

GTÜ BİL MUH BİL 496 İKİNCİ İZLEME KÖŞE YAZISI UYGULAMASI

Fatih Selim YAKAR

Proje Danışmanı: Doç. Dr. Habil KALKAN

Mayıs 2021

İÇERİK



PROJENİN ŞEMASI
VE TANIMI



YAPILANLAR



YAPILACAKLAR



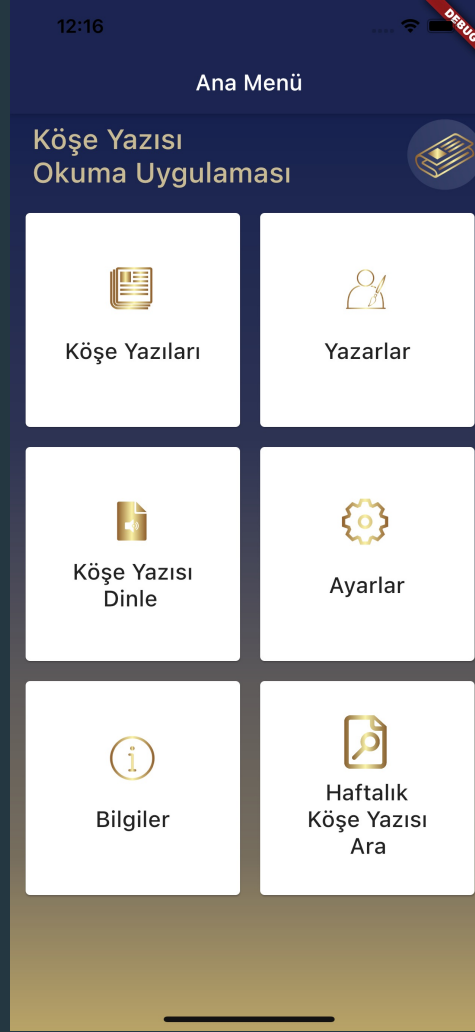
KAYNAKLAR

PROJE ŞEMASI VE TANIMI

Takip edilen
yazarların
haberlerini
okuma

Mevcut
haberleri
dinleme

Uygulamanın
kullanım
talimatlarını
görme



Takip edilen yazar
ekleme veya
çıkarma

Günlük güncelleme
özellikliğini açma veya
kapatma

Haftalık köşe yazısı
arama

Tanım: Bu projede kullanıcı takip ettiği yazarları seçerek onların güncel yazılarını okuyabilecek ve dinleyebilecek. Ayrıca kullanıcının ayarlar kısmından açabileceği özellik ile internet erişimi olduğu takdirde haberler günlük olarak güncellenecek. İnternet erişimi olmadığı durumda ise kullanıcı en son güncellediği zamana ait olan haberleri okuyup dinleyebilecek. Köşe yazısı aramak istediğinde ise bir haftalık süreçteki yazılar arasında arama yapabilecek.

