

# Mobil Oyun Programlama

Meryem PAŞAOĞLU  
Bilgisayar Mühendisliği  
220201123@kocaeli.edu.tr

**Abstract—Sunucu istemci mantığıyla çalışan iki oyunculu android mobil kelime tahmin oyunu tasarlanmıştır.**

## I. GİRİŞ

Bu proje, Kotlin ve Firebase kullanılarak Android Studioda geliştirilmiştir.

Oyun, iki kişinin karşılıklı olarak mobil uygulama üzerinden oynayabileceği bir yapıya sahiptir. Sunucu-istemci mantığıyla çalışan bu oyun, genel olarak sunucu üzerinde ilerler ve oyuncular, önceden oluşturulmuş oyun odalarına giriş yaparak rakipleriyle eşleşebilirler. Bu odalar oyuncuların oyuna girdiğinde hazır halde bulunmalıdır, aksi takdirde oyuncular yalnızca uygulamaya giriş yapabilirler. Oyuncuların oyun türünü seçmelerine ve kelime sayısını belirlemelerine olanak tanınır. Oyun sırasında her iki oyuncuya da tahmin edecekleri alanlar sağlanır ve oyuncular kelime tahminlerini yaparlar.

### A. isterler

Projede istenilen isterler şu şekildedir:

- Oyuniki kişinin karşılıklı oynayabileceği şekilde tasarlanacaktır.
- Sunucu-istemci mantığı kullanılacak ve oyun genel olarak sunucu üzerinde ilerleyecektir. Oyun içerisinde daha sonra bahsedilecek oyun odaları oluşturulacaktır. Oyuncular bu odalara giriş yaparak rakipleri ile eşleşebilecektir. Bahsedilen odalar oyuncular oyuna girdiğinde hazır halde bulunmalıdır. Aksi takdirde oyuncular yalnızca uygulamaya giriş yapabilecektir.
- İki tür oyun seçeneği sunulacak; birinde rastgele harf sabiti üretilerek, diğerinde ise harf sabiti kısıtlaması olmadan oyuncudan direkt kelime istenecektir.
- iki kanal olacaktır 5 ve 6 harfli.
- Kanallar tasarlanırken öncelikle oyun türüne göre, daha sonra kelime sayısına göre bölünecektir.
- Oyuncu mobil uygulamaya giriş yaptığında, üye değilse üye olma seçeneğine tıklayarak üyelik işlemlerini yapmalıdır. Üyeliği varsa kullanıcı adı ve şifresi ile giriş yapacaktır.
- Oyuncu oyun türünü seçecek, daha sonra ise kelime sayısını seçerek ilgili kanala giriş yapacaktır. Aynı kanalda bulunan kullanıcılar birbirlerini görebilecekler.
- kullanıcı yalnızca birisine istekte bulunabilecektir. Diğer oyuncu kabul ederse oyun başlayacaktır. Diğer oyuncu 10sn içerisinde cevap vermezse veya isteği reddederse oyuncuya red mesajı gelecektir. Oyuncu red mesajı aldığıda aynı kullanıcıya veya farklı bir kullanıcıya

tekrardan oyun isteğinde bulunabilecektir. Aynı anda birden fazla oyuncuya istek atmayacaktır.

- İsteği kabul etmesi durumunda oyuncular kanal özelliğine göre (örneğin Harf sabiti olmadan 5 harfli oda olsun) kelime giriş ekranı açılacaktır.
- Açılan ekranda kullanıcılar rakiplerine soracağı kelimeleri girecektir ve daha sonrasında her iki oyuncu da birbirlerine sordukları kelimeyi tahmin etmeye çalışacaktır
- Oyuncuların kelime girmesi için 1 dakika süreleri bulunmaktadır. Bu süre sağ üst köşede geri sayım olarak gösterilmelidir. Oyunculardan birisi rakibinin tahmin edeceği kelimeyi 1 dakika içinde girip onaylarsa, diğeri ise girip onay butonuna basamaz ise oyunu kaybedecektir. Her ikiside 1 dakika içinde rakibinin tahmin etmesi gereken kelimeyi onaylamaz ise kelime yazma ekranı tekrar açılacak kelime girmesi için tekrardan 1 dakika süre verilecektir
- Oyun sırasında her iki oyuncuya da harf sayısı kadar tahmin edecekleri alanlar oluşturulacaktır. Örneğin, 5 harfli bir oyunda ise 5 kez tahmin hakkı olacak şekilde alanlar oluşturulacaktır.
- Oyunda bulunan satır sayısı kadar (aynı zamanda kelime sayısı) deneme şansı bulunan oyuncular, her bir satırda tahmin ettiği kelimeyi oluşturduktan sonra enter tuşuna basacaktır.
- Girilen kelimenin geçerli bir kelime olması gereklidir. Geçerli bir kelime değil ise oyuncuya geçerli bir kelime olmadığı bildirilecek ve geçerli bir kelime girmesi istenecektir.
- Tahmin edilen kelimedeki harfler rakibin sorduğu kelime içinde varsa; doğru yerde bulunan harfler, yeşil yanlış yerde bulunan harfler ise sarı renkte gösterilecektir. Eğer harf yok ise gri renkte gösterilecektir.
- kelime listesi içerisinde bulunan kelimelerden rastgele bir kelime seçilip onaylanacaktır. Bu rastgele girilen kelimedeki benzer şekilde kontrol edilip harf durumlarına göre renklendirilecektir. Eğer iki oyuncuda doğru tahmin edemedi ise en son satırda yeşil olan harfler 10 ile çarpılacak ve toplanacaktır. Benzer şekilde sarı olan harfler ise 5 ile çarpılarak toplanacaktır
- Oyun bitiminde her iki oyuncunun sonuç ekranları aynı ekran üzerinde oyunculara gösterilecek ve galibiyet durumları ekranda verilecektir. Bu ekranda oyuncuların tahmin ekranları, tahmin etme süreleri ve puanları (her bir puan kalem de belirtilerek) gösterilecektir.
- Oyun sonucunda açılan ekranda sağ üst köşede düello

