

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.Сәтбаев атындағы қазақ ұлттық техникалық зерттеу университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Жанибекова Гульдана Саматовна

Студенттердің сабаққа қатысуы мен үлгерімін есепке алатын
ақпараттық жүйе құру

ДИПЛОМДЫҚ ЖҰМЫС


5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ



Қ.Сәтбаев атындағы қазақ ұлттық техникалық
зерттеу университеті
Ақпараттық және телекоммуникациялық
технологиялар институты
Киберқауіпсіздік, ақпаратты өңдеу және сақтау
кафедрасы

«Қорғауға жіберілді»
КАӨЖС кафедра меңгерушісі,

Н.А.Сейлова

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Студенттердің сабаққа қатысуы мен үлгерімін есепке алатын
ақпараттық жүйе құру»

5В070300 – «Ақпараттық жүйелер» мамандығы

Орындаған:



Жанибекова Г.С.

Ғылыми жетекші



сеньор-лектор, т.ғ.к., доцент
Ш.М.Байматаева

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ


Қ.Сәтбаев атындағы қазақ ұлттық техникалық зерттеу университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

БЕКІТЕМІН

КАӨС кафедра меңгерушісі,
тех.ғыл.канд, ассоц. доцент

 Н.А.Сейлова
« 31 » 05 2021 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Жанибекова Гүльдана Саматовна

Тақырыбы: «Студенттердің сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйе құру»

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

Аяқталған жұмысты тапсыру мерзімі 2021 жылғы « 20» мамыр

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

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) ақпараттық қамтаманы құру

б) программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар

Сызба материалдар: 15 слайдпен көрсетілген


Ұсынылатын негізгі әдебиет: 10 атау

Дипломдық жұмысты дайындау

КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	28.02.2021	
Ақпараттық қамтаманы құру	7.04.2021	
Программалық қамтаманы құру	3.05.2021	

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

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

Ғылыми жетекші 

Ш.М.Байматаева

Тапсырманы орындауға алған білім алушы 

Г.С.Жанибекова

Күні

« 24» 11 2021 ж.

АҢДАТПА

Дипломдық жұмыстың мақсаты білім беру орындарындағы студенттердің сабаққа қатысу мен үлгерімін есепке алатын ақпараттық жүйе құру. Дипломдық жұмысты орындау барысында пәндік аймаққа зерттеу жүргізілді; жүйенің функционалдық және инфологиялық модельдері құрылды; ER-диаграммасы жасалды; мәліметтер базасы жобаланды; жобаланған мәліметтер базасы негізінде бағдарламалық қамтама жасалды.

Дипломдық жұмысты орындау барысында Microsoft SQL Server мәліметтер базасы, Microsoft Visual Studio ортасы және C# программалау тілі пайдаланылды.

Жұмысты жасау кезінде пәндік аймақтағы кезінде оқулықтар, ғылыми мақалалар, ғылыми еңбектер қолданылды.

Дипломдық жұмыстың жаңалығы: C# негізінде жасалған оқушылардың сабаққа қатысуын және үлгерімін есепке алатын ақпараттық жүйе жасалды.

АННОТАЦИЯ

Целью дипломной работы является создание информационной системы, учитывающей посещаемость и успеваемость обучающихся.

В ходе выполнения дипломной работы было проведено исследование предметной области; разработаны функциональные и инфологические модели, ER-диаграмма; спроектирована база данных; разработано программное обеспечение на основе спроектированной базы данных.

В ходе работы использовалась база данных Microsoft SQL Server, среда Microsoft Visual Studio и язык программирования C #.

При создании работы были использованы учебники, научные статьи, научные труды в предметной области.

Новизна дипломной работы: построена информационная система на базе C #, учитывающая посещаемость и успеваемость студентов.

ANNOTATION

The aim of the diploma is to create an information system that takes into account the attendance and performance of students. In the course of the thesis, a study of the subject area was conducted; functional and infological models, ER-diagram were developed; a database was designed; software was developed based on the designed database.

In the course of work, a Microsoft SQL Server database, Microsoft Visual Studio environment and the C # programming language were used.

When developing the diploma work, numerous textbooks, scientific articles, scientific works were used.

The novelty of the diploma: an information system based on C # has been built, taking into account the attendance and progress of students.

МАЗМҰНЫ

КІРІСПЕ	9
1 АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ НЕГІЗДЕМЕСІ.....	10
1.1 Пәндік аймақтың сипаттамасы.....	10
1.2 Ақпараттық жүйені құрудың негіздемесі	11
1.3 Есептің қойылымы	12
2 АҚПАРАТТЫҚ ЖҮЙЕНІ ЖОБАЛАУ	13
2.1 Жүйенің функционалды моделін құру	13
2.2 Жүйенің инфологиялық моделін құру.	14
2.2.1 ER-диаграмма	15
2.3 Мәліметтер базасын жобалау	16
2.4 Мәліметтер базасының кестелерін сипаттау	17
3 АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ	20
3.1 АЖ жасауда қолданылатын құралдар	20
3.2 Ақпараттық жүйенің құрылымын жасау.....	21
3.3 Ақпараттық жүйенің жұмысын сипаттау.....	23
ҚОРЫТЫНДЫ.....	34
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	35
ҚОСЫМША.....	36

КІРІСПЕ

Бүгінгі күні ақпараттық технологиялардың дамуы қоғамның барлық салаларын жаңартуға алып келді.

Барлық типтегі салаларда өңделетін ақпарат ағындарының едәуір көлемі басқаруға қажетті құжаттарды жасауға бағытталған. Мұндай басқару құжаттары өз кезегінде басқару шешімдерін реттеуге бағытталған, яғни олар кез-келген мекеменің негізгі қызметі болып табылады.

Автоматтандырылған ақпараттық жүйелерді әр түрлі мекемелердің қызметіне енгізу менеджменттің құжаттамамен қамтамасыз ету сапасын жоғарылатуға және басқару жүйесінің өнімділігін және сапасын арттыруға мүмкіндік береді.

Қазіргі таңда білім беру процесін, оның ішінде орта және жоғарғы білім беру орталарында білім жүйесін автоматтандыру мәселесі өзекті болып саналады.

Дипломдық жұмыстың объектісі - автоматтандырылмаған орта және жоғарғы оқу орны.

Зерттеудің мақсаты - студенттердің сабаққа қатысуы мен үлгерімін есепке алудың автоматтандырылған ақпараттық жүйесін құру.

Дипломдық жұмыс барысында жасалған ақпараттық жүйе келесі көрсетілген процесстерді орындауы керек:

- оқу орнының әр бөлімдері үшін студенттердің тізіміне талдау жасау;
- электронды журналмен қамтамасыз ету, оған қолжетімділікті саралау ;
- барлық құжаттардың сақтық көшірмесін жасау;
- мұғалімдер жұмысын оңтайландыру.

Зерттеудің ғылыми міндеті- студенттердің үлгерімі мен сабаққа қатысуын автоматтандырылған есепке алу, бірыңғай ақпараттық базаны қалыптастыру және ақпараттық ағымдарды оңтайландыру үшін ақпараттық қолдаудың кешенді моделін жасау.

Зерттеу әдістері. Теориялық зерттеулер аналитикалық және имитациялық модельдеу, техникалық жүйелердегі басқару теориясын қолдану арқылы жүзеге асырылды.

Зерттеудің практикалық маңыздылығы- білім беру ұйымын басқару жүйесін жетілдіру, оңтайландыру бойынша ұсыныстарды әзірлеу.

Бұл жобаның теориялық негізі осы саладағы әдебиет болып табылады.

1 АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ НЕГІЗДЕМЕСІ

1.1 Пәндік аймақтың сипаттамасы

Күн сайын әлемдік желі кеңістігінде миллиондаған гигабайт ақпарат алмасатын жағдайда ақпараттық технологияларды қолданатын мамандарды даярлау қажеттілігі күннен-күнге артып келеді. Осыған байланысты білім берудегі ақпараттық технологиялардың да рөлі артуда. Заманауи ақпараттық технологияларды қолдану оқытудың тиімді тәсілдерін жасап, оқыту әдістерін жетілдірудің алғышарты болып табылады.

Қазіргі таңда ақпараттық технологиялар оқытуға қосымша емес, оның тиімділігін арттыратын білім беру процесінің ажырамас бөлігіне айналды. Олар оқытуды дараландырады, студенттерді өз бетінше жұмыс істеуге баулиды, білім алушылардың белсенділігін арттырады. Білім беру саласында ақпараттық технологияларды пайдалану мұғалімдер жұмысын оңтайландырады, студенттерді өзін-өзі тәрбиелеуге, үздіксіз білім алуға, кадрлардың біліктілігін арттыруға және қайта даярлауға ықпалын тигізеді. Білім беру саласында ақпараттық технологияларды пайдаланудың маңызы зор. Біздің елімізде әлі күнге дейін кейбір оқу орындарында білім беру жүйесі автоматтандырылмаған. Сондықтан пәндік аймақ ретінде орта және жоғарғы білім беру орындарын алдым.

Зерттеудің мақсаты- студенттердің үлгерімі мен сабаққа қатысуын есепке алатын ақпараттық жүйе құру.

Білім беру ұйымдарының жұмысын автоматтандыру- бұл білім беру орнының қалыпты жұмыс жасауының алғышарты. Дамыған елдердің білім министрлігі бұны әлдеқайда түсінген және біздің елде де автоматтандыру жұмысын қолға алынып жатыр.

Мұғалімнің ғана емес, сонымен қатар, білім беру мекемесінің барлық қызметкерлерінің қызметін толықтай автоматтандыратын бірыңғай жүйені құру- қазіргі автоматтандырудың өзекті мәселелерінің бірі.

Бағдарламалық шешім білім беру процестерін басқаруды, күнделіктер мен журналдарды жүргізуді, тесттер құруды және тексеруді, ата-аналарға оқу үлгерімі мен сабаққа қатысуы туралы хабарламалар шығаруды түрлендіру арқылы мұғалімнің қызметін оңтайландыруға мүмкіндік береді. Сонымен қатар, жүйе студенттердің үлгерімі мен сабаққа қатысуын талдауды автоматтандыру арқылы басшылықтың жұмысын жеңілдетуге мүмкіндік береді.

Бағдарламалық жасақтама қолданысқа ыңғайлы және икемді, яғни кез-келген оқу орындарында: жалпы, орта, жоғарғы оқу орындарында қолданыла алады.

Автоматтандырылған ақпараттық жүйе келесі нәтижелерді ұсынады:

- білім беру ұйымына қабылдау процесін автоматтандыру;
- білім беру бағдарламаларын есепке алып, олардың орындалуын бақылау;

- ата-аналарға оқу процесін ұйымдастыру, емтихандарды өткізу тәртібі мен оқушы үлгерімі туралы ақпарат беру;
 - студенттердің жұмысын жеке талдау;
 - мұғалімнің іс-әрекетін электронды форматқа аудару: электронды журнал, күнделіктер құру, жүргізу, сабақ жоспарлау, тестілеу.
- Жоба барысында орындалуы керек тапсырмалар:
- оқу орны, мұғалімдер, түлектер, студенттер және т.б. туралы ресми ақпарат жіберу;
 - студенттермен, мекеме қызметкерлерімен және ата-аналармен тиімді қарым-қатынас ұйымдастыру;
 - әр группаның өзінің виртуалды кабинетімен қамтамасыз ету, студенттер сабақ кестесіне, үй тапсырмасына, оқулықтарға үнемі қол жеткізе алуын қамтамасыз ету;
 - ата-аналарға студенттің үлгерімі мен үй тапсырмаларын бақылау, эдвайзермен және басқа мұғалімдермен байланысты ұйымдастыру;
 - студенттердің білімі мен рейтингін бақылау процестерін автоматтандыру, студенттердің сабаққа қатысуы мен үлгерімін есепке алу;
 - қашықтықтан оқыту мүмкіндігін қамтамасыз ету.

1.2 Ақпараттық жүйені құрудың негіздемесі

Оқытушылар білім беру порталында білім беру мүмкіндіктерін дамыта алатын, студенттерге арналған тапсырмаларды жүйеге енгізе алатын жеке парақшаға ие бола алады.

Білім беру орындарының басшылары заманауи оқыту технологияларын қолдану және үлгерімді бақылау арқылы білім сапасын бақылауды қамтамасыз ете алады. Жүйе оқыту үдерістерін автоматтандыру және қашықтықтан білім беру технологияларын енгізіп, шығындарды азайту, мекеменің қызметкерлері, студенттері мен оқытушылары туралы деректерді талдау үшін қажет.

Жасалған жүйені енгізу келесі топтарға арналған:

Білім беру ұйымының басшыларына:

- білім беру мекемесінің қызметін жоспарлау және басқару үдерісіне кететін уақытты үнемдеу;
- ақпаратқа жылдам қол жеткізу;
- оқу үдерісін және білім беру мекемесін басқару тиімділігін арттыру;
- студенттерді және олардың ата-аналарына қолжетімді қосымша ақпараттық ресурстармен қамтамасыз ету.

Оқытушылар, білім беру бөлімінің қызметкерлеріне:

- білім беру процесін реттейтін құжаттарға: оқу жоспарларына, сабақ кестелеріне, бақылау іс-шараларының тізімдеріне онлайн қолжетімділікті алу;
- студенттер, олардың сабаққа қатысуы, үлгерімі туралы ақпаратқа жылдам қол жеткізуге;

- студенттер саны бойынша статистикалық мәліметтерді автоматтандырылған жинауды ұйымдастырады;
- білім беру мекемесі ішінде электронды құжат айналымы жүйесін ұйымдастырады;
- әр түрлі құжатты бекітуге кететін уақытты қысқарту.

Студенттер:

- web- интерфейс арқылы оқу процесіне қатысты ақпаратқа жылдам, қарапайым және жедел қол жеткізуге мүмкіндік алады: сабақтар, емтихандар, тесттер кестесі;

- студенттердің өздерінің үлгерімі жайында ақпарат алу.

Ата-аналар:

- балаларыңыздың үлгерімін бақылау;
- web- интерфейс арқылы оқу процесіне қатысты ақпаратқа жылдам, қарапайым және жедел қол жеткізуге: сабақтар, емтихандар, тесттер кестесі;
- білім беру мекемесінің қызметкерлерімен және әкімшілігімен онлайн режимінде сөйлесу мүмкіндігін алу.

1.3 Есептің қойылымы

Білім беру орындарына арналған білім алушылардың сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйе ең алдымен мұғалімдер жұмысын, одан кейін басшылықтың да жұмысын оңтайландырады, студенттерге өз үлгерімі туралы оңтайланған қолжетімділікті қамтамасыз етеді. Бұл қосымшаны жалпы, орта және жоғарғы оқу орындары да бірдей қолдана алады.

