**АРХИТЕКТУРЕН ПРОЕКТ**

**НА ЗДРАВЕН РАКОВ РЕГИСТЪР**

Автори на проекта:

Кристина Николаева Герчева, фак.№ 471218029

Андрей Ангелов Стрински, фак.№471218010

Любомир Драгомиров Филипов, фак.№471218015

Виктория Емилова Георгиева, фак.№471218040

Дата: 27.10.2020г.

СЪДЪРЖАНИЕ

1. Въведение
   1. Представяне на документа
   2. За проекта
   3. Участници в проекта
2. Предназначение на проекта
   1. Обхват на проекта
   2. Актьори
   3. Използвани термини и символи
   4. Източници
3. Архитектурен обзор

3.1. Use-Case изглед

3.2. Логически изглед

3.3. Процесен изглед

3.4. Изглед на внедряването

1. Нефункционални изисквания

## Въведение

* 1. **. Представяне на документа**

Целта на настоящия документ е да опише архитектурния план на проекта – Здравен Раков Регистър. Описаната структурата в документа е съвкупност от елементите на софтуера, свързани помежду си с връзки или взаимотношения. За целта те са представени чрез клас диаграми, комуникационни диаграми, диаграми на внедряването и други.

**1.2. За проекта**

В момента България няма национален антираков план. Последните данни в раковия регистър са от 2015г., и те не дават достатъчно информация за стадиите на болестта , а оттам нататък и за всички останали елементи от системата и лечението на пациентите. Ракът на гърдата е сред най-честите туморни образувания според Световната здравна организация.

Целта на проекта е да се отвори раков регистър, така да се даде начало на антираковия план, който трябва да обхваща всеки аспект на превенцията и лечението на онкологичното заболяване. По този начин може да се обедини цялата тази информация в големи бази данни, обхващащи демографските аспекти, медицинската история, и ефекта на съществуващите лекарства, с цел след анализ да се определи кои терапии са били най-добри за отделните пациенти.

**1.3. Участници в проекта**

Учатниците в проекта са както следва:

Кристина Николаева Герчева, фак.№ 471218029

Андрей Ангелов Стрински, фак.№471218010

Любомир Драгомиров Филипов, фак.№471218015

Виктория Емилова Георгиева, фак.№471218040

Посочените участници в проекта са отговорни за успешната реализация на Системата.

## Предназначение на проекта

## Обхват на проекта

Софтуерът ще бъде изграден в сферата на здравеопазването. Основните дейности, които осъществяват целите на проекта и формират обхвата на проекта са:

* Създаване и поддържане на ЗРР
* Възможност за запазване на час за медицински преглед от страна на пациентите;
* Извършването на изследвания и започване на определено лечение (дейности, изпълнявани от лекари) на базата на техните симптоми;
* Получаване на информация в периода на лечението на пациента, за да може най-рано да се премине на друг вид лечение;

Тези дейности се изпълняват с цел подобряване на състоянието и излекуване на пациентите.

Нужен е по-добър начин за съхраняване, обработка и следене на данните свързани с лечението на хора болни от рак на гърдата. Необходимост от подобряване на организацията на разпределение на докторите относно медицинските прегледи с възможност на отдалечено показване на резултати от прегледа. Софтуерът ще улесни дейностите на докторите и наложи внимание повече върху изследването на данните получени от пациентите. Целевите групи, за които ще бъде предназначен разработвания софтуер, включват: лекари-онколози, пациенти болни от рак на гърдата.

### Актьори

Участниците, заинтересовани от документа са следните: системен инженер, програмист и системен администратор. Основната част на документа, която касае страните е Архитектурния обзор на системата, където е представена архитектурата и каква информация носи всеки един от представените изгледи. ([виж тук](#_Архитектурен_обзор)). Частите от документа, които касаят трите страни са както следва:

* Системен инженер – [Процесен изглед](#Процес), [Изглед на внедряването](#Внедряване);
* Програмист/Разработчик – [Use-case изглед](#UseCase), [Логически изглед](#Логика), [Процесен изглед](#Логика);
* Системен администратор – [Activity диаграма](#Activity), [Use-case изглед](#UseCase).

