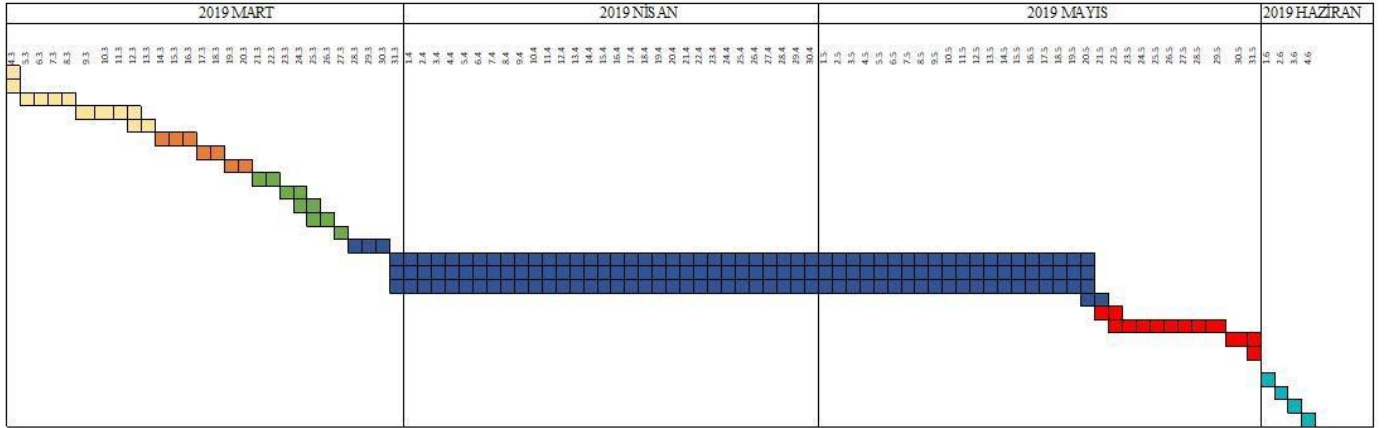
Yakın bir zaman içerisinde ilan ve kullanıcı iline göre ilan ve bildirimin gösterilmesi kısmı bitirilecek ve iletişim modülü de tamamlanarak, uygulamamız genel kullanım testleri için hazır hale getirilerek uygulama tamamlanacaktır. Tamamlanan uygulama Google Play Studio’da yayınlanacaktır.

Uygulamamıza ait iş paketi ve gantt şeması aşağıda verilmiştir.

KANINLA KURTAR ANDROID APP İŞ PAKETİ					
Aşama	Görev	Yapan	Başlangıç	Bitiş	Süre(Gün)
1	Fatih Koçan ile görüşme	Ahmet Bedirhan SAĞIR	4.03.2019	4.03.2019	1
1,1	Proje Fikrinin alınması	Ahmet Bedirhan SAĞIR	4.03.2019	4.03.2019	1
1,2	Proje detaylarının belirlenmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	5.03.2019	8.03.2019	4
1,3	Hedef belirleme	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	9.03.2019	12.03.2019	4
1,4	2. Aşamaya hazırlık ve 1. aşamanın gözden geçirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	12.03.2019	13.03.2019	2
2	İhtiyaçları belirleme	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	14.03.2019	16.03.2019	3
2,1	Sistem kabiliyetlerini belirleme	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	17.03.2019	18.03.2019	2
2,2	3. Aşamaya hazırlık ve 2. aşamanın gözden geçirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	19.03.2019	20.03.2019	2
3	Tasarım ilkelerinin belirlenmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	21.03.2019	22.03.2019	2
3,1	Kullanılacak programlama dilinin belirlenmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	23.03.2019	24.03.2019	2
3,2	Use case diagramının çıkarılması	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	24.03.2019	25.03.2019	2
3,3	Görsel tasarımın ilkelerinin belirlenmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	25.03.2019	26.03.2019	2
3,4	4. Aşamaya hazırlık ve 3. aşamanın gözden geçirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	27.03.2019	27.03.2019	1
4	Kodlamaya başlama	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	28.03.2019	30.03.2019	3
4,1	Android programlama	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	31.03.2019	20.05.2019	51
4,2	Veritabanı programlama	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	31.03.2019	20.05.2019	51
4,3	Görsel tasarım	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	31.03.2019	20.05.2019	51
4,4	5. Aşamaya Hazırlık ve 4. aşamanın gözden geçirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	20.05.2019	21.05.2019	2
5	Test ilkelerinin belirlenmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	21.05.2019	22.05.2019	2
5,1	Sistem testi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	22.05.2019	29.05.2019	7
5,2	Senaryoların test edilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	30.05.2019	31.05.2019	2
5,3	6. Aşamaya hazırlık ve 5. aşamanın gözden geçirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	31.05.2019	31.05.2019	1
6	Sunuma hazırlık ve sunum dosyası oluşturma	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	1.06.2019	1.06.2019	1
6,1	Projenin API haline getirilmesi	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	2.06.2019	2.06.2019	1
6,2	Jüri Sunumu	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	3.06.2019	3.06.2019	1
6,3	Uygulamanın Google Play Store’da yayınlanması ve bitiş	Ahmet Bedirhan SAĞIR, Mohammed ALRAWI	4.06.2019	4.06.2019	1