butonu bulunacaktır. Oyuncuların birisinin düello butonuna basması durumunda karşı oyuncuya düello isteği düşecektir. Eğer onaylaması durumunda oyun tekrardan açılacak ve yeni bir oyun başlatılacaktır. Oyuna düello ile devam edilmesi durumunda, oyun sonunda önceki oyun skorları da dahil edilerek sonuçlar gösterilecektir (Puan kalemine önceki oyun sonucu da eklenecektir.). Düello isteğinde yapılacak süre ve işlemler, oyuna başlama isteğindeki süre ve işlemler ile aynı olacaktır.

- Oyuncular kanalda rakip listesine geri dönecek ve yeni oyun için rakip oyunculara istekte bulunabilecektir. Bu sırada farklı kanallar arasında geçiş yaparak diğer kanallardaki oyuncuları görebilir ve istekte bulunarak oyun oynayabilecektir.

## II. YÖNTEM

Projede bulunan değişkenler ve metotlar şu şekilde özetlenebilir: MainActivity: TasarimmEkranii() fonksiyonu: Bu fonksiyon, giriş ekranının tasarımını ve içeriğini oluşturur. emailKullaniciAdi ve siiffree adında iki değişken, kullanıcının e-posta ve şifresini tutar. Bir Column bileşeni kullanarak arayüzü dikey bir düzende oluşturur. Text bileşeni ile bir başlık görüntüler. OutlinedTextField bileşenleri, kullanıcıdan e-posta ve şifre bilgilerini girmesini sağlar. Button bileşeni, kullanıcının kaydolma işlemini başlatır. GirissButonuu(emailKullaniciAdi: String, siiffree: String) fonksiyonu: Bu fonksiyon, "Giriş Yap" butonunu oluşturur ve kullanıcının giriş yapabilmesi için gereken kontrolleri sağlar. emailKullaniciAdi ve siiffree parametreleri, kullanıcının girdiği e-posta ve şifre bilgilerini alır. Eğer giriş bilgileri boş değilse, giriş işlemini gerçekleştirir. Eğer boş ise, kullanıcıya uygun bir uyarı gösterir. kaayiittOll(emailKullaniciAdi: String, siiffree: String) fonksiyonu: Bu fonksiyon, kullanıcının kaydolma işlemini gerçekleştirir. emailKullaniciAdi ve siiffree parametreleri, kullanıcının girdiği e-posta ve şifre bilgilerini alır. Eğer giriş bilgileri boş değilse, Firebase Authentication kullanarak yeni bir kullanıcı oluşturur. Oluşturulan kullanıcının bilgilerini Firebase Realtime Database'e kaydeder. İşlem başarılıysa, kullanıcıya başarılı bir kayıt mesajı gösterir. İşlem başarısızsa, hata mesajı gösterir. giiriissYapp(emailKullaniciAdi: String, siiffree: String) fonksiyonu: Bu fonksiyon, kullanıcının giriş yapma işlemini gerçekleştirir. emailKullaniciAdi ve siiffree parametreleri, kullanıcının girdiği e-posta ve şifre bilgilerini alır. Firebase Authentication kullanarak kullanıcı girişi yapar. Giriş işlemi başarılıysa, kullanıcıya başarılı bir giriş mesajı gösterir ve yeni bir aktiviteye yönlendirir. Giriş işlemi başarısızsa, hata mesajı gösterir.

