

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі
ф.-м.ғ.д., проф. Құралбаев З.К.

(КОЛЫ)

« _____ » _____ 2014 ж.

ДИПЛОМДЫҚ ЖОБА

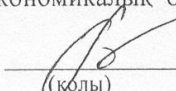
Тақырыбы: «Алматы энергетика және байланыс университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған ақпараттық жүйесін құру»
5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша

Орындаған Достанов Абзал Сайлаубайұлы тобы: ВТк-10-1

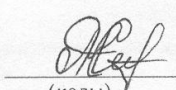
Жетекші ф.-м.ғ.к., доцент Қоржымбаев Т.Т

Кеңесшілер :

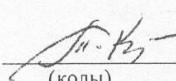
Экономикалық бөлім бойынша :


(КОЛЫ) доцент Боканова Г.Ш.
« 26 » 05 2014ж.

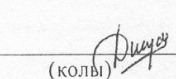
Өмір тіршілігі қауіпсіздігі бойынша:


(КОЛЫ) т.ғ.к., аға оқытушы Муташева Г.С.
« 10 » 05 2014 ж.


Есептеу техникасын қолдану бойынша :


(КОЛЫ) ф.-м.ғ.к., доцент Қоржымбаев Т.Т
« 05 » 06 2014 ж.

Мөлшер бақылаушы:


(КОЛЫ) аға оқытушы Тусупов Д.М.
« 09 » шолпан 2014 ж.

Пікір жазушы :

т.ғ.д., ҚазЭУ ҚИ кафедрасының проф. Абдылдаев Э.К.

(КОЛЫ) « 30 » 05 2014 ж.

Алматы 2014

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

*Суреттер 3.3.1 Түрлер екітсе мерде, 3.4.2 Бағдарламаны
ишантыу бөліміне кеңес тізімі.*

Негізгі ұсынылатын әдебиеттер:

1. Алексей Вишневский. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
2. Полякова Л.Н. Основы SQL. Курс лекций: учеб. пособие. – Москва, 2004. – 368с.
3. Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
4. Агапов В.П. Основы программирования на языке С# . учеб.пособие.- Москва,2012.-128с.
5. Голицына О.Л., Попов И.И.Основы алгоритмизации и программирования. учеб.пособие.-Форум 2008.-432с.

Жоба тараулары бойынша кеңес берушілер және онын мерзімі:

Бөлім	Кенесші	Мерзімі	Қолы
Негізгі бөлім	Қоржымбаев Т.Т.	05.06.2014	<i>[Signature]</i>
Гіршілік қауіпсіздігі	Муташева Г.С.	20.05.2014	<i>[Signature]</i>
Экономикалық бөлім	Боканова Г.Ш.	25.04 - 25.05.14	<i>[Signature]</i>
Норма бақылаушы	Ержан А.А.	09.06.2014	<i>[Signature]</i>
Есептеу техникасын қолдану	Қоржымбаев Т.Т.	05.06.2014	<i>[Signature]</i>

Менің дипломдық жобамда Алматы энергетика және байланыс университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған ақпараттық жүйесін құру. Бағдарламаның деректер қорын My SQL 2008 r2 жасалынып ал интерфейсі Microsoft Visual Studio 2013 C# тілінде жасалынды.

Техникалық бөлімінде пайдаланылатын технологиялардың маңыздылығы және бағдарлама құру кезеңдері көрсетілген.

«Өміртіршілік қауіпсіздігі» бөлімінде еңбек жағдайларына, жұмыс орнына талдау жасалынды.

«Техника-экономикалық негізделуі» бөлімінде енгізілген ақпараттық жүйеден алынған экономикалық пайданы есептеу жүргізілді.

Аннотация

Цель моей дипломной работы – разработка информационной системы для приемной комиссии Алматинского Университета Энергетики и Связи. База данных программы была создана на основе системы My SQL 2008 r2, а интерфейс написан на языке Microsoft Visual Studio 2013 C#.

В технической части работы описаны значение используемых технологий и этапы создания программы.

В разделе «Правила техники безопасности» указаны требования к условиям труда и рабочему месту.

В части «Технико-экономическое обоснование» произведены расчеты по исчислению условного эффекта, получаемого от внедрения разработанной программы.

Abstract

The aim of my thesis – cultivating information system for the selection committee of University of Power Engineering and Telecommunications in Almaty. The database program was created on the My SQL 2008 r2 basis, and the interface was written in Microsoft Visual Studio 2013 C #.

In the technical part of the paper the value of the technology used and the steps of creating the program were described.

The "Safety Rules" part maintains the requirements for working conditions and workplace.

"Feasibility Study" part includes accounts based on calculations of the conditional effect, resulting from the application of the developed program.

Мазмұны

1	Техникалық тапсырма.....	13
1.1	Тағайындалуы және қолдану аймағы.....	13
1.1.1	Сенімділікке қойылатын талаптар	13
1.1.2	Техникалық құрылғыларға қойылатын талаптар	13
1.1.3	Бағдарламалық құжатқа қойылатын талаптар	13
1.2	Құру кезеңдері мен сатылары	14
1.3	Автоматтандырылған жұмыс орнының (АЖО) мен құрылымы.....	14
1.3.1	АЖО құрылымы.....	15
1.3.2	АЖО–ға қойылатын жалпы талаптар	16
1.3.3	АЖО–ды жобалау сатылары.....	17
1.3.4	Қабылдау комиссия жұмысының жалпы сипаттамасы.....	18
2	Ақпараттық жүйелер. Деректер қорын жобалау.....	19
2.1	Ақпараттық жүйелер туралы қысқаша мағлұматтар.....	19
2.1.1	Ақпараттық жүйе жобалау сатылары.....	20
2.1.2	Ақпараттық жүйенің тіршілік циклі	21
2.1.3	Ақпараттық жүйелер ұғымы және олардың маңызы	23
2.2	Деректер қорын жобалау.....	26
2.2.1	Деректер моделі.....	26
2.2.2	Реляционды деректер қорына қойылатын талаптар	28
2.2.3	SQL сұраныс тілі.МҚБЖ үшін интерфейсті стандарттау	29
2.2.4	Қосымшаларды өңдеу кезінде SQL тілінің қолданылуы	31
3	Дерекқорды жобалау және оны іске асыру	33
3.1	ДҚ құрудың бастапқы кезеңі	33
3.2	Концептуалды жобалау.....	33
3.2.1	Қажетті ақпарат. Пайдаланушылардың талабы	33
3.2.2	Дерекқор және қосымшаны жобалау құралдарын таңдаудың негізі	34
3.3	Бағдарламаның ER моделін жасау.....	35
3.3.1	Бизнес ережелер.....	36
3.3.2	Реляционды қатынастарды құрастыру	39
3.3.3	Логикалық жобалау.....	41
3.4	Физикалық жобалау	42
3.4.1	Дерек қорына қосымша құру	42
3.4.2	Бағдарламаның негізгі алгоритмі	42
3.4.3	Администратор бөлімінің коды	43
3.4.4	Бағдарламаны сипаттау	54
4	Тіршілік қауіпсіздігі.....	63

4.1 Ауа алмасу (кондиционер) жүйесінің құрылғысы және есебі	63
4.2 Кондиционерлеу және ауаны жанарту жүйелерін қарастыру	67
4.3 Температура айырымы нәтижесінде алынатын жылу және жылу жоғалу	68
5 Техника – экономикалық негізделуі	75
5.1 Бағдарламалық қамтаманың артықшылығы мен сипаттамасы.....	76
5.2 Маркетингтік стратегия.....	76
5.3 Бағдарламаның сатылым жоспары.....	77
5.4 Жұмыс сипаттамасы және қажеттілік дәлелдемесі	78
5.5 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі.....	78
5.6 Бағдарламалық қамтаманы өңдеуге кеткен шығын есептелуі	80
5.7 Қосымша шығындар статьясы	81
5.8 Өзіндік құн нәтижесінің	83
5.9 Бағдарламаның бағасын есептеу	84
5.10 Бағдарламаны енгізудегі шартты үнемдеуді есептеу.....	84
5.11 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	86
5.12 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу 3 оқу үшін...	87
5.13 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу АЭЖБУ	88
Қорытынды	89
Пайдаланылған әдебиеттер тізімі	90
А қосымшасы.....	91
Б қосымшасы	92

Кіріспе

Бұл дипломдық жобада АУЭС университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған дерекқор негізінде клиенттік программа жобаланды.

Бұл программа АУЭС университетіне түсу үшін келген адамдардың құжаттарын қабылдап, оларды тандаған мамандықтары бойынша жіктеу, әрі тез болу үшін, барлық деректерді жүйелі түрде сақтау, қажетті ақпаратқа тез қатынас алу, деканаттарды автоматтандыруға арналған .

Дерекқор құру үшін алдымен зерттелетін облысының талдауы жасалады. Келесі кезеңде оқуға түсу үшін қандай мәліметтердің керек екенін біліп алу қажет.

Программа MS SQL Server-жүйесінде жасалады. АУЭС университетінің дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейін, пайдаланушылардың барлық сұраныстарын қанағаттандыратын программа жасалынды. Бағдарлама Microsoft visual studio 2014 C# тілінде жазылды.

Бұл бағдарлама ыңғайлы, әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, аз уақыт ішінде бағдарламаны игеріп, оңай пайдалануға болады.

1 Техникалық тапсырма

1.1 Тағайындалуы және қолдану аймағы

Ұсынылып отырған бағдарлама Алматы энергетика және байланыс университетінің қабылдау комиссиясының жұмысын жақсартуға арналған. Бұл бағдарлама бойынша талапкерлердің барлық мәліметтері таңдап алған мамандығы бойынша іріктеліп, тек қана бакалавриатта ғана емес, магистратурада да және басқару жұмысында да қолдануға болады.

1.1.1 Сенімділікке қойылатын талаптар

Бағдарлама сенімді жұмыс жасау үшін қолданушылар кейбір талаптарды орындау керек:

- бағдарлама белгілі бір лицензиямен қамтылу керек;
- ақпараттарды қорғау үшін ГОСТ 51188–98 [1] ережелері орындалу керек;
- қолданылатын құралдарды үздіксіз электроэнергиямен қамтамасыз ету керек;
- бағдарламалық құралдар әрдайым компьютерлік вирусқа тексеріліп отыру керек.

Осындай ұйымдық – техникалық шараларды орындаған кезде тапсырыс берушілер толық сенімді бағдарламаға ие болады.

1.1.2 Техникалық құрылғыларға қойылатын талаптар

Бағдарлама жүзеге асу үшін :

- операционды жүйе Windows 2000 Server , Windows 2003,XP,7 және Windows 8;
- процессор Pentium–2.4Hz;
- жедел жады 5Гигабайт көлемінен кем емес;
- Microsoft SQL Server 2008 r2 ;
- HDD, 50 Гигабайт;
- Microsoft visual studio 2014 ultimate бар серверлік қызмет көрсететін дербес компьютер қажет.

1.1.3 Бағдарламалық құжатқа қойылатын талаптар

Бағдарлама жүзеге асу үшін оның техникалық тапсырмасы, байқаулардың әдістері, пайдаланушыға көмек бере алатын бағдарламалық құжаттары болу керек.

1.2 Құру кезендері мен сатылары

Бағдарламаны құру бірнеше сатыға бөлінеді. Атап айтқанда, бірінші сатыда техникалық тапсырманы құру керек; екінші сатыда жобалау жұмыстары жүргізіледі; үшінші сатыда бағдарламаны енгізу керек.

Бағдарламаны енгізу бірнеше кезеңнен тұрады. Ол үшін алдымен ақпарат жүйесін құру керек. Содан кейін бағдарламалық құжаттарды құрамыз. Келесі кезеңде бағдарламаның жақсы жұмыс істеуі үшін белгілі бір мысалдарды алып тексеру керек.

Бағдарлама дұрыс жұмыс жасау үшін құру кезеңдерімен сатылары жүйелі түрде орындалуы керек.

1.2.1 Кезең бойынша жұмыстардың мазмұны

Ұсынылып отырған бағдарлама жүйелі түрде қызмет істеуі үшін, ең алдымен тапсырманың мақсатын анықтап алу керек. Содан кейін техникалық құрылғыларға және бағдарламаның өзіне қойылатын талаптарды анықтап алу керек [4].

1.3 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы

Анықталған маманның атқаратын функцияларының автоматтандыру үшін керек құралдармен жабдықталған жұмыс орнын автоматтандырылған жұмыс орны (АЖО) немесе шетел терминологияда «жұмыс станса» (work-station) деп атайды. АЖО-ның басқа да қосалқы құрылғылармен қамтамасыз етілуі керек. Олардың қатарына дискілік жинақтауыштар, басу құрылғылары, оптикалық оқитын құрылғылар, штрих кодасын оқитын құралдар, графика құрылғылары, басқа АЖО-мен және басқа жергілікті есептеу желілерімен т.б. түйіндесетін құрылғылармен толықтырылатын әдетте дербес компьютер болып табылады.

Көбінесе қолданымда жүрген бағдарламалардың концепциясына сай ақпарат көлемдері ұйымдық иерархияда қолданылуына байланысты ақпараттарды төменнен жоғары қарай жіберуге арналған. Ақпараттық жүйенің дұрыс қызмет істеуі үшін өңделген мәліметтер мен бастыпқы деректер жергілікті дерекқорларда сақталуы керек. Осы мақсатта ақпаратты жүйенің кезкелген деңгейі үшін автоматтандырылған жұмыс орындарын құру талап етіледі [2].

Әрбір маманның өз дерекқорларын қолданумен қатар, басқа желілермен, керек болса, ЭЕМ-дары бар ауқымды есептеу тораптарына қосылу мүмкіндігін қамтамасыз ете алатын жұмыс орны АЖО-ның негізгі арналуы деп аталады.

Шартты түрде АЖО–да шешілетін мәселелерді ақпараттық және есептеуіш деп екі топқа бөлеміз.

Бірінші топқа ақпараттарға кодтарды бекіту, оларды жіктеу, топтастыру, түзету және сақтау жатады. Көбінесе олар арифметикалық немесе мәтіндік сипатта болып, қарапайым есептеу мен логикалық процедураларды қажет етеді. Дегенмен бұл топтағы мәселелер күрделі болғандықтан, мамандардың көп уақытын алады [4].

Екінші топқа формальдандырылатын және жартылай формальдандырылатын есептеу мәселелері жатады. Өз кезегінде формальдандырылатын мәселелер формальды алгоритм негізінде шешіледі де, тура есептеу мәселелері мен математикалық модельдер негізіндегі мәселелерден тұрады. Олардың біріншісі қарапайым алгоритмдер көмегімен шешіледі десек те, күрделілері математикалық модельдерді қолдану арқылы шешіледі. Қазіргі уақытта осындай мәселелерді шешу үшін семантикалық амалдар экономикалық объектілерді басқару үшін, кейбір кездерде ақпараттар толық болмаған жағдайда кеңінен қолданылып жүр.

1.3.1 АЖО құрылымы

АЖО ұйымдастыру, функционалды және қамтамасыз ету бөліктерінен тұратын кешенді жүйе болып табылады.

АЖО-ның ұйымдастыру бөлігі негізінен қолданушылардың атқаратын қызметтерін жіктеп таратуға арналған. Сонымен қатар оның құрамына мамандар даярлау, олардың әрі қарай өсуін қамтамасыз ету, оны ұйымдастыру жатады. Осы процестерді басқару үшін жұмысты жүйелі түрде жоспарлау, тіркеу, бақылау, сараптау, реттеу керек. Оған қоса қолданушылардың атқаратын міндеттерін олардың құқықтарын құжатты түрде рәсімдеу керек.

АЖО–ның екінші бөлігі экономика-математикалық тәсілдер кешені. Ол жоспарлау шешімін, техника-экономикалық көрсеткіштердің тіркеуін және сараптауын жүзеге асырады. Белгілі бір болжамдар бойынша, осы бөлімі арқылы АЖО-ның функцияларын орындайтын жүйе бірнеше тәуелсіз ішжүйеден тұратыны анықталған.

Әрбір жұмыс орны мен ұйымдық құрылымдықтың атқаратын қызметтеріне байланысты қолданылатын ішжүйелерді ерекшелендіреді.

АЖО–ның үшінші бөлігі қамтамасыз ету бөлігі болып табылады да, оның құрамына:

- функционалды ішжүйелердің ақпараттық қамтамасыз етуі (АҚЕ);
- техникалық құралдар кешені (ТҚК);
- ішжүйелер функцияларын жүзеге асырудың программалық құралдары кіреді.

Ақпараттық қамтамасыз ету құжаттардан, ақпарат массивтерінен (ДК) және кодалаудың стандартталған және бәрыңғай формалар жиынтығынан тұрады.

Барлық ақпараттар бірнеше белгілер бойынша, атап айтқанда, тұрақтылық, арналуы және қозғалу бағыты бойынша жіктеледі (Кесте 1.1)

Кесте 1.1 – Ақпараттардың жіктелуі

Жіктелуі	Түрлері
Тұрақтылық бойынша	шартты тұрақты және айнымалы тұрақты
Арналуы бойынша	нормативті-анықтамалық, жоспарлау, есептемелік, сараптамалық және оперативті-өндірістік
Қозғалу бағыты бойынша	кіріс, шығыс және аралық

Енді АЖО–ның техникалық құралдар кешеніне (ТҚК) келетін болсақ, ол мәліметтерді жинау, жеткізу, өңдеу және сақтау үшін қолдатылатын құрылғылардан тұрады.

Біртекті операцияларды орындайтын құрылғылар техникалық құралдардың функционалды топтарын құрайды.

Қамтамасыз етудің үшінші бөлігіне бірінші кезекте орындайтын адамды, программалық тіл құралдарына арнап, «жұмыс ортасын» қалыптастыратын операциялық жүйелерді жатқызамыз. Ал АЖО–ның программалық тіл құралдары оның функционалды мәселелерін шешуге, оны деңгейлі басқаруға арналған интерфейстерден тұрады [8].

1.3.2 АЖО–ға қойылатын жалпы талаптар

АЖО – ға қойылатын талаптар сурет 1.1 көрсетілген.



Сурет 1.1 – АЖО – ға қойылатын талаптар

АЖО кез келген маманға арналған құралдар мен программалық камтамасыз етудің жиынтығы және әр уақытта арнаулы сипаты бар жүйе. Ол бухгалтерге де, заңгерге де, студентке де, мұғалімге де, дәрігерге де арналуы мүмкін.

АЖО ендіру өңдеу уақытын үнемдейді, жұмысты жеңілдетеді, машина мен адам арасындағы байланысты тездетеді.

АЖО еңбек өнімділігін мен басқарудың тиімділігін арттырып қана қоймайды, сонымен қатар өте қолайлы.

АЖО жеке (дербес) түрде де , топтық түрде де болады. Егер топтық не коллективті АЖО болса, тиімділікті арттыру үшін мамандардың нақты атқаратын міндеттерін анықтау керек. Сонымен қатар АЖО мамандарға да белгілі талаптар қояды,мәселен, олар икемді, үнемі даму мен жетілуге бейімді болулары қажет.

Осы жүйеде жұмыс істеу үшін олар ақпаратты өңдеу машина құралдарын жақсы білулері;диалогты режимде жұмыс жасай алулары; компьютерді жақсы меңгеруі; олардың жақсы жұмыс жасауына кәсіпорын тұрғысынан белгілі бір жағдайлар жасалуы, атап айтқанда, еңбекақыны көбейту, жұмысшылардың біліктілігін арттыратындай мүмкіншіліктер болуы керек.

1.3.3 АЖО–ды жобалау сатылары

АЖО–ның жақсы қызмет етуі үшін басқару жүйесін жақсылап зерттеп, сараптама жасау керек, оны құру үшін техникалық тапсырманы дұрыс құрастыру керек, сонымен қатар АЖО-ны жоғары деңгейде техникалық

жобалап қана қоймай, жұмысты жүйелі түрде жобалау керек. Осы сатылардың жақсы орындалуы АЖО-ның қызметін жақсартады.

1.3.4 Қабылдау комиссия жұмысының жалпы сипаттамасы

Орта білімі бар жас талапкерлердің барлық құжаттарын тез тексеріп, қабылдап, қалаған мамандықтары бойынша жіктеп көрсету керек. Администратор қабылданған құжаттарды және қабылдау комиссияның жұмысын тексеріп, керек жағдайда жаңа комиссия мүшесін енгізе алады. Ұсынылатын бағдарламада әр факультеттің өз қабылдау мүшесі болады.

2 Ақпараттық жүйелер. Деректер қорын жобалау

2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар

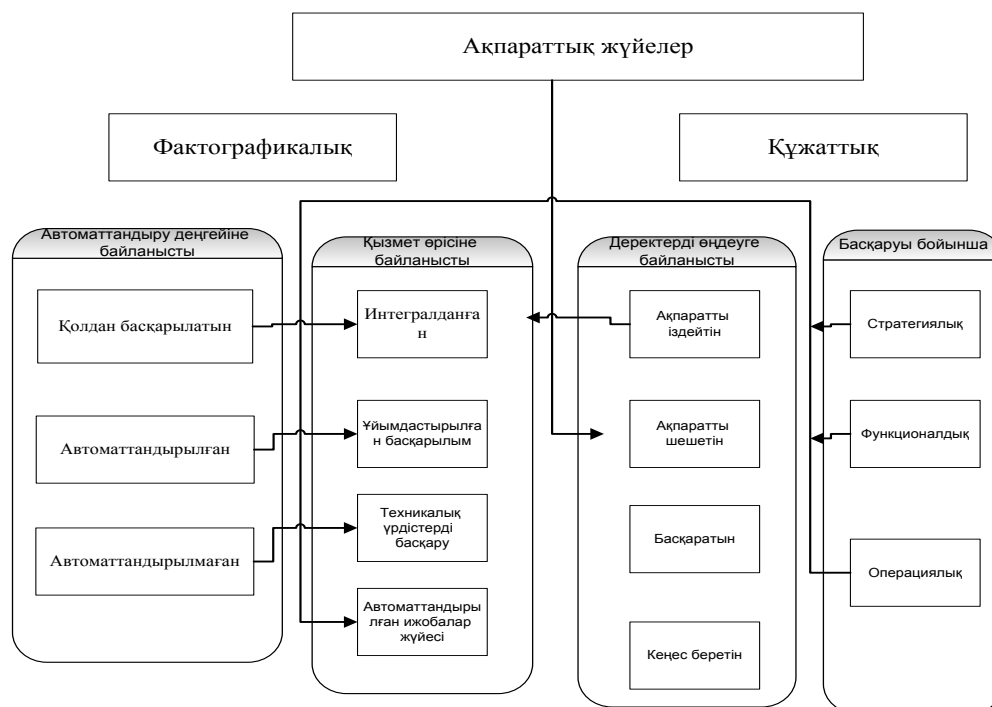
Адам қызметіне арналған ақпараттық модельді ақпараттық жүйе деп қабылдаймыз. Әр жүйе сақтау, ақпарат ауысу, ақпаратты түрлендіру үдерістерінен тұрады. Және де бүгінгі таңда ақпараттық жүйелердің маңызы өте зор.

Ақпарат жүйесінің жіктелуі негізінде олардың арқасында шешілетін тапсырмалардың өзіндік ерекшеліктері жатыр.

Ақпараттық жүйелер бірнеше белгілер бойынша жіктеледі. Олардың негізгілері ақпаратты жүйелердің функционалдық мүмкіндігі мен құрылу ерекшеліктері, орындайтын тапсырмаға, қолданатын техникалық құралдарға байланыстылығы.

Ақпараттық жүйелер фактографикалық және құжаттық болып деректерді сақтау факторы бойынша жіктеледі. Олардың бірінші түрі сандық және мәтіндік деректерді сақтау мен өңдеу үшін қолданылса, екіншісінде деректер тек құжат түрінде болып келеді. Соңғы түрінде мәліметтер пайдаланушыларға беріледі, ал оларды өңдеу бұл жүйелерде қарастырылмайды.

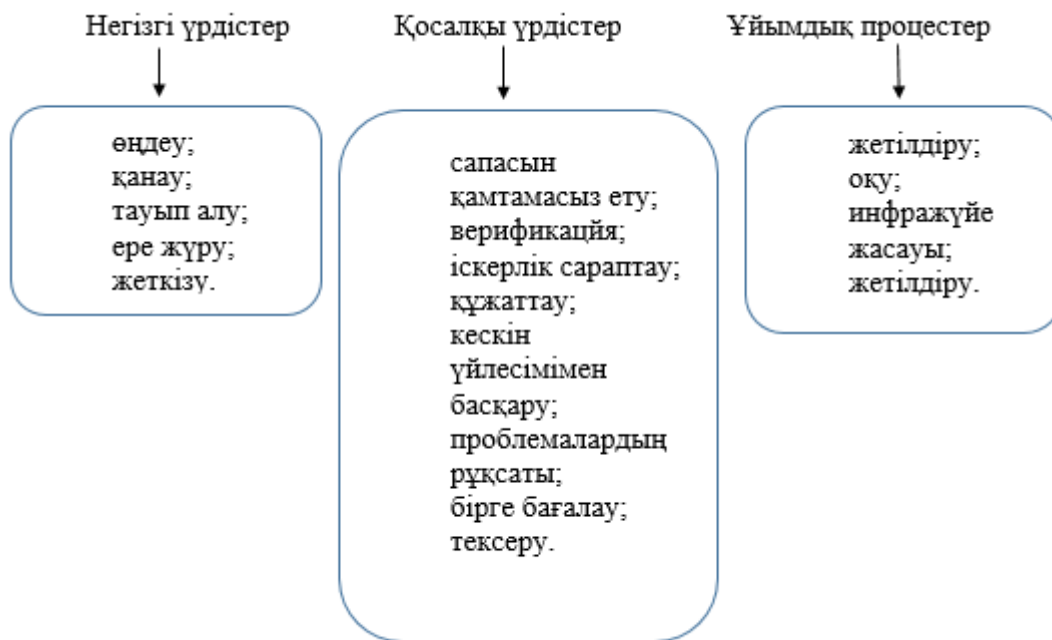
Барлық процесстердің автоматтандырылғанына байланысты ақпараттық жүйелер қолмен басқарылатын, автоматтандырылған және автоматтандырылмаған болып бөлінеді (Сурет 2.1).



Сурет 2.1 – Ақпараттық жүйелердің бөлінуі

2.1.1 Ақпараттық жүйе жобалау сатылары

ISO/IEC 12207 стандартқа сәйкес ақпараттық жүйені жобалау бірнеше үрдістерден тұрады. Олар негізгі, қосалқы үрдістерден және ұйымдық процестерден тұрады (Сурет 2.2).



Сурет 2.2 – Ақпараттық жүйе сатылары

Ал күрделі жүйені жобалау өте ауыр шығармашылық жұмыс болып есептеледі, себебі онда қолданылатын ұйымдасқан шығармашылық технология фундаментальдік зерттеу мен дайындау және қолдану жүйелерінің жиынтығын қажет етеді.

Ұйымдасқан шығармашылық технологияның бірінші сатысында пәндік облысты зерттеу керек.

Қолданымда жүрген бірқатар жүйелер иерархиялық құрылымға ие. Оларға координация – жүйенің негізгі мақсатына жетуге бағытталған арнайы элементтің жиынтығын қамтамасыз ету; арнайыланған жеке жұмыстарды белгілеу және оны арнайы дайындалған элементтермен бөлек беру жүйенің ерекшелігі тән.

Иерархиялық жүйені қолдану арқылы жеке шешім қабылдайтын элементтердің өзара байланысын жүйелі түрде анықтауға болады.

Ал қолдану кезіндегі қиыншылықтарды, қайшылықтарды шешуге пайдаланатын ережелер жиынтығын стратегия деп атаймыз.

Күллі ақпараттық жүйе қабылдау сатысында жүйелік жақындауды қолданады. Бұл кезде жүйелік модель арқылы іске асатын сан алуан саладағы, типтегі, кластардағы ұйымдардың және пәндік облыстардың зерттелуі және

өңдеуі қарастырылады. Мамандардың есебі бойынша, жүйелік жақындауды жоққа шығару осы жүйеге енудің жақсы жолы. Ол әбден қолданыста бірнеше мәрте тексерілген қарапайым зерттеу әдісіне қайшы.

Күрделі жүйе коммерциялық әдіспен де жасалмайды. Нарық заманында ұсынысты зерттеу және алдын ала өңдеу процестеріне ғана бәсекелестіктің қатысы бар. Ал қалған процестерге, атап айтқанда, техникалық және жұмысты жобалауға белгілі бір кәсіпорынмен келісімшартқа отырғаннан кейін ғана көшеміз. Бағдарламаны тапсырыс беруші мақұлдағаннан кейін басқа нарыққа ұсынылмайды. Осы себептен де АЖ жүйелік талдау сатысына көп көңіл бөлінеді. Мысал ретінде АҚШ – тың қорғаныс министрлігі 1965 жылы №32009 директивасын шығарған кезде құрылым концепциясы сатысынсыз күрделі жүйені өңдеу туралы келісім қабылдауға болмайды деп көрсеткен.

Белгілі бір заттың, құбылыстың, процестің түсіндіріліп жеткізілуі концепция түсінігін береді. Ал кейбір сатыларда өзара байланысқан білім, факт, тапсырма мен қызығушылықтар жинақталады.

Көбінесе мұндай жағдай адам санасы мен даму процестері жоғары облыстарда көп кездеседі; соның ішінде тез жетіліп келе жатқан облыстарда осындай үрдіс дағдарысқа соқтырады.

Кез келген жүйеге барлық компоненттерінің өзара байланысын анықтайтын модельдің болуы, көбінесе жеке компоненттерінің қасиеттерін жүйенің қасиетімен байланыстыратын модельдің болуымен ерекшеленулері тән.

Әр жүйенің шешімі теңдік жүйесін құру арқылы орындалады. Теңдік жүйесі түрлі компоненттер қасиеттерінің бір – біріне және жүйе қасиетіне әсерін көрсетеді. Жүйелік түрде кез келген компонент жеке қасиетін өзгерте алмайды, себебі ол өзгеріс басқа компоненттердің өзгеруіне әкеліп соғады. Өз кезегінде ол ішкі жүйеге кері әсерін тигізеді, ал кейбір жағдайларда жалпы жүйенің бұзылуына әкеледі.

Сондықтан да тапсырманы тек теңдік жүйесін құру арқылы шешуге болады. Жеке компоненттер белгілі бір жүйеге арналып жобаланып, жасалады. Арнайы жасалған компоненттерді қолдану арқылы түскен табысты компонентті құрғандарға бөліп беруді қамтамасыз ету үшін нарық керек және нарыққа сәйкес келетін жүйе жеткілікті үлкен болуы қажет.

Тапсырылым бойынша құрылған ақпараттық жүйені жасау үшін рационалды жоспарлау және ұйымды жұмыстардың үйлесімімен ерекшеленетін тіршілік циклдары керек. Оның бірнеше модельдері:

- аралық бақылаумен жүретін тарауланған модель;
- каскадты;
- спиральды модель [7] бар.

