ClinicApi

Task Summary

1	☐	Introduction
2	☐	Main entities
3	☐	entity: Booking
4	☐	entity: Document
5	☐	entity: Disease
6	☐	entity: User
7	☐	entity: Clinic
8	☐	entity: ClinicClinican
9	☐	entity: Notification
10	☐	entity: RefreshToken
11	☐	User roles and functionalities
12	☐	Authorization
13	☐	Technologies
14	☐	Hosting

1 □ Introduction

Main goal

This app will help people to easier find appropriate clinic and clincian and made booking in convenient way.

The ***main idea*** is to make global application (for now only web, in future mobile app for android and iPhone) for all clinic. Somethis simillar to 'Local' in Lviv. So in this app clinician can see all registered clinic and decide where it is convenient to make booking in some clinician.

Why is it valuable?

For patients it`s give convenient possibility to browr through closest clinic and its clinician, view rate for clinician from other patient and view comments.

For clinics it`s good opportunity to promote their service and compare it`s with other nearest clinic. Also it give good way to find the most skilled clinicians and hire them.

For clinician it give possibility to promote thier skills and work in several clinic in convenient way.

Also there are possibility to save discoverages from some booking (with analysis, results and diagnoses). Clinician can save result of his completed booking but only if patient approve it.

It will help young doctors at the beginning of his careare and increase productivity of all clinician.

Each patient can view results of his booking and also other if Clinic administrator allow it for him.

Clinic administrator can view all booking created at his clinic.

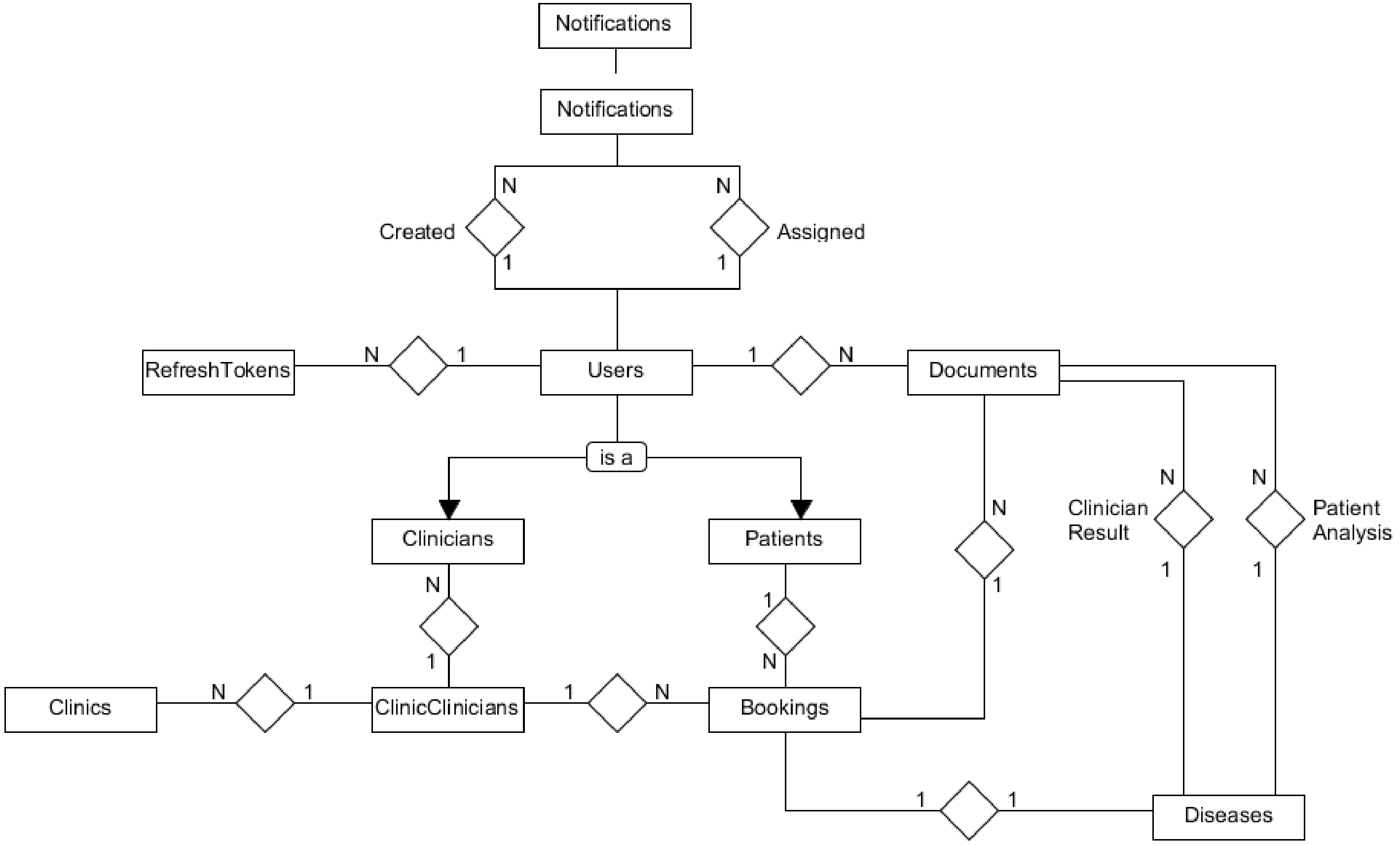
Also this infomation can be used to review work of some clinician and increase his sallary. It will help to find the most experianced clinician.

