



İSTİNYE ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ 2
FİNAL RAPORU

Mayıs 2022

PROJE BAŞLIĞI

MAKİNE ÖĞRENMESİ TAHMİNLERİNİ KULLANAN WEB SERVİSLERİ İÇİN
YAZILIM ALTYAPILARININ GELİŞTİRİLMESİ, SPOR ENDÜSTRİSİ ANALİTİKLERİ
ÜZERİNE BİR UYGULAMA

PROJE YAZARI

ALPERAN ÇAKAR 180701017
GÖKSEL YILMAZ 180701012

DANIŞMAN

DR. ÖĞR. ÜYESİ GÜRSAN ÇOBAN

ÖZET

Spor endüstrisinde teknolojinin kullanımı yıllardan beri süre gelen yatırımlar, araştırmalar ve yapılanmalar sonucu gün geçtikçe daha güvenilir ve daha hızlı sonuç veren bir yapı haline gelmektedir. Her geçen gün yeni yapılanmalar, AR-GE çalışmaları ve girişimler artmaktadır. Yeni hesaplama sistemleri, yeni istatistikler, yeni kavramlar ve daha optimal sonuçlar elde edilen bu oluşumlara yeni bir yaklaşım da bu projeyle ulaşılması hedeflenmektedir.

Yapılan bu proje, günümüz futbol severlerine, yorumculara, gazetecilere, vb. kategorilerdeki topluluklara her gün dünya üzerinde oynanacak maçlardan seçilen bir maç grubu için çeşitli tahmin veya tahminleri kullanıcılara çeşitli kategoriler altında servis eden bir mobil uygulamadır. Bu tahminlerin dünya üzerinde aktif olarak en çok kitleyi barındıran futbol organizasyonunun programlarında kullanılması hedeflenmektedir. Bu büyük organizasyon belirli bir kesim ya da sınır içerisinde kalmadığı için uygulamanın hedef kitlesi de sınırlanmamaktadır. Tüm dünyaya hitap eden bu uygulama sayesinde insanlar, varsa tuttuğu takımların kazanabilme durumlarının istatistiksel veriler ışığında tahminine ulaşabileceklerdir. Verilen bu tahminler analiz ekipleri gibi insan gücü ve öznel yargılardan ziyade yapay zeka analiz sistemini de içinde barındırarak istatistiksel ve matematiksel kavramlar çerçevesince değerlendirilip kullanıcıya sunulacaktır.

Bu uygulamanın oluşturulması için yazılım aracı olarak Android Studio kullanılmıştır. Android Studio, Android uygulamaları geliştirmek için özel olarak tasarlanmış bir mobil uygulama geliştirme platformudur. Android Studio, akıllı bir kod düzenleyici ve hata ayıklayıcı, performans analiz araçları, emülatörler ve daha pek çok araç dahil olmak üzere uygulama geliştirmek için kullanıcıların ihtiyacını karşılayabilecek birçok özellik barındıran bir geliştirme ortamıdır. Bu geliştirme ortamında programlama dili olarak Java kullanılmıştır. Java, herhangi bir bilgisayar mimarisine bağlı olmadan ve platformlardan bağımsız olarak çalışabilen bir programdır. Uygulama içerisinde yapılan tüm işlerin arka plan komutlarının çalıştırılmasını üstlenir.

Yapay zeka sistemi ile ortaya ıkarılacak tahmin iin dinamik olarak hareket eden ve ok ynl veriler ieren bir API kullanılması gerekmektedir. Bu noktada “RapidApi” isimli API veri hizmeti kullanılacaktır. RapidApi pek ok kategoriden belirli API'leri bnyesinde barındıran bir veri hizmet platformudur. Burada bulunan ve futbol kategorisinde pek ok platforma hizmet vermekte olan “API-FOOTBALL” API'si kullanılacaktır. BU API sayesinde yapay zeka sistemine ilenmek zere kullanılabilir her trl oyuncu, takım, lig ve organizasyon hakkında dinamik ekilde gncellenen ve pek ok kategoriden ierik bulunduran bu havuza ulamak mmkn olacaktır. Bu toplanan veriler web servis hizmeti sunan “Gzel Hosting” sayesinde bir veri tabanında tutulacak ve takibinde AI sisteminde analiz edilmek zere buradan tahsis edilecektir.

Geliştirilen uygulama insanların giriş yapabilmesi ve ye olabilmesi iin bir giriş ekranı bulundurmaktadır. Burada bulunan giriş butonu ile Google Signin hizmeti kullanılarak Google hesabı ile giriş yapılabilir. ye olan kullanıcılar oturum atıkları zaman ana sayfaya ynlendirilmektedir. Burada daha nce yapılan tahminler ekranı ve sonuları bulunmaktadır. Geiş saėlanabilen bu ekrandan farklı kategorilere gemek mmkndr. “Free” isimli kategori tm kullancıların erişimine aıktır. Erişime kapalı olan kategoriler iin yeni yelik oluturmak gerekmektedir. Tm bu kategorilerin altında birok ma iin tahminler bulunmaktadır. Bu tahminler malar sonulandıktan sonra kategorilerin altından kaldırılmakta ve ana sayfada bulunan sonular ekranına getirilmektedir. Daha nce yayınlanan tahminler ve maların sonuları karılaştırılmakta, sonucunda kullancılara baarılı tahminler retilip retilmediėi gsterilmektedir.

İÇİNDEKİLER

ÖZET.....	3
FİĞÜR LİSTESİ	6
GİRİŞ	8
LİTERATÜR TARAMASI.....	12
METOT, UYGULAMA VE TESTLER	17
YAZILIM PLATFORMUNUN GELİŞTİRİLMESİ VE UYGULAMA	39
SONUÇ	66
KAYNAKÇA.....	67

FIGÜR LİSTESİ

Şekil 1 Android Studio Anasayfası [47]	18
Şekil 2 act_login.xml kodları	39
Şekil 3 act_login.xml tasarım ekranı	40
Şekil 4 onUpdateNeeded isimli fonksiyonun kodu	41
Şekil 5 showMyCustomAlertDialog fonksiyonu kodu	42
Şekil 6 Sözleşme kabulü uyarı ekranı	42
Şekil 7 girisCheck isimli fonksiyonun kodu	43
Şekil 8 handleResultGoogleSignIn fonksiyonu kodları	44
Şekil 9 Oturumdan çıkış ekranı	45
Şekil 10 onNavigationItemSelected isimli fonksiyonun kodları.....	46
Şekil 11 Menü ekranı	46
Şekil 12 act_contact.xml kodları	47
Şekil 13 İletişim ekranı	48
Şekil 14 act_contact.xml sayfasında bulunan buton kodları	48
Şekil 15 act_howtouse.xml kodları	49
Şekil 16 Uygulama nasıl kullanılır ekranı	50
Şekil 17 Howtouse sınıfında bulunan tanımla isimli fonksiyonun kodları	50
Şekil 18 act_terms.xml kodları.....	51
Şekil 19 Şartlar ve koşullar ekranı	52
Şekil 20 Terms sınıfında bulunan tanımla isimli fonksiyonun kodları	52
Şekil 21 act_privacy.xml kodları	53
Şekil 22 Gizlilik politikası ekranı.....	54
Şekil 23 Privacy sınıfında bulunan tanımla isimli fonksiyonun kodları	54
Şekil 24 frag_proof.xml kodları	55
Şekil 25 Sonuçlanan maçlar ekranı	56
Şekil 26 frag_free.xml kodları	57
Şekil 27 Free kategorisi maçları ekranı	57
Şekil 28 frag_vip.xml kodları.....	58

Şekil 29 Vip kategorisi maçları ekranı	59
Şekil 30 vip kategorisi üyelik ekranı	62
Şekil 31 Maç girişi ekranı	64
Şekil 32 Maç ekleme ekranı	64
Şekil 33 Üye soruları ekranı	65
Şekil 34 Üye yönetim ekranı	65
Şekil 35 Kategori üyelikleri yönetim ekranı	65
Şekil 36 Kayan yazı düzenleme ekranı	65
Şekil 37 Ülke bayrakları düzenleme ekranı	66

GİRİŞ

Bu çalışmada makine öğrenmesi tahminleri yardımıyla web servisleri ve spor endüstrisi için geliştirilen bir yazılım aracının, geliştirme sürecinin anlatımını da içeren analiz ve uygulamalar bulunmaktadır. Spor endüstrisinin içerisinde bulunan tüm kurum, kuruluş ve bireylerin yaşadığı analizlere dayalı tahminlemede bulunan, iş gücünün çokluğu ve ortaya çıkan analizlerin daha çok insan yorumlamasına dayalı olmasından kaynaklı eksik yapıların bulunması, seçilen problem olarak belirlenmiştir. Yapılan tüm analiz sürecinde iş gücünü minimuma indirerek paradan ve zamandan tasarruf edilmesi, daha optimum sonuçlar elde edilmesi adına bu proje seçilmiştir. Tüm bu organizasyon için yapay zeka destekli analiz süreci işlemektedir. Yapay zeka ile daha önce belirlenen parametreler doğrultusunda çıktılar üretilir. Üretilen bu çıktılar belirli kategoriler altında kullanıcılara uygulama üzerinden servis edilir. Kullanılan yapay zeka, süreç devam ettikçe kendini geliştirecek bir yapıda kurulacak ve her bir yeni öğretilde daha optimum sonuçlar çıkartacaktır. Çalışmanın yazılım çıktılarını kullanan kurum, kuruluş ve bireylerin paradan, zamandan ve iş yükünden tasarruf etmesi beklenmektedir. Her gün gelişen ve öğrenen bir sistem ile kullanıcılara daha optimum sonuçlar içeren servis hizmeti sağlanması mümkün olacaktır.

Yapay zekanın alt kümesi olarak kabul edilen makine öğrenmesi (ML), bir bilgisayarın doğrudan yönergeler olmadan öğrenmesine yardımcı olan matematiksel modelleri kullanma işlemidir. Tahmine dayalı analiz türü olan makine öğrenmesi, tahmin yapabilen bir veri modeli oluşturmak için belli kalıplar kullanır. Bu kalıpların belirlenmesi için algoritmalar kullanılır. Tahmin süreci boyunca sistemin girdi olarak aldığı veri sayısı ve deneyim arttıkça makine öğrenmesi sonuçları da daha doğru sonuçlar üretir, hata payı minimuma indirilir. Makine öğrenmesi, tahmine dayalı bir analiz türüdür. Ancak daha fazla veri aldıkça gerçek zamanlı olarak güncelleştirilen makine öğrenmesini uygulamak çok daha kolaydır. Tahmine dayalı analiz, genellikle statik bir veri kümesi ile çalışır ve güncelleştirmeler için yenilenmesi gerekir. Sürekli yenilenme sayesinde sonuçlar da aynı oranda iyileşmektedir. [43]

“Makine öğrenme sistemleri yeni olarak düşünülse de, kökenleri İkinci Dünya Savaşına, 1950'lere uzanır. 1950'de Alan Turing, Turing Testini bilgisayarın bir insanı geçip geçmeyeceğini görmek için tasarlamıştır. Reading Üniversitesi'nde Eugene adında bir chatbot, Turing testini geçmiştir. Böylece öğrenme kabiliyetine sahip ilk sinir ağları 1950 yıllarında inşa

edilmiştir. 1967'de K-en yakın komşuları algoritması (KNN) ile makine öğrenmesi oluşturulmuştur. Araştırmacılar 1980 ve 1990'larda beynimizin yapısından esinlenerek bir tür makine öğrenme modeli olan yapay sinir ağlarını eğitmek için yeni bir algoritma sınıfı daha oluşturmuştur. 1950'li dönemlerden bu yana yapay zeka teknolojilerinde hızlı bir ilerleme görülememiş lakin zaman geçtikçe, gelişen bilgisayarlar ile birlikte yapay zeka teknolojileri de gelişmiştir. Son yirmi yılda veri kullanılabilirliği bin kata, temel algoritmaların gelişmişlik seviyesi yüz kata ve donanım hızı en az yüz kat kadar artmıştır. Günümüzde veri miktarındaki artış, çok geliştirilmiş algoritmalar ve daha güçlü bilgisayar donanımları yapay zeka teknolojilerinin önünü epey açmıştır.” [44]

Özünde, tahmine dayalı analitik, çeşitli istatistiksel teknikleri (makine öğrenimi, tahmine dayalı modelleme ve veri madenciliği dahil) kapsamaktadır ve gelecekteki sonuçları tahmin etmek için istatistikleri (hem geçmiş hem de güncel) kullanmaktadır. Tahmine dayalı analitik, geçmiş analiz ederek gelecekteki olası olayları anlamamıza yardımcı olur. Makine öğrenimi ise, Arthur Samuel'in 1959'daki tanımına göre, 'bilgisayarlara açıkça programlanmadan öğrenme yeteneği' veren bilgisayar biliminin bir alt alanıdır. Makine öğrenimi, örüntü tanıma çalışmasından gelişen, algoritmaların verilerden öğrenebileceği ve veriler üzerinde tahminlerde bulunabileceği fikrini araştırmaktadır. Tahmine dayalı analitik, tahmine dayalı modelleme tarafından yönlendirilir. Bu bir süreçten çok bir yaklaşımdır. Tahmine dayalı analitik ve makine öğrenimi el ele gider, çünkü tahmine dayalı modeller tipik olarak bir makine öğrenimi algoritması içerir. Bu modeller, yeni verilere veya değerlere yanıt vermek için zamanla eğitilebilir ve işletmenin ihtiyaç duyduğu sonuçları sağlayabilir. İki tür tahmin modeli vardır. Bunlar, sınıf üyeliğini tahmin eden Sınıflandırma modelleri ve bir sayıyı tahmin eden Regresyon modelleridir. Bu modeller daha sonra algoritmalarından oluşur. Algoritmalar, verilerdeki eğilimleri ve kalıpları belirleyerek veri madenciliği ve istatistiksel analiz gerçekleştirir. Tahmine dayalı analitik yazılım çözümleri, tahmine dayalı modeller oluşturmak için kullanılacak yerleşik algoritmalara sahip olacaktır. Algoritmalar, verilerin hangi kategorilere ait olduğunu belirleyen 'sınıflandırıcılar' olarak tanımlanır. En yaygın olarak kullanılan tahmin modelleri, Karar ağaçları, Regresyon (doğrusal ve lojistik), Sinir ağlarıdır. Karar ağaçları, çok değişkenli analizin basit ama güçlü bir şeklidir. Verileri dal benzeri bölümlere ayırmanın çeşitli yollarını tanımlayan algoritmalar tarafından üretilirler. Karar ağaçları, verileri girdi değişkenlerinin kategorilerine göre alt kümelere bölerek birinin karar yolunu anlamamıza yardımcı olur. Regresyon, istatistikte en popüler yöntemlerden biridir. Regresyon analizi, değişkenler arasındaki ilişkileri tahmin eder, büyük ve çeşitli veri

kümelerinde anahtar kalıpları ve bunların birbirleriyle nasıl ilişkili olduğunu bulur. Sinir ağları, İnsan beynindeki nöronların çalışmasından sonra modellenen sinir ağları (yapay sinir ağları olarak da adlandırılır), çeşitli derin öğrenme teknolojileridir. Genellikle karmaşık örüntü tanıma sorunlarını çözmek için kullanılırlar ve büyük veri kümelerini analiz etmek için inanılmaz derecede faydalıdır. Verilerdeki doğrusal olmayan ilişkileri yönetmede harikadırlar ve belirli değişkenler bilinmediğinde iyi çalışırlar. Bunların haricinde diğer sınıflandırıcılar ise Zaman Serisi Algoritmaları, Kümeleme Algoritmaları, Aykırı Değer Algılama Algoritmaları, Topluluk Modelleri, Faktör Analizi, Naive Bayes sınıflandırıcısı ve Destek Vektör Makineleri olarak sıralanmaktadır. Zaman Serisi Algoritmaları, verileri sırayla çizer ve zaman içinde sürekli değerleri tahmin etmek için kullanışlıdır. Kümeleme algoritmaları, verileri benzer üyeleri olan gruplar halinde düzenler. Aykırı değer algılama algoritmaları, bir veri kümesi içinde beklenen bir modele veya standarda uymayan öğeleri, olayları veya gözlemleri belirleyerek anormallik algılamaya odaklanır. Topluluk modelleri, birden çok algoritmadan elde edilebilecek daha iyi tahmin performansı için makine öğrenimi algoritmaları kullanılmasıdır. Faktör analizi, değişkenliği tanımlamak için kullanılan ve bağımsız değişkenleri bulmayı amaçlayan bir yöntemdir. Naive Bayes sınıflandırıcısı, olasılığı kullanarak belirli bir özellik kümesine dayalı bir sınıf/kategori tahmin etmeyi sağlamak amacıyla kullanılır. Destek vektör makineleri, verileri analiz etmek ve kalıpları tanımak için ilişkili öğrenme algoritmalarını kullanan bir makine öğrenimi tekniğidir. Görüldüğü üzere her sınıflandırıcı verilere farklı bir şekilde yaklaşır, bu nedenle kuruluşların ihtiyaç duydukları sonuçları alabilmeleri için doğru sınıflandırıcıları ve modelleri seçilmesi gerekir. [45]

Yapay zeka sistemleri, çok daha fazla erişilebilir veriye ilişkin daha kapsamlı bir anlayış sunması, daha önce insan gücünü gerekli kılan süreçleri veya görevleri otomatikleştirerek kurumsal performansı ve üretkenliği artırması, güvenlik ihlallerini tespit etmesi ve engellemesi, kullanıcıların teknolojiye ilişkin sorunlarını çözmesi, üretim yönetimi çalışmalarını azaltması, uygun fiyatlı, yüksek performanslı bilgi işlem özelliğine kolayca erişilebilmesi, eğitim için yüksek miktarda veriye erişebilmesi, doğru ve etkin kullanımı sonucunda rekabette avantaj sağlaması gibi pek çok avantajı ve özelliğinden kaynaklı yoğun bir şekilde hemen hemen her sektörde kullanılır. [46]

Spor endüstrisinde birçok alanda yazılım kullanılmaktadır. Yapay zeka hakemleri, antrenman planlayıcılar, oyuncu performanslarının analizi ve benzeri konular spor endüstrisinde yazılımın kullanıldığı yerlerdir. Yazılım ve spor endüstrisini bir bütün halde

tanımlamak mümkündür. Spor müsabakalarının yayınları, tüm spor kurum ve kuruluşlarının web siteleri yazılım tabanlıdır. Bu durum, spor endüstrisinin gelişiminde büyük rol oynamıştır. Bu sayede tüm spor müsabakaları dünyaya ulaşabilmekte, insanların müsabakaları takip edebilmesi ve bu müsabakalar hakkında yorum yapabilmelerini mümkün kılmaktadır. Yazılım sayesinde spor endüstrisi ağırlığını televizyondan sosyal medyaya çekmiştir. Bu durum da insanlara ulaşma hızını arttırmıştır. Sonuç olarak spor endüstrisi yazılımı efektif olarak kullanmış ve kullanmaya devam edecektir.

Teknolojinin ve dijitalleşmenin geliştiği günümüz Dünyasında hızla büyüme gösteren piyasalardan biri de mobil piyasasıdır. Akıllı telefon kullanımının artması (Android/IOS) ve bu platformlara sürekli yeni içerik üretilmesiyle beraber insanlar internette market alışverişi gibi gündelik ve banka ödemeleri gibi zorunlu ihtiyaçlarını dijital yönlerden gidermektedir. Bunun pek çok sebebi vardır. Bunlardan en önemlisi zamandan ve enerjiden tasarruf etme anlayışı ve arayışıdır. Bu doğrultuda her geçen gün gelişen ve yenilenen mobil platform ürünlerinin piyasaya çıkmasıyla birlikte doğru orantılı olarak kişilerin mobil uygulama kullanımları ve tercihlerinde de artış gözlenmektedir.

Bölüm 2’de literatür araştırması ve alanda daha önce yapılmış benzer uygulamaların içerikleri ve süreçlere değinilmiştir. Bölüm 3 Metot, Uygulama ve Testler kısmında, uygulama ve yazılımın geliştirilmesi sırasında kullanılan araçlar, bu araçların kullanımı ve yazılım için gerekli olan tüm kütüphane ve sınıf yapıları açıklanmıştır. Bölüm 4 Yazılım Platformunun Geliştirilmesi ve Uygulama kısmında uygulamanın gelişim aşamasında takip edilen tüm adımlar ve aşamalar açıklanmış, bu aşamalarda kullanılan tüm fonksiyonlara ve kütüphanelere değinilmiştir. Bölüm 5 Sonuç kısmında uygulamada yapılan ve sonuca ulaşan bölümler açıklanmıştır.

