

# Mobil Oyun Programlama Wordle

Eren GÜNEŞ

200202026

Kocaeli Üniversitesi Bilgisayar Mühendisliği (İÖ)  
gunes18\_eren@hotmail.com

Yusuf GÖKSU

210202027

Kocaeli Üniversitesi Bilgisayar Mühendisliği (İÖ)  
goxuyusuf@gmail.com

## I. ÖZET

Rapor Yazılım Laboratuvarı II dersinin 2.Projesi'nin nasıl yapıldığını anlatmak ve içeriğin açıklanması için yazılmıştır. Bu rapor içerisinde Giriş başlığı altında Uygulama İsterleri adında 1 adet alt başlık bulunmaktadır. Bu başlık ile proje kısaca anlatılmaya çalışılmıştır.

Daha sonrasında Yöntem başlığı altında projenin kodunun nasıl yazıldığından bahsedilip Deneysel Sonuç başlığında ise yapılan program programı çalıştıran kişinin gözünden nasıl gözükeceği anlatılmıştır.Kazanımlar bölümünde proje yapılırken projenin bize katkılarından bahsedilmiştir. Sonuç kısmında ise proje kapsamında yapılan adımların sonuçları anlatılmıştır. Raporun sonundaysa proje yapılırken kullanılan kaynaklara, kaynakçalara yer verilmiştir.

## II. GİRİŞ

Bu rapor belgesi 2023-2024 Bahar dönemi Yazılım Laboratuvarı II dersinin 2. projesini açıklamaya yönelik bilgiler içermektedir. Projede programlama dili olarak Java, Kotlin, Flutter, React Native, Swift dillerinden bir tanesinin kullanılması istenmiştir. Veritabanı ve sunucu tarafında ise serbest bırakılmıştır.

Bu proje kapsamında bizden bir mobil oyun yapılması istenmiştir. Android ya da IOS ile geliştirilmelidir. Kullanıcıların oyuna girip karşılıklı olarak kelime oyunu oynayabilecekleri bir proje yapılması istenmiştir. Geliştirilecek uygulama kelime tabanlı bir oyundur. Bundan dolayı oyunda kelime listesinin kullanılması gereklidir. Oyunda kullanıcıların giriş yapabilmesi için üyelik sisteminde olmalıdır.

### A. Uygulama İsterleri

- 1) Uygulama sunucu-istemci mantığı kullanılarak yapılmalıdır. Oyun genel olarak sunucu üzerinden ilerleyecektir. Kullanıcıların oyun oynayabilecekleri odalar sunucu ile oluşturulmalıdır. Odalardaki diğer kullanıcılar ile oyunu karşılıklı olarak oynayabilecekleri şekilde tasarlanmalıdır
- 2) Oyun karşılıklı olarak iki kişi ile oynanabilir olmalıdır.
- 3) İki tür oyun modu olmalıdır. Oyunculardan kelimeler istenirken birinci oyun modu rastgele harf sabiti üretecektir. İkinci oyun modu ise harf sabiti

üretmeyecektir. Kullanıcılar kelimelerini buna göre gireceklerdir.

