



GÖMÜLÜ SİSTEM ÇÖZÜMLEMELERİ

Dr. Öğr. Üyesi Fırat AYDEMİR

Uyku Apnesi Tespit Sitemi

2022-2023 Bahar Dönemi

08/05/2023-22/05/2023

6.Rapor

HASAN MÜNİR DOĞRUEL-201913171801

YUSUF SARUKAN-201813171058

EMİRHAN AKDİN-201813171049

GİRİŞ

Bir önceki rapor döneminden sonra yapmış olduğumuz doktor uygulamasının ihtiyacı olan verileri üretecek olan Arduino entegre devresi yerine, Arduino'yu taklit edecek ve rastgele değerler üretip clientı besleyecek bir android uygulaması yaptık. Fakat bluetooth üzerinden veri gönderirken, Google'un yeni almış olduğu kararlar doğrultusunda onaylanmamış yayıncıların bluetooth üzerinden veri aktarımında sınırlamalar ve engellemeler (Permissions) getirildi.

Bluetooth tek bir framede 1024bayt yani 1kilobayt veri aktarabiliyor. Bunları byte array üzerinden server to client yada client to server olarak aktarım gerçekleştiriyor. Biz projenin daha performanslı olması için veri aktarım sınırı olmayan ve nispeten daha hızlı olan web socket kullanmayı tercih ettik. Bu şekilde Bluetooth'un kısıtlamalarından ve mesafeye bağlı veri kaybından kurtulmuş olduk.

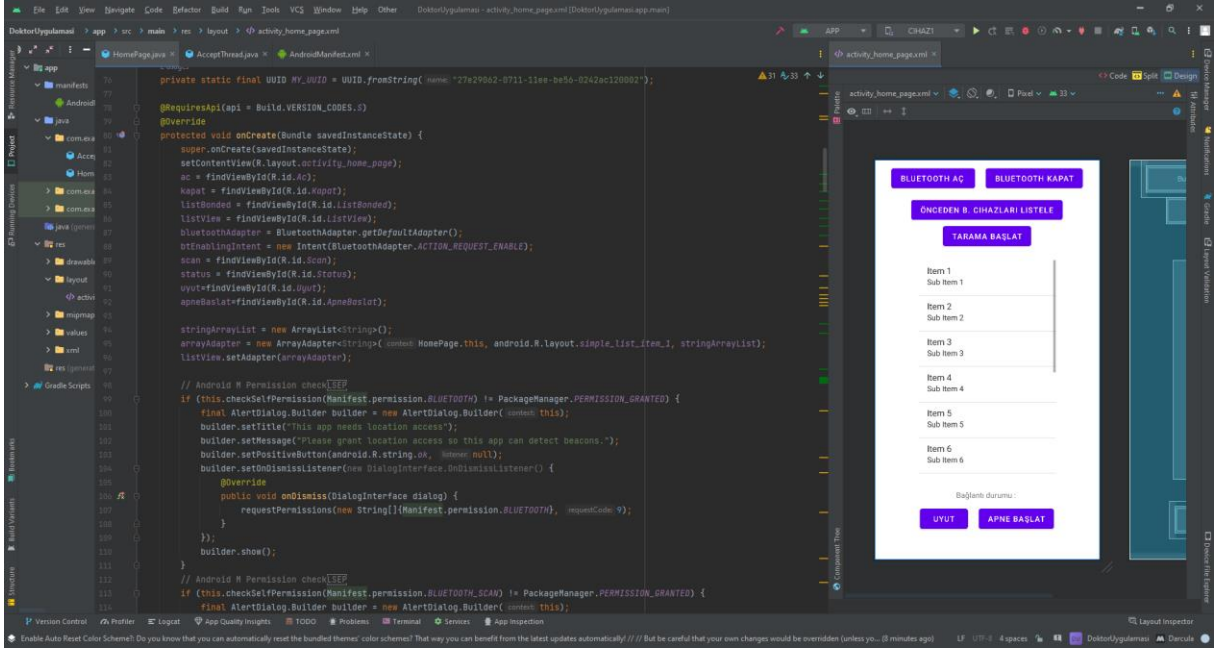
İŞ PLANI

Yapılacak iş	Rapor 1	Rapor 2	Rapor 3	Rapor 4	Rapor 5	Rapor 6	Rapor 7
Firebase Kurulumu, Normalizasyon	X						
Firebase, mobil bağlantısı		X					
Windows Form Tasarımı			X				
Windows Form Firebase bağlantısı				X			
Güncellemeler					X		
Veri taklit edecek server						X	
Verilerin yakalanması ve kullanıcı etkileşimi							

DipNot: Rapor 5 ve Rapor 6 dönemleri değiştiği için 1 rapor dönemimizi önceki yaptıklarımızı güncellemeye harcadık.

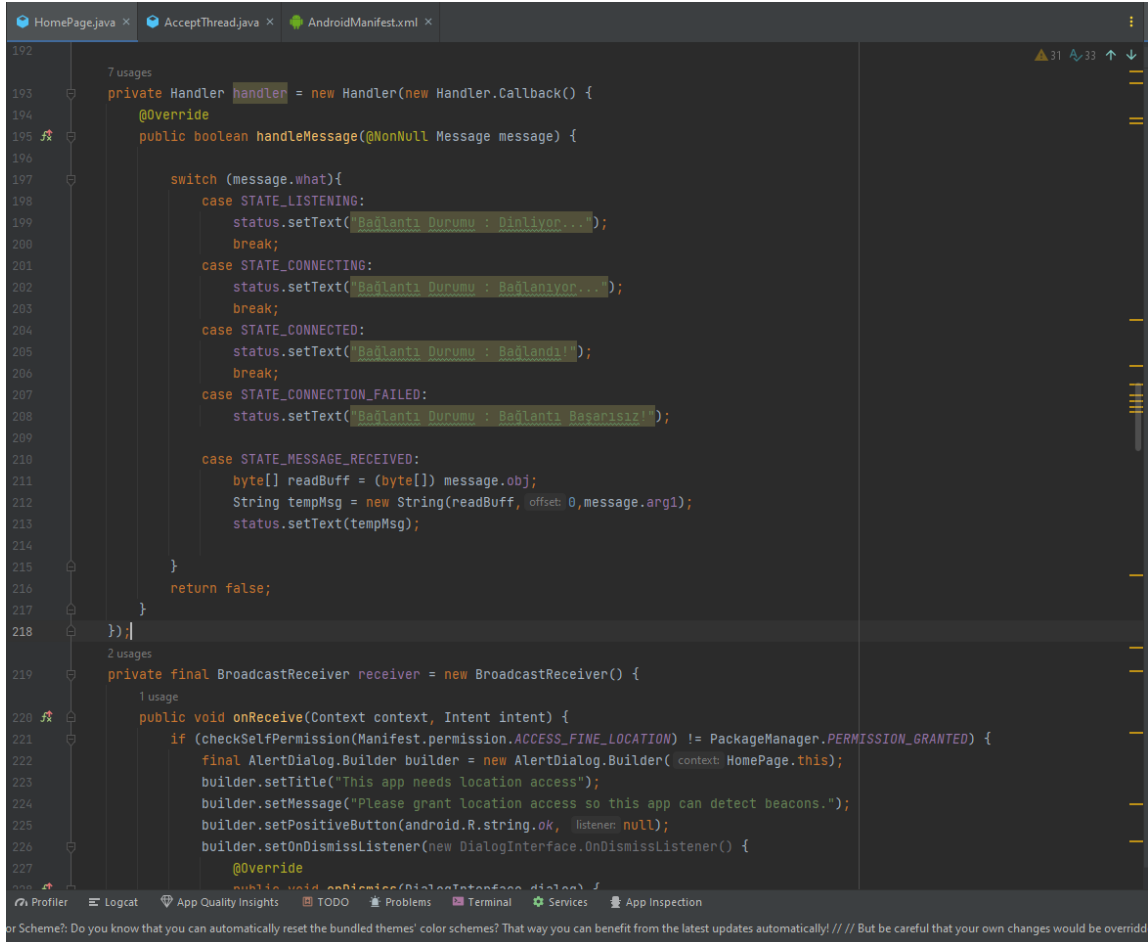
YAPILAN İŞLER

1.Erişim İzinlerinin Ayarlanması (Permissions)

