

Üniversiteni Tanı

**Harran Üniversitesi
Bilgisayar Mühendisliği Bölümü
Mobil Uygulama Dersi**

**2022-2023 Güz Dönemi
Final Proje Ödevi**

-İçindekiler-

(Introduction)

	Sayfa Numarası
1-) Uygulama Hakkında.....	4
1-1-) Nedir? Ne İşe Yarar?.....	4
1-2-) Sağladığı Faydalar.....	5
2-) Genel Tasarım Ve Uygulanması Hakkında.....	5
3-) Giriş Ekranı.....	10
3-1-) FireBase Hakkında.....	10
3-2-) Giriş Ekranı Tasarımı Ve Butonların İşlevi.....	10
3-3-) Kayıt Ekranı Tasarımı Ve Butonların İşlevi.....	11
3-3-) Giriş Yapma.....	13
3-4-) Sisteme Kayıt.....	15
4-) Anasayfa Dizaynı.....	17
5-) İçerik Uygulamaları.....	18
5-1-) Harita Uygulaması.....	18
5-2-) Web Sitelerine Yönlendirme.....	21
5-3-) Galeri.....	22
5-4-) Hesap Makinesi.....	24
6-) Proje Ekibinin Tanıtımı.....	29
7-) Kaynakça.....	30

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, uygulama 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 gerçekleşmesinde 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? Tabii ki 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 aşıkardı.

1-2-) Sağladığı Faydalar

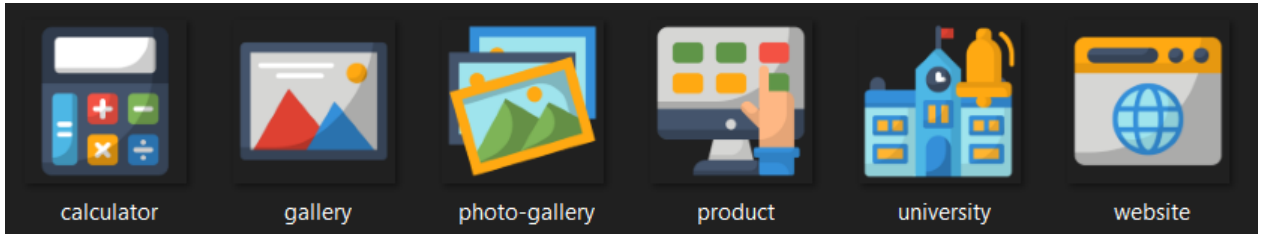
Bir önceki paragrafta ne olduğu ve ne işe yaradığı hakkında konuşuldu. Aslında gereklilikleri anlatılırken bu paragrafın konusu olan sağladığı faydalara da değinilmiş oldu. Genel hatlarıyla uygulamamız dağınık ve erişilmesi zahmetli olan üniversitemizin bilgi kaynaklarını yormayan, sade, zarif tasarım 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

