**Draft of the Idea and Basic Information**

1. Name of the application   
     
   1.1 E-Gov   
   1.2 Health System

1.3 Hospital

1.4

1. Models – Id is Guid !? nor BaseModel that creates the Ids

* Hospital – Id, Name , Address, List<Doctor> , List<Patient>, List<Room> , List<Department>, Type??
* Clinic – Id, Name , Address, List<Doctor> , List<Patient>, List<Room>
* Emergency – Id, Hospital hospital , List<Doctor>, List<Patient>, List<Nurse>, List<Driver>
* Department – Id, Name (Oncology, Pediatric, Surgery, Traumatology, etc), List<Manager who is a person >
* Pharmacy – Id, Name, Address one to one
* Doctor – Id , Name, List<Specialty> , List <Patient>, List<Hospital>
* Nurse – Id, Name,
* Driver – Id , Name, List<Category>, List<Vehicle>, scan of driving license
* Vehicle – Id, Make, Model (selection ) type, carplate , capacity , fueltype(petrol, diesel , hybrid, electric), distance ++
* Category – Id, Letter … ??
* Address(can be null? Or default No Address)
* City – Id, Name, Country one to one
* Country – Id, Name, List<City>, Country Code (2CapitalLetters)
* Medicine – Id, Name, text with description(text), Type (??? Homeopathy, no prescription, etc )
* Prescription – Id, List<Medicine>
* Room – Id, Capacity, Count of beds = Capacity (int), List<Bed> If not full… List<Patient> (see how we make the rooms full )
* Bed – Id, Room room – one-to-one
* Patient – Id, Name, List<Diagnose>, List<IEntityHealth??>
* Diagnose – List<Patient>
* Class with messages , constants
* Class with Exceptions – static class
* GitHub Clean Architecture details about distribution
* Guid class to create new Id for all classes

Controllers – for each page(Index, Login, Users, etc) we need a class name+ Controller

Views has the html stuff

([CallerMemberName] string viewname = null)!!!!!

Wwwroot has the css files

HttpClient and FileStream classes have async and await methods

View -> Tweets -> Create.cshtml displays when the Controller -> TweetController says return this.View(); …

ViewModel is in Models Folder , not in View Folder/Project

* 1. Interfaces   
     - IEntity (Hospital, Pharmacy, Clinic, School, Police station, )

- IMinistry (Health, Defense, MOI, MOE, Education, Justice, Environment, Finance/Tax/ Social security)

- IPerson(Id, Name, Age, ID )

-IPosition (Patient, Doctor, Administration(Secretary, Purchase/ Procurement), IT, Janitor, Guard, Nurse, Janitor, Driver, HR, etc )

- IAddress (City, Country, street, etc or text )

-IVehicle – Car, Bus, Truck, etc

-IDepartment – all kind of institutions

2.2 Type of access

- full access – CRUD - approve registrations

- view access - request registration , request orders, book visitation

- authentication form – log in page

- sha256 pass hash

-   
  
  
  
  
  
  
3. Database

- DTO with **automapper** that reads from the Database and sends to the Controller

Or View that shows the Index/ Initial page

Or service.ForumPosts<DTO> Service gets from the Controller (via automapper)????

4.Functionality

- pop up buttons

- clock, date

- remaining time

- reporting ???? how to do and how to download and display it , counter for each class (patients served , etc ). Download reports via HTTP -> HTML lection page38

- statistics – count site visits , show date and time ? Redis

- people pictures , pictures at all

- help windows

- accept cookies

- greeting (Good Morning, Good Evening, Good Afternoon)

- request access

- commercials (robot.txt, ads.txt)

- report problem – csv format , return excel

- exceptions

- where to host, how to upload ?

- login with FB, Google , etc

- pass verification

- maps where address ?

- error message / under construction

- upload pictures

- automated mails sent to users

- external systems integration

- HTTPS – security

- address of the application / site

- automapper!!

- feedback form

- Buttons send HTML requests

- HttpUtility.UrlDecode() – decodes the string

- bootstrap goes to wwroot folder!

- include somehow machine learning – 18.07.2019 Niki Kostov’s video on this

- isDeleted

- GitHub, NikiIt , take the template , as it is done to be used

- Services are the ones communicating with the models that load the DB content

- Controller communicates with the Service

- upload videos(link maybe?!) , pictures, etc

- search field on the whole site ?

- sendGrid to send mails for registration

-layout to have ifs to change the theme / css when logged and not ...

-every page has a Controller (domain/controller/method(action)). Its methods are the buttons

-every Controller class is related to a folder in Views

- Views folder - page appearance, related to layout

- Home in View - Home page appearance - Index and Privacy for example in extension cshtml, where HTML is . "@" sign starts the Csharp code here

5.Legal stuff – GDPR, link to law and legislation, privacy, trade marks, etc

6.Testing of the solution – Moq, Users , etc