

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Төле Ақерке Асылханқызы

“Түгендеу журналы” Windows қосымшасын әзірлеу

ТҮСІНДІРМЕ ЖАЗБА

ДИПЛОМДЫҚ ЖОБАҒА

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы



ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд, профессор

Мо А.Н.Молдагулова

" 20 " 05 2022 ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жұбаға

Тақырыбы: "Түгендеу журналы" Windows қосымшасын әзірлеу

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

Орындаған

Төле А.А.

Рецензент

PhD, қауымдастырылған профессор

Кашаганова Г.Б. Кашаганова

" 19 " 05 2022 ж.

Ғылыми жетекші

PhD, ассистент-профессор

Алибиева Ж.М. Алибиева

" 16 " 05 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы



БЕКТЕМІН

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд, профессор

М.С. А.Н.Молдагулова

" 20 " 05 2022 ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға *Тәле Ақерке Асылханқызы*

Тақырыбы: «Түгендеу журналы» Windows қосымшасын әзірлеу»

Академиялық мәселелер жөніндегі проректордың бұйрығының

№ *119-ПБ Д.М.* 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " *23* " *05* 2022 ж.

Дипломдық жобаның бастапқы мәліметтері: *Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма, жоба диаграммалары түрінде ақпаратты жинау, деректер қорына сақтау, тестілеу, тексеруге арналған программалық қамтамаларды жасау жүргізілген.*

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) тақырып бойынша талдау;

b) жобаны жобалау;

в) жобаның интерфейсін жобалау және дамыту;

г) бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):



презентацияның 20 слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер: *20 пайдаланылған әдебиеттер тізімінен.*


Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жобаның жоспарын құру	14.01.2022	Орындалды
2. Тапсырма қойылымы және программалау ортасын таңдау	18.01.2022	Орындалды
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім бойынша есеп беру жазбасын дайындау	01.02.2022	Орындалды
4. Дипломның екінші және үшінші бөлімдерін, жобалау сызбаларын дайындау	15.02.2022	Орындалды
5. Жобаның веб-қосымшасын тегілеуден өткізу	18.03.2022	Орындалды
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2022	Орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Жекамбаева М.Н. PhD, қауымдастырылған профессор	19.05.22	
Бағдарламалық бөлім	Марғұлан Қ. техн.ғыл.магистірі, лектор	18.05.22ж	

Ғылыми жетекші  Алибиева Ж.М.

Тапсырманы орындауға қабылдап алған студент  Төле А.А.

Күні  11 2021ж.

АҢДАТПА

Бұл дипломдық жобада студенттер жатаханасының қызметіне әзірленген «Түгендеу журналы» Windows-қосымшасы қарастырылады. Қосымша жатаханаға тиесілі инвентарды түгендеп тіркеуге бағытталған.

Қазіргі таңда аталған жүйе қолмен жүзеге асырылатындықтан ұсынылған бағдарламалық қамтама құжаттарды толтыру барысында кездесетін қателіктерді болдырмайды, қағазбастылықты жояды. Осылайша жатахана қызметкерлерінің жұмысын жеңілдетуге мүкіндік береді.

Дипломдық жобаны сипаттау мәтіні кіріспе, төрт тараудан тұратын негізгі бөлім (Жобаға жалпы шолу, Windows-қосымшасын жасау кезінде қолданылған программалық құралдар, Жобалау бөлімі, Программалау бөлімі), қорытынды, пайдаланылған әдебиеттер тізімін қамтиды.

Дипломдық жұмысты орындауда бағдарламалау тілі ретінде C# және программалық платформа .NET framework қолданылды. Мәліметтер базасымен жұмыс жасау барысында SQL Server қолданыста болды.

АННОТАЦИЯ

В данном дипломном проекте рассматривается Windows-приложение «Инвентарный журнал», разработанное для деятельности студенческой общешития. Приложение направлено на инвентаризацию принадлежащего общешитию инвентаря.

В настоящее время, поскольку данная система реализуется вручную, предлагаемое программное обеспечение сокращает время на заполнение и подтверждение документов, исключает ошибки при заполнении, устраняет бумажную волокиту. Таким образом, можно упростить работу сотрудников общешития.

Текст описания дипломного проекта включает введение, основную часть, состоящую из четырех глав (общий обзор проекта, программные средства, использованные при разработке Windows-приложения, проектный раздел, раздел программирования), заключение, список использованной литературы.

В качестве языка программирования при выполнении дипломной работы использовались C# и программная платформа .NET framework . При работе с базами данных использовался SQL Server.

ANNOTATION

In this graduation project, the Windows application "Inventory Magazine" is considered, developed for the activities of the student dormitory. The application is aimed at inventory of the inventory belonging to the hostel.

Currently, since this system is implemented manually, the proposed software reduces the time for filling out and confirming documents, eliminates errors when filling out, eliminates paperwork. Thus, it is possible to simplify the work of the hostel staff.

The text of the description of the diploma project includes an introduction, the main part consisting of four chapters (a general overview of the project, software tools used in the development of Windows applications, project section, programming section), conclusion, references.

C# and the software platform were used as a programming language when completing the thesis .NET framework. When working with databases, SQL Server was used.

Мазмұны

	Кіріспе	9
1	Жобаға жалпы шолу	10
1.1	Жобаның негізгі мақсаты және міндеттері	10
1.2	Жобаға жүргізілген салыстырмалы талдау	10
1.3	Жобаның өзектілігі	12
2	Windows-қосымшасын жасау кезінде қолданылған программалық құралдар	14
2.1	C# бағдарламалау тілі	14
2.2	.NET Framework	15
2.3	Visual Studio 2022	15
2.4	MS SQL Server	16
2.5	ADO NET	19
2.6	Windows Forms	21
3	Жобалау бөлімі	23
3.1	Қосымшаның контекстік диаграммасы	23
3.2	ER диаграмма	24
3.3	Прецеденттер диаграммасы	26
3.4	Реттілік диаграммасы	27
4	Программалау бөлімі	28
4.1	Қосымшаны әзірлеу	28
4.2	Windows-қосымшаның функционалды құрылымын жасау	29
4.3	Windows қосымшаның интерфейсі	30
	Қорытынды	39
	Қолданылған әдебиеттер тізімі	40
	А қосымшасы. Техникалық тапсырма	42
	Б қосымшасы. Бағдарлама мәтіні	45

КІРІСПЕ

Бухгалтерлік есеп пен есеп қағазда дұрыс болуы мүмкін, бірақ олардың дұрыстығын тек түгендеу арқылы анықтауға болады.

Мүлікті түгендеу - бұл көп уақытты қажет ететін жұмыс, егер тексерілетін кәсіпорын қоймалар мен өнімдердің көптігі бар үлкен болса, оны жүргізу уақыты бірнеше күнге созылуы мүмкін. Сондай-ақ, алынған мәліметтер мен бухгалтерлік ақпарат арасындағы алшақтықты көрсететін тексеру нәтижелерін өңдеуге көп күш жұмсалады. Түгендеу-бұл барлық кәсіпорындар үшін жыл сайынғы міндетті рәсім. Басты мақсат - бухгалтерлік есеп деректерінің нақты жағдайға сәйкестігін тексеру. Бірақ барлық ұйымдар түгендеуді жүргізудің маңыздылығын түсінбейді, оны формальдылыққа айналдырады. Алайда, тек осы процедура барысында мүліктің артық немесе жетіспеушілігін анықтауға, объектілердің нақты жағдайын анықтауға, мүліктік қатынастарды реттеуге, есепте көрсетілген берешектің шынайылығын бағалауға, шығындарды азайту мүмкіндігін анықтауға, бухгалтерлік есепті түзетуге және ең бастысы салықтық тәуекелдерді азайтуға болады.

Жоғары сапалы нәтижеге жету мақсатында ақпаратты енгізу арқылы қол жеткізуге көмектесетін "Түгендеу журналы" Windows қосымшасы әзірленді.

Қосымшаны әзірлеуде .NET фреймворкі таңдалынды. .NET – C# бағдарламалау тілінде жазылған фреймворк. Аталған фреймворк кез келген жоба үшін үлкен құндылықтарды ұсынады: қосымшаларды құру процесін жылдамдатуға және жеңілдетуге көмектесетін дайын компоненттер жиынтығымен қамтамасыз етілген. .NET фреймворкі ашықтығы мен қолжетімділігінің, жетілдіруге арналған тегін әрі ыңғайлы құралдардың бар болуының арқасында көп қолданысқа ие.

Дипломдық жұмыстың мақсаты: қазіргі заманғы технологияларды қолдана отырып жатақхана қызметкерлерінің жұмысын жеңілдетуге және оңтайландыруға бағытталған Windows қосымша әзірлеу.

Қосымшаның қолдану аймағы – жатақхана жүйесінде қызмет ететіндер болып табылады.

1 Жобаға жалпы шолу

1.1 Жобаның негізгі мақсаты және міндеттері

Негізгі мүлікті түгендеу жүргізу мақсаттары:

- мүліктің тиісті жай-күйін, сақтау және пайдалану шарттарын бағалау;
- қойманы басқарудағы қателерді анықтау;
- құжаттама мен нақты деректерді салыстыру;
- жетіспеушілік пен артық шығу себептерін жою.

Түгендеу барысында келесі көрсеткіштер тексеріледі:

- тауарлар;
- шикізат және материалдар;
- негізгі құралдар;
- материалдық емес активтер;
- резервтер.

Ал енді менің жобамның негізгі мақсаты жоғарыда көрсетілген талаптарды орындай отырып “Түгендеу журналы” қосымшасын жобалау және бағдарламалау.

Алға қойға мақатымызға жету үшін мынадай міндеттерді орындауымыз қажет:

- Visual Studio- ортасына және C# бағдарламалау тіліне талдау жүргізу;
- Салыстырмалы түрде тақырып бойынша жан жақты ақпарат жинау;
- Кемшіліктерді түзете отырып бағдарламаның концептуалды моделін құру;
- “Түгендеу журналы” бағдарламасын әзірлеу;
- Бағдарламаны тестілеуден өткізу.

1.2 Жобаға жүргізілген салыстырмалы талдау

Мүлікті түгендеу кез-келген көлемдегі ұйымның ажырамас бөлігі болып табылады. Инвентарьды есепке алу мүлікті өзекті күйде ұстап тұруға және оған қызмет көрсетуге артық шығындарды болдырмауға мүмкіндік береді.

Әрбір құрылғы туралы ақпаратты қолмен жинау ұзақ уақыт бойы оның күрделілігіне байланысты маңызды емес. Мүлікті түгендеу үшін арнайы бағдарламалар тапсырманы шешуге және мүмкіндігінше қысқа мерзімде нақты ақпарат алуға мүмкіндік береді. Осындай бағдарламаны бір рет орнатқаннан кейін сіз кез-келген уақытта әр құрылғы үшін егжей-тегжейлі және нақты қорытынды ала аласыз.

Түгендеу журналы Windows қосымшасын құруға мүмкіндік беретін нарықтағы бар әзірлемелерді талдау кезінде жатақханаларды түгендеу журналы жүргізілетін документтері қарастырылды.

Көптеген жатақханаларда түгендеу журналы excel файлдары арқылы

жүргізіледі. Мысал ретінде төмендегі суретте жатақхана түгендеу журналын файл түріндегі көрінісін көре аламыз.

национальный исследовательский технический университет Организация (индивидуальный предприниматель) имени К.И. Сатпаева"		ИНН/БИН	150140008602
ИНВЕНТАРИЗАЦИОННАЯ ОПИСЬ ДОЛГОСРОЧНЫХ АКТИВОВ		Номер документа	в составе
		00000000020	11.09.2020
Объекты долгосрочных активов			
Местонахождение			
РАСПИСКА			
бухгалтерию, и все объекты долгосрочных активов, поступившие на мою ответственность, оприходованы, а выбывшие списаны с баланса.			
Материально-ответственное(ые) лицо(а):	Менеджер должность		подпись зашифровка подписи

1.1-сурет – Excel форматындағы түгендеу журналы

Инвентаризация начата

При инвентаризации установлено следующее:

Номер по порядку	Наименование, характеристика	Дата выпуска (постройки)	Дата приобретения (регистрации)	Номер		По данным бухгалтерского учета	
				инвентарный	заводской	количество	стоимость, в KZT
1	2	3	4	5	6	10	11
1	Принтер Лазерный	13.04.2006	13.04.2006	2410-26570		1	25 538
2	Принтер HP LJ 1022	14.08.2007	14.08.2007	2410-26712		1	35 000
3	Телевизор Самсунг рабочая станция	24.12.2008	24.12.2008	2410-787		1	374 252,27
4	комплект(монитор LCD22 Philips)Незавод. СХХVB8721009310 для	09.12.2010	09.12.2010	2410-5414		1	235 898
5	Кресло для руководителя	28.03.2011	28.03.2011	2410-5664		1	44 233
6	МФУ Canon MF4410 (JZY	30.05.2011	30.05.2011	2410-5955		1	75 080
7	Ресепшен	08.01.2020	08.01.2020	241059493		1	123 245,15
8	Контейнер для мусора	11.10.2012	11.10.2012	2410-13583		1	12 720
9	Контейнер для мусора	11.10.2012	11.10.2012	2410-13584		1	12 720
10	Контейнер для мусора	11.10.2012	11.10.2012	2410-13585		1	12 720
11	Контейнер для мусора	11.10.2012	11.10.2012	2410-13586		1	12 720
12	Контейнер для мусора	11.10.2012	11.10.2012	2410-13587		1	12 720
13	ПЭВМ-Рабочая станция Моноблок Lenovo S710 AIO.21.5W LED	02.06.2014	02.06.2014	2410-20264	WS70355878	1	155 950
14	ПЭВМ-Рабочая станция Моноблок Lenovo S710 AIO.21.5W LED	02.06.2014	02.06.2014	2410-20298	WS70356283	1	155 950
15	ПЭВМ-Рабочая станция Моноблок Lenovo S710 AIO.21.5W LED	02.06.2014	02.06.2014	2410-20549	WS70355712	1	155 950
16	Плита электрическая Flama	25.06.2019	25.06.2019	241057669		1	63 001
17	Плита электрическая Flama	25.06.2019	25.06.2019	241057670		1	63 001
18	Плита электрическая Flama	25.06.2019	25.06.2019	241057671		1	63 001
19	Плита электрическая Flama	25.06.2019	25.06.2019	241057672		1	63 001
20	Плита электрическая Flama	25.06.2019	25.06.2019	241057673		1	63 001
21	Плита электрическая Flama	25.06.2019	25.06.2019	241057674		1	63 001
22	Плита электрическая Flama	25.06.2019	25.06.2019	241057675		1	63 001
23	Плита электрическая Flama	25.06.2019	25.06.2019	241057676		1	63 001