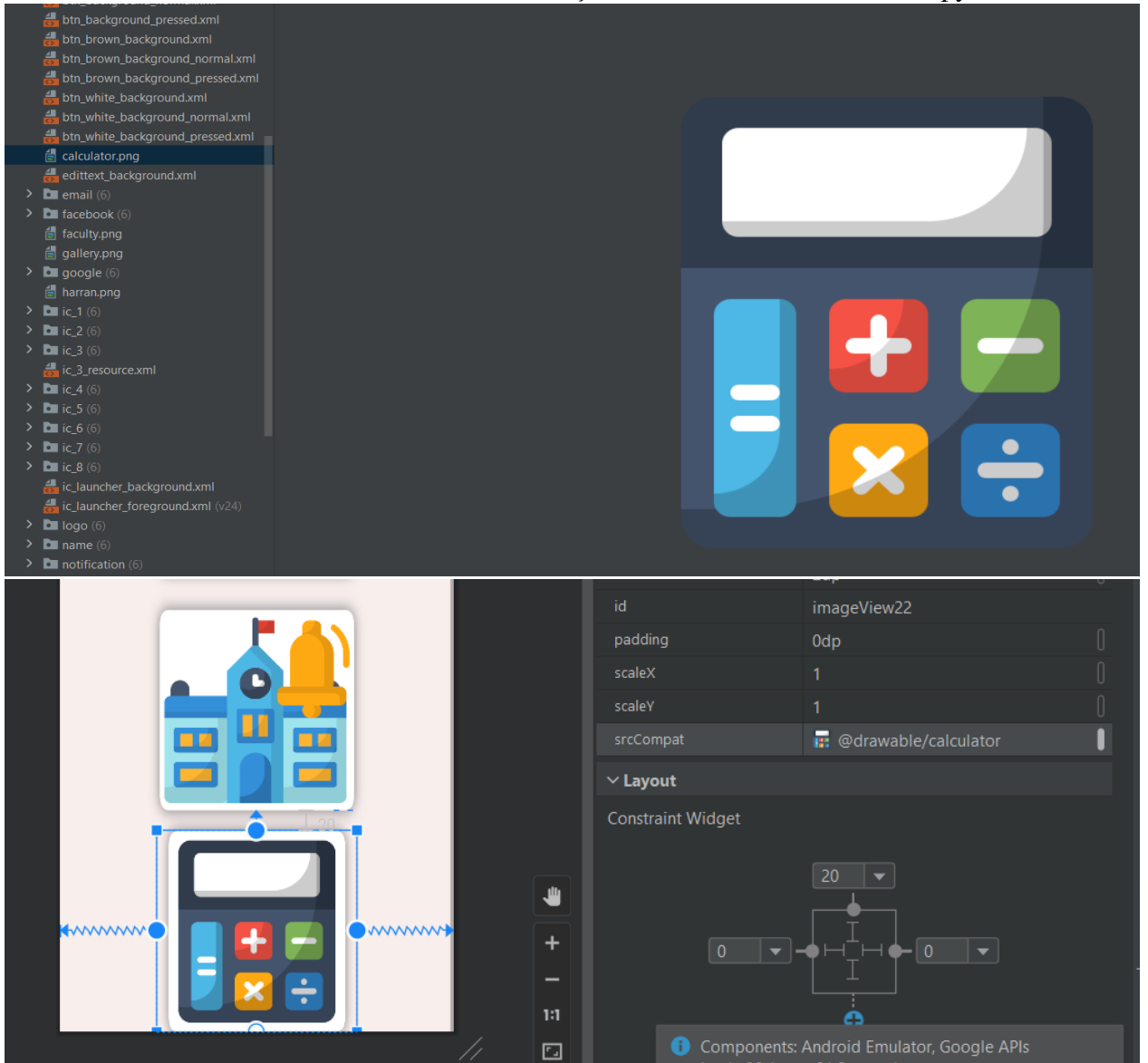
TASARIM

İkonlar

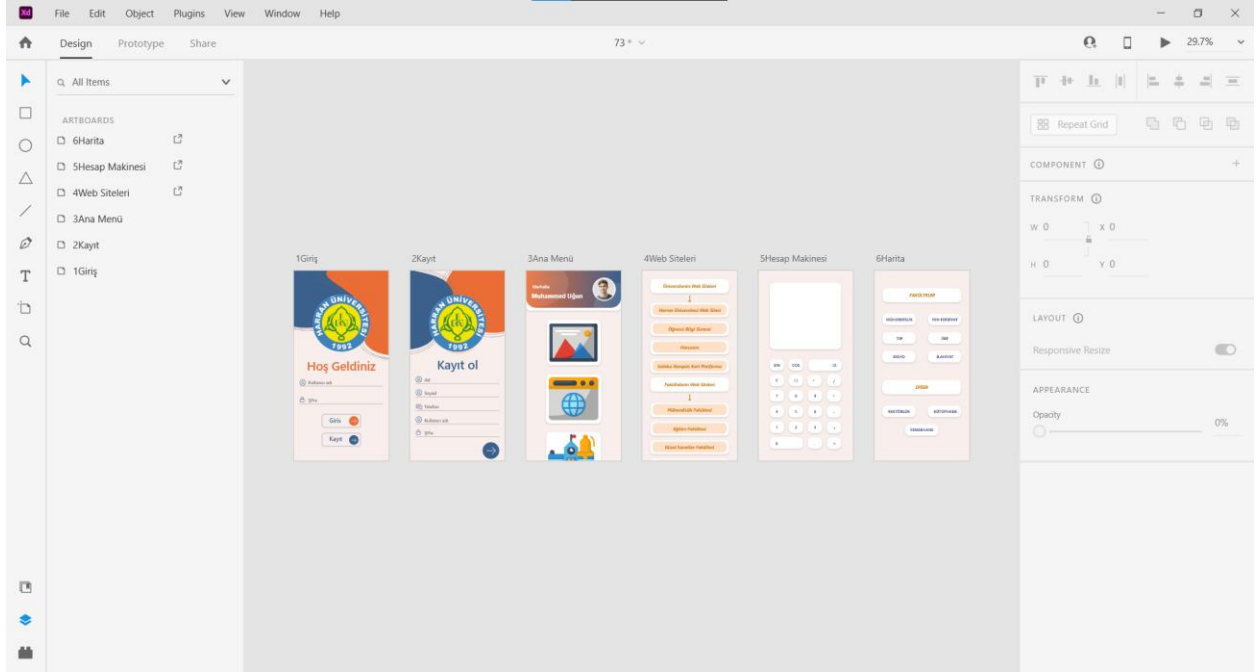
- Gerekli ikonları www.flaticon.com sitesinden bulduk.