2.1.2 Ақпараттық жүйенің тіршілік циклі

Жүйені құру үрдісі мен оны пайдалану кезіндегі болатын оқиғалар түрінде ақпараттық жүйенің тіршілік циклін көрсетуге болады.

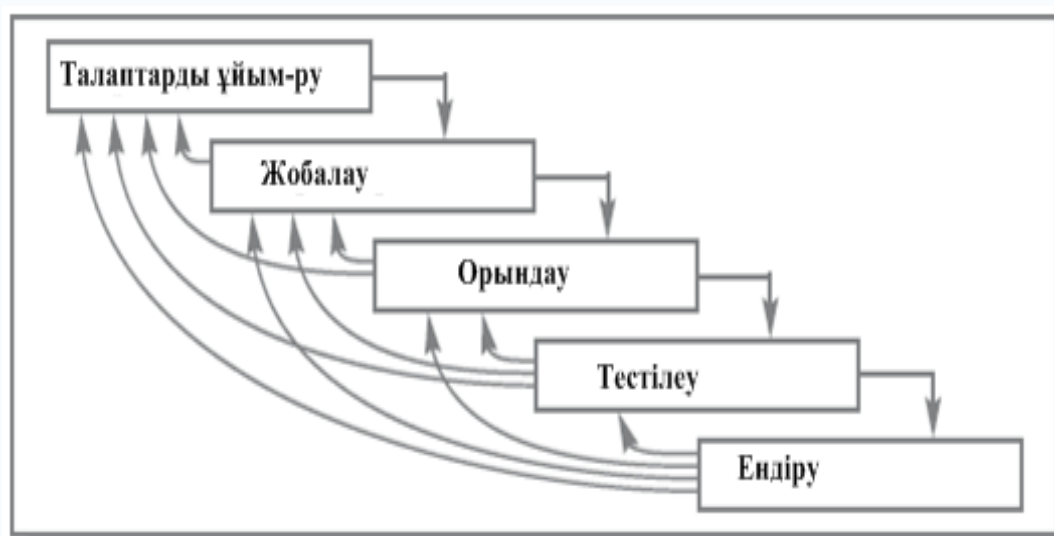
Тіршілік циклі арқылы жүйенің бастапқы қажеттілігенен бастап толық пайдалануынан тыс қалғанша дейінгі күйін көруге болады.

Тіршілік моделінің қолданыста аралық бақылаумен жүретін тарауланған моделі (Сурет 2.3), каскадты модель (Сурет 2.4), спиральды модель (Сурет 2.5) сияқты түрлері қолданылады:

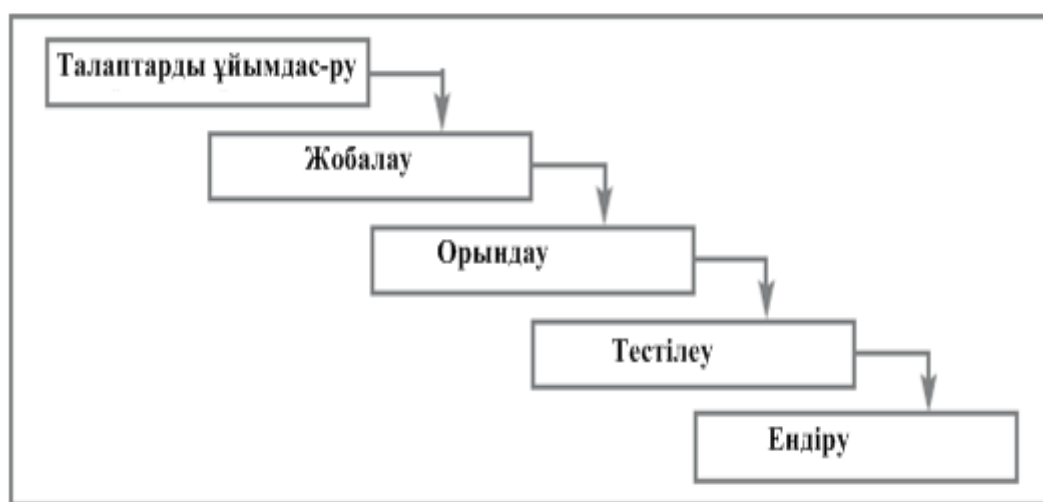
– Біріншісі ақпараттық жүйенің құрылуы, кезеңдер аралығымен кері байланыс циклының интеграциясымен басқарылады.

– Екіншісі жобаның бекітілген ретімен барлық кезеңдерінің орындалуын қамтамасыз етеді. Келесі кезеңге өту алдындағы кезеңдегі жұмыстын толықтай аяқталуын білдіреді;

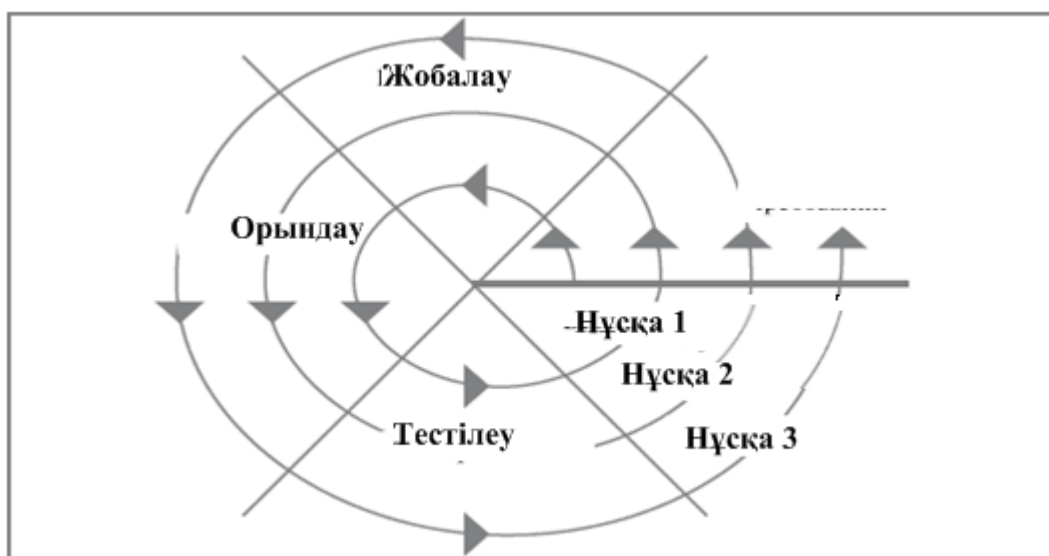
– Үшіншісінде әр спиральдың орамында өнімнің келесі нұсқасы құрылады, жобаға талаптар қарастырылады және оның сапасы анықталады.



Сурет 2.3 – Аралық бақылаумен жүретін тарауланған модель



Сурет 2.4 – Каскадты модель



Сурет 2.5 – Спиральды модель

Күнделікті тіршілікте осындай циклдардың 1970-1985 жылдар аралығында каскадты, ал 1986 жылдан бастап спиральды модельдің қолданылып жүргенділігін атап көрсеткіміз келеді.

2.1.3 Ақпараттық жүйелер ұғымы және олардың маңызы

Пайдаланылатын ресурстардың көптігі мен операциялық ортаның жылдам өзгеретінін ескере отырып, қазіргі заманғы ұйымды тиімді басқару өте күрделі мәселе болып табылады. Басқарудың ерекше қызметіне ұйым қызметінің көпөлшемді әр түрлі салаларында жүзеге асырылатын жоспарлау, ұйымдастыру, ойдағыдай іске асыру, үйлестіру, бақылау мен талдау жатады. Жоғарыда атап өтілген атқарымдарды орындау барысында қалыптастырылатын басқару шешімдері нақты орындаушылар үшін бастапқы мезет болып табылады. Нарық жағдайында осы экономикалық объектінің біртұтас ақпараттық кеңістігін құрайтын толық, шынайы және уақытылы мәліметті иеленіп бәсекелік артықшылыққа жетуге болады. Мұндай кеңістікті тек басқару шешімін қолдайтын автоматтандыру құралдарын тартып электрондық құжат айналымының базасында құруға болады.

Жалпы экономика мен оның жекелеген құрамдастарын (кәсіпорындар, компаниялар, мекемелер мен т.б.) динамикалық жүйеге жатқызуға болады. Мұндай жүйелердің жұмысы өзгеретін сыртқы ортаның әсерімен және өте көп ақпаратты өңдеумен байланысты.

Жүйе – белгілі бір мақсатқа жету үшін атқарылатын жергілікті өзара байланысты құрамдастар жиынтығы.

Жүйені сипаттау үшін келесі ұғымдар пайдаланылады:

- құрылым (көп элементтер мен олардың арасындағы өзара байланыстар);

- кіріс пен шығыс (жүйеге кіретін және одан шығатын материалдық, қаржы және ақпараттық ағындар);
- мінез-құлық заңдары (жүйенің кірісі мен шығысын байланыстыратын қызметтер);
- мақсаттар мен шектеулер (бірқатар өзгермелі жүйенің жұмыс істейтін үдерістерді; жекелеген өзгермелілерге әдетте шектеу салынады).

Басқару – қойылған мақсаттарға жетуге жол ашатын жүйенің жай-күйінің өзгеруі.

Жүйені басқару үдерісі басқару мақсатымен, қоршаған орта және ішкі жағдайлармен анықталады.

Кибернетика тұрғысынан осы үдеріс жүйенің элементтеріне қол жеткізіп, әсер ету үшін жасалады және оны сыртқы ортаны, объект пен басқару аппаратын байланыстыратын ақпараттық үдеріс түрінде көрсетуге болады.

Жүйені басқару үдерісі негізделетін ақпаратпен алмасуда келесі рәсімдер циклімен жүзеге асырылады:

- басқарылатын объектінің ағымдағы жай-күйі жөнінде ақпарат жинау;
- алынған ақпаратты талдау мен объектінің ағымдағы жай-күйін қажет жай-күймен салыстыру;
- басқарылатын объектіні қажет жай-күйге көшіру мақсатында басқаратын іс-әрекет ойластыру;
- объектіге басқару әсерін беру.

Компьютерлік техника мен байланыс құралдары дамуының қазіргі кезеңінде басқару үдерістерін автоматтандырудың арқасында АЖ-ны пайдаланушылардың әр түрлі санаттары өздерінің алдына қойылған міндеттерді жылдам әрі тиімді шеше алады. АЖ-ны пайдаланушыларды келесі төрт санатқа бөлуге болады:

- жүйе әкімшісі – бұл жүйені пайдалану және оның жұмыс істеу қабілетін қамтамасыз ету үшін жауап беретін, түпкілікті пайдаланушылардың қажеттілігін түсінетін, олармен тығыз байланыс та жұмыс істейтін және дерекқорды анықтау, жүктеу, қорғау мен оның тиімділігі үшін жауап беретін маман;
- қолданбалы программалаушылар – қолданбалы міндеттерді, деректер қорына қойылатын сұратуларды іске асыратын программаларды әзірлеумен айналысады;
- жүйелік программалаушылар – ақпараттық жүйені қолдауды жүзеге асырады және оның жұмыс істеу қабілетін қамтамасыз егеді, компьютерлік программалық қамтамасыз етуді (операциялық жүйелерді, деректер қорын басқару жүйелерін, трансляторлар, жалпы тағайындалыстағы сервистік программалар) әзірлеумен және сүйемелдеумен айналысады;

- түпкілікті пайдаланушы (ақпарат тұтынушы) – АЖ–ның мүддесінде істейтін адам немесе ұжым. Түпкілікті тұтынушы АЖ–мен күнделікті жұмыс істейді, шектеулі қызметі саласымен байланысты және әдетте программалаушы болып жұмыс істемейді. Мысалы, ол бухгалтердің, маркетингтің, қаржы менеджерінің, бөлімше басшысының лауазымын атқаруы мүмкін.

Автоматтандырылған АЖ–ге мамандардың көп арнайы мамандандырылған жұмыс орындары (АЖО), коммуникация және ақпарат алмасу құралдары, қызметкерлердің жұмысын автоматтандыруға мүмкіндік беретін басқа да құралдар мен жүйелер кіреді.

Қазіргі заманғы автоматтандырылған АЖ–да экономикалық талдау мен басқару шешімдерін қабылдау үшін қажет ақпаратты сақтау, беру және өңдеу жөніндегі ең жаңа компьютерлік технологиялар пайдаланылады; олар ақпарат өңдейтін қазіргі заманғы техникалық және программалық ақпарат өңдеу құралдарымен, әлемдік ақпараттық кеңістікте жұмыс істейтін телекоммуникация құралдарымен жабдықталған.

Экономикалық объектілерді (кәсіпорындарды, банктерді, сауда ұйымдарын, мемлекеттік мекемелерді және т.б.) басқаруға арналған АЖ–ның тиімділігі олардың басқару қызметтерін қамтудың кеңдігі мен біріктірілуіне, басқару шешімдерін жедел дайындауына, сыртқы ортаның өзгеруі мен пайдаланушылардың ақпараттық қажеттілігіне орай бейімделуіне байланысты.

Шетелдік мамандар ақпараттық технологиялар дамуының бес негізгі үдерісін жеке атап өтеді. Оларға қысқаша тоқталайық.

Бірінші үдеріс есептеу–талдау жұмысының нәтижесі мен жеке пайдаланушыға көрсетілетін арнайы қызметтің арасындағы буданға айналып бара жатқан ақпараттық өнімнің өзгеруімен байланысты.

АТ қисынды элементтерінің қатарлас өзара ықпалдасуға, ақпараттың барлық түрін олардың сезім мүшелері арқылы адамның бір уақытында қабылдайтындай етіп үйлесуіне бейімділігі.

Ақпарат көзінен бастап оны тұтынушыға дейінгі барлық аралық буындардың жойылатынын болжау, мысалы, автор мен оқырманның, сатушы мен сатып алушының, әнші мен тыңдарманның, ғалымдардың өзара , оқытушы мен оқушының, кәсіпорындағы мамандардың бейнеконференция, электрондық дүңгіршек, электрондық почта арқылы тікелей тілдесуі мүмкін болады.

Адамдар өзара және ғаламшардың кез келген нүктесінде жер серіктестік байланыс пен бүкіл әлемдік Интернет желісін пайдаланудың арқасында бір–бірімен тілдесе алатын ақпараттық технологиялардың ғаламдану үрдісі белең алууда.

Конвергенция (жақындасу) АТ–ның дамуының қазіргі кезеңдегі үдерісінің ең соңғы белесі ретінде қарастырылады, осының нәтижесінде материалдық өндіріс пен ақпараттық бизнестің арасындағы

айырмашылықтар жойылып, ұйымдар мен корпорациялардың қызмет түрлері әртараптандырылады, өнеркәсіптің , қаржы секторы мен қызмет көрсету саласының әр түрлі салалары бір–бірімен тығыз байланыста болады.

Қорыта келе айтатынымыз, экономикалық объектіні басқарудың кез келген жүйесіне өзінің экономикалық ақпараттық жүйесі сәйкес келеді. Қоғамды ақпараттандырудың қазіргі кезеңдегі деңгейінде экономикалық объектілердің ақпараттық жүйелерінде ең жаңа техникалық, технологиялық, программалық құралдар пайдаланылады. Автоматтандырылған ақпараттық жүйе – бұл компьютерлік және коммуникациялық жабдықтар, программалық қамтамасыз ету, лингвистикалық құралдар, ақпараттық ресурстар, сондай-ақ жүйе қызметкерлері кіретін жәнepайдаланушылардың ақпараттық қажеттіліктерін қанағаттандыру үшін нақты әлемнің кейбір бөлігінің динамикалық ақпараттық моделін қолдауды қамтамасыз ететін кешен деп білеміз.

2.2 Деректер қорын жобалау

2.2.1 Деректер моделі

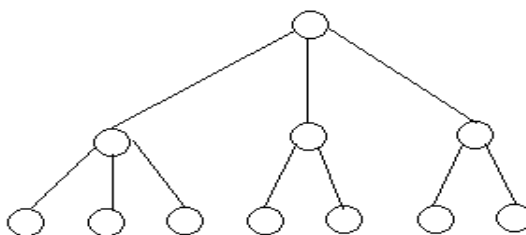
Деректер қорын басқару жүйесінде (ДҚБЖ) деректердің әр түрлі модельдері, атап айтқанда, иерархиялық, желілік, реляционды түрлері, ал кейінгі кездерде белсенді деректер моделі кеңінен өлданылып жүр . Белсенді деректер моделіне постреляционды, көп өлшемді, нысанды–бағытталған.

Осы модельдерге негізделі отырып, қазіргі кезде басқа да деректер моделі құрылуда. Олардың қатарына нысанды–реляционды, деуктивті–нысанды–бағытталған және осылардың негізінде жасалған семантикалық, концептуалдық және бағытталған моделдер жатады.

ДҚБЖның кейбір түрлеріне бірнеше деректер моделі қолданылуы мүмкін. Мәселен, ИНТЕРБАЗА жүйесінде желілік деректер қоры пайдаланушы интерфейсінде SQL және QBE тілдерінен құрылған.

Иерархиялық модельге тоқталамыз.

Бұл модельде деректер арасындағы байланыстар графа (ағаш тәріздес) түрінде болады. Иерархиялық модельдің сызбалық көрінісі келесідей болады (Сурет 2.6).



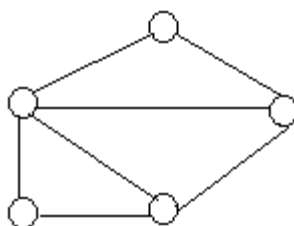
Сурет 2.6 – Иерархиялық модельдегі байланыстар көрінісі

Көрсетілген деректер қорын құру кезде арнайы бағдарламалау тіліндегі бірнеше подтиптерден құралған ағаш типі қолданылады. Сол кезде әр ағаштың өзіне сәйкес «тамырлары» (түбірі), әр тамырдан таралатын құрылымдық ағаш тәрісдес типтері де болады.

Сипатталып отырған модельдің артықшылығы ретінде ЭЕМ жадысын тиімді қолдану және қолданатын операциялардың жылдамдылығын көрсетуге болады.

Желілік модель.

Бұл модель деректер элементтерінің өзара байланысын әр түрін көрсетеді. Желілік модельдің сызбалық кескіні келесідей болады (Сурет 2.7).



Сурет 2.7 – Желілік модельдегі байланыстар

Екінші модель иерархиялық модельмен салыстырғанда байланыстардың әр түрлілігімен сипатталады. Осы фактор оның біріншімен салыстырғандағы артықшылығы болып табылады.

Реляционды модель.

Бұл модельді IBM фирмасының қызметкері Эдгаром Коддом ұсынған. Модельдің негізінде қатынас (relation) ұғымы жатыр.

Қатынас, ол элементтердің жиынынан (кортеж) тұрады.

Кесте жолдар (жазбалар) мен бағандардан тұрады. Кесте жолдарына кортеж сәйкес келсе, бағанға қатынас атрибуттары сәйкес келеді.

Бұл модель өте қарапайым, мүмкін сондықтан да іс жүзінде жиі қолданылады және де деректерді өңдеу жылдам орындалады.

Осы модельге төмендегілер жатады: dBaselll Plus және dBase IV (Ashton–Tate фирмасы), DB2 (IBM), R:BASE (Microrim), FoxPro және FoxBase (Fox Software), Windows (Borland) үшін Paradox және dBASE for, Visual FoxPro және Access (Microsoft), Clarion (Clarion Software), Ingres (ASK Computer Systems), Oracle (Oracle) және Microsoft SQL Server, сонымен қатар ресейлік реляционды ДҚБЖ: ПАЛЬМА (ИК АН УССР) және HyTech (МИФИ).

Егер реляционды ДҚБЖ нысанды–бағытталған жүйелердің кейбір қасиеттерін қолданса, онда ол нысанды–реляционды деп аталынылады. Оның мысалы ретінде Oracle 8.x. алуға болды. Ал, оның алдындағы нұсқасы, яғни Oracle 7.x таза реляционды болып табылады. Microsoft SQL Server 2005, PostgreSQL, сол сияқты Sav Zigzag, IBM Cloudscape, FirstSQL бұлардың бұлардың бәрі нысанды–реляционды болып табылады.

Постреляционды модель.

Реляционды модельдің кеңейтілген түрі ретінде постреляционды модельді қарастырамыз. Осы модель арқылы кестеде сақталған деректерді бөлуге болады, өріс көп мәндік бола алады, яғни бір мән бірнеше мәндерден құралуы мүмкін, ал жаңа кестелер негізгі кестеге бағынады. Реляционды кестелердің жиынын бір постреляционды кестемен көрсету мүмкіндігі, ақпараттық жоғары дәрежеде көрсетілуі және оны өңдеудің тиімділігін осы модельдің артықшылығы деп көрсетуге болады.

Оның ұатарына ДҚБЖ uniVers моделін жатқызамыз.

Көп өлшемді модель.

Бұл модель, реляционды модельмен бірге пайда болса да, ол реляционды модельмен салыстырғанда аз қолданылған. Осы модель ақпаратты интерактивті өңдеуге мүмкіндік беретіндіктен, тек 90 жылдардың ортасынан бастап оған қызығушылық артқан. Оның артықшылығы ретінде нысандардың өзара күрделі байланыста болуын атасақ болады.

Көп өлшемді модельге мына жүйелер жатады: Essbase (Arbor Software), Media Multi-matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems).

Нысанды-бағытталған модель.

Нысанды-бағытталған модельде деректердің бөлек жазбаларын идентификациялау арқылы оларды өңдейтін функцияларды анықтай алады. Ал кемшілігі, сұраныстардың жай орындалуында.

Нысанды-бағытталған модельдер 90 жылдары тәжірибе жүзінде қолданыла бастады, қазіргі кезде оның мына түрлері : ДҚБЖ: POET (POET Software), Jasmine (Computer Associates), Versant (Versant Technologies), O2 (Ardent Software), ODB-Jupiter («Интелтек Плюс» ғылыми-өндіріс орталығы), Iris, Orion және Postgres кеңінен өолданылады.

2.2.2 Реляционды деректер қорына қойылатын талаптар

Реляционды деректер қорына қойылатын талаптардың негізгісі - деректердің қалыптастырылуы. Деректерді қалыптастырудың алты формасы бар, негізінен алғанда тәжірибеде деректер алдыңғы үш қалыптың талаптарын қанағаттандыратын болса жеткілікті және қалыптастыру кестелердегі деректердің қайшылықтары және үйлесімсіздігі болмауы тиіс.

Деректер базасының теориясы қалыптастыру принципіне негізделінеді. Ал, қалыптастыру нысаны бойынша функционалдық тәуелділік ұғымы қолданылады:

– R – қатынасының A атрибуттарының әрбір мәніне B атрибутының тек бір ғана мәні сәйкес болса, онда $B \rightarrow A$. B атрибуты A атрибутына функционалды тәуелді;

– егер $A \rightarrow B$ -ға, $B \rightarrow A$ тәуелді болса, онда A және B атрибуттары өзара функционалды өзара тәуелді;

– егер кілт емес атрибуттар құрама кілттің бір бөлігіне ғана тәуелді болса, онда ол кілтке жартылай тәуелді деп аталады;

– егер R қатысының кілт емес B атрибуты. A атрибуттар тобына, яғни құрама кілттің барлық атрибутына тәуелді болса және ешбір жартылай функционалдық тәуелділік болмаса, онда B атрибуты A атрибуттар тобына толық тәуелді;

– егер R қатынасында $A \rightarrow B$ -ға, $B \rightarrow C$ -ға тәуелді болса, онда C атрибуты A атрибутына транзитивті тәуелді.

ДҚБЖ, SQL бағдарламалау тілімен жүзеге асады. Енді SQL бағдарламалау тіліне қысқаша тоқталайық.

2.2.3 SQL сұраныс тілі.МҚБЖ үшін интерфейс ті стандарттау

МҚ–мен өзара әрекеттесу үшін қолданылатын SQL Server сұраныстар құру тілі 1970-жылдардың ортасында пайда болады.Тілдің шығу атауы–SEQUEL сөзінен шыққан.Бұл тіл реляциялық МҚ–ғы сұраныстардың қалыптасуын түсінікті және ыңғайлы бейнелеуге бағытталған.Негізінде,іс жүзінде толық МҚ–ның тілі болып есептеледі.Реляциялық МҚ–ның негізгі екі сұраныстар тілі болып реляциялық алгебра тілі және реляциялық санау тілі есептеледі.SQL Server сұраныстар құру тілінің мүмкіншілігі мол. SQL Server-дің интерактивті немесе қолданушымен орнатылған қосымша түрінде ҚПП орналасады.

Оның құрамында:

- мәліметтерді анықтау мүмкіндігі;
- мәліметтерді таңдаудағы сұраныстар,SELECT;
- мәліметтерді модификациялау;
- мәліметтерді басқару мүмкіндіктері;
- арифметикалық амалдар орындау;
- кестедегі жолдар мен бағаналарды реттеу;
- МБ-ның көлемін ұлғайту;
- кестенің сұраныстағы мазмұнын еске сақтай білу және шығару;
- мәліметтерді агрегаттау,яғни амалдарна қатысты топтастыру.

Сұраныстарды құру операторлары мен МҚ манипуляциялаудан басқа тілдің құрамында:

- МҚ сұлбасын анықтау құралдары және сұлбамен манипуляциялау;
- бүтіндікті шектеу және триггерлерді анықтау оперторлар;
- МҚ ұсынуларын анықтау құралдары;
- транзакцияларды басқару құралдары.

Қазіргі кезде кеңінен тараған SQL тілінің реляциялық мәліметтер қоры тілінің тарихы 25 жылдан асқанын көрсетеді.Ең бірінші SQL тілінің функционалды толық,бірақ семантикалық және синтаксистік толық емес нұсқауы жасалып,эксперименталды жүзеге асыруда SYSTEMR МҚБЖ-де

қолданылып жүрді (1974–1979 жж. Сан–Хосе жобасы, Калифорния қаласының IBM компаниясының ғылыми іздену лабораториясында жасалды).

SQL тілінің атауы азғандай ғана мағына береді. Әрине, тұтынушыларға бұл тіл реляциялық мәліметтер қорын жобалау оңай болу үшін енгізілген. Мұнымен SQL–ді білгендіктен МҚ–да кез келген әрекетті жасай аламыз деп түсінеміз. Сұраныс және МҚ басқарудан басқа операторлар бар:

- МҚ сұлбасын анықтау және оны басқару;
- бүтіндік шегі;
- МҚ анықтау құралы;
- транзакцияны басқару құралы.

Басқа тілмен айтқанда, SQL тілі мәліметтердің реляциялық моделін толық көрсетуге күші жететінін айқындайды, ол дегеніміз, Код терминінде реляциялық мәліметтер қорының барлық аспектілерін көрсетуге болады.

Сонымен қатар, көп жылдық тарихына қарамай көптеген реляциялық МҚБЖ қолдануды жаңадан үйренушілерге SQL тілі әр алуан болып келетіндігіне сену қиын. Бұл сұраққа әрі оңай жауап беру қиын.

Шын мәнісінде бұл программалау тілінің әр алуан түрін жүзеге асыруға жатады. Стандартқа сәйкес келетін, кез келген компиляторға жүретін халықаралық Си тілі бар. Ашық стандартты жүйемін кездесуі қолданбалы программаларды шығаруға мүмкіндік береді.

1989 ж. Алғашқы халықаралық SQL тілінің стандартты шыққаннан кейін, 1982 ж. екінші халықаралық SQL–92 стандарты қабылданды. Осыдан кейін МҚБЖ бейімделген SQL тілі туралы айтуға болатын болды. Кез келген SQL тілді қолдану үшін стандарт тілдерін білу қажет.

SQL стандарттау сатылары.

1989 ж. қабылданған SQL Халықаралық стандартында (SQL/89) МҚ сұлбасымен манипуляциялау және динамикалық SQL бөлімшелері мүлдем жоқ.

1992 жылы наурыз айында SQL/92 стандарты шығарылды. Бұл стандарт біршама толық және барлық іске асыруға керекті аспектілерден тұрады: МБ сұлбасымен манипуляциялау, транзакция және сессияларды басқару, МБ қосылу, динамикалық SQL.

SQL2 стандартының анықталуы аяқталып жатқан уақытта SQL3 триггерлер механизмін және мәліметтердің абстрактілі типтерін қолдану мүмкіншілігі болады деп жорамалданды. Стандартты 1999 жылы қабылданды деп жобаланды.

Орнатылған SQL. SQL System R–де SQL–н оператор орнатуын қолдайтын арнайы операторлар бар. SQL–ді программалау тіліне орнатудың негізгі мәселесі, SQL–реляциялық тіл, сондықтан ол көбінесе жиынтықпен жұмыс жасайды, ал программалау тілдерін скаляр операциялар негізгі болып табылады.

Динамикалық SQL. Интерактивті SQL–ге бағытталған жүйелерді құруды жеңілдету үшін SQL System R–ға транзакцияны орындау кезінде SQL–дің кез келген операторын орындап және компиляциялауға мүмкіндік беретін операторлар орнатылған.

PREPARE операторы SQL операторының динамикалық компиляциясын шақырады. DESCRIBE операторы бұрын PREPARE операторы көмегімен дайындалған SQL операторында көрсетілген ақпаратты алуға арналған.

ORION нысанды–бағытталған МҚБЖ-ның сұраныс тілінде нысандар инкапсуляциясының принципі толықтай сақталады. Тіл синтаксисі SQL–ге бағытталған. Селекциялардың мүмкін предикаттарының жиынтығы өте жетілген. Атап айтсақ, домені үздік класс болып табылатын атрибут үшін қызығушылық білдірген ішкі класс қолданушысының атын сілтеуге болады.

IRIS жүйесінің сұраныс тілі айтарлықтай деңгейде реляциялық парадигмалар әсерінде болып отыр. Тіпті бұл тілдің атауы OSQL оның реляциялық тілмен SQL тығыз байланысын көрсетеді. OSQL – бұл қалыптастырылмаған қатынастармен жұмысқа арналған реляциялық тіл.

O2 RELOOP жүйесі сұраныстарын декларативті тілі. Бұл SQL – бағытталғын синтаксисі бар сұраныстарының декларативті тілі O2 моделіне арналып жасалған нысандар мен мәндер алгебрасына негізделген. RELOOP тілінің ғажайып қасиеті болып O2 моделінің ортақ контекстінде құрылу шынайылығы табылады. Сұраныс әрқашан жиынтық – мәнде немесе тізімді беріледі. Сұраныс нәтижесі нысан, мән-жиынтық немесе мән-тізімді болуы мүмкін.