### Използвани термини и символи

**ЗРР - Здравен раков регистър**

**ЗАПСП - Закона за авторското право и сродните му права**

**Изходен код (Source Code) - Изходният код е сбор от инструкции, написан на разбираем за човека език за програмиране обикновено като текст.**

**Load Balancer - Означава разпределител на натоварването. Балансиране на натоварването се налага, когато в развитието на даден уеб сайт се достига до момент, в който един сървър не е достатъчен за натоварването, което генерира самият сайт.**

**Firewall - Защитната стена, срещано и като файъруол в компютърните мрежи е специализиран хардуер или софтуер, който проверява мрежовия трафик, преминаващ през него и разрешава или забранява достъпа по определени правила.**

**ESMO - Европейското дружество за медицинска онкология**

**2.4. Източници**

Използваната литература в настоящия документ е от следните източници:

Европейското дружество по медицинска онкология - <https://www.esmo.org/>;

Уикипедия - <https://bg.wikipedia.org/wiki/>;

Рамка за техническо задание -  <https://www.mtitc.government.bg/sites/default/files/pril-nar-obsi-iziskv-info-sistemi-registri-el-adm-usl-14012020.pdf>;

Онлайн мрежа за учебни материали - <https://www.docsity.com/en/>;

Материали по предмет Софтуерни архитектури, II курс ИСН;

<https://www.uml-diagrams.org/deployment-diagrams-overview.html>;

## Архитектурен обзор

## Архитектурата на приложението ще бъде монолитно приложение, следващо архитектурен шаблон ще бъде MVC. Това е така, тъй като приложението няма нужда от високо ниво на скалируемост, защото не се очаква функционалността му да бъде разширявана екстензивно или да бъде приобщено към други системи след завършване на разработка. MVC моделът обаче ни позволява да имаме разграничимост между потребителски интерфейс и бизнес логика, така че в бъдеще да може да се преправи едното без това да афектира другото. От гледна точка на сигурност, системата ще бъде защитена с Basic Authentication. Проектът ще бъде разделен на три основни части:

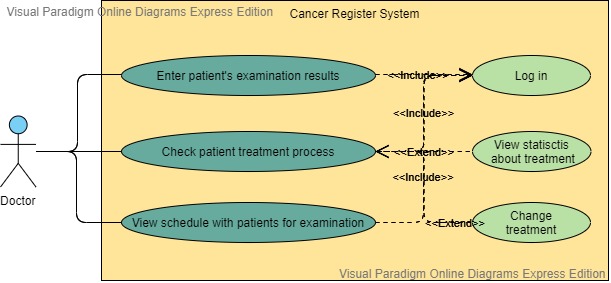
* 1. Клиентски портал: Потребителският интерфейс чрез който ще се достъпва приложението и неговите функционалности.
  2. RESTful API: Компонент в който се разполага цялата бизнес логика. API-то само ще отговаря за собствената си сигурност изисквайки валиден ауторизационен header при всяка защитена заявка.
  3. База данни: Хранилище за данните на приложението в което ще се държи информацията с която ще разполагат потребителите.

Можем да използваме следните 5 архитектурни изгледа, за да опишем софтуерната система:

1. **Use-case изглед** – той изобразява функционалните изисквания спрямо софтуера (основните сценарии), които включват взаимодействието на докторите-онколози, пациентите и администратора със системата. Целта на този изглед е да даде основата, според която ще се разработи софтуерът.
2. **Логически изглед -**  софтуерът се представя като композиция от модули, пакети и класове. Използват се диаграми (*UML class diagrams*), за да се илюстрират връзките между архитектурно зависимите компоненти, в комбинация с текстово описание;
3. **Процесен изглед** – изобразява процесите, извършващи се в системата, които включват регистрация на пациент с рак на гърдата.
4. **Изглед на внедряването** – той изобразява начина на разположението на хардуерните възли спрямо един друг и тяхното взаимодействие, също така се представя и как различните системни файлове и използваните приложения са разпределени по възлите.

**3.1.** **Use-case изглед**

1. *Взаимодействие на доктор със системата*



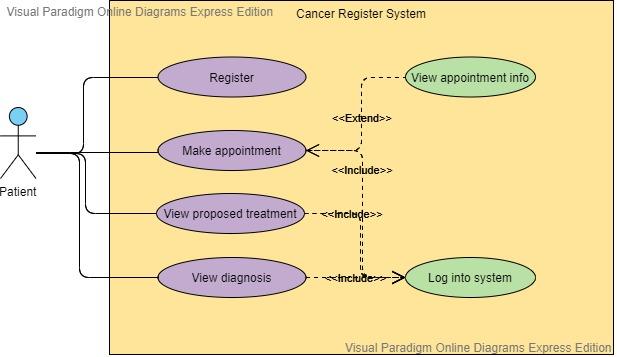
-Докторът-онколог трябва да може да въвежда в системата информация свързана с текущото състояние на пациента след извършване на медицински преглед.

