**Архитектурна документация по проект  
за създаване на раков регистър**

Проектна група 3:

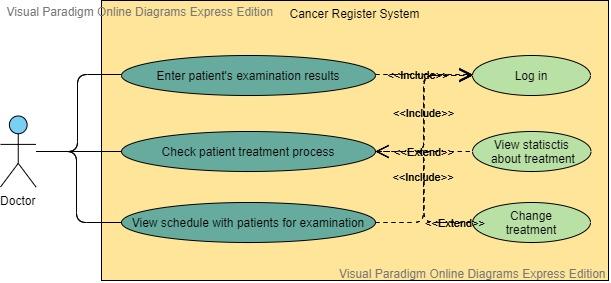
Кристина Николаева Герчева, фак.№ 471218029

Андрей Ангелов Стрински, фак.№471218010

Любомир Драгомиров Филипов, фак.№471218015

Виктория Емилова Георгиева, фак.№471218040

Дата: 26.10.2020г.

1. **Въведение**
2. **Предназначение**
3. **Архитектурен избор**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.
4. **Use-case изглед**
5. Взаимодействие на доктор със системата

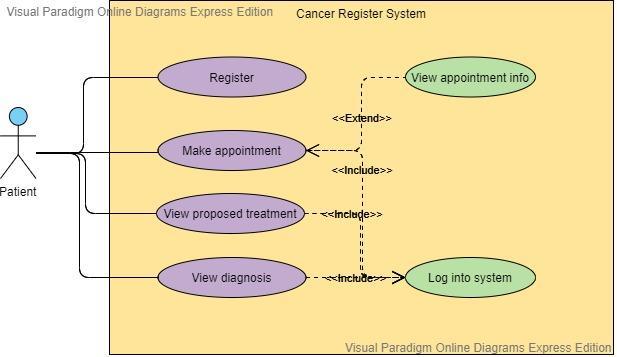
-Докторът-онколог трябва да може да въвежда в системата информация свързана с текущото състояние на пациента след извършване на медицински преглед.

-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



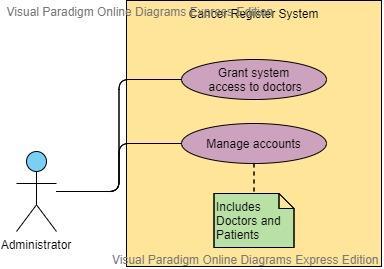
-Пациентът трябва да може да се записва в системата с необходимата за него лична информация.

-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.

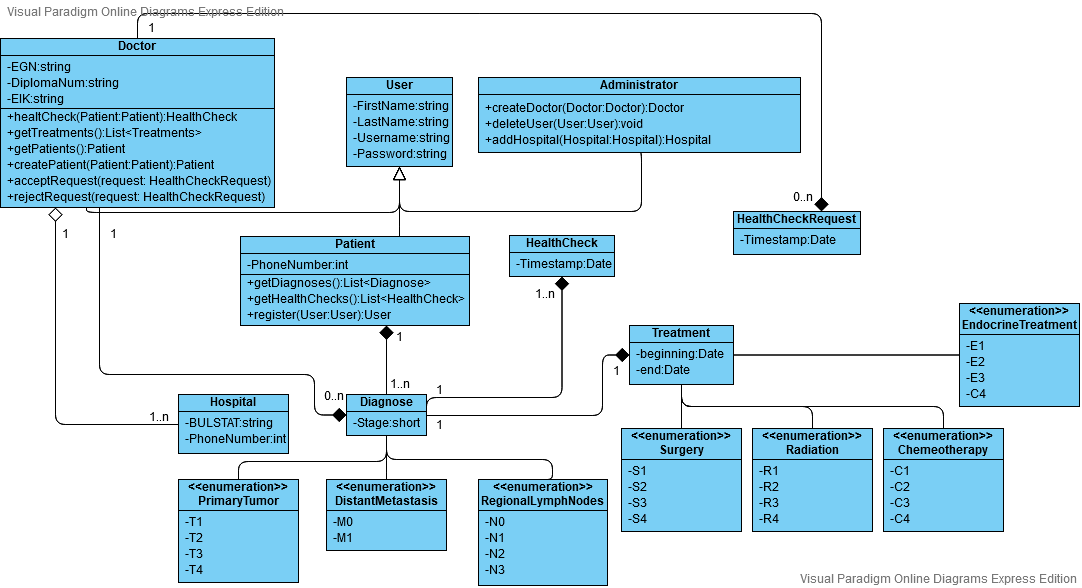
-Системата да показва текущата диагноза на пациента.

