

KOCAELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM LAB. II

2021-2022 GÜZ DÖNEMİ

PROJE I

180201018-Ayşenur Akpınar

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

aysenurakpinar98@gmail.com

190201100-Rukiye Canlı

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

rukiyecanli435@gmail.com

1.ÖZET

Projede Java programlama dili, Netbeans geliştirme ortamı ve HTML kodları içerisine Java komutlarını kolay bir şekilde yerleştirmek için Java EE API JSP kütüphanesini kullanılmıştır. Web sunucusu olarak Glassfish 4.1 ilişkisel veri tabanı olarak Java DB ve ilişkisel olmayan veri tabanı olarak MongoDB tercih edilmiştir. Harita altlığı için Google API map kullanılmıştır. Bu projede bizden birden fazla kaynaktan (kullanıcı araçlarından) akan GPS verileri alınarak bu verileri ilişkisel olmayan bir veri tabanı aracılığıyla anlık olarak harita üzerinde sergilememiz beklenmektedir.

2.GİRİŞ

Bu kısım projenin açıklamasını okuyup edindiğimiz ön bilgiden yararlanılarak yazılmıştır. Projenin istekleri detaylı olarak aşağıdaki gibidir:

Verilerin alınması için bir mesaj simsarıcısı (message broker) kullanılmalıdır. Ayrıca müşteriler kendilerine verilen hesap bilgileri kapsamında sisteme giriş yapabilmelidir. Her müşteriye sahibi olduğu veya hizmetini aldığı araçların anlık veya en son mevki bilgisi gelen güncellemeler kapsamında, harita altlığı üzerinde sunulacaktır. Müşteri bir aracı seçip, 1-24 saat arasında bir seçim yaparak, sadece o araca yönelik geçmiş güzergahı sergileyebilecektir. Hiç bir şey belirtilmemiş ise en son yarım saatlik mevki bilgileri her bir araç için sergilenecektir.

Proje akışı genel hatlarıyla şu şekilde olmalıdır: Müşteri internet gezgini üzerinde A şirketinin araç takip hizmeti bağlantısını girer veya daha önceden kayıt edilmiş bir sayfa işaretçisini tıklar. Arayüz modülü ana giriş

sayfasını üreterek bir kullanıcı adı ve şifre giriş alanının bulunduğu giriş sayfasını müşteriye sergiler. Müşteri daha önceden açılmış olan hesabı kapsamında kendisine verilen kullanıcı adı ve şifre bilgilerini girer. Arayüz modülü, girilen bilgileri Yönetim/Model modülüne gönderir, Yönetim/Model modülü kullanıcı bilgilerini doğrular. Yönetim/Model modülü tarafından müşterinin sistem giriş zamanı SQL veri tabanına kayıt edilir. Yönetim/Model modülü kullanıcı hesabına göre araç bilgilerin ve her bir araç için son 30dk. verileri hazırlayarak Arayüz modülüne gönderir. Son 30dk veriler performans açısından Arayüz modülü tarafından NoSQL veritabanından direkt çekilebilir. Arayüz modülü müşterinin sahip olduğu araçlara yönelik güncel durumları ve 30dk.lık geçmiş bilgileri ile gösterir.

Yönetim modülü müşteri oturma odasının açıklığını takip eder ve oturma odası açık kaldığı sürece 30dk bilgileri ve son mevki bilgileri günceller. Performans veya gerçekleştirme kolaylığı için arayüz modülü anlık veriyi mesaj brokardan direkt alabilir. Müşteri ekranda bir liste vb. kabiliyet üzerinde gösterilen araç listesinden bir aracı seçerek çift tıklar. Müşteri seçtiği araç için zaman aralığını girer ve giriş tuşuna basar veya varsayılanı onaylar. Arayüz modülü girilen bilgileri Yönetim/Model modülüne gönderir, Yönetim/Model modülü girilen zaman aralığına karşılık gelen bilgileri çıkarıp Arayüz modülüne gönderir. Arayüz modülü gerekli sergilemeyi sağlar. Bu sayfa açık kaldığı sürece araç bilgilerine güncelleme yapılmaz ve sadece sorgulanan araç bilgisi gösterilir. Müşteri çıkış linkine tıklayarak sistemden çıkar. Müşteri oturumu sona erer. Yönetim/Model tarafından müşterinin çıkış zamanları veri tabanına kayıt edilir.