-За всеки пациент докторът-онколог трябва да може му се представи статистика за определен период от време, през което е приложено дадено лечение, за да може да се видят разликите в състоянието на пациента. Това означава сравнение на параметри свързани с тумора: дали големината му намалява, дали възпалението на регионалните лимфни възли спада. Ако състоянието на пациента е много усложнено, главната цел на лечението е да потисне разпространяването на тумора, затова се очаква състоянието на пациента да не се влошава.

-Докторът-онколог трябва да може по всяко време да има достъп до това през какво лечение е минал пациента от началото на неговата регистрация и какво е текущото лечение.

-Докторът трябва да разполага с график с пациенти за преглед(изследване) за деня, предоставен от софтуерът.

1. *Взаимодействие на пациент със системата*



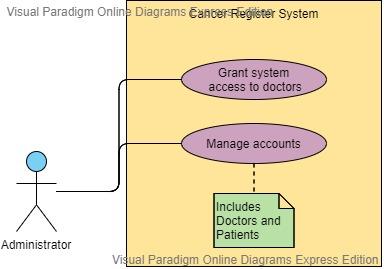
-Пациентът трябва да може да се записва в системата с необходимата за него лична информация.

-В системата той трябва да може да си избере час за преглед(изследване), избирайки свободен в това време доктор.

-След извършване на преглед и регистрация на данните от страна на доктора, пациентът автоматично да получава избраното от системата лечение.

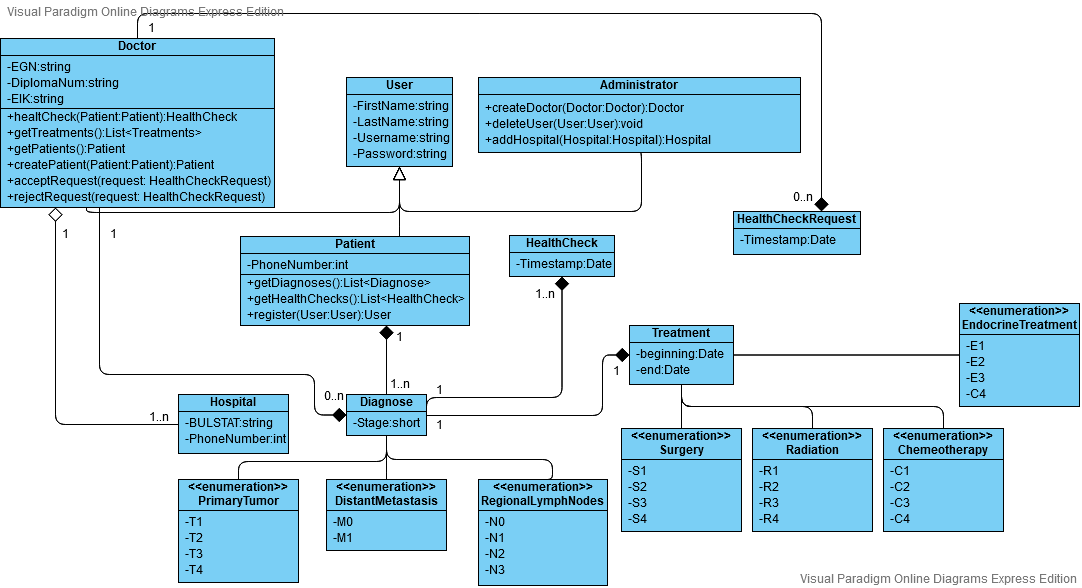
-Системата да показва текущата диагноза на пациента.