LİTERATÜR TARAMASI

Vangelis Sarlis, Christos Tjortjis tarafından (2020) yılında yapılan çalışmada gelecek tahminleri, takım kompozisyonu, sporcu kariyer gelişimi ve bunun nasıl olabileceğinin değerlendirilmesi için sporda Makine Öğrenimi (ML) ve Veri Madenciliği (DM) teknikleri incelenmiştir. Yapılan inceleme sonucunda varılan sonuçlar, Basketbol bir takım sporudur ve yalnızca en verimli oyuncular, takımları değil, aynı zamanda takım oyuncularında en uygun kombinasyonları bulmak gereklidir. Amaç sahadaki performansı optimize etmek olup tüm takımdan maksimum verim almaktır. Bu yaklaşımın takım başarısında büyük bir rol aldığı gözlemlenmiştir. Bu çalışmada önemli reyting parametrelerini analiz ederek belirsizliğin daha önceki çalışmalara kıyasla daha minimuma indirilmiş olmasına karşın, maçın kritik anlarında veya baskı altında doğru karar almak gibi büyük bir etkenin tahmin edilmesi zor bir durumda kalmıştır. Basketbol takımı sadece analiz edilen metriklerle dayandırılmaz. Bunun başlıca sebebi nitel göstergeler ve belirlenemeyen oyuncu becerileridir. Belirli eşleşmelerin tahmini, birkaç dakika dikkate değer bir doğrulukla yerleştirilir fakat tüm maç boyunca bu yüksek seviyelerde kalmaz. Farklı kombinasyonlar nedeniyle bir diziliş seçimi çok karmaşıktır ve sahadaki oyuncuların en iyi performansını bulmak bu sebepten oldukça zordur. [52]

Daniela Loreti, Marco Lippi, Paolo Torroni tarafından (2019) yılında yapılan araştırmada, dağıtık bir mimari ve dağıtılmış mimarinin artırdığı hesaplama kazancının önemi sunulmuştur. Tipik bir ML sistem boru hattını paralelleştirmek için kullanılabilir. Giderek artan sayıda ML uygulaması son kullanıcılar için kullanıma hazır ve önceden eğitilmiş hizmetler olarak maddelenir. Bu aramalar bu tür ölçekleri büyütebilen dağıtılmış mimarileri uygulamak için daha geniş kullanıcı topluluklarına ve daha büyük veri koleksiyonlarına yönelik hizmetlerdir. Bu yazıda, dağıtılmış bir mimariden ilham alınmıştır. Paralelleştirmek için kullanılacak MapReduce paradigması tarafından tipik bir ML işlem hattının tahmin aşaması ele alınmıştır. Yürütülen gerçek dünya metin madenciliği ve uygulama vakası üzerindeki deneysel sonuçlar incelenmiştir. Diğer birçok farklı uygulanabilir senaryonun akış işleme ve performansı incelenmiştir. İnceleme sonucu bu mimari ile birçok açıdan elde edilebilecek kazancın daha fazla olduğu gözlemlenmiştir. [53]

Fethi A. Rabhi, Madhushi Bandara, Kun Lu, Saif Dewan tarafından (2020) yılında yapılan çalışmada amaç, yenilikçi bir Bilgi Teknolojisi (BT) platformunun tasarımını sunar. Günlük karar vermeyi desteklemek için veri analitiği yapmak isteyen bir kuruluş, genellikle analistlerin araştırma değişkenleri, veri kümeleri, analitik modeller ve problemi çözerken kullanılacak en iyi kombinasyonu etkili bir şekilde belirler. Çoğu zaman, bu tür bilgiler kuruluş tarafından açıkça ele alınmaz. Bu sorunu çözmek adına bu çalışma ele alınmıştır. Bu belge için önerilen BT platformu, farklı analitikler arasında veri paylaşımını sağlayan modüler bir şekilde tasarlanmıştır. Önerilen BT platformun mimarisi, bilgi tabanı, veri kümeleri ve tahmin modellerinden oluşur. Bu platformun temel bileşenleri ve altyapısı bir veya birkaç analitik uygulaması aracılığıyla son kullanıcıya sunulur. Herhangi bir tahmin uygulamasını altyapıya bağlamak için uygulama Programlama Arayüzleri (REST API'leri) geliştirilmiştir. Bilgi tabanını sorgulayabilen API'ler, kullanıcı girdilerine ve API'lere dayalı olarak veri kaynaklarından veri getirir. Çalışmanın sonucunda analitik bilgisini, uygulanacağı organizasyonel ortamı dikkate alarak yönetmek için tasarlanmış yenilikçi bir BT platformu sunulmuş olur. Bu BT platformu, kullanıcıların analitik modellerini yeni veri kümelerine ve ortaya çıkan yeni veri kümelerine göre uyarlama olanağıyla kullanmasını ve sürdürmesini sağlar. Bu platformun temel amacı bilgiyi korumaktır. Sonuç olarak bu faktörler dikkate alındığında analitik modelleri yönetmek için bu BT platformunu uygulayarak, kuruluşların veri analizi uygulamalarını uzun vadede verimli bir şekilde yönetebilir olmaları hedeflenmiştir. [54]

Carson K. Leung, Kyle W. Joseph (2014) tarafından yazılan makalede spor tahmin yapılarındaki nesnel tahmin, basit istatistiksel yöntemler ve geleneksel tahminlerdeki subjektif verilere dayanan tahmin yapılarına eleştirel bir bakış açısı ile birlikte, tahmin yapılarını geliştirmek için spor veri madenciliği kavramı incelenmiştir. Yaklaşım, istatistiksel verileri taradıktan sonra çıkan sonuçları 2 veri yapısında saklayıp incelenen takım çifti için bir harita modellemesi oluşturmasıyla başlar. Haritada takımlar birer nokta ile temsil edilmekte olup bu noktalar arası uzaklıkları belirler. Analiz sonucunda takımlara eklenen puanlarda hangi takım daha fazla puana sahipse sistem tarafından kazanan olarak belirlenmiş olur. Çalışma sonucu bir futbol sezonu analiz edilmiş olup maç tahminlerinde yüksek oranlı bir başarı değişimi gözlenmiştir. [55]

Gabriel Fialhoa, Aline Manhães, João Paulo Teixeira tarafından (2019) yapılan çalışmada bir müsabaka sonucu tahmininde bulunmak için farklı Yapay Zeka tekniklerini kullanan bazı araştırmaların gözden geçirilmesi konu alınmıştır. Yapay Sinir Ağları, Destek Vektör Makinesi, Bulanık Mantık ve Bulanık Sistem teknikleri incelenmiştir. Purucker, 5 özelliğten oluşan bir setle futbolu tahmini yapmış ve %61 doğruluk elde etmiştir. Daha sonra Kahn daha fazla özellik eklemiş ve %75 doğruluk elde etmiştir. Bayesian, Farzin Owrapur ve arkadaşları psikolojik ve psikolojik olmayan olarak ayrılan 13 özellik ile %92 doğruluk elde etmiştir. Igiri, özelliklere bir katman daha eklemiş, ev sahibi ve deplasman özelliklerine ayrılarak, futbol sonuçlarını tahmin etmede %85 doğruluk elde etmiştir. Tüm modeller incelendiğinde Igiri, özelliklere bir katman daha ekledi ve sinir ağı ile yüksek doğrulukta bir tahmin elde etmenin mümkün olduğunu kanıtlamıştır. Tüm veriler, sonuçları tahmin etmek için bir sinir ağına girdi olarak kullanılabilir. Bu sinir ağının yanıtı vektörle aynı formda olacaktır ancak girdi veri setindeki ikili değerler, bize sayısal çıktılar verebilir ve yüzdeler çıkarılabilir. Örneğin, yanıt [0.49, 0.31, 0.20] ise, ev sahibi takımın şansı %49 olacaktır. %31 beraberlik ve deplasman takımının kazanması için de %20 şans olacaktır. Bahsedilen bu sistem makale sonucunu ortaya çıkarmıştır. [56]

Kashif Naseer Qureshi, Sadia Din , Gwanggil Jeon , Francesco Piccialli tarafından 2020 yılında yapılan makalede sağlık sektöründeki, hastalık teşhisi ve uygun maliyetli hizmet sunumu ile ilgili çeşitli sorunlarla karşılaşılması üzerine geliştirilen makineye dayalı akıllı sistemler üzerinde durulmuştur. Öğrenme ve veri odaklı yaklaşımı benimseyen bu çalışmalarda veri toplama, bilgi ilişkilendirme, veri çıkarma ve karar verme gibi pek çok kapsam dahilinde çalışır. Amaç uygun tedaviye ulaşmaktır. Bilgi ve iletişim teknolojileri, özellikle acil durumlar için mobil sistemlerin gelişmesine yol açmıştır. “MHealth”, uydu iletişimi, bulut gibi yeni ve gelişmiş kablosuz ve kablolu teknolojilere dayanmaktadır. Temel sistemler, 4. ve 5. Nesil hücresel sistemler ve Mobil İletişim için Küresel Sistemdir (GSM). Sınıflandırma ve tahmin için birçok makine öğrenimi yöntemi tasarlanmıştır. En sık kullanılan makine öğrenme yöntemleri Yapay Sinir Ağları (YSA), Destek Vektör Makinesi (SVM), Naive Bayes Yöntemi, Karar Ağaçları (DT), Lojistik Regresyon (LR) ve K-En Yakın Komşu (K-NN) Algoritmasıdır. YSA, sinir sistemlerini canlandırmak için nöronların bağlanabilirliğini kullanan, tahmin ve sınıflandırma için popüler olan bir hesaplama modelidir. ANN ise son otuz yılda son derece önemli bir yöntem olarak kabul edilmektedir. Bu yöntemin popüleritesinin arkasındaki nedenlerden biri haritalama yeteneğidir. İki girdiyi tahmin etmek için veri kümesinden bir grup girdiyi analiz edebilir. Örnek zaman veri kümesinin geniş bir örnek koleksiyonu vardır ve her

örnek bir veya iki sınıfa aittir. Daha sonra, SVM bir model oluşturur. Örnekleri iki sınıfa eşleyen bir uzayda noktalar şeklinde örnekler tahsis eder. Sınıflar, bir boşluk açısından konumları kullanılarak ayrıca analiz edilir. Kerner hilesi ile de örnekleri yüksek boyutlu uzaya eşlemek için kullanılır. Hastalık analizinin tahmininde doğruluk için makine öğrenme teknikleri ve sınıflandırma kullanılmıştır. Önerilen makine öğrenimi modeli, CVD tahmini için mükemmel bir seçenek gibi görünmektedir. Bir veri setinden hasta özelliklerinin analizine dayalıdır. Karşılaştırmalı bir değerlendirmede mevcut son teknolojiden daha iyi performans gösterdiği gözlemlenmiştir. Sonuç olarak doğruluk, duyarlılık ve özgüllük açısından uygun bir model olduğu belirlenmiştir. [57]

Eric Gossetta, Cormac Tohera , Corey Osesa, Olexandr Isayevc, Fleur Legraind, Frisco Rosea, Eva Zurekf, Jesús Carreteg, Natalio Mingod , Alexander Tropshac, Stefano Curtarolo tarafından (2018) yapılan çalışmada RestApi kavramı ve kullanımı incelenmiştir. Kapsamlı malzeme özellikleri veri tabanlarının ortaya çıkmasıyla sağlanan makine öğrenmesi yaklaşımları, malzeme analizi için verimli bir yön haline gelmektedir. Sonuç olarak, çok sayıda model oluşturulmuş ve yeni sistemlerin özelliklerini tahmin etmek için mevcut veriler üzerinde eğitim yapılmıştır. Bu güçlü yöntemler, araştırmacıların sadece ilginç materyallerinde çalışır, sentezlenemeyen sistemleri ve arzu edilmeyen sistemleri ihmal eder. Ancak, bu tahmine dayalı modelleri kullanmak her zaman kolay değildir. Genellikle, genel kullanıcılar için engeller yaratarak bir dizi teknik uzmanlık gerektirirler. AFLOW-ML (AFLOW Makine Öğrenmesi) içinde geliştirilen makine öğrenimi yöntemlerinin kullanımını kolaylaştırarak sorunun üstesinden gelir. Bu çerçevede, sürekli güncellenenlere doğrudan erişmek için açık bir RESTful API sağlanır. Tahminleri almak için herhangi bir iş akışına şeffaf bir şekilde entegre edilebilen algoritmalar ve bu tür birbirine bağlı bulut tabanlı uygulamaların, makine öğrenimi yöntemlerinin malzeme geliştirmede benimsenmesini daha da hızlandırabilir. AFLOW-ML API, bir Temsili Durum etrafında yapılandırılmıştır. Kaynaklara erişime izin veren transfer (REST) mimarisi HTTP istek yöntemleri kullanılır. Her kaynak bir uç noktada bulunur ve tanımlayıcı isimlerden oluşan bir URL ile tanımlanır. Bir URL, yol biçiminde özel değişkenler de içerebilir. API içindeki kaynaklar JavaScript Nesnesinde temsil edilir. Notasyon (JSON) ve nesneler olarak adlandırılır. Kullanıcı, nesneyle nasıl etkileşimde bulunacağını belirtmelidir ve buna atıfta bulunmalıdır. Bu şekilde API'den istekte bulunurlar. Bütün bu sürecin ve araştırmanın sonucunda AFLOW-ML, kolaylaştırılmış özellikler sağlayarak malzeme keşfini geliştirmiştir. Sonuç olarak AFLOW-ML ile tahmine dayalı modellere açık erişim sağlanabileceği gözlemlenmiştir. [58]

Rory P. Bunker, Fadi Thabtah tarafından 2017 yılında yapılan bu çalışmada, Yapay Zeka uygulamalarına odaklanarak makine öğrenimindeki literatürün analizini yapmıştır. İstatistik ve yöneylem araştırması literatüründeki birçok çalışma daha önce spor sonuçlarını dikkate almasına karşın bu amaç için NN paradigmasının kullanılması daha yeni bir çalışma alanıdır. Buna rağmen güçlü bir NN tekniği bu kısa sürede kendini kanıtlamış ve son derece doğru sınıflandırma modellerinin türetilmesinde etkili olmuştur. Spor sonuçları tahmini için bu akıllı modelleri kullanırken ortaya çıkan zorluklarla ilgili tartışmalar da mevcuttur. Temel katkımız, spor sonucu tahmini için CRISP-DM tipi bir çerçevenin (SRP-CRISP-DM) önerilmesidir. Sporda iyi tahmin doğruluğu gerektiren hayati uygulamalardan biri maç sonucu tahminidir. Geleneksel olarak, sonuçlar maçların matematiksel ve istatistiksel verileri kullanılarak tahmin edilir. Maçla ilgili özelliklerin farklı, kendine özgü doğası nedeniyle, tahmin sonuçları çoğunlukla doğru olmaz. Bu nedenle ML, maçı tahmin edebilen tahmine dayalı modeller ürettiği için spor tahmini için uygun bir metodoloji gibi görünmektedir. Yine de ML modellerinin artan kullanımına rağmen spor tahmini için daha doğru modellere ihtiyaç vardır. Bunun nedeni gelecekteki eşleştirme stratejilerini modellemek için faydalı bilgiler aramaktan geçmektedir. Bu makalede, sporla ilgili bazı yeni araştırmalar analiz edilmiştir. Sonuç olarak YSA kullanan tahmin modeli ve bunu takiben, spor sonucu tahmininin karmaşık sorunu için bir spor sonucu tahmini olan "SRP-CRISP-DM" çerçevesi önerilmiştir. [59]

METOT, UYGULAMA ve TESTLER

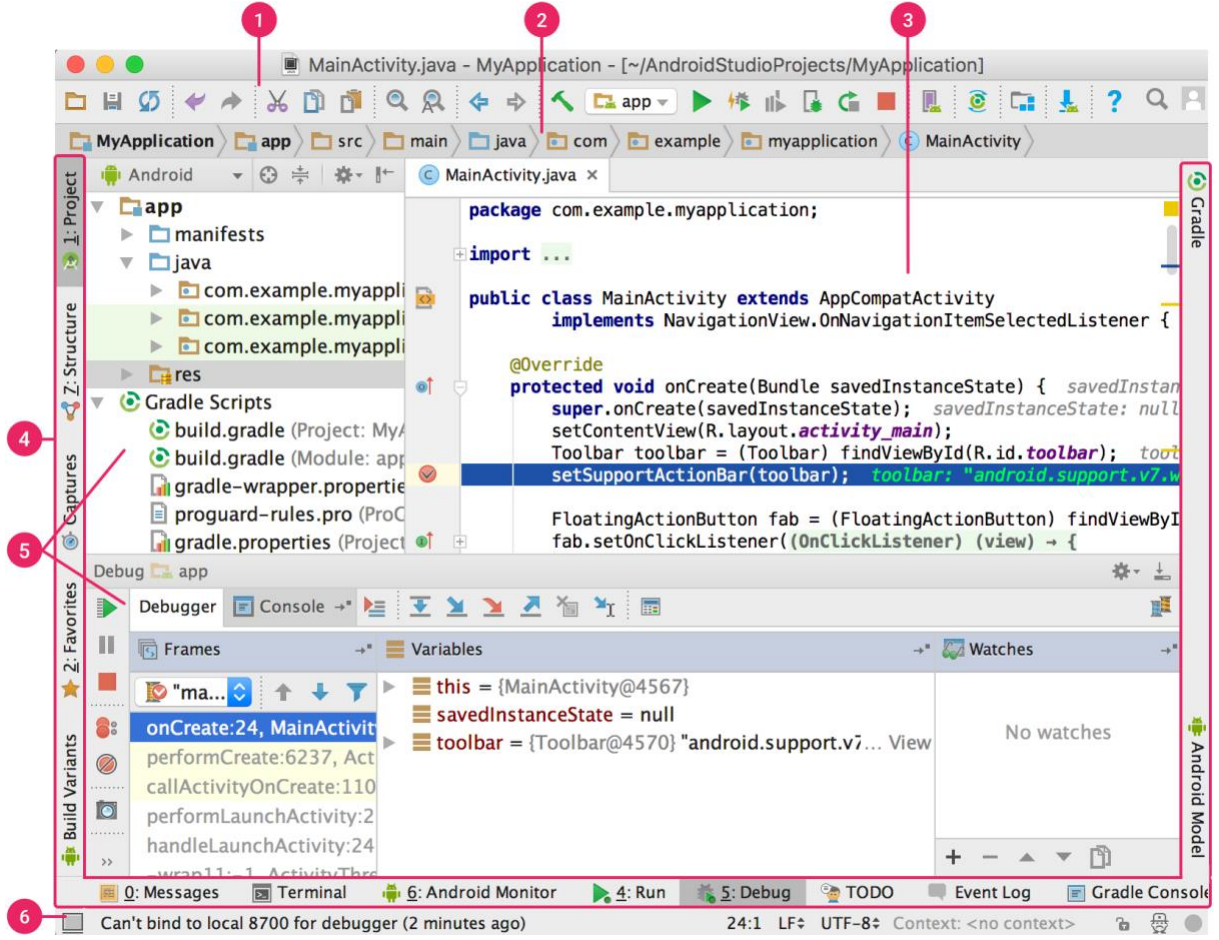
Spor analitikleri üzerine kurulan bu uygulamanın oluşturulması için Android Studio SDK isimli geliştirme platformu kullanılmaktadır. Android Studio, IntelliJ IDEA'ya dayalı Android uygulama geliştirme için resmi Entegre Geliştirme Ortamıdır.

