# Üniversiteni Tanı

**Harran Üniversitesi**

**Bilgisayar Mühendisliği Bölümü**

**Mobil Uygulama Dersi**

**2022-2023 Güz Dönemi**

**Final Proje Ödevi**

# -İçindekiler-
# (Introduction)

Sayfa Numarası

# Rapor Hakkında

Bu rapor Harran Üniversitesi Bilgisayar Mühendisliği Bölümü 2022-2023 güz dönemi Mobil Uygulama dersi final ödevi için hazırlanmış olan proje ödevinin tanıtımını içermektedir.

Uygulama işlevleri, kullanılabilirlik özellikleri, sağladığı faydaların yanı sıra hazırlanmış olan bu raporda uygulamanın teknik özellikleri, tasarım detayları, kullanılan yazılım teknikleri, uygulma içeriği, teknik detaylar, örneklerle yazılan kodlar ve işlevleri anlatılmıştır.

Genel akış, içindekiler kısmında belirtilen kronolojiye bağlı olarak hazırlanmıştır. Burada birazda giriş seviyesinde uygulama tanıtımı uygun görüldüğünden raporun bu ilk kısmı kısa bir özet niteliğinde değerlendirilebilir.

Birazda uygulamamızdan kısaca bahsedecek olursak;

Proje adı kapakta da belirtilen ; ''Üniversiteni Tanı'' olarak uygulamayı oluşturan ekip tarafından tayin edilmiştir. Proje kapsamında tasarlanan giriş ekranında iki seçenek mevcuttur. Bunlardan biri giriş yapmak için, diğeri ise kayıt olmak için gerekli sayfalara yönlendirme yapacaktır. Bu aşama atlanıp giriş başarılı olursa kullanıcıyı anasayfa karşılayacaktır. Anasayfada tasarım detaylarının yanı sıra zengin bir menü karşılayıp içeriğinde üniversitenin bölümleri başlığı altında üniversitemizin Osmanbey yerleşkesinde mevcut olan bölümlerin, sosyal tesislerin ve idari binaların menüleri oluşturulmuştur. Bununla beraber üniversitemizin fotoğraflarının bulunduğu bir galeri uygulaması, ilgili (obs ve haruzem gibi) web sitelerine kısa yoldan ulaşmamızı sağlayan butonlar ve hesap makinesi bulunmaktadır. Bahsi geçen bu anahatların detaylı anlatım ve oluşturulma süreçleri, yararlanılan kaynaklar, teknik detayları ve benzeri ayrıntılara raporun ilerleyen kısımlarında ayrıntılı olarak değinilmiş ve açıklanmıştır. Başta da belirtildiği üzere bu metin bir özet niteliğindedir.

# Uygulama Hakkında

**1-1-) Nedir? Ne işe yarar?**

Oluşturulan uygulama üniversitemizin Osmanbey yerleşkesindeki bölümleri ile ilgili kolayca bilgi sahibi olabileceğimiz ilgili web sitelerine kolayca erişebileceğimiz bağlantılar sağlamanın yanı sıra üniversitemizin içinde yer alan bölümlere kolayca gidebileceğimiz bir harita uygulaması sağlamaktadır. Ayrıca bu harita uygulamasında bazı teknik detaylar düşünülerek kullanıcı dostu hale dönüştürülmüştür. Kullanıcı dostu kelimesi kullanım kolaylığı, uygulamayı hızlı kavrama, her hangi bir eğitim gerektirmeme gibi konularda özenli davranan ekibimizin özellikle dikkat ettiği bir kavram olmuş ve gerek tasarımın oluşturulması sürecinde gerekse de uygulama yazılımının gerçeklenmesinde en hassasiyetle davrandığı adeta kılı kırk yararcasına titizlikle ilgilendiği konu olmuştur. Ekip olarak şunun farkına vardık ki bir uygulamanın oluşturulmasından sonra başarılı olmasındaki en etkili faktörlerden biri kullanıcı memnuniyetidir. Bu gerekçeyle her aşamada testler yapılmış ve kullanım kolaylığı özellikle dikkate alınmıştır.

Bu uygulama sayesinde oldukça dağınık ve parça parça bulunan üniversitemizin sınav takvimleri, ders izlenceleri, yaşayan kampüsümüzün bölümleri ve öğrencileriyle ilgili bilgileri tutan önemli web siteleri bir araya getirilerek kolaylıkla erişilebilir olmuştur. Şu unutulmamalıdır ki bilgisayar teknolojilerinin ve yazılım geliştirmenin en önemli amaçlarından biri insan hayatını kolaylaştırmaktır. Ve bu doğrultuda atılabilecek en önemli adımlar zaman kaybının önüne geçmesi yöneliminde olmalıdır. Kullanım kolaylığı bu aşamada devreye girmektedir. Peki kullanım kolaylığı nasıl bir zaman tasarrufuna sebep olabilir? Tabiiki başta kulağa garip gesle de bu husutaki dikkat çekilecek en önemli konu budur zira kullanım kolaylığı kullanıcının bir bilgi kaynağını aramakla zaman kaybetmemesini sağlamakla beraber aranan şey bulunmaya çalışırken yormaz ve hızlıca erişilebilir. Bu nedenle kullanıcı dostu bir tasarım sağlamak zaman tasarrufuna sebeptir.

