

Üst seviye tasarım dökümanı

Alperen Alkan

1. Genel Bakış

1.1 Projenin tanımı

Proje mentorların ilgili oldukları alanlarda mentee'lere mentörlük sağladığı bir web uygulamasıdır.

1.2 Projenin Özellikleri

Projeyi iki farklı kullanıcı rolü kullanır : Admin ve userlar

User olan kullanıcılar projede şu özellikleri kullanabilir:

1. Mentor arama
2. Mentörlük başvurusu yapma
3. Mentörlük süreci başlatma
4. Mentörlük ilişkisi detayı görüntüleme
4. Mentörlük süreç faz belirleme

Admin olan kullanıcılar şu özellikleri kullanabilir:

1. Mentör başvuru onaylayıp reddetme
2. Konu ve alt konu düzenleme

1.3 Kullanılan Teknolojiler

Spring boot: Uygulamanın sunucu tarafı işlevlerini rahat gerçekleştirebilmek için geniş bir community e sahip olan Spring bootu seçtim.

Spring security: Spring boot ile uyumlu olan bu frameworkle LDAP ve Oauth2 kimliklendirmeleri kullanıcı rollerini kullandım. Bunun yanında giriş yapan kullanıcının usernameini kolay bir şekilde spring security yardımıyla backend'e aldım.

Postgresql: Uygulamanın mentee, mentorship application, topic gibi tablolanması gereken verileri tutan bir SQL DB seçilmiştir.

HTML/CSS: Frontend react kısmında kullanılmıştır.

Maven: Dependencyleri uygulamaya entegre edilmek için kullanılmıştır.

React(Javascript): Sektör de çok kullanılan bu teknolojiyi frontend tarafında kullandım. Frontend için başka bir kütüphane veya teknoloji kullanmamamdaki sebep ise React'ı öğrenip gelişme yapacağının iyi bir başlangıç olduğunu düşündüm.

Material ui: Reactle kullanımı oldukça rahat olan bu framework frontendteki tasarımı ve component kullanımını kolaylaştırdı ve güzelleştirdi.

ElasticSearch: NoSQL data base kullanmadığım için ayrıntıya göre search yapmak için bu teknolojiyi seçtim. Bu teknolojiyi docker ile kullandım

Swagger: Api dokümantasyonu yapmak için kullandım. <http://localhost:8080/swagger-ui/#/> de api dökümantasyonu görsel bir şekilde gözükmetedir.

2- Çözüm mimarisi ve Alınan Kararların Sebebi

Uygulamanın giriş kısmında kullanıcılar Embedded Ldap(Ldap-data.Idif) ile authenticate ediliyor ve rolleri bu dosyadaki rollerine göre belirleniyor. Eğer adminler ise frontendte user ile ilgili page lere gitmeleri yasak ve eğer userlarsa admin pagelerine giremiyorlar. Bunlara ek olarak login olmamış bir kullanıcı sayfalarda gezemiyor çünkü react tarafında protected routes oluşturuldu. Her mentee mentor olabileceğinden ve mentorlar da menteelik işlevlerini gösterebildiğinden bunları

mentee adı verilen tek bir modelde oluřturdum (Farklı databaselere kaydetmeye gerek duymadım). Lakin isMentor flagı ile mentor olanların rolleri belirlenebildi ve ayrıca mentorlar daha sonraki search kısmı için Elasticsearch repositoryisine kaydedildi (burda zaten elasticsearch reposuna kaydettiğim için ekstra database e kaydetmek bir kere daha mantıklı gelmedi). Sonuç olarak menteeler metorluk başvurusuyla adminler tarafından onaylanıp isMentor flagıyla mentor olur ve elasticsearch reposuna kaydedilir. Mentorluk detayı ise doldurulacak olan bir phase list ve mentorluk sürecine dair genel bilgilerden oluşur. Ve bu listeye userlar fazlar ekleyerek daha sonra bu fazları değerlendirip bir diğer faza geçebilirler.

Hata yönetimi

1- Eğer free text search sonucu bir sonuca ulaşılmazsa normalde frontend backendden doğru bir response almadığı için çöküyordu ama bunun backend tarafında error handlingi yapıldı.

2- Eğer bir mentee birden fazla süreçte mentee olmak isterse bu durum response olarak dönülür ve eğer bir mentor 2den fazla menteeye mentorluk yapmak isterse bu da response olarak dönülür.

3- Proje Diagramları

1- Database Diagramı

mentorship	
id	bigint
current_phase	integer
has_phase	boolean
mentee_name	varchar(255)
mentee_thoughts	varchar(255)
mentor_name	varchar(255)
mentor_thoughts	varchar(255)
number_of_phases	integer
start_date	varchar(255)
status	varchar(255)
main_topic	varchar(255)

mentor_applications	
id	bigint
main_topic	varchar(255)
mentee_name	varchar(255)
thoughts	varchar(255)

topic	
id	bigint
title	varchar(255)

topic_sub_title	
topic_id	bigint
sub_title	varchar(255)

mentor_applications_sub_topic	
mentor_applications_id	bigint
sub_topic	varchar(255)

phase	
id	bigint
assessment_of_mentee	varchar(255)
assessment_of_mentor	varchar(255)
end_date	varchar(255)
end_time	varchar(255)
is_complete	boolean
is_mentee_finish	boolean
is_mentor_finish	boolean
name	varchar(255)
phase_id	integer
rating_of_mentee	integer
rating_of_mentor	integer
status	varchar(255)
mentorship_id	bigint

mentee	
id	bigint
is_mentor	boolean
main_topic	varchar(255)
name	varchar(255)
sub_topic	varchar(255)
thoughts	varchar(255)