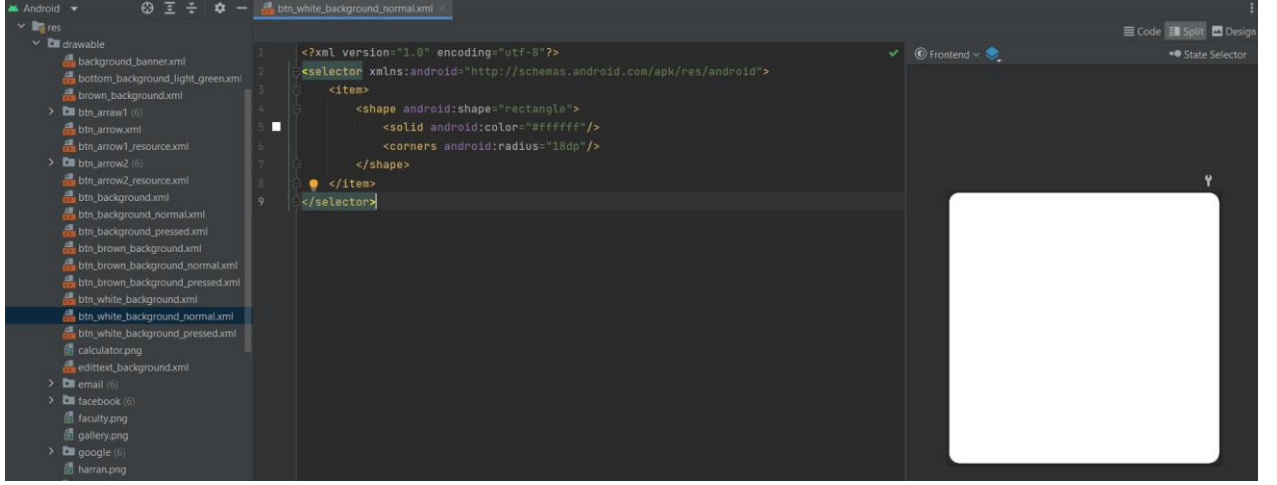
- İkonları Android Studio’da kullanmak için drawable kısmına kopyaladık.