Bir başka deyişle var olan bir sorun inovatif bir yaklaşımla ve sistematik bir çalışma sayesinde çözüme kavuşturulmuştur. Böyle bir uygulamanın gerekliliği ise aşikardı.

### 1-2-) Sağladığı Faydalar

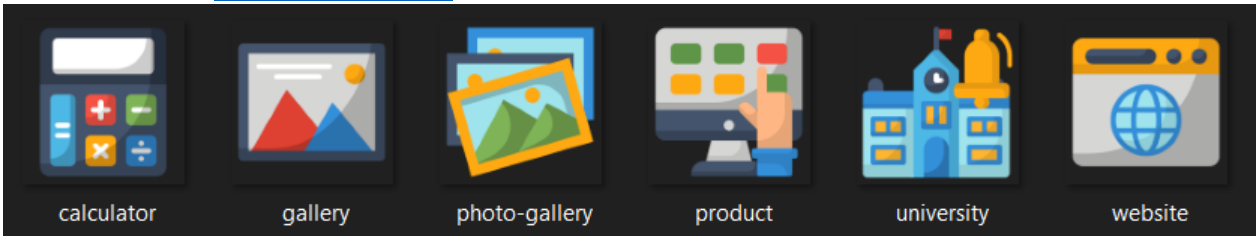
Bir önceki paragrafta ne olduğu ve ne işe yaradığı hakkında konuşuldu. Aslında gereklilikleri anlatılırken bu paragrafın konusu olan sağladığı faydalara da değinilmiş oldu. Genel hatlarıyla uygulamamız dağınık ve erişilmesi zahmetli olan üziversitemizin bilgi kaynaklarını yormayan, sade, zarif tasarım ve kolay kayıt ve kullanım olanaklarıyla hizmete sunmuştur. Ortada olan soruna inovatif çözüm üretildi ve kolaylık öncelendi. Bunlara ek olarak uygulama bünyesinde bulunan hesap makinesi, harita ve web yönlendirmeleri ile içerik zenginleştirildi. Ayrıca galeri uygulamasında veri tabanına aktarılan seçkin fotoğraflar ile özellikle üniversitemizi yeni tercih edecek ve tanımak isteyen öğrencilere üniversitemizin nasıl bir yer olduğu hakkında bilgi sahibi olmalarını sağladık. Geniş ve kapsamlı kampüsümüz fotoğraflarla tanıtıldı. Kısaca uygulamamız bir bütünlük ve erişilebilirlik sağladı.

