**BEACON İŞARETÇİLERİ KULLANILARAK GERÇEKLEŞTİRİLEN ÖĞRENCİ YOKLAMA SİSTEMİ**

Bu çalışmada beacon işaretçileri kullanılarak öğrenci yoklama sistemi gerçekleştirilmiştir. Sistem mobil uygulama, veritabanı, web servisler ve beaconlardan oluşmaktadır. Beaconlar geliştirilen bu sistemin donanımsal bileşenleridir. Mobil uygulama android işletim sistemi için geliştirilmiştir. Uygulama beaconlardan gelen sinyalleri algılayacak ve yoklama kaydı için geliştirilen web servise istek gönderecektir. Veritabanında öğrenciler ve katılım bilgileri tutulmaktadır. Web servis ise gelen isteklere göre ilgili kontrolleri yapıp yoklama kaydı için gerekli veritabanı kaydını gerçekleştirmektedir.

**SENARYO**

* Öğrenci mobil uygulama üzerinden sisteme giriş yapar.
* Öğrenci ders için sınıfa geldiğinde akıllı cihazının internet ve bluetooth bağlantılarını aktif duruma getirir.
* Uygulama arka planda çalışır ve sınıfta bulunan beacon’ un yaydığı sinyalleri yakalar.
* Uygulama, sisteme giriş yaptıktan sonra yayılan sinyaller süresi içinde ders varsa eğer derse imza işlemi yapılır.

**BEACON**

* Beacon cihazlar, BLE teknolojisi ile donatılmış, düşük güçlü ve düşük maliyetli elektronik vericilerdir.
* Android ve iOS cihazlar ile haberleşebilmektedir.
* Beacon cihazlar, BLE Advertising olarak ta adlandırılan belirli sıklıkta kablosuz sinyaller yayar.
* Mobil cihazlar, Beacon cihazlardan aldıkları verilere göre tepki vermektedirler. (uygulama çalıştırmaktadır)
* Böylelikle mikro konum belirleme, bulunulan mekâna göre aktivitelerde bulunma gibi uygulamalar gerçekleştirilmektedir.

Beacon cihazlarda yayılan Bluetooth Sinyaller 4 farklı veri taşımaktadır.

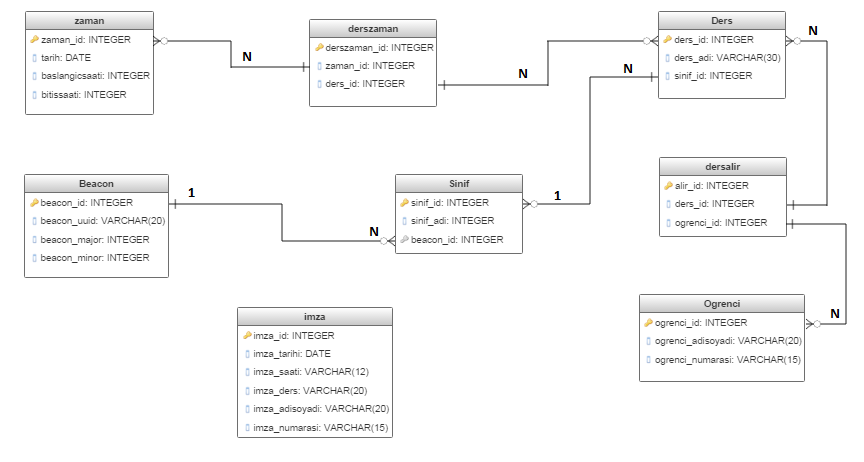
* Universal Unique Identifier (UUID)
  + Tüm bir beacon ağını tanımlamak için kullanılır (Üretici firma bilgisi)
  + 16 bayt uzunluğundadır.
  + Örneğin, mobil cihazdaki uygulama kendine ait UUID algıladığında veri paketinin geri kalan kısmını dinlemeye başlar.
* Majör
  + Üst ağ adresidir. 2 bayt uzunluğundadır.
* Minör
  + Alt ağ adresidir. 2 bayt uzunluğundadır.
  + Aynı üst ağ içerisindeki ilgili cihazı göstermek için kullanılır.

**VERİTABANI**

Sistem içerisinde kalıcı olarak tutulması gereken veriler veritabanında saklanmaktadır. Veritabanında öğrenciler, eğitmenler, dersler, sınıflar, ders programları, ders kayıtları, katılımlar ve bu katılımların saatleri tutulmaktadır. Bu tabloları inceleyecek olursak;

* Öğrenciler tablosunda sisteme kayıtlı öğrenciler ve bu öğrencilere ait bilgiler tutulmaktadır.
* Eğitmenler tablosunda sisteme kayıtlı eğitmenler ve bu eğitmenlere ait bilgiler tutulmaktadır.
* Dersler tablosunda açılan dersler ve bu derslere ait bilgiler tutulmaktadır.
* Ders alir tablosunda ilgili döneme ait öğrencilerin seçtiği dersler ve döneme ait bilgiler tutulmaktadır.
* Sınıflar tablosunda sınıf isimleri ve bu sınıflarda bulunan beacon işaretçilerine ait bilgiler tutulmaktadır.
* Zaman tablosunda derslere ait zaman bilgileri tutulmaktadır.
* Ders zaman tablosunda derslerin birkaç zamanı olabileceğinden tutulmaktadır.
* İmza tablosunda öğrencilere ait yapılan ders katılımları tutulmaktadır.

Veritabanı ER(Varlık – İlişki) diyagramı aşağıda gösterilmiştir.



**WEB SERVİS**

Geliştirilen web servis ile yoklama kaydı yapılmaktadır. Php ile geliştirilmiştir. Bu web servise, öğrenciye ait veritabanında kayıtlı olduğu id, majör ve minör bilgileri ile HTTP POST isteği gönderildiğinde sırasıyla aşağıdaki işlemler yapılır.

**1-) GİRİS.PHP**

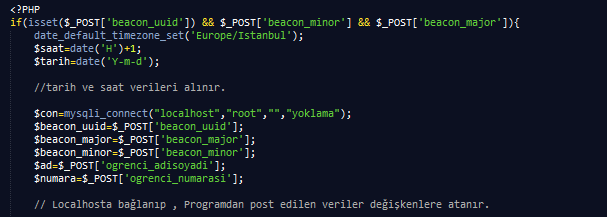
* Öncelikle POST isteği sisteme giriş yapılır.



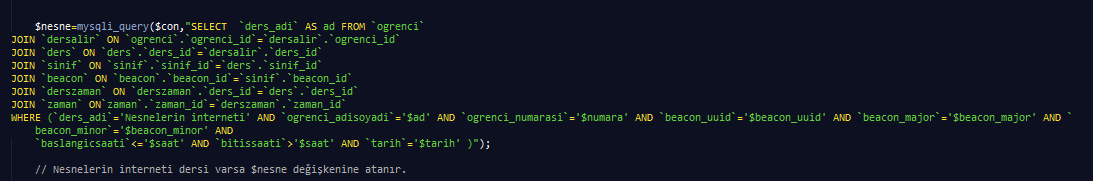
* + Eğer sisteme sorunsuz giriş yaparsa JSON olarak {“succes”,1] verisini gönderiyoruz.
  + Sisteme giriş yaparken Öğrenci adı soyadı ve Numara yanlış girildiyse eğer, JSON {“error”,0} verisini gönderiyoruz.

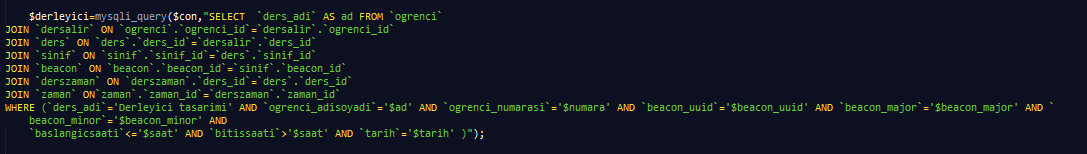
**2-) YOKLAMA.PHP**

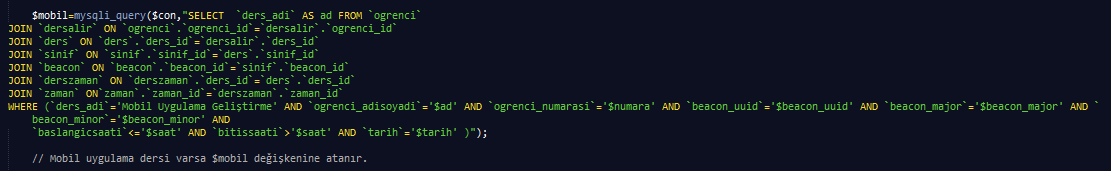
* Sisteme giriş yaptıktan sonra en yakın mesafede olan beacon ‘ ın UUID,Major ve Minor idsini POST isteği ile web servise gönderiyoruz.



