

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü (İÖ)

Yazılım Laboratuvarı II

1. Proje

Çok Katmanlı Web Uygulaması

Ahmet Aydın
190202081@kocaeli.edu.tr

Ömer Faruk Delipoyraz
190202129@kocaeli.edu.tr

Özet:

Bu rapor Yazılım Laboratuvarı 2 dersinin 1. projesi için hazırlanmakta olup projenin tanıtımından başlayarak sırasıyla projenin nasıl tamamlandığı, karşılaşılan sorunların çözümünde kullanılan yöntemler, kullanılan programlama dili ve IDE, kod bilgisi ve kaynakça içermektedir.

Proje Tanıtımı:

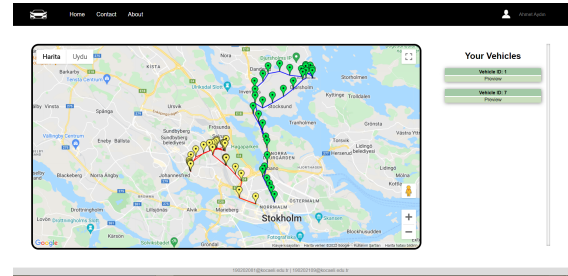
Yazılım Laboratuvarı 2 dersinin 1. projesi olarak bizden çok katmanlı bir web uygulaması geliştirmemiz istendi.

Bu web uygulamasında birden fazla araçtan akan GPS verileri alınarak bir NoSQL veritabanında depolanacaktır. Müşteri hesap bilgileri bir SQL veritabanında tutulacaktır. Gelen anlık araç bilgileri harita altyapısının olduğu bir web arayüzünde sergilenecektir. Müşteri kendisine verilen hesap bilgileri ile sisteme giriş yapabilecektir. Her müşteriye sahibi olduğu yada hizmetini aldığı araçların en son veya anlık verileri gelen güncellemeler kapsamında harita altlığı üzerinde sunulacaktır. Müşteri sahibi olduğu olduğu araçların listesinden istediği aracı seçerek bu araçla ilgili geçmişe dönük konum sorgusu yapabilecektir. Müşteri ilgili listeden istediği aracı seçerek 1-24 saat aralığında sorgu yapabilecektir.

Geliştirilen web uygulaması çok katmanlı mimariye sahip olacaktır. Model,

Arayüz ve Yönetim kalıbının kullanılması ile arayüz, işlem mantığı ve verinin ayrıştırılması beklenmektedir.

Projede kullanılan sistem mimarisi raporun sonunda Akış Diyagramı bölümünün hemen ardında yer almaktadır.



Resim 1: Kullanıcı arayüzü

Araştırma ve Yöntem:

Projeyi geliştirmeye ilk olarak kullanılacak teknolojileri seçerek başladık. Çok katmanlı bir web uygulaması geliştireceğimiz proje kapsamında kullanmamız gereken teknolojiler araç verilerinin akışını gerçekleştirecek bir message broker, araç bilgilerinin kaydedildiği bir NoSQL veritabanı, müşteri bilgilerinin bulunduğu bir SQL veritabanı ve araçların konum bilgilerinin gösterildiği bir harita altyapısı olacaktı.

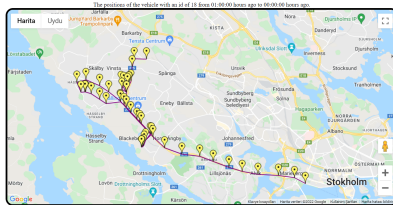
İlk olarak harita yapısı için Google Maps Api kullanmaya karar verdik. Bu seçimi yapmamızdaki etkenler hem kapsamlı olarak

yazılmış dokümantasyonu hem de ulaşma ve kullanım kolaylığı oldu.

NoSQL veritabanı olarak MongoDB, sql veritabanı olarak ise SQL Server kullanmaya karar verdik.

Message broker için başlangıçta ZeroMq kullanmaya karar versek de geliştirmenin ileriki aşamalarında projenin yapısındaki değişiklikleri de göz önünde bulundurarak message broker kullanımından vazgeçtik ve araç bilgilerini doğrudan NoSQL veritabanına kaydettik.