# Genel Tasarım Ve Uygulanması Hakkında

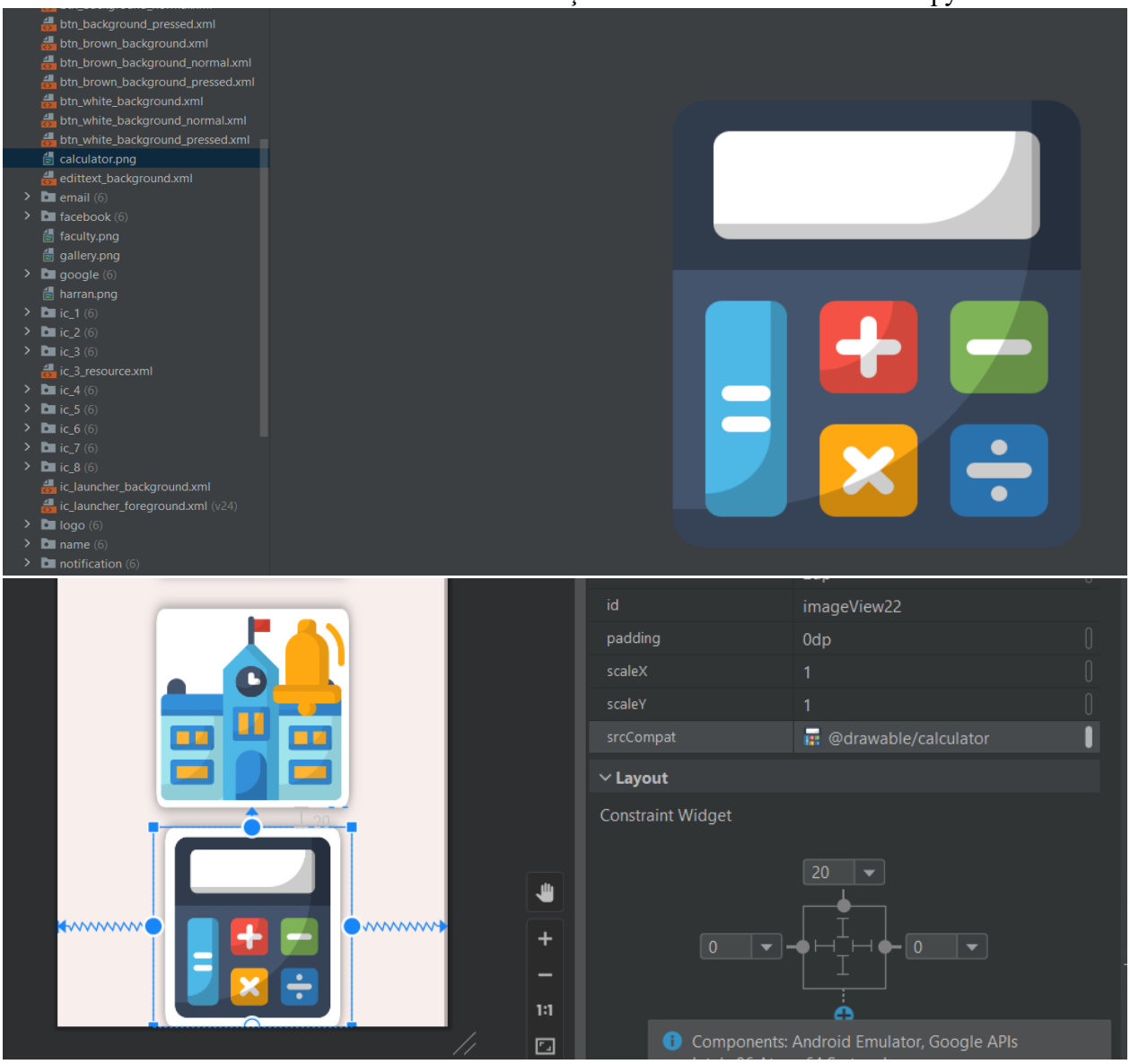
**TASARIM**

### İkonlar

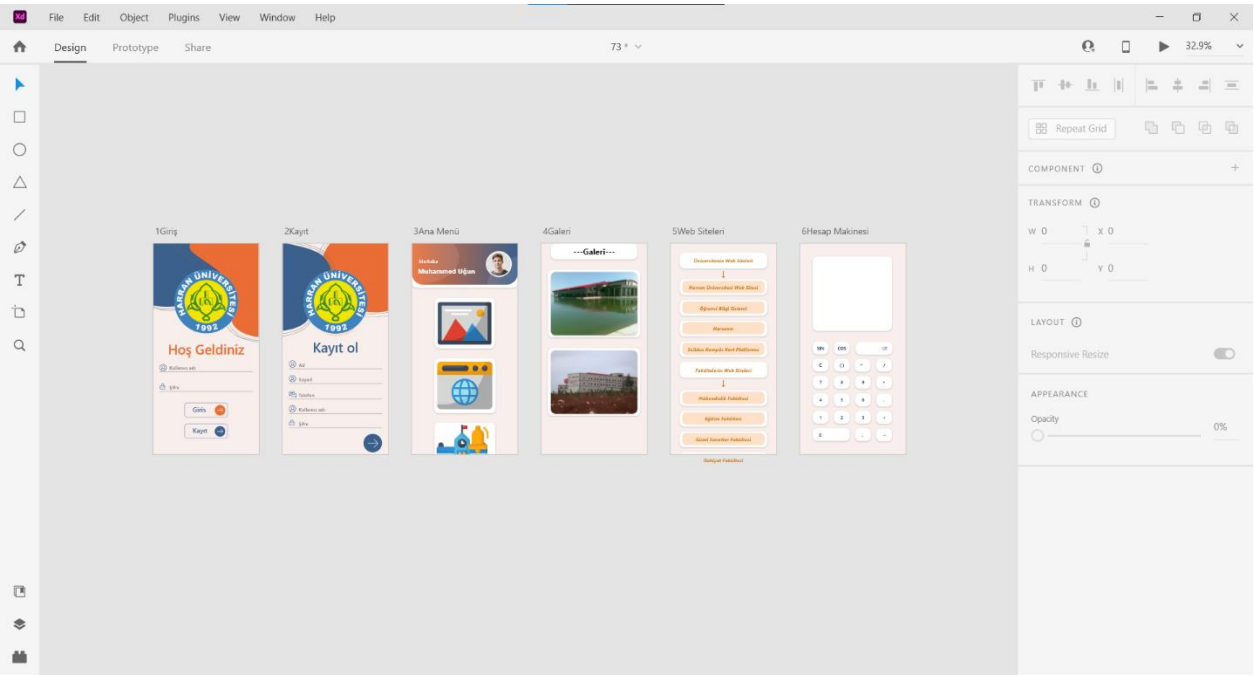
- Gerekli ikonları www.flaticon.com sitesinden bulduk.