- 4) Kelime girdileri minimum 4 harfli maksimum 7 harfli olacaktır. Odalar da kelimelerin uzunluklarına göre açılmalıdır.
- 5) Odalar oluşturulurken oyun türüne ve kelime uzunluğuna göre bölünecektir.
- 6) Kullanıcılar oyun türü seçimi yapabilmelidir. Oyun türü seçtikten sonra kelime uzunluğu seçmelidir. Oyuncu seçtiği kanaldaki diğer kullanıcıları görebilmelidir.
- 7) Odalardaki oyuncular birbirlerinin aktiflik (aktif,oyunda) durumlarını görebilmelidir.
- 8) Kullanıcı oyundaki oyunculara istek yollayamaz ama aktif oyunculara istek yollayabilmelidir. Bir oyuncuya istek yollandığında 10 saniye içerisinde cevap gelmezse ya da istek reddedilirse başka bir kullanıcıya istek atılabilir. İstek kabul edilir ise de oyun başlatılır.
- 9) Oyuncular isteği kabul ettiklerinde oyun ekranında girdikleri kanala uygun şekilde kelime girişi yapabilecekleri bir ekran açılmalıdır.
- 10) Açılan oyun ekranında kullanıcılar rakiplerine soracağı kelimeyi girmelidir. Oyuncuların kelimeleri girmesi için 1 dakika süre verilmelidir. 1 dakika içerisinde kelime girilmezse kullanıcı kaybedecektir. Kelimeler girildikten sonra karşılıklı olarak girilen kelimeleri tahmin etmeleri gerekir.
- 11) Kullanıcıların her ikisinde kelime girme ekranında 1 dakika içerisinde bir kelime giremezse ekran tekrar açılacak ve tekrar 1 dakika süre verilecektir.
- 12) Kullanıcılar tahmin edilecek kelimeleri girip onayladığında oyun eş zamanlı başlayarak her iki oyuncuya da girdikleri odanın kelime uzunluğuna göre kelime girişleri yapabilecekleri alanlar oluşturulacaktır. Satır ve sütun sayısı birbirine eşit olmalıdır. Kullanıcılar burada rakibin girdiği kelimeyi tahmin etmeye çalışmalıdır.

- 13) Kullanıcıların girdikleri her kelime geçerli bir kelime olmak zorundadır. Eğer kelime geçerli değilse kullanıcıya kelimenin geçersiz olduğu uyarısı verilmelidir.
- 14) Kullanıcı kelime girdikten sonra eğer kelime doğru ise aranan kelime ile girilen kelime karşılaştırılacaktır. Kelime eğer doğru tahmin edildiyse oyun bitecektir. En önce tahmin eden kullanıcı oyunu kazanır. Eğer 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. Örnek tahmin işlemleri aşağıda gösterilmiştir.

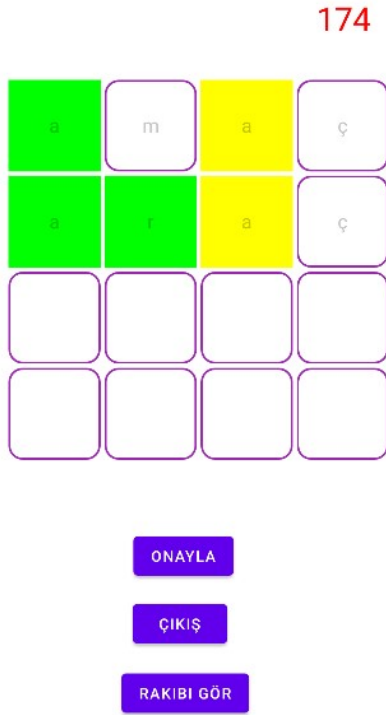


Fig. 1. Oyun Ekranı

- 15) Herhangi bir kullanıcı oyun sırasında oyundan çıkarsa oyunu kaybetmiş sayılacaktır. Kullanıcıya bunun uyarısı yapılmalıdır.
- 16) Eğer kullanıcılardan birinin oyun oynanırken bağlantısı kopar ise 10 saniye geri sayım başlatılacaktır. Sayım bitene kadar kullanıcı dönmezse kaybedecektir.
- 17) Oyun oynanırken rakibi gör butonuna basılarak rakibin ilerlemesi görülebilmelidir.

- 18) Kullanıcılardan bir tanesi bütün tahmin haklarını kullandıysa diğer kullanıcıya her tahmin işlemi için 10 saniye verilecektir. Eğer kullanıcı bu 10 saniye içerisinde bir kelime giremez ise o zaman kelime listesinden rastgele bir kelime girilerek kullanıcı girmiş gibi kabul edilecektir.

- 19) Oyunun sonunda eğer iki kullanıcıda doğru kelimeyi tahmin edemediyse en son satırdaki yeşil harf adeti 10 ile, sarı harf adedi 5 ile çarpılacaktır, ayrıca a oyunun başında rakibe sorulacak kelimenin girildiği ekranda kelimenin girilmesi istenen dakikadan geriye kalan saniye de puanlamaya dahil edilerek toplam puan elde edilecektir. Örnek bir hesaplama aşağıda gösterilmiştir.