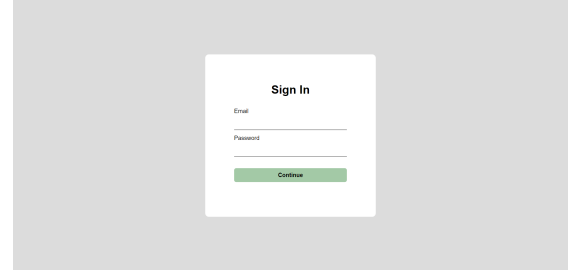
Kullanılacak teknolojilere karar verdikten sonra sıra web arayüzünün tasarımına geldi. Proje kapsamında müşteriye sunmamız gereken üç farklı ekran gerekiyordu. Müşterinin sisteme giriş yapacağı login ekranı, araç listesini ve araçların son yarım saatlik verilerini harita üzerinde görüntüleyeceği ana ekran ve listeden seçeceği araç ile ilgili konum bilgilerini görüntüleyeceği harita ekranı. Bu üç ekranın tasarımını tamamladıktan sonra sıra kodlamaya ve web sitesini ayağa kaldırmaya geldi.



Resim 2: Aralığın gösterildiği arayüz

Visual Studio üzerinden bir ASP.Net MVC projesi başlatarak projeyi geliştirmek için ilk adımımızı attık. Sonrasında gerçekleştirdiğimiz tasarım kapsamında bir login arayüzü oluşturduk. Login arayüzünü oluşturduktan sonra giriş işlemlerini gerçekleştirecek olan fonksiyonları gerçekleştirdik. Burada dikkat ettiğimiz nokta kullanıcının 3 defa ve daha fazla yanlış giriş yaptığında sisteme erişememesi gerektiği sorunu oldu. Bunun için SQL veritabanında kullanıcı tablosuna yanlış giriş sayısı ve askıya alınma durumu ile ilgili iki ayrı sütun ekledik. Aynı zamanda kullanıcı sisteme giriş

yaptığında giriş zamanını veritabanına kaydeden fonksiyonu gerçekleştirdik.



Resim 3: Login arayüzü

Daha sonra kullanıcının araç listesini ve lokasyon izlerini görüntüleyeceği ekranı hazırladık. Bu ekran üzerinde kullanıcı bilgileri, araç listesi ve harita altyapısı görüntülenecektir. Bu ekran açıldığında veri tabanlarından kullanıcı ve araç bilgileri çekerek sayfa üzerinde gösterdik. Kullanıcının bu ekranda liste üzerinden üzerinden bir araç seçerek o araçla ilgili 1-24 saat aralığında sorgu gerçekleştirebilmesi gerekiyordu. Bunun için kullanıcı araçların üzerine tıkladığında açılan bir diyalog ekranı oluşturduk. Bu ekran üzerinde girilen bilgileri kontrolcü modülüne gönderdik ve bu modülde yaptığımız sorgu sonucu dönen verileri arayüz modülüne geri döndürdük.

Sonrasında ise kullanıcı sorguyu oluşturduğunda açılacak olan sayfayı ve bu sayfanın fonksiyonlarını gerçekleştirdik. Bu sayfada kontrolcü modülünden dönen verileri bir harita altyapısı üzerinde gösterdik.

Son olarak kullanıcının çıkış işlemleri ile ilgili fonksiyonları gerçekleştirdik ve kullanıcının çıkış zamanını veritabanına kaydeden fonksiyonu yazdık.

1. Message Broker:

Proje kapsamında message broker araçlardan gelen anlık verinin bir kuyruğa sokularak NoSQL veri tabanında depolanması ve kullanıcı arayüzünde gerçek zamanlı olarak güncellenmesi için kullanılacaktı. Güncelleme istisinin iptal edilmesi ile projede message broker kullanılmamıştır. Araç verileri doğrudan NoSQL veritabanına kaydedilmiştir.

Kazanımlar:

Yazılım Laboratuvarı 2 dersi 1. proje kapsamında çok katmanlı bir web uygulaması geliştirerek dağıtık bir sistem mimarisi kullanarak çözüm oluşturma, güncel bir web kütüphane altyapısının kullanımının öğrenilmesi ve uygulanması ve değişik tipte veri tabanlarının kullanılması ile gerçek dünya ihtiyaçlarını dikkate alan uygulama yeteneği kazandık.

Uygulamanın İşleyişi:

Uygulama açıldığında kullanıcıyı **Resim 3**'te görülen giriş ekranı karşılamaktadır. Kullanıcı bu giriş ekranı üzerinden kendisine verilen hesap bilgileri ile giriş yapabilmektedir. Eğer kullanıcı 3 ve daha fazla sayıda hatalı giriş yapar ise hesabı askıya alınacaktır ve artık sisteme giriş gerçekleştiremeyecektir. Kullanıcı hesabına giriş yaptığından giriş zamanı SQL veritabanına kaydedilecektir.

