

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Казиева Айдана Батырбайқызы

«UML тілі көмегімен ақпараттық жүйелерді жобалау»

**ДИПЛОМДЫҚ ЖҰМЫС**

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ



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

**ҚОРҒАУҒА ЖІБЕРІЛДІ**  
КАӨЖС кафедра меңгерушісі,  
тех.ғыл.канд, ассоц.  
профессор  
\_\_\_\_\_ Н.А.Сейлова  
« \_\_\_\_ » \_\_\_\_\_ 2021 ж.

## ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «UML тілі көмегімен ақпараттық жүйелерді жобалау»


5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған: 

Казиева А.Б.

Ғылыми жетекші:

сеньор-лектор, т.ғ.к., доцент

 Ш.М.Байматаева

«20» мамыр 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

**БЕКІТЕМІН**

КАӨС кафедра меңгерушісі,

тех.ғыл.канд, ассоц.

профессор

Н.А.Сейлова

«\_\_\_» \_\_\_\_\_ 2021 ж.

**Дипломдық жұмысты орындауға  
ТАПСЫРМА**

Білім алушы: Қазиева Айдана Батырбайқызы

Тақырыбы: «UML тілі көмегімен ақпараттық жүйелерді жобалау»

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

Аяқталған жұмысты тапсыру мерзімі 2021 жылғы « 20» мамыр

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

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) ақпараттық қамтаманы құру

б) программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар

Сызба материалдар: \_\_\_ слайдпен көрсетілген

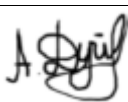
Ұсынылатын негізгі әдебиет: 15 атау

Дипломдық жұмысты дайындау

**КЕСТЕСІ**

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

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

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Норма бақылаушы	Дүйсенбаева А.Н. лектор, тех-ғыл магистрі	27.05.2021	
Программалық қамтама			

Ғылыми жетекші  Ш.М.Байматаева

Тапсырманы орындауға алған білім алушы  А.Б.Казиева

Күні

« 24 » 11 2020 ж.

## АҢДАТПА

Бұл жұмыста UML тілі көмегімен ақпараттық жүйелерді жобалау сұрақтары қарастырылған. Оны орындау барысында келесі нәтижелер алынды: қолданыстағы жүйелерге шолу жасалды; объектілі бағытталған тәсілді қолданып жүйенің моделі құрылды; мәліметтер базасы жобаланды; мәліметтер моделі жасалды; жасалған модельдерге негізделіп программалық қамтама іске асырылды.

Жасалған жүйенің көмегімен кітаптар туралы мәліметтер базасын жүргізуге, кітаптар мен оқырмандарды тіркеуге, мәліметтерді филтрлеу мен сұрыптауға, есеп беру құжаттарын шығаруға мүмкіндік береді.

Дипломдық жұмысты орындау үшін UML объектілі бағытталған ортасы; Microsoft Visual Studio ортасы; C# бағдарламалау тілі; Microsoft SQL Server Management Studio қолданылды.

## **АННОТАЦИЯ**

В данной работе рассмотрены вопросы проектирования информационных систем с применением языка UML. В ходе выполнения были получены следующие результаты: проведен обзор существующих систем; разработаны модели системы с применением инструментов объектно-ориентированного подхода; спроектирована база данных; разработана модель данных; реализовано программное обеспечение на основе разработанной базы данных.

Созданная система обеспечивает ведение базы данных о книгах, регистрировать книги и читателей, производить фильтрацию и сортировку данных, формировать отчетные документы.

Для выполнения дипломной работы были использованы: объектно-ориентированная среда UML; среда Microsoft Visual Studio; язык программирования C#; Microsoft SQL Server Management Studio.

## ANNOTATION

In this paper, the issues of designing information systems using the UML language are considered. During the implementation, the following results were obtained: an overview of existing systems was conducted; system models were developed using the tools of the object-oriented approach; a database was designed; a data model was developed; software was implemented based on the developed database.

The created system provides for maintaining a database of books, registering books and readers, filtering and sorting data, and generating reporting documents.

To complete the thesis, we used: object-oriented UML environment; Microsoft Visual Studio environment; C# programming language; Microsoft SQL Server Management Studio.

## МАЗМҰНЫ

КІРІСПЕ .....	9
1. АҚПАРАТТЫҚ ЖҮЙЕЛЕРДІ МОДЕЛЬДЕУ ТӘСІЛДЕРІНЕ ШОЛУ ...	10
1.1 Тақырып таңдаудың негіздемесі .....	11
1.2 Программалық қамтаманы жобалаудың методологиялары .....	13
1.2.1 Объектілі бағытталған тәсіл .....	14
1.2.2 Құрылымдық тәсіл.....	15
1.3 Есептің қойылымы.....	17
2. АҚПАРАТТЫҚ ЖҮЙЕНІҢ ОБЪЕКТІЛІ МОДЕЛІН ЖАСАУ .....	18
2.1 Прецеденттер диаграммасы .....	22
2.2 Іс әрекеттер диаграммасы .....	22
2.3 Тізбектер диаграммасы .....	23
2.4 Класстар диаграммасы .....	24
2.5 Мәліметтер базасы.....	25
3. ПРОГРАММАНЫ СИПАТТАУ.....	30
3.1 Программаның құрылымы мен жұмысын сипаттау.....	30
ҚОРЫТЫНДЫ.....	36
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ .....	37
ПРОГРАММА ЛИСТИНГІ .....	38



## КІРІСПЕ

Біздің уақытымызда компьютерлік технологияның құрылуына байланысты көптеген қозғалыстарды автоматтандыруға рұқсат етілді және өңделетін ақпарат көлемі де өсті. Қазіргі таңда адам еңбегінің үлкен үлесін автоматтандыру қажеттілігі артып отыр. Білім беруде, медицинада, түрлі кітапханаларда, өндірісте, қаржы менеджменті және саяси қайраткерлер саласында негізделген және тиімді қорытынды жасау үшін қазіргі заманғы маман ақпаратты жинау, сақтау және өңдеу, ақпаратты визуалды құжаттар түрінде көрсету, компьютерлік ресурстарды пайдалану, сондай-ақ өзара байланыс қабілетіне ие болуы керек. Кітапхана жұмысы қазіргі уақытта электронды жұмыстарының жылдам дамып келе жатқан саласы. Жыл сайын кітапхананың электронды қызметі әр түрлі онлайн-кітап тапсырыстарын жүзеге асыратын ең көп таралған құрылғы болып табылады. Қазіргі уақытта кез-келген жұмыс істейтін кітапханаларда - оқырмандардың көп болуына байланысты, егер кітапхана қызметкерлері қарапайым қағаздарымен жұмыс істей алмайтын жағдай туындауы мүмкін. Осы қарама-қайшылықтардың шешімін табу үшін пәндік аймақтың модификациясын құру теориясы көмектесе алады, сонымен қатар ақпарат негіздерін де қолдануға болады. Қазіргі қоғамда барлығы электронды кітапханаларының қызметтерін жиі пайдаланады, ал кітапхана қызметтері қоғамның өмір сүруіне таптырмайтын қолдау көрсететін құрал. Сонымен, бәсекеге қабілеттілігін арттыру үшін жылдам дамып келе жатқан шағын кітапханаларды, электр кітапханасын өзгерту керек, оның кейбір аспектілерін автоматтандыру қажет.

Осы технологиялармен бірге ақпараттық тұжырымдамалар, сондай-ақ ақпараттық технологияларды енгізу саласы қалыптасады. Әлеуметтік және экономикалық сипаттағы проблемалар 1-кезеңнен бастап компьютерлендірілген мәселелерді қолдауға мүмкіндігі жетеді. Дербес компьютерлер ұзақ уақыт бойы олар алғашқы толықтырулар болып саналмады, олар ғылыми-техникалық есептеулер жүргізді.

Ақпараттық жүйе – бұл, ақпараттық іс-қимылдарды жүзеге асыру үшін жалпы жиынтыққа біріктірілген өндірістік, бағдарламалық, ақпараттық, дәл, үйлестіруші, ғылыми-техникалық, сондай-ақ лингвистикалық ресурстар кешені-тұжырымдаманы пайдаланушылардың мақсаттары үшін қажетті деректерді алуды, сақтауды, редакциялауды, өңдеуді, іріктеуді, сондай-ақ беруді қамтиды. Жалпы жағдайда, ақпараттық ұғымдар, осы әрекеттерден басқа, деректерді қорғау, біріктіру, тазарту және нақтылау сияқты басқа да ақпараттық қозғалыстарды жүзеге асыру сияқты көптеген мүмкіндіктерге ие. Жүйенің құрылымдық компоненттері әдетте қамтамасыз ететін ішкі жүйелер немесе қауіпсіздік деп аталады. Осыған байланысты жұмыстың тақырыбы өзекті болып табылады.

# 1 АҚПАРАТТЫҚ ЖҮЙЕЛЕРДІ МОДЕЛЬДЕУ ТӘСІЛДЕРІНЕ ШОЛУ

Ақпараттық тұжырымдамаларды жобалау 3 негізгі тенденцияны қамтиды:

- ақпарат негізінде ақпараттарды құрастыру;
- жобаның дизайны, экранның конфигурациясы, ақпарат, ақпараттық қажеттілікті қанағаттандыру;
- белгілі бір аумақты немесе технологиялық процестерді, атап айтқанда желі топологиясын, архитектурасын, есептелген мәліметтерді өндеуді, синхронды өндеуді ескеретін құрылғының құрылымы (құжаттық компьютер немесе тұтынушы компьютері).

Ақпараттық тұжырымдамаларды жоспарлау - жоспардың миссиясын анықтаудан басталады. Жоспардың мақсаты - өзара тәуелді проблемалар жүйесін, оның ішінде оның қолданылу кезеңінде тұжырымдаманың іске қосылуының ресми болуын шешуге бағытталған.

Ақпараттық ұғымдарды жобалаудың әдістемелік негіздері.

АЖ жоспарының негізін қалаушылар:

- жобалау әдістері;
- технологиялық жобалау процестері;
- жобалау көздері (CASE құралдары).

Әдістерді, сонымен қатар ақпараттық тұжырымдамаларды жобалаудан алынған формальды үзіндінің маңызды бөлігін құрайтын құрылғыларды (CASE құрылғыларын) жобалайды.

Ақпараттық түсініктердің ағымдық ағымын қолдайтын бағдарламалық қамтамасыздандыру CASE құрылғысы деп аталады. CASE (Computer Aided Software Engineering) - бағдарламалық жасақтаманы механикалық құру.

Ақпараттық тұжырымдамаларды құру тәсілі тұжырымдаманың әртүрлі нюанстарын сипаттайтын және нақты нақты белгілердің көмегімен жасалынатын түзетулер жасауға арналған әрекеттер жиынтығын береді.

Модификацияда қолданылатын графикалық объектілер жиыны белгілеу деп аталады. Әдіс және басқа бөліктер жиынтығы көрсетілген.

– тұжырымдамалар - бұл жоғары құрылымдалған немесе нысанға бағытталған аспектке ұқсайтын дерексіз негізгі қағидалар;

– динамикалық модификация тұжырымдамасының, сондай-ақ ұсынылған тұжырымдаманың тұрақты текстурасының болуымен қолданылатын жазба; Графикалық диаграммалар әдетте белгілердің қасиеттерінде қолданылады, өйткені олар ең көрінетін және қабылдауға оңай. Мысалы: мәліметтер ағынының диаграммалары, «мәндер коэффициенті» графигі - альтернативті диаграммалар жоғары құрылымдалған келісімнің болуын толықтырады, бірақ 0-жолды орналасудың болуы - класс графигі;

– әдісті нақты қолдануды анықтау тәртібі: сондай-ақ модификацияның принциптеріне, жалпы ұпайлардың болуына қатысты аспектілерге басымдық беріледі.

Жобалау әдістемесі ағымдағы цикл әрекеттерін, сондай-ақ олардың типтік әдістері мен жабдықтарын жүзеге асыруды қамтамасыз ететін белгілі бір технологияларды қолдана отырып жүзеге асырылады.

