

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Нұрғалиқызы Динара

Пациенттер мен дәрігерлерге арналған «Аурухананы басқару жүйесі» WEB-
қосымшасын әзірлеу

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021


ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ
УНИВЕРСИТЕТІ



Сәтбаев университеті
Кибернетика және ақпараттық технологиялар
институты
Киберқауіпсіздік, ақпаратты өңдеу және сақтау
кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
КАӨЖС кафедра меңгерушісі,
тех.ғыл.канд, ассоц.

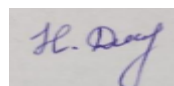
профессор
 Н.А.Сейлова
« 27 » _____ мая _____ 2021 ж..

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: Пациенттер мен дәрігерлерге арналған «Аурухананы басқару
жүйесі» WEB-қосымшасын әзірлеу

5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған:



Д.Нұрғалиқызы

Ғылыми жетекші:

 Зиро А.А.
«20» мамыр 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты


Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

БЕКІТЕМІН

КАӨС кафедра меңгерушісі,

тех.ғыл.канд, ассоц.

профессор

 Н.А.Сейлова

« 27 » _____ мая _____ 2021 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Нұрғалиқызы Динара

Тақырыбы: Пациенттер мен дәрігерлерге арналған «Аурухананы басқару жүйесі» WEB-қосымшасын әзірлеу

Университет Ректорының 2020 жылғы «24» 11 №2131-б бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі 2021 жылғы « 20» мамыр

Дипломдық жұмыстың бастапқы берілістері: диплом алдындағы практикалық жұмыс қорытындысы, тақырып бойынша әдебиеттерге шолу нәтижелері, теориялық мәліметтердің жиыны

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а)қойылған мәселенің қазіргі жағдайын пайымдау

ә)ақпараттық қамтаманы құру

б)программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар

Сызба материалдар: ___ слайдпен көрсетілген


Ұсынылатын негізгі әдебиет: 15 атау

Дипломдық жұмысты дайындау

КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	28.02.2021	
Ақпараттық қамтаманы құру	07.04.2021	
Программалық қамтаманы құру	03.05.2021	

Дипломдық жұмысының бөлімдерінің кеңесшілері мен норма бақылаушыларының аяқталған жобаға қойған **қолтаңбалары**

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Норма бақылаушы	Кабудллин Максат		
Программалық қамтама			

Ғылыми жетекші



Лектор Зиро А.А.

Тапсырманы орындауға алған білім алушы



Д.Нұрғалиқызы

Күні: « 20» мамыр 2021 ж.

АҢДАТПА

Бұл дипломдық жұмыста аурухананы басқаруға арналған WEB-қосымшасын әзірлеу қарастырылады. Веб-қосымша аурухана процесін автоматтандыратын жүйе ретінде даярланатын болады. Бұл дипломдық жұмыстың мақсаты пациентке қажетті дәрігердің қабылдауына тез әрі оңай жазылуға, зертханалық талдаулардың нәтижелерін тексеруге және аурухана қызметкерлеріне науқас туралы негізгі ақпаратқа қол жеткізуге мүмкіндік беретін, қолданушы интерфейсі ыңғайлы қосымшаны әзірлеу. Бағдарламаның серверлік жағы C # бағдарламалау тілінде ASP.NET Core веб-қосымшаларын құруға арналған танымал фреймворк Model-View-Controller үлгісімен қолданылып жүзеге асырылады. Клиент жағы HTML, CSS, JavaScript технологияларын қолданып жазылатын болады.

АННОТАЦИЯ

В данной дипломной работе рассматривается разработка веб-приложения для управления больницей. Веб-приложение будет реализовано в виде системы для автоматизации больничного процесса. Целью данной дипломной работы является разработка приложения, с удобным пользовательским интерфейсом, которое позволяет пациенту быстро и легко записаться на приём к нужному врачу, проверять результаты лабораторных анализов, а сотрудникам больницы иметь доступ к базовой информации о пациенте. Серверная часть приложения будет реализовано на языке программирования C# с использованием популярного фреймворка для создания веб приложений ASP.NET Core с помощью паттерна Model-View-Controller. Клиентская часть будет написана с применением технологий.

SUMMARY

This thesis deals with the development of a web application for hospital management. The web application will be implemented as a system to automate the hospital process. The purpose of this thesis is to develop an application with a convenient user interface that allows the patient to quickly and easily make an appointment with the necessary doctor, check the results of laboratory tests, and the hospital staff have access to basic information about the patient. The server side of the application will be implemented in the C # programming language using the popular framework for creating ASP.NET Core web applications using the Model-View-Controller pattern. The client side will be written using HTML, CSS, JavaScript technologies.

МАЗМҰНЫ

КІРІСПЕ	10
1 АУРУХАНАНЫ БАСҚАРУ ЖҮЙЕСІ ҚЫЗМЕТІНЕ ЖАЛПЫ ШОЛУ	11
1.1 Аурухананы басқару веб-қызметіне зерттеу мен талдау жасау	11
1.2 Пациенттер мен дәрігерлерге арналған веб-қосымшаның қажеттілігі	12
1.3 Есептің қойылымы	12
2. Аурухананы басқару жүйесінің WEB-қосымшасын жобалау	14
2.1 Веб-қосымшаның архитектурасы	14
2.2 Веб-қосымшаны әзірлеуге арналған технологиялар мен құралдарға шолу	15
2.3 Ақпараттық жүйені жобалау	18
3 БАҒДАРЛАМАЛЫҚ ҚАМТАМАНЫ ҚҰРУ	23
3.1 Веб-қосымшаның бағдарламалық құрылымы	23
3.2 Серверлік бөлімнің компоненттерін әзірлеу	24
3.3 Клиенттік бөлімнің компоненттерін әзірлеу	26
3.4 Функционалдық тағайындалуы	27
ҚОРЫТЫНДЫ	
ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	
А ҚОСЫМШАСЫ	
В ҚОСЫМШАСЫ	

КІРІСПЕ

Бүгінгі таңда ғылыми-өнеркәсіптік революцияның ғасырында ақпараттық технологиялар процестері адам өмірінің барлық салаларын дерлік қамтиды, қалыптастыру және жетілдіру мақсатында жаңа мүмкіндіктерді қамтамасыз етеді. Қазіргі уақытта, ДК болмаған жағдайда, қоғамдағы бірде-бір компания мекеме болып табылмайды, сонымен қатар сырттай ақпараттық технологияларының мүмкіншіліктерінсіз ХХІ ғасырды елестету мүмкін емес. Біздің барлық күнделікті жұмысымыз, тіршілігіміз күн сайын дамып жатқан заманауи технологиялармен тікелей байланысты.

Ақпараттандырудың өсуі және таралуы денсаулық сақтау саласын да қамтыды. Инновациялық IT-зерттеулер халыққа медициналық көмек көрсетудің жаңа әдістерін қалыптастыруға оң әсерін тигізеді. Көптеген мемлекеттер денсаулық сақтау саласындағы соңғы технологиялық процестерді ұзақ уақыт бойы қарқынды қолданып келеді. Пациенттер мен қызметкерлерді телеконсультациялауды жүзеге асыру, әртүрлі мекемелер арасында пациенттер туралы мәліметтермен алмасу, физиологиялық параметрлерді қашықтықтан бекіту-осының бәрі медицина саласына ақпараттық технологиялық процестерді енуіне ықпал етеді. Бұл денсаулық сақтауды ақпараттандырудың қалыптасуын жаңа деңгейге шығарады, оның жұмысының барлық тұстарына жағымды әсер етеді.

Кең таралған технологиялардың бірі аурухананың қызметін басқару мақсатындағы веб-қосымшалар болып саналады.

Веб-қосымша – бұл құрылғының операциялық жүйесінде (ОЖ) жергілікті түрде іске қосылатын, өзге қосымшаларға қарағанда веб-серверде жұмыс жасайтын қолданбалы бағдарламалық жасақтама.

Веб-қосымшаның жүйелілігі клиент және сервер арасында бөлінеді, дегенмен деректер алмасу сілтемелер арқылы жүзеге асырылады. Қондырмалардың бұл түрі платформалық қызметтерге жатады, сондықтан ол пайдаланушы операторының тұжырымдамасына тәуелді емес, бұл жұмыс үстеліндегі қондырмаларға қарағанда үлкен артықшылық болып саналады.

Бұл жұмыста мен аурухананы басқаруды оңтайландыру мақсатында практикалық және көп функциялы веб-қосымшаны жобалауды және құруды қарастырып, дәрігерлерге, медициналық қызметкерлерге құжаттарды тез реттеуге және күнделікті аурухана мәселелерін үлкен тиімділікпен шешуге көмектесетін функционалды веб-қосымшаны әзірлеп шықтым.

1 АУРУХАНАНЫ БАСҚАРУ ЖҮЙЕСІ ҚЫЗМЕТІНЕ ЖАЛПЫ ШОЛУ

1.1 Аурухананы басқару веб-қызметіне зерттеу мен талдау жасау

Кез-келген басқа саладағыдай, аурухана өз қызметкерлерін үйлестіру, дәрігерлермен қарым-қатынас жасау мақсатында тиісті басқару тұжырымдамасын қажет етеді. Мамандандырылған бағдарламалық қамтамалар пайда болғанға және ақпараттық технологиялар дамығанға дейін ауруханалар кітаптапшаларға жазбалар жүргізіп, әр пациентті тіркеп және олардың ауру тарихы туралы жазбаларды сақтаған.

Арнайы жүйелер пайда болғанға және ақпараттық технологиялар дамығанға дейін ауруханалар кітаптарға жазбалар жүргізіп, әр пациентті тіркеді және олардың ауру тарихы туралы жазбаларды сақтады.

Қазіргі уақытта ауруханалардың едәуір бөлігі автоматтандырылған бағдарламаларға көшті. Мұндай шешімнің пайда болуы аурухана қызметкерлеріне пациенттерге қызмет көрсетуді айтарлықтай жеңілдетті, ал пациенттер бұдан былай өз кезеңдерін күтіп кезекте тұрмайды және медициналық тестілеу нәтижелерін алу үшін тікелей аурухана мекемесіне бару қажеттілігі төмендеді.

Соңғы жағдайларға байланысты онлайн-кеңестер өзекті бола бастады, науқастар мен медицина қызметкерлерінің тікелей байланысу процесі жеңілдетіледі.

Мұндай жүйелер науқастар туралы мәліметтерді жинау, оларды сақтау мақсатында қолданылады. Сонымен қатар, олар бас жеткізушілердің, қосалқы мердігерлердің, медицина қызметкерлерінің, сондай-ақ реттелуі керек басқа да қызметкерлердің жеке ақпараттарына қол жеткізу үшін қолданылады. Осындай күрделі жүйелердің арқасында аурухана қызметкерлері күнделікті мәселелерді көп күш салмай шешуге көмектесетін көп функционалды сандық механизмге ие.

Осында аурухананы басқару бағдарламалық қамтамасын енгізу басқару барысын оңтайландыру, қызметкерлердің өнімділігін арттыру секілді ауруханаға айтарлықтай артықшылықтар береді.

Науқастар, сондай-ақ дәрігерлер әрбір кезеңде науқас туралы деректерге қол жеткізуге барлық мүмкіндіктері бар, аурудың шежіресі, белгілі бір терапияны қарастыру нәтижелері. Пациенттердің қабылдауға жазылуға барлық мүмкіндіктері бар, сондай-ақ өз зерттеулерінің нәтижелерін бақылауға болады, дәрігерлер науқастың әл-ауқатына байланысты диагноз қою үшін осы ақпаратты қолдануға барлық мүмкіндіктері бар. Бұл қате ықтималдығын айтарлықтай төмендетеді.

Науқастар мен дәрігерлер кез-келген уақытта науқастың медициналық тарихы, тест нәтижелері және тағайындалған емдеу сияқты ақпаратқа қол жеткізе алады. Науқастар қабылдауға жазылып, тест нәтижелерін тексере алады, ал дәрігерлер бұл ақпаратты диагноз қою және науқастың денсаулығын бақылау үшін қолдана алады. Бұл қате мүмкіндігін айтарлықтай азайтады.

Диагноз қою үшін дәрігерлерге тест нәтижелері қажет. Қызметкерлер үйлестіруді және топтық жұмысты жақсартуы керек. Басқару бағдарламалық жасақтамасынсыз дәрігерлер қажетті сынақ нәтижелері мен тағайындалған дәрі-дәрмектерді алуға көп уақыт алады.

