

**T.C**

**İNÖNÜ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**STAJ RAPORU**

**HAZIRLAYAN**

**AD SOYAD:** HAMİT KÖROĞLU

**ÖĞRENCİ NO:**  02140001042

**MALATYA**

**2018**

**İÇİNDEKİLER**

Giriş-Kurum Tanıtımı...........................................................................1

Projelerin Ortamları..............................................................................2

Projelerin Tanıtımı................................................................................4

Android studio kurulumu......................................................................6

Seyahat defterim(Proje ön çalışması)..................................................10

Seyahat defterim (Projenin oluşturulması ve tasarım Yapılması).......11

Seyahat defterim(Projeye harita ekleme,google-harita ayarlarını yapma,haritanın projeden ulaşılabilir olmasını sağlamak.).................12

Seyahat defterim(Gerekli nesne tanımlamalarının yapılması,Kullanıcı izinlerinin alınması,haritada gezinebilmeyi sağlamak.)......................13

Seyahat defterim(En son lokasyonun tekrar açılışta programda gösterilmesi,adresleri almak)..............................................................15

Seyahat defterim(Marker eklenmesi Alınan enlem ve boylam ve kaydedilen yerin isim değerlerini sqlLite veritabanına kaydetmek)...17

Seyahat defterim(Kaydedilen verilerin veritabanından çekilmesi).....18

Seyahat defterim(Listviewde tıklanılan yere yönlendirme yapılması,ekledikten hemen sonra listview'de eklenen adresin gözükmesi)..........................................................................................19

Seyahat kitabım(Projeden görseller)...................................................22

Survivor Bird(Proje ön çalışması).......................................................23

Survivor Bird(LibGdx hakkında araştırma yapılması.Kurulumunun yapılması)............................................................................................24Survivor bird(Oyun görsellerinin bulunması.Tasarımın yapılması)...26

Survivor bird(Oyunda yerçekimi oluşturulması.Kuşa uçma yeteneği eklemek.Arıları oyunumuza dahil etmek.Çarpışmaları anlamak).......29

Survivor bird(Skorlama yapmak.Yanınca bitti uyarısı vermek).........34

Survivor bird(Projeden görseller)........................................................38

Ders Notlarım(Proje ön çalışması)......................................................39

Ders Notlarım(Projenin oluşturulması ve tasarım Yapılması)............40

Ders Notlarım(2. Activity Tasarımı(Main2Activity) tasarımı)...........41

Ders Notlarım(Kullanıcı izinlerinin alınması)....................................42

Ders Notlarım(Kamera bağlantısının yapılması.Çekilen fotoğrafın imageviewe aktarılması).....................................................................44

Ders Notlarım(Bazı arayüz kısıtlamalarının yapılması).....................45

Ders Notlarım(Veritabanı kayıt işlemi)..............................................47

Ders Notlarım(Kayıtlı verilerin çekilmesi Listviewe aktarılması ve tıklanıldığında diğer ekrana aktarılması).............................................48

Ders Notlarım(Projeden görseller)......................................................50

Instagram Klon(Proje ön çalışması)....................................................51

Instagram Klon(Parse ve back4app nedir araştırma yapılması uygulamaya dahil edilmesi)................................................................52

Instagram Klon(Back4app kurulumu ve uygulama ile server bağlantısı)............................................................................................53

Instagram Klon(Login ekranı tasarımı ve login işlemi)......................55

Instagram Klon(Kullanıcı hatırlatma ve menü tasarımı).....................57

Instagram Klon(Kullanıcı izinleri post ekleme ekranı tasarımı ve post yükleme işlemi)...................................................................................58

Instagram Klon(Kayıtların gösterimi için tasarım yapılması ve bağlayıcı sınıf yazmak(custom view ile postActivity bağlantısı))......60

Instagram Klon(Bağlayıcı sınıfın bağlantısının yapılması ve kayıtların getirilmesi)..........................................................................................62

Instagram Klon(Projeden Görseller)...................................................64

Uygulamalar google play'e nasıl yüklenir...........................................65

Robot Kol projesi(Proje ön çalışması)................................................67

Robot Kol projesi(Gerekli ekipmanların tespiti ve eksiklerin temin edilmesi)..............................................................................................68Robo Kol projesi(Dış iskelet çizimlerinin yapılması).........................69

Robot Kol projesi(Dış iskeletin 3d yazıcı ile çıktı alınması)..............70

Robo Kol projesi(Dış iskeletin birleştirilmesi ve yanlış baskıdan kaynaklı kusurların giderilmesi)..........................................................71

Robot Kol projesi(Elektronik kartın hazırlanması).............................72

Robo Kol projesi(Robot kol'un arduino kod kısmının algoritmasının oluşturulmasıyla birlikte kodlamaya geçilmesi ve giriş-çıkış birimlerinin tanımlanması)..................................................................73

Robot Kol projesi(Button hareketlerine göre kontrolün arduino kodlarının yazılması ve projenin tamamlanması)...............................74

**GİRİŞ**

Maker çocuk teknoloji ve danışmanlıkta yaptığım 40 günlük stajın ardından edindiğim tecrübenin ve yaptığım projelerin raporudur

**KURUM TANITIMI**

**İşyerinin adı:Maker Teknoloji danışmanlık san. tic. ltd. şti**

Çocuklara yeni nesil üretim araçlarını ve üretme kültürünü öğretmek için Mayıs 2015'te kuruldu. Kurulduğu günden beri 10000'den fazla çocuk ve 1000'den fazla yetişkin, öğretmen ve eğitimene maker hareketini ve maker hareketine dair araçların kullanımı öğretildi. İçerisinden kısa zamanda onlarca proje çıkan Maker Atölye'de devamlı olarak atölyeler yapılmaktadır.

Maker çocuk ekibi farklı disiplinlerden oluşan, değişim ve farklılık yaratmak isteyen bir takımdır.Ekipte Bilgisayar Öğretmenleri de var, Elektronik Mühendisler de, tasarımcılar da. Böylece birbirlerinden öğrenerek öğretebiliyor, çocuklara farklı bakış açıları kazandırabiliyorlar.

**İletişim Bilgileri**

**Adres:** Platform İstanbul Eski Büyükdere Cd. Behçet Sk. Büyükdere İş Merkezi, 5.Kat 4.Levent, İstanbul

**Telefon:** 0552 205 8125

**Mail:**sendeyap@makercocuk.com

C:\Users\Hamit\Desktop\Yaz Stajı\Makercocuk_logo.fw_.png

**PROJELERİN ORTAMLARI**

4 tane android 1 tane de arduino olmak üzere toplamda 5 proje bulunmaktadır.

Android projeler;

1-)Seyahat Defterim

2-)Survivor Bird

3-)Ders Notlarım

4-)Instagram Klon

**Android proje geliştirme ortamı;**

Android Studio, Android uygulamaları geliştirmek için kullanılan resmi bir yazılımdır. Resmi olması Google tarafından geliştirilmesidir.[IntelliJ IDEA](https://www.jetbrains.com/idea/) tabanlı olarak geliştirilen özel bir geliştirme ortamıdır. Sadece Android uygulamaları geliştirmek için tasarlanmıştır.

**Android Studio ile ilgili temel özellikler**

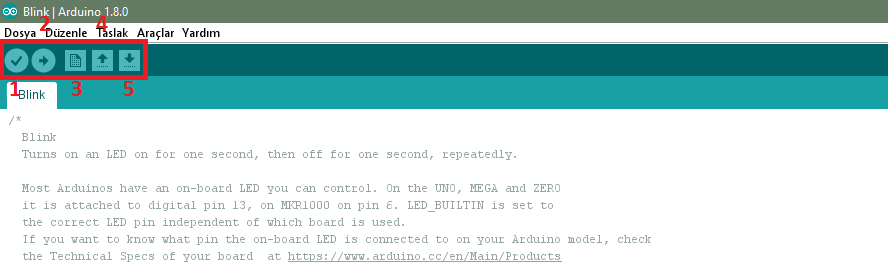
* Gradle tabanlı esnek bir yapıya sahip olması (Android Studio, yazdığımız kodları derlemek için Gradle kullanır.)
* Uygulamalarımızı kontrol edip çalıştırabileceğimiz emülatör ortamına sahip olması
* Tüm Android cihazlarına uygun uygulama geliştirme ortamına sahip olması
* GitHub entegrasyonuna sahip olması
* Kapsamlı test araçlarına sahip olması
* C++ ve NDK(Native Development Kit) desteğinin olması gibi birçok özelliğe sahiptir.

Arduino projesi;

1-)Robot kol projesi

**Arduino proje geliştirme ortamı;**

Türkçe dil desteği de bulunan Arduino IDE oldukça basit, anlaşılır ve kullanışlı bir geliştirme ortamına sahip. Kısaca inceleyecek olursak;

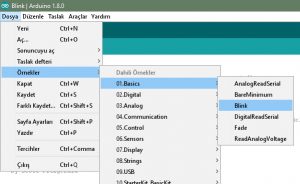
[](https://www.mobilhanem.com/wp-content/uploads/2017/01/Ekran-Al%C4%B1nt%C4%B1s%C4%B1-1.png)

Yukarıdaki görüntüye göre;

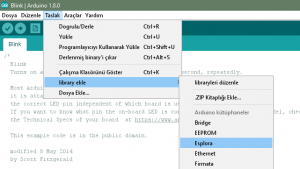
Kontrol et: Kodumuz da bir yazım hatasının bulunup bulunmadığını kontrol etmemizi,

1. Yükle: Kodumuzu önce kontrol edip ardından aygıtımıza yüklememizi,
2. Yeni: Yeni bir .ino dosyası oluşturmamızı,
3. Aç: Kayıtlı bir dosyayı açmamızı,
4. Kaydet: Kodumuzun son halini kaydetmemizi sağlamaktadır.

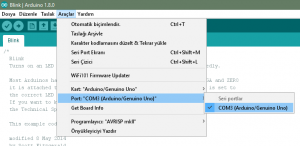
Ayrıca menümüzden bahsedecek olursak;

[](https://www.mobilhanem.com/wp-content/uploads/2017/01/dosya-1.png)

**Dosya > Örnekler** ile bizler için hazırlanmış çok sayıda örnek uygulamaya erişebiliriz.

[](https://www.mobilhanem.com/wp-content/uploads/2017/01/ekle.png)

**Taslak > Library Ekle** ile hali hazırda Arduino IDE ile gelen ve kendi oluşturabileceğimiz kütüphaneleri projemize ekleyebiliriz.

[](https://www.mobilhanem.com/wp-content/uploads/2017/01/port.png)**Araçlar > Kart** ile kullanmış olduğumuz Arduino kartımızın modelini seçebilir,

**Araçlar > Port** ile bilgisayarımıza bağlı olan kullanmak istediğimiz Arduino kartlarını seçebiliriz.

**PROJELERİN TANITIMI**

**1-)Seyahat Defterim:**

Kullanıcı daha önceden gidip gezdiği gördüğü beğendiği yerleri harita üzerine iğne ekleyerek kaydetmek istediği yerin bilgilerinin alınması sağlanacak.

Alınan bilgiler sqlite veritabanı sayesinde kaydedilecek.Bu sayede uygulama kapatılıp tekrar açıldığından verilerin kaybolmaması sağlanabilecek.

Daha sonradan kullanıcı istediği zaman kaydettiği yere istediği zaman tıkladığında kaydettiği yerin adres bilgilerini harita üzerinde görebilecek.

Kullanılacak Dil:Java

Kullanılacak Editör:Android Studio

Kullanılan API:API26:Android 8.0(OREO)

**2-)Survivor Bird:**

Oyun flappy bird oyunu tarzında uçan bir kuşumuz olacak yer çekimi etkisiyle aşağıya doğru düşerken kullanıcımızın ekrana her tıkladığında belli bir yüksekliğe çıkar,sonra yer çekimi etkisiyle tekrar aynı şekilde aşağıya doğru iner.Bu döngüsel olarak devam eder.Bir yandanda kullanıcımız random bir şekilde gelen arılara temas etmekten kaçınmak zorundadır.Her geçtiği arı kümesinden sonra skor tabelası 1 artmaktadır.Arılara temas ettiği takdirde veya yere düştüğü zaman yanmaktadır ve oyun sonlanmaktadır.

Kullanılacak Dil:Java

Kullanılacak Framework:libGdx

Kullanılacak Editör:Android Studio

Kullanılan API:API26:Android 8.0(OREO)

**3-)Ders Notlarım:**

Kullanıcı uygulamada cihazın kamerası yardımıyla ders notunun fotoğrafını çekebilecek.

Çektiği nota kendisi isim verebilecek örneğin:Algoritma ve Programlama 1 gibi.

Daha sonradan kullanıcı istediği zaman kaydettiği yere istediği zaman tıkladığında kaydettiği notu görebilecek.

Uygulama fotoğrafları kendi içerisinde saklayacak bu sayede kişisel telefonunda karmaşıklığa sebep olan not fotoğrafların kurtarılmış olacak.

Kullanılacak Dil:Java

Kullanılacak Editör:Android Studio

Kullanılan API:API23:Android 6.0(MARSHMALLOW)

**4-)Instagram Klon:**

Bir login ekranı tasarlanacak login işlemleri gerçekleştirilecek.

Bir login ekranı tasarlanacak login işlemleri gerçekleştirilecek.

Kullanıcı daha önceden çekmiş olduğu fotoğrafları uygulamaya yükleyebilecek ve yorum yapabilecek.

Daha sonradan kullanıcı istediği zaman hesabına girdiğinde paylaşılmış gönderileri görebilecek

Tüm bu paylaşılan fotoğraflar yorumlar kullanıcı kayıt işlemleri vs depolama gerektiren tüm işlemleri parse backend service'i üzerinden back4app.com da kuracağımız parse server üzerinden yapılacak.

Kullanılacak ekstra teknolojiler:Parse backend servis ve back4app

Kullanılacak Dil:Java

Kullanılacak Editör:Android Studio

Kullanılan API:API26:Android 8.0(OREO)

**5-)Robot Kol Projesi:**

Arduino elektronik devre kartıyla kontrol edilen açısal hareketleri Servo motorlar yardımıyla yapılan bir robot kol projesidir.