Жобалау әдістемесі келесі бөліктер жиынтығымен анықталады:

- ғылыми-техникалық жобалаудың реттілігін сипаттайтын бірізді операциялар;
- аспектілер - бұл ғылыми-техникалық қызметтің нәтижелерін бағалауда қолданылатын принциптер;
- зерттелген тұжырымдаманы бейнелеу үшін қолданылатын графикалық және мәтіндік құрылғылар;
- ғылыми-техникалық басшылық, технологиялық процестердің негізгі сипатының элементтері, орындалатын операцияларға қатысты ғылыми-техникалық операцияларды жүргізудің реттілігі, мән-жайлары мен әдістерінің иллюстрациясын қамтиды.

Ақпараттық тұжырымдамаларды жобалау, зерттеу және қолдау әдістемесі келесі шарттарға сәйкес келуі керек:

- әдістеме ақпараттық тұжырымдаманың ағымдағы бағытын толығымен сақтауға міндетті;
- әдістеме көрсетілген сапаға сәйкес ақпараттық ұғымдарды қалыптастырғаннан гөрі, нәтижеге кепілдік беруге міндетті;
- технологияның үлкен жоспарлары мен ішкі жүйелерін жүзеге асыруға міндетті. Қаріп жоспардың ыдырау ықтималдығына кепілдік береді;
- жеке ішкі жүйелерді жобалау мақсатында шағын топтардың (3-7 адам) техникалық енгізілуін қамтамасыз ету;
- әдістеме толыққанды ақпараттық тұжырымдаманы уақытша жеткізуге кепілдік беруге міндетті;
- ғылыми-техникалық жоспардың конфигурациясын, жоспарланған құжаттаманың автоматтандырылған өндірісін және оның нұсқаларын синхрондауды басқаруды қамтамасыз етеді;
- тұжырымдаманың негізгі қайнар көздерінің әдістемесі
- технологиялық процестердің ағымдағы циклінің барлық кезеңдерінде орындалатын әрекеттерді автоматтандыру үйлестірілген CASE құрылғылар жиынтығының көмегімен жүзеге асырылуы тиіс.

### **1.1 Тақырып таңдаудың негіздемесі**

Ақпараттық жүйе - бұл пайдаланушылардың компьютерлік және коммуникациялық құрылғыларға, бағдарламалық жасақтамаға, мердігер ұйымдарға және ақпараттық ресурстарға деген ақпараттық қажеттіліктерінің жиынтығы. Менің дипломдық жұмыс барысында таңдап алып отырған пәндік облысым – «Кітапхана».

«Кітапхана» нақты саласындағы ақпараттық тұжырымдамалардың негізгі міндеттері - кез-келген жағдайда оқырманға, кітапхана мекемелерінің

әкімшілігіне және ақпараттық қоғамға ақпараттық қолдау көрсететін мүмкіндігінше пайдалы базаны сақтау тұжырымдамасын қамтамасыз ету.

Кітапхананың ақпараттық тұжырымдамасының қалыптасуы осындай мәселелерді шешуде үнемі мәліметтер көлемін беру қажеттілігімен, сондай-ақ кітаптар санының артуымен, стандартты емес қорды жетілдірумен, электронды кітаптардың сипаттамаларын арттырумен байланысты.

Кітапхана қызметінің нақты саласында ақпараттық технологияларды пайдалану негізінен уақытты үнемдеуге мүмкіндік береді. Оқырмандарды электронды тіркеу кітапхананың әр қызметкеріне оқырман туралы бірден ақпарат алуға мүмкіндік береді, болашақта электронды кітапханасының ресурстарын қолданған жөн. Сонымен қатар, электронды кітапхана қызметін кез-келген жерден, қашықтықтан алуға мүмкіндік береді.

«Кітап - алтын қазына» - бұл онлайн қызмет көрсететін кітапхана желісі. Менің дипломдық жұмыс барысында «Кітап-алтын қазына» деп аталатын кітапхана ісінің автоматтандырылған ақпараттық тұжырымдамасын жасау қарастырылады. Бұл кітап қоймасы толығымен автоматтандырылмаған, бұл оның қызметкерлер тұрғысынан тиімсіз, сонымен қатар жұмыс қарқындылығын төмендетеді.

«Кітап - алтын қазына» - бұл бірегей онлайн кітап қоймасы, онда әрбір пайдаланушы кітаптардың сапалы әрі оңай пайдалана алады. «Кітап - алтын қазына» кітапхана қызметі пайдаланушылардың қызығушылығын, мемлекеттің мәдениетін, халықтың білімін арттыруға ықпал етеді.

Кітапхана қызметінің негізі келесідей:

- Қазақстанды заманауи электронды кітапхана жүйелерімен қамтамасыздандыруға өз үлесін қосу;
- оқырман талаптарына сай, әрбір сұраныс иесі үшін сенімді жеткізуші көзі болуы;
- оқырман мұқтаждығын қанағаттандыру;
- онлайн кітапханалар технологиясын және оларды басқарудың жаңа, өзгеше форматын енгізу;
- әрбір қызметкердің әлеуметтік шарттарын жақсарту;
- оқырманға электронды кітапханаға оңай тіркелу мүмкіндігі;
- оқырман қызығушылығын арттыру;
- оқырманның сенімінен шығу;
- ұсыныстар мен шағымдарға көңіл бөліп, олармен жұмыс жасау;
- бәсекелестермен кәсіби және этикалық қағидаларды сақтау.

Кітапхананың негізгі міндеттері мен құндылықтары – әрбір оқырманның көңілінен шығуға тырысып, әлемдік сапа стандартына сәйкес қамтамасыз етіп, Қазақстандағы қарапайым және электронды кітапхана жұмыстарын дамыту болып табылады.

Осы жұмыс барысында «Кітапхана» пәндік бағыты бойынша кітапхана жұмысының автоматтандырылған жүйесі жасалды, ол оқырман мен қызметкер арасындағы жұмыс қатынастарын стандарттауға, сонымен қатар қосымшада халыққа негізгі техникалық қызмет көрсетуге мүмкіндік береді. Сол сияқты,

компьютерлік қызметке қол жетімділік орнатылуы керек, операциялар ұғымы дыбысты өшіру режимінде де жасалуы керек, сонымен қатар жасалған тұжырымдамадағы кез-келген процесс компьютермен тез жазылуы керек, ал кітапханадағы соңғы кітаптар уақытында кітаптар базасына көшірілуі керек. Осы себепті бұл жоба оқырманға сапалы қызмет көрсету үшін үлкен маңызға ие. Бұл жағдайда қарастырылатын негізгі нәрселердің бірі-іс-әрекеттің дұрыс тұжырымдамасын қамтамасыз ету үшін жақсы ойластырылған, қалыптасқан, тіпті нақты қатынасы, жоғары құрылымдалған тұжырымдамасы болуы керек.

«Кітапхана» пәндік облысында ақпараттық технологияларды қолданудың мүмкіншіліктері:

- қазіргі заманғы кітапханалардың алдында тұрған, яғни, жоғарыда атап айтылған проблемаларға сүйене отырып, кітапханадағы ақпараттық технологияларды оңтайлы және ұтымды есебін жүргізу;
- кітапхананың және ондағы оқырмандардың санын, сапасын, жай-күйін қашықтан боқылау жүргізу;
- оқырманның толық тарихын сақтау;
- оқырман тарихына қарап, қазіргі таңдағы өзекті әдеби шығармаларды, үздік кітаптар тізімін анықтауға мүмкіндік алу;
- кітапханаларға жаңа электронды зерттеулер, әзірлемелер, және технологиялар туралы ақпарат алу.

Сандық технологиялық процестер кітапхана қызметінің барлық кезеңдерінде - пайдалы құжаттаманы аяқтаудан және жүргізуден бастап кітапхана капиталын жобалауға дейін, сондай-ақ тиісті өңдеу жобасына дейін және т.б. Кітапхана ісі саласындағы ішкі ұсыныстарды әкімшілендіруді автоматтандыру оқырмандармен қарым-қатынас жасау мақсатында кітапхана саласын қалыптастыру шарттарының бірі болып саналады, сондай-ақ ішкі іс-әрекеттерде ақпараттық технологияларды қолданудың негізі болып табылады. Интернеттегі кітаптар санының сапасы оффлайн кітапханаларының сапасынан артық болса, сонымен қатар онлайн құралдарымен айтарлықтай бәсекелестік туындайды, егер бұл механизм кітаптармен жұмыс жасаудағы ақпараттық технологиялық процестерде жойылмаса, бұл бөліктер жойылғанға дейін тиімді бола алмайды.

## **1.2 Программалық қамтаманы жобалаудың методологиялары**

Бағдарламалық жасақтаманы жоспарлау - бұл бағдарламалық жасақтаманың жоспарын құру процесі, сонымен қатар жобалау әдістерін зерттейтін ғылым. Сәйкес жоспарлау өнімнің дизайны мен іс-әрекетінің жеке жағдайы болып саналады.

Бағдарламалық жасақтама ұсынысының мақсаты - тұжырымдаманың ішкі сапаларын анықтау және оның (проблеманың бастапқы талабы) сәйкес тапсырыс беруші жасаған жағдайлар негізінде оның сыртқы (айқын) қасиеттерін егжей-тегжейлі пысықтау. Бұл жағдайлар бақылауға жатады.

Жоспарлау келесі негізгі жұмыс түрлерін қамтиды:

- шешім әдісі мен стратегиясын таңдау;
- ішкі ақпарат тұжырымдамасын таңдау;
- негізгі әдісті құру;
- бағдарламалық қамтаманы құжаттау;
- тестілеу - бұл зерттеуді таңдау;
- кіріс ақпаратының тұжырымдамасын таңдау.

Бастапқыда жоба қара қорап ретінде көрінеді. Жобалау процесінің процесі, сонымен қатар оның нәтижелері тек шарттардың құрамына ғана емес, сонымен қатар таңдалған курстың өзгеруіне, дизайнердің шеберлігіне байланысты. Ағымдағы сфераның нысаны бизнес логикасын және ақпарат құрылымын шектейді.

Жобалау процесіне сәйкес құрылған сыныпқа байланысты жобалау процесіне, сондай-ақ «қолмен» жобалауға, яғни оны автоматтандырудың әртүрлі құралдарына кепілдік беруге болады. Жобалау кезінде оның мәліметтерін - бастапқы диаграмма, ER-диаграммалар, UML-диаграммалар, DFD-диаграммаларды тұжырымдау мақсатында әртүрлі белгілер, сонымен қатар макеттер қолданылады.

Жобалау әдетте:

- сәйкес құрылым;
- жабдықтың бөлшектері сәйкес;
- пайдаланушы интерфейстері.

Қазіргі таңда практикада жоспарлау біртіндеп реттелетін кезеңдерге сәйкес жүзеге асырылады:

- өндірістік жаттығулар;
- өнеркәсіптік ұсыныстар;
- шамамен жоспар;
- өндірістік жоспар;
- жұмыс жоспары.

Әр кезеңде жарналардың жеке жиынтығы жасалады, оны жоспар деп атайды (жоспарланған құжаттама). Әртүрлі тәжірибеде, мысалы, нормативтік құжаттар бағдарламалық жасақтама құжаты болып саналады.

### **1.2.1 Объектілі бағытталған тәсіл**

UML (бірыңғай модельдеу тілі) - берілген нысандарды модельдеуге арналған оңтайландырылған болжаушы тіл.

UML - бағдарламалық жасақтама, үйлестіру, қаржылық, өндірістік және әр түрлі тұжырымдамаларды құру, ойлау, жобалау және құжаттау үшін айтылатын тіл. UML - бұл әр түрлі белгілердің кең таралған схемаларының жиыны.

UML типтік модификацияның келесі диаграммалар жиынтығын ұсынады:

- қолдану баламаларының схемалары (Use case diagrams) - компанияның кәсіпкерлік қызметін, сондай-ақ тұжырымдамалық тұжырымдамаларды болжау үшін қолданылады;

- сынып диаграммалары (Class diagrams) - тұжырымдамалық кластардың тұрақты құрылымын және олардың арасындағы байланысты болжау мақсатында берілген;

- іс-әрекеттің тұжырымдамалық схемалары (Behavior diagrams).