Ауруханалар аурухана қаржысын оңай басқара алады. Олар пациенттерге шот-фактуралар жасай алады, сатушылар мен жеткізушілерге шот-фактуралар жібере алады және табыс салығын жоспарлай алады. Қаржылық хабардарлық Бизнестің болашағын талдауға және дұрыс бағытта қозғалуға көмектеседі.

Процесті автоматтандырудың арқасында пациенттер туралы деректерді файлдарға жазуға көп уақыт жұмсаудың қажеті жоқ. Бұл қағазбастылықтан, сондай-ақ файлдарды басқарудан уақытты үнемдейді.

1.2 Пациенттер мен дәрігерлерге арналған веб-қосымшаның қажеттілігі

Денсаулық сақтау жүйесі – әлеуметтік-экономикалық қызметтің маңызды тармақтарының бірі, демек, ол ұтымды және тиімді басқаруды талап етеді. Бұл медициналық орталықтарда пайда болған мәліметтер ағынын тиісті түрде бақылауға мүмкіндік беретін құралды қажет етеді. Соңғы кездері денсаулық сақтаудың ақпараттық жүйелері біздің қоғамның барлық салаларында біздің өмір сүру сапамызды жоғарлатуға ықпалын тигізді, сондықтан технологиялар кешенінің өрбүді қажет ететіні сөзсіз. Ауруханалар денсаулық сақтау жүйесінің негізгі мүшелері ретінде айтарлықтай ақпарат көлемін жинайды, бірақ көп жағдайда ол шашыраңқы немесе жүйеленбеген. Сол себепті аурухана мекемесінің қызметін орталықтандыратын бағдарламалық қамтама әзірлеу пациенттерге орталықтың қызметтерімен және мамандарымен танысуға, және онлайн-қызметтерді қолдануға қолайлы орта болып табылады. Мұнда тек пациенттерге ғана емес, сонымен қатар дәрігерлерге пациенттер жайлы ақпаратты ұсынатын екі жақты функцияларға ие веб-қосымша қарастылады. Бұл жүйенің қолданылу аясы кең, шағын аурухана мекемелерінен бастап жекеменшік орта көлемдегі медициналық орталықтарды қамтиды.

1.3 Есептің қойылымы

Дипломдық жұмыстың мақсаты – аурухана процестерін тиімді басқаруды қамтамасыз ететін веб-қосымшаны әзірлеу.

Осы мақсатқа жету үшін келесі міндеттерді шешу қажет:

- пәндік облысқа талдау жүргізу;
- жобалауды өзекті құралдарын және бағдарламалық қамтамасыз етуді әзірлеу құралдарын зерттеу;
- веб-қосымшаға қойылатын талаптарды талдау;
- веб-саладағы заманауи сәулет шешімдерін талдау;
- дерекқорды басқару жүйесін (ДҚБЖ) таңдау);

- мәліметтер базасының тұжырымдамалық моделін құру;
- мәліметтер базасының дизайнын жасау;
- веб-қосымшаның құрылымы мен дизайнын жасаңыз;
- практикалық интерфейс құру;
- жалпы қабылданған стандарттар мен технологияларға сәйкес веб-қосымшаны әзірлеу;
- пайдаланушы интерфейсін тестілеу.

Әзірленген бағдарламалық жасақтама келесі талаптарға сәйкес келуі керек:

- веб-қосымшаның серверлік компоненті клиенттен алынған сұраныс бойынша жауап қалыптастырып, деректерді өңдеуі керек;
- серверге сұраныстар жіберетін клиенттік бөліктің болуы;
- дерекқорды басқару жүйесі ретінде базалық талаптарға жауап беретін реляциялық клиент-серверлі ДҚБЖ таңдалуы керек;
- қолданбада беттерді дұрыс шығару үшін бейімделген орналасу болуы керек.

Веб-қосымша келесі қызметтерді ұсынуы керек:

- пациенттер мен докторларды олардың дербес логині мен паролі бойынша авторизациялау және аутентификациялау;
- аурухананың көрсетілетін қызметтерінің беті және олардың құны болуы керек;
- аурухана мамандарының қызметтері, білімі, кәсіби тәжірибесі, мамандануы, жетістіктері мен марапаттары туралы ақпарат беру;
- онлайн-жазылым бөлімінде барлық қолжетімді дәрігерлердің тізімі болуы керек;
- авторизацияланған пациенттерге дәрігердің қабылдауына жазылуға мүмкіндік беру, талдау нәтижелерін қарау;
- дәрігерлерге өз пациенттерінің жеке деректерін(ауру тарихын), медициналық талдау нәтижелерін ұсыну;
- дәрігерлерге өз онлайн-жазылымдарының уақытын өзгертуге мүмкіндік беру.

Жүйеге қойылатын минималды талаптар:

- Intel Core i5 немесе AMD Ryzen 5 процессоры бар жеке компьютер;
- 8 ГБ жедел жады;
- кемінде 500 ГБ SSD.

Жүйе келесі талаптарға сай болуы керек:

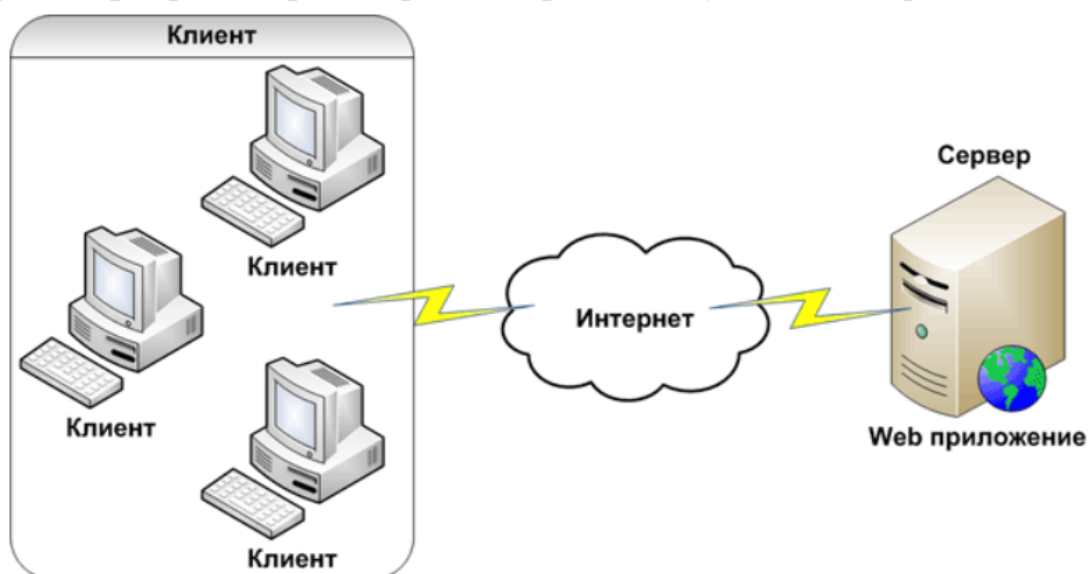
- бағдарлама C # / C ++, JavaScript тілдерінде жазылуы тиіс;
- Visual Studio немесе кез-келген басқа орта сияқты даму платформасы дайындалуы керек;
- мәліметтер базасын MS SQL Server немесе PostgreSQL-де құрылуы міндетті;
- ДК-де Windows 10 орнатылуы қажет;
- пайдаланушы интерфейсін тестілеу.

2 АУРУХАНАНЫ БАСҚАРУ ЖҮЙЕСІНІҢ WEB-ҚОСЫМШАСЫН ЖОБАЛАУ

2.1 Веб-қосымшаның архитектурасы

Веб-қосымшалардың архитектурасы бірнеше қосымшалардың бірлескен жұмысын қамтамасыз ету үшін қосымшалар, өтпелі бағдарламалық қамтамасыз ету тұжырымдамалары және мәліметтер базасы арасындағы өзара әрекеттесуді анықтайды. Пайдаланушы URL мекенжайын енгізіп, "өту" батырмасын басқан кезде, шолғыш веб-сайт орналасқан Интернетке кіретін компьютерді тауып, сол бетті сұрайды.

Содан кейін сервер файлдарды шолғышқа жіберу арқылы жауап береді. Осы әрекеттен кейін браузер сұралған бетті пайдаланушыға көрсету үшін осы файлдарды іске қосады. Енді пайдаланушы сайтпен өзара әрекеттесе алады. Әрине, бұл әрекеттердің барлығы бірнеше секунд ішінде орындалады.



2.1 Сурет – Веб-қосымшаның архитектурасы

Мұнда браузер талдаған код маңызды. Бұл кодта браузерге кірістердің кең ауқымына қалай жауап беру керектігі туралы нақты нұсқаулар болуы мүмкін немесе болмауы мүмкін. Нәтижесінде, веб-қосымшаның архитектурасы барлық ішкі компоненттерді және бүкіл бағдарламалық қосымшаның сыртқы қосымшаларымен алмасуды қамтиды.

Әрине, ол өзінің тиімді жұмыс істеу мақсатында мамандандырылған, сонымен қатар өзінің белгілі бір қажеттіліктерін қанағаттандырады. Веб-қосымшалардың архитектурасы өте маңызды, өйткені ғаламдық желілік трафиктің көп бөлігі және әрбір жеке бағдарлама мен құрылғы веб-байланысты пайдаланады. Ол масштабпен, тиімділікпен, сенімділікпен және қауіпсіздікпен айналысады.

Веб-қосымшалардың көмегімен сізде сервер және клиент жағы бар. Бір уақытта екі бағдарлама жұмыс істейді:

– браузерде өмір сүретін және пайдаланушының енгізуіне жауап беретін код

– серверде өмір сүретін және HTTP сұраныстарына жауап беретін код.

HTTP – бұл хабарламалар негізінде құрылған екі тараптың (клиент пен сервер) өзара әрекеттесуін сипаттайтын хаттама сұрау (Request) және жауап (Response). Әр хабарлама үш бөліктен тұрады: бастапқы жол, тақырыптар және дене. Оның үстіне тек бастапқы жол міндетті.

Сұрау мен жауаптың бастапқы жолдары әртүрлі форматқа ие, бізге тек сұраудың бастапқы жолы қажет:

METHOD URL HTTP/VERSION,

METHOD – бұл HTTP сұрау әдісі, URI-ресурс идентификаторы, version-протокол нұсқасы.

Тақырыптар – қос нүктемен бөлінген атау-мән жұптарының жиынтығы. Тақырыптарда әртүрлі қызметтік ақпарат беріледі: хабарламаны кодтау, шолғыштың атауы мен нұсқасы, клиент келген мекен-жай (Referrer) және т.б.

Хабарламаның денесі- тікелей берілетін ақпарат.

Біздің архитектурамыз HTTP протоколына негізделгендіктен, қосымшаның ресурстарын басқару GET, POST, PUT және DELETE сияқты әдістерді қолдану арқылы жүзеге асады.

GET– ресурс алу.

POST– ресурс құру.

PUT– ресурсты жаңарту.

DELETE – ресурсты жою.

Шын мәнісінде, HTTP сұраныстарына жауап бере алатын кез-келген код серверде жұмыс жасай алады. Сервер жағындағы кодтың тағы бірнеше атрибуттары:

– пайдаланушы ешқашан көрмейді (сирек кездесетін жағдайларды қоспағанда);

– пайдаланушы профилдері, твиттер, парақтар және т.б. сияқты деректерді сақтайды;

– пайдаланушы сұраған бетті қайтарды.

Клиенттік кодта келесі тілдер қолданылады:

– CSS

– Javascript

– HTML

Веб-қосымшалардың жақсы архитектурасы жобалау керек, себебі бұл алдағы уақытта сұраныстың артуына және сенімділіктің жоғарылауына ықпалын тигізіп, болашақ өсуді қолдау жоспары болады.

2.2 Веб-қосымшаны әзірлеуге арналған технологиялар мен құралдарға шолу

Бүгінгі таңда бағдарламалық жасақтаманы әзірлеудің көптеген құралдары бар, олар жыл сайын жетілдіріліп, қайта әзірленеді. Өз бағдарламалық жасақтаманы дамытқысы келетіндер үшін кең таңдау мен әр

түрлі ұсынылатын баламалар бар.Әр технология бір-бірінен өзінің функционалдары мен мүмкіндіктерімен ерекшеленеді. Әзірлеу платформасы мен құралдарды таңдауға жауапкершілікпен келу керек, себебі ол заманауи ІТ стандарттарына сәйкес келуі керек.

Бұдан әрі бағдарламалық қамтамасыз етуді құруды қолдайтын құрылғылардың тізімі келтірілген.