- 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.
-

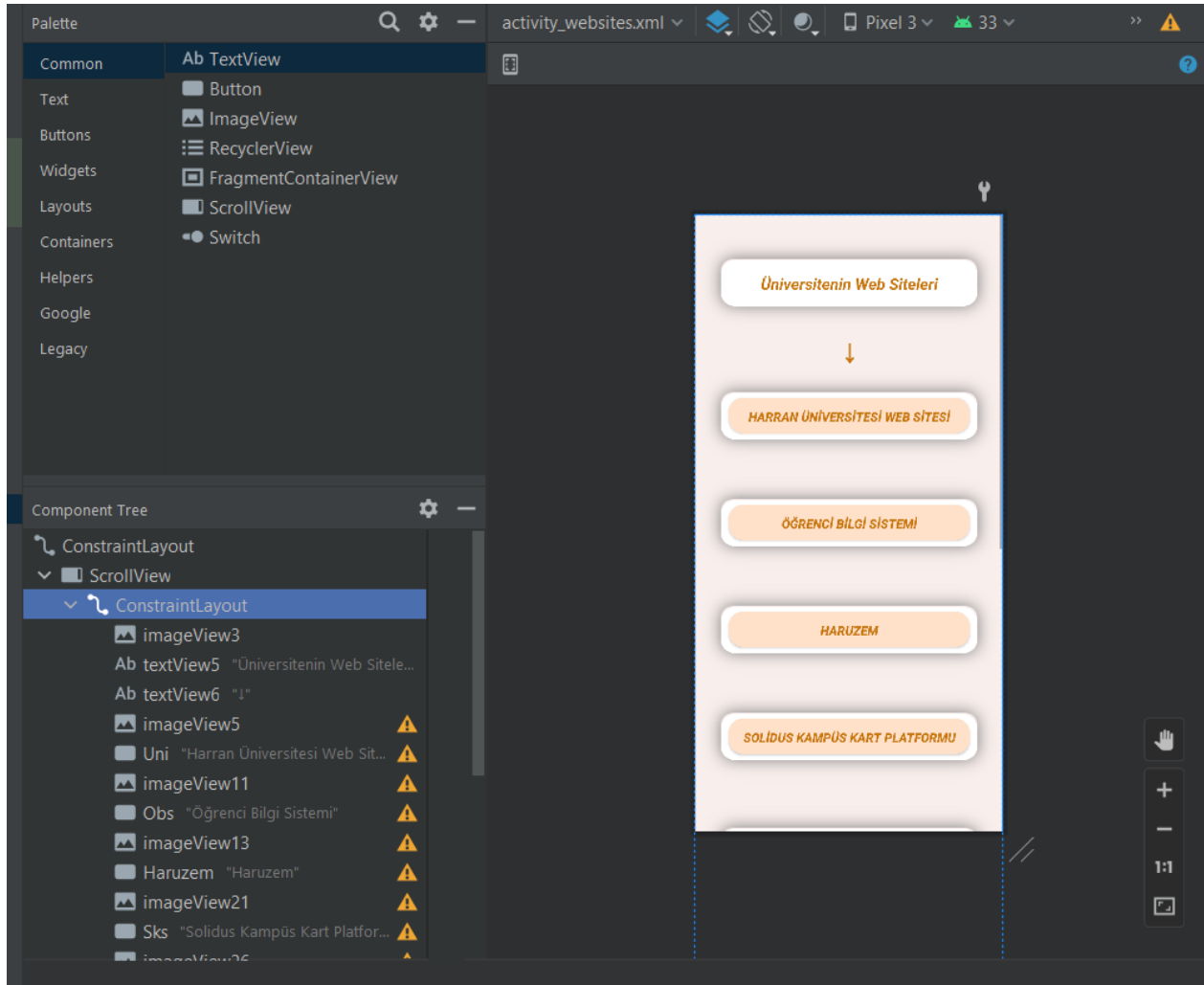
```

• <?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 tamamıyla 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 kullanıcı bilgilerinin veri tabanına kaydedilmesini sağlar.

0:05 100%

HARRAN ÜNİVERSİTESİ
1992

Kayıt Ol

Ad

Soyad

Telefon

Kullanıcı adı

Şifre

→

Giriş Yapma

Kullanıcı adının boş olup olmadığını kontrol eder. Eğer boş ise “Kullanıcı adı girin” uyarısı verecektir.

```
public Boolean dogrulamaID() //Kullanıcı adının boş olup olmadığını kontrol eder
{
    String sId = edt_Id.getText().toString();
    if(sId.isEmpty())
    {
        edt_Id.setError("Kullanıcı adı girin");
        return false;
    }
    else{
        edt_Id.setError(null);
        return true;
    }
}
```



Şifrenin boş olup olmadığını kontrol eder. Eğer boş ise “Şifre girin” uyarısı verecektir.