-сурет – Excel форматындағы түгендеу журналы

1					
2		в секции (2комната, туалет)	65	2	130
3		коридор 1 эт		0	0
4		коридор 2 эт		0	0
5		бытовка	1	4	4
6		служебный туалет		0	0
7		подвал		0	0
8		фое	1	1	1
9		уличное под козырьком		0	0
10		актовый зал 2 эт	1	2	2
11		читальный зал 1 эт		1	0
12		читальный зал 2 эт	1	1	1
13		склад		0	0
14		прачка	1	3	3
15		менеджер	1	1	1
16		лестница	1	3	3
17					
18					145
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					

-сурет – Excel форматындағы түгендеу журналы

1.3 Жобаның өзектілігі

Жылдық есептілікті дайындау процесінде барлық компаниялар "бухгалтерлік есеп және қаржылық есептілік туралы"Қазақстан Республикасының 2007 жылғы 28 ақпандағы № 234-ІІІ Заңының (28.12.2016 жылғы ред.) негізінде өздерінің негізгі құралдарына түгендеу жүргізеді. Негізгі құралдарды, олардың құны мен орналасуын есепке алу — қазіргі уақытта еңбек шығыны болып табылады.

Түгендеу журналы менің жұмысымда жатақхана қызметінің түгендеу журналына бағатталған.

Қазіргі таңда «Түгендеу журналы» қолмен жүзеге асырылатындықтан ұсынылған Windows қосымша құжаттарды толтыру мен растау уақытын азайтады, толтыру барысында кездесетін кателіктерді болдырмайды, қағазбастылықты жояды. Осылайша жатақхана қызметкерлерінің жұмысын жеңілдетуге мүкіндік береді.

Windows қосымша арқылы:

- деректерді енгізе аламыз;
- деректерді өзгерте аламыз;
- қандай да бір белгісі бойынша іздеу жүргізе аламыз (атауы, орналасқан жері, жауапты тұлға және т. б.);
- енгізілген түгендеу журналын excel форматында да ашып көруге мүмкіндігіміз болады;

2 Windows-қосымшасын жасау кезінде қолданылған программалық құралдар

2.1 C# бағдарламалау тілі

Қосымшаның жүзеге асырылу кезінде бағдарламалау тілі ретінде C# бағдарламалау тілі қарастырылды. C# —объектіге бағытталған бағдарламалау тілі.

C# бағдарламашыларға(программисттерге) .NET жүйесінде жұмыс істейтін қауіпсіз және сенімді веб-қосымшалардың әртүрлі түрлерін жасауға мүмкіндік береді.

C# синтаксисі басқа бағдарламалау тілдеріне қарағанда әлдеқайда ерекшеленеді. Ол автоматты түрде жадыдан қолданылмайтын немесе жойылған қосымшалардың объекттерін жойып отырады. Сонымен қатар ол көптеген құралдар мен фреймворктарды қолдануға мүмкіндігі жетеді.

2.1-кесте- C# программалау тілінің ерекшеліктері

Параметр	
Экожүйе	Microsoft платформаларына арналған бағдарламалық жасақтаманы әзірлеу үшін қолданылады
Генериктерді қолдау	CLI - ге біріктірілген және түрі туралы ақпаратты жұмыс уақытында қол жетімді етуге мүмкіндік береді
Делегаттарды қолдау	Мақсатты нысанды білместен шақыруға болатын әдістер ретінде қызмет ететін делегаттар бар
Тексерілген ерекшеліктер	Белгіленген және тексерілмеген ерекшеліктерді ажыратады
Полиморфизм	Әдепкі полиморфизмді қамтиды
Кірістірілген деректер түрі	Мән бойынша берілетін кірістірілген деректер түрлері қарабайыр типтер деп аталады
Шартты компиляцияны қолдау	C# препроцессор директивалары арқылы шартты компиляция функциясын қолдайды

2.1-кесте жалғасы

Goto операторын қолдау	C# goto операторын қолдайды
Платформалық қолдау	Java - мен салыстырғанда c# бұл мүмкіндікті жақсартуды қажет етеді
	Microsoft C# API басқарады
Танымал компаниялар	Stack Exchange, Microsoft, Coderus, Docplanner және т.б.

2.2 .NET Framework

Microsoft. NET Framework- Microsoft компаниясы әртүрлі бағдарламалау тілдерін қолдана отырып жазылған бағдарламалық өнімдердің үйлесімділігін қамтамасыз ету мақсатында шығарған бағдарламалық платформа. Егерде біз .NET Framework пен жұмыс жасайтын болсақ бағдарламалар операциялық жүйенің қажетті компоненттерімен өзара әрекеттесе алады, яғни бағдарламалар қай бағдарламалау тілінде жасалғанына қарамастан сұраныстарды жібереді және өңдейді.

Жоба .NET Framework тің 4.7.2 версиясы бойынша жұмыс жасайды



Пакет нацеливания для платформы .NET Framework 4.7.2

2.1-сурет - .NET Framework тің 4.7.2 версиясының орнатылуы

Microsoft. NET Framework 4.7.2-бұл алдыңғы версиялардың (. NET Framework 4, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7 және 4.7.1.) жаңартылуы нәтижесінде жасалған платформа. Офлайн пакетті интернет байланысы болмай қалған жағдайда пайдалануға мүмкіндік береді.

2.3 Visual Studio 2022

ASP.NET-пен жұмыс істегенде, веб-қосымшаларды әзірлеудің бірнеше нұсқалары бар. Біріншісі, әр веб-бет пен сыныптың кодын қарапайым мәтіндік редакторда қолмен жазуға болады. Бұл тәсіл өте түзу, бірақ көп күш жұмсауды қажет етеді және күрделі беттерде қателесу мүмкіндігі жоғары болуы мүмкін. Сондықтан кәсіби ASP.NET бағдарламашылары(програмисттер) оған сирек

жүгінеді. Оның орнына ауқымды ASP.NET веб-сайттарының барлығы дерлігі Visual Studio бағдарламасымен жасалынады.

Visual Studio 2022 нің қолдану ерекшеліктері:

- IntelliCode-бұл кодтың мәтін мәнін танытын кодты автоматты түрде аяқтаудың қуатты жиынтығы: айнымалы атаулар, функциялар және жасалған код түрі. Бұл IntelliCode-ге бүкіл жолды бірден аяқтауға мүмкіндік береді.
- CodeLens маңызды аналитикалық ақпаратты оңай табуға көмектеседі, мысалы, қандай өзгерістер жасалса, олар не әкелді, сонымен қатар сіздің әдісіңізге блокты тестілеу жүргізіледі. Сілтемелер, авторлар, тесттер және бекіту тарихы сияқты негізгі ақпарат әрқашан қолжетімді.
- Интеграцияланған күйін келтіру Visual Studio өнімдерінің маңызды құрамдас бөлігі болып табылады. Кодты талдап, айнымалыларда сақталған мәндерді зерттеп, мәндердің өзгеруін бақылау үшін айнымалылардың басқару мәндерін реттей аламыз, кодтың орындалу жолын және бағдарламалық жасақтаманың барлық басқа ерекшеліктерін біле аламыз.

2.4 MS SQL Server

MS SQL Сервері-бұл Microsoft дерекқорының өнімі. Бұл пайдаланушыларға SQL сұрауларын орындауға және оларды орындауға мүмкіндік береді. Бұл мәліметтер базасы үшін ең тұрақты, қауіпсіз және сенімді шешімдердің бірі. Ол корпоративтік it орталарында транзакцияларды, аналитиканы және бизнес-аналитиканы өңдеуге арналған көптеген қосымшаларды қолдайды.

Microsoft SQL Server ерекшеліктері:

- SQL Server Profiler қолдау құралдары, бизнесті талдау құралдары, SQL Server Management Studio және мәліметтер базасын құру көмекшісі;
- онлайн қолдау мен құжаттаманы, сондай-ақ нақты уақыттағы өнімді қолдауды ұсынады;
- деректер түрлерін салыстыру және нысандарды жою және қайта атау үшін кеңейтілген теңшеу мүмкіндігін ұсынады;
- орындау терезесінде қате туралы хабарламалар мен көші-қон туралы ескертулерді көрсетеді;
- SQL Server ДҚБЖ ядросын басқару және авторизациялау үшін бірыңғай интеграцияланған орта;
- диалогтың өзгермелі өлшемдері тілқатысу терезесі ашық болған кезде бірнеше құралдарға қол жеткізуге мүмкіндік береді;
- сүзгілеу және автоматты жаңарту қызметі мониторының функциясы;
- SQL Server Management Studio-дан импорттау және экспорттау.

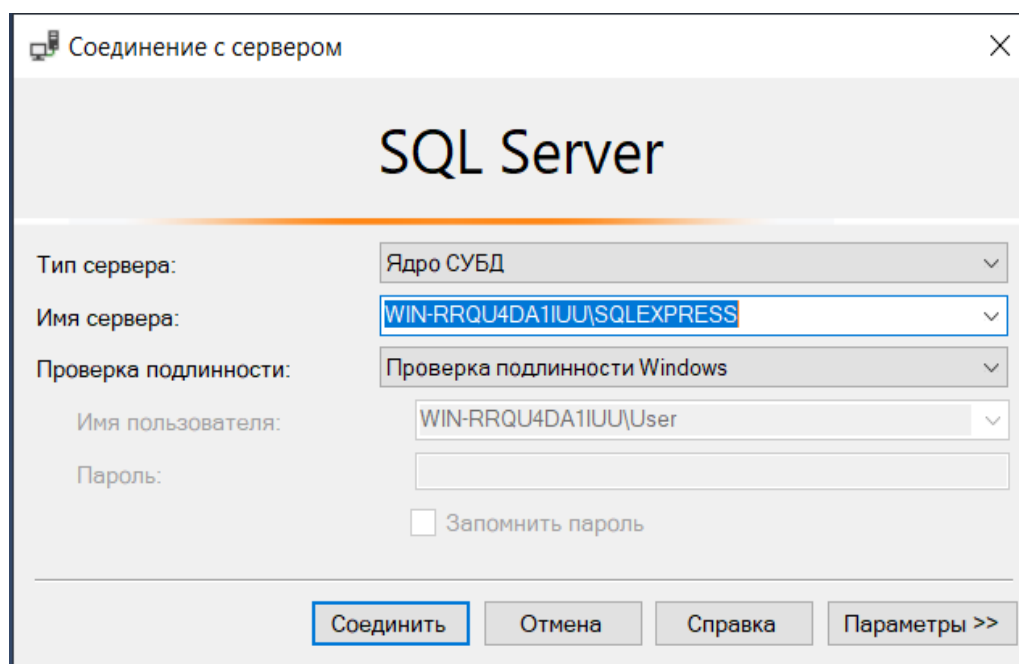
-кесте – MS SQL Server және Oracle салыстырмалы сипаттамасы

Параметр		
Синтаксис	Қарапайым және жеңіл синтаксис	Күрделі және тиімді синтаксистер
Тапсырмаларды жоспарлау	SQL Server агенті арқылы тапсырмаларды жоспарлау	Oracle жоспарлаушысы немесе OEM арқылы тапсырмаларды жоспарлау
Бит карталарының индекстері	Бит карталарының индекстері кері пернелер мен функцияларға негізделмейді	Нүктелік кескінді, функцияға негізделген индекстерді және кері кілттерді қолданады
Сұраныстарды оңтайландыру	Сұраныстарды оңтайландыру жоқ	Star сұрауларын оңтайландыруды қолданады
Қолдау және ақаулықтарды жою	Техникалық жазбаларды, қате сипаттамаларын, сценарийлерді, түзетулерді және жүктеуді қосымша ақысыз ұсынады	Қолдау қызметіне хабарласудың әрбір жағдайы бойынша төленетін қолдау қызметіне қоңырау шалу
Мәннің өзгеруі	Мәндер бекітілгенге дейін өзгереді	Бекітпес бұрын мәндер өзгермейді
Параллель орындау	серверінде INSERT, UPDATE, DELETE операторлары рет-ретімен орындалады	Oracle INSERT, UPDATE, DELETE және MERGE нұсқауларын параллель орындайды
Ортақ пайдалану	Әрбір дерекқорда серверде ортақ пайдаланылмай, өзінің жеке диск файлы бар	Деректер базасының барлық объектілері схемалар бойынша топтастырылған. Дерекқор нысандарының ішкі жиыны және барлық дерекқор нысандары барлық схемалар мен пайдаланушылармен қолданылады

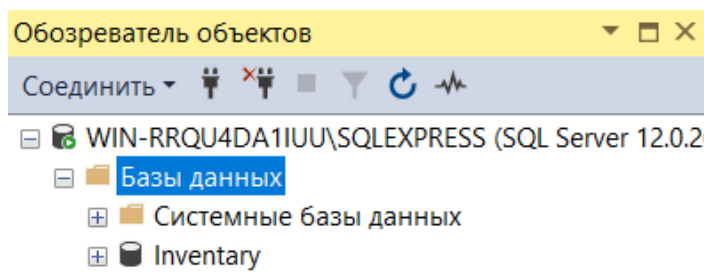
2.2-кесте жалғасы

Сақтық көшірме жасау	Толық, ішінара және қосымша сақтық көшірмелер жасауға мүмкіндік береді.	Деректер базасының толық, файлдық, қосымша және дифференциалды сақтық көшірмелерін жасауға мүмкіндік береді
----------------------	---	---

Дипломдық жобаға MS SQL Server-ді игерудің жеңілдігі мен қол жетімділігін ескере отырып таңдалды, өйткені бұл ДҚБЖ қосымша жабдықтар мен пайдалану бойынша зерттеу шығындарын сатып алмай, қарастырылған функцияларды толығымен автоматтандыруға мүмкіндік береді. Бағдарламалық жасақтаманың толық жұмыс істеуі үшін мәліметтер базасының файлдарын сервермен байланыстыру керек. Ол үшін SQL Server Management Studio бағдарламасын ашып, дерекқорды қосу керек.