Son kelimede 2 tane yeşil olduğunu varsayalım  $10 \times 2 = 20$  puan

3 tane sarı olduğunu varsayalım  $5 \times 3 = 15$  puan

Kelime yazma ekranında kalan saniyenin 25 saniye olduğunu varsayalım bu durumda kullanıcının puanı  $20 + 15 + 25 = 60$  puan olacaktır.

- 20) Oyun bittiğinde kullanıcıların oyun ekranlarında yapmış oldukları işlemler ekranda gösterilecektir. Bunun yanı sıra tahmin etme süreleri ve puanları da gösterilmelidir.
- 21) Oyun bittikten sonra kullanıcılar tekrar oynamak için düello seçeneğini seçebilmelidir. Düello butonuna basıldığında karşıdaki kullanıcıya bir istek gönderilmelidir. İsteğin kabul edilmesi durumunda aynı skor üzerinden oyun tekrar başlamalıdır. Düellonun reddedilmesi durumunda ise oyun sona ererek giriş ekranına yönlendirilecektir. Rakiplerden bir tanesi oyundan dahi çıksa düello isteği gönderilebilir olmalıdır.

### III. YÖNTEM

#### A. Sunucu ve Odalar

Harf sabitli ve sabitsiz olarak odalar oluşturduk. Bu odalara üzerlerinde sabitli ve sabitsiz yazan butonlara basarak giriş yapabiliyoruz. Butona basıldığında bizi GameActivity'ye yönlendiriyor burada odalarımızın bulunduğu bir spinner yani dropdownbox var, spinnerin alt kısmında ise ListViewimiz var, spinnerda 4-5-6-7 kişilik odalara geçiş yapmak için üzerine tıklamamız yeterli. ListView kısmında kendi isimimiz ve bulunduğumuz oyun modundaki bulunduğumuz odadaki aktif kişiler yer almaktadır. Kişilerin isimlerinin üzerine tıkladığımızda o kişiye istek oluşturuluyor ve karşıdaki kişinin verilerine bizim idmiz ekleniyor eğer kişi kabul ederse preGameActivity classımız açılacaktır. Burada sabitli ise kişilere rastgele bir harf verilerek bu harfe uygun kelime girilmesi istenmektedir. Bunu girerse ve karşı rakip girmemişse onu beklemek için AlertDialog kullandık. Rakip de uygun kelime girerse oyun başlatılıyor. Birisi kelime girip biri girmezse kelime giren oyunu kazanmaktadır. ikisi de girmezse iki oyuncuya da 1 dakika ekstra süre verilmektedir. Oyunun içerisinde

karşılıklı birbirlerinin kelimelerini bulmaya çalışmaktadırlar. Birisi kelimeyi bildiğinde EndGameActivity'ye yönlendiriliyor. Orda da tekrardan düello butonu ve ana menüye dön butonları mevcuttur. Düello butonu ile rakip oyuncuya tekrar istek atılır kabul ederse oyun tekrardan eski skorlar üzerine eklenerek tekrardan başlanır. Anamenü butonuyla da anamenüye dönülür.