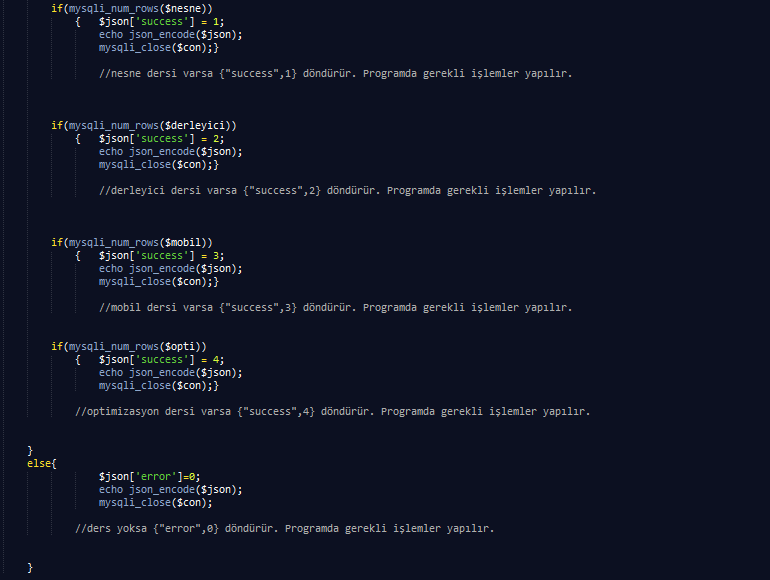
* + Daha sonra Post ettiğimiz verilere göre ve ders adına göre $nesne, $derleyici, $optimizasyon ve $mobil değişkenlerine atarız.





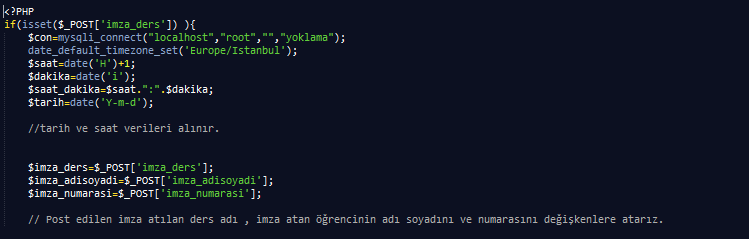


* Daha sonra bu verilerin sayısına göre bakıyoruz.1’ den büyük olan ife giriş yapıyor. JSON veriyi gönderiyoruz.

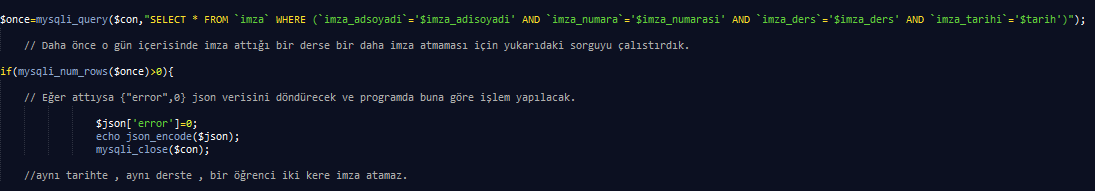


**3-) İMZA.PHP**

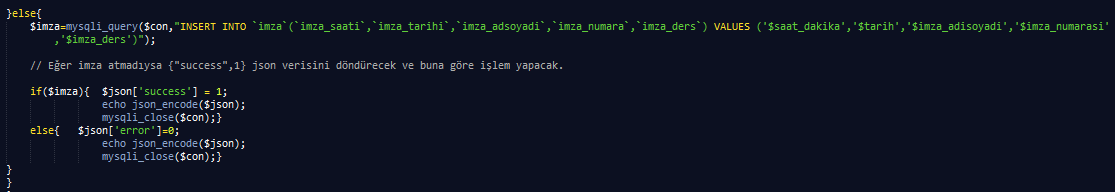
* Son olarak öğrenci imza atmak için öğrenci adı soyadı , numarası ve ders adı POST işlemi ile web servise yollanır.



* + Öğrencinin o derse o gün içerisinde imzası var mı diye kontrol edilir. Eğer varsa {“error”,0} JSON verisi geri döndürür.



* + Öğrencinin o tarihte o derse ait yoklama kaydı yoksa yeni yoklama kaydı imza tablosuna eklenir ve {“succes”,1} JSON verisi geri döndürür.



**MOBİL UYGULAMA**

Mobil uygulama android işletim sistemi için geliştirilmiştir. Uygulama tarafında beaconlardan gelen sinyalleri algılayabilmek için ek bir kütüphane kullanılmıştır. Bununla ilgili birçok kütüphane bulunmaktadır. Bu çalışmada estimote sdk kullanılmıştır. Estimote sdk içerisinde beacon sinyallerini yakalamak/işlemek için iki farklı yöntem söz konusudur. Bunlar monitoring ve ranging olarak adlandırılmaktadır. Bunun dışında sdk içerisinde region olarak adlandırılan bir yapı da bulunmaktadır. Bu kavramlarla ilgili detaylı bilgi aşağıda verilmiştir.

