

YAZLAB - PROJE 3

Harry Potter: Memory Master

1st Melih Selami URKMEZ
Kocaeli University
Computer Engineering
Kocaeli,Turkey
200202010@kocaeli.edu.tr

2nd Taha PEK
Kocaeli University
Computer Engineering
Kocaeli,Turkey
200202046@kocaeli.edu.tr

Abstract—Bu proje kapsamında Java/Kotlin dilinde yazılan, bağımlılıklarının Bulut Platformu üzerinde bulunacağı bir mobil uygulama geliştirilmesi amaçlanmaktadır. Bu mobil uygulama bir hafıza kart oyunu olacaktır. Tek ve çift olmak üzere hem tek hem de çift kişi ile oynanabilir. Ayrıca kolay,orta,zor olarak 3 seviyesi bulunmaktadır.

I. GİRİŞ

Projede Harry Potter filminden karakterler bize verilmiştir. Verilen karakterlerin bilgilerini (adı, evi, puanı, kartı resmi) ve oyunu oynayacak olan kullanıcıların bilgilerini (kullanıcı adı, şifresi, ID bilgisi, e-posta hesabı) bulut tabanındaki bir veri tabanında tutulması gerekmektedir .

Giriş ekranı olarak oyun ilk açıldığında ekranda açılacak sayfa giriş ekranı olması istenmektedir. Kullanıcı bu ekranda, kullanıcı adı ve şifresi ile giriş yapabilmeli, şifre değiştirebilmeli ve kaydolabilmesi istenmektedir.

Oyun ekranı olarak kullanıcı giriş yaptıktan sonra karşısına gelecek ekran oyun ekranı olması istenmektedir. Burada Tek Oyuncu ve Çoklu Oyuncu Olarak iki farklı seçenek bulunmalıdır. Oyun ekranı ilk açıldığında “BAŞLA” butonu bulunmalıdır. Oyuncu BAŞLA butonuna tıkladığında oyun ve süre başlatılması istenmektedir.

Oyun başlatıldığında kartlar kapalı şekilde dağıtılması istenmektedir. Oyundaki kartların her birinden birer çift bulunmaktadır. Buradaki amaç açılan kartın diğer çiftini bulabilmektir. Oyunda kartlar ilk olarak rastgele dağıtılır.

Oyun zorluk seviyesi için oyunda 2*2, 4*4 ve 6*6 olmak üzere 3 farklı zorluk seviyesi olması istenmektedir.

Oyun sırasında ve belirli olaylarda da belirtilmiş olan müziklerin çalınması beklenmektedir.

II. TERMINOLOJİ

Bulut Platform Nedir?

Bulut Teknolojisi ya da Cloud Computing olarak da bilinen Bulut Tabanlı; akıllı telefon, tablet ya da bilgisayar gibi çeşitli

teknolojik cihazlar ile ulaşılabilen, veri paylaşımı sağlayan, internet tabanlı bir veri depolama sistemidir.

Docker Nedir?

Docker, konteynerleştirme hizmetinin bir ürünüdür. Konteyner dediğimiz kavram ise bir uygulamanın,servisin vb. herhangi bir kodun tüm bağımlılıklarıyla beraber işletim sisteminden izole bir şekilde bulunduğu ortam olarak adlandırılabilir. Biz de projemizde kullandığımız MySQL ve Redis bağımlılıklarını, Web API’leri konteynerleştirerek Kubernetes’te koşturduk.

Kubernetes Nedir?

Kubernetes genel tanımıyla aslında bir konteyner orkestrasyon platformudur. Bu ifadeyi biraz daha açacak olursak, birden fazla konteynerin tek bir yapı tarafından yönetilmesini sağlar. Containerleri sürekli kontrol ederek en az sayıda kesintiye amaçlar.

MySQL Nedir?

MySQL, ilişkisel bir veritabanı olarak 1995 yılında kullanıma sürülmüştür. Web tasarım dünyasında kullanılan en popüler açık kaynaklı ilişkisel veri tabanı yönetim sisteminden birisi olma özelliğine sahiptir. SQL ise, MySQL’in temelini, yani çekirdek yapısını oluşturmaktadır. MySQL ismi ise “SQL (Structured Query Language)” ve Michael Widenius’un kızının adının (My) birleşiminden almaktadır.

Redis Nedir?

Redis, açık kaynaktır ve kaynak kodlarına GitHub üzerinden erişilebilmektedir. C dili ile yazıldığı için yüksek performanslı sonuçlar vermektedir. Linux ve türevi işletim sistemleri tarafından desteklenmekte fakat Windows tarafı için resmi bir destek olmasa da community tarafından desteklenmektedir.

GKE(Google Kubernetes Engine) Nedir?