2.2.4 Қосымшаларды өңдеу кезінде SQL тілінің қолданылуы

SQL/89 стандартындағы сұраныстар құрылымы жайында айту үшін синтаксистік ережелер мәліметтерінен бастаған жөн:

ODBC қолданбалы құрушыларға мәліметтер қорын өңдеу бойынша, қол жеткізіліп отырған мәліметтер типіне, яғни ISAM мәліметтер қоры, мәтіндік мәліметтер (Excel) немесе SQL мәліметтер қоры байланыссыз функционалды мүмкіншіліктерді ұсынуға арналған. Бұл мақсат ODBC –тің әрбір драйверін алдын ала анықталған сәйкес деңгейге жалғау арқылы іске асады. ODBC драйвері болу үшін драйвер ODBC ядросының егжей-тегжейіне үйлесуі керек. Бұл талаптар қосымша құрушысы әрқашан қандай мәліметтерге қатынас жасалып жатқанына байланыссыз сол бір баяғы функционалды мүмкіншіліктерге сенім артуына кепілдік береді. Егер қолданыстағы мәліметтердің форматы негізгі функционалды мүмкіншіліктерді қолдамаса ODBC драйвері бұл функцияларды эмуляциялау керек. ODBC көмегімен кез келген МҚБЖ мәліметтерімен (тіпті мәліметтер қорына тікелей қатысы жоқ мәліметтермен де, мысалы, электрондық кесте файлдарындағы немесе мәндік файлдардағы мәліметтермен), егер олар үшін ODBC драйвері бар болса, манипуляциялауға болады. Әрбір қолданыстағы МҚБЖ үшін өзіндік ODBC драйвері қажет. ODBC жайында айтқанда, ODBC айрықша ерекшелігі ODBC – драйверіне бірнеше стандарттарды білдіретінін атамай кетуге болмайды. Бұл стандарттар осындай драйверде іске асуы тиіс әр түрлі функционалды жағдайда ажыратылады. Барлық драйверлер үшін мәліметтер көзімен байланысу әдісі, қате жайындағы хабарларды алу, сонымен қатар тіркеудің стандартты

интерфейстері ортақ болып табылады. Драйверлер бірыңғайлықты қамтамасыз ету үшін негізгі талаптарды сақтайды. Мәліметтер қорына қол жеткізудің ашық интерфейсі қолданбалы программаларға әр түрлі МҚБЖ–ларға қатынауға мүмкіндік беретін функциялар кітапханасы болып табылады. Сұраныстардың құрылымдық тілі SQL пайдаланылады. Осылайша, қолданбалы программа құрушысы виртуалды мәліметтер қорына арналған программа құра алады және жүктелетін драйверге логикалық мәліметтерді берілген қолданбалы программа қолданып жатқан нақты МҚБЖ немесе жүйе мәліметтеріне айналдыруға мүмкіндік бере алады. ODBC тартымдылығы оның ықшамдылығымен және қолданбалы программа кодымен шартталған. ODBC қолданбалы программа құрушыларға, сонымен қатар драйверлер кітапханасын жасаушыларға арналған стандартты интерфейс болып функцияланады. ODBC архитектурасының төрт негізгі компоненті бар: қолданбалы программа; драйверлер диспетчері; драйвер; көз немесе мәліметтер көзі.

ODBC интерфейсін пайдаланушы қосымша келесі есептерді орындайды: мәліметтер көзімен байланысты сұрайды. Мәліметтер көзіне SQL сұраныстарды жібереді. Сақтау аймағын және SQL сұраныстардың нәтижесіне арналған форматты сипаттайды. Мәліметтерді сұрайды. Қателерді өңдейді. Қателер жайында хабарлайды. Тұрақтылықты немесе транзакция режимінде іс-әрекеттердің шегінуін іске асырады. Мәліметтер көзімен байланысты жабады.

Драйверлер диспетчері ODBC құралдары жиынтығымен бірге динамикалық байланысқан кітапхана (DLL) болып табылады. Ол әр түрлі драйверлерге ODBC функциясына кірудің жалғыз нүктесімен қамтамасыз ете отырып, драйверлерді жүктейді.

ODBC интерфейсінң функциясы жеті топқа бөлінеді:

Тағайындау және тағайындауды болдырмау;

- орта идентификаторы, байланыс идентификаторы, оператор идентификаторы;
- байланыс;
- SQL – операторлардың орындалуы;
- нәтижелердің алынуы;
- транзакциялармен басқару;
- қателер идентификациясы;
- аралас функциялар.

3 Дерекқорды жобалау және оны іске асыру

3.1 ДҚ құрудың бастапқы кезеңі

Жүктемені бөлу жұмысының жалпы сипаттамасы.

АЭЖБУ университетінің қабылдау комиссиясының жұмысын жеңілдетуге арналған бағдарламаның деректер қорын құру керек. Жүктемелерді бөліп, бөлінбеген жүктемелерді көріп, оларға өзгеріс енгізе алады.

Мәселелер және шектеулер:

– құпия сөзсіз кестені өңдеуге және өзгеріс енгізуге мүмкіндіктің болмауы;

– жүктеменің дұрыс бөлінуі;

– есеп беру жүйелі түрде болуы.

Дерекордағы шешілетін мәселелер.

Қабылдау комиссиясының жүктемесін бөлетін орталықтандырылған деректер қорларын тиімді басқаратын жүйесін құру, құжаттарды таңдаған мамандықтары бойынша жүктеу керек, жаңа комиссия мүшесін қосуға мүмкіндіктері болу керек және құжаттарды сұрыптауға мүмкіндігі болу қажет.

3.1.1 Дерекқордың функционалдығын сипаттау

АЭЖБУ университетінің қабылдау комиссия мүшесінің жұмысын жеңілдетеді, бағдарлама арқылы жұмыскерлерді азайтады, құжаттарды қолмен толтырудан босатады бір сөзбен айтқанда қабылдау комиссия жұмысын толығымен автоматтандырады.

3.2 Концептуалды жобалау

Жобаланатын дерекқор және бағдарламалық интерфейс қабылдау комиссия жүктемесін бөлуге арналған. Бағдарламалық қосымшада келесі функциялар іске асырылған:

– ақпараттық функция, жүктеме, бөлінген және бөлінбеген жүктемелер туралы толық ақпаратты береді;

– енгізу функциясы, дерекқорға жаңа ақпарат енгізуге мүмкіндік береді.

3.2.1 Қажетті ақпарат. Пайдаланушылардың талабы

Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары қабылдау комиссия мен администратор болады. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:

- жүйе қолданыста қарапайым болу керек. интерфейс ыңғайлы болу керек;
- жүйе қауіпсіздікті қамтамасыз ету керек;
- жүйе дерекқордың тұтастығын қамтамасыз ету керек;
- жүйе толығымен интеграциялану керек, яғни мәліметтердің артылуы және артық жаңартулар болмау керек.

3.2.2 Дерекор және қосымшаны жобалау құралдарын таңдаудың негіздемесі

Дерекқорларды басқару жүйесі.

Барлық талаптарды қамтамасыз ететін информациондық жүйені жүзеге асыру үшін бірінші кезекте реляциондық дерекқорын және серверлі архитектура клиентін қолданудың анық шешіміне әкелетін орталық деректер сақтаудың әдісін таңдау қажет.

Қазіргі кезде қойылған тапсырмалар талаптарын ДҚБЖ мүмкіншілігіне қарай және тапсырыс берушінің таңдауына байланысты таңдауға болатын бірнеше ДҚБЖ көрсетілген [2,4,7]:

- орнату және қолданудың жеңілдігі;
- кең таралған;
- қымбат серверлік қуатты орнатуды қажет етпейді;
- өндіргіштігінің жақсы көрсеткіші;
- деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдары;
- коммерциялық емес қолдану үшін өнімді сызғышта арзан түрлерінің болуы.

Microsoft SQL Server төмен өндіргіштік серверлік қуаттарға есептелген, сонымен қатар қолданушылық жұмыс станциясында қолданушыға өндіргіштіктің төмендеуінің білінуінсіз жұмыс істеуге мүмкіндігі бар. Сонымен қатар кең таралғандық критерийларына және деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдарының бар екендігіне жауап береді.

SQL компьютерлік дерекқорда сақталатын деректерді өңдеу және оқу үшін арналған құрал болып табылады, және қолданушының дерекқормен өзара қатынас ұйымдастыру үшін қолданылатын программалау тілі болып табылады. Шындығында SQL тек бір белгілі типті – реляционды дерекқормен жұмыс істейді. SQL ДҚБЖ қолданушыға ұсынатын барлық функционалдық мүмкіншіліктерді жүзеге асыру үшін қолданылады, соның ішінде:

- деректерді ұйымдастыру. SQL қолданушыға деректердің көрсетілімі құрылымын өзгертуге, сонымен қатар дерекқордың элементтері арасында қатынас құруға мүмкіндік береді;

– деректерді оқу. SQL қолданушыға немесе қосымшаға дерекқордың ішіндегі деректерді оқуға және олармен қолдануға мүмкіндік береді;

– деректерді өңдеу. SQL қолданушыға немесе қосымшаға дерекқорды өзгертуге, яғни оған жаңа деректер енгізуге, оған қоса бұрыннан бар деректерді өшіруге немесе жаңартуға мүмкіндік береді;

– қатынас құруды басқару. SQL-дің көмегімен қолданушының деректерді оқу және өзгерту және рұқсат етілмеген қатынас құрудан қорғауға мүмкіндігін шектеуге болады;

– деректерді бірге қолдану. SQL параллельді түрде жұмыс істейтін қолданушыларға бір-біріне кедергі келтірмес үшін деректерді бірге қолдануын қадағалайды;

– деректердің тұтастығы. SQL дерекқордың тұтастығын қамтамасыз етіп, оны келісімсіз өзгерістерден немесе жүйенің бас тартуынан қираудан қорғауға рұқсат береді.

SQL – дерекқорды программалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды. Бұл әдістеме қолданушылар жазған бағдарламаларда да, дерекқорлардың қызметтік бағдарламаларында да қолданылады (есеп беру генераторы және деректерді енгізу құралдарында). Осылайша, SQL адамдарға, бағдарламаларға және есептеуіш жүйелерге реляционды дерекқорлардағы ақпараттарға қатынас құруды қамтамасыз ететін қажетті және мықты құралға айналды.

3.3 Бағдарламаның ER моделін жасау

ДҚ құрудың бастапқы кезеңінде алғашқы көрсеткіштердің жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен администратор көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады. АЭЖБУ дерекқоры кесте 3.1 көрсетілген нысандарды қолданады.

К е с т е 3.1 – «АЭЖБУ университетінің АЖО» дерекқорының нысандары

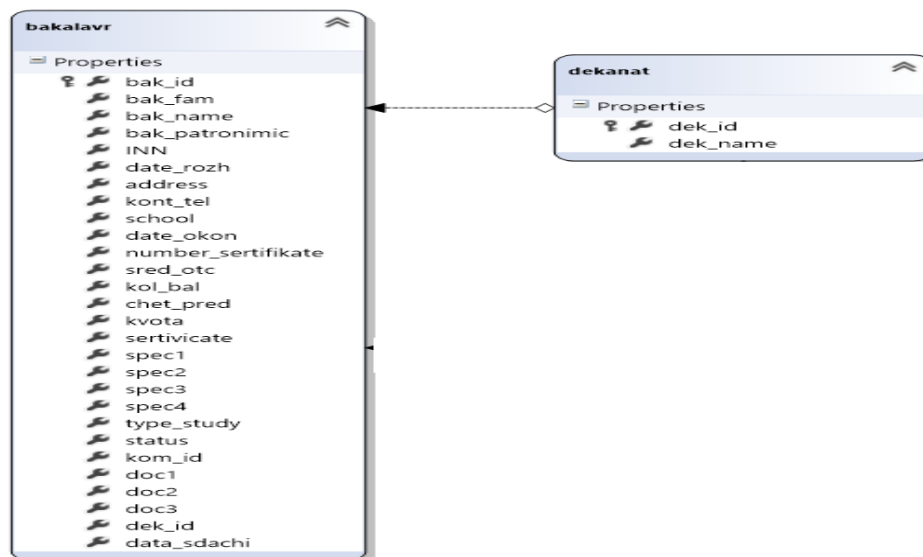
Нысан аты	Сипаттамасы
Bakalavr	Бакалавр
Dekanat	Деканат
Komissiya	Қабылдау комиссиясы
Kafedra	Кафедра
Magistr	Магистратура

Жобалаушы және ақырғы пайдаланушы нысандарды анықтау кезінде келісімге келу керек. Жобалаушы нысандар арасындағы байланыстарды операциялардың сипаттамасына негізделіп анықтайды. Нысандар арасындағы байланыстар операциялардың толық сипаттамасы көрсетілген. Құжаттарды қабылдау комиссиясы тексеруге жібереді. Комиссия мүшесі логинімен кіріп

құжаттарын дұрыстығын тексеріп өткізуге немесе қарастыруға жібереді. Барлық құжаттар администратор бөлімінде болады.

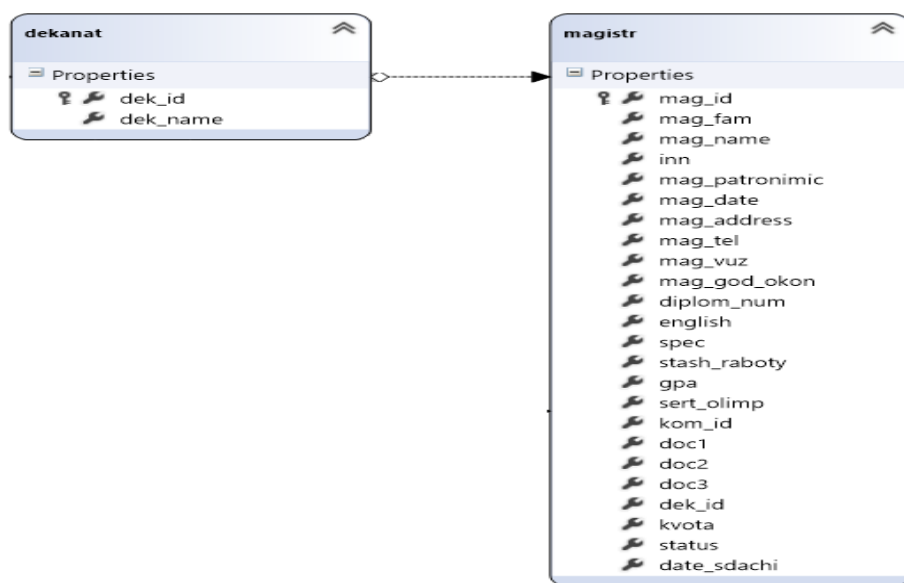
3.3.1 Бизнес ережелер

1 Талапкердің құжаттары өзінің таңдап алған мамандықтары бойынша деканатқа жіберіледі (Сурет 3.1).



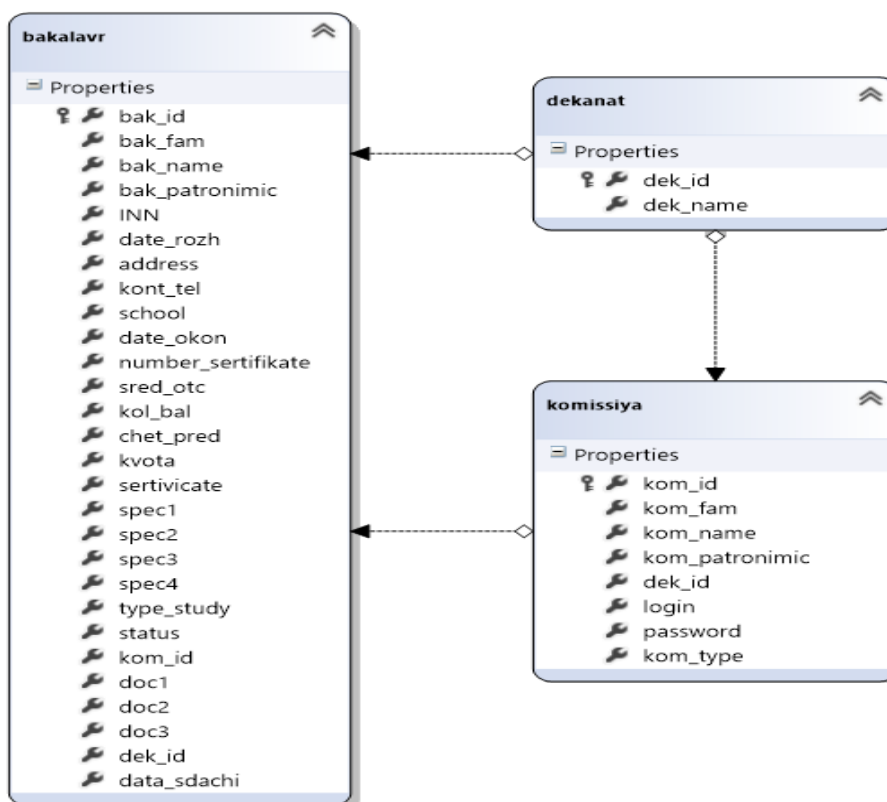
Сурет 3.1 – Бакалавр кестесі бизнес ереже үшін ER сегменті

2 Магистранттың құжаттары өзінің таңдап алған мамандығы бойынша деканатқа жіберіледі (Сурет 3.2).



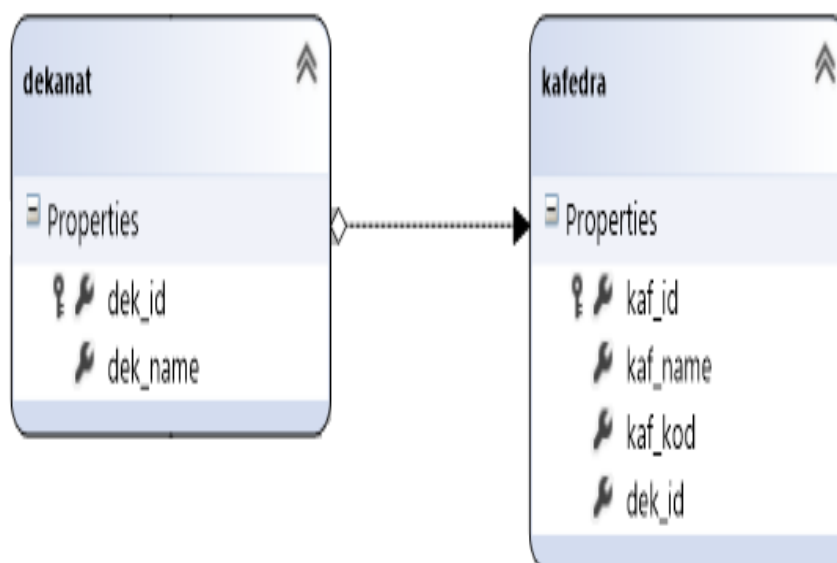
Сурет 3.2 – Магистрант кестесі бизнес ереже үшін ER сегменті

3 Талапкердің құжаттары деканатқа жіберілгеннен кейін құжаттарды қабылдау комиссиясы тексеріп оны өткізуге рұқсатын береді (Сурет 3.3).



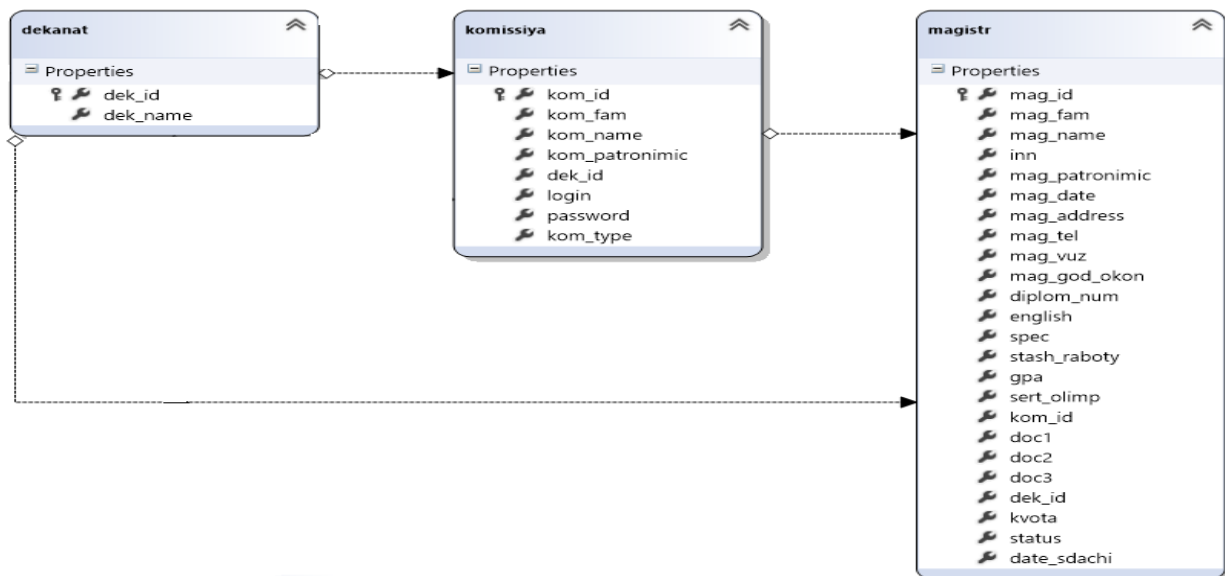
Сурет 3.3 – Деканат кестесі бизнес ереже үшін ER сегменті

4 Деканатқа түскен құжаттарды кафедралар бойынша бөлуі (Сурет 3.4).



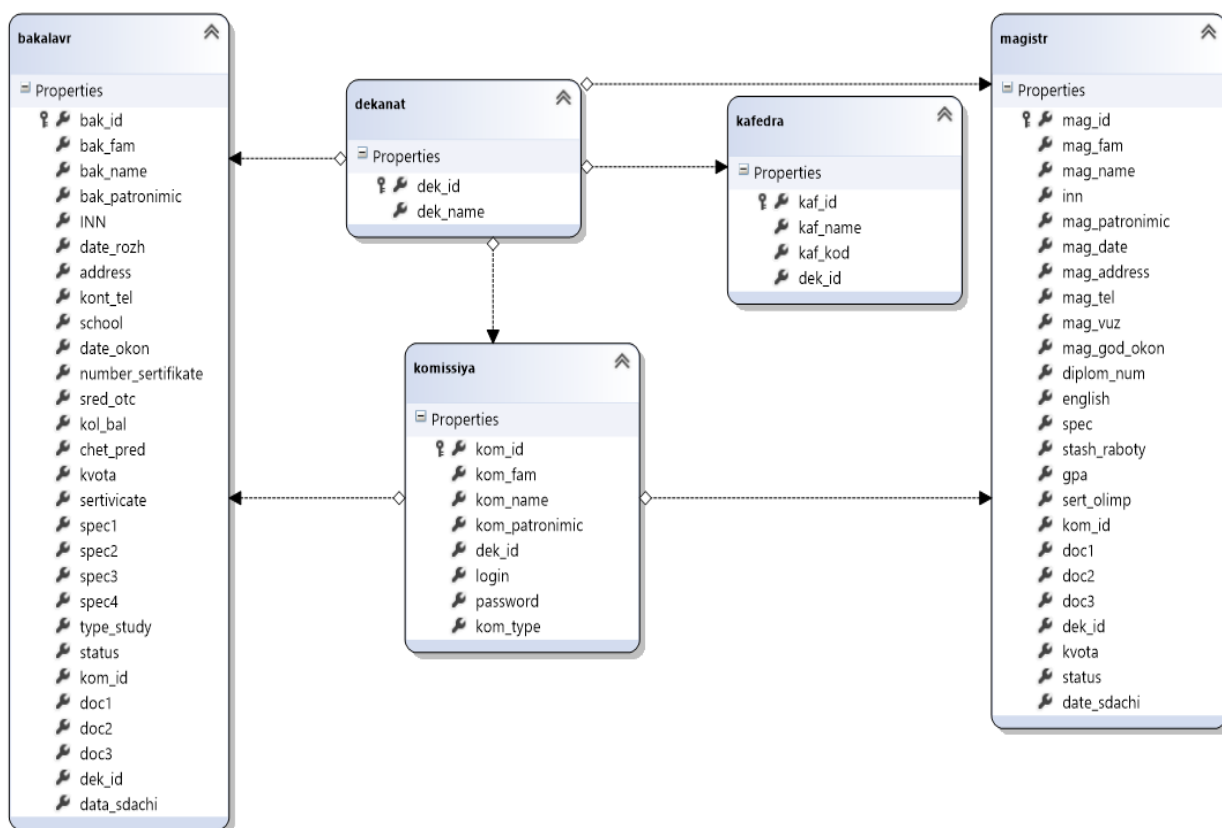
Сурет 3.4 – Кафедра кестесі бизнес ереже үшін ER сегменті

5 Магистранттын құжататры деканатқа жіберілгеннен кейін құжаттарды қабылдау комиссиясы тексеріп оны өткізуге рұқсатын береді (Сурет 3.5).



Сурет 3.5 – Қабылдау комиссия бизнес ереже үшін ER сегменті

6 Бағдарламаның ER диаграммасы (Сурет 3.6).



Сурет 3.6 – ER диаграмма

3.3.2 Реляционды қатынастарды құрастыру

Әрбір реляционды қатынас бір нысанға сәйкес келеді және оған нысанның барлық атрибуттары енгізіледі. Әрбір қатынас үшін бастапқы кілт және сыртқы кілттерді (егер бар болса) анықтау қажет. Қатынастар кесте 3.2 – 3.6 – келтірілген.

Әрбір қатынас үшін ішкі аты, типі мен ұзындығы бар атрибуттар көрсетіледі Мәліметтер типі келесідей белгіленеді: int – сандық, char – символдық.

К е с т е 3.2 – bakalavr

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Бакалавр коды	bak_id	int	Бастапқы кілт
Бакалавр тегі	bak_fam	varchar	Керекті өріс
Бакалавр аты	bak_name	varchar	Керекті өріс
Әкесінің аты	bak_patronimic	varchar	Қажетті өріс
ЖСН	INN	varchar	Қажетті өріс
Туылған жылы	date_rozh	varchar	Қажетті өріс
Мекен-жайы	address	varchar	Қажетті өріс
Байланыс телефоны	kont_tel	varchar	Қажетті өріс
Мектеп	school	varchar	Керекті өріс
Бітірген жылы	date_okon	date	Керекті өріс
Сертификат нөмері	number_sertifikate	varchar	Керекті өріс
Орташа бағасы	sred_otc	real	Керекті өріс
Ұпай	kol_bal	int	Керекті өріс
4 пән ұпайы	chet_pred	int	Керекті өріс
Жеңілдік	kvota	int	Керекті өріс
Сертификат	sertivicate	int	Керекті өріс
1 мамандық	spec1	varchar	Керекті өріс
2 мамандық	spec2	varchar	Керекті өріс
3 мамандық	spec3	varchar	Керекті өріс
4 мамандық	spec4	varchar	Керекті өріс
Сырттай оқу	type_study	int	Керекті өріс
Күндізгі оқу	status	int	Керекті өріс
Комиссия коды	kom_id	int	Сыртқы кілт
1 Сурет	doc1	varchar	Керекті өріс
2 сурет	doc2	varchar	Керекті өріс
3 Сурет	doc3	varchar	Керекті өріс
Деканат коды	dec_id	int	Сыртқы кілт
Өткізген күні	data_sdachi	datetime	Керекті өріс

К е с т е 3.3 – dekanat

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Деканат коды	dek_id	int	Бастапқы кіл
Деканат аты	dek_name	varchar	Керекті өріс

К е с т е 3.4 – kafedra

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Кафедра коды	kaf_id	int	Бастапқы кіл
Кафедра аты	kaf_name	varchar	Керекті өріс
Кафедра коды	kaf_kod	varchar	Керекті өріс
Деканат коды	dek_id	varchar	Сыртқы кілт

К е с т е 3.5 – komissiya

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Комиссия коды	kom_id	int	Бастапқы кіл
Комиссия тегі	kom_fam	varchar	Керекті өріс
Комиссия аты	kom_name	varchar	Керекті өріс
Әкесінің аты	kom_patronimic	varchar	Керекті өріс
Деканат коды	dek_id	int	Сыртқы кілт
Логин	login	varchar	Керекті өріс
Құпия сөз	password	varchar	Керекті өріс
Комиссия	kom_type	varchar	Керекті өріс

К е с т е 3.6 – magistr

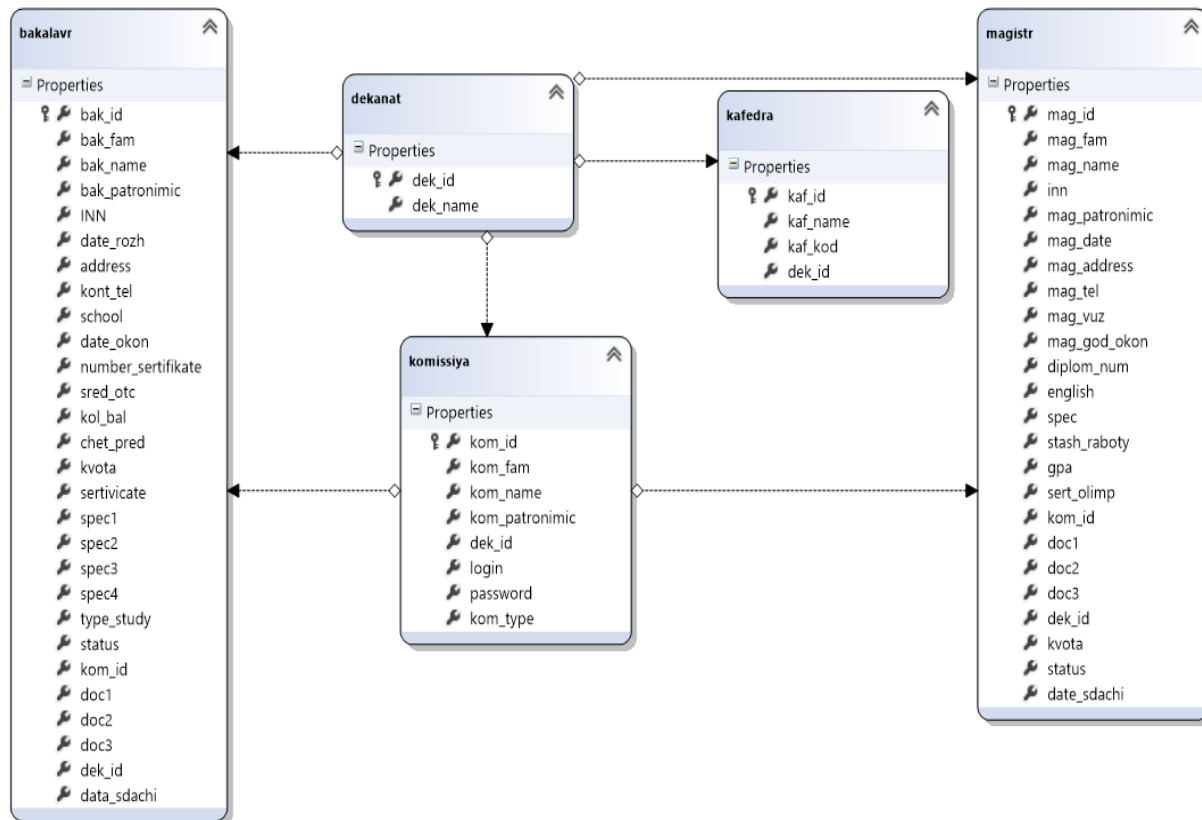
Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Маг. коды	mag_id	int	Бастапқы кіл
Маг. тегі	mag_fam	varchar	Керекті өріс
Маг. аты	mag_name	varchar	Қажетті өріс
ЖСН	Inn	varchar	Керекті өріс
Әкесінің аты	mag_patronimic	varchar	Керекті өріс
Туылған жылы	mag_date	date	Керекті өріс
Мекен-жайы	mag_address	varchar	Керекті өріс
Байланыс телефоны	mag_tel	varchar	Керекті өріс
Бітірген ЖОО	mag_vuz	varchar	Керекті өріс
Бітірген жылы	mag_god_okon	date	Керекті өріс
Диплом нөмері	diplom_num	varchar	Керекті өріс
Ағылшын	english	int	Керекті өріс
Мамандық	spec	varchar	Керекті өріс
Еңбек тәжірибесі	stash_raboty	int	Керекті өріс
Орташа бал	gpa	real	Керекті өріс

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Олимпиада жүлдегері	sert_olimp	int	Керекті өріс
Комиссия коды	kom_id	int	Сыртқы кілт
1 Сурет	doc1	varchar	Керекті өріс
2 Сурет	doc2	varchar	Керекті өріс
3 Сурет	doc3	varchar	Керекті өріс
Деканат коды	dec_id	int	Сыртқы кілт
Жеңілдік	kvota	int	Керекті өріс
Мамандық	status	int	Керекті өріс
Өткізген күні	date_sdachi	datetime	Керекті өріс

3.3.3 Логикалық жобалау

Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (MS SQL Server) үшін жарамды форматқа трансформаланады. Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын MS SQL Server көмегімен жасаймыз.