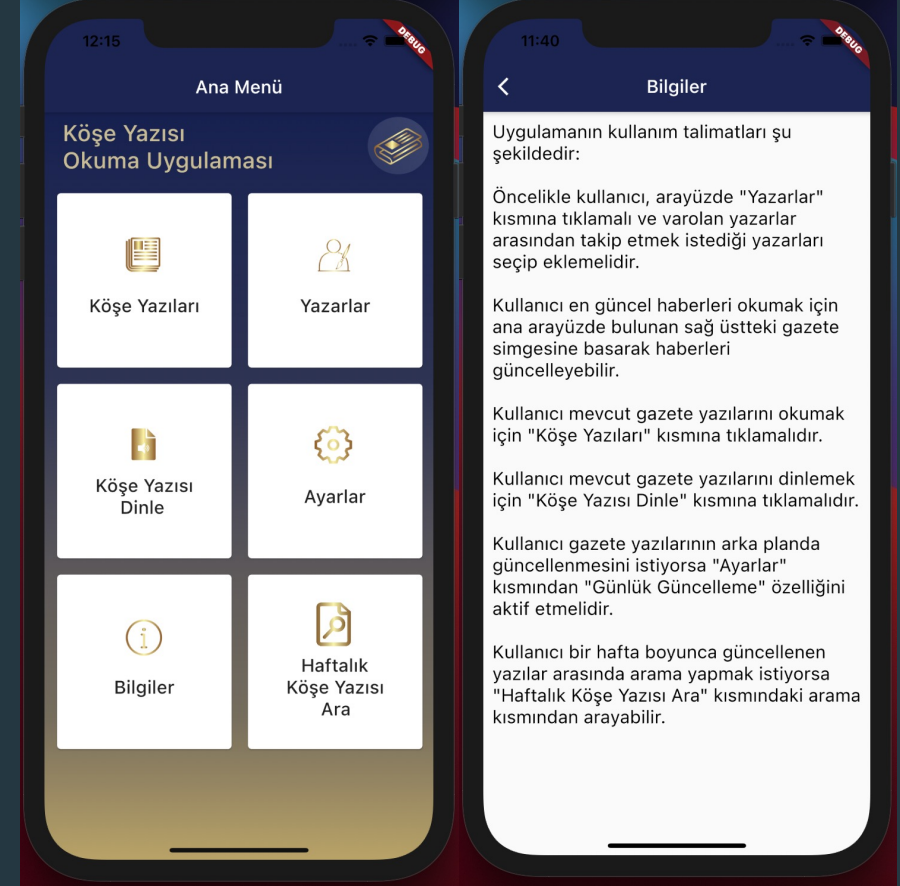
ANA EKTRAN VE BİLGİLER KISMINA "HAFTALIK KÖŞE YAZISI ARA" KISMI EKLENDİ

Android'in dahili araçlarından CardView, ImageView, TextView kullanılarak basılabilir alan yapıldı. Bilgiler kısmında sadece TextView veri tipine ekleme yapıldı.



Android Studio

Flutter'ın Card , Container ve GestureDetector veri tipleri ile basılabilir alan yapıldı. Bilgiler kısmında sadece Text veri tipine ekleme yapıldı.



Flutter

HAFTALIK KÖŞE YAZISI TUTABILMEK İÇİN VERİ TABANI YARDIMCISINA YENİ FONKSİYONLAR EKLENDİ.

SQLiteOpen
Helper
kullanılarak
veri tabanı
yardımcısı
güncellendi

```
public boolean insertNewsFilter(int newsId,String author, String title, String description,  
String url, String urlToImage, String publishedAt, String content){...}  
  
public List<NewsModel> getAllNewsFilter(){...}  
  
public NewsModel getNewsFilterModel(int newsId){...}  
  
public boolean updateNewsFilter(int newsId,String author, String title, String description,  
String url, String urlToImage, String publishedAt, String content){...}  
  
public int deleteNewsFilter(int newsId){...}  
  
public void deleteAllNewsFilter(){...}  
  
public boolean insertFilterDay(int day){...}  
  
public List<FilterDayModel> getAllFilterDays(){...}  
  
public FilterDayModel getSettingModel(int filterDay){...}  
  
public boolean updateFilterDays(int filterDay){...}  
  
public int deleteFilterDays(int filterDay){...}  
  
public void deleteAllFilterDays(){...}  
}
```

Android Studio

```
245 Future<void> insertFilterableNews(News news) async {...}  
259  
260 Future<List<News>> filterableNews() async {...}  
282  
283 Future<void> updateFilterableNews(News news) async {...}  
297  
298 Future<void> deleteFilterableNews(int id) async {...}  
311  
312 Future deleteAllFilterableNews() async {...}  
319  
320 Future<void> insertFilterDay(FilterDay filterDay) async {...}  
334  
335 Future<List<FilterDay>> filterDays() async {...}  
350  
351 Future<void> updateFilterDay(FilterDay filterDay) async {...}  
365  
366 Future<void> deleteFilterDay(int id) async {...}  
379
```

Flutter

sqflite paketi
kullanılarak
yapılan veri
tabanı
yardımcısına
yeni
fonksiyonlar
eklenerek
güncellendi.

