



**T.C
DOKUZ EYLÜL ÜNİVERSİTESİ
FEN FAKÜLTESİ
BİLGİSAYAR BİLİMLERİ BÖLÜMÜ**

YABANCI DİL ÖĞRENME UYGULAMASI

Müslüm Berkay YILMAZ

Sefa Berk BIYIKCI

Danışman: DR. ÖĞR. Üyesi Ayşe Övgü KINAY

Mayıs, 2022

İZMİR

Müslüm Berkay YILMAZ ve Sefa Berk BIYIKCI tarafından **DR. ÖĞR. Üyesi Ayşe Övgü KINAY** yönetiminde hazırlanan “**YABANCI DİL ÖĞRENME UYGULAMASI**” başlıklı rapor tarafımızca okunmuş, kapsamı ve niteliği açısından bir Bitirme Projesi olarak kabul edilmiştir.

DR. ÖĞR. Üyesi
Ayşe Övgü KINAY

Proje alıřmaları suresince hem proje hem de Tbitak bařvurusu ařamalarında desteęini eksik etmeyen danıřmanımız Dr. Ayře vg KINAY'a, ve altyazı arřivini bizimle paylařan OpenSubtitles.org yetkililerine teřekkr bor biliriz.

ÖZET

Bu projede, yabancı dil öğrenenlerin kelime hazinesini etkin bir şekilde genişletmelerini sağlayacak bir mobil uygulama geliştirilmiştir. Öğrenme süreçlerini kişiselleştiren ve kullanıcıların hafızalarını güçlendiren bir tekrar algoritması içeren uygulama, kullanıcı deneyimini gamifikasyon öğeleri ile zenginleştirerek motivasyon sağlamayı hedefler. Çok çeşitli kelimeleri içeren geniş bir veritabanına sahip olan uygulama, kullanıcılarının öğrenme ihtiyaçlarına göre uyum sağlayabilen interaktif bir arayüze sahiptir. Android ve iOS platformları için tasarlanan bu uygulama, kullanıcıların yeni kelimeler öğrenirken eğlenmelerini ve bu kelimeleri uzun vadede hatırlamalarını sağlamayı amaçlamaktadır.

Anahtar kelimeler: Yabancı Dil Eğitimi, Kelime Hazinesi Geliştirme, Mobil Uygulama, Gamification, Kişiselleştirilmiş Öğrenme.

ABSTRACT

This project develops a mobile application designed to efficiently expand the vocabulary of individuals learning a foreign language. Featuring a repetition algorithm that personalizes the learning process and strengthens memory retention, the application aims to enhance user engagement through gamification elements. It boasts an extensive database with a wide variety of words and an interactive interface that adapts to the learning needs of its users. Developed for both Android and iOS platforms, the application seeks to make learning new words enjoyable for users and to ensure long-term retention.

Keywords: Foreign Language Education, Vocabulary Enhancement, Mobile Application, Gamification, Personalized Learning

İÇİNDEKİLER

ÖZET	4
ABSTRACT	5
İÇİNDEKİLER.....	6
ŞEKİLLER DİZİNİ.....	7
1. PROJE KONUSU	8
2. GİRİŞ.....	8
2.1. Problem.....	8
2.2. Çözüm.....	9
3. YÖNTEMLER.....	10
3.1. Flutter.....	10
4. MİMARİ VE ALGORİTMALAR	12
4.1. Genel Hatlar	12
4.2. Algoritma	12
4.2.1. Kelimelerin ham veriden ayrıştırılması	12
4.2.2. Kelime frekanslarına göre kullanıcının bilmediği düşünülen kelimeler saptanması	13
4.2.3. GloVe aşaması.....	13
4.2.4. FastText	13
4.2.5 LSTM	14
4.2.6 Seviye değişimlerini tespit edilmesi	14
4.2.7 Hedef Belirleme.....	14
4.2.8 Testlerdeki sorunlar	14
4.3. Backend	15
4.4. SQL	17
4.5. Firestore:	18
4.6 Mobil İstemci:	21
5. UYGULAMA	22
5.1. Mobil Uygulama Arayüzü	22
5.2. Kelime Listesi Seçimi.....	23
5.3 Kelimeleri Öğrenme Aşaması	25
6. SONUÇ VE DEĞERLENDİRME	26
KAYNAKÇA.....	28

ŞEKİLLER DİZİNİ

Şekil 4.1. Kelime tokenizasyon ve lematizasyon kodları.....	16
Şekil 4.2. Backend'den Firebase'e upload işlemleri	17
Şekil 4.3. SQL veritabanı yapısı.....	18
Şekil 4.4 Firebase üzerinde seviye belirleme veri yapısı	19
Şekil 4.5. Firebase Kullanıcı Modeli ve Seviye Belirleme Testi Yapısı.....	20
Şekil 4.6. Firebase Kullanıcı Veri Modeli.....	21
Şekil 4.7. Uygulama Veri Akış Şeması.....	22
Şekil 5.1 Uygulama Anasayfası	23
Şekil 5.2 Kelime Listeleri.....	24
Şekil 5.3. Öğrenme Ekranları	25
Şekil 5.4. Öğrenme Ekranları 2	26

1. PROJE KONUSU

Bu bitirme projesi kapsamında, yabancı dil öğrenen bireylerin kelime dağarcıklarını etkin bir şekilde genişletmelerine yardımcı olmayı amaçlayan bir mobil uygulamanın tasarımı ve geliştirilmesi ele alınmıştır. Dil öğrenimi sürecinde karşılaşılan en büyük engellerden biri, öğrenilen yeni kelimelerin akılda kalıcı ve etkili bir biçimde pekiştirilmesidir. Bu projede, öğrencilere kişiselleştirilmiş bir kelime öğrenme deneyimi sunmak ve tekrar etme algoritması üzerinden kelimelerin uzun süreli hafızada kalıcılığını artırmak hedeflenmiştir. Uygulama, kullanıcı etkileşimine dayalı olarak bireysel öğrenme eğilimlerini analiz edebilen ve bu analizleri kullanarak kullanıcının öğrenme sürecine en uygun kelimeleri sunan dinamik bir algoritma kullanmaktadır. Gelişmiş bildirim mekanizmaları, gamifikasyon öğeleri ve kullanıcı etkileşimini teşvik eden bir arayüz ile öğrencinin motivasyonunu yüksek tutmayı ve etkili bir öğrenme süreci sağlamayı amaçlayan uygulama, Firebase ve SQL veritabanları üzerine inşa edilmiş ve hem Android hem de iOS platformlarında çalışacak şekilde tasarlanmıştır.

2. GİRİŞ

2.1.Problem

Dil öğreniminde kelime dağarcığı, dilin özünü oluşturur ve geniş bir kelime bilgisi olmadan etkili iletişim kurmak zorlaşır. Gramer yapısı bir duvarın çimentosu gibi dilin yapısal temelini oluştururken, kelimeler ise bu yapıyı oluşturan temel bloklardır. Ancak bir dili akıcı konuşmak, anlamak ve yazmak için gerekli olan geniş kelime dağarcığını geliştirmek öğrenciler için çeşitli zorluklar barındırır.

Problem A, öğrencilerin hangi kelimeleri öğreneceklerine karar verirken ortaya çıkar. Seçilecek kelimelerin öğrencinin dil seviyesine uygun ve günlük hayatta sıkça karşılaşılabilecekleri türden olması önemlidir. Projemizde bu sorunu, kullanıcıların seviyelerine uygun en optimal medyaları belirleyerek çözüyoruz. Dizi, film, müzik, YouTube videoları ve kitap gibi kaynaklardan seçkiler sunarak, kullanıcıların kelime öğrenimini doğal bir sürece dönüştürüyoruz.

Problem B, istenen kelimelere ulaşma sürecidir. Öğrencinin günlük hayatta etkileşimde bulunduğu çeşitli içeriklerden kelimeleri gruplayarak, organizasyon ve erişimi

kolaylaştırıyoruz. Makine öğrenimi algoritmaları ile öğrencinin zaten bildiği kelimeleri elemek, öğrenme sürecini daha da kişiselleştiriyor. Ayrıca, kullanıcıların kendi kelime listelerini oluşturmalarına da olanak tanıyoruz.