ДҚ диаграммасы (Сурет 3.7).



Сурет 3.7 – Программаның ER диаграмма

3.4 Физикалық жобалау

3.4.1 Дерек қорына қосымша құру

Тапсырманың қойылуы.

Әдістемелік тұрғыдан қарағанда, мәліметтер қорымен жұмыс істеу қосымшасын жасау қарапайым программаны құрудан еш айырмашылығы жоқ. Дәл сол сияқты формаға керекті компоненттер енгізіледі, компоненттердің қасиеттерінің мәндері жазылады, керекті оқиғаны өңдеу процедурасы жасалады.

Мәліметтер қорымен жұмыс істеу қосымшасы мәліметтерге енді қамтамасыз ететін компоненттерден, өрістердің мәліметтерін қарап шығу және редакциялау мүмкіндіктерінен тұрады. Мәліметтерге кіруді қамтамасыз ететін компоненттер Data Access компоненттер палитрасы қосымшасында, ал мәліметтерді бейнелеу компоненттері - Data Controls қосымшасында болады [1,5,9].

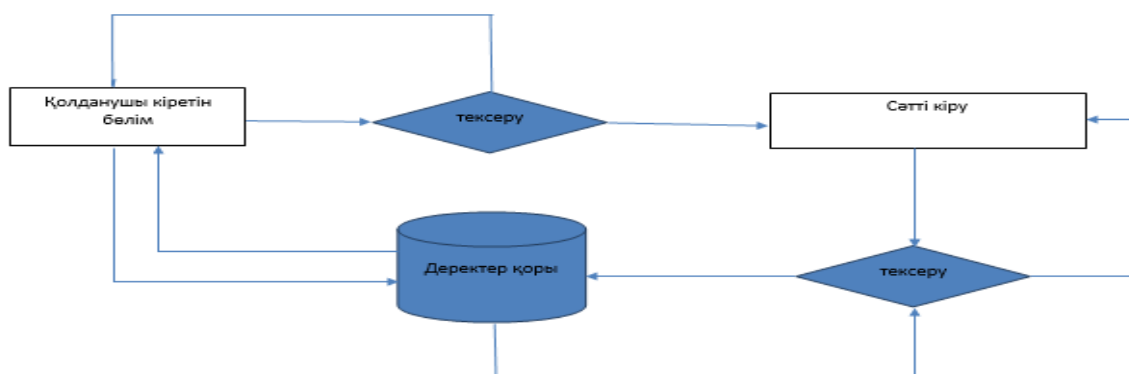
Database компоненті мәліметтер қорын кестелер жиынтығы ретінде, ал Table компоненті – мәліметтер қорының бір кестесі. DataSource (мәліметтер көзі) компоненті мәліметтерді бейнелеу-редакциялау компоненті (мысалы DBGrid компонентін) мен мәліметтер көзі (Table компоненті немесе SQL-сұраныс нәтижесі) арасындағы байланысты қамтамасыз етеді. DataSource компоненті мәліметтер көзін тез таңдауды, сол компоненттердің біреуін пайдаланып, мысалы DBGrid кестеден мәліметтерді көру үшін немесе осы кестеге SQL-сұранысты орындау нәтижесін көруге мүмкіндік береді [2,6].

3.4.2 Бағдарламаның негізгі алгоритмі

Есеп шартына байланысты алгоритм командаларының жай және құрама түрлері болады.

Әр алгоритм қарапайым бір қадамды ақпаратты өңдеудің бір қарапайым операциясын орындайтын жай командадан тұрады. Осындай командалар қатарына меншіктеу, мәлімет енгізу, нәтиже алу командалары жатады.

Қолданушы бөлімі (Сурет 3.8).



Сурет 3.8 – Қолданушы бөлімінің алгоритмі
Администратор бөлімі (Сурет 3.9).



Сурет 3.9 – Администратор бөлімінің алгоритмі

3.4.3 Администратор бөлімінің коды

Бағдарлама 3 бөліктен тұрады. Администратор бөлімінде бүкіл құжаттар сақталады.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;
using Excel = Microsoft.Office.Interop.Excel;
namespace administrator
{
    public partial class adminControl : Form
    {
        private static int rec;
        public adminControl()
        {
            InitializeComponent();
            groupBox1.Enabled = false;
            a1 = "select k.kom_id, k.kom_fam as 'Тегі', k.kom_name
as 'Аты', k.kom_patronymic as 'Әкесінің аты', k.kom_type as
'Қабылдау бөлімі', k.login as 'Логин', k.password as 'Құпия сөз',
  
```

```

d.dek_name as 'Деканат' from komissiya k inner join dekanat d on
d.dek_id = k.dek_id";
    connect();
    dataGridView1.Columns["kom_id"].Visible = false;
    dataGridView1.Rows[0].Cells[1].Selected = true;
    var dekanat = (from i in Program.db.dekanats select
i.dek_name);
    foreach (var dek in dekanat)
    {
        comboBox1.Items.Add(dek);
    }

    comboBox5.Items.Add("Барлық");
    comboBox5.Items.Add("Өткізілген құжаттар");
    comboBox5.Items.Add("Қарастырылатын құжаттар");
    comboBox2.Items.Add("бакалавр");
    comboBox2.Items.Add("магистратура");
    comboBox5.Enabled = false;
    groupBox3.Enabled = false;
    var faculty = (from i in Program.db.dekanats select
i.dek_name);
    foreach (var f in faculty)
    {
        comboBox3.Items.Add(f);
    }
}
private void dataGridView1_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header =
this.dataGridView1.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView1.Rows[index].HeaderCell.Value =
indexStr;
}
private void dataGridView2_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header =
this.dataGridView2.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView2.Rows[index].HeaderCell.Value =
indexStr;
}
string a, a1, b, b1, cn;
public void connect()
{
    SqlConnection cn = Program.conn;

```



```

        a = a1;
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(a, cn);
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
    }
    public void connect2()
    {
        SqlConnection cn = Program.conn;
        b = b1;
        DataSet ds1 = new DataSet();
        SqlDataAdapter da1 = new SqlDataAdapter(b, cn);
        da1.Fill(ds1);
        dataGridView2.DataSource = ds1.Tables[0];
    }
    private void button5_Click(object sender, EventArgs e)
    {
        textBox1.Clear();

        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();
        comboBox1.Items.Clear();
        comboBox2.Items.Clear();
        groupBox1.Enabled = false;
        button1.Enabled = true;
        button2.Enabled = true;
        dataGridView1.Enabled = true;
        var dekanat = (from i in Program.db.dekanats select
i.dek_name);
        foreach (var dek in dekanat)
        {
            comboBox1.Items.Add(dek);
        }
        comboBox2.Items.Add("бакалавр");
        comboBox2.Items.Add("магистратура");
    }
    private void button1_Click(object sender, EventArgs e)
    {
        button2.Enabled = false;
        groupBox1.Enabled = true;
        rec = 1;
    }
    private void button2_Click(object sender, EventArgs e)
    {
        dataGridView1.Enabled = false;
        groupBox1.Enabled = true;
        button1.Enabled = false;
        rec = 0;
        textBox1.Text
dataGridView1.CurrentRow.Cells[1].Value.ToString();
    }

```

```

        textBox2.Text =
dataGridView1.CurrentRow.Cells[2].Value.ToString();
        textBox3.Text =
dataGridView1.CurrentRow.Cells[3].Value.ToString();
        textBox4.Text =
dataGridView1.CurrentRow.Cells[5].Value.ToString();
        textBox5.Text =
dataGridView1.CurrentRow.Cells[6].Value.ToString();
        comboBox1.Text =
dataGridView1.CurrentRow.Cells[7].Value.ToString();
        comboBox2.Text =
dataGridView1.CurrentRow.Cells[4].Value.ToString();
    }
    private void button3_Click(object sender, EventArgs e)
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "";
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Пайдаланушыға жүйеге кіру жабылды",
"Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    private void button4_Click(object sender, EventArgs e)
    {
        try
        {
            var dek_id = (from i in Program.db.dekanats where
i.dek_name == comboBox1.Text select i.dek_id).SingleOrDefault();
            if (rec == 0)
            {
                SqlConnection sqlConnection1 = Program.conn;
                sqlConnection1.Open();
                SqlCommand cmd = new SqlCommand();
                cmd.CommandType =
System.Data.CommandType.Text;
                cmd.CommandText = "update komissiya set
kom_fam='" + textBox1.Text + "', kom_name = '" + textBox2.Text +
"', kom_patronimic = '" + textBox3.Text + "', dek_id = '" + dek_id
+ "', _login = '" + textBox4.Text + "', password = '" +
textBox5.Text + "', kom_type = '" + comboBox2.Text + "' where
kom_id = " + dataGridView1.CurrentRow.Cells[0].Value;
                cmd.Connection = sqlConnection1;
                cmd.ExecuteNonQuery();
                sqlConnection1.Close();
                MessageBox.Show("Ақпарат сәтті өзгертілді!",
"Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
        }
    }
}

```

```

        textBox1.Clear();
        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();
        comboBox1.Items.Clear();
        comboBox2.Items.Clear();
        groupBox1.Enabled = false;
        button1.Enabled = true;
        button2.Enabled = true;
        dataGridView1.Enabled = true;
        var dekanat = (from i in Program.db.dekanats
select i.dek_name);
        foreach (var dek in dekanat)
        {
            comboBox1.Items.Add(dek);
        }
        comboBox2.Items.Add("бакалавр");
        comboBox2.Items.Add("магистратура");
    }
    else if (rec == 1)
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType =
System.Data.CommandType.Text;
        cmd.CommandText = "insert into
komissiya(kom_fam, kom_name, kom_patronimic, dek_id, login,
password, kom_type) values('" + textBox1.Text + "','" +
textBox2.Text + "','" + textBox3.Text + "','" + dek_id + "','" +
textBox4.Text + "','" + textBox5.Text + "','" + comboBox2.Text +
"')";

        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Ақпарат сәтті енгізілді!",
"Назар аударыңыз", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();
        comboBox1.Items.Clear();
        comboBox2.Items.Clear();
        groupBox1.Enabled = false;
        button1.Enabled = true;
        button2.Enabled = true;
        dataGridView1.Enabled = true;
        var dekanat = (from i in Program.db.dekanats
select i.dek_name);

```

```

        foreach (var dek in dekanat)
        {
            comboBox1.Items.Add(dek);
        }
        comboBox2.Items.Add("бакалавр");
        comboBox2.Items.Add("магистратура");
    }
    a1 = "select k.kom_id, k.kom_fam as 'Тегі',
k.kom_name as 'Аты', k.kom_patronimic as 'Әкесінің аты',
k.kom_type as 'Қабылдау бөлімі', k.login as 'Логин', k.password as
'Құпия сөз', d.dek_name as 'Деканат' from komissiya k inner join
dekanat d on d.dek_id = k.dek_id";
    connect();
    dataGridView1.Columns["kom_id"].Visible = false;
    dataGridView1.Rows[0].Cells[1].Selected = true;
}
catch (Exception)
{
    MessageBox.Show("Деректер қоры жабық", "Назар
аударыңыз", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
private void comboBox5_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (radioButton2.Checked == true && comboBox5.Text ==
"Барлық")
    {
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as
'Аты', bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address
as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b";
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
    else if (radioButton2.Checked == true &&
comboBox5.Text == "Өткізілген құжаттар")
    {
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as
'Аты', bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address
as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where status = 2";
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
    else if (radioButton2.Checked == true &&
comboBox5.Text == "Қарастырылатын құжаттар")

```

```

        {
            b1 = "select b.bak_fam as 'Тегі', b.bak_name as
'Аты', bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address
as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where status = 1";
            connect2();
            dataGridView2.Columns["status"].Visible = false;
        }
        else if (radioButton1.Checked == true &&
comboBox5.Text == "Барлық")
        {
            b1 = "select m.mag_fam as 'Тегі', m.mag_name as
'Аты', m.mag_patronimic as 'Әкесінің аты', m.inn as 'ЖСН',
m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gpa as
'GPA', m.english as 'Шет тілі', m.spec as 'Мамандық',
m.date_sdachi as 'Тапсырған уақыты', m.status from magistr m";
            connect2();
            dataGridView2.Columns["status"].Visible = false;
        }
        else if (radioButton1.Checked == true &&
comboBox5.Text == "Өткізілген құжаттар")
        {
            b1 = "select m.mag_fam as 'Тегі', m.mag_name as
'Аты', m.mag_patronimic as 'Әкесінің аты', m.inn as 'ЖСН',
m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gpa as
'GPA', m.english as 'Шет тілі', m.spec as 'Мамандық',
m.date_sdachi as 'Тапсырған уақыты', m.status from magistr m where
status = 2";
            connect2();
            dataGridView2.Columns["status"].Visible = false;
        }
        else if (radioButton1.Checked == true &&
comboBox5.Text == "Қарастырылатын құжаттар")
        {
            b1 = "select m.mag_fam as 'Тегі', m.mag_name as
'Аты', m.mag_patronimic as 'Әкесінің аты', m.inn as 'ЖСН',
m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gpa as
'GPA', m.english as 'Шет тілі', m.spec as 'Мамандық',
m.date_sdachi as 'Тапсырған уақыты', m.status from magistr m where
status = 1";
            connect2();
            dataGridView2.Columns["status"].Visible = false;
        }
        }
        groupBox3.Enabled = true;
    }
    private void radioButton2_CheckedChanged(object sender,
EventArgs e)
    {

```

```

        comboBox5.Enabled = true;
        groupBox3.Enabled = false;
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты',
bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address as
'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where status = 3";
        connect2();
        comboBox5.Items.Clear();
        comboBox5.Items.Add("Барлық");
        comboBox5.Items.Add("Өткізілген құжаттар");
        comboBox5.Items.Add("Қарастырылатын құжаттар");
    }
    private void radioButton1_CheckedChanged(object sender,
EventArgs e)
    {
        comboBox5.Enabled = true;
        groupBox3.Enabled = false;
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты',
bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address as
'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where status = 3";
        connect2();
        comboBox5.Items.Clear();
        comboBox5.Items.Add("Барлық");
        comboBox5.Items.Add("Өткізілген құжаттар");
        comboBox5.Items.Add("Қарастырылатын құжаттар");
    }
    private void comboBox3_SelectedIndexChanged(object sender,
EventArgs e)
    {
        int dekanat_id = (from i in Program.db.dekanats where
i.dek_name == comboBox3.Text select i.dek_id).SingleOrDefault();
        if (radioButton1.Checked == true)
        {
            b1 = "select m.mag_fam as 'Тегі', m.mag_name as
'Аты', m.mag_patronimic as 'Әкесінің аты', m.inn as 'ЖСН',
m.mag_address as 'Мекен-жайы', m.mag_tel as 'Телефон', m.gpa as
'GPA', m.english as 'Шет тілі', m.spec as 'Мамандық',
m.date_sdachi as 'Тапсырған уақыты', m.status from magistr m where
m.dek_id = " + dekanat_id;

            connect2();
            dataGridView2.Columns["status"].Visible = false;
        }
        else if (radioButton2.Checked == true)
        {

```

```

        b1 = "select b.bak_fam as 'Тегі', b.bak_name as
'Аты', bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address
as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where b.dek_id = " + dekanat_id;
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
}
private void textBox6_TextChanged(object sender, EventArgs
e)
{
    if (radioButton1.Checked == true)
    {
        b1 = "select m.mag_fam as 'Тегі', m.mag_name as
'Аты', m.mag_patronimic as 'Әкесінің аты', m.inn as 'ЖСН',
m.mag_address as 'Мекен-жайы', m.mag_tel as 'Телефон', m.gpa as
'GPA', m.english as 'Шет тілі', m.spec as 'Мамандық',
m.date_sdachi as 'Тапсырған уақыты', m.status from magistr m where
m.mag_fam like '" + textBox6.Text + "%'";
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
    else if (radioButton2.Checked == true)
    {
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as
'Аты', bak_patronimic as 'Әкесінің аты', b.INN as 'ЖСН', b.address
as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған
ұпай', b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as
'2 мамандық', b.spec3 as '3 мамандық', b.spec4 as '4 мамандық',
b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b
where b.bak_fam like '" + textBox6.Text + "%'";
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
}
private void button3_Click_1(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count == 0)
    {
    }
    else
    {
        try
        {
            if (radioButton1.Checked == true)
            {

```

```

        string myDocsPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
        SaveFileDialog saveResults = new
SaveFileDialog();
        saveResults.Filter = "Книга Excel
(*.xls)|*.xlsx";
        saveResults.FilterIndex = 2;
        saveResults.RestoreDirectory = true;
        saveResults.InitialDirectory = myDocsPath;
        saveResults.FileName = "Список
поступающих";
        saveResults.Title = "АУЭС: Экспорт
результатов";
        Microsoft.Office.Interop.Excel.Application
ExcelApp;
        ExcelApp = new
Microsoft.Office.Interop.Excel.Application();

ExcelApp.Application.Workbooks.Add(Type.Missing);
        ExcelApp.Columns.ColumnWidth = 15;
        ExcelApp.Cells[1, 1] = "Тегі";
        ExcelApp.Cells[1, 2] = "Аты";
        ExcelApp.Cells[1, 3] = "Әкесінің аты";
        ExcelApp.Cells[1, 4] = "ЖСН";
        ExcelApp.Cells[1, 5] = "Мекен-жай";
        ExcelApp.Cells[1, 6] = "Телефон";
        ExcelApp.Cells[1, 7] = "ГРА";
        ExcelApp.Cells[1, 8] = "Шет тілі";
        ExcelApp.Cells[1, 9] = "Мамандық";
        ExcelApp.Cells[1, 10] = "Тапсырған
уақыты";

        for (int i = 0; i <
dataGridView2.ColumnCount; i++)
        {
            for (int j = 0; j <
dataGridView2.RowCount; j++)
            {
                ExcelApp.Cells[j + 2, i + 1] =
(dataGridView2[i, j].Value).ToString();
            }
        }

ExcelApp.ActiveWorkbook.SaveAs(saveResults.FileName,
saveResults.ShowDialog());
        ExcelApp.Workbooks.Close();
        ExcelApp.Application.Quit();
    }
    else if (radioButton2.Checked == true)
    {
        string myDocsPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