Бұл диаграммалардан басқа:

- өзара әрекеттесу диаграммалары (interaction diagrams), дәйектілік диаграммалары (sequence diagrams) және серіктес диаграммалар (collaboration diagrams) - объектілер арасындағы ескертулер алмасудың дамуын болжау мақсатында берілген;

- күй диаграммалары (statechart diagrams) - тұтас объектілерді болжау мақсатында, бір үлкеннен екіншісіне ауысудың болуы;

- әр түрлі қосымшаларда тұжырымдаманың жұмысын болжау мақсатында қызметтерді модельдейтін жұмыс сызбалары (activity diagrams);

- іске асыру диаграммасы (implementation diagrams);

- бөлшек диаграммалар (component diagrams) - тұжырымдаманы (ішкі жүйені) бөлік иерархиясын болжау мақсатында;

- орналастыру схемалары (deployment diagrams - тұжырымдаманың физиологиялық архитектурасын болжау үшін берілген.

Сыртқы объект орындайтын құбылыс нәтижесінде тұжырымдама орындалатын операциялардың реттілігін қолдану түрі, бұл жағдайда кейіпкер қатысушының әсері бар. Қосымшаның нұсқасы пайдаланушы арасындағы әдеттегі байланысты тұжырымдама ретінде де білдереді. Қарапайым жағдайда қосымша қатысушының орындағысы келетін функцияларын талқылау кезінде анықталады.

## 1.2.2 Құрылымдық тәсіл

Ақпараттық тұжырымдамаларды жобалау мен қарастырудың жоғары құрылымдық аспектісі құрылғылардың негізгі екі санатында қолданылады, олар тұжырымдаманың көпфункционалды құрылымын, сонымен қатар ақпарат арасындағы байланысты білдіреді.

АЖ-ны зерттеудің құрылымдық орналасуының мәні оның автоматтандырылған функцияларға ыдырауында (бөлінуінде) жатыр: тұжырымдама өз ретімен субфункцияларға бөлінетін көпфункционалды ішкі жүйелерге бөлінеді, олар осылайша әрі қарай проблемаларға бөлінеді. Бөлу процесі белгілі бір операцияларға дейін жалғасады. Бұл автоматтандырылған тұжырымдаманың болуы барлық элементтердің элементтерін қоспағанда, бір-бірімен байланыстыратын бүкіл түсінікті сақтайды. «Төменнен жоғары» тұжырымдамасын зерттеудің жеке сұрақтардан тұтастық тұжырымдамасына дейін болуы жоғалады, жекелеген бөліктерді ақпараттық түйістіру кезінде қиындықтар туындайды.

Ерекшеліксіз, сүйектерді орналастыру әдіснамалары бірқатар жалпы негіздерге негізделген. Негізгі екі қасиеті сәйкесінше қолданады:

– «бөліп ал және басқар» ережесі - күрделі мәселелерді шешімдерді қабылдау және қабылдау үшін қарапайым, оларды минималды жеке мәселелердің көп санына бөлу арқылы шешу ережесі;

– иерархиялық ереже - күрделі элементтерді иерархиялық ағаш құрылымына кез-келген дәрежеде ең соңғы элементтерді қосумен біріктіру ережесі.

Негізгі екі ережеге баса назар аудару басқа ережелердің соншалықты маңызды болып саналмайтындығын білдірмейді, өйткені олардың әрқайсысына назар аудармау күтпеген нәтижелерге әкелуі мүмкін (бүкіл жоспардың күйреуін қоса). Ол ережелерге келесілер жатады:

– абстракция ережесі - тұжырымдаманың маңызды нюанстарын атап көрсетуден, сондай-ақ зейінді маңызды емес нәрселерден алшақтатудан тұрады;

– формализация ережесі - қиындықты қорытындылай отырып, қатаң әдістемелік туралау қажеттілігінде.

Құрылымдық ерекшеліктер негізгі 2 функциядан, яғни иллюстрациялық функцияларынан, өндірістік құрылыстар функциясынан көрінеді. Тәуліктің командасы сандар санынан танымал кестелердің нақты айнымалыларын көрсетеді

Кестелердің белгілі бір түрі құрылғылардың әр тобына жауап береді. Ең танымалдары:

– DFD (мәліметтер ағынының диаграммалары) - мәліметтер ағынының диаграммалары;

– SADT (құрылымдық талдау және жобалау техникасы) - дұрыс көпфункционалды тізбектердің диаграммасы;

– ERD (Entity-Relationship Diagrams) - қатынас мәндерінің диаграммалары.

АЖ жобалау кезеңінде кестелер кеңейтіледі, орнатылады және бағдарламалық жасақтаманы жеткізу құрылымын бейнелейтін диаграммаларды қамтиды: архитектура, қаңқа схемалары және экранды конфигурациялау схемалары.

Жоғарыда келтірілген кестелер сіз сүйетін блоктың қайсысы қол жетімді немесе жаңадан жасалғанына қарамастан, абсолютті IP ұсынуын қамтамасыз етеді. Диаграммалардың құрылымы әрбір нақты жағдайда тұжырымдаманың қажетті күрделілігіне байланысты.

Мәліметтер ағынының диаграммалары CASE құралдарында қолданылатын ең танымал түзету түрлері болып саналады.

Ағындық диаграммалардың негізгі элементтері:

- сыртқы мәндер;
- тұжырымдамалар мен ішкі жүйелер;
- бағыттылық;
- деректер жинаушылар;
- ақпарат ағындары.



Сыртқы көрініс объектісі - деректердің негізі немесе алушысы болып саналатын берілген қаржылық объект немесе жеке тұлға. Мысалы, сатып алушылар, жұмысшылар, жеткізушілер, тұтынушылар, қоймалар. Затты немесе тұжырымдаманы құру, сондай-ақ оның сыртқы маңыздылығы, бұл жағдайда олардың қарастырылып отырған ұғымның шекарасымен берілгендігін көрсетеді. Қарастыру барысында белгілі бір сыртқы мәндердің қарастырылып отырған тұжырымдаманың сызбасына еруге барлық мүмкіндігі бар, егер бұл туындайтын болса, және керісінше, белгілі бір қозғалыстар сыртқы мәндер ретінде диаграммадан шығарудың барлық мүмкіндіктеріне ие.

### **1.3 Есептің қойылымы**

Ақпараттық жүйе құру мақсаттарының бірі- мәліметтер моделін жасау, сонымен қатар реляциялық деректер құрылымын құру, нақты өңдеу мүмкіндіктерін жетілдіру. Ағымдағы қолдану саласына сәйкес қайта қаралған интернет-қосымшалардың басқа тұжырымдамаларын талдағаннан кейін, ол оқырман мен кітапханашы арасындағы қарым-қатынасты сақтау үшін ойлап табылды. Сонымен қатар, тағы да бір мақсаты интернет-қосымшада клиент оған қажет кітапты тезірек табуға көмектеседі және оқырманның барлық сұраныстарын тез қанағаттандырады.

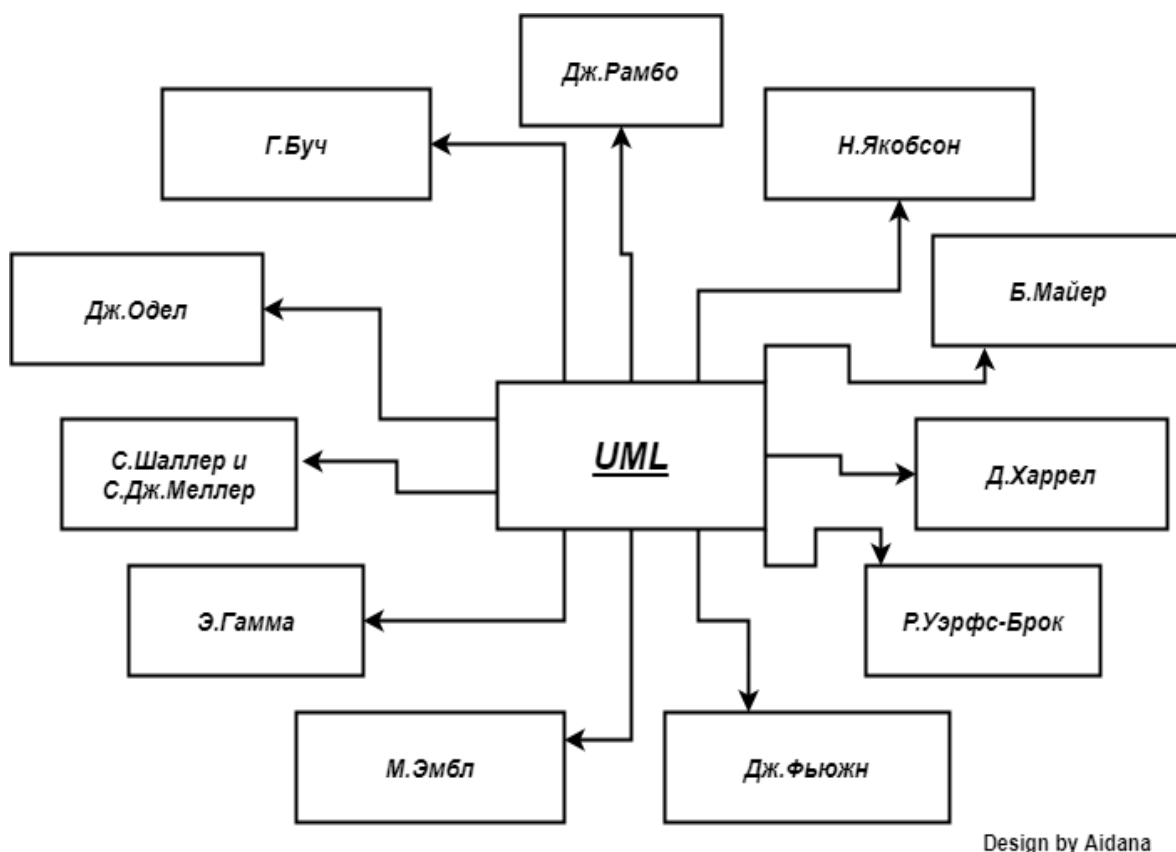
Дипломдық жұмыстың мақсаты - UML тілі көмегімен ақпараттық жүйелерді жобалау болып табылады. Қойылған мақсатқа жету үшін келесі есептер қарастырылуы керек:

- пәндік аймақ бойынша шолу жасау;
- жүйенің объектілі моделін жасау;
- жүйенің іс-әрекеттер, класстар және тізбектер диаграммасын құру;
- жүйенің ER – моделін жасау;
- жасалған модельдерге негізделіп оны іске асыру.

Жүйені құру үшін келесі орталар қолданылуы керек: UML объектілі бағытталған ортасы; Microsoft Visual Studio ортасы; C# бағдарламалау тілі; Microsoft SQL Server Management Studio мәліметтер базасы.