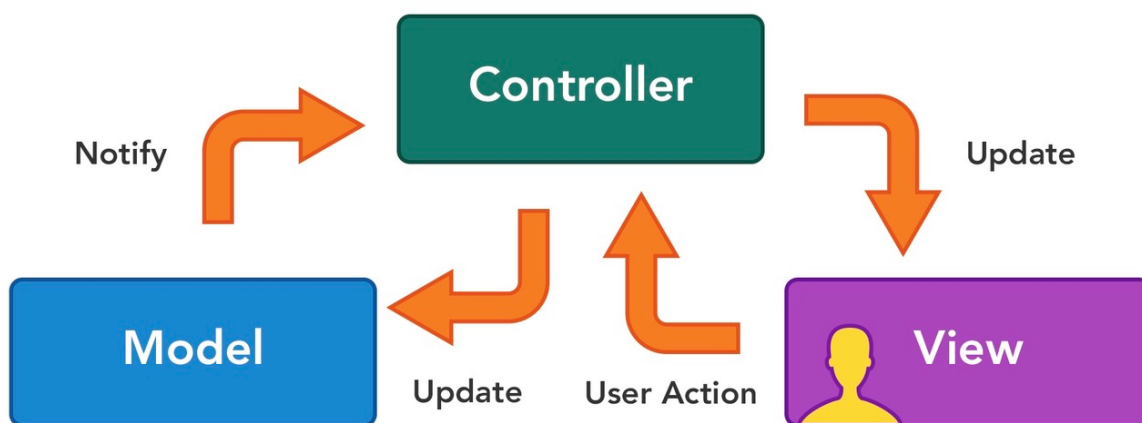
C# объектіге және аспектке бағытталған тұжырымдамаларды біріктіретін бағдарламалау тілі. 1998-2001 жылдары Microsoft .NET платформасына арналған қосымшаларды әзірлеудің негізгі тілі ретінде Microsoft фирмасында Андерс Хейлсберг басқармасы жанында инженерлер тобы ойлап тапқан .

C# -C-тәріздес синтаксисті тілдер тобына тиесілі, олардың арасында оның синтаксисі C++ және Java-ға тілдеріне ұқсас. Тіл қатаң статикалық типизацияны ұстанады, полиморфизмді, класс мүшелерінің әдістерін, операторлардың шамадан тыс жүктелуін, атрибуттарды, оқиғаларды, XML форматындағы түсініктемелерін қолдайды. C++, Java және т.б. C-тәріздес тілдерден C# біраз нәрсені алып, осы аталып өткен тілдердің тәжірибесіне арқа сүйей отырып, бағдарламалық жүйелерді өрбітуде проблемалық деп танылған кейбір модельдерді есепке алмайды: мысалы, C# бірнеше классты, яғни көптік мұрагерлікті қолдамайды (с++-тен ерекшелігі).

C# тілі .NET платформасының негізгі бағдарламалау тілі болып табылады, ол бүгінгі күнге дейін кез-келген түрдегі қосымшаны әзірлеуге мүмкіншілік береді.

ASP.NET Core – ашық бастапқы кодты веб-қосымшаларды құруға арналған еркін таратылатын платформалық орта. Бұл платформаны Microsoft қоғамдастықпен бірлесіп әзірлейді және ASP.NET салыстырғанда үлкен өнімділікке ие.

Модель-көрініс-контроллер (MVC) – бұл бағдарламаны үш негізгі логикалық компоненттерге ажырататын архитектура: модель, көрініс және контроллер. Бұл компоненттердің әрқайсысы қосымшаны әзірлеудің белгілі бір тұстарын өңдеуге арналған. MVC – кеңейтілген жобаларды құру үшін жиі қолданылатын стандартты веб-әзірлеу паттерндердің бірі.



2.2. Сурет – MVC архитектурасы

Model элементі пайдаланушы жұмыс істейтін мәліметтермен байланысты барлық логикаға жауап береді. Ол View және Controller компоненттері арасында берілетін деректерді немесе бизнес логикасына қатысты кез келген басқа деректерді ұсына алады. Мысалы, Customer нысаны тұтынушы туралы ақпаратты дерекқордан алады, оны басқарады және деректерді дерекқорға жаңартады немесе деректерді визуализациялау үшін пайдаланады.

View элементі қосымшаның барлық UI(пайдаланушы интерфейсі) логикасы үшін қолданылады. Мысалы, "Тұтынушы " көрінісінде соңғы пайдаланушы өзара әрекеттесетін мәтіндік жолдар, ашылмалы тізімдер және т.б. сияқты пайдаланушы интерфейсінiң барлық компоненттері болады.

Controller модель мен көріністер арасындағы аралық компонент ретінде бүкіл бизнес логикасы мен кіріс сұраныстарын өңдеуге, модель компонентін қолдана отырып мәліметтерді басқаруға және соңғы нәтижені визуализациялау үшін көріністермен өзара әрекеттесуге қызмет етеді. Мысалы, тұтынушы контроллері қолданушының көрінісіндегі барлық өзара әрекеттесулер мен кірістерді өңдейді және қолданушы моделін қолдана отырып деректер қорын жаңартады. Дәл сол контроллер тұтынушы деректерін қарау үшін қолданылады.

ASP.NET MVC келесі функционалдарды ұсынады:

- күрделі, бірақ жеңіл қосымшаларды әзірлеу үшін өте қолайлы;
- оңай ауыстыруға және конфигурациялауға болатын кеңейтілетін және қосылатын құрылымды ұсынады. Егер сіз кіріктірілген Razor немесе ASPX view қозғалтқышын қолданғыңыз келмесе, сіз кез-келген басқа сыртқы тарап қарау механизмдерін қолдана аласыз немесе барларын өзгерте аласыз;
- бағдарламаның компоненттік дизайнын қолданады, оны Модель, Көрініс және Контроллер элементтеріне логикалық түрде ажыратады. Бұл бағдарлама жасақтаушыларына ауқымды, күрделі жобаларды басқаруға және жеке компоненттермен жұмыс істеуге мүмкіншілік береді;
- MVC құрылымы сынақтан өткізуге негізделген жасақтаманы және қосымшаның сыналуын жақсартады, себебі барлық компоненттерді интерфейс негізінде әзірлеуге және модельдеу элементтерін қолдана отырып тексеруге болады. Демек, ASP.NET MVC Framework веб-әзірлеушілердің үлкен тобы бар жобалар үшін өте ыңғайлы;
- ASP.NET авторизация және аутентификация, басты беттер, деректерді байланыстыру, жеке басқару элементтері, мүшелік, бағыттау сияқты барлық қолданыстағы кең мүмкіндіктерді қолдайды;
- көру күйі ұғымын пайдаланбайды, бұл жеңіл қосымшаларды құруға көмектеседі және жасақтаушыларға толық бақылау береді;

Осылайша, біз MVC Framework-ті компоненттерге негізделген дамуға және сынауға бағытталған қосымша функциялардың үлкен жиынтығын ұсынатын, ASP.NET үстінен құрылған негізгі платформа ретінде қарастыра аламыз.

2.3 Ақпараттық жүйені жобалау

Кез-келген ұйым ақпаратқа уақтылы қол жеткізуді талап етеді. Қазіргі кезде ақпараттың құндылығы өте жоғары бағаланады. Ақпарат тұжырымдамаларының рөлін көп жағдайда мәліметтер базасы атқарады. Деректер базасы ақпараттың сенімді сақталуын, құрылымдалған түрін және оған уақтылы қол жеткізуді қамтамасыз етеді. Іс жүзінде кез-келген заманауи ұйымға деректерді сақтау, басқару бойынша белгілі бір қажеттіліктерді өтейтін мәліметтер қоры керек.

Аурухана-бұл қызметкерлер мен пациенттер туралы өте үлкен ақпарат ағығымен жұмыс істейтін мекеме. Дәрігерлер өз пациенттері туралы, науқастарды емдеу курсы туралы мәліметтерді үнемі бақылап отырады. Ал басшылық өз қызметкерлері туралы ақпараттан хабардар болуы қажет. Ол үшін сізге барлық қажетті ақпаратты қамтитын жалпы мәліметтер базасы қажет. Бағдарлама бүгінгі күні өте өзекті болып табылады, ол дерекқормен жұмысты автоматтандырады және пайдаланушыға (операторға) түсінікті және қарапайым интерфейсті ұсынады.

Барлық талаптарды ескеретін аурухана үшін мәліметтер базасын жасау үшін келесі талаптар орындалуы керек:

- деректер базасы аурухана туралы, атап айтқанда қызметкерлер мен пациенттер туралы барлық ақпаратты қамтуы керек;
- деректер базасында дәрігерлерге диагноздар және оларды емдеу әдістері туралы анықтамалық ақпарат болуы керек;
- ауруханада бар қызметтер және олардың құны жайлы ақпарат;
 - деректерге өзгерістер енгізу және жаңа деректермен толықтыру мүмкіндігі болуы тиіс;
 - бағдарламада іздеу, белгілі бір сұраныстарды орындау функциялары болуы.

Мәліметтер базасын жобалау құралы ретінде Microsoft SQL Server ДҚБЖ таңдалды.

Microsoft SQL Server-бұл қоғамдық IT орталарында транзакцияларды, бизнес-аналитиканы өңдеуге арналған бірқатар қосымшаларды қолдайтын реляциялық компонент ретінде жұмыс атқаратын дерекқорды басқару жүйесі (ДҚБЖ). Microsoft SQL Server-бұл Oracle Database және IBM DB2-мен бірге нарықтағы озық үш дерекқор технологиясының бірі.

Басқа да ДҚБЖ бағдарламалық жасақтамалары секілді, Microsoft SQL Server де дерекқордың әкімшілері (DBAS) және басқа IT-мамандар дерекқорды басқару және олардағы деректерге сұраныс жасау үшін пайдаланатын стандартталған бағдарламалау тілі SQL негізінде құрылған. SQL Server Microsoft-пен құрылған, стандартты бағдарламалау тіліне өз кеңейтілімдерін қосатын Transact-SQL (T-SQL) процедуралық кеңейтіліміне айтарлықтай байланған.

Басқа да ДҚБЖ технологиялары секілді, SQL Server негізінен мәліметтер базасының бірнеше орындарында артық сақтау қажеттілігін

болдырмай, әртүрлі байланысқан кестелердегі мәліметтер элементін өзара қосатын кестелердің және жолдарың негізінде құрылған. Реляциялық модель сонымен қоса мәліметтердің дәлдігін сақтау мақсатында сілтемелік тұтастықты және деректердің дәлдігін қамтамасыз ететін басқа да тұтастық шектеулерін ұстанады. Бақылау деректері атомдық, дәйектілік, оқшаулану негіздерін неғұрлым кең сақтаудың құрамдас бөлігі болып саналады, сонымен қатар ACID сапасы сияқты танымал жиынтықтарда мәліметтер базасының транзакцияларын дұрыс өңдеуді қамтамасыз ету мақсатында қарастырылған.

