

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Әбдикадирова А.П.

Косметологияда сараптамалық жүйені қолдану

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

ПИ кафедра меңгерушісі

техника ғылымдарының  
кандидаты, доцент, ассистент-  
профессор

 Р. Юнусов  
" 14 " мел 2019ж.

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

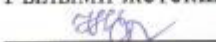
Тақырыбы: "Косметологияда сараптамалық жүйені қолдану"

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Орындаған

А.П.Әбдикадирова

Ғылыми жетекші, лектор

 I.N. Ersari  
" 13 " 05 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

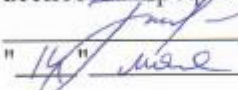
Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

**БЕКІТЕМІН**

ПИ кафедра меңгерушісі,  
техника ғылымдарының  
кандидаты, доцент,  
ассистент-профессор

 Р. Юнусов  
"14" мамыр 2019ж.

**Дипломдық жобаны орындауға**

**ТАПСЫРМА**

Білім алушыға Әбдикадирова Алтынай Пернебекқызы

Тақырыбы: Косметологияда сараптамалық жүйені қолдану

Университет ректоры бұйрығының № 1162-б "16" қазан 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі "14" мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

*a) сараптамалық жүйе көмегімен бет терсінің типін анықтау;*

*b) тұжырымдамаға сәйкес толыққанды оқытуға арналған ақпараттар қорын жинау;*

*в) бағдарлама дәлділігін дамыту, жақсарту;*

*г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;*



Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):  
15 графикалық материалдар бар.

Ұсынылған негізгі әдебиеттер: 17 әдебиеттер тізімінен

Дипломдық жобаны орындау  
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1.Кіріспе	12.12.2018	орындайды
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	15.01.2019	орындайды
3. Бағдарламалық қамтаманы жобалау.	08.02.2019	орындайды
4. Бағдарламаны әзірлеу	14.02.2019	орындайды
5.Бағдарламаның мүмкіндіктерін дамыту	12.03.2019	орындайды
6. Дипломдық жобаға түсіндірме жазба жазу	30.03.2019	орындайды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	А.Б.Рамазан Тьютор	13.05.19	
Бағдарламалық бөлім	А.М.Байгаринов Техн.ғыл.магистр, лектор	13.05.19	

Ғылыми жетекші \_\_\_\_\_  I.N.Eрсари

Тапсырманы орындауға қабылдап алған студент  А.П.Әбдикадилова

Күні

«1» күні 2018 ж.

## КІРІСПЕ

Косметологиялық орталық - клиенттерге косметологиялық қызмет көрсетуге бағытталған мекеме. Бұл нарық үлкен емес, сондықтан күн сайын жаңа салондар саны өсіп келеді.

Клиенттерге қызмет көрсету сапасын жоғарылату және процестер мен тапсырмаларды автоматтандыру үшін, орталықтар ақпараттық технологияларды қолдануға тырысады. Қызмет көрсету секторындағы ақпараттық технологияларды пайдалану заманауи орталықтардың жұмысын жоғары деңгейде қамтамасыз етеді. Тұтынушы кезекте тұрмайды, жүйеге кіріп, қажетті қызметті іздейді, ол туралы кең көлемде ақпарат ала алады және қызметтерге тіркеледі, сонымен қатар алдын-ала тегін сараптамалық тесттен өтіп, өз терісіне қандай қызметтер мен процедуралар алу керек екендігі туралы ақпарат алады.

Осы дипломдық жұмыстың мақсаты косметологиялық орталық үшін сараптамалық жүйесі бар бағдарламалық қамтама құру болып табылады.

Бүгінгі күні жоба менеджменті маңызды болып келеді. Көптеген кәсіпорындар өз қызметінде жобаларды басқару принциптерін қолданады. Бұл қызығушылық басшылар мен менеджерлер қолданатын әдістер мен құралдардың көп қырлы және жобаның кез келген түріне қолайлы екендігіне байланысты. Инновация саласындағы жобалардың күрделілігі күрделі жобаны жүзеге асыру ұзақ уақытты қажет етеді және жиі және кейде маңызды өзгерістерді талап етеді. Жобаның неғұрлым күрделі болғаны соншалық, ол тәуекелге ұшырайды. Сонымен қатар жыл сайын жобалардың саны артып келе жатқандығын атап өту керек, бұл бәсекелестіктің өсуіне алып келеді. Тапсырыс беруші барлық жобаларды уақытында, тиімді және ең төменгі құны бойынша аяқтауды қажет етеді. Тапсырыс берушінің барлық талаптарын орындау үшін жоба менеджері есептеулерді уақытты модельдеуге, жоспарлауға және қысқартуға мүмкіндік беретін арнайы бағдарламаларды пайдаланады.

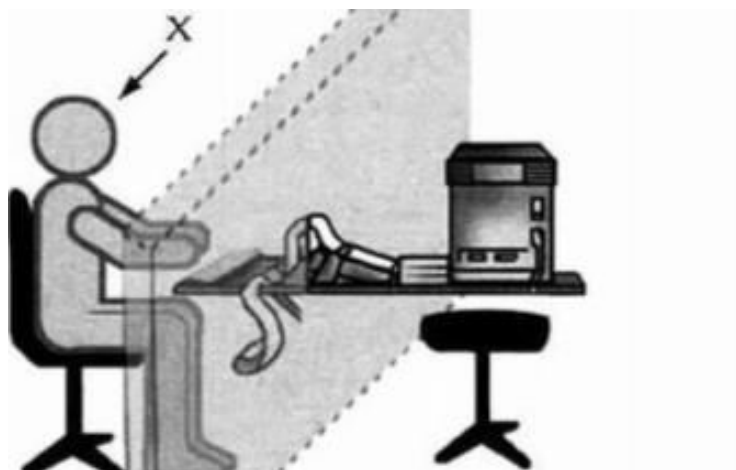
# 1 Жасанды интеллект түсінігі және сараптау жүйесі

## 1.1 Жасанды интеллект түсінігі

Жақында «жасанды интеллект» термині жиі мақалаларда, басылымдарда және кітаптарда кездеседі. Бастапқыда жасанды интеллект адам ретінде ойлауға қабілетті және адамның миымен бәсекелесуге қабілетті машина жасауды білдіреді.

Өткен ғасырдың 80-ші жылдарының басында екі үлкен ғалым Барр мен Фейгенбум жасанды интеллект туралы түсінік берді. «Жасанды интеллект - интеллектуалдық компьютерлік жүйелерді дамытатын информатика саласы, яғни біз дәстүрлі түрде адамның ақыл-ойымен, тілдік түсініктерімен, білімімен, ойлау қабілетімен, проблемаларды шешумен және т.б. мүмкіндіктермен байланыстыратын жүйелер» .

Жасанды интеллекттің негізін қалаушы ағылшын математикесі Алан Тьюринг оның анықтамасын ұсынды. 1950 жылы ол жасанды интеллектті анықтауға арналған тест ұсынды, оның идеясы өте қарапайым: компьютер мен адам түрлі бөлмелерде орналасқаны 1.1-суретте келтірілген.. Адам автокөліктің сұрақтарын сұрайды, және ол өз кезегінде жауап береді. Егер эксперимент барысында адам басқа адаммен сөйлесіп жатқанын ойлауды жалғастырса, бұл машинада ақыл бар .



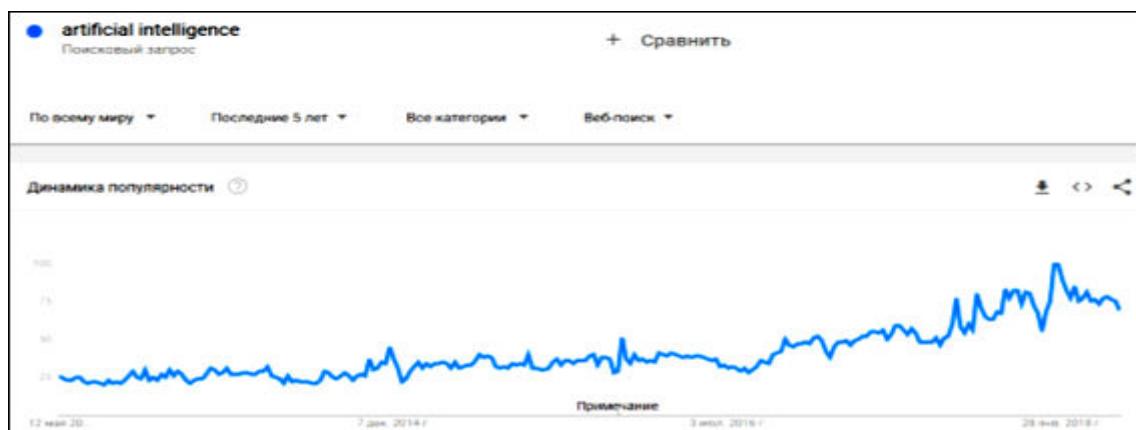
1.1-сурет – Тьюринг тесті

Жасанды интеллекттің жасаған сұрағына ешқандай жауап жоқ. Жасанды интеллект туралы кітабын жазған дерлік әрбір авторға барлық жазбаша бағдарламалардың тек адамның ақылына жақындауға мүмкіндік беретінін түсінетін кез келген анықтамадан қайтады. Адам ретінде жұмыс істеген, ойланып, шешім қабылдаған бағдарлама әлі де жоқ. Көптеген бағдарламалар мен алгоритмдер жазылмаған және адамның ақыл-ойына ұқсамайтын жазылған. Бірақ бағдарламаны табиғи жағдайда қалдырсаңыз, ол бірден

кателіктермен жұмыс істей бастайды.

Жүйе бастапқы деректерге байланысты мақсатты болған кезде, зияткерлік болып табылады, жұмыс істеу параметрлерін және оның мінез-құлқының жолын өзгертеді.

Бұл жағдайда мінез-құлық ағымдағы жағдайдан ғана емес, бұрынғы жүйе күйлерінен де өзгереді. Бұл тақырыптың қызықты және тартымды екенін түсіну үшін Google Trends іздеу жүйесінде зерттеу жүргізілді. Соңғы 5 жылда «жасанды интеллект» (жасанды интеллект) сұраныстың динамикасы 2 есе өсті. Динамикасы 1.2-суретте келтірілген.



**1.2-сурет – Соңғы 5 жылдағы «жасанды интеллект» сұранысының динамикасы**

Қазіргі заманғы әлемде математикалық алгоритмдер және бағдарламалық қамтамасыз ету, олар өздерінің шешімдерін бейнелейтін адам секілді проблемаларды шеше алады, жасанды интеллектке айнала бастады.

Жасанды интеллект зерттеу тақырыбы нашар ресімделген проблемаларды шешу, жауап іздеу, үлкен көлемдегі деректерді өңдеу болып табылады. Жасанды интеллект Сараптау жүйелер, шешімдерді қолдау жүйесі, оқыту жүйесі, сөйлеуді тану жүйесі, паттернді тану және т.б. жүйелерде белсенді қолданылады.

Басқару саласындағы міндеттердің пайда болуы жасанды интеллектпен байланысты нақты әдістерді әзірлеуге әкелді. Осылайша, интеллектуалды басқару жүйесі «дәстүрлі» интеллектуалды жүйемен ерекшеленеді. Негізгі айырмашылық - сыртқы мазасыздықтардың болуы, яғни біз мақсатымызды өзгерту, жобаның параметрлерін өзгерту, ресурстарды өзгерту және т.б.

Жасанды интеллекттің маңызды қасиеттерін атап өту керек: тіл түсіну, оқуға қабілеттілік, «ойлау» қабілеті және шешім қабылдау. Жасанды интеллекттің міндеттері есептеу функциясында ғана емес, сонымен бірге проблеманы шешуге арналған әдісті табуда .

Жасанды интеллекттің негізгі бағыттарының бірі сараптамалық жүйелер болып табылады .