```
public Boolean dogrulamaPw() //Şifrenin boş olup olmadığını kontrol eder
{
    String sPw = edt_Pw.getText().toString();
    if(sPw.isEmpty())
    {
        edt_Pw.setError("Şifre girin");
        return false;
    }
    else{
        edt_Pw.setError(null);
        return true;
    }
}
```



Kullanıcı bilgilerini kontrol eder. Eğer veritabanında olmayan bir ID girerse “Kullanıcı adı yanlış!” uyarısı verir. Eğer ID doğru ve şifre yanlış ise “Şifre yanlış!” uyarısı verir. Girilen bilgiler doğru ise kullanıcının Ad ve soyad bilgilerini adVT ve soyadVT değişkenlerinde tutar (Bu bilgileri Anasayfa sayfasında kullanacağız) ve Anasayfaya yönlendirir.

```
public void checkID() { // Kullanıcı bilgilerini kontrol eder
    String sId = edt_Id.getText().toString().trim();
    String sPw = edt_Pw.getText().toString().trim();
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "kullaniciLAR");
    Query checkUserDatabase = reference.orderByChild("id").equalTo(sId);

    checkUserDatabase.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()){
                edt_Pw.setError(null);
                String sifreVT= snapshot.child(sId).child("sifre").getValue(String.class);
                // sId'de girilen id'ye göre veritabanındaki şifreyi sifreVT'ye atar
                if (sifreVT.equals(sPw)) { // sifreVT ve dışarıdalan girilen şifre eşitse giriş başarılı olur
                    edt_Id.setError(null);
                    String adVT, soyadVT;
                    adVT = snapshot.child(sId).child("ad").getValue(String.class);
                    soyadVT = snapshot.child(sId).child("soyad").getValue(String.class);
                    Intent hintent = new Intent( packageContext LoginActivity.this, HomePage.class);
                    hintent.putExtra( name: "ad",adVT);
                    hintent.putExtra( name: "soyad",soyadVT);
                    startActivity(hintent);
                }
                else {
                    edt_Pw.setError("Şifre Yanlış!");
                    edt_Pw.requestFocus();
                }
            }
            else{
                edt_Id.setError("Kullanıcı Adı Yanlış!");
                edt_Id.requestFocus();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText( context LoginActivity.this,error.toException().getMessage(),Toast.LENGTH_SHORT).show();
        }
    });
}
```

Giriş butonuna tıklandığında ID ve şifrenin boş olup olmadığını kontrol eder, boş değilse eğer ardından kullanıcı kontrolünü yapar. Eğer her iki koşulu da sağlarsa Anasayfaya yönlendirir.

```

public void clckGiris(View v) //Giris butonuna tıklandığında
{
    if(dogrulamaID() && dogrulamaPw()) //ID ve şifre boş değilse
    {
        checkID(); // kullanıcı kontrolün ardından giriş yapar
    }
}

```

Kayıt butonuna tıklandığında Kayıt ol sayfasına yönlendirir.

```

public void clckRegister(View v)// Kayıt butonuna tıklandığında Kayıt ol sayfasına yönlendirir
{
    Intent intent= new Intent( packageContext LoginActivity.this, Register.class);
    finish();
    startActivity(intent);
}

```

3-4-) Sisteme Kayıt

Kayıt ol butonuna tıklandığında aşağıda açıklaması yapılan kodlar çalışır.

```

public void kayıtOl(View view) //kayıt ol butonundaki onClick
{
    sId=edt_Id.getText().toString();
    sPw=edt_Pw.getText().toString();
    sAd=edt_Ad.getText().toString();
    sSoyad=edt_Soyad.getText().toString();
    sTelNo=edt_TelNo.getText().toString();

    //Değerler boş girilmediği sürece
    if(!TextUtils.isEmpty(sId) && !TextUtils.isEmpty(sPw) && !TextUtils.isEmpty(sAd)
        && !TextUtils.isEmpty(sSoyad) && !TextUtils.isEmpty(sTelNo))
    {
        //Data ile Veritabanına bilgileri gireriz
        //("veritabanındaki key kısmı" , key'in dışarıdan girilen değeri)
        mData=new HashMap<>();
        mData.put("id",sId);
        mData.put("sifre",sPw);
        mData.put("ad",sAd);
        mData.put("soyad",sSoyad);
        mData.put("telNo",sTelNo);

        mReference.child("kullanıcılar").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.hasChild(sId)) //Veritabanındaki bütün id'leri sId'de girilen ile kıyaslar ve değer true ise
                {
                    Toast.makeText( context Register.this, text "Böyle bir kullanıcı adı zaten var!",Toast.LENGTH_SHORT).show();
                }
                else // mData'ya girilen değerleri sId'deki değere göre veri tabanına ekle
                {
                    mReference.child("kullanıcılar").child(sId).setValue(mData)
                        .addOnCompleteListener( activity Register.this, new OnCompleteListener<Void>() {

```

Kodun devamı