### B. Oyun

Belirli bir boyutta bir kare ızgara oluşturulur. Her kareye bir EditText (metin giriş kutusu) eklenir. Her EditText, kullanıcıların harf girişi yapabileceği bir alandır. Her bir kutu bir harfi temsil eder. Kullanıcı, ızgaradaki her bir kutuya bir harf girebilir. Kullanıcı bir harf girdiğinde, olay dinleyicileri aracılığıyla kontrol edilir: Eğer kullanıcı bir harf girdiyse, bir sonraki EditText'e geçilir. Eğer bir kullanıcı bir harfi silerse (delete tuşuna basarsa), önceki EditText'e geçilir. Kullanıcı, son EditText'e ulaştığında, klavye gizlenir. Kullanıcı, ızgaradaki herhangi bir kutuya harf girebilir ve silerek, ızgarayı dolaşabilir. Kullanıcının girdiği kelimeyi kontrol eder ve doğruluğunu belirler: Kullanıcı, bir kelimeyi tamamladığında (tüm kutular dolduğunda), girilen kelimeyi kontrol eder. Girilen kelime, bir metin dosyasındaki önceden tanımlanmış kelimelerle karşılaştırılır. Eğer girilen kelime listede varsa, kelime doğru kabul edilir. Doğru bir kelime girildiğinde, oyunun ilerleyişine ve kullanıcı arayüzüne bazı değişiklikler yapılır: Girilen kelime işaretlenir ve aranan kelime ile karşılaştırılır. Eğer tamamen eşleşiyorsa, kullanıcı kazanır ve oyun biter. Eğer tüm harfler doğruysa ama kelime tam olarak eşleşmiyorsa, kullanıcıya bilgi verilir ve oyun devam eder. Eğer doğru bir kelime bulunamazsa, kullanıcıya bilgi verilir ve oyun devam eder. Eğer oyunun sonuna gelinirse, hem kullanıcının hem de rakibin puanları hesaplanır: Kullanıcının puanı, doğru harfler ve renklerin sayısına göre hesaplanır. Rakibin puanı, aynı şekilde hesaplanır. Her iki oyuncunun puanı belirlendikten sonra, oyun bitirilir ve sonuçlar kullanıcıya gösterilir. Bir sonraki adıma geçiş: Kullanıcı, oyunu tamamladığında veya iptal ettiğinde, sonuçları gösteren bir ekran olan "endGameActivity" ekranına yönlendirilir. Bu yönlendirme, bir Intent kullanılarak yapılır. Intent'e veri eklemek: "endGameActivity" ekranına geçişte, çeşitli bilgileri aktarmak için putExtra yöntemi kullanılır. Bu bilgiler arasında kullanıcının e-posta adresi, oyun modu, oda adı, oyuncu kimlikleri vb. bulunur. Zamanlayıcıları iptal etme: Eğer bir zamanlayıcı çalışıyorsa, öncelikle iptal edilir. Bu, oyuncunun oyunu terk ettiğinde veya sonlandığında, geriye dönük sayım veya herhangi bir süreç aktifse bunların durdurulması için gereklidir. Listener'ı kaldırma: Veritabanı dinleyicileri kaldırılır. Bu, oyuncunun oyunu terk ettiğinde, oyunla ilgili veritabanı işlemlerini durdurmak için gereklidir. Aktiviteyi sonlandırma ve yeni aktiviteyi başlatma: Mevcut aktivite sonlandırılır ve sonuçları gösteren aktivite başlatılır.

### C. Sonuç Ekranı

Sonuç ekranına oyun ekranından gelen veriler intent ile çekildi. Sonrasında findViewById ile textViewlar, buttonlar

çekildi. Daha önceden kurulmuş sunucu bağlantısı ile adListenerForSingleValueEvent fonksiyonu kullanılarak kullanıcıların skorları ve girdi olarak girdikleri kelimeler çekildi. Aynı zamanda puan attributesinin kontrolü yapıldı. Kullanıcıların eğer paun attributesi varsa oyunda her iki tarafta doğru kelimeyi bulamamış ve son durumdaki puanlarına bakılarak karar verilecektir demek. Puan attributesi var ise her iki kullanıcının da puanlarını child fonksiyonu ile çekiyoruz. Daha sonra puanlar arasında karşılaştırma yaparak yüksek puana sahip kullanıcının skorunu artırıyoruz. Ekrandaki textView larıda burada düzenliyoruz. Ana menüye dön butonuna basılması durumunda kullanıcıların oyun verilerini silip inten ile ana menü ekranına yönlendiriyoruz. Düello butonuna basılması durumunda da rakio kullanıcının refini kullanarak bir istek oluşturuyoruz. İsteğin kabul edilmesi durumunda tekrardan intent ile sunucu verileri düzenlenerek yönlendiriyoruz. Oyunu gör butonlarında ise oyun içerisinde yaptığımız işlemlerin aynısını yapıyoruz. Kullanıcıların kelimelerini kodun en başında almıştık bu kelimeleri tekrardan döngüye sokup bir dialog üzerinde görselleştirdik. Dialogada kapat tuşu koyup dialogun kapanmasını sağladık.