1. *Взаимодействие на администратор със системата*

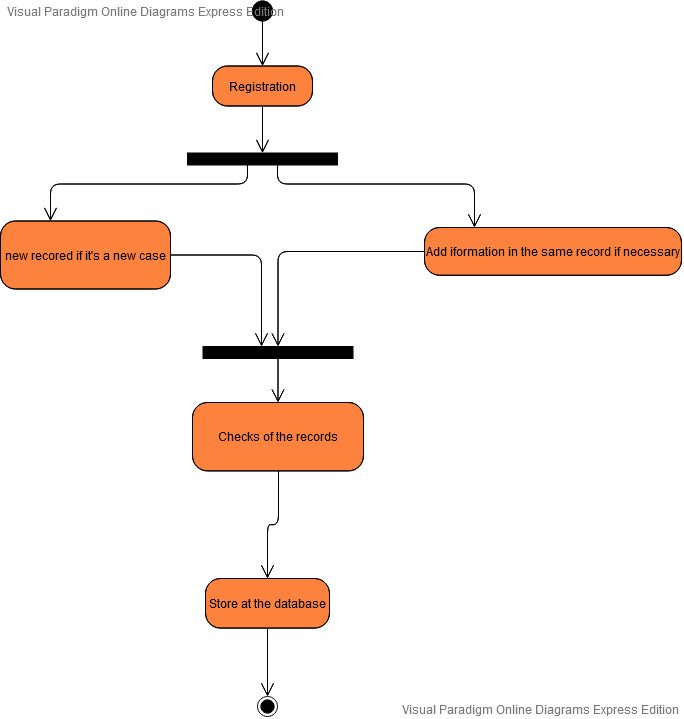


-За управлението на акаунтите на докторите-онколози и пациенти се изисква нова роля – администратор.

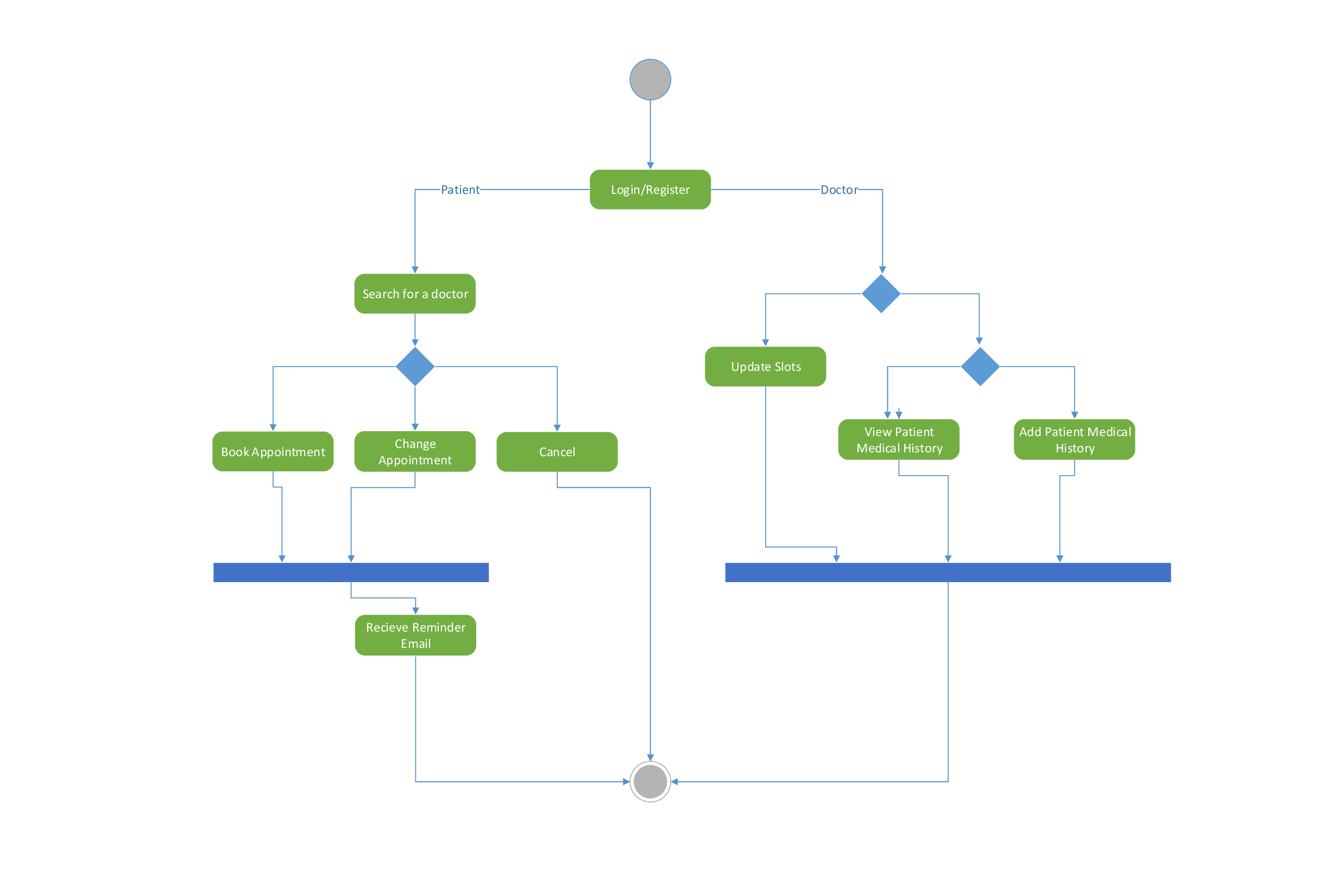
-Администраторът е единственият, който ще дава достъп на докторите до системата (той ще извършва тяхната регистрация).