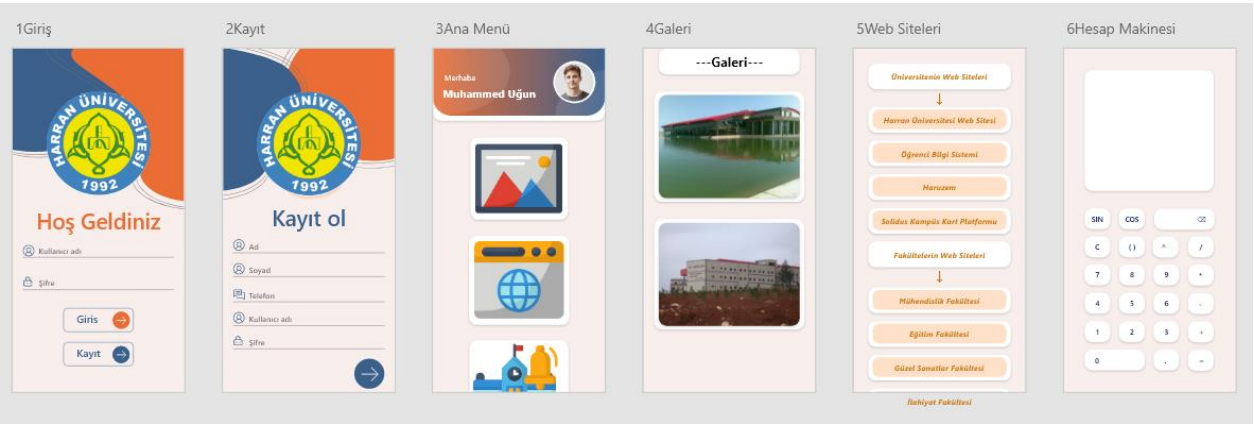


- İkonları Android Studio'da kullanmak için drawable kısmına kopyaladık.

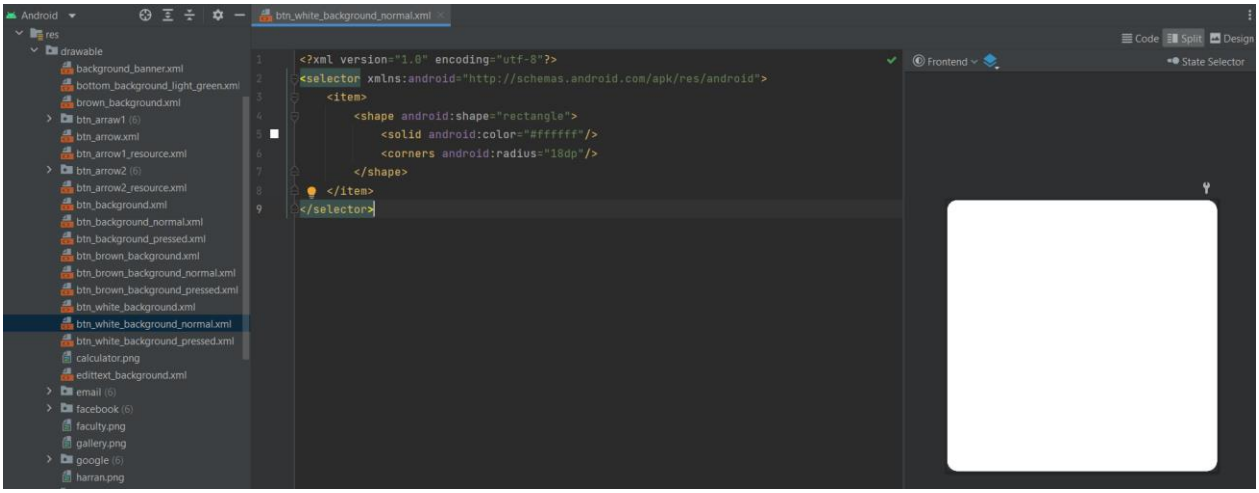


- Activity tasarımları Adobe XD Programı kullanılarak tasarlandı.




- Tasarımı uygulamak için gerekli olan birçok şekli Android Studio'da drawable kısmından xml kodlarıyla oluşturduk.