1 adet Arduino Mega 1 adet 220 Ohm direnç

4 adet Standard servo motor Bağlantılar için kablo ve konnektörler somun ve civatalar

2 adet Mikro servo motor Dış iskelet için yaklaşık 1 kg filament

8 adet push button

1 adet aç/kapa anahtar

1 adet potansiyometre ve knob(pot başlığı)

1 adet yeşil LED

Baskı devre kartı

8 adet 10K Ohm direnç

|  |
| --- |
| **Proje-Resim/Rapor**  Öncelikle https://developer.android.com/studio/ indirme sayfasına gidiyoruz:  android-1Yeşil renkli "Download Android Studio for Windows" düğmesine tıkladıktan sonra lisans sözleşmesi karşımıza gelecek. Sözleşmeyi okuduktan sonra, kabul etmek için "I Agree" düğmesine basmamız gerekiyor.  1  Sözleşmeyi onayladıktan sonra indirme işlemi otomatik olarak başlayacak. İndirme işlemi tamamlandıktan sonra dosyamıza çift tıklıyoruz. UAC etkinse, Windows öncelikle bizden cihaz üzerinde kurulu yönetimsel izinleri isteyecek. Yönetici izni verildikten sonra, Android Studio kurulumu için ilk adım görülecek. Next düğmesine basarak kurulum işlemine başlıyoruz.  android1  Next düğmesi sonrası karşımıza çıkacak ekranda yüklemek istediğimiz bileşenleri seçiyoruz ve tekrar Next düğmesine basıyoruz.  android2  Next düğmesine bastıktan sonra yine lisans sözleşmesi karşımıza gelecek. Sözleşmeyi okuduktan sonra eğer kabul ediyorsak "I Agree" düğmesine basmamız gerekiyor.  android3  Sözleşmeyi onayladıktan sonra karşımıza Android Studio'nun ve SDK'nın nereye yükleneceğini soran bir pencere gelecek. Uygun yerleri belirttikten sonra tekrar "Next" düğmesine basıyoruz.  android5  Eğer Intel Donanım Hızlandırıcı'ya sahipseniz, Bir sonraki adımda ise karşımıza Android uygulamanız için kullanabileceğiniz emülatörün kurulum ayarlarının bulunduğu ekran çıkacak. Buradan emülatörün kullanabileceği bellek miktarını ayarlayabilirsiniz.  Intel Donanım Hızlandırıcı desteği bulunmayan bilgisayarlarda, bu pencere görüntülenmeyip, doğrudan bir sonraki adıma geçilmektedir.  android6  Android Studio'nun "Başlat" çubuğunda kısayol olarak gözükmesini istemiyorsak "Do not create shortcuts" seçeneğini işaretliyoruz ve "Install" düğmesine basıyoruz.android7  Ardından kurulum işlemimiz başlıyor.  android8  Kurulum bittikten sonra "Next" düğmesine tıklıyoruz. Çıkan pencereden "Finish" düğmesine tıklıyoruz ve kurulumu sonlandırıyoruz.  android9 |

|  |
| --- |
| **Seyahat Kitabım Uygulaması**  Android üzerinde çalışan Seyahat Defterim isimli bir projenin yapım aşamalarıdır.  **Proje İçeriği:**  Kullanıcı daha önceden gidip gezdiği gördüğü beğendiği yerleri harita üzerine iğne ekleyerek kaydetmek istediği yerin bilgilerinin alınması sağlanacak.  Alınan bilgiler sqlite veritabanı sayesinde kaydedilecek.Bu sayede uygulama kapatılıp tekrar açıldığından verilerin kaybolmaması sağlanabilecek.  Daha sonradan kullanıcı istediği zaman kaydettiği yere istediği zaman tıkladığında kaydettiği yerin adres bilgilerini harita üzerinde görebilecek.  Kullanılacak Dil:Java  Kullanılacak Editör:Android Studio  Kullanılan API:API26:Android 8.0(OREO)  **Projede elde edilmesi planlanan kazanımlar :**  Tasarım ekranı yapılması sayesinde Android studioda tasarım öğrenilmesi ve tasarım tecrübesi elde edilmesi.  Sqlite veritabanı kullanılmasıyla Android platformu üzerinde veritabanı yaratılması ve yönetimi öğrenilmesi.  Maps ler üzerinde çalışılacağı için Google maps ile projemiz arasındaki bağlantının yapılması ve üzerinde işlem yapılabilmesinin öğrenilmesi ve tecrübe edilmesi. |

|  |  |
| --- | --- |
| **Projenin oluşturulması ve tasarım Yapılması**  Android studioyu açıldığında Start a new Android Studio Projecte tıklanır.Activity ekleme ekranında Empty Activity seçilir.Proje oluşturulur.  Bunları yaptıktan sonra ilk olarak activity\_main.xml dosyasına girilip Paletteden sürükle bırak ile bir listView eklendi.ListView e ID olarak listView verildi.Projemizde tanıtılması için onCreate metodu altında  ListView listView=(ListView)findViewById(R.id.*listView*); diyerek tanımlarız.  menu  Daha sonra, menü ekleyebilmek için app'nin altında res klasörüne sağ tıklanıp Directory'e tıklanır.İsim olarak "menu" verilir.Daha sonra menu klasörüne sağ tıklanıp Menu resource file'a tıklanıp gelen ekranda isim verilir.isim olarak yer\_ekle verdik.  yer\_ekle.xml in Text kısmına  <item android:id="@+id/yer\_ekle" android:title="@string/yer\_ekle"></item>  yazılır.  Elde edilen görüntü yandaki şekildeki gibi olur.  MainActivity.java içinde onCreateOptionsMenu yazılarak override olan onCreateOptionsMenu fonksiyonumuzu oluştururuz.Menümüzü bağlamamızı sağlar.  İçerisinede;  MenuInflater menuInflater=getMenuInflater();  menuInflater.inflate(R.menu.*yer\_ekle*,menu);  @Override public boolean onCreateOptionsMenu(Menu menu) {  MenuInflater menuInflater=getMenuInflater();  menuInflater.inflate(R.menu.*yer\_ekle*,menu);  return super.onCreateOptionsMenu(menu); }  Yine aynı şekilde MainActivity.java içerisinde onOptionsItemSelected yazarak başka bir override methodumuzu oluştururuz buda menüye tıklandığında ne olacağını yazacağımız fonksiyondur.  *//menüye tıklandığında ne olacağıyla ilgili yapılan işlemleri* @Override public boolean onOptionsItemSelected(MenuItem item) {  if(item.getItemId()==R.id.*yer\_ekle*)  {  //Buraya daha sonra Intent Yardımıyla harita bağlanacak.  }  return super.onOptionsItemSelected(item);}   |  | | --- | | **Projeye harita ekleme,google-harita ayarlarını yapma,haritanın projeden ulaşılabilir olmasını sağlamak.**  Projemize harita ekleyebilmek için package'in üzerine sağ tıklanıp activitye tklanır ordanda gallery'e tıklanır.  Google Maps Activity seçilir.isimGerekli ayarlamalar yapılarak activityimiz projemize eklenmiş olur.  Proje açıldığında ilk olarak karşımıza google\_maps\_api.xml isimli bir dosya gelir.Bu dosya values klasörünün hemen altında bulunur.Bunun bize faydası kendi uygulamamızı google'a kaydedeceğiz ve google bize bir anahtar kelime verecektir.Verdiği bu anahtar kelime sayesinde  Map'leri kullanmayı sağlayacağız.Bu anahtar kelimeyi de alabilmek için önce yapmamız gereken bir google hesabına sahip olmak.google hesabına giriş yapılır.Daha sonra google\_maps\_api.xml dosyasının içerisinde ;  https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend  &keyType=CLIENT\_SIDE\_ANDROID&r=8A:B9:94:C5:43:7E:AC:1D:23:1B:B1:0E:8F:2F:  D4:B2:7C:C4:E9:8D%3Bcom.example.hamit.seyahatkitabim  Tarzındaki url yi herhangi bir tarayıcıdan açılır.Gelen ekranda create a new project seçeneği seçilir continue'ye tıklanır.Daha sonra gelen ekranda Create Api Key'e tıklanır.  AIzaSyA5SB0Vsz4FuE-Fmb-vezCWWflj-SN1\_vA  Bu tarzda bir anahtar verir.Bunu google\_maps\_api.xml dosyasının içindeki YOUR\_KEY\_HERE yazan yere yapıştırılır.  Daha sonra onOptionsSelectedItem fonksiyonumuzun içerisindeki if'in içerisine ;  Intent ıntent=new Intent(getApplicationContext(),MapsActivity.class);  startActivity(ıntent);  Yazarız bu sayede yer ekle'ye tıkladığımızda haritayı açmamızı sağlar. | |

|  |  |  |
| --- | --- | --- |
| **Gerekli nesne tanımlamalarının yapılması, Kullanıcı izinlerinin alınması,haritada gezinebilmeyi sağlamak.** |  | |
| Daha sonra harita ayarlarımızı yapmak ve konum üzerinde oynama yapabilmemiz için (google maps activity eklerken verdiğimiz isimle oluşturulan java classı) MapsActivity.java içerisinde global olarak ;  LocationManager locationManager;  LocationListener locationListener;  Tanımlanır.Daha sonra onMapReady fonksiyonu içerisinde ataması yapılır.  locationManager=(LocationManager)this.getSystemService(Context.*LOCATION\_SERVICE*);  locationListener = new LocationListener()  Kullanıcının yerini vs tespit edebilmek için neredeyse her uygulamada yapıldığı gibi burdada kullanıcıdan izin istenmesi gerekmektedir.izin isteyebilmek için ise ;  if (Build.VERSION.*SDK\_INT* >= 23) {//sdk version 23 ten büyükse  if(checkSelfPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*)!=PackageManager.*PERMISSION\_GRANTED*) {//eğer izin yoksa  requestPermissions(new String[] {Manifest.permission.*ACCESS\_FINE\_LOCATION*},1);//izin varmı kontrol et yoksa izin iste} else {//eğer izin var ise  locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*,0,0,locationListener);//kullanıcının yerini al  } else {//sdk 23 ten küçükse izin istemeksizin işlemleri yapar  locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*,0,0,locationListener);  }}  İzinler alındıktan sonra onMapReady'den çıkılarak, onRequestPermissionsResult yazılarak kullanıcını izin vermişse ne olacağını yazacağımız fonksiyonumuzu oluştururuz.  @Override public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  if(grantResults.length>0)  { if(requestCode==1)  {if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS\_FINE\_LOCATION)  ==PackageManager.PERMISSION\_GRANTED);  {  locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,0,0,locationListener);}  Daha sonra onLocationChanged içerisine ;  SharedPreferences sharedPreferences=MapsActivity.this.getSharedPreferences("com.example.hamit.seyahatkitabim",*MODE\_PRIVATE*); boolean firstTimeCheck = sharedPreferences.getBoolean("notFirstTime",false);///kullanıcı bu app'i ilk kez kullanıyorsa false dönecek.  if (!firstTimeCheck) {  LatLng userLocation = new LatLng(location.getLatitude(),location.getLongitude());  mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(userLocation,15));  sharedPreferences.edit().putBoolean("notFirstTime",true).apply(); }  Bu sayede kullanıcı harita üzerinde rahatça gezinebilecek ve harita her seferinde kullanıcıyı kayıtlı enlem ve boylama yönlendirmeyecektir. | |

**En son lokasyonun tekrar açılışta programda gösterilmesi,adresleri almak**

Programımızda en son lokasyonun bir sonraki açılışta karşımıza gelebilmesi için ve çökmeleri engellemek için;

if (Build.VERSION.*SDK\_INT* >= 23) {  
 if (checkSelfPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 requestPermissions(new String[] {Manifest.permission.*ACCESS\_FINE\_LOCATION*},1);  
 } else {  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*,0,0,locationListener);  
 mMap.clear();  
 Location lastLocation = locationManager.getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);  
 if (lastLocation != null) {  
 LatLng lastUserLocation = new LatLng(lastLocation.getLatitude(),lastLocation.getLongitude());  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(lastUserLocation,15));  
 }  
 }  
} else {  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*,0,0,locationListener);  
 Location lastLocation = locationManager.getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);  
 if (lastLocation != null) {  
 LatLng lastUserLocation = new LatLng(lastLocation.getLatitude(),lastLocation.getLongitude());  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(lastUserLocation,15));  
 }

Adresleri almak için

public class MapsActivity extends FragmentActivity implements onMapReadyCallback,GoogleMap.OnMapLongClickListener

satırı bu şekilde düzenlenir.Daha sonra onMapLongClick fonksiyonu projemize eklenir.Bu fonksiyonda kullanıcı ekrana uzun süre basılı tuttuğu durumda ne işlem yapılacağını yazacağız.Bunu yapabilmek için mMap değişkenimizi onMapReady altında eşleriz.bunun için;

mMap.setOnMapLongClickListener(this); yazarız.

onMapLongClick içerisine;

Geocoder geocoder=new Geocoder(getApplicationContext(), Locale.*getDefault*());  
String adres="";  
try {List<Address> addressList=geocoder.getFromLocation(latLng.latitude,latLng.longitude,1);  
 if(addressList!=null&&addressList.size()>0)  
 {if (addressList.get(0).getThoroughfare()!=null)  
 { +=addressList.get(0).getThoroughfare();  
 if(addressList.get(0).getSubThoroughfare()!=null)  
 {  
 adres+=addressList.get(0).getSubThoroughfare();}}  
 }else {  
 adres="Yeni yer";}}  
catch (IOException e) {  
 e.printStackTrace();

}

addressList adında bir liste oluşturulur buraya,uzun süre tıklanıldığında tıklanan yerin enlem ve boylam bilgileri kaydedilir.

**Marker eklenmesi Alınan enlem ve boylam ve kaydedilen yerin isim değerlerini sqlLite veritabanına kaydetmek**

marker ekleyebilmek için ise onMapLongClick içerisine aşağıdaki ;

mMap.addMarker(new MarkerOptions().title(adres).position(latLng));  
Toast.*makeText*(getApplicationContext(),"Yeni yer eklendi",Toast.*LENGTH\_SHORT*).show();

kodu eklenir bu sayede marker eklenmiş olur.

Projemizde MapsActivity.java içerisinde farklı yerlerden ulaşabilmek için static ve global olarak database nesnemizi oluştururuz.

static SQLiteDatabase *database*;

onMapLongClick'in içerisinde veritabanı oluşturma kodlarımızı yazarız.

try  
{  
 Double l1=latLng.latitude;  
 Double l2=latLng.longitude;  
  
 String coord1=l1.toString();  
 String coord2=l2.toString();  
  
 *database*=this.openOrCreateDatabase("Yerler",*MODE\_PRIVATE*,null);  
 *database*.execSQL("CREATE TABLE IF NOT EXISTS yerler(isim VARCHAR,enlem VARCHAR,boylam VARCHAR )");  
  
 String toCompile="INSERT INTO yerler (isim,enlem,boylam) VALUES (?,?,?)";  
 SQLiteStatement sqLiteStatement =*database*.compileStatement(toCompile);  
  
 sqLiteStatement.bindString(1,adres);  
 sqLiteStatement.bindString(2,coord1);  
 sqLiteStatement.bindString(3,coord2);  
  
 sqLiteStatement.execute();  
}catch (Exception e)  
{  
 e.printStackTrace();  
}

**Kaydedilen verilerin veritabanından çekilmesi**

Verileri çekebilmek için önce MainActivity içerisinde global olarak;

static ArrayList<String > *isimler*=new ArrayList<String>();  
static ArrayList<LatLng> *yerler*=new ArrayList<LatLng>();  
static ArrayAdapter *arrayAdapter*;

Arraylistlerimizi oluştururuz.

Daha sonra onCreate'in altında listview tanımlamamızdan hemen sonra.MapsActivity içerisindeki database'e erişim sağlanır.daha sonra isim,enlem,boylamlar in kolonİndex değerleri alınır.cursor aracılığıyla sıra sıra gezinilir.Elde edilen veriler sırasıyla string değişkenlere atanır.daha sonra elde ettiğimiz string değerleri arraylistelere ekleriz.enlem ve boylamı double'a çeviririz.bunları LatLng türüne dönüştürürüz daha sonra yerler.add(veritabanıYerler) diyerek latlng türündeki paketimizi arraylistimize ekleriz.

try  
{

MapsActivity.*database*=this.openOrCreateDatabase("Yerler",*MODE\_PRIVATE*,null);  
 Cursor cursor= MapsActivity.*database*.rawQuery("SELECT \* FROM yerler",null);  
 int isimİndex=cursor.getColumnIndex("isim");  
 int enlemİndex=cursor.getColumnIndex("enlem");  
 int boylamİndex=cursor.getColumnIndex("boylam");

while (cursor.moveToNext())  
 {  
 String veritabanıİsim=cursor.getString(isimİndex);  
 String veritabanıEnlem=cursor.getString(enlemİndex);  
 String veritabanıBoylam=cursor.getString(boylamİndex);  
 *isimler*.add(veritabanıİsim);  
 Double L1=Double.*parseDouble*(veritabanıEnlem);  
 Double L2=Double.*parseDouble*(veritabanıBoylam);  
  
 LatLng veritabanıYerler=new LatLng(L1,L2);  
 *yerler*.add(veritabanıYerler);  
  
 }

cursor.close();  
}catch (Exception e)  
{  
 e.printStackTrace();  
}

Burdan itibaren ise yazılan kodlarda arrayadapter sayesinde veritabanımızdaki verileri listview'imizde göstermemizi sağlar

*arrayAdapter* =new ArrayAdapter(this,android.R.layout.*simple\_list\_item\_1*,*isimler*);  
listView.setAdapter(*arrayAdapter*)

**Listviewde tıklanılan yere yönlendirme yapılması,ekledikten hemen sonra listview'de eklenen adresin gözükmesi**

onCreate fonksiyonunun altında setOnItemClickListener fonksiyonu yazılır bunun sayesinde kullanıcı listviewdeki bir adresemi tıklayıp haritayı açmak istedi yoksa yeni bir yer eklemek isteyiptemi haritayı açmak istedi onun ayrımını yapmamızı sağlar.yani lisview de tıkladığında kayıtlı adresin gelmesini isteriz yer ekleye tıkladığında ise haritada bulunduğu konumu almasını isteriz.