* 1. **Логически изглед**

На диаграмата са показани всички основни класове в системата. При създаване на системата се генерират потребители от клас Administrator. При начало на практиката на нов лекар, тези потребители създават потребител от клас Doctor, който репрезентира този лекар. Той има за задача да поддържа информацията за себе си (EGN, номер на активната диплома, и в кое лечебно заведение практикува). Лечебните заведения също като лекарите се добавят само и единствено от администратор. Потребителят може сам да си създаде профил, а в противен случай, когато нов пациент се изследва, то лекарят при когото се изследва е длъжен да направи нов запис от клас Patient. За потребителско име на този клас се приема ЕГН-то на лицето и му се поставя автоматично генерирана парола. Когато пациента иска да запази час той може да направи заявка за преглед репрезентирана от класа HealthCheckRequest, която лекарят за когото е заявката може да приеме или откаже. При преглед на пациент се създава обект от клас HealthCheck в който се държи дата на прегледа и диагноза поставена през него. Диагнозата се репрезентира от класа Diagnose. Една и съща диагноза може да продължи да бъде валидна дори и за повече от един прегледа. Всяка диагноза предполага предприето лечение което се репрезентира от класа Treatment. Пациентите имат достъп до всички свои прегледи и диагнози настоящи и предишни. Лекарите имат достъп до пълна информация за всички изминали прегледи и диагнози и лечения.

**3.3. Процесен изглед**

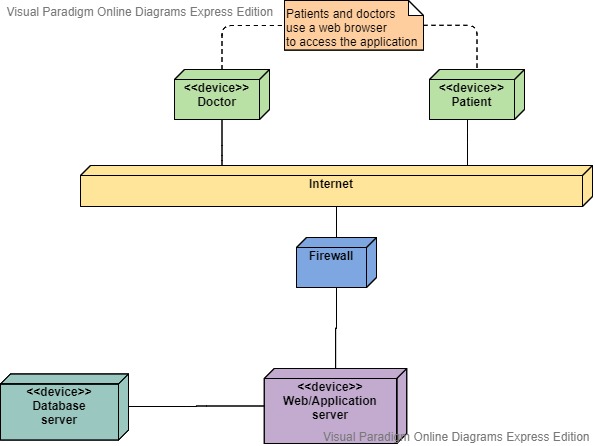
- Тази диаграма представя процеса по регистрация на пациенти и доктори, като последните могат да направят нов запис, ако има наличие на нов случай на рак или да допълнят или коригират информацията във вече съществуващия такъв. След, което данните и от двата вида се запазват и съхраняват в базата данни.



* Тази UML Activity диаграма представя процеса по логване , ако вече имаме направена регистрация или направата на регистрация, при отсъствието на такава, действието се извършва от една страна на лекарите , и от друга страна на пациентите. Всеки пациент, може да извърши търсене и да избере конкретен лекар-онколог, след което може да си запази час при него, да смени своя час за преглед, или да го отмени. Като при всяко запазване на час или промяна в него, пациентът получава напомнящ имейл. Лекарите , от своя страна , могат да актуализират слотовете, да преглеждат епикризите на пациентите или да добавят нови такива.