2.2-сурет- Серверді байланыстыру



2.3-сурет- Деректер қорын байланыстыру

Дерекқорды қосқаннан кейін жобаның конфигурациялық файлында сервер атауын көрсету керек, онда Data Source өрісінде сервер атауын және таблицаның атауын көрсету керек.

```
[ApplicationScopedSetting]
[DebuggerNonUserCode]
[SpecialSetting(SpecialSetting.ConnectionString)]
[DefaultSettingValue("Data Source=WIN-RRQU4DA1IUU\SQLEXPRESS;Initial Catalog=Inventory;Trusted_Connection=true;")]
public string JournalDBConnectionString
{
    get
    {
        return (string) this[nameof (JournalDBConnectionString)];
    }
}
```

2.4-сурет- Деректер қорын байланыстыру

2.5 ADO NET

ADO.NET - деректер базасында немесе басқа көздерде (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, мәтіндік файлдар) сақталған деректерге қол жеткізуді және оларды басқаруды қамтамасыз ететін платформаға негізделген технология болып табылады.

ADO.NET бұл.NET Framework платформасына негізделген деректермен жұмыс істеу технологиясын ұсынады. Бұл технология бізге мәліметтер базасына сұраныстар жіберуге, қосылымдар орнатуға, мәліметтер базасынан жауап алуға және басқа да бірқатар операцияларды жасауға болатын сыныптар жиынтығын ұсынады.

Сонымен қатар, мәліметтер базасын басқару жүйелері көп болуы мүмкін екенін атап өткен жөн. Олардың мәні бойынша олар әртүрлі болуы мүмкін. MS SQL Server, мысалы, сұраныстарды құру үшін T-SQL тілін қолданады, ал MySQL және Oracle PL-SQL тілін қолданады. Әр түрлі дерекқор жүйелерінде әр түрлі мәліметтер болуы мүмкін. Басқа да сәттер әртүрлі болуы мүмкін. Алайда

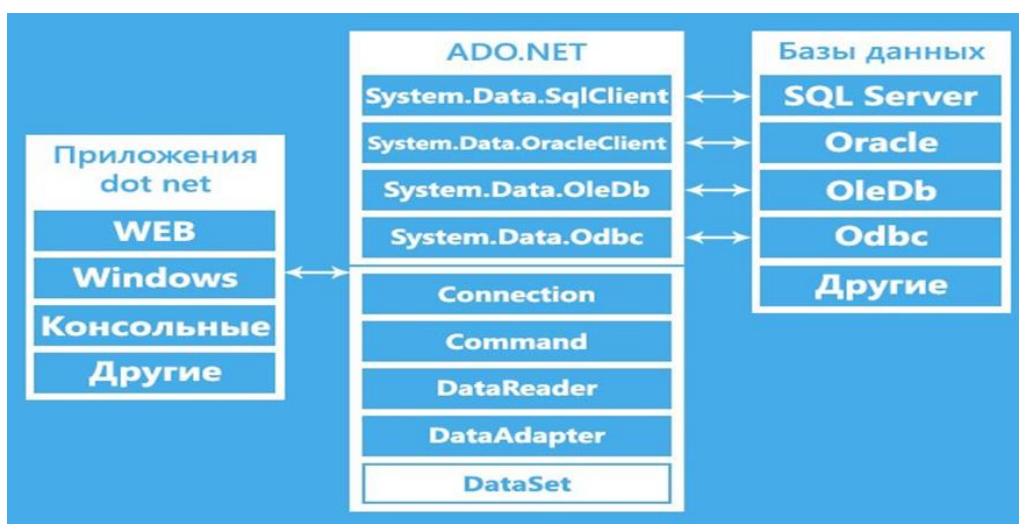
функционалдылық ADO.NET әзірлеушілерге әр түрлі ДҚБЖ-мен жұмыс істеу үшін бірыңғай интерфейсті ұсыну үшін салынған.

Мәліметтер базасымен өзара әрекеттесу интерфейсінің негізі ADO.NET нысандардың шектеулі шеңберін ұсынады: қосылу, пәрмен, `DataReader`, `DataSet` және `DataAdapter`. `Connection` нысанын қолдана отырып, деректер көзіне қосылу орнатылады. Пәрмен нысаны дерекқордағы деректермен операцияларды орындауға мүмкіндік береді. `DataReader` нысаны сұрау нәтижесінде алынған деректерді оқиды. `DataSet` нысаны деректер базасынан деректерді сақтауға арналған және мәліметтер базасынан тәуелсіз жұмыс істеуге мүмкіндік береді. Ал `DataAdapter` нысаны `DataSet` пен деректер көзі арасындағы делдал болып табылады. Негізінен, деректер базасымен жұмыс осы нысандар арқылы жүреді.

Дегенмен, әртүрлі деректер көздері үшін бірдей нысандар жиынтығын пайдалану үшін тиісті деректер провайдері қажет. Іс жүзінде деректер провайдері арқылы ADO.NET және деректер базасымен өзара әрекеттесу жүзеге асырылады. Және әрбір деректер көзі үшін ADO.NET жоғарыда аталған сыныптардың нақты іске асырылуын анықтайтын провайдер болуы мүмкін.

ADO.NET технологиясының келесі провайдерлері бар:

- MS SQL Server үшін провайдер
- OLE DB провайдері (MS SQL Server-дің кейбір ескі нұсқаларына, сондай-ақ access, DB2, MySQL және Oracle дерекқорына қол жеткізуге мүмкіндік береді)
- ODBC үшін провайдер (провайдерлері жоқ деректер көздері үшін провайдер)
- Oracle үшін провайдер
- EntityClient провайдері. ORM Entity Framework технологиясына арналған деректер провайдері
- SQL Server Compact 4.0 серверіне арналған провайдер



2.5-сурет- ADO.NET технологиясының жұмыс істеу принципі

2.6 Windows Forms

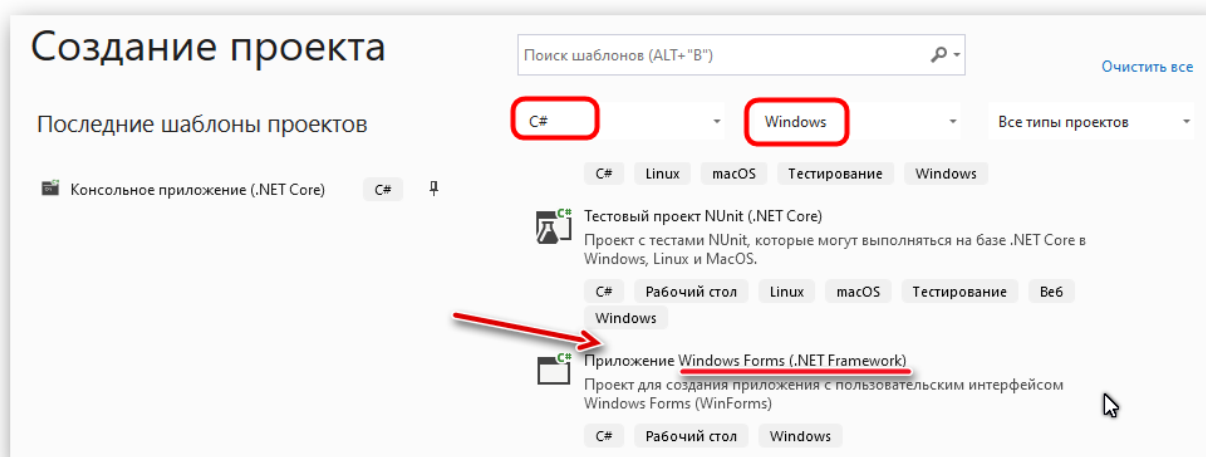
Windows Forms-пайдаланушының графикалық интерфейсіне жауап беретін және Microsoft.NET Framework құрамына кіретін қосымшаларды бағдарламалау интерфейсі. Ол Visual Studio-да визуалды конструкторды қолдана отырып, классикалық қосымшаларды құрудың ең тиімді әдістерінің бірін ұсынады.

Windows формаларында форма-бұл пайдаланушы үшін ақпарат көрсетілетін визуалды бет. Әдетте Windows Forms қосымшасы формаларға басқару элементтерін қосу және тінтуірдің басылуы немесе пернелерді басу сияқты пайдаланушының әрекеттеріне жауап беру үшін код құру арқылы жасалады. Басқару элементі-бұл деректерді көрсетуге немесе енгізуге арналған жеке интерфейс элементі.

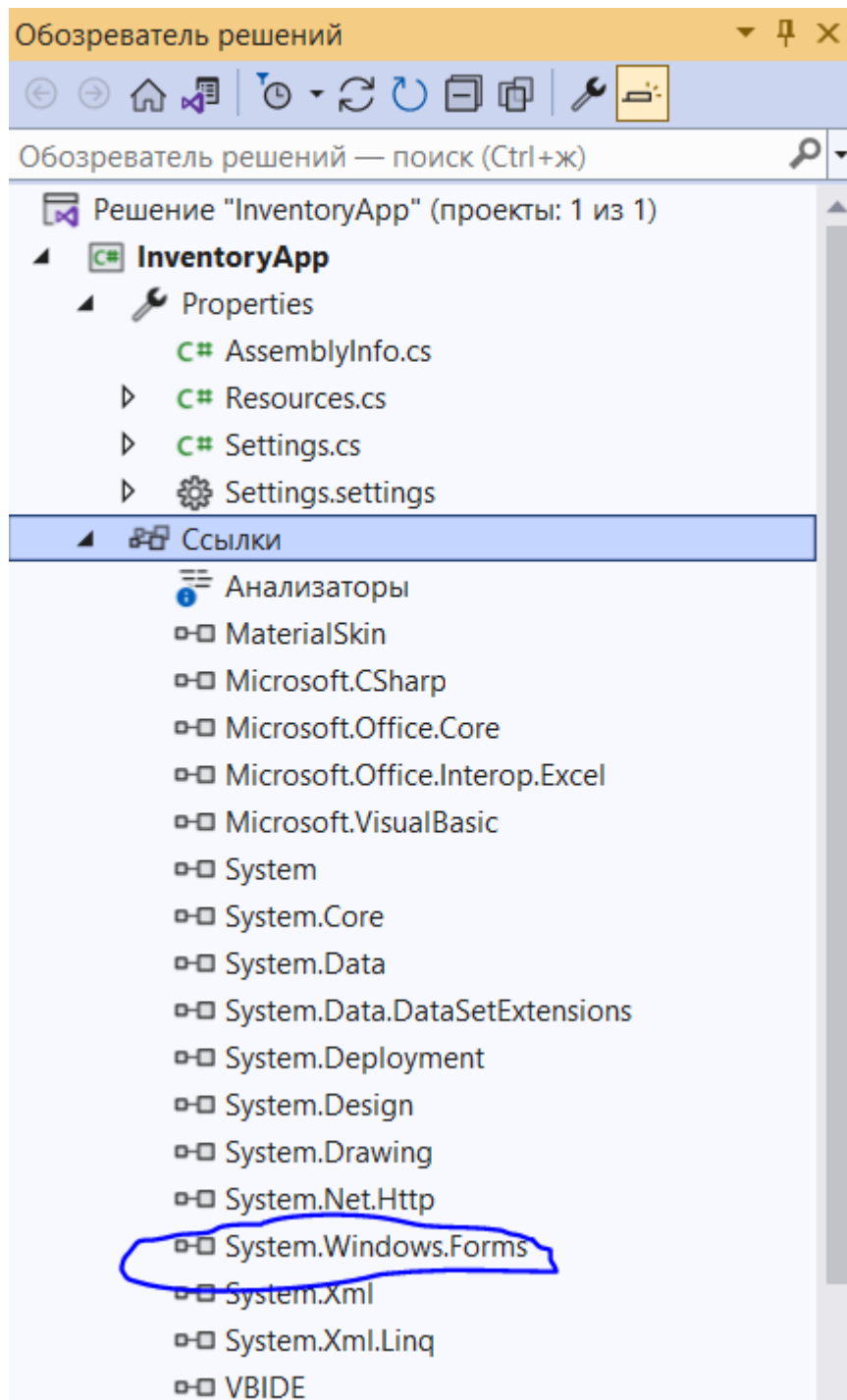
Пайдаланушы кез-келген әрекетті пішінмен немесе оның басқару элементтерінің бірімен орындаған кезде оқиға жасалады. Бағдарлама осы оқиғаларға кодта көрсетілгендей жауап береді және олар пайда болған кезде оқиғаларды өңдейді.