## 1.2 Сараптау жүйесі

«Сараптау жүйелер» бағыты өткен ғасырдың 80-жылдарында қалыптасты. Сараптамалық жүйені зерделеу адамның Сараптаушының жұмысының тиімділігі мен сапасы жағынан кем түспейтін күрделі мәселелерді шешуге байланысты.

«Сараптау жүйесі» деген сөзде сараптау болып табылатын жүйе бар деген мағынасы бар. Сонда келесі сұрақ туындайды: қандай жағдайда компьютерлік жүйені Сараптау деп атауға болады?

Адами тұрғыда Сараптау мамандандырылған білімі бар адам. Тиісінше, сіз аласыз жүйе контекстінде Сараптау білімі бар бағдарлама болады деп есептейік. Кейбір алгоритмді орындаудың әдеттегі қабілеті осы талапқа сай келмейді. Жүйенің белгілі бір тақырыптық аймаққа бағытталуы керек екенін білу керек. Барлық ақпарат бір-біріне қатысты болуы керек, ол сөздер мен әрекеттердің қарапайым жиынтығы болмауы керек. Ақпарат бір тізбектің келесіге жабысатын тізбекті қалыптастыруы керек. Білім мәселені шешуге көмектесуі керек.

Сараптамалық жүйенің қызметі деректерге, ережелерге, ақпаратқа сипатталған жағдайға байланысты ұсыныстарды қалыптастыру болып табылады. Сараптаулардың жүйелері қажетті ақпаратты сұратуға мүмкіндік беретін зияткерлік дерекқорға ие. Мұндай ақпарат анық сақталмайды, бірақ талдау негізінде білім базасынан алынады. Бұл жағдайда ақпарат іздеуде көрсетілген шарттар ұзартылуы тиіс. Мысал «Баға салалық орташа мәннен жоғары элементтердің тізімін көрсету» сұрауы. Бұл сұраққа жауап алу үшін сараптама жүйесі барлық тауарлардың құнын талдап, саланың орташа құнын есептеп, таңдау жасайды. Нақты жағдайда, индустрияның орташа құнын есептеу шартты қосымша анықтау болып табылады. Сараптама жүйесі машина деңгейіне табиғи тіл аударатындығын атап өту керек. Бұл үшін жүйе синтаксистік, морфологиялық және семантикалық талдаудың мәселелерін шеше алады. Морфологиялық талдау жүйенің сөздіктерінде сақталған сөздердің емлесін тексеруге мүмкіндік береді. Синтаксистік талдау алынған бөлімдерді бөлек компоненттерге бөлуге мүмкіндік береді. Семантикалық талдау семантикалық жүктемені анықтайды және оны синтаксистік құрылымдармен салыстырады. Мәселен, сараптама жүйесі - бұл мамандардың білімдерін жинақтайтын күрделі компьютерлік бағдарламабелгілі бір пәндік облыстар, табиғи тілді машина деңгейіне аударады, талдайды, шарттарды анықтайды және нәтижелерді көрсетеді.

Сараптамалық жүйенің артықшылығы - бұл бірегей жағдайларда шешімдер қабылдау мүмкіндігі. проблеманы шешу алгоритмі алдын ала белгілі емес жағдайларда. Бұл жағдайда мәселені шешу бастапқы ақпараттың көп міндетті, толық емес және дұрыс еместігі жағдайында жүзеге асырылады. Сараптамалық жүйе кәсіби емес пайдаланушылар үшін көмекші кеңесші рөлін орындай алады, шешімдерді талдау үшін ассистент көмекшісі, тиісті қызмет

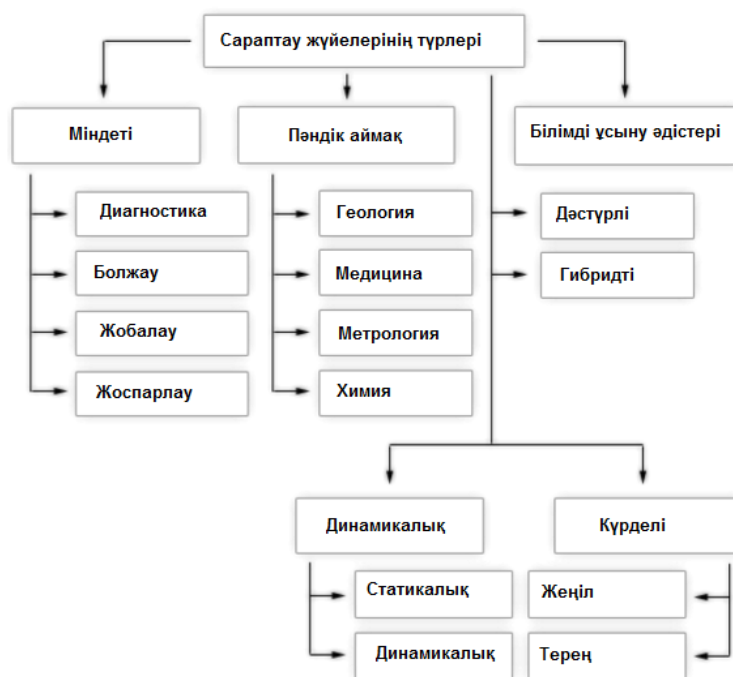


саласынан мәселелерді шешу үшін Сараптауның көмекшісі болады.

### 1.2.1 Сараптамалық жүйелерді қолдану аясы

Эксперттік жүйелердің жалпы қабылданған жіктемесі бар. Диаграмма эксперттік жүйелердің негізгі мақсаты: диагностика, болжау, жобалау және жоспарлау болып табылады. Егер өтінім бойынша жіктелуді қарастырсақ, онда Сараптаулар жүйелері геология, медицина, оқыту, метрология, химия және әскери мәселелерде белсенді қолданылады.

Сараптамалық жүйенің негізгі мақсаты жобаны басқару саласындағы жұмыстың ерекшелігін толық және нақты көрсетеді. Дегенмен, жобаны іске асырумен байланысты пәндік сала осы жіктеуде жоқ. Бұл дегеніміз, осы саладағы шектеулі сандарда Сараптау жүйелер пайдаланылады немесе мүлдем жоқ.



### 1.3-сурет – Сараптамалық жүйелерді жіктеу

Диагностика - бұл жағдайға әкелетін себептерді анықтау процесі. Ақпараттық технологиялардың өсуі медицина саласында байқалады, өйткені бұл сала адам денсаулығына жауапты. Медициналық диагностика белсенді түрде дамып келеді, медициналық сараптамалық жүйелерді қолданып, дәрігерге өз диагностикалық жорамалдарын тексеріп қана қоймай, сонымен қатар жүйеге көмек және кеңестер, онда үлкен көлемді деректерді өңдеу керек (пациенттің белгілері). Дәрігерлердің шешім қабылдаудағы проблемалары

жобаларды басқару саласындағы проблемалар - білімнің жетіспеушілігі, уақыт шектеулері, толық емес ақпарат .

Медициналық диагноз органның бұзылу себептерін анықтау үшін қолданылады. Ең танымал диагностикалық жүйе - MYCIN жүйесі. Бұл жүйе 1970-жылдардың ортасында Стэнфорд университетінде жасалды. Қазіргі уақытта MYCIN жүйесі аурулардың көпшілігін анықтайды .

Жоспарлау - бұл жобаның маңызды кезеңі. Бұл кезеңде жоба менеджері ресурстың түрлерін және олардың санын, ақшалай шығындарын, уақытша сипаттамаларын, топ мүшелерін және жобалық тәуекелдерді анықтайды. Жоспарлау кезеңі - ең күрделі, өйткені бастапқы деректерді дұрыс анықтау жобаға үлкен артықшылық береді.

Жобаны жоспарлау бірыңғай тұжырымдаманы құруды, шешім әдісін таңдауды, әрбір жұмыс кезеңінің дамуын және әр кезеңде ресурстарды бөлуді қамтиды.

Қазіргі уақытта сараптамалық жүйелер маршруттарды жоспарлауда белсенді қолданылады. Эксперименттік жүйелер көліктік және маршруттық базаға негізделген қажетті маршрутты анықтау үшін логистикалық аймақпен біріктіріледі. Сонымен қатар, MOLGEN деп аталатын жүйені бөліп алу қажет. Оның мақсаты молекулалық генетика бойынша эксперименттерді жоспарлау. Бұл жүйе көп деңгейлі және өз саласында белсенді қолданылады .

Болжаумен айналысатын жүйелер мүмкін болатын нәтижелерді және объектінің бастапқы деректеріне негізделген оқиғаларды көрсетеді. Сараптау жүйелер болжау функцияларын керемет орындайды. Мұндай жүйенің мысалы - бұл геологиялық ортаның ластану салдарын болжауға мүмкіндік беретін жүйе. Ол жер қабатының қаншалықты зақымдалғанын анықтауға мүмкіндік беретін топыраққа мұнай өнімдерінің ену тереңдігін анықтайды. Сондай-ақ, бұл бірегей жүйе мұнайдың жер асты суларына жету уақытын есептей алады .

Сараптаулар жүйелері күрделі мәселелерді шешеді және ықтимал нәтижелер мен оқиғаларды болжай алады. Олар сондай-ақ ауа райы болжамы үшін қолданылады.

Әдетте, сараптамалық жүйенің білім жүйесі фактілерді (деректер, синтаксистік деңгейдің қатынастары), білімді (семантикалық, деректер арасындағы семантикалық қатынастар) және ережелерді - белгілі фактілер мен мағынада қолдануға болатын нұсқаулар жиынтығын қамтиды.

Логикалық деректер базасының үлгісі мен білім қорының құрамында олар логикалық дереккөздермен, мысалы, Prolog немесе Smalltalk сияқты компьютерлік тілде жазылған, мысалы, тұжырымдаманы анықтау ережелерін білдіретін фактілерді және шығарылған ережелерді сипаттау үшін негізгі тілді, жалпыланған және нақты ақпарат, сондай-ақ дерекқорлар мен білім негіздеріне нақты және жалпылама сұраулар.

Пролог тіліндегі білім негіздеріне қатысты нақты және жалпыланған сұраулар таңдалған тақырыптық аймақта және білімде жалпыланған және нақты ақпарат пен білімді білдіретін білімдер базасында қолжетімді нәтиже ережелерін анықтайтын және түсіндірме ережелерін білдіретін негізгі тілде

жазылады.

Әдетте білім базасындағы фактілер берілген тақырыптық аймақ үшін тұрақты болып келетін құбылыстарды сипаттайды. Міндеттері белгілі бір тапсырмаға байланысты болған сипаттамалар, СЖ жұмыс барысында пайдаланушыдан алады және оларды жұмыс моделіне сақтайды. Мысалға, медициналық СЖ-де «Сау адамның екі аяғы бар» фактісі білім базасында сақталады, ал «Пациенттің бір аяғы бар» фактісі нақты науқастың жұмыс үлгісінде.

### **1.3 Қорытындылар**

Кез-келген интеллектуалды жүйе пайдаланушы енгізген ақпараттық сұрауларды қабылдап, алынған сұрауларды өңдеуге және нәтиже жасай алады. Ақпараттық жүйеден интеллектуалдық жүйенің айрықша ерекшелігі қиын форматталған проблемаларды шешуге мүмкіндік береді. Бұл қабілет Сараптау жүйелерге ие, бұл зияткерлік жүйелердің ерекше жағдайы. Сараптамалық жүйе сараптамалық білімді компьютердің мүмкіндіктерімен біріктіреді. Ол тапсырманы шешуге көмектесе алады. Жасанды интеллект саласындағы Сараптау жүйелер ең танымал. Сараптық жүйелер тиісті тақырыптық салада күрделі мәселелерді шешуге көмектеседі. Мұндай есептеу жүйесі жүйенің дұрыс жауап беруге және ақылға қонымды шешім қабылдауға мүмкіндік беретін мамандардың білімдерін қамтиды.