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent ıntent=new Intent(getApplicationContext(),MapsActivity.class);  
 ıntent.putExtra("bilgi","eski");  
 ıntent.putExtra("pozisyon",i);  
 startActivity(ıntent);  
 }

MapsActivity içerisinde onMapReady'nin kod bloğu aşağıdaki şekilde düzenlenir.Eğer bilgi yeni ise ekleme yapılır değilse tıklanılan yer listview üzerinde bir yer listviewde nereye tıklanıldığı tespit edilir.Ondan sonra o koordinata gidilir.

@Override  
public void onMapReady(GoogleMap googleMap) {  
 mMap = googleMap;  
 mMap.setOnMapLongClickListener(this);  
  
 Intent intent = getIntent();  
 String info = intent.getStringExtra("bilgi");  
 if (info.matches("yeni")) {  
 locationManager = (LocationManager) this.getSystemService(Context.*LOCATION\_SERVICE*);  
 locationListener = new LocationListener() {  
 @Override  
 public void onLocationChanged(Location location) {  
 SharedPreferences sharedPreferences = MapsActivity.this.getSharedPreferences("com.example.hamit.seyahatkitabim", *MODE\_PRIVATE*);  
 boolean firstTimeCheck = sharedPreferences.getBoolean("notFirstTime", false);  
  
 if (!firstTimeCheck) {  
 LatLng userLocation = new LatLng(location.getLatitude(), location.getLongitude());  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(userLocation, 15));  
 sharedPreferences.edit().putBoolean("notFirstTime", true).apply();  
 }  
 }  
 @Override  
 public void onStatusChanged(String provider, int status, Bundle extras) {  
 }  
 @Override  
 public void onProviderEnabled(String provider) {  
 }  
 @Override  
 public void onProviderDisabled(String provider) {  
 }  
 };  
 if (Build.VERSION.*SDK\_INT* >= 23) {  
 if (checkSelfPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 requestPermissions(new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, 1);  
 } else {  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*, 0, 0, locationListener);  
  
 mMap.clear();  
  
 Location lastLocation = locationManager.getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);

if (lastLocation != null) {  
 LatLng lastUserLocation = new LatLng(lastLocation.getLatitude(), lastLocation.getLongitude());  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(lastUserLocation, 15));  
 }}  
 } else {  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*, 0, 0, locationListener);  
 Location lastLocation = locationManager.getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);  
 if (lastLocation != null) {  
 LatLng lastUserLocation = new LatLng(lastLocation.getLatitude(), lastLocation.getLongitude());  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(lastUserLocation, 15));  
 }}}

Main activity içerisinde kaçıncıya tıklandığını belirleyerek o lokasyona giden kod bloğu yazıldı.

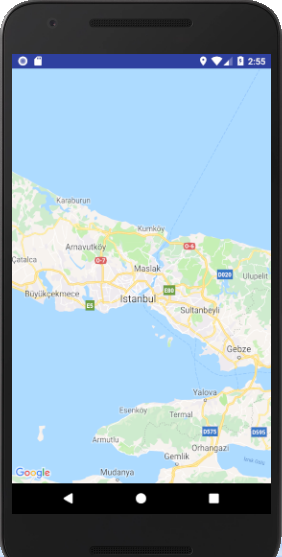
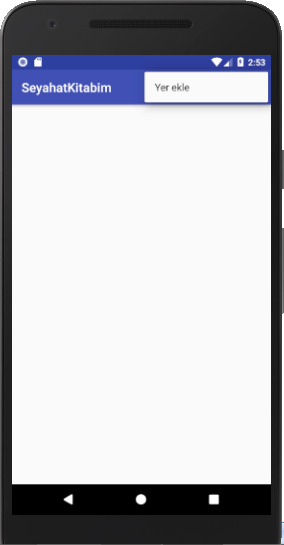
else {  
 mMap.clear();  
 int position = intent.getIntExtra("position",0);  
 LatLng location = new LatLng(MainActivity.*yerler*.get(position).latitude,MainActivity.*yerler*.get(position).longitude);  
 String placeName = MainActivity.*isimler*.get(position);  
  
 mMap.addMarker(new MarkerOptions().title(placeName).position(location));  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(location,15));  
  
}

Kaydedilen adresin geri dönüldüğünde uygulama yeniden başlatmaya ihtiyaç duyulmadan listviewe eklenmesini sağlamak için,daha doğrusu arrayadapterden güncelleme yapmasını istemek için;

MainActivity.*isimler*.add(adres);  
MainActivity.*yerler*.add(latLng);  
MainActivity.*arrayAdapter*.notifyDataSetChanged();

Ekleme bloğundan yani try,catch'den hemen önce yukarıdaki kod yazılır.

**Projeden Görseller**



**Survivor Bird Uygulaması**

Android üzerinde çalışan Survivor Bird isimli oyun projesinin yapım aşamalarıdır.

**Proje İçeriği:**

Oyun flappy bird oyunu tarzında uçan bir kuşumuz olacak yer çekimi etkisiyle aşağıya doğru düşerken kullanıcımızın ekrana her tıkladığında belli bir yüksekliğe çıkar,sonra yer çekimi etkisiyle tekrar aynı şekilde aşağıya doğru iner.Bu döngüsel olarak devam eder.Bir yandanda kullanıcımız random bir şekilde gelen arılara temas etmekten kaçınmak zorundadır.Her geçtiği arı kümesinden sonra skor tabelası 1 artmaktadır.Arılara temas ettiği takdirde veya yere düştüğü zaman yanmaktadır ve oyun sonlanmaktadır.

Kullanılacak Dil:Java

Kullanılacak Framework:libGdx

Kullanılacak Editör:Android Studio

Kullanılan API:API26:Android 8.0(OREO)

**Projede elde edilmesi planlanan kazanımlar :**

Android studio üzerinde basit bir uygulama dahi olsa oyun yazma tecrübesi elde edilmesi.

LibGdx kullanılarak farklı bir teknoloji konusunda fikir sahibi olabilmek.

Android studio projelerine farklı teknolojilerin nasıl entegre edildiği ve nasıl birlikte çalıştırıldığının öğrenilmesi.

**LibGdx hakkında araştırma yapılması.Kurulumunun yapılması**

**LibGDX Nedir ?**

LibGDX farklı cihazlarda çalışabilen 2D ve 3D oyun geliştirebileceğimiz açık kaynaklı (open source)  Cross Platform (Android, iOS, Blackberry, Desktop, HTML5) çatıdır (framework).

# LibGDX Nasıl Kurulur ?

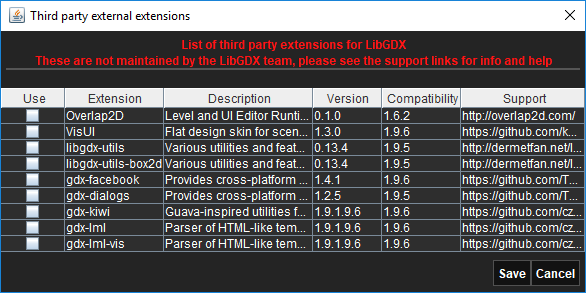
Şimdi de nasıl kurulacağına değinelim. LibGDX kurulumu için yapmanız gereken pek bir işlem yoktur. Bilgisayarınızda Java JDK kurulu ise LibGDX kütüphanesini https://libgdx.badlogicgames.com/download.html adresinden indirerek çalıştırmanız yeterli.  Ardından aşağıdaki ekran açılacaktır.



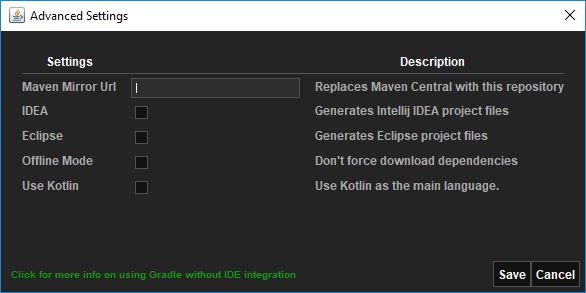
Bu ekranda “Name” kısmına projenizin adını, “Package” kısmında ana paketin adını, “Game class” kısmına main class adını, “Destination” kısmına projenin nerede oluşturulacağını, “Android SDK” kısmına sdk yolumuzu giriyoruz. Ardından “Sub Projects” bölümündeki Desktop, Android, iOS, iOS-MOE ve Html seçeneklerinden android'i seçiyoruz. Son olarak Extensions kısmından ihtiyacımız olan eklentileri seçebiliriz bizim bu projede buradaki eklentilere ihtiyacımız bulunmamaktadır bu sebeple herhangi bir seçeneğe tıklamamız gerekmemektedir. En son “Generate” butonuna tıklarız. BUILD SUCCESSFUL'u uyarısını gördüğümüz anda projemiz başarıyla oluşturulmuş demektir.Böylelikle oyun projesini oluşturmuş oluyoruz.

**Detay Bilgiler:**

Burada dikkat edilmesi gereken kısım “Game class” kısmını doldururken Java dil kurallarına dikkat etmeniz. Boşluk ve özel karakter kullanırsanız projenizi bir editörle açtığınızda her yerden bir hata mesajı çıkacaktır. Ek olarak “Destination” kısmında projenizin yolunu yazarken sonuna mutlaka projenizin klasörünün adını da yazın. Örneğin “D:\Projelerim” klasöründe tüm projelerinizi geliştiriyorsanız yolunu “D:\Projelerim\YeniOyunum” yapın. Eğer böyle yapmazsanız LibGDX projeyi oluştururken projenin dosyalarını  “D:\Projelerim” klasörüne oluşturacak ve diğer projelerinizle karışacaktır.Ek olarak eğer ana ekranda görünenlerin dışında diğer eklentilere de ihtiyaç duyuyorsanız “Show Third Party Extensions” butonuna tıklayarak aşağıdaki ekranı açabilir ve istediğiniz eklentiyi seçebilirsiniz.

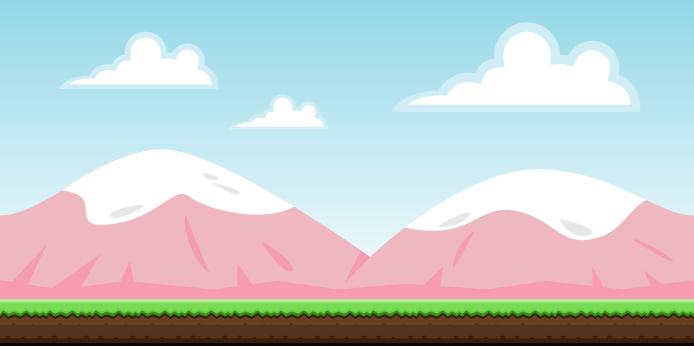


Son olarak eğer isterseniz “Advanced” butonuna tıklayarak oluşturulacak projenin gelişmiş ayarlarını da yapabilirsiniz. Bu ekranda Maven url eklemenize, Kotlin desteği eklemenize veya doğrudan bir editörün projesi gibi oluşturmanıza imkan sağlanıyor. Değişiklikleri yaptıktan sonra “Save” butunu ile kaydedip bu pencereyi kapatıp ana ekrandaki “Generate” butonu ile projenizi oluşturabilirsiniz.



**Oyun görsellerinin bulunması.Tasarımın yapılması**

Oyun görselleri için https://opengameart.org sitesinden ücretsiz olarak 1 kuş 1 arı ve arka plan için 1 adet manzara resmi alındı..



libGdx aracılığıyla oluşturduğumuz uygulamamızı android studioda file open ile açtığımızda karşımıza project'in altında android dosyası haricinde core isimli bir dosya çıkacaktır.Core'un altında java klasörünün içinde bizim ismini verdiğimiz survivorBird isimli classa kodlarımızı yazarız.

İndirdiğimiz görselleri android studio içinde project penceresinde android'in altındaki assets klasörünün içerisine atılır.

Oyun projelerinde resimleri veya başka görselleri projeye ekleyebilmek için SpriteBatch sınıfından bir obje oluşturmamız gerekir;

survivorBird classının içinde global olarak SpriteBatch batch; isimli bir obje tanımlarız.

public class survivorbird extends ApplicationAdapter {

SpriteBatch batch;

}

Daha sonra create içinde yazdığımız nesneyi başlatırız.Create fonksiyonu normal uygulamalardaki onCreate metoduyla aynı görevi görmektedir.Uygulama başlarken çalışması gerekenleri yazmamız gereken kısımdır.

@Override

public void create () {

batch=new SpriteBatch();}

Projemizdeki render metodu ise oyun devam ettiği sürece devamlı çağrılan bir metoddur.Sürekli olması istenen şeyler buraya yazılır.Bunun için batch nesnemizi burada başlatıp bitirmek zorundayız.

@Override

public void render () {

batch.begin();

batch.end();}

Bundan sonra görsellerimizi koyabilmek için Texture kavramını kullanmamız gerekmektedir.bunun içinde global olarak örneğin arka plan için Texture background tanımlarız.

public class survivorbird extends ApplicationAdapter {  
 SpriteBatch batch;  
 Texture background;}

Create içinde;

@Override

public void create () {

batch=new SpriteBatch();

background=new Texture("mountain.png"); yaparak arka planımız background'a atarız

Arka planımızı projemizde çizdirebilmek için son olarak render içerisinde draw özelliğini kullanırız;

@Override  
public void render () {  
 batch.begin();  
 batch.draw(background,0,0, Gdx.*graphics*.getWidth(),Gdx.*graphics*.getHeight());

batch.end();}

Burada 0,0 çizilecek görselin x,y eksenindeki başlangıç noktasını temsil etmekte,

Gdx.*graphics*.getWidth(),Gdx.*graphics*.getHeight() ise genişlik ve yüksekliği temsil eder.

Kuş görselinide eklemek için aynı işlemler yapılır.Farklı olarak sadece konumlandırma farklı olacaktır yani x,y ekseninde 0,0 noktasından başlatılmayacaktır.Başlatılsa dahi oyun içerisinde y ekseni sürekli değişmek zorunda olduğu için belli bir değişkene tanımlayıp onun üzerinden işlem yapmak daha mantıklıdır.

public class survivorbird extends ApplicationAdapter {  
 SpriteBatch batch;  
 Texture background;  
 Texture bird;}

@Override  
public void create () {  
batch=new SpriteBatch();  
background=new Texture("mountain.png");  
bird=new Texture("bird.png");}

Eksenler üzerindeki değerlerle oynama yapabilmek için 2 adet değişken tanımlanır.

public class survivorbird extends ApplicationAdapter {  
 SpriteBatch batch;  
 Texture background;  
 Texture bird;  
 float birdX=0;  
 float birdY=0;

@Override  
public void create () {  
batch=new SpriteBatch();  
background=new Texture("mountain.png");  
birdX=Gdx.*graphics*.getWidth()/2-bird.getHeight()/2;//kuşun x üzerindeki konumu  
birdY=Gdx.*graphics*.getHeight()/3;}//kuşun y üzerindeki konumu

Render içerisine;

batch.draw(bird,birdX,birdY,Gdx.*graphics*.getWidth()/15,Gdx.*graphics*.getHeight()/10);

kodu yazılarak kuşumuz çizilmiş olur.