Android Studio, gradle tabanlı esnek bir yapı sistemidir. Hızlı ve zengin özelliklere sahip bir emülatördür. Tüm Android cihazlar için uygulama geliştirilebilir birleşik bir ortamdır. Çalışan uygulamaya kod ve kaynak gönderimi yapılmasına olanak sağlar. Kapsamlı test araçları ve çerçeveleri bünyesinde bulundurulur. Performans, kullanılabilirlik, sürüm uyumluluğu ve diğer sorunları yakalamak için araçlar bulundurulur. Google Cloud Messaging ve App Engine'i entegre etmeyi kolaylaştıran Google Cloud Platform için yerleşik destek sağlar. [47]

“C++, Bjarne Stroustrup tarafından 1979 yılından itibaren geliştirilmeye başlanmış, C'yi kapsayan ve çok paradigmatlı, yaygın olarak kullanılan, genel amaçlı bir programlama dilidir. Nesne paradigması kullanılarak programlamaya olanak tanır. Sınıflar sayesinde yeni veri türleri yaratılabilir veya var olan türlerden yenileri türetilebilir. Ayrıca çok biçimlilik sayesinde bir sınıf tanımıyla yazılmış kod, o sınıf türünden türetilmiş yeni sınıflarla da çalışabilir.” [48]

Android NDK, uygulamalarda C ve C ++ gibi dillerinde yazılmış kodların Native Android uygulamalarında kullanabilmesini sağlayan bir araç setidir. Aynı şekilde bu diller için hazırlanmış olan kütüphanelerin de kullanılmasını sağlar. Bunlar hem var olan hazır kütüphaneler hem de yüksek performans sağlayan API'ler olabilir. [49]

Android Studio'daki her proje, kaynak kod dosyaları ve kaynak dosyaları içeren bir veya daha fazla modül içerir. Modül türlerine Android uygulama modülleri, Kütüphane modülleri, Google App Engine modülleri örnek verilebilir. Tüm derleme dosyaları Gradle komut dosyaları altında en üst düzeyde görünür ve her uygulama modülü manifests, java ve res klasörlerini içerir. Manifests, AndroidManifest.xml dosyasını içerir. Java, JUnit test kodu da dahil olmak üzere Java kaynak kodu dosyalarını içerir. Res, XML düzenleri, kullanıcı arabirimi dizeleri ve bitmap görüntüleri gibi tüm kod dışı kaynakları içerir. Diskteki Android proje yapısı, bu düzleştirilmiş gösterimden farklıdır. [47]



Şekil 1 Android Studio Anasayfası [47]

1. Araç çubuğu, uygulamanın çalıştırılması ve Android araçlarını başlatma dahil olmak üzere çok çeşitli işlemlerin gerçekleştirilmesine olanak tanır. [47]
2. Gezinme çubuğu, projede gezinmeye ve dosyaları düzenlemek üzere açmaya yardımcı olur. Proje penceresinde görünen yapının daha kompakt bir görünümünü sağlar. [47]
3. Düzenleyici penceresi, kodun oluşturulduğu ve değiştirildiği yerdir. Geçerli dosya türüne bağlı olarak düzenleyici değişebilir. Örneğin, bir Layout dosyasını görüntülerken düzenleyici, Layout Düzenleyici'yi görüntüler. [47]
4. Araç penceresi çubuğu, IDE penceresinin dışında çalışır ve tek tek araç pencerelerinin genişletilmesine veya daraltılmasına izin veren düğmeleri içerir. [47]
5. Araç pencereleri, proje yönetimi, arama, sürüm kontrolü ve daha fazlası gibi belirli görevlere erişilmesini sağlar. [47]
6. Durum çubuğu, projenin durumunu ve IDE'nin yanı sıra herhangi bir uyarı veya mesajı görüntüler. [47]

Bu uygulamanın oluřturulmasında, zamandan, hafızadan tasarruf etmek ve uygulama yükünü minimize etmek için birçok kütüphane ve sınıf yapısı kullanılmıştır.

Kütüphane, bilgisayar programları tarafından yazılım geliştirme amaçlı kullanılan kalıcı kaynakların tümüdür. Sistem çağrılarının her seferinde yeniden yazılması yerine bir kütüphane aracılığıyla yapılması bu duruma örnek gösterilebilir. Yazılım kütüphanesi, yazılımcıların daha önceden yazmış oldukları kodlar, hazırlamış oldukları sınıflar, prosedürler, komut dosyaları, yapılandırma verileri ve daha fazlasını içeren bir pakettir. Kod yazma kolaylığı sağlayan veya tekrar eden işlemleri kolaylařtırmak, otomatikleřtirmek için kullanılan bir yapıdır. [50]

Sınıf, nesne yönelimli programlama dillerinde nesnelerin özelliklerini, davranışlarını ve başlangıç durumlarını tanımlamak için kullanılan şablona verilen addır. Sınıflar genelde şahıs, yer ya da bir nesnenin ismini temsil ederler. Sınıflar metotları ile nesnelerin davranışlarını, deęişkenleri ile ise nesnelerin durumlarını kapsül ederler. Sınıflar hem veri yapısına hem de bir ara yüze sahiptirler. Sınıflar ile nasıl etkileşime girileceęi bu ara yüzler sayesinde sağlanır. [51]

1 Activities

Program için arka planda çalışan tüm yapının baęlı olduęu sınıfların bulunduęu kısımdır.

1.1 MainActivity Class

Uygulama çalıştığı zaman ilk olarak yürütölen kısımdır. Bu sınıf, dięer tüm sınıfların bütönlöşik biçimde yürütölme noktasıdır.

1.1.1 onCreate Fonksiyonu

Class'ı yaratan fonksiyondur. Parametrelerin tanımlamaları ve atamaları gerçekleştirilir. Uygulamanın Main ekranının tasarımına baęlantı sağlanır. Bu sayede burada çalışan arka plan komutları tasarımda işlevsel hale getirilmiş olur.

1.1.2 onBackPressed Fonksiyonu

Main ekranından herhangi bir ekrana geçiş yapıldığı zaman çıkış komutunu gerçekleştiren fonksiyondur. Kullanıcıya bulunduğu ekrandan çıkmak isteyip istemediğinin dönüşü doğrultusunda Main ekranına geri dönüş yapılır ya da mevcut ekranda işleme devam edilir.

1.1.3 profiletanimla Fonksiyonu

Google ile oturum açan kullanıcının giriş bilgileri daha önceden belirlenmiş değişkenlere atamaları yapılır. Bu sayede bu değişkenlerde oturum açan kullanıcının bilgileri tutulur ve profil düzenlemelerinde kullanılır.

1.1.4 onUpdateNeeded Fonksiyonu

Uygulamanın son sürümü kullanıp kullanmadığını test eden fonksiyondur. Eğer kullanıcı artık uygun olmayan bir sürüm kullanıyor ise güncelleme önerisi için uyarı verir.

1.1.5 redirectStore Fonksiyonu

Güncel olmayan bir uygulama kullanan kullanıcıya uyarı verilir. Bu uyarı sonucunda kullanıcı uygulamayı güncellemeyi kabul etmesi durumunda redirectStore fonksiyonu ile uygulamayı güncelleyebilmesi için mevcut store'a yönlendirilir.

1.1.6 onNavigationItemSelected Fonksiyonu

Main Ekranında bulunan menü özelliğinde yapılan seçimlerin yönlendirmesini sağlayan fonksiyondur. Home item'ini seçen kullanıcı main ekranına yönlendirilir. Bu yönlendirmeler “Activities” bünyesinde bulunan sınıfları kullanarak gerçekleşir. Main ekranı için MainActivity Class'a yönlendirme yapılır.

1.2 Login Class

Google Tabanlı giriş işlemlerinin gerçekleştiği sınıftır. Tüm değişkenler tanımlanarak başlanır.

1.2.1 onCreate Fonksiyonu

Class yapısını başlatan fonksiyondur. “tanımla” fonksiyonu ile giriş butonu tanımlanır. Tasarımda bulunan değişkenlerin class yapısına ayarlanması gerçekleşir. GoogleSignIn kütüphanesi bir değişkene atılır.

1.2.2 showMyCustomAlertDialog Fonksiyonu

Giriş için gerekli olan sözleşmeler kabul edilmediğinde uyarı çıkmasını sağlayan fonksiyondur. Çıkan uyarıda kullanıcı sözleşmeleri kabul ettiğini işaretlemesi durumunda kabul butonuna basıldığında giriş ekranındaki sözleşmelerin kabul edildiği işaretlenir.

1.2.3 onStart Fonksiyonu

Giriş yapılacak hesap seçilir ve giriş yapılan hesabın kontrolü için girişCheck fonksiyonuna kullanıcı mail adresini iletilir.

1.2.4 girişCheck Fonksiyonu

Girdi olarak aldığı mail adresinin varlığını kontrol eder. Eğer böyle bir hesap yok ise ekrana uyarı verilir. Hesap mevcut ise onLoggedIn fonksiyonu çağırılır.

1.2.5 onLoggedIn Fonksiyonu

Giriş işleminin son kısmını gerçekleştiren fonksiyondur. Süreci tamamlar, kullanıcı bilgilerini MainActivity class'ına iletir ve Main ekranına geçiş işlemini sağlar.

1.2.6 onActivityResult Fonksiyonu

Giriş yapmak isteyen kullanıcının tüm bilgilerinin değişkenlere atanarak saklandığı fonksiyondur.

1.2.7 register Fonksiyonu

İlk kez kayıt olacak kullanıcının bilgilerinin saklandığı fonksiyondur. Var olmayan bir Google hesabı ile kayıt olmak isteyen kullanıcıya uyarı verilir.

1.3 Logout Class

Uygulama içerisinde açık olan oturumdan çıkıp giriş ekranına dönmeyi sağlayan sınıftır.

1.3.1 onCreate Fonksiyonu

Class yapısını başlatan fonksiyondur. “tanımla” fonksiyonu ile kullanıcı bilgileri çağırılır ve çıkış yapılacak hesap bilgileri ekranda yer alır. Ekranda yer alan başlık oluşturulur ve gösterilir. Çıkış işlemleri için gerekli googleSocialLogin isimli fonksiyon burada çağırılır.

1.3.2 googleSocialLogin Fonksiyonu

Google servis çağrısını başlatan fonksiyondur. Burada googleApiClient oluşturulur.

1.3.3 onBackPressed Fonksiyonu

Bulunduğu ekrandan çıkıp bir önceki ekrana yönlendirme sağlayan fonksiyondur. Logout ekranından Main ekranına dönmeyi sağlar.

1.3.4 tanımla Fonksiyonu

Çıkış yapacak kullanıcının bilgilerini ekranda yazdırmak için tasarımda verilen isimlendirmeleri class yapısındaki değişkenlere atamalarını gerçekleştiren fonksiyon.

1.3.5 onActivityResult Fonksiyonu

Servisten aldığı response bilgisini alıp handleResultGoogleSignIn fonksiyonuna iletir.

1.3.6 handleResultGoogleSignIn Fonksiyonu

Bu fonksiyon içerisinde çıkış yapmak isteyen kullanıcı bilgileri elde tutulur.

1.3.7 googleLogout Fonksiyonu

Oturumu GoogleSignInApi kütüphanesinin signOut fonksiyonu ile sonlandırmasını sağlayan fonksiyondur. Tüm işlemlerin sonunda gerçekleşir ve çıkış işlemi tamamlanır.

1.3.8 onConnectionFailed Fonksiyonu

İnternet erişimi olmayan cihazda uyarı veren bir fonksiyondur.

1.4 Contact Class

Kullanıcıların, e-posta yoluyla iletişimi için gerekli arka plan komutlarının bulunduğu sınıf.

1.4.1 onCreate Fonksiyonu

Class'ı yaratan fonksiyondur. Parametrelerin tanımlamaları ve atamaları gerçekleştirilir. Google hesap bilgilerinin gerekli değişkenlere atamaları gerçekleştirilir. Gerekli fonksiyonların çağırma işlemleri burada gerçekleşir.

1.4.2 onBackPressed Fonksiyonu

Mevcut ekrandan bir önceki ekrana geri dönüş yapılmasını sağlayan fonksiyondur.

1.4.3 validateText Fonksiyonu

E-posta için oluşturulan textboxlar tanımlanır ve değişkenlere atamaları gerçekleştirilir. Bu değişkenler vasıtasıyla textboxların boş olup olmadığı kontrol edilir ve olumlu ya da olumsuz değer döndürülür.

1.4.4 tanımla Fonksiyonu

Kullanıcıya servis edilen ekrandaki itemlerin tasarımı gerçekleştiren fonksiyon. Gönderici olarak oturumu açık olan kullanıcıyı seçili getirir. Nasıl kullanılacağını göstermek için “here” isimli bir text butonu vardır.

1.4.5 click Fonksiyonu

Gönderme butonunun çalışma fonksiyonudur. Textboxlardan alınan verilere göre değişken atamaları gerçekleştirilir. Daha bu değişkenler “SoruSorReg” isimli fonksiyona işlenmesi için gönderilir. Son olarak “delete” isimli fonksiyon çağırılır ve textboxların içeriği temizlenir.

1.4.6 delete Fonksiyonu

Konu ve mesaj textboxlarının içeriğinin temizlenmesini sağlayan fonksiyon.

1.4.7 SoruSorReg Fonksiyonu

Girdi olarak aldığı değişkenleri işleyerek e-posta gönderimini sağlayan fonksiyon. Değişkenleri işledikten sonra “onResponse” isimli fonksiyon vasıtasıyla gönderim gerçekleşir. Bir hata olması durumunda “onFailure” fonksiyonu ile uyarı verilmesi sağlanır.

1.5 HowToUse Class

Kullanıcıların, uygulamayı nasıl kullanacaklarına dair anlatımın olduğu sayfanın arka plan komutlarının bulunduğu sınıf.

1.5.1 onCreate Fonksiyonu

Class’ı yaratan fonksiyondur. Parametrelerin tanımlamaları ve atamaları gerçekleştirilir.

1.5.2 onBackPressed Fonksiyonu

Mevcut ekrandan bir önceki ekrana geri dönüş yapılmasını sağlayan fonksiyondur.

1.5.3 tanımla Fonksiyonu

Uygulamanın web sitesinde bulunan kullanma talimatının bulunduğu html sayfasına url ataması gerçekleştirilerek bu sayfada bulunan verileri uygulama içerisinde kullanıcıya servis eder.

1.6 Privacy Class

Kullanıcıların, gizlilik politikaları hakkında bilgi alabileceği sayfanın arka plan komutlarının bulunduğu sınıf.

1.6.1 onCreate Fonksiyonu

Class'ı yaratan fonksiyondur. Parametrelerin tanımlamaları ve atamaları gerçekleştirilir. Gerekli fonksiyonların çağırılması işlemleri gerçekleştirilir.

1.6.2 onBackPressed Fonksiyonu

Mevcut ekrandan bir önceki ekrana geri dönüş yapılmasını sağlayan fonksiyondur.

1.6.3 tanımla Fonksiyonu

Uygulamanın web sitesinde bulunan kullanma talimatının bulunduğu html sayfasına url ataması gerçekleştirilerek bu sayfada bulunan verileri uygulama içerisinde kullanıcıya servis eder.

1.7 Terms Class

Kullanıcıların, uygulamanın kullanım koşulları hakkında bilgi almasını sağlayan sayfanın arka plan komutlarının bulunduğu sınıf.

1.7.1 onCreate Fonksiyonu

Class'ı yaratan fonksiyondur. Parametrelerin tanımlamaları ve atamaları gerçekleştirilir. Gerekli fonksiyonların çağırılması işlemleri gerçekleştirilir.

1.7.2 onBackPressed Fonksiyonu

Mevcut ekrandan bir önceki ekrana geri dönüş yapılmasını sağlayan fonksiyondur.

1.7.3 tanımla Fonksiyonu

Uygulamanın web sitesinde bulunan kullanma talimatının bulunduğu html sayfasına url ataması gerçekleştirilerek bu sayfada bulunan verileri uygulama içerisinde kullanıcıya servis eder.

1.8 Subscriptions Class

Kullanıcının üye olduğu kategorilerin kullanıcıya listelenmesi için arka plan kodlarının bulunduğu sınıftır. “tanımla” isimli fonksiyon kullanılarak tasarım ekranı ile bağlantı sağlanır, gerekli nesne tanımlamaları yapılır. “getSubsinfo” isimli fonksiyon içerisinde “AdapterSubs” isimli Class kullanılarak kullanıcının tüm üyelik paketleri çağırılır ve kullanıcıya listelenir.

2 Models

Geri dönüş sağlayan kısa fonksiyonların bağlı olduğu sınıfların bulunduğu yapıdır.

2.1 ContactUsModel Class

İletişim bölümünde ihtiyaç duyulan dönüşlü fonksiyonların tanımlandığı class.

2.1.1 isTf Fonksiyonu

Çağırıldığı yapının doğru mu yoksa yanlış mı olduğunu kontrol eder. Hatalı bir yapı ise false, doğru bir yapı ise true return eder.

2.1.2 setText Fonksiyonu

Girdi olarak aldığı text’i bulunduğu konumdaki text değişkenine atamasını gerçekleştirir.

2.1.3 getText Fonksiyonu

setText ile daha önceden atanmış olan text’i ihtiyaç anında çağırmak için kullanılan fonksiyon.

2.1.4 toString Fonksiyonu

Kullanılan fonksiyonların return değerlerinin modellemesini yapan fonksiyon. Return değerlerini bir çıktı olarak ekrana yazılmasını sağlar.

2.2 LoginPageModel Class

Giriş ekranı bölümünde ihtiyaç duyulan dönüşlü fonksiyonların tanımlandığı class.

2.2.1 isTf Fonksiyonu

Çağırıldığı yapının doğru mu yoksa yanlış mı olduğunu kontrol eder.

2.2.2 setText Fonksiyonu

Girdi olarak aldığı text'i bulunduğu konumdaki text değişkenine atamasını gerçekleştirir.

2.2.3 getText Fonksiyonu

setText ile daha önceden atanmış olan text'i ihtiyaç anında çağırmak için kullanılan fonksiyon.

2.3 MailSubsModel Class

Mail ekranı için ihtiyaç duyulan dönüş fonksiyonlarının tanımlandığı class.

2.3.1 setText Fonksiyonu

Girdi olarak aldığı text'i bulunduğu konumdaki text değişkenine atamasını gerçekleştirir.

2.3.2 getText Fonksiyonu

setText ile daha önceden atanmış olan text'i ihtiyaç anında çağırmak için kullanılan fonksiyon.

2.3.3 toString Fonksiyonu

Kullanılan fonksiyonların return değerlerinin modellemesini yapan fonksiyon. Return değerlerini bir çıktı olarak ekrana yazılmasını sağlar.

2.4 RegisterModel Class

Kayıt için ihtiyaç duyulan dönüş fonksiyonlarının tanımlandığı class.

2.4.1 isTf Fonksiyonu

Çağırıldığı yapının doğru mu yoksa yanlış mı olduğunu kontrol eder. Hatalı bir yapı ise false, doğru bir yapı ise true return eder.

2.4.2 setText Fonksiyonu

Girdi olarak aldığı text'i bulunduğu konumdaki text değişkenine atamasını gerçekleştirir.

2.4.3 getText Fonksiyonu

setText ile daha önceden atanmış olan text'i ihtiyaç anında çağırmak için kullanılan fonksiyon.

2.5 GetGames Class

Maçları listelemek için kullanılan fonksiyonların bulunduğu sınıftır.

2.5.1 isTf Fonksiyonu

Çağırıldığı yapının doğru mu yoksa yanlış mı olduğunu kontrol eder. Hatalı bir yapı ise false, doğru bir yapı ise true return eder.

2.5.2 toString Fonksiyonu

Kullanılan fonksiyonların return değerlerinin modellemesini yapan fonksiyon. Return değerlerini bir çıktı olarak ekrana yazılmasını sağlar.

2.5.3 getMacuyelik Fonksiyonu

Kullanıcının hangi kategoriye üye olduğunu bildiren fonksiyondur.

2.5.4 getMactakimler Fonksiyonu

Tahmin edilen maçtaki takımların isimlerini çağıran fonksiyondur. Listeleme işlemi sırasında takım isimlerini listelemek için kullanılır.

2.5.5 getMactarih Fonksiyonu

Tahmin edilen maçın tarihini çağıran fonksiyondur. Listeleme işlemi sırasında maç tarihini listelemek için kullanılır.

2.5.6 getMacsaat Fonksiyonu

Tahmin edilen maçın saatini çağıran fonksiyondur. Listeleme işlemi sırasında maç saatini listelemek için kullanılır.

2.5.7 getMacskor Fonksiyonu

Tahmin edilen maçın skorunu çağıran fonksiyondur. Listeleme işlemi sırasında maç skorunu listelemek için kullanılır.

2.5.8 getMaculke Fonksiyonu

Tahmin edilen maçın bulunduğu ülkeyi çağıran fonksiyondur. Listeleme işlemi sırasında maçın ülkesini listelemek için kullanılır.

2.5.9 getMaclig Fonksiyonu

Tahmin edilen maçın bulunduğu ligi çağıran fonksiyondur. Listeleme işlemi sırasında maçın bulunduğu ligi listelemek için kullanılır.

2.5.10 getMactahmin Fonksiyonu

Maç için yapılan tahmini çağıran fonksiyondur. Listeleme işlemi sırasında maç tahminini listelemek için kullanılır.

2.5.11 getMacdurum Fonksiyonu

Maç sonucunu çağıran fonksiyondur. Listeleme işlemi sırasında maç skorunu listelemek için kullanılır.

2.5.12 getMacbonus

Maçın iptal durumunu çağıran fonksiyondur.

3 Restapi

REST, Web protokolleri ve teknolojilerini kullanan bir dağıtık sistemdir.

3.1 RestApi Class

RestApi web servislerini kullanarak uygulamanın bulunduğu cihazda bellek kullanmadan veri ve haberleşme işlemlerini sağlar. RestApi Class yapısı, “models” içerisinde bulunan class’ları çağırıp, her bir class’ın içerisinde bellekte yer tutmayacak şekilde, daha sonra kullanmak üzere yeni yapılar oluşturur. RegisterModel isimli class çağırılıp registerUser isimli yapı, ContactUsModel isimli class çağırılıp SoruSor isimli yapı, LoginPageModel isimli class çağırılıp LoginUser isimli yapı ve MailSubsModel isimli class çağırılıp MailToSubscription isimli yapı oluşturuldu.

