Üst Seviye Tasarım Dökümanı

1. Genel Bakış

1.1 Projenin Tanımı

Mentoree, çeşitli alanlarda uzman olan kişilerin uzmanlıklarını uygulamayı kullanan diğer kullanıcılara mentörlük şeklinde sunmasını sağlayan bir web uygulamasıdır.

1.2 Projenin Özellikleri

Projeyi kullanabilecek iki farklı kullanıcı rolü tanımlanmıştır. Bunlar admin ve user rolleridir.

User olan kullanıcılar Mentoree uygulamasını kullanırken şu özellikleri kullanabilmektedirler:

- Mentörlük Başvurusu Oluşturma
- Mentör Arama
- Mentör-Mentee İlişkisi Oluşturma
- Mentörlük İlişkisi Detayı Görüntüleme

Admin olan kullanıcılar Mentoree uygulamasını kullanırken şu özellikleri kullanabilmektedir:

- Mentörlük Başvurularını Onaylayıp Reddetme
- Mentörlük Konu ve Alt Konularını Düzenleme

1.3 Kullanılan Teknolojiler

Thymeleaf: Mentoree, geliştirilme sırasında MVC paternini kullandığından ötürü bir view teknolojisine ihtiyaç duyulmuştur. JSP gibi teknolojiler araştırılmış fakat okunabilirlik ve kullanılabilme kolaylığı sayesinde bu template engine teknolojisi seçilmiştir.

Spring Boot: Uygulamanın işlevlerini rahat gerçekleştirebilmek adına dökümantasyonu ve kullanım alanı geniş bir framework olan Spring Boot tercih edilmiştir. Aynı zamanda Spring MVC özelliklerini Spring Boot uygulaması içerisinde kullanılabilmektedir. Bu da Spring Boot'un seçilmesinde ayrı bir nedendir.

Spring Security: Spring Boot ile uyumlu olan bir güvenlik frameworküdür. LDAP ve OAuth2 kimliklendirmelerini ve kullanıcı rollerini tanımlama amaçlı kullanılmıştır.

MySQL: Uygulamanın ana bilgilerini tutan veritabanı için MySQL seçilmiştir çünkü hem kullanıcı arayüzü hem de dökümantasyon olarak kolay bir kullanıma sahiptir.

MongoDB: Uygulamanın Free Text Search kısmı için Mentör bilgilerini tutmaya yarayan noSQL veritabanı için MongoDB tercih edilmiştir çünkü kullanıcı arayüzü, dökümantasyonu ve Spring Boot ile olan entegrasyonu diğer teknolojilere göre daha rahat görünmüştür.

JQuery: Uygulamanın Javascript gerektiren interaktif özellikleri için JQuery kütüphanesi seçilmiştir. Bunun nedeni ise dökümantasyon zenginliği ve köklü bir kütüphane olmasıdır.

Bootstrap 4: Uygulamanın front-end kısmına ek özellikler katabilmek adına dökümantasyonu geniş ve rahat kullanıma sahip bir framework olan Bootstrap 4 tercih edilmiştir.

HTML5/CSS3: Uygulamanın front-end kısmı için kullanılmıştır.

Maven: Gerekli dependencyleri uygulamaya entegre edebilmek ve Spring Boot uygulamasını build edebilmek adına kullanılmıştır.

Spring Mail: Devam eden fazdan 1 saat önce mail gönderebilmek adına Spring Boot ile uyumlu bir mail kütüphanesi olduğundan seçilmiştir.

1.4 Varsayımlar

MySQL veritabanına veriler yöneticiler tarafında elle girilecek.

MongoDB veritabanına girilen bilgiler ile MySQL veritabanında var olan Mentor tablosunun içerikleri aynı olacak, eksik ya da fazla mentör bulunmayacak.

Embedded LDAP için kullanılan ldap-data.ldif dosyasındaki bilgiler ile MySQL veritabanında var olan User tablosunun içerikleri aynı olacak, eksik ya da fazla mentör bilgisi bulunmayacak.

2.Proje Detayları

2.1 Kullanıcı Kimliklendirmesi

Uygulamaya giriş iki türlü yapılabilmektedir:

- Kullanıcı adı ve şifre ile giriş: Projenin içinde bulunan ldap-data.ldif dosyasında bulunan kullanıcılar arasından bir kontrol yapılarak ve MySQL veritabanından kullanıcı kontol edilip uygulamaya giriş sağlanmaktadır.

- Google ile giriş: OAuth2 kimliklendirme sistemi kullanılarak login öncesi MySQL veritabanından kullanıcı kontrol edilip uygulamaya giriş sağlanmaktadır.