Problem C, öğrenilen kelimeleri etkili bir biçimde hafızada tutmakla ilgilidir. Çoklu duyu organlarımız üzerinden beyne gönderilecek veriler ile öğrenme sürecini güçlendiriyoruz. Görsel ve işitsel materyallerle desteklenen sorularla, geri çağırma işlemi yoluyla bilginin beyinde daha kalıcı hale getirilmesini hedefliyoruz. Çoktan seçmeli soruların yanı sıra, resim, telaffuz, çeviri veya anlam üzerinden kelime sorarak, farklı yöntemlerle geri çağırma pratikleri yapıyoruz. Bu yöntemler, öğrenilen kelimelerin beyinde daha etkin ve kalıcı bir şekilde yer etmesini sağlar.

Problem D, öğrenilen kelimelerin uzun vadede hafızada kalmasını sağlamaktır. Unutma eğrisi, tekrarsız öğrenilen bilginin zamanla azaldığını gösterir. Optimum tekrar aralıklarını belirlemek için bilimsel çalışmalardan yararlanıyoruz. Bu çalışmalara dayanarak geliştirdiğimiz bir bildirim sistemi ile öğrencilerin kelimeleri en az eforla en iyi şekilde hatırlamalarını sağlıyoruz.

Özetle, bu proje ile dil öğrenmek isteyen bireylere, ilgi alanlarına uygun ve kendilerinin tercih ettiği çeşitli medyalar aracılığıyla, keyifli ve etkili bir yöntemle ihtiyaç duydukları kelimeleri öğretmeyi amaçlıyoruz.

2.2.Çözüm

Yabancı dil öğrencilerinin karşılaştığı en büyük engellerden biri, geniş bir kelime dağarcığını etkili bir şekilde öğrenmek ve bu kelimeleri uzun süreli hafızada tutabilmektir. Bu proje, kelime öğrenme sürecini optimize eden ve kullanıcıların dil öğrenimini hızlandıran bir mobil uygulama geliştirerek bu soruna çözüm sunmayı hedeflemektedir. Uygulama, Spaced Repetition ve Leitner System gibi kanıtlanmış öğrenme algoritmalarını kullanarak, her bir kullanıcının öğrenme kapasitesine ve unutma eğilimlerine göre kişiselleştirilmiş bir tekrar planı oluşturur. SQLite ve Firebase teknolojilerini kullanarak, geniş bir kelime veritabanını yönetir ve kullaYabancı dil öğrencilerinin karşılaştığı en büyük engellerden biri, geniş bir kelime dağarcığını etkili bir şekilde öğrenmek ve bu kelimeleri uzun süreli hafızada tutabilmektir. Bu proje, kelime öğrenme sürecini optimize eden ve kullanıcıların

dil öğrenimini hızlandıran bir mobil uygulama geliştirerek bu soruna çözüm sunmayı hedeflemektedir. Uygulama, Spaced Repetition ve Leitner System gibi kanıtlanmış öğrenme algoritmalarını kullanarak, her bir kullanıcının öğrenme kapasitesine ve unutma eğilimlerine göre kişiselleştirilmiş bir tekrar planı oluşturur. SQLite ve Firebase teknolojilerini kullanarak, geniş bir kelime veritabanını yönetir ve kullanıcı etkileşimini analiz eder. Böylece, öğrencilere hem bireysel öğrenme yollarını takip etme özgürlüğü sunar hem de onlara özgü hafıza paternlerine en uygun tekrar zamanlamasını sağlar. Mobil platformların yaygın kullanımı sayesinde, bu uygulama kullanıcılara her zaman ve her yerde öğrenme imkanı sunarak kelime hazinesinin geliştirilmesine katkıda bulunur. Mobil platformların yaygın kullanımı sayesinde, bu uygulama kullanıcılara her zaman ve her yerde öğrenme imkanı sunarak kelime hazinesinin geliştirilmesine katkıda bulunur.

3. YÖNTEMLER

Bu bölüm, Vocabulary Booster uygulamasının kullanıcı deneyimini iki temel etkinlik üzerinden ele alacaktır: "Kelime Listesi Seçimi" ve "Kelimeleri Öğrenme Aşaması". Her bir alt başlık, bu süreçlerde kullanıcının deneyimlediği özellikler ve etkileşimler üzerinde yoğunlaşacak ve uygulamanın bu etkinlikleri nasıl optimize ettiğini açıklayacaktır.

3.1. Flutter

Flutter, Vocabulary Booster uygulamasının ön yüzünün geliştirilmesinde kullanılan bir UI toolkitidir. Google tarafından geliştirilen bu açık kaynaklı framework, Dart programlama dilini kullanır ve hızlı, etkili, platformlar arası mobil, web ve masaüstü uygulamaları geliştirme imkanı sağlar.

3.1.1. Kullanıcı Arayüzü Oluşturma

Flutter'ın widget tabanlı yapısı sayesinde, zengin ve özelleştirilebilir kullanıcı arayüzleri oluşturmak mümkündür. Vocabulary Booster uygulamasında, interaktif ve sezgisel bir deneyim sunmak üzere özel tasarlanmış widget'lar kullanılmıştır.

3.1.2. Performans Optimizasyonu

Flutter, yüksek performanslı uygulamalar oluşturma konusunda güçlüdür. "Hot Reload" özelliği ile geliştirme sürecinde yapılan değişiklikler, derleme yapmaya gerek kalmadan doğrudan uygulamada görülebilir, bu da hızlı prototipleme ve hataları azaltma açısından önemlidir.

3.1.3. Platformlar Arası Uyumluluk

Flutter ile geliştirilen uygulama, kod tabanının büyük bir kısmını Android ve iOS platformları arasında paylaşabilir. Bu, uygulamanın her iki sistemde de tutarlı bir kullanıcı deneyimi sunmasını ve geliştirme süreçlerinin verimliliğini artırır.

3.2 Firebase

Firebase, uygulamanın arka uç geliştirme ihtiyaçlarını karşılamak için kullanılan bir BaaS (Backend-as-a-Service) platformudur. Google tarafından sunulan bu servis, uygulama geliştiricilere veritabanı, kimlik doğrulama, hosting gibi hizmetler sağlar.

3.2.1. Veri Depolama ve Senkronizasyon

Firebase, NoSQL tabanlı bir veritabanı olan Firestore'u kullanır. Bu veritabanı, Vocabulary Booster uygulamasının gerçek-zamanlı veri senkronizasyonu ve offline destek gibi özelliklerini destekler. Kullanıcıların kelime listeleri, öğrenme ilerlemeleri gibi dinamik verileri, etkin bir şekilde yönetilir ve güncellenir.

3.2.2. Kullanıcı Kimlik Doğrulama

Firebase Auth, uygulamaya sosyal medya hesapları, e-posta ve şifre veya telefon numarası ile giriş yapma gibi çeşitli kimlik doğrulama seçenekleri sunar. Bu sayede kullanıcılar, Vocabulary Booster uygulamasına kolay ve güvenli bir şekilde erişebilirler.

3.3 Python

Projemizde, Python dilinin okunabilir ve anlaşılır yapısından faydalanarak, kodun bakımını ve ekip çalışmasını kolaylaştırdık. Özellikle veri işleme ve algoritma geliştirme aşamalarında Python'un geniş kütüphane desteği, işlevsel gereksinimlerimizi hızlı ve etkin bir şekilde yerine getirmemize olanak tanıdı. Veri analizinde pandas ve NumPy gibi kütüphaneler kullanarak işlemlerimizi verimli bir şekilde yürüttük ve matplotlib ile görselleştirme yaptık

3.4 SQLite

SQLite, projemizde hafif bir veritabanı çözümü olarak tercih edildi. Basit konfigürasyonu ve sunucu gereksinimleri olmaması, özellikle hızlı geliştirme ve prototipleme aşamalarında bize avantaj sağladı. Kullanıcı tercihleri, uygulama verileri ve sistemin gereksinim duyduğu diğer veriler SQLite veritabanında saklandı. ACID uyumluluğu sayesinde verilerin güvenliğinden emin olurken, projemizin portatif ve hafif kalmasını sağladı.

3.5 Flask

Flask, apılarımızın geliştirilmesinde kullanılan bir mikro web çatısıdır. Kullanıcıların etkileşimini kolaylaştırmak için basit ama güçlü bir arayüz oluşturmamıza olanak verdi. Flask'ın rotalama mekanizmaları, RESTful API yapımızı destekleyerek, kullanıcıların sisteme sorgu göndermelerine ve yanıtlar almalarına imkan tanıdı. Flask'ın genişletilebilir yapısından yararlanarak, projemiz için özel araçlar ve uzantılar ekledik. Örneğin, kullanıcı oturum yönetimi için Flask-Login ve form yönetimi için Flask-WTF gibi uzantılar kullanıldı. Bu sayede, kullanıcı dostu ve etkileşimli bir deneyim sunmayı başardık.