```

else // mData'ya girilen degerleri sId'deki degera göre veri tabanına ekle
{
    mReference.child("kullaniciId").child(sId).setValue(mData)
        .addOnCompleteListener( new Register.this, new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful())
                {
                    Toast.makeText( context, Register.this, text: "Kayıt işlemi başarılı!", Toast.LENGTH_SHORT).show();
                    Clear();
                    edt_Ad.requestFocus();
                }
                else
                {
                    Toast.makeText( context, Register.this,task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        });
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Toast.makeText( context, Register.this,error.toException().getMessage(), Toast.LENGTH_SHORT).show();
}
});
}
else
{
    Toast.makeText( context, this, text: "İstenilen Değerler Boş Geçilemez", Toast.LENGTH_SHORT).show();
}
}
}

```

Kayıt ol butonuna basıldığında şartları sağlıyorsa, kullanıcının kaydı yapılır ardından bütün editTextlerin içi boşaltılır.

```

private void Clear()
{
    edt_Id.setText("");
    edt_Pw.setText("");
    edt_Ad.setText("");
    edt_Soyad.setText("");
    edt_TelNo.setText("");
}

```


Anasayfa Dizaynı

Ana menü tasarımı diğer tasarımlara uygun olması için yani genel konsepti koruması için aynı renk ve şekilleri kullanmaya özen gösterdik. Tasarıma genel olarak bakıldığında üst tarafta kullanıcıyı ismiyle karşılayan bir bölüm ve alt tarafta ise ikonlardan oluşan, activitylere yönlendiren ImageView'lerden oluşan butonlarımız mevcut.



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, "Harita Hazır", Toast.LENGTH_SHORT).show();
    Log.d(TAG, "onMapReady:harita hazır");
    mMap = googleMap;

    if (mLocationPermissionsGranted) {...}
}
```

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



```
onRun.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent( packageContext, HaritaActivity.this, MapMuhendis.class);
        startActivity(intent);
    }
});
onTip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent( packageContext, HaritaActivity.this, MapTip.class);
        startActivity(intent);
    }
});
onFenE.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent( packageContext, HaritaActivity.this, MapFenEdb.class);
        startActivity(intent);
    }
});
onIih.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent( packageContext, HaritaActivity.this, MapIih.class);
        startActivity(intent);
    }
});
});
```

Butonlara basılmadan önce harita initMap() ile yüklenir.

```
private void initMap(){// Haritanın yüklenmesi
    Log.d(TAG, "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();//aratilacak 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: IOException: " + e.getMessage());
    }
    if (list.size()>0){
        Address address = list.get(0);
        Log.d(TAG, msg: "geoLocate: bir yer bulundu:"+ address.toString());//konum üstü bilgiler

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

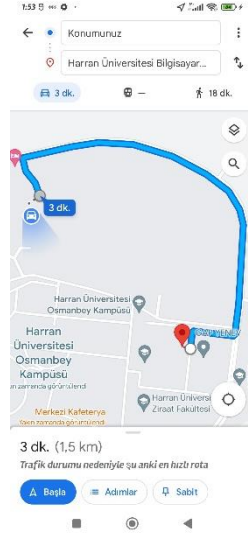
Aynı zamanda konumun üstüne tıklandığında konum bilgileri verilir. geoLocate() methodu içinde tanımlanmıştır.



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

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

İşaretili 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: "onComplete:Bulunan yer");
                        Location currentLocation=(Location) task.getResult();

                        moveCamera(new LatLng(currentLocation.getLatitude(),currentLocation.getLongitude()),
                                DEFAULT_ZOOM, title: "Benim konumum" );
                    }else{
                        Log.d(TAG, msg: "onComplete: 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 istisnası: "+e.getMessage() );
    }
}
```

Eğer kullanıcı kendi gezmek isterse search arama butonu bulunmaktadır. Map java'daki `clcg()` 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("Gölveren, Mehmet Arabacı Beden Eğitimi 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);  
    }  
});
```

5-3-) Galeri

Bu metod veritabanına eklenen fotoğrafları çeker ve ilk fotoğrafı ekranda gösterir.