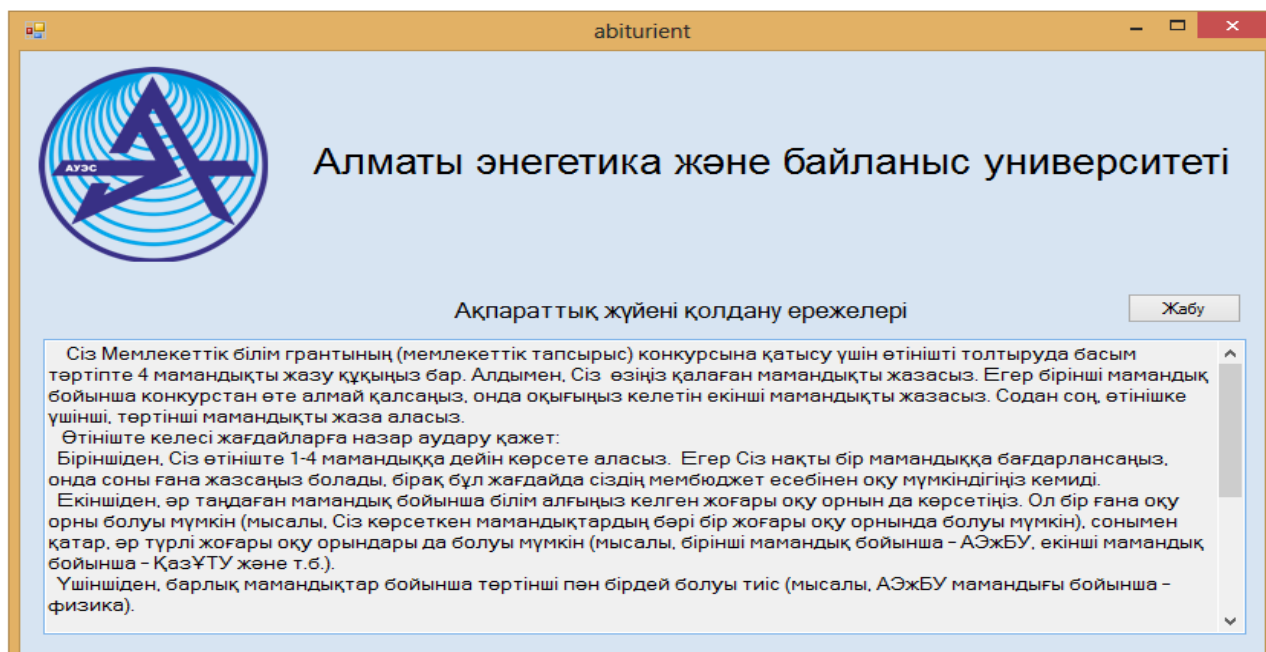
```



```

SaveFileDialog()
SaveFileDialog();
(*.xls)|*.xlsx";
поступающих";
результатов";
ExcelApp;
Microsoft.Office.Interop.Excel.Application
ExcelApp = new
Microsoft.Office.Interop.Excel.Application();
ExcelApp.Application.Workbooks.Add(Type.Missing);
ExcelApp.Columns.ColumnWidth = 15;
ExcelApp.Cells[1, 1] = "Тегі";
ExcelApp.Cells[1, 2] = "Аты";
ExcelApp.Cells[1, 3] = "Әкесінің аты";
ExcelApp.Cells[1, 4] = "ЖСН";
ExcelApp.Cells[1, 5] = "Мекен-жайы";
ExcelApp.Cells[1, 6] = "Телефон";
ExcelApp.Cells[1, 7] = "Жинаған ұпай";
ExcelApp.Cells[1, 8] = "4 пән";
ExcelApp.Cells[1, 9] = "1 мамандық";
ExcelApp.Cells[1, 10] = "2 мамандық";
ExcelApp.Cells[1, 13] = "3 мамандық";
ExcelApp.Cells[1, 14] = "4 мамандық";
ExcelApp.Cells[1, 15] = "Тапсырған
уақыты";
for (int i = 0; i <
dataGridView2.ColumnCount; i++)
{
for (int j = 0; j <
dataGridView2.RowCount; j++)
{
ExcelApp.Cells[j + 2, i + 1] =
(dataGridView2[i, j].Value).ToString();
}
}
ExcelApp.ActiveWorkbook.SaveAs(saveResults.FileName,
saveResults.ShowDialog());
ExcelApp.Workbooks.Close();
ExcelApp.Application.Quit();
}
}
catch (Exception)
{

```

Сурет 3.11 – Қысқаша түсініктеме

bakalavr1

Алматы энегетика және байланыс университеті

Құжаттарды қабылдау

Тегі	<input type="text" value="Шалабаев"/>	4 пән ұпайы	<input type="text" value="23"/>
Аты	<input type="text" value="Арнур"/>	Олимпиада жүлдегері	<input type="text" value="жоқ"/>
Әкесінің аты	<input type="text" value="Маратович"/>	Жеңілдік	<input type="text" value="жоқ"/>
ЖСН	<input type="text" value="123456789999"/>	I мамандық	<input type="text" value="Информатика"/>
Туған жылы	<input type="text" value="28.05.1996"/>	II мамандық	<input type="text" value="Автоматтандыру және басқару"/>
Мекен-жайы	<input type="text" value="Ақсай 5 дом 45 кв 8"/>	III мамандық	<input type="text" value="Ақпараттық технологиялар"/>
Байланыс телефоны	<input type="text" value="87775945994"/>	IV мамандық	<input type="text" value="Есептеу техникасы және бағдарла"/>
Мектеп	<input type="text" value="№173"/>	I құжат	<input type="text" value="Жылу энергетика"/>
Бітірген жылы	<input type="text" value="28.05.2014"/>	II құжат	<input type="text" value="Электроэнергетика"/>
Сертификат нөмері	<input type="text" value="4562288636"/>	III құжат	<input type="text" value="Радиотехника, электроника және т"/>
Орташа бағасы	<input type="text" value="4.7"/>	оқу түрі	<input type="text" value="Ғарыштық техника и технологиялар"/>
Ұпай	<input type="text" value="94"/>		

Сурет 3.12 – Мамандықты тандайтын терезе

Алматы энегетика және байланыс университеті

Құжаттарды қабылдау

Тегі	Шалабаев	4 пән ұпайы	23
Аты	Арнур	Олимпиада жүлдегері	жоқ
Әкесінің аты	Маратович	Жеңілдік	жоқ
ЖСН	123456789999	I мамандық	Есептеу техникасы және бағдар
Туған жылы	28.05.1996	II мамандық	Жылу энергетика
Мекен-жайы	Ақсай 5 дом 45 кв 8	III мамандық	Электроэнергетика
Байланыс телефоны	87775945994	IV мамандық	Информатика
Мектеп	№173	I құжат	Автоматтандыру және басқару
Бітірген жылы	28.05.2014	II құжат	Ақпараттық технологиялар
Сертификат нөмері	1561512355	III құжат	Радиотехника, электроника және т
Орташа бағасы	4.7	оқу түрі	Ғарыштық техника и технологиялар
Ұпай	94		

Кері қайту Тазалау Тапсыру

Сурет 3.13 – Талапкер бір мамандықты екінші рет тандай алмайды

Алматы энегетика және байланыс университеті

Құжаттарды қабылдау

Тегі	Шалабаев	4 пән ұпайы	23
Аты	Арнур	Олимпиада жүлдегері	жоқ
Әкесінің аты	Маратович	Жеңілдік	жоқ
ЖСН	123456789999	I мамандық	Есептеу техникасы және бағдар
Туған жылы	28.05.1996	II мамандық	Жылу энергетика
Мекен-жайы	Ақсай 5 дом 45 кв 8	III мамандық	Электроэнергетика
Байланыс телефоны	87775945994	IV мамандық	Автоматтандыру және басқару
Мектеп	№173	I құжат	C:\Users\Абзал\Desktop\доқ
Бітірген жылы	28.05.2014	II құжат	C:\Users\Абзал\Desktop\доқ
Сертификат нөмері	1545856622	III құжат	C:\Users\Абзал\Desktop\доқ
Орташа бағасы	4.7	оқу түрі	күндізгі
Ұпай	94		

Кері қайту Тазалау Тапсыру

Назар аударыңыз

Ақпарат сәтті енгізілді

OK

Сурет 3.14 – Талапкердің құжаттарын қабылдау терезесі

magistratura

Алматы энегетика және байланыс университеті

Құжаттарды қабылдау

Тегі	<input type="text" value="Ертаев"/>	GPA	<input type="text" value="3.6"/>
Аты	<input type="text" value="Азамат"/>	<input type="text" value="85"/>	
Әкесінің аты	<input type="text" value="Асхатович"/>	бесі	<input type="text" value="жоқ"/>
ЖСН	<input type="text" value="159418566333"/>	үлдегері	<input type="text" value="жоқ"/>
Туған жылы	<input type="text" value="06.09.1991"/>		<input type="text" value="жоқ"/>
Мекен-жайы	<input type="text" value="Астана,Орбита 4"/>		<input type="text" value="Есептеу техникасы және бағдар"/>
Байланыс телефоны	<input type="text" value="87752566652"/>		<input type="text" value="C:\Users\Абзал\Desktop\доқ\"/>
ЖОО	<input type="text" value="АУЭС"/>	II құжат	<input type="text" value="C:\Users\Абзал\Desktop\доқ\"/>
Бітірген жылы	<input type="text" value="11.06.2014"/>	III құжат	<input type="text" value="C:\Users\Абзал\Desktop\доқ\"/>
Диплом нөмері	<input type="text" value="1458936696"/>		

Назар аударыңыз

Ақпарат сәтті енгізілді

OK

Кері қайту Тазалау Тапсыру

Сурет 3.15 – Магистранттың құжаттарын қабылдау терезесі
Комиссия бөлімі (Сурет 3.16 – 3.17).

komForm

Алматы энегетика және байланыс университеті

Логин

Құпия сөз

Сурет 3.16 – Комиссияның кіретін терезесі

komControl

Алматы энегетика және байланыс университеті

Құжаттарды тексеру

Аты-жөні, әкесінің аты	<input type="text" value="Шалабаев Арнур Маратович"/>	4 пән ұпайы	<input type="text" value="23"/>
ЖСН	<input type="text" value="123456789999"/>	Олимпиада жүлдегері	<input type="text" value="жоқ"/>
Туған жылы	<input type="text" value="28.05.1996"/>	Жеңілдік	<input type="text" value="жоқ"/>
Мекен-жайы	<input type="text" value="Ақсай 5 дом 45 кв 8"/>	I мамандық	<input type="text" value="5B070400"/>
Байланыс телефоны	<input type="text" value="87775945994"/>	II мамандық	<input type="text" value="5B071700"/>
Мектеп	<input type="text" value="№173"/>	III мамандық	<input type="text" value="5B071800"/>
Бітірген жылы	<input type="text" value="28.05.2014"/>	IV мамандық	<input type="text" value="5B070200"/>
Сертификат номері	<input type="text" value="1545856622"/>	I құжат	<input type="text" value="C:\Users\Абзал\Desktop\проект"/>
Орташа бағасы	<input type="text" value="4.7"/>	II құжат	<input type="text" value="C:\Users\Абзал\Desktop\проект"/>
Ұпай	<input type="text" value="94"/>	III құжат	<input type="text" value="C:\Users\Абзал\Desktop\проект"/>
		оқу түрі	<input type="text" value="күндізгі"/>

Сурет 3.17 – Құжаттарды қабылдау немесе қарастыруға жіберетін терезесі
Администратор бөлімі (Сурет 3.18 – 3.25).

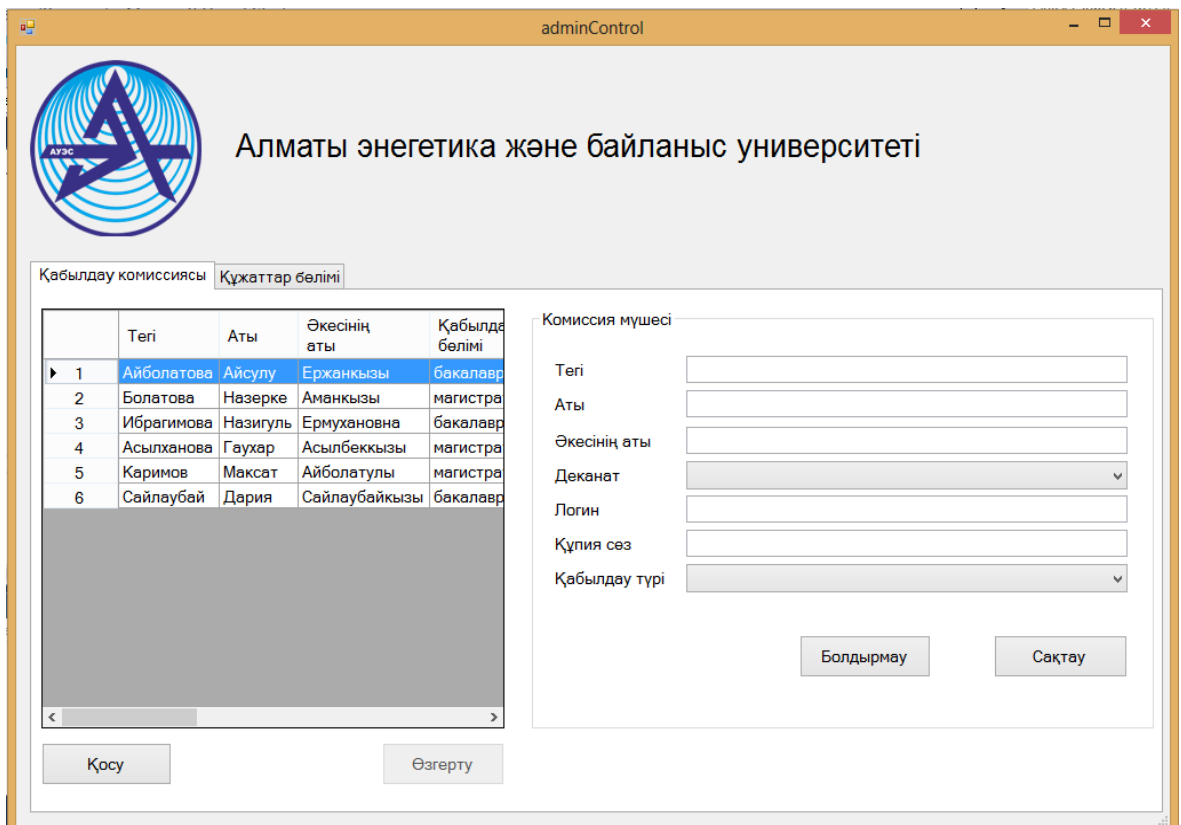
administrator

Алматы энегетика және байланыс университеті

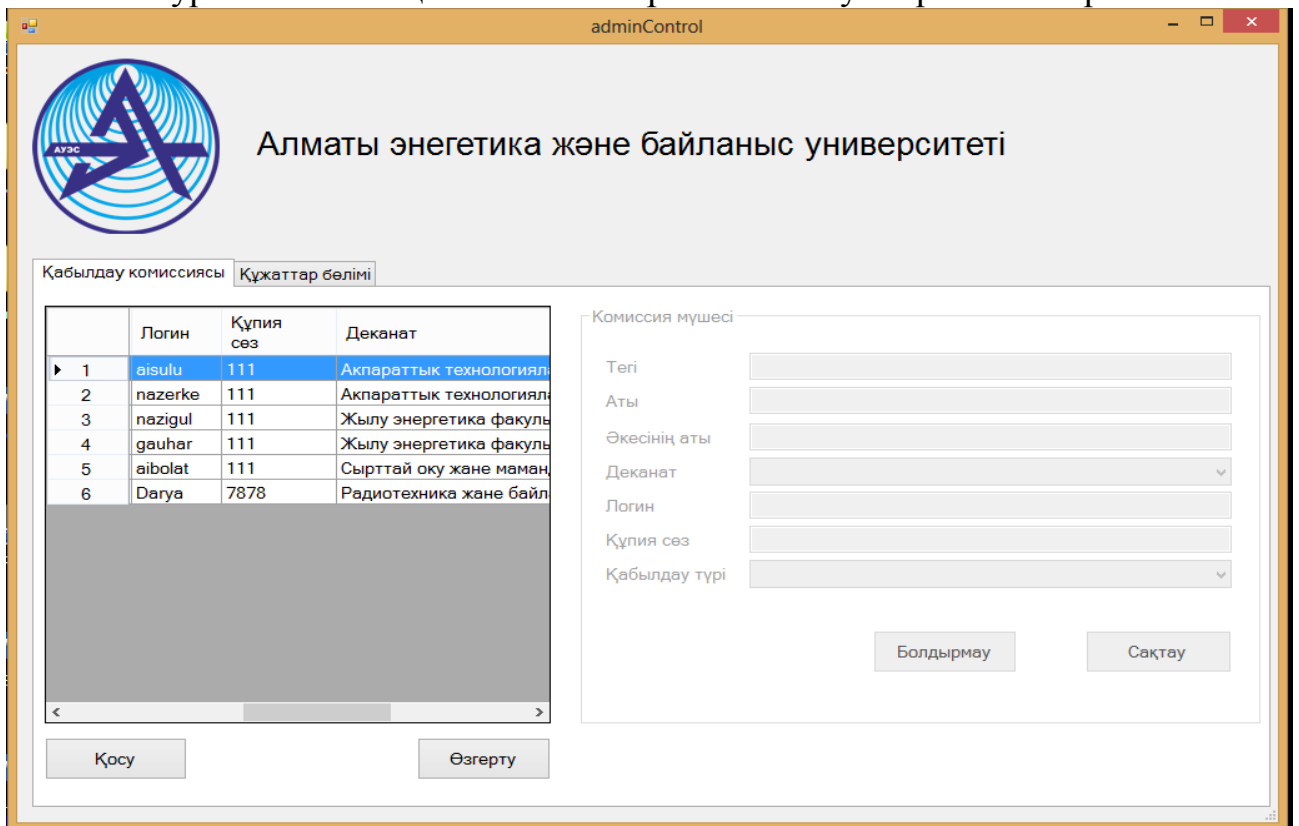
Логин

Құпия сөз

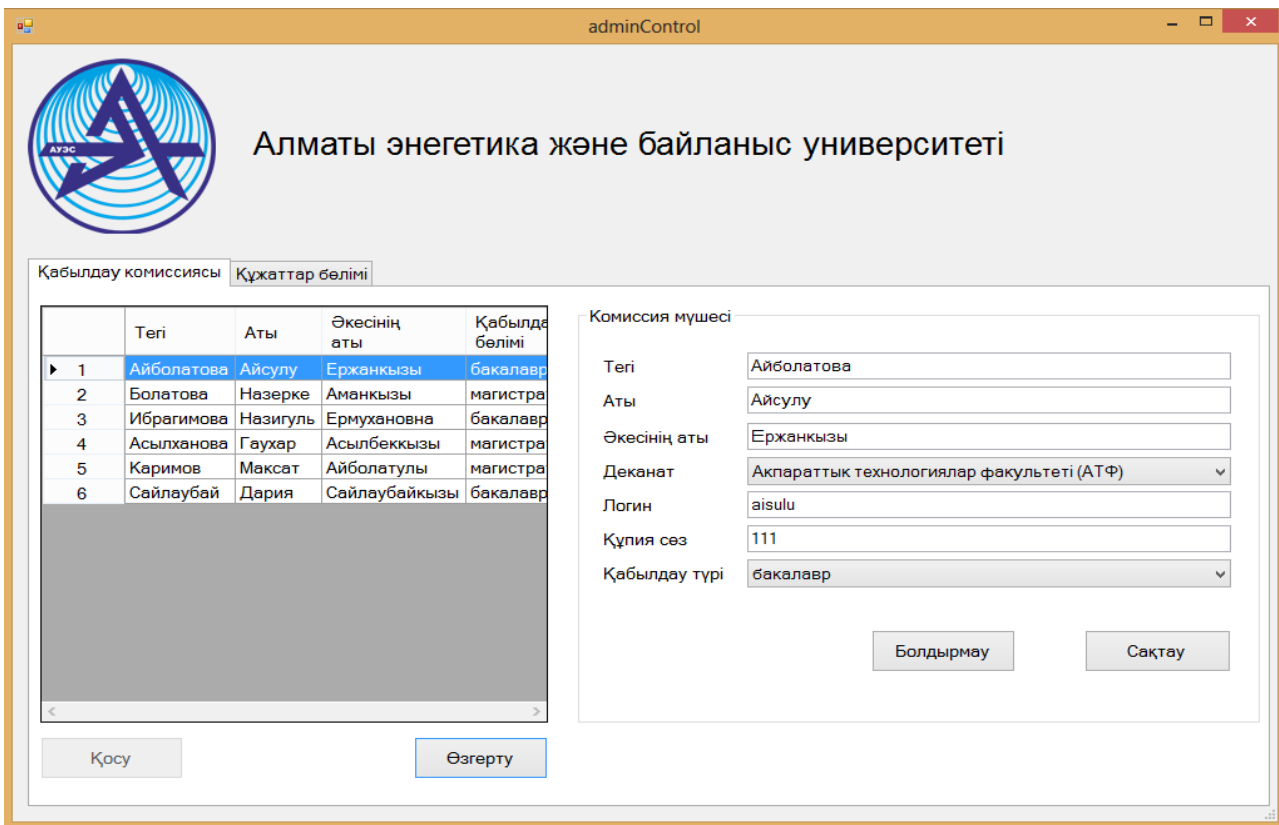
Сурет 3.18 – Администратор кіретін терезе



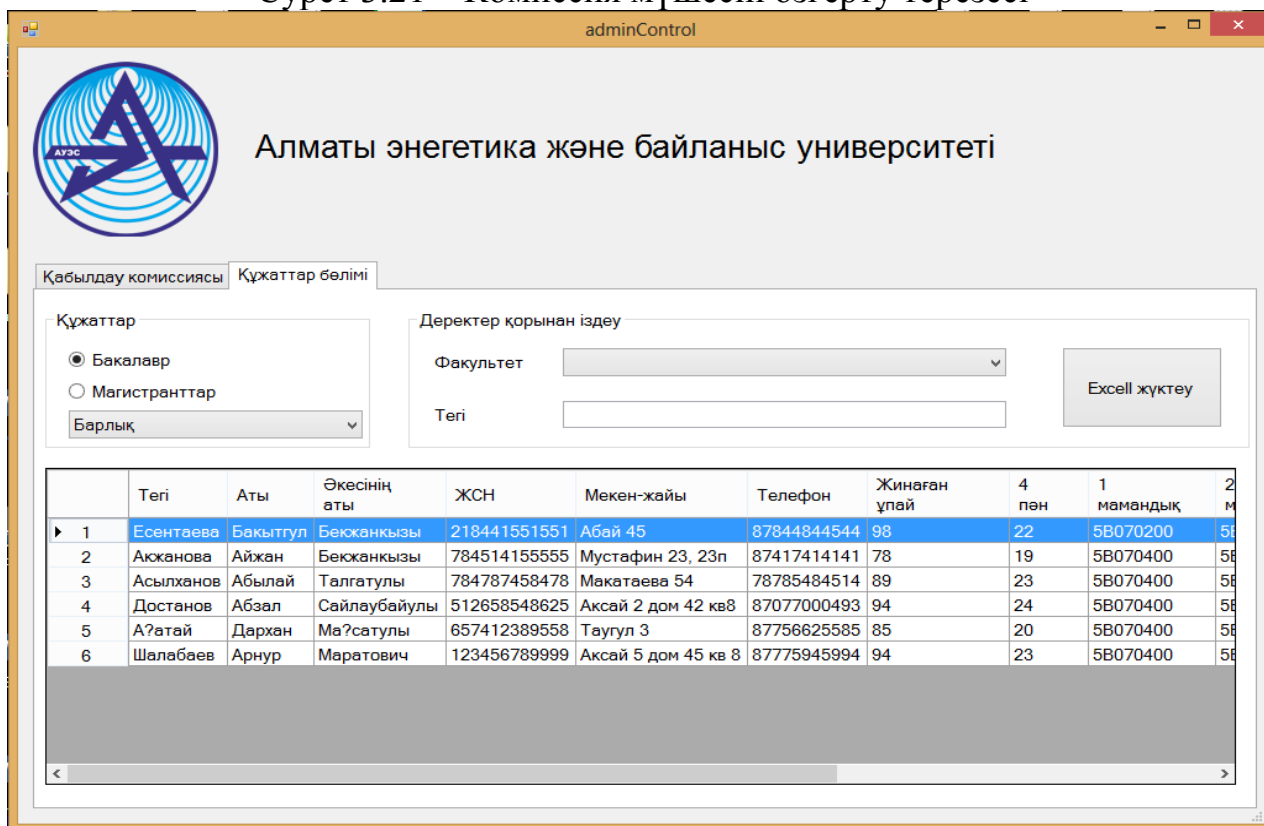
Сурет 3.19 – Жаңа комиссия мүшесін енгізуге арналған терезе



Сурет 3.20 – Комиссияның кіретін логині мен құпия сөзін көрсететін терезе




Сурет 3.21 – Комиссия мүшесін өзгерту терезесі



Сурет 3.22 – Енгізілген барлық құжаттар бөлімі

adminControl



Алматы энегетика және байланыс университеті

Қабылдау комиссиясы Құжаттар бөлімі

Құжаттар

Бакалавр
 Магистранттар
 Өткізілген құжаттар

Деректер қорынан іздеу

Факультет


Тегі

Excell жүктеу

	Тегі	Аты	Әкесінің аты	ЖСН	Мекен-жайы	Телефон	Жинаған ұпай	4 пән	1 мамандық	2 мамандық
▶ 1	Асылханов	Абылай	Талғатулы	784787458478	Мақатаева 54	78785484514	89	23	5B070400	5B070400
2	Шалабаев	Арнур	Маратович	123456789999	Ақсай 5 дом 45 кв 8	87775945994	94	23	5B070400	5B070400

Сурет 3.23 – Комиссия өткізген құжаттар бөлімі

adminControl



Алматы энегетика және байланыс университеті

Қабылдау комиссиясы Құжаттар бөлімі

Құжаттар

Бакалавр
 Магистранттар
 Қарастырылатын құжаттар

Деректер қорынан іздеу

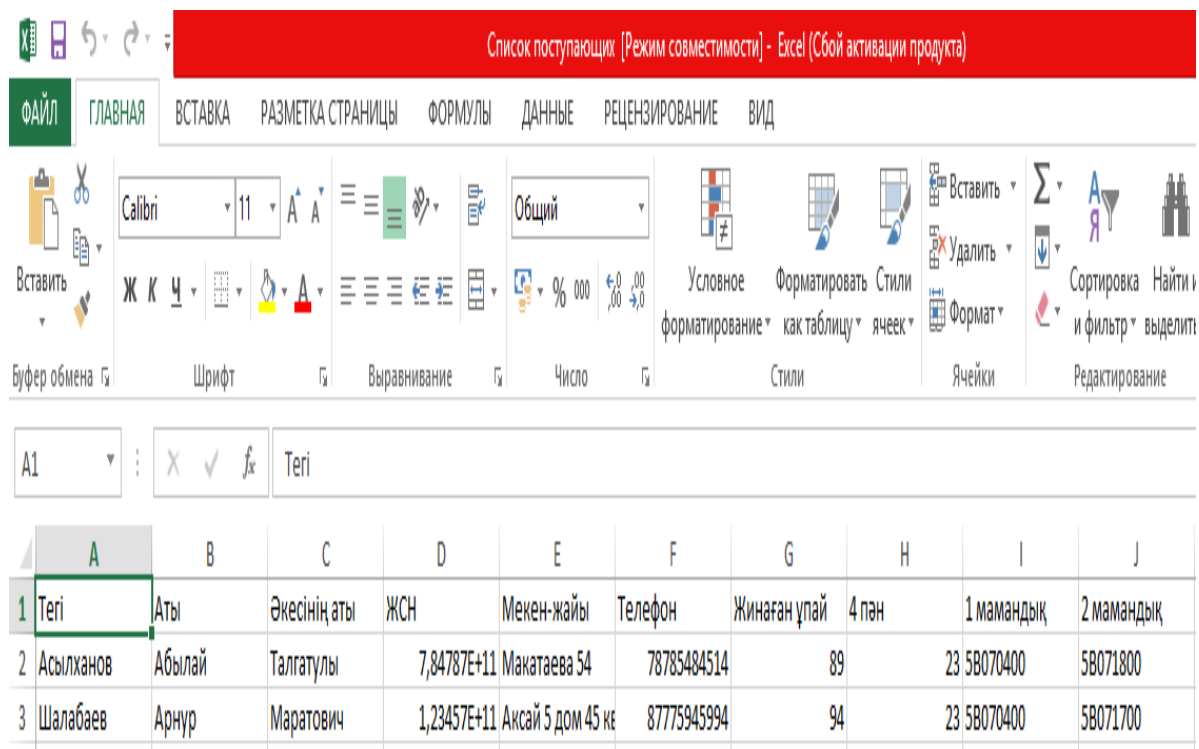
Факультет: Ақпараттық технологиялар факультеті (АТФ)

Тегі: Д

Excell жүктеу

	Тегі	Аты	Әкесінің аты	ЖСН	Мекен-жайы	Телефон	Жинаған ұпай	4 пән	1 мамандық	2 мамандық
▶ 1	Достанов	Абзал	Сайлаубайұлы	512658548625	Ақсай 2 дом 42 кв8	87077000493	94	24	5B070400	5B070400

Сурет 3.24 – Комиссия қарастыруға жіберілген құжаттар



Сурет 3.25 – Енгізілген құжаттарды Excel-ге жүкте

Құжаттарды сақтау (Сурет 2.26 – 2.27).

Аюжанова Айжан	14.03.2014 10:52
Ақатай Дархан	28.05.2014 13:03
Альбина Бектенова	14.03.2014 10:52
Аскарова Альбина	14.03.2014 10:52
Асылханов Абылай	14.03.2014 12:03
Буланова Айым	14.03.2014 10:52
Достанов Абзал	28.05.2014 12:35
Есентаева Айым	14.03.2014 10:52
Есентаева Бакытгул	14.03.2014 10:52
Шалабаев Арнұр	14.03.2014 10:52

Айым Есентаева файлының мәліметтері:

- Дата-создания: 16.03.2014 23:21
- Размер: 449 КБ
- Файлы: 3ZmuydQdX_og images (1), figdell_3

Сурет 3.26 – Деректері енгізілген адамның атына папка ашылады



Сурет 3.27 – Папканың ішінде талапкердің құжаттары болады

4 Тіршілік қауіпсіздігі

4.1 Ауа алмасу (кондиционер) жүйесінің құрылғысы және есебі

Осы дипломдық жұмыста Алматы энергетика және байланыс университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған ақпараттық жүйесін құру. Қабылдау комиссиясы жұмыс істейтін бөлменің ауа алмасуын есептеу керек. Бағдарламаның дерекқоры SQL 2008 r2 жасалынып, Microsoft Visual Studio бағдарламалық ортасымен жасалынды.

Кафедра бөлмесі сурет 4.1 көрсетілген. Кафедра бөлмесінің ұзындығы 6 м, ені 4 м, ұзындығы 2 м екі терезе орналасқан бөлменің биіктігі 3 м тең. Кафедрада 2 адам жұмыс істейді, жұмыс уақыты – аптасына 5 күн, күніне 8 сағат. Кафедра бөлмесіндегі жарықтандыру дербес компьютердің жұмыс істегені байланысты болғандықтан жарықтандыруды есепке алу керек. Физиологиялық үрдістерге байланысты жарықтандыру психикалық жағдайларға әкелуі мүмкін. Кафедра оргтехникалық жабдықпен, компьютерлік құрылғылармен жабдықталған, сондықтан бөлменің персоналы артық жылулық сәулеленуге шалдығады. Сол себептен жұмысшылардың қолайлы еңбек жасауы

үшін микроклимат шарттарын қарастыру керек. Микроклиматтың шарттары бұзылса еңбекке қабілеттілігі төмендейді және көңіл күйі нашарлайды. Жұмысшылардың жұмысы ГОСТ 12.0.003–88. ССБТ [7] сәйкес категориясы жеңіл физикалық жұмыс үшін бірқалыпты микроклиматтық шарттар берілген. Бөлмеде жаз уақыт кезіндегі температура +26°C-ге дейін көтеріліп, ал қыс кезіндегі температура +18-ден +20°C-ге жетеді. Бөлменің қалыпты температурасын сақтап тұру үшін кондиционерді орнатуымыз керек. Кафедра терезелері арқылы түсетін жарықтану болады, түнде жұмыс істеу үшін жасанды жарықтандыруды қолданамыз. Жасанды жарықтану жоғары дәлдікті көру жұмысының III, а разрядының талаптарына сәйкес келеді. Жасанды жарықталу жаңа шамдар арқылы жүзеге асырылады.

Жұмысшылардың қауіпсіздігін қамтамасыз ету үшін бөлме қызметкерлерінің денсаулығына әсер ететін барлық мүмкін факторларды қарау қажет.

Кафедрада құрылғылардың осындай түрлері қолданылады. Қолданатын компьютерлер саны – 2. Оператор компьютерден алысырақ отыру керек, себебі зиянды электромагниттердің сәулелері көп болады. Мониторға қорғаныс экранын орнату керек. Бөлмедегі шаннан, газдан құтылу үшін арнайы құрылғыларды қолдануымыз керек. Көрермен залы мен дыбыстық қамтамасыз етудің аппараттық бөлмесі арасындағы әуе шуының изоляция индексі 50дБ-ден кем болмауы керек. Дыбысты қамтамасыз етудің аппараттық бөлмесінің қабырғалары мен төбесі 500 – 2000 Гц жиіліктер диапазонында дыбысты жұту коэффициенті 0.6–дан кем болмайтын дыбысты жұтқыш материалдармен қапталуы керек. Дыбысты қамтамасыз ету жүйесінің барлық техникалық аппараттық бөлмелерінің едендері шаң тудырмайтын болмауы және күнделікті ылғалды жинастыру жұмыстарын өткізуге мүмкіндік беретін (метлах тақтасы, линолеум) болуы керек.

Дербес электрондық есептеуіш машина қолданушысының жұмыс орнын ұйымдастыруда келесі негізгі талаптар сақталуы қажет:

- жұмыс орнының құрамына кіретін құрылғылардың оптималды орналасуы;
- барлық қажет қозғалыстар мен орын ауыстыруларды жүзеге асыруға мүмкіндік беретін жеткілікті жұмыс аймағы;
- қызметтерді іске асыру үшін табиғи және жасанды жарықтандыру қажет;
- қатты шудың деңгейі рұқсат етілген нормадан аспауы керек.

Бөлмеде келесі құрал-жабдық қолданылады:

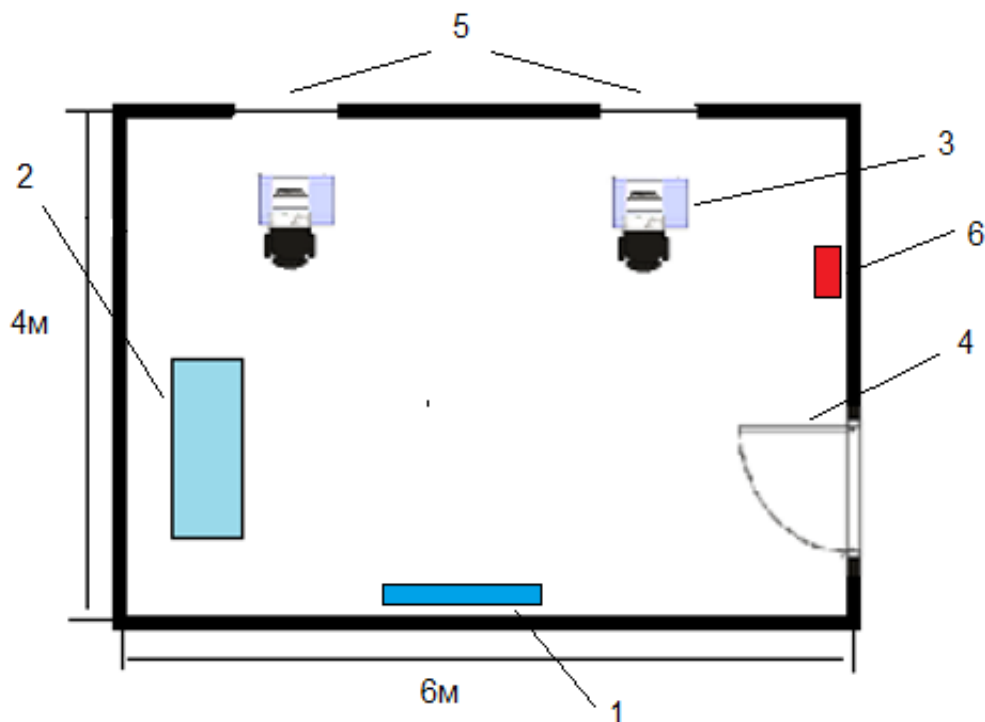
1 Дербес компьютер – 2 дана.

Құрылғының техникалық сипаттамалары:

- Asus core i7 3210/4Gb/250Gb/Combo/DOS дербес компьютері;
- SAMSUNG LS19A100N монитору;
- мөлшерлер 1200x750x1150 мм (дербес компьютер+үстел);
- электрлік қоректену көзі: айнымалы кернеу 220–250 В, 50 Гц жиілігі,

қуаты 400 Вт.

2 Сплит-жүйе плазма Samsung CS/CU-E 18 NKDW– кондиционер, қуаты 5 кВт (Сурет 4.1).



1 –кондиционер, 2 –орындық, 3 –үстел және дербес компьютер, 4 –есік, 5 – терезе, 6 –өрт сөндіргіш
Сурет 4.1 – Кафедра бөлмесі.

ГОСТ 12.0.003–88. ССБТ [7] сәйкес категориясы I а жеңіл физикалық жұмыс үшін қалыпты микроклиматтық шарттар келтірілген (Кесте 4.1).

К е с т е 4.1 – Микроклимат параметрлерінің қалыпты нормалары

Жыл мезгілі	Жұмыс категориясы	Температура, °С	Ауа қозғалысының жылдамдығы, м/с
Салқын	I а	18-26	0,1
Жылы	I а	20-30	0,2

Бөлменің микроклимат шамалары жыл мезгіліне байланысты болады. Суық кездері ауаның жылдамдығы 0.1 м/с, салыстырмалы ылғалдылығы 60 %, ауа температурасы 18–26°С тең болады.

Жаз мезгілінің жылы кездерінде ауа қозғалысының жылдамдығы 0.2 м/с, салыстырмалы ылғалдылығы 60–70% тең. Ұсынылған шамалар адам

организміне қолайлы нормаларды қанағаттандырмайды. Сондықтан кафедра бөлмесінде ауаны кондиционерлеуді қарастыру қажет.

Электр тоғының адамға әсер ететін зиянды мөлшерін анықтайық:

– едендер бір қабатты линолеуммен қаптаған дұрыс, себебі ол ток өткізбейді;

– Кафедра бөлмесі құрғақ болады себебі ауаның салыстырмалы ылғалдылығы 60% –дан аспауы керек;

– Цельсий бойынша плюс 30 градустан аспайды;

– адамның бір уақытта бір жақтан жермен байланысы бар технологиялық жабдықтардың корпустарымен және басқа жерлендірілген бөліктермен, екінші жақтан электр жабдықтарының металл корпустарымен немесе ток өткізуші бөліктермен жанасу мүмкіндіктерінің болмауы (кернеу 1000В мәнінен аспағандықтан сымдардың өте жақсы изоляциясында);

– зиянды химиялық заттар болмайды.

ГОСТ 12.1.013–78. ССБТ[7] сәйкес осы бөлмені маңызды қауіпсіздік жоқ бөлме ретінде классификациялауға болады.

ГОСТ 12.1.030–81 [7] электр қауіпсіздігін қамтамасыз ету үшін қолданып, жерлендіруді есептеуіміз керек. Кернеу – 220В, сондықтан жерлендіру мен нөлдеу міндеттелмейді, бірақ ұсынылады.

Бөлмені салу үшін пеноблок, әйнек, металл, темір және тағы да басқа материалдармен салынған. Бөлменің сыртын жанбайтын материалдармен қаптап, өртке қарсы өртсөндіргіш орнату керек. Бөлмелердің сыртында өрт крандары орналастыру қажет, олар баспалдақтардың қасында орналасу керек. Оператор бөлмелерінде архивті не компьютерді сақтап қалу үшін өрт болған жағдайда, өртті сөндіру үшін кәдімгі су қолданамыз. Қолданған кезде, дербес электрондық есептеуіш машина бар бөлмелерде, ақпаратты тасушыларды сақтау орындарында, аса қымбат құрылғыларды бұзу немесе толықтай істен шығару қаупінен бақылау-өлшеуіш жабдықтары бар бөлмелерде суды қолдану тек кейбір жағдайларда ғана рұқсат етіледі. Судың мөлшері өте аз болуы және дербес электрондық есептеуіш машинаны, дыбыстық құрылғыларды ластикпен немесе матамен жауып судан қорғау керек.

Бөлмелердің барлығын өрттен қорғану үшін стационарлы автоматты өрт өшіргіш қондырғылармен жабдықтылуы қажет. Ауаның құрамындағы оттегіні азайтатын өрт өшіргіш газбен, газбен бөлмені бірден толтыруға негізделген өшіру қондырғыларын қолданған тиімдірек болып табылады.

Ғимаратқа сырттан келетін зиянды заттарды қарастырып, ең алдымен бөлменің орналасқан жерін қарастыру керек. Менің жағдайымда ол жол бойында орналасқан. Жұмыскерлер бөлмесінде даладан бөлмеге келетін жиі ластағыштар шаң, қорғасын, азот және тағы басқалары болып саналады.

Бөлменің қаңқасы торон мен радоннан өте көп ластанады, сонымен қатар ауа концентрация нашар баратын ғимараттар көбінесе бетоннан жасалған үйлерде кездесіп жатады.

Компьютер қолданушысының, басқарушының, жұмыс орнындағы шу көздеріне осылар жатады:

- сөйлесіп тұрған адамдар;
- компьютердің дауысы;
- желдеткіш қондырғының шуы.

Осылардың бәрі шағымсыз шуды тудырады, сол себепті бөлмеге дыбысты жұтқыштар қолдануға болады.

Дыбысты жақсы жұту қасиеті талшықты-ауа көлемді материалдарда: фибролитті плиталарда, ыдыстарда, минералды мақтада, полиуретанды поропластта, ауа көлемді поливинилхлоридте және басқаларында болады. Негізінен дыбыс жұтқыш материалдарға дыбыс жұту коэффициенті 0,3 мәнінен төмен емес материалдар жатады.

Компьютерлік электрондық есептейтін машинамен жұмыс істеу үшін арналған бөлменің жарықтандырылуы табиғи және жасанды жарықтың болуы есебінен жасалынып, жасанды жарықтандыру тиімді болады.

Бөлмеде жарықтандыру жеткіліксіз болса ол адамның тез шаршауына әкеліп, орындайтын жұмыс сапасы төмендейді. Сол себептен қателердің саны көбейеді. Егер бөлмеде жарықтандыру көп болса адамның көзі ауырады, көз қабықшасы қышиды, жұмыс барысында жарақат саны көбейуі мүмкін.

Қарастырып отырған бөлмемізде жұмыс жасау үшін жарықтандыру норма бойынша $E_n=300$ лк, осылайша бөлмедегі қажетті жарықтандыруды қамтамасыз етуіміз керек.

Оператор бөлмесінде жақсы кондиционерлеу және вентиляция жүйесі жоқ. Бөлменің температурасын бірқалыпты ұстап тұру үшін микроклиматтың шарттарын ұстап отырып қарастыру қажет. Мұндай әмбебап жүйе ретінде автономдық кондиционерлер болып табылады.

ГОСТ 12.1.005–88 ССБТ "Жұмыс істеу аймағының ауасы, жалпы санитарлы-гигиеналық талаптар" сәйкес, компьютерлермен жабдықталған бөлмедегі адамдардың жұмысы жеңіл физикалық жұмысқа жатады. Ағзаның энергия жұмсау жұмыстарының категориялары кесте 4.2 келтірілген.

К е с т е 4.2 – Адам ағзасының энергия жұмсау жұмыстарының категориялары

Жұмыс	Категория	Ағзаның энергия жұмсауы, Ккал/сағ, Дж/с	Жұмыс сипаттамасы
Жеңіл	I a	<138	Жұмыс отырып жүргізіледі

4.2 Кондиционерлеу және ауаны жаңарту жүйелерін қарастыру

Қазіргі кезде жұмыс орындарында жұмысшыларға қолайлы жағдай жасалуына байланысты көптеген жұмыс орындарында кондиционерді қолдануда. Кондиционер ауаны тазартып отырып, бөлмедегі температураны

бірқалыпты жағдайда ұстап отырады. Өзінің құрылысы бойынша кондиционер жүйесі қолайлы және технологиялық болып екіге бөлінеді. Қолайлы жүйелер үйлесімді санитарлық-гигиеналық талаптарға жауап беретін ауаның температурасын, ылғалдылығын, тазалығы мен қозғалыс жылдамдығын жасау және автоматты қолдау үшін тағайындалады. Кондиционердің технологиялық жүйелері белгілі бір өндірістік және технологиялық процесс талаптарына басым дәрежеде жауап болатын ауа параметрлерін қамтамсыз ету үшін тағайындалады.

Заманға сай аумағы 18 –тен 130 м² дейінгі тұрғын және қоғамдық бөлмелерде сплит-жүйе кондиционерлері кең түрде қолданады. Негізінде олар сыртқы блоктан (компрессорлы-конденсаторлы) және ішкі блоктан (буландырғыш) тұрады. Атап кетсек сыртқы блок бөлменің қапбырғасына, шатыр астына немесе шатырдың өзіне, балконға бір сөзбен айтқанда конденсатор температурасы төмен жерлерде орналасуы керек. Ішкі блок тікелей кондиционерленетін бөлмеге орнатылады және ауаны салқындату немесе жылыту, сүзу және бөлмедегі ауаның қажетті қозғалысын жасау үшін тағайындалады. Ішкі блоктар берілген температураны ұстайды және бөлмедегі ауаның тең бөлінуін қамтамсыз етеді және шусыз жұмыс істейді (шу деңгейі 35-38 дБ). Кондиционерленген ғимараттың жылулық және ылғал теңгерімін белгілі әдістермен орындалады. Мұнда бөлменің ауа ортасының қалпы өзгеруіне әкеп соғатындай етіп, барлық факторлар есепке алынуы қажет.

Бөлмеге кондиционерді таңдау үшін ең бастысы жылудың қосындысын, содан кейін күннен түсетін радиациясынан бөлінетін жылуды есептеп алу керек. Содан кейін жұмыс орнында қанша адам жұмыс істейтіндігін біліп алуымыз қажет. Бөлмеде салқын бөлінудің қосындысы бойынша шамалы үлкен мәнді таңдап алып сол мәннен ауа алмасуын қамтамсыз ететін кондиционер моделін таңдап аламыз.

Бөлменің жылулық балансы мына формуламен есептелінеді

$$Q_{\text{жылу.б}} = Q_{\text{қоршау}} + Q_p + Q_a^a + Q_{\text{жарықтану}} + Q_{\text{құрал}} \quad (4.1)$$

мұнда $Q_{\text{қоршау}}$ – температура айырымы нәтижесінде алынатын жылу және жылу жоғалту;

Q_p – шынылау арқылы күннің сәулеленуінен келетін жылу;

Q_a^a – адамдардан келетін жылу түсу;

$Q_{\text{жарықтану}}$ – жарықтандыру аспаптарынан келетін жылу;

$Q_{\text{құрал}}$ – оргтехника және құрылғылардан келетін жылу.

4.3 Температура айырымы нәтижесінде алынатын жылу және жылу жоғалту

Әйнек арқылы күннің радиациясынан түсетін жылу.

Күннен бөлінетін жылу әйнектің моделіне байланысты 93%-ға дейін бөлме ортасымен жұтылады, одан қалған бөлігі шағылысады. Ең үлкен жылу

жүктемесі тура және шашырай түсетін күн сәулесінің ең үлкен деңгейінде алынады. Сәуле түсу қарқыны жергілікті кеңдікке, жыл мезгіліне және тәулік уақытына байланысты болады.

Суық мезгіл үшін есептік сыртқы температура ($t_{\text{сырт.есеп}}$) ең салқын айдың 14 сағатындағы орташа температурасына, жылы период үшін – ең ыстық айдың 14 сағатындағы орташа температурасына сәйкес келеді. Ал ішкі ($t_{\text{іш.есеп}}$) жайлылық шартын және өндірістік процесстерде көрсетілетін технологиялық талаптарын ескере отырып таңдалады [10]

$$Q_{\text{қоршау}} = V_{\text{бөлме}} * X_0 * (t_{\text{шыққан}} - t_{\text{келген}}) \quad (4.2)$$

мұнда $V_{\text{бөлме}}$ – бөлменің көлемі, м^3

$$V_{\text{бөлме}} = 6 * 4 * 3 = 72 \text{ м}^3$$

X_0 – меншікті жылулық сипаттама, $\text{Вт}/\text{м}^3\text{°C}$

$$X_0 = 0,42 \text{ Вт}/\text{м}^3\text{°C}$$

Жылдың жылы мезгіліне арналған сыртқы есептік температурасы

$$t_{\text{сырт.есеп}} = 27,6^\circ$$

Жылдың суық мезгіліне арналған сыртқы есептік температурасы

$$t_{\text{сырт.есеп}} = -25^\circ$$

Жылдың жылы мезгіліне арналған ішкі есептік температурасы

$$t_{\text{іш.есеп}} = 24^\circ$$

Жылдың суық мезгіліне арналған ішкі есептік температурасы

$$t_{\text{іш.есеп}} = 20^\circ$$

Жылы мезгіл үшін

$$Q_{\text{қоршау}} = 72 * (27,6 - 24) * 0,42 = 108,86 \text{ Вт}$$

Салқын мезгіл үшін

$$Q_{\text{қоршау}} = 72 * 0,42 * (-25 - 20) = -1360,8 \text{ Вт}$$

Күн сәулесінен шанылу арқылы келетін жылуды есептеу

Күннің сәулеленуінен (радиация) келетін жылу терезе арқылы сәуле бөлмеге кіріп, күннен шынылау сәулелену периоды үшін

$$Q_p = (q_{\text{тура}} + q_{\text{шашыр.}}) * K_1^c * K_2 * \beta_{\text{ж.ө.}} * n * H_0 * B_0 \quad (4.3)$$

Күннің сәулелері терезеден кірмейтін көлеңке периоды үшін (шашыраңқы радиация)

$$Q_p = q_{\text{шашыр.}} * K_1^T * K_2 * \beta_{\text{ж.ө.}} * n * H_0 * B_0 \quad (4.4)$$

мұнда $q_{\text{тура}}$;

$q_{\text{шашыр.}}$ – тура және шашыраңқы радиациядан келетін жылулық ағындар, Вт/м²;

F_0 – жарықтық ойықтың ауданы, м² (n – терезелердің саны, биіктігі H_0 және ені B_0)

$$F_0 = n * H_0 * B_0$$

мұнда K_1 – қапсырмамен шынылаудың көлеңкелену коэффициенті (K_1^c – сәулеленген ойықтар үшін;

K_1^T – көлеңкедегі ойықтар үшін;

K_2 – шынылаудың ластану коэффициенті;

$\beta_{\text{ж.ө.}}$ – жылу өткізу коэффициенті.

1 Алматы қаласындағы институттың кафедра бөлмесіндегі шынылаудың ауданы, 44⁰ СШ [12, кесте 3]

$$F_0 = 2 * 2.5 * 2 = 10 \text{ м}^2$$

2 Шынылаудың бағыты: оңтүстік-шығыс (ОШ).

3 Ішінде жарық перделері бар. $\beta_{\text{ж.ө.}} = 0.4$ [12, кесте 4] деп қабылдаймыз.

Түске дейін ОШ үшін, яғни сағат 8-ден 13-ге дейін 44⁰ СШ ендікте тура радиацияның мәні (П) $q_{\text{тура}} = 387$ Вт/м² және шашыраңқы радиацияның мәні (Р) $q_{\text{шашыр.}} = 101$ Вт/м² тең [12, кесте 5]. 45 – 67⁰ СШ ендік диапазонында металды қапсырмалы екі қабатты шынылау үшін: $K_1 = K_1^c = 0.72$, егер ойық күнмен сәулеленген болса, яғни 9-10 және 13 – 14 сағат аралығында. Әйнектің бірқалыпты ластануы коэффициенті $K_2 = 0.9$ қабылданады.

Тура сәулелену периодында 9 бен 14 сағат аралығында есептелу мына формула арқылы жүреді

$$Q_p = (387 + 101) * 0.72 * 0.9 * 10 * 0.4 = 1265 \text{ Вт}$$

Ал көлеңкелену периодында 14 пен 20 сағат аралығында мына формуламен есептелінеді

$$Q_p = 22 * 1.15 * 0.9 * 10 * 0.4 = 91 \text{ Вт}$$

Максималды есептелу уақыты: 9–10 сағат, жылу түсу 1265 Вт.

Адамдардан тарайтын жылу.

Адамдардан тарайтын жылу қоршаған ауа параметрлеріне және орындалатын жұмыс қарқынына байланысты. Адам бөлетін жылу ауаға конвекция арқылы сезілетін және өкпеден, теріден бөлінетін байқалмайтын жылудан тұрады. Адамдардың жылу таратуы к е с т е 4.3 сипатталады [10].

К е с т е 4.3 – Адамның сыртқы ортаға жылу таратуы

Сыртқы орта температурасы °С	Отырғандағы жағдай			Тұрғанда немесе жеңіл қозғалыс			Ауыр жұмыс		
	Анық	Жасырын	Жалпы	Анық	Жасырын	Жалпы	Анық	Жасырын	Жалпы
24	67	35	102	72	60	132	95	154	249
20	82	21	103	92	42	133	140	110	250

Бөлмедегі адамдардың бөлетін анық жылуы (Кесте 4.4).

К е с т е 4.4 – Адам бөлетін ылғал және көміртегі саны

Параметрлер	Бөлме ауасының температурасындағы мәндер °С				
	15	20	25	30	35
Ылғал г/сағ	40	40	50	75	115
Көміртегі қостотығы г/сағ	45	45	45	45	45

Кафедрада 2 әйел адам-оператор жұмыс істейді. $t = 24 \text{ }^{\circ}\text{C}$ температурада отырған күйде бір ер адам 67 Вт анық жылу, ал жалпы – 102 Вт жылу бөледі [12, кесте 8]. Әйел адам ересек ер адамның жылу бөлу нормасының 85 %-ын, ал кішкентай бала – 75 %-ын бөледі деп саналады. Бөлмедегі адамдардың бөлетін анық жылуы

$$Q_a^a = 67 * 2 * 0.85 = 114 \text{ Вт}$$

Ал жалпы жылу

$$Q_a^ж = 102 * 2 * 0.85 = 173 \text{ Вт}$$

$t = 20 \text{ }^{\circ}\text{C}$ температурада бір ер кісі 82 Вт анық жылу және 103 Вт жалпы жылу бөледі [12, кесте 8].

Бөлмедегі адамдардың бөлетін анық жылуы

$$Q_a^a = 82 * 2 * 0.85 = 139 \text{ Вт}$$

Ал жалпы жылуы

$$Q_a^ж = 103 * 2 * 0.85 = 175 \text{ Вт}$$

$t = 24 \text{ } ^\circ\text{C}$ үшін ылғалдылық және көміртегі қышқылының мөндерін 9-кестеден [12] интерполяция жолымен табамыз. Бір адамнан 50 г/сағ ылғалдылық, 45 г/сағ көміртегі қышқылы бөлінеді. Ал 2 адамның ылғалдылығы 100 г/сағ, көміртегі қышқылы мөлшері 90 г/сағ құрайды.

$t = 20 \text{ } ^\circ\text{C}$ үшін 1 адамнан бөлінетін ылғалдылық – 40 г/сағ, көміртегі қышқылы – 45 г/сағ. 2 адамнан бөлінетін ылғалдылығы 80 г/сағ. 2 адамнан бөлінетін көміртегі қышқылы мөлшері 90 г/сағ (К е с т е 4.5).

К е с т е 4.5– Бөлмедегі адамдардан бөлінетін зиянды заттардың есептелуінің нәтижелері

Жыл мезгілі	Температура $^{\circ}\text{C}$	Жылу, Вт		Ылғалдылық, W г/сағ	CO ₂ г/сағ
		Q _a ^a	Q _a ^ж		
Жылы	24	114	173	100	90
Салқын	20	139	175	80	90

Жарықтану аспаптарынан, оргтехникадан және құрылғылардан келетін жылу.

Шамдардан келетін жылу мына формуламен есептеледі [10]

$$Q_{\text{жарықтану}} = \eta * N_{\text{жарықтану}} \quad (4.5)$$

мұнда η - электр энергиясының жылулыққа ауысу коэффициенті.

$N_{\text{жарықтану}}$ – шамдардың орнатылған қуаты 65 Вт/м²

Кафедраның еденінің ауданы

$$F_{\text{еден}} = 6 * 4 = 24 \text{ м}^2,$$

$$Q_{\text{жарықтану}} = 0,6 * 65 * 24 = 936 \text{ Вт}$$

Оргтехниканың әсерінен пайда болатын жылу ағыны бір компьютерге орташа есеппен 300 Вт алады. Кафедрада 2 дербес компьютер болғандықтан

$$Q_{\text{құрал}} = 2 * 300 = 600 \text{ Вт}$$

Орындалған есептеулерден (4.1) формуласы бойынша кафедра бөлмесіне келетін жылу балансын құрамыз. Жылдың жылы мезгілінде: температура айырымы нәтижесінде келетін жылу $Q_{\text{коршау}} = 108$ Вт; күн радиациясынан $Q_p = 1265$ Вт; адамдардан $Q_a^a = 139.4$ Вт; жарықтану аспаптарынан $Q_{\text{жарықтану}} = 936$ Вт; оргтехника мен құрылғылардан $Q_{\text{құрал}} = 600$ Вт. Кафедраның жылулық балансы жазда