3.2 ManagerAll Class

Senkronizasyon işlemlerini gerçekleştirdikten sonra RestApi Class'ında üretilen her bir yapı için yeni fonksiyon üretildi. Üretilen bu fonksiyonlar bir dönüş değerine sahip olup ihtiyaç duyulduğunda restapi aracılığıyla oluşturulmuş yapıları çağırarak. Oluşturulan bu fonksiyonlar “activities” altında bulunan class yapılarında kullanılacak. Login Class dosyasında kullanılmak üzere “kayitol” isimli fonksiyon oluşturuldu. Oluşturulan bu fonksiyon restapi aracılığı ile “registerUser” isimli yapıyı döndürmektedir. Login Class dosyasında kullanılmak üzere “girisKontrol” isimli fonksiyon oluşturuldu. Oluşturulan bu fonksiyon restapi aracılığı ile “LoginUser” isimli yapıyı döndürmektedir. Contact Class dosyasında kullanılmak üzere “soruSor” isimli fonksiyon oluşturuldu. Oluşturulan bu fonksiyon restapi aracılığıyla “SoruSor” isimli yapıyı döndürmektedir.

3.3 BaseManager Class

RestApiClient Class'ında bulunan restapi bağlantıları ile yeni bir yapı yaratılır. Yaratılan bu yapının BaseUrl Class içerisinde saklanan uygulamanın web sitesinin url'i ile bağlantısı sağlanır. Oluşturulan “getRestApi” fonksiyonu ile bu bağlantı return edilir.

3.4 CallbackWithRetry Class

Bu class, farklı bir class içerisinde bulunan herhangi bir yapının klonunun çağırılmasında kullanılır. “retry” fonksiyonu ile girdi alınan yapı kopyalanarak çağırılır. “onFailure” fonksiyonu ile herhangi bir hata oluşması durumunda uyarı verilir.

3.5 RestApiClient Class

Restapi bağlantılarının temellerinin dayandığı sınıf yapısıdır. Öncelikle girdi alınan url işlenir. Daha sonra “Gson” kütüphanesi kullanılarak json dosyalarını işlemek için bir yapı kurulur. Girdi alınan url işlenir, “Retrofit” kütüphanesi ile daha önceden “Gson” kütüphanesi kullanılarak işlenmiş olan json dosyaları kullanılır, “getUnsafeOkHttpClient” fonksiyonu ile güvenilirlikler ve yürütülebilir işler test edilir. Bunların sonucunda yapıcı işleri tamamlanır. Bağlantı hazır hale getirilir. Oluşturulan “getRestApi” fonksiyonu ile hazır olan bağlantı işlenmek üzere çağırılabilir.

3.6 BaseUrl Class

Uygulamanın web sitesi ile bağlantısının kurulması için url saklayan bir class yapısıdır.

4 Adapter

Birbiri ile uyumsuz ara yüzleri olan sayfaların birbiriyle çalışmasını sağlayan sınıf yapısıdır.

4.1 AdapterGames Class

Tüm kategoriler altındaki tahminlerin sonuçlarını işleyen sınıf yapısıdır. “onBindViewHolder” fonksiyonu altında işlemleri gerçekleştirir. Kategori türüne göre tahmin edilen maçın başarı durumu, takım isimleri, skor, lig adı gibi bilgileri listeler. Tüm bu bilgiler “GetGames.java” klasöründe bulunan fonksiyonlardan elde edilir. “getMacDurum” isimli fonksiyon ile maçın sonucu getirilir ve tahmin ile karşılaştırılır. Karşılaştırma sonucuna göre başarılı ya da başarısız olduğu listelenir.

4.2 AdapterProof Class

Free kategorisi altındaki tahminlerin sonuçlarını işleyen sınıf yapısıdır. “onBindViewHolder” fonksiyonu altında işlemleri gerçekleştirir. Tahmin edilen maçın başarı durumu, takım isimleri, skor, lig adı gibi bilgileri listeler. Tüm bu bilgiler “GetGames.java” klasöründe bulunan fonksiyonlardan elde edilir. “getMacDurum” isimli fonksiyon ile maçın sonucu getirilir ve tahmin ile karşılaştırılır. Karşılaştırma sonucuna göre başarılı ya da başarısız olduğu listelenir.

4.3 AdapterSubs Class

Kullanıcının üye olduğu kategorilerin bilgisinin elde edilmesi için arka plan kodlarının bulunduğu sınıftır. Oluşturulan “onBindViewHolder” isimli fonksiyon içerisinde kullanıcının üye olduğu tüm kategorilerin bilgisi alınır. Bu bilgiler için “GetSubs” sınıfında kullanıcıların bilgilerini döndüren fonksiyonlar kullanılır.

5 Fragments

Ara yüzlerin çalışması için gerekli arka plan komutlarını tutan sınıf yapılarıdır.

5.1 Succcess Class

Tüm kategorilerdeki tahminlerin sonuçlarını bağlı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanimla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh”

fonksiyonu ile sınıfın bağı olduğu sayfa aktif olarak güncellenir. “getSuccessMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.2 Free Class

Free kategorisindeki tahminleri ve sonuçlarını, bağı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağı olduğu sayfa aktif olarak güncellenir. “getFreeMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.3 Silver Class

Silver kategorisindeki tahminleri ve sonuçlarını, bağı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağı olduğu sayfa aktif olarak güncellenir. “getSilverMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.4 Silver_Subscription Class

Silver kategorisine üye olacak kullanıcılar için üyelik işlemlerini gerçekleştiren komutların bulunduğu sınıf yapısıdır. “tanımla” isimli fonksiyon ile Silver kategorisine üye olmak için kullanılan ekranda bulunan nesneler tanımlanır. Nesnelere erişim için gerekli arka plan komutları “setOnClickListener” isimli fonksiyon ile tanımlanmıştır. Seçilen paket türüne göre “billingInterface” sınıfı kullanılarak üyelik alımı gerçekleştirilir.

5.5 Gold Class

Gold kategorisindeki tahminleri ve sonuçlarını, bağı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağı olduğu sayfa aktif olarak güncellenir. “getGoldMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.6 Gold_Subscription Class

Gold kategorisine üye olacak kullanıcılar için üyelik işlemlerini gerçekleştiren komutların bulunduğu sınıf yapısıdır. “tanımla” isimli fonksiyon ile Gold kategorisine üye

olmak için kullanılan ekranda bulunan nesneler tanımlanır. Nesnelere erişim için gerekli arka plan komutları “setOnClickListener” isimli fonksiyon ile tanımlanmıştır. Seçilen paket türüne göre “billingInterface” sınıfı kullanılarak üyelik alımı gerçekleştirilir.

5.7 Htft Class

Htft kategorisindeki tahminleri ve sonuçlarını, bağlı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağlı olduğu sayfa aktif olarak güncellenir. “getHtftMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.8 Htft_ Subscription Class

Htft kategorisine üye olacak kullanıcılar için üyelik işlemlerini gerçekleştiren komutların bulunduğu sınıf yapısıdır. “tanımla” isimli fonksiyon ile Htft kategorisine üye olmak için kullanılan ekranda bulunan nesneler tanımlanır. Nesnelere erişim için gerekli arka plan komutları “setOnClickListener” isimli fonksiyon ile tanımlanmıştır. Seçilen paket türüne göre “billingInterface” sınıfı kullanılarak üyelik alımı gerçekleştirilir.

5.9 Vip Class

Vip kategorisindeki tahminleri ve sonuçlarını, bağlı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağlı olduğu sayfa aktif olarak güncellenir. “getVipMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.10 Vip_ Subscription Class

Vip kategorisine üye olacak kullanıcılar için üyelik işlemlerini gerçekleştiren komutların bulunduğu sınıf yapısıdır. “tanımla” isimli fonksiyon ile Vip kategorisine üye olmak için kullanılan ekranda bulunan nesneler tanımlanır. Nesnelere erişim için gerekli arka plan komutları “setOnClickListener” isimli fonksiyon ile tanımlanmıştır. Seçilen paket türüne göre “billingInterface” sınıfı kullanılarak üyelik alımı gerçekleştirilir.

5.11 Draw Class

Draw kategorisindeki tahminleri ve sonuçlarını, bağlı olduğu sayfada listeleyen sınıf yapısıdır. “onCreateView” fonksiyonu ile gerekli tasarım sayfasına bağlantı kurulur. “tanımla” fonksiyonu ile tasarımda bulunan elemanlar bu sınıf yapısı içerisinde tanımlanır. “onRefresh” fonksiyonu ile sınıfın bağlı olduğu sayfa aktif olarak güncellenir. “getDrawMatches” fonksiyonu ile tüm kategorilerde tamamlanan maçlar tahmin sonuçlarıyla beraber listelenir.

5.12 Draw Subscription Class

Draw kategorisine üye olacak kullanıcılar için üyelik işlemlerini gerçekleştiren komutların bulunduğu sınıf yapısıdır. “tanımla” isimli fonksiyon ile Draw kategorisine üye olmak için kullanılan ekranda bulunan nesneler tanımlanır. Nesnelere erişim için gerekli arka plan komutları “setOnClickListener” isimli fonksiyon ile tanımlanmıştır. Seçilen paket türüne göre “billingInterface” sınıfı kullanılarak üyelik alımı gerçekleştirilir.

6 Kullanılan kütüphaneler ve nesneler

6.1 Android Context

Android sistemi tarafından soyut bir şekilde sunulan, uygulama hakkında global bilgiye sahip ara yüzdür. Bir nevi uygulamanın komponentlerinin kimliğidir denilebilir. Activity başlatmak, broadcast atmak, view güncellemek, shared pref’e erişmek vb. işlemler için her zaman context’e ihtiyaç duyulur. [1]

6.2 Android Intent ve Android Bundle

Bir activity’den başka bir activity’ e geçişi sağlayan veya bilgi aktarımını yapan nesnelerdir. [2]

6.3 Android SpannableString ve ClickableSpan

Bu kütüphaneler istenilen karakterden başlayıp istenmeyen karaktere kadar stringi şekillendirmede kullanılır. [3]

6.4 Android TextPaint

TextPaint, Paint’in metin ölçümü ve çizimi sırasında kullanılan bazı ekstra verilere yer bırakan bir uzantısıdır. [4]

6.5 Android TextUtils

TextUtils, android programlamada çeşitli testler ve sorgular yapmak için kullanılan araçları içinde barındıran bir kütüphanedir. TextUtils'deki birden çok yöntem arasında, TextUtils.isEmpty yaygın bir örnektir. Bu yöntem CharSequence parametresini alır ve parametre null veya uzunluk 0 ise true değerini döndürür. [5]

6.6 Android LinkMovementMethod

Metin arabelleğindeki bağlantıları geçen ve gerekirse kaydıran bir hareket yöntemidir. DPad Center veya Enter ile bağlantılara tıklamayı destekler. [6]

6.7 Android View

Bir Görünümdür. Ekranda dikdörtgen bir alanı kaplar ve hem çizim hem olay işlemeden sorumludur. Görünüm, etkileşimli kullanıcı arabirimi bileşenleri (düğmeler, metin alanları vb.) oluşturmak için kullanılan widget'ların temel sınıfıdır. [7]

6.8 Android Button

Tıklanabilir ve işlevsel butonlar oluşturmak için kullanılan kütüphanedir. [8]

6.9 Android EditText

Kullanıcıların text oluşturmaları için gereken text alanlarını sağlayan yapıdır. [9]

6.10 Android TextView

Uygulamada gözükmesi istenen textler için kullanılan görüntüleme yapısıdır. [10]

6.11 Android Toast

Uygulama içinde ekranda küçük bildirimlerin oluşturulmasını sağlar. [11]

6.12 Android AlertDialog

AlertDialog, Dialogdan türeyen kullanıcının, ekranda pop up şeklinde bir görüntü görmesini sağlayan yapıdır. Kullanıcıya bilgi mesajı gibi çıktılar üretilmesini sağlar. [12]

6.13 Android AppCompatActivity

Son gelen özelliklerin eski apilerde kullanılabilir olmasını sağlar. Geriye dönük uygulama desteği sunan bir yapıdır. [13]

6.14 Android Toolbar

Uygulama içindeki araç çubuklarının ve özelliklerinin kullanılabilmesini sağlayan kütüphanedir. [14]

6.15 Android GoogleSignIn

Google hesabı ile üyelik oluşturmayı sağlayan hazır google kütüphanesidir. [15]

6.16 Android GoogleSignInAccount

Bu yapı, Androidde kullanıcının giriş yaptığı google hesabı üzerinden uygulama için id, token vb. değerleri tutan, istenildiği zaman bu değerleri servis eden bir nesnedir. [16]

6.17 Android WebView

WebView yapısı, uygulama içindeki bağlantıların ve diğer web içeriklerinin açılmasına izin veren bir yapıdır. [17]

6.18 Android ProgressDialog

ProgressDialog ile kullanıcıya bilgi mesajı vererek; ekranda dönecek şekilde bir dialog gösterilir. [18]

6.19 Android DialogInterface

Gösterilebilen, kapatılabilen, iptal edilebilen ve tıklanabilen düğmelere sahip olabilen bir iletişim türü sınıfı tanımlayan arabirimdir. [19]

6.20 Android ColorDrawable

Canvas'ı belirli bir renkle dolduran özel bir Drawable'dır. Canvas sınıfı "draw" çağrılarını tutar. [20]

6.21 Android Uri

Amacı soyut veya fiziksel kaynağı belirlemek olan bu yapı, dataya değeri atamak vb. işlemler amacıyla da kullanılır. Bu metoda giriş olarak Uri nesnesi gönderilir. Uri, nesne bir karakter diziyi alıp böler ve bölünen parçaları içerir. [21]

6.22 Android Log

Hata, uyarı, bilgi, hata ayıklama, verbose şeklinde azdan çoğa sıralamaya sahip olan ve bunların yürütülmesi için yardımcı olan bir yapıdır. [22]

6.23 Android WindowManager

Uygulamaların pencere yöneticisiyle haberleşmek için kullandığı ara yüzdür. [23]

6.24 Android GoogleSignInClient

GoogleSignInClient, GoogleApi'yi genişleten ve Google Sign In API ile etkileşim kurmasına yardımcı olan bir istemcidir. [24]

6.25 Android GoogleSignInOptions

GoogleSignInOptions, Google account'u yapılandırmak için kullanılan seçenekleri içerir. [25]

6.26 Android Exception

Google Play hizmetlerine yapılan bir çağrı başarısız olduğunda görev tarafından döndürülecek istisna yapısıdır. [26]

6.27 Android RippleBackground

Uygulama giriş ekranında hareketli görsel efektlerin bulunduğu bir arayüz tasarım seçeneği sunar. [27]

6.28 JetBrains NotNull

Bir değişkenin, parametrenin veya dönüş değerinin geçersizliğinin kontrol edilmesine izin verir. Yöntem hiyerarşileri boyunca sözleşmelerin kontrol edilmesine yardımcı olur. IntelliJ IDEA, sözleşmenin ihlal edildiğini tespit ederse, tespit edilen sorunu bildirir. NullPointerException'ın oluşabileceği kodu gösterir. [28]

6.29 Android LinearLayout

LinearLayout, düzen yönünü android:orientation niteliğiyle belirtilebildiği ve uygulama dizaynını dikey veya yatay olarak tek bir yönde hizalayan bir görünüm grubudur. [29]

6.30 Android SupressLint

Lint'in açıklamalı öge için belirtilen uyarıları yok sayması gerektiğini belirtir. [30]

6.31 Android ActionBarDrawerToggle

Gezinme çekmeceleri için önerilen tasarımı uygulamak amacıyla DrawerLayout ve ActionBar çerçevesinin işlevselliğini birbirine bağlamak için kullanışlı bir yoldur. [31]

6.32 Android GravityCompat

Gravity'den daha yeni işlevlere erişmek için uyumluluk sağlamaya yarar sağlayan yapıdır. [32]

6.33 Android FragmentManager

FragmentManager, uygulamanın parçaları üzerinde ekleme, kaldırma veya değiştirme gibi eylemler gerçekleştiren ve bunları arka yığına eklemekten sorumlu sınıftır. [33]

6.34 Android FragmentStatePagerAdapter

Parçanın durumunun kaydedilmesini ve geri yüklenmesini yöneten bir adapterdir. Her sayfayı yönetmek için Fragment kullanan bir yapıdır. [34]

6.35 Default Transformer

Ayrıştırıcının biçim bileşenini varsayılan transformatörlerle yapılandırır. [35]

6.36 Android NavigationView

Uygulama için standart bir gezinme menüsünü temsil eder. Menü içerikleri bir menü kaynak dosyası tarafından doldurulabilir. NavigationView tipik olarak bir DrawerLayout içine yerleştirilir. [36]

6.37 Google Gson

Gson, Java Nesnelerini JSON temsillerine dönüştürmek için kullanılabilen bir Java kitaplığıdır. Bir JSON dizesini eşdeğer bir Java nesnesine dönüştürmek için de kullanılır. [37]

6.38 Java HostnameVerifier

Ana bilgisayar doğrulaması için temel arabirimdir. Doğrulama mekanizması bağlantıya izin verilip verilmeyeceğini belirlemek için bu arabirimin uygulayıcılarını geri arayabilir. [38]

6.39 Java SSL

SSL, bir web sitesi ile internet tarayıcısı arasında şifrelenmiş iletişime olanak sağlayan bir tür dijital güvenlik teknolojisidir. SSL kimlik doğrulama için hangi kimlik doğrulama bilgilerinin uzak Ana Bilgisayar'a gönderilmesi gerektiğine karar verir. [39]

6.40 Java TrustManager

TrustManager, uzak bağlantının güvenilir olup olmadığını, yani uzak tarafın istediği kişi olup olmadığını belirler. [40]

6.41 Android ContextThemeWrapper

Sarılmış bağlamın temasının değiştirilmesine izin veren bir bağlam sarmalayıcıdır. [41]

6.42 Google FirebaseRemoteConfig

Uygulamada parametreleri tanımlamak, bunların değerlerini bulutta güncellemek için Firebase Remote Config'i kullanarak uygulama güncellemesi dağıtmadan uygulamanın görünümünü ve davranışını değiştirmede kullanılır. [42]

6.43 LayoutInflater

Girdi olarak XML dosyası alır ve görüntü nesneleri oluşturur. [43]

6.44 RecyclerView

Bir veri kümesindeki öğeleri temsil eden görünümmler sağlamaktan sorumlu, gelişmiş bir sınıftır. [44]

6.45 LinearLayoutManager

Ekrana yatay veya dikey olarak yansıtılmak istenen verinin belirlenen görsellikte kullanıcıya sunulmasını sağlar. [45]

6.46 SwipeRefreshLayout

Dikey kaydırmayı algılayarak çalışan, kullanıcıya sürekli dönen bir Progress Bar gösteren bir kütüphanedir. [46]

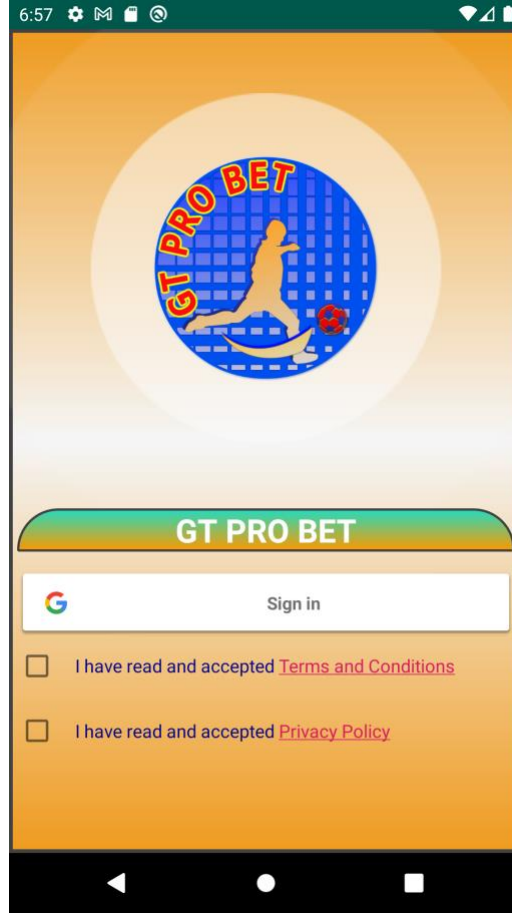
YAZILIM PLATFORMUNUN GELİŞTİRİLMESİ ve UYGULAMA

1) Giriş Ekranı Tasarımı Süreci

Kullanıcıların uygulamaya giriş yapabilmesi için Şekil 2’de kodları bulunan “act_login.xml” isimli bir ekran tasarlandı. Şekil 3’de görüldüğü üzere bu ekranda giriş butonu, kullanım koşulları ve gizlilik politikaları için onay butonları bulunmaktadır. Bu sayfa xml dili ile yazıldı. Arka plan için “loginscreen_background” isimli bir xml dosyası oluşturuldu. Bu dosya sadece renk tasarımını barındırmaktadır. loginscreen_background isimli dosya giriş ekranı klasöründe tanıtıldı. Daha sonra uygulama için tasarlanan logo burada tanımlandı. Google ile giriş yapılabilmesi ve sözleşmeler için butonlar tanımlandı. Tüm bu tanımlanan elemanlar ekranda düzenli olması adına “LinearLayout” isimli yapı kullanıldı. Bu yapı, uygulama penceresinde dikey konumda sıralı bir görünüm oluşmasını sağlamaktadır. Giriş ekranı, kullanıcıların uygulamayı kullanmaya başladıklarında ilk karşılaşacakları, ilk izlenimleri oluşturacak olan ekran olduğu için büyük önem arz etmektedir.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <com.skyfishjy.library.RippleBackground
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="295.00dp"
        android:layout_gravity="center"
        app:rb_color="#FBFBFB"
        app:rb_duration="2000"
        app:rb_radius="32.00dp"
        app:rb_rippleAmount="3"
        app:rb_scale="6">
        <ImageView
            android:id="@+id/loginscreen_centerimage"
            android:layout_width="140.00dp"
            android:layout_height="140.00dp"
            android:layout_centerInParent="true"
            android:background="@drawable/logo" />
        </com.skyfishjy.library.RippleBackground>
    <TextView
        android:id="@+id/loginscreen_banner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5.00dp"
        android:layout_marginRight="5.00dp"
        android:background="@drawable/rounded_edge"
        android:gravity="center"
        android:text="GT PRO BET"
        android:textColor="@android:color/white"
        android:textSize="20.00sp"
        android:textStyle="bold" />
    </LinearLayout>
```

Şekil 2 act_login.xml kodları



Şekil 3 act_login.xml tasarım ekranı

2) Mainactivity Oluşturma Süreci

Oturum açıldığında kullanıcıya servis edilecek “activity_main.xml” isimli bir ana sayfa ekranı tasarlandı. Görsel olarak ayrı bir yapı oluşturması için “toolbar.xml” isimli dosya eklendi. Toolbar, üzerinde simgelerin ve araçların listelenebildiği bir araç çubuğudur. MainActivity, bir uygulamada tüm işlerin yürütülmesi için gerekli olan bir sınıf yapısıdır. Uygulama geliştirilmeye devam ettikçe bu sınıf güncellenir ve yeni eklenen tüm sınıflar burada tanımlanır. Bir başlangıç sınıfı olduğu için MainActivity, en önemli yapıdır. activity_main.xml klasörü MainActivity.java klasörüne bağlandı. Java klasörleri, uygulamada çalışan tüm arka plan komutlarını içeren yapılardır. Uygulama güncellemeleri ihtiyacının kontrolü için şekil 4’te kodları bulunan “onUpdateNeeded” isimli fonksiyon oluşturuldu. Bu fonksiyon, uygulama çalıştığında uygulama sürümünün güncel olma durumunu kontrol etmekte ve yeni sürüm mevcut ise kullanıcıya bunu bildirmektedir. MainActivity.java klasörü tüm sınıfların birleşeceği bir yapıdır ve gelecek tüm değişiklikler bu dosyayı güncelleyecektir.