## 2 АҚПАРАТТЫҚ ЖҮЙЕНІҢ ОБЪЕКТІЛІ МОДЕЛІН ЖАСАУ

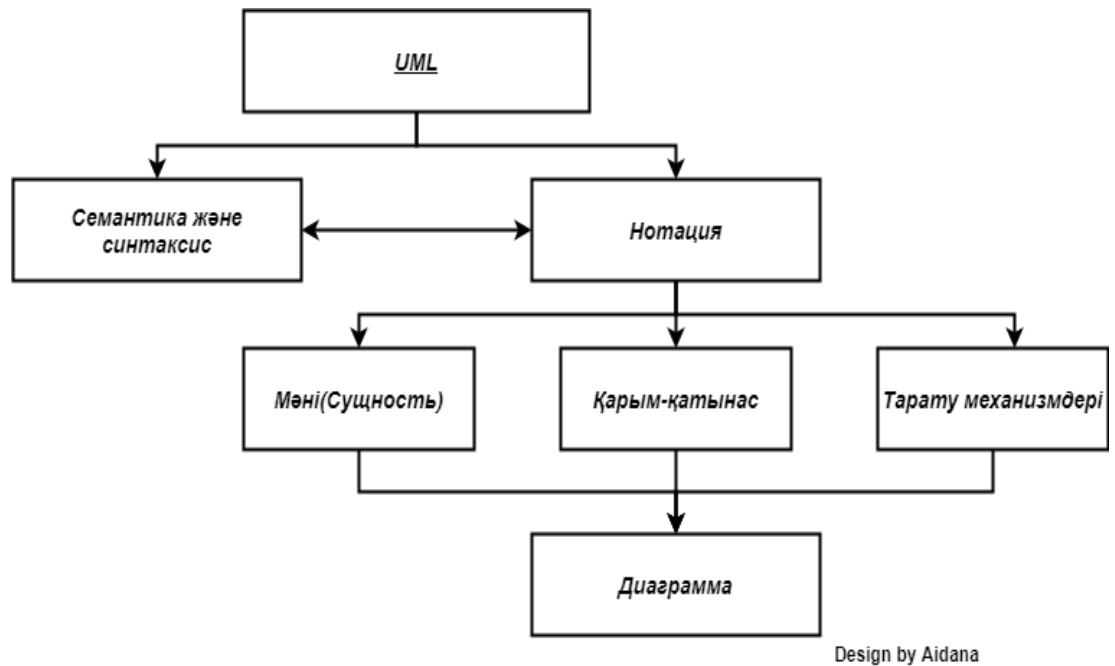
UML (Unified Modeling Language) - бұл унификациялық немесе бірыңғай моделдеу тілі. Яғни, біртекті метомоделдерге негізделген графикалық нотациялар жинытығын айтамыз. UML тілінің негізін қалаушылар Гради Буч, Ивар Якобсон және де Джеймс Рамбо болып табылады(1 сурет) [1, 35-36].



1 сурет – UML тілі авторлары

UML тілі басқа тілдер сияқты – ақпарат ұсыну, сақтау және байланыстыру үшін қолданылатын белгілер жүйесі. UML – бұл ресми жасанды тіл, сондықтан оның спецификациясы жүйесіз болса да келесі міндетті бөлімдерден тұрады(2 сурет):

- сөздік;
- синтаксис (syntax) – тілдік құрылымды құрастыру ережелерінің тізімі;
- семантика (semantic) – тілдік құрылымдарға мағынаны қосу ережелерін анықтау;
- прагматика (pragmatic) – нақты мақсаттарға жету үшін тілдік құрылымдарды қолдану ережелерін анықтау.



2 сурет – UML-дің жалпы құрылымы

UML-дің ерекшелігі – оның сөздігі графикалық элементтерді құрайды, сондықтан, UML – мәтіндік тіл емес. Әрбір "сөз" - графикалық символ болғандықтан, ол нақты семантикаға сәйкес келеді, сондықтанда бір әзірлеуші жасаған модельде басқарылады, сондай-ақ UML түсіндіретін бағдарламалық құрал болғандықтан, оны анық түсіну мүмкін. UML - де ұсынылған жүйенің моделін автоматты түрде бағдарламалау тіліне аударуға болады (Java, c++. VisualBasic және т. б.) яғни, UML - ді қолдайтын визуалды модельдеудің жақсы құралы болған кезде, ол үш модельді қолдана отырып, сізге бағдарламалық кодты, графикалық, графикалық символды ала аласыз, демек, бұл осы модельге жауап береді. UML модельдерінің көмегімен бағдарламалық кодты алуға болады, бірақ UML бағдарламалау тілі емес – бұл моделдеуге арналған тіл. Ақпараттық жүйелердің бағдарламалық жасақтамасын әзірлеу тәжірибесінде модель белгілі бір тілдің көмегімен жасалған талдау және жобалау кезеңдерінің нәтижелері деп аталады. Модельдерді құру процесі модельдеу деп аталады. Ол жүйелерді талдау және жобалау нәтижелерін ұсынуға арналған, сондықтан да UML тілін – модельдеу тілі деп атайды[2, 57-58].

UML объектіге бағытталған дизайн парадигмасын жүзеге асырады және модельдерді графикалық түрде ұсынады, сондықтан UML – объектіге бағытталған визуалды модельдеу тілі болып табылады. Оның негізгі мақсаты - жүйеге қойылатын талаптарды және қабылданған жобалық шешімдерді көрнекі, дәл, бір мағыналы және толық ұсынуды қамтамасыз ету, бұл:

- тапсырыс берушінің АЖ әзірлеушісімен ынтымақтастығына;
- АЖ әзірлеушілердің әртүрлі топтарының бір-бірімен өнімді өзара іс-қимылын ұйымдастыруға мүмкіндік береді.

UML тіліндегі диаграммалар тізімі. UML тілінде келесідей диаграммалар қолданылады(3 сурет).

Құрылым диаграммалары:

- сыныптар диаграммасы (Class diagram);
- құрауыштар диаграммасы (Component diagram);
- күшейту диаграммасы (Deployment diagram);
- құрылымдық бөліктер диаграммасы (Composite structure diagram);
- объекттер диаграммасы (Object diagram);
- пакеттер диаграммасы (Package diagram).

Мінез-құлық диаграммалары:

- өзара әрекеттесу диаграммасы (Interaction diagram);
- қызмет диаграммасы (Activity diagram);
- прецеденттер диаграммасы (Use case diagram);
- соңғы автоматтар диаграммасы (State Machine diagram);
- кезектіліктер диаграммасы (Sequence diagram);
- байланыстар диаграммасы (Communication diagram);
- өзара әрекеттесулерді шолу диаграммасы (Interaction overview diagram);
- уақыт диаграммасы (Timing diagram).



*Design by Aidana*

3 сурет – UML тіліндегі диаграммалар

UML – өңдеушілері қолданылатын негізгі үш режим бар: эскиз режимі, жобалау режимі және программалау тілі режимі. Ал осы үш режимнің ең негізгісі UML-ді эскиз жасауда пайдалану. Осы режимде жобалаушылар UML-ді жүйенің әртүрлі аспектілері туралы ақпарат алмасу үшін қолднылады. Жобалау режимінде UML-ді тік және кері өңдеуде қолдануға болады. Тік өңдеуде(forward engineering) диаграммалар код жазылғанға дейін салынды, ал кері өңдеу(reverse engineering) диаграммалар нәтижелік код дұрыс түсіну үшін тұрғызылады.

Соңғы режимде толық жабамыз дайын болады. Себебі, бұл режимде программасы жасалатын болады.

UML – тіліндегі диаграммалардың қолданылуы. UML тілінің қазіргі түрі нотациялармен метомоделдерді анықтайды. Нотация моделдерде қолданылатын графикалық элементтердің жиынтығы мен UML моделдеу тілінің синтаксисі болып табылады. Метомоделдеу – бұл тілдің түсініктерін анықтайтын диаграммалар жиыны. UML – диаграмманың 13 ресми түрін сипаттайды. Олардың жіктелуі, қолдану аясы 1 кестеде көрсетілген. Негізінен көп адамдар үшін UML сипаттама ережелер жиынынан тұрады[4, 25-26]. Бірақ UML – үлгіқалпы(графикалық моделдер).

1 кесте – UML диаграмма түрлерінің қолданылуы

№	Диаграмма атауы	Қолданылуы
1	Қызмет	Процедуралық және паралелді әрекеттер
2	Класс	Класстар, қасиеттер мен қатынастар
3	Коммуникация	Объектілер арасындағы өзара қатынас: байланысқа қатты көңіл бөледі
4	Компонент	Компоненттер арасындағы өара қатынас
5	Құрамдық құрылым	Орындалу кезіндегі класс декомпозициясы
6	Күшейткіштер	Артефактарды түйіндерге ауыстыру Артефактарды түйіндерге ашу
7	Өзара әрекеттесулерге шолу	Комбинациялар, кезектілік диаграммалары, қызметтік диаграмма
8	Объектілер	Даналардың конфигурацияларының үлгілері
9	Пакеттер	Компиляция уақытының иерархиялық құрылымы
10	Кезектілік	Объектілер арасындағы қатынас: кезектілікке қойылатын акцент
11	Соңғы автоматтар(Күйлер)	Оқиғаларды объектінің өмір сүруі алдында өзгерту
12	Уақыт	Объектілер арасындағы арақатынас: синхронизацияға ерекше көңіл бөлу
13	Прецеденттер	Қолданушылардың жүйемен қатынасы

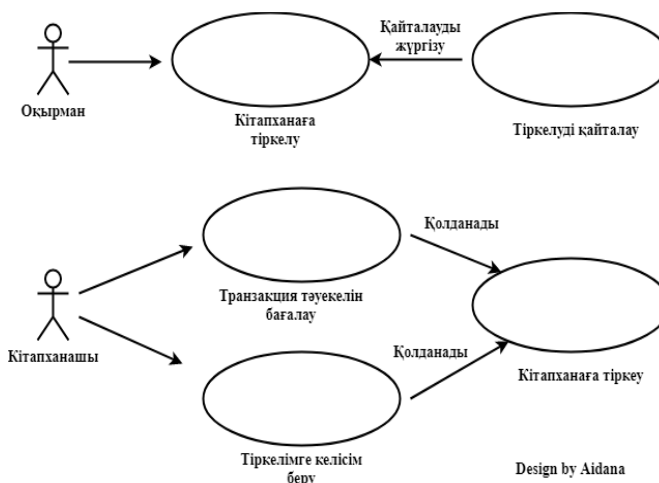
## 2.1 Прецеденттер диаграммасы

UML тілі прецеденттердің құрамы жайлы мағлұмат бермейді, бірақ диаграммалар көмегімен оны көрсетуге болады. Прецеденттер құрастыру кезінде диаграмманы құрастыру үшін көп күш жұмсау керек емес. Оның орнына прецеденттердің мәтінді құрамына назар аударған жөн.

Прецеденттердің диаграммасын ойластыру графикалық кесте көмегімен атқарған жөн. Себебі, онда прецеденттердің құрамын көрсетеді. Ол құрамалы тәсілдерде қолданылатын контекстің диаграммасына ұқсас: себебі ол жүйенің

шекаралары мен оның сыртқы ортамен қарым-қатынасы көрсетіледі. Прецеденттер диаграммасы актерлерді, прецеденттерді және олардың арасындағы байланыстарды көрсетеді:

- қандай актер қай прецедентті орындайтынын көрсетеді;
- қай прецеденттер өз құрамына басқа прецеденттерді қосады.



4 сурет – Прецеденттер диаграммасы

## 2.2 Іс-әрекеттер диаграммасы

Іс-әрекет диаграммасы - бұл жағдай диаграммасының ерекше жағдайы. Іс әрекет диаграммасы басқару ағымының жүйенің ішіндегі бір әрекеттен екіншісіне ауысуын көрсетеді. Диаграмманың бұл түрі әдетте көптеген параллель процестерді қамтитын мінез-құлықты сипаттау үшін қолданылады.

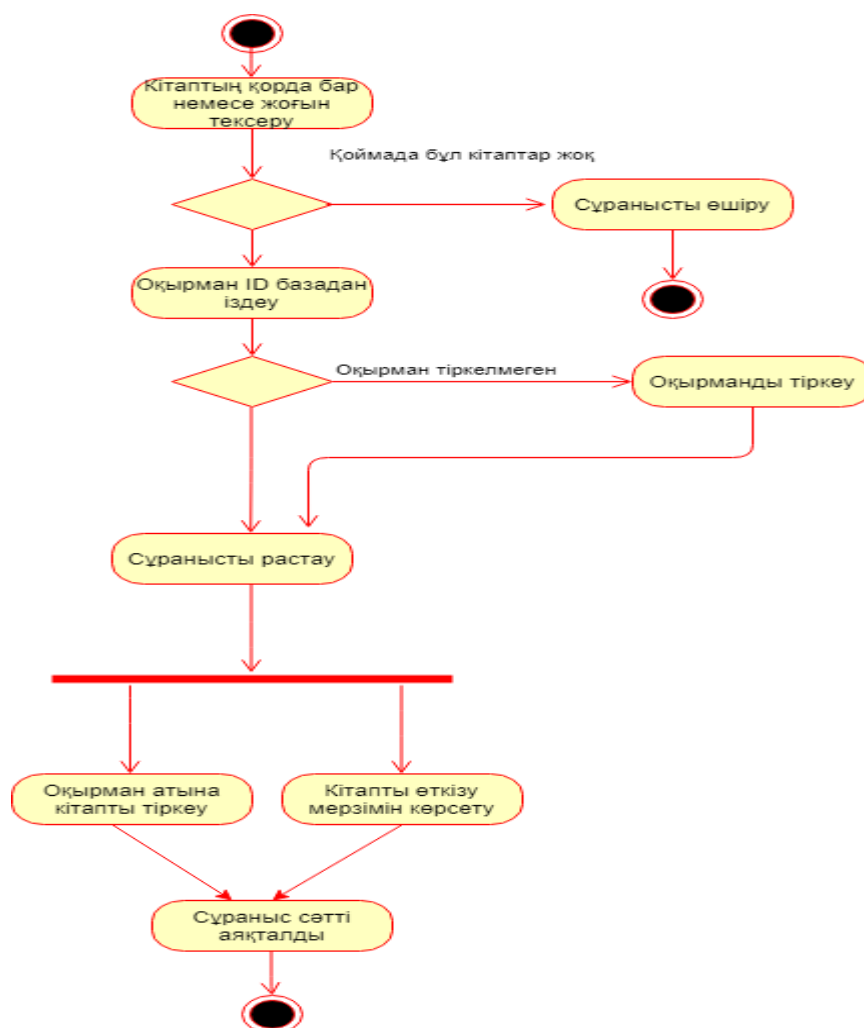
Іс әрекет диаграммаларының негізгі элементтері.