1. Взаимодействие на администратор със системата

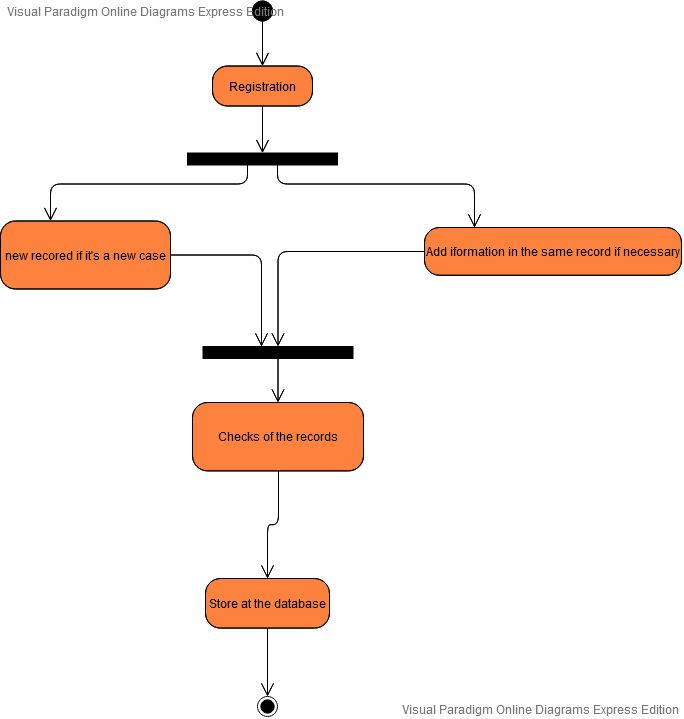


-За управлението на акаунтите на докторите-онколози и пациенти се изисква нова роля – администратор.

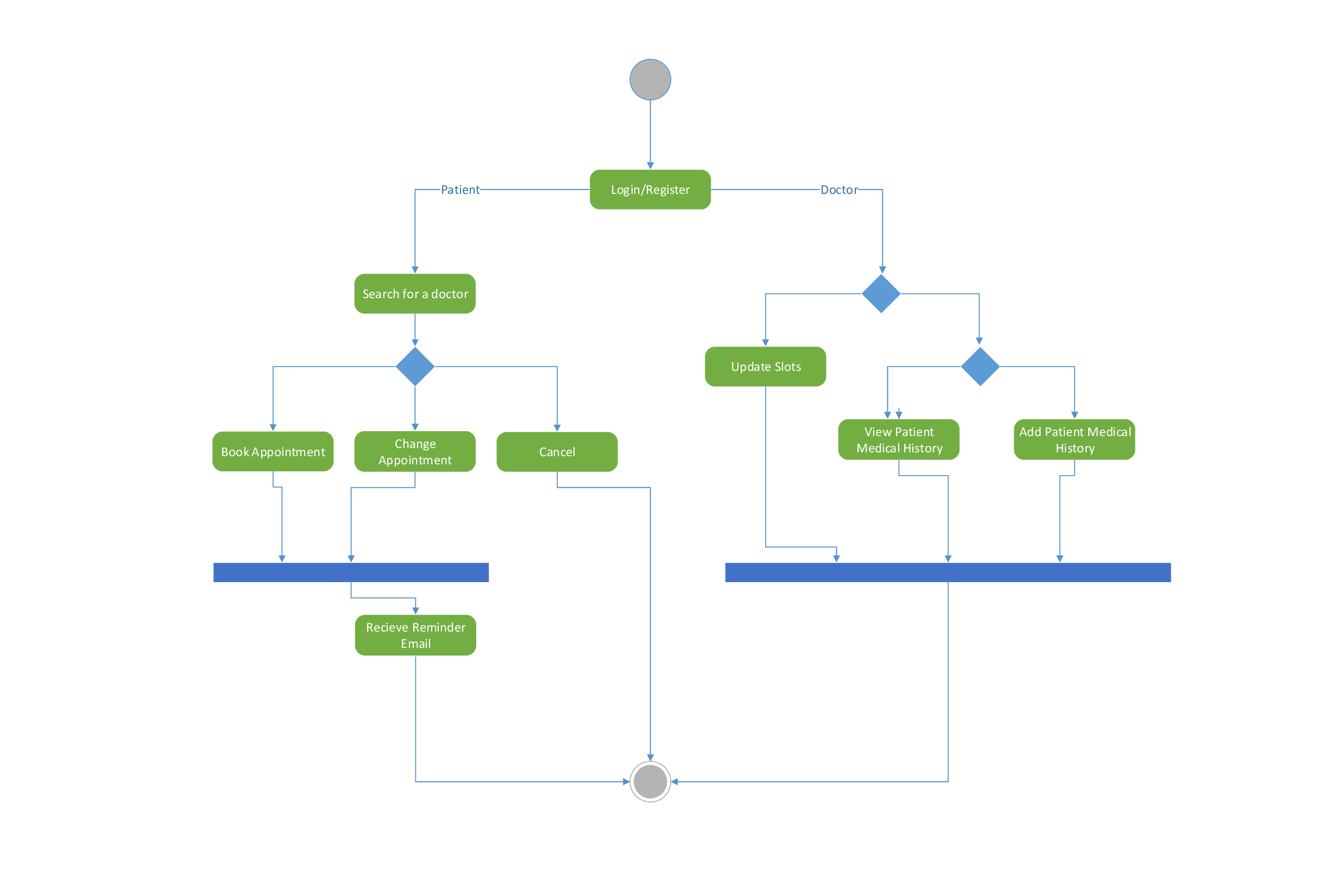
-Администраторът е единственият, който ще дава достъп на докторите до системата (той ще извършва тяхната регистрация).