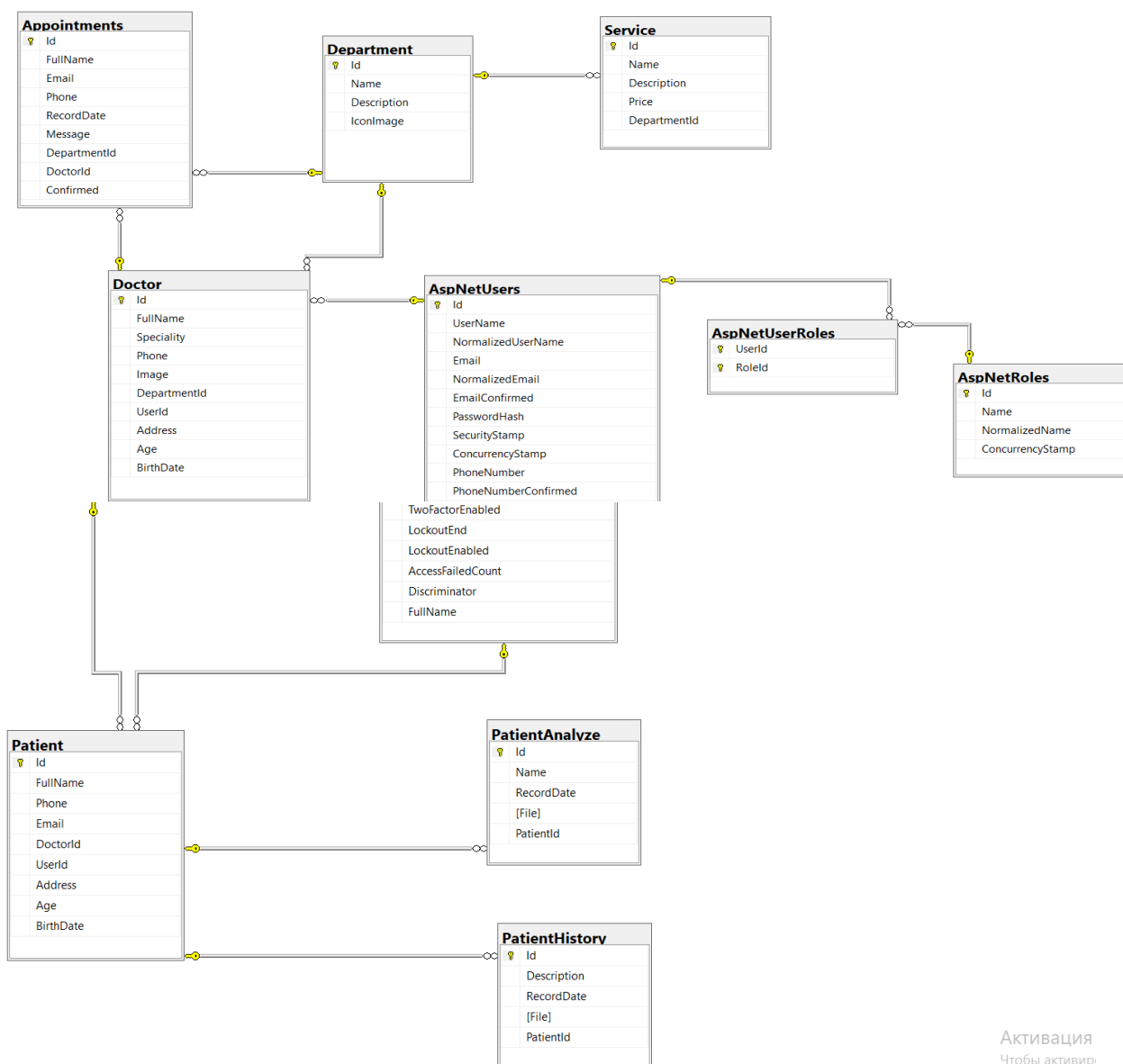
Microsoft SQL Server негізгі компоненті SQL Server Database Engine элементі болып саналады, ол ақпаратты үнемдеуге, өңдеуге және қорғауға жауап береді. Оған командалар мен сұраныстарды өңдейтін реляциялық механизм, сонымен қатар дерекқор файлдарын, кестелерді, беттерді, индекстерді, деректер буферлерін және транзакцияларды басқаратын сақтау механизмі кіреді. Сақталған процедуралар, триггерлер, көріністер және дерекқордың басқа нысандары да Database Engine компонентімен жасалады және орындалады.

MS SQL-де кең мүмкіндіктер бар. Бұл дерекқордың көрнекті ерекшелігі – транзакцияларды интерактивті күйде өңдеуге мүмкіндік беріледі, динамикалық блоктауы бар тұрақты және сенімді, қалпына келтіру процесі күш жұмсамайды және техникалық қызмет көрсету шығындары шығындары мен қол күштері аз. Объект-реляциялық модельді қолдана отырып жасалғандықтан күрделі құрылымдарды және қолданушы анықтаған және анықтайтын мәліметтер түрлерінің кең спектрін қолдайды. Ол деректердің кең сыйымдылығын қамтамасыз етеді және деректердің тұтастығына сенімді. Сонымен қатар ДҚБЖ басқа Microsoft өнімдерімен жұмыс істеуді қолдайды: Access, MS Excel.

Деректер қорын жобалау процесінде келесі нысандар айқындалды:

- Жазылым (Appointments);
 - Дәрігер (Doctor);
 - Қызмет (Service);
 - Бөлім (Department);
 - Пациент (Patient);
 - Пациент анализдері (PatientAnalyze);
 - Пациент жазбалары (PatientHistory);
 - Пайдаланушы(AspNetUsers);
 - Пайдаланушы рөлі(AspNetUserRoles);
 - Рөл(AspNetRoles).

Төмендегі суретте нысан атрибуттары (кестелер) және кестелер арасындағы байланыстары бар мәліметтер базасының моделі көрсетілген.



Активация
Чтобы активир.

2.3 Сурет – Мәліметтер базасының моделі

Субъектілер арасындағы қатынастарды сипаттаймыз.

"Дәрігер" мәні "Жазба" мәнімен "бір-көпке" деген қатынаспен байланысты.

"Бөлім" және "Дәрігер" мәні "бір-көпке" деген көзқараспен байланысты.

"Бөлім" мәні "Қызмет" мәнімен "бір-көпке" қатынасымен байланысты.

"Дәрігер" мәні аталған байланыстардан басқа "Пациент" мәнімен "бір-көпке" қарым-қатынасымен және "бір-бір" қатынасымен "Пайдаланушылар" мәнімен байланысты.

"Пациент" мәні "Пайдаланушылар" мәнімен "бір-бір" деген қатынаспен байланысты.

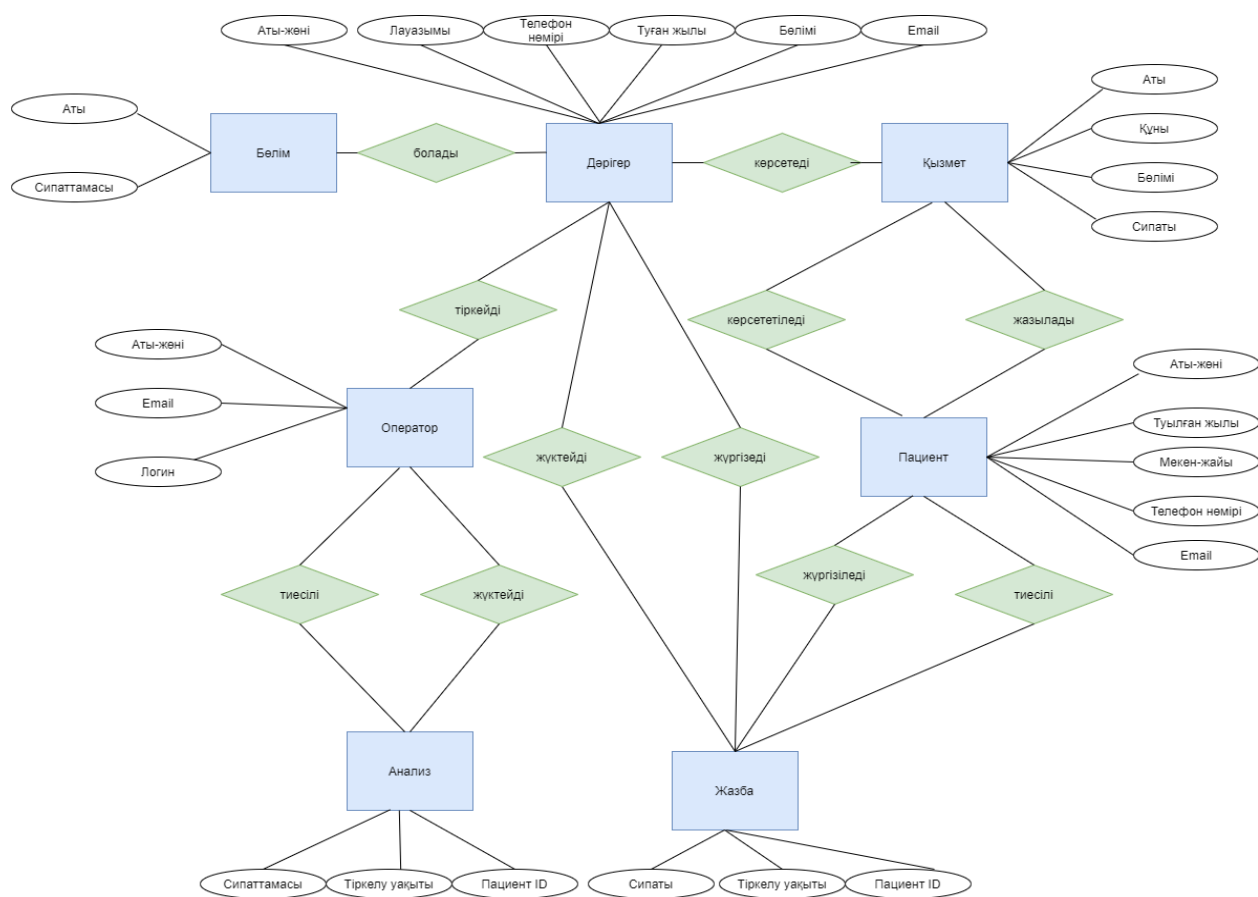
"Пациент" мәні "Жазбалар тарихы" және "Талдаулар" мәнімен "бір-көпке" қатынасымен байланысты.

"Пайдаланушылар" мәні "Рөлдер" мәнімен "көп-көпке" қатынаспен байланысты.

Көп-көпке – әр субъект бір немесе одан да көп басқа субъектілерге сәйкес келетін байланыс түрі. Байланыстың бұл түрін жүзеге асыру үшін байланыстырушы кестелердің сыртқы кілттері бар делдалдық кесте қолданылады.

Бір-көпке – байланыс түрі, онда бірінші нысан екінші нысанның тек бір данасына жатады, ал екінші нысан бірінші нысанның бір немесе одан да көп даналарын қамтиды.

Бір-бірге – байланыс түрі, онда әр субъект басқа объектінің бір данасын ғана көрсетеді. 1: м байланысын болдырмау үшін кестелердің бірінің сыртқы кілтінің шегін ерекше ету керек. Ал, онда бірінші нысан екінші нысанның бір данасына ғана тиесілі, ал екінші нысан бірінші нысанның бір немесе одан да көп даналарын қамтиды.

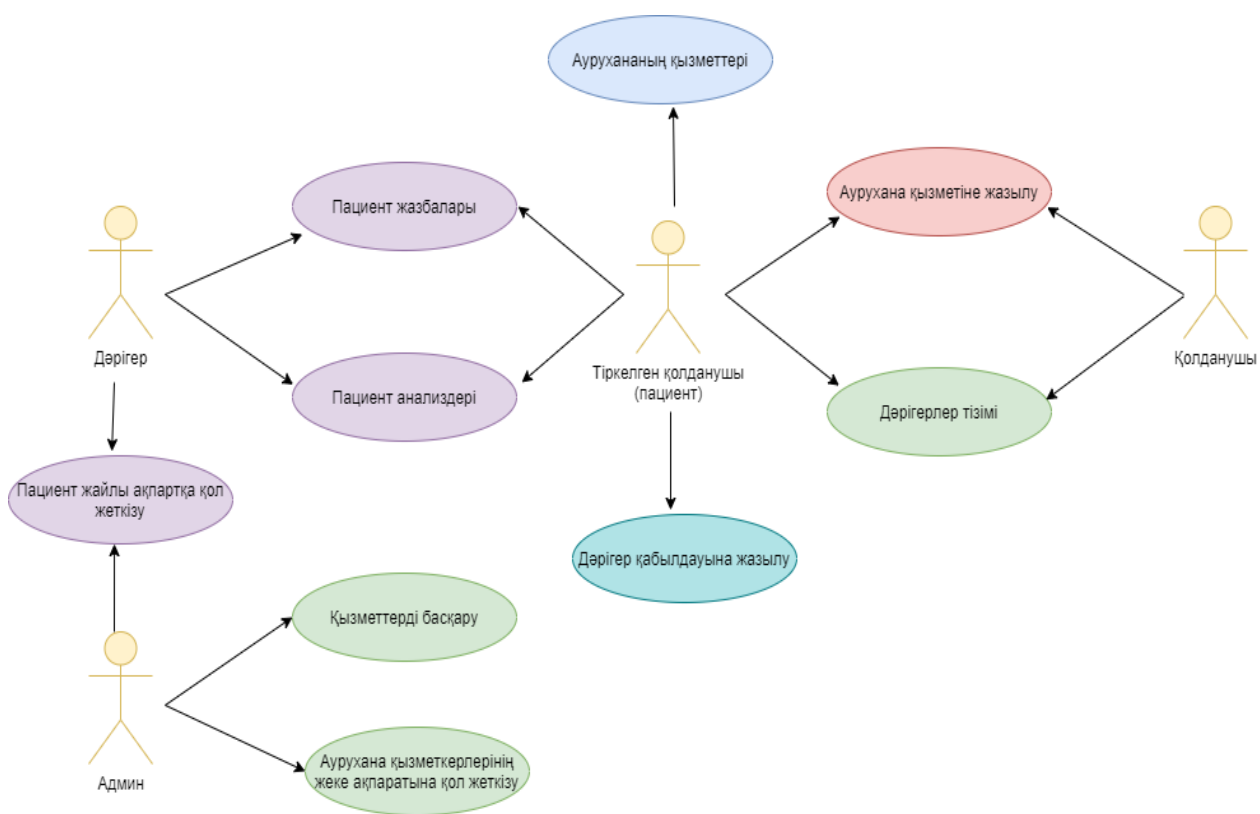


2.4 Сурет – Чен нотациясындағы ER-диаграмма

Суретте аурухана мекемесі аймағында пациенттерді қабылдау мен тіркеу, дәрігерлерге өз пациенттері жайлы мәліметтерге қол жеткізу үшін басқару жүйесінің объектілері арасындағы байланысты айқындайтын “болмыс-байланыс” диаграммасы көрсетілген.