4. MİMARİ VE ALGORİTMALAR

4.1.Genel Hatlar

Uygulama mimarisini genel anlamda Algoritma - Backend - SQL - Firestore - Mobil İstemci başlıklarıyla inceleyeceğiz.

4.2.Algoritma

Kullanıcının öğrenmek istediği şarkının kelimelerinin bildiği kelimelere anlamca yakın olanların sayısına göre bilip bilmediğini tahmin ederek testler oluşturan bir algoritma geliştirildi.

4.2.1. Kelimelerin ham veriden ayrıştırılması

Kelimeler ham veriden ayrıştırılması aşağıdaki adımlardan oluşmaktadır:

1. Stopword'lerin kaldırılması
2. Tokenization
3. Lemmatization

4. Headword'lerin bulunması
5. Kelimelerin word family'lere ayrılması (BNC/COCA DB)

4.2.2. Kelime frekanslarına göre kullanıcının bilmediği düşünölen kelimeler saptanması

Kelimeler elimizdeki İngilizce corpus'daki yaygınlıklarını (her dildeki yaygınlığı aynı olarak varsayıldı) TF-IDF'e benzer bir yöntemle bulunmakta.

Kullanıcıya başlangıçta bir kere ve uygulamayı kullandıkça yapılacak testlerle kullanıcının öğrenmek istediğı dildeki kelime bilgisi hesaplandı (1000'er kelimelik en yaygından en nadire vocabulary level gruplarından, kullanıcının %80 gibi bir eşik değerin üzerinde bildiğı en yüksek level kullanıcının dil seviyesi olarak belirleniyor)

(Bkz. Stoeckel, McLean, Nation, Limitations of size and levels tests of written receptive vocabulary knowledge (2021))

4.2.3. GloVe aşaması

Pretrained bir GloVe modeli çıktı olarak word vector'leri döndürmekte. (cosine distance) ve işlenen medyalardaki tüm kelimeler, modeldeki kelimelerin birbirlerine semantik uzaklıkları hesaplandı, böylece kullanıcı herhangi bir kelimeye yakın anlamda k adet kelimeyi halihazırda biliyorsa o kelimeyi de bildiğı varsayılarak kullanı testleri oluşturulması için gereken kelime sayısının azaltılmasını amaçladık. Böylece kullanıcı daha az bilgi sahibi olduğı alanlardaki kelimeleri daha önce öğrenebilsin. 'k' değerin 3 olarak aldık.

Embedding modelini seçerken aşağıdaki makalelerden faydalandık:

[An empirical study of semantic similarity in WordNet and Word2Vec \(uno.edu\)](#)

[Word2Vec vs GloVe - A Comparative Guide to Word Embedding Techniques \(analyticsindiamag.com\)](#)

4.2.4. FastText

Çalışma mantığı Word2Vec'e çok benzer olmakla beraber diğ er yöntemlerin aksine yazım hatası içeren kelimelerde de çalışmakta. İleride iç erisinde hatalar bulunabilen OCR

(optical character recognition) yöntemiyle elde edilen pdf dokümanları verileriyle iyi çalışmakta. Ancak proje kapsamında film ve dizi verileriyle çalışacağımız için, bu yöntemi diğerlerine kıyasla performans dezavantajı sebebiyle kullanmayacağız.

4.2.5 LSTM

Kelime yakınlığı bilgisi için Word2Vec'in yeterli olduğunu gördüğümüzden LSTM'e ihtiyaç olmadığına karar verdik. Bkz. Overview of LSTMs and " at <https://www.cam.ac.uk>

4.2.6 Seviye değişimlerini tespit edilmesi

Boyut ve seviye testlerinin ikinci yaygın kullanımı vocabulary size'daki gelişmeyi ölçmektir. Bir testin bu işlevini yerine getirebilmesi için kullanıcının sözlük bilgisindeki anlamlı değişiklikler hassas bir şekilde takip edilmeli.

WordBite'a yansması: Kullanıcının yeni öğrendiği her kelime takip edilerek vocabulary-size puanında gerekli güncelleştirmeler yapılır, bu sayede maksimum hassasiyet sağlanır. Yani kullanıcı her yeni kelime öğrenme akışına girdiğinde vocabulary-size puanı doğruluğu artar.

4.2.7 Hedef Belirleme

Vocabulary size ve seviye tespitinde bir diğer önemli amaç öğrencilerin öğrenmeleri için hedefler belirlemelerine yardımcı olmaktır.

Amaç, öğrencilerin hedef belirlemelerine yardımcı olmak olduğunda, hangi yüksek frekanslı kelimelerin bilinmediğini belirleyebilmek önemlidir. Bu noktada da ölçüt olarak sıklığı kullanmak yerine, öğrenme ihtiyacına göre özel bir kelime grubuna odaklanmak verimli olabilir.

Bkz: Stoeckel, Tim & McLean, Stuart & Nation, Paul. (2020)

4.2.8 Testlerdeki sorunlar

Testlerdeki doğruluğu etkileyebilecek 3 parametre var, bunlar; madde formatı, kelime gruplarının gruplama ilkeleri ve sorulacak kelime sayısıdır.

Biz uygulama kapsamında test başına sabit 20 kelime kullanmayı uygun gördük. Kelime gruplarının gruplamasını ise medyada geçme sırası olarak uyguladık.

Orta-Frekans Seviyesi	Örnek Kelimeler
3,001-4,000	Academic, consist, exploit, rapid, vocabulary
4,001-5,000	Agricultural, contemporary, dense, insight, particle
5,001-6,000	Cumulative, default, penguin, rigorous, schoolchildren
6,001-7,000	Axis, comprehension, peripheral, sinister, taper
7,001-8,000	Authentic, conversely, latitude, mediation, undergraduate
8,001-9,000	Anthropology, fruitful, hypothesis, semester, virulent

Tablo: Farklı 1,000 orta-frekans seviyelerinden örnek kelimeler

(Kaynak: 6,000–7,000 word families for spoken discourse and 8,000–9,000 for written discourse)

van Zeeland & Schmitt (Reference van Zeeland and Schmitt, 2012). Nation (Reference Nation, 2006)

Amacımız kullanıcının izlemek istediği içeriği anlamasını sağlamak olsa da o dilde yeterli olduğuna kanaat getirmek için 6000-9000 arası kelimeyi yeterli kabul edeceğiz.

4.3.Backend

Bir sanatçı listesi oluşturduk ve bu sanatçıların en popüler şarkı sözlerini, albüm kapak fotoğraflarını Genius API aracılığıyla işleyip firebase veritabanına mobil uygulamanın okuyacağı formatta otomatik olarak kaydeden bir sistem geliştirdik.

```

# Stopword removal
def filter_out_stopwords(lemmas):
    custom_stopwords = ["something", "ourselves", "hers", "between", "yourself", "but", "again", "the"]

    stop_words = set(stopwords.words("english") + custom_stopwords)

    return [lemma for lemma in lemmas if not lemma in stop_words]

# Returns Lemmas out of given text.
def extract_lemmas(text):
    # Lowercasing
    text = text.lower()

    # Tokenization
    word_tokens = word_tokenize(text)

    # Word extraction
    words = filter_nonalphanumerics(word_tokens)
    print(len(set(words)), ' words in the media.')

    # Lemmatization
    lemmas = lemmatize(words)

    # Stopword removal
    return filter_out_stopwords(lemmas)

# Validate word from dictionary. Return ones with no meaning in dictionary.
def dictionary_validation(words, dictionary="wordnet"):
    in_dictionary = set()
    not_in_dictionary = set()

    if dictionary == "wordnet":
        for word in words:
            if not wordnet.synsets(word):
                not_in_dictionary.add(word)
            else:
                in_dictionary.add(word)

    elif dictionary == "webster":
        from webster_dictionary import is_in_webster
        for word in words:
            if is_in_webster(word) == False:
                not_in_dictionary.add(word)