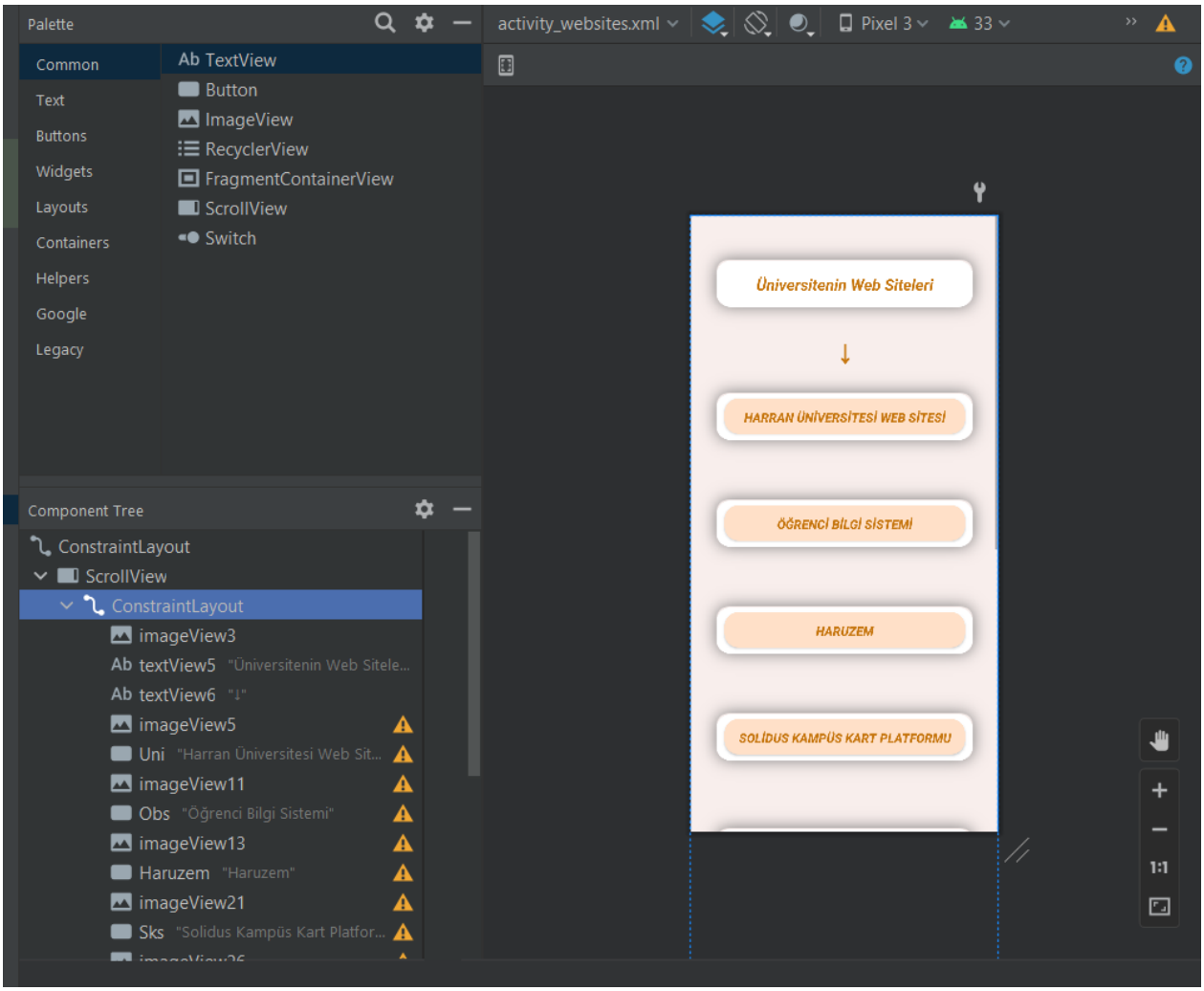



- Butonlara tıklarken tıklama hissi vermesi için bir şekilden diğer şekle geçme yöntemini kullandım ve bunu uygulamak için aşağıdakine benzer kod stilini kullandık.
-