```

@Override
public void onUpdateNeeded(String updateUrl) {
AlertDialog dialog = new AlertDialog.Builder( context: this)
.setTitle("New version available")
.setMessage("Please, update app to new version to continue.")
.setCancelable(false)
.setPositiveButton( text: "Update",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog,int which) {
redirectStore(updateUrl);
finish();
moveTaskToBack( nonRoot: true);
System.exit( status: 0);
}
}).setNegativeButton( text: "No, thanks",
new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
finish();
moveTaskToBack( nonRoot: true);
System.exit( status: 0);
}
}).create();
dialog.show();
}

```

Şekil 4 onUpdateNeeded isimli fonksiyonun kodu

3) Oturum Açma (Google) Yazılım Altyapısının Oluşturulması Süreci

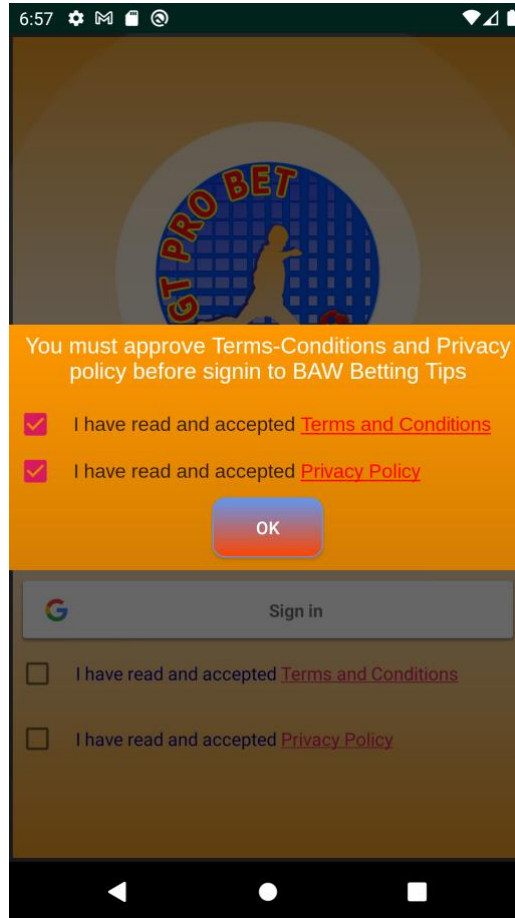
Oturum açma sürecinin arka plan kodları için Login.java isimli sınıf oluşturuldu. Giriş ekranının tasarımının bulunduğu act_login.xml isimli dosyanın tanımlaması, Login.java sınıfında “onCreate” fonksiyonu içerisinde yapıldı. Giriş sayfasının arka plan komutları bu sınıfta etkinleştirilecek. “onCreate” fonksiyonu ile Login sınıfı çalışmaya başlar. onCreate içince tüm tanımlamalar yapılır. Bu tanımlamalar, giriş ekranındaki butonların id’leridir. Giriş butonu işleme girdiğinde öncelikle kullanım koşulları ve sözleşme butonlarının işaretlenmeleri kontrol edilir. İşaretlenmeyen kullanım koşulu ve sözleşme butonu bulunur ise kullanıcıya şekil 5’te kodları bulunan “showMyCustomAlertDialog” isimli fonksiyon ile uyarı verilir. Bu fonksiyon, kullanıcıya şekil 6’da görülen ve doldurulması gereken “checkBox”ları uyarı olarak verir. Kullanıcı, açılan bu ekrandan işaretlemeleri tamamlamalıdır. İşaretlemeler tamamlandıktan sonra butona basılarak giriş sürecine devam edilir.

```

public void showMyCustomAlertDialog() {
    final Dialog dialog = new Dialog(context);
    dialog.setContentView(R.layout.custom_dialog);
    Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
    CheckBox checkBoxTermsDialog = dialog.findViewById(R.id.checkBoxTermsDialog);
    CheckBox checkBoxPrivacyDialog = dialog.findViewById(R.id.checkBoxPrivacyDialog);
    TextView privacyPolicyDialog = dialog.findViewById(R.id.privacyPolicyDialog);
    TextView termsOfAgreementDialog = dialog.findViewById(R.id.termsOfAgreementDialog);
    Button btniPtaI = dialog.findViewById(R.id.button_ipital);
    ClickableSpan clickableSpan1 = new ClickableSpan() {
        @Override
        public void onClick(View widget) {
            Intent intent = new Intent( packageContext: Login.this, Privacy.class);
            startActivity(intent);
            widget.cancelPendingInputEvents(); }
        @Override
        public void updateDrawState(@NonNull TextPaint ds) {
            super.updateDrawState(ds);
            ds.setColor(context.getResources().getColor(R.color.red));
            ds.setUnderlineText(true); }
    };
    ClickableSpan clickableSpan = new ClickableSpan() {
        @Override
        public void onClick(View widget) {
            Intent intent = new Intent( packageContext: Login.this, Terms.class);
            startActivity(intent);
            widget.cancelPendingInputEvents(); }
        @Override
        public void updateDrawState(@NonNull TextPaint ds) {
            super.updateDrawState(ds);
            ds.setColor(context.getResources().getColor(R.color.red));
            ds.setUnderlineText(true); }
    };
};

```

Şekil 5 showMyCustomAlertDialog fonksiyonu kodu



Şekil 6 Sözleşme kabulü uyarı ekranı

Tüm süreç tamamlandıktan sonra GoogleSignin ile oturum açma işlemi çalışmaya başlar. “tanımla” fonksiyonu ile giriş butonu tanımlanır. Tasarımda bulunan değişkenlerin class yapısına ayarlanması gerçekleşir. “tanımla” fonksiyonu, Google Sign in butonunun tanımlandığı fonksiyondur. “onStart” fonksiyonu ile giriş yapılacak hesap seçilir ve giriş yapılan hesabın kontrolü için şekil 7’de kodları bulunan “girisCheck” fonksiyonuna kullanıcı mail adresi iletilir. girisCheck fonksiyonu, “retrofit” kütüphanesi kullanarak girdi olarak aldığı mail adresinin varlığını kontrol eder. Eğer böyle bir hesap yok ise ekrana uyarı verilir. Hesap mevcut ise “onLoggedIn” fonksiyonu çağırılır. onLoggedIn fonksiyonu, giriş işleminin son kısmını gerçekleştiren fonksiyondur. Süreci tamamlar, kullanıcı bilgilerini MainActivity class’ına iletir ve Main ekranına geçiş işlemini sağlar.

```
public void girischeck(String userMailAdres) {
    Call<LoginPageModel> req = ManagerAll.getInstance().girisKontrol(userMailAdres);
    req.enqueue(new CallbackWithRetry<LoginPageModel>() {
        @Override
        public void onResponse(Call<LoginPageModel> call, retrofit2.Response<LoginPageModel> response) {
            if (response.body() != null) {
                if (response.body().isTf())
                {
                    onLoggedIn(account);
                }
                else
                {
                    String mesaj = "this account (" + account.getEmail() + ") " +
                        "has been disabled. Please contact with us via email support@betwinnergame.com";
                    AlertDialog.Builder builder = DialogHelper.alertBuilder(context: Login.this);
                    builder.setMessage(mesaj);
                    builder.setCancelable(false);
                    builder.setPositiveButton(text: "OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    });
                    builder.show();
                    googleSignInClient.signOut();
                    mProgress.dismiss();
                }
            }
        }
    })
}

@Override
public void onFailure(Call<LoginPageModel> call, Throwable t) {
    super.onFailure(call, t);
    Toast.makeText(getApplicationContext(), text: "User login error", Toast.LENGTH_LONG).show();
}
```

Şekil 7 girisCheck isimli fonksiyonun kodu

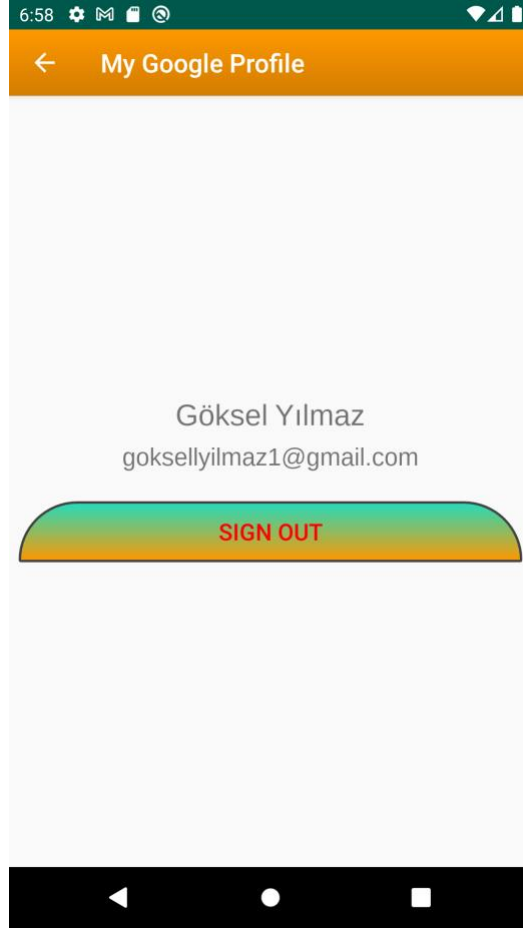
4) Oturum Kapatma (Google) Yazılım Altyapısının Oluşturulması Süreci

Oturum kapatma sürecinin arka plan kodları için Logout.java isimli sınıf oluşturuldu. Logout sınıfı, uygulama içerisinde açık olan oturumdan çıkıp giriş ekranına dönmeyi sağlayan sınıftır. Çıkış butonuna basılınca çıkış için emin olup olunmadığının sorulduğu Şekil 9’da görülen 2. Bir ekrana yönlendirme gerçekleştirilir. Burada çıkış gerçekleştirilmek istenirse Logout.java sınıfının içerisindeki onCreate fonksiyonu çalışır. onCreate Fonksiyonu class yapısını başlatan fonksiyondur. “tanımla” fonksiyonu ile kullanıcı bilgileri çağırılır ve çıkış yapılacak hesap bilgileri ekranda yer alır. tanımla Fonksiyonu, çıkış yapacak kullanıcının bilgilerini ekranda yazdırmak için tasarımda verilen isimlendirmeleri class yapısındaki değişkenlere atamalarını gerçekleştiren fonksiyondur. Ekranda yer alan başlık oluşturulur ve gösterilir. Çıkış işlemleri için gerekli googleSocialLogin isimli fonksiyon çağırılır. “googleSocialLogin” fonksiyonu, Google servis çağrısını başlatan fonksiyondur. Burada “googleApiClient” oluşturulur. “onActivityResult” fonksiyonu ile servisten alınan response bilgisi şekil 8’de kodları bulunan “handleResultGoogleSignIn” fonksiyonuna iletilir. handleResultGoogleSignIn fonksiyonu içerisinde çıkış yapmak isteyen kullanıcı bilgileri tutulur. Bu bilgiler ışığında “googleLogout” fonksiyonu çalıştırılır. googleLogout, oturumu “GoogleSignInApi” kütüphanesinin “signOut” fonksiyonu ile sonlandırmasını sağlayan fonksiyondur. Tüm işlemlerin sonunda gerçekleşir ve çıkış işlemi tamamlanır

```
private void handleResultGoogleSignIn(GoogleSignInResult result)
{
    if (result.isSuccess())
    {
        GoogleSignInAccount account = result.getSignInAccount();
        String socialFirstName = Objects.requireNonNull(account).getGivenName();
        String socialLastName = account.getFamilyName();
        String socialEmail = account.getEmail();
        String socialUserId = account.getId();
        Uri socialImageUrl = account.getPhotoUrl();
        tvName.setText(socialFirstName);
        tvSurname.setText(socialLastName);
        tvEmail.setText(socialEmail);
        Picasso.get()
            .load(socialImageUrl)
            .into(ivUser);
    }
}

public void googleLogout()
{
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback(status ->
    {
        Intent intent = new Intent( packageContext: Logout.this, Login.class);
        startActivity(intent);
        finish();
    });
}
```

Şekil 8 handleResultGoogleSignIn fonksiyonu kodları



Şekil 9 Oturumdan çıkış ekranı

5) Menü Tasarımı Süreci

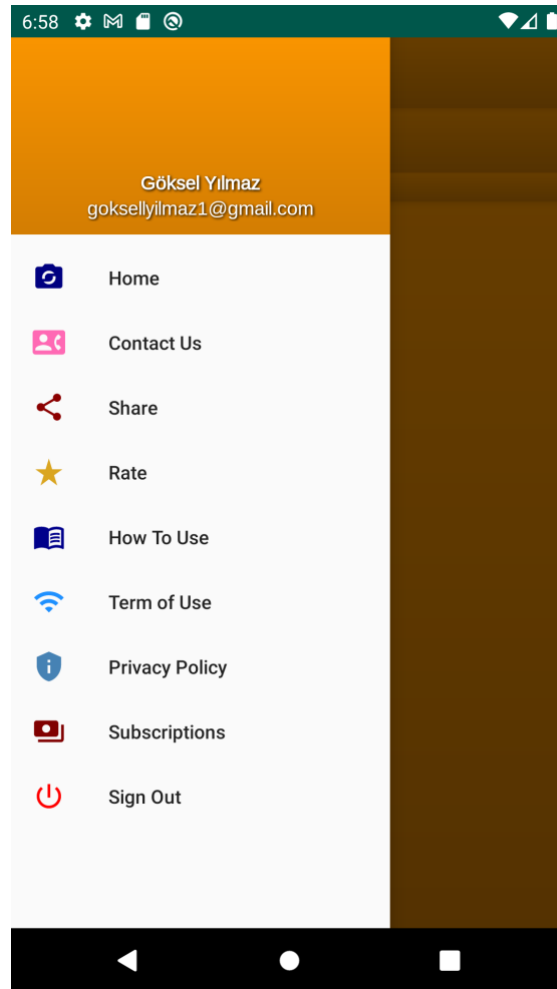
Kullanıcıların sayfalar arasında geçiş yapabilmesi için şekil 11’de görülen menü sayfası tasarlandı. Bu menü içerisinde ana sayfaya dönmek için, iletişim ekranına geçiş için, uygulamayı paylaşmak için, uygulamayı puanlamak için, uygulamanın nasıl kullanıldığını anlatıldığı ekrana geçiş için, sözleşme ve kullanım koşulları ekranına geçiş için, üyelik hakkında bilgi ekranı için ve çıkış yapmak için seçenekler bulunmaktadır. Bu “activity_main_drawer.xml” isimli menü dosyası, uygulamada tüm ekranlarda ulaşılabilir olması için MainActivity sınıfında tanımlanır. Menü içerisindeki butonların işlevlerinin yerine getirilebilmesi için şekil 10’da kodları görülen “onNavigationItemSelectedListener” isimli fonksiyon oluşturuldu. Burada öncelikle seçilen butonun id’si bir değişkende tutulur. Daha sonra burada bulunan koşullu ifadelerin süzgecinden geçer. O id hangi işleme karşılık geliyor ise o döndürülür. Ana sayfa butonu seçilmiş ise ana sayfaya yönlendirme, iletişim butonu seçilmiş ise iletişim ekranına yönlendirme, çıkış butonu seçilmiş ise çıkış ekranına yönlendirmeler gerçekleşir.

```

@Override
public boolean onNavigationItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.nav_home) {
        viewPager.setCurrentItem( item: 0, smoothScroll: true);
    } else if (id == R.id.nav_privacy_policy) {
        Intent intent = new Intent( packageContext: MainActivity.this, Privacy.class);
        intent.putExtra(Contact.GOOGLE_ACCOUNT, googleSignInAccount);
        startActivity(intent);
    } else if (id == R.id.nav_contact_us) {
        Intent intent = new Intent( packageContext: MainActivity.this, Contact.class);
        intent.putExtra(Contact.GOOGLE_ACCOUNT, googleSignInAccount);
        startActivity(intent);
    } else if (id == R.id.nav_how_to_use) {
        Intent intent = new Intent( packageContext: MainActivity.this, Howtouse.class);
        intent.putExtra(Contact.GOOGLE_ACCOUNT, googleSignInAccount);
        startActivity(intent);
    } else if (id == R.id.nav_term_of_use) {
        Intent intent = new Intent( packageContext: MainActivity.this, Terms.class);
        intent.putExtra(Contact.GOOGLE_ACCOUNT, googleSignInAccount);
        startActivity(intent);
    }
}

```

Şekil 10 onNavigationItemSelected isimli fonksiyonun kodları



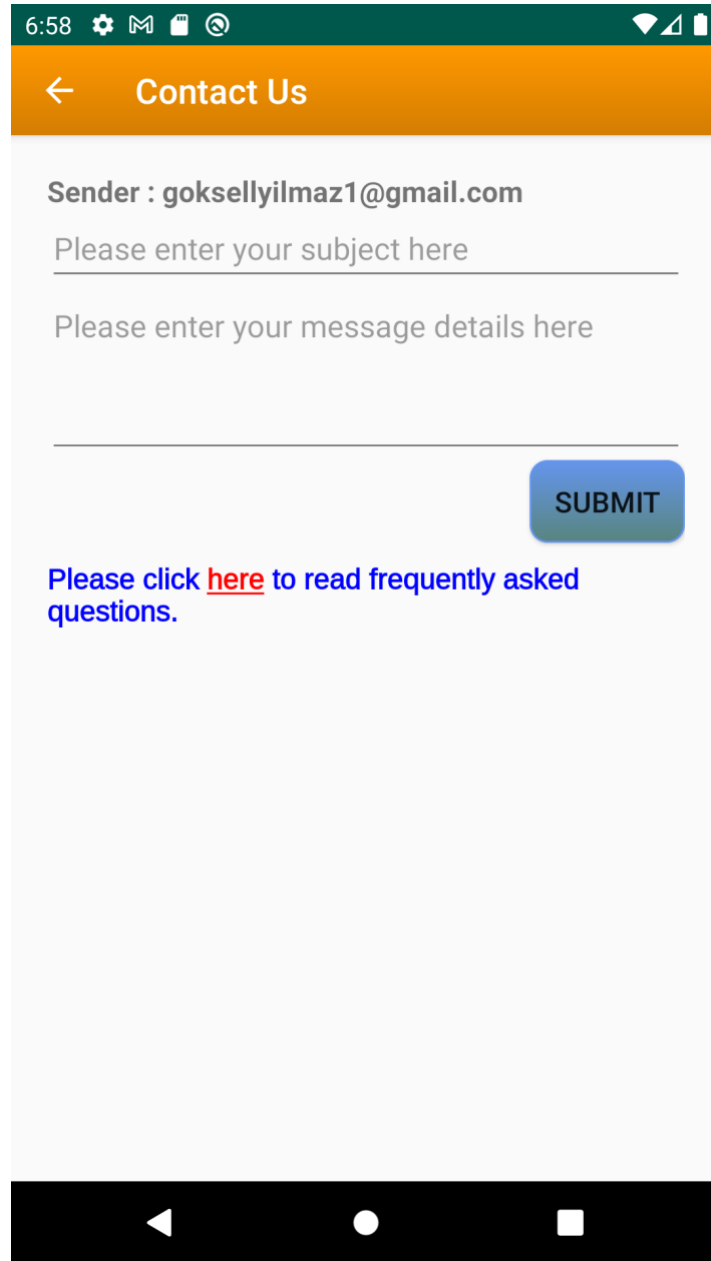
Şekil 11 Menü ekranı

6) İletişim Ekranı Tasarım Süreci

Kullanıcıların, yöneticiler ile iletişim kurabilmesi için bir şekil 13’te görülen iletişim sayfası oluşturuldu. Bu sayfanın tasarımı için Şekil 12’de kodları görülen “act_contact.xml” isimli bir dosya oluşturuldu. Sayfada sıralı bir görüntü oluşmasını sağlayan LinearLayout yapısı içine, sayfa isminin yazılı olacağı bir toolbar eklendi. Göndericinin adının yazılı olacağı bir TextView eklendi. Mesaj konusunun ekleneceği bir EditText eklendi. Mesajın ekleneceği bir EditText eklendi. Tüm bunların altına gönderim sağlaması için şekil 14’te kodları görünen buton eklendi ve tasarım süreci tamamlandı.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <include
        android:id="@+id/toolbarContactUs"
        layout="@layout/toolbar" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="16.00dp"
        android:paddingTop="16.00dp"
        android:paddingRight="16.00dp"
        android:paddingBottom="16.00dp">
        <TextView
            android:id="@+id/txtSender"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="13.00sp"
            android:textStyle="bold" />
        <EditText
            android:id="@+id/txtSubject"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/please_enter_your_subject_here"
            android:autofillHints=""
            android:inputType="text" />
        <EditText
            android:id="@+id/txtMessage"
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:autofillHints=""
            android:inputType="textMultiLine"
            android:gravity="top"
            android:hint="@string/please_enter_your_message_details_here"
```