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.

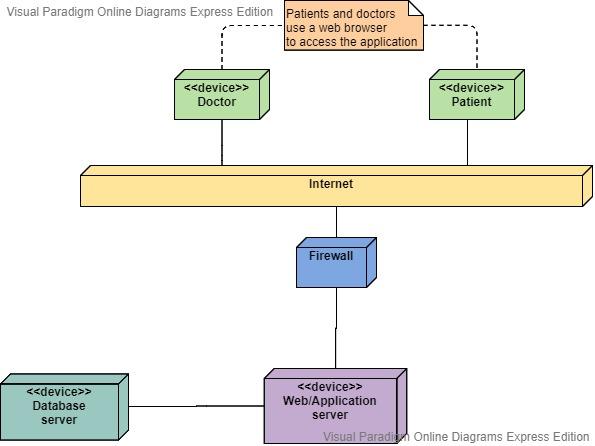
1. **Процесен изглед**

- 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.



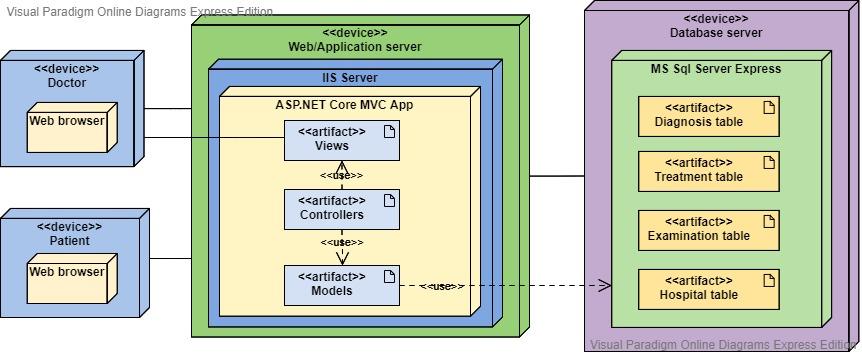
* Тази UML Activity диаграма представя процеса по логване , ако вече имаме направена регистрация или направата на регистрация, при отсъствието на такава, действието се извършва от една страна на лекарите , и от друга страна на пациентите. Всеки пациент, може да извърши търсене и да избере конкретен лекар-онколог, 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.

1. **Изглед на данните**
2. **Изглед на внедряването**
3. 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.

1. **Нефункционални изисквания**