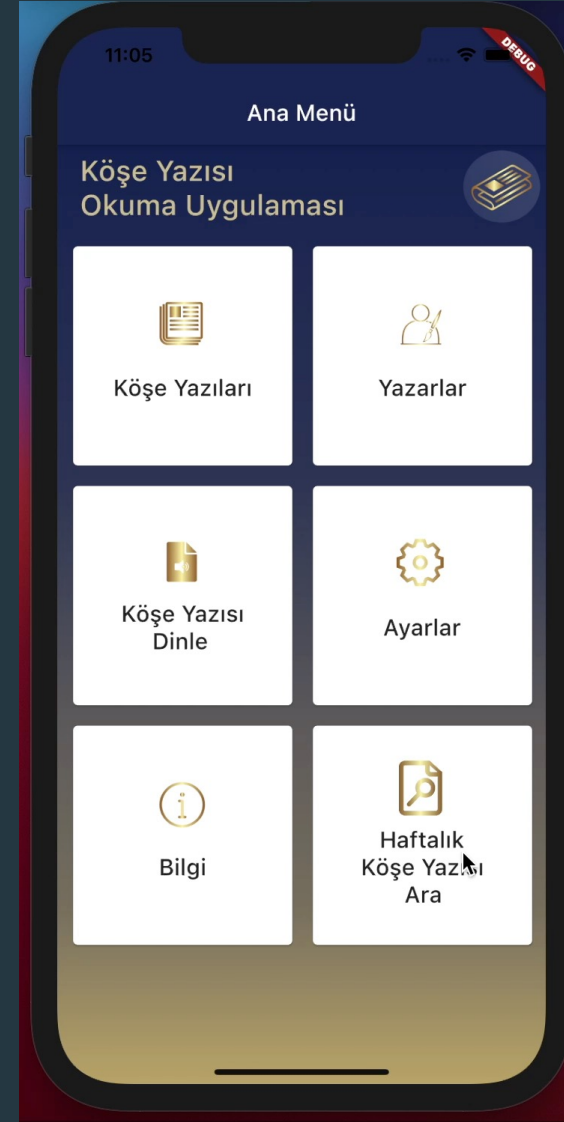
Bu değişiklikler sonucu veri tabanına haberleri tutabilmek ve kaç gün tuttuğunu belirlemek amacıyla FilterDays ve NewsFilter tabloları eklenmiş oldu.

HAFTALIK KÖŞE YAZISI ARAMA ARAYÜZÜ YAPILDI

Köşe yazıları kısmında RecyclerView, Imageview ve TextView kullanıldı. Arama çubuğu için SearchView kullanıldı.