**Oyunda yerçekimi oluşturulması.Kuşa uçma yeteneği eklemek.Arıları oyunumuza dahil etmek.Çarpışmaları anlamak**

Oyunun başlayıp başlamadığını kontrol etmek için gamestate adında bir değişken oluşturulur.

public class survivorbird extends ApplicationAdapter {  
  
 SpriteBatch batch;  
 Texture background;  
  
 float birdX=0;  
 float birdY=0;  
  
 int gameState=0;

float velocity=0;

float gravity=0.1f;

}

render metodu içerisinde gamestate'in kontrolü sağlanır.

if (gameState == 1)  
{

if(Gdx.*input*.justTouched())  
{  
 velocity = -7;  
}

if(birdY>0|| velocity<0)  
{

velocity= velocity+ gravity;

birdY= birdY- velocity;

}

} else

{

if(Gdx.*input*.justTouched())  
{  
 gameState = 1;  
}

}

Bu sayede kuşa uçma yeteneği verilmiş olur.

Arı görsellerini eklemek için batch ile çizdirmemiz gerekir.Texture ile yaratırız.

public class survivorbird extends ApplicationAdapter {  
 SpriteBatch batch;  
 Texture background;  
 Texture bird;  
 Texture bee1;  
 Texture bee2;  
 Texture bee3;  
 float birdX=0;  
 float birdY=0;  
 int gameState=0;  
 float velocity=0;  
 float gravity=0.1f;}

Create metodu içerisinde,tanımladığımız Texture değişkenlere görselleri atarız.

@Override  
public void create () {  
batch=new SpriteBatch();  
background=new Texture("mountain.png");  
bird=new Texture("bird.png");  
bee1=new Texture("bee.png");  
bee2=new Texture("bee.png");  
bee3=new Texture("bee.png");}

3 lü setler halinde 4 adet arı seti oluşturmak için dizilerden faydalanırız.x,y eksenlerini ayarlamak için 1 adet x ekseni 3 adette y ekseni üzerinde oynama yapabilmek için toplamda 4 adet değişkene ihtiyacımız var. Kuşla arının birbiriyle temasını anlamak için daireler içine alacağız bundan dolayı 1 tane birdCircle ve 3 tane enemyCircle değişkenine ihtiyaç olacak bunları çizdirmek için ise shapeRenderer sınıfından faydalanacağız.4 lü setleri random getirmek içinde Random türünden bir değişkene ihtiyaç duyacağız.

public class survivorbird extends ApplicationAdapter {  
 SpriteBatch batch;  
 Texture background;  
 Texture bird;  
 Texture bee1;  
 Texture bee2;  
 Texture bee3;  
 float birdX=0;  
 float birdY=0;  
 int gameState=0;  
 float velocity=0;  
 float gravity=0.1f;  
 float enemyVelocity=2;  
 Random rndm;  
 Circle birdCircle;  
 ShapeRenderer shapeRenderer;  
 int numberOfEnemies=4;  
 float [] enemyX=new float[numberOfEnemies];  
 float [] enemyOffset=new float[numberOfEnemies];  
 float [] enemyOffset2=new float[numberOfEnemies];  
 float [] enemyOffset3=new float[numberOfEnemies];  
 float distance=0;  
 Circle[] enemyCircle;  
 Circle[] enemyCircle2;  
 Circle[] enemyCircle3;

render fonksiyonunu bu şekilde düzenliyoruz.

@Override  
 public void render () {  
 batch.begin();  
 batch.draw(background,0,0, Gdx.*graphics*.getWidth(),Gdx.*graphics*.getHeight());  
 if (gameState == 1)  
 {  
 if(Gdx.*input*.justTouched())  
 {  
 velocity = -7;  
 }  
 for(int i=0;i<numberOfEnemies;i++)  
 {  
 if(enemyX[i]<Gdx.*graphics*.getWidth()/15)  
 {  
 enemyX[i]=enemyX[i]+numberOfEnemies\*distance;  
 enemyOffset[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 enemyOffset2[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 enemyOffset3[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 }else  
 {  
 enemyX[i]=enemyX[i]-enemyVelocity; }batch.draw(bee1,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);  
 batch.draw(bee2,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset2[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);  
 batch.draw(bee3,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset3[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);

enemyCircle[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+enemyOffset[i]+

Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);  
 enemyCircle2[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+enemyOffset2[i]+

Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);  
 enemyCircle3[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+enemyOffset3[i]+

Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);  
 }

if(birdY>0)  
 { velocity=velocity+gravity;  
 birdY=birdY-velocity;  
 }else  
 {  
 gameState=2;  
 }}  
 else if(gameState==0)  
 {  
 if(Gdx.*input*.justTouched())  
 {  
 gameState = 1;  
 }  
 }else if(gameState==2)  
 {  
 if(Gdx.*input*.justTouched())  
 {  
 gameState = 1;  
 birdY=Gdx.*graphics*.getHeight()/3;  
  
 for (int i=0;i<numberOfEnemies;i++)  
 {  
 enemyOffset[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset2[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset3[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyX[i]=Gdx.*graphics*.getWidth()-bee1.getWidth()/2\*i\*50;  
  
 enemyCircle[i]=new Circle();  
 enemyCircle2[i]=new Circle();  
 enemyCircle3[i]=new Circle();  
 }

velocity=0;  
 }  
 }  
 batch.draw(bird,birdX,birdY,Gdx.*graphics*.getWidth()/15,Gdx.*graphics*.getHeight()/10);  
 batch.end();  
birdCircle.set(birdX+Gdx.*graphics*.getWidth()/30,birdY+Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);

for(int i=0;i<numberOfEnemies;i++)  
 {  
 if(Intersector.*overlaps*(birdCircle,enemyCircle[i])||Intersector.*overlaps*(birdCircle,enemyCircle2[i])||

Intersector.*overlaps*(birdCircle,enemyCircle3[i]))*//çarpışmaları anlamak için yapılan kontrol* {  
 gameState=2;  
 }}}

Create fonksiyonunu ise bu şekilde düzenliyoruz.

@Override  
public void create () {  
  
batch=new SpriteBatch();  
background=new Texture("mountain.png");  
bird=new Texture("bird.png");  
bee1=new Texture("bee.png");  
bee2=new Texture("bee.png");  
bee3=new Texture("bee.png");  
  
shapeRenderer=new ShapeRenderer();  
  
birdX=Gdx.*graphics*.getWidth()/2-bird.getHeight()/2;  
birdY=Gdx.*graphics*.getHeight()/3;  
  
birdCircle=new Circle();  
enemyCircle=new Circle[numberOfEnemies];  
enemyCircle2=new Circle[numberOfEnemies];  
enemyCircle3=new Circle[numberOfEnemies];  
distance=Gdx.*graphics*.getWidth();  
rndm=new Random();

for (int i=0;i<numberOfEnemies;i++)  
{  
 enemyOffset[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset2[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset3[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyX[i]=Gdx.*graphics*.getWidth()-bee1.getWidth()/2\*i\*50;  
  
 enemyCircle[i]=new Circle();  
 enemyCircle2[i]=new Circle();  
 enemyCircle3[i]=new Circle();

}

}

**Skorlama yapmak.Yanınca bitti uyarısı vermek**

Skor hesaplamak için 1 adet score değişkeni 1 adette scoreEnemy değişkenine ihtiyaç vardır.Ayrıyeten skoru göstermek için 1 adet font değişkenine oyunun bittiğine dair uyarı vermek için ise 1 adet daha font değişkenine ihtiyacımız vardır.Değişkenlerimiz bu şekilde olacak;

public class survivorbird extends ApplicationAdapter {  
  
 SpriteBatch batch;  
 Texture background;  
 Texture bird;  
 Texture bee1;  
 Texture bee2;  
 Texture bee3;  
 float birdX=0;  
 float birdY=0;  
 int gameState=0;  
 float velocity=0;  
 float gravity=0.1f;  
 float enemyVelocity=2;  
 Random rndm;  
 int score=0;  
 int scoreEnemy=0;  
 BitmapFont font;  
 BitmapFont font2;  
 Circle birdCircle;  
 ShapeRenderer shapeRenderer;  
 int numberOfEnemies=4;  
 float [] enemyX=new float[numberOfEnemies];  
 float [] enemyOffset=new float[numberOfEnemies];  
 float [] enemyOffset2=new float[numberOfEnemies];  
 float [] enemyOffset3=new float[numberOfEnemies];  
 float distance=0;  
 Circle[] enemyCircle;  
 Circle[] enemyCircle2;  
 Circle[] enemyCircle3;

Create fonksiyonumuz şu şekilde olacaktır.

@Override  
public void create () {  
  
batch=new SpriteBatch();  
background=new Texture("mountain.png");  
bird=new Texture("bird.png");  
bee1=new Texture("bee.png");  
bee2=new Texture("bee.png");  
bee3=new Texture("bee.png");  
shapeRenderer=new ShapeRenderer();  
birdX=Gdx.*graphics*.getWidth()/2-bird.getHeight()/2;  
birdY=Gdx.*graphics*.getHeight()/3;  
birdCircle=new Circle();  
enemyCircle=new Circle[numberOfEnemies];  
enemyCircle2=new Circle[numberOfEnemies];  
enemyCircle3=new Circle[numberOfEnemies];  
distance=Gdx.*graphics*.getWidth();  
rndm=new Random();  
font=new BitmapFont();  
font.setColor(Color.*WHITE*);  
font.getData().setScale(4);  
  
font2=new BitmapFont();  
font2.setColor(Color.*WHITE*);  
font.getData().setScale(6);  
  
  
  
for (int i=0;i<numberOfEnemies;i++)  
{  
 enemyOffset[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset2[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset3[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyX[i]=Gdx.*graphics*.getWidth()-bee1.getWidth()/2\*i\*50;  
  
 enemyCircle[i]=new Circle();  
 enemyCircle2[i]=new Circle();  
 enemyCircle3[i]=new Circle();  
}}

Render fonksiyonumuz ise şu şekilde olacaktır.

@Override  
 public void render () {  
 batch.begin();  
 batch.draw(background,0,0, Gdx.*graphics*.getWidth(),Gdx.*graphics*.getHeight());  
 if (gameState == 1)  
 {  
 if(enemyX[scoreEnemy]<Gdx.*graphics*.getWidth()/2-bird.getHeight()/2)  
 {  
 score++;  
 if(scoreEnemy<numberOfEnemies-1)  
 {  
 scoreEnemy++;  
 }else  
 {  
 scoreEnemy=0;  
 }}  
 if(Gdx.*input*.justTouched())  
 {  
 velocity = -7;  
 }  
 for(int i=0;i<numberOfEnemies;i++)  
 {  
 if(enemyX[i]<Gdx.*graphics*.getWidth()/15)  
 {  
 enemyX[i]=enemyX[i]+numberOfEnemies\*distance;  
 enemyOffset[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 enemyOffset2[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 enemyOffset3[i]=(rndm.nextFloat()-0.5f)\*(Gdx.*graphics*.getHeight()-200);  
 }else  
 {  
 enemyX[i]=enemyX[i]-enemyVelocity;  
 }batch.draw(bee1,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);  
 batch.draw(bee2,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset2[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);  
 batch.draw(bee3,enemyX[i],Gdx.*graphics*.getHeight()/2+enemyOffset3[i],Gdx.*graphics*.getWidth()/15,

Gdx.*graphics*.getHeight()/10);

enemyCircle[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+

enemyOffset[i]+Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);

enemyCircle2[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+

enemyOffset2[i]+Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);

enemyCircle3[i]=new Circle(enemyX[i]+Gdx.*graphics*.getWidth()/30,Gdx.*graphics*.getHeight()/2+

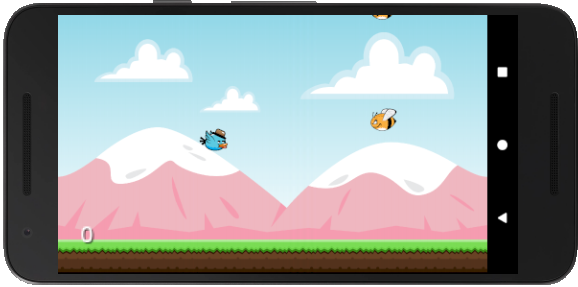
enemyOffset3[i]+Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);  
 }  
 if(birdY>0)  
 {  
 velocity=velocity+gravity;  
 birdY=birdY-velocity;  
 }else  
 {  
 gameState=2;  
 }  
 }  
 else if(gameState==0)  
 {  
 if(Gdx.*input*.justTouched())  
 {  
 gameState = 1;  
 }  
 }else if(gameState==2)  
 {  
 font2.draw(batch,"Game over Tap to play again ",Gdx.*graphics*.getWidth()/2,Gdx.*graphics*.getHeight()/3);  
 if(Gdx.*input*.justTouched())  
 {  
 gameState = 1;  
 birdY=Gdx.*graphics*.getHeight()/3;  
  
 for (int i=0;i<numberOfEnemies;i++)  
 {  
 enemyOffset[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset2[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyOffset3[i]=(rndm.nextFloat())\*(Gdx.*graphics*.getHeight());  
 enemyX[i]=Gdx.*graphics*.getWidth()-bee1.getWidth()/2\*i\*50;  
  
 enemyCircle[i]=new Circle();  
 enemyCircle2[i]=new Circle();  
 enemyCircle3[i]=new Circle();  
 }  
 velocity=0;  
 score=0;  
 scoreEnemy=0;  
 }  
 }  
 batch.draw(bird,birdX,birdY,Gdx.*graphics*.getWidth()/15,Gdx.*graphics*.getHeight()/10);  
 font.draw(batch,String.*valueOf*(score),100,200);  
 batch.end();  
 birdCircle.set(birdX+Gdx.*graphics*.getWidth()/30,birdY+

Gdx.*graphics*.getHeight()/20,Gdx.*graphics*.getWidth()/30);  
 for(int i=0;i<numberOfEnemies;i++)  
 {  
 if(Intersector.*overlaps*(birdCircle,enemyCircle[i])||Intersector.*overlaps*

(birdCircle,enemyCircle2[i])||Intersector.*overlaps*

(birdCircle,enemyCircle3[i]))*//çarpışmaları anlamak için yapılan kontrol* {  
 gameState=2;  
 }  
 }  
}

**Projeden Görseller**



**Ders Notlarım**

Android üzerinde çalışan Ders Notlarım isimli projenin yapım aşamalarıdır.

**Proje İçeriği:**

Kullanıcı uygulamada cihazın kamerası yardımıyla ders notunun fotoğrafını çekebilecek.

Çektiği nota kendisi isim verebilecek örneğin:Algoritma ve Programlama 1 gibi.

Daha sonradan kullanıcı istediği zaman kaydettiği yere istediği zaman tıkladığında kaydettiği notu görebilecek.

Uygulama fotoğrafları kendi içerisinde saklayacak bu sayede kişisel telefonunda karmaşıklığa sebep olan not fotoğrafların kurtarılmış olacak.

Kullanılacak Dil:Java

Kullanılacak Editör:Android Studio

Kullanılan API:API23:Android 6.0(MARSHMALLOW)

**Projede elde edilmesi planlanan kazanımlar :**

Tasarım ekranı yapılması sayesinde Android studioda tasarım öğrenilmesi ve tasarım tecrübesi elde edilmesi.

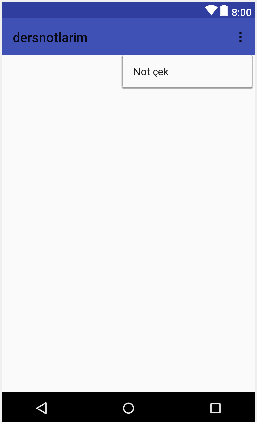
Sqlite veritabanı kullanılmasıyla Android platformu üzerinde veritabanı yaratılması ve yönetimi öğrenilmesi.

Android uygulamalarında kamera bağlantısı,kamera kullanılması için gerekli izinlerin nasıl alındığının öğrenilip tecrübe edilmesi.