Windows Forms формаларға қосуға болатын көптеген басқару элементтерін ұсынады. Мысалы, басқару элементтері мәтіндік өрістерді, түймелерді, ашылмалы тізімдерді, қосқыштарды және тіпті веб-беттерді көрсете алады. Егер берілген басқару элементтері сіздің мақсаттарыңызға сәйкес келмесе, Windows Forms-те UserControl класын қолдана отырып, жеке басқару элементтерін жасауға болады.

Windows Forms пен жұмыс жасау үшін ең алдымен Visual Studio дан шаблон таңдап алу керек 2.6-суретте көрсетілгендей



2.6-сурет- Windows Forms шаблоны



2.7-сурет- Windows Forms кітапханасы

3 Жобалау бөлімі

3.1 Қосымшаның контекстік диаграммасы

Контекстік диаграмма жүйені "қара жәшік" деңгейінде сипаттайтын диаграммалар санатына жатады — атап айтқанда тек сыртқы қасиеттер (бұл жағдайда деректер ағындары), бірақ жүйенің мазмұны емес. Контекстік диаграммада 3 негізгі компонент бар:

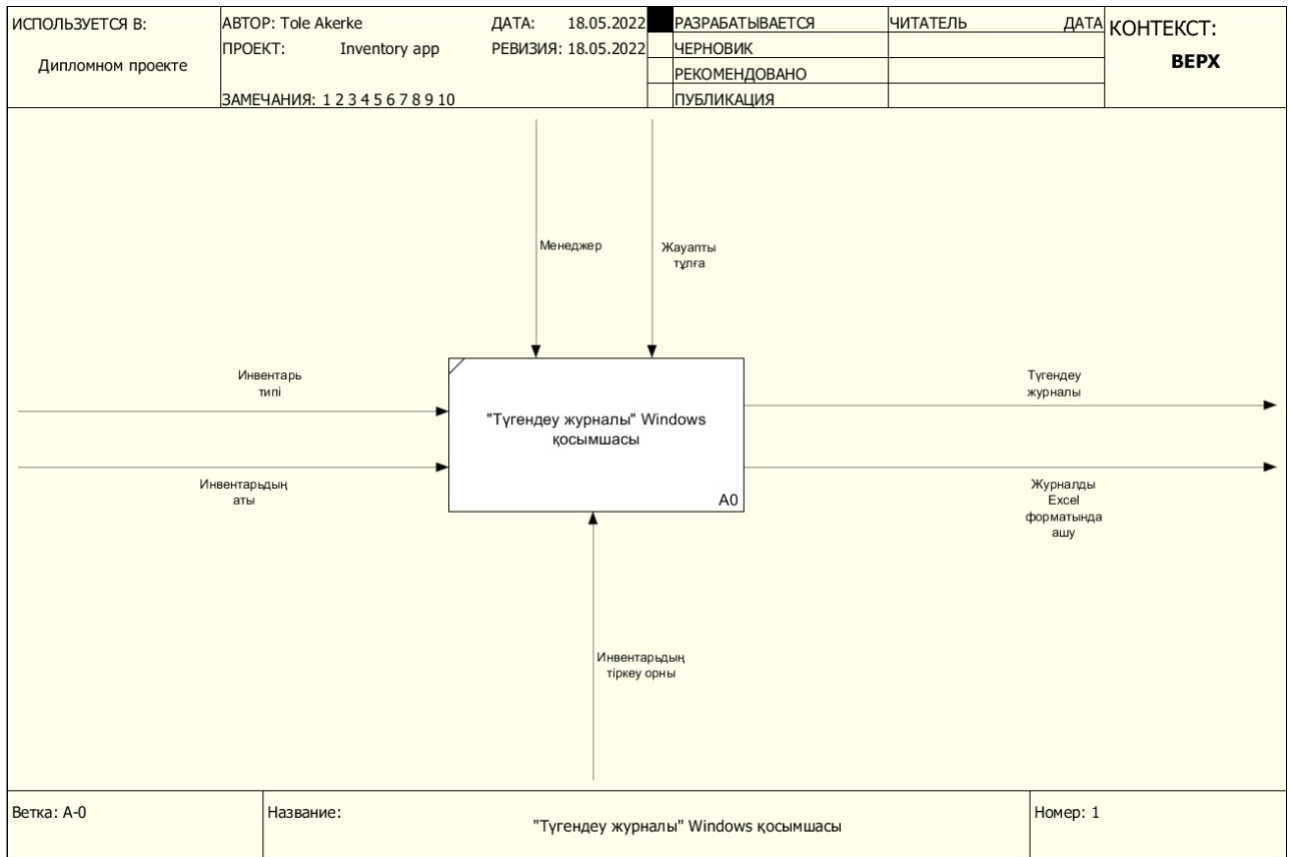
- Жобаланған нысан (мысалы, жүйе)
- Жобаланған объектімен өзара әрекеттесетін қоршаған орта элементтері (пайдаланушылар топтары, іргелес жүйелер)
- Деректер ағындары (Шығыс және кіріс)

Деректер ағындары қоршаған орта мен (бағдарламалық) жүйе арасында кез-келген жолмен-графикалық пайдаланушы интерфейсі (GUI), пәрмен жолы (CLI), бағдарламалық қоңыраулар (API), пошта хабарламалары және т. б. арқылы берілуі мүмкін. Егер жүйеде физикалық интерфейстер болса, онда бұл әртүрлі Джойстиктер, басқару тұтқалары, арнайы пернетақталар, қозғалысты тану сенсорлары, суреттер, кимылдар және т. б. болуы мүмкін.

Стандартты түрде диаграмманы қиындатпау үшін және бастапқы шешімдер қабылданғанға дейін қайталама шешімдер қабылдауға тырыспау үшін өзара әрекеттесу интерфейстерінің түрлерін және әсіресе хаттамаларды көрсету әдеттегідей емес.

Мәтінмәндік диаграмма жоғарғы деңгейдің функциясын, оның кірістерін, шығуларын, басқару элементтерін және механизмдерін сипаттайтын бір блоктан тұрады, сонымен қатар модель мақсаты мен модель құрылатын көзқарас тұжырымдарымен бірге. Мәтінмәндік диаграммадағы көрсеткілер жүйенің сыртқы элеммен өзара әрекеттесуін сипаттау үшін қолданылады. Олар диаграмманың шекарасынан басталып, жұмыста аяқталуы мүмкін немесе керісінше. Мұндай көрсеткілер шекара деп аталады.

Түгендеу журналының контекстік диаграммасы төмендегі 3.1-суретте бейнеленген. Сурет бойынша жобаланатын нысан-қосымша, кіріс деректерге-инвентарь аты, типі, менеджер, жауапты тұлға және де инвентарьдың тіркеу орны кіреді. Ал, шығыс деректерге түгендеу журналы мен журналдың excel форматындағы нұсқасы.



3.1-сурет-Түгендеу журналының контекстік диаграммасы

3.2 ER диаграмма

Нысан-байланыс схемасы (сонымен қатар ERD немесе ER диаграммасы)- бұл әртүрлі "нысандар" (адамдар, Нысандар, ұғымдар және т.б.) жүйе ішінде қалай өзара байланысты екендігі көрсетілген блок — схеманың бір түрі. ER диаграммалары көбінесе білім беру саласындағы реляциялық мәліметтер базасын жобалау және жөндеу, бизнеске арналған бағдарламалық жасақтама мен ақпараттық жүйелерді зерттеу және дамыту үшін қолданылады.

ER диаграммаларында нысандарды бейнелеу үшін тіктөртбұрыштарды, атрибуттарды және қатынастарды көрсету үшін ромбоидты формаларды қолданатын әртүрлі таңбалар бар.

Бір қарағанда, ER диаграммасы ағын схемасына өте ұқсас. Алайда, ER-диаграммада көптеген мамандандырылған таңбалар бар және олардың мағыналары бұл модельді ерекше етеді. ER диаграммасының мақсаты-объект құрылымының инфрақұрылымын ұсыну.

ER моделі Entity Relationship Model дегенді білдіреді және деректер моделінің жоғары деңгейлі тұжырымдамалық диаграммасы болып табылады. ER моделі жақсы жобаланған мәліметтер базасын құру үшін деректерге қойылатын

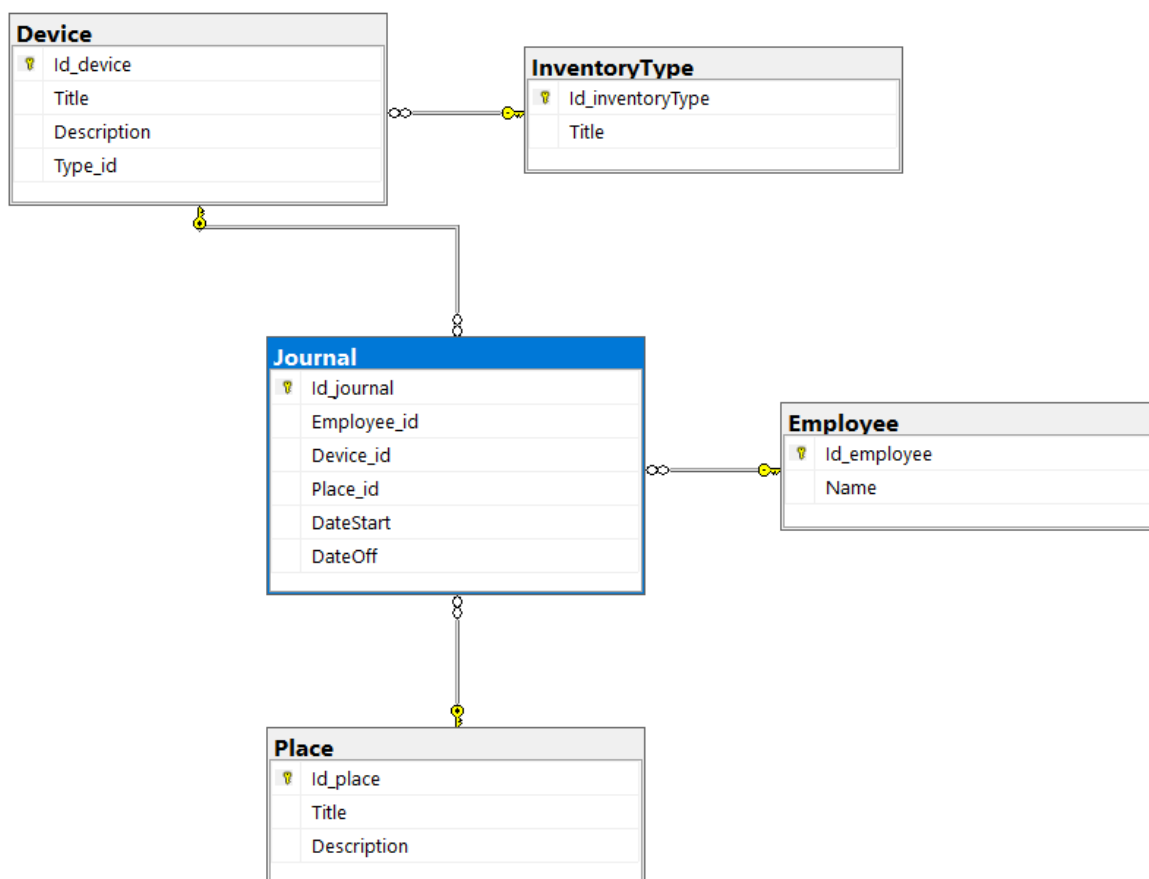
талаптарды жүйелі түрде талдауға көмектеседі. ER моделі нақты әлем объектілері мен олардың арасындағы қатынасты білдіреді. Деректер базасын іске асырмас бұрын ДҚБЖ-де ER моделін құру ұсынылады.

ER-модельдеу жақсы жобаланған мәліметтер базасын құру үшін деректерге қойылатын талаптарды жүйелі түрде талдауға көмектеседі. Осылайша, мәліметтер базасын жасамас бұрын ER модельдеуді аяқтау ұсынылады.

Төменде көрсетілген сурет бойынша ER диаграмма арқылы жобаның деректер қоры кестесінің жобасын көруімізге болады. Яғни біздің жобамыз нақты :

- InventoryType
- Device
- Journal
- Employee
- Place

атты таблицалар арқылы бір бірімен байланысқан.