OdaActivity: OdaActivity sınıfı, Android uygulamasında odaları listelemek için kullanılan bir aktivitedir. Bu sınıf, odaları listelemek için bir arayüz sağlar ve kullanıcının belirli odalara yönlendirilmesini sağlar.

onCreate() fonksiyonu, aktivite oluşturulduğunda çağrılır ve içeriği belirler. setContent fonksiyonu, Compose kullanarak arayüzü oluşturur. ButtonTasaraimii() fonksiyonu, kullanıcı arayüzünü oluşturur. Bir Column bileşeni kullanarak düzen sağlanır ve butonlar yerleştirilir. İlk iki buton, kullanıcının

belirli bir harf sayısı ile kelime oluşturmaya sağlar. onClick olayları, belirli bir sayıda harf içeren kelimeleri oluşturmak için yazılacak işlemleri belirtir. Son iki buton, belirli bir harf sayısı ile kelime oluşturmaya yanı sıra, belirli bir kelime uzunluğunda kelime oluşturmaya sağlar. onClick olayları, belirli bir harf sayısı ve kelime uzunluğunda kelimeleri oluşturmak için yönlendirme işlemlerini gerçekleştirir. yonlendir() ve yonlendir() fonksiyonları, belirli butonlara tıklama durumunda kullanıcının ilgili aktivitelere yönlendirilmesini sağlar.

SunucuIstemciActivity:

SunucuIstemciActivity, bir oyun sunucusu ve istemcisi olarak hareket eden bir Android aktivitesidir. Bu aktivite, kullanıcıların oyun isteklerini göndermelerini, kabul etmelerini veya reddetmelerini sağlar. Ayrıca, kullanıcıların diğer oyuncularla iletişim kurmalarına olanak tanır. onCreate() Bu fonksiyon, aktivite oluşturulduğunda çağrılır ve içeriği belirler. Firebase Authentication ile mevcut kullanıcının bilgisini alır. Compose kullanarak arayüzü oluşturur ve içeriği belirler. kullaniciMailleriniSiralaa() Bu fonksiyon, Firebase Realtime Database'deki "kullanici" düğümündeki kullanıcıları listeler. ValueEventListener kullanarak veritabanı işlemlerini dinler. Her kullanıcının e-posta adresini kullanıcı listesine ekler. kullaniciIstekleriniSiralaa(girisYapanKulEmail: String) Bu fonksiyon, kullanıcının alıcı olduğu oyun isteklerini listeler. Firebase Realtime Database'deki "OyunIstekleri" düğümündeki istekleri dinler. Kullanıcının alıcı olduğu istekleri algılar ve ilgili verileri işler. istekGonder(oyunIsteği: String, gondereEmail: String, aliciEmail: String) Bu fonksiyon, kullanıcının belirli bir kullanıcıya oyun isteği göndermesini sağlar. İstek gönderildiğinde Firebase veritabanına isteği ekler. İsteğin gönderildiği kullanıcıya bildirim gönderir. kabulEtButonuu(iisteekk: OyunIsteği) Bu fonksiyon, kullanıcının aldığı bir oyun isteğini kabul etmesini sağlar. İstek kabul edildiğinde Firebase veritabanından isteği kaldırır. Kabul edilen isteğin gönderenine bir bildirim gönderir ve oyunu başlatır. reddetButonuu(iisteekk: OyunIsteği) Bu fonksiyon, kullanıcının aldığı bir oyun isteğini reddetmesini sağlar. İstek reddedildiğinde Firebase veritabanından isteği kaldırır. Reddedilen isteğin gönderenine bir bildirim gönderir. mesajGonder(receiverEmail: String, message: String) Bu fonksiyon, kullanıcının diğer oyunculara mesaj göndermesini sağlar. Henüz tamamlanmamış bir özelliktir ve mesaj gönderme işlemlerini içermez. zamanlayiciyiBaslatt() Bu fonksiyon, belirli bir süre içinde isteğin kabul edilmemesi durumunda otomatik olarak yeniden gönderilmesini sağlayan bir zamanlayıcı başlatır. Zamanlayıcı, CountdownTimer sınıfını kullanarak gerçekleştirilir. yeniIsteekk() Bu fonksiyon, yeni bir oyun isteği göndermek veya başka bir kullanıcıya istek göndermek için gerekli durumları işler. İsteğin gönderileceği kullanıcıyı sıfırlar veya kullanıcı listesinden başka bir kullanıcı seçer. onDestroy() Bu fonksiyon, aktivite yok edildiğinde zamanlayıcıyı iptal eder. Zamanlayıcı, aktivite yok edilmeden önce durdurulur ve kaynakların serbest bırakılmasını sağlar.