$$Q_{\text{жылу.б}} = Q_{\text{коршау}} + Q_p + Q_a^a + Q_{\text{жарықтану}} + Q_{\text{құрал}}$$

$$Q_{\text{жылу.б}} = 1265 + 113 + 108 + 936 + 600 = 3023 \text{ Вт} = 3,023 \text{ кВт}$$

$$Q_{\text{жылу.б}} = 3.023 * 3600 = 10883 \text{ кДж/сағ құрайды}$$

Жылдың салқын мезгілінде: температура айырымы нәтижесінде жоғалатын жылу $Q_{\text{коршау}} = -1360.8$ Вт; күн радиациясынан келетін жылу $Q_p = 1265$ Вт; адамдардан $Q_a^a = 139$ Вт; жарықтану аспаптарынан $Q_{\text{жарықтану}} = 936$ Вт; оргтехника және құрылғылардан $Q_{\text{құрал}} = 600$ Вт. Кафедраның жылулық балансы қыста

$$Q_{\text{жылу.б}} = Q_{\text{коршау}} + Q_p + Q_a^a + Q_{\text{жарықтану}} + Q_{\text{құрал}}$$

$$Q_{\text{жылу.б}} = 1265 + 139 - 1361 + 936 + 600 = 1579 \text{ Вт} = 1.579 \text{ кВт}$$

$$Q_{\text{жылу.б}} = 1.579 * 3600 = 5684 \text{ кДж/сағ құрайды}$$

Ауа алмасуды есептеу

$Q_{\text{жылу.б}}$ жазда $> Q_{\text{жылу.б}}$ қыста болғандықтан, $Q_{\text{жылу.б}}$ жазда мәнімен ауаның жылу кернеулігін мына формуламен есептейміз

$$Q_k = \frac{Q_{\text{жылу.б}} * 860}{V_{\text{бөлме}}} \quad (4.6)$$

$$Q_k = \frac{3.023 * 860}{6 * 4 * 3} = \frac{36.1 \text{ ккал}}{\text{м}^3}$$

$Q_n > 20$ ккал/м³ болғанда $\Delta t = 8^\circ \text{C}$

Бөлмеге қажет ауаның мөлшері жылулық баланстан алынып, мына формуламен анықталады

$$L = \frac{Q_{\text{жылу.б}} * 860}{c * \Delta t * \gamma} \quad (4.7)$$

$$L = \frac{3,023 * 860}{0,24 * 8 * 1.206} = 1123 \text{ м}^3 / \text{сағ}$$

мұнда $C = 0.24$ ккал/кг⁰C – ауаның жылу сыйымдылығы;
 $\gamma = 1,206$ кг/м³ – ағынды ауаның сыбағалы массасы.

Барлық артық жылулар 10883 кДж/сағ немесе $10883 : 3600 = 3,02$ кВт құрайды. Бөлмеге қажетті ауа мөлшері $L = 1123$ м³/ч = $31,8$ м³/мин. Өз таңдауымызды Samsung CC/CU-E 18 сплит-жүйесі кондиционеріне тоқтатамып, орнатуымыз қажет. Кафедра бөлмесінде кондиционер қондыруымыз керек. Жұмысшылардың еңбек нәтижелерін арттыру үшін, жұмысшылардың денсаулығына микроклиматтық жағдайлар әсерін тигізбеу керек.

Кондиционердің техникалық сипаттамалары (Сурет 4.2):

- салқын 5.20 кВт; жылу 5.80 кВт;
- қорек кернеуі $220\text{В}, 50$ Гц;
- салқынның жұмсайтын қуаты, кВт $1,47$;
- жылудың жұмсайтын қуаты, кВт $1,54$
- салқын/жылу жұмыс тогы, А $2,3 / 3,1$;
- EER, А 4.36 ;
- COP, А 4.41 ;
- жылдық ток пайдалануы 940 кВт*сағ
- шудың деңгейі, ішкі (жоғ/орт/төм), дБ(А) $44/37/34$;
- шудың деңгейі, сыртқы, дБ(А) 47 ;
- габаритті өлшемдері, Ш/В/Г, Ішкі, мм $290*1070*240$;
- габаритті өлшемдері, Ш/В/Г, сыртқы, мм $695*875*320$;
- 2 жылға кепілдік;
- қосымша параметрлері;
- ТТ, А 4.43
- TER, 5.53
- салмағы, кг 11 .



Сурет 4.2–Кондиционер суреті

5 Техника – экономикалық негізделуі

5.1 Бағдарламалық қамтаманың артықшылығы мен сипаттамасы

Деректер қорын жобалауға арналған бұл бағдарлама оқу орындарындағы қабылдау комиссиясының ақпараттармен жұмыс жасауын жақсартатын бағдарлама. Бағдарлама деректер қорына қызмет түрлерін енгізу, олар жайлы мәліметтер алу, жұмыс барысын бақылау, қызметкерлерді енгізу және т.б мәліметтерді алып отыруға мүмкіндік береді. Қазіргі таңда оқу орындарының жұмысын басқаруға қажетті бағдарлама болып табылады. Бағдарлама қабылдау комиссиясының жұмысын ұтымды, тиімді және жеңіл етеді. Бағдарламаны қолдану арқылы қызметкердің уақытын ұтымды пайдалануға болады, ақпараттарды оңай ала-аласыз, деректерді енгізудегі қателіктердің алдын аласыз.

5.2 Маркетингтік стратегия

Қабылдау комиссия жұмысында тиімді түрде қолдануға болады. Бағдарлама жұмысының барысы оқу орындарына жұмысшыларына ақпаратты өңдеуде зор мүмкіндік тудырады (Кесте 5.1 – 5.2).

К е с т е 5.1 – Бағдарламаның SWOT – анализі

<i>Мықты жақтары</i>	<i>Әлсіз жақтары</i>
Бағдарлама мүмкіншіліктерінің көптігі; Қолжетімділігі; Қызмет бірақ рет орнатылады қосымша жаңартуларды жыл сайын қажет етпейді;	Бағдарлама арнайы оқу орындары үшін пайдалануға арналған. Басқа қызмет түрлері үшін тиімсіз болып табылады.
<i>Мүмкіндіктері</i>	<i>Қаупі</i>
Қосымша табыс көзіне қол жеткізу. Қабылдау комиссиясының жұмысын 100% жеңілдетеді.	Жоқ

К е с т е 5.2 – Маркетинг – микс элементі

Маркетинг-микс элементі	Анализ және қажет әрекеттер
Product	
Оқу орындарының деректер қорын жобалауға арналған бағдарлама	Жоспарланып отырғанымыз өзіне бірнеше комплексті бөлшектерді кіргізеді: Бұл қызмет түрі негізінен арнайы оқу орындарына сатылымға негізделген (дерекқоры орташа немесе үлкен оқу орындарына) ; Бағдарлама интерфейсі Microsoft Visual Studio ,C# тілдерінде орындалмақ; Дерекқоры MySQL ортасында орындалады.
Price	
Бастапқы баға	“Деректер қорын жобалауға арналған бағдарлама” қызметі негізінен мекеменің қажеттілік ерекшеліктерін ескере отырып ақысы аланады.
Promotion	
Жарнама	Интернетке жарнамаға беру.
Place	
Мекеме	Программаны 2 жылдың ішінде 3 жоғарғы оқу орнына сатамын.

5.3 Бағдарламаның сатылым жоспары

Біздің стратегия оқу орындарындағы қабылдау комиссиясының деректер қорын жобалауда қарапайым және қолдануға ыңғайлы, мүмкіндіктері жоғары бағдарламалық өнім ұсынып, нарықтағы өзіндік артықшылығын дамыту. Бұл біздің нақты бағытымызды анықтап және тұрақты тұтынушыларды қамтамасыз етеді. Қазіргі таңда оқу орындарында осындай бағдарламаның жетіспеушілігін сезінуде.

Бұл бағдарлама арнайы тапсырыс арқылы жасалғандықтан бірінші жылы ауарлық белгіні кең түрде тарату және тұрақты тұтынушылардың санын арттыру мақсатында жоспарлап отырмыз.

5.4 Жұмыс сипаттамасы және қажеттілік дәлелдемесі

Берілген дипломдық жобаның тақырыбы – «Алматы энергетика және байланыс университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған ақпараттық жүйесін құру» болып табылады.

Нарықтық экономика жағдайында әрбір кәсіпорын, оның ішінде жоғарғы оқу орындары бәсекелестігін арттырып, қаржылық мүмкіндігін неғұрлым жоғарлату қажет. Осыған жетудің бірден бір жолы жас талаптарды қабылдаудың ақпараттық жүйесін одан әрі жетілдіру болып табылады. Ақпараттық жүйенің бағдарламасы университет басқармасының ғана емес, жас талаптардың да сұраныстарына толық жауап беру керек. Ең негізгі қойылатын талап – ақпаратты тез алу және оларды дұрыс сараптап, топтастыру. Осы тұрғыдан қарастырғанда таңдап алынған дипломдық зерттеудің тақырыбы көкейтесті.

Ұсынылып отырған ақпараттық жүйенің экономикалық тұрғыдан қарастырғандағы тиімділігі, мүмкіншілігі төменде қарастырылады.

5.5 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері мына формула бойынша есептеледі

$$Q=q*c*(1+p) \quad (5.1)$$

мұнда q – бағдарламалық өнімдегі (бастапқы команда) операторлар саны, 1550-ға тең [1, 5 бет];

c – бағдарлама күрделілігінің коэффициенті;

p – бағдарламаны өңдеу кезіндегі коррекциялау коэффициенті, қойылған тапсырманың нақтылығына байланысты 0.12 [2, 9 бет] деп қабылдаймыз.

Бағдарлама күрделілігінің коэффициентін дипломдық жұмыстың экономикалық бөлімін орындауға арналған Методикалық нұсқаудан аламыз. Ол бойынша біздің жүйе 2 «Күрделілік топқа» – тіркеу тапсырмалары, статистика және есеп-қисап және «Жаңалық дәрежесі» бойынша Б – оригинальды бағдарламаын өңделуі болып табылады. Осы нұсқау бойынша бағдарламаның күрделілік коэффициенті $c = 1.37$ [1, 5 бет]. Осы әдіспен базалық көрсеткішті табамыз

$$Q=q*c*(1+p) = 1550*1.37*(1+0.12) = 2378$$

Ары қарай еңбек шығынының құраушы компоненттерін есептейміз: алгоритмді жазу мен дайындауға кеткен шығын, алгоритмді зерттеуге кеткен

шығын, алгоритмді өңдеуге кеткен еңбек шығыны, бағдарламалауға және құжаттаманы дайындауға кеткен еңбек шығыны. Бүкіл параметрлер базалық көрсеткішке байланысты болады.

Алгоритмді жазу мен дайындауға кеткен шығын эмпириялық немесе келесі формула бойынша анықталады

$$t_{on} = \frac{T_{\min} + 4 * T_{не} + T_{\max}}{6} \quad (5.2)$$

$$t_{on} = \frac{30 + 4 * 60 + 90}{6} = 60 \text{ адам/сағ}$$

мұнда T_{\max} – операцияның ең қолайлы емес уақыттағы еңбек өнімділігі (пессимистік баға);

T_{\min} – операцияның қолайлы уақыттағы еңбек өнімділігі (оптимистическая баға);

$T_{не}$ – операцияның қалыпты уақыттағы еңбек өнімділігі (ықтималдылығы жоғары баға). Еңбек өнімділігінің бағытталған мәнінің q операторына тәуелділігі $[A, 1]$ қосымшада көрсетіледі.

Тапсырманы шешу алгоритмін зерттеуге кететін еңбек шығыны мына формула бойынша анықталады

$$t_{uc} = \frac{Q * B}{(75 \div 85)k} \quad (5.3)$$

мұнда Q – базалық коэффициент(5.1) формуласы арқылы анықталады,

B – есепті шешу жеткіліксіздігінің коэффициенті, 1.35-ге тең [1, 5 бет];

k – программист біліктілігінің коэффициенті жұмыс өтілі 2 жылға дейін 0,8-ге тең деп табылды [1, 7 бет] (болашақ жас маман).

Осылайша тапсырманы шешу алгоритмін зерттеуге кеткен еңбек шығынын табамыз

$$t_{uc} = \frac{2378 * 1.35}{85 * 0.8} = 47 \text{ адам/сағ}$$

Алгоритмнің блок-схемасын жасауға кеткен еңбек шығыны осылайша есептеледі

$$t_{al} = \frac{2378}{25 * 0.8} = 119 \text{ адам/сағ}$$

Блок-схема бойынша алгоритмді бағдарламалауға кеткен еңбек шығыны осылайша есептеледі

$$t_{np} = \frac{2378}{25 * 0.8} = 119 \text{ адам/сағ}$$

$$t_{omi} = \frac{2378}{5 * 0.8} = 595 \text{ адам/сағ}$$

Құжаттарды дайындауға кететін еңбек шығыны қолжазба дайындауға кеткен еңбек шығыны мен құжатты әсемдеуге кеткен еңбек шығынымен анықталады

$$t_{\phi} = t_{pyk} + t_{o\phi} = \frac{Q}{(15 \div 20)k} + 0.75 * t_{pyk} \quad (5.4)$$

$$t_{\phi} = \frac{2378}{20 * 0.8} + 0.75 * \frac{2378}{20 * 0.8} = 149 + 111 = 260 \text{ адам/сағ}$$

Жалпы еңбек шығыны құрамдас еңбек шығындарының қосындысымен анықталады

$$t_{\Sigma} = t_{on} + t_{uc} + t_{al} + t_{np} + t_{omi} + t_{\phi} \quad (5.5)$$

$$t_{\Sigma} = 60 + 47 + 119 + 595 + 260 = 1081 \text{ адам/сағ}$$

5.6 Бағдарламалық қамтаманы өндеуге кеткен шығын есептелуі

Еңбек құны екі құрамнан құралады: негізгі еңбек құны және қосымша. Негізгі еңбек құны

$$Z_{ocn} = t_{\Sigma} * TC / (t_{cp} * 8) \quad (5.6)$$

мұнда t_{Σ} – есептелетін жалпы еңбек шығыны;

t_{cp} – айдағы орташа күн саны, 21 күнге тең; жұмыс күніндегі 8 сағатқа көбейтіледі;

TC – тарифтік қойылым.

Тарифтік қойылым МРОТ-ты білдіреді (еңбек ақының минимальды өлшемі), біздің мысалда ол 19966 д.ед. [ҚР-да МРОТ 01.01.2014ж-дан 19966 теңгені құрайды], [3].

Осылайша негізгі еңбек ақы

$$Z_{ocn} = 1081 * 19966 / (21 * 8) = 128472 \text{ теңге}$$

Қосымша еңбек ақы негізгі еңбек ақының 20% – ін құрайды [1,8 бет] осылайша анықталады

$$Z_{дон} = 0.2 * 128472 = 25694 \text{ теңге}$$

Жалпы еңбек ақысы (еңбек төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады

$$\Phi OT = Z_{дон} + Z_{осн} \quad (5.7)$$

$$\Phi OT = 128472 + 25694 = 154166 \text{ теңге}$$

5.7 Қосымша шығындар статьясы

Бағдарламалық қамтамасыз етудегі қосымша шығыстар: материалдар мен құрамындағыларға шығыс (құрылғының өз құны, яғни компьютерлер есепке алынбайды), социалды сақтандыруға аударым, жүкқұжат шығындар, амортизациялық аударым, техникалық қондырғылар қызметіне шығын, компьютермен жұмыс істеу кезінде кеткен электроэнергиялық шығындар.

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, қосымша шығындар статьясындағы есептеулерде қолданылады. ЭВМ-ге бағдарлама жазу кезінде құрылғы ретінде дербес компьютер қолданылады. Құны $C_{обор} = 114990$ теңгені құрайды.

Бағдарламалық өнімді жазу кезінде қолданылған материалдар мен құрамындағыларға шығын, сонымен қатар техникалық қызметпен жөндеуге сәйкесінше құрылғы құнынан алғанда 1,5% және 2.5% [2,6 бет]

$$C_{мик} = 0.015 * C_{обор} \quad (5.8)$$

$$C_{мик} = 0.015 * 114990 = 1725 \text{ теңге}$$

$$C_{ТО} = 0.025 * C_{обор} \quad (5.9)$$

$$C_{ТО} = 0.025 * 114990 = 2875 \text{ теңге}$$

ҚР-да $T_{норм} = 2.5$ жыл. Себебі Салық Кодексі бойынша компьютерлерге берілген норма – 40 % [1,9 бет]. Осы бойынша бізде амортизациялық аударымды есептедік

$$A_{зод} = 114990 * \frac{40}{100} = 45996 \text{ теңге}$$

Бағдарламалық қамтаманы жазу кезіндегі электроэнергияның шығынын қоса есептеу керек. Электроэнергияның құны мына формула бойынша есептеледі

$$C_{ЭЭ} = M * k_3 * F_{эф} * C_{кВт-ч} \quad (5.10)$$

мұнда M – ЭВМ қуаты (450 Вт); k_3 – жүктеу коэффициенті (0.8);

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны 14.47 теңге [А, 2];
 $F_{эф}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі.

$$F_{эф} = D_{ном} * d * (1 - f/100) \quad (5.11)$$

$$F_{эф} = 258 * 8 * (1 - 2/100) = 2023$$

мұнда $D_{ном} = 258$ – жылдағы жұмыс күнінің номиналды саны [1,9 бет];
 $d = 8$ – жұмыс күнінің ұзақтығы [сағ];
 $f = 2\%$ шамамен – ЭВМ-ды жөндеуге кеткен жоспарланған уақыт.
 Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны

$$C_{ЭЭ} = 450 * 0.8 * 2023 * 14.47 = 10538 \text{ теңге}$$

Дегенмен, алынған амортизациялық аударым мен электроэнергияға шығын – жылдық шығынның мәні, оны жалпы жылдық эксплуатациондық шығынмен анықталатын уақыт коэффициентімен коррекциялау керек

$$\mathcal{E}_3 = t_{\Sigma'} * C_3 / F_{эф}, \quad (5.12)$$

$$C_3 = C_{ЭЭ} + C_{ТО} + A_{год} \quad (5.13)$$

$$C_3 = 10538 + 2875 + 45996 = 59409 \text{ теңге}$$

мұнда $F_{эф}$ – жұмыс уақытының эффективті фонды, формуласы бойынша анықталды;

$t_{\Sigma'}$ - ЭВМ-нің есепті шешу үшін жалпы қолданылған уақыты, формуласы бойынша анықталған, компьютерде ғана істелген жұмыс уақыты.

$$t_{\Sigma'} = t_{np} + t_{омл} + t_{\delta} \quad (5.14)$$

$$t_{\Sigma'} = 119 + 595 + 260 = 974 \text{ сағ}$$

Сәйкесінше, ЭВМ-ның жалпы жылдық эксплуатационды шығынқұны

$$\mathcal{E}_3 = 974 * 59409 / 2023 = 28603 \text{ теңге}$$

Ал уақыт коэффициенті мына формула бойынша есептеледі

$$w = \frac{\mathcal{E}_3}{C_3}$$

(5.15)

$$w = \frac{\mathcal{E}_3}{C_3} = \frac{28603}{59409} = 0.4815$$

Осылайша, жалпы эксплуатационды шығыннан уақыт коэффициентін ескеріп коррекциялаймыз:

- Электроэнергияға шығын: $C_{\text{ЭЭ}}^* = 10538 * 0.4815 = 5074$ теңге;
- Амортизационды аударым: $A_{\text{зод}}^* = 45996 * 0.4815 = 22147$ теңге.

Сонымен қатар, (5.11) формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар.

Әлеуметтік аударымдар қазіргі таңда былайша жүргізіледі. Алдымен 5% мөлшерінде Әлеуметтік аударымдардың Мемлекеттік қорына есептелінеді. Содан кейін [11%, ҚР Салық Кодексі бойынша әлеуметтік салық есептелінеді], (5.11) формуласы бойынша анықталады [1, 8 бет]

$$\mathcal{E}C = (154166 - 0.1 * 154166) * 0.11 = 15262$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін қосымша шығындар еңбек жалақысы фондының 50% [1,10 бет] формуласы бойынша нақтыланады

$$C_{\text{накл}} = 0.5 * 154166 = 77083 \text{ теңге}$$

5.8 Өзіндік құн нәтижесінің кестесі

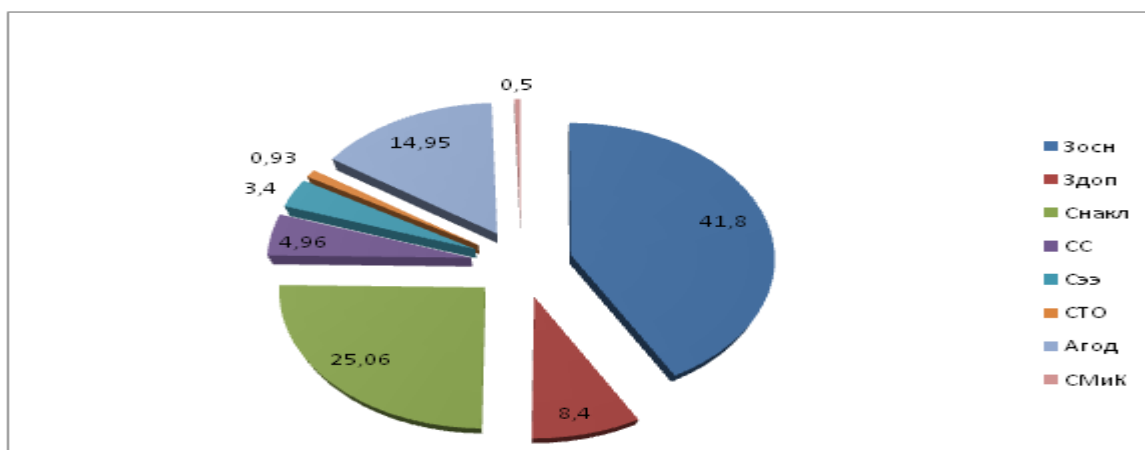
К е с т е 5.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сумма, д. ед	Жалпы суммадан процентпен
ФЗП	$Z_{\text{осн}}$	128472	41,8
	$Z_{\text{доп}}$	25694	8,4
Жүктеме шығындар, $C_{\text{накл}}$		77083	25,06
Социальды сақтандыру, CC		15262	4,96
Эксплуатационды Шығындар	$C_{\text{ЭЭ}}$	10538	3,4
	$C_{\text{ТО}}$	2875	0,93
	$A_{\text{зод}}$	45996	14,95

5.3 – кестенің соңы

Шығын статьялары	Сумма, д. ед	Жалпы суммадан процентпен
Материалдар мен жинақтар, $C_{Мик}$	1725	0,5
Нәтижесі:	307645	100

Өзіндік құн нәтижесінің диаграммасы (Сурет 5.1)



Сурет 5.1 – Өзіндік құн нәтижесінің диаграммасы

5.9 Бағдарламаның бағасын есептеу

Бағдарлама 1 өндіріс орнына енгізіледі деп жоспарланады.

Өнімнің өзіндік құны 307645 теңге болды делік, онда бір енгізу объектісіне арналған жүйенің толық өзіндік құны (C_{nc})

$$C_{nc} = 307645 / 3 = 102548 \text{ теңге}$$

Өнімнің минималды бағасы $r=25\%$ болған кезде [4]

$$C_{min} = 102548 * 1.25 = 128185 \text{ теңге}$$

Минималды баға келесі формуламен есептеледі (ҚР да НДС 12%)[5]

$$C_{min_{НДС}} = C_{min} + H_{ндс} * C_{min} \quad (5.16)$$

$$C_{min_{НДС}} = 128185 + 0.12 * 128185 = 143567 \text{ теңге}$$

5.10 Бағдарламаны енгізудегі шартты үнемдеуді есептеу

Жоғарыда көрсетілген есептеулерден бағдарламаны еңгізген кезде барлығы 6 адам (1081/168) қызмет атқаратындығын көруге болады. Енді бағдарламаны қолданбаған кезде әр факультетке 2 адамнан (2*4) + 1 ортақ кеңесші институт бойынша + 1 қабылдау комиссиясының төрағасы, барлығы кем дегенде 10 адам жас түлектерді қабылдауға жұмылдырылады.

Сонда институт үшін 4 адамды жұмыспен қамтамасыз етуге кеткен шығындар шартты үнемдеу болып табылады. Есептеп көрелік

$$1 \text{ ФЗП} = 79864 + 15973 = 95837 \text{ тенге}$$

$$Z_{\text{осн}} = 4 * 19966 = 79864 \text{ тенге}$$

$$Z_{\text{ооп}} = 79864 * 0.2 = 15973 \text{ тенге}$$

2 Жүктеме шығындар

$$C_{\text{накл}} = 95837 * 0.5 = 47919 \text{ тенге}$$

3 Социальды сақтандыру

$$CC = (95837 - 0.1 * 95837) * 0.11 = 9488 \text{ тенге}$$

4 Жалпы шартты үнемдеу сомасы

$$95837 + 47919 + 9488 = 153244 \text{ тенге}$$

Сондықтан бағдарламаны еңгізген кезде жұмысшылар санын қысқартуынан түскен шартты пайда 153244тг.тең.

Келесі фактор, бағдарламаны қолдану арқылы қызметкердің уақытын ұтымды пайдалануға болады, ақпараттарды оңай ала-аласыз, деректерді енгізудегі қателіктердің алдын аласыз. Сондықтан мәліметтерді өңдеуге жұмсалатын шығындардан да уақыт факторы арқылы шартты үнемдеу сомасын шығаруға болады.

Негізгі шығындар түрлері:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқада шығындар.

Есептеу жылына бір рет ПК операторымен 3000теңге/сағ сағаттық мөлшерде үш сағат көлемінде жүргізіледі [1, 12 бет]

$$C_o = 3000 * 3 = 9000 \text{ теңге}$$

Әлеуметтік салыққа жұмсалатын аударымдар(ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі)

$$O_{CH} = (9000 - 9000 * 0.1) * 0.11 = 891 \text{ теңге}$$

Электроэнергияға кететін шығындар

$$C_{ЭЭ} = 0.8 * 450 * 3 * 14.47 = 15.63 \text{ теңге}$$

мұндағы 0.8 – жұмсалған қуат мөлшері, кВт – сағ;

14.47 – 1 кВт – с электроэнергияның құны, теңге.

К е с т е 5.4 – Кәсіпорынның жылдық бірмезгілдегі шығындарын есептеу нәтижесі

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	9000
Әлеуметтік салыққа аударымдар	891
Электроэнергияға төлем	15.63
Қорытынды:	9907

5.11 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу

Үнемдеу мөлшерін есептейік. Ол үшін өткізу қабілетін есептеуге кететін кәсіпорынның шығындарын қолмен есептеу керек.

Қол еңбегімен келесі шығындар түрлері есептеледі:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқа да шығындар.

Қолмен есептеу барысында еңбек өнімділігін талдау үшін 3000 теңге/сағ мөлшерінде жұмыс істейтін мамандандырылған инженер қажет. Өткізу қабілетін есептеу үшін инженер 5 күн жұмсайды. Қолмен есептеу барысында еңбектің жылдық төлем қоры

$$C_o = 5 * 8 * 3000 = 120000 \text{ теңге}$$

Әлеуметтік салыққа жұмсалатын аударымдар(ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі) [1,8 бет]

$$\Theta C = (120000 - 120000 * 0.1) * 0.11 = 11880 \text{ теңге}$$

К е с т е 5.5 – Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығындарды есептеу

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	120000
Әлеуметтік салыққа аударымдар	11880
Қорытынды:	131880

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 9907 теңге болды.

Осыған сәйкес, шығындардың уақыт факторы арқылы шартты үнемделуі мынаған тең болады

$$\mathcal{E}_{\text{ж}} = 131880 - 9907 = 121973 \text{ теңге}$$

5.12 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу 3 оқу орны үшін

3 оқу орны үшін шартты пайда кесте 5.6 көрсетілген.

К е с т е 5.6 – Шартты үнемдеу

№	Көрсеткіштер	Қолданбаған кезде	Қолданған кезде	Жалпы экономия		
				Барлығы	Соның ішінде	
					1 фактор	2 фактор
1	ФЗП	361003	154166	-206837	-95837	-111000
2	Жүктеме шығындар	125002	77083	-47919	-47919	
3	Әлеуметтік шығындар	35739	15262	-20477	-9488	-10989
4	Эксплуатационды шығындар	59393	59409	+16		+16
5	Материалдар мен жинақтар	1725	1725			
	Өзіндік шығындар	582862	307645	-275217	-153244	-121973

Жоғарыда көрсетілген есептеулер бағдарламаны енгізу да, сатып алушыларға да тиімді екендігін көрсетеді.

Бағдарламаны ұсынушылар өзіндік құны 102548 (307645/3) теңге бағдарламаны 143567 теңге сатады (25% рентабельдік деңгейі + 12 % НДС).

Сатып алушыларға да тиімді, себебі әрбір оқу орны бағдарламаны 143567 теңге сатып алады да, 275217 теңге ақша қаражаттарын үнемдейді.

5.13 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу АЭЖБУ

АЭЖБУ үшін шартты пайда кесте 5.7 көрсетілген.

К е с т е 5.7 – Шартты үнемдеу

№	Көрсеткіштер	Қолданбаған кезде	Қолданған кезде	Жалпы экономия		
				Барлығы	Соның ішінде	
					1 фактор	2 фактор
1	ФЗП	258226	51389	-206837	-95837	-111000
2	Жүктеме шығындар	73613	25694	-47919	-47919	
3	Әлеуметтік шығындар	25564	5087	-20477	-9488	-10989
4	Эксплуатационды шығындар	19787	19803	+16		+16
5	Материалдар мен жинақтар	575	575			
	Өзіндік шығындар	377765	102548	-275217	-153244	-121973

Жоғарыда көрсетілген есептеулер бағдарламаны енгізу да, сатып алушыларға да тиімді екендігін көрсетеді.

Бағдарламаны ұсынушылар өзіндік құны 102548 (307645/3) тенге бағдарламаны 143567 тенге сатады (25% рентабельдік деңгейі + 12 % НДС).

Сатып алушыларға да тиімді, себебі әрбір оқу орны бағдарламаны 143567 тенге сатып алады да, 275217 тенге ақша қаражаттарын үнемдейді.

Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. АЭЖБУ университетінің қабылдау комиссиясының жұмысын автоматтандыруға арналған бағдарлама құрылды. Дерекқорды құру барысында барлық ерекшеліктер ескерілді.

Дерекқор MS SQL Server дерекқорларды басқару жүйесінде жасалды, жаңа түскен ақпаратқа байланысты деректерді тек құпия сөз арқылы енгізе алады. Мұндай қатынау принципі пайдаланушылар үшін онша ыңғайлы емес, сондықтан олар үшін Microsoft Visual studio 2013 ultimate программалық ортасында клиенттік қосымша жасалды. Сонымен қатар, құпия сөз қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

Бұл бағдарламалық өнімді кез келген жоғары оқу орнындағы кез келген кафедрада пайдалануға болады.

Тіршілік қауіпсіздігі бөлімінде кафедра бөлмесінің ауа алмасу (кондиционер) жүйесін есептедім. Мұнда сыртқы және ішкі микроклимат параметрлерін, температура, ылғалдылық және ауаның қозғалыс жылдамдығы, адам мен компьютерден бөлінетін жылу, зиянды химиялық заттар және радиацияны, бөдме көлемін ескере отырып кафедра бөлмесіне плазмалық Samsung CS/CU-E 18 NKDW сплит-жүйесі кондиционері таңдалып алынды.

Техника-экономикалық негізделуі нәтижесі бойынша бұл программаны құру ақшалай қаржыландыруды қажет етеді. Бағдарламаның өзіндік құны қосымша құн салығынсыз 102548 теңгені қамтиды. Шығындардың көп бөлігі 128472 жалақы төлеуге жұмсалады. Бағдарламаны жүзеге асыру бағасы **307645 теңге**.

Берілген бағдарламаны қолдану мынадай мүмкіндіктер береді:

- қабылдау комиссияның жұмысын жеңілдетеді;
- қабылдау комиссияның қателерін азайтады;
- жұмыс істейтін адамдардың санын азайтады.

АЭЖБУ университетіне арналған бағдарламаны құру қиын әрі ұзақ уақытты қажет етуіне қарамастан қабылдау комиссияның жұмысын толығымен автоматтандырады.

Пайдаланылған әдебиеттер тізімі

1. Гаврилова Т.А.,Хоропинский В.Ф. Базы знаний интеллектуальных систем. – СПб:Питер,2001.
2. Гареев А.,Корнеев В.,Райх В.,Васютин С. Базы данных.Интеллектуальная обработка информации. – М.:Нолидж,2003.
3. Гринберг А.С.,Горбачев Н.Н.,Бондаренко А.С.Информационные технологии управление:Учебное пособие. – М.:ЮНИТИ,2004.
4. Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных: Методическое указание к выполнению лабораторных работ. – Алматы: АИЭС, 2004.
5. Айтхожаева Е.Ж. Стандартный язык баз данных SQL: Учебное пособие. – Алматы: АИЭС, 2005. – 48 с.
6. Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург,2006.
7. Голицына Л.База данных.2002
8. Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
9. Глушаков С.В.,Ломатко Д.В.,База данных:Учебный курс.Издание Фолио:Харьков,2000.
10. Хомоненко А.Д.,Цыганков В.М. и др.Базы данных. – СПб,Корона,2000.
11. Мамаев Е.В.,Администрирование MS SQL 2008 2000. –СПб:Издание ВHV.
12. Зәуірбеков Н.С.,Сәрсенбай А.С.,Мәнжу М.Д.,Мәліметтер қорын жобалау.Оқу құралы/– Алматы:Экономика,2011. – 228 б.
13. Биярова А.Ө.,Иманбаев Қ.С.,Экономикалық ақпаратты жүйелер:Оқу құралы. – Алматы:Экономика,2011. – 456 бет.
14. <http://egov.kz>
15. <http://prepod2000.kulichki.net>
16. <http://samruk-energy.kz>
17. Смирнов Г.Н.,Сорокин А.А.,Тельноа Ю.Ф.проектирование экономических информационных систем:Учебник/Под ред.Ю.Ф.Тельнова. – М.:Финансы и статистика,2001.
18. Романов А.Н.,Одинцов Б.Е. Информационные системы в экономике:Учебное пособие. – М.:Вузовский учебник,2006.

А қосымшасы

Экономикалық бөлімге қажетті кестелер

Операторлар санын қарайтын (Кесте А 1).

К е с т е А 1 – q коэффициентінің мәні

Тапсырма түрлері	Коэффициенттің өзгеру аралығы
септеу тапсырмалары	1400 ден 1500
Оперативті басқару тапсырмалары	1500 ден 1700
Жоспарлау тапсырмалары	3000 ден 3500
Көп вариантты	4500 ден 5000
Комплекстік тапсырма	5000 ден 5500

Еңбек сыйымдылығын есептейтін (Кесте А 2).

К е с т е А 2 – Еңбек сыйымдылығын есептейтін коэффициент

Бағдарлама тілі	Күрделік тобы	Жаналықтық дәрежесі				В коэффициенті
		А	Б	В	Г	
Жоғарғы деңгей	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Төменгі деңгей	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

Алматы қаласы бойынша 3 раундты тарифтер (Кесте А 3).

К е с т е А 3 – Алматы қаласы бойынша 3 раундты тарифтер

Тариф	Мөлшерлеме	Сағат бойынша
Т-1	14,47	7.00-19.00
Т-2	30,31	19.00-21.00
Т-3	4,08	21.00-7.00

Б қосымшасы

Жобада қолданылатын бағдарламаның кодтары

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;
```

```
namespace abiturient  
{  
    public partial class abiturient : Form  
    {  
        public abiturient()  
        {
```

```
            InitializeComponent();  
            panel1.Visible = false;  
            string instr = " Сіз Мемлекеттік білім грантының (мемлекеттік тапсырыс)  
конкурсына қатысу үшін өтінішті толтыруда басым тәртіпте 4 мамандықты жазу құқыңыз  
бар. Алдымен, Сіз өзіңіз қалаған мамандықты жазасыз. Егер бірінші мамандық бойынша  
конкурстан өте алмай қалсаңыз, онда оқығыңыз келетін екінші мамандықты жазасыз. Содан  
соң, өтінішке үшінші, төртінші мамандықты жаза аласыз." + Environment.NewLine + "  
Өтініште келесі жағдайларға назар аудару қажет:" + Environment.NewLine + " Біріншіден,  
Сіз өтініште 1-4 мамандыққа дейін көрсете аласыз. Егер Сіз нақты бір мамандыққа  
бағдарлансаңыз, онда соны ғана жазсаңыз болады, бірақ бұл жағдайда сіздің мембюджет  
есебінен оқу мүмкіндігіңіз кемиді." + Environment.NewLine + " Екіншіден, әр таңдаған  
мамандық бойынша білім алғыңыз келген жоғары оқу орнын да көрсетіңіз. Ол бір ғана оқу  
орны болуы мүмкін (мысалы, Сіз көрсеткен мамандықтардың бәрі бір жоғары оқу орнында  
болуы мүмкін), сонымен қатар, әр түрлі жоғары оқу орындары да болуы мүмкін (мысалы,  
бірінші мамандық бойынша – АЭЖБУ, екінші мамандық бойынша – ҚазҰТУ және т.б.)." +  
Environment.NewLine + " Үшіншіден, барлық мамандықтар бойынша төртінші пән бірдей  
болуы тиіс (мысалы, АЭЖБУ мамандығы бойынша – физика)." + Environment.NewLine + "  
Республикалық конкурстық комиссия Қазақстан бойынша Мембюджет қаражаты есебінен  
оқу конкурсын ҰБТ және ТКТ нәтижелерін ескере отырып өткізеді." + Environment.NewLine  
+ " Егер Сіз конкурстан өте алмай қалсаңыз, онда толық ақылы негізде университетке түсе  
аласыз (тестілеуден кемі 50 бал алған шарт бойынша)." + Environment.NewLine + "  
Қабылдау комиссиясы" + Environment.NewLine + " Оқу жылының аяқталуы жақындаған  
сайын біздің оқырмандарымыздың бүгінгі білім беру саласының қарқынды дамуына, оның  
ішінде Ұлттық бірыңғай тестілеу мәселесіне ерекше қызығушылықтары артуда. ҰБТ бүгінгі  
күннің талаптарына, білім сапасын арттыру міндеттеріне, оның ішінде шараның ашық та  
таза, объективті өткізілуі жөніндегі ел Президенті Нұрсұлтан Әбішұлы Назарбаев қойған
```