Şekil 12 act_contact.xml kodları



Şekil 13 İletişim ekranı

```
<Button
    android:id="@+id/btnOK"
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:background="@drawable/contactbuttonshapesubmit"
    android:textSize="13.00sp"
    android:text="Submit" />
```

Şekil 14 act_contact.xml sayfasında bulunan buton kodları

7) Uygulama Kullanım Kılavuzu (How To Use) Ekranı Oluşturma Süreci

Kullanıcıların uygulamayı nasıl kullanacaklarının bilgilendirmesi için şekil 16’da görülen sayfa oluşturuldu. Bu sayfanın tasarımının kodlaması için şekil 15’te kodları görülen “act_howtouse.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde bir toolbar ve bilgilendirmelerin bir web siteden alınabilmesi için WebView isimli yapı eklendi. WebView, internet sitelerindeki bilgilerin mevcut ekrana yansıtılmasını sağlamaktadır. act_howtouse.xml isimli tasarım dosyasının arka plan komutları için “Howtouse.java” isimli sınıf oluşturuldu. Oluşturulan bu sınıfın içerisinde tasarım dosyası olan act_howtouse.xml dosyası tanımlandı. Tasarımda eklenen WebView içerisinde bilgilendirmeleri yansıtmak için şekil 17’de kodları görülen “tanımla” isimli fonksiyon oluşturuldu. Bu fonksiyon içerisinde “loadUrl” fonksiyonu ile bir web sitesi url’i çalıştırılmakta ve WebView içerisinde listelenme işlemi sağlanmaktadır.

```
<?xml version="1.0" encoding="utf-8"?>

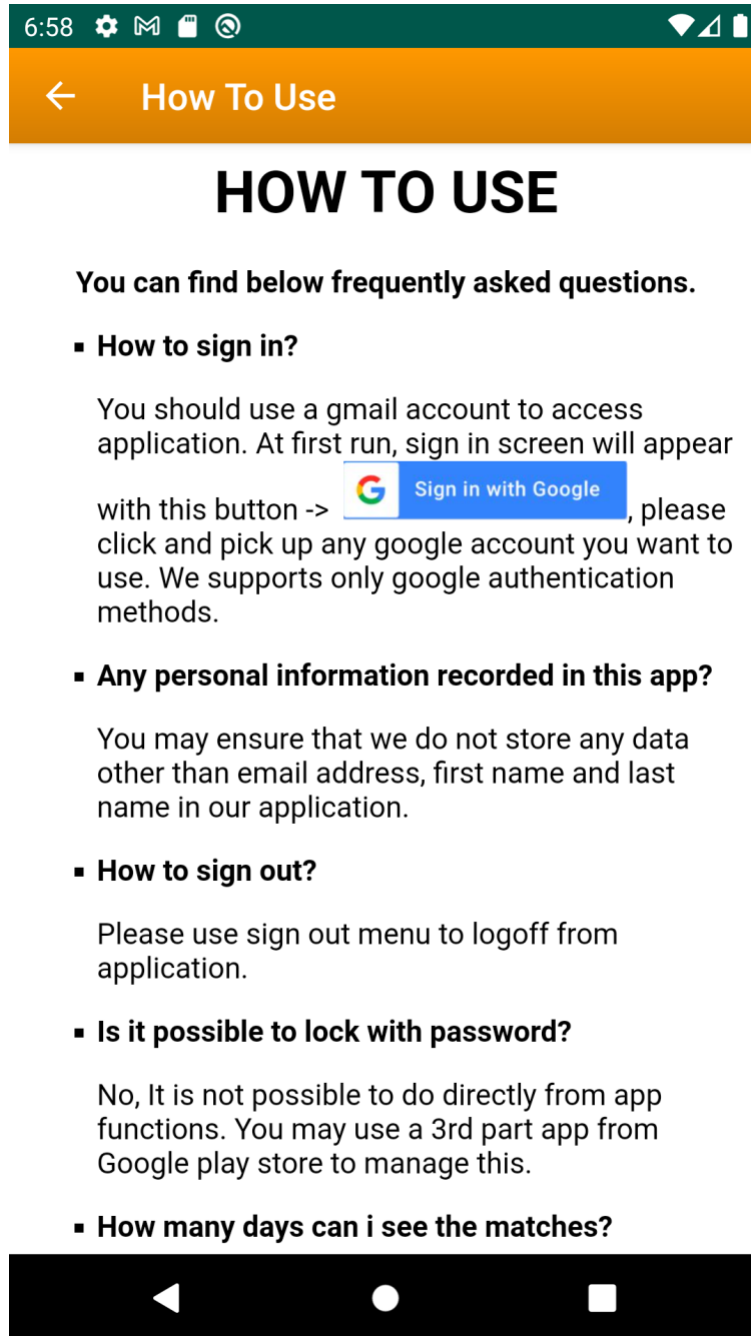
<LinearLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:fitsSystemWindows="true"
    android:orientation="vertical">

    <include
        android:id="@+id/toolbarHowtouse"
        layout="@layout/toolbar" />
    <WebView
        android:id="@+id/webViewhowtouse"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        />

</LinearLayout>
```

Şekil 15 act_howtouse.xml kodları



Şekil 16 Uygulama nasıl kullanılır ekranı

```
public void tanimla()  
{  
    web = findViewById(R.id.webViewhowtouse);  
    web.loadUrl("howtouse/index.html");  
}
```

Şekil 17 Howtouse sınıfında bulunan tanimla isimli fonksiyonun kodları

8) Uygulama Kullanım Koşulları (Term Of Use) Ekranı Oluşturma Süreci

Kullanıcıların kullanım koşulları ve sözleşmeler hakkında bilgilendirilmesi için şekil 19’da görülen bilgilendirme sayfası oluşturuldu. Bu sayfanın tasarımının kodlaması için şekil 18’de kodları görülen “act_terms.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde bir toolbar ve bilgilendirmelerin bir web siteden alınabilmesi için WebView isimli yapı eklendi. WebView, internet sitelerindeki bilgilerin mevcut ekrana yansıtılmasını sağlamaktadır. Burada sergilenecek bilgiler için “terms.html” dosyası oluşturuldu ve internet sitesindeki mimari kullanılarak tüm bilgiler girildi. act_terms.xml isimli tasarım dosyasının arka plan komutları için “Terms.java” sınıfı oluşturuldu. Oluşturulan bu sınıfın içerisinde tasarım dosyası olan act_terms.xml dosyası tanımlandı. Tasarımda eklenen WebView içerisinde bilgilendirmeleri yansıtmak için şekil 20’de kodları görülen “tanımla” isimli fonksiyon oluşturuldu. Bu fonksiyon içerisinde “loadUrl” fonksiyonu ile bir web sitesi url’i çalıştırılmakta ve WebView içerisinde listelenme işlemi sağlanmaktadır.

```
<?xml version="1.0" encoding="utf-8"?>

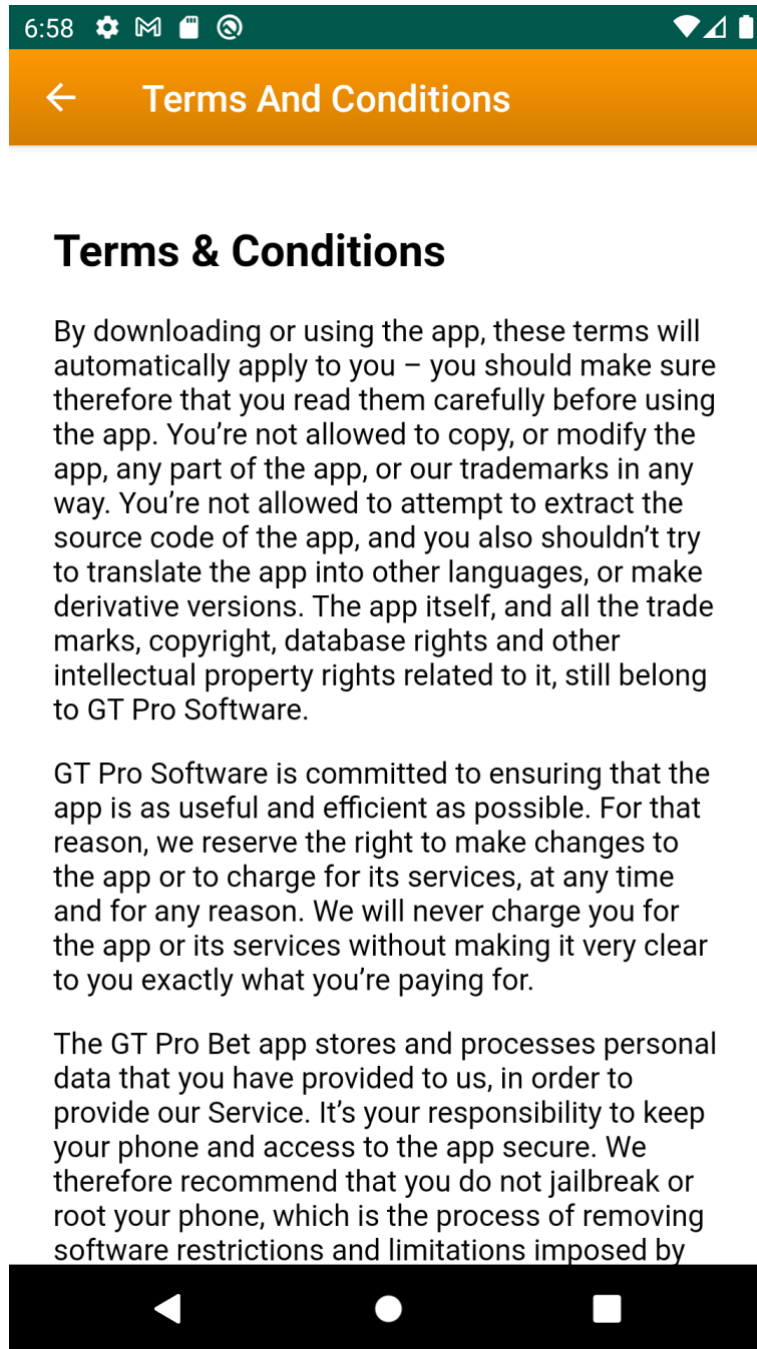
<LinearLayout
    |
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:fitsSystemWindows="true"
    android:orientation="vertical">

    <include
        android:id="@+id/toolbarTermsCond"
        layout="@layout/toolbar" />

    <WebView
        android:id="@+id/webViewTerms"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp"
        tools:ignore="MissingConstraints" />

</LinearLayout>
```

Şekil 18 act_terms.xml kodları



Şekil 19 Şartlar ve koşullar ekranı

```
public void tanimla() {  
    web = findViewById(R.id.webViewTerms);  
    web.loadUrl("file:///android_asset/terms.html");  
}
```

Şekil 20 Terms sınıfında bulunan tanimla isimli fonksiyonun kodları

9) Gizlilik Politikası (Privacy Policy) Ekranı Oluşturma Süreci

Kullanıcıların gizlilik politikası hakkında bilgilendirilmesi için şekil 22’de görülen bilgilendirme sayfası oluşturuldu. Bu sayfanın tasarımının kodlaması için şekil 21’de kodları görülen “act_privacy.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde bir toolbar ve bilgilendirmelerin bir web siteden alınabilmesi için WebView isimli yapı eklendi. WebView, internet sitelerindeki bilgilerin mevcut ekrana yansıtılmasını sağlamaktadır. Burada sergilenecek bilgiler için “privacy.html” dosyası oluşturuldu ve internet sitesindeki mimari kullanılarak tüm bilgiler girildi. act_privacy.xml isimli tasarım dosyasının arka plan komutları için “Privacy.java” sınıfı oluşturuldu. Oluşturulan bu sınıfın içerisinde tasarım dosyası olan act_privacy.xml dosyası tanımlandı. Tasarımda eklenen WebView içerisinde bilgilendirmeleri yansıtmak için şekil 23’de kodları görülen “tanımla” isimli fonksiyon oluşturuldu. Bu fonksiyon içerisinde “loadUrl” fonksiyonu ile bir web sitesi url’i çalıştırılmakta ve WebView içerisinde listelenme işlemi sağlanmaktadır.

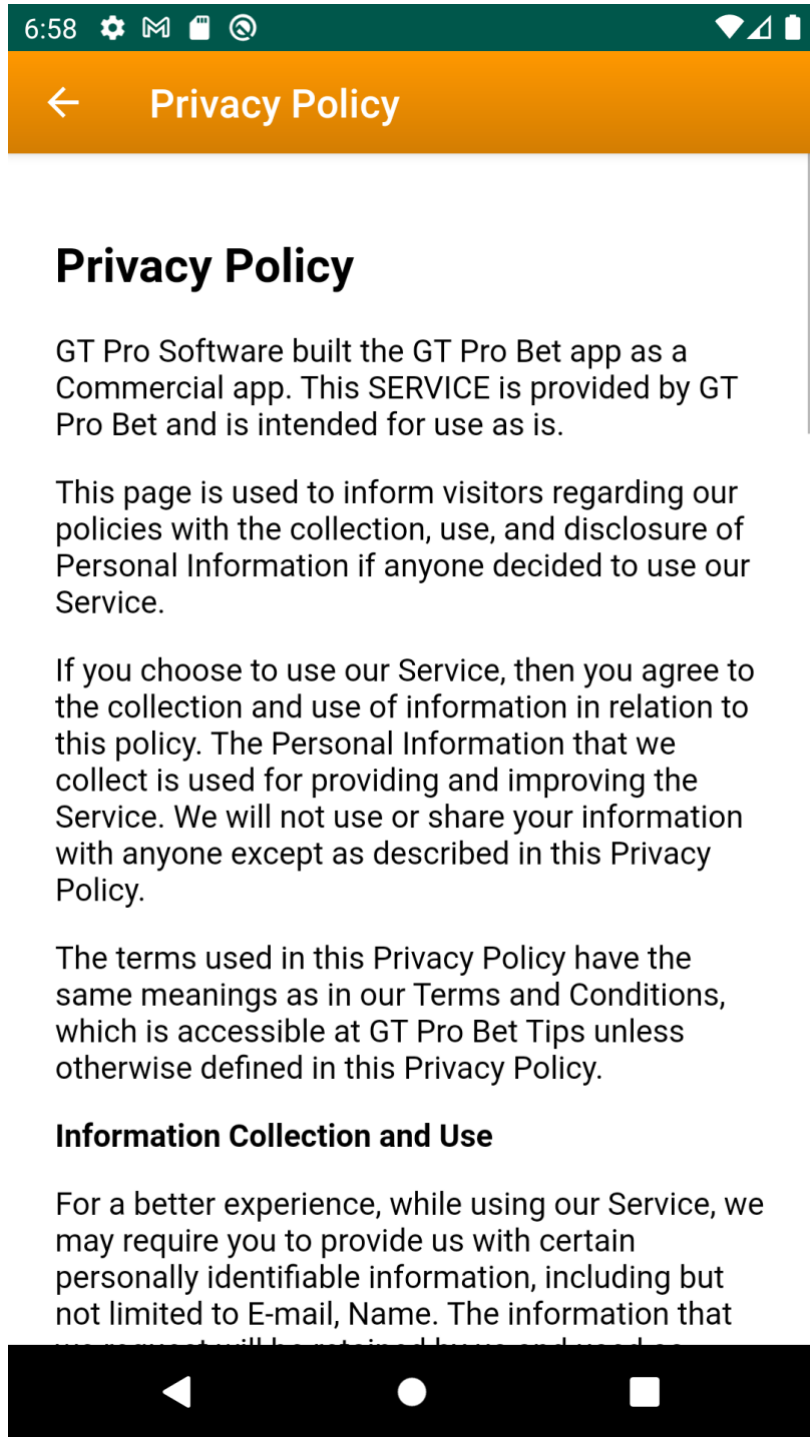
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:fitsSystemWindows="true"
    android:orientation="vertical">

    <include
        android:id="@+id/toolbarPrivacy"
        layout="@layout/toolbar" />

    <WebView
        android:id="@+id/webViewPrivacy"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp"
        tools:ignore="MissingConstraints" />

</LinearLayout>
```

Şekil 21 act_privacy.xml kodları



Şekil 22 Gizlilik politikası ekranı

```
public void tanimla() {  
    web = findViewById(R.id.webViewPrivacy);  
    web.loadUrl("file:///android_asset/privacy.html");  
}
```

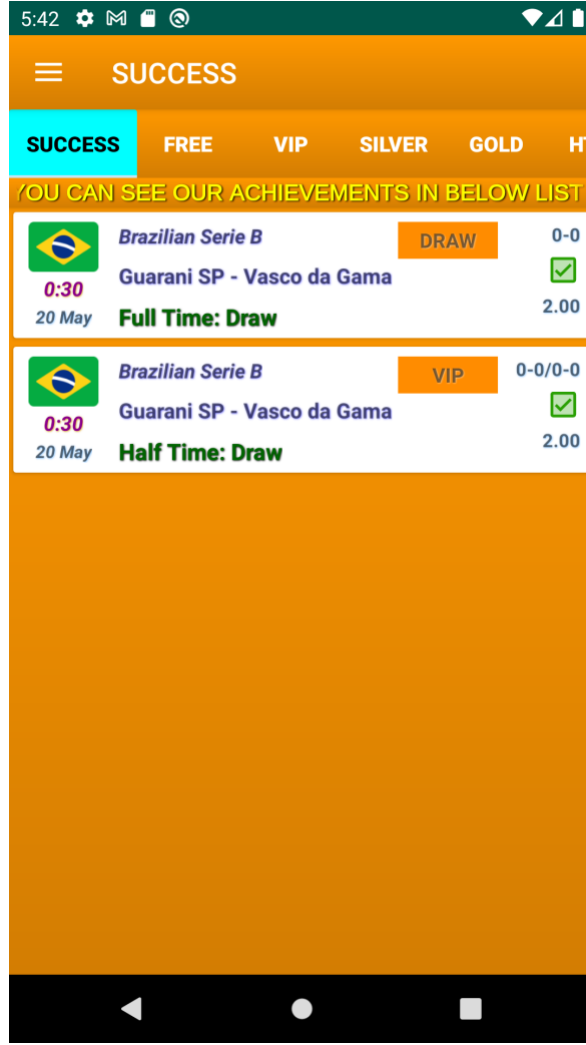
Şekil 23 Privacy sınıfında bulunan tanimla isimli fonksiyonun kodları

10) Sonuçlanan Maçlar Ekranı Oluşturma Süreci

Kullanıcıların sonuçlanan maçların detaylarını öğrenebilmeleri için şekil 25'te görülen tahmin sonuçları bilgilendirme sayfası oluşturuldu. Bu sayfanın tasarımının kodlanması için şekil 24'te kodları görülen "frag_proof.xml" isimli dosya oluşturuldu. LinearLayout yapısı içerisinde "RecyclerView" elemanı eklendi. RecyclerView, alt alta ürün listelemek için kullanılan bir yapıdır. frag_proof.xml isimli tasarım dosyasının "Success" isimli sınıfa bağlantısı sağlandı. Success sınıfında tasarımda kullanılan nesneler tanımlandı. "getSuccessMatches" isimli fonksiyon ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen maçlar, maçın sonuçları, başarı durumu bilgileri listelendi.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_marginTop="?attr/actionBarSize"
android:layout_alignParentBottom="true"
android:background="@drawable/gradient_one"
android:id="@+id/MacScreenProof"
tools:context=".fragments.Success">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/Proof_Match_Recyclerview"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        android:layout_width="match_parent"
        android:clipToPadding="false"
        android:layout_height="wrap_content">
    </androidx.recyclerview.widget.RecyclerView>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/proofnomatchtxt"
        android:visibility="invisible"
        android:layout_gravity="center_vertical"
        android:layout_marginTop="40.00dp"
        android:textSize="12.00dp"
        android:gravity="center"
        android:text="@string/there_are_no_match_to_display">
    </TextView>
</LinearLayout>
```