3.2-сурет- Деректер қорының ER диаграммасы

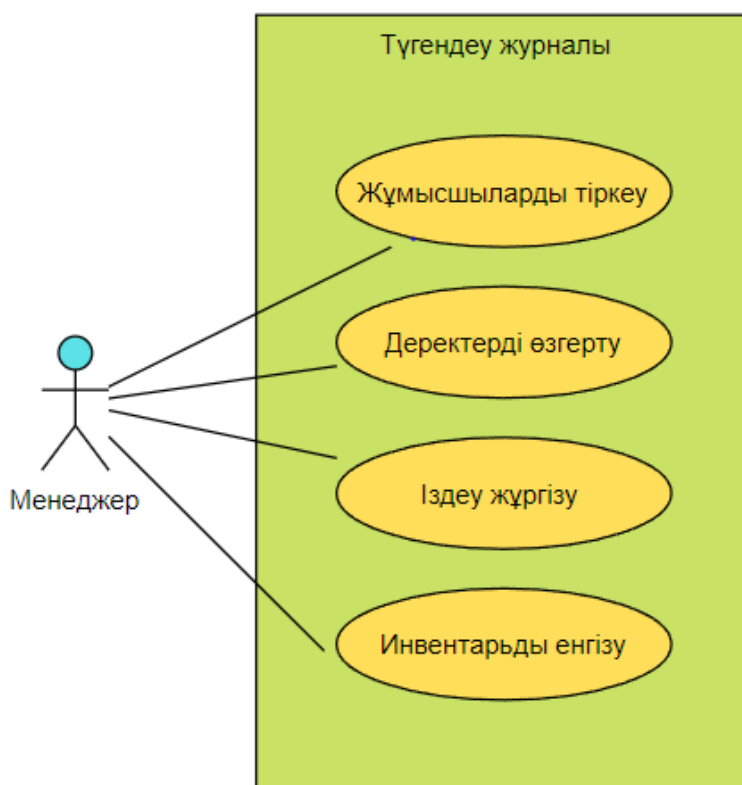
3.3 Прецеденттер диаграммасы

Пайдалану нұсқаларының диаграммасы (ағылш. case diagram) UML — де актерлер мен прецеденттер арасындағы қатынасты көрсететін және жүйені тұжырымдамалық деңгейде сипаттауға мүмкіндік беретін прецеденттер моделінің құрамдас бөлігі болып табылатын диаграмма.

Диаграмманың негізгі элементтері-қатысушы (actor) және прецедент .

Қатысушы-бұл прецеденттермен немесе субъектілермен (жүйе, ішкі жүйе немесе сынып) өзара әрекеттесу кезінде орындалатын логикалық байланысты рөлдердің жиынтығы. Қатысушы субъектіден тыс нәрсені білдіретін адам немесе басқа жүйе, ішкі жүйе немесе сынып бола алады. Графикалық түрде қатысушы "адам" ретінде бейнеленген.

Прецедент (use case) - қатысушы байқайтын нәтижеге әкелетін жүйе орындайтын көптеген жүйелі оқиғалардың (нұсқаларды қоса алғанда) сипаттамасы. Прецедент қатысушылар мен жүйенің өзара әрекеттесуін сипаттайтын субъектінің әрекетін білдіреді. Прецедент белгілі бір нәтижеге "қалай" қол жеткізілгенін көрсетпейді, тек "не" дәл орындалады. Прецеденттер өте қарапайым түрде белгіленеді - эллипс түрінде, оның ішінде оның атауы көрсетілген.



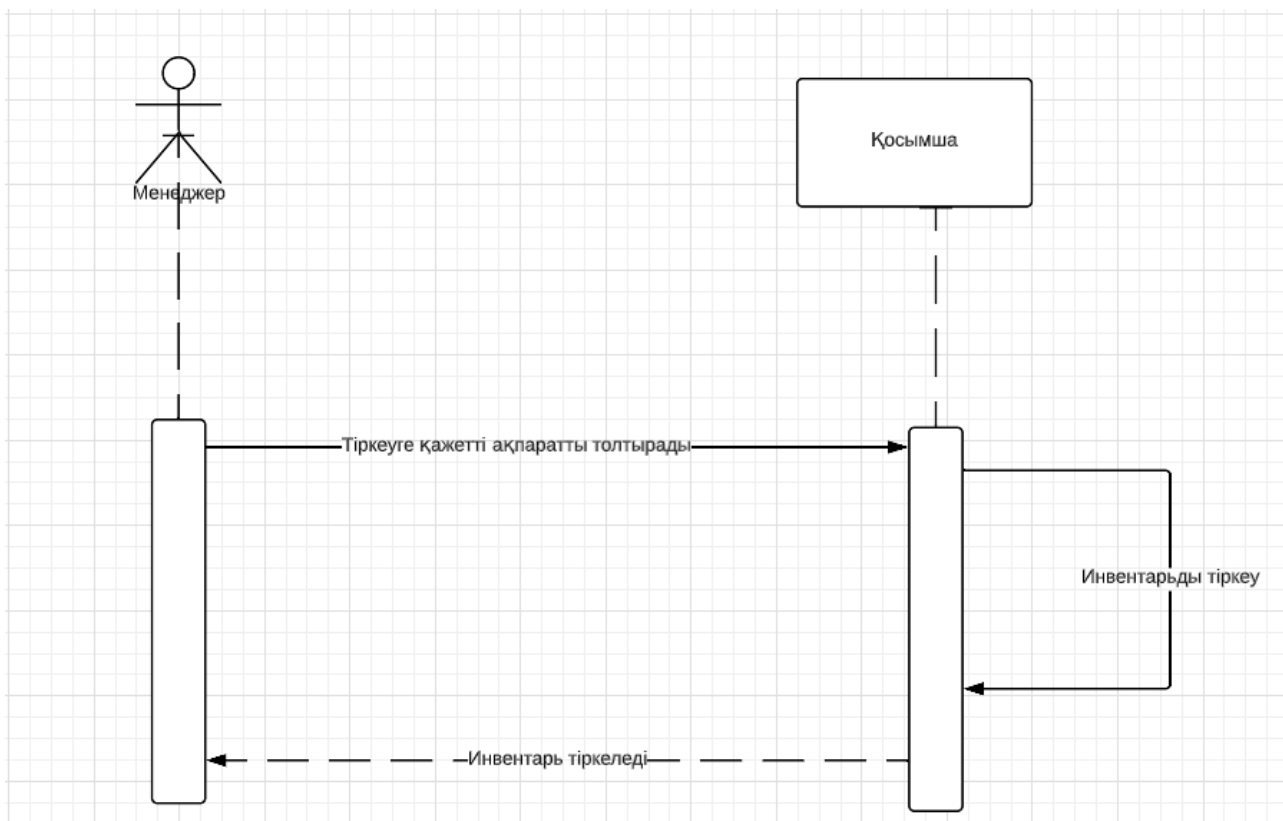
3.3-сурет- Қосымшаның прецеденттер диаграммасы арқылы көрсетілген жұмыс жасау принципі

3.4 Реттілік диаграммасы

Реттілік диаграммасы (ағылш. sequence diagram)-UML диаграммасы, онда бір уақыт осіндегі объектілердің кейбір жиынтығы үшін объектінің өмірлік циклі (белгілі бір нысанды құру-Қызмет-жою) және прецедент шеңберіндегі ақпараттық жүйенің актерлерінің (актерлерінің) өзара әрекеті көрсетілген.

Реттілік диаграммасының негізгі элементтері объектілердің белгілері (объектілердің атаулары бар тіктөртбұрыштар), тік "өмір сызықтары" (ағылш. lifeline), уақыт ағынын, объектінің қызметін немесе оның белгілі бір функцияны орындауын көрсететін тіктөртбұрыштарды (нүктелі "өмір сызығындағы" тіктөртбұрыштар) және объектілер арасындағы сигналдармен немесе хабарламалармен алмасуды көрсететін көрсеткілер.

Реттілік диаграммасы прецеденттік диаграммаларды нақтылау үшін қолданылады-жүйенің мінез-құлық аспектілерін сипаттайды. Реттілік диаграммасы уақыт өте келе нысандардың өзара әрекеттесуін көрсетеді. Бұл жағдайда ақпарат хабарлама түрін алады, ал объектілердің өзара әрекеттесуі сценарий аясында осы хабарламалармен алмасуды қамтиды.



3.4-сурет- Қосымшаның реттілік диаграммасы

4 Программалау бөлімі

4.1 Қосымшаны әзірлеу

Қосымшаны әзірлеу кезінде қосымшада қажетті функцияларды орындайтын көптеген сыныптар іске асырылды.

4.1.1 Main.cs

Бұл модуль қосымшаның негізгі беті болып табылады. Модульдің ішінде графикалық интерфейс (Windows forms) арқылы формачкалар жасалған . Форманың ішінде біз мынадай функцияларды орындауға мүмкіндік аламыз: журнал бойынша инвентарьды іздеу, тіркелген инвентарьды өзгерте аламыз, өшіре аламыз.

4.1.2 Addjournal.cs

Бұл модульдің ішінде біз жаңа инвентарьды тіркей аламыз, яғни бізге керекті ақпараттарды толтыру арқылы инвентарьды журналға тіркейміз. Егерде қандай да ақпаратты бос қалдырсақ жүйе бізге қате деген хабарлама жібереді. Ақпаратты толықтай толтырған жағдайда инвентарь сәтті тіркеледі.

4.1.3 Edit.cs

Бұл модуль арқылы тіркелген инвентарьдың қандай да бір ақпаратты қате толтырылғанын байқасақ сол инвентарьды өңдеп қайтадан тіркеуге мүмкіндік аламыз.

4.1.4 Employee.cs

Бұл модульде инвентарьға жауапты тұлғаларды енгізе аламыз . Жауапты тұлға жатақхананың менеджері немесе жатақхананың басқа да қызметкері болуы мүмкін. Жауапты тұлғаны тіркеп, өзгертіп және де өшіру сияқты функцияларды орындай аламыз.

4.1.5 Inventory.cs

Бұл модульдің ішінде инвентарьдың аты және типі сәйкестендіріп алдын ала толтыруға мүмкіндік аламыз. Әрбір инвентарьдың аты, типін толтырған кезде жанына инвентарьға сипаттамасын толтырып ақпаратты толықтыра аламыз.

4.1.6 Place.cs

Бұл модуль арқылы инвентарьды тіркейтін жерді алдын-ала тіркеп аламыз. Жаңадан инвентарьды тіркеу кезінде жеңіл болуы үшін. Бөлмені тіркеу кезінде жанына сипаттама толтырамыз.

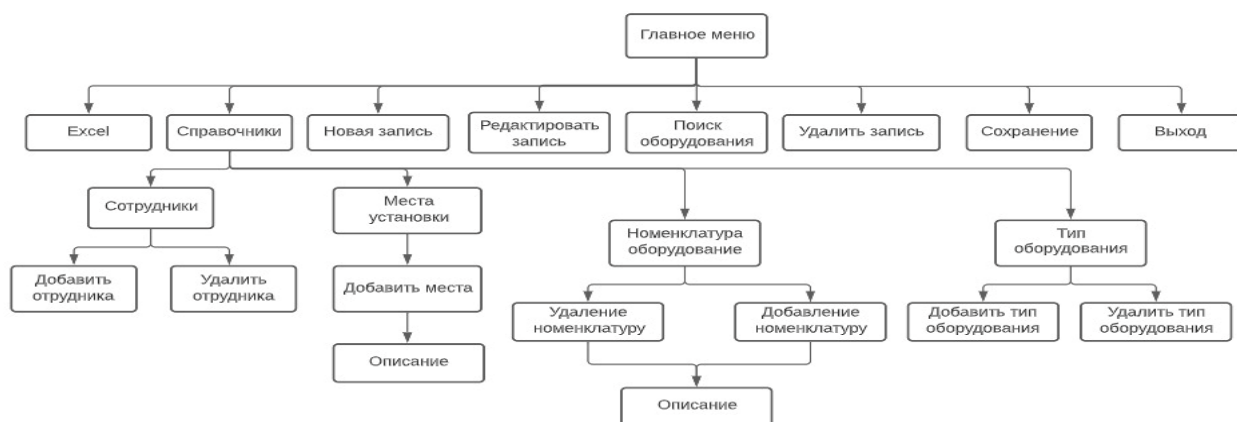
4.1.7 Type.cs

Бұл модульде инвентарьдың типтерімен жұмыс жасауға мүмкіндік аламыз. Типті алдын ала тіркеп, оны өшіріп сонымен қатар өзгертіге мүмкіндік аламыз.

4.2 Windows қосымшаның функционалды құрылымын жасау

Windows қосымша функционалды құрылымын жасау – қосымшаға кірген сәттен бастап, қосымшадан шыққан сәтке дейінгі қадамдар көрсету.

Қосымшаның функционалды құрылымын төмендегі суреттен көруге болады.



4.1-сурет- Қосымшаның функционалды құрылымы

4.3 Windows қосымшаның интерфейсі

Қосымшаға кірген кезде бізге ең алдымен осы терезе ашылады,яғни бұл қосымшаның негізгі беті. Бұл терезеде көріп тұрғандарыңыздай қосымшаға тіркелген инвентарьдың тізімін көре аламыз. Қандай да бір сипаты (тип устройств,устройстваның аты, жауапты тұлға, орналасқан жері, тіркелген күні) бойынша іздеу жұмыстарын жүргізе аламыз.