Necessary resources:

- GitHub Repository: https://github.com/VolodymyrKLymenko/clinicApi (https://github.com/VolodymyrKLymenko/clinicApi)
- Firebase app: https://console.firebase.google.com/project/webapp-1d45e/hosting/main (https://console.firebase.google.com/project/webapp-1d45e/hosting/main)
- Trello link: https://trello.com/b/QhF4eYDP/clinicapi (https://trello.com/b/QhF4eYDP/clinicapi)
- Travis CI: https://travis-ci.org/VolodymyrKLymenko/clinicApi/ (https://travis-ci.org/VolodymyrKLymenko/clinicApi/)
- Documentaiton: https://www.process.st/checklist/clinicapi/
- Azure -> in Future (work on subscription)

2 □ Main entities

ER Diagram:

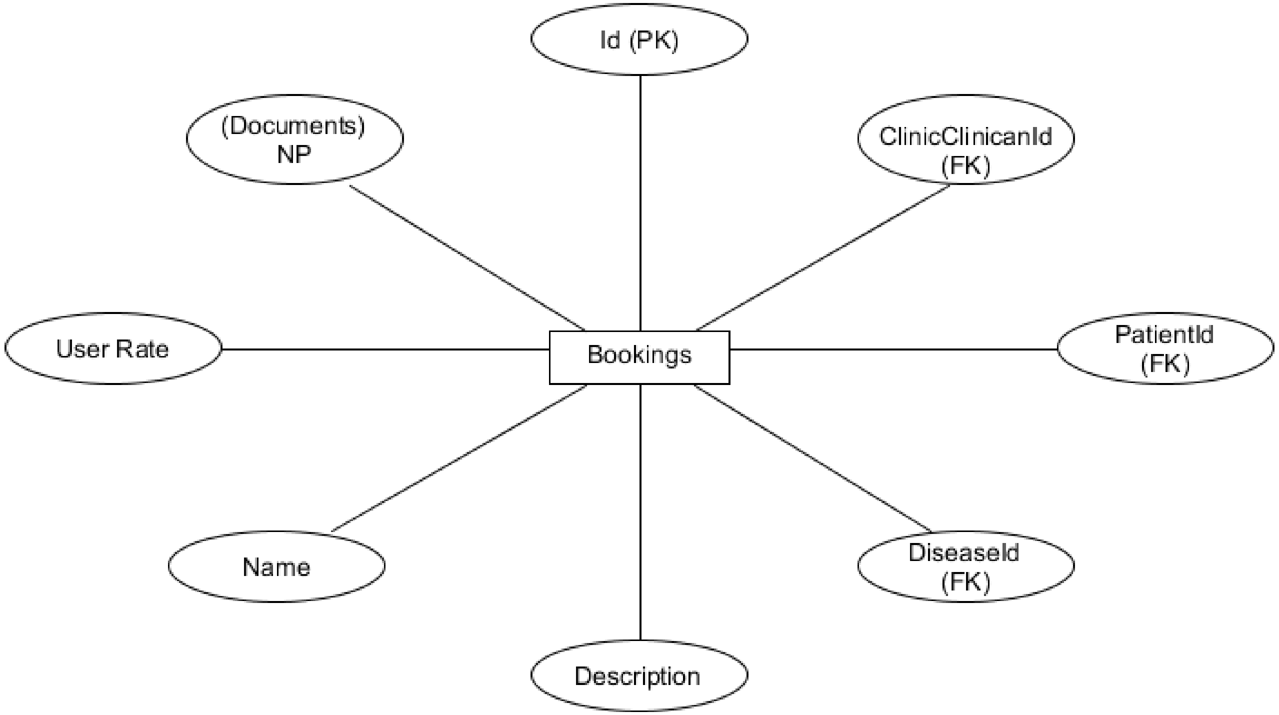


UXF

Download

ERDiagram.uxf

3 □ entity: Booking



ID: Unique, Primary Key

ClinicClinicianId: Foreign key to ClinicCLinician table

PatientId: Foreign Key to Patient table

DiseaseId: Foreign Key to Diseases table

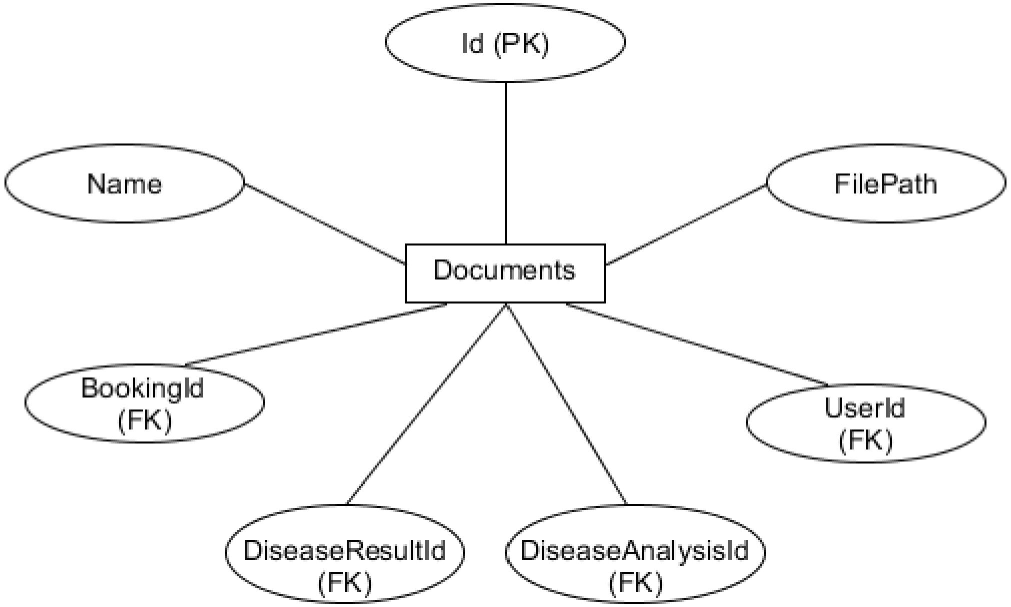
Descripton: Short description of treat; Not long, detailed discoverages will be saved in file (Document table)

Name(optional): Clinician can named his bookings

User rate(optional): Patient can rate work of clinician. Clinician can`t view this rate (only avarage calculated from all his booking rates)