талаптарға қаншалықты жауап бере алады? ҰБТ туралы бүгінгі көпшіліктің көкейінде жүрген сұрақтарға Білім және ғылым министрі, академик Бақытжан Жұмағұлов жауап береді.";

```
textBox1.Text = instr;
```

Б қосымшасы жалғасы

```
}

private void button1_Click(object sender, EventArgs e)
{
    new bakalavr1().ShowDialog();
}

private void button2_Click(object sender, EventArgs e)
{
    new magistratura().ShowDialog();
}

private void label3_Click(object sender, EventArgs e)
{
    panel1.Visible = true;
}

private void button3_Click(object sender, EventArgs e)
{
    panel1.Visible = false;
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace abiturient
{
    public partial class bakalavr1 : Form
    {
        public bakalavr1()
        {
            InitializeComponent();

            comboBox1.Items.Add("иә");
            comboBox1.Items.Add("жоқ");
        }
    }
}
```

```

comboBox2.Items.Add("бар");
comboBox2.Items.Add("жоқ");

comboBox3.Items.Add("күндізгі");
                                Б қосымшасы жалғасы

comboBox3.Items.Add("сырттай");
var sp1 = (from i in Program.db.kafedras select i.kaf_name);
foreach (var s in sp1)
{
    comboBox4.Items.Add(s);
    comboBox5.Items.Add(s);
    comboBox6.Items.Add(s);
    comboBox7.Items.Add(s);
}

comboBox4.Enabled = true;
comboBox5.Enabled = false;
comboBox6.Enabled = false;
comboBox7.Enabled = false;
}

private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
        string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox6.Text) ||
        string.IsNullOrEmpty(textBox7.Text) || string.IsNullOrEmpty(textBox8.Text) ||
        string.IsNullOrEmpty(textBox11.Text) || string.IsNullOrEmpty(textBox13.Text) ||
        string.IsNullOrEmpty(textBox14.Text) || string.IsNullOrEmpty(textBox15.Text) ||
        string.IsNullOrEmpty(textBox22.Text) || string.IsNullOrEmpty(comboBox1.Text) ||
        string.IsNullOrEmpty(comboBox2.Text) || string.IsNullOrEmpty(comboBox3.Text) ||
string.IsNullOrEmpty(comboBox4.Text) || string.IsNullOrEmpty(comboBox5.Text) ||
string.IsNullOrEmpty(comboBox6.Text) || string.IsNullOrEmpty(comboBox7.Text))
    {
        MessageBox.Show("Барлық жолдарды толтырыңыз", "Ескерту",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

    else
    {
        int zd = 0;
        string[] prov_inn = (from i in Program.db.bakalavrs select i.INN).ToArray();
        for (int i = 0; i < prov_inn.Length; i++)
        {
            if (textBox4.Text == prov_inn[i].ToString())
            {
                zd = 1;
            }
        }
    }
}

```

```

        if (zd == 1)
        {
            MessageBox.Show("Сіз жүйеге құжаттарыңызды өткізгенсіз, сізге жақын арада
хабарласамыз"

```

Б қосымшасы жалғасы

```

    }

```

```

        else if (zd == 0)
        {
            string document1 = textBox15.Text;
            string document2 = textBox14.Text;
            string document3 = textBox13.Text;

            string fileName1 = System.IO.Path.GetFileName(document1);
            string fileName2 = System.IO.Path.GetFileName(document2);
            string fileName3 = System.IO.Path.GetFileName(document3);

            string targetPath = Application.StartupPath + @"..\..\..\documents\bakalavr\" +
textBox1.Text + " " + textBox2.Text + @"\";
            string sourceFile1 = System.IO.Path.Combine(document1);
            string sourceFile2 = System.IO.Path.Combine(document2);
            string sourceFile3 = System.IO.Path.Combine(document3);

            string tardoc1 = targetPath + @"\" + fileName1;
            string tardoc2 = targetPath + @"\" + fileName2;
            string tardoc3 = targetPath + @"\" + fileName3;

            string destFile1 = System.IO.Path.Combine(targetPath, fileName1);
            string destFile2 = System.IO.Path.Combine(targetPath, fileName2);
            string destFile3 = System.IO.Path.Combine(targetPath, fileName3);

            if (!System.IO.Directory.Exists(targetPath))
            {
                System.IO.Directory.CreateDirectory(targetPath);
            }

            System.IO.File.Copy(sourceFile1, destFile1, true);
            System.IO.File.Copy(sourceFile2, destFile2, true);
            System.IO.File.Copy(sourceFile3, destFile3, true);

            var fam = textBox1.Text;
            var name = textBox2.Text;
            var patr = textBox3.Text;
            var inn = textBox4.Text;
            var date_rozh = dateTimePicker1.Value;
            var address = textBox6.Text;
            var tel = textBox7.Text;
            var shcool = textBox8.Text;
            var date_okon = dateTimePicker2.Value;
            var sert_number = textBox10.Text;

```

```
var sred_bal = textBox11.Text;
var ent_bal = textBox12.Text;
var chet_pr = textBox22.Text;
```

Б қосымшасы жалғасы

```
var olimp = 0;
if (comboBox1.Text == "иә")
{
    olimp = 1;
}
else if (comboBox1.Text == "жоқ")
{
    olimp = 0;
}

var kvot = 0;
if (comboBox2.Text == "бар")
{
    kvot = 1;
}
else if (comboBox2.Text == "жоқ")
{
    kvot = 0;
}

var spec1 = (from i in Program.db.kafedras where i.kaf_name == comboBox4.Text
select i.kaf_kod).SingleOrDefault();
var spec2 = (from i in Program.db.kafedras where i.kaf_name == comboBox5.Text
select i.kaf_kod).SingleOrDefault();
var spec3 = (from i in Program.db.kafedras where i.kaf_name == comboBox6.Text
select i.kaf_kod).SingleOrDefault();
var spec4 = (from i in Program.db.kafedras where i.kaf_name == comboBox7.Text
select i.kaf_kod).SingleOrDefault();
var dekanat_id = (from i in Program.db.kafedras where i.kaf_kod == spec1 select
i.dek_id).SingleOrDefault();
var doc1 = textBox15.Text;
var doc2 = textBox14.Text;
var doc3 = textBox13.Text;

var type_study = 0;
if (comboBox3.Text == "күндізгі")
{
    type_study = 1;
}
else if (comboBox3.Text == "сырттай")
{
    type_study = 0;
}
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
```

```
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into bakalavr(bak_fam, bak_name, bak_patronymic,
INN, date_rozh, address, kont_tel, school, date_okon, number_sertifikate, sred_otc, kol_bal,
```