**KAVRAMLAR**

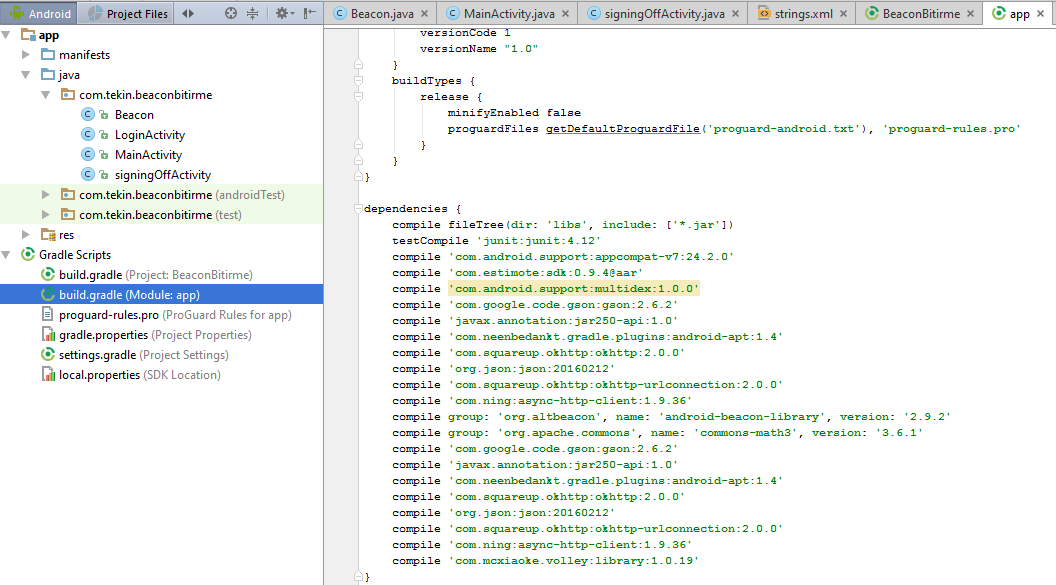
**Region:** Yakalanacak sinyallerin kapsamını daraltmak için kullanılmaktadır. Beaconlardan gelen sinyaller önceden belirlenen bu region’ un içerisinde yer alıyorsa işlemler yapılır. Region’ un dışında kalan beaconlardan gelen sinyaller göz ardı edilir. Region tanımlarken 4 değişken kullanılmaktadır. Bunlar taramaya ait unique bir string, uuid, majör ve minör bilgileridir. Örneğin string, null, null, null olarak region tanımlanırsa uygulamanın çalıştığı cihazın kapsama alanına giren tüm beacon sinyalleri yakalanacak ve işlemler gerçekleştirilecektir. Bir başka örnekte string, uuid, null, null olarak region tanımlaması yapılırsa sadece cihazın kapsama alanındaki belirlenen uuid’ ye sahip beacon sinyalleri yakalanacak ve işlemler gerçekleştirilecektir.

**Monitoring:** Bu sinyal yakalama/işleme yönteminde önceden belirlenen region da sürekli tarama yapılır. İçerisinde bulunan override metotlar sayesinde sadece önceden belirlenen bu region’ a bir beacon’ un girmesi çıkması durumlarında işlemler yapılabilir. Kapsama alanına giren beaconlara ait verilere ulaşılamaz.

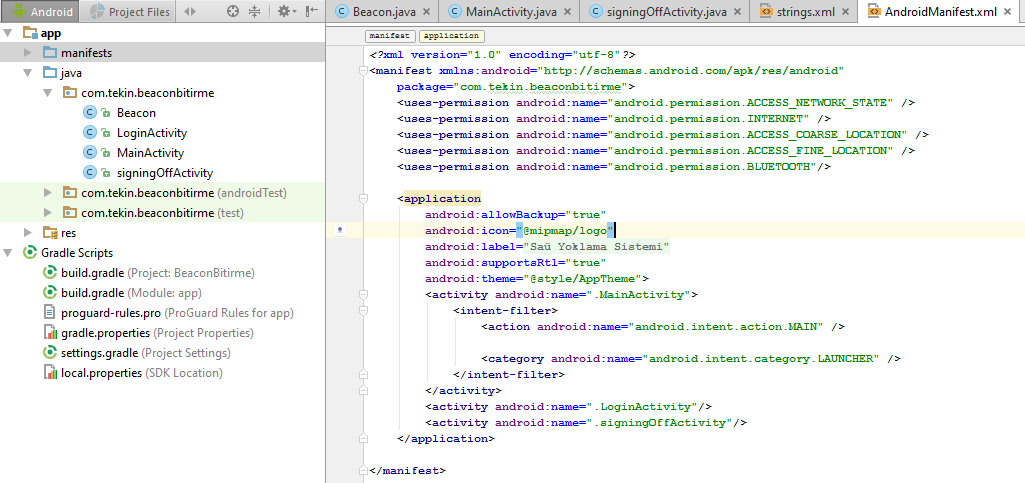
**Ranging:** Bu sinyal yakalama/işleme yönteminde de yine önceden belirlenen region da sürekli tarama yapılır. Uygulamanın çalışacağı cihazın kapsama alanına giren ve belirlenen region değerlerine sahip olan beaconlar ve bu beaconlara ait diğer veriler bu yöntem ile gözlemlenebilir. Monitoring’ den farkı ilgili region’ a giren beaconlara ait verilere ulaşılabilmesidir. Bir başka fark ise monitoring de bulunan region’ a beacon girmesi/çıkması durumunda tetiklenen override metotların bu yöntemde olmamasıdır.