Kullanıcı sisteme giriş yaptıktan sonra **Resim 1**'de görülen kullanıcı arayüzü açılacaktır. Bu arayüz üzerinde kullanıcı sahip olduğu araçların listesini ve araçlarının son 30 dakikalık lokasyon izlerini görebilecektir. Kullanıcı araç listesinden bir araç seçip üzerine tıkladığında bir diyalog ekranı açılacaktır. Bu diyalog ekranı üzerinde kullanıcı istediği saat aralığını seçerek ilgili aracın girilen saat aralığındaki güzergah bilgilerini görüntüleyebilecektir. Bu güzergah bilgileri açılan yeni bir sekme üzerindeki harita altyapısında kendisine sunulacaktır. Kullanıcı istediği araçla ilgili istediği sayıda sorgulama yapabilmektedir.

Kullanıcı sistemde işini bitirip çıkmak istediğinde ana ekranın sağ üstünde isminin bulunduğu kısma mouse'unu hareket ettirerek açılan listeden çıkış seçeneğini seçtiğinde sistemden sistemden çıkışı gerçekleştirilecektir. sistemden çıkış yapan kullanıcının çıkış zamanı veritabanına işlenecektir.

Tasarım:

1. Akış Diyagramı:

Akış diyagramı raporun sonunda Referanslar bölümünün hemen ardında yer almaktadır.

2. Sitem Mimarisi:

Sistem mimarisi raporun sonunda Akış Diyagramı bölümünün hemen ardında yer almaktadır.

3. ER Diyagramı:

SQL veritabanı ER diyagramı raporun sonunda sistem Mimarisi bölümünün hemen ardında yer almaktadır.

Programlama Dili ve Geliştirme Ortamı:

Bu proje Windows işletim sistemi üzerinde Visual studio 2019 ortamında Asp.Net kullanılarak gerçekleştirilmiştir.

sql veritabanı olarak sql server, Nosql veritabanı olarak ise MongoDB kullanılmıştır. Harita altyapısı için Google Maps Api'den yararlanılmıştır. Projede Message Broker kullanılmamıştır.

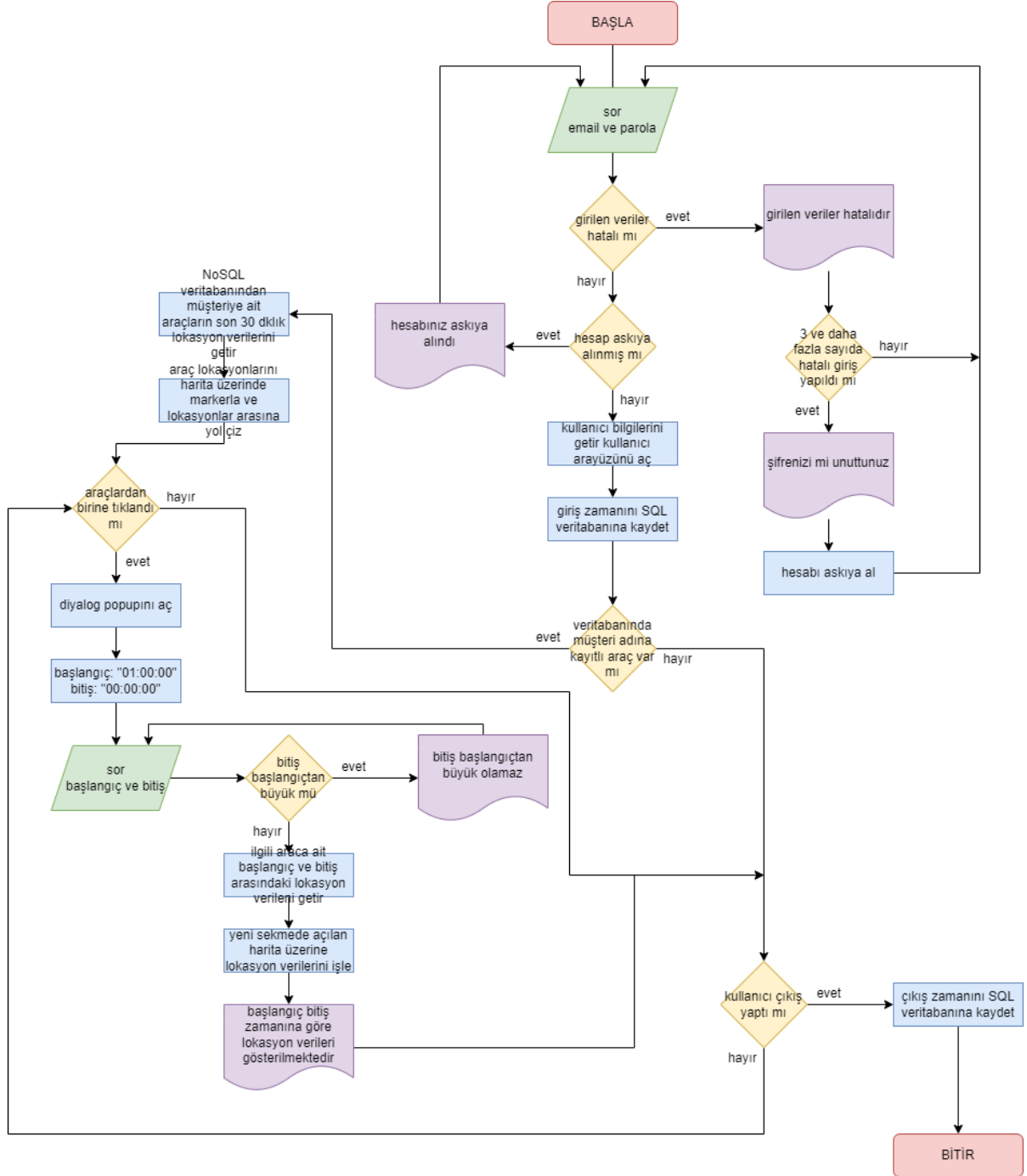
Referanslar:

Web Sitesi:

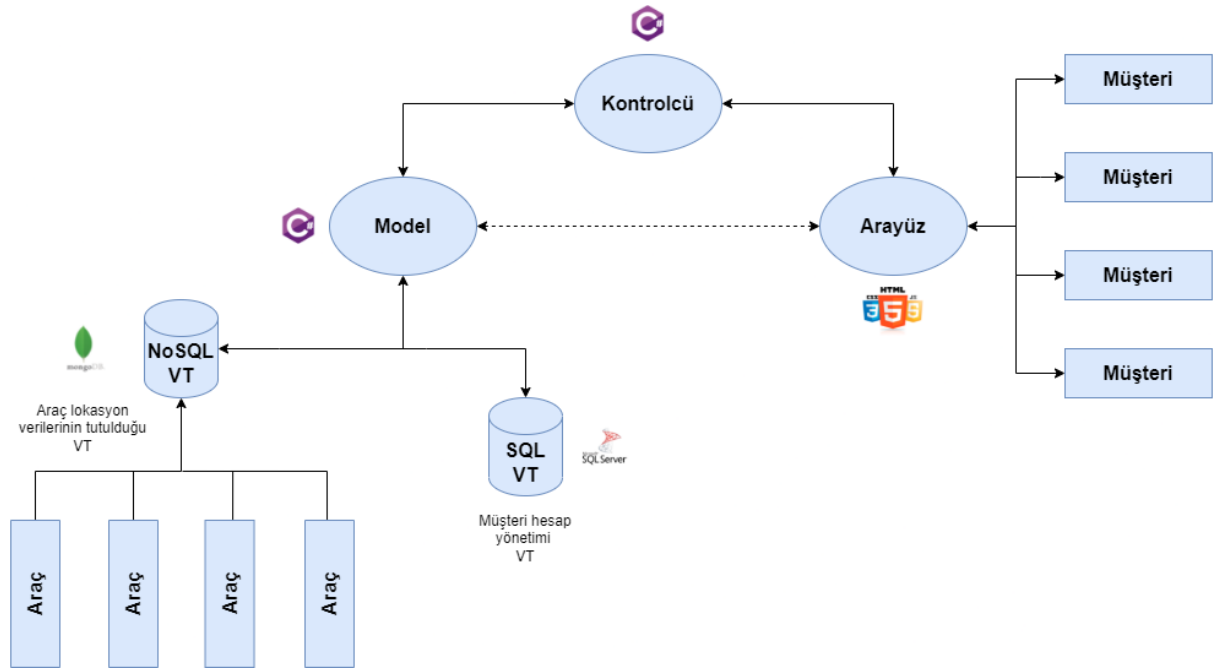
1. <https://zeromq.org/get-started/>
2. <https://docs.mongodb.com/>
3. <https://docs.microsoft.com/tr-tr/aspnet/mvc/overview/getting-started/introduction/getting-started>
4. <https://developers.google.com/maps/documentation>
5. <https://stackoverflow.com/>

Tasarım:

1. Akış Diyagramı



2. Sitem Mimarisi:



3. ER Diyagramı:

