

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Серікова Камила Арманқызы

Азық-түлік дүкені тауарларын есепке алу үшін мобильді қосымша құру

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

А.Н.Молдагулова

" 27 " 05 2022 ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: «Азық-түлік дүкені тауарларын есепке алу үшін мобильді қосымша құру»

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

Орындаған

Серікова К.А.

Рецензент

Ғылыми жетекші

техн. ғыл. канд., доцент

лектор

" 30 " 05 2022 ж. Д.М.Ескендирова

" 25 " 05 2022 ж. Т.Құрметқан

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



БЕКІТЕМІН

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд. профессор

М.Н. Молдагулова А.Н.Молдагулова

"22" "05" 2022 ж.

Дипломдық жобаны орындауға

ТАПСЫРМА

Білім алушыға *Серікова Камила Арманқызы*

Тақырыбы: «Азық-түлік дүкені тауарларын есепке алу үшін мобильді қосымша құру»

Академиялық мәселелер жөніндегі проректор бұйрығының № 429-П/6 "24" 12 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі: "01" "06" 2022 ж.

Дипломдық жобаның бастапқы мәліметтері: *Жобаның тәлқұжаты, 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 және есептің қойылымын жасау;*
- b) жобаны жобалау және пәндік сала бойынша талдау;*
- в) пайдаланушы интерфейсін жобалау және дамыту;*
- г) бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу;*
- д) А Қосымшасы – техникалық тапсырма;*
- е) Б Қосымшасы – бағдарлама мәтіні.*

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):

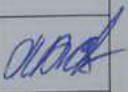
презентацияның 30 слайдпен берілген құжат түрінде ұсынылған.

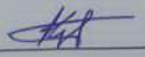
Ұсынылған негізгі әдебиеттер: 32 пайдаланылған әдебиеттер тізімінен

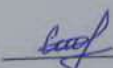
Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жобаның жоспарын күру.	14.01.2022	орындалды
2. Тапсырма қойылымы және бағдарламалау 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. Жобаның бөліміне сызбаларды дайындау.	15.02.2022	орындалды
5. Жобаның веб-қосымшасын тестілеуден 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	01.06.22	
Бағдарламалық бөлім	Марғұлан Қ. техн.ғыл.магистрі, лектор	25.05.22	

Ғылыми жетекші  Құрметқан Т

Тапсырманы орындауға қабылдап алған студент  Серікова К.А.

Күні «17» 11 2021 ж.

АҢДАТПА

Дипломдық жобада Android платформасына негізделген азық-түлік дүкені тауарларын есепке алатын және кіріс-шығысын бақылайтын мобильді қосымша құрылған. Құрылған жүйе компаниядағы тауарларды есепке алуды оңтайландыруға және жеңілдетуге, олардың компаниядағы қозғалысын бақылауға және ыңғайлы статистика жүргізуге жағдай жасайды. Сонымен бірге деректерді енгізу және редакциялау кезінде қосымша енгізілетін ақпараттың дұрыстығын бақылау, пайдаланушылар тарапынан қате ісәрекеттерді болдырмау және алдын алу мүмкіндігі қарастырылған.

Android қосымшасы Java бағдарламалау тілін қолдану арқылы жүзеге асырылған. Құрал ретінде IntelliJ Idea 15 ортасы таңдалған. Деректер қорын басқару жүйесі ретінде MySQL сервері қолданылған.

Дипломдық жұмыс 5 бөлімнен тұрып, мобильді қосымшаларды қолданудың ерекшеліктері мен артықшылықтары талданып, жүйені жобалау және құру барысы баяндалған. Сондай-ақ қолданушылар үшін қосымшаны пайдалануға нұсқаулық берілген.

АННОТАЦИЯ

В дипломном проекте создано мобильное приложение для учета и контроля за входом и выходом товаров продуктового магазина на базе платформы Android. Созданная система позволит оптимизировать и упростить учет товаров в компании, отслеживать их перемещение в компании и вести удобную статистику. Также предусмотрена возможность контроля достоверности дополнительно вводимой информации при вводе и редактировании данных, предотвращения и устранения некорректных действий со стороны пользователей.

Приложение для Android реализовано с использованием языка программирования Java. В качестве инструмента выбрана среда IntelliJ Idea 15. В качестве системы управления базами данных использовался сервер MySQL.

Дипломная работа состоит из 5 разделов, в которых проанализированы особенности и преимущества использования мобильных устройств, изложен ход проектирования и создания системы. Также для пользователей дана инструкция по использованию приложения.

ANNOTATION

The diploma project created a mobile application for accounting and monitoring the entry and exit of grocery store products based on the Android platform. The created system will optimize and simplify the accounting of goods in the company, track their movement in the company and keep convenient statistics. It is also possible to control the reliability of additionally entered information when entering and editing data, to prevent and eliminate incorrect actions on the part of users.

The Android application is implemented using the Java programming language. The IntelliJ Idea 15 environment is selected as the tool. The MySQL server was used as a database management system.

The thesis consists of 5 sections in which the features and advantages of using mobile devices are analyzed, the course of designing and creating the system is outlined. There are also instructions for users to use the application.

МАЗМҰНЫ

	Кіріспе	9
1	Әдебиетке шолу	11
1.1	Мобильді қосымшаның мақсаты мен міндеттері	11
2	Логикалық модельдеу	23
2.1	Модельдеу әдістері мен құралдарын таңдау	23
2.2	Пайдалану диаграммаларын әзірлеу	24
2.3	Талдау сыныптарын сәйкестендіру	28
2.4	Даму тақырыбының сипаттамасы	29
2.5	Нысандардың өзара әрекеттесуі және экрандық нысандар	30
2.6	Логикалық деректер моделі	35
2.7	Даму тақырыбының статикалық моделі	37
3	Жобалау бөлімі	39
3.1	Даму ортасын, бағдарламалау тілін және даму құралдарын таңдау	39
3.2	Деректердің физикалық моделі	39
3.3	Әзірлеу пәнінің компоненттері	41
3.4	Әзірлеу пәнін орналастыру	43
4	Бағдарламалық жасақтаманы іске асыру және тестілеу	45
4.1	Бағдарламалық жасақтама компоненттерінің мақсаты мен сипаттамасы	45
4.2	Бағдарламалық жасақтаманы тестілеу	48
5	Пайдаланушы нұсқаулығы	49
5.1	Бағдарламаны орнату және конфигурациялау бойынша нұсқаулық	49
5.2	Кез келген пайдаланушы рөлінде жұмыс істеу	49
	Қорытынды	56
	Пайдаланылған әдебиеттер тізімі	57
	А қосымшасы	59
	Б қосымшасы	62

КІРІСПЕ

Бағдарламалық жасақтаманың қарқынды дамуы тез дамып келе жатқан смартфон нарығымен бірге қоғамды Ақпараттық технологиялар қоғамның дамуын анықтайтын және қамтамасыз ететін сапалы жаңа кезеңге жақындатты. Енді бағдарламалық жасақтама мен заманауи технологияларды пайдаланбай Бизнесінің бірде-бір сегментін елестету мүмкін емес, ал нарықта смартфондардың пайда болуымен жаңа технологияларды қолдану және енгізу тезірек және ыңғайлы бола бастады.

Кең мағынада, тауарларды есепке алудың мобильді қосымшасы – бұл бағдарламалық жасақтама, оның функциялары компаниядағы тауарларды есепке алу, жаңа өнімдерді қосу, статистиканы қарау, пайдаланушыларға ыңғайлы және оңай игерілетін интерфейсті ұсыну және осының бәрі қарапайым смартфонның көмегімен. Әдетте, мұндай жүйелермен жұмыс істеуге тура келетін ақпарат көлемі өте үлкен, ал ақпараттың өзі өте күрделі құрылымға ие. Қазіргі әлемде банктер мен деректер базасы ұйымдардың, мекемелердің, министрліктердің және т. б. кез келген ақпараттық жүйесінің ажырамас негізгі құрауышына айналды. Ақпараттық жүйелердің классикалық мысалдары – банк жүйелері, авиациялық немесе теміржол билеттерін брондау жүйелері, қонақ үй орындары, оқу және тестілеу ақпараттық жүйелері, кесте жүйелері және т. б [1].

Тақырыптың өзектілігі. Интернет-технологияларды қолдана отырып, тауарлардың шексіз және өте ыңғайлы қозғалысы мен есебін жүргізу мүмкіндігі пайда болды. Кәсіпорынның әр қызметкері немесе иесі кез-келген уақытта интернет бар кез-келген жерде ең маңызды ақпаратты ала алады. Бұл артықшылықтар кәсіпорындарда дәстүрлі бизнес моделіне ақпараттық технологияларды енгізуге ықпал етті.

Әр кәсіпорынның қызметкерлері, жеткізушілері және клиенттері бар. Қызметкерлер тауарлардың алдын-ала жасалған санаттарына тауарлар қосады, тауарлардың ағымдағы саны, сатылған тауарлар мен аяқталған тауарлар саны бар. Жеткізушілер мен клиенттерді тіркеу.

Өте маңызды фактор – компанияның теңгеріміне және қолданыстағы жеткізушілерге тауарлар бойынша өзекті ақпараттың қолжетімділігі. Әрбір қызметкер смартфонның көмегімен кез келген уақытта кеңсеге бармай-ақ немесе компьютерді пайдаланбай-ақ, тауарлар бойынша өзіне қажетті барлық ақпаратты көре алады.

Біздің ақпараттық дәуірде бір қалам мен қағаздың көмегімен құжаттама мен бухгалтерлік есепті тиімді жүргізу мүмкін емес екендігіне ешкім күмән келтірмейді. Алайда, егер бүгінгі таңда нарықта менеджер немесе бухгалтер жұмысының тиімділігін арттыруға арналған бағдарламалар жеткілікті болса, онда смартфондар үшін ұқсас техникалық әзірлемелер өте аз.

Дипломдық жобаның мақсаты - Android платформасына негізделген азық-түлік дүкені тауарларын есепке алу үшін есеп қосымшасының мобильді нұсқасын жобалау және іске асыру, жүйенің параметрлерін конфигурациялау және оңтайландыру.

Қосымшаның негізгі мақсаты - дүкендегі тауарлардың есебі мен қозғалысын ішінара автоматтандыру. Сатудың жоғары деңгейі бар мұндай жүйе компаниядағы тауарларды есепке алуды оңтайландыруға және жеңілдетуге, олардың компаниядағы қозғалысын бақылауға және ыңғайлы статистика жүргізуге қабілетті [2].

Деректерді енгізу және редакциялау кезінде қосымша енгізілетін ақпараттың дұрыстығын бақылауы және пайдаланушылар тарапынан қате іс-әрекеттерге әкелуі мүмкін жағдайларды мүмкіндігінше болдырмауы тиіс.

Тауарларды есепке алу және қозғалысы жөніндегі қосымшаны қолдау мен сүйемелдеуді жеңілдету, сондай-ақ оны одан әрі кеңейту мақсатында ол бағдарламалық жүйелерді құрудың үш деңгейлі архитектурасы негізінде жобалануға және құрылуға, клиенттік және серверлік бөліктері болуға тиіс.

Сервер бөлігі деректерге қол жеткізу үшін жауап беруі керек және қосымшаның іскери логикасын қамтуы керек. Клиент бөлігі өз кезегінде қосымшаның пайдаланушы интерфейсін жүзеге асыруы керек.

1 Әдебиетке шолу

1.1 Мобильді қосымшаның мақсаты мен міндеттері

Дипломдық жобаның мақсаты - Android платформасына негізделген азық-түлік дүкені тауарларын есепке алу үшін есеп қосымшасының мобильді нұсқасын жобалау және іске асыру, жүйенің параметрлерін конфигурациялау және оңтайландыру.

Қосымшаның негізгі мақсаты - дүкендегі тауарлардың есебі мен қозғалысын ішінара автоматтандыру. Сатудың жоғары деңгейі бар мұндай жүйе компаниядағы тауарларды есепке алуды оңтайландыруға және жеңілдетуге, олардың компаниядағы қозғалысын бақылауға және ыңғайлы Статистика жүргізуге қабілетті.

Қосымшада тауарлар бойынша жалғыз пайдаланушы - менеджер үшін қол жеткізу құқығы қамтамасыз етілген [3].

Жалпы, жүйе барлық пайдаланушылар үшін келесі мүмкіндіктерді ұсынуы керек:

- бар тауарларды қарау;
- әр түрлі өлшемдер бойынша өнімдерді сүзу және іздеу;
- фирманың клиенттерін тіркеу;
- жеткізушінің деректерін өзгерту мүмкіндігі;
- жеткізушіні жою мүмкіндігі;
- фирманың клиентін өзгерту мүмкіндігі;
- фирма клиентін жою мүмкіндігі;
- өнім санатын қосу;
- тауарлар санатының өзгеруі;
- тауарлар санатын алып тастау;
- барлық кеңестер бар ыңғайлы, интуитивті интерфейс.

Деректерді енгізу және редакциялау кезінде қосымша енгізілетін ақпараттың дұрыстығын бақылауы және пайдаланушылар тарапынан қате іс-әрекеттерге әкелуі мүмкін жағдайларды мүмкіндігінше болдырмауы тиіс.

Тауарларды есепке алу және қозғалысы жөніндегі қосымшаны қолдау мен сүйемелдеуді жеңілдету, сондай-ақ оны одан әрі кеңейту мақсатында ол бағдарламалық жүйелерді құрудың үш деңгейлі архитектурасы негізінде жобалануға және құрылуға, клиенттік және серверлік бөліктері болуға тиіс.

Сервер бөлігі деректерге қол жеткізу үшін жауап беруі керек және қосымшаның іскери логикасын қамтуы керек. Клиент бөлігі өз кезегінде қосымшаның пайдаланушы интерфейсін жүзеге асыруы керек.

Деректерге қол жеткізу деңгейі ретінде MySQL ДҚБЖ пайдалану керек, ол жүйенің барлық деректерін орталықтандырылған құрылымдық сақтауды қамтамасыз етеді, олардың тұтастығы мен дәйектілігін қамтамасыз етеді, сонымен қатар көптеген төмен деңгейлі қызметтерді ұсынады: деректерді

коймадан оқу, деректерді сақтау, құрылымын өзгерту және т.б. Бұл деңгейде жүйенің барлық деректерін сақтайтын мәліметтер базасы құрылуы керек. Деректерді іріктеу командаларын іске асыру, деректердің тұтастығы мен дәйектілігін бақылау тиісті сақталған процедуралар, триггерлер және сервер ұсынатын басқа объектілер арқылы жүзеге асырылуы керек.

Бизнес логикасының деңгейі қосымшалар серверінде пайда болады және жүйенің өзегі болып табылады. Бұл деңгейде жүйенің бизнес логикасының көп бөлігі шоғырлануы керек:

- жүйе пайдаланушысының авторизация алгоритмдері, қолжетімділік құқығын тексеру жүйесі;
- деректерді өңдеу ережелері, мысалы: пайдаланушының деректерді дұрыс толтыруын тексеру, деректердің өзара байланысын тексеру және ұйымдастыру, жүйе ішіндегі ақпарат қозғалысы ережелері;
- дерекқорға қосылу және транзакцияларды орындау класы;
- дерекқор кестелерімен жұмыс істеуге және серверде тиісті сақталған процедуралар мен функциялардың орындалуын бастауға арналған сыныптар мен алгоритмдер.

Қарапайым бизнес логикасын ұсыну деңгейіне шығару керек: авторизация интерфейсі, шифрлау алгоритмдері, деректерді енгізу және көрсету функциялары, енгізілген мәндерді рұқсат ету және форматқа сәйкестігі үшін бастапқы тексеру, клиенттің жұмыс станцияларына жүктелген мәліметтермен қарапайым операциялар (сұрыптау, топтау, мәндерді есептеу).

Жүйе заманауи Android қосымшаларын және Java бағдарламалау тілін дамытудың негізгі технологиясын қолдана отырып, Android Studio және одан жоғары интеграцияланған ортада Android операциялық жүйесіне арналған.

Бүгінгі таңда кез-келген кәсіпорын немесе компания өз жұмысында жұмыс процесін автоматтандыру үшін барлық Ақпараттық жүйелерді немесе қосымшаларды пайдаланады. Зерттеулер көрсеткендей, олардың бір бөлігі әмбебап дайын жүйелер, ал екіншісі жоғары мамандандырылған және белгілі бір кәсіпорынға, компанияға байланысты. Бұл жүйелер қолдану саласымен, іске асыру технологиясымен, қол жетімділік пен ашықтық деңгейімен ерекшеленеді. Осы саладағы ең жақсы мобильді қосымшалардың бірі – «1С: шағын компанияны басқару», «тауарларды есепке алу» және басқалары [4].

«1С: шағын компанияны басқару» қосымшасын қарастырамыз, бұл компаниядағы тауарларды есепке алуды автоматтандырудың кешенді шешімі. «1С: шағын фирманы басқару» есепті тиімді ұйымдастыруға мүмкіндік береді. Жүйе сізге тауарлар туралы қажетті ақпаратты табуға, статистиканы көруге немесе бірнеше секунд ішінде жаңа өнімді қосуға мүмкіндік береді.

Оқу мазмұны мен тапсырмаларын басқару үшін Class Server құрамына Android құрылғысына орнатылған және мүмкіндік беретін клиенттік бағдарламалық жасақтама кіреді:

- тауарлардың түсуі, сату және түгендеу;
- баспа нысаны шығыс жүкқұжаты;

- тауарларды сату және олардың қалдықтары жөніндегі есептер;
- деректерді мұрағаттау;
- деректерді жою;
- сату Кестесі.

Компанияда тауарларды есепке алуға арналған бағдарламалық жасақтаманың мобильді даму нарығында келесі функционалдылыққа ие қарапайым бухгалтерлік жүйелер басым:

- тауарларды құру және сақтау;
- жеткізушіні тіркеу;
- статистика жүргізу;
- клиенттердің есебін жүргізу.

Мұндай жүйелердің негізгі кемшіліктері – көптеген өнімдермен тиімді жұмыс істеудің мүмкін еместігі және олардың бағдарламадағы абсолютті хаотикалық орналасуы. Тауарлар кездейсоқ сақталады және ешқандай санатқа жатпайды. Егер кәсіпорында көптеген тауарлар болса, қажетті тауарларды табу өте қиын.

Азық-түлік дүкеніндегі тауарларды есепке алу және қозғалысы үшін әзірленген Мобильді қосымша:

- Android негізіндегі смартфонның көмегімен фирмадағы тауарлардың есебі мен қозғалысын барынша жеңілдету;
- қызметкерлердің тиімділігін арттыру, тауарлар, клиенттер мен жеткізушілер туралы барлық ақпарат бір жерде сақталады, қызметкерлерге әртүрлі жерлерде қажетті ақпаратты іздеудің қажеті жоқ;
- кәсіпорынның есебінде және статистикада ақша үнемделеді, өйткені барлығы смартфонда;
- компанияның кез-келген қызметкеріне қосымшамен өте қысқа мерзімде жұмыс істеуге мүмкіндік беретін жеңіл интерфейс;
- көзге көрінбейтін деректердің толық қауіпсіздігі. Қосымшаға кіру тек пароль арқылы қол жетімді;
- көзге көрінбейтін деректердің толық қауіпсіздігі. Қосымшаға кіру тек пароль арқылы қол жетімді;
- деректердің тасымалдануында проблемалар жоқ. Деректер интернетте хостингте сақталады. Смартфонды ауыстырған кезде қосымшаны қайтадан жүктеп алу керек және өнім туралы мәліметтер қызметкерге қайтадан қол жетімді болады.

Жүйеде тіркелген әрбір пайдаланушы автоматты түрде өзінің иелігіне жеке кеңістікті алады, онда ол тек өзіне қол жетімді жеке құжаттарды орналастыра алады және жеке пайдаланушыларға/пайдаланушылар топтарына қол жетімділікті қамтамасыз ете отырып, жалпы пайдалану үшін ақпаратты орналастыра алады [5].

Бұл жүйенің негізгі кемшіліктерінің бірі, көптеген ұқсас жүйелер сияқты, оның параметрлері мен мүмкіндіктерімен конфигурацияның

күрделілігі болып табылады, бірақ тұтастай алғанда "1С: шағын компанияны басқару" - бұл оқу процесін басқарудың қуатты және ыңғайлы, жақсы біріктірілген кешенді жүйесі.

«Тауарды есепке алу» – бұл стандартты функционалдылық пен параметрлермен тауарларды есепке алуға мүмкіндік беретін жүйе. Жүйе шағын фирмаларда қолданылуы мүмкін.

Функционалдық мүмкіндіктері:

- тапсырыстарды тіркеу;
- жаңа түсімдерді есепке алу;
- тауарларға бағалар мен жеңілдіктерді тағайындау;
- клиенттерді есепке алу;

Кемшіліктері-қолданба дамымайды және кемшіліктер бар.

1.1-кесте – Еркін таратылатын оқыту жүйесін талдау

	1С: шағын фирманы басқару	Тауарды есепке алу	Контейнер: сауда + қойма 1С
Лицензия	GNU/GPL	GNU/GPL	GNU/GPL
Жүктеулер саны	100000	5000	5000
Көптілді интерфейс	Жоқ	Жоқ	Жоқ
Орыс тілін қолдау	Иә	Иә	Иә
SCORM қолдау	Иә	Иә	Иә
Құрылымы	ядро + Модульдер жиынтығы	монолитті	монолитті
Кеңейту мүмкіндігі	иә, сыртқы Модульдер есебінен	әзірлеушілерге байланысты	әзірлеушілерге байланысты
Платформа	Android	Android	Android
Қолданбаны жаңарту	Қосымша дамып келеді	Қосымша дамымайды	Қосымша дамымайды

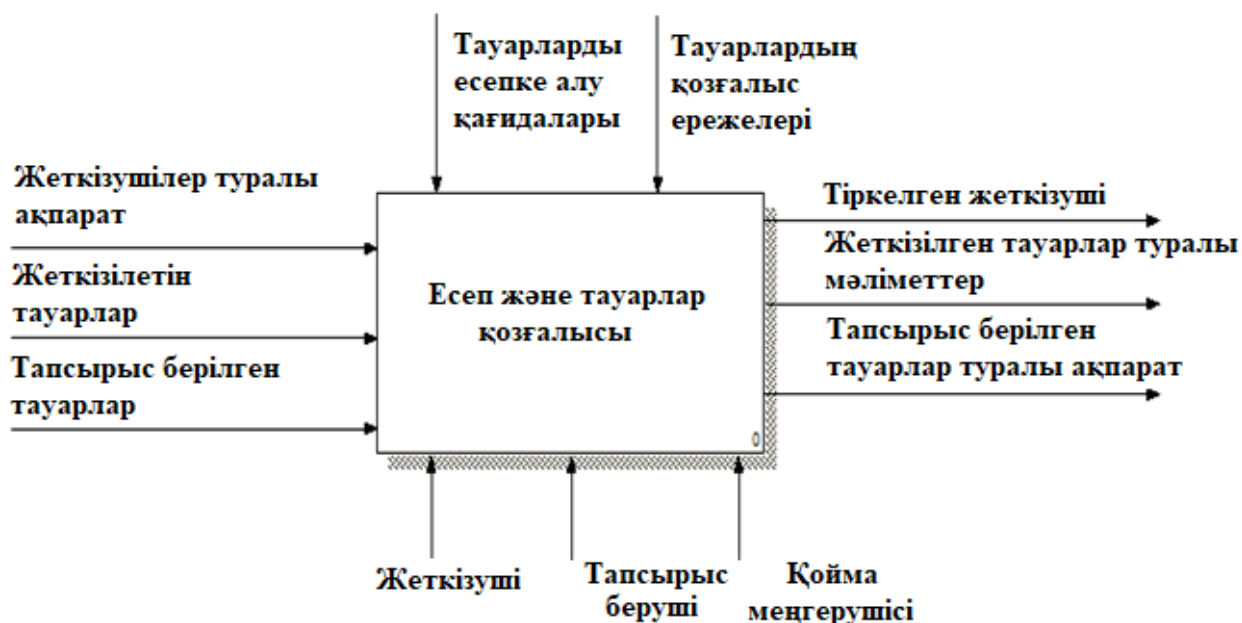
AS-IS Моделі

AS – IS - "сол сияқты" моделі, ұйымның қазіргі жағдайының моделі. Бұл модель қазіргі уақытта болып жатқан процестерді, сондай-ақ пайдаланылатын ақпараттық объектілерді жүйелеуге мүмкіндік береді. Осының негізінде бизнес-процестерді ұйымдастырудағы және өзара әрекеттесудегі қиындықтар анықталады, қолданыстағы құрылымдағы белгілі бір өзгерістердің қажеттілігі анықталады.

As-is моделін құру кезеңінде олардың идеализацияланған көрінісіне емес, процестердің нақты ағындарына негізделген шындыққа жақын модель құру маңызды болып саналады [6].