Şekil 24 frag_proof.xml kodları



Şekil 25 Sonuçlanan maçlar ekranı

11) Free Kategorisi Ekranı Oluşturma Süreci

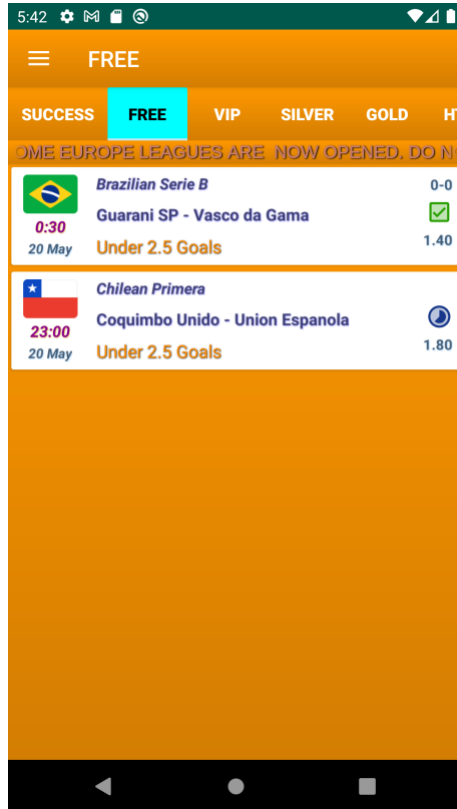
Tüm kullanıcıların erişimine açık olan şekil 27’de görülen tahmin ekranı oluşturuldu. Bu sayfada maç tahminleri ve maçlar sona erdikten sonra tahmin başarı durumları listelenmektedir. Bu sayfanın tasarımının kodlanması için şekil 26’da kodları görülen “frag_free.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta ürün listelemek için kullanılan bir yapıdır. frag_free.xml isimli tasarım dosyasının “Free” isimli sınıfa bağlantısı sağlandı. Free sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına bağlantı sağlandı. “getFreeMatches” isimli fonksiyon kullanılarak bağlantı sağlanan diğer sınıflardan çekilen maç verileri ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen maçlar, maçların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe göre güncellemektedir ve sürekli aktiftir.


```

<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginTop="?attr/actionBarSize"
    android:layout_alignParentBottom="true"
    android:background="@drawable/gradient_one"
    android:id="@+id/MacScreenFree"
    tools:context=".fragments.Free">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/Free_Match_Recyclerview"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        android:layout_width="match_parent"
        android:clipToPadding="false"
        android:layout_height="wrap_content">
    </androidx.recyclerview.widget.RecyclerView>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/freenomatchtxt"
        android:visibility="invisible"
        android:layout_gravity="center_vertical"
        android:layout_marginTop="40.00dp"
        android:textSize="12.00dp"
        android:gravity="center"
        android:text="There are no match to display">
    </TextView>
</LinearLayout>
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