2.2 Hata Yönetimi

Uygulamada var olabilecek çoğu hata kullanıcıya bildirilmiştir. Bunlar:

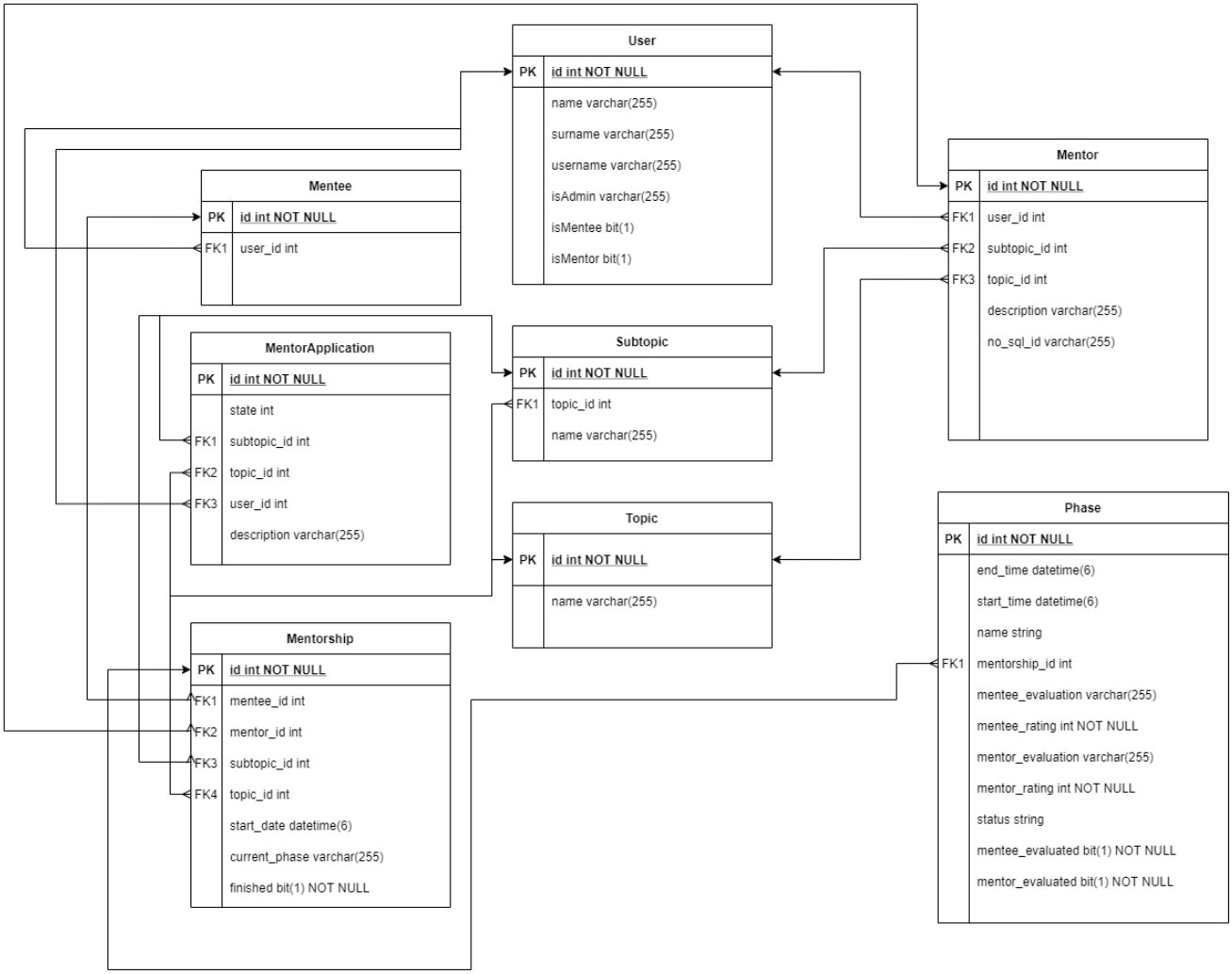
- Adminin var olan bir konuyu tekrar eklemeye çalışması,
- Adminin var olan bir alt konuyu tekrar eklemeye çalışması,
- Adminin mentörlük ve mentör-mentee ilişkisine ait bir konuyu silmeye çalışması,
- Adminin mentörlük ve mentör-mentee ilişkisine ait bir alt konuyu silmeye çalışması,
- Kullanıcının hali hazırda mentör olduğu bir alana tekrar mentörlük başvurusu yapmaya çalışması,
- Kullanıcının beklemede olan bir başvurusu ile aynı konu ve alt konuyu içeren başka bir başvuru yapmaya çalışması,
- Mentör-Mentee ilişkisi kurulurken eğer mentörün mentee sayısı 2'ye ulaştıysa,
- Kullanıcı zaten varolan bir Mentör-Mentee ilişkisini tekrar oluşturmaya çalışıyorsa,
- Kullanıcı giriş yaparken yanlış bilgi girerse,
- Mentör ve menteeler başlamış bir süreci tekrar başlatmaya çalışırsa,
- Mentör ve menteeler faz tanımlanmamış bir süreci başlatmaya çalışırsa,

gibi durumlarda uygulamanın çalışmasını engelleyecek unsurlar oluşmaması adına hata kontrolleri yapılmış ve kullanıcı bilgilendirilmiştir.