Бұл дипломдық жұмыстың мақсаты білім алушылардың сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйе құру болып табылады. Қойылған мақсатты орындау келесілерді орындауды қажет етеді:

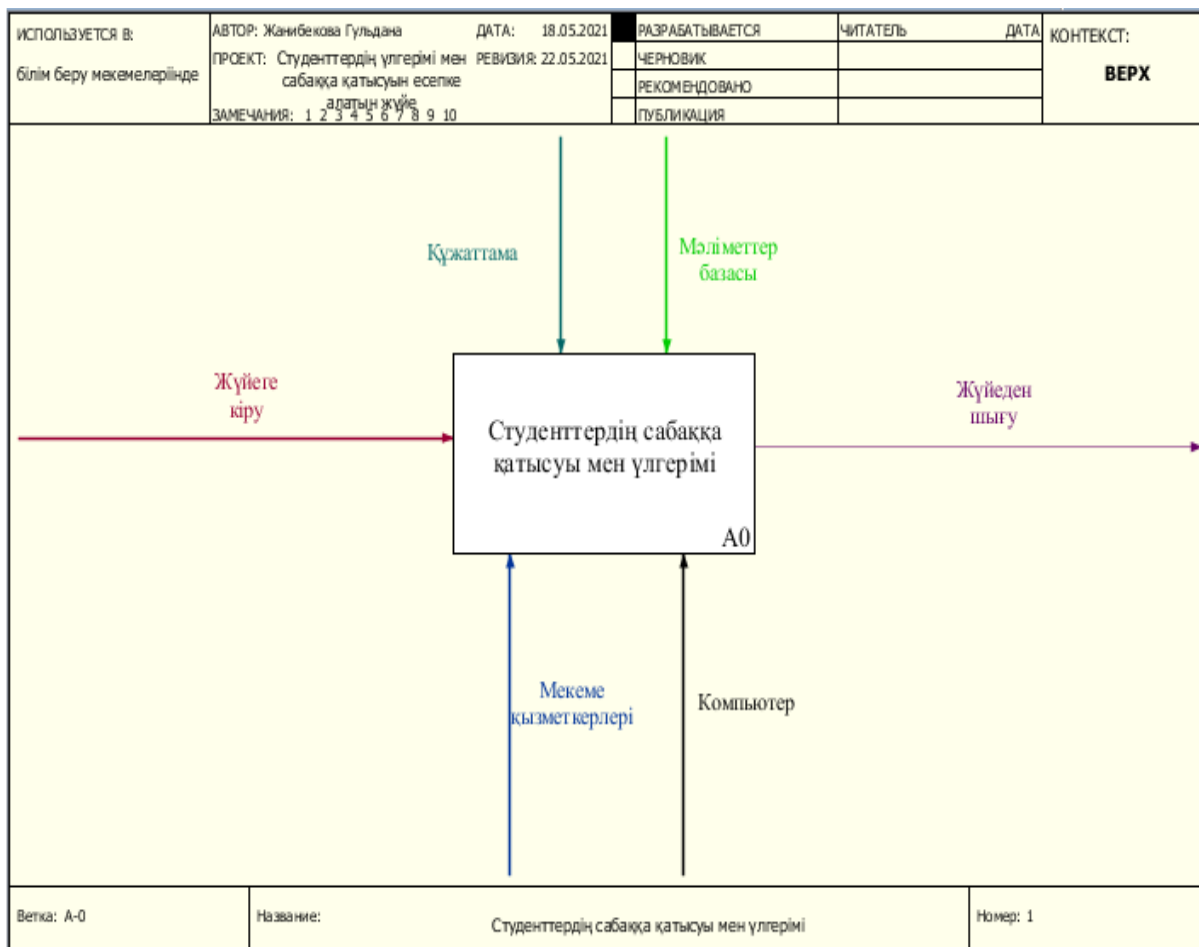
- ақпараттық жүйенің функционалдық құрылымын дайындау;
- ақпараттық жүйенің инфологиялық моделін жасау;
- мәліметтер базасын жобалау;
- қолданушыға ыңғайлы интерфейс құру;
- ақпараттық жүйенің бағдарламалық құралдарын таңдау;
- ақпараттық жүйені іске асыру.

Дипломдық жұмысты жүзеге асыру барысында C# программалау тілі, Microsoft Visual Studio 2019 ортасы, мәліметтер базасын құру үшін Microsoft SQL Server қолданылуы керек.

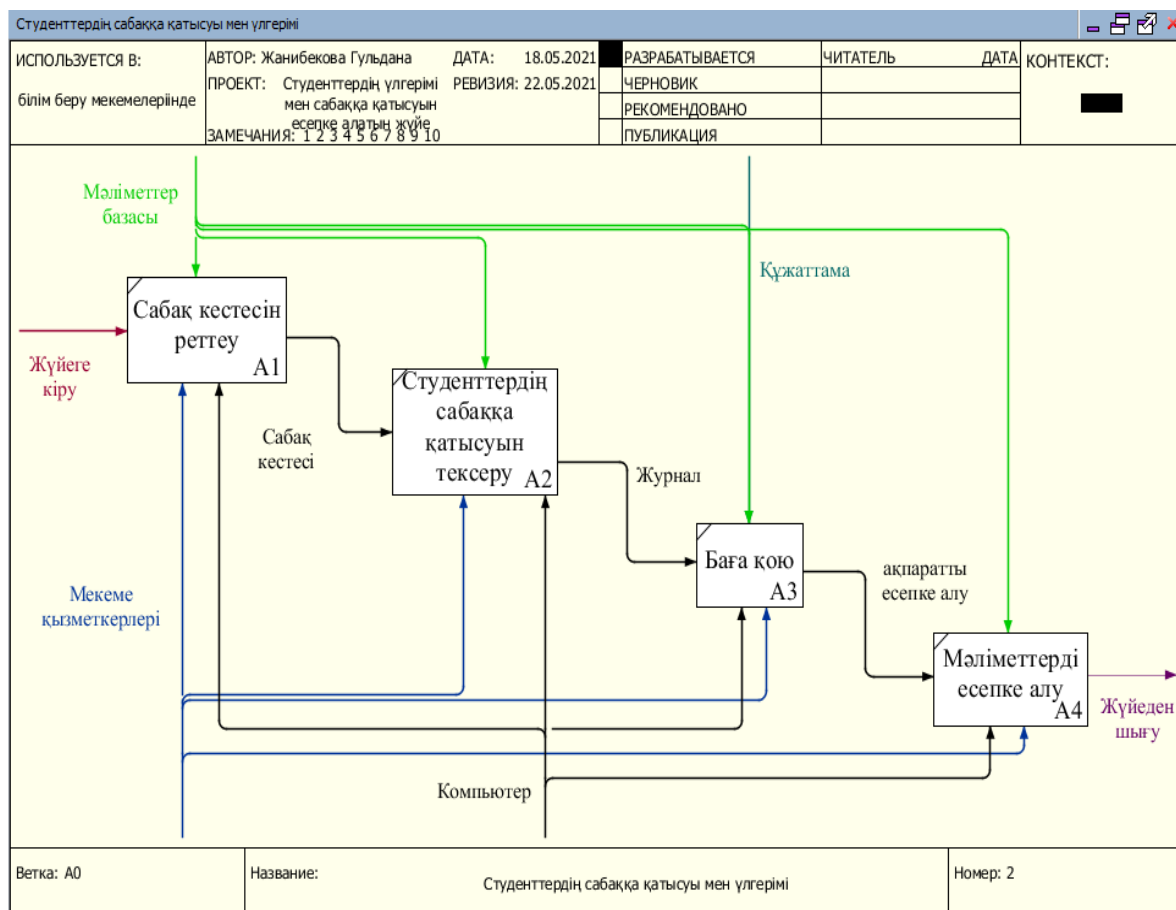
2 АҚПАРАТТЫҚ ЖҮЙЕНІ ЖОБАЛАУ

2.1 Жүйенің функционалды моделін құру

Жүйенің функционалды моделі қосымшаның ең үлкен құрылымдық бөліктерін таңдауға мүмкіндік береді, ал ол өз кезегінде әзірлеушінің мақсатын дәлірек анықтауға, мақсаттың орындалуына қажетті міндеттерге қол жеткізуге және ең жақсы сапалы жүйеге қол жеткізуге мүмкіндік береді. 2.1-суретте IDEF0 процестің функционалды моделінің контекстік диаграммасы көрсетілген.



2.1 сурет - IDEF0 процестің функционалды моделінің контекстік диаграммасы



2.2 сурет - Контексті диаграмманың декомпозициясы

IDEF0 стандартына сәйкес диаграммадағы төртбұрыштар блоктар немесе әрекеттер, олар нәтижеге апаратын процестерді, функцияларды, операцияларды бейнелейді. Құрылатын ақпараттық жүйе келесі процесстерге бөлінді:

- сабақ кестесін реттеу;
- студенттердің сабаққа қатысуын тексеру;
- баға қою;
- мәліметтерді есепке алу.

Блоктың сол жағындағы көрсеткіш кірісті білдіреді. Жоғарыдағы көрсеткіштер - жүйені жүзеге асыру үшін қолданылатын ереже, нұсқаулықтар, мәліметтер, бұл жүйеде мәліметтер базасы мен оқу орнына қатысты ережелер. Төменгі жақтағы көрсеткіштер - жұмысты орындауға арналған құрылғылар, ресурстар, құрылатын жүйеде білім беру мекемесінің қызметкерлері мен компьютер. Оң жақтағы көрсеткіштер - шығатын көрсеткі, соңғы іс-әрекет, жүйеден шығу процесі.

2.3 Жүйенің инфологиялық моделін құру

Жүйенің инфологиялық моделін құрудың мақсаты - мәліметтер базасында сақталатын ақпараттарды жинап, қолданушыға түсінікті жолмен ұсыну.

Инфологиялық модельдің негізгі элементтері: субъектілер, олардың арасындағы қатынастар және атрибуттар.

Келесі бөлімде «Студенттердің сабаққа қатысу мен үлгерімін есепке алатын ақпараттық жүйенің» инфологиялық моделі ER-диаграммасы құрылды.

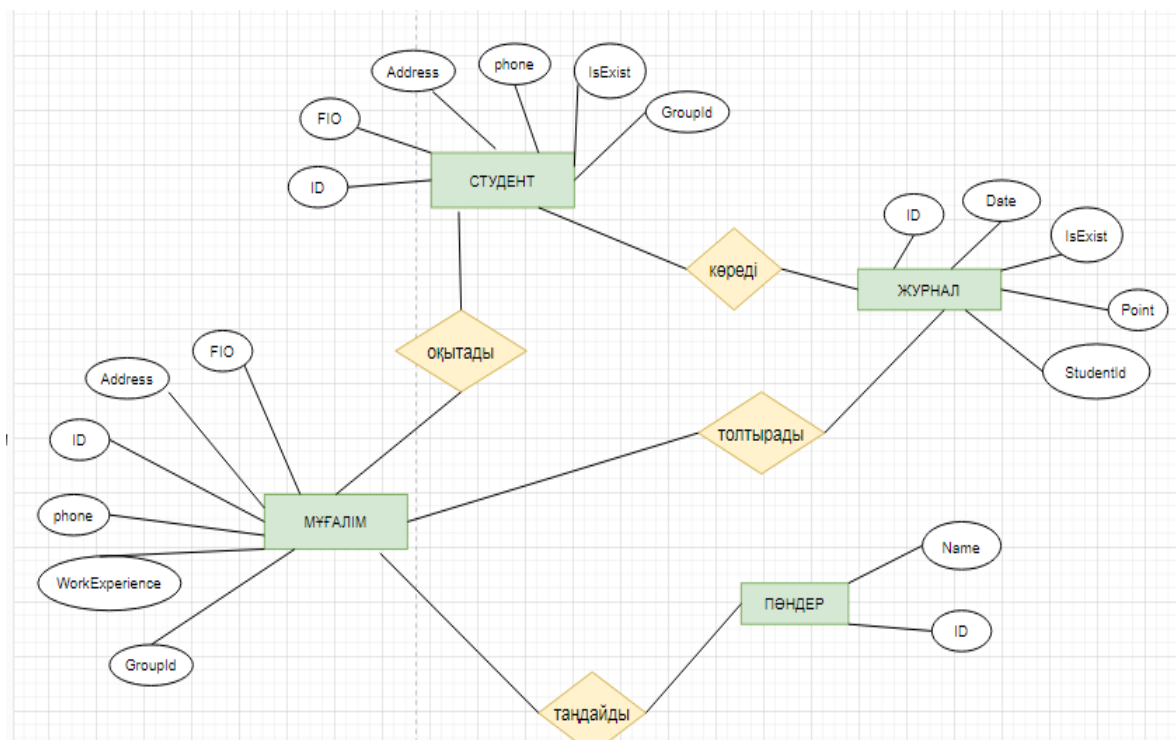
Жүйе бойынша барлық іс-әрекетті жүзеге асыратын бір бағдарлама мен компьютер бар. МҰҒАЛІМ тізімдегі СТУДЕНТТІ оқытады. МҰҒАЛІМ ЖУРНАЛ толтырады. МҰҒАЛІМ ПӘНДЕР таңдай алады. СТУДЕНТТЕР ЖУРНАЛДЫ көреді.

СТУДЕНТ, МҰҒАЛІМ, ПӘНДЕР субъектілері- жүйеде әрқашан болатын тәуелсіз деректер. ЖУРНАЛ- жүйенің басқа субъектілеріне тәуелді ақпарат. Оқытады, таңдайды, толтырады және тағы да басқалары- субъектілер арасындағы шартты қатынастар.

Барлық талаптарға сай субъектінің барлық атрибуттарымен, сипаттамаларымен және субъектілер арасындағы байланыс та қамтылған.

2.3.1 ER-диаграмма

«Субъект-қатынас» диаграммасы (ER-диаграмма)- бұл жүйе ішіндегі әртүрлі «объектілердің» (адамдар, объектілер, ұғымдар және т.б.) өзара байланысын көрсететін блок-схема. ER-диаграммалары көбінесе білім беру саласындағы реляциялық мәліметтер базасын жобалау, бизнеске арналған бағдарламалық және ақпараттық жүйелерді зерттеу үшін пайдаланады. ER-диаграмма нысандарды, олардың атрибуттары мен байланыстарын бейнелеу үшін стандартты белгілер жиынтығына сүйенеді.



2.3 сурет - Деректер базасының ER-диаграммасы

2.4 Мәліметтер базасын жобалау

Ақпараттық жүйе мәліметтер базасынан және мәліметтермен ыңғайлы әрі тез жұмысты қамтамасыз ететін клиенттік қосымшадан тұрады. Өз кезегінде, мәліметтер базасы студенттер мен оқытушылар, пәндер, группалар, мекеме туралы маңызды мәліметтерден тұратын кестелерді қамтиды.

Деректер базасы клиент-сервер технологиясының көмегімен құрылды, және клиент интерфейсі мәліметтер базасымен үйлесімді программалау тілінде құрылды. Клиент-сервер технологиясы мәліметтер базасына тез өзгерістер енгізуге, оның құрылымын оңай өзгертуге мүмкіндік беретіндігіне байланысты және файл-сервер технологиясымен салыстырғанда арзан болғандықтан таңдалды. Деректер базасы бірнеше байланыстарды қолдайтын деректер қорын басқару жүйесі (ДҚБЖ) негізінде дайындалады, осылайша бірнеше пайдаланушылар егер оларға құқықтары болса, бірден мәліметтермен жұмыс істей алады. Деректерге қолжетімділік сараланған: студенттер өздерінің сабаққа қатысуы мен үлгерімін көре алады, ал мұғалімдер оқушылардың сабаққа қатысуы мен үлгерімін, өз құзыретіне кіретін деректерді өңдеуге және жоюға қабілетті (бағалау, қатысу, оқушылар тізімі кестелері, оқушылар туралы ақпарат қосу, өшіру). Жүйе деректермен жұмыс процессін жылдамдатады, және оны ыңғайлы етеді. Жүйе деректер қауіпсіздігінің тиісті деңгейіне ие.

Оқу процессін автоматтандыруға арналған қолданыстағы өнімнің көп бөлігі білім беру процессінің кейбір бөліктерін ғана автоматтандыруға мүмкіндік береді, мысалы, оқушылардың жеке іс құжаттары, дипломға арналған жазбалар және басқа да сол сияқты. Бірақ бұл процесстердің барлығы бір-бірімен тығыз байланысты, сондықтан оларды бөлек автоматтандыру тиімсіз. Бөлек автоматика негізгі мәселені шешпейді, яғни автоматтандырылатын пәндік аймаққа қатысты кез-келген ақпаратқа жылдам қол жеткізуді қамтамасыз етеді. Менің ойымша, білім беру мекемесіндегі барлық процесстер кешенді түрде автоматтандырылуы керек, сондықтан оларға қатысты барлық ақпарат бірыңғай мәліметтер базасында болуы керек.

Әлемдегі ең танымал деректер қорын басқару жүйелерінің (ДҚБЖ) бірі - SQL Server. Бұл МҚБЖ әртүрлі жобаларға жарамды: кішігірім қосымшалардан бастап үлкен жүктемелі жобаларға дейін.