```
public void resimYukleniyor()
{
    progressDialog= new ProgressDialog( context: Photos.this);
    progressDialog.setMessage("Yükleniyor..");
    progressDialog.setCancelable(false);
    progressDialog.show();
    storageReference= FirebaseStorage.getInstance().getReference( location: "images/1.jpg");
    try {
        File localfile= File.createTempFile( prefix: "tempfile", suffix: ".jpg");
        storageReference.getFile(localfile)
            .addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();

                    Bitmap bitmap= BitmapFactory.decodeFile(localfile.getAbsolutePath());
                    binding.ivPhotos.setImageBitmap(bitmap);
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();
                    Toast.makeText( context: Photos.this, text: "Fail", Toast.LENGTH_SHORT).show();
                }
            });
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Geri butonuna tıklandığında bir önceki fotoğrafa döner

```

public void ilerileri(View v) { // İleri butonuna tıklandığında bir sonraki fotoğrafa gider
    progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Yükleniyor..");
    progressDialog.setCancelable(false);
    progressDialog.show();
    if(sayac>8)
    {
        sayac=1;
    }
    String imageID= Integer.toString(sayac);
    sayac++;
    storageReference= FirebaseStorage.getInstance().getReference().child("images/"+imageID+".jpg");
    try {
        File localFile= File.createTempFile("tempfile", ".jpg");
        storageReference.getFile(localFile)
            .addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();

                    Bitmap bitmap= BitmapFactory.decodeFile(localFile.getAbsolutePath());
                    binding.ivPhotos.setImageBitmap(bitmap);
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();

                    Toast.makeText(this, "Fail", Toast.LENGTH_SHORT).show();
                }
            });
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

İleri butonuna tıklandığında bir sonraki fotoğrafa geçer

```

public void geri(View v) { // Geri butonuna tıklandığında bir önceki fotoğrafa döner
    progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Yükleniyor..");
    progressDialog.setCancelable(false);
    progressDialog.show();
    if(sayac<1)
    {
        sayac=8;
    }
    String imageID= Integer.toString(sayac);
    storageReference= FirebaseStorage.getInstance().getReference().child("images/"+imageID+".jpg");
    try {
        File localFile= File.createTempFile("tempfile", ".jpg");
        storageReference.getFile(localFile)
            .addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();
                    sayac--;
                    Bitmap bitmap= BitmapFactory.decodeFile(localFile.getAbsolutePath());
                    binding.ivPhotos.setImageBitmap(bitmap);
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    if(progressDialog.isShowing())
                        progressDialog.dismiss();

                    Toast.makeText(this, "Fail", Toast.LENGTH_SHORT).show();
                }
            });
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

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

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

```
case R.id.btn_0:
    display.setText("");
    break;
case R.id.btn_brackets:
    addBrackets();
    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 updateDisplay(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 rightSideOnDisplay = oldDisplay.substring(cursor_pos);  
        String newText = leftSideOnDisplay + s + rightSideOnDisplay;  
        display.setText(newText);  
    }  
    display.setSelection(cursor_pos + 1);  
}
```

Burada hesaplamamanın 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(" ")) countBracets++;  
        if (textDisplay.substring(i, i + 1).equalsIgnoreCase("(")) countBracets--;  
    }  
    String lastCharOfTextDisplay = textDisplay.substring(textDisplay.length() - 1);  
    if (countBracets == 0 || lastCharOfTextDisplay.equals("(")) updateDisplay( s: "(");  
    else if (countBracets > 0 && !lastCharOfTextDisplay.equals("(")) updateDisplay( s: "(");  
}
```

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

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

Kaynakça

<https://www.youtube.com/playlist?app=desktop&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt>

https://www.youtube.com/watch?v=TLEEdVlXKJk&ab_channel=LogicofComputer

<https://tr.wikipedia.org/wiki/Firebase#:~:text=Firebase%2C%20Google%20taraf%C4%B1ndan%20mobil%20ve,konusunda%20Google'%C4%B1n%20%C3%B6nc%C3%BC%20hizmetidir.>

www.flaticon.com

<https://mathparser.org/>