Сараптаулар жүйелері - олардың түсіндірулерінің жүйелілігін түсіндіре алатын, сонымен қатар олардың білімдерін арттыруға мүмкіндік беретін тар бағдарламалар.

Сараптамалық жүйелерді пайдалану салаларын талдап шыққаннан кейін, сараптамалық жүйелерді жобаны басқаруда шешім қабылдау жөніндегі көмекші ретінде қолдануға болады деп қорытынды жасауға болады. Олар деректерді талдайды және оларды өңдей алады және салыстыра алады. Мұндай жүйелер атом электр станцияларында, авиацияда, медицинада, қаржыландыруда және басқа да салаларда жұмыс істейді, онда критикалық жағдайлардағы шешімді дамытуда көмек қажет.

Жұмыс кезінде консультант (клиенттерді қабылдау, пайдалану, сұлулық нұсқаулығын және т.б.) тестен алынған мәліметтерге, сұрақтарға жауаптарға, зертханалық сынақтарға, қысымға, импульстікке, температураға негізделген нақты әрекеттерге алып келетін нақты қорытынды жасауға тиіс және басқа көрсеткіштер қарастырады.

Сонымен қатар, ол тәжірибеден алынған, арнайы әдебиеттерді оқыған, өзінің тәжірибесі, түйсігі және әріптестерінен келген кеңестерді басшылыққа алады. Шешім қабылдауды жеңілдету үшін, компьютердің сараптамалық кометологиялық жүйесін бұрын жасалған ақпарат негізінде клиентке сұрақтардың нұсқаларын ұсынатын интеллектуалдық анықтамалық ретінде

жұмыс істейтін түрі кеңінен таралған. Енгізілген ақпарат мәтіндік ғана емес, сонымен бірге әртүрлі графика болуы мүмкін. Жүйе дерекқорда сақталған клиенттің космологиялық тарихын ескере отырып, барлық параметрлерді талдайды және клиенткебәт әлпетке байланысты нақты ем-шаралар және ықтимал әрекеттерді ұсынады. Дегенмен, бұл бағдарламаны әзірлеу кезінде құрылған білікті мамандардың білім базасына негізделген. Сондықтан мұндай жүйені «анықтамалық-кеңесші» деп атау дұрысырақ. Әрине, түпкілікті шешім космолог тарапынан жасалады.

Қолдану саласы өте жауапты болғандықтан, қолданыстағы сарапшылар жүйелері қате әрекеттердің ықтималдығын азайтатын күрделі алгоритмдерді пайдаланады, сондықтан мұнда әмбебаптық пен көпесептікке сүйену қиындау болады. Кәсіби Косметологиялық пакеттер өте мамандандырылған. Мысалы, шұғыл диагностика және шұғыл емдеу бағдарламалары, тері диагностикасы, әртүрлі асқынуларының алдын алу, анестезиология және т.б.

Сонымен қатар, компьютерлік жүйе, сондай-ақ клиенттерді тану теориясына негізделген және интеллектуалды (көбінесе сарапшылардың пікірінше) Косметологиялық жүйе оларды құру кезінде келесі критерийлерге негізделуі керек:

Компьютерлік диагностикалық жүйелердің медициналық фокусы міндеттерді және келесі пайдалану деңгейлерімен анықталады:

1) дифференциалды сериясын құрумен зерттеудің алдын ала зертханалық кезеңінде алдын-ала диагностика жасау;

2) патологияның соңғы нозологиялық сәйкестендіруін оңтайлы зерттеу әдісін таңдау арқылы диагностикалау;

3) бар ақпаратқа (клиникалық және параклиникалық) негізделген нақты диагнозды қалыптастыру (негіздеу) көмегімен нозологиялық диагностика;

4) қауіптілік жағдайында шешім қабылдау (көмек көрсету деңгейін басқару) үшін негіз ретінде жағдайдың ауырлығын бағалау арқылы индикативті диагностика.

Автоматтандырылған жүйені құру әдістемесін таңдауға қатысты жасанды интеллект теориясына негізделген сараптамалық жүйелердің (ЭС) класы үлкен қызығушылық тудырады. Бұл жүйенің белгілі бір проблемалық аймақта білім базасын қамтитынын және хаттамамен Косметологиялық жағдайды тану нәтижелерімен және жүйе ұсынған шешімнің логикалық түсіндірілуімен танысу мүмкіндігін береді.

Косметологияда ең қызықты (олардың барлығы Ресей мен ТМД-да әр түрлі мекемелерде жұмыс істейтін) жеке жүйелерді қарастырсақ. Төменде біз Ресейдің медициналық ғылыми мектептерінің ерекшеліктерін ескере отырып, біз үшін табиғи болып табылатын жүйені енгізуді ескере отырып, негізінен осы саладағы ішкі оқиғаларға назар аударамыз.

Мысал ретінде, Мәскеудің педиатрия және балалар хирургиясы ғылыми-зерттеу институтында құрылған «DIN» балалардағы төтенше жағдайлар синдромын диагностикалаудың автоматтандырылған жүйесімен қарастырайық.

Бұл ЕС диагностикалық гипотезалардың тізімі болып табылатын 42 синдром туралы ақпаратты қамтиды. Емдеуді таңдау ықтимал асқынулардың болжауымен анықталғандықтан, жүйе себеп-салдарлық, уақытша және ассоциациялық қатынастармен анықталған синдромдардың өзара байланысын сипаттайды. Себептік қарым-қатынастар осы синдромның себебі болуы мүмкін синдромдар туралы немесе, керісінше, оның салдары болуы мүмкін екендігін көрсетеді. Соңғы жағдайда балада байқалатын синдромның туындауы мүмкін асқынуларды болжау туралы мәселе. Уақытша қосылулар қазіргі уақытта болған күйге әкелуі мүмкін алдыңғы синдромдар туралы ақпаратты қалпына келтіруге мүмкіндік береді, бұл әсіресе анамнезсіз бала реанимация бөліміне кірген жағдайларда маңызды.

Дәрігер балада қазіргі уақытта қол жетімді жетекші синдром немесе синдром туралы (аурудың дәрежесін немесе кезеңін көрсететін) туралы ақпаратты алады, ал егер қаласа, диагностика процесінде жүйе қарастыратын гипотетикалық жағдайлардың толық тізімі.

Шұғыл диагноз қою және қарқынды терапия мәселелеріне бағытталған жүйелер мүгедектік пен балалар өлімінің төмендеуімен тікелей байланысты болып келетін күрделі жағдайларда емдеудің тиімділігін арттыру үшін жаңа мүмкіндіктер туғызады.

Диагностика үшін сөзсіз қызығушылық, балаларда жедел қан айналымы бұзылуының емдеу жіктеу және түзету бағдарламасы кешені «О, бұл Hurts қалай», жүрек-тамыр хирургиясы орталығы Бакулева құрылды және белсенді хирургия және реанимация бөлімшесіне емдеу операциядан кейінгі таңдау кезінде пайдаланылады. Ол математикалық үлгілерді, имитациялық әдістерді қолданады. Сипаттардың алғашқы бағалауы қан айналымы бұзылыстарын бақылауды бақылау процесінде алынған пациенттердің деректерін өңдеу негізінде жасалады. Жүйе «доктор-компьютер» диалогында нақты уақыт режимінде жұмыс істейді, өмірлік маңызды жүйелердің жай-күйін бақылайтын медициналық жабдықтан алынған деректерді өңдейді.

Ақпараттың өсуі, соңғы диагностикалық және емдік технологиялардың пайда болуы медициналық деректерді өңдеу мен түсіндіруге, білікті мамандардың тәжірибесін жинақтау, сақтау және пайдалану мүмкіндігіне түбегейлі әртүрлі көзқарасты тудырды. Медициналық зерттеулер үшін материалды өңдеу үшін статистикалық әдістердің стандартты жиынтықтарын пайдалану жеткіліксіз. Биология мен медицинаның барлық салаларына қолданылатын, ақпараттық сыйымдылығы жоғары және шуылға қарсы иммунитеті бар тапсырмаларды икемді тұжырымдау мүмкіндігі бар әмбебап құралдарды қажет ететін ұзақ уақыт қажет емес. Барлық жоғарыда көрсетілген талаптарды проблемаларды шешу үшін өздігінен реттеуге қабілетті жүйелермен қанағаттандыруға болады. Бастапқыда бірдей ғана құрылған, олар әртүрлі мәселелерді шешуге үйренулері керек.

Сараптамалық жүйелердің жаңа класын құру мүмкіндігі жеке қолданушыға, мысалы, сараптамалық жүйені өзінің тәжірибесі бойынша және осы дәрігерге қол жетімді деректер негізінде оқытуға мүмкіндік беретін нақты

дәрігерге дейін кеңейтілуі керек. Бұл дегеніміз, мұндай жүйелердің өндіріс технологиясында түбегейлі өзгеріс қажет. Түпнұсқалық шешімдер қабылдаудың нақты жүйелері негізінде, әдетте, мамандар тобы, оның ішінде математиктер, бағдарламашылар және тапсырмаларды орындаушы пән мамандары құрылды.

Осындай жүйелерді соңғы пайдаланушыға теңестіру қабілеті әрдайым әрдайым азайып кетеді және осындай жүйені сатып алған кезде ол көбінесе нақты жұмыс жағдайларына (мысалы, осы клиникада пайдаланылатын зертханалық сынақтардың немесе сынақ әдістерінің әртүрлі ауқымы) қолданылмайды. Шешім - маманға нақты жағдайлар, тәжірибе және әріптестердің тәжірибесі негізінде сараптама жүйесін құруға мүмкіндік беру. Мұндай дизайн тек қана компьютерде әдеттегі дағдыларды талап ететін математикалық аппараттың пән маманын білместен жүзеге асырылуға тиіс. Сонымен қатар, осы жағдайда, жұмыс істейтін сараптамалық жүйенің қорытындыларына сенімділік мәселесі оны құрастырған маманға, оның сенім білдірген әріптестеріне, сондай-ақ ол бақылаулардың нәтижесінде алынған нақты деректерге негізделген тәжірибесі мен білімдеріне негізделеді.

Жасанды нейрондық желілерге негізделген нейро-ақпараттық технологиялар бұл талаптарды өте жақсы қанағаттандырады. Нейрондық желілердің жұмыс істеуі жүйке жүйесінің нейрондық және синапстарында сигналдардың таралуын имитациялайтын алгоритмдерге негізделген.

Нейрондық желілердің жұмыс істеуі үшін көптеген сәулет және мета-алгоритмдер жиынтығы бар, ал қазіргі заманғы нейроинформатика арқылы шешілетін мәселелер көбінесе нейрондық желілерді оқытудың архитектурасы мен алгоритмдерін нақты тапсырмалар класына немесе белгілі бір тапсырмаға сәйкестендіруді талап етеді. Сондықтан медицина сараптамалық жүйелерін құрудың әмбебап технологиясын жасау, соның ішінде нейрондық желілердің жұмыс істеуі үшін архитектураларды және мета-алгоритмдерді оңтайландыру, биомедициналық ақпаратпен жұмыс істеу кезінде нейрондық желілердің жұмыс істеуінің теориялық және әдіснамалық негіздері маңызды міндет болып табылады.

Нейро-ақпараттық технология тұрғысынан адамның барлық тапсырмаларын екі топқа бөлуге болады:

1) белгілі және анықталған алгоритмге нақты, анық, анық, нақты жауап алу қажет болатын шарттар белгілі және анықталған жиынтығы бар тапсырмалар;

2) жауаптың барлық нақты жағдайларын ескеру мүмкін емес міндеттер, бірақ сіз ең маңызды шарттардың шамамен жиынтығын ғана таңдай аласыз (кейбір шарттар есепке алынбайды, жауап дұрыс емес, жақын және алгоритм) жауапты дәл табу мүмкін емес).