**Projenin oluşturulması ve tasarım Yapılması**

Android studioyu açıldığında Start a new Android Studio Projecte tıklanır.Activity ekleme ekranında Empty Activity seçilir.Proje oluşturulur.

Bunları yaptıktan sonra ilk olarak activity\_main.xml dosyasına girilip Paletteden sürükle bırak ile bir listView eklendi.ListView e ID olarak listView verildi.Projemizde tanıtılması için onCreate metodu altında



ListView listView=(ListView)findViewById(R.id.*listView*); diyerek tanımlarız.

Daha sonra, menü ekleyebilmek için app'nin altında res klasörüne sağ tıklanıp Directory'e tıklanır.İsim olarak "menu" verilir.Daha sonra menu klasörüne sağ tıklanıp Menu resource file'a tıklanıp gelen ekranda isim verilir.isim olarak not\_ekle verdik.

not\_ekle.xml in Text kısmına

<item android:id="@+id/not\_ekle" android:title="Not çek"></item>

yazılır.

Elde edilen görüntü yandaki gibi olur.

MainActivity.java içinde onCreateOptionsMenu yazılarak override olan onCreateOptionsMenu fonksiyonumuzu oluştururuz.Menümüzü bağlamamızı sağlar.

İçerisinede;

MenuInflater menuInflater=getMenuInflater();  
 menuInflater.inflate(R.menu.*not\_ekle*,menu);

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater menuInflater=getMenuInflater();  
 menuInflater.inflate(R.menu.*not\_ekle*,menu);  
 return super.onCreateOptionsMenu(menu);  
}

Yine aynı şekilde MainActivity.java içerisinde onOptionsItemSelected yazarak başka bir override methodumuzu oluştururuz buda menüye tıklandığında ne olacağını yazacağımız fonksiyondur.

Daha sonra bir tane daha empty activity ekliyoruz projemize adını Main2Activity koyuyoruz.

*//menüye tıklandığında ne olacağıyla ilgili yapılan işlemleri*

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 if(item.getItemId()==R.id.*not\_ekle*)  
 {   
 Intent ıntent=new Intent(getApplicationContext(),Main2Activity.class);  
 startActivity(ıntent);  
 }  
 return super.onOptionsItemSelected(item);

}

Buraya daha sonra Intent Yardımıyla kamera bağlanacak.

**2. Activity Tasarımı(Main2Activity) tasarımı**

2. activityimizin tasarımını yapmak için 1 imageview 1 editText 1 tanede buton kullandık.Butona id olarak btnKaydet imageview'e imageview editText'e ise editTextName id verdik.

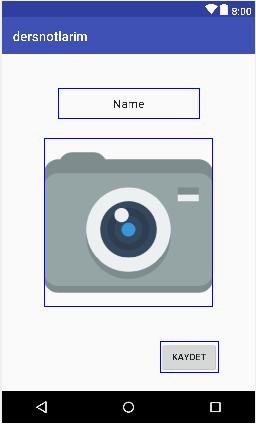
İmageview kamera resmini koyabilmek için projemizdeki res klasörünün altındaki drawable klasörüne sağ tıklanır.Show in explorer'a tıklanır.

Çıkan klasörün içerisine projeyle ilgili istenilen görsel atılır.

Daha sonra imageview projeye konur konduktan sonra imageviewin projedeki standart görünümü için android studio bir resim veya renk seçilmesini ister burada yönlendirilen sayfada projectin altında koyduğumuz resmi seçeriz ve android studio imageviewimize bu görseli koyar.

Main2Activity içerisinde kaydet ve sec isimli 2 metod yazdık.

Butonun onclick özelliğine kaydet imageview'in onclick özelliğinede sec yazdık.Bu sayede bunlar button ve imageview'imizle bağlı çalışacaklar.

public void kaydet(View view)  
{  
  
}

public void sec(View view)  
{  
  
}

**Kullanıcı izinlerinin alınması**

Android uygulamalarda apı 23 ten itibaren kullanıcının cihazında galeri kamera vs.. kişisel bilgi veya belgelere erişim izin izin alınması gerekmektedir.Uygulamamızda kamera kullanacağımız için öncelikle kullanıcıdan buna dair bir izin almamız gerekmektedir.İzinleri AndroidManifest.xml üzerinde yazacağımız bir kaç satır kodla alırız.

**Android Manifest nedir?**

AndroidManifest.xml, tüm kuralların tanımlandığı yerdir. Uygulamanın çalışması için verilmesi gereken izinler burada verilir. Açtığımız bütün activity sınıflarını burada tanımlamamız gerekir. Versiyon ayarlamaları da burada yapılır. Uygulamanın önemli dosyalarından biridir.

Aklımıza gelebilecek birçok şey bu dosya içindeki verilere göre şekillenir. Örneğin; sms gönderme iznini vermediğimiz bir uygulama sms gönderemez veya internet erişimi vermediğimiz bir uygulamanın internete bağlanma imkânı yoktur. Ya da bir activity sınıfını bu dosyada tanıtmadığımız sürece uygulamamız içinde o sınıfa erişmemiz mümkün olmayacaktır.

**Android manifeste yazmamız gereken kodlar;**

<uses-permission android:name="android.permission.CAMERA"></uses-permission>  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"></uses-permission>  
<uses-feature android:name="android.hardware.Camera"></uses-feature>

İzinlerin kontrolünün sağlanması;

Main2Activity classımızda sec fonksiyonu içerisine

if(checkSelfPermission(Manifest.permission.*CAMERA*)!= PackageManager.*PERMISSION\_GRANTED*)  
 {  
 requestPermissions(new String[]{Manifest.permission.*CAMERA*},2);  
 }else {  
 Intent ıntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 startActivityForResult(ıntent, 1);  
 }

yazarak izni kontrol ederiz.Baştaki ifte eğer izin yoksa izin istemeye yönlendiriyor.Eğer izin varsa yani else'e girer ise uygulama kamerayı açıyor.

Eğer önce izin yoksa kullanıcı uygulamayı ilk açtığında izin istiyor ve onaylıyorsa o zamanonRequestPermissionResult fonksiyonunu kullanırız.

@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 if(requestCode==2)  
 {

if(grantResults.length>0&&grantResults[0]==PackageManager.*PERMISSION\_GRANTED*)

{  
 Intent ıntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 startActivityForResult(ıntent, 1);  
 }  
 }  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
}

**Kamera bağlantısının yapılması.Çekilen fotoğrafın imageviewe aktarılması**

Uygulamada daha önce kullanıcı izinleri gibi detayla yapılımıştı.Kamera bağlantısını yapmak ve görüntüyü imageviewe aktarmak için önce global olarak bir imageview tanımlarız

public class Main2Activity extends AppCompatActivity {

ImageView resim;

Daha sonra tanımladığımız imageviewi onCreate metodu altında ilan ederiz.

@Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
 resim=(ImageView)findViewById(R.id.*imageView*);

}

Kamera bağlantımızı oluşturmak ve görseli imageviewe aktarmak için yapmamız gereken son aşama ise bir kontrol ile izinin olup olmadığının kontrolü ve gelen verinin boş olup olmadığının kontrolünün yapılması işlemidir.

onActivityResult fonksiyonu ile kameradan görüntü alıp imageviewe aktarma işlemini gerçekleştirmiş oluruz.

@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 try{  
 if(requestCode==1&&resultCode==*RESULT\_OK*&&data!=null){  
 Bitmap image=(Bitmap)data.getExtras().get("data");*//Çekilen resim id olarak bitmap şeklinde alındı ve imageview'e atandı* resim= (ImageView)findViewById(R.id.*imageView*);  
 resim.setImageBitmap(image);  
 }  
 }catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
  
  
 super.onActivityResult(requestCode, resultCode, data);  
}

**Bazı arayüz kısıtlamalarının yapılması**

Kullanıcı ana ekranda daha önce çekdiği notlardan birine tıkladığı takdirde gelecek olan ekranda kaydet butonunun aktif olmaması sağlandı.Yani kullanıcının aynı kaydı birden fazla yapması engellendi.Kullanıcı yeni bir kayıt yapmak istediğinde imageview ve edittextte eski verilerinin olmaması içinde kısıtlamalar eklendi.Bu sayede yeni kayıt durumunda edittext boş imageview ise standart olarak koyduğumuz resim ile gelicek.

Bunun için butonun visible özelliğinden edittext'in setText ve imageview'in ise setImageBitmap özelliğinden faydalanıldı.Önce kullandığımız araçları Main2Activity içerisinde tanımladık;

public class Main2Activity extends AppCompatActivity {  
 ImageView resim;  
 EditText isimText;  
 Button kaydet;

Daha sonra onCreate metodunu şu şekilde düzenledik;

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
 resim=(ImageView)findViewById(R.id.*imageView*);  
 isimText=(EditText)findViewById(R.id.*editTextName*);  
 kaydet=(Button)findViewById(R.id.*btnKaydet*);  
  
 Intent ıntent=getIntent();  
 String bilgi=ıntent.getStringExtra("bilgi");  
 if(bilgi.equalsIgnoreCase("yeni"))  
 {  
 Bitmap arkaplan= BitmapFactory.*decodeResource*(getApplicationContext().getResources(),R.drawable.*camera*);  
 resim.setImageBitmap(arkaplan);  
 kaydet.setVisibility(View.*VISIBLE*);  
 isimText.setText("");  
 }else  
 {  
 kaydet.setVisibility(View.*VISIBLE*);}}

MainActivity içerisinde onOptıonsSelectedItem metodu şu şekilde düzenlendi.

Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 if(item.getItemId()==R.id.*not\_ekle*)  
 {  
 Intent ıntent=new Intent(getApplicationContext(),Main2Activity.class);  
 ıntent.putExtra("bilgi","yeni");  
 startActivity(ıntent);  
 }  
 return super.onOptionsItemSelected(item);  
}

Bu sayede yeni ekleme yapılmak istendiğinde sistem "yeni" metni gidicek arkaplan normalkoyduğumuz görsel olucak buton gözükecek ve edittext'in içerisi boş olacak.

**Veritabanı kayıt işlemi**

Projemizde veritabanı resim ve ismiyle kaydetmek için 1 adet static sqliteDatabase türünden bir veritabanı ve Bitmap türünden 1 değişken tanımlarız.

public class Main2Activity extends AppCompatActivity {  
  
 ImageView resim;  
 EditText isimText;  
 Button kaydet;  
 static SQLiteDatabase *veritabani*;  
 Bitmap cekilenNot;}

Resmin ismini normal veritabanına kayıt gibi kaydedebilirken resim kaydederken biraz daha farklı bir yöntem kullanmamız gerekir.

İsimi notİsim adlı bir değişkene atarız.ByteArrayOutputStream türünden bir değişken oluştururur.Daha sonra imageview denn aldığımız görseli kompres ederiz.yani sıkıştırırız.Daha sonra bu sıkıştırılmış verileri byteArray isimli bir byte dizisine atarız.Daha sonra herhangi bir hata durumunda sistemin çökmemesi için try-catch içerisine önce veritabanımızı oluştururuz daha sonra tablomuzu oluşturup ekleme sorgusuyla ekleme işlemini yaparız.Bunun içinde VALUES(?,?) kısmındad 1. soru işaretine statement.bindString(1,notİsim) ile ismi göndeririz statement.bindBlob(2,byteArray) ilede sıkıştırılmış resim verilerini göndeririz.execute ilede kayıt işlemimiz gerçekleşmiş olur

Veritabanın kayıt işlemi için daha önceden kaydet butonunun onClick özelliğine tanıttığımız kaydet fonksiyonumuzu şu şekilde düzenleriz;

public void kaydet(View view)  
{  
 String notİsim=isimText.getText().toString();  
 ByteArrayOutputStream outputStream=new ByteArrayOutputStream();  
 cekilenNot.compress(Bitmap.CompressFormat.*PNG*,50,outputStream);  
 byte[]byteArray=outputStream.toByteArray();  
 try{  
 *veritabani*=this.openOrCreateDatabase("Notlar",*MODE\_PRIVATE*,null);  
 *veritabani*.execSQL("CREATE TABLE IF NOT EXISTS notlar (isim VARCHAR,resim BLOB)");  
 String sqlSorgu="INSERT INTO notlar(isim,resim) VALUES(?,?)";  
 SQLiteStatement statement=*veritabani*.compileStatement(sqlSorgu);  
 statement.bindString(1,notİsim);  
 statement.bindBlob(2,byteArray);  
 statement.execute();  
 }catch (Exception e){  
 e.printStackTrace();}

Intent ıntent=new Intent(getApplicationContext(),MainActivity.class);  
 startActivity(ıntent);}

**Kayıtlı verilerin çekilmesi Listviewe aktarılması ve tıklanıldığında diğer ekrana aktarılması**

Veritabanımızda kayıtlı verilerin çekilebilmesi için 1 String ve 1 Bitmap türünden Arraylistlere ve 1 adet Arrayadapter'e ihtiyacımız vardır.

public class MainActivity extends AppCompatActivity {  
 static ArrayList<Bitmap> *notResim*;

Bunları onCreate metodunda şu şekilde tanımlarız.  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 ListView listView=(ListView)findViewById(R.id.*listview*);  
 final ArrayList<String> notİsim=new ArrayList<String>();  
 *notResim*=new ArrayList<Bitmap>();  
 ArrayAdapter arrayAdapter=new ArrayAdapter(this,android.R.layout.*simple\_list\_item\_1*,notİsim);  
 listView.setAdapter(arrayAdapter);

Herhangi bir sistem veritabanı hatasına karşın try-catch yapısında gerekli kodlarımızı yazarız.Verileri çekmek için Cursor türünden br değişken tanımlarız içerisine select sorgumuzu yazarak veritabanındaki tüm verileri çekeriz daha sonra isim sütunu için indexleri tutacak bir değişken resim sütunu indexleri için başka bir değişken tanımlarız.Sonra cursorumuzu tablonun en başına göndeririz.while döngüsü içerisinde cursor boşluğu gelene kadar işlemlerimizi yaptırırız.Bu işlemler notİsim arraylistine isimIndexten gelen verinin atanması ve sıkıştırılmış veri dizisi halindeki resmimizin dönüştürülüp notResim isimli arrayliste atanması ve cursor un moveToNext komutu ile her seferinde bir sonraki satıra geçmesini sağlamak.arrayAdapter.notifyDataSetChaned(); kodu ise bir datayı değiştirildiğinde arrayAdapter'e haber verip listview içinde hemen güncellemesini sağlar.

try{

Main2Activity.*veritabani*=this.openOrCreateDatabase("Notlar",*MODE\_PRIVATE*,null);  
 Main2Activity.*veritabani*.execSQL("CREATE TABLE IF NOT EXISTS notlar(isim VARCHAR,resim BLOB)");  
 Cursor cursor=Main2Activity.*veritabani*.rawQuery("SELECT \* FROM notlar ",null);  
 int isimIndex=cursor.getColumnIndex("isim");  
 int resimIndex=cursor.getColumnIndex("resim");  
 cursor.moveToFirst();  
 while (cursor!=null)  
 {  
 notİsim.add(cursor.getString(isimIndex));  
 byte[]byteArray=cursor.getBlob(resimIndex);  
 Bitmap resim= BitmapFactory.*decodeByteArray*(byteArray,0,byteArray.length);  
 notResim.add(resim);  
 cursor.moveToNext();  
 arrayAdapter.notifyDataSetChanged();  
 }  
}catch (Exception e)  
{  
 e.printStackTrace();  
}

Listedeki isimlerden birisine tıklanıldığında ekrana aktarılabilmesi için MainActivity içerisindeki onCreate metodunun içinde listview in setOnClickListener özelliğinden faydalanırız.