Resim 2: İş Paketi

KANINLA KURTAR ANDROID APP GANTT ŞEMASI



Resim 3: Gantt Şeması

Renk	Aşama
Yellow	Aşama 1: Sistem Tanımlama
Orange	Aşama 2: İhtiyaç Analizi
Green	Aşama 3: Tasarım
Blue	Aşama 4: Kodlama
Red	Aşama 5: Test
Cyan	Aşama 6: Sunum ve Yayınlama

Resim 4: Lejand

METOT, UYGULAMA VE TESTLER

METOT

Uygulamamızda Agile Software Development (Çevik Yazılım Geliştirme) metodu kullanılmıştır. Kısaca bu metottan bahsedelim.

Öncelikle bunun için Agile Manifesto'dan bahsetmemiz gerekiyor.

Çevik Yazılım Geliştirme Manifestosu

“Bizler daha iyi yazılım geliştirme yollarını uygulayarak ve başkalarının da uygulamasına yardım ederek ortaya çıkartıyoruz.

Bu çalışmaların sonucunda:

Süreçler ve araçlardan ziyade bireyler ve etkileşimlere
Kapsamlı dokümantasyondan ziyade çalışan yazılıma
Sözleşme pazarlıklarından ziyade müşteri ile iş birliğine Bir
plana bağlı kalmaktan ziyade değişime karşılık vermeye
değer vermeye kanaat getirdik.

Özetle, sol taraftaki maddelerin değerini kabul etmekle birlikte, sağ taraftaki maddeleri daha değerli bulmaktayız.” (Agile Manifesto 2019)

Çevik, hepsini mümkün olan en kısa sürede vermeye çalışmak yerine, artan teslimat, takım iş birliği, sürekli planlama ve sürekli öğrenmeyi vurgulayan yazılım geliştirme yaklaşımlarını tanımlamak için kullanılan bir terimdir.

Agile, süreci basit tutmaya ve bir şey sona ermeden birkaç yinelemeden geçen minimum uygulanabilir ürünler (MVP'ler) oluşturmaya odaklanır. Geribildirim sürekli toplanır ve uygulanır ve herkesin bir amaç için birlikte çalıştığı çok daha dinamik bir süreçtir.

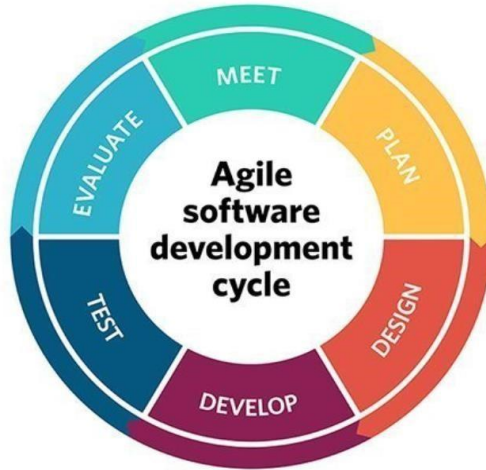
Çevik Manifesto'nun on iki ilkesi

2001'de hafif yazılım geliştirme kavramlarını tartışmak için 17 yazılım geliştirme uzmanı bir araya geldi ve Çevik Manifesto'yu yarattı. Manifesto, Çevik'in temel değerlerini ana hatlarıyla belirtir ve Manifesto'nun yararlılığını artırıp artırmadığı konusunda tartışma olsa da, Çevik hareketin özünde olmaya devam eder.

Manifesto, süreçlerden ve araçlardan ziyade insanlara ve iletişime önem veren, devrimci bir kavramdır. Manifesto'nun diğer önemli hususları arasında müşterileriyle doğrudan çalışmak, işleri doğrudan küçük parçalara bölmek, işleri düzenli tutmak ve işlemin sonunda bile değişikliklere açık olmak için günlük toplantı yapmak yer alıyor.