3.YÖNTEM

Projeyi gerçekleştirebilmek için ilk olarak veri setinde bazı düzenlemeler yaptık. Verileri eksik olan araçlara veri ekledik. Tüm verileri birer dakika artacak şekilde ayarladık. Ayrıca dosyada sadece dört araç için veri tuttuk. Veri ön işlemlerinden sonra Front-end kısmına geçtik.

İlk olarak projemizin giriş sayfasını tasarladık. Giriş sayfası aracılığıyla kullanıcılar önceden sahip oldukları hesap bilgileri sayesinde sisteme başarılı bir şekilde giriş yapabilirler. Eğer giriş yapma işlemi üç defadan fazla tekrar edilirse kullanıcı bir uyarı mesajı alacak ve sisteme giriş yapamayacaktır. Giriş sayfası HTML, CSS ve JSP dilleri ile gerçekleştirdik.

Sisteme giriş yapan kullanıcı, araç ve zaman aralığı seçim sayfasına yönlendirilecektir. Bu sayfada kullanıcı sahip olduğu araçlar arasında harita sayfasında görmek istediği tüm araçları seçebilir. Ayrıca bu araçları göstermek istediği zaman aralıklarını ekranda gördüğü seçenekler sayesinde seçebilir. Bu işlemlerden sonra kullanıcı harita sayfasına yönlendirilecektir. Bu sayfada kullanıcı araçların yol güzergâhlarını takip edebilir. Eğer kullanıcı zaman aralığı seçimi yapmadıysa otomatik olarak araçların son yarım saatlik yol güzergâhını görebilecektir.

Kullanıcı verilerini ilişkisel bir veri tabanında tutmamız gerektiğinden MySQL veri tabanında tuttuk. Burada kullanıcı adı, şifresi, siteye giriş ve çıkış yaptığı saatler için alanlar oluşturulmuştur. Araç verilerini çekip ilişkisel olmayan bir veri tabanına atmamız gerektiği için fakat bu verileri MongoDB veri tabanında tuttuk. Burada bir database yarattıktan sonra her bir araç için bir doküman(document) oluşturduk ve her bir aracın araç id, saat, enlem ve boylam bilgileri için alanlar oluşturduk. Proje kapsamında bizimle paylaşılan İsveç taksi verisini kullandık. Bu verilerin allCars.csv dosyasından veri tabanına bir mesaj simsarcısı aracılığıyla çekilmesi istenmişti ancak bu kısmı gerçekleştiremediğimiz için verileri csv dosyasından veri tabanına direkt olarak import ettik.

Front-end işlemleri sonrasında Back-end kısmına geçtik. Kullanıcı harita göster butonuna bastığında seçilen araç ve zaman bilgileri Java'da yazdığımız iki fonksiyondan birine gönderilir. Eğer kullanıcı zaman aralığı seçimi yapmadıysa ilk fonksiyon olan FileWriter(String carId, String time) fonksiyonuna parametre olarak araç id ve o anki saat bilgisi gönderilir. Kullanıcı zaman aralığı seçimi yaptıysa ikinci fonksiyon olan FileWriter(String carId, String time1, String time2) fonksiyonuna saat bilgileri ve araç id bilgisi gönderilir. Bu fonksiyon aracılığıyla Java ve MongoDB arasında bağlantı kurulur. Kullanıcının seçmiş olduğu araç ve zaman bilgilerine göre veri tabanından lokasyon bilgileri çekilir. Verilerimiz birer dakika arayla olduğu için örneğin

18:00-19:00 arasında kendi araç güzergahını görmek isteyen bir kullanıcı için veri tabanından 120 lokasyon

verisi çekilecektir. Çekilen bu veriler fonksiyon içerisinde bir matrise atılır. Elde edilen bu matrisi

ekranda harita göstermek için oluşturduğumuz Javascript dosyasına göndermemiz gerekiyordu. Bunu nasıl yapabileceğimize dair araştırmalar yaptık. Bir veriyi Json formatıyla Javascript'e gönderebileceğimizi okuduk. Ayrıca REST-API, Ajax gibi yeni nesil tekniklerin de kullanılabileceğini gördük fakat bu işlemleri gerçekleştiremediğimiz için oluşturduğumuz matrisi bir text dosyasına attık. Js dosyasında bu text dosyasını okutup matrisi elde etti. Js dosyası aracılığıyla harita üzerinde matristen alınan her bir lokasyon bilgileri için yeni bir işaret oluşturulur. İşaretler sayesinde kullanıcı, araçlarının nerede olduklarını rahatlıkla gözlemleyebilir.