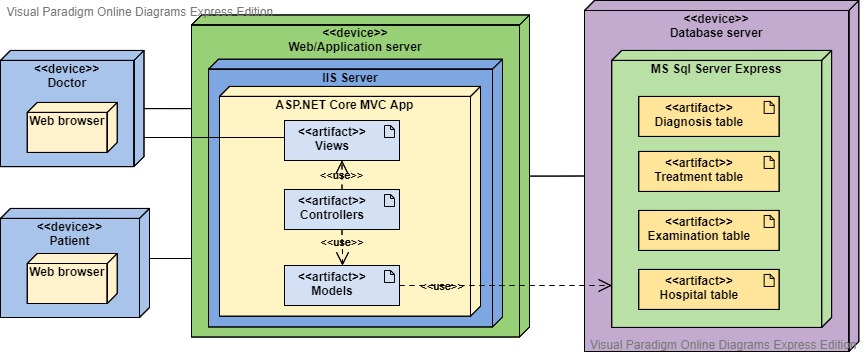
**3.4. Изглед на внедряването**

1. *Мрежова диаграма на внедряването*

****

- От една страна са докторите и пациентите, свързани към интернет, използват уеб браузър, за да достъпят услугите предоставени от софтуера. Главното приложение се изпълнява на друга машина(Web/Application server), като е между нея и връзката и с интернет поставяме защитна стена(Firewall), за да може само оторизирани лица да достъпват системата. Web/Application сървърът е отговорен както за заявки от статичен вид, също и за динамични заявки, използвайки database сървърът. Той е отговорен за съхраняването и натрупването на информацията генерирана от докторите и пациентите.

1. *Диаграма на внедряването описваща разположението на системните файлове и програми*



- Докторите и пациентите, използвайки уеб браузър, изпращат заявки към възела, които има задачата да отговори на заявката, като върне статична страница или динамично да генерира страницата в зависимост от подадените данни, като си взаимодейства с машината, на която е разположена базата данни, използвана от приложението. В центъра на web/application сървърът е разположен ASP.Net Core MVC приложението, хостнато на IIS server. Приложението има достъп до група от файлове, които използва за обработката на заявките, това са Models, Views и Controllers. Данните са разположени на отделна машина. Като за тяхното управление се използва MS SQL Server Express.

**IV. Нефункционални изисквания**

*Системата на ЗРР трябва да отговаря на следните нефункционални изисквания:*

**Натоварване**

ЗРР трябва да позволява едновременна работа на не по-малко от 30 потребители, с възможност за увеличаване на техния брой, при необходимост до 70.

***Производителност***

***-*** Производителността е директно свързана с управлението на ресурсите, с които разполага сървърът, които е отговорен за обработката на потребителската заявка и изпращането на отговор. Така когато трафикът към системата не е голям, ресурсите, които сървърът използва, могат да бъдат занижени. В ситуации, при които трафикът към системата нарасне, ресурсите могат да бъдат увеличени, за да може той да се справи с товара. Ако заявките броят на заявките нарасне значително много, се предвижда внедряване на нова инстанция на приложението на друга машина, като разпределението на товара между двата сървъра ще се осъществи чрез **Load Balancer**.

**Интерфейс**

**-** Да предлага бърза и интуитивна навигация;

**-** Системата трябва да бъде достъпна и да работи чрез основните видове интернет браузъри. Като минимум тя трябва да поддържа последните версии на Mozilla Firefox, Google Chrome, Internet Explorer, Opera и Safari, както и да бъде разработена чрез модерни способи, позволяващи достъпването на ИС и работата с нея през браузър на мобилно устройство;

**-** Дизайнът на системата трябва да позволява бъдещо разширение и подобрения;

**-** Да бъде предоставена възможност за съхранение и движение на документи и файлове в формат -.doc/.docx. и .pdf.

**Сигурност на системата**

**-** Базата данни и системата да включват мерки за сигурност на данните.

**-** Системата не трябва да допуска въвеждането на некоректни данни. Трябва да оповестява потребителя за възникнали грешки и да изисква задължително потвърждаване за извършване на необратими действия.

- Системата да използва надеждни технологии за криптиране на пароли и служебна информация.

**-** Системата да поддържа административен интерфейс за управление на системни данни, номенклатури, потребители и права.

*В допълнение ЗРР трябва да отговаря и на следните изисквания:*

- Всички компютърни програми, които се разработват за реализиране на Системата, трябва да отговарят на критериите и изискванията за софтуер с отворен код;

- Всички авторски и сродни права върху произведения, обект на закрила на Закона за авторското право и сродните му права, включително, но не само, компютърните програми, техният изходен програмен код, структурата и дизайнът на интерфейсите и базите данни, чието разработване е включено в предмета на поръчката, възникват за Възложителя в пълен обем без ограничения в използването, изменението и разпространението им и представляват произведения, създадени по поръчка на Възложителя съгласно чл. 42, ал. 1 от ЗАПСП;

- Изходният код (Source Code), разработван по проекта, както и цялата техническа документация трябва да бъде бъдат публично достъпни онлайн, като софтуер с отворен код.