Çevik metodolojilerin çeşitleri

Herhangi bir Çevik ortamda, kullanılmakta olan birçok Çevik yöntem vardır. En eskilerinden biri, başarılı bir gelişim için testlerin düzenli olarak yapılması gerektiği fikrine dayanarak aşırı programlamadır. Çoğu durumda, testler koddan önce yazılmalıdır.



Resim 5: Çevik Yazılım Geliştirme Döngüsü

Avantajları

Yıllar boyunca Çevik ve Şelale yaklaşımları birçok şeyi karşılaştırdı.

Yazılım geliştirme Şelale döneminde, kodlayıcılar, yazılım test edicilere teslim edilmeden önce ve daha sonra üretime geçmeden, az girdili olsun veya olmasın, yalnız çalıştı. Hatalar, komplikasyonlar ve özellik değişiklikleri iyi bir şekilde ele alınmadı veya proje ciddi bir şekilde geciktiğinde veya hurdaya ayrılırken bu süreçte çok geç ele alındı.

İş tarafı da dahil olmak üzere geliştirme sürecine katılan herkesin arkasındaki Çevik model fikri, hem kültürde hem de bir şirketin daha hızlı pazarlama için daha iyi yazılım elde etme kabiliyetinde köklü bir değişime neden oldu.

İşbirliği ve iletişim teknoloji kadar önemli hale geldi ve Çevik Manifesto yorumlamaya açık olduğundan, Çevik her boyutta ve türde kuruluşlara uyacak şekilde uyarlandı ve uyarlandı. Çevik kültürel değişim aynı zamanda, yazılım geliştirmenin en son evrimi olan DevOps'a da yol açtı.

Dezavantajları

Birçoğu, Çevik'in en büyük dezavantajının, değiştirildikleri gerçeği olduğunu söyleyecektir - bazıları, çoğu kurum tarafından sulandırıldıklarını söyleyebilir. Bu fenomen o kadar yaygındır ki, "Çevik yol" uygulayıcıları "Örgütümüzde Scrum" yaptığımız gibi "Scrumbuts" olarak bilinir.

Agile, geliştiricilerle iş dünyası arasındaki iletişim hatlarını açmış olmasına rağmen, test ve süreçleri bu karışıma getirmek daha az başarılı oldu - DevOps fikrinin daha da artırılmasına yardımcı olabilecek bir ihmal.

Agile ile ilgili bir başka potansiyel endişe de, teknolojiyi vurgulamaması, kavramın yazılım geliştirmedeki rolünü anlamayan kıdemli yöneticilere satılmasını zorlaştırmasıdır.

UYGULAMA

Uygulamamız temel olarak üç adımdan oluşmaktadır. Bunlardan ilki sisteme üye olmaktır. İkincisi giriş yapmaktır. Sonuncusu ise ilan vermektir.

Potansiyel bir kullanıcı sistemin kendisinden istediği bilgileri kullanarak sisteme üye olur. Bu bilgiler sırasıyla; TC Kimlik No, Ad, Soyad, Yaşadığı il, Kan grubu, Telefon numarası, E-Posta adresi ve sisteme giriş şifresidir.

İlanların görüntülenebilmesi için kullanıcının yaşadığı ili doğru olarak girmesi gerekir.

Üyelik işlemi başarıyla tamamlandığında, potansiyel kullanıcı gerçek kullanıcı olur. Gerçek bir kullanıcı sisteme e-posta adresi ve daha önce belirlediği şifresiyle sisteme giriş yapar.

Sisteme giren bir kullanıcı, ilan ekleyebilir, verdiği ilanları görebilir, verilmiş ilanlara erişebilir ve bilgilerini güncelleyebilir.

İlan vermek için kişi öncelikle isim ve soy ismini, hastanın olduğu ili, hastanın yattığı hastaneyi, iletişim için telefon numarasını ve ilan detaylarını girmek zorundadır. İlan verildiğinde, ilan verilen il, herhangi bir kullanıcının iliyle aynıysa, sistem o kullanıcıya bir bildirim göndermektedir.

Bildirimi alan kullanıcı, eğer bağış yapacaksa, “BEN DE VARIM” butonuna basmalıdır. Bu butona basıldığında, kan/kan ürünü verme şartlarını sağlayıp sağlamadığına dair bir sorgulamaya tabi tutulur. Eğer şartlar sağlanıyorsa, ilgili kullanıcı ihtiyaç sahibine bağış yapabilir.