Google Cloud üzerinde, PAAS olarak sunulan bir hizmettir. Kullanıcı tarafından istenen kaynaklara ve node sayısına göre bir Kubernetes clusteri hizmeti sunar.

DockerFile Nedir?

Kabaca Dockerfile açıklarsak; içerisinde yayınlanacak olan

uygulamanın nasıl bir ortamda çalışacağına dair talimatları barındırmaktadır.

Projede hem Redis hem de MySQL kullanma sebebimiz her zaman database ile işlem yapıp vakit kaybetmeyi engellemektir. Çünkü Redis Mysql'den oldukça hızlı bir NOSQL veritabanıdır. İlgili query eğer Rediste bulunmuyorsa bunu Databasede sorgula ve ilgili cevabı ve query'i Rediste depola. Böylece gelen cevap süresini 2 kat daha hızlandır.

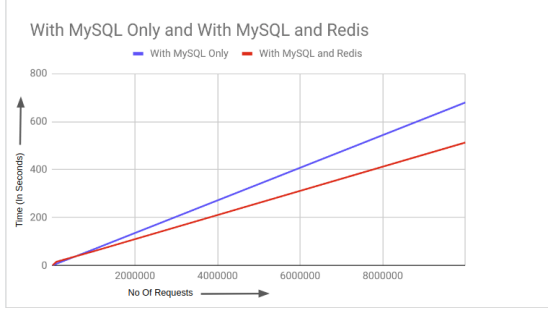


Fig. 1. Redis vs MySQL Hız Grafiği

III. TEMEL İSTERLER

Uygulama hem tek kişi ile hem de çift kişi ile oynanabilir olmalıdır. Bu yüzden hem tek kişilik için hem de çift kişilik için olan gerekli isterler aşağıda listelenmiştir.

A. Tek Oyunculu

- Kartlar oyunun başında rastgele arka yüzleri kapalı olacak şekilde dağıtılır. Oyuncu bir kartın üzerine tıklar ve kart açılır. Daha sonra oyuncu farklı bir karta tıklayarak kartın eşini bulmaya çalışır.
- Oyun skoru: Oyun süresi 45 saniyedir. Oyunda her kartın bir puanı ve ait olduğu bir ev bulunmaktadır. Oyun skoru her hamle sonrasında ekranda anlık olarak gösterilecektir.
 - Örn- Harry Potter (Puan :10 , Ev: Gryffindor)
 - Oyuncu doğru bir eşleştirme yaparsa $[(2 * \text{kartın puanı} * \text{evin katsayısı}) * (\text{kalan süre} / 10)]$ kadar puan kazanır.
 - Yanlış bir eşleştirme durumunda iki kart aynı evden ise $[(\text{kartların toplam puanı} / \text{evin katsayısı}) * (\text{geçen süre} / 10)]$ kadar puan kaybeder.
 - Yanlış bir eşleştirme durumunda iki kart farklı evden ise $[(\text{kartların puan ortalaması} * \text{Ev1katsayı} * \text{Ev2katsayı}) * (\text{geçen süre} / 10)]$ kadar puan kaybeder

B. Çok Oyunculu

- Kartlar oyunun başında rastgele arka yüzleri kapalı olacak şekilde dağıtılır. 1. Oyuncu oyuna başlar ve bir kartı seçer. Daha sonrasında kartın eşini bulmaya çalışır. Eğer kartın eşini bulursa aynı oyuncu oyuna devam eder. Eğer kartın eşini bulamazsa sıra rakip oyuncuya geçer.
- Oyun skoru: Oyun süresi 60 saniyedir. Oyunda her kartın bir puanı ve ait olduğu bir ev bulunmaktadır. Her oyuncu

sırayla seçim yapar. Doğru bir eşleştirme yapan oyuncu tekrar oynama hakkına sahiptir. Oyun skoru her hamle sonrasında ekranda anlık olarak gösterilecektir.

- Örn - Harry Potter (Puan :10 , Ev: Gryffindor)
- Oyuncu doğru bir eşleştirme yaparsa $(2 * \text{kartın puanı} * \text{evin katsayısı})$ kadar puan kazanır.
- Yanlış bir eşleştirme durumunda iki kart aynı evden ise $(\text{kartların toplam puanı} / \text{evin katsayısı})$ kadar puan kaybeder.
- Yanlış bir eşleştirme durumunda iki kart farklı evden ise $(\text{kartların puan ortalaması} * \text{Ev1 katsayı} * \text{Ev2 katsayı})$ kadar puan kaybeder.

NOT: Rastgele dağıtılan kartların bilgisi (ön yüzlerinde hangi karakterin bulunduğu bilgisi) ayrıca bir not defterinde tutulup anlık olarak takip edilebilecektir.