Компаниядағы тауарларды есепке алуды ұйымдастырудың ағымдағы процестерін талдау негізінде келесі as-is моделі құрылды, ол осы жүйеде жұмыс істеген кезде болып жатқан процестерді оқшаулауға және жүйелеуге

мүмкіндік береді. Бұл модельдің негізгі мәтінмәндік диаграммасы 1-суретте көрсетілген.



1.1-сурет – Негізгі мәтінмәндік диаграмма (AS-is моделі)

Ағымдағы проблемалық аймақта болып жатқан бизнес-процестердің логикасын толығырақ түсіну үшін жоғарыда аталған контекстік диаграмма келесі үш процеске бөлінді:

- жеткізушіні тіркеу;
- тауарларды есепке алу және тіркеу;
- тауар тапсырысын тіркеу.

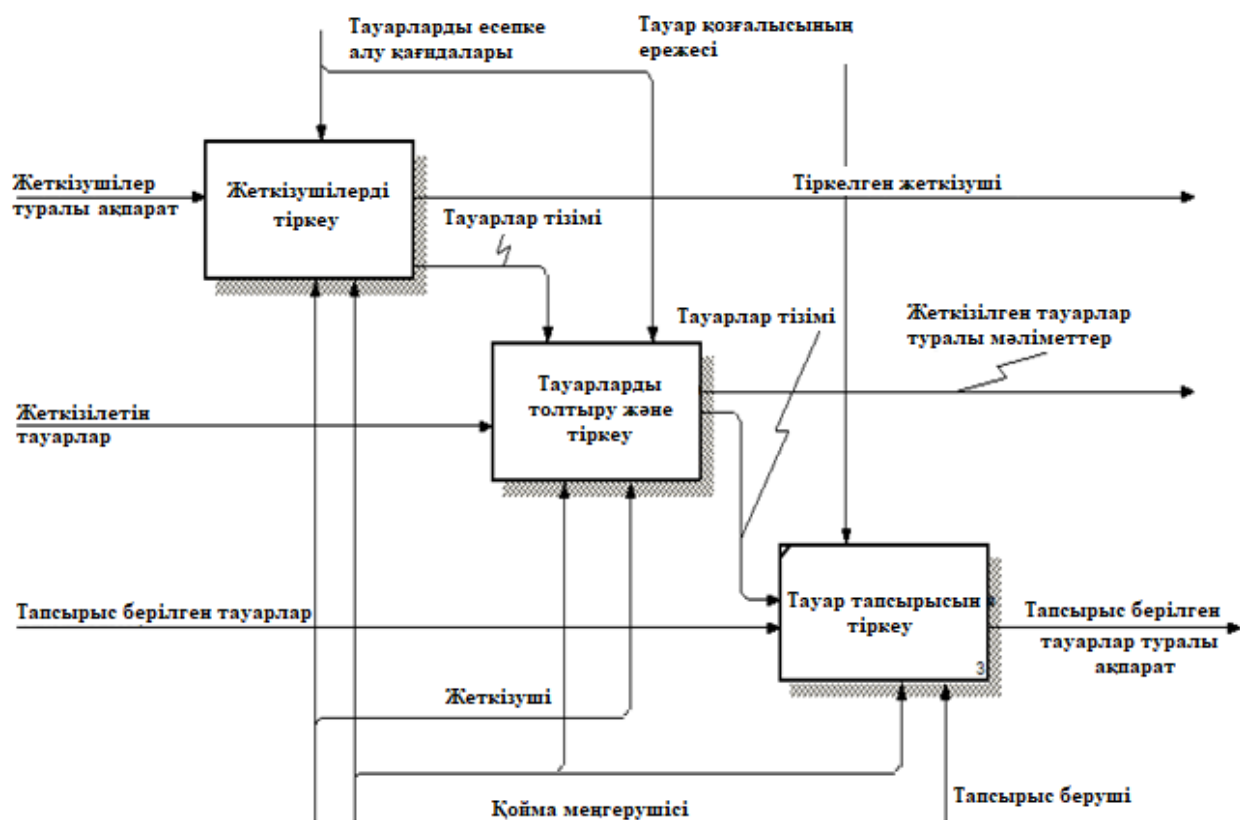
Мәтінмәндік диаграмманың ыдырауы 2-суретте көрсетілген.

Диаграммада оқу процесінің кезеңдері байқалады: басында жеткізуші тіркеледі. Жеткізуші тауарларды жеткізеді. Әрі қарай, қызметкер, қойма меңгерушісі тауарларды тіркейді және клиенттерден тапсырыс береді.

"Жеткізушіні тіркеу" процесін егжей-тегжейлі түсіну үшін ол келесі үш процеске бөлінеді:

- жеткізушілердің деректерін тексеру;
- жеткізушіні тіркеу;
- жүйеге кіру.

"Жеткізушіні тіркеу" процесінің декомпозициясы 3-суретте көрсетілген.



