

Activity Tracking App Proje Raporu

Entity :

- int - Id ; (PK)
- DateTime - createDate ;
- DateTime - updatedAt (null) ;

EventType : Entity

- string Name; // Okuma, Dinleme ve İzleme

Events : Entity

- string Name; // Kitap, Dergi, Film, Belgesel, Podcast, Müzik
- int EventTypeId (FK)

EventTopic : Entity

- string Name; // Bilim kurgu, Aksiyon, Drama, Romantik

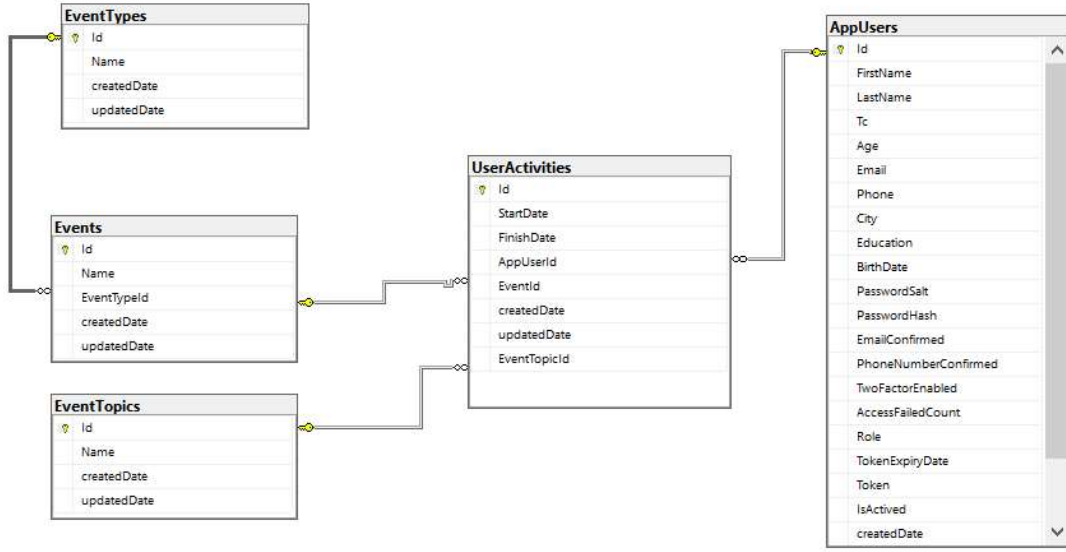
UserActivities : Entity

- DateTime StartDate
- DateTime FinishDate
- int AppUserId (FK)
- int EventId (FK)
- int EventTopicId (FK)

AppUser : Entity

- string FirstName
- string LastName
- string Tc
- int Age
- string Gender{ get; set; }
- string Email
- string Phone
- string City
- string Education
- DateTime BirthDate
- byte[] PasswordSalt
- byte[] PasswordHash
- bool EmailConfirmed
- bool PhoneNumberConfirmed
- bool TwoFactorEnabled
- int AccessFailedCount
- string Role
- DateTime TokenExpiryDate
- string? Token
- bool IsActivated

Veritabanı Tasarım Diagramı :



Uygulamanın İşleyişi ve Çalışma şekli :

Activity			^
POST	/api/Activity/StartActivity	✓	🔒
POST	/api/Activity/FinishActivity	✓	🔒
Event			^
POST	/api/Event/AddEvent	✓	🔒
GET	/api/Event/GetEventList	✓	🔒
EventTopic			^
POST	/api/EventTopic/AddEventTopic	✓	🔒
EventType			^
POST	/api/EventType	✓	🔒
Report			^
GET	/api/Report/GetMostFrequentlyActivityForWomen	✓	🔒
GET	/api/Report/GetBestActivityandTotalTime	✓	🔒
GET	/api/Report/GetWorstActivityandTotalTime	✓	🔒
UserAccount			^
POST	/api/UserAccount/CreateUser	✓	🔒
POST	/api/UserAccount/CreateToken	✓	🔒

EventTopic ekleyerek sisteme (Korku, Komedi , Drama) gibi event topic'ler tanımlanabilmektedir. Daha sonra bir Aktivite başlatırken buradaki EventTopicId bizden istenecektir.

Event Type aktivitenin tipini belirtir. Örneğin izleme, dinleme , okuma gibi değerler olabilir. Ardından Event eklerken buradaki eventTypeld bizlerden istenecektir. Event'ler ise Kitap, dergi, müzik, podcast gibi değerler alabilir. Ayrıca EventController altında eventListesi getirilebilir.

En üstteki aktivite methodlarını çalıştırmak için sisteme kayıtlı bir kullanıcı olmalıyız. Aktivite metodlarından startActivity methodu aracılığıyla girdiğimiz bilgilere göre bir aktiviteye başlarız. Ardından bize döndürülen nesnedeki Id değerini alarak FinishActivity methoduna parametre olarak gönderirsek bu aktiviteyi sonlandırmış oluruz ve sonlardırdığımız esnada finishDate alanı güncellenir.

Eğer kaydımız yoksa User Account altındaki CreateUser Methodu aracılığıyla kayıt işlemini gerçekleştirebiliriz. Kayıt işlemini başarılı bir şekilde gerçekleştirdikten sonra CreateToken methoduna giderek kayıt olduğumuz mail adresi ve şifre bilgilerini doğru şekilde girersek bir token değeri üretiriz. Ardından bu token değerini kopyalayarak swagger üzerinde yer alan Authorize alanına giderek "Bearer {token}" şeklinde vererek sisteme authorize olabiliriz.

Available authorizations x

Bearer (apiKey)

Enter JWT Bearer authorization token

Name: Authorization

In: header

Value:

Bearer eyJhbGciOiJIUzI1NiIs

Authorize

Close

Sistemde yer alan Raporlar için sisteme giriş yapmış olmaya gerek yoktur.

- a. Kadınların en çok zaman ayırdığı aktivite türü ve ayrılan toplam süre
- b. En sevilen aktivite ve ayrılan toplam süre
- c. En seilmeyen aktivite ve ayrılan toplam süre

Get isteği ile gibi yukarıdaki raporları elde edebiliriz.

Kullanılan Kütüphaneler ve Teknolojiler:

- .NET 6 MVC Web API
- N Layer Architecture
- Fluent Validation
- AutoMapper
- AutoFac
- EntityFrameworkCore for Sql Server
- Json Web Token