**UGYULAMA ADIMLARI**

* Öncelikle beacon sinyalleri ile işlem yapabilmek için estimote kütüphanesi android projesine dâhil edilir. Kütüphane, projeye bağımlılık olarak eklenir. Bunu yapmak için android studio editöründe sol tarafta bulunan proje penceresinden build.gradle dosyası açılır. Daha sonra dosya içerisindeki dependencies bölümü altına altbeacon kütüphanesinin adresi eklenir ve proje build edilerek kütüphane ekleme işlemi tamamlanır.

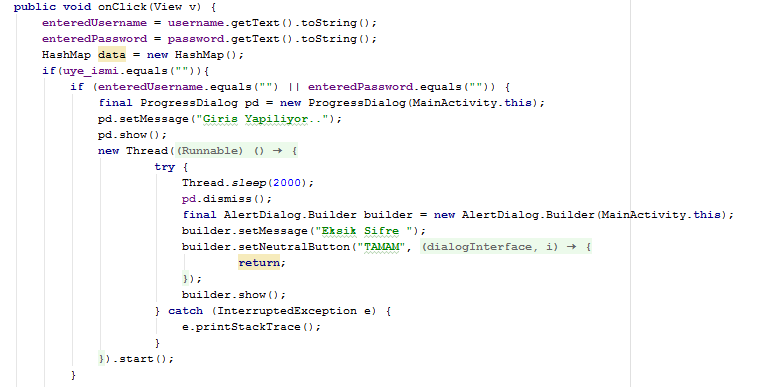


* Kütüphane ekleme işleminden sonra beacon sinyallerini yakalamak ve http isteğinde bulunabilmek için gerekli olan bluetooth ve internet izinleri uygulamaya AndroidManifest dosyası yardımıyla eklenir. Arka planda çalışmasını istediğimiz kodların yer alacağı sınıf da bu dosya içerisinde belirtilir. Bu işlemleri yapmak için android studio editöründe sol tarafta bulunan proje penceresinden AndroidManifest dosyası açılır. Gerekli izinler aşağıdaki şekilde de görüldüğü gibi <uses-permission> etiketleri arasına yazılarak eklenir. Arka plan kodlarının çalışacağı sınıf ise <application> etiketi altına name değeri eklenerek belirtilir.



**1-) MainActivity.java**

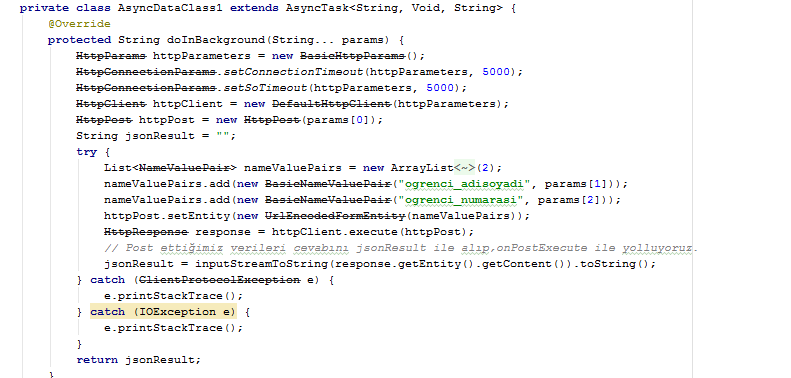
* Bu adımda girilen Öğrenci adının ve numarasının alınıp POST işlemi ile web servise gönderileceği geri dönen JSON verisine göre de sisteme giriş yapacağı bölümdür.
  + Eğer Öğrenci adı ve numarası boş olarak girildiyse “EKSİK ŞİFRE” diye hata verecektir. Yeniden öğrenci adı ve öğrenci numarasının girilmesi istenilecektir.



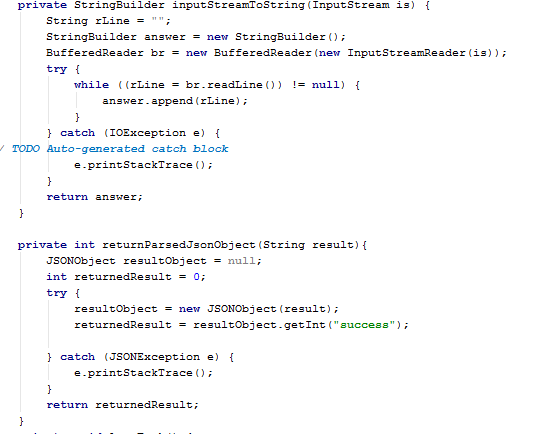
* + Uye ismi olarak SharedPrefences olarak tuttuğumuz değere her giriş yaptığımızda değer atanır. Eğer öğrenci daha önce sisteme giriş yaptıysa “Zaten giriş yapmış bulunmaktasınız..” diye uyarı gelir. Daha sonra girilen değerler parametre olarak web servise yollanmak amacıyla oluşturduğumuz sınıfa yollanır.