```

Şekil 26 frag_free.xml kodları



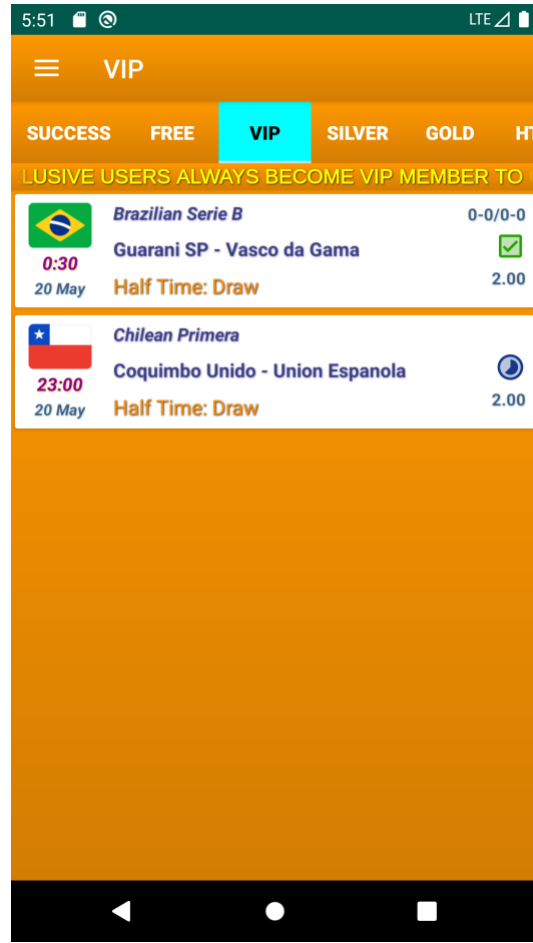
Şekil 27 Free kategorisi maçları ekranı

12) Vip Kategorisi Ekranı Oluşturma Süreci

Vip kategorisine üye olan kullanıcıların erişimine açık olan bir tahmin ekranı oluşturuldu. Bu sayfaya giriş yapabilmek için üye olmak gerekmektedir. Bu sayfada maç tahminleri ve maçlar sona erdikten sonra tahmin başarı durumları listelenmektedir. Sayfanın tasarımının kodlanması için şekil 28’de görülen “frag_vip.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta ürün listelemek için kullanılan bir yapıdır. frag_vip.xml isimli tasarım dosyasının “Vip” isimli sınıfa bağlantısı sağlandı. Vip sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına bağlantı sağlandı. “getVipMatches” isimli fonksiyon kullanılarak bağlantı sağlanan diğer sınıflardan çekilen maç verileri ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen maçlar, maçların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe göre güncellemektedir ve sürekli aktiftir.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/Vip_Match_Recyclerview"
        app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli..."
        android:layout_width="match_parent"
        android:clipToPadding="false"
        android:layout_height="wrap_content">
    </androidx.recyclerview.widget.RecyclerView>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/vipnomatchtxt"
        android:visibility="invisible"
        android:layout_gravity="center_vertical"
        android:layout_marginTop="40.00dp"
        android:textSize="12.00dp"
        android:gravity="center"
        android:text="There are no match to display">
    </TextView>
</LinearLayout>
```

Şekil 28 frag_vip.xml kodları



Şekil 29 Vip kategorisi maçları ekranı

13) Silver Kategorisi Ekranı Oluşturma Süreci

Silver kategorisine üye olan kullanıcıların erişimine açık olan bir tahmin ekranı oluşturuldu. Bu sayfaya giriş yapabilmek için üye olmak gerekmektedir. Bu sayfada maç tahminleri ve maçlar sona erdikten sonra tahmin başarı durumları listelenmektedir. Sayfanın tasarımının kodlanması için “frag_silver.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta ürün listelemek için kullanılan bir yapıdır. frag_silver.xml isimli tasarım dosyasının “Silver” isimli sınıfa bağlantısı sağlandı. Silver sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına bağlantı sağlandı. “getSilverMatches” isimli fonksiyon kullanılarak bağlantı sağlanan diğer sınıflardan çekilen maç verileri ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen maçlar, maçların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe göre güncellemektedir ve sürekli aktiftir.

14) Gold Kategorisi Ekranı Oluřturma Sreci

Gold kategorisine ye olan kullanıcıların erişimine açık olan bir tahmin ekranı oluşturuldu. Bu sayfaya giriş yapabilmek için ye olmak gerekmektedir. Bu sayfada ma tahminleri ve malar sona erdikten sonra tahmin başarı durumları listelenmektedir. Sayfanın tasarımının kodlanması için “frag_gold.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta rn listelemek için kullanılan bir yapıdır. frag_gold.xml isimli tasarım dosyasının “Gold” isimli sınıfa bağlantısı sağlandı. Gold sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına bağlantı sağlandı. “getGoldMatches” isimli fonksiyon kullanılarak bağlantı sağlanan diğerk sınıflardan çekilen ma verileri ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen malar, maların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe göre gncellemektedir ve srekli aktiftir.

15) Draw Kategorisi Ekranı Oluřturma Sreci

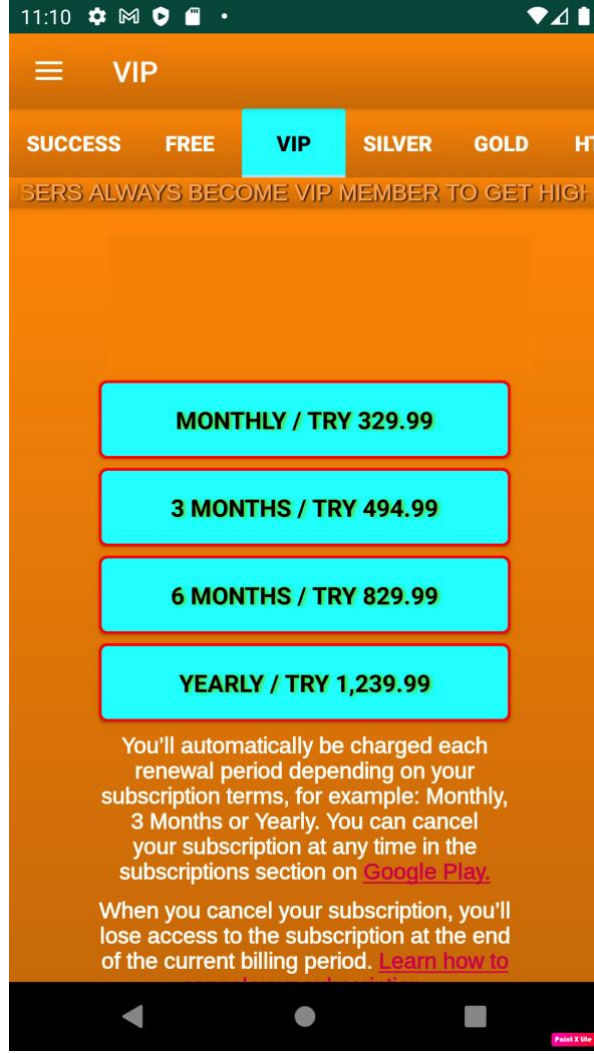
Draw kategorisine ye olan kullanıcıların erişimine açık olan bir tahmin ekranı oluşturuldu. Bu sayfaya giriş yapabilmek için ye olmak gerekmektedir. Bu sayfada ma tahminleri ve malar sona erdikten sonra tahmin başarı durumları listelenmektedir. Sayfanın tasarımının kodlanması için “frag_draw.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta rn listelemek için kullanılan bir yapıdır. frag_draw.xml isimli tasarım dosyasının “Draw” isimli sınıfa bağlantısı sağlandı. Draw sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına bağlantı sağlandı. “getDrawMatches” isimli fonksiyon kullanılarak bağlantı sağlanan diğerk sınıflardan çekilen ma verileri ile tasarımda oluşturulan RecyclerView içerisine tahmin edilen malar, maların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe göre gncellemektedir ve srekli aktiftir.

16) HT/FT Kategorisi Ekranı Oluřturma Sreci

HT/FT kategorisine ye olan kullanıcıların erişimine açık olan bir tahmin ekranı oluşturuldu. Bu sayfaya giriş yapabilmek için ye olmak gerekmektedir. Bu sayfada ma tahminleri ve malar sona erdikten sonra tahmin başarı durumları listelenmektedir. Sayfanın tasarımının kodlanması için “frag_htft.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde “RecyclerView” elemanı eklendi. RecyclerView, alt alta rn listelemek için kullanılan bir yapıdır. frag_htft.xml isimli tasarım dosyasının “Htft” isimli sınıfa baėlantısı saėlandı. Htft sınıfında, “tanımla” isimli fonksiyon ile tasarımda kullanılan nesneler tanımlandı. Bu sınıf içerisinde “AdapterGames” ve “GetGames” sınıflarına baėlantı saėlandı. “getHtftMatches” isimli fonksiyon kullanılarak baėlantı saėlanan diėer sınıflardan çekilen ma verileri ile tasarımda oluşturulan RecyclerView içerisinde tahmin edilen malar, maların sonuçları, başarı durum bilgileri listelendi. Sayfa yenilemesi için “onRefresh” isimli fonksiyon oluşturuldu. Bu fonksiyon sayfayı mevcut tarihe gre gncellemektedir ve srekli aktiftir.

17) Vip Kategorisi yelik Ekranı Oluřturma Sreci

Vip kategorisine erişmek isteyen kullanıcıların yelik almaları için řekil 30’da grlen ye olma ekranı oluşturuldu. Bu sayfanın tasarımı için “frag_vip_subs.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde yelik trlerine gre butonlardan oluşan bir tasarım oluşturuldu. frag_vip_subs.xml isimli tasarım sayfasının “Vip_Subscription” isimli sınıfa baėlantısı saėlandı. Vip_Subscription sınıfında, tasarım ekranında oluşturulan yelik trlerine gre oluşturulan butonlar “tanımla” isimli fonksiyon içerisinde tanımlandı. Butonlar, trlerine gre “billingInterface” ile yelik tipine gre ynlendirildi. yelik adımları tamamlandıktan sonra mevcut kategorinin tahmin sayfasına ynlendirme gerekleřtirildi.



Şekil 30 vip kategorisi üyelik ekranı

18) Silver Kategorisi Üyelik Ekranı Oluşturma Süreci

Silver kategorisine erişmek isteyen kullanıcıların üyelik almaları için üye olma ekranı oluşturuldu. Bu sayfanın tasarımı için “frag_silver_subs.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde üyelik türlerine göre butonlardan oluşan bir tasarım oluşturuldu. frag_silver_subs.xml isimli tasarım sayfasının “Silver_Subscription” isimli sınıfa bağlantısı sağlandı. Silver_Subscription sınıfında, tasarım ekranında oluşturulan üyelik türlerine göre oluşturulan butonlar “tanımla” isimli fonksiyon içerisinde tanımlandı. Butonlar, türlerine göre “billingInterface” ile üyelik tipine göre yönlendirildi. Üyelik adımları tamamlandıktan sonra mevcut kategorinin tahmin sayfasına yönlendirme gerçekleştirildi.

19) Gold Kategorisi Üyelik Ekranı Oluşturma Süreci

Gold kategorisine erişmek isteyen kullanıcıların üyelik almaları için üye olma ekranı oluşturuldu. Bu sayfanın tasarımı için “frag_gold_subs.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde üyelik türlerine göre butonlardan oluşan bir tasarım oluşturuldu. frag_gold_subs.xml isimli tasarım sayfasının “Gold_Subscription” isimli sınıfa bağlantısı sağlandı. Gold_Subscription sınıfında, tasarım ekranında oluşturulan üyelik türlerine göre oluşturulan butonlar “tanımla” isimli fonksiyon içerisinde tanımlandı. Butonlar, türlerine göre “billingInterface” ile üyelik tipine göre yönlendirildi. Üyelik adımları tamamlandıktan sonra mevcut kategorinin tahmin sayfasına yönlendirme gerçekleştirildi.

20) Draw Kategorisi Üyelik Ekranı Oluşturma Süreci

Draw kategorisine erişmek isteyen kullanıcıların üyelik almaları için üye olma ekranı oluşturuldu. Bu sayfanın tasarımı için “frag_draw_subs.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde üyelik türlerine göre butonlardan oluşan bir tasarım oluşturuldu. frag_draw_subs.xml isimli tasarım sayfasının “Draw_Subscription” isimli sınıfa bağlantısı sağlandı. Draw_Subscription sınıfında, tasarım ekranında oluşturulan üyelik türlerine göre oluşturulan butonlar “tanımla” isimli fonksiyon içerisinde tanımlandı. Butonlar, türlerine göre “billingInterface” ile üyelik tipine göre yönlendirildi. Üyelik adımları tamamlandıktan sonra mevcut kategorinin tahmin sayfasına yönlendirme gerçekleştirildi.

21) Htft Kategorisi Üyelik Ekranı Oluşturma Süreci

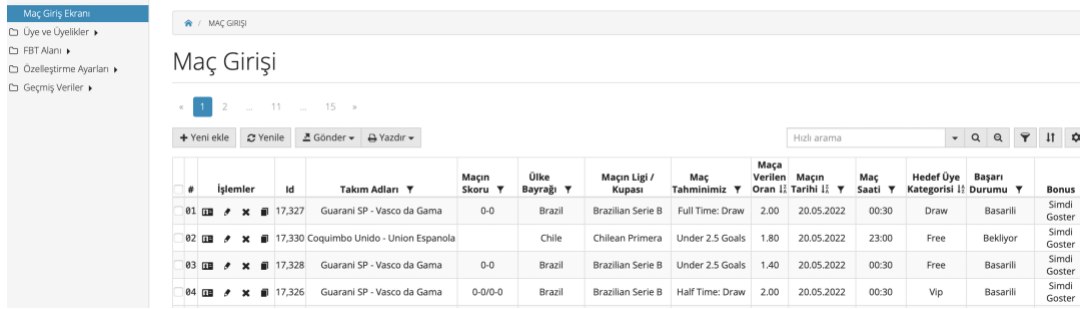
Htft kategorisine erişmek isteyen kullanıcıların üyelik almaları için üye olma ekranı oluşturuldu. Bu sayfanın tasarımı için “frag_htft_subs.xml” isimli dosya oluşturuldu. LinearLayout yapısı içerisinde üyelik türlerine göre butonlardan oluşan bir tasarım oluşturuldu. frag_htft_subs.xml isimli tasarım sayfasının “Htft_Subscription” isimli sınıfa bağlantısı sağlandı. Htft_Subscription sınıfında, tasarım ekranında oluşturulan üyelik türlerine göre oluşturulan butonlar “tanımla” isimli fonksiyon içerisinde tanımlandı. Butonlar, türlerine göre “billingInterface” ile üyelik tipine göre yönlendirildi. Üyelik adımları tamamlandıktan sonra mevcut kategorinin tahmin sayfasına yönlendirme gerçekleştirildi.

22) Veri Tabanı Oluşturma Süreci













Maç verilerinin saklanması için “Güzel Hosting” isimli sistem vasıtasıyla bir veri tabanı oluşturuldu. Güzel Hosting; domain, web, SQL vb. hosting hizmetleri veren bir platformdur. Oluşturulan bu veri tabanı içerisinde kullanıcıya sunulacak olan maç tahmin verileri saklanmaktadır. Oluşturulan bir uzantı sayesinde uygulamaya bağlantısı sağlanmıştır.

23) Yönetici Paneli Oluşturma Süreci

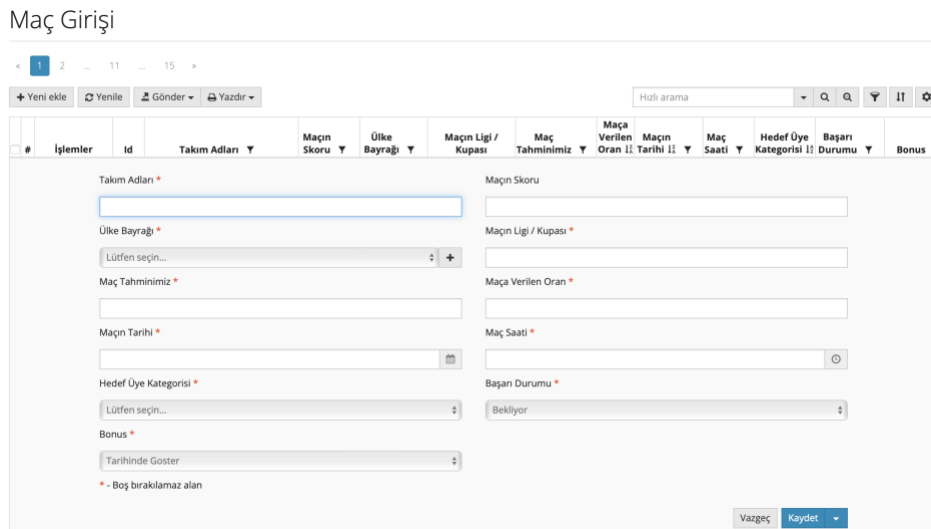
Güzel Hosting ile oluşturulmuş olan veri tabanı yönetici paneline bağlandı. Yönetici panelini oluşturulmak için “PHP Generator” isimli bir yardımcı araç kullanıldı. Bu araç, kullanıcıdan aldığı veri tabanına göre hazır bir şablon oluşturmaktadır. Oluşturulan bu şablon, yönetici panelinde bulunan özellikleri içerisinde hazır olarak bulundurmaktadır. Ekle, sil, güncelle ve listeleme gibi özellikleri SQL komutlarından bağımsız olarak kullanabilmeyi sağlamaktadır. Veri tabanı bağlantısı ardından panel tasarımı ve kullanılacak özellikler yapılandırılır. Bu sayede panel, kullanıma hazır hale gelmiştir.



The screenshot shows the 'Maç Girişi' (Match Entry) screen. On the left is a sidebar with navigation links: 'Maç Girişi Ekranı', 'Üye ve Üyelikler', 'FBI Alanı', 'Özellikler Ayarları', and 'Geçmiş Veriler'. The main area has a breadcrumb 'MAÇ GİRİŞİ' and a title 'Maç Girişi'. Below the title is a pagination bar showing '1' of 11 pages. There are buttons for '+ Yeni ekle', 'Yenile', 'Gönder', and 'Yazdır'. A search bar labeled 'Hızlı arama' is on the right. The table below has columns: #, İşlemler, Id, Takım Adları, Maçın Skoru, Ülke Bayrağı, Maçın Ligi / Kupası, Maç Tahminimiz, Maç Verilen Oran, Maçın Tarihi, Maç Saati, Hedef Üye Kategorisi, Başarı Durumu, and Bonus. The table contains 4 rows of match data.

#	İşlemler	Id	Takım Adları	Maçın Skoru	Ülke Bayrağı	Maçın Ligi / Kupası	Maç Tahminimiz	Maç Verilen Oran	Maçın Tarihi	Maç Saati	Hedef Üye Kategorisi	Başarı Durumu	Bonus
01	  	17,327	Guarani SP - Vasco da Gama	0-0	Brazil	Brazilian Serie B	Full Time: Draw	2.00	20.05.2022	00:30	Draw	Basarili	Simdi Goster
02	  	17,330	Coquimbo Unido - Union Espanola		Chile	Chilean Primera	Under 2.5 Goals	1.80	20.05.2022	23:00	Free	Bekliyor	Simdi Goster
03	  	17,328	Guarani SP - Vasco da Gama	0-0	Brazil	Brazilian Serie B	Under 2.5 Goals	1.40	20.05.2022	00:30	Free	Basarili	Simdi Goster
04	  	17,326	Guarani SP - Vasco da Gama	0-0/0-0	Brazil	Brazilian Serie B	Half Time: Draw	2.00	20.05.2022	00:30	Vip	Basarili	Simdi Goster

Şekil 31 Maç girişi ekranı



The screenshot shows the 'Maç Girişi' (Match Entry) screen with the form for adding a new match. The form has fields for: 'Takım Adları' (Team Name), 'Ülke Bayrağı' (Country Flag), 'Maçın Skoru' (Match Score), 'Maçın Ligi / Kupası' (Match League / Cup), 'Maç Tahminimiz' (Our Match Prediction), 'Maç Verilen Oran' (Match Odds), 'Maçın Tarihi' (Match Date), 'Maç Saati' (Match Time), 'Hedef Üye Kategorisi' (Target User Category), 'Başarı Durumu' (Success Status), and 'Bonus'. There are also buttons for '+ Yeni ekle', 'Yenile', 'Gönder', and 'Yazdır'. A search bar labeled 'Hızlı arama' is on the right. The form is titled 'Maç Girişi' and has a pagination bar showing '1' of 11 pages.

Şekil 32 Maç ekleme ekranı

Üye soruları

12...11»

+ Yeni ekle

Yenile

Gönder

Yazdır

Hızlı arama

Q

Q

#	İşlemler	Id	E-posta Gönderen	Üye Adı ve Soyadı	Soru başlığı	Soru metni	Durum
01	   	557	goksellyilmaz1@gmail.com	Göksel Yılmaz	uygulama hk.	Uygulama nasıl kullanılıyor?	1
02	   	556	goksellyilmaz1@gmail.com	Göksel Yılmaz	gold kategorisi hk.	gold kategorisine nasıl erişirim?	1
03	   	555	goksellyilmaz1@gmail.com	Göksel Yılmaz	vip üyelik hk.	vip kategorisine nasıl erişirim?	1

Şekil 33 Üye soruları ekranı

Üye Yönetimi

«12...11...51...101...644»

+ Yeni ekle

Yenile

Gönder ▾

Yazdır ▾

Hızlı arama

▾

<input type="checkbox"/>	#	İşlemler	Id	E-posta Adresi	Üye Adı ve Soyadı	Parola Doğrulama Kodu	Aktivasyon Durumu	Üyelik Tipi	Üye Fotoğraf	Lokasyonu
<input type="checkbox"/>	01	<div><div><div></div><div></div><div></div><div></div></div></div>	32,521	goksellyilmaz1@gmail.com	Göksel Yılmaz	28114734	Aktif	google		

Şekil 34 Üye yönetim ekranı

Üyelik Yönetimi

12»

+ Yeni ekle

Yenile

Gönder ▾

Yazdır ▾

Hızlı arama

Q

Q

Y

I

S

#	İşlemler	Id	E-posta Adresi	Adı ve Soyadı	Üyelik Başlangıç Tarihi ▾	Üyelik Bitiş Tarihi ▾	Oluşturma Tarihi 11 ▾	Üyelik Paketi Adı	Sipariş ID'si	Token Id
01	<div><div></div><div></div><div></div><div></div></div>	4,635	goksellyilmaz1@gmail.com	göksel yilmaz	20-05-2022	30-06-2024	15.05.2022 00:00:00	bwg_vip_subs_monthly	External	External

Şekil 35 Kategori üyelikleri yönetim ekranı

Kayan Yazı

1

+ Yeni ekle

Yenile

Gönder

Yazdır




















Hızlı arama

#	İşlemler	Id	Kayan yazı metni	Android sayfası	Metin renkleri	Yazı boyutu	Aktivasyon durumu
01	<div><div></div><div></div><div></div><div></div></div>	1	SOME EUROPE LEAGUES ARE NOW OPENED. DO NOT MISS TO BET WEEKEND GAMES. WE ... daha	Free	#ca7c68	16	Aktif
02	<div><div></div><div></div><div></div><div></div></div>	2	BE MEMBER OF GOLD TO WIN. HIGH ACCURACY RATES. DAILY PUBLISHED 3- 7 MATCHES... daha	Gold	#ffae5e	16	Aktif
03	<div><div></div><div></div><div></div><div></div></div>	3	SILVER PROVIDES YOU MID LEVEL ODDS. DAILY PUBLISHED BETWEEN 3-7 GAMES AND ... daha	Silver	#ffae5e	16	Aktif

Şekil 36 Kayan yazı düzenleme ekranı

Ülke Bayrağı Yönetimi



#	İşlemler	Id	Ülke Adı	Ülke Kısa Kodu
01	   	3	Albania	AL
02	   	4	Algeria	DZ
03	   	5	American Samoa	AS
04	   	6	Andorra	AD
05	   	7	Angola	AO
06	   	8	Anguilla	AI

Şekil 37 Ülke bayrakları düzenleme ekranı

24) Makine Öğrenmesi Tahmin Sistemlerinden Verilerin Alınma Süreci

Tahmin için gerekli olan maç verileri “rapidAPI” isimli sistemde bulunan “API-Football” isimli paketten elde edilmektedir. Güncel olarak tüm maç verilerini bulunduran bu paketten alınan maçlar, “Python” isimli programlama dili üzerinde yapılan yapay zeka analizleri sonucu bir tahmin çıktısı olarak elde edilmektedir. Yapay zeka ile oluşturulan maç tahminleri bir liste olarak alınır. Alınan bu tahminler uygulamanın veri tabanına bağlı bulunan bir yönetim paneli vasıtasıyla kullanıcıya servis edilir.

SONUÇ

Maç analizi ve tahminleri yapan bu uygulamanın geliştirilmesi için gerekli altyapı araştırılmaları ve literatür taraması yapıldı. Elde edilen sonuçlar karşılaştırıldı ve gerekli çıktılar dokümanite edildi. Uygulama kullanıcı ara yüzleri tasarlandı. Uygulama içerisinde çeşitli kategori ekranları tasarlandı. Tüm ara yüzlerin arka plan kodlarının çalışması için gerekli sınıf yapıları oluşturuldu. Uygulama giriş işlemleri, çıkış işlemleri, menü işlemleri ve ara yüzler arası geçiş işlemleri tamamlandı. Sınıflar arası bağlantılar oluşturuldu. Maç tahminlerini saklayacak veri tabanı oluşturuldu. Oluşturulan veri tabanı baz alınarak yönetici paneli tasarlandı ve arka plan işlemleri gerçekleştirildi. Veri tabanının uygulama ile bağlantısı oluşturuldu, bu sayede kategori ekranlarında girilen maçların listelenmesi sağlandı.

KAYNAKÇA

- [1] 2021. *Context*. 12 Ağustos. Erişildi: Kasım 16, 2021.
<https://developer.android.com/reference/android/content/Context>.
- [2] 2021. *Bundle*. 17 Mart. Erişildi: Kasım 16, 2021.
<https://developer.android.com/reference/android/os/Bundle>.
- [3] Kırca, Taha. 2016. *Android SpannableString ve ClickableSpan Kullanımı*. 10 Nisan. Erişildi: Kasım 17, 2021. <https://www.mobilhanem.com/android-spannablestring-clickablespace-kullanimi/>.
- [4] 2021. *TextPaint*. 24 Şubat. Erişildi: Kasım 17, 2021.
<https://developer.android.com/reference/android/text/TextPaint>.
- [5] 2021. *TextUtils*. 24 Şubat. Erişildi: Kasım 18, 2021.
<https://developer.android.com/reference/android/text/TextUtils>.
- [6] 2021. *LinkMovementMethod*. 24 Şubat. Erişildi: Kasım 19, 2021.
<https://developer.android.com/reference/android/text/method/LinkMovementMethod>.
- [7] 2021. *View*. 12 Ağustos. Erişildi: Kasım 19, 2021.
<https://developer.android.com/reference/android/view/View>.
- [8] 2021. *Button*. 12 Ağustos. Erişildi: Kasım 20, 2021.
<https://developer.android.com/reference/android/widget/Button>.
- [9] 2021. *EditText*. Aralık. Erişildi: Kasım 20, 2021.
<https://developer.android.com/reference/android/widget/EditText>.
- [10] 2021. *TextView*. Erişildi: Kasım 21, 2021.
<https://developer.android.com/reference/android/widget/TextView>
- [11] Beyler, Alper. 2019. *Android Toast Mesajı Kullanımı*. 30 Nisan. Erişildi: Kasım 22, 2021. <https://www.mobilhanem.com/android-toast-mesaji-kullanimi/>.
- [12] Beyler, Alper. 2018. *Android AlertDialog Kullanımı*. 3 Eylül. Erişildi: Kasım 22, 2021.
<https://www.mobilhanem.com/android-alertdialog-kullanimi/>.
- [13] 2021. *AppCompatActivity*. 9 Ocak. Erişildi: Kasım 23, 2021.
<https://developer.android.com/reference/kotlin/androidx/appcompat/app/AppCompatActivity>.
- [14] 2021. *Toolbar*. 12 Ağustos. Erişildi: Kasım 23, 2021.
<https://developer.android.com/reference/android/widget/Toolbar>.

- [15] 2018. *Android ile Google Kullanıcı Girişi Örneği*. 28 Ocak. Erişildi: Kasım 25, 2021. <https://medium.com/etiya/android-ile-google-kullanici-girisi-orneği-1ee302aa56fa>
- [16] 2021. *GoogleSignInAccount*. Erişildi: Kasım 25, 2021. <https://developers.google.com/android/reference/com/google/android/gms/auth/api/signin/GoogleSignInAccount>.
- [17] 2021. *WebView*. 12 Ağustos. Erişildi: Kasım 29, 2021. <https://developer.android.com/reference/android/webkit/WebView>.
- [18] Beyler, Alper. 2018. *Android ProgressDialog ve Progressbar Kullanımı*. 9 Kasım. Erişildi: Kasım 29, 2021. <https://www.mobilhanem.com/android-progressdialog-ve-progressbar-kullanimi/>.
- [19] 2021. *DialogInterface*. 24 Şubat. Erişildi: Kasım 30, 2021. <https://developer.android.com/reference/android/content/DialogInterface>.
- [20] 2021. *ColorDrawable*. 14 Temmuz. Erişildi: Kasım 30, 2021. <https://developer.android.com/reference/android/graphics/drawable/ColorDrawable>.
- [21] 2014. *URI Nedir* . 16 Nisan. Erişildi: Aralık 1, 2021. <https://www.technopat.net/2014/04/16/android-programlama-ders-34-uri-nedir-arastiriyoruz/>.
- [22] 2021. *Log*. 24 Şubat. Erişildi: Aralık 1, 2021. <https://developer.android.com/reference/android/util/Log>.
- [23] 2021. *WindowManager*. 14 Temmuz. Erişildi: Aralık 2, 2021. <https://developer.android.com/reference/android/view/WindowManager>.
- [24] 2021. *GoogleSignInClient*. 21 Temmuz. Erişildi: Aralık 2, 2021. <https://developers.google.com/android/reference/com/google/android/gms/auth/api/signin/GoogleSignInClient>.
- [25] 2021. *GoogleSignInOptions*. Erişildi: Aralık 3, 2021. <https://developers.google.com/android/reference/com/google/android/gms/auth/api/signin/GoogleSignInOptions>.
- [26] 2021. *ApiException*. Erişildi: Aralık 3, 2021. <https://developers.google.com/android/reference/com/google/android/gms/common/api/ApiException>.
- [27] 2015. 6 Şubat. Erişildi: Aralık 4, 2021. <https://github.com/skyfishjy/android-ripple-background>.
- [28] 2021. *Nullable and NotNull*. 17 Kasım. Erişildi: Aralık 5, 2021. <https://www.jetbrains.com/help/idea/nullable-and-notnull-annotations.html>.

- [29] 2020. *Linear Layout*. 18 Kasım. Erişildi: Aralık 6, 2021.
<https://developer.android.com/guide/topics/ui/layout/linear>.
- [30] 2021. *SuppressWarnings*. 21 Nisan. Erişildi: Aralık 6, 2021.
<https://developer.android.com/reference/android/annotation/SuppressWarnings>.
- [31] 2021. *ActionBarDrawerToggle*. 24 Şubat. Erişildi: Aralık 7, 2021.
<https://developer.android.com/reference/androidx/appcompat/app/ActionBarDrawerToggle>.
- [32] 2021. *GravityCompat*. 24 Şubat. Erişildi: Aralık 7, 2021.
<https://developer.android.com/reference/androidx/core/view/GravityCompat>.
- [33] 2021. *Fragment manager*. 27 Ekim. Erişildi: Aralık 8, 2021.
<https://developer.android.com/guide/fragments/fragmentmanager>.
- [34] 2021. *FragmentStatePagerAdapter*. 27 Ekim. Erişildi: Aralık 8, 2021.
<https://developer.android.com/reference/androidx/fragment/app/FragmentStatePagerAdapter>.
- [35] 2021. *Default transformers*. 3 Mart. Erişildi: Aralık 8, 2021.
<https://docs.informatica.com/data-integration/b2b-data-transformation/10-4-0/user-guide/transformers/defining-transformers/default-transformers.html>.
- [36] 2021. *NavigationView*. 7 Temmuz. Erişildi: Aralık 9, 2021.
<https://developer.android.com/reference/com/google/android/material/navigation/NavigationView>.
- [37] 2021. *Gson*. 29 Ekim. Erişildi: Aralık 10, 2021.
<https://github.com/google/gson>.
- [38] 2021. *HostnameVerifier*. Erişildi: Aralık 10, 2021.
<https://developer.android.com/reference/javax/net/ssl/HostnameVerifier>.
- [39] K., Gürkan. 2021. 9 Kasım. Erişildi: Aralık 11, 2021.
<https://www.hostinger.web.tr/rehberler/https-tls-ve-ssl-nedir-ne-ise-yararlar>.
- [40] 2019. *TrustManager*. 27 Aralık. Erişildi: Aralık 11, 2021.
<https://developer.android.com/reference/javax/net/ssl/TrustManager>.
- [41] 2021. *ContextThemeWrapper*. 14 Temmuz. Erişildi: Aralık 12, 2021.
<https://developer.android.com/reference/android/view/ContextThemeWrapper>.
- [42] 2021. *Firebase Remote Config*. Erişildi: Aralık 13, 2021.
<https://firebase.google.com/docs/remote-config/get-started?platform=ios>.
- [43] *Makine öğrenmesi nedir*. Erişildi: Ocak 20, 2022.
<https://azure.microsoft.com/tr-tr/overview/what-is-machine-learning-platform/>

- [44] 2018. *Makine Öğrenmesi Tarihçesi*. 7 Mayıs. Erişildi: Ocak 20, 2022.
<http://bbyogrencisi.blogspot.com/2018/05/makine-ogrenmesinin-ksa-tarihcesi.html>.
- [45] Wakefield, Katrina. *Predictive Modeling Analytics and Machine Learning*. Erişildi: Ocak 20, 2022. https://www.sas.com/en_gb/insights/articles/analytics/a-guide-to-predictive-analytics-and-machine-learning.html
- [46] tarih yok. *AI Yapay Zeka*. Erişildi: Ocak 20, 2022.
<https://www.oracle.com/tr/artificial-intelligence/what-is-ai/#ai-help-organization>.
- [47] 2022. *Meet Android Studio*. 10 Ocak. Erişildi: Ocak 22, 2022.
<https://developer.android.com/studio/intro>.
- [48] KARADAĞ, Caner. 2020. *C++ Tarihçesi*. 18 OCAK. Erişildi: Ocak 23, 2022.
<http://canerkaradag.com/cpp/basics/cpp-history/>.
- [49] Cansızoğlu, Taha. 2020. *Android NDK Nedir*. 11 Temmuz. Erişildi: Ocak 23, 2022.
<https://medium.com/hardwareandro/android-ndk-nedir-f98004b3689c>.
- [50] 2021. *Kütüphane (bilgisayar bilimi)*. 23 Şubat. Erişildi: Ocak 24, 2022.
[https://tr.wikipedia.org/wiki/K%C3%BCt%C3%BCphane_\(bilgisayar_bilimi\)](https://tr.wikipedia.org/wiki/K%C3%BCt%C3%BCphane_(bilgisayar_bilimi)).
- [51] 2021. *Sınıf (programlama)*. 18 Mart. Erişildi: Ocak 24, 2022.
[https://tr.wikipedia.org/wiki/S%C4%B1n%C4%B1f_\(programlama\)](https://tr.wikipedia.org/wiki/S%C4%B1n%C4%B1f_(programlama)).
- [52] Sarlis, Vangelis, ve Christos Tjortjis. 2020. «Sports analytics — Evaluation of basketball players and team performance.» *Information Systems* 1-19.
- [53] Loreti, Daniela , Marco Lippi, ve Paolo Torroni. 2019. «Parallelizing Machine Learning as a service for the end-user.» *Future Generation Computer Systems* 275-286.
- [54] Rabhi, Fethi A. , Madhushi Bandara, Kun Lu, ve Saif Dewan. 2020. «Design of an innovative IT platform for analytics knowledge management.» *Future Generation Computer Systems* 209-219.
- [55] Leung, Carson K., ve Kyle W. Joseph. 2014. «Sports data mining: predicting results for the college football games.» *Procedia Computer Science* içinde, yazar Carlos Eduardo Ferreira, Orlando Lee ve Flavio Keidi Miyazawa, 710-719. Winnipeg: ELSEVIER.
- [56] Fialho, Gabriel, Aline Manhães, ve João Paulo Teixeira. 2019. «Predicting Sports Results with Artificial Intelligence – A Proposal Framework for Soccer Games Framework for Soccer Games.» *Procedia Computer Science* içinde, yazar Carlos

Eduardo Ferreira, Orlando Lee ve Flavio Keidi Miyazawa, 131-136. Bragança: ELSEVIER.

- [57] Qureshi, Kashif Naseer, Sadia Din, Gwanggil Jeon, ve Francesco Piccialli. 2020. «An accurate and dynamic predictive model for a smart M-Health system using machine learning.» *Information Sciences* 486-502.
- [58] Gossett, Eric, Cormac Toher, Corey Oses, Olexandr Isayev, Fleur Legrain, Frisco Rose, Eva Zurek, ve diğerleri. 2018. «AFLOW-ML: A RESTful API for machine-learning predictions of materials T properties.» *Computational Materials Science* 134-145.
- [59] Bunker, Rory P., ve Fadi Thabtah. 2017. «A machine learning framework for sport result prediction.» *Applied Computing and Informatics* 27-33.