Әрекет диаграммасы процестерді блок-схема түрінде ұсынуға балама (альтернатив) болып табылады және әрекеттер мен таңдаулардың реттілігін сипаттау үшін қолданылады.

Ол келесі элементтерден тұрады:

- процестің басталуы - сипатталған процестің басталуын білдіреді, бағдарламаның немесе жаһандық процестің басталуымен сәйкес келмеуі мүмкін;
- әрекет - алгоритмді орындаудың ағымдағы кезеңіндегі әрекеттердің сипаттамасын қамтиды;
- шешім - диаграммалар блогындағы сияқты, ромбпен белгіленеді, бірақ мәтін жоқ. Тармақталу шарттарының мәтіні шешімнен шығатын басқару ағындарында көрсетіледі;
- басқару ағыны - әрекеттер тізбегін көрсетеді;
- бөлу - тәуелсіз операциялар блогының басталуы;
- қосылу - тәуелсіз операциялар блогының аяқталуы;

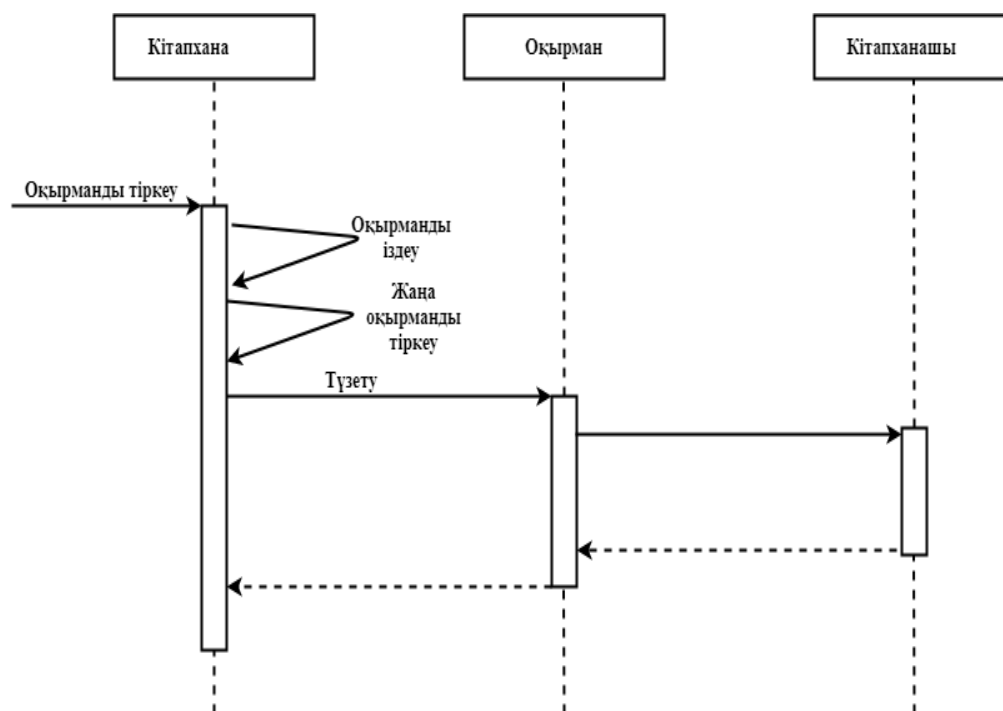
– процестің аяқталуы - сипатталған процестің аяқталуы бағдарламаның немесе жаһандық процестің аяқталуымен сәйкес келмеуі мүмкін.



5 сурет – Іс-әрекеттер диаграммасы

### 2.3 Тізбектер диаграммасы

Диаграмманың бұл түрі қолдану жағдайының сценарийін дәл анықтау үшін қолданылады. Бірізділік диаграммаларында қолдану жағдайымен өзара әрекеттесетін нысандардың түрлері, олар бір-біріне жіберілетін хабарламалар және осы хабарламалармен байланысты кез келген қайтару мәндері көрсетілген. Тік сызықтардағы тіктөртбұрыштар объектінің «қызмет ету мерзімін» көрсетеді. Көрсеткіштер мен әдістердің атаулары жазылған жолдар объектіге шақырудың әдісін көрсетеді(6 сурет).



Desing by Aidana

6 сурет – Тізбектер диаграммасы

## 2.4 Кластар диаграммасы

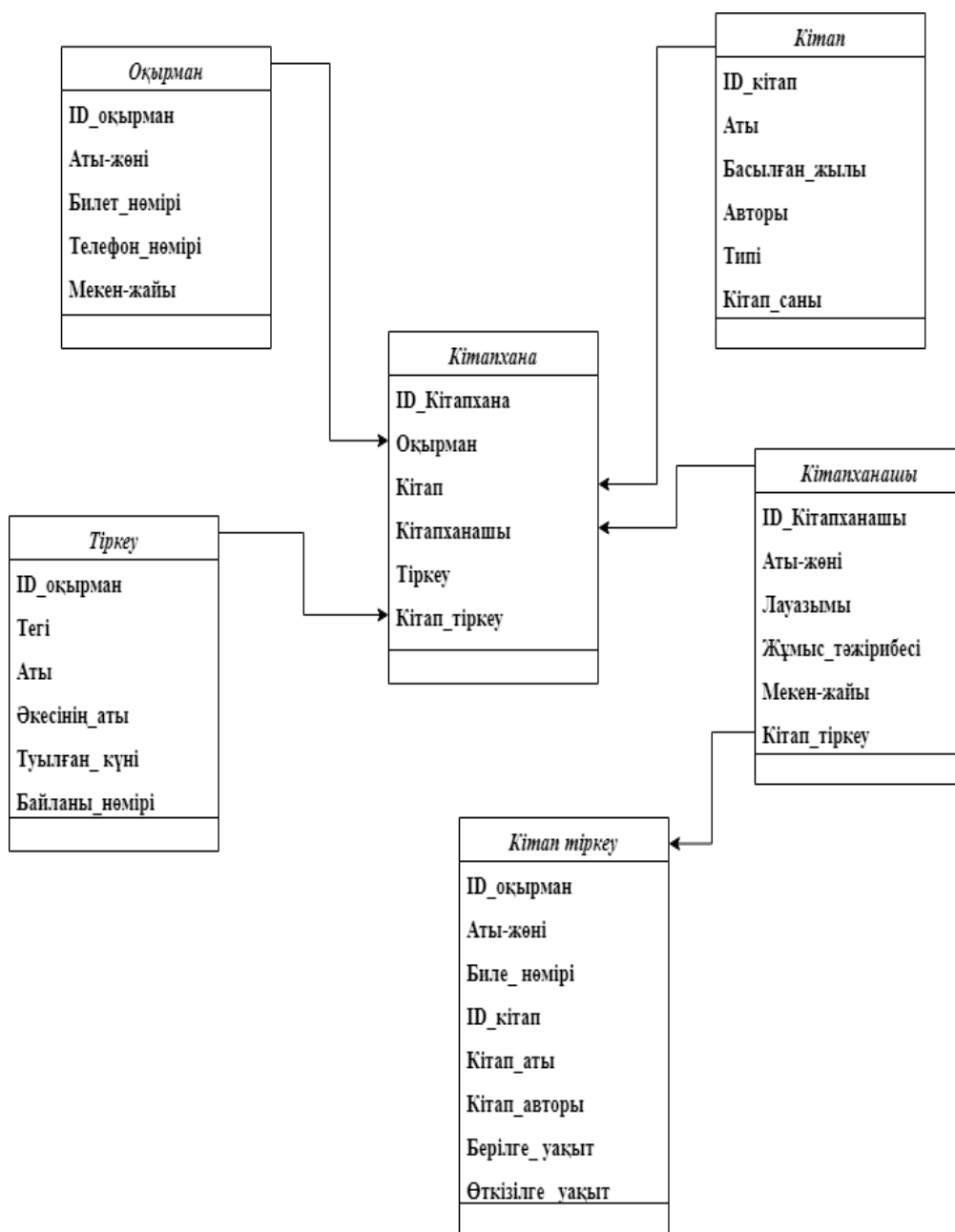
UML-дегі кластар жүйені статикалық күйде сипаттауға мүмкіндік беретін классикалық диаграммаларда бейнеленген - жүйе нысандарының типтерін және олардың арасындағы әртүрлі статикалық байланыстарды анықтауға мүмкіндік береді.

Сыныптар жүйелік нысандардың түрлерін көрсетеді.

- сыныптар арасындағы пайдалану қатынастарын сипаттайтын тәуелділіктер;
- жалпылама сыныптарды мамандандырылған сыныптармен байланыстыру;
- сынып объектілері арасындағы құрылымдық қатынасты көрсететін бірлестіктер.

Кластар диаграммасы келесі суретте көрсетілген:





Design by Aidana

7 сурет – Кластар диаграммасы

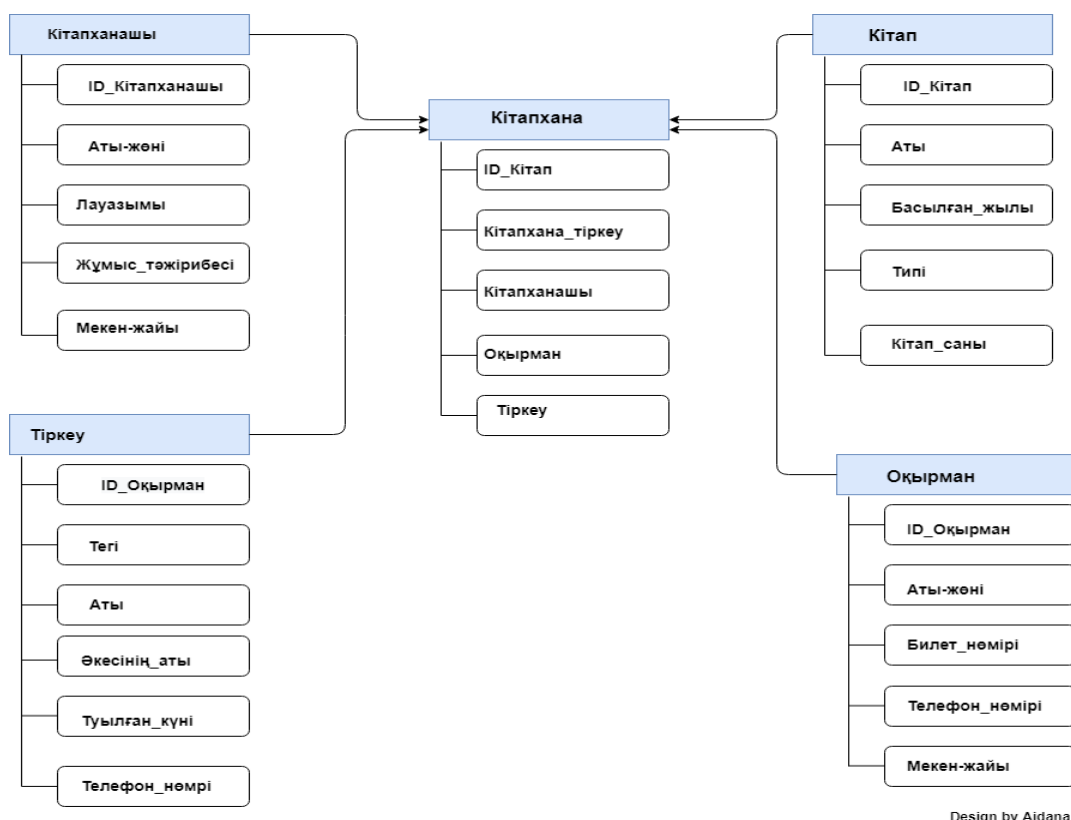
## 2.5 Мәліметтер базасы

MySQL ақпараттарды енгізу мақсатында ақпараттық база ретінде қолданылады. MySQL - бұл жоғарыда аталған функцияларды орындауға арналған барлық маңызды құрылғылардан тұратын жұмыс үстелінің ақпараттық тұжырымдамасы. Негіздер бойынша нұсқаулық MySQL - жұмыс үстеліндегі ДҚБЖ (Деректер қорын басқару жүйесі) тобына ең көп таралған қосымшалардың бірі. MySQL-дің барлық нұсқаларын қоспағанда, ақпараттарды енгізу, ақпараттарды өңдеу сияқты қызметтер атқарады, бұл кесте, кесте және ақпарат түріндегі мәліметтерді таңдау мен түсінуді едәуір жеңілдетеді. Бұл

ақпарат веб-сайттарда, соның ішінде пайдаланушыға MySQL нұсқасынан көрінетін Internet Explorer-де бар.