Программа Справочники

Инвентарный журнал

Для того чтобы не включать параметр в поиск оставьте поле пустым

Тип устройств Устройство Отв. лицо Место Дата учета Дата списания

ПОИСК

	Устройство	Отв. сотрудник	Место установки	Постановка на учет	Списание
▶	12569	▼ Tole Akerke	▼ 1	▼ Вт 10.05.22	Ср 11.05.22
	12569	▼ Zhanieva Aigerim	▼ 1	▼ Ср 11.05.22	
	12569	▼ Tole Akerke	▼ 1	▼ Ср 11.05.22	
	12567	▼ Tole Akerke	▼ 1	▼ Чт 12.05.22	
	12566	▼ Tole Akerke	▼ 1	▼ Чт 12.05.22	
	12568	▼ Zhanieva Aigerim	▼ 2	▼ Ср 18.05.22	
	12565	▼ Zhanieva Aigerim	▼ 3	▼ Ср 18.05.22	

НОВАЯ ЗАПИСЬ РЕДАКТИРОВАТЬ EXCEL СОХРАНИТЬ ВСЕ ИЗМЕНЕНИЯ ВЫХОД

4.2-сурет- Қосымшаның негізгі беті

Төменде келтірілген суреттер бойынша қосымшада инвентарьды әр түрлі сипатына байланысты іздеу (поиск) жүргізілген.

Программа Справочники

Инвентарный журнал

Для того чтобы не включать параметр в поиск оставьте поле пустым

Тип устройств Устройство Отв. лицо Место Дата учета Дата списания

Стол [] [] [] [] [] [] ПОИСК

Устройство	Отв. сотрудник	Место установки	Постановка на учет	Списание
▶ 12569	Tole Akerke	1	Вт 10.05.22	Ср 11.05.22
12569	Zhanieva Aigerim	1	Ср 11.05.22	
12569	Tole Akerke	1	Ср 11.05.22	

НОВАЯ ЗАПИСЬ РЕДАКТИРОВАТЬ EXCEL СОХРАНИТЬ ВСЕ ИЗМЕНЕНИЯ ВЫХОД

4.3-сурет- Инвентардың типі бойынша іздеу

Программа Справочники

Инвентарный журнал

Для того чтобы не включать параметр в поиск оставьте поле пустым

Тип устройств Устройство Отв. лицо Место Дата учета Дата списания

[] [] Zhanieva Aigerim [] [] [] ПОИСК

Устройство	Отв. сотрудник	Место установки	Постановка на учет	Списание
▶ 12569	Zhanieva Aigerim	1	Ср 11.05.22	
12568	Zhanieva Aigerim	2	Ср 18.05.22	
12565	Zhanieva Aigerim	3	Ср 18.05.22	

НОВАЯ ЗАПИСЬ РЕДАКТИРОВАТЬ EXCEL СОХРАНИТЬ ВСЕ ИЗМЕНЕНИЯ ВЫХОД

4.4-сурет- Жауапты тұлға бойынша іздеу

Программа Справочники

Инвентарный журнал

Для того чтобы не включать параметр в поиск оставьте поле пустым

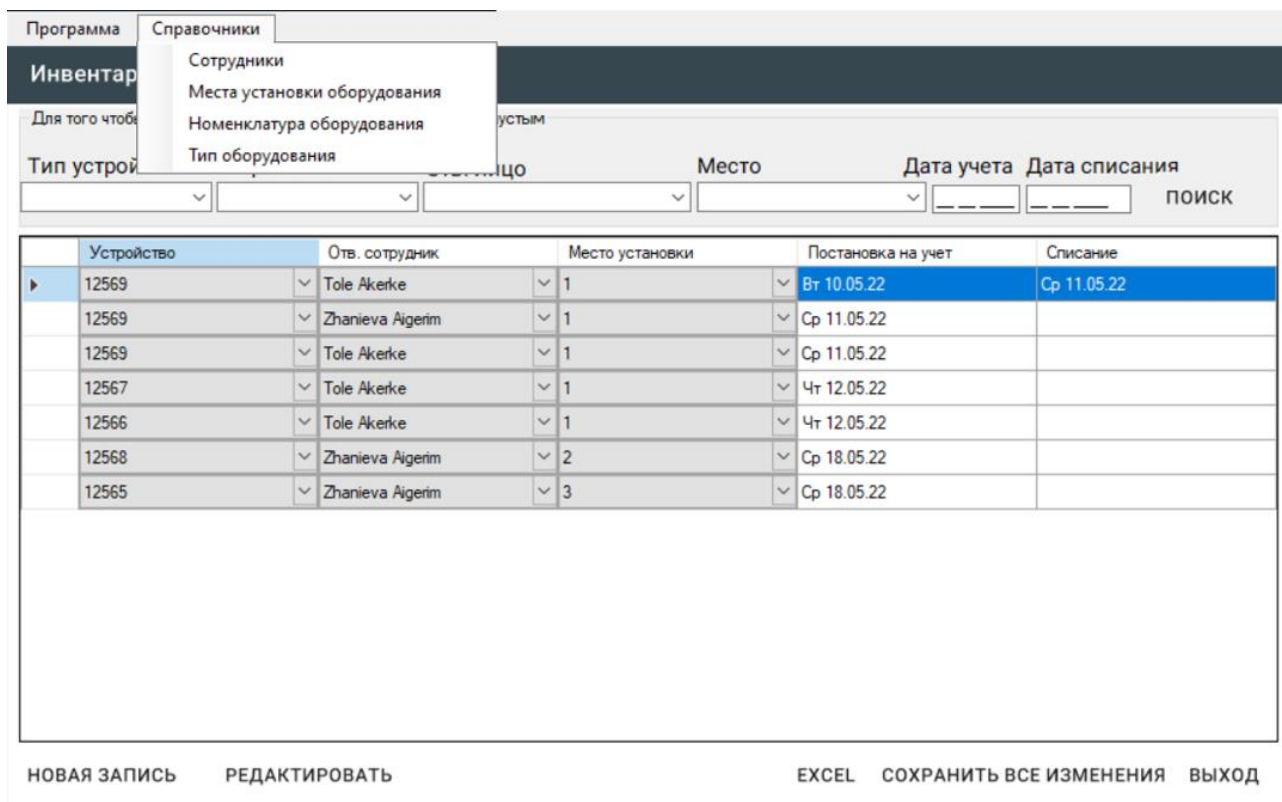
Тип устройств Устройство Отв. лицо Место Дата учета Дата списания

Устройство	Отв. сотрудник	Место установки	Постановка на учет	Списание
▶ 12569	▼ Tole Akerke	▼ 1	Вт 10.05.22	Ср 11.05.22

4.7-сурет-Тіркелген және есептен шыққан күні бойынша іздеу

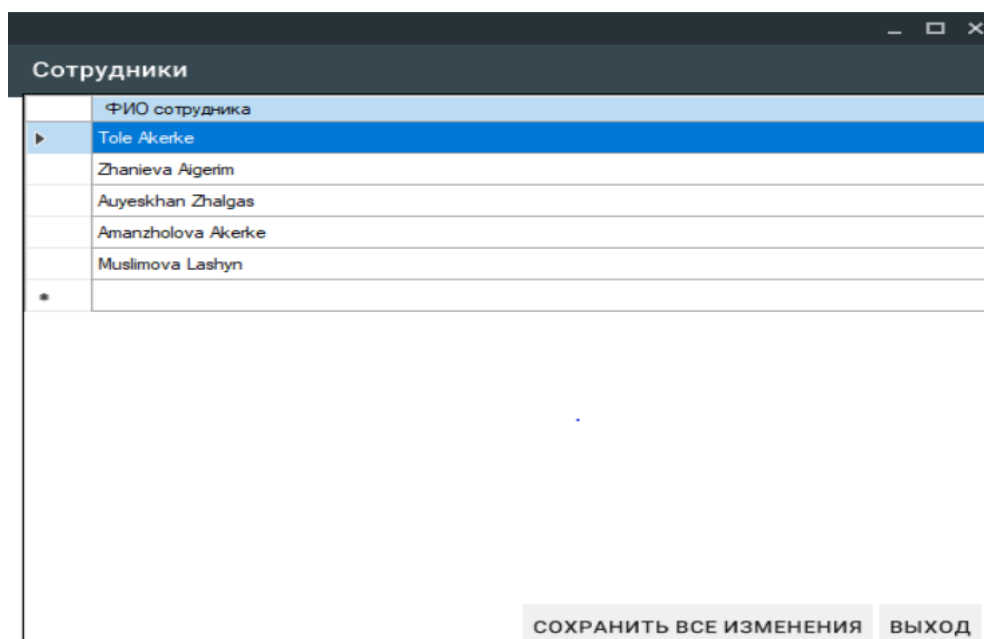
Төменде көрсетілген “Справочники” бөлімін басқан кезде бізге

- “Сотрудники” жауапты тұлғалардың тізімі;
- “Места установки оборудования” инвентарьдың орналастырылатын орындарының тізімі;
- “Номенклатура оборудования” инвентарьдың атын, типтерін көрсетеміз;
- “Тип оборудования” тіркелетін инвентарьдың типтерін алдын ала тіркеп аламыз;



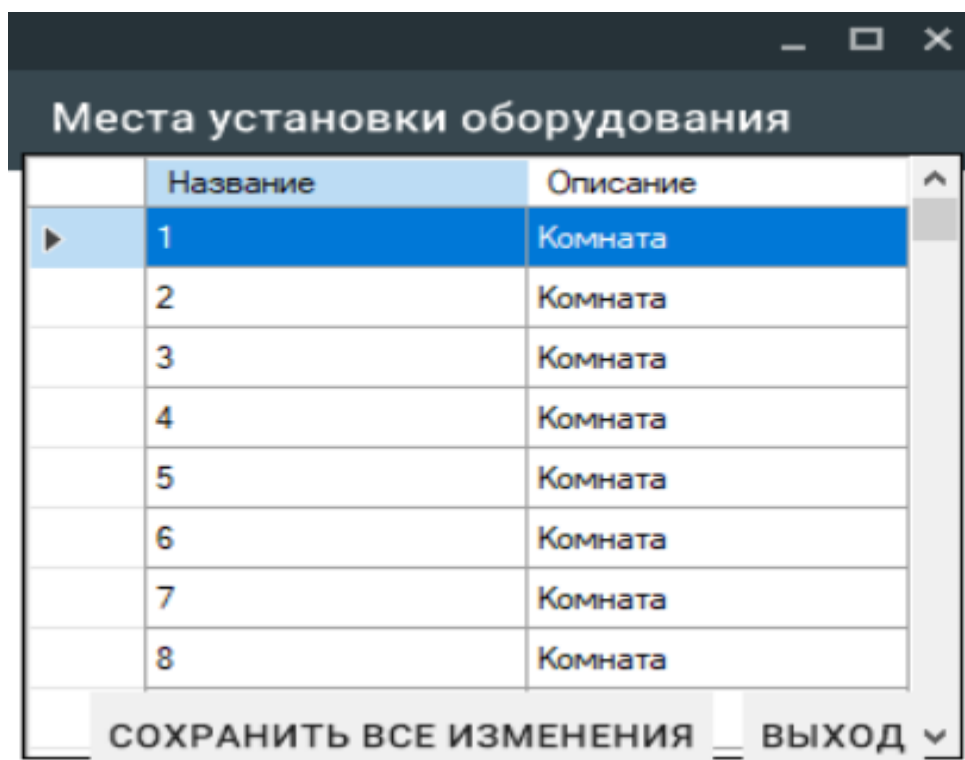
4.8-сурет- Справочники бөлімі

Сотрудники терезесін ашқан уақытта жауапты тұлғалардың тізімін көре аламыз. Сондай ақ жаңа қызметкерлерді енгізіп, қызметкерлерді тізімнен алып тастай аламыз.



4.9-сурет- Қызметкерлерді тіркеу терезесі

Места установки оборудования терезесін ашқан кезде бөлмелер мен олардың сипаттамасын көре аламыз.

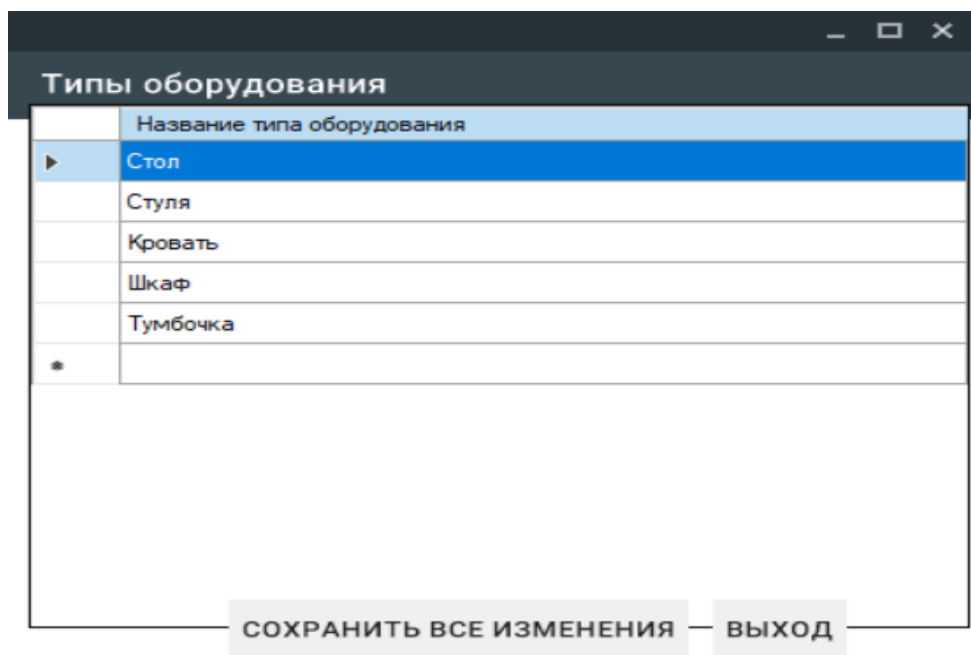


	Название	Описание
▶	1	Комната
	2	Комната
	3	Комната
	4	Комната
	5	Комната
	6	Комната
	7	Комната
	8	Комната

СОХРАНИТЬ ВСЕ ИЗМЕНЕНИЯ ВЫХОД

4.10-сурет- Инвентарьды тіркеу орны

Типы оборудования терезесінде инвентарьдың типтерін алдын ала бөліп тіркей аламыз.