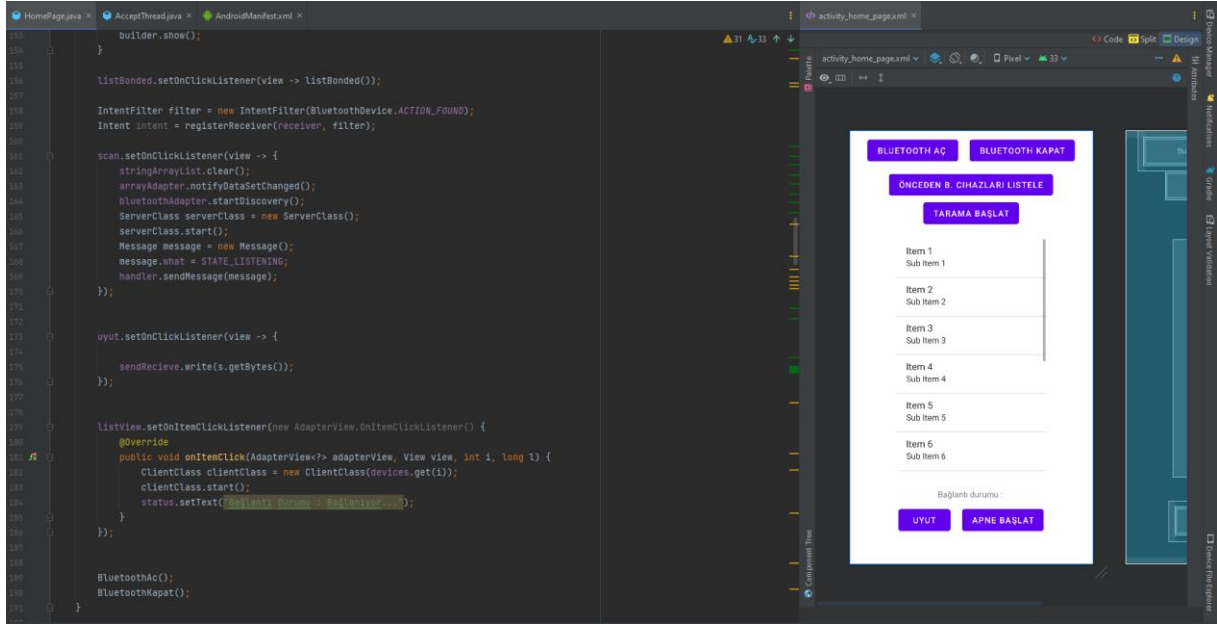


Görseldeki kod bloğunda 99. satırda başlayan kod parçası bluetooth ile etkileşime geçmek için gerekli izni alması için yazılmış fakat bu izin sadece entegre bluetooth cihazının MAC adresine ulaşmak için kullanıldı. Bunun gibi 8 parça kod bloğu gerek bağlantı gerekse tarama için kullanıldı. Gerekli görsel nesneler (View'ler) java classlarına (Object) bağlandı ve tıklama eventleri için hazırlıklar yapıldı.