OyunActivity: Değişkenler ve Sabitler: kelimeListesi: Oyun

için kullanılacak olan kelimelerin bir listesi. gecerliKelimeIndexi: O anki oyun için seçilen kelimenin indeksi. kalanDenemeHakki: Kullanıcının oyun sırasında yapabileceği kalan deneme hakkı. skorPuanı: Kullanıcının kazandığı puan. gecerliKelime: O anki oyun için seçilen kelime. kelimeninHarfleri: Seçilen kelimenin harflerinin bir listesi. kutularinDuzeni: Harf kutularını içeren GridLayout. skorTexti: Kullanıcının puanını gösteren TextView. kelimeGirisi: Kullanıcının kelime tahminlerini gireceği EditText. onCreate() Fonksiyonu: Aktivite oluşturulduğunda çağrılır. Arayüz öğelerini oluşturur, rastgele bir kelime seçer ve harf kutularını oluşturur. Tahmin yapma, yeni oyun başlatma gibi işlemleri gerçekleştirmek için butonlar ekler. rastgeleKelimeSec() Fonksiyonu: Kelime listesinden rastgele bir kelime seçer ve bu kelimeyi ve harflerini değişkenlere atar. kutulariOlustur() Fonksiyonu: GridLayout içinde harf kutularını oluşturur ve düzene ekler. buttonEkle() Fonksiyonu: Verilen metin ve tıklama işlevi ile bir buton oluşturur ve düzene ekler. tahminiYerlestir() Fonksiyonu: Kullanıcının girdiği kelimeyi kontrol eder ve doğru harf ve pozisyonları işaretler. Doğru tahminler için kutular yeşil, yanlış tahminler için sarı renkte gösterilir. Kullanıcının kalan deneme hakkı kontrol edilir ve oyunun sonucuna göre mesajlar gösterilir. oyunuSifirla() Fonksiyonu: Oyunu sıfırlar: yeni bir kelime seçer, harf kutularını yeniden oluşturur ve kullanıcının deneme hakkını sıfırlar. skoruGuncelle() Fonksiyonu: Kullanıcının puanını günceller ve skoru ekranda gösterir. toastMesaji() Fonksiyonu: Kullanıcıya kısa mesajlar göstermek için bir Toast mesajı görüntüler.

kanalSabitsiz6: Bu bir oyun sunucusu ve istemcisi olarak hareket eden bir Android aktivitesidir. Bu aktivite, kullanıcıların oyun isteklerini göndermelerini, kabul etmelerini veya reddetmelerini sağlar. Ayrıca, kullanıcıların diğer oyuncularla iletişim kurmalarına olanak tanır.

onCreate: Bu fonksiyon, aktivite oluşturulduğunda çağrılır. Bu kod bloğunda, kullanıcı girişi işlemleri gerçekleştirilir ve arayüz belirlenir. setContent fonksiyonu aracılığıyla ButtonnTasaraimii adlı kompozisyon belirlenir ve bu kompozisyon ekranda görüntülenir. Ayrıca, kullanıcıların listesi ve isteklerin listesi sıralanır. ButtonnTasaraimii: Bu fonksiyon, arayüzdeki butonları ve diğer bileşenleri tasarlar. İki ana bölüme sahiptir: biri gelen istekleri gösterir, diğeri ise istek gönderme işlemlerini yapar. Ayrıca, zamanlayıcı başlatma butonu ve oyunu başlatma butonu da bulunmaktadır. kulaniiMailleriniSiralaa: Bu fonksiyon, Firebase Realtime Database'den kullanıcı e-postalarını alır ve bir liste olarak saklar. kulaniiIstekleriniSiralaa: Bu fonksiyon, Firebase Realtime Database'den kullanıcıya gelen istekleri kontrol eder. istekGonder: Bu fonksiyon, bir kullanıcıya oyun isteği gönderir. İsteğin gönderildiğini işaretler ve Firebase veritabanına isteği ekler. kabulEtButtonuu: Bu fonksiyon, kullanıcıya gelen isteği kabul etme işlemini gerçekleştirir. İsteği kabul ettiğinde, ilgili kullanıcıya mesaj gönderir, isteği veritabanından siler ve oyunu başlatır. reddetButtonuu: Bu fonksiyon, kullanıcıya gelen isteği reddetme işlemini gerçekleştirir. İsteği reddettiğinde, ilgili kullanıcıya mesaj gönderir ve isteği veritabanından siler. mesajGonderr: Bu fonksiyon, bir kullanıcıya mesaj

göndermek için kullanılır. Ancak, şu anda bu fonksiyonun içi boş. zamanlayiciyiBaslatt: Bu fonksiyon, bir zamanlayıcı başlatır. Örneğin, bir oyun isteğinin belirli bir süre içinde kabul edilmesi veya reddedilmesi için kullanılabilir. yeniIsteekk: Bu fonksiyon, bir oyun isteğinin tekrar gönderilmesi veya başka bir kullanıcıya gönderilmesi gereken durumları ele alır. İsteği gönderilecek kullanıcıyı sıfırlar ve zamanlayıcıyı iptal eder. onDestroy: Bu fonksiyon, aktivite yok edildiğinde çağrılır. Bu noktada, zamanlayıcıyı iptal eder.