SQL Server төмендегідей анықтамалармен сипатталады:

- өнімділік, яғни SQL Server өте жылдам.
- сенімділік және қауіпсіздік. SQL Server деректерді шифрлауды қамтамасыз етеді.
- қарапайымдылық. Бұл МҚБЖ-мен жұмыс істеу және басқару салыстырмалы түрде қарапайым.

MS SQL Server-де кез-келген ДҚБЖ-дағы сияқты орталық аспект мәліметтер базасы болып табылады. Мәліметтер қоры - дегеніміз бұл белгілі бір жолмен ұйымдастырылған мәліметтер қоймасы. Деректер базасын басқару үшін мәліметтер базасын басқару жүйелері қолданылады, және MS SQL Server де осындай деректер қорын басқару жүйелерінің бірі болып табылады.

Деректер базасын ұйымдастыру үшін MS SQL Server реляциялық модельді пайдаланады. Бұл мәліметтер базасының моделін 1970 жылы Эдгар Кодд жасаған. Қазіргі таңда бұл мәліметтер базасын ұйымдастырудың стандарты.

Реляциялық модель деректерді кесте түрінде сақтайды, олардың әрқайсысы жолдар мен бағандардан тұрады. Әр қатарда жеке объект сақталады, ал бағандарда сол объектінің атрибуттары болады. Кесте ішіндегі әр жолды анықтауға арналған бастапқы кілт қолданылады. Бастапқы кілт арқылы біз кестенің белгілі бір жолына жүгіне аламыз. Сәйкесінше, екі қатарда бірдей негізгі кілт бола алмайды. Кілттер арқылы бір кестені екінші кестемен байланыстыруға болады, яғни екі кесте арасында сілтемелер ұйымдастыруға мүмкіндік береді. Ал кестенің өзін қатынас ретінде ұсына аламыз.

Мәліметтер қорымен өзара әрекеттесуі үшін SQL (Structured Query Language) тілі қолданылады. ДҚБЖ сұранысты түсіндіреді және орындайды, кейін орындау нәтижесін клиентке жібереді.

2.5 Мәліметтер базасының кестелерін сипаттау

Мәліметтер базасының құрылымын құру үшін кестелердің қатынастары мен олардың арасындағы байланыстары анықталуы қажет. Мәліметтер кесте құрылымында сақталады. Жобаланған мәліметтер базасында барлық кестелер үшін басты кілт арнайы енгізілген идентификатор (id) болып табылады, мысалы мен құрған мәліметтер базасының ER- диаграммасы бойынша:

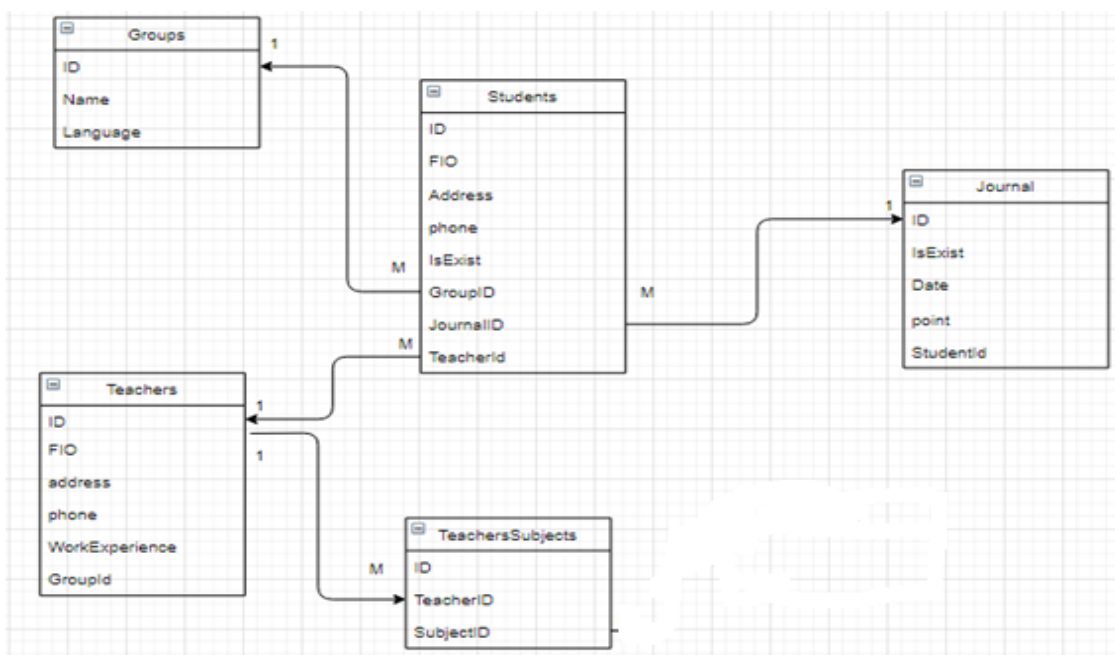
Teachers кестесі үшін teachers_id;

Groups кестесі үшін groups_id;

TeachersSubjects кестесі үшін teachersSubjects_id;

Journal кестесі үшін journal_id;

Students кестесі үшін students_id.



2.4 сурет - Кестелер арасындағы байланыс

Ақпараттық жүйе құру барысында жасалған мәліметтер базасының кестелері төменде көрсетілген.

Кесте 2.1 – Teachers

Кесте атауы	Өріс атауы	Мәліметтер типі
Teachers	ID	Int AUTOINCREMENT
	FIO	Nvarchar(200)
	Address	Nvarchar(50)
	Phone	Nvarchar(20)
	WorkExperience	Int
	GroupId	Int

Кесте 2.2 - Students

Кесте атауы	Өріс атауы	Мәліметтер типі
Students	ID	int AUTOINCREMENT
	FIO	Nvarchar(200)
	Address	Nvarchar(50)
	Phone	Nvarchar(20)
	IsExist	Bit
	TeachersId	Int
	JournalId	Int
	GroupId	Int

Кесте 2.3 – TeachersSubjects

Кесте атауы	Өріс атауы	Мәліметтер типі
TeachersSubjects	ID	int AUTOINCREMENT

	TeachersId	Int
	SubjectId	Int

Кесте 2.4 – Journal

Кесте атауы	Өріс атауы	Мәліметтер типі
Journal	ID	int AUTOINCREMENT
	IsExist	Bit
	Date	Date
	Point	Int
	StudentID	Int

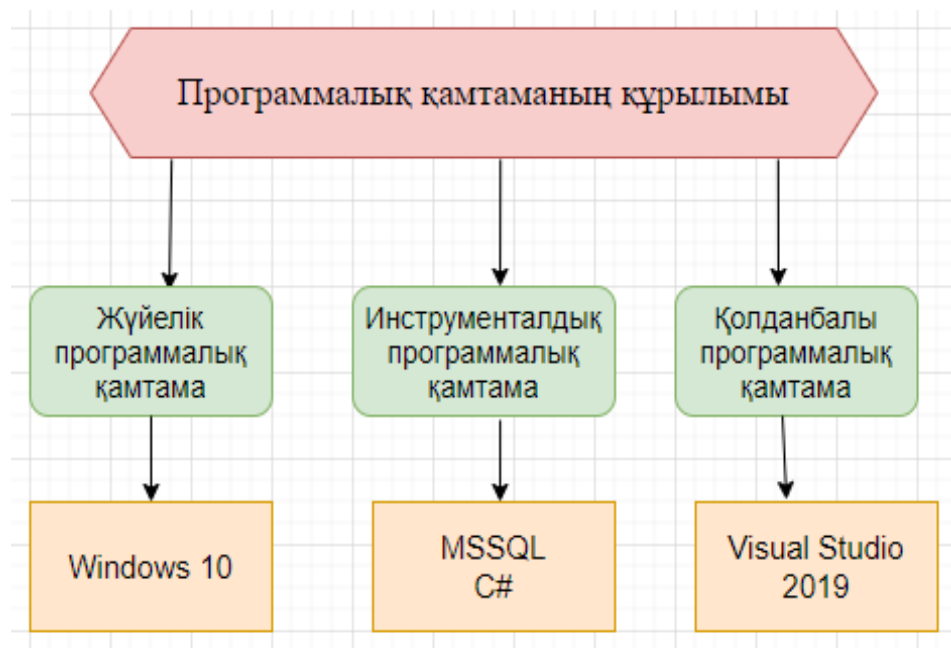
Кесте 2.5 - Groups

Кесте атауы	Өріс атауы	Мәліметтер типі
Groups	ID	int AUTOINCREMENT
	Name	Nvarchar(50)
	Language	Nvarchar(50)

3 АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ

3.1 АЖ жасауда қолданылатын құралдар

Бұл дипломдық жұмысты жүзеге асыру барысында программалық қамтаманың құрылымы төменде көрсетілгендей жүйелік, инструменталдық, қолданбалы программалық қамтамалар қолданылды.



3.1 сурет - Жүйенің программалық қамтаманың құрылымы

Microsoft Visual Studio – бұл C, C ++, C #, JavaScript, Python секілді атақты программалау тілдерін қолдайтын толық функционалды интеграцияланған даму ортасы (IDE). Visual Studio-ның функционалдылығы программалық жасақтаманы әзірлеудің барлық кезеңдерін, атап өткенде код жазуға, графикалық интерфейстерді жобалауға, қосымшаларды құруға, түзетуге және тестілеуге арналған заманауи құралдарды қамтиды. Visual Studio мүмкіндіктерін арттыру үшін өзімізге қажетті кеңейтілімдерді қосу арқылы көбейтуге болады.

Visual Studio бағдарламасының кірістірілген түзеткіші жазылған кодтағы қателерді табу және уақытылы түзету үшін қолданылады. Визуалды студияның форма дизайнері графикалық интерфейсі бар бағдарламаларды, болашақ қосымшаның сыртқы көрінісін және интерфейсстің әрбір элементінің жұмысын нақты жобалауға көмектеседі.

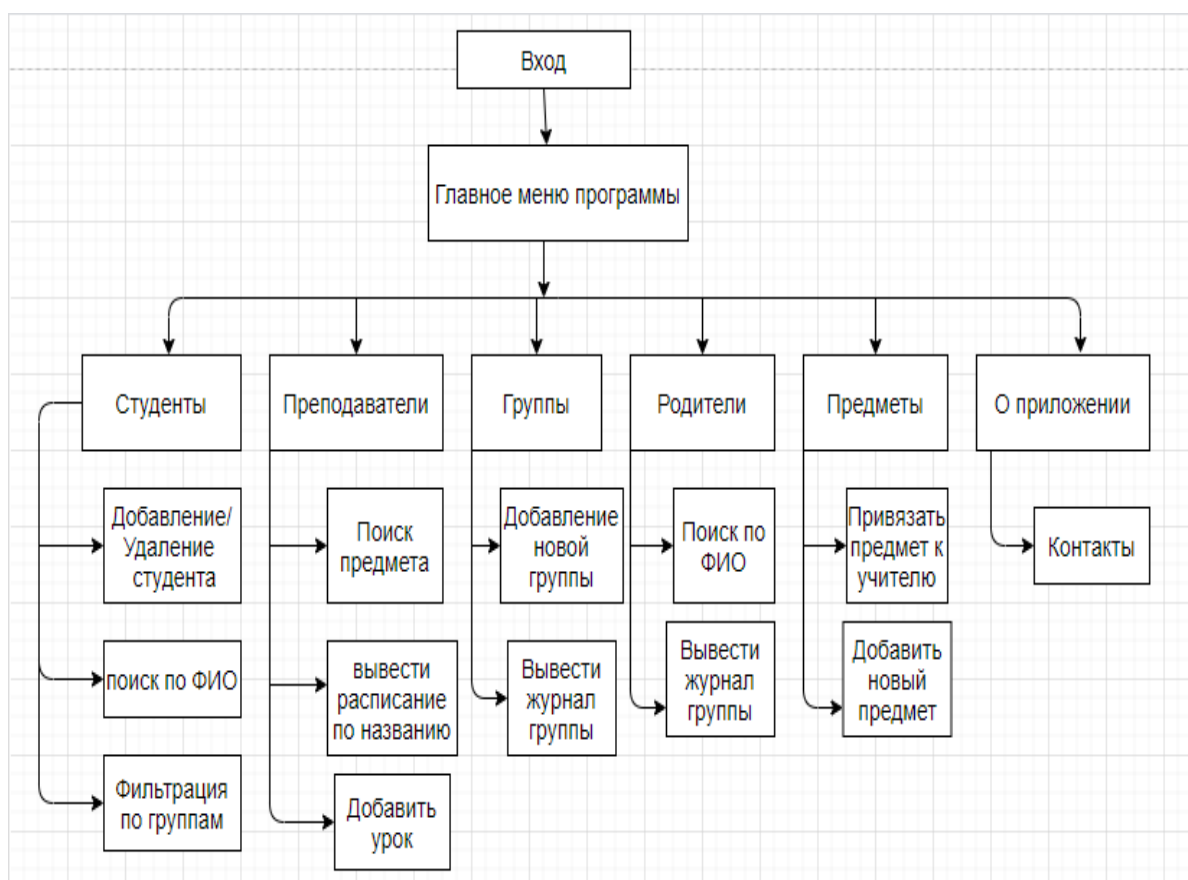
C# (Си шарп)- бұл объектілік бағытталған программалау тілі. Microsoft .NET платформасы үшін қосымшаларды әзірлеудің негізгі тілі ретінде Microsoft корпорациясында Андерс Хейлсберг бастаған инженерлер тобы 1998-2001 жылдары жасаған. C# компиляторы .NET стандартты қондырғысының құрамына кіреді.

C# синтаксисі C++ және Java-ға жақын синтаксисі бар тілдер тобына кіреді. C# типті жүйені тез және визуальды құру ортасы дейді, себебі ол өзіне компилятормен қоса редактор және 200-ге дайын компоненттер жиынтығын қамтиды. C# қолданушыға өте ыңғайлы және тез программа құруға мүмкіндік береді.

C# негізгі ерекшелігі - онда қосымша құруда компоненттік және объектілік тәсілдер қолданылады. Компоненттік тәсілдің мәнісі жеңіл: әр қосымша арнайы іс-әрекеттерді орындайтын компоненттерден құралады. Компоненттер визуальды компоненттер кітапханасында сақталған, пайдаланушы жаңа компонент дайындап, оны осы панельге қосуына да болады.

3.2 Ақпараттық жүйенің құрылымын жасау

Microsoft Visual Studio және SQL Server программалау құралдарын пайдаланып, білім беру мекемелерінің жұмысын автоматтандыруға арналған ақпараттық жүйе құрылды. Білім беру орталарына арналған автоматтандырылған жүйе мекеменің қызметкерлерінің жұмысын оңтайландыруға, студенттердің өз үлгерімі туралы ақпаратқа қолжетімділікті қамтамасыз етуге арналған. Бағдарламаны қолданушылар - білім беру мекемесінің қызметкерлері, студенттер. Бағдарлама студенттің келесі семестрге өту/өтпеуін енгізілген мәліметтерге сай автоматты түрде анықтайды.