	Название типа оборудования
▶	Стол
	Стуля
	Кровать
	Шкаф
	Тумбочка
*	

СОХРАНИТЬ ВСЕ ИЗМЕНЕНИЯ ВЫХОД

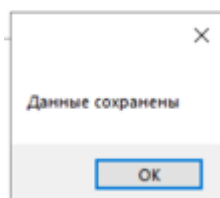
4.11-сурет- Инвентарьдың типін тіркеу терезесі

Инвентарьдың аты мен типін сәйкестендіріп , жанына сипаттамасын Инвентарь терезесінде толтыра аламыз.

Инвентарь			
	Код оборудования	Тип	Описание
▶	12569	Стол	Для жiтья
	12568	Стуля	Для жiтья
	12567	Кровать	Для жiтья
	12566	Шкаф	Для жiтья
	12565	Тумбочка	Для жiтья
*			

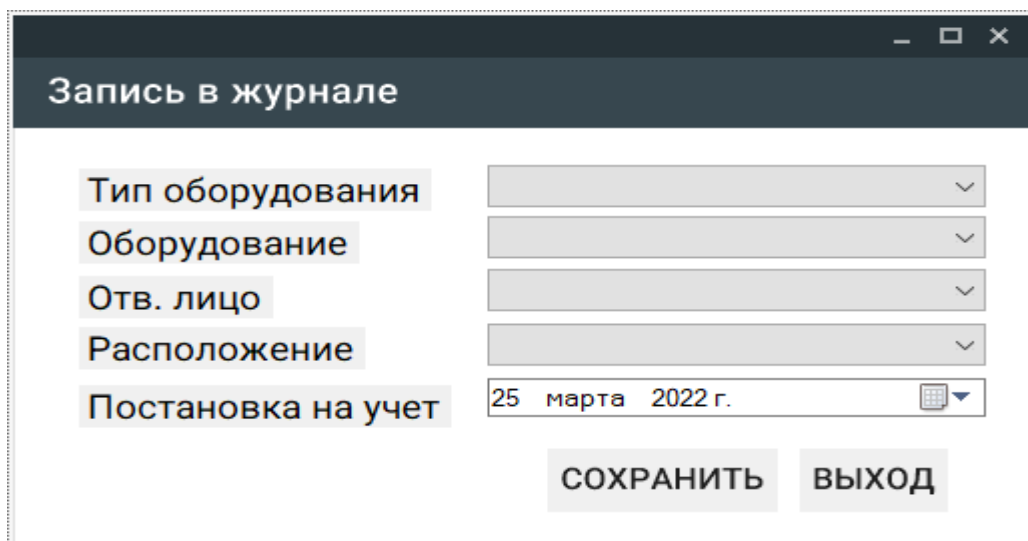
4.12-сурет- Инвентарьды код бойынша тіркеу

Ақпараттың барлығын дұрыс, толық енгізіп сақтаған жағдайда қосымша төмендегідей хабарлама көрсетеді.



4.13-сурет- Енгізілген ақпаратты сақтаған кездегі хабарлама

Жаңа инвентарьды тіркеу кезінде бізге төмендегідей терезе ашылады . Яғни ол жерде инвентарьдың типін, атын, жауапты тұлға және де орналасқан орнына байланысты ақпаратты толтырып инвентарьды тіркей аламыз. Тіркелген күні автоматты түрде шығып тұрады.



Запись в журнале

Тип оборудования

Оборудование

Отв. лицо

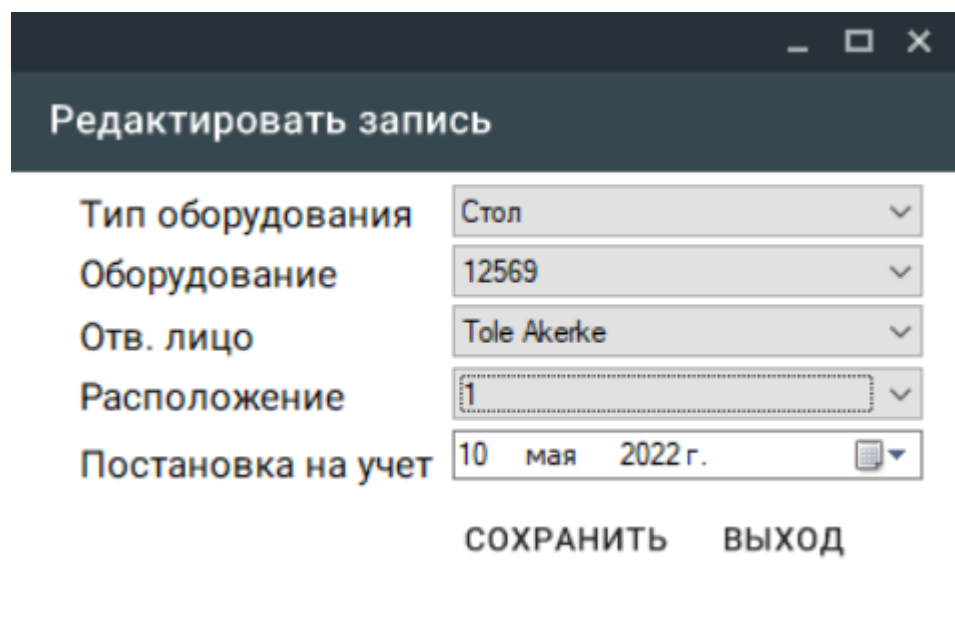
Расположение

Постановка на учет 25 марта 2022 г.

СОХРАНИТЬ **ВЫХОД**

4.14-сурет- Инвентарьды тіркеу терезесі

Инвентарьды тіркеу үстінде қандай да бір ақпаратты қате толтырған жағдайда инвентарьды қайта тіркеу жүргізе аламыз төмендегі терезеде көрсетілгендей.



Редактировать запись

Тип оборудования Стол

Оборудование 12569

Отв. лицо Толе Акеке

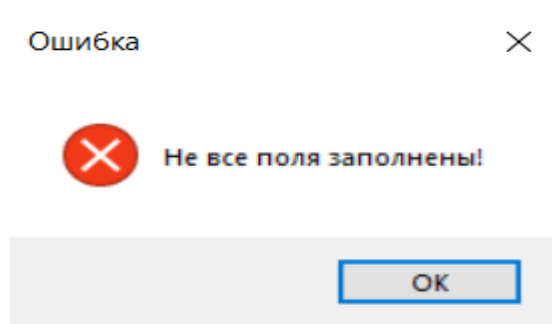
Расположение 1

Постановка на учет 10 мая 2022 г.

СОХРАНИТЬ **ВЫХОД**

4.15-сурет- Тіркелген инвентарьды өзгерту терезесі

Егерде қандай да бір ақпаратты толтырмай бос қалдырып, сақтаған жағдай да жүйе төмендегідей хабарлама жібереді. Ақпараттын бәрін толық толтырған кезде ғана инвентарь журналға тіркеледі.



4.16-сурет- Қателік болған жағдайда келетін хабарлама

Қорытынды

Жоғарыда көрсетілген міндеттер бойынша жобаны дайындадық. Жоба бойынша жатақхана жұмысының түгендеу журналы дайындалды. Әзірленген жоба бойынша жатақхана жұмысшылары мүлікті(инвентарьды) журналға тіркеп, олармен жұмыс жасауды жеңілдетеді. Яғни кез-келген тіркелген инвентарьды қандайда бір сипаты бойынша іздеу жүргізе алады. Инвентарьға қатысты ақпаратты кез-келген уақытта өзгертуге мүмкіндік береді. Инвентарьды жарамсыз болып қалған жағдайда есептен шығару функциясында оңай орындауға болады. Осы және бұлардан өзге қосымша функционалдарға ие “Түгендеу журналы” Windows қосымшасы жатақхана қызметкерлерінің жұмысын оңтайландырады деген ойдамын. Windows қосымша құжаттарды толтыру мен растау уақытын азайтады, толтыру барысында кездесетін қателіктерді болдырмайды, қағазбастылықты жояды. Осылайша жатақхана қызметкерлерінің жұмысын жеңілдетуге мүмкіндік береді.

Жоба Visual Studio ортасында C# бағдарламау тілі арқылы жасалынды.

Жобаны жасау кезінде жатақхана мүлкіне байланысты көптеген ақпараттар жиналды.

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 SQL және дерекқор сабақтары // сайттағы электрондық нұсқа
<http://www.site-do.ru/db/db4.php>
- 2 Деректер базасымен жұмыс // сайттағы электрондық нұсқа
<http://www.intuit.ru/studies/courses/3439/681/lecture/14017>
- 3 SQL тілінің түсінігі және оның негізгі бөліктері // сайттағы электрондық нұсқа
http://www.intuit.ru/studies/higher_education/3407/courses/364/lecture/8659?page=2
- 4 MySQL ДҚБЖ-ге кіріспе // сайттағы электрондық нұсқа
<http://www.intuit.ru/studies/courses/111/111/lecture/3246?page=3>
- 5 MySQL Workbench-пен жұмыс істеу негіздері: жылдам бастау, деректер схемасын басқару // сайттағы электрондық нұсқа
<http://mithrandir.ru/professional/soft-and-hardware/mysql-workbench-basics.html>
- 6 Learning Web Design, Fifth Edition; O'Reilly; Jennifer Niederst Robbins; 2018; 810 с.; ISBN: 978-1-491-96020-2
- 7 Хомоненко А.Д., Цыганков В. М., Мальцев М. Г., мәліметтер базасы: жоғары оқу орындарына арналған оқулық / ред. проф. А. Д. Хомоненко. — 6-шы басылым., доп. - СПб.: Тәж-ғасыр, 2009. - 736 б., ISBN 978-5-7931-0527-9
- 8 Гольцман В. MySQL 5.0. Программист кітапханасы: Питер; Санкт-Петербург; 2010, ISBN 978-5-49807-135-0
- 9 SQL сабақтары және мәліметтер базасы. // Сайттағы электрондық нұсқа
<http://www.site-do.ru/db/db.php>
- 10 Жылдам бастау: MySQL Workbench-те визуалды мәліметтер базасын жобалау // сайттағы электрондық нұсқа
<http://www.pvsm.ru/mysql/31556/index.html>
- 11 Библия С#. 3- е изд.; Фленов М. Е.; СПб.: БХВ-Петербург; 2016; 546 с.; ISBN 97 8-5-9775-3533-5
- 12 С# бағдарламалау тіліне сабақтар
<https://metanit.com/sharp/aspnet5/29.1.php>
- 13 Әр түрлі программалау тілдерін үйренуге мүмкіндік беретін сайт
<https://w3schools.com>
- 14 Daniel Roth, Rick Anderson, and Shaun Luttin “Overview to ASP.NET Core” – Лондон, 2017. – 20 б.
- 15 Ақпараттық жүйелер мен технологияларды жобалаудың әдістері мен құралдары: оқу құралы / С.Ю.Петрова. - Ставрополь: Логос, 2018. - 114б.
- 16 ASP.NET и его архитектура [Электрондық ресурс]. Сілтеме: <https://www.guru99.com/what-is-asp-dot-net.html>
- 17 Ақпараттық жүйелер мен технологиларды жобалаудың әдістері мен құралдары: оқу құралы / С.Ю.Петрова-Ставрополь: Логос,2018. -114б
- 18 Анализ и проектирование на UML // Новиков Ф.А-Санкт-Петербург,2007.-66
- 19 “Overview to ASP.NET Core” / Daniel Roth, Rick Anderson, and Shaun Luttin – Лондон, 2017.-40б

20 Язык UML.Руководство пользователя.//Гради Буч.Джеймс Рамбо.Ивар
Якобсан.-Москва,2006.-196

А қосымшасы (Міндетті)

Техникалық тапсырма

А.1 Жалпы сипаттама

“Түгендеу журналы ” Windows қосымшасы жатақхана инвентарын түгендеуге қажетті қосымша.

А.2 Терминдер және қысқартулар

Төменде көрсетілген 2-кестеде жоба мәтінінде қолданылған барлық терминдер мен қысқартулар және жобаны бағдарламалық дайындау кезінде пайдаланылатын технологиялар анықтамалары көрсетілген.

1-кесте- Терминдер және қысқартулар

Аббревиатура немесе термин	Анықтама
Visual Studio	бұл кодты жазуға, жөндеуге және құруға, сондай-ақ қосымшаларды жариялауға арналған іске қосу алаңы
C#	қазіргі заманғы объектіге бағытталған және типті қауіпсіз бағдарламалау тілі. C# тілі әзірлеушілерге .NET жүйесінде жұмыс істейтін қауіпсіз және сенімді қолданбалардың әртүрлі түрлерін жасауға мүмкіндік береді.
SQL	қарапайым сөзбен айтқанда, бұл құрылымдық сұранысты программалау тілі (SQL, Structured Query Language), ол деректерді сақтаудың, оның бөліктерін іздеудің, жаңартудың, мәліметтер базасынан алудың және жоюдың тиімді әдісі ретінде қолданылады.
Фреймворк	ІТ дағы дайын үлгі, бланкі, бағдарламалық платформаға арналған үлгі, оның негізінде сіз өзіңіздің кодыңызды қоса аласыз
Қосымша, қолданба	белгілі бір пайдаланушылық тапсырмаларды орындауға және пайдаланушымен тікелей қарым-қатынас жасауға арналған бағдарлама.