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 Intent ıntent=new Intent(getApplicationContext(),Main2Activity.class);  
 ıntent.putExtra("bilgi","eski");  
 ıntent.putExtra("isim",notİsim.get(i));  
 ıntent.putExtra("i",i);  
  
 startActivity(ıntent);  
 }});

Main2Activity deki onCreate metodunu ise şu şekilde düzenlememiz gerekir;

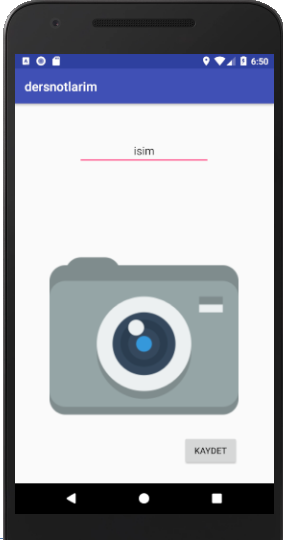
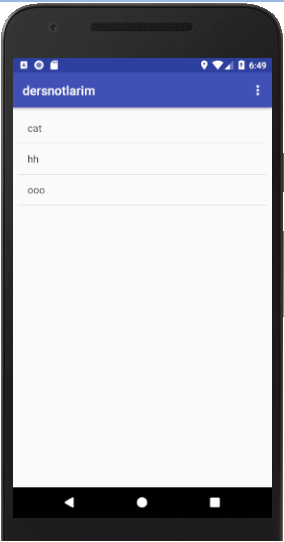
@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
 resim=(ImageView)findViewById(R.id.*imageView*);  
 isimText=(EditText)findViewById(R.id.*editTextName*);  
 kaydet=(Button)findViewById(R.id.*btnKaydet*);  
  
 Intent ıntent=getIntent();  
 String bilgi=ıntent.getStringExtra("bilgi");  
 if(bilgi.equalsIgnoreCase("yeni"))  
 {  
 Bitmap arkaplan=BitmapFactory.*decodeResource*(getApplicationContext().getResources(),R.drawable.*camera*);  
 resim.setImageBitmap(arkaplan);  
 kaydet.setVisibility(View.*VISIBLE*);  
 isimText.setText("");  
 }

Else den itibaren olan kodlarda isim ve resmimizi getirip isimi edittext'e resimi ise imageview'e aktarır.Kaydet butonunu görünmez yapıp imageview'imizin tıklanılabilirliğini etkisiz hale getiriyor.

else{  
 String isim=ıntent.getStringExtra("isim");  
 isimText.setText(isim);  
 int notResmi=ıntent.getIntExtra("resim",0);  
 resim.setImageBitmap(MainActivity.*notResim*.get(notResmi));  
 kaydet.setVisibility(View.*INVISIBLE*);

resim.setEnabled(false);  
 }}

**Projeden Görseller**



**Instagram Klon**

Android üzerinde çalışan Instagram Klon uygulaması yapılmasına karar verildi.

**Proje İçeriği:**

Bir login ekranı tasarlanacak login işlemleri gerçekleştirilecek.

Kullanıcı daha önceden çekmiş olduğu fotoğrafları uygulamaya yükleyebilecek ve yorum yapabilecek.

Daha sonradan kullanıcı istediği zaman hesabına girdiğinde paylaşılmış gönderileri görebilecek

Tüm bu paylaşılan fotoğraflar yorumlar kullanıcı kayıt işlemleri vs depolama gerektiren tüm işlemleri parse backend service'i üzerinden back4app.com da kuracağımız parse server üzerinden yapılacak.

Kullanılacak ekstra teknolojiler:Parse backend servis ve back4app

Kullanılacak Dil:Java

Kullanılacak Editör:Android Studio

Kullanılan API:API26:Android 8.0(OREO)

**Projede elde edilmesi planlanan kazanımlar :**

Tasarım ekranı yapılması sayesinde Android studioda tasarım öğrenilmesi ve tasarım tecrübesi elde edilmesi.

Parse backend service kullanılması ve tecrübesi edilmesi.

Back4app.com üzerinde parse server kurulumu yapılması ve uygulama ile bağlantısının yapılması

**Parse ve back4app nedir araştırma yapılması uygulamaya dahil edilmesi**

Parse kullanılarak daha önceden yapılmış uygulamalar incelendi.Uygulamaya entegre edilmesi parse server kurulumu gibi kısımlar incelendi.

**Parse nedir?**

2011’de mobil uygulamalar için backend desteği sağlamak üzere kurulan open source olan Parse daha sonra Facebook tarafından 2013 yılında 85 Milyon Dolar’a satın alınmıştı. Android, iOS, Js gibi bir çok alanda hem server hem veritabanı sağlamakla birlikte push notification yollamak gibi işleri de bizler için kolaylaştırıyordu.



**Back4app nedir?**

Parse server kurulumu yapmak için kurulmuş bir sitedir.

**Parse'ı uygulamamıza dahil etmek için;**

Parseplatform.org'a gidilir android kısmında view on github'a tıklanır.Daha sonra androidde oluşturduğumuz projemizin içinde build.gradle(app)'in içine gireriz.

implementation "com.parse:parse-android:=$parseVersion"

implementation "com.parse:parse-fcm-android:$parseVersion"

ext{

parseVersion="1.17.3"

}

Kodlarını yazarız bu sayede parse'ı uygulamamıza dahil ederiz.

**Back4app kurulumu ve uygulama ile server bağlantısı**

Daha önce uygulamamıza dahil ettiğimiz parse'ın back4app üzerinden server kurulumunu yapmak için uygulamamızda yeni bir java classı oluştururuz.İsminede parseClass deriz.Classımızı application'unumuza extends ederiz.Daha sonra onCreate metodunu oluştururuz ve içine şu kodları yazarız :  
@Override  
public void onCreate() {  
 super.onCreate();  
 Parse.*setLogLevel*(Parse.*LOG\_LEVEL\_DEBUG*);  
 Parse.*initialize*(new Parse.Configuration.Builder(this)  
 .applicationId()  
 .clientKey()  
 .server()  
 .build());}

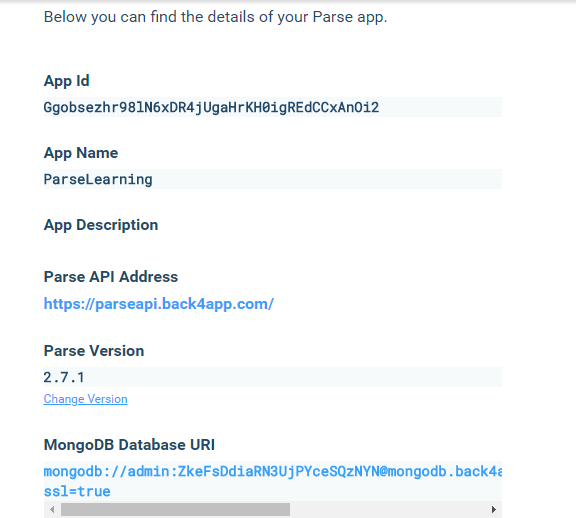
Bunlar serverimizle uygulamamızı bağlamamızı sağlayacaktır.Server bilgilerimizi buraya yazacağız.

Back4app.com'a girdikten sonra mail ve şifre ile üye olunur.Üye olunduktan sonra My apps kısmından build new app'e tıklarız.Uygulamamıza isim veririz ve istenirse açıklama yazılabilir.Create butonuna basılarak serverimiz oluşturulur.

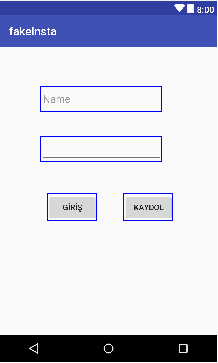
Kurulum tamamlandıktan sonra karşımıza gelen ekranda core settingse tıklarız.Burada karşımıza bize gerekli olan tüm bilgiler çıkar.Application ıd,clintkey ve server adresi.  
@Override  
public void onCreate() {  
 super.onCreate();  
 Parse.*setLogLevel*(Parse.*LOG\_LEVEL\_DEBUG*);  
 Parse.*initialize*(new Parse.Configuration.Builder(this)  
 .applicationId("Ggobsezhr98lN6xDR4jUgaHrKH0igREdCCxAnOi2")  
 .clientKey("F9vlYRPjAIJmhGHlvyuQftTY4czogYByGhQbT7Tg")  
 .server("https://parseapi.back4app.com/")  
 .build());}

Aldığımız o bilgileri kullanarak kodlarımızı bu şekilde düzenleriz. Daha sonra android manifestte

android:name=".parseClass" yaparak clasımızı tanıtırız.

****

**Login ekranı tasarımı ve login işlemi**

Activity\_main.xml içerisinde login ekranı tasarımımızı yaparız 2 adet edittext 2 button kullanırız.Edittextlerde kullanıcı adı ve şifre bilgilerini alırız.butonların birinde giriş diğerinde ise kayıt için kullaanacağız.Butonlardan giriş butonunun onClick özelliğine giris kaydol butonunun onClick özelliğine ise kayit yazarız bu sayede giriş ve kayıtlarla ilgili yazdığımız fonksiyonu direk bu şekilde butonlara yönlendirebiliriz.

Daha sonra mainactivity içerisinde edittextlerimizi tanımlarız.

EditText nameEditText;  
EditText passwordEditText;

Daha sonra onCreate metodu altında bunları ilan ederiz.

nameEditText=(EditText) findViewById(R.id.*sign\_up\_activity\_name\_text*);  
passwordEditText=(EditText) findViewById(R.id.*sign\_up\_activity\_password\_text*);

onClick metodlarına yazdığımız fonksiyonlarıda mainactivity içerisinde tanımlarız.kayıttan sonra yönlendireceğimiz activity'ide oluştururuz isminede postActivity deriz.

public void giris(View view)   
{   
ParseUser.*logInInBackground*(nameEditText.getText().toString(), passwordEditText.getText().toString(), new LogInCallback() {  
 @Override  
 public void done(ParseUser user, ParseException e) {  
 if(e!=null)  
 {  
 Toast.*makeText*(getApplicationContext(),e.getLocalizedMessage(),

Toast.*LENGTH\_LONG*).show();  
 }else  
 {  
Intent ıntent=new Intent(getApplicationContext(),postActivity.class);  
startActivity(ıntent);  
Toast.*makeText*(getApplicationContext(),"Hoşgeldin. \n"+user.getUsername().toString(),Toast.*LENGTH\_LONG*).show();  
}}});}   
public void Kayit(View view)

{  
 ParseUser user=new ParseUser();  
 user.setUsername(nameEditText.getText().toString());  
 user.setPassword(passwordEditText.getText().toString());  
 user.signUpInBackground(new SignUpCallback() {  
 @Override  
 public void done(ParseException e) {  
 if(e!=null)  
 {  
 Toast.*makeText*(getApplicationContext(),e.getLocalizedMessage(),

Toast.*LENGTH\_LONG*).show();  
 }else  
 {  
 Toast.*makeText*(getApplicationContext(),"Kullanıcı Kaydedildi.",Toast.*LENGTH\_LONG*).show();  
 }}});}

**Kullanıcı hatırlatma ve menü tasarımı**

Kullanıcı hatırlatmayı yapabilmek için;

MainActivity altında onCreate metodu içerisinde bir kontrol mekanizması yazarız.

ParseUser parseUser=ParseUser.*getCurrentUser*();  
if(parseUser!=null)  
{  
 Intent ıntent=new Intent(getApplicationContext(),FeedActivity.class);  
 startActivity(ıntent);  
}

Menü tasarımı yapabilmek için res klasörüne sağ tıklayıp new directory deriz isminede menu veririz.menu klasörüne sağ tıklayıp new menu resource file'a tıklarız ismine ana\_menu deriz.içerisine;

<item android:id="@+id/gonderi\_ekle" android:title="Gönderi Ekle"></item>  
<item android:id="@+id/cıkıs" android:title="Çıkış yap"></item>

Yazarız ve menümüzü oluştururuz.

Bunu uygulamamıza bağlamak için postActivity içerisinde onCreateOptionsMenu ve onOptionsItemSelected'ı override ederiz.

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 *//menümüzü bağlarız.* MenuInflater menuInflater=getMenuInflater();  
 menuInflater.inflate(R.menu.*ana\_menu*,menu);  
 return super.onCreateOptionsMenu(menu);}  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 *//hangi item seçilirse ne işlem yapılacak belirlenir.* if(item.getItemId()==R.id.*gonderi\_ekle*)  
 {  
 Intent ıntent=new Intent(getApplicationContext(),postSaveActivity.class);  
startActivity(ıntent);}  
 else if(item.getItemId()==R.id.*cıkıs*)  
 {  
 ParseUser.*logOutInBackground*(new LogOutCallback() {  
 @Override  
 public void done(ParseException e) {  
 if(e!=null)  
 {  
 Toast.*makeText*(getApplicationContext(),e.getLocalizedMessage(),

Toast.*LENGTH\_LONG*).show();  
 }else  
 {  
 Intent ıntent=new Intent(getApplicationContext(),MainActivity.class);  
 startActivity(ıntent); }}});}  
 return super.onOptionsItemSelected(item);}

**Kullanıcı izinleri post ekleme ekranı tasarımı ve post yükleme işlemi**

Daha önce oluşturduğumuz postSaveActivity de yani Yükleme ekranı tasarımı için 1 imageview bir edittext ve bir buton koyarız.Butonun onClick özelliğine yukle deriz.Edittexte ise kullanıcı yorumlarını yazabilecek.Imageviewinde onclick özelliğine resimsec diyeceğiz.

postSaveActivity içerisinde

EditText commentText;  
ImageView ımageView;

Edittext ve imageview imizi tanımlarız.onCreate içindede ilan ederiz;

commentText=findViewById(R.id.*upload\_activity\_comment\_text*);  
ımageView=findViewById(R.id.*upload\_activity\_ımageview*);

androidManifest içerisinde izinleri alabilmek için;

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"></uses-permission> yazarız.

Daha sonra postSaveActivity içinde izinlerin kontrolünü yaparız izinler yoksa izin aldırırız varsa yapılacak olan işlemleri yazarız.

resimSec metodu içerisinde;

if(ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_EXTERNAL\_STORAGE*)!= PackageManager.*PERMISSION\_GRANTED*)  
{  
 *//eğer izin yoksa izin istenilir.* ActivityCompat.*requestPermissions*(this,new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},2);  
}else  
{  
 *//eğer izin varsa* Intent ıntent=new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 startActivityForResult(ıntent,1);  
} Kodlarını yazarız.

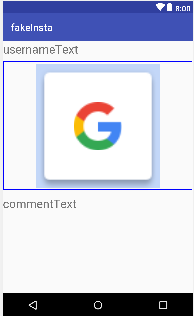
resimSecin dışında ise;

@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 *//izin sonucunda yapılacak şey* if(requestCode==2){ if(grantResults.length>0&&grantResults[0]==PackageManager.*PERMISSION\_GRANTED*)  
 {  
 Intent ıntent=new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 startActivityForResult(ıntent,1);  
 }}  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);}  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 *//startactivityforresult çalışınca ne olucak* if(requestCode==1&&resultCode==*RESULT\_OK*&&data!=null)  
 {  
 Uri uri =data.getData();  
 try {  
 Bitmap bitmap=MediaStore.Images.Media.*getBitmap*(this.getContentResolver(),uri);  
 ımageView.setImageBitmap(bitmap);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }}  
 super.onActivityResult(requestCode, resultCode, data);  
}

Yükleme işlemi yapabilmek için ise yukle metodumuzun içerisini şu şekilde düzenleriz.Resimi yüklemek için bir byte dizisi türüne çeviririz ve bu şekilde yükleriz.;

public void yukle(View view)  
{  
 String yorum=commentText.getText().toString();  
  
 ByteArrayOutputStream byteArrayOutputStream=new ByteArrayOutputStream();  
 secilenResim.compress(Bitmap.CompressFormat.*PNG*,50,byteArrayOutputStream);  
 byte [] bytes=byteArrayOutputStream.toByteArray();  
 ParseFile parseFile=new ParseFile("resim.png",bytes);  
 ParseObject object=new ParseObject("Gonderiler");  
 object.put("resim",parseFile);  
 object.put("Yorum",yorum);  
 object.put("kullaniciAdi", ParseUser.*getCurrentUser*().getUsername());  
 object.saveInBackground(new SaveCallback() {  
 @Override  
 public void done(ParseException e) {  
 if(e!=null)  
 {  
 Toast.*makeText*(getApplicationContext(),e.getLocalizedMessage(),

Toast.*LENGTH\_LONG*).show();  
 }else  
 {  
 Toast.*makeText*(getApplicationContext(),"Gönderi Yüklendi",Toast.*LENGTH\_LONG*).show();  
 Intent ıntent=new Intent(getApplicationContext(),postActivity.class);  
 startActivity(ıntent); }}});}

**Kayıtların gösterimi için tasarım yapılması ve bağlayıcı sınıf yazmak(custom view ile postActivity bağlantısı)**

Postların gösterimi için activity\_post.xml içerisine 1 adet listview koyarız.Id olarka listview atarız.postActivity.java içerisinde tanımlarız.onCreate altındada ilan ederiz.