```

Şekil 4.1. Kelime tokenizasyon ve lematizasyon kodları


```

84         pass
85
86         word_to_media_words(media_id, word_id)
87
88     def artist_data_to_firebase(artist_name, song):
89         db.collection('Songs').document(artist_name).set(song)
90
91     def artist_list_data_to_dict(lists_dict, titles, artist_name, media_ids, cover_urls):
92
93     # # Artist List Data Format
94     # artist_list_data = {
95     #     artist_name : {
96     #         0 : {
97     #             'title' : title,
98     #             'artist_name' : artist_name,
99     #             'media_id' : media_id,
100            #             'cover_url' : cover_url
101            #         }
102            #         1 : {
103            #             'title' : title,
104            #             'artist_name' : artist_name,
105            #             'media_id' : media_id,
106            #             'cover_url' : cover_url
107            #         }
108            #     }
109            #     artist_name : {
110            #         0 : {
111            #             'title' : title,
112            #             'artist_name' : artist_name,
113            #             'media_id' : media_id,
114            #             'cover_url' : cover_url
115            #         }
116            #     }
117            # }
118            #
119            #
120            #
121            #
122
123     artist_list_data = []
124
125     for i in range(len(titles)):
126         artist_list_data.append({
127             'title' : titles[i],
128             'artist_name' : artist_name,
129             'media_id' : media_ids[i],
130             'cover_url' : cover_urls[i]
131         })

```

Şekil 4.2. Backend'den Firebase'e upload işlemleri

4.4.SQL

Backend üzerinde bir SQLite veritabanı çalışmaktadır ve şarkı sözlerinden işlediği veri bu veritabanına yazılır, ilişkisel veritabanında, kullanıcı dil seviyeleri - kelime - kelime anlamları - kelime çevirileri - en yakın kelimeler, kelime başına anlamca en yakın 6 kelime - şarkı - sanatçı - şarkı sözü dosya path verileri tutulmaktadır.

Yeni Veritabanı Veritabanı Aç Değişiklikleri Kaydet Değişiklikleri Geri Al Proje Aç Projeyi Kaydet Veritabanı Ekle Veritabanı Kapat		
Veritabanı Yapısı Veriyi Görüntüle Pragmaları Düzenle SQL kodunu yürüt		
Tablo Oluştur Index Oluştur Yazdır		
İsim	Tip	Şema
▼ Tablolar (11)		
> de		CREATE TABLE "de" ("word_id" INTEGER NOT NULL UNIQUE, "translation" TEXT, FOREIGN KEY("word_id") REF
> en		CREATE TABLE "en" ("word_id" INTEGER NOT NULL UNIQUE, "translation" TEXT NOT NULL UNIQUE, PRIMARY K
> en_most_similars		CREATE TABLE "en_most_similars" ("word_id" INTEGER NOT NULL UNIQUE, "similar_1" TEXT, "similar_2" TEXT
word_id	INTEGER	"word_id" INTEGER NOT NULL UNIQUE
similar_1	TEXT	"similar_1" TEXT
similar_2	TEXT	"similar_2" TEXT
similar_3	TEXT	"similar_3" TEXT
similar_4	TEXT	"similar_4" TEXT
similar_5	TEXT	"similar_5" TEXT
similar_6	TEXT	"similar_6" TEXT
> fr		CREATE TABLE "fr" ("word_id" INTEGER NOT NULL UNIQUE, "translation" TEXT, PRIMARY KEY("word_id"))
> media		CREATE TABLE "media" ("media_id" INTEGER NOT NULL UNIQUE, "media_type" TEXT NOT NULL, "media_name
media_id	INTEGER	"media_id" INTEGER NOT NULL UNIQUE
media_type	TEXT	"media_type" TEXT NOT NULL
media_name	TEXT	"media_name" TEXT NOT NULL
genius_id	INTEGER	"genius_id" INTEGER DEFAULT NULL UNIQUE
opensubtitles_id	INTEGER	"opensubtitles_id" INTEGER DEFAULT NULL UNIQUE
imdb_id	INTEGER	"imdb_id" INTEGER DEFAULT NULL UNIQUE
> media_words		CREATE TABLE "media_words" ("media_id" INTEGER NOT NULL, "word_id" INTEGER NOT NULL, FOREIGN KEY("
media_id	INTEGER	"media_id" INTEGER NOT NULL
word_id	INTEGER	"word_id" INTEGER NOT NULL
> ru		CREATE TABLE "ru" ("word_id" INTEGER NOT NULL UNIQUE, "translation" INTEGER, PRIMARY KEY("word_id"))
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
> tr		CREATE TABLE "tr" ("word_id" INTEGER NOT NULL UNIQUE, "translation" TEXT, PRIMARY KEY("word_id"), FOR
> tr_most_similars		CREATE TABLE "tr_most_similars" ("word_id" INTEGER NOT NULL UNIQUE, "similar_1" TEXT, "similar_2" TEXT
word_id	INTEGER	"word_id" INTEGER NOT NULL UNIQUE
similar_1	TEXT	"similar_1" TEXT
similar_2	TEXT	"similar_2" TEXT
similar_3	TEXT	"similar_3" TEXT
similar_4	TEXT	"similar_4" TEXT
similar_5	TEXT	"similar_5" TEXT
similar_6	TEXT	"similar_6" TEXT
> words		CREATE TABLE "words" ("word_id" INTEGER NOT NULL UNIQUE, "frequency" INTEGER DEFAULT NULL, PRIMAR
> İndisler (0)		
> Görünümler (0)		
> Tetikleyiciler (0)		

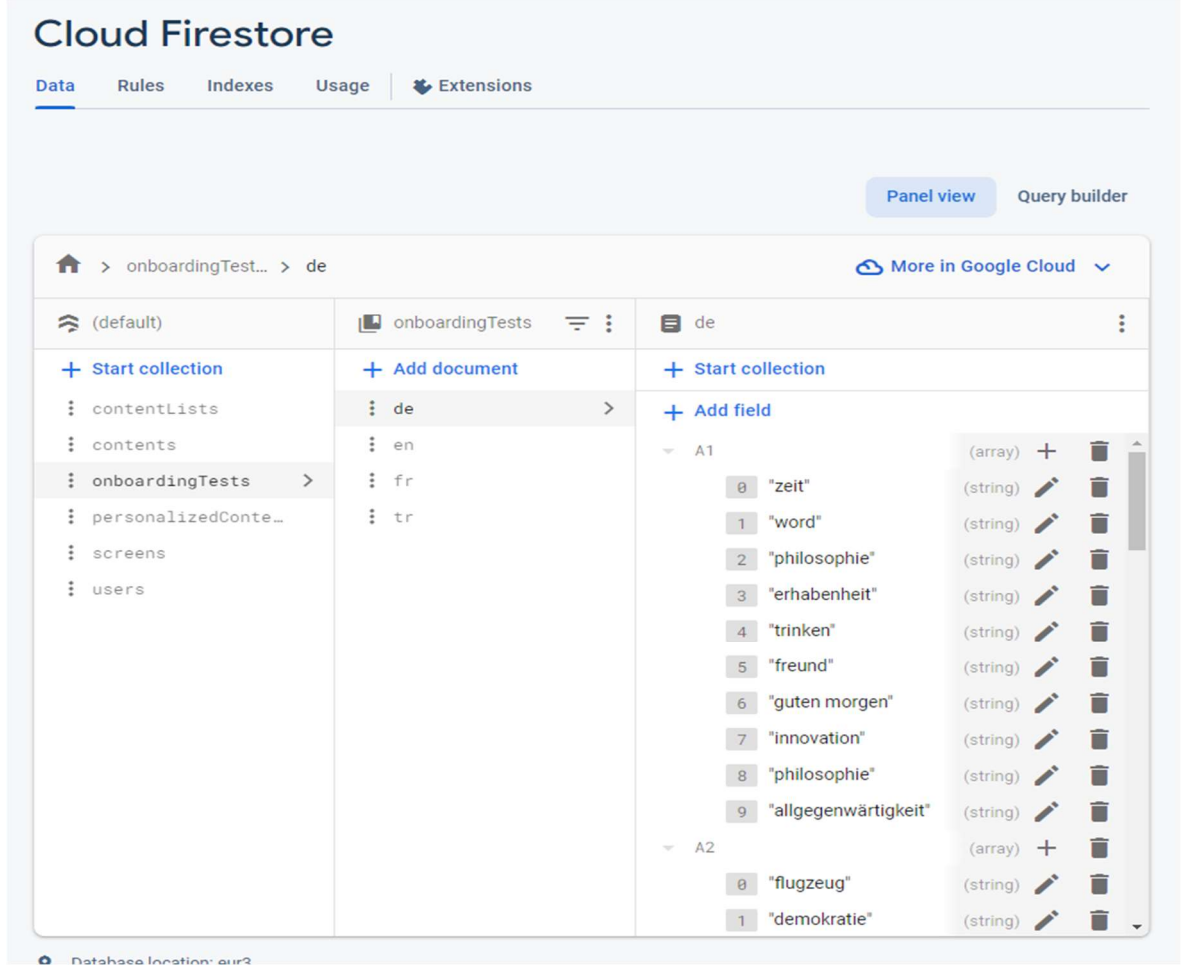
Şekil 4.3. SQL veritabanı yapısı

Mobil uygulama test talep ettiğinde ilgili şarkıyı kullanıcıya göre işleyerek response döndürmektedir.