Kullanıcı bildirimleri kullanıcının iline ve ilanın iline göre verilir.

Örnek vermek gerekirse;

```
if(user.il==ilan.il){  
  sendnotification();  
}
```

Şeklinde bir somutlama yapılabilir.

Notification bildirimi sistem mesajı şeklinde verilir ve kişi bu mesaja tıklarsa, uygulamada sadece ilan gösterme kısmı açılır.

İlanı gören kişi, BEN DE VARIM butonuna basarak, karşısına çıkacak olan sorgulama sayfasındaki 4 soruyu cevaplamak zorundadır. Bu sorular;

- 1-) Sürekli kullandığınız ilaç var mıdır?
- 2-) Alkol alıyor musunuz?
- 3-) Hepatit, Aids vb. bulaşıcı hastalığınız var mı?
- 4-) Kalp, tansiyon benzeri hastalığınız var mı?

Eğer kullanıcı bu 4 sorunun tamamına “HAYIR” cevabı vermişse, kan bağışı yapabilirsiniz şeklinde bir toast mesajı alır ve ilan yeniden gösterilir. Sorulardan en az birisine bile “EVET” denilmişse, kişi kan bağışçısı olamaz.

İlanlar kısmında ise kullanıcının şehrindeki verilmiş olunan tüm ilanlar gösterilir.

TESTLER

Uygulama geliştirme aşamasında sürekli olarak uygulamanın doğru olarak çalışıp çalışmadığına dair, gerçek kullanıcı senaryolu testler yapılmıştır. Testler sırasında ortaya çıkan hatalar hemen giderilerek, uygulamanın işlerliği garanti altına alınmıştır.

Senaryo 1: Üyelik

Herhangi bir kişi üyelik formunu doldurmalıdır.

Amaç: Üyelik bilgileri veritabanına kaydedilerek, üyelik işleminin sağlanması

Yapılanlar: Ahmet Bedirhan SAĞIR ve Mohammed ALRAWI adlı kullanıcılar aynı anda kendilerine ait bilgileri girmişlerdir.

Sonuç: Her iki kullanıcı da sisteme üye olmuşlardır.

Senaryo 2: Sisteme Giriş

Herhangi bir kişi e posta ve şifresinde oluşan giriş formunu doldurmalıdır.

Amaç: e-posta ve şifreyle sisteme giriş yapılması

Yapılanlar: Ahmet Bedirhan SAĞIR ve Mohammed ALRAWI adlı kullanıcılar kendi e postaları ve şifrelerini giriş paneline girmiştir

Sonuç: Her iki kullanıcı da sisteme giriş yapmıştır.

Senaryo 3: İlan Gönderme

Herhangi bir kişi ilan vermelidir

Amaç: Kan veya kan ürünü ihtiyacının ilan edilmesi

Yapılanlar: Ahmet Bedirhan SAĞIR ve Mohammed ALRAWI adlı kullanıcılar ihtiyaç duydukları kan veya kan ürününe ait ihtiyaç ilan formunu doldurmuşlardır

Sonuç: Her iki kullanıcının ilanları veritabanına kaydedilmiştir ama bildirim yapılamamıştır.

GELECEKTE YAPILACAK İŞLER

Gelecekte; hastane ile kullanıcının bulunduğu yer arasında navigasyon sağlanacak, anlık konum bilgisi kullanılacak ve menü türü navigation drawer menü yapılacaktır. Bu üç işin yapılmasından sonra uygulamamızın Android versiyonu tamamlanmış olacaktır. Bundan sonra ise sadece uygulamayı işler halde tutacağız.

Kullanılan veritabanı türünü değiştirebiliriz. Uygulamamızın İOS (İphone/İpad) sürümü ise ilerleyen zamanlarda yapılacaktır.

KAN BAĞIŞI HAKKINDA BAZI BİLGİLER

Kan, sıvı (plazma) ve çeşitli hücre tiplerinden oluşur. Ortalama bir erkeğin vücudunda yaklaşık 5-6 litre kan vardır; Bir kadının biraz daha azına sahip. Kanın birçok farklı işlevi vardır - aşağıda detaylandırılmıştır.

Kan nerede bulunur?

Kan damarlarında bulunur. Kan damarları (atardamarlar, atardamarlar, kılcal damarlar, venüller ve damarlar) vücudunuzun her yerinden ve kandan kan alır. Kan, kan damarlarından kalbiniz tarafından pompalanır.