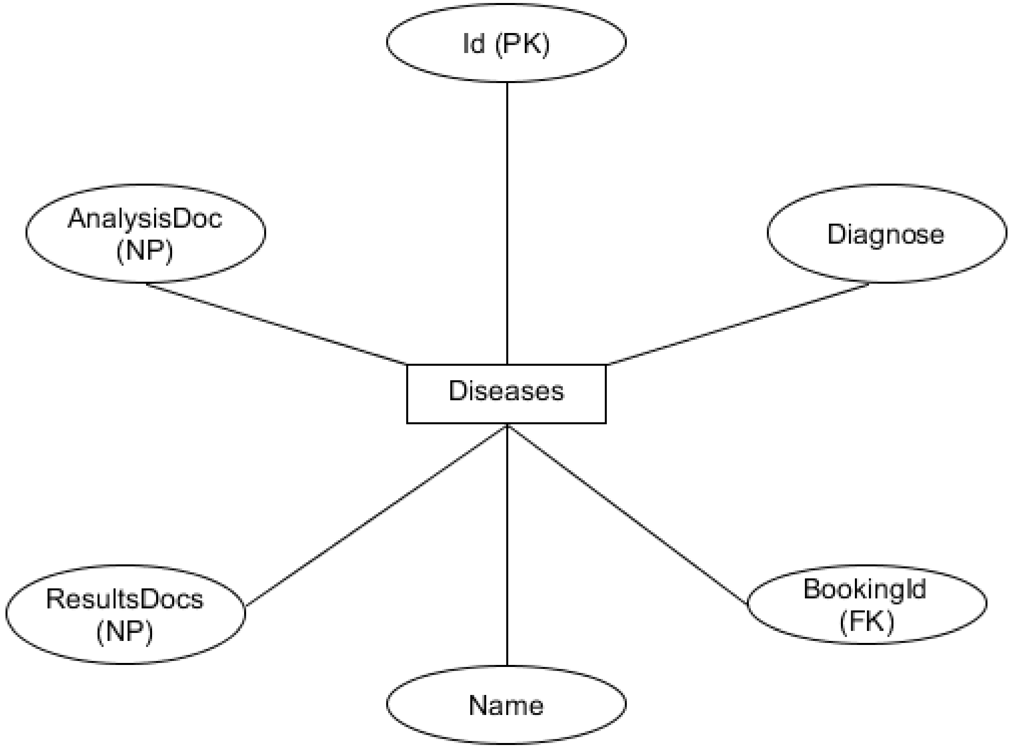
Documents(navigation property): each booking contain documents(files) with analysis and some discoverages

4 □ entity: Document



Documents will be stored at host file system. (Maybe will use some auzre storage for if in future)

5 □ entity: Disease



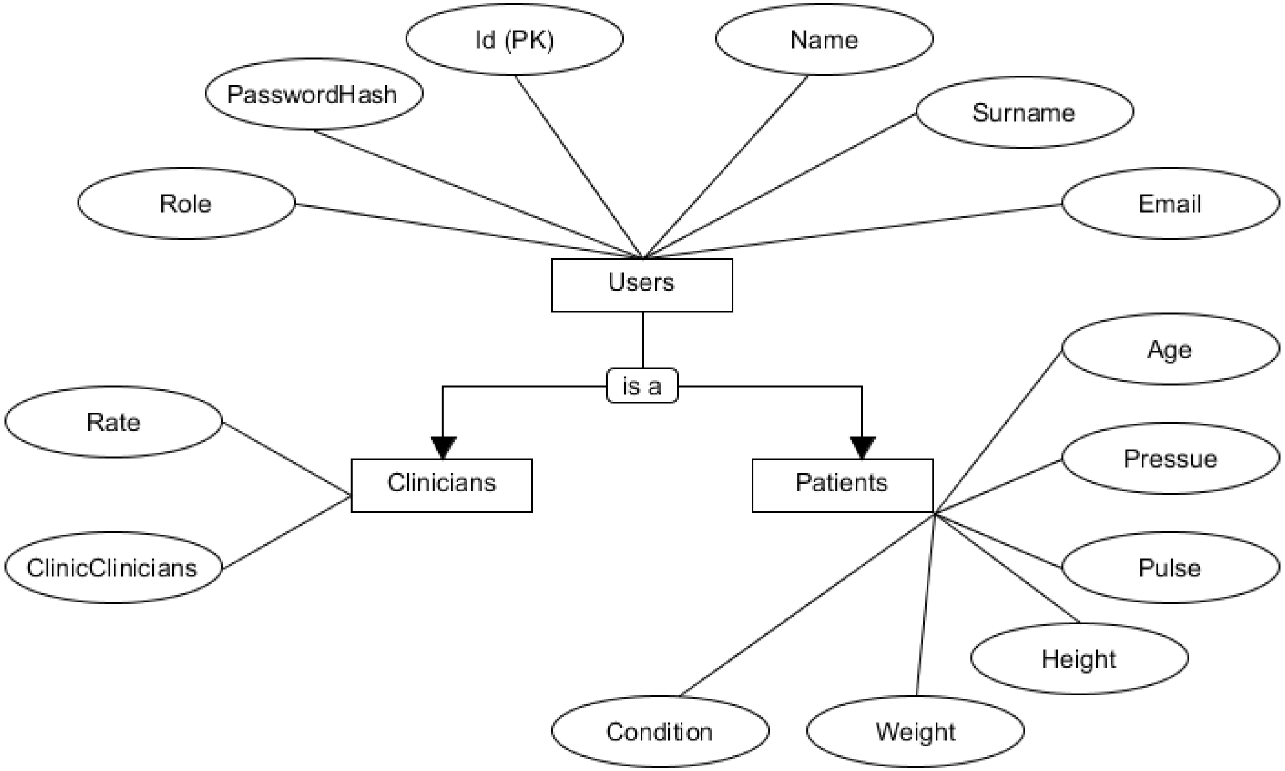
ResultsDocs: It`s navigation property for all documents created by clinician about diagnos and some investigation.