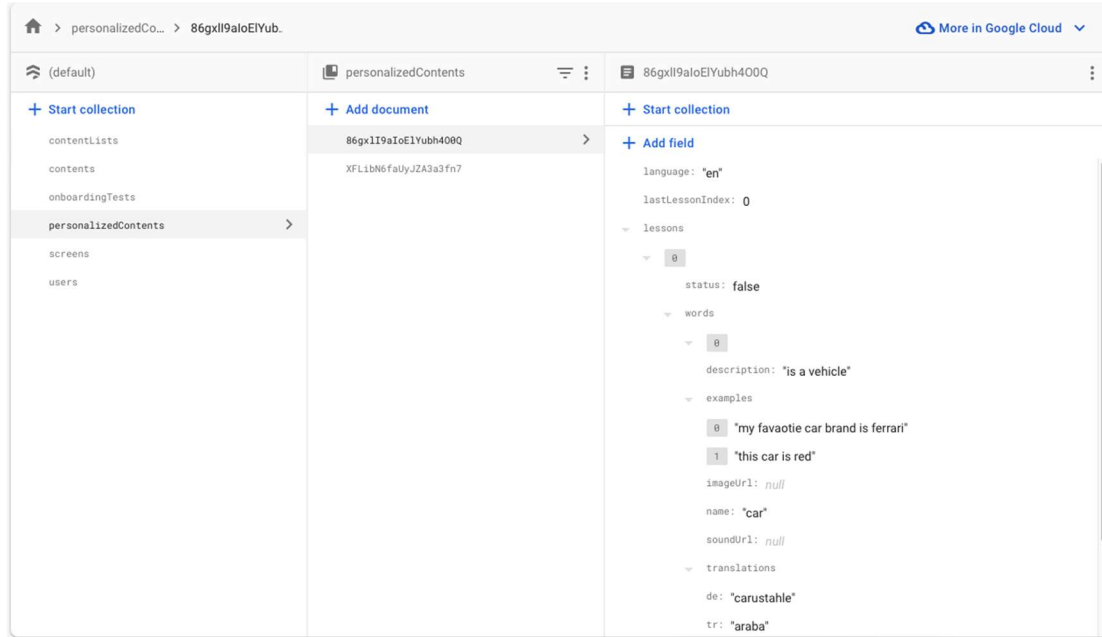
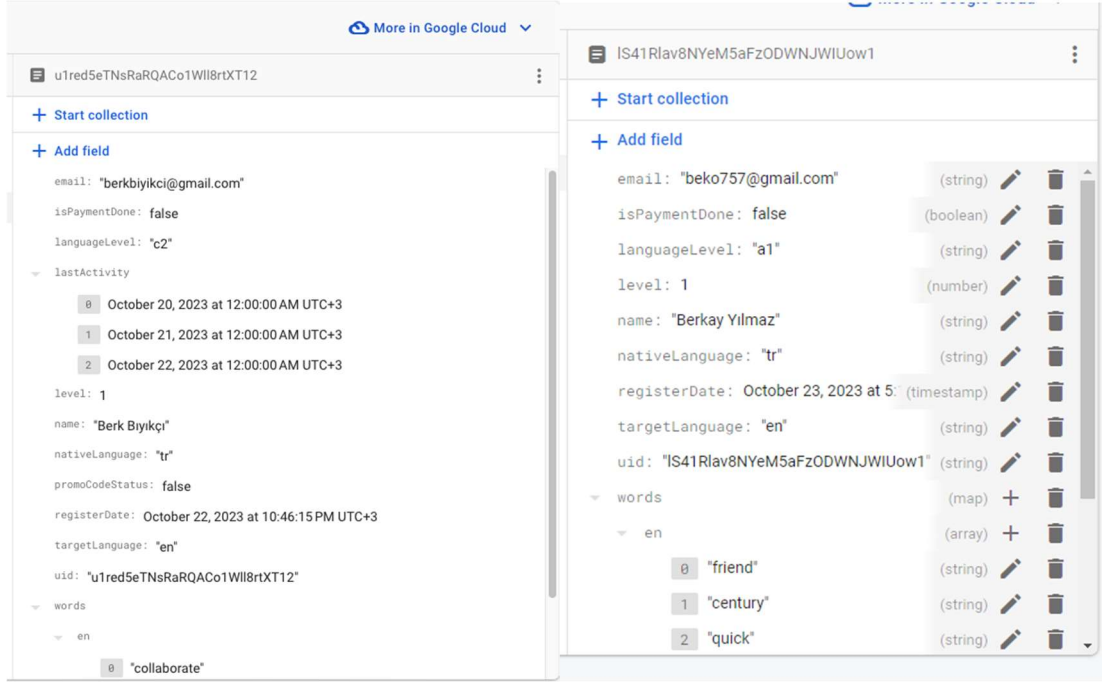
4.5. Firestore:

Firestore, uygulamamızın hayati verilerini saklayan, doküman tabanlı NoSQL bir veritabanı hizmetidir. Mobil uygulama, kullanıcının ilerlemesine, kişiselleştirilmiş kelime listelerine ve oturum bilgilerine ilişkin tüm canlı verileri Firestore'dan alır. Öğrenme modüllerinden alınan kullanıcı verileri ve performans istatistikleri bu veritabanında güvenli bir şekilde saklanır. Veritabanının gerçek-zamanlı senkronizasyon yetenekleri sayesinde, kullanıcılar herhangi bir cihazdan giriş yaptıklarında en güncel verilere anında erişebilirler.

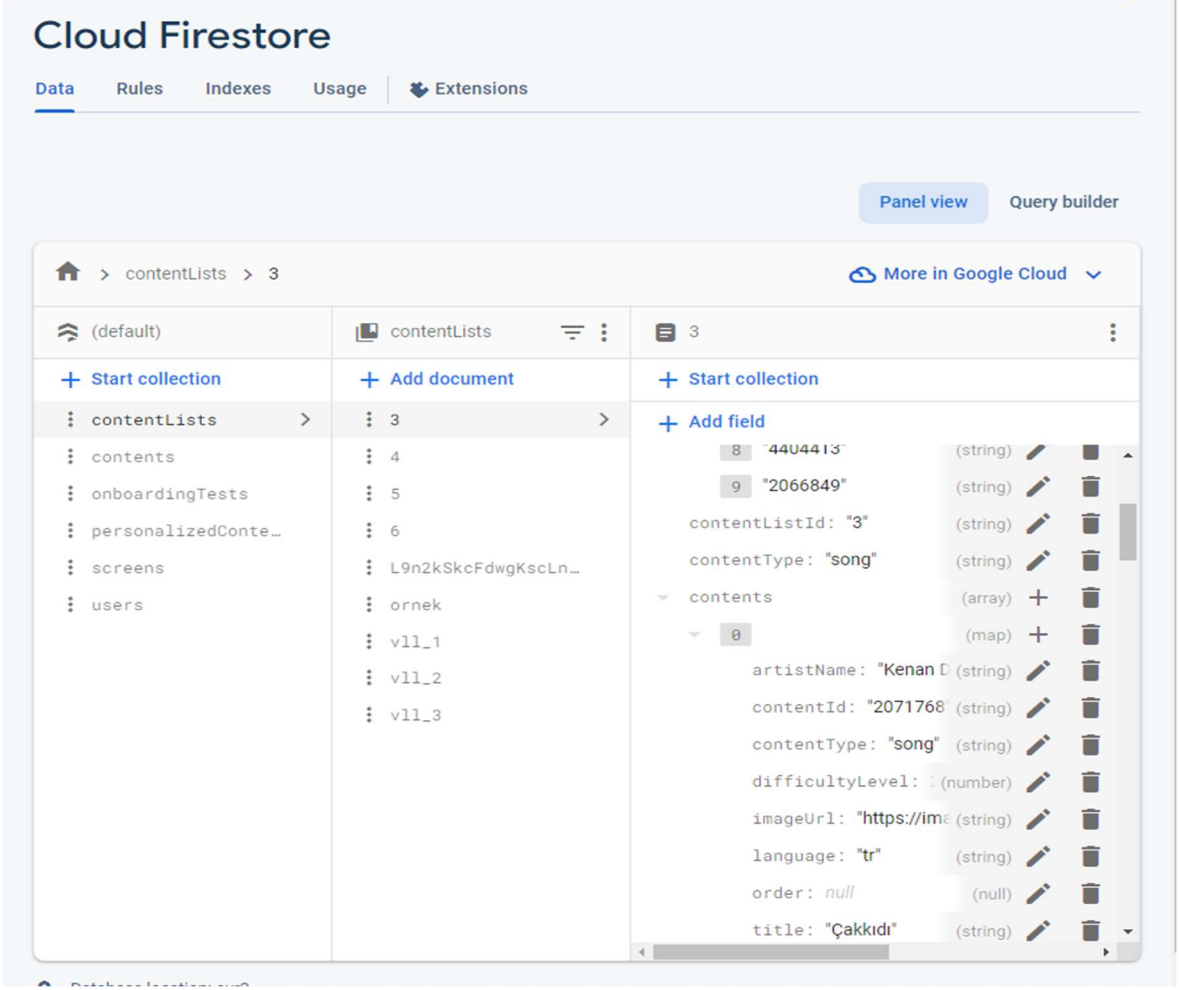
Ayrıca, Firestore'un güçlü sorgu yetenekleri sayesinde, uygulama içi aramalar ve veri filtrelemeleri etkin ve hızlı bir şekilde gerçekleştirilir.