Normal kan neyi içerir?

Kan hücreleri

Bunlar bir mikroskop altında görülebilir ve kan hacminin yaklaşık% 40'ını oluşturur.

Kan hücreleri üç ana tipe ayrılır:

Kırmızı hücreler (eritrositler). Kanı kırmızı yaparlar. Bir damla kan, yaklaşık beş milyon kırmızı hücre içerir. Bozulmuş eski hücrelerin yerine geçmek için sürekli yeni bir kırmızı kan hücresi sağlanmalıdır. Her gün milyonlarca kırmızı kan hücresi yapılır. Kırmızı hücreler hemoglobin denilen bir kimyasal içerir. Bu oksijene bağlanır ve oksijeni akciğerlerden vücudun tüm kısımlarına alır.

Beyaz hücreler (lökositler). Nötrofiller (polimorflar), lenfositler, eozinofiller, monositler ve bazofiller adı verilen farklı beyaz hücreler vardır. Bağışıklık sisteminin bir parçası. Başlıca rolü vücudu enfeksiyonlara karşı korumaktır. Nötrofiller mikropları (bakterileri) tutar ve özel kimyasallarla imha eder. Eozinofiller ve monositler ayrıca vücuttaki yabancı parçacıkları yutarak da çalışırlar. Bazofiller iltihabı yoğunlaştırmaya yardımcı olur. Enflamasyon kan damarlarını sızdırıyor. Bu, uzman beyaz kan hücrelerinin ihtiyaç duyulan yere ulaşmalarına yardımcı olur. Lenfositlerin çeşitli farklı işlevleri vardır. Virüslere ve diğer mikroplara (patojenler) saldırırlar. Ayrıca patojenleri yok etmeye yardımcı olan antikorlar yaparlar.

Trombositler. Bunlar küçük ve eğer kendimizi kesersek, kanın pıhtılaşmasına yardımcı olabilirler.

Plazma

Bu kanın sıvı kısmıdır ve kan hacminin yaklaşık% 60'ını oluşturur. Plazma esas olarak sudan yapılır, ancak aynı zamanda birçok farklı protein ve başka kimyasallar da içerir:

Hormonlar.

Antikorlar. enzimler.

Glikoz.

Yağ parçacıkları Tuzlar.

Vücudunuzdan kan döküldüğünde (veya kan örneği düz cam tüpe alındığında), hücreler ve bazı plazma proteinleri bir pıhtı oluşturmak üzere bir araya toplanır. Kalan berrak sıvıya serum denir.

Kan ne yapar?

Kanın çeşitli işlevleri vardır. Bunlar şunları içerir:

Taşıma. Kan, akciğerlerden oksijeni vücudun hücrelerine çeker. Karbondioksiti vücudun hücrelerinden solup kaldığı ciğerlere götürür. Kan vücuttaki besinleri, hormonları ve atık ürünleri taşır.

Yönetmelikler. Kan vücudun asit-alkali dengesini kontrol altında tutmaya yardımcı olur. Aynı zamanda vücut ısısının düzenlenmesinde rol oynar. Cilde yakın akan kan miktarını artırmak, vücudun ısınıyı kaybetmesine yardımcı olur.

Koruma. Beyaz kan hücreleri istilacı mikroplara (bakteri) ve diğer patojenlere saldırır ve tahrip eder. Yaralanma sonrasında vücudun çok fazla kan kaybetmesini önleyen kan pıhtıları.

Kan grupları

Dört ana kan grubu vardır (kan grubu) - A, B, AB ve O. Kan grubunuz, ailenizden devraldığınız genler tarafından belirlenir.

Her grup Rh pozitif veya Rh negatif olabilir, yani toplamda sekiz ana kan grubu vardır.

Antikorlar ve antijenler

Kan, kırmızı kan hücreleri, beyaz kan hücreleri ve trombosit adı verilen bir sıvıdaki plazmadan oluşur. Kan grubunuz, kandaki antikorlar ve antijenler tarafından tanımlanır.

Antikorlar, plazmada bulunan proteinlerdir. Vücudun doğal savunmasının bir parçası. Mikroplar gibi yabancı maddeleri tanırlar ve onları yok eden bağışıklık sisteminizi uyarırlar.

Antijenler, kırmızı kan hücrelerinin yüzeyindeki protein molekülleridir.