А қосымшасының жалғасы

1-кесте жалғасы

Бағдарламалау тілі	орындаушы үшін (мысалы, компьютер) бағдарламалар жазуға арналған формальды тіл.
API	Application Programming Interface, сұраулар қабылдайтын және жауаптар жіберетін сервердің құрамдас бөлігі.

А.3 Жүйенің жұмыс істеуін ұйымдастыруға қойылатын талаптар

“Түгендеу журналы” қосымшасы қызметкерлердің компьютерінде орнатылуы тиіс. Орнатылғаннан кейін пайдаланушылар сыртқы түгендеу журналынан қолмен енгізу, көшіру арқылы мүлікті жүйеге тіркей алады. Инвентарды тіркеп, журнал бойынша түгендегеннен кейін ақпаратты журнал түрінде Excel форматында экспорттай алады.

А.4 Сыртқы ортаның сипаттамасы

Менің жүйем үшін сыртқы орта жүзеге асыратын арнайы талаптар жоқ.

А.5 Техникалық құралдар кешеніне қойылатын талаптар

Қосымшамен жұмыс істеу үшін мынадай параметрлері бар компьютер талап етіледі: Жұмыс жиілігі 166 МГц-тен кем емес Intel Pentium-дан төмен емес процессор, жедел жад 32 Мбайттан төмен емес, графикалық карта 1 Мбайттан төмен емес, қатты дискідегі орын кемінде 100 Мбайт, USB немесе Floppy 3.5 дюйм.

А.6 Бағдарламалық құралдарға қойылатын талаптар

Қосымшаның жұмысы үшін пайдаланушының компьютерінде Microsoft Windows, сондай-ақ компьютерде орнатылған Visual Studio 2022 және MS SQL Server орнатылуы тиіс. Жүйені жазу үшін бағдарламалау тілдері мен деректерді таңбалау қолданылды:

А қосымшасының жалғасы

- C#.
- SQL.
- .NET Framework

А.7 Жүйені пайдаланушылар

Бұл жүйенің пайдаланушылары жатақхана менеджерлері, олардың көмекшілері, сондай-ақ жатақхана ұйымының директоры болып табылады. Менеджерлер мен олардың көмекшілері құжаттар жасайды және оны тексерістен өткізеді.

А.8 Жүйе функцияларына қойылатын талаптар

Жүйенің функциялары бірнеше түрге бөлінеді. Олар: Ақпаратты енгізу-бұл құжаттарды жасау, көшіру, импорттау функциялары. Деректерді қарау және экспорттау-басып шығару, қарау, үзінділерді экспорттау, үзінділерді қарау және басып шығару.

Б қосымшасы (Міндетті)

Бағдарлама мәтіні

AddJournal.cs

```
using InventoryProject;
using InventoryProject.JournalDataSetTableAdapters;
using MaterialSkin;
using MaterialSkin.Controls;
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 InventoryApp
{
    public partial class AddJournal : MaterialForm
    {
        private BindingSource bindingSource1;
        private JournalDataSet journalDataSet;
        private JournalTableAdapter journalTableAdapter;
        private ComboBox type_cmbx;
        private BindingSource inventoryTypeBindingSource;
        private InventoryTypeTableAdapter inventoryTypeTableAdapter;
        private MaterialLabel label1;
        private ComboBox device_cmbx;
        private MaterialLabel label2;
        private ComboBox empl_cmbx;
        private MaterialLabel label3;
        private ComboBox place_cmbx;
        private MaterialLabel label4;
        private BindingSource foreignKeyConstraintTypeDeviceBindingSource;
        private DeviceTableAdapter deviceTableAdapter;
        private BindingSource employeeBindingSource;
        private EmployeeTableAdapter employeeTableAdapter;
        private BindingSource placeBindingSource;
        private PlaceTableAdapter placeTableAdapter;
        private DateTimePicker dateTimePicker1;
        private MaterialLabel label5;
        private MaterialFlatButton button2;
        private MaterialFlatButton save_btn;
        public AddJournal()
        {
```

Б қосымшасының жалғасы

```
InitializeComponent();
var materialSkinManager = MaterialSkinManager.Instance;
materialSkinManager.AddFormToManage(this);
materialSkinManager.Theme = MaterialSkinManager.Themes.LIGHT;
materialSkinManager.ColorScheme = new ColorScheme(Primary.BlueGrey800,
Primary.BlueGrey900, Primary.BlueGrey500,
Accent.LightBlue200, TextShade.WHITE);
}
private void AddNewJournal_Load(object sender, EventArgs e)
{
    this.placeTableAdapter.Fill(this.journalDataSet.Place);
    this.employeeTableAdapter.Fill(this.journalDataSet.Employee);
    this.deviceTableAdapter.Fill(this.journalDataSet.Device);
    this.inventoryTypeTableAdapter.Fill(this.journalDataSet.InventoryType);
    this.journalTableAdapter.Fill(this.journalDataSet.Journal);
    this.empl_cmbx.SelectedItem = (object)null;
    this.device_cmbx.SelectedItem = (object)null;
    this.place_cmbx.SelectedItem = (object)null;
    this.type_cmbx.SelectedItem = (object)null;
}
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
private void save_btn_Click(object sender, EventArgs e)
{
    if (this.type_cmbx.SelectedItem != null && this.device_cmbx.SelectedItem != null
    && this.empl_cmbx.SelectedItem != null && this.place_cmbx.SelectedItem != null)
    {
        JournalDataSet.JournalRow row = (this.bindingSource1.AddNew() as
DataRowView).Row as JournalDataSet.JournalRow;
        row.Place_id = (int)this.place_cmbx.SelectedValue;
        row.Employee_id = (int)this.empl_cmbx.SelectedValue;
        row.Device_id = (int)this.device_cmbx.SelectedValue;
        row.DateStart = this.dateTimePicker1.Value.Date;
        this.bindingSource1.EndEdit();
        try
        {
            this.journalTableAdapter.Update(this.journalDataSet.Journal);
            this.Close();
        }
        catch (Exception ex)
        {
            int num = (int)MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Hand);
        }
    }
    Else
    {
```

Б қосымшасының жалғасы

```
        int num1 = (int)MessageBox.Show("Не все поля заполнены!", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
}
}
Employee.cs
using InventoryProject;
using InventoryProject.JournalDataSetTableAdapters;
using MaterialSkin.Controls;
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 InventoryApp
{
    public partial class Employee : MaterialForm
    {
        private BindingSource bindingSource1;
        private JournalDataSet journalDataSet;
        private EmployeeTableAdapter employeeTableAdapter;
        private DataGridView dataGridView1;
        private MaterialFlatButton save_btn;
        private MaterialFlatButton button2;
        private DataGridViewTextBoxColumn idemployeeDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn nameDataGridViewTextBoxColumn;
        private MaterialContextMenuStrip contextMenuStrip1;
        private MaterialToolStripMenuItem удалитьToolStripMenuItem;
        public Employee()
        {
            InitializeComponent();
        }
        private void EmployeeForm_Load(object sender, EventArgs e)
        {
            this.employeeTableAdapter.Fill(this.journalDataSet.Employee);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void save_btn_Click(object sender, EventArgs e){
            try
            {
                this.bindingSource1.EndEdit();
                this.employeeTableAdapter.Update(this.journalDataSet.Employee);
            }
        }
    }
}
```

Б қосымшасының жалғасы

```
        int num = (int)MessageBox.Show("Данные сохранены");
    }
    catch (Exception ex)
    {
        int num = (int)MessageBox.Show(ex.Message);
    }
}
private void удалитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if ((uint)this.dataGridView1.SelectedRows.Count <= 0U)
        return;
    this.bindingSource1.Remove(this.dataGridView1.SelectedRows[0].DataBoundItem);
}
}
```

Inventory.cs

```
using InventoryProject;
using InventoryProject.JournalDataSetTableAdapters;
using MaterialSkin.Controls;
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 InventoryApp
{
    public partial class Inventory : MaterialForm
    {
        private MaterialFlatButton button2;
        private MaterialFlatButton save_btn;
        private DataGridView dataGridView1;
        private MaterialContextMenuStrip contextMenuStrip1;
        private MaterialToolStripMenuItem удалитьToolStripMenuItem;
        private BindingSource bindingSource1;
        private JournalDataSet journalDataSet;
        private DeviceTableAdapter deviceTableAdapter;
        private JournalDataSet journalDataSet1;
        private BindingSource journalDataSet1BindingSource;
        private BindingSource inventoryTypeBindingSource;
        private InventoryTypeTableAdapter inventoryTypeTableAdapter;
        private DataGridViewTextBoxColumn iddeviceDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn titleDataGridViewTextBoxColumn;
        private DataGridViewComboBoxColumn typeidDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn descriptionDataGridViewTextBoxColumn;
        public Inventory()
        {

```


Б қосымшасының жалғасы

```
InitializeComponent();
var topLeftHeaderCell = dataGridView1.TopLeftHeaderCell;
}
private void InventoryForm_Load(object sender, EventArgs e)
{
    this.inventoryTypeTableAdapter.Fill(this.journalDataSet1.InventoryType);
    this.deviceTableAdapter.Fill(this.journalDataSet.Device);
}
private void save_btn_Click(object sender, EventArgs e)
{
    try
    {
        this.bindingSource1.EndEdit();
        this.deviceTableAdapter.Update(this.journalDataSet.Device);
        int num = (int)MessageBox.Show("Данные сохранены");
    }
    catch (Exception ex)
    {
        int num = (int)MessageBox.Show(ex.Message);
    }
}
private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

private void удалитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if ((uint)this.dataGridView1.SelectedRows.Count <= 0U)
        return;
    this.bindingSource1.Remove(this.dataGridView1.SelectedRows[0].DataBoundItem);
}
}
}
```

Place.cs

```
using InventoryProject;
using InventoryProject.JournalDataSetTableAdapters;
using MaterialSkin.Controls;
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 InventoryApp
{
```

Б қосымшасының жалғасы

```
public partial class Place : MaterialForm
{
    private MaterialFlatButton button2;
    private MaterialFlatButton save_btn;
    private DataGridView dataGridView1;
    private MaterialContextMenuStrip contextMenuStrip1;
    private MaterialToolStripMenuItem удалитьToolStripMenuItem;
    private BindingSource bindingSource1;
    private JournalDataSet journalDataSet;
    private PlaceTableAdapter placeTableAdapter;
    private DataGridViewTextBoxColumn idplaceDataGridViewTextBoxColumn;
    private DataGridViewTextBoxColumn titleDataGridViewTextBoxColumn;
    private DataGridViewTextBoxColumn descriptionDataGridViewTextBoxColumn;
    public Place()
    {
        InitializeComponent();
    }
    private void PlaceForm_Load(object sender, EventArgs e)
    {
        this.placeTableAdapter.Fill(this.journalDataSet.Place);
    }

    private void save_btn_Click(object sender, EventArgs e)
    {
        try
        {
            this.bindingSource1.EndEdit();
            this.placeTableAdapter.Update(this.journalDataSet.Place);
            int num = (int)MessageBox.Show("Данные сохранены");
        }
        catch (Exception ex)
        {
            int num = (int)MessageBox.Show(ex.Message);
        }
    }
    private void button2_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void удалитьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if ((uint)this.dataGridView1.SelectedRows.Count <= 0U)
            return;
        DataGridViewRow selectedRow = this.dataGridView1.SelectedRows[0];
    }
}
}
Type.cs
using InventoryProject;
using InventoryProject.JournalDataSetTableAdapters;
```

Б қосымшасының жалғасы

```
using MaterialSkin.Controls;
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 InventoryApp
{
    public partial class Type : MaterialForm
    {
        private BindingSource bindingSource1;
        private JournalDataSet journalDataSet;
        private MaterialFlatButton button2;
        private MaterialFlatButton save_btn;
        private DataGridView dataGridView1;
        private MaterialContextMenuStrip contextMenuStrip1;
        private MaterialToolStripMenuItem удалитьToolStripMenuItem;
        private InventoryTypeTableAdapter inventoryTypeTableAdapter;
        private DataGridViewTextBoxColumn
idinventoryTypeDataGridViewTextBoxColumn;
        private DataGridViewTextBoxColumn titleDataGridViewTextBoxColumn;
        public Type()
        {
            InitializeComponent();
        }
        private void TypeForm_Load(object sender, EventArgs e)
        {
            this.inventoryTypeTableAdapter.Fill(this.journalDataSet.InventoryType);
        }

        private void save_btn_Click(object sender, EventArgs e)
        {
            try
            {
                this.bindingSource1.EndEdit();
                this.inventoryTypeTableAdapter.Update(this.journalDataSet.InventoryType);
                int num = (int)MessageBox.Show("Данные сохранены");
            }
            catch (Exception ex)
            {
                int num = (int)MessageBox.Show(ex.Message);
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {

```

Б қосымшасының жалғасы

```
    this.Close();
}
private void удалитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if ((uint)this.dataGridView1.SelectedRows.Count <= 0U)
        return;
    this.bindingSource1.Remove(this.dataGridView1.SelectedRows[0].DataBoundItem);
}
}
}
```