AnalysisDocs: It`s navigation property for all documents with analysis for patient, can be uploaded by patient or clinician.

Name: It`s optional parametr which can be specified by clinician like key phrase for booking.

Diagnose: For now it`s just string property.

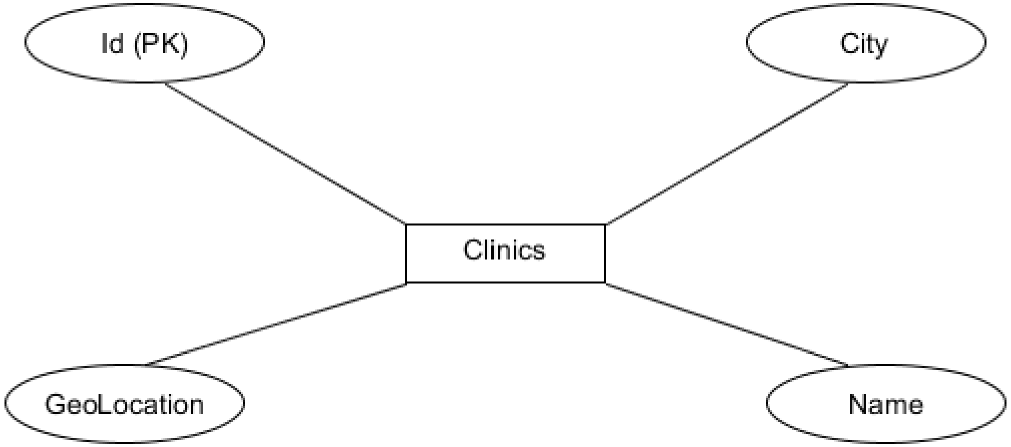
6 □ entity: User



Will add another useful health characteristic to Patient entity

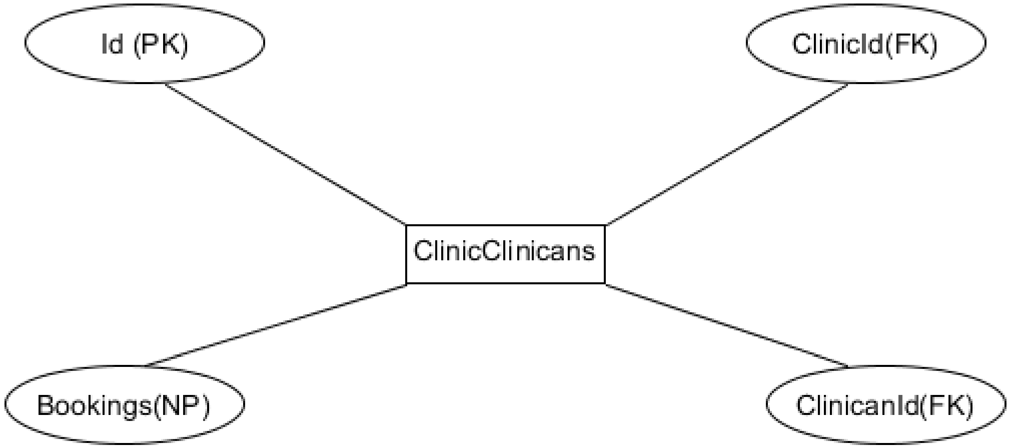
Patient table will contain all common information about health condition, which will be used to make some statistic and will be displayed like diagrams on UI part at user tab.