Аурухана операторының тіркеу жүргізетін бір БАҒДАРЛАМАСЫ және ДЕРБЕС КОМПЬЮТЕРІ бар. АУРУХАНЫҢ ӨКІЛЕТТІ ҚЫЗМЕТКЕРІ жүйеге ДӘРІГЕРЛЕРДІ ТІРКЕЙДІ. Өз кезегінде ДӘРІГЕР бір БӨЛІМГЕ қарайды, ПАЦИЕНТЕРДІҢ ЖАЗБАЛАРЫН жүргізеді. ҚЫЗМЕТТІ бір БӨЛІМГЕ тиесілі болуы мүмкін, ал ПАЦИЕНТ 0 немесе одан да көп болуы мүмкін. ПАЦИЕНТ 1 немесе одан да көп ҚЫЗМЕТКЕ жазыла алады. Өз кезегінде, ОПЕРАТОР АНАЛИЗ нәтижелерін жүктей алады

ОПЕРАТОР, ДӘРІГЕР, ПАЦИЕНТ, ҚЫЗМЕТ, ЖАЗБА, АНАЛИЗ және ҚЫЗМЕТ субъектілері жүйеде әрқашан болатын тәуелсіз деректерді ұсынады. ПАЦИЕНТ - жүйенің басқа субъектілеріне тәуелді мәлімет. БОЛАДЫ НЕМЕСЕ ТІРКЕЙДІ шартсыз қатынастар болып табылады. Қалғандары - бұл болмыстар арасындағы шартты қатынастар.



2.5 Сурет – Прецеденттер диаграммасы

Прецеденттер диаграммасы – бұл бағдарламалық жасақтаманың немесе процестің негізгі функцияларын көрнекі түрде көрсетудің оңай әдісі

Менің веб-қосымшамда ұсынылатын бір қатар мүмкіндіктерге қол жеткізу үшін жүйені қолданушыларға рөл тағайындалады. Әр қолданушы өз рөліне сәйкес функционалдарға қол жеткізе алады

Веб-қосымша маңайында анықталған рөлдер келесідегідей:

- Жүйеде тіркелмеген қолданушылар
- Тіркелген қолданушылар(шартты түрде пациенттер)
- Дәрігерлер

– Операторлар(админдер)

Прецеденттер диаграмасында осы рөлдер бойынша бөлінген функционалдар сипаттамасы бейнеленген.

3. БАҒДАРЛАМАЛЫҚ ҚАМТАМАНЫ ҚҰРУ

3.1 Веб-қосымшаның бағдарламалық құрылымы

Төмендегі 3.1– суретте көрсетілгендей бұл дипломдық жұмыстың бағдарламалық құрылымы жүйелік, инструменталдық және қолданбалы бағдарламалық қамтамаларды қолдана отырып даярланды.



3.1 Сурет– Веб-қосымшаның бағдарламалық құрылымы

3.2 Серверлік бөлімнің компоненттерін әзірлеу

ASP.NET Core MVC архитектурасында орталық компонент болып табылатын контроллерің жасақталу кезеңінде құрылады. Дегенмен контроллер мәліметтермен жұмыс жасайтын болғандықтан деректерді ұсынатын модельдің құрылуы шарт. А Қосымшасында анықталған Appointments моделін талдайық. Мұндағы ар жол кестенің бағанын сипаттайды. Модельде жазылым идентификаторы, қолданушы аты-жөні, почта мекен-жайы, телефон нөмірі, жазылу уақыты секілді қасиеттер сипатталады. Модельді анықтау процесінен кейін оның мәліметтерін сақтайтын орынға, яғни деректер қорында кесте бейнесінде анықтауымыз қажет. Бұл мақсатта мәліметтерге қол жеткізуге және әрекеттесуге мүмкіндік беретін Entity Framework Core технологиясын қолдана аламыз. Ол бізге белгілі бір деректер қорына тәуелді болмай абстракциялануға мүмкіндік береді. Entity Framework Core-да мәліметтер базасымен өзара әрекеттесу арнайы деректер класы – контекст арқылы жүзеге асырылады. Сондықтан біз жобамызға HospitalDbContext деп аталатын және келесі коды бар жаңа класс қосамыз.

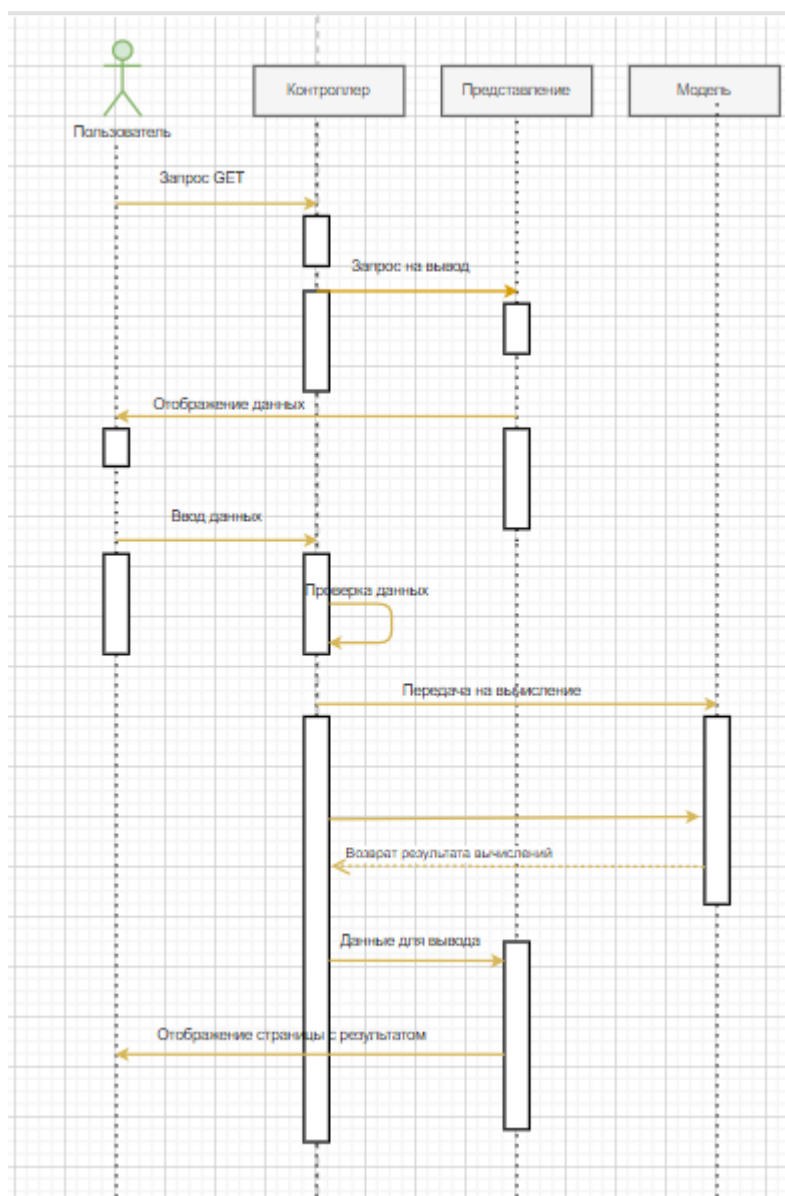
Ал осы модель бойынша деректер қорында кесте құрылу үшін миграция технологиясын қолдансақ жеткілікті. Миграция арқылы деректер қорында жаңа модель құруға, бар кестелерді олардың модельдері арқылы өзгертіге болады.

```
public partial class AddDepartmentToDatabase : Migration
{
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.CreateTable(
            name: "Department",
            columns: table => new
            {
                Id = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Name = table.Column<string>(type: "nvarchar(max)",
                    nullable: true)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_Department", x => x.Id);
            });
    }

    protected override void Down(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropTable(
            name: "Department");
    }
}
```

3.2 Сурет – Миграция коды

Ендігі кезекте контроллер қолданушаға көріністер қайтару үшін әдістерді қайтарады, мұндағы әдістерді көбіне әрекеттер деп те атайды. Контроллер, Көрініс моделінің іс-жүзінде әрекеттесу үлгісін қарастрайық.



3.3 Сурет – Тізбектілі диаграммасы

Бірінші кезекте қолданушы GET сұранысын жіберіп, оны контроллер өз әдісінде көрініске бағыттайды. Қолданушы интерфейсінде контроллермен генерацияланған көрініс пайда болып, ол көріністі қандай-да бір мәліметпен толтырады. Толтырылған мәліметтер валидация тексеруінен өтеді. Егер бәрі дұрыс болса, мәліметтер модельге өңдеге жіберілінеді. Модель деректерге қатысты логикамен жұмыс жасап, нәтижені контроллерге POST әдісі арқылы сілтейді. Алынған мәліметтермен толыққан көрініс қолданушаға қайтады. Дәл осы сипатталған әдіс бойынша қосымшадағы барлық құраушы компоненттер жұмыс жасайды.

3.3 Клиенттік бөлімнің компоненттерін әзірлеу

Қолданушы интерфейсін әзірлеу – бұл мәліметтермен толтырылған, пайдаланушы әрекеттесе алатын графикалық интерфейс жасақтау процесі. Бұл дипломдық жұмыста қолданушы интерфейсін көбінесе HTML тілінде UI коды бар cshtml кеңейтіліміндегі файлдар немесе html кодынан C#кодына ауысуға мүмкіндік беретін арнайы көрініс – Razor қозғалтқышы болып табылады. Бұл көрініс кәдімгі html бетіне ұқсайды. Мұнда html белгілеудің барлық стандартты элементтері анықталуы мүмкін, стильдер мен сценарийлер қосылуы мүмкін. Бірақ толығымен html беті әлі де көрініс деп айтуға болмас, себебі іс жүзінде бұл көріністер компиляцияланып, одан әрі қолданушы өзінің шолғышында көретін html беттерін генерациялау үшін қолданылады. Порт ретінде "http://localhost:44369" пайдаланылады. Әзірлеу процесін жеңілдету үшін көріністі модель сипатына қарай құрастырған жөн.

```
@model HospitalSystem.Models.Service

<div class="col-sm-3 filter @Model.Department.Name.Replace(' ',
    '_")">
    <div class="card" style="border-color: #2c4964;border-radius: 8px
7px;margin-bottom: 5px;margin-top: 10px; background-color: #e8f3fc">
        <div class="card-body">
            <h5 class="card-title">@Model.Name</h5>
            <p class="card-text" style="color:
rgba(0,0,0,.64);">@Model.Price</p>
            <a asp-controller="Service" asp-route-Id="@Model.Id" asp-
action="AppointmentToService" class="btn btn-
warning">Записаться</a>
        </div>
    </div>
</div>
```

3.4 Сурет – Көрініс коды

Код тізбегі мәліметтер контрольмен берілген моделді анықтаудан басталады. Әр жолда модельден керек деректер баған аты бойынша шақырылады. Пайдаланушы интерфейсінде генерацияланған код арқылы аурухана қызметтеріне жазылуға және ол туралы маңызды ақпараттарды алуға қолданылады.

3.4 Функционалдық тағайындалуы

Веб-қосымшаның құрылымын қарастырайық. Жасақталған веб-қосымшада бірнеше негізгі бөлімдер бар: негізгі бет, регистрация, авторизация, аурухана дәрігерлерінің тізімі, көрсетілетін қызметтер және олардың құны, әр бөлім жайлы қысқаша мәлімет, егер қолданушы авторизацияланған болса, оған дәрігер қабылдауына жазылу мүмкіндігі қол жетімді болады. Айта кететіні әр қолданушы рөліне қарай пайдаланушы интерфейсі ерекшеленеді. 3.5 – суретте жасақталған веб-қосымшаның негізгі беті көрсетілген. Басты бетте дәрігердің онлайн-қабылдауына жазылым, аурухана жайлы қысқа ақпарат, бөлімдер және дәрігерлер тізімі анықталған.