NOT: 4*4 ve 6*6 dağıtılan destelerde her evden eşit sayıda karakter bulunması gerekmektedir.

IV. YÖNTEM

Proje'nin verimli bir şekilde çalıştırabilmesi için Andorid Studio ortamında Kotlin diliyle yazılmıştır. Ayrıca Android Studio'da bulunan emülatör de aktif bir şekilde kullanılmıştır. İlgili kart ve kullanıcı bilgileri bulut platformunda gke üzerinde mysql veritabanında saklanmıştır.

Veritabanı işlemleri ile mobil uygulamayı haberleştirmek için Flask frameworkü sayesinde WEB API kodlanmıştır.

WEB API Nedir ?

Application Programming Interface anlamına gelen dilimize uygulama programlama arayüzü olarak çevirebileceğimiz bir servis teknolojisidir. Platform bağımsızdır ve yazılın bir Web Api servisi birden fazla veri formatına ve birden fazla platforma destek verebilir.

Http protokolü üzerinden haberleşir ve MVC Desing Pattern(tasarım kalıbı) uygulanmıştır. MVC yapısında bulunan Routing, Controlles, Action, Filters, Model Binders yapılarını Web Api teknolojisinde birebir görmemiz mümkündür.

Flask Nedir ?

Flask bir python frameworküdür. Web servislerinde de hızlı sonuç elde etmek için pythonun "Flask" frameworkünden kullanılır. Flask hem çabuk öğrenilebilen hem de benchmarklarına bakıldığında performansı gayet yüksek bir frameworktür.

A. Proje Hiyerarşisi

- DbOperations
DbOperations içerisindeki ImageEncode.py dosyasında verilmiş olan görsellerin base64 formatına çevrilmesi ve çevrilen verilerin veri tabanında kaydedilme işlemleri gerçekleştirilmiştir. WritetoDb.py dosyasında ise kartların bilgilerinin ilişkisel veri tabanına kayıt işlemi gerçekleştirilmiştir.
- Image
Projenin arayüzünde kullanılan arkaplan resminin

bulunduğu klasördür.

- Service
Service klasörünün içerisinde kullanıcı authentication işlemleri için ve kart bilgilerinin oyuna yüklenmesi için WEP API kodları bulunmaktadır. Ayrıca ilgili servislerin Dockerfile'leri de bulunmaktadır.
- MemoryGame
Oyun klasörüdür. İçerisinde android studio için gerekli eklentiler bulunur. .gradle dosyasında ilgili konfigürasyon işlemleri bulunur. Main klasöründe oyunun ilgili kodları bulunur. Layout klasöründe ise oyunun ekranı bulunur.

V. FONKSİYON İŞLEVLERİ

- PlayAudio Fonksiyonu
Uygulama müziğinin çalıştırmasını sağlar.
- StopAudio Fonksiyonu
Uygulamada çalan müziği durdurmayı sağlar
- OyunBittiBasarili Fonksiyonu
Oyun bittiğinde başarılı bir şekilde alert döndürmeye yapar
- OyunBittiBasarisiz
Oyun Bittiğinde başarısız bir şekilde alert döndürmeye yapar.
- CalculateTrueResult
Doğru eşleştirmedeki puanı hesaplamaya yapar.
- CalculateFalseReult
Hatalı eşleştirme puan hesaplamayı yapar.
- UpdateView
Kart görünülerini düzenler.
- UpdateModel
Card class'ı içerisindeki Faceup property'sine true değeri verir.
- restoreCard
Eşleşmemiş kartların faceUp larını false yapar.
- checkForMatch
Kartlardaki eşleştirme işlemlerinin kontrolünü yapar.
- myTimer
Oyun içerisindeki zamanlayıcı anlar

A. Tasarlanan Mimari

VI. PSEUDO KOD(KABAKOD)

1. Giriş ekranı gelir.
2. Eğer kullanıcı adı ve şifre girerse bunları apiye ilet cevaba göre onayla veya alert dön.
3. Eğer şifre unuttuma basarsa git. Eski şifresini ve yeni şifresini al. Apiye yolla. Api cevabına göre şifreyi değiştir.