## IV. DENEYSEL SONUÇLAR

### A. Giriş Ekranı

Oyun ilk açıldığında kullanıcıları giriş yapabilecekleri ve kayıt olabilecekleri bir ekran karşılıyor. Ekranda e-posta ve şifrenin girilebileceği iki adet input ve iki adet buton bulunmaktadır. Butonlardan bir tanesi giriş işlemi için kullanılırken diğeri kayıt etme işlemi için kullanılır. Giriş butonuna basıldığında eğer var olan bir kullanıcı iseniz oyun modu seçebileceğiniz ekrana yönlendirilirsiniz.

Fig. 2. Giriş Ekranı

### B. Oyun Modları Ve Odalar

Giriş yapan kullanıcıları ilk başta oyun modu seçebilecekleri bir ekran karşılar. Ekrandan iki adet buton bulunmaktadır. Butonlardan bir tanesi harf sabitli oyun modunu diğeri ise harf sabitsiz oyun modunu seçmek içindir.

Oyunun modu seçildikten sonra kullanıcı kelime uzunluğu seçeceği ekrana gelir bu ekranda ise 4 tane buton bulunmaktadır. Bu butonlar 4 harfli, 5 harfli, 6 harfli ve 7 harfli yazan butonlardır. Butonlardan bir tanesine basılması durumunda odadaki kullanıcılar sıralı bir şekilde gözüktür. Her kullanıcının müsaitli durumu kullanıcıların üzerinde yazar. Müsait olan kullanıcıların isimlerinin üzerine tıklanması durumunda kullanıcılara istek gitmektedir. Gelen istek ekranda gözüktür kabul etme ve etmeme seçenekleri çıkar. Kullanıcı eğer kabul ederse oyun ekranına yönlendirilir kabul etmez ise de oyun modu ekranında kalmaya devam eder.

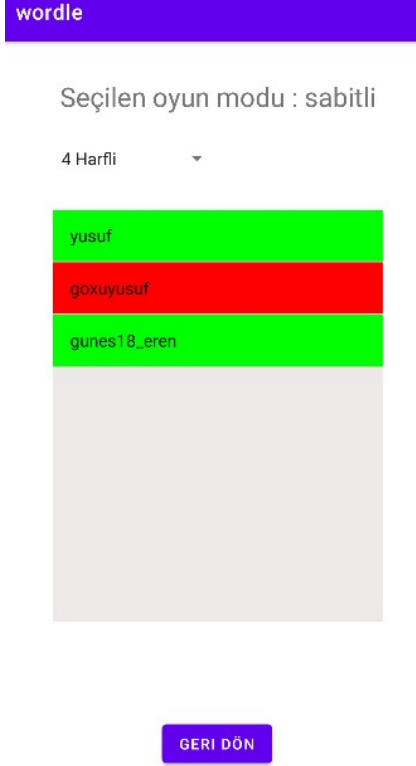


Fig. 3. Odalara Giriş Ekranı

### C. Oyun

Oyun açıldığında kullanıcıyı seçtiği oyun moduna göre bir ekran karşılar. Kullanıcının seçtiği kelime uzunluğuna göre örneğin 4 ise 4x4 lük bloklar gelir. Kullanıcı bloklara tıkladığında kelime girebilir. Blokların altında onayla ve çıkış yap butonları bulunmaktadır. Kullanıcı bir satıra kelime girişi yaptığında satır sonuna geldiğinde klavye kapanmaktadır. Eğer değiştirmek istediği bir harf var ise o harfe tıklayarak değiştirebilmektedir aynı zamanda klavyedeki silme tuşuna basarakta bloklar arası geri gidebilmektedir. Kullanıcı ilgili satıra kelime girişi yaptıktan sonra onayla butonuna basmalıdır. Eğer kullanıcının girdiği kelime geçerli bir kelime ise bir alt satıra geçmektedir. Kullanıcı kelime girdiğinde harflerin arkalarında yeşil, sarı ve gri renklerini görecektir. Yeşil renk harfin arana kelime içerisinde olduğunu ve o harfin doğru