Бірінші топтың үлкен жетістіктері бар проблемаларын шешу үшін сіз дәстүрлі компьютерлік бағдарламаларды пайдалана аласыз. Алгоритмнің қаншалықты қиын болғанына қарамастан, көптеген шарттардың (кіріс параметрлері) шектеулілігі осы мәселені шешетін белгілі бір бағдарламаны

шешу және жазу үшін алгоритм құру мүмкіндігін береді. Себебі, мұндай проблемаларды шешу үшін нейро-ақпараттық технологияларды қолданудың ешқандай мәні жоқ. Бұл жағдайда нейрондық желілік әдістер осындай проблемаларды шешу үшін априорндық болады. Жауапты есептеудің алгоритмі тым үлкен және ауыр болып шықса, және осы алгоритм бойынша белгілі бір тапсырманы шешу уақыты практикалық талаптарға сәйкес келмейді; Сонымен қатар, жауап алу кезінде абсолютті дәлдік талап етілмейді.

Екінші топтың проблемаларын шешу кезінде нейротехнологияны пайдалану барлық параметрлерде өзін ақтайды, алайда, егер екі шарт орындалса: біріншіден, әмбебап типтегі архитектураның болуы және бірыңғай әмбебап оқыту алгоритмі (оларды әрбір тапсырма түрі үшін дамытудың қажеті жоқ), екіншіден, нейрондық желілерді оқыту негізінде мысалдардың болуы (фон, стационарлық тәжірибе). Бұл шарттар орындалған кезде сарапшылар жүйесін құру жылдамдығы он есе артады және, тиісінше, олардың құны азаяды.

Іс жүзінде барлық медициналық және биологиялық ғылымдар екінші топқа жататын міндеттерден тұрады және осы міндеттердің көпшілігінде екінші шартты орындау үшін қажетті санын жинау жеткілікті. Бұл диагностиканың, дифференциалды диагностиканың, болжаудың, стратегияны таңдаудың және емдеу тактикасының проблемалары болып табылады. Медициналық тапсырмалар әрдайым бірнеше шешімдерге ие және нейрондық желілер арқылы нәтиже шығару әдісімен сәйкес келетін жауаптың «анық емес» сипаты бар.

Жіктеу міндеттері - биомедициналық тапсырмалардың негізгі және өте кең тобы. Оларға жауап - сынып - бұрынғы белгілі нұсқалар жиынтығынан бір нұсқаны таңдау. Жіктеу бинарлық (қарапайым жіктеу) болуы мүмкін - бұл жағдайда ықтимал жауаптардың жиынтығы екі нұсқадан (сыныптардан) тұрады - және  $n$ -ағу, онда сынып саны екіден көп. Бинарлы классификацияның мысалдары объективті санаттар (адамның жынысы еркек немесе әйел, ісіктің табиғаты жақсы немесе қатерлі) және субъективті (сау адам науқас немесе науқас, қатерлі ауруларға бейімділік бар немесе болмауы) болуы мүмкін. Кейбір жағдайларда міндеттің жауабын объективті немесе субъективті категорияға жіктеу мүмкін емес, бұл жүйке жүйесіндегі сарапшылар жүйесін оқыту және жұмыс істеуде түбегейлі маңызды емес.

Классификациялаудың маңызды ерекшелігі, анықтамалар бойынша, бір және бір ғана шешімді (класс) таңдау мүмкіндігін береді, сондықтан диагноз бір жіктеу міндеті ретінде қарастырыла алмайды. Бір адамда бірнеше патология болуы мүмкін. Егер бір жауапты таңдау мүмкін болмаса (таңдаудың көптігі), тапсырма әрқайсысы жіктеу міндеті болып табылатын қосалқы тапсырмаларға бөлінеді.

Нейрологиялық желілердің тағы бір түрі - болжамды немесе болжау проблемасы. Олар санның болжауына (бір өлшемді болжау) және векторға бөлінеді (векторлық болжау, жалпы жағдайда). Жіктелу проблемаларының айырмашылығы, болжау проблемаларындағы жауаптар бөлшек болуы мүмкін

және кез-келген аралығындағы кез келген мәндерді қабылдауы мүмкін.

Векторлық болжау жауаптың бір-бірінен тәуелсіз бірнеше сан ретінде ұсынылуы мүмкін, яғни өлшемі болжанған сандар санына тең көп өлшемді кеңістікте нүктені (немесе векторды) қалыптастырады деп есептейді. Вектордың координаталарының саны жауап векторының өлшемі деп аталады.

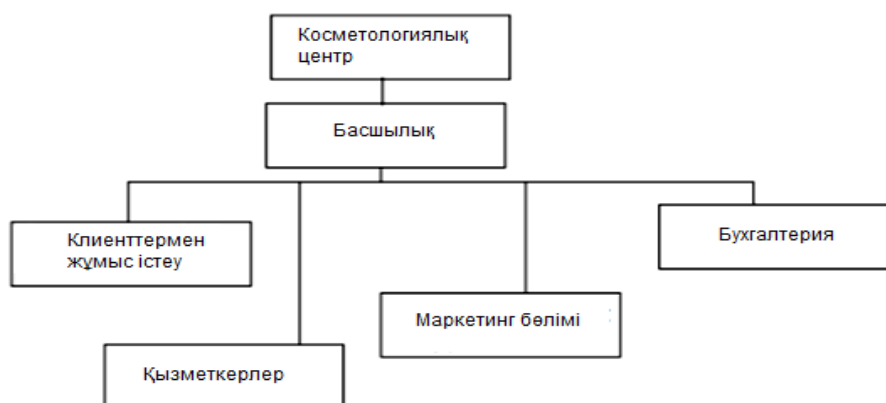


## 2 Тақырыптық аумақты талдау және талап

### 2.1 Пәндік аумақты талдау

Заманауи әлемдегі сұлулықтың маңыздылығын бағаламау қиын. Адамның терісінің денсаулығына және оның сыртқы түріне байланысты мәселелер әр адамның өміріне байланысты. Косметологиялық центр сыртқы көріністі жақсартып қана қоймай, денсаулығын жақсарту үшін түрлі рәсімдерді де жасай алады.

Бұл қызметтер күнделікті тұрмыстағы ең сұранысқа ие болатын үй шаруашылығының қызметтеріне байланысты. Салон бизнес-сегментінің сегменті жеке қызмет көрсету саласының айналымы арасында табыс көлемі бойынша ең үлкені болып табылатыны таңқаларлық емес. Косметикалық бизнесте қызмет көрсетумен айналысатын компания 2.1-суретте ұсынылған келесі құрылымға ие.



2.1-сурет – Кәсіпорынның құрылымы

Косметологиялықө центр маркетинг бөлімінің деректері негізінде шешім қабылдайтын нұсқаулықты қамтиды. Клиенттермен жұмыс істеу бөлімі тікелей қызмет көрсетуші персонал болып табылады. Кадр бөлімі жаңа және қолданыстағы кадрлармен жұмыс жүргізеді. Маркетингтік бөлім клиенттерді тартудың жаңа тәсілдерін іздеуге және қолданыстағы тәсілдерін жақсартуға бағытталған.

Негізгі жұмысты әкімші жүргізеді, оған бухгалтерлік мәселелерді шешу керек, атап айтқанда клиенттерді есепке алу, қызмет көрсетілген клиенттерден төлем қабылдау, жалақыны төлеуге және даму стратегиясына қаражат бөледі. Ақпараттық технологиялар пайда болғанға дейін клиенттерді есепке алу және басқа да көптеген мүмкіндіктер Microsoft office бағдарламалық жасақтамасы арқылы жүзеге асырылды. Үлкен көлемге байланысты уақытылы тапсырмаларды орындаудың кідіруіне алып келді, бұл нарықта бәсекеге қабілеттіліктің төмен болуына алып келді, осылайша салон қызметін

автоматтандыру өзекті(қажетті) міндет болып табылады.

### **2.1.1 Есептің қойылуы**

Әзірленген бағдарламалық қамтамасыз ету (БҚ) салонның барлық қызметкерлерінің жұмыс сапасын арттыруға, сондай-ақ осы салонның клиенттеріне ыңғайлы болу үшін арналған. Жүйе ақпаратты сақтауға және өңдеуге және қызметтер, клиенттер, сондай-ақ салон қызметкерлері туралы деректерді сақтауға тиіс. Жүйе Есептеу техникасы саласында жоғары біліктілігі жоқ Соңғы пайдаланушыға бағдарланған. Сондықтан автоматтандырылған ақпараттық жүйе қарапайым, ыңғайлы және оңай игерілетін интерфейске ие болуы тиіс.

Сондай-ақ сенімді сақтау мен ақпаратқа қол жеткізуді қамтамасыз ететін деректер базасы әзірленуі тиіс. Болжанатын аудитория тері күтіміне, оның денсаулығы мен сыртқы түріне мүдделі тұлғалар тобы болып табылады.

### **2.2 Пәндік саладағы бизнес-үдерістерді талдау**

Бөлінген бизнес-процестер келешекте әзірленетін бағдарламалық өнімнің серверлік бөлігін жобалау және іске асыру кезінде пайдаланылатын болады. Осы жүйенің жұмысын өзіне барынша толық көрсету үшін оны пәндік саланы, оның қасиеттері мен сипаттарын сипаттау үшін пайдаланылатын ұғымдарда және ондағы процестердің өту заңдарын қарастыру керек. Бөлінген бизнес-процестер келешекте әзірленетін бағдарламалық өнімнің серверлік бөлігін жобалау және іске асыру кезінде пайдаланылатын болады.

Тұжырымдамалық модель - бұл модельдеуші жүйенің құрылымын, оның элементтерінің қасиетін және жүйеге тән себеп-салдарлық байланысты анықтайтын және модельдеудің мақсатына жету үшін маңызды абстрактілі модель.

Жүйеде өтетін бизнес-үдерістерді бөліп, оларды 1-кестеге енгіземіз.