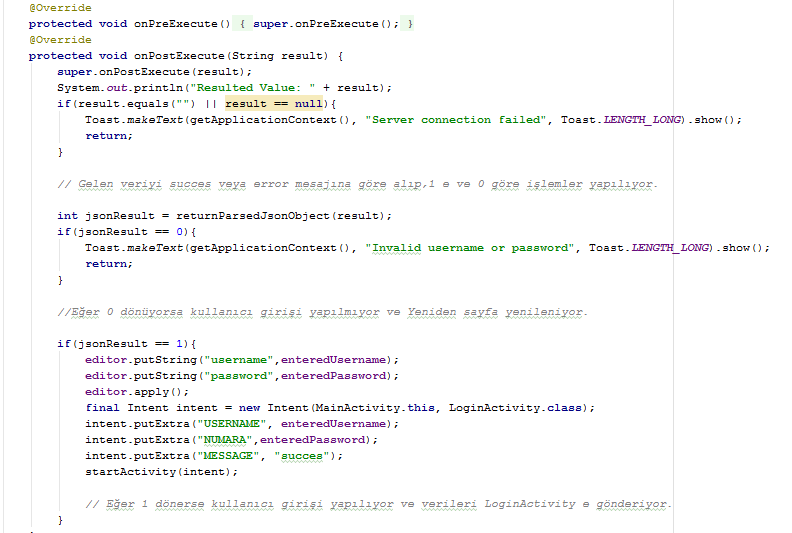
* + Web servise POST işlemi ve web servisten gelecek sonuç ile GET işlemi için oluşturduğumuz sınıfta doInBackground’ in içinde gelen parametreleri POST işlemine tabi tutarız.



* + Geri dönen JSON verisini parser işlemine tabi tutarak sonucu alırız.

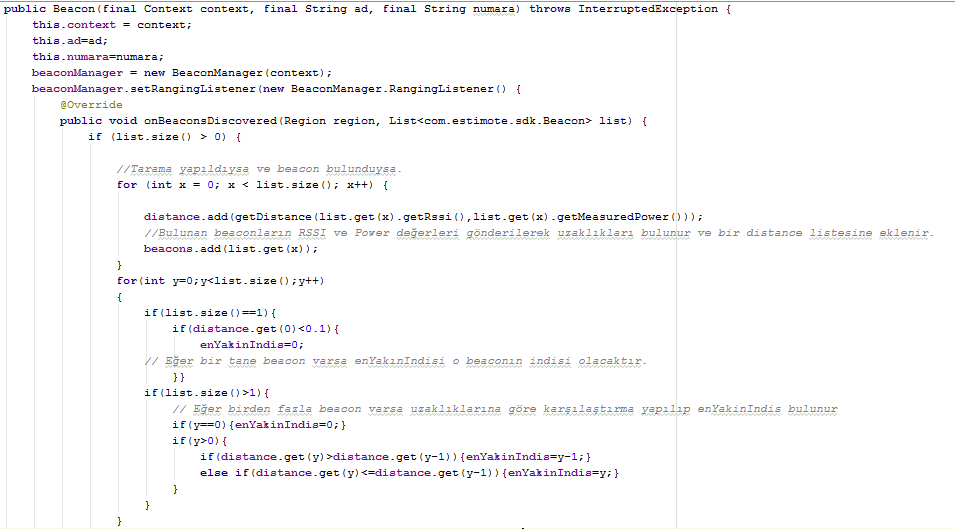


* + Parser işleminde sonra da sonucu göre sonuç 1 ise başarılı olarak sisteme giriş yapar. 0 ise sisteme giriş yapamaz.

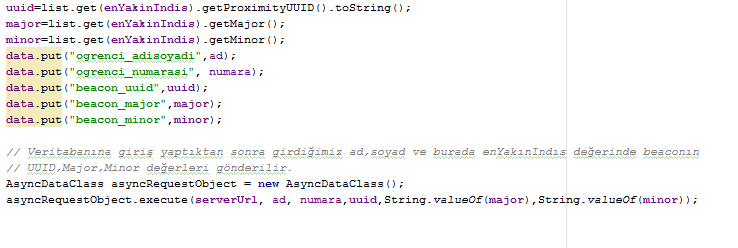


**2-) Beacon.java**

* Sisteme giriş yapıldıktan sonra ,öğrencinin gireceği sınıfta derse dersin olup olmadığını bulmak için beaconların algılanıp web servise POST işleminin yapılacağı ve ders olup olmadığını sorgulanacağı bölümdür.
  + setRangingListener bölümünde beaconların tarıyoruz ve en yakın mesafedeki beaconı buluyoruz.



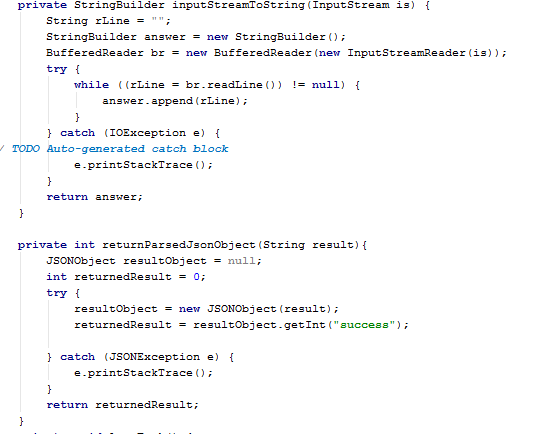
* + En yakın beaconın UUİD, major , minor idsini ve sisteme giriş yaptığımız öğrenci adı ve öğrenci numarasını web servise yollamak üzere AsyncDataClass sınıfına parametre olarak veriyoruz.



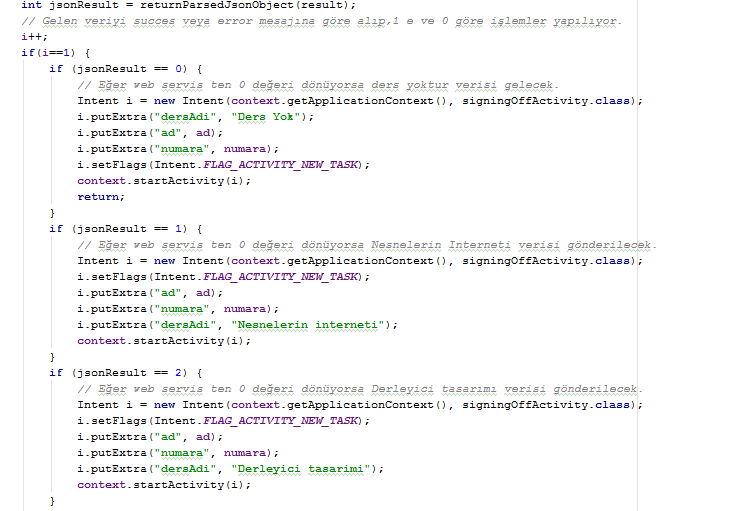
* + Parametre olarak web servise yolluyoruz. Geri dönecek değeri de döndürüyoruz.

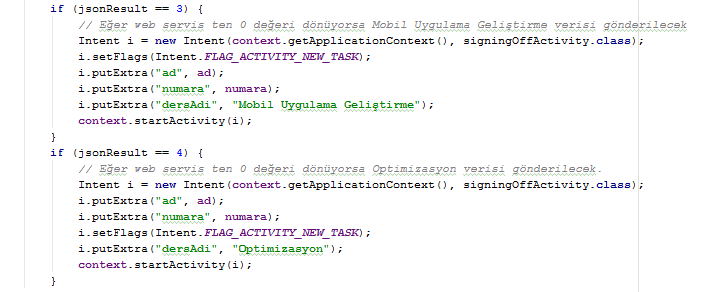


* + JSON parser ile geri dönen değerimizin sonucunu alıyoruz.



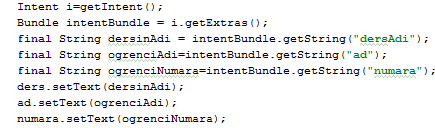
* + Parser işleminde sonra da sonucu göre sonuç;
    - 0 ise “Ders Yok” geri döner.
    - 1 ise “Nesnelerin İnterneti” geri döner.
    - 2 ise “ Derleyici Tasarımı” geri döner.
    - 3 ise “Mobil Uygulama ve Geliştirme” geri döner.
    - 4 ise “Optimizasyon” geri döner.