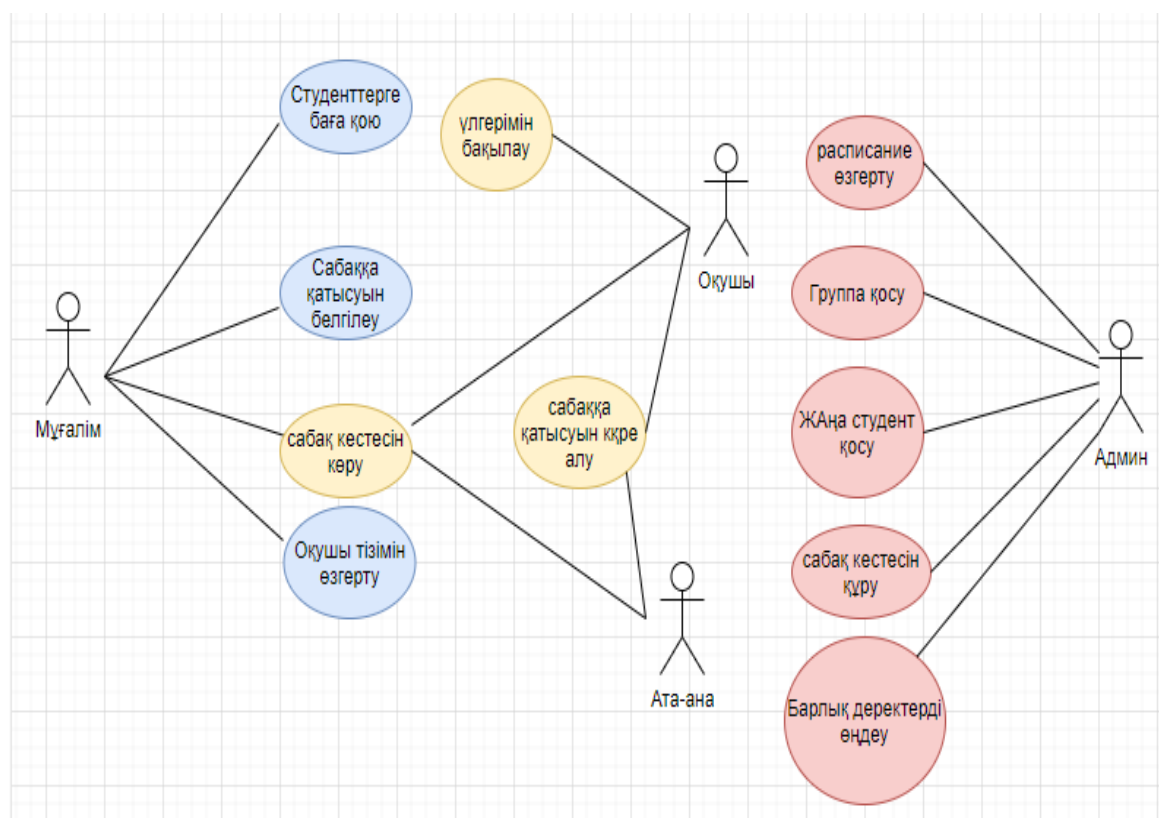


3.2 сурет - Ақпараттық жүйенің құрылымы

Білім беру жүйесінің қосымшасына кірген кезде басты бет пайда болады. Басты бетте оқу орны туралы мағлұмат және мекеме жұмысын оңтайландыратын бірнеше бөлімшеге бөлінген:

- авторизация;
- басты бет;
- мұғалімдерге;
- студенттерге;
- ата-аналарға;
- группа;
- пәндер;
- қосымша туралы;
- шығу.

Әр бөлімшеде сәйкесінше іс-әрекеттер орындалады.



3.3 сурет - Прецеденттер диаграммасы

Оқу орнының жұмысын жүргізу үшін студент туралы, оның бағалары және үлгерімі туралы нақты әрі сенімді ақпарат қажет.

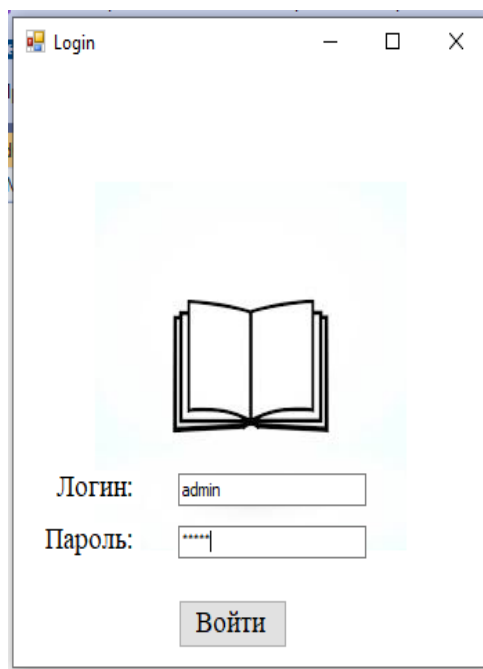
Дипломдық жұмыс барысында орта және жоғарғы білім беру орталықтарының студенттерінің оқу үлгерімі мен сабаққа қатысуын есепке алатын ақпараттық жүйе құрылды.

Кіріс мәліметтері: біріншіден, қолданушы енгізетін өзгерістер мен баға қойылымы; екіншіден, қолданушының студент туралы ақпаратқа сұраныстары.

Монитор арқылы жүйе өз жұмысы туралы ақпарат беріп, пайдаланушының әрі-қарайғы іс-әрекеттеріне бағыттайды.

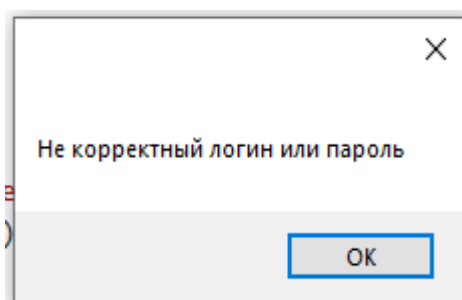
Шығыс мәліметтері: біріншіден, жаңартылған мәліметтер базасы және өзгерістер нәтижесін көрсететін монитордағы ақпараттар; екіншіден, іздеудегі сұраныстар бойынша шыққан мәліметтер.

3.3 Ақпараттық жүйенің жұмысын сипаттау



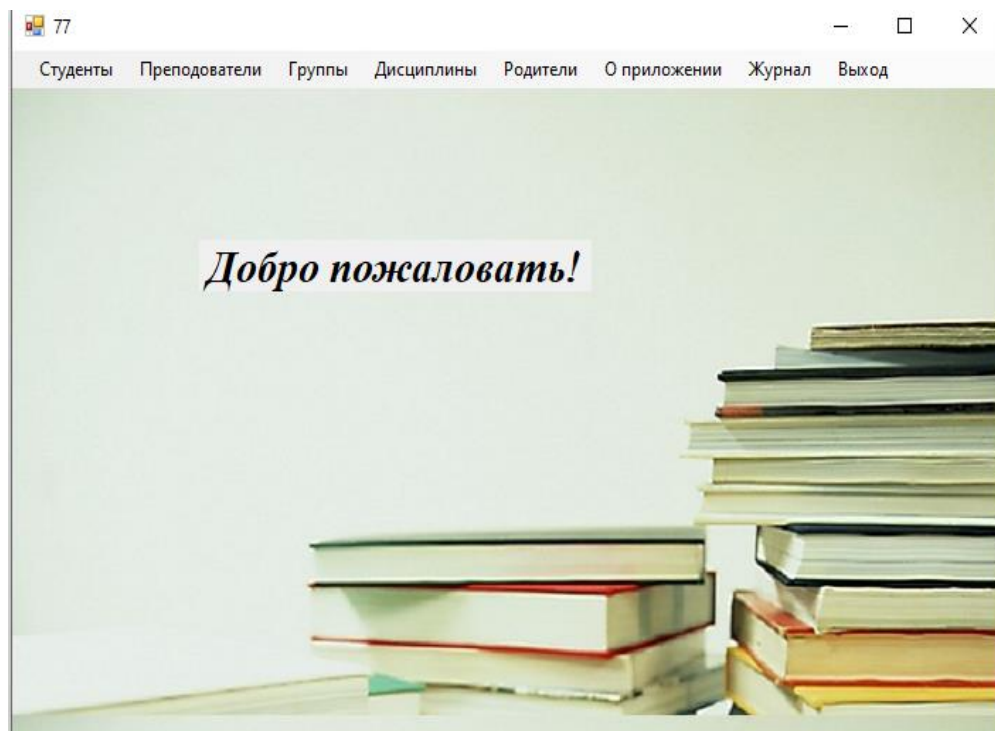
3.4 сурет - Жүйеге кіру терезесі

«Войти» батырмасы арқылы жүйеге кіре аламыз.



3.5 сурет – Авторизация терезесі

Егер логин мен пароль базада тіркелмеген болса немесе қате терілген болса, жүйеге кіргізбейді.

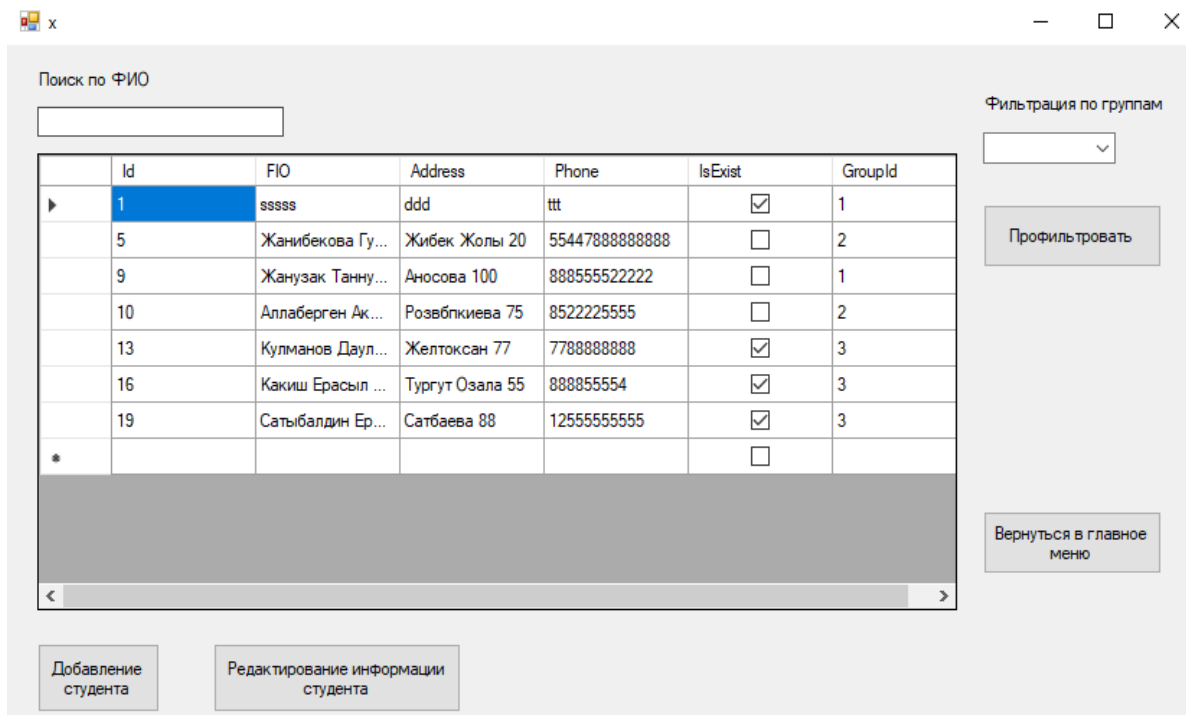


3.6 сурет - Бас мәзір бөлімі

Басты бетте оқу орны туралы мағлұмат және мекеме жұмысын оңтайландыратын бірнеше бөлімшеге бөлінген:

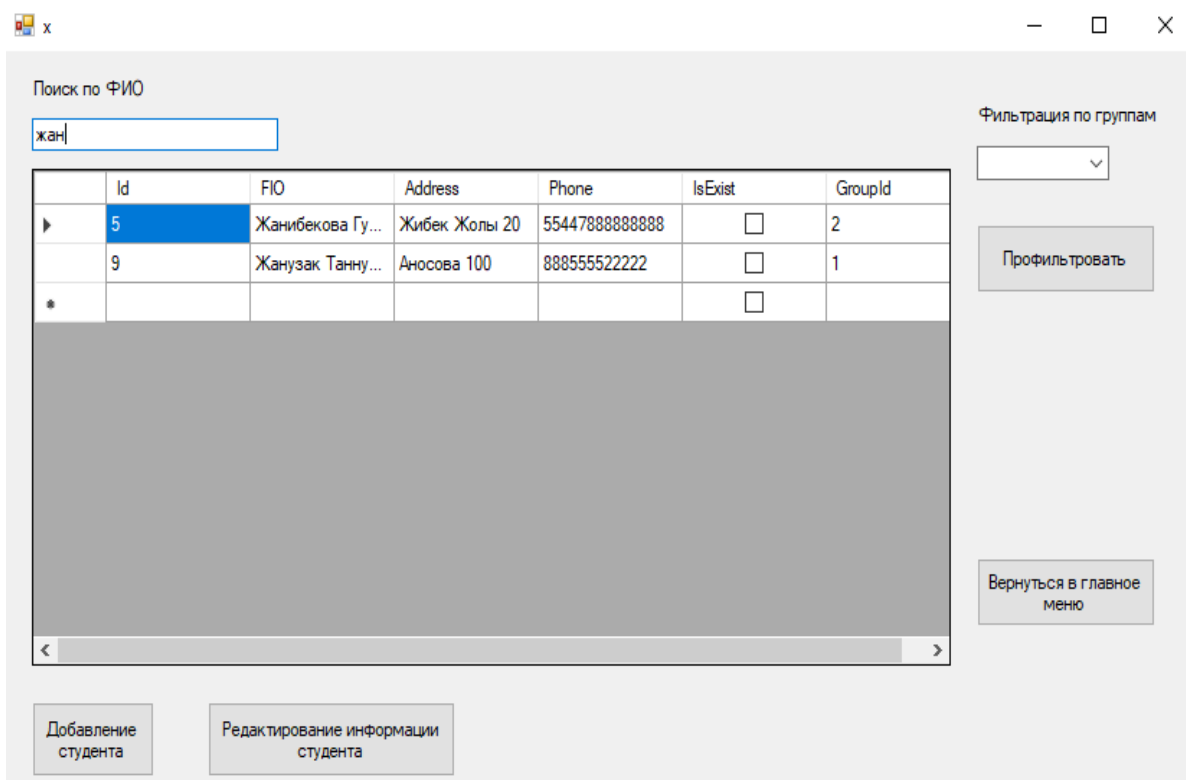
- Авторизация;
- Басты бет;
- Мұғалімдерге;
- Студенттерге;
- Ата-аналарға;
- Группа;
- Пәндер;
- Қосымша туралы;
- Шығу;

Әр бөлімшеде сәйкесінше іс-әрекеттер орындалады.



3.7 сурет - Студенттер бөлімі

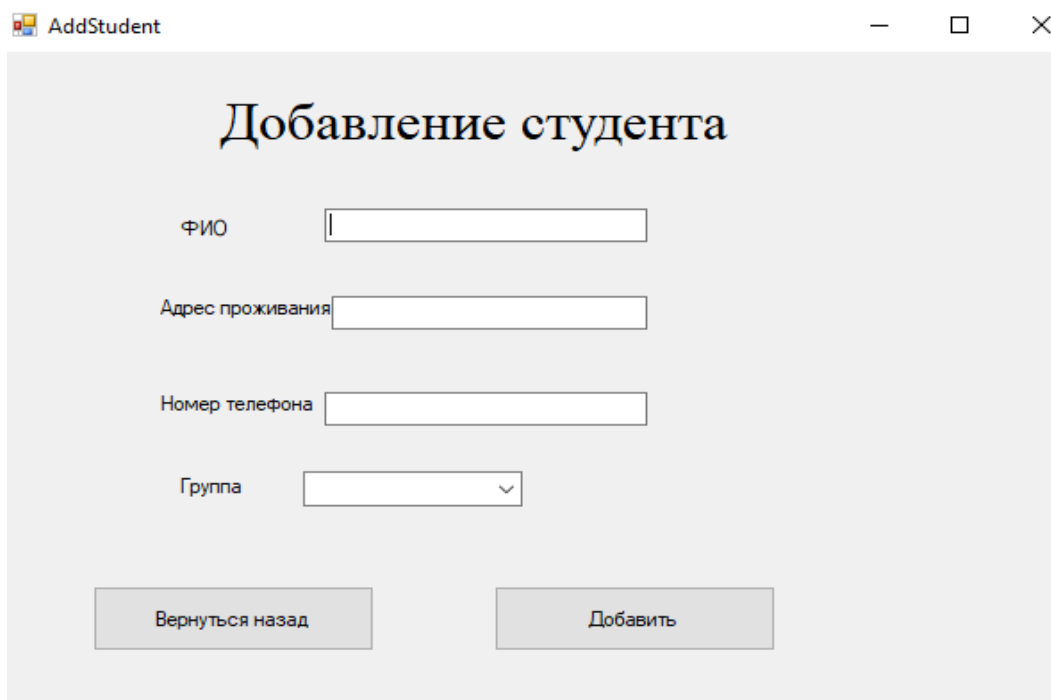
Бұл бөлімде студенттердің аты-жөні бойынша іздеуге болады, группа нөмері бойынша фильтрация жасалынады. Келесі 3.8 суретте студенттердің аты-жөні бойынша іздеу функциясы көрсетілген:



3.8 сурет – Аты –жөні бойынша іздеу функциясы

Сонымен қатар, «Жаңа студент қосу» және «Студенттің мәліметтерін өзгерту» батырмалары бар.