## 1-кесте-пәндік саладағы Бизнес-үдерістер

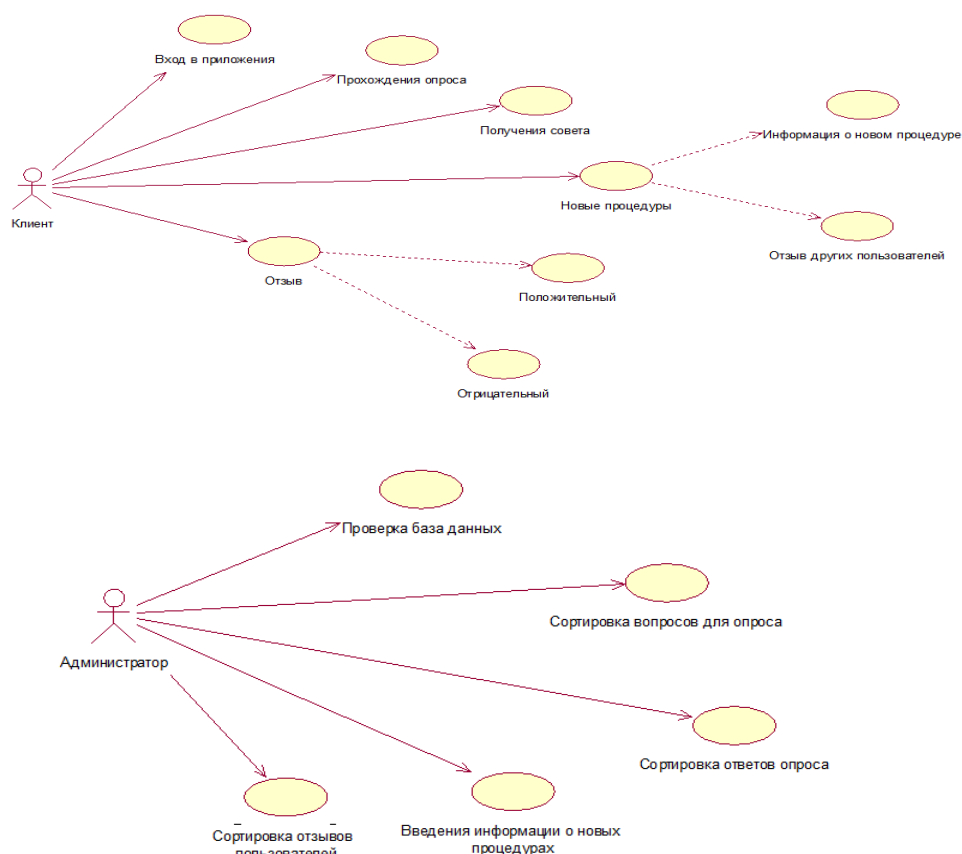
	Бизнес процесс	Орындаушы	Кіріс мәліметтер		Шығыс мәндері		Шектеулер
			Поставщик	Мазмұны	Тұтынушы	Мазмұны	
1	2	3	4	5	6	7	8
1	Клиентті тіркеу	Клиент	Клиент  Администратор	Клиент туралы ақпарат  Қызметтер туралы ақпарат	Қызмет көрсетуші қызметкерлер	Қолданушылардың тізімі	Әр қолданушыға жеке идентификатор тағайындалады  Қолданушылар туралы қайталанатын ақпараттарды болырмау
2	Қолданушылар туралы ақпараттарды өзгерту/өшіру	Клиент	Клиент	Клиент туралы ақпаратты өзгерту	Қызмет көрсетуші қызметкерлер	Қолданушылардың тізімі	-
3	Қолданушыларды авторизациялау	Клиент	Клиент	Клиент туралы ақпаратты қосу	Қызмет көрсетуші қызметкерлер	Қолданушылардың тізімі	-
4	Орталыққа келу туралы ақпаратты қосу	Клиент	Клиент	Клиент туралы ақпарат	Қызмет көрсетуші қызметкерлер	Қолданушылардың тізімі	-
5	Қызмет түрін қосу	Админ	Орталық иесі	Қосымша ақпарат	Администратор	Жаңа тізім	Әр қолданушыға жеке идентификатор тағайындалады
6	Материал сатып алу	Админ	Орталық иесі	Қосымша ақпарат	Администратор	-	-
7	Орталық жұмысын талдау	Админ	Админ	Орталық жұмыс нәтижелері	Орталық иесі	Талдау нәтижесі	-

### 2.3. Функционалдық талаптарды талдау

Функционалдық талаптар пайдалану нұсқаларының диаграммасын жақсы бейнелейді.

Жүйенің жұмыс істеуін егжей-тегжейлі түсіну үшін функционалдық талаптар мен пайдаланушылардың типтері негізінде жобаланған пайдалану нұсқаларының диаграммаларын қарастыру қажет.

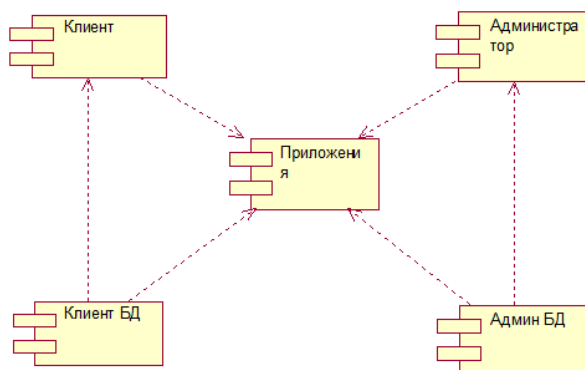
Прецеденттер диаграммасы бағдарламалық қаматамадағы қолданушылар мен администратордың іс-әрекеттерін көрсетеді. Менің пк-да администратор сауалнамаларды қолданушыларға дайындап береді. Оның басты мақсаты қолданушылардың деректер қорын басқара отырып, жаңа процедуралар туралы мағлұматтар енгізу. Ал қолданушылар пк-ға жазылу үшін алдымен тіркеледі, сосын сауалнама сұрақтарына жауап береді. Прецеденттер диаграммасы 2.2-суретте бейнеленген.



**2.2-сурет – Прецеденттер диаграммасы**

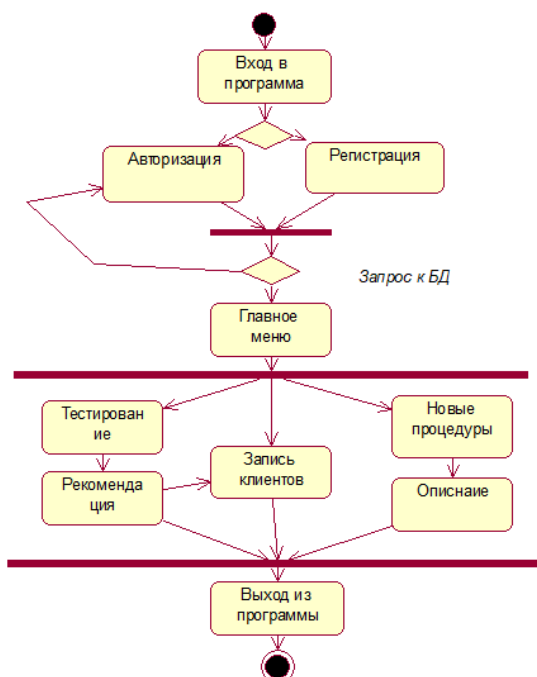
Прецеденттер диаграммасын құрғаннан кейін және актерлер анықтағаннан кейін әрбір прецедентті дәл анықтауға кіріседі. Бұл процесс прецеденттерді нақтылау деп аталады.

Компоненттер диаграммасында бағдарламалық қаматамада қолданылған деректер қорымен жүргізілетін жұмысты бейнелеп көрсетеді. Ең басты деректер қоры ол клиенттер деректер қоры. Бұл деректер қорында қолданушылар туралы деректер сақталады. Қолданушы аты-жөні, логин, пароль, сауалнамаға жауаптары сақталады. Администратор деректер қорында админ логин, пароль, жаңа өнімдер сақталады. Компоненттер диаграммасы 2.3-суретте бейнеленген.



**2.3-сурет – Компоненттер диаграммасы**

Күйлер диаграммасында бағдарламалық қаматамаға кірген кезден бастап бейнеленеді. Ең алдымен бағдарламалық қаматамаға кіру, одан кейін авторизация/регистрация процесі қарастырылады. Бұл процессте деректер қорына сұраным болады. Деректер қорында бар немесе жоқ екендігі тексеріледі. Басты бетте сауалнамалардан өту, жаңа өнім туралы, отзывтарды қарау араларында тандау жасай алады. Күйлер диаграммасы 2.4 - суретте бейнеленген.



**2.4-сурет – Күйлер диаграммасы**

Басқару ағынын бір әрекеттен екіншісіне ауыстыру үдерісін бейнелеу үшін, осы мәселеде қажет болатын басқа UML диаграммаларынан айырмашылығы абстракцияның ең жоғары деңгейіне ие әрекет диаграммасын қарау қажет.

### 2.3.1 Функционалды емес талаптар

Косметологиялықө центр сараптаца жүйесінің бағдарламалық жасақтамасы пайдаланушыға ыңғайлы, қарапайым және түсінікті, айқын графикалық интерфейске ие болуы керек.

Бағдарламалық қамтамасыз ету серверге тұрақты қосылуымен тұрақты жұмыс істеуін қамтамасыз етуі керек.

Серверден деректерді күту уақыты отыз секундтан аспауы керек. Тапсырыстар мен тапсырыс берушілер тізімі автоматты түрде жаңартылады. Жүйеде жаңа пайдаланушы тіркелсе, клиенттік қосымшаның әр данасында

Пайдаланушы бар кестеге енгізіледі. Клиенттік қосымшаға қол жеткізу пайдаланушы түпнұсқалық растамасы арқылы жүргізілуі керек. Пайдаланушы әкімшілік функцияларды орындай алмауы керек.

Функционалды талаптарды талдауға және тақырыптық аумақты талдау негізінде веб-сайтты әзірлеу туралы шешім қабылданды, себебі ол жүйенің негізгі мақсаттарын іске асыру үшін қолайлы.

Веб-торап - веб-серверлерде кез-келген әрекеттердің орындалуын автоматтандыруға арналған қосалқы бағдарламалық құралдар. Бұл жағдайда, веб-браузерлер компьютер арқылы Интернетке қашықтан басқару үшін пайдаланушы интерфейсі ретінде пайдаланылады.

Төмендегі критерийлер функционалды емес талаптарға қолданылады:

- пайдаланушы интерфейстері;
- орындау талаптары;
- бағдарламалық қамтамасыз ету сапасының атрибуттары;
- қауіпсіздік талаптары;
- архитектураға қойылатын талаптар.

### 2.4 Құралдарды таңдау

Қолданбалар туралы ақпаратты құрылымдық сақтау үшін, MS SQL Server 2014 дерекқорды басқару жүйесі (DBMS) және SQL ақпараттық-логикалық дерекқор тілі ретінде реляциялық дерекқорды пайдалану туралы шешім қабылданды. Таңдау SQL кеңінен қолданылатын дерекқорлармен жұмыс істеу үшін бірінші және қазіргі уақытта жалғыз стандартты тіл болып табылатындығына байланысты. Іс жүзінде барлық ірі ДҚБЖ әзірлеушілері өз өнімдерін SQL тілін немесе SQL интерфейсі арқылы жасайды. Ол девелоперлерден де, пайдаланушылардан да үлкен инвестициялар жасады. Ол көптеген ірі және беделді ұйымдардың стратегиялық таңдауын қолдану сәулетінің бөлігі болды.

Ақпараттық жүйенің логикалық және физикалық моделі ERWin Data Modeler құралының көмегімен жүзеге асырылады. Интеграцияланған даму

ортасы - Microsoft Visual Studio 2015.

Жобаны жоспарлау, ресурстарды жұмысты тапсыру және жұмыс көлемін талдаудағы прогрессті қадағалау үшін, Microsoft Project 2013 жобасының менеджеріне жоспарларды әзірлеуде, ресурстарды тапсырмаларға бөлу, жұмыстың көлемін талдау және қадағалау.

## 2.5 Жүйелік бағдарламалық құралдар

### 2.5.1 Dapper ASP .Nnet Core

Әзірлеу негізі платформаны пайдалану болып табылады ASP.NET Core. Бұл платформа заманауи бұлтты веб-қосымшаларды жасау үшін алдын ала ойластырылған. Қосымшалар ASP.NET Core`. NET Core астында немесе толық нұсқада іске қосылуы мүмкін.NET Framework. Фреймворк модульдік компоненттерден тұрады, бұл шешім жасау кезінде икемділік береді. Windows, Mac және Linux астында бағдарламаны жүзеге асыруға болады.

Dapper-sql-сұраулардың нәтижелерін с#сыныптарымен салыстыру (маппинг) технологиясы. Бұл тұрғыда Dapper Entity Framework сәл ұқсас. Сонымен қатар, оның оңай dapper арқасында үлкен өнімділікті қамтамасыз етеді және Entity Framework қарағанда сұраныстарды жылдам орындауға мүмкіндік береді.

Бірінші алдын ала релиз ASP.NET 15 жыл бұрын пайда болды.NET Framework. Содан бері миллиондаған әзірлеушілер оны веб-қосымшаларды жасау және іске қосу үшін пайдаланды, қазіргі уақытта қосымшаларды іске асыру үшін ең өзекті платформа болып табылады.

ASP.NET-те Core архитектуралық өзгерістер көп болды, нәтижесінде фреймворк ықшам және модульдік болды. ASP.NET Core System негізделген емес.Web.dll. Ол NuGet пакеттерінің жиынтығына негізделеді. Бұл қолданба тек қана NuGet пакеттерін қосу үшін бағдарламаны оңтайландыруға мүмкіндік береді.