Şekil 4.4 Firebase üzerinde seviye belirleme veri yapısı



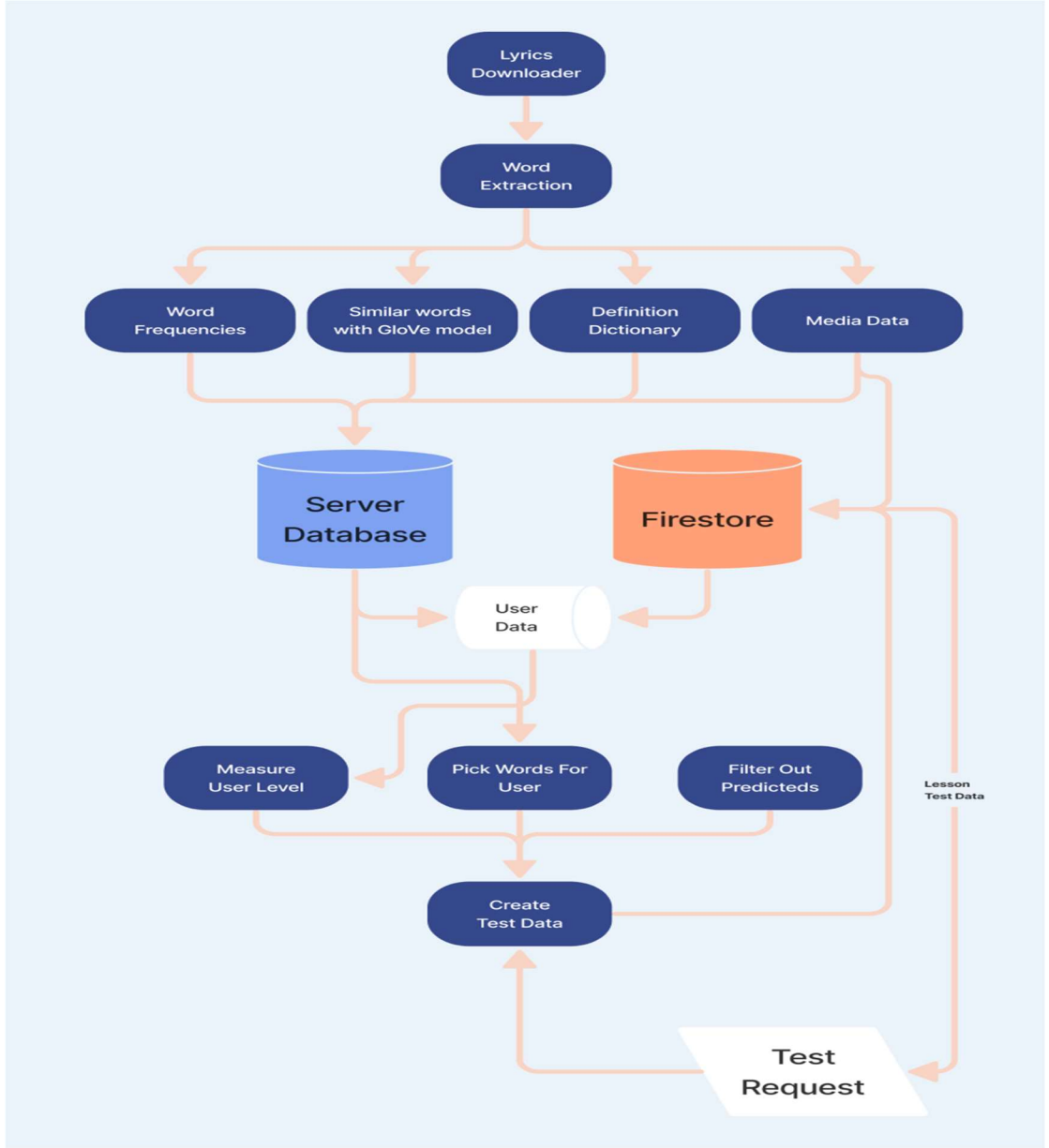
Şekil 4.5. Firebase Kullanıcı Modeli ve Seviye Belirleme Testi Yapısı



Şekil 4.6. Firebase Kullanıcı Veri Modeli

4.6 Mobil İstemci:

Mobil istemci, uygulamanın kullanıcı tarafı etkileşimlerini yönetir ve kullanıcı deneyimini en üst düzeye çıkarır. Flutter ile geliştirilen bu kısım, kullanıcıların uygulama içindeki şarkı listelerine erişimini, kelime testlerini ve diğer özelleştirilmiş öğrenme aktivitelerini kolaylaştırır. Mobil istemci ayrıca, kullanıcıya özel etkileşimli eğitim modüllerini ve öğrenme araçlarını sunarak, kelime öğrenimi sürecini daha verimli ve keyifli hale getirir. Uygulama içi testler ve quizler, kullanıcıların öğrendikleri kelimeleri pekiştirmelerine olanak tanırken, arayüz tasarımı kullanıcı dostu ve sezgisel bir deneyim sağlamak üzere özenle hazırlanmıştır. Her öge, uygulama içerisinde kullanıcının dil öğrenme yolculuğunu destekleyecek şekilde düşünülmüş ve entegre edilmiştir.



Şekil 4.7. Uygulama Veri Akış Şeması

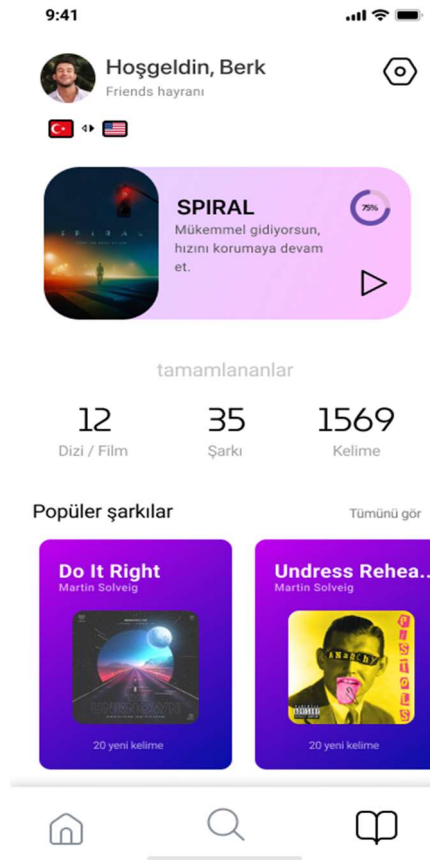
5. UYGULAMA

5.1. Mobil Uygulama Arayüzü

Mobil uygulama arayüzümüz, kullanıcı dostu, sezgisel ve erişilebilir olacak şekilde tasarlandı. Arayüz, kullanıcıların uygulamamızı kolayca navige etmeleri ve istedikleri işlemleri sorunsuz bir şekilde gerçekleştirebilmeleri için minimalist bir yaklaşımla oluşturuldu. Bu kısımda, mobil uygulama arayüzünün özelliklerine, tasarımına ve kullanıcı etkileşimine odaklandık.