7 □ **entity: Clinic**



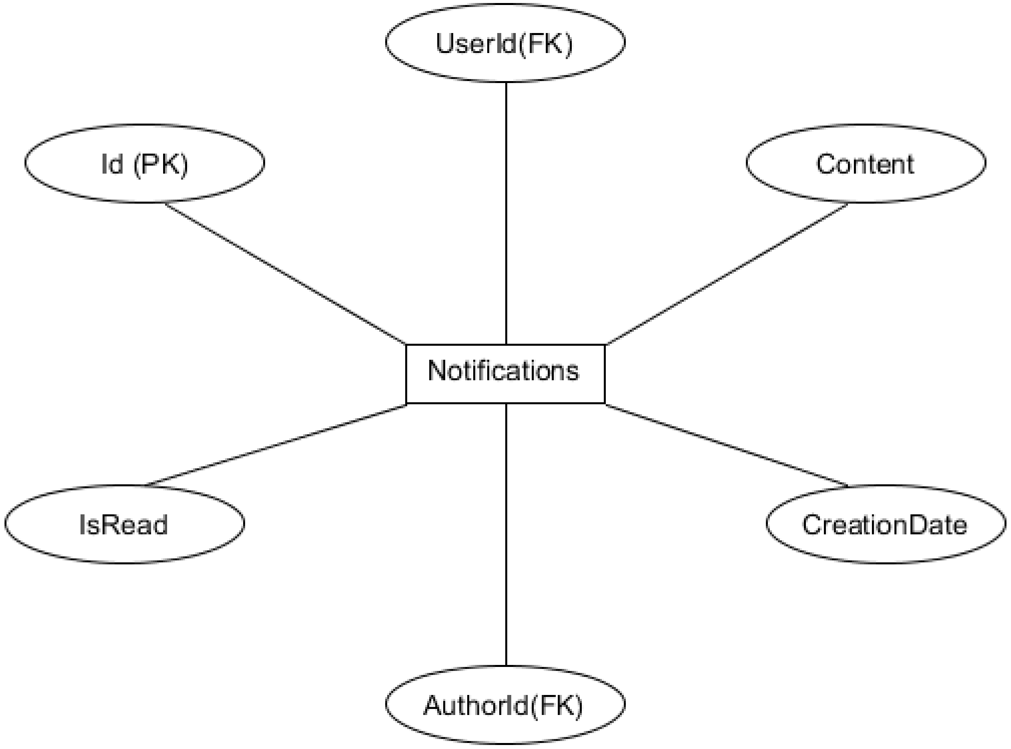
Geolocation: Contain information about longitude and latitude of clinic. Special type supported by MSSQLServer DB and EF

8 □ **entity: ClinicClinican**



NP -> Navigation properties (absent in db)

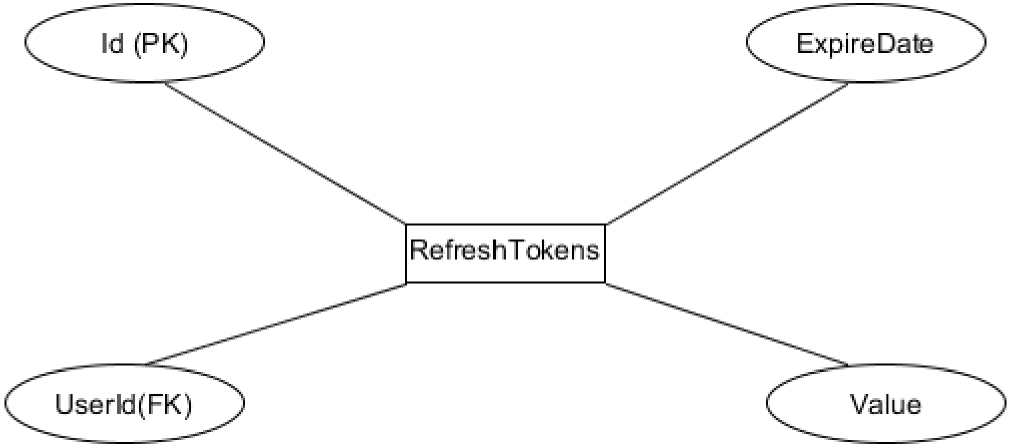
9 □ entity: Notification



In Future will store it in quicker storage like: firebase db or azure cosmos db.