Сонымен қатар, MySQL оны басқа кесте қызметкерлерінен, сонымен қатар серверлік мәліметтер базасынан қосымша кестелермен толықтыруға қажет деректерді сақтау мақсатында пайдалануға мүмкіндік береді. Ол MySQL кестелеріне ұқсас негіздерімен, сыртқы кестелерге қосымша қатармен жұмыс істейді. Сонымен қатар, басқа пайдаланушыларда бұл жағдайда олар жасаған әрекеттерді жалғастыру үшін осы деректермен жұмыс жасауға барлық мүмкіндіктер бар. MySQL-дің артықшылықтары. Біреуде қарапайым және күрделі керек-жарақтарды зерттеу үшін қарапайым графикалық тартпа бар. Басқа кестеде тағы бір айта кететін жайт, MySQL барлық мәліметтерді 1 файлға ерекше жағдайсыз сақтайды және әртүрлі кестелерге бөлінеді[6, 45-49].

Әртүрлі зерттеулер саласы нақты қоғамның белгілі бір үлесін білдіреді, ол қоғамның мақсатымен зерттеледі, сонымен бірге басқаруды автоматтандырады. Тапсырмалар бөліктер жиынтығында ұсынылған, мысалы, бизнес-семинарлар, басқару, санау және бақылау әртүрлі конфигурациялармен және әрекеттермен сипатталады. Барлық пайдаланушыларда әр түрлі мониторлар проблемасы бар. Дипломдық жұмыста негізгі мақсаттардың бірі ақпараттық мәліметтер базасын құру. Осы себептен оны іске асыру барысында зерттеулер жүргізілді. Зерттеу барысында мәліметтер моделі сызылды(8 сурет)



8 сурет – Мәліметтер моделі диаграммасы

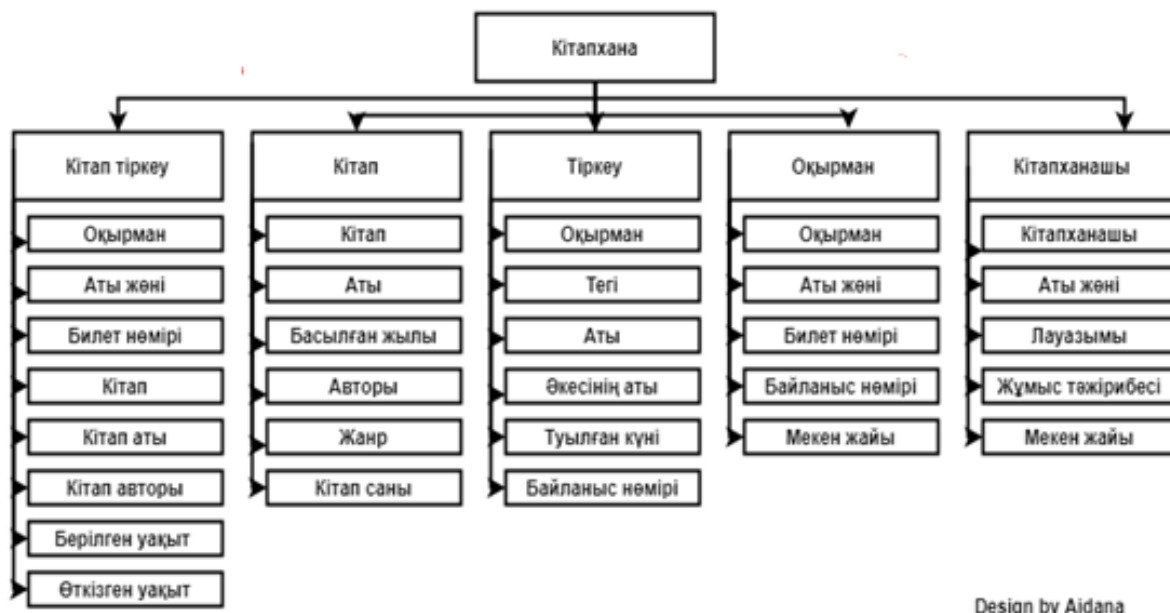
Параметрлерге талап - ақпаратты интерактивті терезеге енгізу талабын, олардың орындалуын көрсетеді. Сіз кейбір ақпаратты жүзеге асыруға сұраныс жасай аласыз.

Сондай-ақ, ол сұранысты зерттеуді жеңілдететін құрылымдық ақпаратты тексеру үшін протоколдық мақсатта қолданылады. Біріктірілген сұраулардың, орташа мағыналардың, мағыналардың санын немесе басқа статистикалық есептеулердің қажетті санын есептейді, содан кейін барлығы екі кестеге топтастырылады, олардың біреуі баған атауларын белгілейді, ал екіншісі - жол аттары.

Өзгерістерді талап дегеніміз - бірнеше жазбалар әсер ету мақсатында өзгертілген немесе өзгертілген маңызды құжатты өзгерту туралы берілген өтініш. Өзгерту туралы сұранысты алып тастау үшін, мәтін журналдарын жаңарту журналына қосады. Үстелді қалыптаудың 4 түрі бар.

ДҚБЖ (Деректер қорын басқару жүйесі) - бұл ақпарат негіздеріне негізделген кейбір қолданушыларды (жобаларды) қалыптастыру, пайдалану және пайдалану мақсатында ақпаратқа негізделген көмекке арналған бағдарламалық жасақтама. ДҚБЖ мүмкіндіктері:

- жаңа ақпараттық базаны құру;
- ақпаратты түрлендіру;
- ақпаратты толтыру;
- ақпаратты іздеу;
- деректер қорындағы құпия ақпарат;
- жаңартылған ақпаратты жазу;
- ақпаратты басып шығару және талаптарды шешу[6, 40-45].

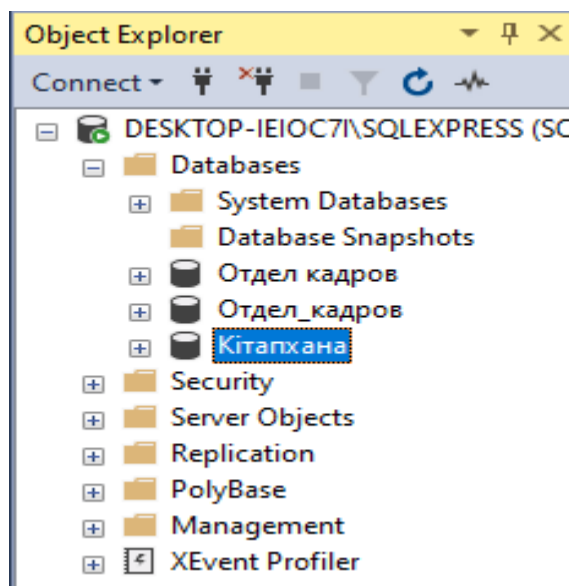


9 сурет – Жобаның мәліметтер базасының құрылымы

Кітапхана мәліметтік базасын құрамыз. Жобалаудың бірінші кезеңінде келесі қатынастар алынды:

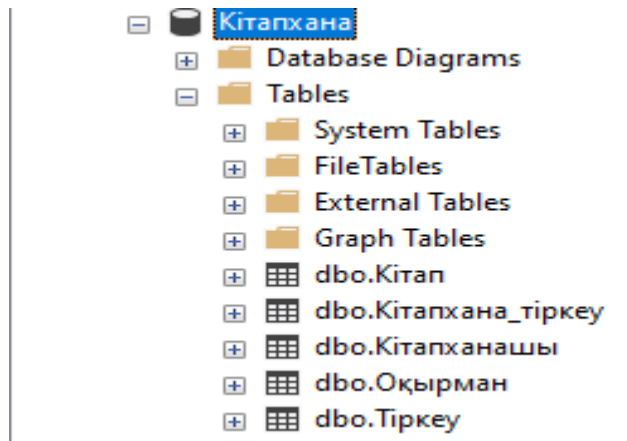
- КІТАПХАНА (ID\_Кітапхана, Оқырман, Кітап, Кітапханашы, Тіркеу, Кітап\_тіркеу );
- КІТАПХАНАШЫ (ID\_Кітапханашы, Аты-жөні, Лауазымы, Жұмыс тәжірбиесі, Мекен-жайы, Кітап\_тіркеу);
- ОҚЫРМАН (ID\_Оқырман, Аты-жөні, Билет нөмірі, Телефон нөмірі, Мекен-жайы);
- КІТАП (ID\_Кітап, Аты, Басылған жылы, Автор, Типі, Кітап саны);
- ТІРКЕУ (ID\_Оқырман, Тегі, Аты, Әуесінің\_аты, Туылған\_күні, Байланыс\_нөмірі);
- Кітап\_тіркеу (ID\_Оқырман, Аты-жөні, Билет\_нөмірі, ID\_Кітап, Кітап\_аты, Кітап\_авторы, Берілген\_уақыт, Өткізілген\_уақыт).

Алдымен SQL – жаңа database құрылады, оны «Кітапхана» деп атаймыз. Оны 10 суреттен көре аламыз.