Kullanıcı birden fazla aracı aynı anda ekranda görebilir. Ayrıca iki ayrı kullanıcı aynı anda farklı haritalar üzerinde kendi araçlarını görebilir. Kullanıcının birden fazla aracı aynı anda ekranda görebilmesi kısmı işimizi biraz zorlarsa da bazı işlemler sonucunda bu isteri de yerine getirebildik. Lokasyon bilgilerini text dosyasına yazdırdığımız için seçilen ikinci aracı da bu text dosyasında tutmamız gerekiyordu. Bunun için FileWriter(file, append) metodunu kullandık.

Append=true olarak ayarladığımızda dosyaya en baştan veri yazdırmak yerine birinci araç verilerinin hemen altına ikinci araç verilerini yazdırmış olduk.

Projenin minimum değerlendirme koşulu, bir müşteriye ait bir araç verisinin anlık olarak veri tabanında depolanması ve web arayüzünde sergilenmesi, gelen güncellenmelerin web arayüzünde gösterilebilmesiydi. Özellikle bu kısımlara dikkat edip projemizi tamamladık. Projemiz araç verilerini veri tabanında depolayabilir, bu bilgileri harita üzerinde sergileyebilir ve güncellemeler ekranda gösterilebilir.

4.YALANCI KOD

Kullanıcı araç seçimi yapıp zaman aralığı seçimi yapmadan “Haritayı göster” butonuna basarsa

getLocation(carId, currentTime) fonksiyonunu çağır

```
getLocation(carId, currentTime){
```

Mongo ile Java arasında bağlantı kur

Adresses koleksiyonunu çağır

Bu koleksiyonda doküman aramak için query nesnesi yarat

Lokasyonları tutacak String lokasyonlar[][] matrisini oluştur

Yarım saati tutmak için bir integer sayaç oluştur

Fonksiyona gelen şimdiki saati(currentTime) String'e dönüştür

Şimdiki saatten yarım saat öncesini hesaplatıp yeni bir String'e(newTime) ata

```
While(!newTime.equals(currentTime)){
```

Query.append ile veri tabanında araç ve saate göre veri araması sorgusu oluştur

Find(query) ile veri tabanında aranan veriler bul

Obj.getString("lat") ve Obj.getString("long") metodları ile alınan verinin lokasyonlarını String'e çevir

currentTime değerini bir dakika geri al

Sayac++;

```
}
```

```
File file = new File(123.txt);
```

```
FileWriter out;
```

```
İf(file.isNull()){
```

```
Out = new FileWriter("123.txt");
```

```
For(int i =0;i<lokasyon.length(); i++){
```

```
Out.write(lokasyon[i][0]+" "+lokasyon[i][1]);
```

```
}
```

```
}Else{
```

```
Out = new FileWriter("123.txt");
```

```
For(int i =0;i<lokasyon.length(); i++){
```

```
Out.write(lokasyon[i][0]+" "+lokasyon[i][1]);
```

```
}
```

```
}
```

Kullanıcı araç ve zaman aralığı seçimi yapıp "Haritayı göster" butonuna basarsa

getLocation(carId, time, time2) fonksiyonunu çağır

```
getLocation(carId, time, time2){
```

Mongo ile Java arasında bağlantı kur

Adresses koleksiyonunu çağır

Bu koleksiyonda dokuman aramak için query nesnesi yarat

Lokasyonları tutacak String lokasyonlar[][] matrisini oluştur

İki saat arasındaki farkı tutmak için bir integer sayaç oluştur

Fonksiyona gelen son saati String'e dönüştür

String olan son saati(time2) gerçek bir saat tipine dönüştür

```
While(!time.equals(time2)){
```

Query.append ile veri tabanında araç ve saate göre veri araması sorgusu oluştur

Find(query) ile veri tabanında aranan veriler bul

Obj.getString("lat") ve Obj.getString("long") metodları ile alınan verinin lokasyonlarını String'e çevir

Time2 değerini bir dakika geri al

Sayac++;

```
}
```

```
File file = new File(123.txt);
```

```
FileWriter out;
```

```
İf(file.isNull()){
```

```
Out = new FileWriter("123.txt");
```

```
For(int i =0;i<lokasyon.length(); i++){
```

```
Out.write(lokasyon[i][0]+" "+lokasyon[i][1]);
```

```
}
```

```
}Else{
```

```
Out = new FileWriter("123.txt");
```