harfSabitsiz6: onCreate(savedInstanceState: Bundle?): Bu fonksiyon, aktivite başlatıldığında çağrılır. Aktivite oluşturulurken yapılacak işlemler burada yapılır. Bu fonksiyon içinde aşağıdaki işlemler gerçekleştirilir: Layout ve bileşenler oluşturulur. Rastgele bir kelime seçilir. Harf kutuları oluşturulur. Skor göstermek için TextView oluşturulur. Onayla ve Yeni Oyun butonları eklenir. rastgeleKelimeSec(): Bu fonksiyon, kelime listesinden rastgele bir kelime seçer ve seçilen kelimenin harflerini bir listeye dönüştürür. kutulariOlustur(layoutDuzeni: LinearLayout): Bu fonksiyon, GridLayout kullanarak harf kutularını oluşturur. Her bir kutu, bir TextView bileşeni olarak oluşturulur ve GridLayout'a eklenir. buttonEkle(layoutDuzeni: LinearLayout, text: String, parametreler: LinearLayout.LayoutParams, onClick: () -> Unit): Bu fonksiyon, bir düzene bir buton eklemek için kullanılır. Butonun metni, parametreler ve tıklama işlevi belirtilir. tahminiYerlestir(girilennKelime: String, layoutDuzeni: LinearLayout): Bu fonksiyon, kullanıcının girdiği kelimeyi kontrol eder ve doğru/hatalı tahminleri işaretler. Her doğru tahminde doğru pozisyon sayacı artar. Eğer tüm harfler doğru tahmin edilirse oyunu kazanır ve puanını günceller. Eğer yanlış tahmin ederse deneme hakkını azaltır ve eğer deneme hakkı kalmazsa oyunu kaybeder. oyunuSifirla(layoutDuzeni: LinearLayout): Bu fonksiyon, oyunu sıfırlar. Yeni bir kelime seçer, harf kutularını yeniden oluşturur, deneme hakkını sıfırlar ve tahmin yapmayı etkinleştirir. skoruGuncelle(): Bu fonksiyon, skoru günceller ve skor TextView'ında gösterir. toastMesaji(message: String): Bu fonksiyon, kullanıcıya kısa bir mesaj göstermek için Toast kullanır.

### III. SONUÇLAR

Oyun, iki kişinin karşılıklı olarak mobil uygulama üzerinden oynayabileceği bir yapıya sahiptir. Sunucu-istemci mantığıyla çalışan bu oyun, genel olarak sunucu üzerinde ilerler ve oyuncular, önceden oluşturulmuş oyun odalarına giriş yaparak rakipleriyle eşleşebilirler. Bu odalar oyuncuların oyuna girdiğinde hazır halde bulunmalıdır, aksi takdirde oyuncular yalnızca uygulamaya giriş yapabilirler.

Oyuncuların oyun türünü seçmelerine ve kelime sayısını belirlemelerine olanak tanınır. Ardından ilgili kanala giriş yapılır ve aynı kanalda bulunan diğer oyuncular görülebilir. Oyuncular, birbirlerine istekte bulunarak oyun başlatma talepleri gönderebilirler. İstek kabul edilirse, oyun ilgili kanal özelliklerine göre başlatılır.

Oyun sırasında her iki oyuncuya da tahmin edecekleri alanlar sağlanır ve oyuncular kelime tahminlerini yaparlar. Tahmin edilen kelimedeki harfler, rakibin sorduğu kelime içinde varsa doğru yerde bulunan harfler yeşil, yanlış yerde bulunanlar sarı renkte gösterilir. Eğer harf yoksa gri renkte gösterilir. Belirlenen süre içinde tahmin yapılmazsa sistem tarafından rastgele bir kelime seçilir ve aynı işlem uygulanır.

Eğer her iki oyuncu da doğru tahmin edemezse, yeşil ve sarı renkteki harflere göre puanlama yapılır. Oyuncular daha sonra rakip listesine geri dönebilir, yeni oyunlar için istekte bulunabilir ve farklı kanallar arasında geçiş yaparak diğer oyuncularla iletişim kurabilirler. Bu yapı, oyunculara dinamik bir oyun deneyimi sunar ve rekabetçi bir ortam oluşturur.

## REFERENCES

- [1] <https://www.youtube.com/watch?v=rWamixHIKmQ>
- [2] <https://firebase.google.com/docs/auth/android/start?hl=tr>
- [3] <https://www.youtube.com/watch?v=jbHfJpoOzkI>
- [4] <https://www.youtube.com/watch?v=wm626abfMM8>
- [5] <https://www.youtube.com/watch?v=qkQGActKadQ>
- [6] <https://www.youtube.com/watch?v=7Vxrl-d1Oo>