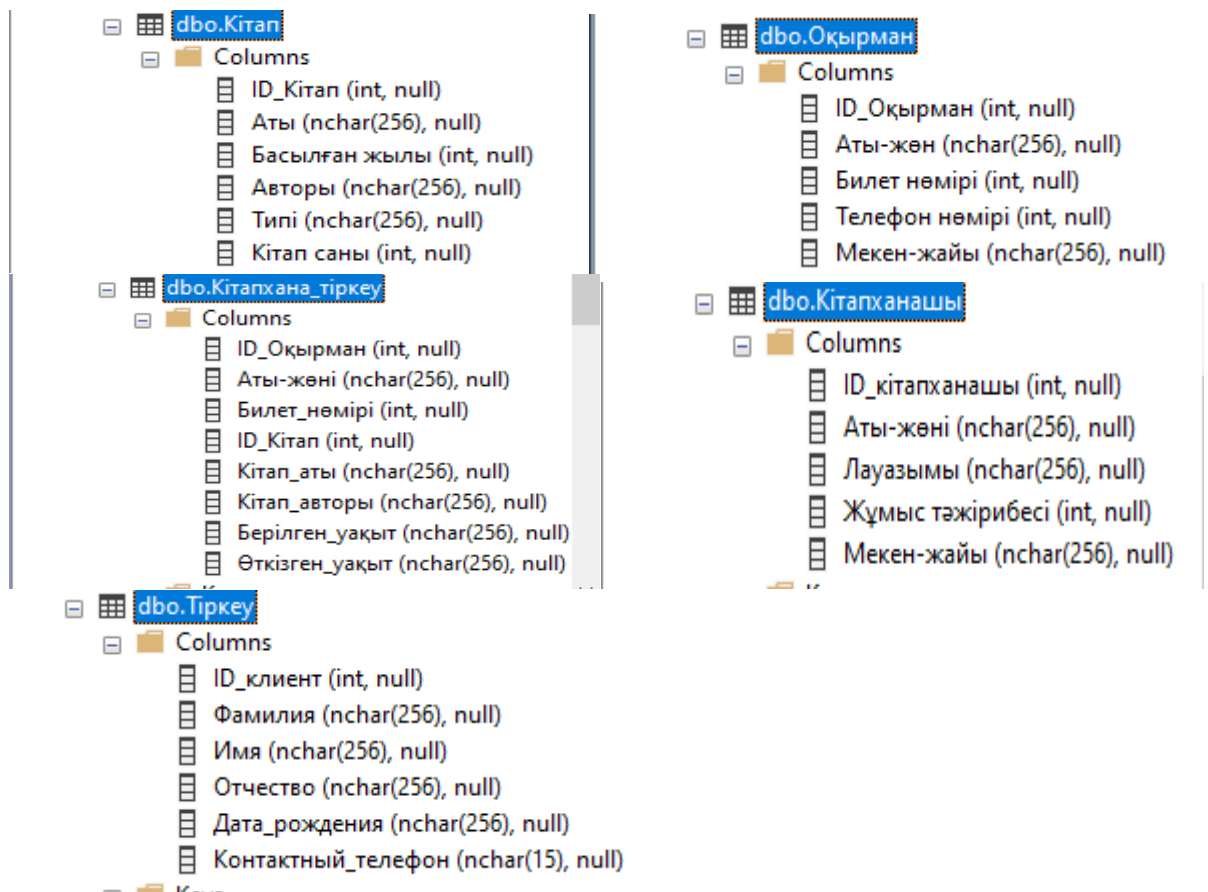
10 сурет – «Кітапхана» базасы

Содан соң осы база ішіне бірнеше Table(11 сурет), яғни кестелер қостым. Оларды: «Кітап», «Кітапхана\_тіркеу», «Кітапханашы», «Оқырман», «Тіркеу» деп атадым.



11 сурет – SQL-де «Кітапхана» кестесі

Әрбір кестеге өзіне тән бағандар қосып, олардың типін көрсеттім.



### 3 ПРОГРАММАНЫ СІПАТТАУ

#### 3.1 Программаның құрылымы мен жұмысын сипаттау

Visual Studio 2019 платформасында C# тілінде «Тіркеу» бетін жасау үшін Label, TextBox, Button элементтерін қолдана отырып “Регистрация” терезесі жасалады. Ол 12 суретте көрсетілген.

The screenshot shows a registration form with the following fields and labels:

- ID: [input field]
- Фамилия: [input field]
- Имя: [input field]
- Отчество: [input field]
- Дата рождения: [input field]
- Контактный телефон: [input field]
- Регистрация: [button]

12 сурет – Оқырманды тіркеу беті

”Регистрация” формасын толтырып, SQL Server-дегі “Клиенты” кестесіне жаңа қолданушыны қосамыз. Қосылған әрбір қолданушы автоматты түрде SQL базасына түседі (13 сурет).

The screenshot shows the registration form after successful registration. The form is filled with sample data:

- ID: 5
- Фамилия: Советкинова
- Имя: Жауқазын
- Отчество: Жайдырмықовна
- Дата рождения: 02.07.1999
- Контактный телефон: 87475144114

A dialog box displays the message: "Вы успешно зарегистрированы!" (You have been successfully registered!).

Below the form is a table with the following columns: IP\_клиент, Фамилия, Имя, Отчество, Дата\_рождения, Контактный.

Buttons: Показать, Следующая страница

13 сурет – Оқырманды тіркеу бетін толтыру

«Регистрация» бетінде толтырылған барлық мәліметтерді DataGridView-де көрсету. Мұнда SQL-ге «Регистрация» кестесінен түскен мәліметтер орналасады. Яғни, DataGridView-ге мәліметтер тікелей SQL мәліметтер базасынан келеді, ол 14 суретте бейнеленген.

**Қытап - асыл Қазына**

ID:

Фамилия:

Имя:

Отчество:

Дата рождения:

Контактный телефон:

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_т...
2	Сағындық ...	Канат ...	Маратович ...	05.02.1989 ...	87472564123...
3	Казиева ...	Айдана ...	Батырбайқызы...	25.05.1999 ...	8702589896...
4	Колбай ...	Акбота ...	Нурланқызы ...	29.12.1999 ...	8707125948...
5	Советханова...	Жауказын ...	Жайдарбекқызы...	02.04.1999 ...	8747514411...

14 сурет – DataGridView беті

«Регистрация» кестесінің “SQL”-дегі көрінісі. Бұл «Регистрация» бетінде толтырылған мәліметтерді сақтайтын SQL бағаны(15 сурет).

DESKTOP-IEIOC7\N...пхана - dbo.Тіркеу

	ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_т...
▶	1	Кайсарова ...	Дина ...	Жанатовна ...	12.02.1995 ...	87073210303
	2	Сағындық ...	Канат ...	Маратович ...	05.02.1989 ...	87472564123
	3	Казиева ...	Айдана ...	Батырбайқызы...	25.05.1999 ...	87025898969
	4	Колбай ...	Акбота ...	Нурланқызы ...	29.12.1999 ...	87071259484
	5	Советханова ...	Жауказын ...	Жайдарбекқызы...	02.04.1999 ...	87475144114
*	NULL	NULL	NULL	NULL	NULL	NULL

15 сурет – Оқырмандар туралы мәліметтер

«Запуск» жасағаннан кейін бізде алғашқы форма беті ашылады. Button элементі арқылы келесі формаға әрекеті орындалады. Екінші форма ашылған сәтте DataGridView элементіне SQL Server – ден “Тіркеу” кестесі жүктелінеді. Формада SQL Server-ден кестені жүктеу үшін DataGridView элементі, сұрыптауды кестенің белгілі бағанын орындау мақсатында ListBox элементі, сұрыптауды өсу немесе кему ретімен жасау үшін RadioButton 2 элементі, іздеу жолы және фильтрация ретінде TextBox элементі қолданылды. Button элементтері арқылы жоғарыда айтылған операциялары жүзеге асырылады (16 сурет).

Form1

### Кітап - асыл қазына

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_телефон
1	Кайсарова ...	Дина ...	Жанатовна ...	12.02.1995 ...	87073210303
2	Сағындық ...	Канат ...	Маратович ...	05.02.1989 ...	87472564123
3	Казиева ...	Айдана ...	Батырбайқызы ...	25.05.1999 ...	87025898969
4	Колбай ...	Ақбота ...	Нурланқызы ...	29.12.1999 ...	87071259484

**Критерий:**

**ФИО:**

**Сортировка**

**Поле для сортировки:**

ID\_клиент  
Фамилия  
Имя

Сортировка по возрастаню  
 Сортировка по убыванию

16 сурет – Фильтрлеу терезесі

«Сортировка» батырмасын қолдану. Сұрыптар 17, 18 суретте көрсетілгендей «ID\_клиент», «Фамилия», «Имя», «Отчество», «Дата\_рождения», «Контактный\_телефон» бойынша өсу және кему ретінде сұрыпталады. Типі string болған кезде алфавит бойынша жасалады, ал типі int болған кезде сандардың өсу және кему реті бойынша жасалады.

Кітап - асыл қазына

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_телефон
2	Сағындық ...	Канат ...	Маратович ...	05.02.1989 ...	87472564123
1	Кайсарова ...	Дина ...	Жанатовна ...	12.02.1995 ...	87073210303
3	Казиева ...	Айдана ...	Батырбайқызы ...	25.05.1999 ...	87025898969
4	Колбай ...	Ақбота ...	Нурланқызы ...	29.12.1999 ...	87071259484

**Критерий:**

**ФИО:**

**Сортировка**

**Поле для сортировки:**

Отчество  
Дата\_рождения  
Контактный\_номер

Сортировка по возрастаню  
 Сортировка по убыванию

17 сурет – Сұрыптау терезесі



***Кітап - асыл қазына***

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_телефон
2	Сағындық	Канат	Маратович	05.02.1989	87472564123
1	Кайсарова	Дина	Жанатовна	12.02.1995	87073210303
4	Колбай	Ақбота	Нурланқызы	29.12.1999	87071259484
3	Казиева	Айдана	Батырбайқызы	25.05.1999	87025898969

**Критерий:**

**ФИО:**

**Сортировка**

**Поле для сортировки:**

Сортировка по возрастаню  
 Сортировка по убыванию

18 сурет – Сұрыптау мысалы

«Фильтр» батырмасын қолдану. Бұл команда «ID\_клиент», «Фамилия», «Имя», «Отчество», «Дата\_рождения», «Контактный\_телефон» бойынша жасалады.

***Кітап - асыл қазына***

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_телефон
2	Сағындық	Канат	Маратович	05.02.1989	87472564123

**Критерий:**

**ФИО:**

**Сортировка**

**Поле для сортировки:**

Сортировка по возрастаню  
 Сортировка по убыванию

19 сурет – Фильтрлеу мысалы

«Поиск» батырмасын жасау үшін RichTextBox және Button пайдаланылды. Бұл операция кестедегі барлық бағандар бойынша жасалады. 20 суретте «Поиск» батырмасының қолданылуы көрсетілген.

***Кітап - асыл қазына***

ID_клиент	Фамилия	Имя	Отчество	Дата_рождения	Контактный_телефон
1	Кайсарова ...	Дина ...	Жанатовна ...	12.02.1995 ...	87073210303
3	Казиева ...	Айдана ...	Батырбайқызы ...	25.05.1999 ...	87025898969
*					

**Критерий:**

**ФИО:**

**Сортировка**

**Поле для сортировки:**

ID\_клиент  
Фамилия  
Имя

Сортировка по возрастианию  
 Сортировка по убыванию

20 сурет – Мәліметтерді іздеу

21 суретте Visual Basic ортасында Microsoft RDLC Report арқылы деректер қорындағы барлық енгізілген оқырманды шығару үшін «Отчет» яғни, есеп беру командасын жасаймыз. Шығарылған есеп беруді дайын excel форматында шығаруға немесе консольде көруге болады.

Отчет таблицы "Тіркеу"	
ID клиента	1
ФИО сотрудника	Казиева Айдана
Дата рождения	25.05.1999
Пол	Ж
Номер телефона	+77759395954
Адрес сотрудника	Коксай-5394
ID клиента	2
ФИО сотрудника	Колбай Акбота

ID клиента		2
ФИО сотрудника	Колбай Акбота	
Дата рождения	24.09.1989	
Пол	Ж	
Номер телефона	+77478675645	
Адрес сотрудника	Саина 79	
ID клиента		3
ФИО сотрудника	Советханова Жауказын	
Дата рождения	01.05.1999	
Пол	Ж	
Номер телефона	+77756784534	
Адрес сотрудника	Таугуль-23а	

21 сурет – Есеп беру мысалы

## ҚОРЫТЫНДЫ

Дипломдық жұмыста UML тілі көмегімен ақпараттық жүйелерді жобалау сұрақтары қарастырылған. Оны орындау барысында келесі нәтижелер алынды:

- пәндік аймақ бойынша шолу жасалды;
- жүйенің объектілі моделін жасауда қолданылатын диаграммалар таңдалды;
- жүйенің іс-әрекеттер диаграммасы жасалды;
- кластар диаграммасы жасалды;
- тізбектер диаграммасы құрылды;
- мәліметтер базасы жобаланды;
- мәліметтер моделі жасалды;
- жасалған модельдерге негізделіп программалық қамтама іске асырылды.

Жасалған жүйенің көмегімен кітаптар туралы мәліметтер базасын жүргізуге, кітаптар мен оқырмандарды тіркеуге, мәліметтерді филтрлеу мен сұрыптауға, есеп беру құжаттарын шығаруға мүмкіндік береді.