ДОБРО ПОЖАЛОВАТЬ

Консультация специалистов по любым интересующим вас вопросам

[ЗАПИСАТЬСЯ](#)

О НАС

MediCenter-одна из лучших клиник страны, располагающая «прорывными» лечебно-диагностическими технологиями, оказывающая широкий спектр медицинских услуг, соответствующих международным стандартам.

*MediCenter оказывает медицинскую помощь КРУГЛОСУТОЧНО, без праздников и выходных. Здесь 24/7 ведут прием все узкоспециализированные, квалифицированные врачи, которые тщательно разбираются в каждой ситуации, устраняют имеющиеся симптомы,

У нас вы можете пройти медицинское обследование

Получить консультацию узких специалиста

Лечиться у лучших докторов по инновационным технологиям

В центре можно сдать широкий спектр ЛАБОРАТОРНЫХ анализов, — воспользоваться услугами DETOX стационара, — сделать стандартное, двойное, премиум, VIP очищение организма, чтобы вывести токсины, избавиться от бессонницы, чувства тревожности и страха. Сотрудники центра соблюдают закон о врачебной тайне, сохраняют анонимность пациентов, не разглашают их имена

[Узнать больше](#)

УСЛУГИ



Уникальные программы лечения и эксклюзивные препараты, разработанные ведущими учеными



Комплексный подход в диагностировании и лечении заболеваний с применением новейшего медицинского оборудования



"Золотой стандарт" обслуживания, максимально ориентированный на нужды и интересы пациента

Диагностика

Гастроэнтерология

Амбулаторная хирургия

Кардиология

Гинекология

Дерматалогия

Физиотерапия

Урология

НАШИ СПЕЦИАЛИСТЫ



Абдраимова Аида

Кардиолог, хирург



Жумалиева Бакыт

Дерматалогия



Владитровна Раиса

Кардиолог, хирург



Дыбов Михаил Александрович

Уролог



3.5 Сурет – Веб-қосымшаның негізгі беті



Гастроэнтерологический центр занимается профилактикой, диагностикой и лечением заболеваний желудочно-кишечного тракта, печени и желчного пузыря. Гастроэнтерологам MedCenter в 95% случаев удается добиться ремиссии такой сложной патологии как язвенная болезнь. Деятельность центра апробирована Европейской и Российской ассоциациями гастроэнтерологов.

Услуги

Комплексное обследование 47500 тг	Записаться
Первичный приём врача-гастроэнтеролога от 5000 тг	Записаться
Видеоэзофагогастродуоденоскопия(ФГДС) от 4000 тг	Записаться
Скрининг желудка от 8000 тг	Записаться
Эндоскопия от 3000 тг	Записаться

Заболевания



Энцефалит Подробнее ->	Хеликобактер пилори Подробнее ->	Язва желудка Подробнее ->
---	---	--

[При каких симптомах следует обращаться к гастроэнтерологу?](#)

[Какие проблемы решает гастроэнтеролог?](#)

[Что входит в стоимость комплексной диагностики?](#)

Специалисты

 <p>Байменов Ерасыл Гастроэнтеролог Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Людмила Зайцева Гастроэнтеролог Записаться</p> <p>Twitter Facebook Instagram</p>
--	---

3.6 Сурет – “Гастроэнтерология” бөлімінің негізгі беті













“Отделения” бетінде мекеменің бөлімдері бойынша ақпарат алу мүмкіндігі бар. “Гастроэнтерология” бөлімін егжейлі-тегжейлі қарастырайық. Мұнда бөлім жайлы ақпарат жинақталған: көрсетілетін қызметтер, емделетін ауру түрлері, көптеп қойылатын сұрақтар және мамандары.

“Наши специалисты” бетінде мекеменің қызметкерлері (дәрігерлері) тізіміне қол жеткізуге болады.

medicenter@gmail.com +7 707 555 0000 Байзакова 75A, ALA

MediCenter Наши специалисты Услуги Отделения Пользователи Контент Пациенты Hello admin@admin.com! Logout

All Диагностика Кардиология Физиотерапия Хирургия Гастроэнтерология Урология Дерматология Гинекология

 <p>Абдраимова Аида Кардиология Кардиолог, хирург Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Жумалиева Бакыт Физиотерапия Дерматология Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Владимирова Раиса Кардиология Кардиолог, хирург Записаться</p> <p>Twitter Facebook Instagram</p>
 <p>Дыбов Михаил Александрович Урология Уролог Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Логоinov Руслан Физиотерапия Физиотерапевт Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Байменов Ерасыл Гастроэнтерология Гастроэнтеролог Записаться</p> <p>Twitter Facebook Instagram</p>
 <p>Стригунова Елена Гинекология Гинеколог, акушер Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Степанова Александра Гастроэнтерология Гастроэнтеролог Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Манапбаева Алия Диагностика Терапевт Записаться</p> <p>Twitter Facebook Instagram</p>
 <p>Семёновна Алёна Дерматология Дерматолог Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Муслимова Гаухар Физиотерапия Физиотерапевт Записаться</p> <p>Twitter Facebook Instagram</p>	 <p>Айтмаганбетов Алишер Хирургия Хирург Записаться</p> <p>Twitter Facebook Instagram</p>

© 2021 - HospitalSystem - Privacy

3.7 Сурет – “Дәрігерлер” тізімі

Мұнда дәрігерлерді бөлімдер арқылы фильтрациялау мүмкіндігі қол жетімді. Сонымен қатар тікелей қалаған дәрігерге жазылу мүмкіндігі қолжетімді.

Тіркелмеген қолданушыға ауруханда көрсетілетін қызметтер мен процедуралардың құнын саралап және жазылуға болады.

3.5 – суретте онлайн қабылдауға жазылу батырмасын басқанда ол бізді “Авторизация” бетіне сілтейді. Егер біз жүйеде тіркелген болсақ, онда осы формаға логиніміз бен құпия сөзімізді енгіземіз. Сонымен қатар біз құпия сөзімізді ұмытып қалған жағдайда оны қайта қалпына келтіру мүмкіндіген ие боламыз.

Авторизоваться

Используйте локальный
аккаунт для авторизации.

Email

The Email field is required.

Password

The Password field is required.

Remember me?

Авторизация

[Забыли свой пароль?](#)

[Зарегистрироваться](#)

3.8 Сурет – “Авторизация” терезесі

Шартты түрде біз жаңа пайдаланушы делік, онда “Регистрация” бетіне көшеміз.

Регистрация

Создать новый аккаунт.

FullName

Нургаликызы Динара

Email

dina.30.08@mail.ru

PhoneNumber

87758894976

Password

.....

Confirm password

.....

Регистрация

3.9 Сурет – “Регистрация” терезесі

Біздің веб-қосымшада пайдаланушыларды тіркеу және авторизациялау мақсатында ASP.NET Identity кіріктірілген жүйесі қолданылады. Жаңа қолданушыны тіркеу үшін толық аты-жөні, почта мекен-жайы, телефон нөмірі және құпия сөзді енгізу талап етіледі.

Жаңа пайдаланшының тіркелгенін тексеру үшін деректер базасынан тіркелген қолданушылар кестесінен біле аламыз.

```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) *
FROM [HospitalSystem].[dbo].[AspNetUsers]
```

Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	PasswordHash
1	customer@customer.com	CUSTOMER@CUSTOMER.COM	customer@customer.com	CUSTOMER@CUSTOMER.COM	0	AQAAAAEAAcQA/
2	sinatranur3462@gmail.com	SINATRANUR3462@GMAIL.COM	sinatranur3462@gmail.com	SINATRANUR3462@GMAIL.COM	0	AQAAAAEAAcQA/
3	75457e68-3b04-448d-a4f7-0420468d7786	RATOASUMI@GMAIL.COM	ratoasumi@gmail.com	RATOASUMI@GMAIL.COM	0	AQAAAAEAAcQA/
4	7fb0e809-fb77-4e4a-bd30-bf9184af925e	DOCTOR2@DOCTOR.COM	doctor2@doctor.com	DOCTOR2@DOCTOR.COM	0	AQAAAAEAAcQA/
5	8c7d5c98-5130-47c0-862a-a2e9522c4275	DOCTOR@DOCTOR.COM	doctor@doctor.com	DOCTOR@DOCTOR.COM	0	AQAAAAEAAcQA/
6	96203a5d-80f5-40ef-9f5c-f9d181243170	dina.30.08@mail.ru	dina.30.08@mail.ru	DINA.30.08@MAIL.RU	0	AQAAAAEAAcQA/
7	d28905b0-003c-4e02-95e8-6ce10c48db8f	ADMIN@ADMIN.COM	admin@admin.com	ADMIN@ADMIN.COM	0	AQAAAAEAAcQA/

3.10 Сурет– “AspNetUsers” кестесі

Тіркелген қолданушыға “Анализдер” беті және дәрігердің қабылдауына онлайн жазылу ұсыналады.

3.10 Сурет “Пациент” қолданушысының навигациялық панелі

Біз авторизациялан қолданушы болған соң жазылым формасында бірқатар жолдар біз тіркелу кезінде енгізген мәліметтермен автоматты түрде толтырылады.

ЗАПИСЬ К ВРАЧУ

Вы можете указать удобный для вас промежуток. Наши специалисты подберут для вас подходящее время.

Нурғалиқызы Динара	dina.30.08@mail.ru	87758894976
01.01.0001	Отделение	Доктор
Message (Optional)		

Записаться

3.11 Сурет – “Жазылым терезесі”

Осы формадан біз өзімізге қажетті бөлімді, кейіннен дәрігіерлі таңдап, бос жолға бізге ыңғайлы уақытты немесе шарттарды жаза кетсек болады.

ЗАПИСЬ К ВРАЧУ

Вы можете указать удобный для вас промежуток. Наши специалисты подберут для вас подходящее время.

Нурғалиқызы Динара	dina.30.08@mail.ru	87758894976
01.01.0001	Кардиология	Абдраимова Аида
После обеда или рано утром		

Записаться

3.12 Сурет – Толықтарылған “Жазылым” терезесі

“Записаться” батырмасы арқылы сұраныс жібереміз. Қолданушы интерфейсінде жазылымның сәтті түрде өңделуге жіберілгенін хабарлайтын бет шығады.



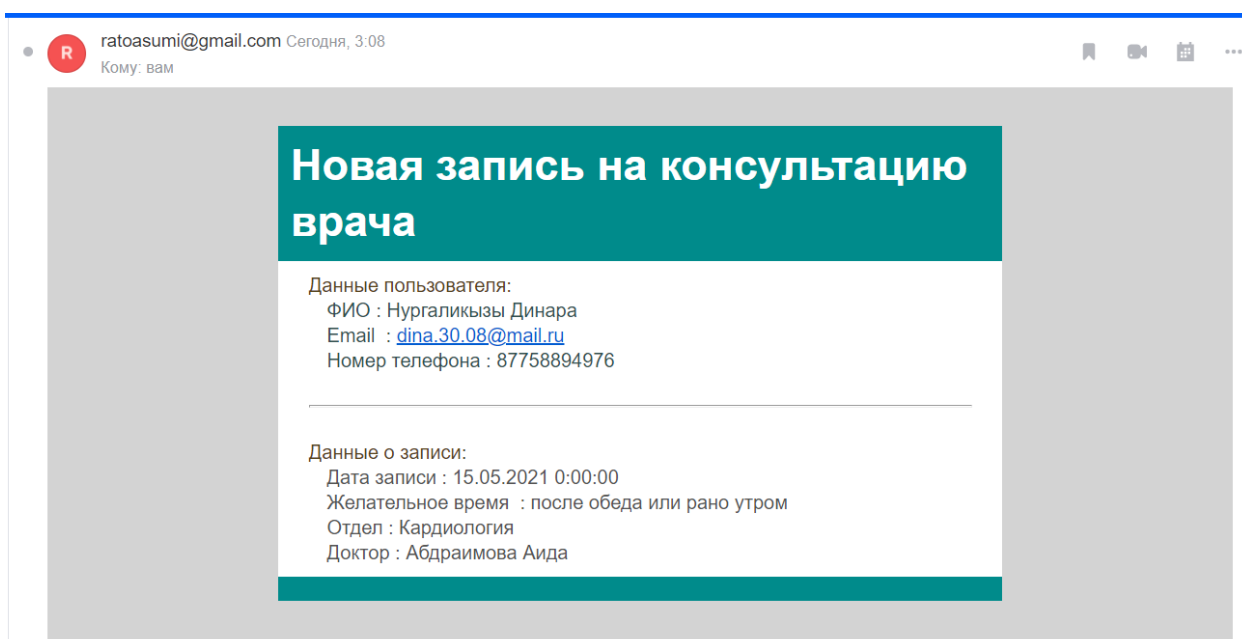
Вы были успешно записаны!