Б қосымшасы жалғасы

```
chet_pred,sertivicate, kvota, spec1, spec2, spec3, spec4, doc1, doc2, doc3, type_study, dek_id,
status) values("'" + fam + "'", "'" + name + "'", "'" + patr + "'", "'" + inn + "'", "'" + date_rozh + "'", "'" +
address + "'", "'" + tel + "'", "'" + shcool + "'", "'" + date_okon + "'", "'" + sert_number + "'", "'" + sred_bal + "'",
"'" + ent_bal + "'", "'" + chet_pr + "'", "'" + olymp + "'", "'" + kvot + "'", "'" + spec1 + "'", "'" + spec2 + "'", "'" +
spec3 + "'", "'" + spec4 + "'", "'" + tardoc1 + "'", "'" + tardoc2 + "'", "'" + tardoc3 + "'", "'" + type_study + "'", "'" +
dekanat_id + "'", 0)";
```

```
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox10.Clear();
textBox11.Clear();
textBox12.Clear();
textBox22.Clear();
textBox13.Clear();
textBox14.Clear();
textBox15.Clear();
```

```
comboBox1.Items.Clear();
comboBox2.Items.Clear();
comboBox3.Items.Clear();
comboBox4.Items.Clear();
comboBox5.Items.Clear();
comboBox6.Items.Clear();
comboBox7.Items.Clear();
```

```
comboBox1.Items.Add("иә");
comboBox1.Items.Add("жоқ");
```

```
comboBox2.Items.Add("бар");
comboBox2.Items.Add("жоқ");
```

```
comboBox3.Items.Add("күндізгі");
comboBox3.Items.Add("сырттай");
var sp1 = (from i in Program.db.kafedras select i.kaf_name);
foreach (var s in sp1)
```

```

        {
            comboBox4.Items.Add(s);
            comboBox5.Items.Add(s);
            comboBox6.Items.Add(s);
                Б қосымшасы жалғасы

            comboBox7.Items.Add(s);
        }
        comboBox4.Enabled = true;
        comboBox5.Enabled = false;
        comboBox6.Enabled = false;
        comboBox7.Enabled = false;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
    this.openFileDialog1.RestoreDirectory = true;
    this.openFileDialog1.Multiselect = false;
    this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
    DialogResult dr = this.openFileDialog1.ShowDialog();
    if (dr == System.Windows.Forms.DialogResult.OK)
    {
        textBox15.Text = openFileDialog1.FileName;
    }
}

private void button4_Click(object sender, EventArgs e)
{
    this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
    this.openFileDialog1.RestoreDirectory = true;
    this.openFileDialog1.Multiselect = false;
    this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
    DialogResult dr = this.openFileDialog1.ShowDialog();
    if (dr == System.Windows.Forms.DialogResult.OK)
    {
        textBox14.Text = openFileDialog1.FileName;
    }
}

private void button5_Click(object sender, EventArgs e)

```

```

    {
        this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
        this.openFileDialog1.RestoreDirectory = true;

```

Б қосымшасы жалғасы

```

this.openFileDialog1.Multiselect = false;
    this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
    DialogResult dr = this.openFileDialog1.ShowDialog();
    if (dr == System.Windows.Forms.DialogResult.OK)
    {
        textBox13.Text = openFileDialog1.FileName;
    }
}

```

```

private void comboBox4_SelectedIndexChanged(object sender, EventArgs e)
{
    var sel1 = comboBox4.Text;
    var sel2 = comboBox5.Text;
    var sel3 = comboBox6.Text;
    var sel4 = comboBox7.Text;

    comboBox5.Items.Remove(sel1);
    comboBox5.Items.Remove(sel2);
    comboBox5.Items.Remove(sel3);
    comboBox5.Items.Remove(sel4);

    comboBox6.Items.Remove(sel1);
    comboBox6.Items.Remove(sel2);
    comboBox6.Items.Remove(sel3);
    comboBox6.Items.Remove(sel4);

    comboBox7.Items.Remove(sel1);
    comboBox7.Items.Remove(sel2);
    comboBox7.Items.Remove(sel3);
    comboBox7.Items.Remove(sel4);

    comboBox4.Enabled = false;
    comboBox5.Enabled = true;
    comboBox6.Enabled = false;
    comboBox7.Enabled = false;
}

```

```

private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
{
    var sel1 = comboBox4.Text;
    var sel2 = comboBox5.Text;
    var sel3 = comboBox6.Text;
    var sel4 = comboBox7.Text;
}

```

```
comboBox6.Items.Remove(sel1);
comboBox6.Items.Remove(sel2);
comboBox6.Items.Remove(sel3);
comboBox6.Items.Remove(sel4);
```

Б қосымшасы жалғасы

```
comboBox7.Items.Remove(sel1);
comboBox7.Items.Remove(sel2);
comboBox7.Items.Remove(sel3);
comboBox7.Items.Remove(sel4);
comboBox4.Enabled = false;
comboBox5.Enabled = false;
comboBox6.Enabled = true;
comboBox7.Enabled = false;
}
```

```
private void comboBox6_SelectedIndexChanged(object sender, EventArgs e)
{
    var sel1 = comboBox4.Text;
    var sel2 = comboBox5.Text;
    var sel3 = comboBox6.Text;
    var sel4 = comboBox7.Text;

    comboBox7.Items.Remove(sel1);
    comboBox7.Items.Remove(sel2);
    comboBox7.Items.Remove(sel3);
    comboBox7.Items.Remove(sel4);

    comboBox4.Enabled = false;
    comboBox5.Enabled = false;
    comboBox6.Enabled = false;
    comboBox7.Enabled = true;
}
```

```
private void comboBox7_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox4.Enabled = false;
    comboBox5.Enabled = false;
    comboBox6.Enabled = false;
    comboBox7.Enabled = false;
}
```

```
private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(Char.IsDigit(e.KeyChar)))
    {
        if (e.KeyChar != (char)Keys.Back)
        {
            e.Handled = true;
        }
    }
}
```



```

    }

    private void textBox7_KeyPress(object sender, KeyPressEventArgs e)
    {
        Б қосымшасы жалғасы

        if (!(Char.IsDigit(e.KeyChar)))
        {
            if (e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }

        private void textBox12_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!(Char.IsDigit(e.KeyChar)))
            {
                if (e.KeyChar != (char)Keys.Back)
                {
                    e.Handled = true;
                }
            }
        }

        private void textBox22_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!(Char.IsDigit(e.KeyChar)))
            {
                if (e.KeyChar != (char)Keys.Back)
                {
                    e.Handled = true;
                }
            }
        }

        private void button6_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            textBox6.Clear();
            textBox7.Clear();
            textBox8.Clear();
            textBox10.Clear();
            textBox11.Clear();
            textBox12.Clear();
            textBox22.Clear();
            textBox13.Clear();

```

```

textBox14.Clear();
textBox15.Clear();

comboBox1.Items.Clear();
        Б қосымшасы жалғасы

comboBox2.Items.Clear();
comboBox3.Items.Clear();
comboBox4.Items.Clear();
comboBox5.Items.Clear();
comboBox6.Items.Clear();
comboBox7.Items.Clear();
comboBox1.Items.Add("иә");
comboBox1.Items.Add("жоқ");

comboBox2.Items.Add("бар");
comboBox2.Items.Add("жоқ");

comboBox3.Items.Add("күндізгі");
comboBox3.Items.Add("сырттай");
var sp1 = (from i in Program.db.kafedras select i.kaf_name);
foreach (var s in sp1)
{
    comboBox4.Items.Add(s);
    comboBox5.Items.Add(s);
    comboBox6.Items.Add(s);
    comboBox7.Items.Add(s);
}

comboBox4.Enabled = true;
comboBox5.Enabled = false;
comboBox6.Enabled = false;
comboBox7.Enabled = false;
}
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace abiturient
{
    public partial class magistratura : Form
    {

```

```

public magistratura()
{
    InitializeComponent();

                                Б қосымшасы жалғасы

    comboBox1.Items.Add("иә");
    comboBox1.Items.Add("жоқ");

    comboBox2.Items.Add("бар");
comboBox2.Items.Add("жоқ");
    comboBox3.Items.Add("бар");
    comboBox3.Items.Add("жоқ");

    var spq = (from i in Program.db.kafedras select i.kaf_name);
    foreach (var s in spq)
    {
        comboBox4.Items.Add(s);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox5.Text) ||
string.IsNullOrEmpty(textBox6.Text) || string.IsNullOrEmpty(textBox7.Text) ||
string.IsNullOrEmpty(textBox8.Text) || string.IsNullOrEmpty(textBox10.Text) ||
string.IsNullOrEmpty(textBox11.Text) || string.IsNullOrEmpty(textBox15.Text) ||
string.IsNullOrEmpty(textBox18.Text) || string.IsNullOrEmpty(textBox19.Text) ||
string.IsNullOrEmpty(comboBox4.Text))
    {
        MessageBox.Show("Барлық жолдарды толтырыңыз", "Ескерту",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

    else
    {

        int zd = 0;
        string[] prov_inn = (from i in Program.db.magistrs select i.inn).ToArray();
        for (int i = 0; i < prov_inn.Length; i++)
        {
            if (textBox4.Text == prov_inn[i].ToString())
            {
                zd = 1;
            }
        }

        if (zd == 1)
        {

```

```
        MessageBox.Show("Сіз жүйеге құжаттарыңызды өткізгенсіз, сізге жақын арада  
хабарласамыз");  
    }
```

```
else if (zd == 0)
```

Б қосымшасы жалғасы

```
{  
    string document1 = textBox19.Text;  
    string document2 = textBox18.Text;  
    string document3 = textBox15.Text;  
string fileName1 = System.IO.Path.GetFileName(document1);  
    string fileName2 = System.IO.Path.GetFileName(document2);  
    string fileName3 = System.IO.Path.GetFileName(document3);  
  
    string targetPath = Application.StartupPath + @"..\..\..\documents\magistratura\" +  
textBox1.Text + " " + textBox2.Text + @"\";  
    string sourceFile1 = System.IO.Path.Combine(document1);  
    string sourceFile2 = System.IO.Path.Combine(document2);  
    string sourceFile3 = System.IO.Path.Combine(document3);  
  
    string tardoc1 = targetPath + @"\" + fileName1;  
    string tardoc2 = targetPath + @"\" + fileName2;  
    string tardoc3 = targetPath + @"\" + fileName3;  
  
    string destFile1 = System.IO.Path.Combine(targetPath, fileName1);  
    string destFile2 = System.IO.Path.Combine(targetPath, fileName2);  
    string destFile3 = System.IO.Path.Combine(targetPath, fileName3);  
  
    if (!System.IO.Directory.Exists(targetPath))  
    {  
        System.IO.Directory.CreateDirectory(targetPath);  
    }  
  
    System.IO.File.Copy(sourceFile1, destFile1, true);  
    System.IO.File.Copy(sourceFile2, destFile2, true);  
    System.IO.File.Copy(sourceFile3, destFile3, true);  
  
    var fam = textBox1.Text;  
    var name = textBox2.Text;  
    var patr = textBox3.Text;  
    var inn = textBox4.Text;  
    var date_rozh = dateTimePicker1.Value;  
    var address = textBox6.Text;  
    var tel = textBox7.Text;  
    var vus = textBox8.Text;  
    var date_okon = dateTimePicker2.Value;  
    var diplom_num = textBox10.Text;  
    var gpa = textBox5.Text;  
    var engl = textBox11.Text;  
    var olimp = 0;
```

```

if (comboBox1.Text == "бар")
{
    olimp = 1;
}
else if (comboBox1.Text == "жоқ")
    Б қосымшасы жалғасы

{
    olimp = 0;
}

var kvot = 0;
if (comboBox2.Text == "бар")
{
    kvot = 1;
}
else if (comboBox2.Text == "жоқ")
{
    kvot = 0;
}
var spec = (from i in Program.db.kafedras where i.kaf_name == comboBox4.Text
select i.kaf_kod).SingleOrDefault();
var dekanat_id = (from i in Program.db.kafedras where i.kaf_kod == spec select
i.dek_id).SingleOrDefault();
var work = 0;
if (comboBox3.Text == "бар")
{
    work = 1;
}
else if (comboBox3.Text == "жоқ")
{
    work = 0;
}

var doc1 = textBox19.Text;
var doc2 = textBox18.Text;
var doc3 = textBox15.Text;

SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into magistr(mag_fam, mag_name, mag_patronymic,
inn, mag_date, mag_address, mag_tel, mag_vuz, mag_god_okon, diplom_num, gpa, english,
stash_raboty, sert_olimp, kvota, spec, doc1, doc2, doc3, dek_id, status) values('" + fam + "','" +
name + "','" + patr + "','" + inn + "','" + date_rozh + "','" + address + "','" + tel + "','" + vus + "','" +
date_okon + "','" + diplom_num + "','" + gpa + "','" + engl + "','" + work + "','" + olimp + "','" + kvot
+ "','" + spec + "','" + tardoc1 + "','" + tardoc2 + "','" + tardoc3 + "','" + dekanat_id + "', 0)";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();

```

```
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз",  
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
textBox1.Clear();  
textBox2.Clear();
```

Б қосымшасы жалғасы

```
textBox3.Clear();  
textBox4.Clear();  
textBox15.Clear();  
textBox6.Clear();  
textBox7.Clear();  
textBox8.Clear();  
textBox10.Clear();  
textBox11.Clear();  
textBox5.Clear();  
textBox19.Clear();  
textBox18.Clear();  
comboBox1.Items.Clear();  
comboBox2.Items.Clear();  
comboBox3.Items.Clear();  
comboBox4.Items.Clear();
```

```
comboBox1.Items.Add("иә");  
comboBox1.Items.Add("жоқ");
```

```
comboBox2.Items.Add("бар");  
comboBox2.Items.Add("жоқ");
```

```
comboBox3.Items.Add("бар");  
comboBox3.Items.Add("жоқ");
```

```
var spq = (from i in Program.db.kafedras select i.kaf_name);  
foreach (var s in spq)  
{  
    comboBox4.Items.Add(s);  
}
```

```
}  
}
```

```
private void button2_Click(object sender, EventArgs e)  
{  
    this.Close();  
}
```

```
private void button3_Click(object sender, EventArgs e)  
{
```

```

        this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files
(*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
        this.openFileDialog1.RestoreDirectory = true;
        this.openFileDialog1.Multiselect = false;
        this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
        DialogResult dr = this.openFileDialog1.ShowDialog();
                Б қосымшасы жалғасы

        if (dr == System.Windows.Forms.DialogResult.OK)
        {
            textBox19.Text = openFileDialog1.FileName;
        }
    }
    private void button4_Click(object sender, EventArgs e)
    {
        this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files
(*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
        this.openFileDialog1.RestoreDirectory = true;
        this.openFileDialog1.Multiselect = false;
        this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
        DialogResult dr = this.openFileDialog1.ShowDialog();
        if (dr == System.Windows.Forms.DialogResult.OK)
        {
            textBox18.Text = openFileDialog1.FileName;
        }
    }

    private void button5_Click(object sender, EventArgs e)
    {
        this.openFileDialog1.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files
(*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
        this.openFileDialog1.RestoreDirectory = true;
        this.openFileDialog1.Multiselect = false;
        this.openFileDialog1.Title = "Құжаттарыңызды таңдаңыз";
        DialogResult dr = this.openFileDialog1.ShowDialog();
        if (dr == System.Windows.Forms.DialogResult.OK)
        {
            textBox15.Text = openFileDialog1.FileName;
        }
    }

    private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!(Char.IsDigit(e.KeyChar)))
        {
            if (e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }
    }
}

```

```

private void textBox7_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(Char.IsDigit(e.KeyChar)))
    {
        if (e.KeyChar != (char)Keys.Back)
            Б қосымшасы жалғасы

            {
                e.Handled = true;
            }
    }
}
private void textBox11_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(Char.IsDigit(e.KeyChar)))
    {
        if (e.KeyChar != (char)Keys.Back)
        {
            e.Handled = true;
        }
    }
}

```

```

private void button6_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox15.Clear();
    textBox6.Clear();
    textBox7.Clear();
    textBox8.Clear();
    textBox10.Clear();
    textBox11.Clear();
    textBox5.Clear();
    textBox19.Clear();
    textBox18.Clear();
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    comboBox3.Items.Clear();
    comboBox4.Items.Clear();
    comboBox1.Items.Add("иә");
    comboBox1.Items.Add("жоқ");

    comboBox2.Items.Add("бар");
    comboBox2.Items.Add("жоқ");

    comboBox3.Items.Add("бар");
    comboBox3.Items.Add("жоқ");
}

```



```

var spq = (from i in Program.db.kafedras select i.kaf_name);
foreach (var s in spq)
{
    comboBox4.Items.Add(s);
}

```

Б қосымшасы жалғасы

```

}
}
}
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Runtime.Remoting.Contexts;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;

namespace abiturient
{
    static class Program
    {
        public static SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["priemContext"].ConnectionString);
        public static aipetDataContext db = new aipetDataContext(conn);
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new abiturient());
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;
using Excel = Microsoft.Office.Interop.Excel;

```

```

namespace administrator
{
    public partial class adminControl : Form
    {
        private static int rec;

        public adminControl()
        {
            InitializeComponent();
            groupBox1.Enabled = false;
            a1 = "select k.kom_id, k.kom_fam as 'Тегі', k.kom_name as 'Аты', k.kom_patronimic as 'Әкесінің аты', k.kom_type as 'Қабылдау бөлімі', k.login as 'Логин', k.password as 'Құпия сөз', d.dek_name as 'Деканат' from komissiya k inner join dekanat d on d.dek_id = k.dek_id";
            connect();
            dataGridView1.Columns["kom_id"].Visible = false;
            dataGridView1.Rows[0].Cells[1].Selected = true;

            var dekanat = (from i in Program.db.dekanats select i.dek_name);
            foreach (var dek in dekanat)
            {
                comboBox1.Items.Add(dek);
            }
            comboBox5.Items.Add("Барлық");
            comboBox5.Items.Add("Өткізілген құжаттар");
            comboBox5.Items.Add("Қарастырылатын құжаттар");

            comboBox2.Items.Add("бакалавр");
            comboBox2.Items.Add("магистратура");
            comboBox5.Enabled = false;
            groupBox3.Enabled = false;

            var faculty = (from i in Program.db.dekanats select i.dek_name);
            foreach (var f in faculty)
            {
                comboBox3.Items.Add(f);
            }
        }

        private void dataGridView1_RowPrePaint(object sender,
        DataGridViewRowPrePaintEventArgs e)
        {
            int index = e.RowIndex;
            string indexStr = (index + 1).ToString();
            object header = this.dataGridView1.Rows[index].HeaderCell.Value;
            if (header == null || !header.Equals(indexStr))
                this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;
        }
    }
}

```

Б қосымшасы жалғасы

```

private void dataGridView2_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView2.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        Б қосымшасы жалғасы

        this.dataGridView2.Rows[index].HeaderCell.Value = indexStr;
}

string a, a1, b, b1, cn;
public void connect()
{
    SqlConnection cn = Program.conn;
    a = a1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(a, cn);
    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}

public void connect2()
{
    SqlConnection cn = Program.conn;
    b = b1;
    DataSet ds1 = new DataSet();
    SqlDataAdapter da1 = new SqlDataAdapter(b, cn);
    da1.Fill(ds1);
    dataGridView2.DataSource = ds1.Tables[0];
}

private void button5_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    groupBox1.Enabled = false;
    button1.Enabled = true;
    button2.Enabled = true;
    dataGridView1.Enabled = true;
    var dekanat = (from i in Program.db.dekanats select i.dek_name);
    foreach (var dek in dekanat)
    {
        comboBox1.Items.Add(dek);
    }
}

```

```

        comboBox2.Items.Add("бакалавр");
        comboBox2.Items.Add("магистратура");
    }

    private void button1_Click(object sender, EventArgs e)
    {
        button2.Enabled = false;

        Б қосымшасы жалғасы

        groupBox1.Enabled = true;
        rec = 1;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        dataGridView1.Enabled = false;
        groupBox1.Enabled = true;
        button1.Enabled = false;
        rec = 0;
        textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
        textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
        textBox3.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        textBox4.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
        textBox5.Text = dataGridView1.CurrentRow.Cells[6].Value.ToString();
        comboBox1.Text = dataGridView1.CurrentRow.Cells[7].Value.ToString();
        comboBox2.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "";
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Пайдаланушыға жүйеге кіру жабылды", "Назар аударыңыз",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    private void button4_Click(object sender, EventArgs e)
    {
        try
        {
            var dek_id = (from i in Program.db.dekanats where i.dek_name == comboBox1.Text
            select i.dek_id).SingleOrDefault();
            if (rec == 0)
            {
                SqlConnection sqlConnection1 = Program.conn;

```

```

sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update komissiya set kom_fam =" + textBox1.Text + ",
kom_name = " + textBox2.Text + ", kom_patronimic = " + textBox3.Text + ", dek_id = " +
dek_id + ", login = " + textBox4.Text + ", password = " + textBox5.Text + ", kom_type = " +
comboBox2.Text + " where kom_id = " + dataGridView1.CurrentRow.Cells[0].Value;

```

Б қосымшасы жалғасы

```

cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті өзгертілді!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
comboBox1.Items.Clear();
comboBox2.Items.Clear();
groupBox1.Enabled = false;
button1.Enabled = true;
button2.Enabled = true;
dataGridView1.Enabled = true;
var dekanat = (from i in Program.db.dekanats select i.dek_name);
foreach (var dek in dekanat)
{
    comboBox1.Items.Add(dek);
}
comboBox2.Items.Add("бакалавр");
comboBox2.Items.Add("магистратура");
}

else if (rec == 1)
{
    SqlConnection sqlConnection1 = Program.conn;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "insert into komissiya(kom_fam, kom_name, kom_patronimic,
dek_id, login, password, kom_type) values(" + textBox1.Text + "," + textBox2.Text + "," +
textBox3.Text + "," + dek_id + "," + textBox4.Text + "," + textBox5.Text + "," +
comboBox2.Text + ")";
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    sqlConnection1.Close();
    MessageBox.Show("Ақпарат сәтті енгізілді!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

```

textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
comboBox1.Items.Clear();
comboBox2.Items.Clear();

```

Б қосымшасы жалғасы

```

groupBox1.Enabled = false;
button1.Enabled = true;
button2.Enabled = true;
dataGridView1.Enabled = true;
var dekanat = (from i in Program.db.dekanats select i.dek_name);
foreach (var dek in dekanat)
{
    comboBox1.Items.Add(dek);
}
comboBox2.Items.Add("бакалавр");
comboBox2.Items.Add("магистратура");
}

```

```

a1 = "select k.kom_id, k.kom_fam as 'Тегі', k.kom_name as 'Аты', k.kom_patronymic as
'Әкесінің аты', k.kom_type as 'Қабылдау бөлімі', k.login as 'Логин', k.password as 'Күпия сөз',
d.dek_name as 'Деканат' from komissiya k inner join dekanat d on d.dek_id = k.dek_id";
connect();
dataGridView1.Columns["kom_id"].Visible = false;
dataGridView1.Rows[0].Cells[1].Selected = true;

```

```

}
catch (Exception)
{
    MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

```

}

```

```

private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked == true && comboBox5.Text == "Барлық")
    {
        b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты', bak_patronymic as 'Әкесінің аты',
b.INN as 'ЖСН', b.address as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған ұпай',
b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as '2 мамандық', b.spec3 as '3 мамандық',
b.spec4 as '4 мамандық', b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b";
        connect2();
        dataGridView2.Columns["status"].Visible = false;
    }
}

```

```

else if (radioButton2.Checked == true && comboBox5.Text == "Өткізілген құжаттар")
{
    b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты', bak_patronimic as 'Әкесінің аты',
b.INN as 'ЖСН', b.address as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған ұпай',
b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as '2 мамандық', b.spec3 as '3 мамандық',
b.spec4 as '4 мамандық', b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b where
status = 2";

```

Б қосымшасы жалғасы

```

    connect2();
    dataGridView2.Columns["status"].Visible = false;
}
else if (radioButton2.Checked == true && comboBox5.Text == "Қарастырылатын
құжаттар")
{
    b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты', bak_patronimic as 'Әкесінің аты', b.INN as
'ЖСН', b.address as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған ұпай',
b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as '2 мамандық', b.spec3 as '3 мамандық',
b.spec4 as '4 мамандық', b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b where
status = 1";
    connect2();
    dataGridView2.Columns["status"].Visible = false;
}

else if (radioButton1.Checked == true && comboBox5.Text == "Барлық")
{
    b1 = "select m.mag_fam as 'Тегі', m.mag_name as 'Аты', m.mag_patronimic as 'Әкесінің
аты', m.inn as 'ЖСН', m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gra as 'GPA',
m.english as 'Шет тілі', m.spec as 'Мамандық', m.date_sdachi as 'Тапсырған уақыты', m.status
from magistr m";
    connect2();
    dataGridView2.Columns["status"].Visible = false;
}
else if (radioButton1.Checked == true && comboBox5.Text == "Өткізілген құжаттар")
{
    b1 = "select m.mag_fam as 'Тегі', m.mag_name as 'Аты', m.mag_patronimic as 'Әкесінің
аты', m.inn as 'ЖСН', m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gra as 'GPA',
m.english as 'Шет тілі', m.spec as 'Мамандық', m.date_sdachi as 'Тапсырған уақыты', m.status
from magistr m where status = 2";
    connect2();
    dataGridView2.Columns["status"].Visible = false;
}
else if (radioButton1.Checked == true && comboBox5.Text == "Қарастырылатын
құжаттар")
{
    b1 = "select m.mag_fam as 'Тегі', m.mag_name as 'Аты', m.mag_patronimic as 'Әкесінің
аты', m.inn as 'ЖСН', m.mag_address as 'Мекен-жай', m.mag_tel as 'Телефон', m.gra as 'GPA',
m.english as 'Шет тілі', m.spec as 'Мамандық', m.date_sdachi as 'Тапсырған уақыты', m.status
from magistr m where status = 1";
    connect2();
    dataGridView2.Columns["status"].Visible = false;
}

```

```

    }
    groupBox3.Enabled = true;
}

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    comboBox5.Enabled = true;
        Б қосымшасы жалғасы

    groupBox3.Enabled = false;
    b1 = "select b.bak_fam as 'Тегі', b.bak_name as 'Аты', bak_patronimic as 'Әкесінің аты',
b.INN as 'ЖСН', b.address as 'Мекен-жайы', b.kont_tel as 'Телефон', b.kol_bal as 'Жинаған ұпай',
b.chet_pred as '4 пән', b.spec1 as '1 мамандық', b.spec2 as '2 мамандық', b.spec3 as '3 мамандық',
b.spec4 as '4 мамандық', b.data_sdachi as 'Тапсырған уақыты', b.status from bakalavr b where
status = 3";
    connect2();
    comboBox5.Items.Clear();
    comboBox5.Items.Add("Барлық");
    comboBox5.Items.Add("Өткізілген құжаттар");
    comboBox5.Items.Add("Қарастырылатын құжаттар");
}

```


1	Техникалық тапсырма.....	13
1.1	Тағайындалуы және қолдану аймағы.....	13
1.1.1	Сенімділікке қойылатын талаптар	13
1.1.2	Техникалық құрылғыларға қойылатын талаптар	13
1.1.3	Бағдарламалық құжатқа қойылатын талаптар	13
1.2	Құру кезеңдері мен сатылары	14
1.3	Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы	14
1.3.1	АЖО құрылымы	15
1.3.2	АЖО–ға қойылатын жалпы талаптар	16
1.3.3	АЖО–ды жобалау сатылары.....	17
1.3.4	Қабылдау комиссия жұмысының жалпы сипаттамасы.....	18
2	Ақпараттық жүйелер. Деректер қорын жобалау.....	19
2.1	Ақпараттық жүйелер туралы қысқаша мағлұматтар.....	19
2.1.1	Ақпараттық жүйе жобалау сатылары.....	20
2.1.2	Ақпараттық жүйенің тіршілік циклі	21
2.1.3	Ақпараттық жүйелер ұғымы және олардың маңызы	23
2.2	Деректер қорын жобалау	26
2.2.1	Деректер моделі.....	26
2.2.2	Реляционды деректер қорына қойылатын талаптар.....	28
2.2.3	SQL сұраныс тілі.МҚБЖ үшін интерфейсті стандарттау	29
2.2.4	Қосымшаларды өңдеу кезінде SQL тілінің қолданылуы	31
3	Дерекқорды жобалау және оны іске асыру	33
3.1	ДҚ құрудың бастапқы кезеңі	33
3.2	Концептуалды жобалау	33
3.2.1	Қажетті ақпарат. Пайдаланушылардың талабы.....	33
3.2.2	Дерекқор және қосымшаны жобалау құралдарын таңдаудың негіздемесі. 34	
3.3	Бағдарламаның ER моделін жасау	35
3.3.1	Бизнес ережелер	36
3.3.2	Реляционды қатынастарды құрастыру.....	39
3.3.3	Логикалық жобалау.....	41
3.4	Физикалық жобалау	42
3.4.1	Дерек қорына қосымша құру	42

3.4.2 Бағдарламаның негізгі алгоритмі	42
3.4.3 Администратор бөлімінің коды	43
3.4.4 Бағдарламаны сипаттау	54
4 Тіршілік қауіпсіздігі.....	63
4.1 Ауа алмасу (кондиционер) жүйесінің құрылғысы және есебі	63
4.2 Кондиционерлеу және ауаны жаңарту жүйелерін қарастыру	67
4.3 Температура айырымы нәтижесінде алынатын жылу және жылу жоғалту .	68
5 Техника – экономикалық негізделуі	75
5.1 Бағдарламалық қамтаманың артықшылығы мен сипаттамасы.....	76
5.2 Маркетингтік стратегия.....	76
5.3 Бағдарламаның сатылым жоспары.....	77
5.4 Жұмыс сипаттамасы және қажеттілік дәлелдемесі	78
5.5 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі.....	78
5.6 Бағдарламалық қамтаманы өңдеуге кеткен шығын есептелуі	80
5.7 Қосымша шығындар статьясы	81
5.8 Өзіндік құн нәтижесінің	83
5.9 Бағдарламаның бағасын есептеу	84
5.10 Бағдарламаны енгізудегі шартты үнемдеуді есептеу.....	84
5.11 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	86
5.12 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу 3 оқу орны үшін 87	
5.13 Бағдарламаны енгізуден түскен шартты үнемдеуді есептеу АЭЖБУ	88
Қорытынды	89
Пайдаланылған әдебиеттер тізімі	90
А қосымшасы.....	91
Б қосымшасы	92