«Жаңа студентті қосу» батырмасын басқан кезде келесі бөлім ашылады:

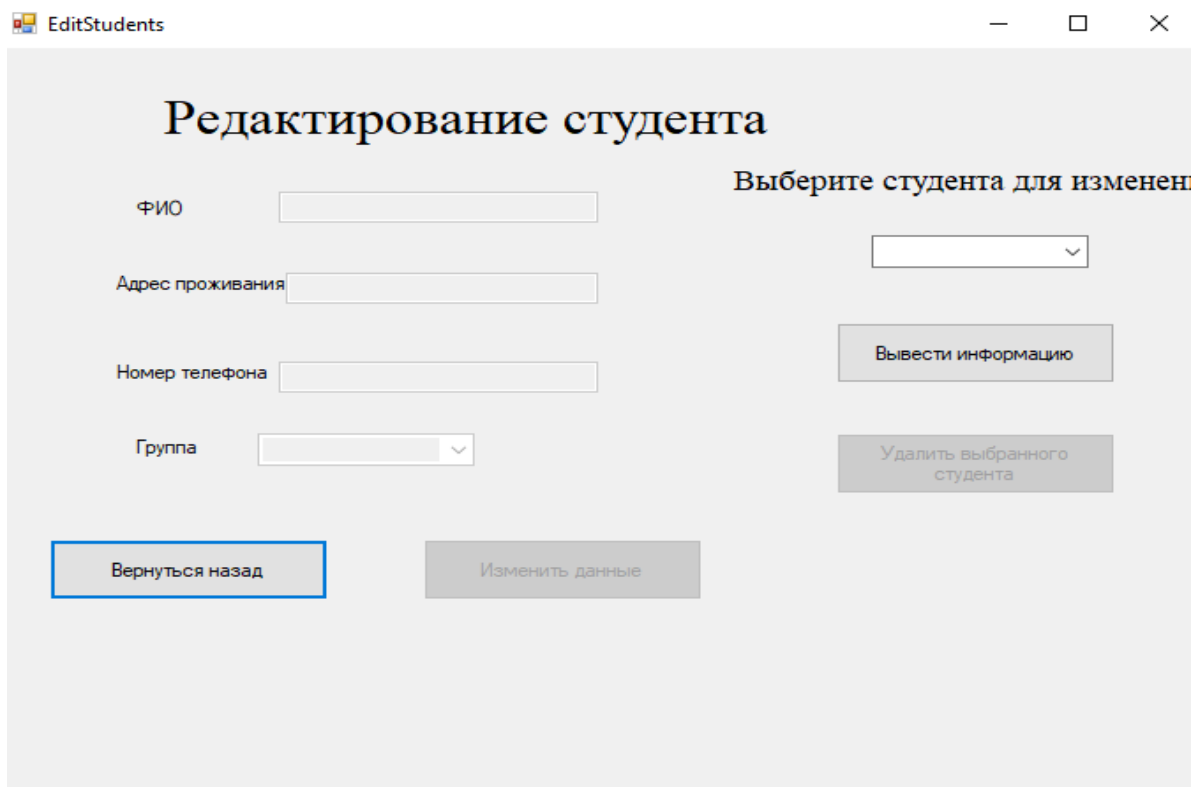


The screenshot shows a window titled 'AddStudent' with standard window controls (minimize, maximize, close). The main content area has a title 'Добавление студента' (Adding a student). Below the title are four input fields: 'ФИО' (Full name), 'Адрес проживания' (Residence address), 'Номер телефона' (Phone number), and 'Группа' (Group) which is a dropdown menu. At the bottom of the form are two buttons: 'Вернуться назад' (Go back) and 'Добавить' (Add).

3.9 сурет - Жаңа студентті қосу процесі

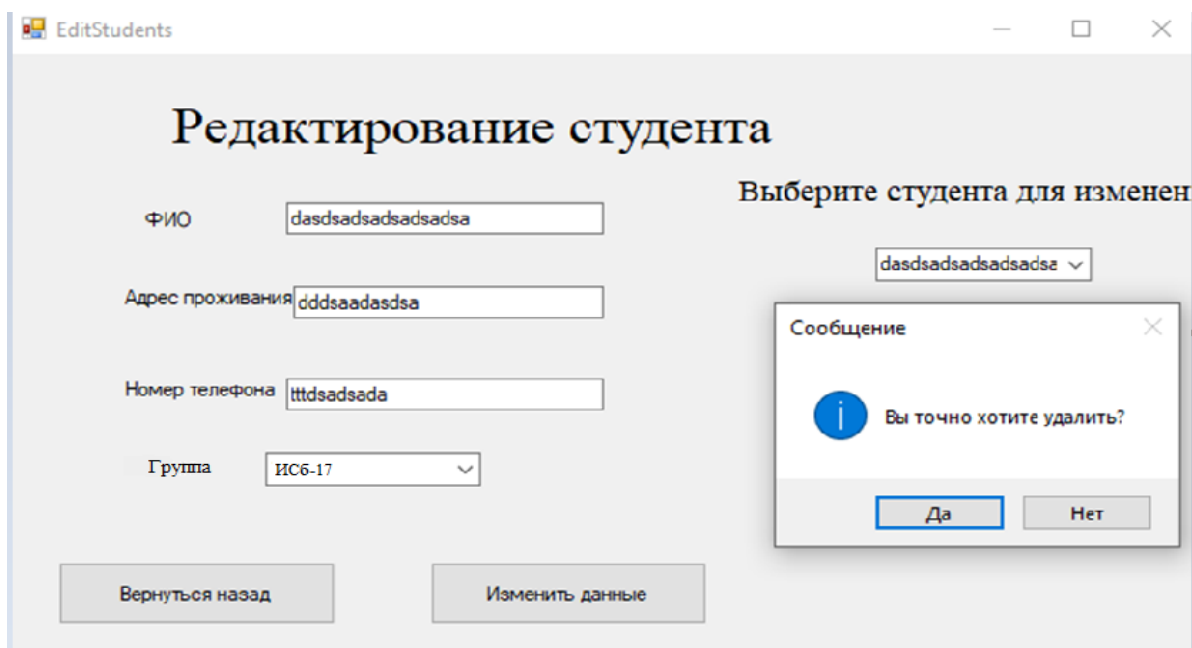
Барлық мәлімет толтырылып, «Добавить» батырмасын басқаннан кейін студенттер тізіме жаңа оқушы қосылады.

«Студенттің мәліметтерін өзгерту» батырмасын басқанда келесі бөлім ашылады:



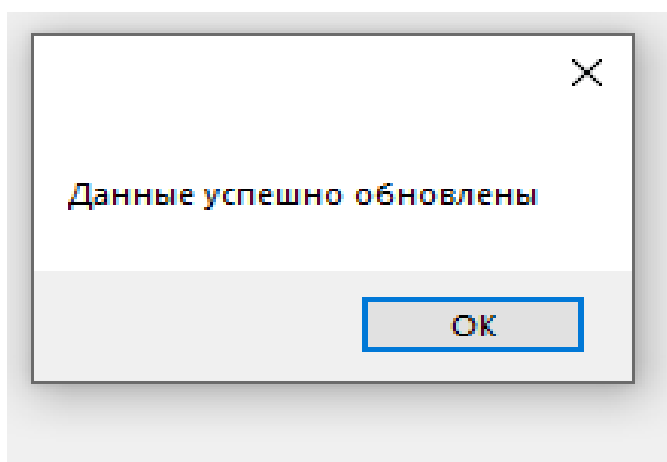
3.10 сурет - Тізімде бар студенттің мәліметтерін өзгерту процесі

Студент таңдалғаннан кейін ғана мәліметтер шығып, соларға өзгерту енгізе аламыз немесе толықтай өшіре аламыз. 3.11 суретте таңдалған студенттің мәліметтері шыққанын көруге болады.

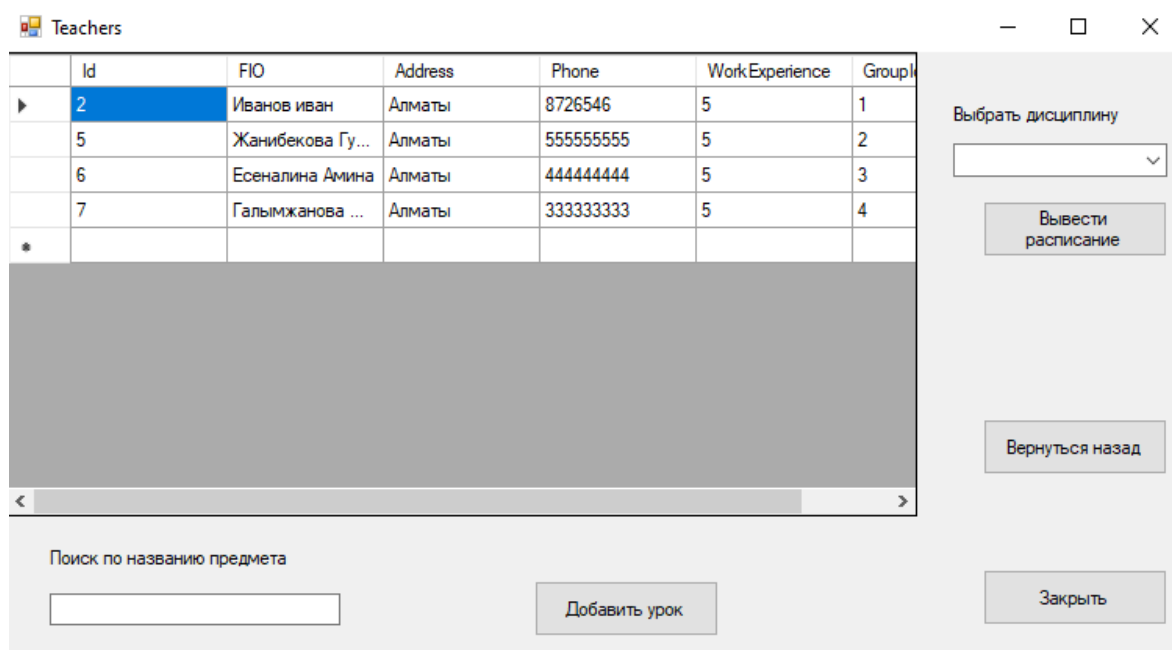


3.11 сурет - Тізімде бар студенттің мәліметтерін өзгерту процесі

Егер студентті тізімнен өшірмей, тек мәліметті өзгерткен жағдайда келесі терезе пайда болады:

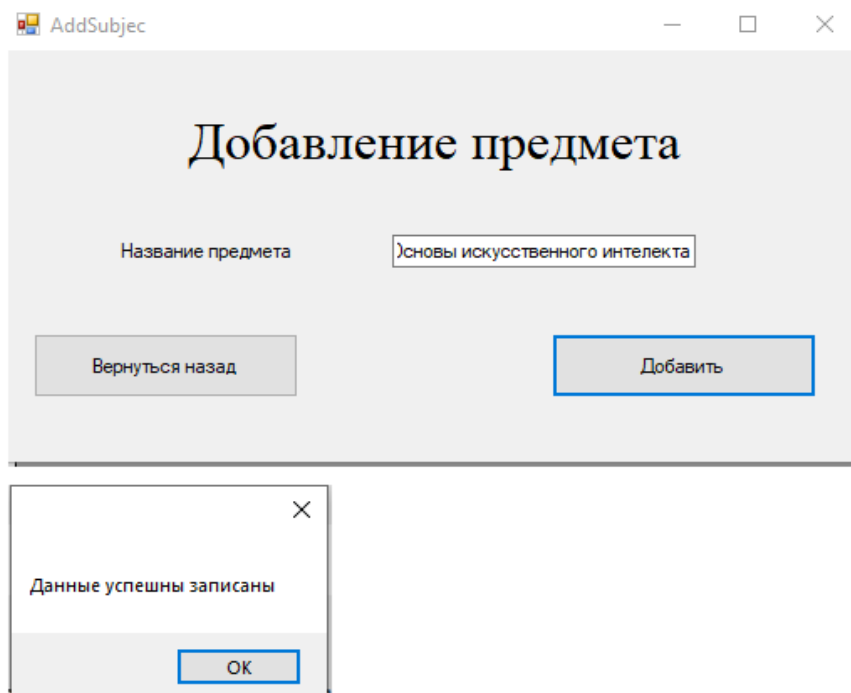


3.12 сурет - Студенттің мәліметін өзгерту процесі



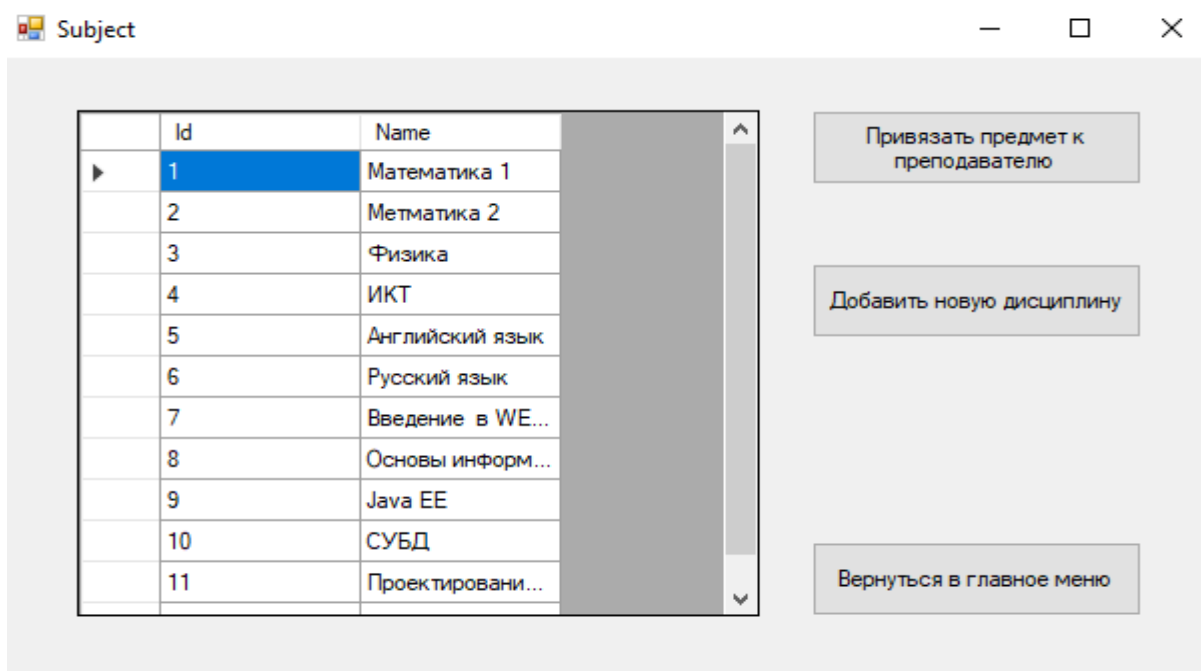
3.13 сурет - Мұғалімдер бөлімі

Дисциплина атауы бойынша сабақ кестесін көруге, пән атауы бойынша іздеуге, жаңа пән қосу мүмкіндіктері бар. Жаңа пән қосуға арналған батырма басқанда келесі бөлім ашылады:



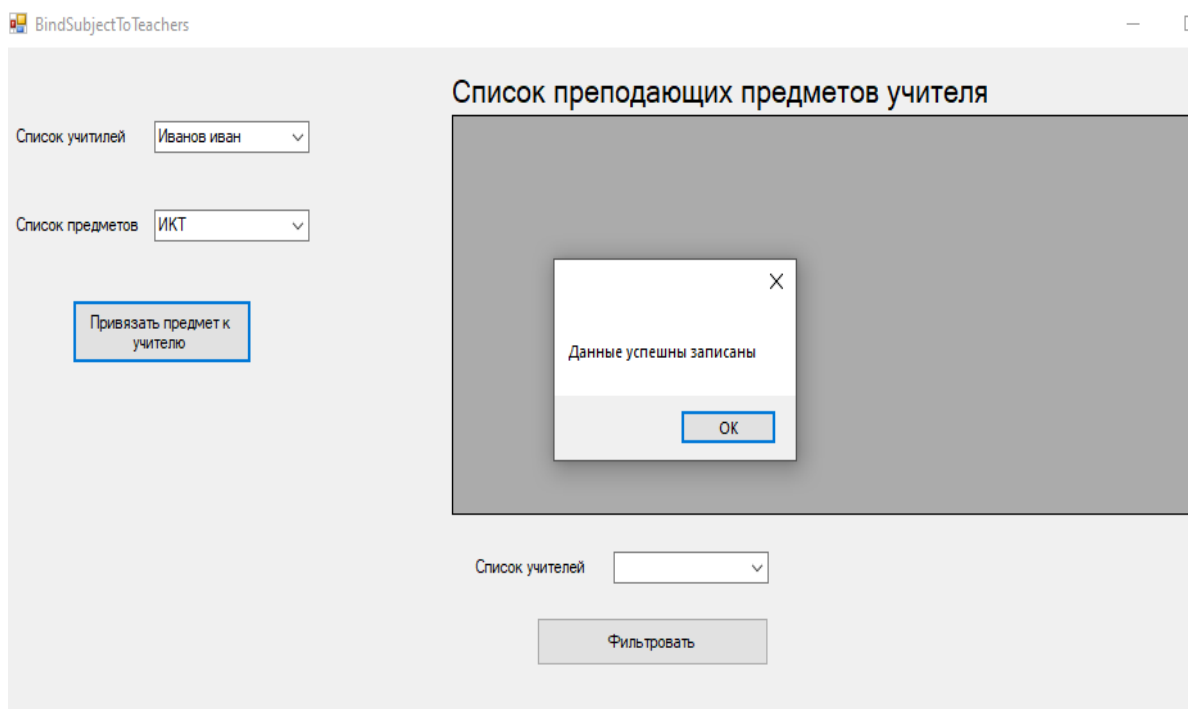
Сурет 3.14- Жаңа пән қосу процесі

Жаңа пән тізімге қосылғаннан кейін «Данные успешно записаны» деген терезе шығады.



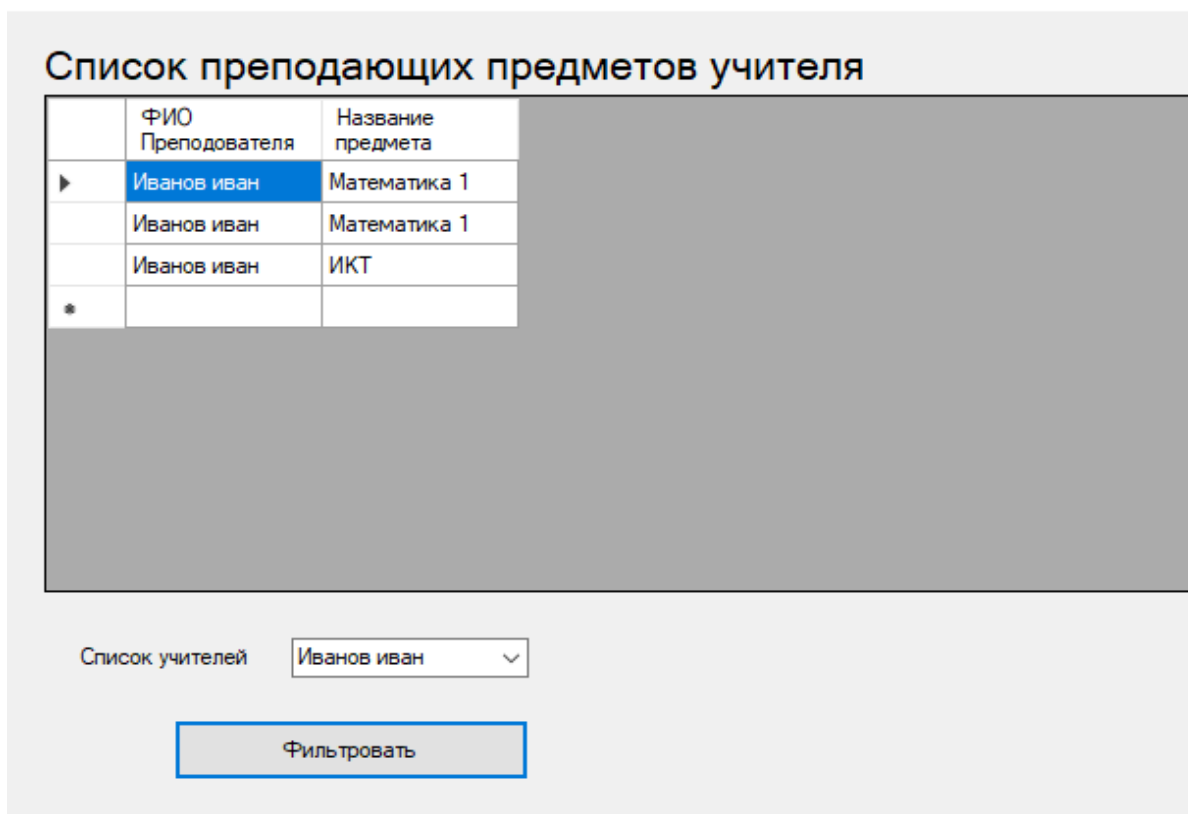
3.15 сурет - Пәндер бөлімі

Мұнда жоғарыда көрсетілген жаңа пән қосатын бөлімге өтуге болады. Жаңа мұғалімге пәнді немесе жаңа пәнді мұғалімге байланыстыру мүмкіндігі бар батырмасын басқан кезде:



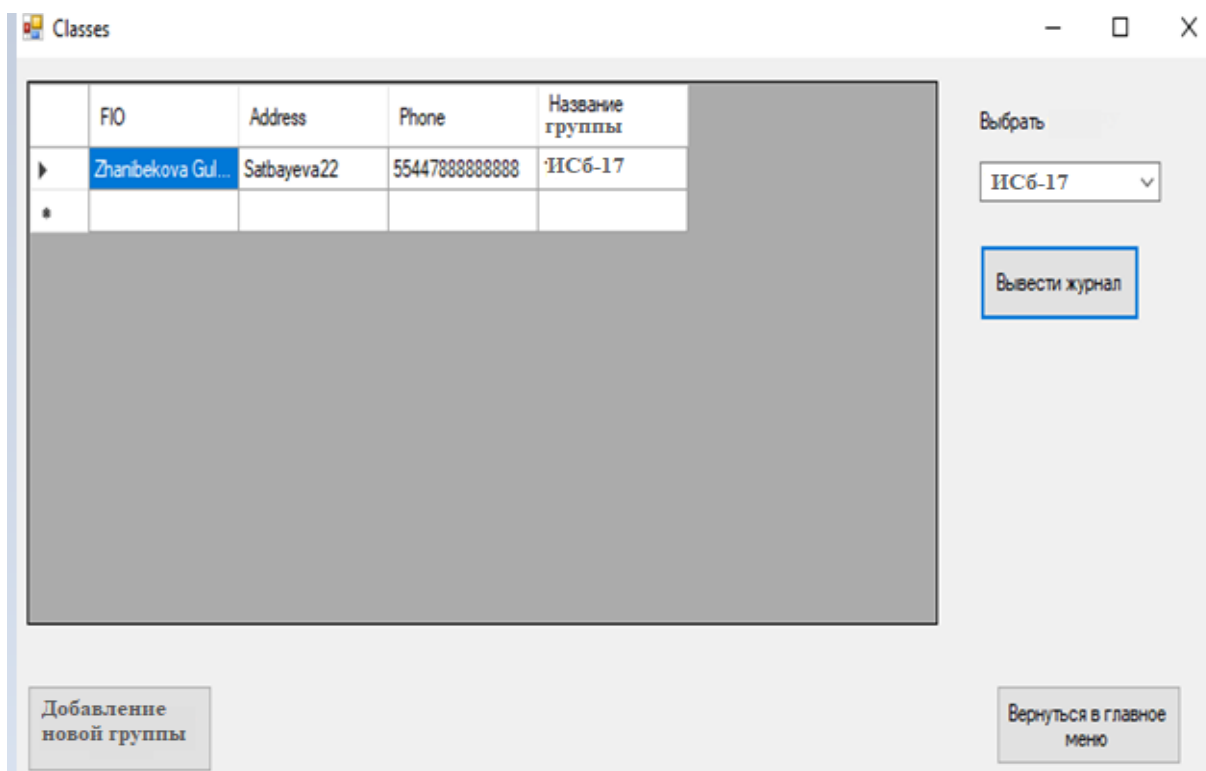
3.16 сурет - Мұғалімге жаңа пән қосу процессі

Негізгі процестен бөлек мұғалімдер тізіміне фильтр жасауға болады.



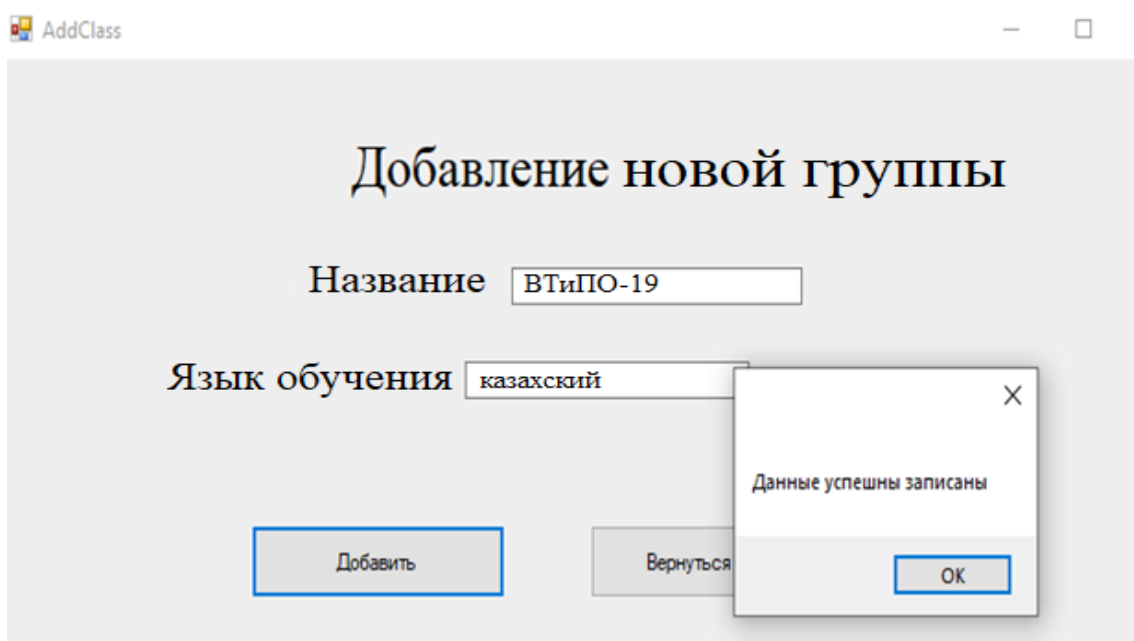
3.17 сурет - Мұғалімдерге тіркелген пәндер

Алдыңғы процессті тексеруге арналған бөлім

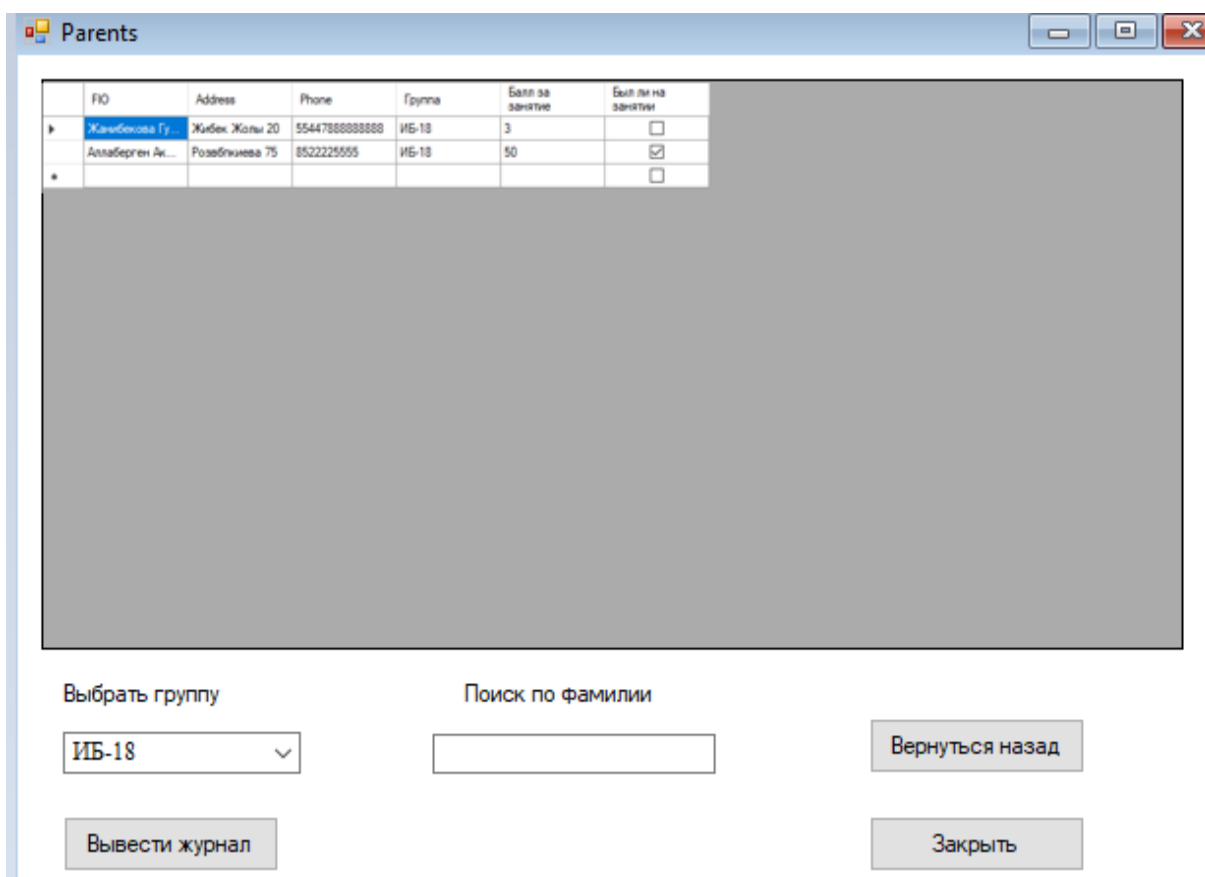


3.18 сурет – Группа бөлімі

Батырманы басқан кезде келесі терезе ашылады:

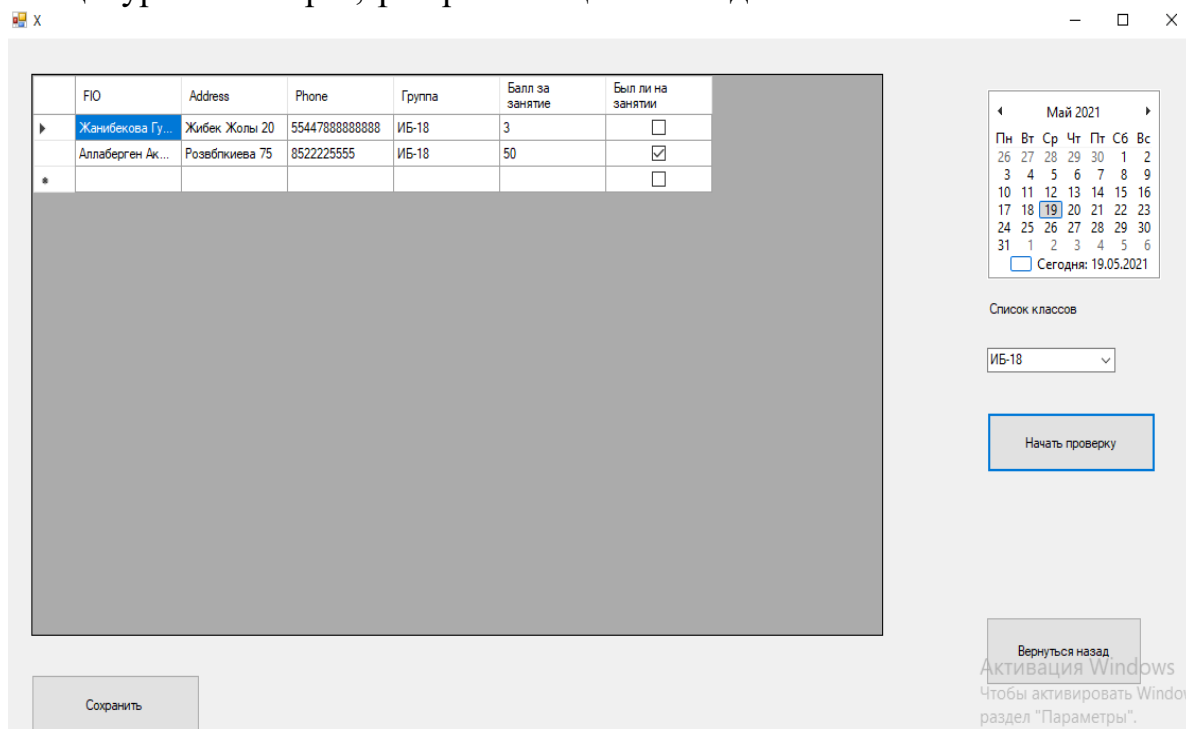


3.19 сурет – Жаңа топ қосу процессі



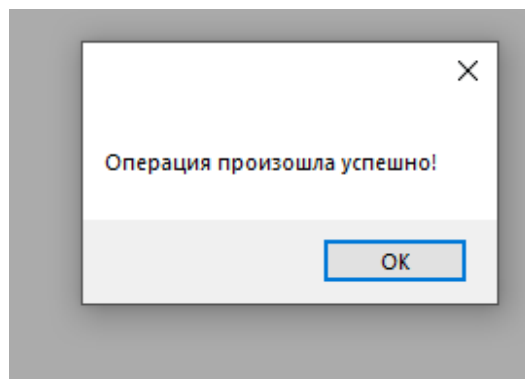
3.20 сурет – Ата-аналар бөлімі

Бұл бөлімде ата-аналар балаларының тобының атауы бойынша толық группаның журналын көріп, үлгерімін бақылай алады.



3.18 сурет - Журнал бөлімі

Бағдарламаның ең басты бөлімі, себебі бұл жерде студентке баға қойылады, сабаққа үлгерімін тексеріп белгілеуге болады және енгізілген мәліметтерді сақтауға болады. Алдымен группаны таңдап алу қажет, содан кейін журнал тізімі шыққаннан кейін, күнтізбе бойынша күнді таңдап, студенттерге баға мен сабаққа қатысуын белгілеуге болады.



3.19 сурет - Баға мен сабаққа қатысуына енгізілген өзгерістерді сақтау процесі

ҚОРЫТЫНДЫ

Дипломдық жұмыс барысында орта және жоғарғы білім беру орталықтарына арналған студенттердің үлгерімі мен сабаққа қатысуын есепке алатын ақпараттық жүйе жасалды.

Дипломдық жұмысты орындау нәтижесінде келесі нәтижелер алынды:

- қолданыстағы ақпараттық жүйелерге шолу жасалды;
- ақпараттық жүйенің функционалдық моделі жасалды;
- жүйенің инфологиялық моделі жасалды;
- ER-диаграмма құрылды;
- мәліметтер базасы жобаланды;
- жүйені жасауда қолданылатын құралдар таңдалды;
- интерфейс жобаланды;
- құрылған мәліметтер базасына негізделіп ақпараттық жүйе іске

асырылды.

Дипломдық жұмысты орындау барысында Microsoft SQL Server мәліметтер базасы, Microsoft Visual Studio ортасы және C# программалау тілі пайдаланылды.

Жасалған ақпараттық жүйе оқытушылардың, басшылықтың жұмысын оңтайландырады, студенттердің оқу үлгерімі мен сабаққа қатысуын есепке алуды қамтамасыз етеді.

ПАЙДАЛАНЫЛГАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Баженов Р.И., Баженова Н.Г., Белов И.В., Кардаш А.С. О разработке информационной системы оценки учебных достижений студентов // Современные научные исследования и инновации. 2014. № 12. Ч. 3 [URL:<https://web.snauka.ru/issues/2014/12/41514>]
- 2 Глушаков С.В., Ломотько Д.В. База данных: учебник / С.В. Глушаков –М.;Фолио,2006. -209с.
- 3 Избачков Ю.С., Петров В.Н. Информационные системы для вузов: учебное пособие. – СПб.:Питер,2005.-656с.
- 4 Красильникова М.В. Проектирование информационных систем: Учебное пособие- М.:2004.- 106с.
- 5 Карташов С.С. Образование и XXI век информационные и коммуникационные технологии.: учебное пособие. – М: Наука,2005.-191с.
- 6 Рудаков А.В. Технология разработки программных продуктов.: -М: Москва,2012г.
- 7 Семакин И.Г., Шестаков А.П. Основы программирования.-Москва, 2011г.
- 8 Чуйко О.И., Ещенко Р.А. Электронный журнал: анализ применения в школах и перспективы внедрения в вузах // Международный академический вестник,2014. №6. С.27-31.
- 9 Холзнер С. Программирование на С#. Учебный курс; Питер, 2001.- 576с.
- 10 Кузин А.В., Левонисова С.В. Базы данных:Учебное пособие для студентов высших учебных заведений. –М: ИЦ Академия, 2012.-320с.
- 11 Норенков И.П. автоматизированные информационные системы: Учебное пособие. –М.: МГТУ им.Баумана, 2011.-342с.
- 12 Робинсон С., Корнес О., Глинн Д. С# для профессионалов.- М.: Лори,2005.-396с.
- 13 Язык программирования С# и платформа .NET. Учебный портал. [URL: <https://metanit.com/sharp/>]

ҚОСЫМША

```
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 MyDiploma
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Equals("admin") && textBox2.Text.Equals("admin"))
            {
                this.TopMost = true;
                this.Hide();
                var form = new Form1();
                form.Closed += (s, args) => this.Close();
                form.Show();
            }
            else
            {
                MessageBox.Show("Не корректный логин или пароль");
            }
        }
    }
}
namespace MyDiploma
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void выходToolStripMenuItem_Click(object sender, EventArgs e)
```

```

        {
            this.Hide();
            this.Close();
        }
private void студентыToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Students();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
private void преподавателиToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Teachers();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
private void классыToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Groups();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
private void предметыToolStripMenuItem_Clic(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Subject();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
private void родителиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Parents();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
}

```

```

private void журналToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();

    var form = new Journal();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
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 MyDiploma
{
    public partial class Students : Form
    {
        public Students()
        {
            InitializeComponent();
            using (SqlConnection connection = new SqlConnection(@"Data Source=DESKTOP-
LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(String.Format("use diploma Select * from Students"), connection);
                    DataTable dataTable = new DataTable();
                    sqlDataAdapter.Fill(dataTable);
                    dataGridView1.DataSource = dataTable;
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
    }
}

```

```

    }
}
}

private void Students_Load(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma Select * from Students"), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;

            SqlCommand sqlCommand = new SqlCommand("use diploma select
id,concat(g.Name) from groups as g", connection);
            using (SqlDataReader reader = sqlCommand.ExecuteReader())
            {
                while (reader.Read())
                {
                    comboBox1.Items.Add(reader[1].ToString());
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect

```

```

Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            var selectedGroup = comboBox1.Text.Split(' ');

            SqlCommand sqlCommand = new SqlCommand(
                String.Format("use diploma select * from Students where GroupID =
'{0}'",
                    selectedGroup[1], selectedGroup[0]), connection);
            int groupId = Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
            SqlDataAdapter sqlDataAdapter = new
            SqlDataAdapter(String.Format("use diploma select * from Students where
Students.GroupID = " + groupId), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"))
    {
        try
        {
            SqlDataAdapter sqlDataAdapter = new
            SqlDataAdapter(String.Format("use diploma select * from Students as s where s.FIO
like '%{0}%", textBox1.Text), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
        }
    }
}

```



```

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

}

private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new MyDiploma.AddStudent();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

private void button3_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new MyDiploma.EditStudents();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

private void button4_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

}
}
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 MyDiploma
{
    public partial class Teachers : Form
    {
        public Teachers()
        {
            InitializeComponent();
        }
        private void Teachers_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(String.Format("use diploma Select * from Teachers"), connection);
                    DataTable dataTable = new DataTable();
                    sqlDataAdapter.Fill(dataTable);
                    dataGridView1.DataSource = dataTable;

                    SqlCommand sqlCommand = new SqlCommand("use diploma select
id,concat(g.Name) from groups as g", connection);
                    using (SqlDataReader reader = sqlCommand.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox1.Items.Add(reader[1].ToString());
                        }
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
    }
}

```

```

    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
        {
            try
            {
                SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma select * from Teachers as s where s.FIO
like '%{0}%", textBox1.Text), connection);
                DataTable dataTable = new DataTable();
                sqlDataAdapter.Fill(dataTable);
                dataGridView1.DataSource = dataTable;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new Form1();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new AddSubject();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
}

```

```

}
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 MyDiploma
{
    public partial class Groups : Form
    {
        public Groups()
        {
            InitializeComponent();
        }
        private void Classes_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlCommand sqlCommand = new SqlCommand("use diploma select
from Groups as g", connection);
                    using (SqlDataReader reader = sqlCommand.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox1.Items.Add(reader[1].ToString());
                        }
                    }
                }
                catch (Exception ex)
                {

```

```

        MessageBox.Show(ex.Message);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            var selectedGroup = comboBox1.Text.Split(' ');
            SqlCommand sqlCommand = new SqlCommand(String.Format("use
diploma select id from Groups as g where g.Name = '{0}'",
                selectedGroup[1], selectedGroup[0]), connection);
            int groupId =
Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
            SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma select s.FIO , s.Address,s.Phone
,CONCAT(g.ID, g.Name) as 'Название группы' from Students as s join Groups as g
on g.Id = s.GroupId where s.GroupId = " + groupId), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

```

```

    }
    private void button2_Click(object sender, EventArgs e)
    {
        this.TopMost = true;
        this.Hide();
        var form = new AddClass();
        form.Closed += (s, args) => this.Close();
        form.Show();
    }
    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {
    }
}
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 MyDiploma
{
    public partial class Subject : Form
    {
        public Subject()
        {
            InitializeComponent();
        }

        private void Subject_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {

```

```

        connection.Open();
        SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma Select * from Subject"), connection);
        DataTable dataTable = new DataTable();
        sqlDataAdapter.Fill(dataTable);
        dataGridView1.DataSource = dataTable;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

private void button1_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new AddSubject();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

private void button3_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new BindSubjectToTeachers();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}
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 MyDiploma
{
    public partial class Parents : Form
    {
        public Parents()
        {
            InitializeComponent();

            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma Select * from Students"), connection);
                    DataTable dataTable = new DataTable();
                    sqlDataAdapter.Fill(dataTable);
                    dataGridView1.DataSource = dataTable;
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try

```



```

        {
            SqlDataAdapter sqlDataAdapter = new
            SqlDataAdapter(String.Format("use diploma select * from Students as s where s.FIO
            like '{0}%", textBox1.Text), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        {
            using (SqlConnection connection = new SqlConnection(@"Data
            Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
            Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
            ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    var selectedClass = comboBox1.Text.Split(' ');
                    SqlCommand sqlCommand = new SqlCommand(
                    String.Format("use diploma select id from groups as g where g.Name
                    = '{0}'",
                    selectedClass[1], selectedClass[0]), connection);
                    int groupId =
                    Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
                    SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(String.Format("use diploma select * from Students where
                    Students.GroupID = " + groupId), connection);
                    DataTable dataTable = new DataTable();
                    sqlDataAdapter.Fill(dataTable);
                    dataGridView1.DataSource = dataTable;
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
    }
}

```

```

        }
    }
}
private void button3_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}
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 MyDiploma
{
    public partial class Journal : Form
    {
        public Journal()
        {
            InitializeComponent();
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                button2.Enabled = false;
                try
                {
                    connection.Open();
                    SqlCommand sqlCommand = new SqlCommand("use diploma select
Name from Groups", connection);
                    using (SqlDataReader reader = sqlCommand.ExecuteReader())

```

```

        {
            while (reader.Read())
            {
                comboBox1.Items.Add(reader[0].ToString());
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void button1_Click(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"))
    {
        try
        {
            if (comboBox1.SelectedItem != null)
            {
                connection.Open();
                SqlCommand sqlCommand = new SqlCommand(
                    String.Format("use diploma select id from groups where name =
'{0}'", comboBox1.Text), connection);
                int groupId =
                Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
                SqlDataAdapter sqlDataAdapter = new
                SqlDataAdapter(String.Format("use diploma select s.Id , s.FIO , s.Address,s.Phone ,
c.Name as 'Группа' from Students as s " +
                    "join Groups as c on c.Id = s.GroupID where s.GroupId = " +
                    groupId), connection);
                DataTable dataTable = new DataTable();
                sqlDataAdapter.Fill(dataTable);
                string selectedDate =
                monthCalendar1.SelectionRange.Start.ToString("dd/MM/yyyy");
                if (dataTable.Rows.Count > 0)
                {

```

```

        SqlCommand checkCommand = new
SqlCommand(String.Format("Select count(*) from Journal where Date = '{0}' and
StudentId = {1}", selectedDate, dataTable.Rows[0].ItemArray[0]), connection);
        int result = Convert.ToInt32(checkCommand.ExecuteScalar());
        if (result == 0)
        {
            foreach (DataRow sss in dataTable.Rows)
            {
                int studentId = Convert.ToInt32(sss.ItemArray[0]);
                bool isExist = false;
                int point = 0;
                string queryAdd = String.Format("use diploma Insert Journal
(IsExist,Date,Point,StudentId) " +
                    "values ('{0}','{1}','{2}','{3})", isExist, selectedDate, point,
studentId);
                SqlCommand sqlCommandInsert = new
SqlCommand(queryAdd, connection);
                sqlCommandInsert.ExecuteNonQuery();
            }

            SqlDataAdapter newSqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma select s.FIO , s.Address,s.Phone ,
c.Name as 'Группа',j.Point as 'Балл за занятие', j.IsExist as 'Был ли на занятии' from
Students as s " +
                "join Groups as c on c.Id = s.GroupID " +
                "join Journal as j on j.StudentId = s.Id " +
                "where s.GroupId = {0} and j.Date = '{1}'", groupId,
selectedDate), connection);
            DataTable newDataTable = new DataTable();
            newSqlDataAdapter.Fill(newDataTable);
            dataGridView1.DataSource = newDataTable;
        }
        else
        {
            SqlDataAdapter newSqlDataAdapter = new
SqlDataAdapter(String.Format("use diploma select s.FIO , s.Address,s.Phone ,
c.Name as 'Группа',j.Point as 'Балл за занятие', j.IsExist as 'Был ли на занятии' from
Students as s " +
                "join Groups as c on c.Id = s.GroupID " +
                "join Journal as j on j.StudentId = s.Id " +
                "where s.GroupId = {0} and j.Date = '{1}'", groupId,
selectedDate), connection);
            DataTable newDataTable = new DataTable();
            newSqlDataAdapter.Fill(newDataTable);