**2-) signingOffActivity.java**

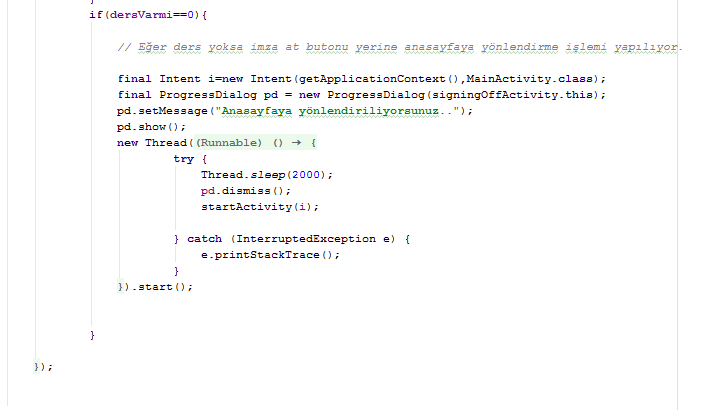
* Ders olup olmadığını bulduktan sonra , ders yoksa anasayfaya dönüleceği varsa imza atılacağı sınıftır.
  + Beacon sınıfından yolladığımız verileri alıyoruz.



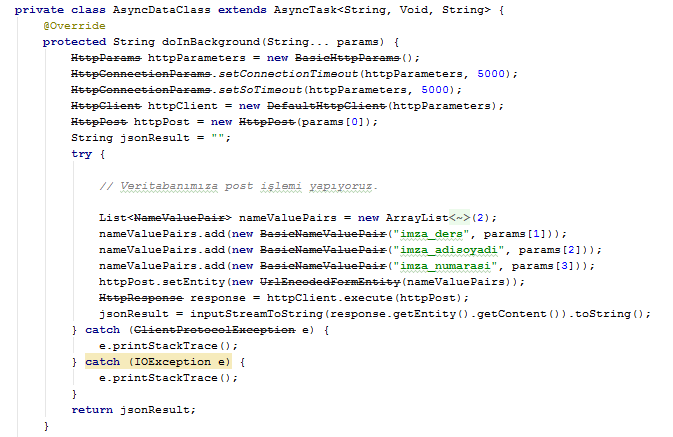
* + “Ders Yok” diye geri göndermediyse web servise göndermek üzere AsyncDataClass sınfımıza parametre olarak dersin adını , öğrenci adını ve numarasını göndeririz.



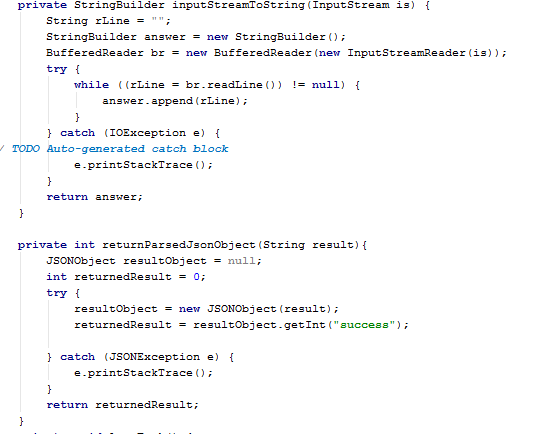
* + “Ders Yok” diye geri döndüyse anasayfaya geri diye button oluştururuz.



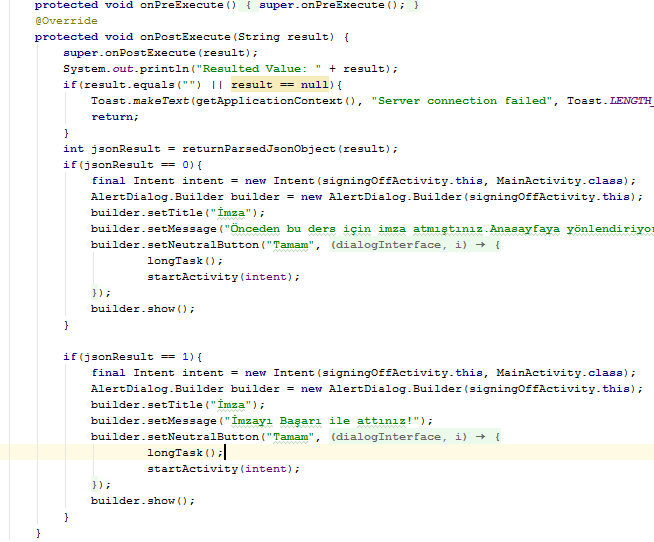
* + Gelen parametreleri web servise POST işlemini uygularız.



* + Dönen veriye göre sonucunu alırız.



* + Sonuç 1 ise ; “imzayı başarılı attınız.” Diye uyarı gelir. 0 ise eğer ; öğrenci o derse o tarih ve ders saatleri içinde imza atmıştır ve “Önceden bu ders için imza atmıştınız , Anasayfaya yönlendiriliyorsunuz.” Uyarısı gelir.



**KAYNAKLAR**

1 - Cüneyt Bayılmış ve Mehmet Özdemir, "Bluetooth Düşük Enerji Teknolojisine Sahip İşaretçi ve Akıllı Telefon Temelli Öğrenci Yoklama Sistemi", Bilişim Teknolojileri Dergisi, 9 (3), 249–254, 2016.

2 - https://github.com/ozdemirmehmet/beacon-yoklama-sistemi