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true"
android:drawable="@drawable/btn_background_pressed"/>
    <item android:state_pressed="false"
android:drawable="@drawable/btn_background_normal"/>
</selector>
```

- Örneğin burada butona tıklandığı zaman “btn_background_pressed” şekli aktif edilecek, basılmadığı zaman ise “btn_background_normal” şekli aktif olacak.

- Tasarımları uygularken ConstraintLayout kullanarak objeler arasında ilişkisel bir hizalama yaptık.



# Giriş Sayfası

Giriş sayfası uygulama çalıştırıldığında kullanıcıyı ilk karşılayan sayfa olduğundan hem tasarım hemde işlevsellik olarak zamanımızın ve emeğimizin önemli bir kısmını harcadığımız ve üzerinde titizlikle durduğumuz kısım oldu. Sade, göze hitap eden ve erişilebilir dizaynda iki adet editText ve yine iki adet button yapısı kullanıldı.

### 3-1-) Firebase Hakkında

Firebase, Google tarafından geliştirilen ve sunulan bir mobil ve web uygulamaları geliştirme platformudur. Firebase, uygulamamız için gerekli olan veri depolama, veri yönetimi, güvenlik, giriş ve yetkilendirme, analitik ve raporlama özelliklerini bir arada sunar. Ayrıca, Firebase sayesinde diğer özelleştirilmiş özellikler eklenebilir. Firebase, mobil ve web uygulamaları için popüler bir platformdur ve genellikle Backend-as-a-Services(Baas) olarak adlandırılır.

### 3-2-) Giriş Ekranı Tasarımı Ve Butonların İşlevi

Giriş sayfasının dizaynından bahsedecek olursak; Ekranın üst kısmında soft renklere işlenmiş orta boyut üniversitemizin logosu konumlandırıldı. Logonun hemen altında tasarıma uygun olarak seçilmiş turuncu renkte ‘’Hoş Geldiniz’’ yazısı kullanıcıyı karşılıyor. Hemen yazının altında ekranın sonuna kadar tıklama yapılabilmesine imkan sağlayan editText’ler bu özellik dikkate alınarak boyutlandırıldı. Bahsi geçen editText’lerin altında ise iki elle kullanıma da olanak sağlaması sebebiyle ekranın ortasına alt alta konumlandırılmış iki adet buton mevcut. İlk editText kullanıcı adı girişi, ikincisi ise şifre girişi olacak şekilde atandı. Üst taraftaki buton giriş ve alt taraftaki buton ise kayıt olma seçeneğini sağlayacak şekilde programlandı.


### 3-2-) Kayıt Ekranı Tasarımı Ve Butonların İşlevi

Kayıt ekranının genel tasarımında giriş ekranında karar kılınan tasarıma sadık kalınmıştır. Ekranın kullanıcı kaydı için gerekli bilgilerin alındığı bölüme kadar olan üst kısım tamaıyla giriş ekranıyla benzerdir. Kullanıcı bilgileri için beş adet editText kullanılmıştır. Bunlar sırasıyla ad, soyad, telefon, kullanıcı adı ve şifre bilgilerini kayıt olmak isteyen kullanıcıdan beklemektedir. Bu alanların her hangi birinin boş geçilmemesi kodlarla kısıtlanmıştır. Bahsi geçen editText'lerin sağ altında konumlandırılmış oval mavi buton kayıt işleminin tamamlanmasını ve veri tabanıyla bağlantı kurarak kullnaıcı bilgilerinin veri tabanına kaydedilmesini sağlar.


# Anasayfa Dizaynı

# İçerik Uygulamaları

## 5-1-) Harita Uygulaması

Map javalarında onMapReady ile haritanın ilk nerde başlayacağını belirleriz .mMap=googleMap;

```
@Override
public void onMapReady(@NonNull GoogleMap googleMap) { // Haritanın başlangıç konumundan başlatır
    Toast.makeText( context: this,  text: "Harita Hazır", Toast.LENGTH_SHORT).show();
    Log.d(TAG,  msg: "onMapReady:harita hazır");
    mMap = googleMap;

    if (mLocationPermissionsGranted) {...}

}
```

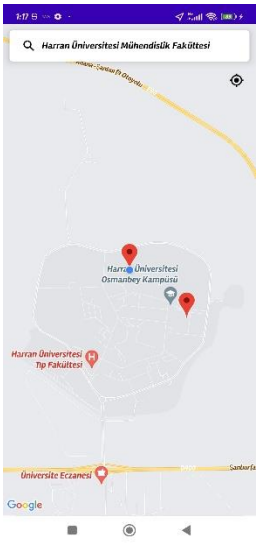
Harita uygulamasını empty activity ile oluşturulmuş olan projede uygulamaya ilk girdiğinizde HaritaActivity. Java ile oluşturulmuş sayfanın tasarımıyla karşılaşacksınız. Butonlar ve kodları;


```
btMuh.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent=new Intent( packageContext: HaritaActivity.this, MapMuhendis.class);
                startActivity(intent);

    }
});
btTip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent=new Intent( packageContext: HaritaActivity.this, MapTip.class);
        startActivity(intent);

    }
});
btFenE.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent=new Intent( packageContext: HaritaActivity.this, MapFenEdb.class);
        startActivity(intent);

    }
});
btIlh.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent=new Intent( packageContext: HaritaActivity.this, MapIlh.class);
        startActivity(intent);

    }
});
```

Butonlara basılmadan önce harita initMap() ile yüklenir.

```
private void initMap(){// Haritanın yüklenmesi
    Log.d(TAG, msg: "initMap:harita başlatılıyor");
    SupportMapFragment mapFragment=(SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);

    mapFragment.getMapAsync( callback: MapBes.this);
}
```

Mühendislik Fakültesi butonuna basıldığında ise Mühendislik Fakültesinin haritadaki konumu geoLocate() methodu ile gerçekleştirilir.




```
public void geoLocate(){// aranılan yere yönlendirme
    Log.d(TAG, msg: "geoLocate: geoLocate");
    String searchString= mSearchText.getText().toString();//aratılacak yer

    Geocoder geocoder = new Geocoder( context: MapBes.this);
    List<Address> list=new ArrayList<>();
    try {//hata kodu
        list=geocoder.getFromLocationName(searchString, maxResults: 1);
    }catch (IOException e){
        Log.e(TAG, msg: "geoLocate: IOExpection: " + e.getMessage());
    }
    if (list.size()>0){
        Address address = list.get(0);
        Log.d(TAG, msg: "geoLocate: bir yer buldu:"+ address.toString());//konum üstü bilgiler

        moveCamera(new LatLng(address.getLatitude(),address.getLongitude()),DEFAULT_ZOOM,
                address.getAddressLine( index: 0));
    }
}
```

Aynı zamanda konumun üstüne tıklandığında konum bilgileri verilir. geoLocate() methodu içinde tanımlanmıştır.