Yorumları resimleri ve paylaşan bilgilerini bir arada gösterebilmemiz için yeni bir dizayn yapmamız gerekir.bunun için layoutta new layout resource file deriz.İsminede custom\_view deriz.

Custom\_view in içinde text kısmına gelip layoutumuzu linear layout yaparız.orientation ayarınıda vertical yaparız.

Daha sonra custom\_view ile postActivity birbirine bağlayabilmek için bir sınıf yazmamız gerekicek.Bir arrayadapter sınıfı yazacağız bunun için oluşturduğumuz postClass.java yı ArrayAdapter<String> içerisine extends ederiz.

public class postClass extends ArrayAdapter<String> {  
 private final ArrayList<String> kullaniciAdi;  
 private final ArrayList<String> yorum;  
 private final ArrayList<Bitmap> resim;  
 private Activity context;  
  
 public postClass(ArrayList<String> kullaniciAdi,ArrayList<String> yorum,ArrayList<Bitmap> resim,Activity context)  
 {  
 super(context,R.layout.*custom\_view*,kullaniciAdi);  
 this.kullaniciAdi=kullaniciAdi;  
 this.yorum=yorum;  
 this.resim=resim;  
 this.context=context;  
 }  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater layoutInflater=context.getLayoutInflater();  
 View ozelView=layoutInflater.inflate(R.layout.*custom\_view*,null,true);  
 TextView kullaniciAdiText=ozelView.findViewById(R.id.*custom\_view\_username\_text*);  
 TextView yorumText=ozelView.findViewById(R.id.*custom\_view\_comment\_text*);  
 ImageView resimView=ozelView.findViewById(R.id.*custom\_view\_imageview*);  
  
 kullaniciAdiText.setText(kullaniciAdi.get(position));  
 resimView.setImageBitmap(resim.get(position));  
 yorumText.setText(yorum.get(position));  
 return ozelView;  
  
 }  
}

Bu sayede postActivityden gönderilen Arraylisti buraya kaydettik.Burdaki arraylistte hangi pozisyondaysa alıp custom\_viewe ilettik

**Bağlayıcı sınıfın bağlantısının yapılması ve kayıtların getirilmesi**

Custom\_view ile postActivityi birbirine bağlamak için postActivity içerisinde;

ArrayList<String> kullaniciAdiParse;  
ArrayList<String> kullaniciYorumParse;  
ArrayList<Bitmap> kullaniciResimParse;  
PostClass postClass;

Tanımlarız.onCreate altında ise bunları ilan ederiz.

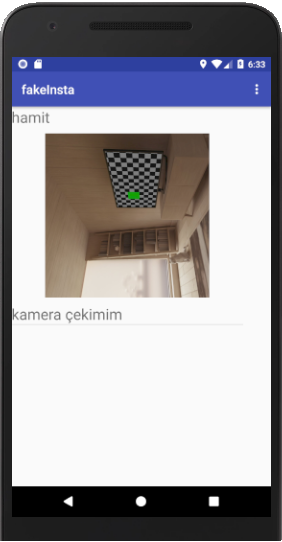
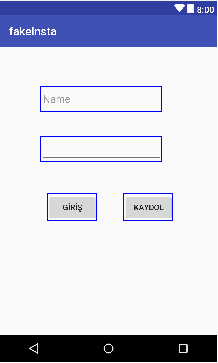
@Override   
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_feed);  
 listView=findViewById(R.id.listView);  
 kullaniciAdiParse=new ArrayList<>();  
 kullaniciYorumParse=new ArrayList<>();  
 kullaniciResimParse=new ArrayList<>();  
 postClass=new PostClass(kullaniciAdiParse,kullaniciYorumParse,kullaniciResimParse,this);  
 listView.setAdapter(postClass);  
}

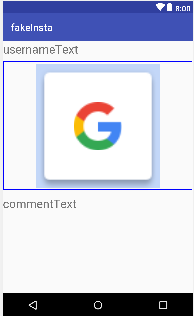
Daha sonra onCreate dışında indir fonksiyonu oluştururuz.

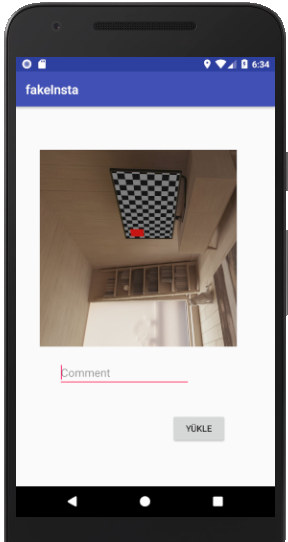
İndir fonksiyonunun içerisini şu şekilde düzenleriz.

public void indir()  
{  
 ParseQuery<ParseObject> query=ParseQuery.*getQuery*("Gonderiler");  
 query.findInBackground(new FindCallback<ParseObject>() {  
 @Override  
 public void done(List<ParseObject> objects, ParseException e)  
 {  
 if(e!=null)  
 {  
 Toast.*makeText*(getApplicationContext(),e.getLocalizedMessage(),Toast.*LENGTH\_LONG*).show();  
 }else {  
 if(objects.size()>0) {  
 for(final ParseObject object:objects) {  
 ParseFile parseFile=(ParseFile) object.get("resim");  
  
 parseFile.getDataInBackground(new GetDataCallback() {  
 @Override  
 public void done(byte[] data, ParseException e)  
 {  
 if(e==null&&data!=null)  
 {  
 *//byte dizisine çevirdiğimiz resmi decodebytearray diyerek tekrar resim haline çeviriyoruz.*  
 Bitmap bitmap= BitmapFactory.*decodeByteArray*(data,0,data.length);  
 kullaniciResimParse.add(bitmap);  
 kullaniciAdiParse.add(object.getString("kullaniciAdi"));  
 kullaniciYorumParse.add(object.getString("Yorum"));  
  
 postClass.notifyDataSetChanged();  
 }  
 }  
 });  
 }  
 }  
 }  
  
 }  
 });  
}

**Projeden Görseller**



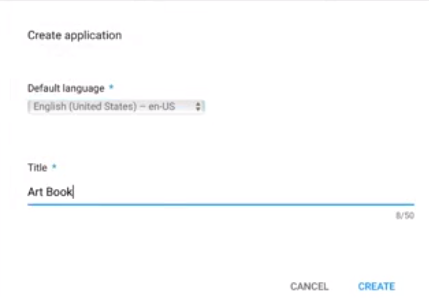
****



**Uygulamalar google play'e nasıl yüklenir**

Uygulamaları google play'e yükleyebilmek için https://developer.android.com/distribute/ adresine gideriz.Buradan launch play console'a tıklarız.Eğer bir geliştirici hesabınız varsa sizi play.google.com'a yönlendirir.Yoksada yeni bir hesap açtırız ve senelik 25$ kayıt ücreti ödeyerek yeni hesap alırsınız..

**Uygulama nasıl yüklenir?**

Karşımıza gelen ekranda Publish An Android App On Google Play'e tıklarız.Karşımıza aşağıdaki gibi bir görsel çıkar burda default dili seçeriz ve uygulamaya isim veririz.Create deriz ve bizi ayrıntılı ayarların bulunduğu bir ekrana yönlendirir.Burda product details'de kısa açıklama ve uzun açıklama tarzında 2 tane açıklama isteyecek uygulama hakkında.Bir soonraki aşama ise asıl önemli olan aşama.Graphics assets aşaması telefonda tablet saatte vs ürünlerde nasıl gözükürle ilgili uygulama içi screen shootları yükleyeceğimiz bölüm karşımıza çıkar.Daha sonra icon koyacağımız kısım geliyor burda tasarladığımız bir icon varsa onu koyarız.Daha sonra categorization kısmı gelicek.Burda uygulama tipini(oyun,normal uygulama) seçeriz.oyunsa ne tür bi oyun uygulamaysa ne tür uygulama diye seçeriz alt bir kategoride.Daha sonra bir privacy policy yüklemesi istiyor.Yani kullanım koşullarının yasal düzenlemesi.Bunları yaptıktan sonra save draft'a tıklarız.Daha sonra pricing & distribution kısmına geliriz bu kısımda ücretsiz bir uygulamamı yoksa ücretli bir uygulamamı onu ayarlarız.Eğer ücretli bir uygulamaysa koyacağınız bir ticari hesap ister google sizden.Daha sonra hangi ülkelerde yayınlanmasını istediğinizi sorar.Bunu siz belirlersiniz.Daha sonra pricing kısmında bir kaç soru daha geliyor karşınıza 13 yaş altı için uygunmu amerikan kanunlarına uygunmu tarzında bunları da cevapladıktan sonra save draft deriz bu kısmada.Uygulama içinde satın almalar varsa In app products kısmında ayarlamalar yapılması gerekir.Daha sonra App releases'e geliriz.upload new apk'e tıklarız.Burda alpha,beta ve production seçenekleri var biz production'ı seçiyoruz.Create release'e tıklarız.New release production sayfası gelir karşımıza.Sonra ENROLL'a tıklarız hizmet şartları sözleşmesini kabul ederiz.Daha sonra upload apk butonuna tıklarız.Browse files'a tıklarız.apk'imizi seçeriz.Apk yüklendikten sonra release not yazmamızı isteyecek bizden bbu versiyonda ne var neler değişti tarzında. birşeyler yazmamızı isteyecek.Review'e tıklarız.

**Eğer apk'miz hazır değilse;**

Android studio da projeyi açarız build variantsta build variant ayarını debugdan release'e getiririz.Daha sonra buildden generate signed apk deriz.Key store path isteyecek.Create'e tıklarız.Burda keystore path ve şifre belirleriz ilk aşamada.Daha sonra key bölümünde alias ve şifre belirleriz.Sertifika kısmında ise sim soyisim şirket varsa onun bilgileri lokasyon ülke ve ülke kodu belirleriz.Ok'a tıklarız.next deriz daha sonraki ekranda bizden bir signature seçmemizi isteyecek v1 ve v2 signature güvenlikle ilgili önemli durumlar söz konusuysa v2 seçilir. finish deriz.apk'imiz hazır olur.

**Robot Kol projesi**

1 Mekatronik mühendisi 2 Bilgisayar öğretmeni ve 1 Bilgisayar mühendisi olmak üzere 4 kişilik bir ekiple birlikte bir robot kol projesi yapılmasına karar verildi.

* Projenin kaba-taslak çizimleri kağıt üzerinde yapıldı.
* Dış iskelet 3d printer yardımıyla oluşturulacağı kararlaştırıldı.
* İç mekanizma ise 1 arduino ve bağlantıları sağlamak için ekstra bir elektronik devre kartı tasarlanacak.
* Robot koldaki hareketler servo motorlar yardımıyla sağlanacak.

**Proje İçeriği:**

Robot bir kontrol kumandasından gelen komutlara göre servo motorların hareketiyle kontrol edilecek.

**Projede elde edilmesi planlanan kazanımlar :**

3 boyutlu tasarımın nasıl yapıldığının görülmesi.

3d printer kullanımının tecrübe edilmesi.

Arduino elektronik devre kartının öğrenilmesi ve proje ile tecrübe edilmesi.

Ekstra bağlantılar için gerekli devre kartının oluşturulması ve tecrübe edilmesi.

**Gerekli ekipmanların tespiti ve eksiklerin temin edilmesi**

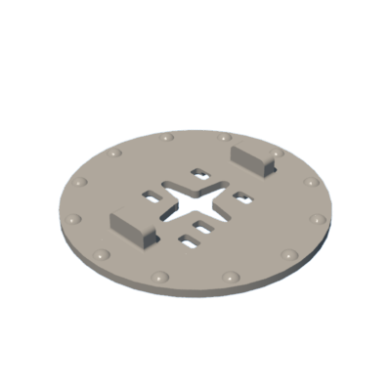
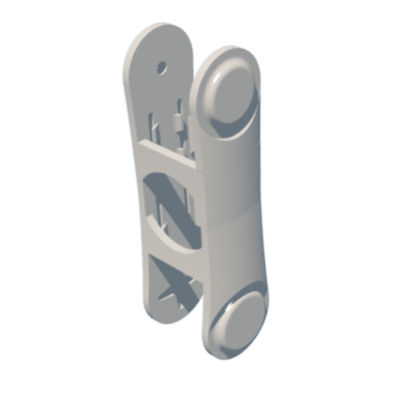
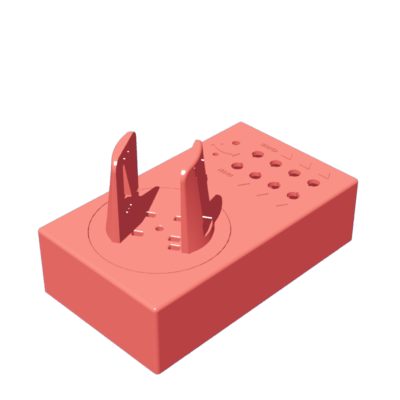
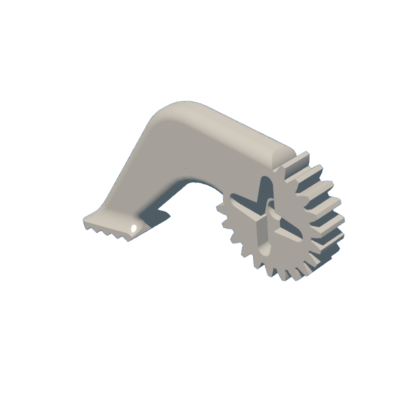
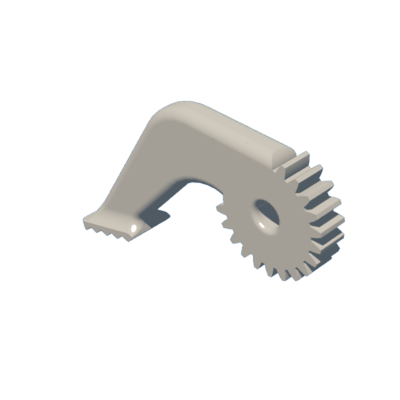
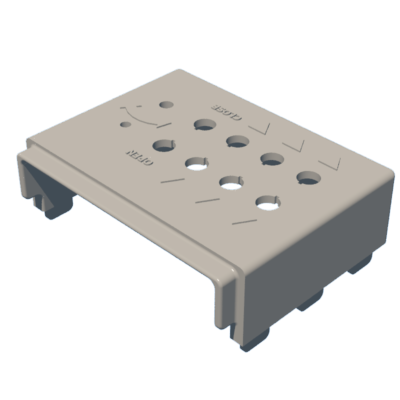
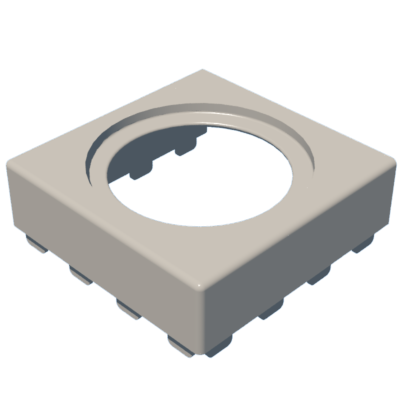
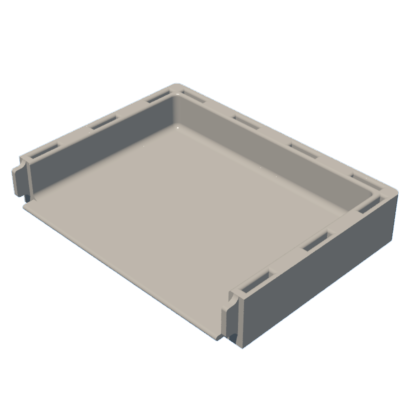
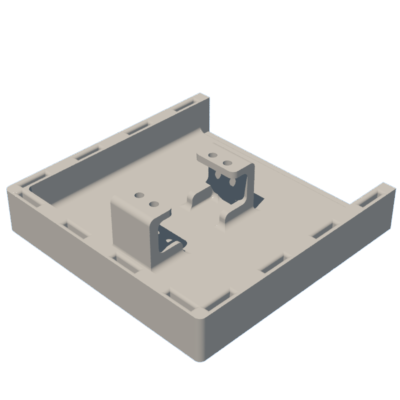
Kullanılacak tasarım ile ekipmanların belirlenmesi ve eksik ekipmanların temin edilmesi işlemleri yapıldı.