Arayüz tasarımı, genel kullanıcı deneyimi ilkeleri göz önünde bulundurularak renk paleti, yazı tipi seçimi ve buton yerleşimleri optimize edildi. Ana sayfada, kullanıcıların çoğu zaman geçireceği bölümlere hızlı erişim sağlanacak şekilde en önemli özellikler ön plana çıkarıldı.

Uygulamanın temel özelliklerinden biri olan kelime öğrenme ve pratik yapma bölümleri, interaktif ve kullanıcıya geri bildirim sağlayacak şekilde tasarlandı. Kullanıcıların öğrenme süreçlerini takip edebilmeleri için ilerleme göstergeleri ve özelleştirilebilir hatırlatıcılar eklendi.



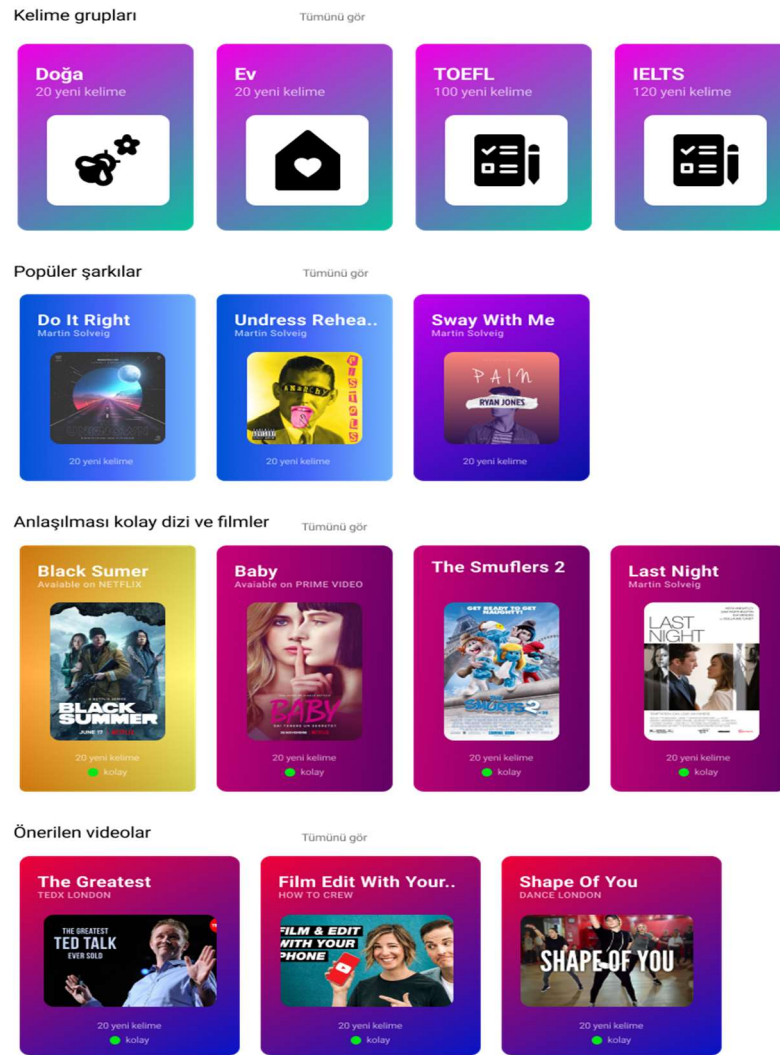
Şekil 5.1 Uygulama Anasayfası

5.2. Kelime Listesi Seçimi

Uygulamadaki kelime listesi seçimi, kullanıcıların dil öğrenme deneyimlerini kişiselleştirme ve kendi ihtiyaçlarına uygun hale getirme sürecidir. Bu aşama, kullanıcının seviyesini, ilgi alanlarını ve öğrenme hedeflerini temel alır. Uygulama, geniş bir medya

veritabanını kullanarak kullanıcının tercihlerine en uygun içerikleri önerir ve bu içeriklerden derlenen kelimelerle kişiselleştirilmiş listeler oluşturur.

Kullanıcılar, arzu ettikleri türde medya (dizi, film, müzik, YouTube videosu veya kitap) üzerinden ilerleyerek, öğrenmek istedikleri kelimeleri seçerler. Uygulama, seçilen kelimelerin sıklığını ve kullanım alanlarını analiz ederek, öğrenme sürecinin verimliliğini artırmak adına en önemli kelimeleri belirler. Bu süreçte, makine öğrenimi algoritmaları kullanıcıların daha önce maruz kaldıkları ve muhtemelen öğrendikleri kelimeleri filtreleyerek, öğrenme sürecini daha da kişiselleştirir.