Пожалуйста, ожидайте звонка. Наши сотрудники в скором времени с вами свяжутся.

[Вернуться на главную страницу](#)

3.13 Сурет – “Жазылымның” сәтті түрде жіберілгенін растау беті

Жіберілген сұраныс аурухана қызметкерлерімен өңделеді. Келіп түскен жазылымдарды жылдам өңдеу мақсатында қызметкер почтасына жазылымға сұраныс жайлы толық ақпаратпен толықтырылған хабарлама жеткізіледі.



3.14 Сурет – Жаңа “Жазылым” жайлы хабарлайтын электронды хабарлама

Хабарламаны алғаннан кейін, мекеме қызметкері сізге хабарласып, қабылдаудың нақты күнін белгілейді. Бұл іс-әрекеттер тізбегін орындау жүйеде “Admin” ретінде тіркелген қолданушыға қолжетімді. Осы тұрғыда веб-қосымша қандай функционалдарға ие екендігін қарастырайық.











3.15 Сурет – “Админ” қолданушысының навигациялық панелі

“Админ” панелінде жай қолданушыда болмаған функционалдар бар.

Бірінші кезекте жазылымдарды басқару бетіне көшеміз.

Записи на приём

[+ Создать новую запись](#)

Имя пациента	Номер телефона	Email	Время записи	Подтверждение	
Aisha Murtaza	87758894978	Жумалиева Бакыт	01.05.2021 0:00:00	Подтверждена	 
Nazerke Tumenbay	87471232030	Жумалиева Бакыт	01.05.2021 0:00:00	Не подтверждена	 
Kenzhe Almatov	87475896363	Жумалиева Бакыт	01.05.2021 0:00:00	Не подтверждена	 
Асель Аскаркызы	87759880288	Абдраимова Аида	01.01.0001 0:00:00	Не подтверждена	 
Нурғалиқызы Динара	87758894976	Абдраимова Аида	15.05.2021 0:00:00	Не подтверждена	 

3.1 Сурет – “Админ” қолданушысының жазылымдарды басқару беті


Егер сұраныс өңделмесе қызыл түсті ұяшықпен көрсетіледі. Жоғарыда жіберілген сұранысты растау үшін өзгерту бетіне өтеміз.

Редактировать запись

FullName

Phone

Email

RecordDate 

Message

Confirmed











[Edit](#) [Back](#)

3.17 Сурет – Жазылымдарды растау және өзгерту беті

Аурухана қызметкері сұраныс иесімен байланысып, қабылдау уақытын белгілеп жазылымды растайды.

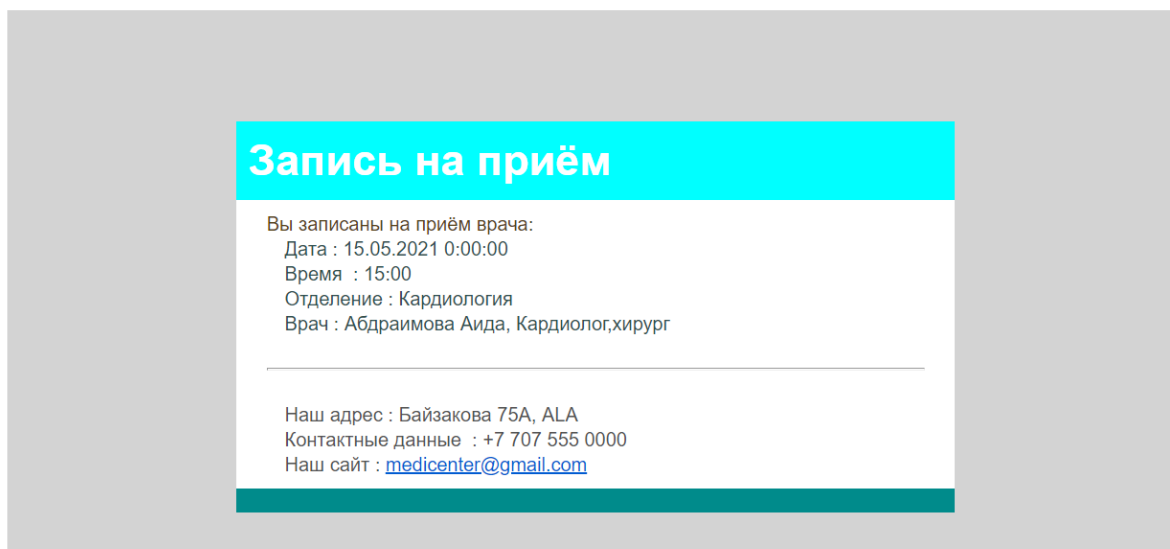
Записи на приём

[+ Создать новую запись](#)

Имя пациента	Номер телефона	Email	Время записи	Подтверждение	
Aisha Murtaza	87758894978	Жумалиева Бакыт	01.05.2021 0:00:00	Подтверждена	 
Nazerke Tumenbay	87471232030	Жумалиева Бакыт	01.05.2021 0:00:00	Не подтверждена	 
Kenzhe Almatov	87475896363	Жумалиева Бакыт	01.05.2021 0:00:00	Не подтверждена	 
Асель Аскарқызы	87759880288	Абдраимова Аида	01.01.0001 0:00:00	Не подтверждена	 
Нурғалиқызы Динара	87758894976	Абдраимова Аида	15.05.2021 0:00:00	Подтверждена	 

3.18 Сурет – “Админ” қолданушысының жазылымдарды басқару беті

Жазылым жайлы ескерту үшін сұраныс иесінің электронды почтасына хабарлама жіберіледі.



3.19 Сурет – Жазба иесін хабарлау электронды хаты

“Admin” рөліндегі қолданушы жаңа қызметті қосып, өзгертуге, өшіруге құқылы. Яғни “CRUD” операциялары арқылы басқару жүргізілінеді. Сонымен қатар ол қажет жағдайда жүйеге жаңа дәрігерді тіркей алады. Дәрігерлер мен қызметкерлер пациенттердің қабылдауы жайлы есепті, медициналық анализ нәтижелерін файл түрінде жүктесе болады.

Создать новый аккаунт.

FullName

Семёнова Алёна

Email

alena@mail.com

PhoneNumber

87073211038

Password

.....

Confirm password

.....






Регистрация пользователя доктора

3.20 Сурет – Жаңа “Дәрігер” қолданушысын тіркеу

Дәрігерлер мен қызметкерлер пациенттердің қабылдауы жайлы есепті, медициналық анализ нәтижелерін файл түрінде жүктесе болады.

Пациенты

Show entries Search:

ID	ФИО	Телефон	Email	
7	Анар Арапбай	87073211032	anar@gmail.com	 Edit
9	Асель Аскарқызы	87471455222	asel@mail.com	 Edit
11	Александр Владимирович Миронов	87065544556	aleksandr@mail.com	 Edit
12	Нурмолдаева Сауле Болатовна	87471455222	saule@mail.com	 Edit
13	Нурғалиқызы Динара	87471451220	dina.30.08@mail.ru	 Edit

3.21 Сурет – Пациенттер тізімі

Өзгерту батырмасы арқылы дәрігер пациенттің барлық талдау нәтижелері мен жазбаларына қол жеткізеді және өзі де жазба мен талдау нәтижелерін жүтей алады.

[+ Результат анализов](#) [+ История болезни](#)

Пациент

FullName	<input type="text" value="Анар Арапбай"/>
Email	<input type="text" value="anar@gmail.com"/>
Phone	<input type="text" value="87073211032"/>

Анализы

Описание	Дата	Файл
Тест на антитела	25.04.2021 0:00:00	Download
Тест на ковид	18.04.2021 0:00:00	Download
Тест на бактерии	21.03.2021 0:00:00	Download
Тест на хеликобактерии	12.03.2021 0:00:00	Download


Записи

Описание	Дата регистрации	Файл
Осмотр терапевта	25.01.2021 0:00:00	Download

3.22 Сурет – Пациенттің жеке жазбалары

Пациенттің талдау нәтижелерін жүктейік.

Загрузить результаты анализа

FullName	<input type="text" value="Анар Арапбай"/>
Name	<input type="text" value="Тест на эритроциты"/>
RecordDate	<input type="text" value="21.05.2021"/> 
File	<input type="button" value="Выбрать файлы"/> <input type="text" value="complex.png"/>
<input type="button" value="Загрузить"/> <input type="button" value="↩ Back"/>	

3.23 Сурет – Пациенттің жаңа талдау нәтижесін жүктеу

MediCenter

Наши специалисты Услуги Отделения Анализы

Hello anar@mail.com! Logout

Анализы

Описание	Дата	Файл
Тест на антитела	25.04.2021 0:00:00	Download
Тест на ковид	18.04.2021 0:00:00	Download
Тест на бактерии	21.03.2021 0:00:00	Download
Тест на хеликобактерии	12.03.2021 0:00:00	Download
Тест на эритроциты	21.05.2021 0:00:00	Download

3.24 Сурет – Пациенттің талдау нәтижелері

Талдау нәтижелерінің таңдалынған пациентке жүктелгенін тексеру үшін “Пациент” аккаунтымен жүйеге. Көріп тұрғанымыздай бәрі дұрыс жүктеліп көрсетіліп тұр. Нәтижелерді қолданушы жеке құрылғысына жүктеп ала алады.

Қорытынды

Бұл дипломдық жұмыстың мақсаты – аурухананы тиімді басқару үшін заманауи веб-қосымшаны құру және дамыту. Осы мақсатқа жету үшін пәндік салаға талдау жасалды, тақырыптың өзектілігі туралы мәселе, осындай жүйелердің дамуына көзқарас және оның ерекшеліктері зерттелді. Бағдарламаның кезең-кезеңімен орындалуы және қойылатын талаптар жүйеде егжей-тегжейлі баяндалды. Менің бағдарламалық жасақтамама веб-қосымша болғандықтан, веб-қосымшаның архитектурасы өте маңызды рөл ойнайды.

Архитектураға, клиенттің сұраныстарын өңдеуге және логикалық байланысты кері байланысқа ерекше назар аударылды. Жобаның бірдей маңызды компоненті – бұл ақпараттардың сенімді сақталуын және оған уақтылы қол жеткізуді қамтамасыз ететін мәліметтер базасын жобалау. Веб қосымшаны құрудағы негізгі компонент қазіргі талаптар мен шарттарға сәйкес келетін заманауи және қуатты даму құралы болып табылады. Осы мақсатта серверлік компоненттер ретінде ASP.NET Core веб-қосымшаларын әзірлеудің танымал платформасы таңдалды, танымал бағдарламау тілі C#, ал MVC құрылымы фреймворк болып таңдалды. Қолданушы интерфейсі үшін HTML/CSS/JavaScript қолданылды. Құралдар мен технологияларды ұтымды таңдаудың арқасында қосымшаның мүмкіндіктері мен функционалдық шекаралары нақты анықталды.

Осылайша, мақсат қосымшаны әзірлеуге, жүйені автоматтандыруға және аурухана қызметкерлерінің жұмысын жеңілдетуге, қызмет көрсетуді оңай әрі қарапайым процеске айналдыруға бағытталып, өз мақсатына жетті.

ПАЙДАЛАНЫЛГАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 <https://selleo.com/blog/how-to-build-a-hospital-management-system>
- 2 <https://searchdatamanagement.techtarget.com/definition/SQL-Server>
- 3 <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio>
- 4 Богданенко Д. А. Подходы к архитектурному проектированию веб – приложений (рус.) // Молодой ученый : статья в журнале - научная статья. — 2018. — № 9 (195). — С. 24—29.
- 5 Марко Беллиньясо. Разработка Web-приложений в среде ASP.NET 2.0: задача — проект — решение = ASP.NET 2.0 Website Programming: Problem - Design - Solution. — М.: «Диалектика», 2007. — С. 640.
- 6 Кристиан Нейгел, Билл Ивьен и др. С# 4.0 и платформа .NET 4 для профессионалов = Professional C# 4 and .NET 4. — М.: «Диалектика», 2010. — С. 1440.
- 7 Адам Фримен. ASP.NET MVC 5 с примерами на С# 5.0 для профессионалов, 5-е издание = Pro ASP.NET MVC 5. — М.: «Вильямс», 2014. — 736 с.