Content: It`s string property with message of event.

10 □ entity: RefreshToken



Should be removed on:

- User try to access with expired time
- User logout
- On refresh token create new and remove existing

11 □ User roles and functionalities

There are three main roles:

- Patient (user)
- Clinician (doctor)
- Clinic (administrator)

Patient:

1. Create booking but (clinic admin or clinician can reject it with notification for patient)
2. View all clinic in two way
    a. Select clinic -> clinician from drop down select field (sorted by distance from user)
    b. Special map tab with closest clinic and then select clinician

3. View user information with
    a. Created booking (and their state)
    b. Personal information
    c. Statistic with (graphics for pulse, pressure, weight etc)
4. Can rate some booking but not clinician at all (clinician rate will be calculated based on all rates for his booking)

Clinician:

1. View assigned booking and edit it
2. View/Edit personal information/assigned booking/his rate/saved diseases
3. Upload documents with some analysis and discoverages
4. Can assign to some clinic for specific time duration (but need confirmation from clinic admin)
5. Can confirm/reject patient booking (but for reject need aprove from clinic admin)
6. Add information about booking to special diseases storage

Clinic (admin):

1. View/Edit information aboutclinic
2. View information about booking and reject some (but needed aprove from clinician and notification will be created)
3. View booking filtered by
    a. Clinician
    b. Creation date
    c. Duration
4. Fire clinician with notification

12 □ Authorization

OAuth 2.0 standard protocol is used for authentication/authorization with access and refresh tokens.

Access token is JWT token generated on authorization and contain:

- Sequrity information to encrypt it;
- Expire time = 30 minutes;
- Claims information about
    - user id
    - user role
    - user email

Refresh token is stored in database and is send to user on authorization. Is created with standart encryption classes from ***System.Security.Cryptography*** namespace in .NET Framework.

It is special entity which contain

- User id
- Expiration date (creation date + 7 days)
- Encyped values for security

On expire access token user have to call specific api route and send both expired access token and refresh token:

- API retrive information about user from access token
- Then take needed refresh token entity from database for this user
- Check if secured values form entity and send values is same
- If success generate new access and refresh tokens
- Remove old refresh token entity

All password is encryped with random generated stirng value(hash) to make haker`s life more fun.

13 □ Technologies

The application consist of two parts BackEnd (.Net Web Api 2) and UI (Angular 7)

Back end:

- .Net Framework 4.7 (will change to .net core in future)
- Web API 2 platform
- AutoFac for Dependency Injection
- AutoMapper for mapping models with entities (possibly chnge to simple extensions methods in future)
- BCrypt.Net (It`s special package to encrypt password with specific random salt. Also it allow to set time or memory complexity for decryption of passwordHash that made haker`s life much more difficult)
- EntityFramework (6.2.0) as ORM
- Microsoft.SqlServer.Types (special package which allow to store in DB special types for geolocation(DbGeography) and math(DbGeometry). Which simplify wotk with geolocation data. It work with special type Spatial in MSSqlServer

UI Part:

- Angular 7
- Bootstrap 4 (for styling)
- Prime NG (big library with some common components)
- Agm/core (to integrate app with google maps)

14 □ Hosting

Full application is sored at GitHub (https://github.com/VolodymyrKLymenko/clinicApi (https://github.com/VolodymyrKLymenko/clinicApi))

UI

- Angular application is deployed on firebase host
- Firebase databse to fast access for notification (in future)
- Also use ***travis-ci*** to configure continues integration with github (automatically build and deploy angular application on push into master branch)

Back-end (not yet deployed)

- Will be deployed on Azure Services