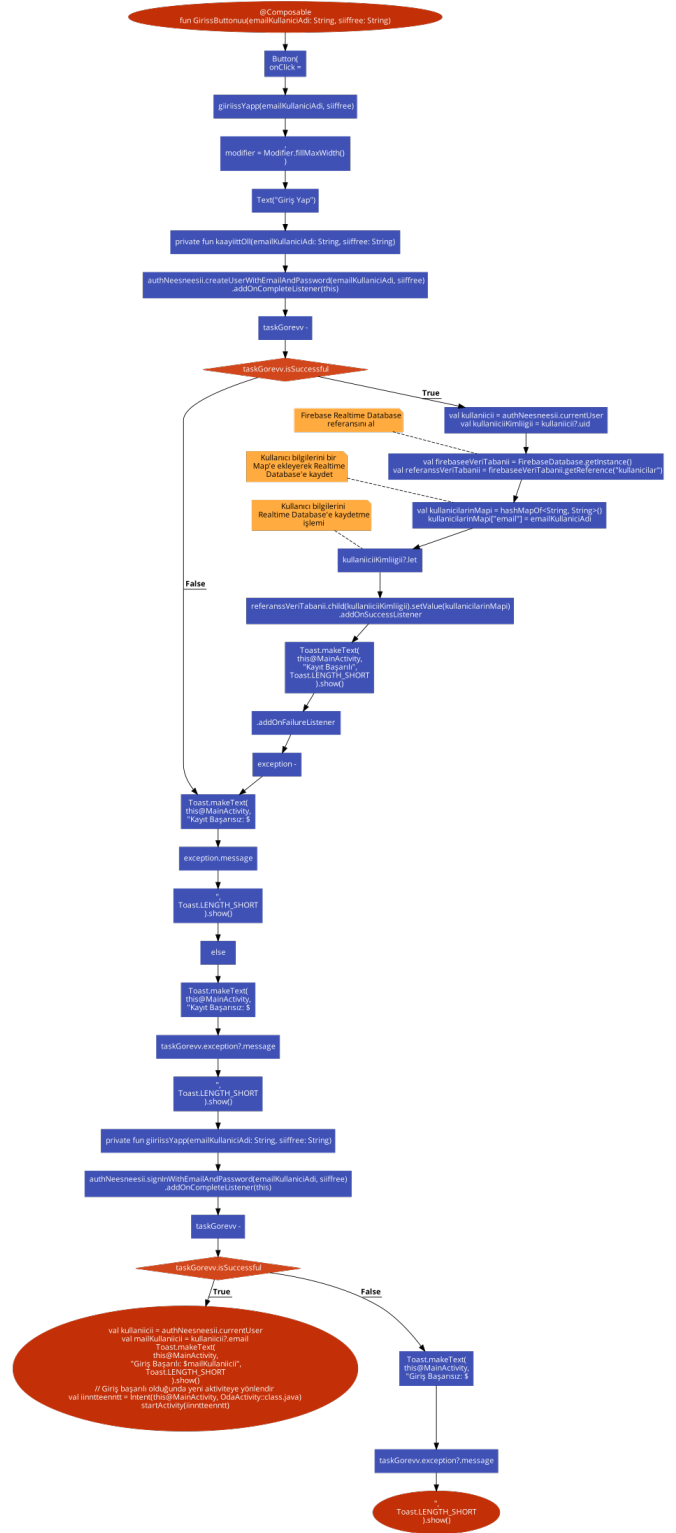


Fig. 1: MainActivity

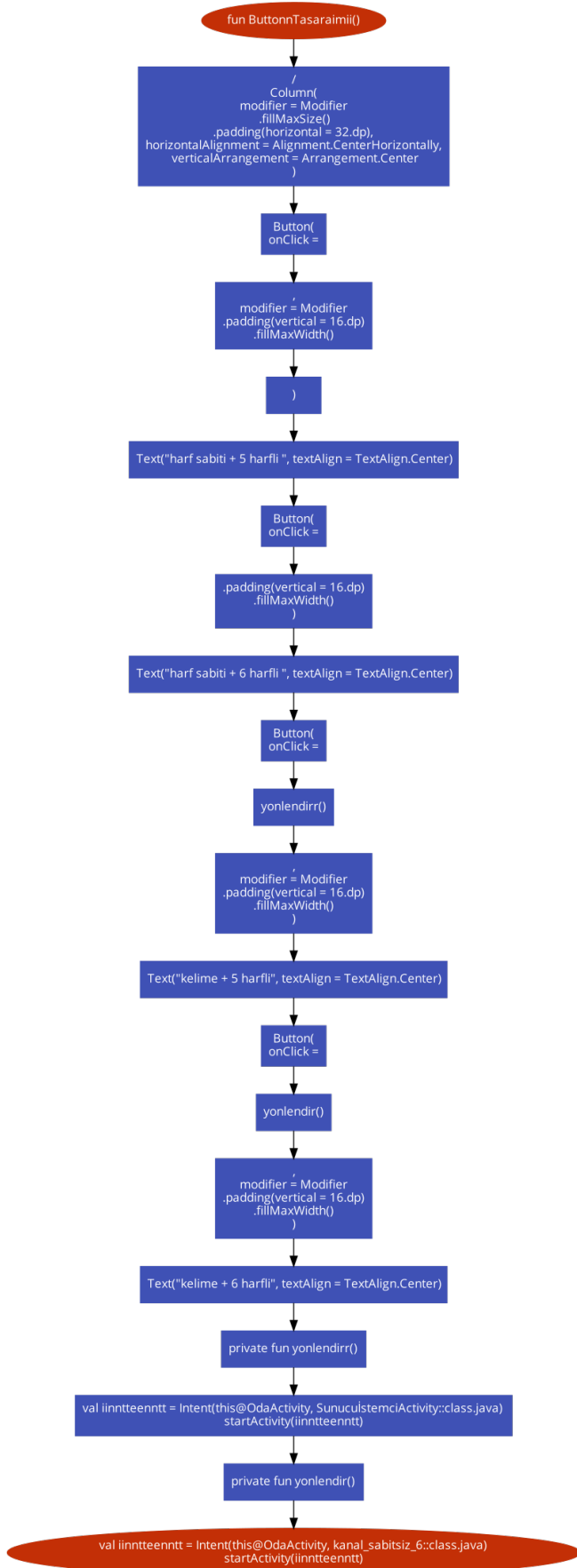


Fig. 2: OdaActivity