yerde olduğunu gösterir. Sarı renk harfin kelime içerisinde olduğunu ama harfin yanlış yerde olduğunu gösterir. Gri renkteki harfler ise o harfin kelime içerisinde olmadığını gösterir. Kullanıcı eğer kelimeyi doğru tahmin ederse veya rakibi daha önceden kelimeyi doğru tahmin ederse iki kullanıcıda sonuç ekranına yönlendirilir. Kullanıcı çıkış butonuna basması durumunda oyundan çıkıp çıkmak istemediği sorulur. Çıkış yapmaya basar ise tekrardan oda seçim ekranına geri döner ver oyunu kaybetmiş sayılır.

### D. Sonuç Ekranı

Sonuç ekranına gelindiğinde ekranda oyuncuların skorları, isimleri, kendi ve rakiplerinin oyun ekranlarını görebilecekleri butonlar, ana ekrana dönme butonu ve düello butonları bulunmaktadır.

Eğer kullanıcıların ikisinde oyun esnasında doğru kelimeyi bilemediyse hesaplama puan değerleride kullanıcıların isimlerini yanlarında gösterilir. Oyun ekranını gör butonları ise oyun esnasında ortaya çıkan oyun ekranını kullanıcılara gösterir. Bu şekilde kullanıcılar kendi ve rakip oyun ekranlarını görebilirler. Ana menüye dön butonuna basılması durumunda kullanıcı ana ekrana dönerek tekrar oyun modu seçimine yönlendirilir. Düello butonuna basılması durumunda ise rakibe bir düello isteği gönderilir. Eğer rakip gelen isteği kabul ederse oyun kaldığı skordan tekrar başlar.

## V. KATKILAR

Bu proje kapsamında ilk defa bir mobil uygulama geliştirdik Android Stuido ilk defa bu proje ile öğrenilmiştir.

Daha önce sunucu kullanarak bir proje geliştirmemiştir. İlk defa sunucu kullanarak sunucu mantığı ve kullanım şekli anlaşıldı.

Bağlantı kopması, oyundan çıkma durumları ve oyun süresinin kontrol edilmesi gibi işlemler öğrenilmiştir

Sunucu ve database arasındaki fark proje sayesinde anlaşılmıştır.

## VI. SONUÇ

Bizden bu projede veritabanı, sunucu ve mobil programlama yapacağımız bir kelime oyunu yapılması istenmiştir.

Bu projede mobil uygulamalarda işleyen gui ve frontend mantığını anlamış olduk. Bunları bir arada kullanmayı, yönetmeyi ve mobil kavramlarını öğrenmiş olduk. Sunucu işlemlerinin nasıl yapılacağını, nasıl kontrol edileceğini öğrendik. Ayrıca ilk defa kullandığımız android stuidoyu kullanmayı öğrendik.

Bu projedeki amaç bir mobil uygulamanın nasıl yapıldığını ve sunucu mantığı ile nasıl birleştirilebileceğini anlamaktır.

Projenin çalıştırılabilmesi için projenin indirilip run edilmesi yeterlidir.

Kod takımca yazılmıştır.

## VII. KAYNAKÇA

- 1) <https://www.geeksforgeeks.org/how-to-build-a-wordle-game-application-in-android/>
- 2) <https://firebase.google.com/docs/database?hl=tr>
- 3) <https://www.youtube.com/watch?v=9ZntK2uz9i4list=PLhe7JosY15tJlZMSRYM89JrVM3fosY3D>
- 4) <https://yannahmet.medium.com/android-firebase-realtime-database-67162d0ce48f>
- 5) <https://www.youtube.com/watch?v=oKM2nQdQkIU>