**Kullanılacak ekipman listesi;**

* **1 adet Arduino Mega**
* **4 adet Standard servo motor**
* **2 adet Mikro servo motor**
* **8 adet push button**
* **1 adet aç/kapa anahtar**
* **1 adet potansiyometre ve knob(pot başlığı)**
* **1 adet yeşil LED**
* **Baskı devre kartı**
* **8 adet 10K Ohm direnç**
* **1 adet 220 Ohm direnç**
* **Bağlantılar için kablo ve konnektörler somun ve civatalar**
* **Dış iskelet için yaklaşık 1 kg filament**

**Dış iskelet çizimlerinin yapılması**

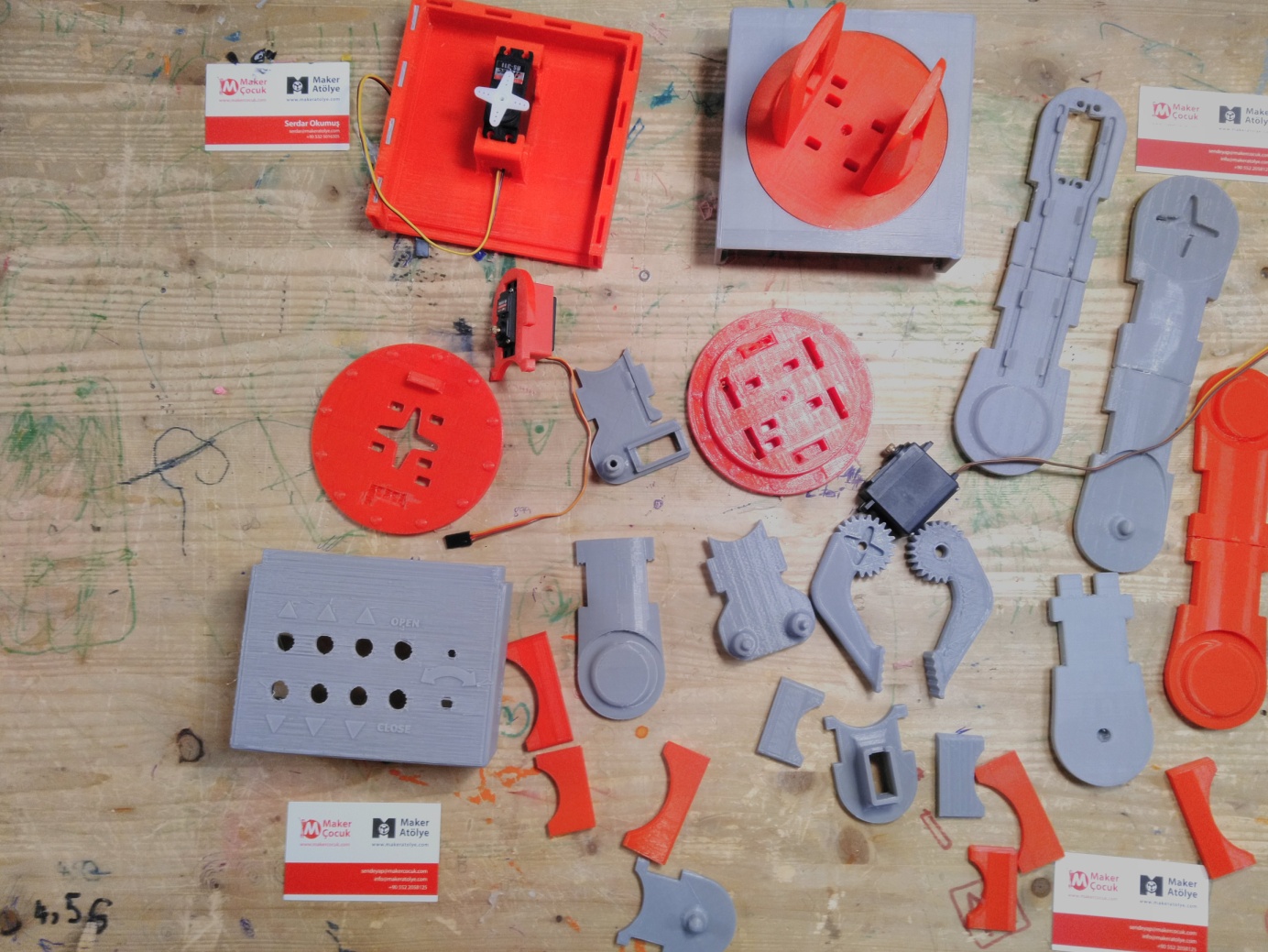
Solidworks 2017 programında dış iskeletin çizimleri yapıldı.

**Dış iskeletin 3d yazıcı ile çıktı alınması**

3 boyutlu olarak yapılan proje parçaları 3d yazıcılar yardımıyla basıldı.

Parçaların çokluğu ve basımlarının çok zaman alması nedeniyle, zamanı ekonomik kullanabilmek için farklı kalitelerde baskı yapan birden çok makineden çıktı alındı buda bazı yazıcıların kalitesiz baskı yapmasından dolayı bazı parçalarda istenen sonucun tam olarak elde edilememesine neden oldu.

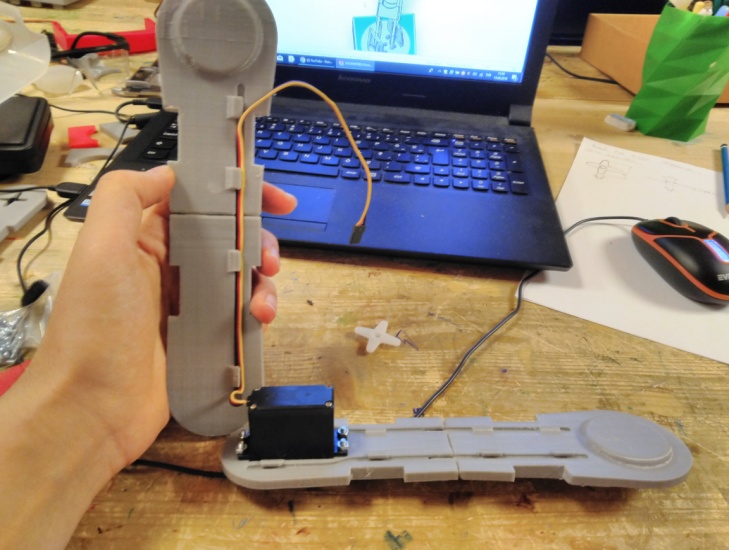


**Dış iskeletin birleştirilmesi ve yanlış baskıdan kaynaklı kusurların giderilmesi**

3D yazıcı ile alınan baskıların servo motorlar ve diğer parçalarla dış iskelete montajı yapıldı.Montajda,baskıdan oluşan yanlışlar ve hatalı baskılardan kaynaklı kısımlar zımpara ve 3d yazıcı kalemi ile giderildi.

Montaj görüntüleri;



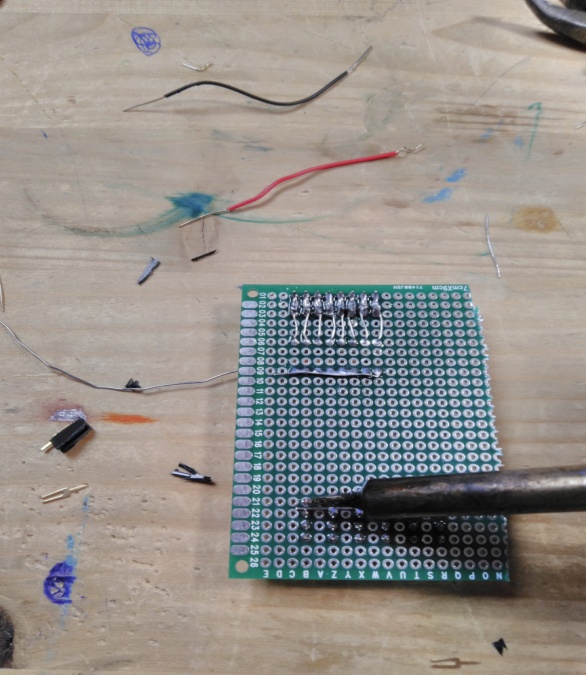
**Elektronik kartın hazırlanması**

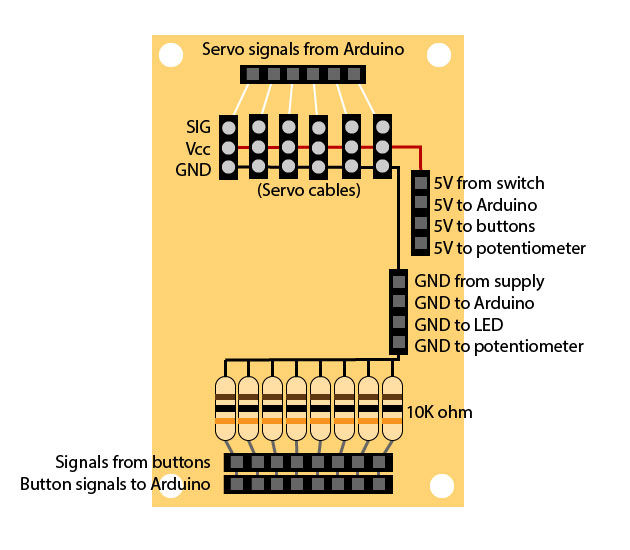
**Elektronik Bağlantılar;**

Robot kol kontrol paneli ve servo motorların bağlantılarında kolaylık sağlanması için bir devre kartı hazırlandı. 5V GND bu kart üzerinden dağıtıldı.Kontrol panelindeki butonlar, led, potansiyometre bağlantıları bu kart ile kontrol edilecek.

**Karşılaşılan problemler;**

Güç kaynağından çektiğimiz voltaj 12V. Ancak arduino 12V ile servolar ise fazla akım çektiği için arduinodan değil dışarıdan 5V ile besleme ile beslendiği için regüle kullanmamız gerekti. Kulandığımız regüle bu değeri düşürmeye yeterli olmadı. 5 V güç kaynağı kullanıp arduinoyu dışardan 9V luk pille besleme gibi bir çözümle sorun halledildi.

. 

**Robot kol'un arduino kod kısmının algoritmasının oluşturulmasıyla birlikte kodlamaya geçilmesi ve giriş-çıkış birimlerinin tanımlanması**

**Programlama;**

Robot kol için mikrodenetleyici olarak ucuz ve programlaması daha kullanışlı olan arduino seçildi.Ancak çok fazla giriş ve çıkış olduğu için arduino mega modeli seçildi.Algoritması oluşturulan program yapısında daha sonra Arduino IDE üzerinde programlaması yazıldı.

**Robot kol arduino kodları şu şekildedir.**

Değişkenlerimizi bu şekilde tanımlarız.servo motor kullanacağımız için servo.h kütüphanesini dahil ederiz.Sonra kullanacağımız servoları sırayla anlaşılabilir birer isimle tanımlarız.Hangi servo hangi pinden sinyal alacak onu belirleriz.

|  |  |
| --- | --- |
| #include<Servo.h>  Servo servotabla;  Servo servoalt1;  Servo servoalt2;  Servo servoorta;  Servo servoon;  Servo servoagiz;  int aci1=90;  int aci2=90;  int aci3=90;  int aci4=70;  int greenpin=46;  int bluepin=45;  int butontablailerigeri=A1;  int butonaltileri=26;  int butonaltgeri=28;  int butonortaileri=30;  int butonortageri=32;  int butononileri=34;  int butonongeri=36;  int butonagizileri=38;  int butonagizgeri=40; | void setup() {   Serial.begin(9600);  servotabla.attach(2);  servoalt1.attach(3);  servoalt2.attach(4);  servoorta.attach(5);  servoon.attach(6);  servoagiz.attach(7);  servoalt1.write(90);   servoalt2.write(90);   servoorta.write(90);   servoon.write(90);   servoagiz.write(70);     pinMode(45,OUTPUT);   pinMode(46,OUTPUT);  for(int i=26;i<=40;i=i+2)  {  pinMode(i,INPUT);  }  pinMode(A1,INPUT);  } |

**Button hareketlerine göre kontrolün arduino kodlarının yazılması ve projenin tamamlanması**

**Loop fonksiyonunun içerisi şu şekilde düzenlenir;**

void loop() {

 int but1=digitalRead(26);

 int but2=digitalRead(28);

 int but3=digitalRead(30);

 int but4=digitalRead(32);

 int but5=digitalRead(34);

 int but6=digitalRead(36);

 int but7=digitalRead(38);

 int but8=digitalRead(40);

 int val = analogRead(A1);

int val1 = map(val, 0, 1023, 0, 180);

 servotabla.write(val1);

Serial.println(val);

if (but1==HIGH)

{

       aci1+= 1; // increase position of servo

       if (aci1 > 175)

{

aci1 = 175; // aci1 175'i aşınca 175 de sabitle.

}

       servoalt2.write(aci1); // yeni pozisyonu yaz

       servoalt1.write(180-aci1);

       delay(25); // bekle

}

 if (but2==HIGH)

{

       aci1-= 1; // increase position of servo

       if (aci1 > 175)

{

aci1 = 175; // aci1 175'i aşınca 175 de sabitle.

}

       servoalt2.write(aci1); // yeni pozisyonu yaz

       servoalt1.write(180-aci1);

       delay(25); // bekle

}

 if ((but3==HIGH)&&(aci2<=170))

{

   aci2=aci2-2;

servoorta.write(aci2);

delay(25);

}

if ((but4==HIGH)&&(aci2>=10))

{

aci2=aci2+2;

servoorta.write(aci2);

delay(25);

}

if ((but5==HIGH)&&(aci3<=160)){

   aci3=aci3+1;

servoon.write(aci3);

delay(15);

}

if ((but6==HIGH)&&(aci3>=20))

{

aci3=aci3-1;

servoon.write(aci3);

delay(15);

}

if ((but7==HIGH)&&(aci4<=160))

{

   aci4=aci4+1;

servoagiz.write(aci4);

delay(15);

}

if ((but8==HIGH)&&(aci4>=20))

{

aci4=aci4-1;

servoagiz.write(aci4);

delay(15);

}

if ((but8==LOW)&&(but7==LOW)&&(but6==LOW)&&(but5==LOW)&&(but4==LOW)&&

(but3==LOW)&&(but2==LOW)&&(but1==LOW))

{

 digitalWrite(46,180);

}

if ((but8==HIGH)||(but7==HIGH)||(but6==HIGH)||(but5==HIGH)||(but4==HIGH)||

(but3==HIGH)||(but2==HIGH)||(but1==HIGH))

{

 digitalWrite(45,180);

}

}