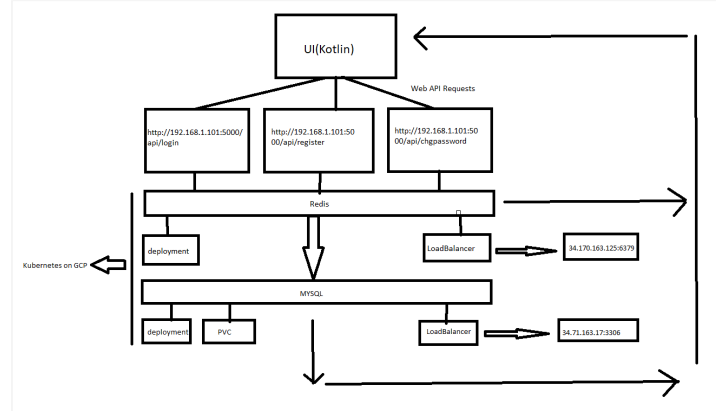


Fig. 2. Uygulama Mimarisi

4. Yeni kullanıcı oluşturmak istiyorsa git. Bilgileri al. Kaydet apiye yolla. Api'de kayıt işlemini gerçekleştir. Response dön.
5. Kullanıcı başarılı bir şekilde giriş yaparsa oyun tercihleri yapacağı sayfayı getir.
6. Kullanıcıdan tercihleri iste. Her tercihi girdikten sonra ilgili oyunu CardAPI'sine giderek kart bilgilerini çek.
- O ara kullanıcıya SplashScreen'i göster.
7. Cardları kotlin üzerinde spesifik bir class'a çek. Sonrasında kartları shufflela.
8. Her bir butonu kart bilgileriyle eşleştir.
9. Timer'i başlatarak oyunu başlat. Oyun başladığı anda müziği de çal.
10. Eğer kullanıcı doğru eşleşme yaparsa ilgili müziği çal. Kazandığı puanı hesapla. Arayüzde göster.
11. Eğer kullanıcı yanlış eşleşme yaparsa kaybettiği puanı hesapla. Skorundan çıkar. Bastığı kart bilgisini textview olarak ScrollBar'a ekle.
12. Süreyi sürekli kontrol et. Tüm kartlar eşleşmeden süre bittiyse kaybettin yaz ve kaybettin müziğini çaldır.
13. Süre bitmeden kartlar eşleştirilmişse kazandın yaz ve kazandın müziğini çal.
14. Oyun bittiğinde kullanıcıyı ana menüye dön ve tekrar oyna olarak 2 butonla karşıla.
15. Bu döngüye sürekli böyle devam et.

VII. SONUÇ

Bu projede birçok kazanım elde ettik. Ayrıca birçok sorunla karşılaştık. Bu durumun bizi oldukça geliştirdiğini öğrendik. Elde ettiğimiz kazanımlardan bahsedecek olursak;

- Android Studio ortamını öğrenmek,
- Kotlin ile yazılım geliştirmeyi öğrenmek,
- Event Listener yapılarını kullanmayı öğrenmek,
- Bulut platformunu öğrenmek,
- Veri tabanı CRUD işlemlerini öğrenmek,
- API ile farklı diller arasında veri akışı sağlamayı öğrenmek,

Kazanımlarını elde ettik.

VIII. KAYNAKÇA

- <https://medium.com/kodlayan-nesil/flask-nedir-9364c1bb5f41>
- <https://aws.amazon.com/tr/what-is/api/>
- <https://kotlinlang.org/>
- <https://developer.android.com/studio>
- <https://github.com/>
- <https://www.btkakademi.gov.tr/portal/course/kotlin-ile-android-mobil-uygulama-gelistirme-egitimi-temel-seviye-10274>
- <https://bilgisayarkavramlari.com>
- <https://medium.com/@ecemsuren/kotlin-temelleri-2c9ad608af7d>
- <https://stackoverflow.com/>
- <https://www.protan.com.tr/mysql-nedir/>
- <https://devnot.com/2020/redis-nedir-temel-kullanim-alanlari-nelerdir/>
- <https://deniz-turkmen.medium.com/dockerfile-a177c7796d53>
- <https://www.google.com/url?sa=iurl=https>

IX. DENEYSEL SONUÇLAR

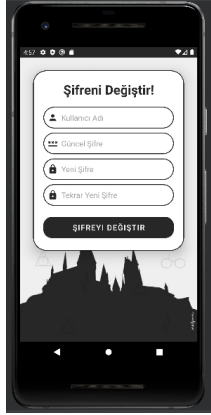


Fig. 3. Şifre Değiştirme Ekranı



Fig. 4. Oyundan Bir Görüntü

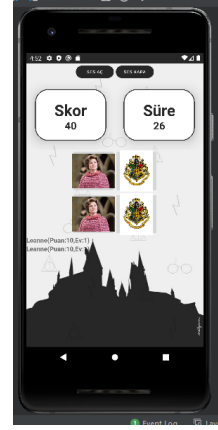


Fig. 5. Oyundan Bir Görüntü

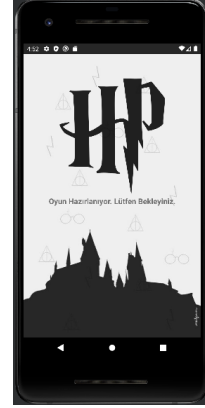


Fig. 6. Yüklem Ekranı



Fig. 7. Ana Menü