1.2-сурет – Мәтін-мәндік диаграмманың ыдырауы

"Тауарды тіркеу және тіркеу" процесін егжей - тегжейлі түсіну үшін ол келесі екі процеске бөлінеді:

- 1) тауарды тексеру;
- 2) тауарды қоймаға түсіру;
- 3) тауарды санаттар бойынша бөлу;
- 4) қоймада сақтау.

ТО-ВЕ Модельі

ТО-ВЕ моделі ("қалай болу керек") AS-IS негізінде, бизнес-процестерді ұйымдастырудағы кемшіліктерді жоюмен, сондай-ақ AS-IS талдау негізінде анықталған қиындықтарды жою арқылы оларды жетілдірумен және оңтайландырумен жасалады [7].

ТО-ВЕ моделіне сәйкес, даму тақырыбының мақсаты – компаниядағы тауарлардың есебі мен қозғалысын ұйымдастыруды жеңілдету. Бұл жағдайда әзірлеу пәні мыналарды қамтамасыз етуі керек:

- деректер базасын қалыптастыру, тұтынушылар мен жеткізушілер тауарлары туралы ақпарат;
- деректерді өңдеу, қосу, жою үшін осы дерекқорға кіру құралдары;
- дерекқордағы деректерді сұрыптау және сүзу;
- деректер базасында қажетті ақпаратты іздеу құралдары;
- сұралған ақпаратты ыңғайлы көру.

Жоғарыда жасалған as-IS проблемалық аймақ процестерінің моделін талдау негізінде TO-BE моделі құрылды, оның контекстік диаграммасы 3-суретте келтірілген.

Ағымдағы проблемалық аймақта болып жатқан бизнес-процестердің логикасын түсінуді нақтылау үшін контекстік диаграмма келесі төрт процеске бөлінді:

- 1) пайдаланушыларды тіркеу және қолдау;
- 2) тауарлармен жұмыс;
- 3) тауарлардың тапсырыстарымен жұмыс.

Егжей-тегжейлі қарау үшін "пайдаланушыларды тіркеу және қолдау" процесі төрт процеске бөлінеді:

- 1) студенттер мен оқытушыларды тіркеу;
- 2) тіркеудің жеке және құпия деректерін өзгерту;
- 3) құпия сөзді қалпына келтіру;
- 4) пайдаланушыны жою.

Олардың қызметі нәтижесінде барлық процестер мәліметтер қоймасымен ақпарат алмасады: олар жаңа ақпарат қосады, оны шығарады және өзгертеді, сонымен қатар қажет емес деректерді жояды, ал ыдыраудың осы кезеңінде тек "әкімші" бұл функцияларды орындай алатындығын көрсетеді.

"Жаңа тауарларды жеткізумен жұмыс" процесі келесі үш процеске бөлінді:

- 1) Өнім туралы деректерді қосу;
- 2) тауар туралы деректерді өзгерту;
- 3) өнім туралы деректерді жою.

Процестер жұмыс істейтін объектілер ретінде тауарлар туралы мәліметтер болады. Олардың қызметі нәтижесінде барлық процестер "Mysql" мәліметтер базасымен ақпарат алмасады: деректерді қосады, өзгертеді және жояды, ал енгізілген деректер мәліметтер базасының талаптарына сәйкес келуі керек. Бұл әрекеттерді "әкімші" орындай алады.

"Жаңа тапсырыстармен жұмыс" процесі келесі үш процеске ыдырады:

- 1) Жаңа тапсырыс қосу;
- 2) жаңа тапсырысты өзгерту;
- 3) жаңа Тапсырысты жою.

Олардың қызметі нәтижесінде барлық процестер "Mysql" мәліметтер базасымен ақпарат алмасады: деректерді қосады, өзгертеді және жояды, ал енгізілген деректер мәліметтер базасының талаптарына сәйкес келуі керек. Бұл әрекеттерді тек "әкімші" орындай алады [8].

Ақпараттық ресурстар, жүйелер мен технологиялар қазіргі заманғы адам қызметінің ажырамас, тез дамып келе жатқан элементтері болып табылады. 1997 жылы WAP (Wireless Application Protocol - Сымсыз қолданба хаттамасы) технологиясы ұялы байланыс нарығында пайда болды, ол компьютерге

қосылу үшін арналған кабельді пайдаланбастан ұялы телефондарға тікелей ғаламтордан бағдарламаларды орнатуға мүмкіндік берді. Содан бері қоғамның «мобильдендіру» процесі басталды. 2000-жылдардың басында мобильді қосымшаларды телефондармен үлкен сенсорлық экрандармен дамыту мобильді қосымшаларды жасауда сапалы қимылмен жаңа деңгейдегі мобильді қосымшаларды жасауға мүмкіндік берді. 2010-шы жылдардан бастап мобильді құрылғылары қосымшаларды дамыту үшін заманауи ақпараттық технологияларды қолдануға мүмкіндік беретін аса қуатты процессорлармен жабдықталған. Көптеген бағдарламашылар кәсіби қызметтің жаңа бағыттарын үйреніп, нарықтық үрдістерге бейімделе бастады.

Қазіргі уақытта мобильді қосымшаларды әзірлеу ақпараттық технологиялар саласындағы ең танымал іс-әрекеттердің бірі болып табылады. Мобильді қосымшаларды құрастыру тұтынушылық қажеттілікті алдын ала орындауға мүмкіндігі бар шешімі қолданушыға белгісіз алгоритмдер мен тапсырмаларды орындауға негізделген.