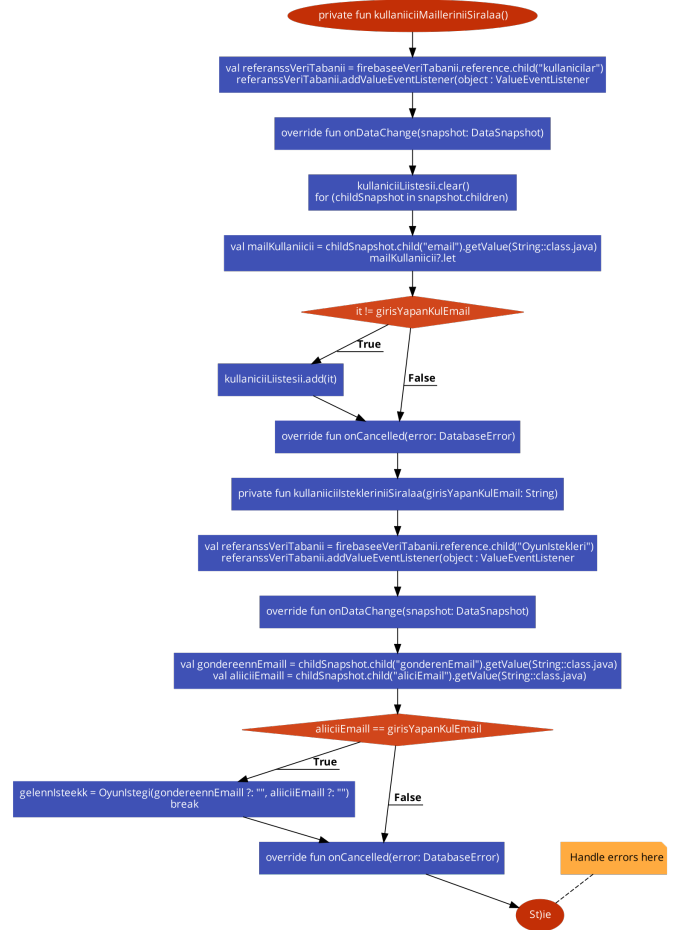


Fig. 3: SunucuIstemciActivity-1

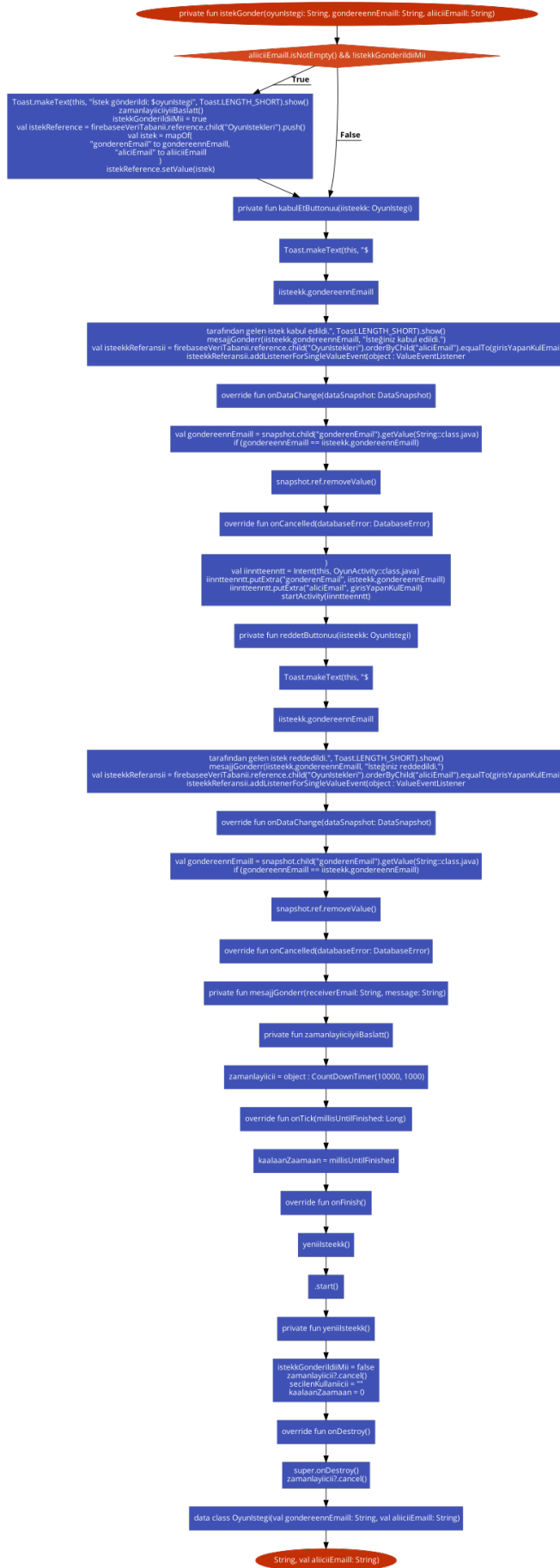


Fig. 4: SunucuIstemciActivity-2