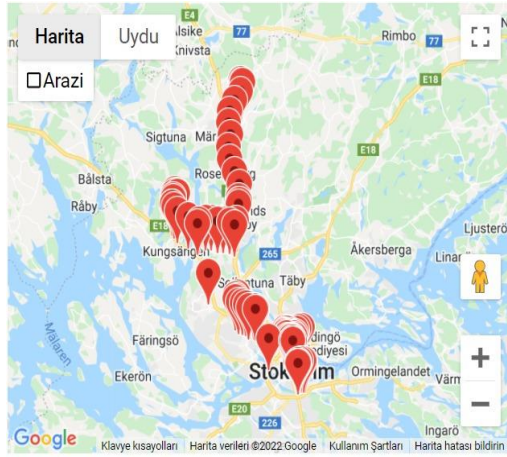
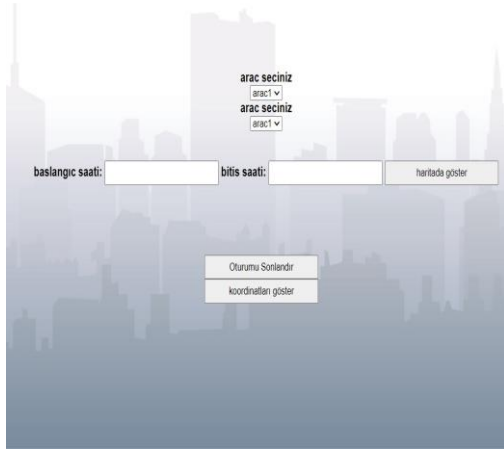
```
For(int i =0;i<lokasyon.length(); i++){
```

```
Out.write(lokasyon[i][0]+" "+lokasyon[i][1]);
```

```
}
```

```
}
```

4. DENEYSEL SONUÇLAR



6. SONUÇ

Java dili kullanarak oluşturduğumuz web sitesi aracılığıyla kullanıcı kendisine verilen hesap bilgileri aracılığıyla sisteme giriş yapabilir. Üç defadan fazla giriş yaparsa bir uyar mesajı alır. Sisteme giriş yapabilen kullanıcılar arayüz aracılığıyla ekranda görmek istedikleri araçları ve araçların zaman aralıklarını seçebilir. Zaman aralığı seçmeyen bir kullanıcı son yarım saatlik verileri görebilir. Kullanıcılar bu araçların lokasyonlarını harita üzerinde rahatlıkla anlık olarak gözlemleyebilir. Proje sürecinde karşılaştığımız bazı zorluklara(message broker, back-end ve front-end arası veri aktarımı) rağmen projemizi tamamladık ve proje sonucunda proje dökümanında istenilen çoğu isteri gerçekleştirmiş olduk.

7. KAYNAKÇA

https://www.natro.com/blog/mongodb-nedir-kullaniciya-sundugu-avantajlar-nelerdir/?gclid=Cj0KCQiAmpyRBhC-ARIsABs2EAphuyay7GARCUatbaSZIJqurTOIxORWrtiIiJQtTu5L5fJ2Zz9xtmwaAiGVEALw_wcB

https://www.youtube.com/watch?v=sX_mKdZ3qp4

<https://www.youtube.com/watch?v=6hsXeNSFWnw>

<https://mongodb.github.io/mongo-java-driver/>

<https://www.youtube.com/watch?v=v3kQ-0xJgzQ>

<https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/android-101/javada-dosya-islemleri>

<https://stackoverflow.com/questions/9015536/how-to-add-10-minutes-to-my-string-time>

<https://stackoverflow.com/questions/59769376/search-for-a-value-inside-a-document-mongodb>

<https://stackoverflow.com/questions/49351629/mongodb-java-3-4-get-fields-from-embedded-document>

<https://stackoverflow.com/questions/19129414/how-to-get-values-from-mongodb-document-java>

<https://stackoverflow.com/questions/12670195/mongodb-query-in-java-search-find-in-nested-object>

<https://www.geeksforgeeks.org/mongodb-findone-method/>

<https://developers.google.com/maps/documentation/javascrypt/markers>

<https://laratutorials.com/add-multiple-markers-on-google-map-javascript/>

https://www.w3schools.com/jsref/jsref_split.asp

<https://stackoverflow.com/questions/15212832/how-to-save-current-date-and-time-to-database-using-java>