```

```

        dataGridView1.DataSource = new DataTable;
    }
    button2.Enabled = true;
}
}
else
{
    MessageBox.Show("Сначала выберите дату и группу!");
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}

private void button2_Click(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            foreach (DataGridViewRow s in dataGridView1.Rows)
            {
                if (s.Cells[0].Value != null)
                {
                    SqlCommand currentStudent = new
SqlCommand(String.Format("Select id from Students where FIO = '{0}'",
s.Cells[0].Value.ToString()), connection);
                    int studentId = Convert.ToInt32(currentStudent.ExecuteScalar());
                    bool isExist = Convert.ToBoolean(s.Cells[5].Value);
                    int point = Convert.ToInt32(s.Cells[4].Value);
                    string selectedDate =
monthCalendar1.SelectionRange.Start.ToString("dd/MM/yyyy");
                    SqlCommand updateQuery = new
SqlCommand(String.Format("Update Journal set Point = {0} , IsExist = '{1}' where
StudentId = {2} and Date = '{3}'", point, isExist, studentId, selectedDate), connection);
                    updateQuery.ExecuteNonQuery();
                }
            }
        }
        catch { }
    }
}

```

```

        }
    }
    MessageBox.Show("Операция произошла успешно!");
    dataGridView1.DataSource = null;
    dataGridView1.Rows.Clear();
    comboBox1.SelectedItem = null;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
private void button3_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}

}

}
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 MyDiploma
{
    public partial class AddClass : Form
    {
        public AddClass()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {

```

```

if (textBox1.Text == "" || textBox2.Text == "")
{
    MessageBox.Show("Сначала заполните все поля!!");
}
else
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            string letter = textBox1.Text;
            int studyYear = Convert.ToInt32(textBox2.Text);
            string queryAdd = String.Format("use diploma Insert Class
(Letter,StudyYear) " +
                "values ('{0}','{1}]", letter, studyYear);

            SqlCommand sqlCommandInsert = new SqlCommand(queryAdd,
connection);
            sqlCommandInsert.ExecuteNonQuery();
            MessageBox.Show("Данные успешно записаны");
            textBox1.Text = "";
            textBox2.Text = "";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Groups();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}

```

```

}
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 MyDiploma
{
    public partial class AddStudent : Form
    {
        public AddStudent()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" ||
                comboBox1.Text == "")
            {
                MessageBox.Show("Сначала заполните все поля!!");
            }
            else
            {
                using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
                {
                    try
                    {
                        connection.Open();
                        string fio = textBox1.Text;
                        string address = textBox2.Text;
                        string phone = textBox3.Text;
                        var selectedClass = comboBox1.Text.Split(' ');
                        SqlCommand sqlCommand = new SqlCommand(
                            String.Format("use Diploma select id from class as c where c.Letter
                            = '{0}' and c.StudyYear = {1}",

```



```

        selectedClass[1], selectedClass[0]), connection);
        int classId =
Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
        string queryAdd = String.Format("use diploma Insert Students
(FIO,Address,Phone,ClassId) " +
        "values ('{0}','{1}','{2}','{3}')", fio, address, phone, classId);
        SqlCommand sqlCommandInsert = new SqlCommand(queryAdd,
connection);
        sqlCommandInsert.ExecuteNonQuery();
        MessageBox.Show("Данные успешно записаны");
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        comboBox1.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}

private void AddStudent_Load(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            SqlCommand sqlCommand = new SqlCommand("use diploma select
id,concat(c.StudyYear,' ',c.Letter) from class as c", connection);
            using (SqlDataReader reader = sqlCommand.ExecuteReader())
            {
                while (reader.Read())
                {
                    comboBox1.Items.Add(reader[1].ToString());
                }
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Students();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}
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 MyDiploma
{
    public partial class AddSubject : Form
    {
        public AddSubject()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "")
            {
                MessageBox.Show("Сначала заполните все поля!!!");
            }
            else
            {
                using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated

```

```

Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            string subjectName = textBox1.Text;

            string queryAdd = String.Format("use diploma Insert Subject (Name) "
+
            "values ('{0}')" , subjectName);

            SqlCommand sqlCommandInsert = new SqlCommand(queryAdd,
connection);
            sqlCommandInsert.ExecuteNonQuery();
            MessageBox.Show("Данные успешны записаны");
            textBox1.Text = "";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Subject();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}
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 MyDiploma
{
    public partial class BindSubjectToTeachers : Form
    {
        public BindSubjectToTeachers()
        {
            InitializeComponent();
        }
        private void BindSubjectToTeachers_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlCommand sqlCommandTeacher = new SqlCommand("use diploma
select * from teachers", connection);
                    using (SqlDataReader reader = sqlCommandTeacher.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox1.Items.Add(reader[1].ToString());
                            comboBox3.Items.Add(reader[1].ToString());
                        }
                    }
                    SqlCommand sqlCommandSubject = new SqlCommand("use diploma
select * from subject", connection);
                    using (SqlDataReader reader = sqlCommandSubject.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox2.Items.Add(reader[1].ToString());
                        }
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
    }
}

```

```

    }
}
private void button2_Click(object sender, EventArgs e)
{
    using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
    {
        try
        {
            connection.Open();
            var teacherName = comboBox3.Text;
            SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(String.Format("select t.FIO as 'ФИО Преподавателя',s.Name as
'Название предмета' from Teachers as t join TeachersSubjects as ts on t.Id =
ts.TeacherId join Subject as s on s.Id = ts.SubjectId where t.FIO = '{0}'",
teacherName), connection);
            DataTable dataTable = new DataTable();
            sqlDataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button1_Click(object sender, EventArgs e)
{
    if (comboBox1.Text == "" || comboBox2.Text == "")
    {
        MessageBox.Show("Сначала заполните все поля!!!");
    }
    else
    {
        using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
        {
            try

```

```

        {
            connection.Open();
            SqlCommand sqlCommandTeacherId = new
SqlCommand(String.Format("use Diploma select id from teachers where FIO =
'{0}'",comboBox1.Text), connection);
            int teacherId =
Convert.ToInt32(sqlCommandTeacherId.ExecuteScalar());
            SqlCommand sqlCommandSubjectId = new
SqlCommand(String.Format("use Diploma select id from subject where Name = '{0}'",
comboBox2.Text), connection);
            int subjectId =
Convert.ToInt32(sqlCommandSubjectId.ExecuteScalar());
            string queryAdd = String.Format("use diploma Insert TeachersSubjects
(TeacherId,SubjectId) " +
            "values ('{0}',{1})", teacherId, subjectId);
            SqlCommand sqlCommandInsert = new SqlCommand(queryAdd,
connection);
            sqlCommandInsert.ExecuteNonQuery();
            MessageBox.Show("Данные успешно записаны");
            comboBox1.Text = "";
            comboBox2.Text = "";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button3_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Form1();
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}
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 MyDiploma
{
    public partial class EditStudents : Form
    {
        public int studentId;
        public EditStudents()
        {
            InitializeComponent();
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            comboBox1.Enabled = false;
            button1.Enabled = false;
            button4.Enabled = false;
        }
        private void EditStudents_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlCommand sqlCommand = new SqlCommand("use diploma Select *
from Students", connection);
                    using (SqlDataReader reader = sqlCommand.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox2.Items.Add(reader[1].ToString());
                        }
                    }
                    SqlCommand sqlCommandForClasses = new SqlCommand("use diploma
select id,concat(c.StudyYear,' ',c.Letter) from class as c", connection);
                    using (SqlDataReader reader = sqlCommandForClasses.ExecuteReader())
                    {
                        while (reader.Read())

```

```

        {
            comboBox1.Items.Add(reader[1].ToString());
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
private void button3_Click(object sender, EventArgs e)
{
    if (comboBox2.SelectedItem == null)
    {
        MessageBox.Show("Сначала выберите сотрудника!");
    }
    else
    {
        textBox1.Enabled = true;
        textBox3.Enabled = true;
        textBox2.Enabled = true;
        comboBox1.Enabled = true;
        button1.Enabled = true;
        button4.Enabled = true;
        try
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                connection.Open();
                string fio = comboBox2.SelectedItem.ToString();
                SqlDataAdapter adapter = new SqlDataAdapter("use diploma Select *
from Students where FIO ='" + fio + "'", connection);
                DataSet ds = new DataSet();
                adapter.Fill(ds);
                DataTable dt = ds.Tables[0];
                DataRow dataRow = dt.NewRow();
                foreach (DataRow i in dt.Rows)
                {
                    var cells = i.ItemArray;

```



```

        studentId = Convert.ToInt32(cells[0]);
        textBox1.Text = cells[1].ToString();
        textBox2.Text = cells[2].ToString();
        textBox3.Text = cells[3].ToString();
        SqlCommand sqlCommand = new SqlCommand("select
concat(c.StudyYear,' ',c.Letter) from class as c where id = " +
Convert.ToInt32(cells[5]), connection);
        comboBox1.Text = sqlCommand.ExecuteScalar().ToString();
    }
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
private void button1_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show(
        "Вы точно хотите изменить?",
        "Сообщение",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1);
    if (result == DialogResult.Yes)
    {
        using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
        {
            try
            {
                connection.Open();
                string fio = textBox1.Text;
                string address = textBox2.Text;
                string phone = textBox3.Text;
                var selectedClass = comboBox1.Text.Split(' ');
                SqlCommand sqlCommand = new SqlCommand(
                    String.Format("use Diploma select id from class as c where c.Letter
= '{0}' and c.StudyYear = {1}",
                    selectedClass[1], selectedClass[0]), connection);

```

```

        int classId =
Convert.ToInt32((sqlCommand.ExecuteScalar().ToString()));
        String queryUpdate = String.Format("Update Students set Fio = '{0}' ,
Address = '{1}' , Phone = '{2}' , ClassId = {3} where id = {4} ", fio, address, phone,
classId, studentId);
        SqlCommand sqlCommandUpdate = new SqlCommand(queryUpdate,
connection);
        sqlCommandUpdate.ExecuteNonQuery();
        MessageBox.Show("Данные успешно обновлены");
        comboBox2.Items.Clear();
        SqlCommand sqlCommandStudents = new SqlCommand("Select *
from Students", connection);
        using (SqlDataReader reader =
sqlCommandStudents.ExecuteReader())
        {
            while (reader.Read())
            {
                comboBox2.Items.Add(reader[1].ToString());
            }
        }
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        comboBox1.Text = "";
        comboBox2.Text = "";
        textBox1.Enabled = false;
        textBox2.Enabled = false;
        textBox3.Enabled = false;
        comboBox1.Enabled = false;
        button1.Enabled = false;
        button4.Enabled = false;
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void button2_Click(object sender, EventArgs e)
{
    this.TopMost = true;
    this.Hide();
    var form = new Students();
    form.Closed += (s, args) => this.Close();
}

```

```

        form.Show();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        DialogResult result = MessageBox.Show(
            "Вы точно хотите удалить?",
            "Сообщение",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Information);
        if (result == DialogResult.Yes)
        {
            using (SqlConnection connection = new SqlConnection(@"Data
Source=DESKTOP-LNSVKAU\SQLEXPRESS;Initial Catalog=Diploma;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWr
ite;MultiSubnetFailover=False"))
            {
                try
                {
                    connection.Open();
                    SqlCommand sqlCommandDelete = new SqlCommand("use Diploma
Delete Students where id =" + studentId, connection);
                    sqlCommandDelete.ExecuteNonQuery();
                    MessageBox.Show("Студент успешно удален");
                    textBox1.Text = "";
                    textBox2.Text = "";
                    textBox3.Text = "";
                    comboBox1.Text = "";
                    comboBox2.Text = "";
                    comboBox2.Items.Clear();
                    textBox1.Enabled = false;
                    textBox2.Enabled = false;
                    textBox3.Enabled = false;
                    comboBox1.Enabled = false;
                    button1.Enabled = false;
                    button4.Enabled = false;
                    SqlCommand sqlCommandStudents = new SqlCommand("use diploma
Select * from Students", connection);
                    using (SqlDataReader reader =
sqlCommandStudents.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            comboBox2.Items.Add(reader[1].ToString());
                        }
                    }
                }
            }
        }
    }
}

```

```
        }  
    }  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
}  
}  
}  
}
```

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ШҚІР

Жанибекова Гульдана Саматовна

5B070300 – Ақпараттық жүйелер

Тақырыбы: «Студенттердің сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйе құру»

Жанибекова Г. дипломдық жұмысының мақсаты студенттердің сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйесін жобалау болып табылады.

Дипломдық жұмыс кіріспеден, үш бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Дипломдық жұмыстың тақырыбы жоғары оқу орнында студенттердің сабаққа қатысуы мен үлгерімін автоматтандыру қажеттігімен таңдалды. Сондықтан, дипломдық жұмыстың тақырыбы өзекті болып табылады.

Дипломдық жұмысты жасау барысында жүйенің ақпараттық моделі құрылды; мәліметтер базасы мен интерфейсі қарастырылды. Заманауи ақпараттық жүйені жасайтын ортаны қолданып программалық қамтама жасалды.

Дипломдық жұмысты жасау барысында Жанибекова Г. жақсы теориялық дайындық және ақпараттық жүйелерді жасай алатын маман ретінде көрсетті. Сондай ақ, қазіргі кездегі ақпараттық жүйелерді жасауда қолданылатын құралдармен жұмыс істей білді.

Дипломдық жұмыс жоғарғы ғылыми техникалық деңгейде орындалған. Дипломдық жұмыстың барлық тараулары және программалық орта автордың өзімен жасалған.

Жоғарыда айтылғандарды ескеріп студент Жанибекова Г. «Студенттердің сабаққа қатысуы мен үлгерімін есепке алатын ақпараттық жүйе құру» тақырыбындағы жұмысы қорғауға жіберілуі мүмкін.

Ғылыми жетекші,
сенior-лектор, т.ғ.к.

Байматаева Ш.М.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

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

Автор: ~~Жанибекова Гульдана~~

Название: ~~Студенттердің сабаққа катысуы мен үлгерімінің есепке алатын ақпараттық жүйе құру~~

Координатор: ~~Шолпан Байматаева~~

Коэффициент подобия 1: 1,73

Коэффициент подобия 2: 0,0

Замена букв: 1

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

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

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

Обоснование:

Заимствования являются добросовестными и не обладают признаками плагиата.

Дата 31.05.2021



Сейдова Н.А., зав. кафедрой КБОиХИ

Ф ~~КазНУТУ~~ 711-27. Протокол анализа Отчета подобия заведующего кафедрой

Протокол анализа Отчета подобия Научным руководителем

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

Автор: ~~Жанибекова Гульдана~~

Название: ~~Студенттердің сабаққа қатысуы мен үлгерімінің есепке алатын аппараты: жүйе кету~~

Координатор: ~~Шолпан Байматаева~~

Коэффициент подобия 1: 1,73

Коэффициент подобия 2: 0.00

Замена букв: 1

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:


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

Обоснование:

Заимствования являются добросовестными и не обладают признаками плагиата.

27.05.2021

Дата


Подпись Научного руководителя