2.3 Veritabanı Tasarımı

Uygulamada iki farklı veritabanı kullanılmıştır bunlardan biri ilişkisel bir MySQL veritabanıdır, NoSQL bir veritabanı olan MongoDB veritabanıdır. Bu iki veritabanının kullanılmasının nedeni uygulamanın ilişkisel bir yapısı olması fakat sadece mentör arama kısmı için free text search uygulanabilecek bir NoSQL veritabanı gereksinimidir.

2.3.1 MySQL Veritabanı Tasarımı



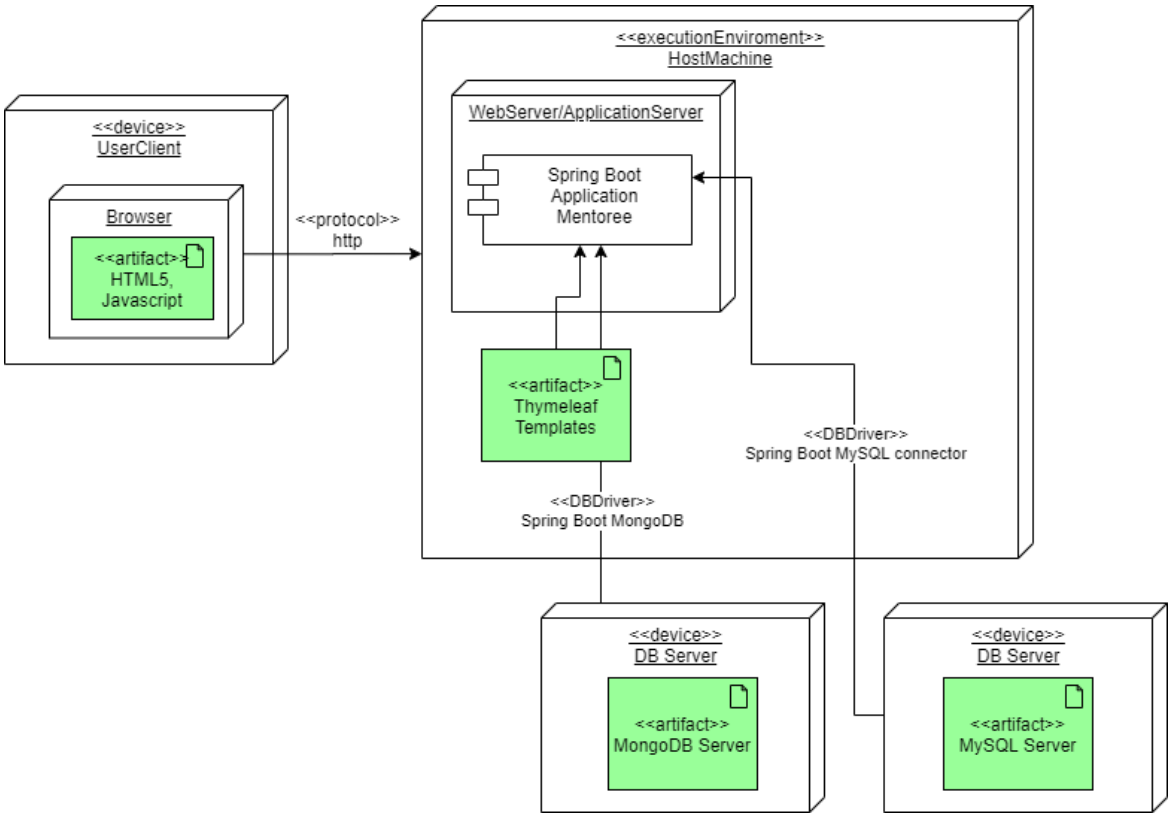
2.3.2 MongoDB Veritabanı Tasarımı

MongoDB'de tutulan mentor yapısı aşağıdaki gibidir. Bu yapı, uygulamanın içerisinde MentorNOSQL modeli tarafından temsil edilmektedir.

Mentor
_id: ObjectId
name: string
surname: string
topic: string
subtopic: string
username: string
description: string

3. Tasarım Detayları

3.1 Fiziksel Deployment Diyagramı



3.2 Use Case Diyagramı