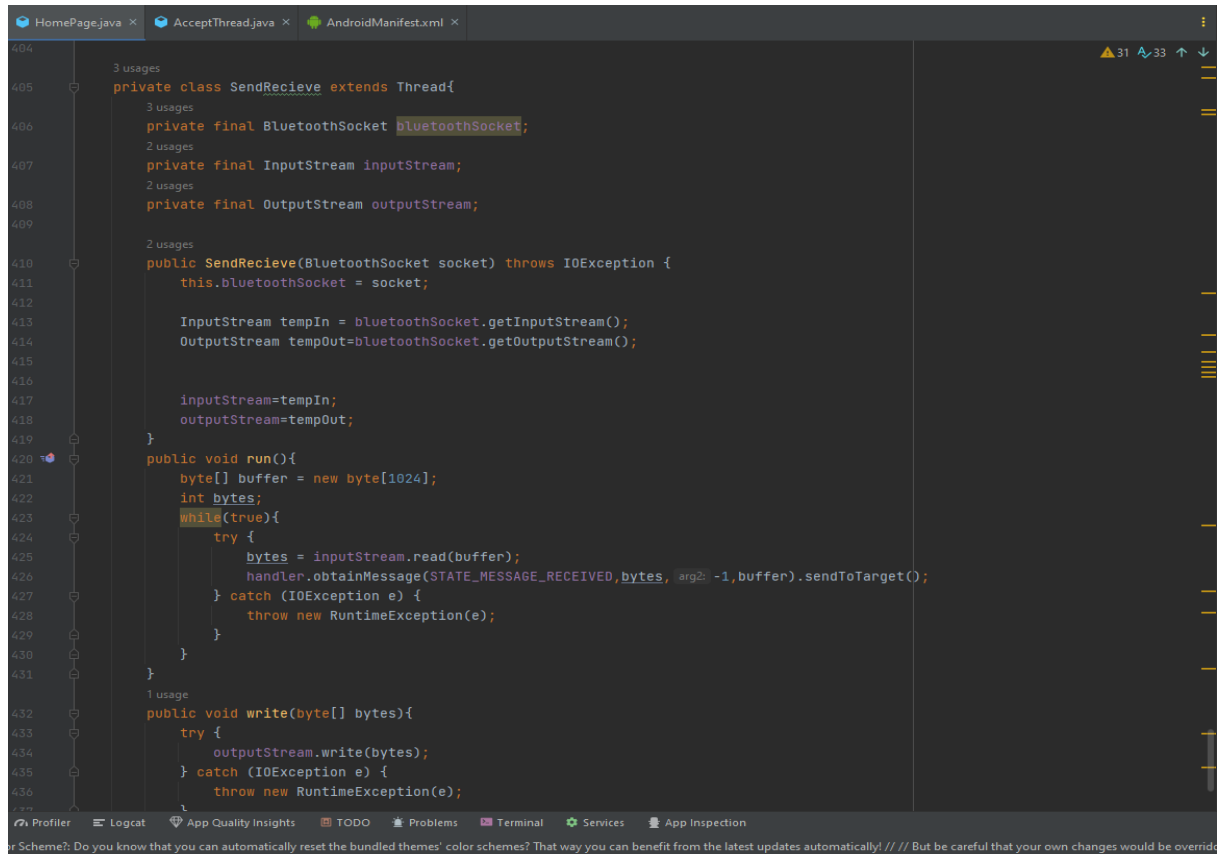
2.Statelerin Belirlenmesi ve Broadcast Receiver Ayarlanması