ABO sistemi

ABO sistemi tarafından tanımlanan dört ana kan grubu vardır:

kan grubu A - A, plazmada anti-B antikorları olan kırmızı kan hücrelerinde antijenlere sahiptir.

kan grubu B - plazmada anti-A antikorları olan B antijenleridir

Kan grubu O - antijen içermez, fakat plazmada hem anti-A hem de anti-B antikorları vardır.

AB kan grubu - hem A hem de B antijenlerine sahiptir ancak antikorları yoktur

Kan grubu O en yaygın kan grubudur. İngiltere nüfusunun neredeyse yarısında (% 48) kan grubu O.

Yanlış ABO grubundan kan almak hayatı tehdit edici olabilir. Örneğin, A grubuna A grubuna kan verilirse, anti-A antikorları A grubu hücrelerine saldırarak.

Bu nedenle, A grubu kanı, B grubu kanı olan birine verilmemelidir ve bunun tersi de geçerlidir.

Kırmızı kan hücrelerinin grubu, A veya B antijenleri olmadığından, diğer herhangi bir gruba güvenle uygulanabilir.

NHS Kan ve Nakil (NHSBT) web sitesi farklı kan grupları hakkında daha fazla bilgiye sahiptir.

Rh sistemi

Kırmızı kan hücreleri, bazen RhD antijeni olarak bilinen bir antijene sahiptir. Bu mevcutsa, kan grubunuz Rh pozitif. Değilse, kan grubunuz Rh negatif.

Bu, sekiz kan grubundan biri olabileceğiniz anlamına gelir:

Rh pozitif (A +)

Rh negatif (A-)

B Rh-pozitif (B +)

B Rh negatif (B-)

O Rh pozitif (O +)

O Rh negatif (O-)

AB Rh pozitif (AB +)

AB Rh negatif (AB-)

Çoğu durumda, O Rh negatif kanı (O-) herkese güvenle uygulanabilir. Kan grubu hemen bilinmediğinde genellikle tıbbi acil durumlarda kullanılır.

Çoğu alıcı için güvenlidir çünkü hücrelerin yüzeyinde A, B veya Rh antijenleri yoktur ve diğer ABO ve Rh kan gruplarıyla uyumludur.

Kan uyum tablosu				
Alıcı Kan Grubu	Verebilen kan grupları			
AB Rh+	Tüm kan grupları			
AB Rh-	O Rh-	A Rh-	B Rh-	AB Rh-
A Rh+	O Rh-	O Rh+	A Rh-	A Rh+
A Rh-	A Rh-	A Rh+		
B Rh+	O Rh-	O Rh+	B Rh-	B Rh+
B Rh-	O Rh-	B Rh-		
O Rh+	O Rh-	O Rh+		
O Rh-	O Rh-			

Resim 6: Kan Grupları Tablosu

KAN VERME ŞARTLARI

Çoğu insan sağlık koşullarında kan verebilir. Kan bağışı için yerine getirilmesi gereken bazı temel şartlar vardır. İşte bazı temel uygunluk kuralları:

Yaş:

18 ile 65 yaş arasındasın.

* Bazı ülkelerde, ulusal yasalar, gerekli fiziksel ve hematolojik kriterleri yerine getirmeleri ve uygun onayları almaları şartıyla, 16 ila 17 yaş arasındaki çocukların bağış yapmalarına izin vermektedir.

* Bazı ülkelerde, 65 yıldan daha eski düzenli bağışlar sorumlu hekimin takdirine bağlı olarak kabul edilebilir. Bazı ülkelerde üst yaş sınırı 60'dır.

Ağırlık:

En az 50 kg ağırlığındasın.

* Bazı ülkelerde, tam kan bağışçıları 350 ml \pm % 10 bağış yapmak için en az 45 kg ağırlığında olmalıdır.

Sağlık:

Bağış yaparken sağlıklı olmalısın.

Soğuk algınlığı, grip, boğaz ağrısı, soğuk algınlığı, mide böceği veya başka bir enfeksiyon varsa bağış yapamazsınız.

Son zamanlarda dövmeleğiniz veya piercinginiz varsa, işlem tarihinden itibaren 6 ay boyunca bağıştta bulunamazsınız. Vücutta piercing kayıtlı bir sağık profesyoneli tarafından yapılır ve iltihaplar tamamen giderilirse, 12 saat sonra kan bağışı yapabilirsiniz.

Küçük bir işlem için diř hekimini ziyaret ettiyseniz, bağıştan önce 24 saat beklemelisiniz; Büyük işler için bir ay bekleyin.