Startup класында сұраныстарды өңдеу ағыны анықталады, сондай-ақ қосымшаға қажетті барлық сервистер теңшеледі. Startup класы әдістерді қамтиды.

Сервис-бұл қосымшадағы жалпы пайдалануға арналған компонент. Сервистер тәуелділікті енгізу арқылы қол жетімді. ASP.NET Core кірістірілген IOC контейнерін қамтиды, ол әдепкі бойынша конструктор енгізуді қолдайды, бірақ оны IOC контейнермен оңай ауыстыру мүмкіндігі бар. Әлсіз байланыс артықшылығына қосымша, DI сервистер барлық қолданбаға қол жетімді.

ASP.NET –те Core біз байланыстырушы бағдарламалық қамтамасыз етуді (Middleware) пайдалана отырып, сұраныстар легін құрастырамыз. Байланыс ASP.NET Core HttpContext үшін асинхронды логиканы орындайды, содан кейін келесі байланыстырушы БҚ тізбегіне шақырады немесе тікелей

сұрау салуды бұзады. Әдетте байланыстырушы БҚ үшін Use қолданылады, NuGet пакеті үшін тәуелділікті қабылдай отырып және Configure әдісі бойынша IapplicationBuilder үшін UseXYZ кеңейтудің тиісті әдісін шақырады.

Хостинг моделі ASP.NET Core сұрауларды тікелей тыңдамайды-ол app сұрауларын жіберу үшін HTTP серверлік жүзеге асыруға сүйенеді. Берілген сұраныс feature интерфейстерінің жиынтығы ретінде ұсынылады, ол бағдарлама содан кейін HttpContext-те компоненттейді. ASP.NET Core IIS немесе nginx сияқты өндірістік веб серверге әдетте іске қосылатын Kestrel, кроссплатфорлы веб серверді қамтиды. Түбірлік директория-қолданбада пайдаланылатын кез келген мазмұнға, мысалы, оның көріністері мен веб мазмұнына негізгі жол. Әдепкі бойынша мазмұнның түбірлік директориясы - орындалатын хостинг үшін қолданбаның негізгі жолы сияқты; балама WebHostBuilder көмегімен көрсетуге болады.

Веб (web root) негізгі директориясы - css, js файлдары және сурет файлдары сияқты ашық статикалық ресурстар үшін директория. Статикалық файлдарды әдепкі бойынша тек осы директориядан (және қосалқы директориядан) өңдейді. ASP.NET Core қарапайым жұптарды өңдеу үшін жаңа конфигурациялық модельді пайдаланады. Жаңа конфигурациялық модель System-ге негізделмейді.Configuration, ол реттелген конфигурациялық провайдерлер жиынтығынан жасалады. Кірістірілген конфигурациялық провайдерлер көптеген файл пішімдерін (XML, JSON, INI) және айнымалы ортаны қолдайды. Сондай-ақ, жеке теңшелетін конфигурациялық провайдерлерді жазу мүмкін.

## 2.5.2 MVC архитектурасы

Қолданба интерфейсі MVC архитектурасы арқылы жазылған. Егер ұғымға негізделсе, онда MVC архитектуралық үлгісі MVC қосымшасы төменде сипатталған үш бөлікке бөлінетінін білдіреді: пайдаланушылар жұмыс істейтін деректерді қамтитын немесе ұсынатын модельдер.

Олар көріністер мен бақылаушылар арасында берілетін деректерді ғана ұсынатын көріністердің қарапайым модельдері болуы мүмкін, немесе олар бизнес-деректерден, сондай-ақ операциялардан, түрлендірулерден және осы деректермен айла-шарғы жасау ережелерінен тұратын пәндік саланың модельдері болуы мүмкін.

Модельдің кейбір бөлігін пайдаланушы интерфейсі түрінде визуализациялау үшін қолданылатын көріністер. Контроллерлер, өңдейді келіп түсетін сұрау салуларды орындайды операциялар моделі және таңдайды ұсыну үшін визуализация пайдаланушыға.

Мысалы, банк-қосымшасында моделі ұсынады барлық аспектілері банк қызметін қолдаумен қоса, мұндай есеп айырысу шоттары, негізгі бухгалтерлік кітап және несие лимиттері клиенттер үшін, сол сияқты операциялар үшін



пайдаланылуы мүмкін айла-шарғы жасау деректер моделін сияқты ақша қаражаттарын енгізу және есептен шығару.

Модель деректердің жалпы жай– күйі мен бүтіндігін сақтау үшін де жауап береді-мысалы, барлық транзакциялар Бас кітапқа енгізілгенін, ал клиент шоттан оған құқығы бар немесе банктің иелігінде бар құжаттан артық ақша қаражатын алмайды. Сонымен қатар, олар жауап бермейтінімен де анықталады: модельдер пайдаланушы интерфейстерін визуалаумен немесе сұраныстарды өңдеумен айналысатын істерге ие емес, өйткені бұл көріністер мен контроллерлердің жауапкершілігі.

Көріністер пайдаланушыға модель элементтерін көрсету үшін қажетті логиканы қамтиды - және ештеңе артық. Олардың модель туралы ешқандай тікелей мәліметтері жоқ және деректермен тікелей алмаспайды.

MVC архитектурасының әрбір бөлігі нақты және өзін – өзі қамтамасыз ету болып табылады-бұл жауапкершілікті бөлу деп аталады. Бұл модельде деректерді манипуляциялайтын Логика тек модельде ғана қамтылған, деректерді көрсететін логика – тек көріністе ғана, ал сұранымдарды өңдейтін және пайдаланушыларды енгізетін код – тек контроллерде ғана. Барлық бөліктерді нақты ажырату кезінде қосымша қаншалықты үлкен болғанына қарамастан, оның қызмет ету мерзімі ішінде бірге жүру және кеңейту оңай болады.

MVC инфрақұрылымында контроллерлер C# класстары, әдетте System класынан туынды Web.Mvc.Controller. Контроллерден алынған сыныпта әрбір ашық әдіс маршрутизация жүйесі арқылы әрекет ету әдісі болып табылады ASP.NET конфигурацияланатын URL-мен байланыстырылады. Сұрау әрекет әдісімен байланысты URL арқылы жіберілген кезде, контроллер класындағы операторлар пән аймағының моделіне кейбір операцияны жүргізу үшін және клиентке көрсету үшін Көріністі таңдау үшін орындалады.

Инфрақұрылымда ASP.NET MVC Framework шолғыш үшін жауапты генерациялау мақсатында көріністі өңдеуге жауап беретін компонент - визуализация механизмі қолданылады.

Ерте MVC нұсқаларында стандартты визуализация механизмі қолданылды ASP.NET Web Forms белгілеу синтаксисінің жаңартылған нұсқасын пайдалана отырып, ASPX - беттерін өңдеген. MVC 3-те MVC 4 нұсқасында (және MVC 5 нұсқасында өзгеріссіз қалған) жетілдірілген razor визуализациялау механизмі пайда болды, онда мүлдем басқа синтаксис қолданылады.

Visual Studio ортасы Razor қозғалтқышы үшін IntelliSense құралына қолдау көрсетеді, контроллер жіберген презентация деректеріне орындауды және жауапты жеңілдетеді.

ASP.NET MVC инфрақұрылымы домен үлгісін іске асыруға қандай да бір шектеулер енгізбейді. Сіз әдеттегі C # нысандарымен модель жасай аласыз және кез-келген дерекқорды, объект-реляциялық карталар инфрақұрылымын немесе .NET қолдайтын басқа деректерді өңдеу құралдарын пайдалану арқылы табандылықты қолдана аласы

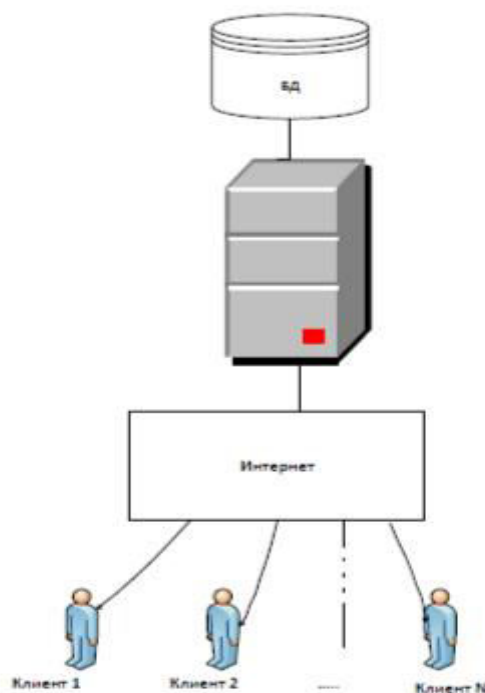
## 3 Жобалау

### 3.1 Негізгі жүйе архитектурасы

Қаралып жатқан жүйе екі деңгейлі клиенттік-серверлік архитектураға ие. Автоматтандырылған ақпараттық жүйенің құрылымы 3.1-суретте келтірілген.

Бұл архитектурада жүйенің 3 негізгі құраушысы бар.

Бірінші компонент деректер базасы болып табылады. Онда тұрақты деректер мен салонның қызметін жүргізу фактісі сақталады.



**3.1-сурет – Жүйенің архитектурасы**

Екінші компонент деректер базасының солтүстігін білдіреді, онда барлық бизнес логика салынған. Үшінші компонент-пайдаланушы мен ақпараттық жүйенің серверлік бөлігі арасындағы өзара іс-қимыл интерфейсі болып табылатын клиенттік бөлік. Клиенттер қызметке клиенттік қосымшада өзінің жұмыс станциясында жазылады. Клиенттік қосымша интернет арқылы деректерді деректер қорының серверіне жібереді. Дерекқор сервері алынған ақпаратты сақтайды. Кезекті тексеру нәтижелері бойынша әрбір станцияның клиенттік қосымшасы Деректерді жаңарту мақсатында серверге сұрау жібереді. Сервер әр станцияның жаңартылған деректерін жібереді.

### 3.2 Логикалық дерекқор үлгісі

Логикалық модель пәндік саладағы түсініктерді, олардың қарым-қатынастарын, сондай-ақ тақырыптық аймаққа енгізілген деректерге шектеулерді сипаттайды. Логикалық деректер моделі болашақ дерекқордың бастапқы прототипі болып табылады. Логикалық модель ақпараттық бірліктерде құрастырылған, бірақ белгілі бір ДҚБЖ сілтемесіз. Сонымен қатар, логикалық деректер моделі реляциялық деректер моделі арқылы міндетті түрде білдіруге міндетті емес. Қазіргі кезде логикалық деректер моделін құрудың негізгі құралы ER-диаграммалардың әртүрлі нұсқалары (субъект-қатынас, субъект-қатынас диаграммалары) болып табылады. Сол ER үлгісі реляциялық деректер үлгісіне, сондай-ақ иерархиялық және желілік ДББЖ үшін деректер үлгісіне немесе пост-реляциялық деректер үлгісіне айналуға болады.

Алдыңғы деңгейде домендік модельді әзірлеу кезінде қабылданған шешімдер логикалық деректер моделі жасалуы мүмкін кейбір шектеулерді анықтайды, ал осы шектерде әртүрлі шешімдер қабылдануы мүмкін .

Қалыпты пішін реляциялық деректер үлгісіндегі қатынас сипаты болып табылады, ол оны іріктеу немесе өзгерту деректерінің логикалық қате нәтижелеріне әкелуі мүмкін артықшылықты сипаттайтын. Қалыпты пішін қатынасы қанағаттандыруы тиіс талаптар жиынтығы ретінде анықталады. Әдеттегі қалыпты формалар (3NF) бар, бірақ әдетте практикалық шешуге арналған.