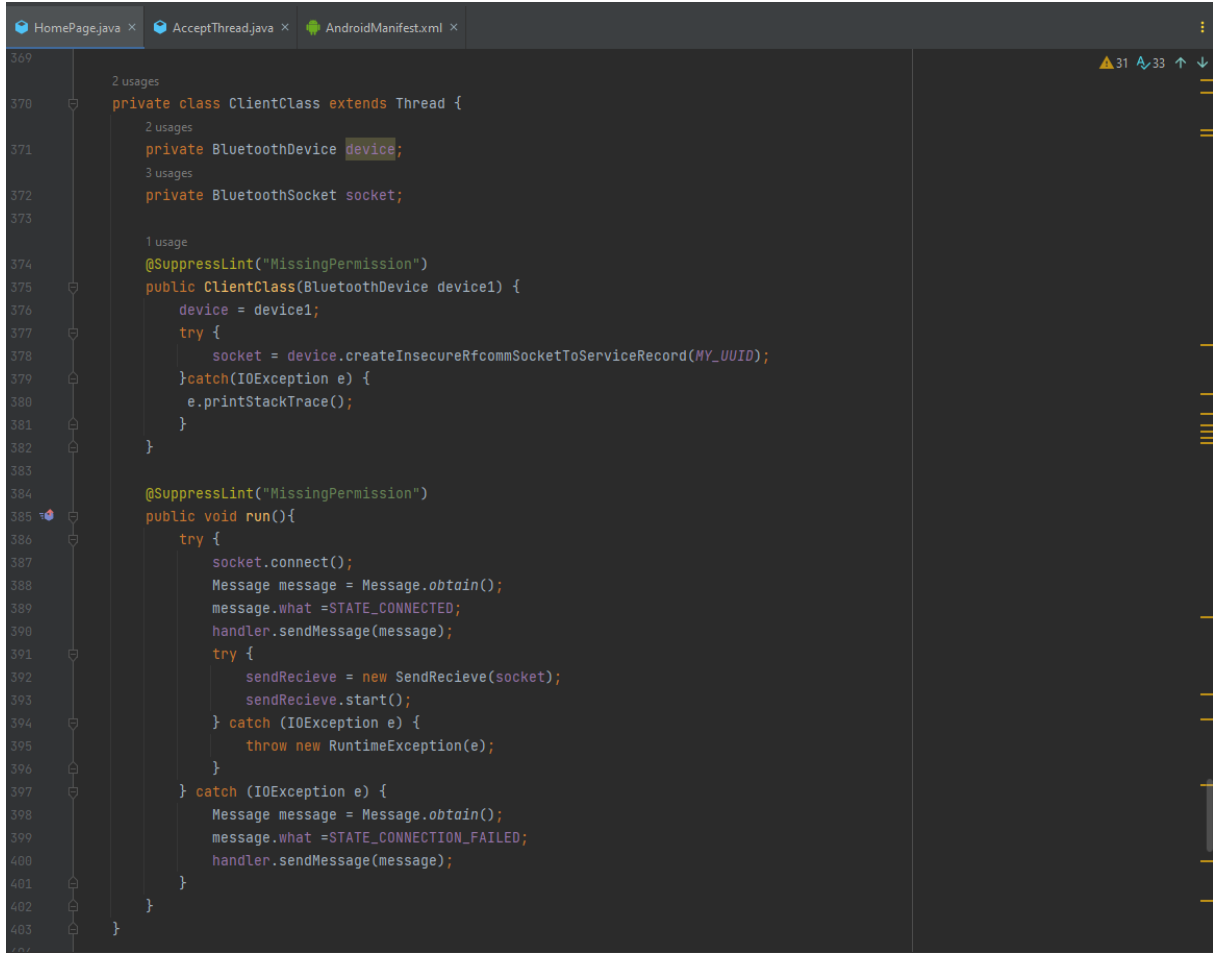
Görseldeki kod bloğunda 195. satırda bağlantı statelerinin gelen mesajlara göre durumları ekrana yazdırıldı. Mesaj alıcı state'ini (STATE_MESSAGE_RECEIVERD) alındığında gelen byte array string buffer yardımı ile geçici bir değişkene atıldı. Bu statü ekrana yazdırıldı. Receiver kısmında gerekli izinler alındı. Receiver kısmında cihazdan tekrar erişim izni alındı. Daha önceden bağlanmış cihazların ismi bir döngü ile bluetooth device ve bluetooth adapterdan alındı. Bunlar RecyclerView (ListView)'e yazdırıldı. Gerekli tıklama eventleri alınıp, sonradan yazılacak olan client sınıfına istek göndermek üzere hazır hale getirildi.



3. Bluetooth Client ve Bluetooth Server'ın Yazılması



Server thread sınıfından türetildi, böylece run isimli methodu implemente edebildi. Bu method sayesinde verileri taşıyan array parçalanarak hedefe aktarılmak üzere hazırlandı.



```
369
370 private class ClientClass extends Thread {
371     private BluetoothDevice device;
372     private BluetoothSocket socket;
373
374     @SuppressWarnings("MissingPermission")
375     public ClientClass(BluetoothDevice device1) {
376         device = device1;
377         try {
378             socket = device.createInsecureRfcommSocketToServiceRecord(MY_UUID);
379         } catch (IOException e) {
380             e.printStackTrace();
381         }
382     }
383
384     @SuppressWarnings("MissingPermission")
385     public void run(){
386         try {
387             socket.connect();
388             Message message = Message.obtain();
389             message.what = STATE_CONNECTED;
390             handler.sendMessage(message);
391             try {
392                 sendRecieve = new SendRecieve(socket);
393                 sendRecieve.start();
394             } catch (IOException e) {
395                 throw new RuntimeException(e);
396             }
397         } catch (IOException e) {
398             Message message = Message.obtain();
399             message.what = STATE_CONNECTION_FAILED;
400             handler.sendMessage(message);
401         }
402     }
403 }
```

Client thread sınıfından türetildi, böylece run isimli methodu implemente edebildi. Bu method sayesinde taşınan veriler parçalanmak üzere handler'a gönderildi.

4.Testler

Yazılan uygulama test edildi, izin hataları onaylanmamış kullanıcı olmak, uygulamanın performansının düşük olması, uzaklığa bağlı bağlantı problemleri nedeni ile “**Web Socket**” üzerinden veri aktarımı yapmaya karar verdik.

YAPILACAK OLAN İŞLER

- Bir sonraki rapor döneminde Web Socket server ve Client kısmı yazılacak.
- Projenin testleri yapılacak.