Жүйені құру үшін келесі орталар қолданылды: UML объектілі бағытталған ортасы; Microsoft Visual Studio ортасы; C# бағдарламалау тілі; Microsoft SQL Server Management Studio.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 В.В.Белов, В.И.Чистякова «Проектирование информационных систем»
- 2 Ш.Ә.Джомартова, Д.Р.Рахимова, Ә.Т.Тұрарбек «UML тілінің негіздері (оқу құралы)»
- 3 Г.С.Мухашева, И.И.Томилова, Б.О.Мухамеджанова, А.Н.Томилов «UML практикумы»
- 4 Брежнева, В.В. Информационное обслуживание: продукты и услуги, предоставляемые библиотеками и службами информации предприятий
- 5 Холзнер С. Программирование на C#. Учебный курс
- 6 Кузин А.В. Базы данных: Учебное пособие для студ.
- 7 Байжұманов М.Қ., Жапсарбаева Л.Қ. Информатика
- 8 Кречетников, К.Г. Рекомендации по проектированию мотивационной и содержательной составляющих образовательной среды вуза на основе информационных технологий [Электронный ресурс]
- 9 <https://socscan.ru/kk/vytyazhki/vidy-diagramm-uml-diagrammy-uml-uml-diagrammy.html>
- 10 Мацящек Л.А. «Анализ и проектирование информационных систем с помощью UML 2.0»
- 11 Пирогов В.Ю. «Информационные системы и базы данных: организация и проектирование (учебное пособие)»
- 12 Уэнди Боггс, Майкл Боггс «UML и Rational Rose»
- 13 Ф.А.Новиков, Д.Ю.Иванов «Моделирование на UML. Теория и практика»
- 14 <https://intuit.ru/studies/courses/2195/55/info>
- 15 [https://ru.wikipedia.org/wiki/Проектирование\\_программного\\_обеспечения](https://ru.wikipedia.org/wiki/Проектирование_программного_обеспечения)

## 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;
using System.Data.SqlClient;
using System.Data.Common;

namespace WindowsFormsApp3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
```

```

    }
    //connection
    SqlConnection myConnection = new SqlConnection(@"Data
Source=DESKTOP-IEIOC7I\SQLEXPRESS;Initial Catalog=Кітапхана;Integrated
Security=True");
    private DataTable SearchData(string valueToFind)
    {

        DataTable table = new DataTable();
        SqlCommand command = myConnection.CreateCommand();
        command.CommandText = @"SELECT * FROM Тіркеу Where Фамилия
like '%" + valueToFind + @"'%"";
        command.CommandType = CommandType.Text;
        DataSet data = new DataSet();
        SqlDataAdapter adapter = new SqlDataAdapter(command);
        adapter.Fill(data);
        table = data.Tables[0];
        return table;

    }
    private DataTable DoubleSearchData(string valueToFind2, string
valueToFind3)
    {
        DataTable table = new DataTable();
        SqlCommand command = myConnection.CreateCommand();

```

```

        command.CommandText = @"SELECT * FROM Tipkey Where Фамилия
like '%" + valueToFind2 + @"%' and Имя like '%" + valueToFind3 + @"%'";
        command.CommandType = CommandType.Text;
        DataSet data = new DataSet();
        SqlDataAdapter adapter = new SqlDataAdapter(command);
        adapter.Fill(data);
        table = data.Tables[0];
        return table;
    }
    private DataTable ViewData()
    {
        DataTable table = new DataTable();
        SqlCommand command = myConnection.CreateCommand();
        command.CommandText = @"SELECT * FROM Tipkey ORDER BY
ID_КЛИЕНТ";
        command.CommandType = CommandType.Text;
        DataSet data = new DataSet();
        SqlDataAdapter adapter = new SqlDataAdapter(command);
        adapter.Fill(data);
        table = data.Tables[0];
        return table;
    }
    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void label3_Click(object sender, EventArgs e)
    {

```



```
}
```

```
private void linkLabel1_LinkClicked(object sender,  
LinkLabelLinkClickedEventArgs e)
```

```
{
```

```
}
```

```
private void label4_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```
    dataGridView1.DataSource = DoubleSearchData(" ", " ");
```

```
}
```

```
private void button4_Click(object sender, EventArgs e)
```

```
{
```

```
    dataGridView1.DataSource = SearchData(" ");
```

```
}
```

```
private void button5_Click(object sender, EventArgs e)
```

```
{
```

```

Form2 f2 = new Form2();
this.Hide();
f2.Show();
}

private void button3_Click(object sender, EventArgs e)
{
    dataGridView1.DataSource = ViewData();
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    string text = richTextBox1.Text;
    dataGridView1.DataSource = SearchData(text);
}

private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked)
    {
        if (listBox1.Text == "ID_КЛИЕНТ")
        {
            dataGridView1.Sort(dataGridView1.Columns[0],
ListSortDirection.Ascending);
        }
        else if (listBox1.Text == "Фамилия")
        {

```

```

        dataGridView1.Sort(dataGridView1.Columns[1],
ListSortDirection.Ascending);
    }
    else if (listBox1.Text == "Имя")
    {
        dataGridView1.Sort(dataGridView1.Columns[2],
ListSortDirection.Ascending);
    }
    else if (listBox1.Text == "Отчество")
    {
        dataGridView1.Sort(dataGridView1.Columns[3],
ListSortDirection.Ascending);
    }
    else if (listBox1.Text == "Дата_рождения")
    {
        dataGridView1.Sort(dataGridView1.Columns[4],
ListSortDirection.Ascending);
    }
    else if (listBox1.Text == "Контактный_номер")
    {
        dataGridView1.Sort(dataGridView1.Columns[5],
ListSortDirection.Ascending);
    }
}
else if ((radioButton2.Checked))
{
    if (listBox1.Text == "ID_клиент")
    {

```

```

        dataGridView1.Sort(dataGridView1.Columns[0],
ListSortDirection.Descending);
    }
    else if (listBox1.Text == "ФАМИЛИЯ")
    {
        dataGridView1.Sort(dataGridView1.Columns[1],
ListSortDirection.Descending);
    }
    else if (listBox1.Text == "Имя")
    {
        dataGridView1.Sort(dataGridView1.Columns[2],
ListSortDirection.Descending);
    }
    else if (listBox1.Text == "Отчество")
    {
        dataGridView1.Sort(dataGridView1.Columns[3],
ListSortDirection.Descending);
    }
    else if (listBox1.Text == "Дата_рождения")
    {
        dataGridView1.Sort(dataGridView1.Columns[4],
ListSortDirection.Descending);
    }
    else if (listBox1.Text == "Контактный_номер")
    {
        dataGridView1.Sort(dataGridView1.Columns[5],
ListSortDirection.Descending);
    }
}

```

```
}
```

```
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void button5_Click_1(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void richTextBox2_TextChanged(object sender, EventArgs e)
```

```
{
```

```
    string text = richTextBox1.Text;
```

```

        string text2 = richTextBox2.Text;
        dataGridView1.DataSource = DoubleSearchData(text, text2);
    }

    private void button6_Click(object sender, EventArgs e)
    {
        Form3 f3 = new Form3();
        this.Hide();
        f3.Show();
    }
}
}
}

```

### **«Тіркеу» үшін программа листингі**

```

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;
using System.Data.Common;
using System.Data.SqlClient;

```

```

namespace WindowsFormsApp3

```

```

{
public partial class Form5 : Form
{
    public Form5()
    {
        InitializeComponent();
    }

    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void label3_Click(object sender, EventArgs e)
    {

    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        int ID_клиент = Convert.ToInt32(richTextBox1.Text);
        string Фамилия = richTextBox2.Text;
        string Имя = richTextBox3.Text;
    }
}

```

```

string Отчество = richTextBox4.Text;
string Дата_рождения = richTextBox5.Text;
string Контактный_телефон = richTextBox6.Text;

string mark = @"Data Source=DESKTOP-IEIOC71\SQLEXPRESS;Initial
Catalog=Китапхана;Integrated Security=True";
SqlConnection connection = new SqlConnection(mark);
try
{
    connection.Open();
    string commandText = "insert into Тиркеу(ID_клиент, Фамилия, Имя,
Отчество, Дата_рождения, Контактный_телефон ) values (@ID_клиент,
@Фамилия, @Имя, @Отчество, @Дата_рождения, @Контактный_телефон)";
    SqlCommand command = new SqlCommand(commandText, connection);
    command.CommandType = CommandType.Text;
    command.Parameters.AddWithValue("@ID_клиент", ID_клиент);
    command.Parameters.AddWithValue("@Фамилия", Фамилия);
    command.Parameters.AddWithValue("@Имя", Имя);
    command.Parameters.AddWithValue("@Отчество", Отчество);
    command.Parameters.AddWithValue("@Дата_рождения",
Дата_рождения);
    command.Parameters.AddWithValue("@Контактный_телефон",
Контактный_телефон);

    command.ExecuteNonQuery();
    MessageBox.Show("Вы успешно зарегистрированы!");
}
catch (Exception ex)
{

```



```

        MessageBox.Show(ex.Message.ToString());
    }
    finally
    {
        connection.Close();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string connectionString = @"Data Source=DESKTOP-
IEIOC71\SQLEXPRESS;Initial Catalog=Книпхана;Integrated Security=True";

    SqlConnection myConnection = new SqlConnection(connectionString);

    myConnection.Open();

    string query = "SELECT * FROM Тиркеу ORDER BY ID_клиент";

    SqlCommand command = new SqlCommand(query, myConnection);

    SqlDataReader reader = command.ExecuteReader();

    List<string[]> data = new List<string[]>();

    while (reader.Read())
    {
        data.Add(new string[6]);
    }
}

```

```

        data[data.Count - 1][0] = reader[0].ToString();
        data[data.Count - 1][1] = reader[1].ToString();
        data[data.Count - 1][2] = reader[2].ToString();
        data[data.Count - 1][3] = reader[3].ToString();
        data[data.Count - 1][4] = reader[4].ToString();
        data[data.Count - 1][5] = reader[5].ToString();
    }

    reader.Close();
    myConnection.Close();
    foreach (string[] s in data)
        dataGridView1.Rows.Add(s);
}

private void button3_Click(object sender, EventArgs e)
{
    Form1 f1 = new Form1();
    this.Hide();
    f1.Show();
}
}
}

```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
Сәтбаев университеті

## ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Казиева Айдана Батырбайқызы

Дипломдық жұмыстың тақырыбы: «UML тілі көмегімен ақпараттық жүйелерді жобалау»

Казиева А. дипломдық жұмысының мақсаты UML тілі көмегімен ақпараттық жүйелерді жобалау болып табылады.

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

Дипломдық жұмысты жасау барысында пәндік аймақ бойынша шолу жасалды; жүйенің функционалдық үлгісі жасалды; жүйенің іс-әрекеттер, класстар және тізбектер диаграммалары құрылды; жүйенің ER – моделі жасалды; жасалған мәліметтер базасы негізінде ақпараттық жүйе құрылды.

Жүйені құруда қолданылған орталар: UML объектілі бағытталған ортасы; Microsoft Visual Studio платформасы; Бағдарламалау тілі: C#, Microsoft SQL Server Management Studio

Дипломдық жұмысты жасау барысында Казиева А. 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 Казиева А. «UML тілі көмегімен ақпараттық жүйелерді жобалау» тақырыбындағы жұмысы қорғауға жіберілуі мүмкін.

Ғылыми жетекші,  
сенъор-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:** Казиева Айдана

**Название:** UML тілі кемелімен ақпараттық жүйелерді жобалау

**Координатор:** Шолпан Байматаева

Коэффициент подобия 1: 0,67

Коэффициент подобия 2: 0,0

Замена букв: 6

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 отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

Заимствования являются добросовестными и не обладают признаками плагиата.

Дата

Сейлова Н.А., зав. кафедрой КБОиХИ

## 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: Казиева Айлана

Название: UML тілі көмегімен аспаптық жүйелерді жобалау

Координатор: Шолпан Байматаяв

Коэффициент подобия 1: 0,67

Коэффициент подобия 2: 0,00

Замена букв: 6

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