Kan bağışı için minimum hemoglobin seviyesine uymazsanız

* Bağış yerinde bir test yapılacaktır. Birçok ülkede, hemoglobin düzeyleri kadınlar için 12.0 g / dl'den düşük, erkekler için 13.0 g / dl'den düşük değildir.

Seyahat:

Örneğın sıvrisineğ kaynaklı enfeksiyonların endemik olduğı bölgelere yolculuk; Sıtma, dang ve Zika virüsü enfeksiyonları geçici bir erteleme ile sonuçlanabilir.

Birçok ülke ayrıca, belirli ülkelerde veya bölgelerde tanımlanmış kümülatif maruz kalma süreleri için kan bağışçılarını seyahat veya ikamet tarihleriyle erteleme politikası uygulamaktadır; Varyant Creutzfeldt-Jakob Hastalığı'nın (vCJD) kan transfüzyonu yoluyla kontaminasyon riskini azaltmak için bir önlem olarak.

davranışlar:

Kan vermemelisin:

“Son 12 ayda cinsel ilişkide risk altındaysanız”

Aşağıdaki davranışlara sahip kişiler kalıcı olarak ertelenir:

Hiç HIV için pozitif test yaptınız mı?

Eğlenmek için hiç ilaç enjekte etmedin.

* Ulusal kan bağışı seçim rehberinde daha fazla davranış uygunluk kriteri vardır.
Kriterler farklı ülkelerde farklılık gösterebilir.

Hamilelik ve emzirme:

Hamilelikten sonra, gecikme süresi hamilelik süresi kadar sürmelidir.

Emzirme döneminde kan bağışı önerilmez. Doğumdan sonra erteleme süresi en az 9 aydır (hamilelikte olduğu gibi) ve bebeğinizin ciddi bir şekilde sütten kesilmesinden 3 ay sonra (yani beslenmesinin çoğunu katı maddelerden veya şişelerden beslemek)

KAYNAKÇA

2019. "Agile Manifesto." *Agile Manifesto*. Accessed Mayıs 21, 2019.
<https://agilemanifesto.org/iso/tr/manifesto.html>.
2017. "Bir Damla Umut." *Google Play*. Accessed Nisan 2019.
<https://play.google.com/store/apps/details?id=com.heromobile.birdamlaumut>.
2016. "Kan Bankası." *Google Play*. Accessed Mart 2019.
<https://play.google.com/store/apps/details?id=tr.com.besimvatan.mobilkanbankasi>.
2017. "Kan Lazım." *Google Play*. Eylül. Accessed Nisan 2019.
<https://play.google.com/store/apps/details?id=com.kanlazim.app>.
2019. "Kan Verme." *Medical Park Hastanesi*. Accessed Nisan 10, 2019.
<https://www.medicalpark.com.tr/kan-verme/hg-92>.
2019. "Stat Counter Türkiye Telefon Kullanım Bilgileri." *Stat Counter*. Accessed Nisan 15, 2019. <http://gs.statcounter.com/osmarketshare/mobile/turkey>.
2019. "Stat Counter Worldwide Smartphone infos." *Stat Counter*. Accessed Nisan 15, 2019.
<http://gs.statcounter.com/os-market-share/mobile/worldwide>.

EKLER

```
package com.example.bedirhan.kaninla_kurtar;

import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class StartActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT=3000; //3 saniyelik bekleme süresi tanımlandı

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent homeIntent = new Intent(StartActivity.this,
AnasayfaActivity.class);
                startActivity(homeIntent);
                finish();
            }
        }, SPLASH_TIME_OUT);
    }
}
```

```

package com.example.bedirhan.kaninla_kurtar;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.widget.Toolbar;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity {

    EditText Tc,Ad,Sad,Tel,İl,Kan,Mail,Pass;
    Button btn_reg;
    FirebaseAuth auth;
    DatabaseReference reference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        Tc =findViewById(R.id.Tc);
        Ad =findViewById(R.id.Ad);
        Sad =findViewById(R.id.Sad);
        Tel =findViewById(R.id.Tel);
        İl =findViewById(R.id.İl);
        Kan =findViewById(R.id.Kan);
        Mail =findViewById(R.id.Mail);
        Pass =findViewById(R.id.Pass);
        btn_reg =findViewById(R.id.btn_reg);
        auth = FirebaseAuth.getInstance();

        btn_reg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String txt_Tc= Tc.getText().toString();
                String txt_Ad= Ad.getText().toString();
                String txt_Sad= Sad.getText().toString();
                String txt_Tel= Tel.getText().toString();
                String txt_İl= İl.getText().toString();
            }
        });
    }
}