А ҚОСЫМШАСЫ

```
using System;
using Microsoft.EntityFrameworkCore.Migrations;
namespace HospitalSystem.Migrations
{
    public partial class AddAppointmentToDatabaseLast : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Appointments",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    FullName = table.Column<string>(type: "nvarchar(max)", nullable:
false),
                    Email = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Phone = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    RecordDate = table.Column<DateTime>(type: "Date", nullable: false),
                    Message = table.Column<string>(type: "nvarchar(max)", nullable:
true),
                    DepartmentId = table.Column<int>(type: "int", nullable: false),
                    DoctorId = table.Column<int>(type: "int", nullable: true)
                },
                constraints: table =>
                {
```

А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
table.PrimaryKey("PK_Appointments", x => x.Id);
table.ForeignKey(
    name: "FK_Appointments_Department_DepartmentId",
    column: x => x.DepartmentId,
    principalTable: "Department",
    principalColumn: "Id",
    onDelete: ReferentialAction.Cascade);
table.ForeignKey(
    name: "FK_Appointments_Doctor_DoctorId",
    column: x => x.DoctorId,
    principalTable: "Doctor",
    principalColumn: "Id",
    onDelete: ReferentialAction.Restrict);
});
migrationBuilder.CreateIndex(
    name: "IX_Appointments_DepartmentId",
    table: "Appointments",
    column: "DepartmentId");
migrationBuilder.CreateIndex(
    name: "IX_Appointments_DoctorId",
    table: "Appointments",
    column: "DoctorId");
}
```

А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "Appointments");
    migrationBuilder.CreateTable(
        name: "Appointment",
        columns: table => new
        {
            Id = table.Column<int>(type: "int", nullable: false)
                .Annotation("SqlServer:Identity", "1, 1"),
            DepartmentId = table.Column<int>(type: "int", nullable: false),
            DoctorId = table.Column<int>(type: "int", nullable: false),
            Email = table.Column<string>(type: "nvarchar(max)", nullable: false),
            FullName = table.Column<string>(type: "nvarchar(max)", nullable:
false),
            Message = table.Column<string>(type: "nvarchar(max)", nullable:
true),
            Phone = table.Column<string>(type: "nvarchar(max)", nullable: false),
            RecordDate = table.Column<DateTime>(type: "Date", nullable: false)
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_Appointment", x => x.Id);
            table.ForeignKey(
                name: "FK_Appointment_Department_DepartmentId",
```

А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
column: x => x.DepartmentId,  
principalTable: "Department",  
principalColumn: "Id",  
onDelete: ReferentialAction.Cascade);  
table.ForeignKey(  
    name: "FK_Appointment_Doctor_DoctorId",  
    column: x => x.DoctorId,  
    principalTable: "Doctor",  
    principalColumn: "Id",  
    onDelete: ReferentialAction.Cascade);  
});  
migrationBuilder.CreateIndex(  
    name: "IX_Appointment_DepartmentId",  
    table: "Appointment",  
    column: "DepartmentId");  
migrationBuilder.CreateIndex(  
    name: "IX_Appointment_DoctorId",  
    table: "Appointment",  
    column: "DoctorId");  
}  
}  
}
```

В ҚОСЫМШАСЫ

```
namespace HospitalSystem.Controllers
{
    public class AppointmentController : Controller
    {
        private readonly HospitalDbContext _db;
        public readonly IWebHostEnvironment _webHostEnvironment;
        private readonly IEmailSender _emailSender;
        public AppointmentController(HospitalDbContext db, IWebHostEnvironment
webHostEnvironment, IEmailSender emailSender)
        {
            _db = db;
            _webHostEnvironment = webHostEnvironment;
            _emailSender = emailSender;
        }
        [Authorize]
        public IActionResult Index()
        {
            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
            AppointmentVM appointmentVM = new AppointmentVM()
            {
                ApplicationUser = _db.ApplicationUser.FirstOrDefault(u => u.Id ==
claim.Value),
                Appointments = new Appointments(),
                DepartmentSelectList = _db.Department.Select(i => new SelectListItem
                {
                    Text = i.Name,
                    Value = i.Id.ToString()
                }),
                DoctorSelectList = _db.Doctor.Select(i => new SelectListItem
                {
                    Text = i.FullName,
                    Value = i.Id.ToString()
                }),
            };
            return View(appointmentVM);
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Index(AppointmentVM appointmentVM)
        {
            if (ModelState.IsValid)
```

В ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
{
    _db.Appointments.Add(appointmentVM.Appointments);
    _db.SaveChanges();
    var department = _db.Department.FirstOrDefault(u => u.Id ==
appointmentVM.Appointments.DepartmentId);
    var doctor = _db.Doctor.FirstOrDefault(u => u.Id ==
appointmentVM.Appointments.DoctorId);
    //Send email
    var PathToTemplate = _webHostEnvironment.WebRootPath +
Path.DirectorySeparatorChar.ToString()
+ "templates" + Path.DirectorySeparatorChar.ToString() +
"Inquiry.html";
    var subject = "New Inquiry";
    string HtmlBody = "";
    using (StreamReader sr = System.IO.File.OpenText(PathToTemplate))
    {
        HtmlBody = sr.ReadToEnd();
    }
    string messageBody = string.Format(HtmlBody,
appointmentVM.Appointments.FullName,
appointmentVM.Appointments.Email,
appointmentVM.Appointments.Phone,
appointmentVM.Appointments.RecordDate,
appointmentVM.Appointments.Message,
department.Name,
doctor.FullName);
    await _emailSender.SendEmailAsync(WC.EmailAdmin, subject,
messageBody);
    return RedirectToAction(nameof(Confirm));
}
appointmentVM.DepartmentSelectList = _db.Department.Select(i => new
SelectListItem
{
    Text = i.Name,
    Value = i.Id.ToString()
});
appointmentVM.DoctorSelectList = _db.Doctor.Select(i => new
SelectListItem
{
    Text = i.FullName,
    Value = i.Id.ToString()
});
return View(appointmentVM);}
```

В ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
public IActionResult Confirm()
{
    return View();
}
public IActionResult Record()
{
    var role = User.IsInRole("Doctor");
    if (role)
    {
        var claimsIdentity = (ClaimsIdentity)User.Identity;
        var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
        var doctor = _db.Doctor.Include(u =>
u.ApplicationUser).FirstOrDefault(p => p.UserId == claim.Value);
        IEnumerable<Appointments> objlist = _db.Appointments.Include(u =>
u.Doctor).Where(d => d.Doctor.Id == doctor.Id).Where(c => c.Confirmed ==
true);
        return View(objlist);
    }
    else
    {
        IEnumerable<Appointments> objlist = _db.Appointments.Include(u =>
u.Doctor);
        return View(objlist);
    }
}
//GET - Create
public IActionResult Create()
{
    return View();
}
//POST
[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Create(Appointments obj)
{
    if (ModelState.IsValid)
    {
        _db.Appointments.Add(obj);
        _db.SaveChanges();
        return RedirectToAction("Record");
    }
    return View(obj)
}
```

В ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
//GET - Edit
public IActionResult Edit(int? id)
{
    AppointmentVM appointmentVM = new AppointmentVM()
    {
        Appointments = new Appointments(),
        DoctorSelectList = _db.Doctor.Select(i => new SelectListItem
        {
            Text = i.FullName,
            Value = i.Id.ToString()
        }),
        DepartmentSelectList = _db.Department.Select(i => new SelectListItem
        {
            Text = i.Name,
            Value = i.Id.ToString()
        })
    };
    if (id == null)
    {
        //this is for create
        return View(appointmentVM);
    }
    else
    {
        appointmentVM.Appointments = _db.Appointments.Include(u =>
u.Doctor).Include(u => u.Department).FirstOrDefault(u => u.Id == id);
        if (appointmentVM.Appointments == null)
        {
            return NotFound();
        }
        return View(appointmentVM);
    }
}
//POST - Edit(Update)
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(AppointmentVM appointmentVM)
{
    if (ModelState.IsValid)
    {
        if (appointmentVM.Appointments.Id == 0)
```


В ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
{
    _db.Appointments.Add(appointmentVM.Appointments);
}
else
{
    _db.Appointments.Update(appointmentVM.Appointments);
}
_db.SaveChanges();
var department = _db.Department.FirstOrDefault(u => u.Id ==
appointmentVM.Appointments.DepartmentId);
var doctor = _db.Doctor.FirstOrDefault(u => u.Id ==
appointmentVM.Appointments.DoctorId);
if (appointmentVM.Appointments.Confirmed == true)
{
    var PathToTemplate = _webHostEnvironment.WebRootPath +
Path.DirectorySeparatorChar.ToString()
+ "templates" + Path.DirectorySeparatorChar.ToString() +
"InquiryConfirm.html";
var subject = "New Inquiry";
string HtmlBody = "";
using (StreamReader sr =
System.IO.File.OpenText(PathToTemplate))
{
    HtmlBody = sr.ReadToEnd();
}
string messageBody = string.Format(HtmlBody,
appointmentVM.Appointments.RecordDate,
appointmentVM.Appointments.Message,
department.Name,
doctor.FullName,
doctor.Speciality);
await
_emailSender.SendEmailAsync(appointmentVM.Appointments.Email, subject,
messageBody);
return RedirectToAction(nameof(Index));
}
return RedirectToAction("Index");
}
else
{
    var errors = ModelState
.Where(x => x.Value.Errors.Count > 0)
.Select(x => new { x.Key, x.Value.Errors })
}
```

В ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
        .ToArray();
    }
    appointmentVM.DoctorSelectList = _db.Doctor.Select(i => new
SelectListItem
    {
        Text = i.FullName,
        Value = i.Id.ToString()
    });
    return View(appointmentVM);
}

//GET - Delete
public IActionResult Delete(int? id)
{
    if (id == null || id == 0)
    {
        return NotFound();
    }
    var obj = _db.Appointments.Find(id);
    if (obj == null)
    {
        return NotFound();
    }
    return View(obj);
}

//POST - Delete
[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult DeletePost(int? id)
{
    var obj = _db.Appointments.Find(id);
    if (obj == null)
    {
        return NotFound();
    }
    _db.Appointments.Remove(obj);
    _db.SaveChanges();
    return RedirectToAction("Record");
}
}
```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ғылыми жетекшінің пікірі

Дипломдық жұмыс

Нұрғалиқызы Динара

5B070300 – Ақпараттық жүйелер

Тақырыбы: Пациенттер мен дәрігерлерге арналған «Аурухананы басқару жүйесі» WEB-қосымшасын әзірлеу

Бұл дипломдық жұмыс өзінің жүйелік құрылымымен ерекшеленген. Түсіндірме жобаның құрамы кіріспеден, 3 бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Менің ойымша, диплом жобалаушы алдына қойылған тапсырманы толығымен орындады және заманауи технологиялар кешенін меңгергендігін көрсетті.

Жалпы дипломдық жоба кәсіби деңгейде орындалған. Түсіндірме жазба сауатты жазылған, жоба бойынша барлық қажетті мәліметтер бар.

Кемшілік ретінде кейбір шағын стилистикалық қателерді атап кетуге болады.

Жоғарыда айтылғандарға байланысты, дипломдық жұмыс 5B070300 – «Ақпараттық жүйелер» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және дипломдық жұмыс қорғауға жіберіле алады, ал оның авторы Нұрғалиқызы Динара бакалавр академиялық дәрежесін алуға лайықты деп есептеймін.

Ғылыми жетекші

Лектор

«24» мая 2021 ж.



Зиро А.А.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Нұрғалиқызы Динара

Название: Пациенттер мен дәрігерлерге арналған «Аурухананы басқару жүйесі» WEB-қосымшасын әзірлеу

Координатор: Зиро А.А.

Коэффициент подобия 1: 0,33

Коэффициент подобия 2: 0.00

Замена букв: 3

Интервалы: 0

Микропробелы: 24

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

После анализа отчета по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, так в основном связаны с применением общеизвестных терминов.

26.05.2021

*Дата
руководителя*

.....


Подпись Научного

**Протокол анализа Отчета подобия
заведующего кафедрой / начальника
структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Нұрғалиқызы Динара

Название: Пациенттер мен дәрігерлерге арналған «Аурухананы басқару жүйесі» WEB-қосымшасын әзірлеу

Координатор: Сейлова Н.А.

Коэффициент подобия 1: 0,33

Коэффициент подобия 2: 0.00

Замена букв: 3

Интервалы: 0

Микропробелы: 24

Белые знаки: 0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.


Обоснование:

После анализа отчета по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, так в основном связаны с применением общеизвестных терминов.

.....

...
27.05.2021

Дата

.....


Подпись заведующего кафедрой

начальника структурного подразделения