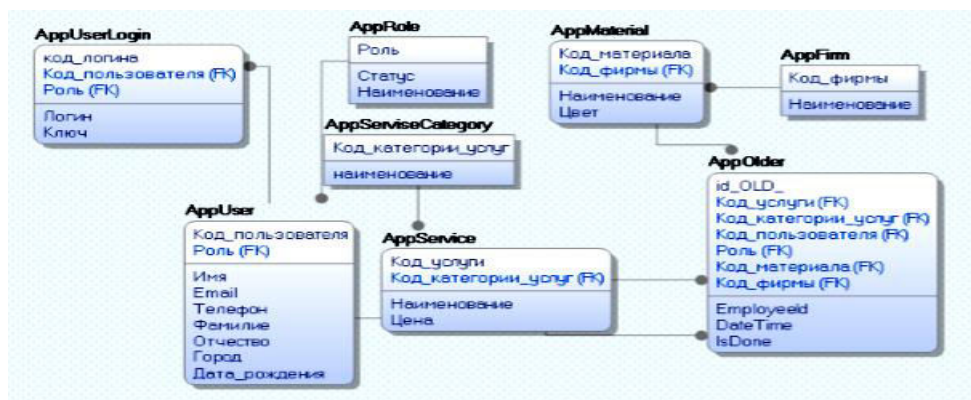
Логикалық деректер моделі 3 нысанға бөлінеді.

Бірінші қалыпты пішін (1NF) жалпы қатынас. Байланысты анықтауға байланысты, кез-келген қатынас автоматты түрде 1NF-де. Әр түрлі кестелерге келетін болсақ, олар қатынастардың дұрыс көрінісі болмауы және сәйкесінше 1NF-де болмауы мүмкін .

Егер қатынасы 1NF-да болса және күрделі кілттің бөлігіне байланысты кілтсіз атрибуттар болмаса, ара қатынасы екінші қалыпты түрде (2NF) болады.

Егер байланыс 2NF-де болса және барлық кілттік емес атрибуттар бір-бірінен тәуелсіз болса, қатынас 3-қалыпты қалыпта (3NF). Біздің жағдайда әрбір қарым-қатынастың кілтсіз емес төлсипаты тек бастапқы кілтке байланысты болады .

ERIGN Data Modeler құралының көмегімен IDEF1X белгілерінде логикалық домен үлгісін құрыңыз. Логикалық модель 3.2-суретте келтірілген.

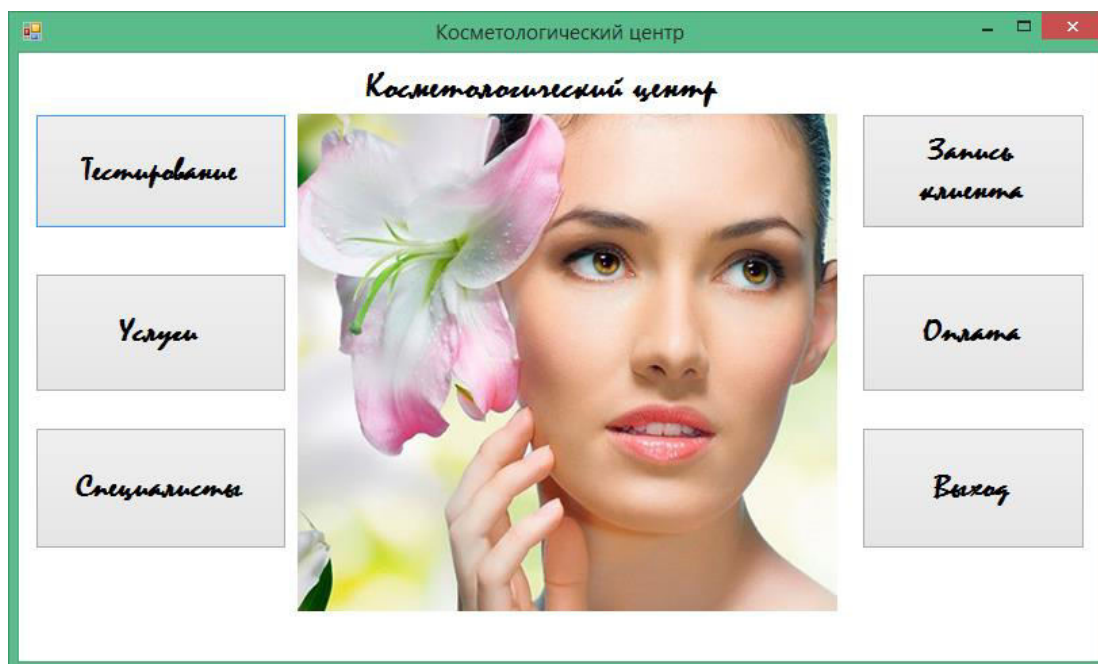


3.2-сурет – Дерекқордың логикалық моделі

Жасалған логикалық деректер базасының үлгісі транзиттік тәуелділіктерді қамтымайды. Осылайша, модель үшінші қалыпты пішінге сәйкес келеді деген тұжырым жасауға болады. Әрі қарай жұмыс барысында логикалық модельден физикалық мәнге көшу жүзеге асырылады.

### 3.3 Пайдаланушы интерфейсін жасау

Қолдану диаграммасына сүйене отырып, графикалық пайдаланушы интерфейсін жасалды, онда функцияның рұқсат етілген пайдаланушыға байланысты бөлінуі 3.3-суретте көрсетілген.



3.3-сурет – Визуалды компоненттер графикалық интерфейс

15 класс жобаланды. Төменде олардың қысқаша сипаттамасы:

1) Program.cs-веб-серверді құру, Директория жолдарын орнату, IIS параметрлерін орнату, веб-серверді іске қосу болатын басты Main () кіру функциясын қамтиды

2) Startup.cs-конфигурацияны қамтиды және конфигурация бойынша инициализацияны орындайды;

3) FirmService.cs-фирмалар тізімін алуға, жеке фирманы алуға, фирманы құруға, өңдеуге және жоюға арналған;

4) FirmController.cs-фирмаға қатысты сұрауларды қабылдайды, фирманы алуға мүмкіндік береді, немесе тізім, құру, өзгерту немесе жою;

5) DateTimeHelper.cs – орыс уақыт форматына сәйкес күнді жолға және кері түрлендіру әдістері бар;

6) CreateFirmViewModel.cs-Контроллерден көрініске және кері жіберілетін деректер бар

7) FirmRepository фирмаға қатысты іс-қимылдарға арналған

8) Stores - datarepositories үстінен орама;

9) DataBaseExecutor-деректер қоры бойынша командаларды орындауға арналған;

10) ConnectionFactory-қосылу фабрикасы.

Firms.cs, Material.cs, Older.cs, Rose.cs, Service.cs, ServiceCategory.cs, User.cs-деректер қорының кестелерімен өзара әрекеттесуді қамтамасыз ететін сыныптар.

### **3.4 Жобаны жоспарлау және бюджетті бағалау**

Жобаны жоспарлаудың негізгі түрлерінің бірі құрылымдық жоспарлау болып табылады, ол келесі кезеңдерді қамтиды: жобаны орындау жобаны іске асыру үшін қажет жеке жұмыстардың жиынтығына жобаны бөлу; жұмысты орындау жүйелілігін сипаттайтын желілік графикті құру; жұмыстың уақыт сипаттамаларын бағалау және желілік графикті талдау. Жобаларды басқару-жұмыс көлемі, еңбек және материалдық ресурстар, уақыт, сапа және тәуекелдер арасындағы теңгерімдеу кезінде жобаның мақсаттары анықталатын және оған қол жеткізілетін қызмет саласы. Жобалық басқару жетістігінің негізгі факторы нақты, алдын ала белгіленген жоспардың болуы болып табылады.

### 3.4.1 Жобаны жоспарлау

Microsoft Project бағдарламасындағы жоспарлау процесі өзіндік ерекшелігіне ие. Жобаның мақсаты анықталғаннан кейін пәндік сала мен функционалдық талаптарды талдауда көрсетілген негізгі кезеңдер мен олардың мазмұны тұжырымдалған жоба жоспары құрылады; бұдан әрі жоба міндеттерінің тізімін анықтау және құрылымдау қажет, әрбір міндет үшін міндеттер арасында ұзақтықты, тәуелділікті орнату; содан кейін ресурстар тізімін құру: қызметкерлер, жабдықтар мен материалдар, ресурстарды міндеттерге тағайындау қажет. Енгізілген Microsoft Project ақпараты негізінде кесте жасайды. Осылайша жасалған кестені теңшеуге және оңтайландыруға болады.

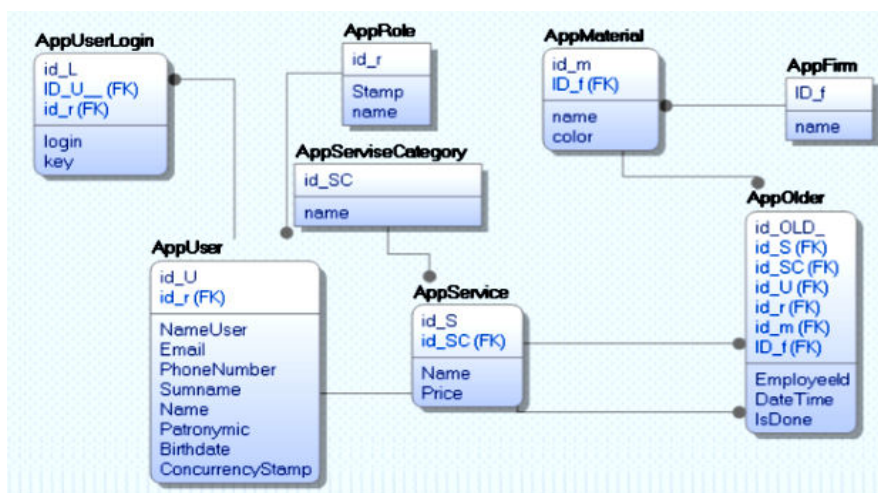
Физикалық модель логикалық деректер моделіне негізделеді және мақсатты ДҚБЖ сипаттамаларын ескереді. SQL Server 2017 мақсатты DBMS ретінде таңдалған. ERWin-де логикалық модельден жекеге оңай ауыса аласыз, алайда, жеке нысан түрлері физикалық деректер базасында кестелерге айналады;

- атрибуттар физикалық деректер базасында бағандарға айналады;
- бірегей идентификаторлар бастапқы кілттер деп аталатын NULL емес бағандарға айналады;

Қатынастар сыртқы кілттермен модельденеді .

ERWin Data Modeler-дың маңызды қасиеттерінің бірі - алға қойылған дизайнды және керісінше өткізу қабілеттілігі.

Дерекқордың физикалық моделі 3.4-суретте көрсетілген.



3.4-сурет – Дерекқордың физикалық моделі

ERWin Data Modeler-да құрылған DDL сценарийі MS SQL Server 2017 DBMS-де орындалды, нәтижесінде 3.4-суретте ұсынылған дерекқор схемасы алынды.

## ҚОРЫТЫНДЫ

Сұлулық салондары мен сұлулық пен денсаулық саласының басқа да мекемелері бүгінгі күні танымал болып келеді. Сондықтан осы қызмет түріне тән процестер мен тапсырмаларды автоматтандыратын ақпараттық жүйелерге сұраныс артты. Дегенмен, мұндай кәсіпорындардың барлық басшылары қызметтердің сапасын жақсарту және еңбек жағдайларын жақсарту үшін ақпараттық технологияларды пайдаланбайды, кейде тіпті электрондық компьютерлерді пайдаланудан толық бас тартады, бұл қосымша шығындарға әкеледі.

Бұл жұмыстың мақсаты косметикалық бизнес саласында қызмет көрсетумен айналысатын кәсіпорындар үшін автоматтандырылған сараптаудың ақпараттық жүйесін құру болып табылады.