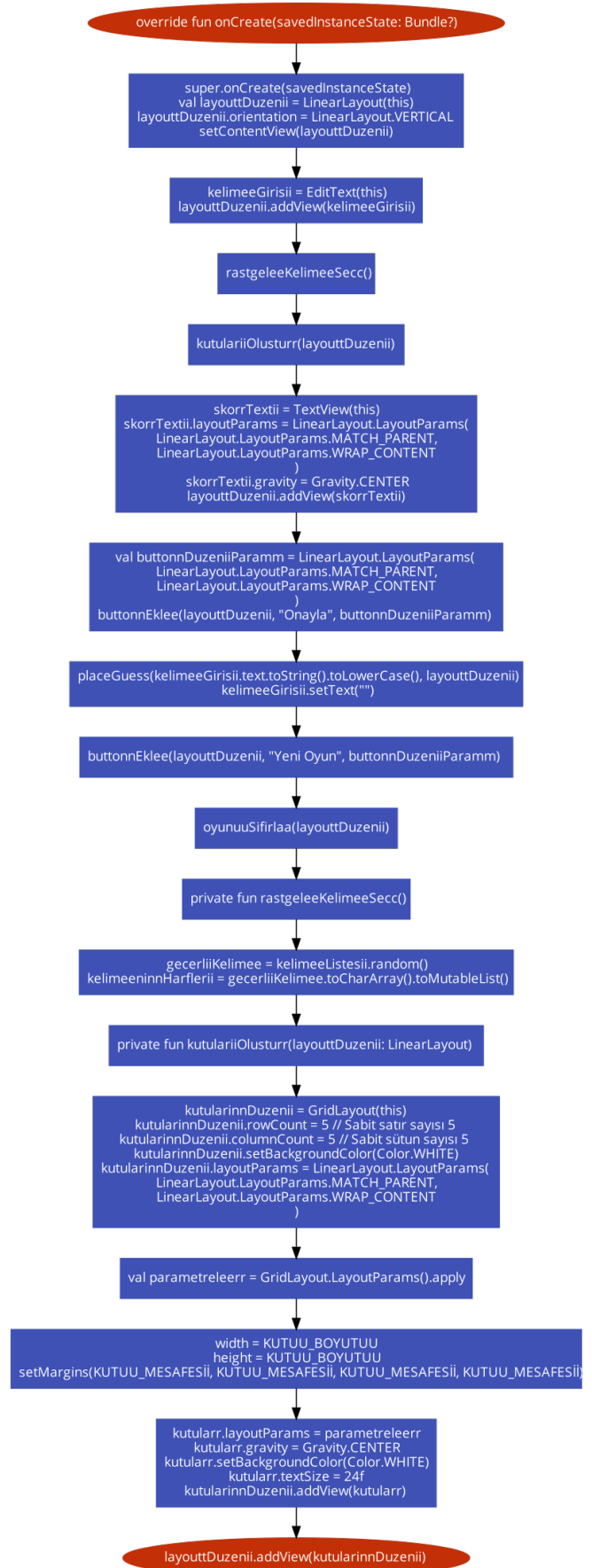


Fig. 5: OyunActivity-1

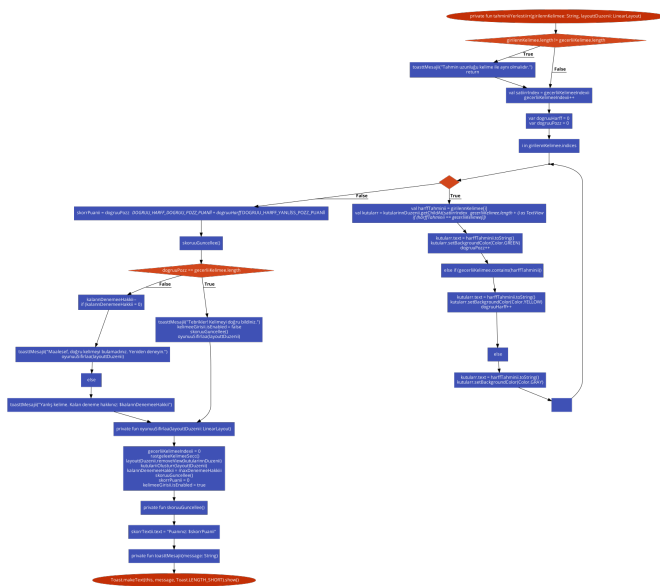


Fig. 6: OyunActivity-2