```
if (list.size()>0){
    Address address = list.get(0);
    Log.d(TAG, msg: "geoLocate: bir yer buldu:"+ address.toString());//konum üstü bilgiler

    moveCamera(new LatLng(address.getLatitude(),address.getLongitude()),DEFAULT_ZOOM,
            address.getAddressLine( index: 0));
}
```

İşaretli olan butona tıklandığında bizi Google haritalara gönderir ve rota çizer.

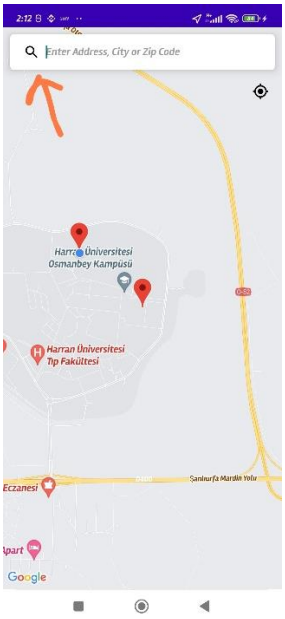


Bulunduğumu lokasyona geri dönmeyi getDeviceLocation () methodu ile yaparız.

```
Location currentLocation=(Location) task.getResult();
```

```
private void getDeviceLocation(){// lokasyona geri döndür
    Log.d(TAG, msg: "getDeviceLocation: cihazların mevcut konumunu alma");
    mFusedLocationProviderClient= LocationServices.getFusedLocationProviderClient( activity: this);
    try {
        if (mLocationPermissionsGranted){
            Task location= mFusedLocationProviderClient.getLastLocation();
            location.addOnCompleteListener(new OnCompleteListener() {
                @Override
                public void onComplete(@NonNull Task task) {
                    if (task.isSuccessful()){
                        Log.d(TAG, msg: "onComplate:Bulunan yer");
                        Location currentLocation=(Location) task.getResult();

                        moveCamera(new LatLng(currentLocation.getLatitude(),currentLocation.getLongitude()),
                                DEFAULT_ZOOM, title: "Benim konumum" );
                    }else{
                        Log.d(TAG, msg: "onComplate: geçerli konum konumu boş");
                            Toast.makeText( context: MapBes.this,  text: "mevcut konum alınamıyor", Toast.LENGTH_SHORT).show();
                    }

                }
            });
        }

    }catch (SecurityException e){
        Log.e(TAG, msg: "getDeviceLocation:Güvenlik İstisnası: "+e.getMessage() );

    }
}
```

Eğer kullanıcı kendi gezinmek isterse search arama butonu bulunmaktadır. Map java'daki clcgo() methodu ile bu buton aktif edilmiştir.




Not: MapBes, MapFenEdb, MapIIBF, MapIlh, MapKtp, MapMuhendis, MapRektr, MapTip, MapYM java sınıfları her bir buton için oluşturulmuştur kodlar birbirinin aynısı olup sadece resimde gösterilen kısımdaki yere konum bilgileri farklı olarak aktarılmaktadır.

```
mSearchText.setText("Gülveren, Mehmet Arabacı Beden Eğitmi ve Spor Yüksekokulu, 63290 Şanlıurfa Merkez/Şanlıurfa");
```

### 5-2-) Web Sitelerine Yönlendirme

Butona basıldığı zaman ilgili siteye yönlendirebilmesi için aşağıdaki kod parçasını kullandık. android.net.Uri kütüphanesinden gelen Uri sınıfını sayesinde link yönlendirme işlemlerini yapabildik.

```
UniWeb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Uri linkimiz = Uri.parse("http://www.harran.edu.tr/"); //butona vermek istediğimiz linki buraya yazıyoruz.
        Intent intentimiz = new Intent(Intent.ACTION_VIEW, linkimiz);
        startActivity(intentimiz);
    }
});
```

### 5-4-) Hesap Makinesi

Bu bölümde de genel uygulamamızın tasarım çizgisine sabit kalınarak bir tasarım uygulanmıştır. Genel formatta bilinen basit işlemleri yapan ve bunun yanı sıra sinüs ve cosinüs gibi trigonometrik fonksiyonların sayısal değerlerini hesaplayabilen ve üs alıp parantez işlemleri yapabilen detaylı düşünülmüş ve ince tasarlanmış bir hesap makinesi oluşturduk.

### 5-4-1-) Yapısal Olarak Hesap Makinesi

Tasarım gereği değiştirilen Themes.xml ve color xml dışında yapısal olarak java kodlarına etki eden strings.xml dosyasına şu komut eklenerek başlandı;

```
<string name="tap_here">0</string>
```

Bundan sonra yapılan değişiklikler direkt olarak Calculator.java dosyasında gerçekleşmiştir.

Sırasıyla kodlar açıklanacaktır. İlk olarak gerekli kütüphaneler import edildi.

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.mariuszgromada.math.mxparser.*;
```

Burada dikkat edilmesi gereken nokta; MathParser kütüphanesinin import edilmesidir. Bu kütüphane konsola yazılan şu komutun çalıştırılmasının ardından import edilebilir.