```



```

        String txt_Kan= Kan.getText().toString();
        String txt_Mail= Mail.getText().toString();
        String txt_Pass= Pass.getText().toString();

        if(TextUtils.isEmpty(txt_Tc)&&TextUtils.isEmpty(txt_Ad)&&TextUtils.isEmpty(txt_Sad)&&TextUtils.isEmpty(txt_Tel)&&TextUtils.isEmpty(txt_İl)&&TextUtils.isEmpty(txt_Kan)&&TextUtils.isEmpty(txt_Mail)&&TextUtils.isEmpty(txt_Pass)){
            Toast.makeText(getApplicationContext(),"Boş alanları doldur", Toast.LENGTH_LONG).show();
        }else if (txt_Pass.length()<6)
        {
            Toast.makeText(getApplicationContext(),"Parola en az 6 karakter olmalı", Toast.LENGTH_LONG).show();
        }else {
            register(txt_Tc,txt_Ad,txt_Sad,txt_Tel,txt_İl,txt_Kan,txt_Mail,txt_Pass);
        }
    }
}

//Vazgeç butonuna basıldığında, kayıt işleminden vazgeçilerek anasayfaya dönülür.
Button btn_canc = (Button)findViewById(R.id.btn_canc);
btn_canc.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent gerri = new Intent(RegisterActivity.this,AnasayfaActivity.class);
        startActivity(gerri);
    }
});

}

//Kullanıcı kayıt girişleri ve buton tanımlama
private void register(final String Tc, final String Ad, final String Sad, final String Tel, final String İl, final String Kan, final String Mail, final String Pass){
    auth.createUserWithEmailAndPassword(Mail,Pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(task.isSuccessful()){
                FirebaseUser firebaseUser = auth.getCurrentUser();
                assert firebaseUser !=null;
                String userid =firebaseUser.getId();
                reference = FirebaseDatabase.getInstance().getReference("Users").child(userid);
                HashMap<String,String> hashMap = new HashMap<>();
                hashMap.put("Id: ", userid);
                hashMap.put("Tc: ", Tc);
                hashMap.put("Ad: ", Ad);
                hashMap.put("Soyad: ", Sad);
                hashMap.put("Telefon: ", Tel);
                hashMap.put("İli : ", İl);
            }
        }
    });
}

```

```

        hashMap.put("Kan Grubu: ", Kan);
        hashMap.put("EMail: ", Mail);
        hashMap.put("Parola: ", Pass);

        reference.setValue(hashMap).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if(task.isSuccessful()){
                    Intent baslangic = new
Intent(RegisterActivity.this,AnasayfaActivity.class);
                    startActivity(baslangic);
                }
            }
        });
    }else {
        Toast.makeText(getApplicationContext(),"Kayıt Başarısız, Yeniden
dene", Toast.LENGTH_LONG).show();
    }
}
});
}
}
}

```

```

package com.example.bedirhan.kaninla_kurtar;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class LoginActivity extends AppCompatActivity {
    private EditText email,pass;
    Button btn giris ;
    FirebaseAuth auth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        auth =FirebaseAuth.getInstance();
        email =findViewById(R.id.email);
        pass =findViewById(R.id.pass);
        btn giris =findViewById(R.id.btn_giris);
        btn_giris.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String txt_EMail =email.getText().toString();
                String txt_Pass =pass.getText().toString();

                if(TextUtils.isEmpty(txt_EMail)||TextUtils.isEmpty(txt_Pass)){
                    Toast.makeText(LoginActivity.this,"Tüm Alanları
Doldurunuz",Toast.LENGTH_LONG).show();
                }else {
                    auth.signInWithEmailAndPassword(txt_EMail,txt_Pass)
                        .addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                            @Override
                            public void onComplete(@NonNull Task<AuthResult>
task) {
                                if(task.isSuccessful()){
                                    Intent intentt = new
Intent(LoginActivity.this,Menu KullaniciActivity.class);
                                    startActivity(intentt);
                                }else {
                                    Toast.makeText(LoginActivity.this,"Giriş
Hatası",Toast.LENGTH_LONG).show();
                                }
                            }
                        });
                }
            }
        });

        Button btn_geri2 =(Button) findViewById(R.id.btn_geri2);
        btn_geri2.setOnClickListener(new View.OnClickListener() {

```

```
        @Override
        public void onClick(View v) {
            Intent geri2 = new
Intent(LoginActivity.this, AnasayfaActivity.class);
            startActivity(geri2);
        }
    });
}
}
```