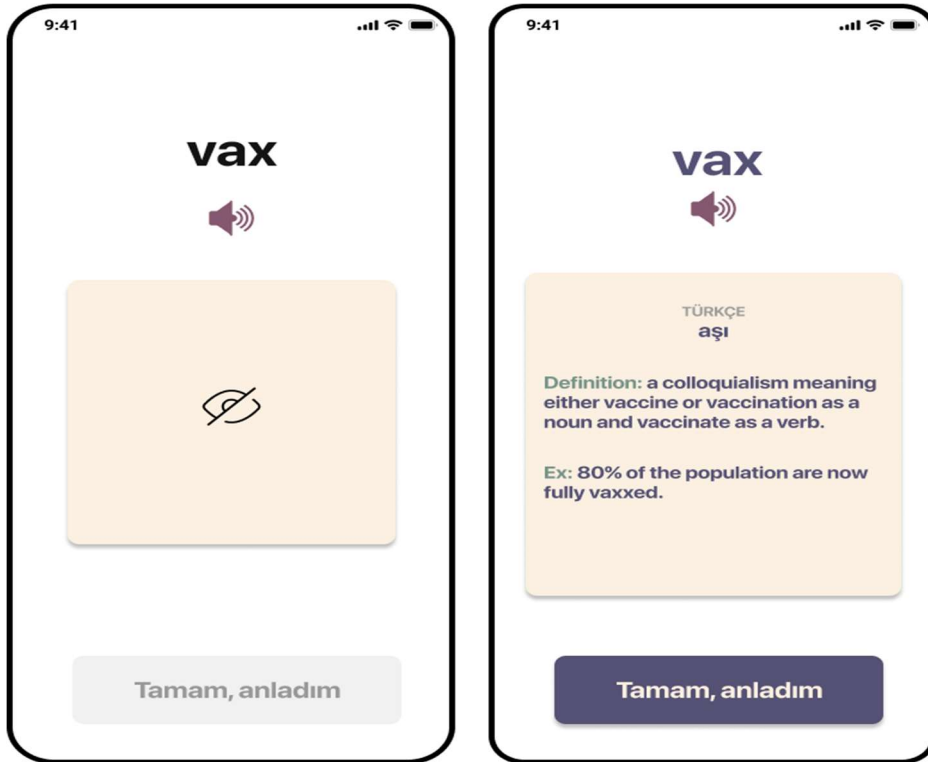


Şekil 5.2 Kelime Listeleri

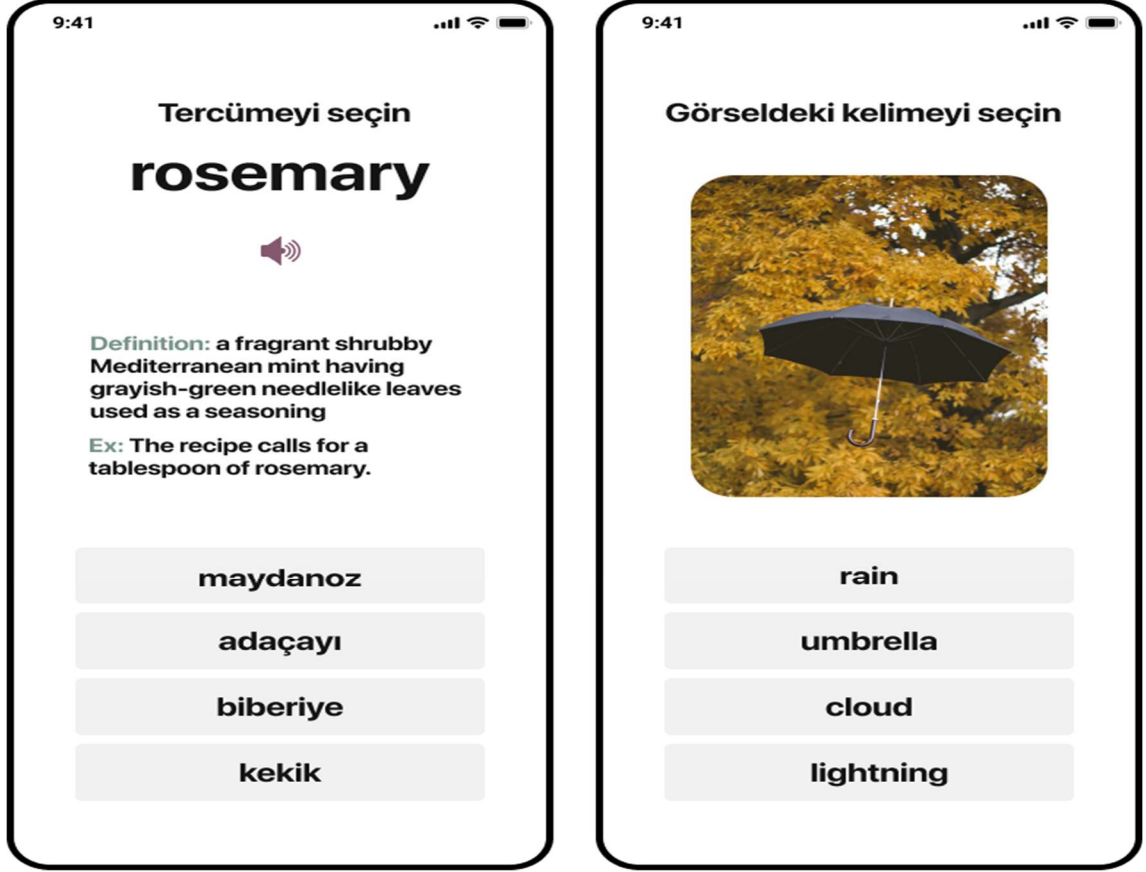
5.3 Kelimeleri Öğrenme Aşaması

"Kelimeleri Öğrenme Aşaması", uygulamanın interaktif öğrenme modüllerini ve tekniklerini içerir. Bu aşama, kullanıcının kelimeleri görsel ve işitsel materyallerle öğrenmesini, hafızasına almasını ve uzun vadede hatırlamasını amaçlar. Uygulama, birden fazla duyu organını kullanarak öğrenme sürecini güçlendirir ve recall yöntemleriyle bilginin hafızadan etkin bir şekilde geri çağrılmasını sağlar.

Kelimelerle ilgili çeşitli egzersizler, kullanıcının bilgiyi aktif bir şekilde kullanmasını ve pekiştirmesini teşvik eder. Klasik sınavlardan çoktan seçmeli testlere, resimli tahmin oyunlarından telaffuz ve çeviri pratiklerine kadar geniş bir yelpazede etkinlik sunar. Ayrıca, Spaced Repetition Systems (SRS) teknolojisi kullanarak, öğrenilen kelimelerin tekrarını optimize eder ve böylece kelimelerin hafızada kalıcılığını maksimize eder. Kullanıcıların geri bildirimleri ile sürekli güncellenen bu sistem, her kullanıcı için en etkili öğrenme yol haritasını oluşturur.



Şekil 5.3. Öğrenme Ekranları



Şekil 5.4. Öğrenme Ekranları 2

6. SONUÇ VE DEĞERLENDİRME

Projenin gerçekleştirilmesiyle, kelime öğrenimine yönelik olarak geliştirilen vocabulary uygulaması, dil ediniminde karşılaşılan zorlukların üstesinden gelmek için öğrencilere yenilikçi bir destek sunmuştur. Dil öğrenirken karşılaşılan en büyük engellerden biri olan geniş ve etkili bir kelime dağarcığına ulaşma süreci, uygulama vasıtasıyla kolaylaştırılmıştır. Öğrencilerin ihtiyaç duydukları ve kullanım sıklıklarına göre önem taşıyan kelimeleri etkin bir şekilde öğrenmeleri ve bu bilgiyi pratik kullanımlarla pekiştirmeleri hedeflenmiştir.

Uygulamanın tasarımı ve kullanımı, öğrencilerin dil seviyeleri ve öğrenme hedefleri doğrultusunda kişiselleştirilmiştir. Uygulama, her bir kullanıcıya uygun bir öğrenme yolu sunarak, kelime öğrenim sürecini daha verimli hale getirmek için kişisel ilerleme izleme ve uyarlanabilir öğrenme teknikleri sağlamıştır. Uygulamada yer alan etkileşimli alıştırmalar ve testler ile kelime dağarcığının pekiştirilmesi, hafıza teknikleri ile desteklenmiştir.

Yapılan uygulama testleri ve kullanıcı geri bildirimleri, uygulamanın etkinliğini ve kullanıcı memnuniyetini ortaya koymuştur. Öğrenciler, uygulamanın kapsamlı kelime

listeleri, gerçekçi kullanım örnekleri ve öğrenmeyi teşvik eden interaktif unsurları sayesinde, öğrenme sürecinde daha motive olduklarını ve kelime dağarcıklarının genişlediğini belirtmişlerdir.

Uygulamanın başarısı, yalnızca teknolojik yeniliklerin ve eğitim materyallerinin kalitesinin bir sonucu değildir; aynı zamanda, öğrencilerin bireysel ihtiyaçlarını anlama ve bu ihtiyaçlara uygun çözümler geliştirme anlayışının bir göstergesidir. Geleneksel ezber yöntemlerine kıyasla çok daha interaktif ve kişiye özgü bir öğrenme deneyimi sunarak, dil öğrenim süreçlerinde devrim niteliğinde bir adım atılmıştır.

Sonuç olarak, vocabulary uygulaması, kullanıcı dostu bir arayüz ve etkileşimli öğrenme materyalleri ile kelime öğrenim sürecini derinlemesine destekleyen, kullanıcıların dil becerilerini geliştirmelerine yardımcı olan ve özellikle kelime dağarcığı konusunda kapsamlı bir kaynak sunan bir proje olmasını hedefledik.

KAYNAKÇA

Nation, I.S.P., & Waring, R. (1997). Vocabulary size, text coverage and word lists. In N. Schmitt & M. McCarthy (Eds.), *Vocabulary: Description, acquisition and pedagogy* (pp. 6-19). Cambridge: Cambridge University Press.

Nation, P. (2006). How large a vocabulary is needed for reading and listening? *The Canadian Modern Language Review*, 63(1), 59-82.

Schmitt, N., & Schmitt, D. (2014). A reassessment of frequency and vocabulary size in L2 vocabulary teaching. *Language Teaching*, 47(4), 484-503.

Stoeckel, T., & Bennett, P. (2015). Refining the construct of classroom-based reading proficiency: Comparing the predictive validity of two measures on the TOEFL iBT test. *Language Testing*, 32(2), 173-190.

van Zeeland, H., & Schmitt, N. (2013). Lexical coverage in L1 and L2 listening comprehension: The same or different from reading comprehension? *Applied Linguistics*, 34(4), 457-479.

Webb, S. (2017). *The Routledge Handbook of Vocabulary Studies*. London: Routledge.

[Word2Vec vs GloVe - A Comparative Guide to Word Embedding Techniques \(analyticsindiamag.com\)](http://analyticsindiamag.com)

"An empirical study of semantic similarity in WordNet and Word2Vec," available at <https://www.uno.edu>.

"Overview of LSTMs and word2vec - and a bit about compositional distributional semantics if there's time" at <https://www.cam.ac.uk>.

Nation, I.S.P., & Bergsleithner, J.M. (2007). *Productive vocabulary tests and the search for concurrent validity*. *Studies in Second Language Acquisition*, 29(3), 445-461

Wiktionary. (n.d.). *Wiktionary frequency lists*. Retrieved from <https://www.wiktionary.org>

Rutgers University. (n.d.). *Rutgers University Center for Cognitive Science*. Retrieved from <https://www.rutgers.edu>

Masrai, Ahmed. (2019). Vocabulary and Reading Comprehension Revisited: Evidence for High-, Mid-, and Low-Frequency Vocabulary Knowledge. SAGE Open. 9. 215824401984518. 10.1177/2158244019845182.

[\(PDF\) Vocabulary and Reading Comprehension Revisited: Evidence for High-, Mid-, and Low-Frequency Vocabulary Knowledge \(researchgate.net\)](#)