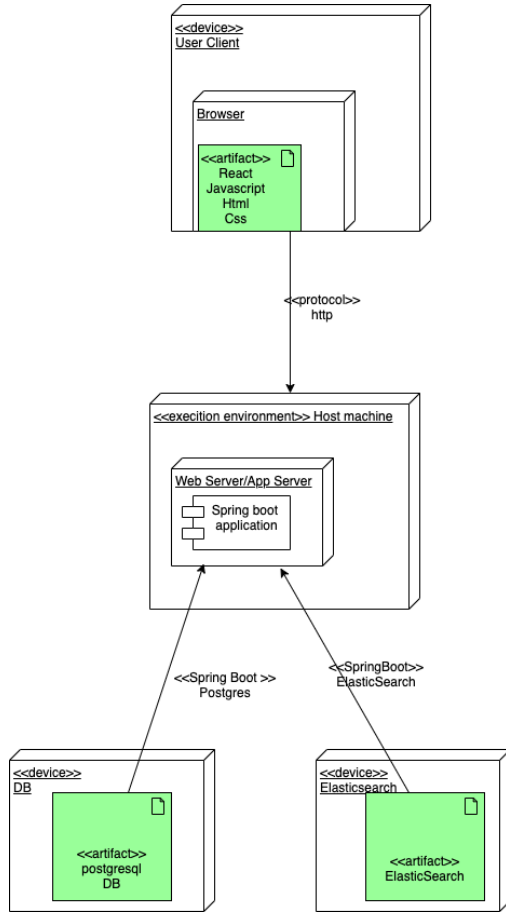
mentorship_id:id

mentor_applications_id:id

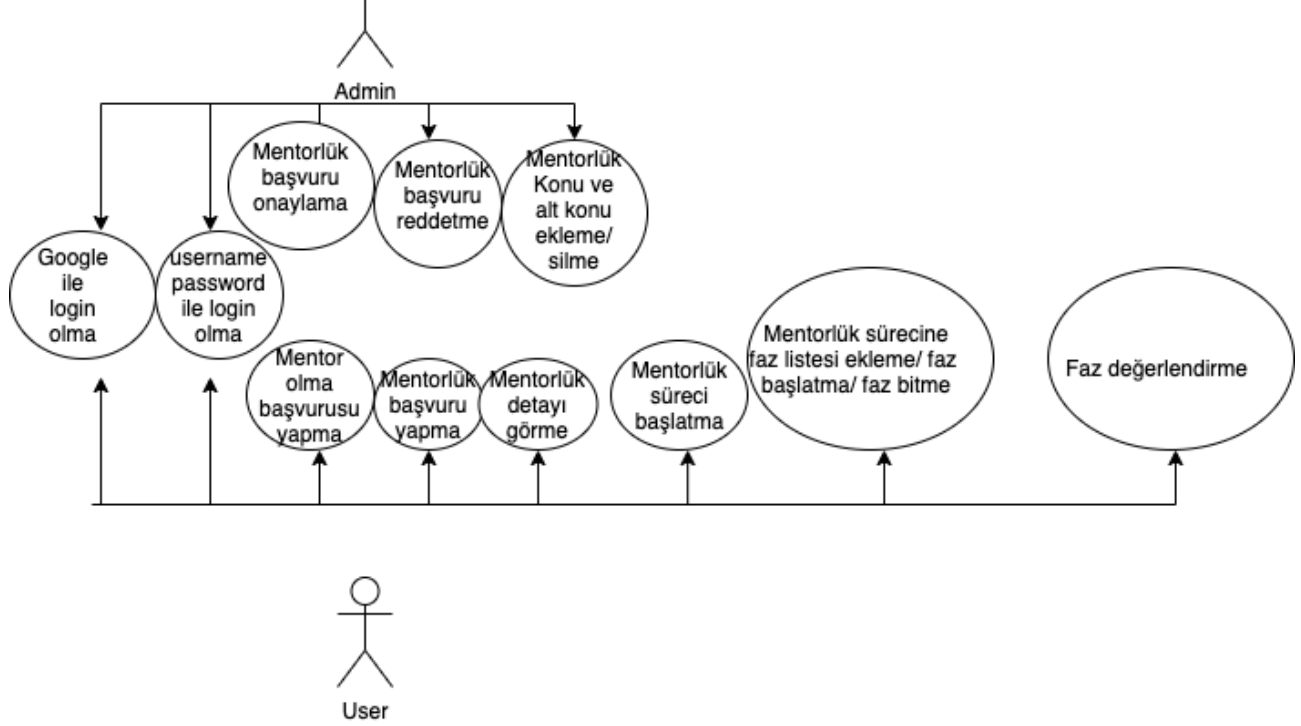
topic_id:id

phase_id:id

2- Fiziksel Deployment Diagramı



3- Use Case Diagramı



4-Class Diagramı