Android Studio



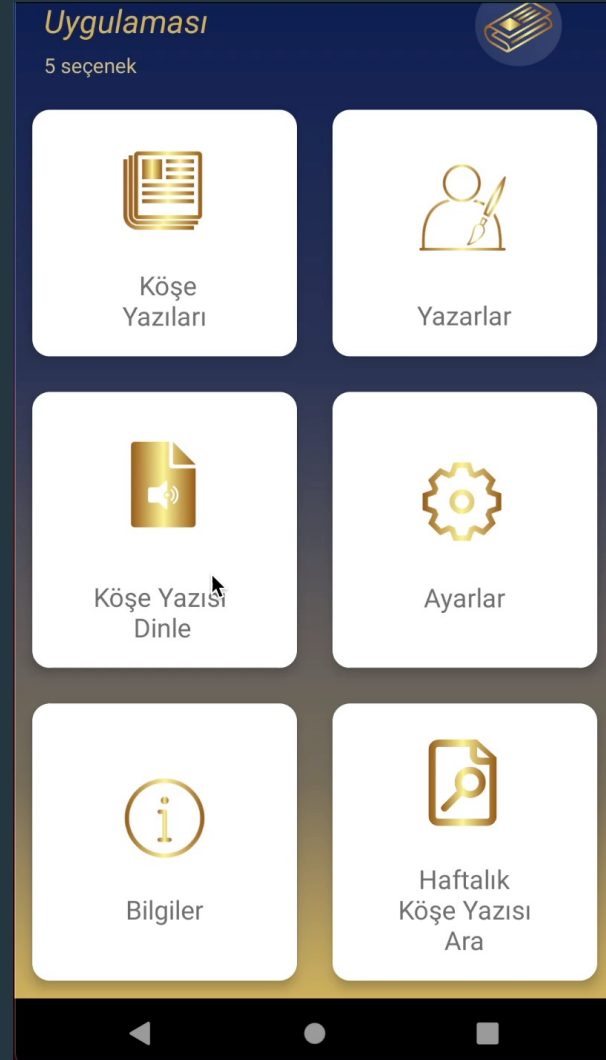
Flutter

Köşe yazıları kısmında Image, Card ve Column veri tipi kullanıldı. Arama çubuğu için TextField veri tipi kullanıldı.

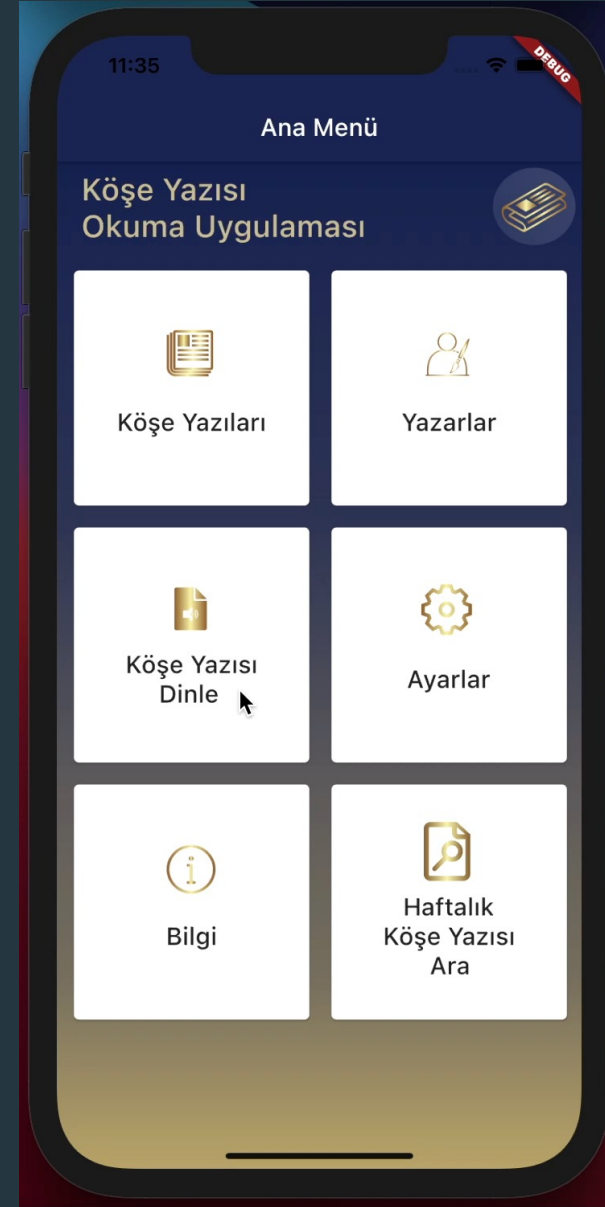
Arama; yazı içeriği, yayın kuruluşu ve yazı başlıklarına bakılarak gerçekleştiriliyor.

KÖŞE YAZISI DİNLEME ARAYÜZÜ YAPILDI.

RecyclerView, Buton ve TextView kullanıldı. RecyclerView'da kullanılmak amacıyla Holder ve Adapter sınıfları yazıldı.



Android Studio



Flutter

Flutter'daki Card, ElevatedButton, IconButton, Text, Padding, ListTile veri tipleri kullanılarak yapıldı.

ANDROID İÇİN ARKAPLAN ALARM SERVİSİ YAZILDI.

```
package com.example.newsdashboard.service;

import ...

public class MakeAlarmTask extends Service {

    // This method run only one time. At the first time of service created and running
    @Override
    public void onCreate() {
        HandlerThread thread = new HandlerThread( name: "ServiceStartArguments",
            Process.THREAD_PRIORITY_BACKGROUND);
        thread.start();
        Log.d( tag: "onCreate()", msg: "After service created");
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        //Here is the source of the TOASTS :D
        new RequestHelper( context: this, isService: true).updateNews();
        Toast.makeText( context: this, text: "New Service Task!", Toast.LENGTH_SHORT).show();

        return START_STICKY;
    }

    @Override
    public IBinder onBind(Intent intent) {
        // We don't provide binding
        return null;
    }
}
```

```
package com.example.newsdashboard.receiver;

import ...

public class AlarmReceiver extends BroadcastReceiver
{
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onReceive(Context context, Intent intent)
    {
        Intent in = new Intent(context, MakeAlarmTask.class);
        context.startService(in);
        setAlarm(context);
    }

    @RequiresApi(api = Build.VERSION_CODES.M)
    public void setAlarm(Context context)
    {
        int updateIntervalMinute=Integer.parseInt(new DatabaseHelper(context).getAllSettings().get(0).getSetting());
        AlarmManager am =( AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
        Intent i = new Intent(context, AlarmReceiver.class);
        PendingIntent pi = PendingIntent.getBroadcast(context, requestCode: 0, i, flags: 0);
        assert am != null;
        if(updateIntervalMinute!=0)
            am.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, triggerAtMillis: (System.currentTimeMillis())/1000L +
                (60L*updateIntervalMinute)) *1000L, pi); //Next alarm in 5s
        //Toast.makeText(context,"UPDATE INTERVAL SET:"+updateIntervalMinute,Toast.LENGTH_LONG).show();
    }
}
```



Günlük güncelleme özelliği açıldığında databasedeki güncelleme aralığı 1440 dakika olarak belirlenip alarm kuruluyor. Kurulan alarm haber güncelleme fonksiyonunu çalıştırıyor. Sonuç olarak databasedeki haberler güncelleniyor.

YAPILACAKLAR

Arayüz Altında Çalışacak İşlevlerin Yazılması (flutter servis, flutter ve android için yazıdan sese çeviri vb.)

Haber API' in Kontrol Edilmesi

Kullanıcı Testi Ve Gerçek Cihaz Testi Yapılması

Tüm Bu Sürecin Bir Rapora Dökülmesi

KAYNAKLAR

- Google, Android Studio Developer [Çevrimiçi], https://developer.android.com/studio/* [Ziyaret Tarihi: 22 Mart 2021].
- Stackoverflow [Çevrimiçi], https://stackoverflow.com/questions/* [Ziyaret Tarihi: 15 Mart 2021].
- Youtube, Create SQLite Database [Çevrimiçi], <https://www.youtube.com/watch?v=UdNn4S60l3k> [Ziyaret Tarihi: 10 Mart 2021].
- Youtube, Android XD Tutorial [Çevrimiçi], <https://www.youtube.com/watch?v=IVE5ETGEoA4> [Ziyaret Tarihi: 17 Mart 2021].
- Lucidchart [Çevrimiçi], <https://lucid.app/lucidchart> [Ziyaret Tarihi: 22 Nisan 2021].
- Geeksforgeeks [Çevrimiçi], https://www.geeksforgeeks.org/* [Ziyaret Tarihi: 7 Nisan 2021].
- Youtube, News Feed In Android Studio [Çevrimiçi], https://www.youtube.com/watch?v=9oNZAzlhL7s&list=PLT3-dzFEBix16zontJjPJPeYUWtE_HREq [Ziyaret Tarihi: 5 Nisan 2021].
- Looka, Free Logo Maker, https://looka.com/onboarding?company_name=kYOU [Ziyaret Tarihi: 15 Nisan 2021]
- Flaticon[Çevrimiçi], <https://www.flaticon.com/> [Ziyaret Tarihi: 18 Nisan 2021]
- Pub Dev[Çevrimiçi], <https://pub.dev/>, [Ziyaret Tarihi: 10 Nisan 2021]
- Flutter[Çevrimiçi], <https://flutter.dev/docs/development> , [Ziyaret Tarihi: 10 Nisan 2021]