Install-Package MathParser.org-mXparser -Version 5.1.0

Sonrasında build.gradle dosyasına şu komut eklendi;

implementation 'org.mariuszgromada.math:MathParser.org-mXparser:5.1.0'

Sonrasında ekrandaki çıktı düzenlendi

```
EditText display;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_calculator);

    display = findViewById(R.id.display);
    display.setShowSoftInputOnFocus(false);


    display.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (getString(R.string.tap_here).equals(display.getText().toString())) {

                display.setText("");
            }

        }
    });

}
```

Ardından switch-case yapısı kullanılarak butonlara basıldığında updateDisplay adlı fonksiyona karakter ataması yapılır.

```
case R.id.btn_nine:
    updateDispaly( s: "9");
    break;
case R.id.btn_dot:
    updateDispaly( s: ".");
    break;
```

Aşağıdaki switch-case lerdeki işlemde farklı olarak başka fonksiyonlar çağrılmıştır.

```
case R.id.btn_C:
    display.setText("");
    break;
case R.id.btn_brackets:
    addBracets();
    break;
```

Aşağıdaki kod parçasında cursor pozisyonunun sayılar arasındaki konumu ayarlanmıştır.

```
private void reverse() {

    int cursor_pos = display.getSelectionStart();

    if (cursor_pos > 0) {
        String oldDisplay = display.getText().toString();
        String leftSideOnDisplay = oldDisplay.substring(0, cursor_pos - 1);
        String rightSightOnDisplay = oldDisplay.substring(cursor_pos);
        String newText = leftSideOnDisplay + rightSightOnDisplay;
        display.setText(newText);
        display.setSelection(cursor_pos - 1);
    }
}
```

Result fonksiyonu ile MathParser kütüphanesi yardımıyla sonuç hesaplama işlemi yapılmıştır.

```
private void result() {

    String textDisplay = display.getText().toString();
    String reTextDisplay = textDisplay.replaceAll( regex: "/", replacement: "/");
    reTextDisplay = reTextDisplay.replaceAll( regex: "X", replacement: "*");
    Expression exp = new Expression(reTextDisplay);
    String result = String.valueOf(exp.calculate()).toString();

    if (!result.equals("NaN")) {
        display.setText(result);
        display.setSelection(result.length());//cursoru en saga atar...
    } else {
        showToast( text: "Wrong value!!!");
    }
}
```

Update display metodu ekrana her girdi girildiğinde işlem yapmasını ve hesaplamada sonuç döndürmesini sağlar.

```
private void updateDispaly(String s) {
    int cursor_pos = display.getSelectionStart();

    if (getString(R.string.tap_here).equals(display.getText().toString())) {

        display.setText(s);
    } else {
        String oldDisplay = display.getText().toString();
        String leftSideOnDisplay = oldDisplay.substring(0, cursor_pos);
        String rightSightOnDisplay = oldDisplay.substring(cursor_pos);
        String newText = leftSideOnDisplay + s + rightSightOnDisplay;
        display.setText(newText);
    }
    display.setSelection(cursor_pos + 1);
}
```

Burada hesaplamanın doğruluğu ve mantıksal okumaların doğruluğu için en önemli husus parantez eklemedir oda şekildeki kod parçacığı ile sağlanmıştır.

```
private void addBracets() {

    String textDisplay = display.getText().toString();
    //  int cursor_pos = display.getSelectionStart();
    int countBracets = 0;

    for (int i = 0; i < textDisplay.length(); i++) {
        if (textDisplay.substring(i, i + 1).equalsIgnoreCase( anotherString: "(")) countBracets++;
        if (textDisplay.substring(i, i + 1).equalsIgnoreCase( anotherString: ")")) countBracets--;
    }
    String lastCharOfTextDisplay = textDisplay.substring(textDisplay.length() - 1);
    if (countBracets == 0 || lastCharOfTextDisplay.equals("(")) updateDispaly( s: "(");
    else if (countBracets > 0 && !lastCharOfTextDisplay.equals(")")) updateDispaly( s: ")");

}
```

Bu hesap makinesinde son olarak trigonometrik hesaplamalar da yapması aşağıdaki metot sayesindedir.

```
public void clcSin(View v) {
    display.setText("sin(" + display.getText().toString() + ")");
    result();
}


public void clcCos(View v) {
    display.setText("cos(" + display.getText().toString() + ")");
    result();
}
```