Сұлулық пен денсаулық сақтау мекемелері үшін дайын бағдарламалық өнімдерді талдағаннан кейін, оларды әрі қарай талдау үшін әр түрлі есептерді жасауға мүмкіндік беретін қуатты аналитикалық бірлікке ие екендіктерін айтуға болады. Осы жүйелерде іске асырылатын функциялар сұлулық индустриясының институттарында басқарудың, ұйымдастырудың және есепке алудың көптеген міндеттерін шешеді.

Косметологиялық орталық үшін автоматтандырылған жүйелердің көпшілігі күрделі интерфейске ие, ол көптеген өрістер, функциялар мен әрекеттерге толы. Бұл тіпті тәжірибелі компьютер пайдаланушысы үшін жүйемен жұмыс істеу тиімділігін төмендетеді. Соны ескере отырып, ұсынылып отырған жұмыста құрылған ақпараттық жүйенің интерфейсін мүмкіндігінше түсінікті етуге тырыстық. Сондықтан, осы ортада жұмыс істеуге қызметкерлерді оқыту үшін қосымша уақыт пен қаржы шығындары талап етіледі. Іске асырудың жоғары деңгейін және техникалық қызмет көрсетудің жоғары құны, сондай-ақ қаралған бағдарламалық өнімдердің күрделілігіне байланысты кәсіпорын басшыларының өз қызметін автоматтандыру үшін шешімдерін қабылдауға асықпайды.

Дегенмен, сұлулық индустрия институттарының мамандандырылған процестерін автоматтандыру ақпараттық технологияларды дамытудың келешегі зор бағыт болып табылады, бұл өз кезегінде ұсынылатын қызметтердің сапасын арттыруға мүмкіндік береді. Сондықтан осы жұмыстың өзектілігі айқын, себебі қазіргі таңда ең қарқынды дамып келе жатқан косметология саласы бойынша сараптау жүйесі енгізілген ақпараттық жүйені ұсынып отыр.

## ПАЙДАЛАНЫЛҖАН ӘДӘБИЕТТЕР ТІЗІМІ

- 1 Новожилова Н.В., Алякина Л.А. Вопросы проектирования информационных систем //Вестник Чувашского университета. 2011. №1. С. 432-438.
- 2 Целигорова Е.Н. Современные информационные технологии и их использование для исследования систем автоматического управления //Инженерный вестник Дона. 2010. №3. С. 140-144.
- 3 Описание программного продукта «Universe-красота» URL: <http://www.universe-soft.ru> (дата обращения: 10.12.15)
- 4 Описание программного продукта «Sycret Saloon» URL: <http://www.sycret.ru/>
- 5 Алиев, Р.А. Производственные системы с искусственным интеллектом / Р.А. Алиев, Н.М. Абдикеев, М.М. Шахназаров. - М.: Радио и связь, 2016. - 264 с.
- 6 Васильев, В.И. Распознающие системы / В.И. Васильев. - К.: Наукова думка; Издание 2-е, перераб. и доп., 2016. - 425 с.
- 7 Габец, А. Профессиональная разработка в системе 1С: Предприятие 8 / А. Габец, Д. Гончаров, Д. Кухлевский. - М.: 1С: Паблишинг, 2014. - 808 с.
- 8 Джордж Основы кибернетики / Джордж, Фрэнк. - М.: Радио и связь, 2016. - 272 с.
- 9 Любарский, Ю.Я. Интеллектуальные информационные системы / Ю.Я. Любарский. - М.: Наука, 2013. - 232 с.
- 10 Марка, Д. Методология структурного анализа и проектирования SADT / Д. Марка, К. Макгоуэн. - М.: МетаТехнология, 2015. - 240 с.
- 11 Матэ Добердо: моногр. / Матэ, Залка. - М.: СПб: Художественная литература, 2014. - 280 с.
- 12 Методы и средства преобразования информации. - М.: Рига: Зинатне, 2016. - 136 с.
- 13 Мичи, Д. Компьютер - творец / Д. Мичи, Р. Джонстон. - М.: Мир, 2017. - 255 с.
- 14 Мкртчян, С.О. Нейроны и нейронные сети / С.О. Мкртчян. - М.: Энергия, 2017. - 232 с.15. Нариньяни, А.С. Моделирование языковой деятельности в интеллектуальных системах / ред. А.Е. Кибрик, А.С. Нариньяни. - М.: Наука. Главная редакция физико-математической литературы, 2012. - 280 с.
- 15 Нейлор, К. Как построить свою экспертную систему / К. Нейлор. - М.: Энергоатомиздат, 2013. - 286 с.17. Нильсон, Н. Принципы искусственного интеллекта / Н. Нильсон. - М.: Радио и связь, 2014. - 373 с.
- 16 Пенроуз, Р. Новый ум короля. О компьютерах, мышлении и законах физики / Р. Пенроуз. - М.: Едиториал УРСС, 2012. - 384 с.
- 17 Реальность и прогнозы искусственного интеллекта / ред. В.Л. Стефанюк. - М.: Мир, 2016. - 247 с.



18 Розенблатт, Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга / Ф. Розенблатт. - М.: Мир, 2013. - 480 с.

19 Системы автоматизированного проектирования в 9 ти книгах. Книга 9 Иллюстрированный словарь / ред. И.П. Норенков. - М.: Высшая школа, 2012. - 159 с.

## **А қосымшасы** (міндетті)

### Техникалық тапсырма

#### **А.1.5 Косметологияда сараптамалық жүйені қолдануға техникалық тапсырма**

Техникалық тапсырма косметологияда сараптамалық жүйені қолдануға бағытталған. Бұл бағдарламалық қамтама қолданушылардың бет терісінің типін анықтап, беттің терісіне күтім көрсету арқылы жақсарту мақсатында жасап шығарылады. Косметологиялық орталықта ұсынылатын процедураларға жазылу алдын бет терісі типін анықтау үшін сараптамалық жүйе қолданылды.

##### **А.1.5.1 Мақсаты**

Бұл программалық қамтаманың мақсаты қолданушыларына бет терісінің күтіміне көмек көрсететін кеңестер беру.

##### **А.1.5.2 Жүйенің функционалды талаптары**

Жүйе келесі функцияларды орындай алу керек:

- жүйенің инициализациясы;
- кіріске берілген суреттің дұрыс өңделуі және берілген сурет негізінде
- жасалған қорытынды ақпарат;
- кіріске берілген сурет негізінде керекті ақпаратты көрсету;

Бастапқы деректер:

- адам бет терісінің түрлерімен артықшылықтары;
- сараптамалық жүйені дайындау;

Нәтиже:

- класс атауын экранға шығару;

## **А қосымшасының жалғасы**

### **А.1.5.3 Қауіпсіздік талаптары**

Кіріс туралы ақпаратты бақылауды қамтамасыз етіңіз. Жүйемен жұмыс істеу кезінде қате пайдаланушы әрекеттерін блоктауды қамтамасыз етіңіз. Сақталған ақпараттың тұтастығын қамтамасыз етіңіз.

### **А.1.5.4 Техникалық талаптар**

Жүйе IBM үйлесімді дербес компьютерлерде жұмыс істеуі керек. Ең аз конфигурация: процессор түрі - Pentium және одан жоғары; Жад сыйымдылығы - 32 МБ және одан жоғары.

### **А.1.5.5 Бағдарламалық үйлесімділік талаптары**

Жүйе Win32 API-ді іске асыратын Windows операциялық жүйелерінің бақылауында жұмыс істеуі керек.

**Б қосымшасы**  
(міндетті)

Бағдарлама коды

```
namespace WindowsFormsApp1
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен
        быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        /// <summary>
        /// Требуемый метод для поддержки конструктора — не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
        }
    }
}
```

## Б қосымшасының жалғасы

```
this.button3 = new System.Windows.Forms.Button();
this.button4 = new System.Windows.Forms.Button();
this.button5 = new System.Windows.Forms.Button();
this.button6 = new System.Windows.Forms.Button();
this.button7 = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.SuspendLayout();
//
// pictureBox1
//
this.pictureBox1.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
this.pictureBox1.Location = new System.Drawing.Point(-3, -2);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(800, 417);
this.pictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom;
this.pictureBox1.TabIndex = 0;
this.pictureBox1.TabStop = false;
//
// button1
//
this.button1.BackColor = System.Drawing.Color.White;
this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button1.Location = new System.Drawing.Point(12, 114);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(227, 59);
this.button1.TabIndex = 1;
this.button1.Text = "Услуги";
this.button1.UseVisualStyleBackColor = false;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// button2
//
this.button2.BackColor = System.Drawing.Color.White;
this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button2.Location = new System.Drawing.Point(553, 304);
```

## Б қосымшасының жалғасы

```
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(227, 59);
this.button2.TabIndex = 1;
this.button2.Text = "КЛИЕНТЫ";
this.button2.UseVisualStyleBackColor = false;
this.button2.Click += new System.EventHandler(this.button2_Click);
//
// button3
//
this.button3.BackColor = System.Drawing.Color.White;
this.button3.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button3.Location = new System.Drawing.Point(12, 205);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(227, 59);
this.button3.TabIndex = 1;
this.button3.Text = "Тестирование";
this.button3.UseVisualStyleBackColor = false;
this.button3.Click += new System.EventHandler(this.button3_Click);
//
// button4
//
this.button4.BackColor = System.Drawing.Color.White;
this.button4.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button4.Location = new System.Drawing.Point(553, 205);
this.button4.Name = "button4";
this.button4.Size = new System.Drawing.Size(227, 59);
this.button4.TabIndex = 1;
this.button4.Text = "Оплата";
this.button4.UseVisualStyleBackColor = false;
this.button4.Click += new System.EventHandler(this.button4_Click);
//
// button5
//
this.button5.BackColor = System.Drawing.Color.White;
this.button5.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button5.Location = new System.Drawing.Point(12, 304);
```

## Б қосымшасының жалғасы

```
this.button5.Name = "button5";
this.button5.Size = new System.Drawing.Size(227, 59);
this.button5.TabIndex = 1;
this.button5.Text = "Специалисты";
this.button5.UseVisualStyleBackColor = false;
this.button5.Click += new System.EventHandler(this.button5_Click);
//
// button6
//
this.button6.BackColor = System.Drawing.Color.White;
this.button6.Font = new System.Drawing.Font("Microsoft Sans Serif",
20.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button6.Location = new System.Drawing.Point(553, 114);
this.button6.Name = "button6";
this.button6.Size = new System.Drawing.Size(227, 59);
this.button6.TabIndex = 1;
this.button6.Text = "Запись";
this.button6.UseVisualStyleBackColor = false;
this.button6.Click += new System.EventHandler(this.button6_Click);
//
// button7
//
this.button7.Font = new System.Drawing.Font("Monotype Corsiva", 14.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))), System.Drawing.GraphicsUnit.Point,
((byte)(204)));
this.button7.Location = new System.Drawing.Point(318, 339);
this.button7.Name = "button7";
this.button7.Size = new System.Drawing.Size(143, 59);
this.button7.TabIndex = 2;
this.button7.Text = "Выход";
this.button7.UseVisualStyleBackColor = true;
this.button7.Click += new System.EventHandler(this.button7_Click);
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoScroll = true;
this.AutoSize = true;
this.ClientSize = new System.Drawing.Size(809, 427);
```

## Б қосымшасының жалғасы

```
this.Controls.Add(this.button7);
this.Controls.Add(this.button6);
this.Controls.Add(this.button5);
this.Controls.Add(this.button4);
this.Controls.Add(this.button3);
this.Controls.Add(this.button2);
this.Controls.Add(this.button1);
this.Controls.Add(this.pictureBox1);
this.Name = "Form1";
this.Text = "Косметологический центр";
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button button4;
private System.Windows.Forms.Button button5;
private System.Windows.Forms.Button button6;
private System.Windows.Forms.Button button7;
}
}
```



<i>Формат</i>	<i>Зона</i>	<i>Поз.</i>	<i>Обозначение</i>	<i>Наименование</i>	<i>Кол.</i>	<i>Примечание</i>