Мобильді қосымша - белгілі бір тұтынушылар тобына арналып құрастырылып, олардың қандай да бір мәселелері мен қиындығын шешуге бағытталған. Мобильді қосымша - нақты платформаға орнатылған, белгілі бір әрекеттерді шешуге болатын функционалдығы бар арнайы бағдарлама [9]. Бұл әртүрлі ақпаратпен өзара әрекеттесуге көмектесетін жүйенің бір түрі. Осыған байланысты мынадай түрлерге жіктеледі:

- қосымша-оқиға: әртүрлі іс-шараларды, оқиғаларды көрсетуге мүмкіндік береді, мысалы, спорттық, мәдени, білім-ғылымға қатысты орын алған оқиғаның көрсетілімі және т.б.;

- қызметтік қосымша: ұйымдардың қызметін көрсететін сайттарға ұқсас сервистік қосымшалар;

- үйретуге, дамытуға арналған әртүрлі ойындарға арналған қосымшалар;

- онлайн режимде сатылым жасауға арналған интернет дүкендер; - әртүрлі брендтерді жарнамалауға қолданылатын промо-қосымшалар;

- бизнес-қосымша: ұйымның үдерісін оңтайландыруға, бизнес ақпаратқа қол жетімділікті қамтамасыз етуге және деректер базасымен біріктіруге мүмкіндік береді;

- жүйелік қосымша: телефон мен оның бағдарламалық жасақтамасы үшін кеңейтілген параметрлер мен опцияларды пайдалану;

- телефонды толық навигатор ретінде пайдалануға мүмкіндік беретін GPS модулін пайдаланатын шарлау және іздеу қызметтері бар қосымша;

- бейне және аудио ақпаратпен жұмыс істеу кезінде телефонның мүмкіндігін кеңейтетін мультимедиялық бағдарламаладағы қосымшалар;

- әлеуметтік желілер, байланыс үшін онлайн қызметтер, ақпарат таратуы және әлеуметтік қатынастарды ұйымдастырушы қосымшалар;

- контенттік қосымшалар және т.б.

Кез келген құрастырылған қосымшалар болсын, дайын бағдарламалық өнім болсын ең алдымен идея ретінде басталады. Егер бағдарламалық өнім

немесе мобильді қосымша құру идеясы жоқ болса, онда мұндай жұмысты бастаудың тиімдісі проблемалар және әлеуетті шешімдер тұрғысынан заттарды ойлауға даярлау болып табылады. Егерде белгілі бір жағдайларға байланысты туынған мәселелерге қызығушылық танытып, «бұл мәселенің туындауына не себеп?» «бұл мәселені шешудің жақсы жолы бар ма?» деген секілді сұрақтар арқылы мәселелер мен нарықтық тиімсіздікті анықталса, онда идеяның жартысына қол жеткізгендік болып табылады.

Келесіде бұл мәселенің неліктен екенін түсіну және бұрын соңды бұл мәселе бойынша қосымша жасалғаны жөнінде ақпарат іздестіру. Мәселе кеңістігіне мүмкіндігінше көп зерттеу жүргізу. Мәселені толық түсінгеннен кейін, мобильді қосымша мәселені қалай шеше алатынын бағалауды бастау.

Бұл кезеңде алдымен қосымшаның ақпараттық архитектурасын жобалап алған жөн. Ақпараттық архитектура - бұл қосымшада қандай деректер мен функцияларды ұсыну керектігін және осы деректер мен функциялардың қалай ұйымдастырылғандығын анықтайтын процесс. Әдетте, бұл процесті бағдарламаны орындауда қатысатын функциялардың тізімін және қосымшада қандай жерде көрсетілетінін көрсету арқылы басталады. Бұл қосымшаны модельдеу кезіндегі сұлбаның негізгі құрастырушы блогы болып саналады. Әрі қарай қосымшаның терезесін жасауды және әр функцияларды және деректерді тағайындауды бастау қажет. Осы орайда әр нысанның өз орны бар екеніне көз жеткізу керек.

Пайдаланушылар қосымшамен жұмыс істеу кезіндегі қолайсыздық болған жағдайларға алдын ала талдау жасап шығу қажет. Қосымшада әрбір нысан мен мәзірлер ішіндегі орын алатын click-термен бастапқы бетке ауысу немесе кез келген беттен іс-әрекетті аяқтау үшін қанша click қажет екенін көргіңіз келетін нәрселердің әрқайсысын қарастыру керек. Бұл жердегі click – нысанды басу, шерту дегенді білдіреді. Әрбір басудың интуитивті екенін тексерген жөн. Егер қандай да бір әрекетті жасау үшін бір реттік шерту болса, онда ол қолданушыға ыңғайлы, бірақ жалпы тапсырмаларды орындау үшін бірнеше рет шерту ұсынылмайды [10].

Келесі қадам – шерту арқылы ауысу моделі. Шерту арқылы ауысу моделі жұмыс үрдістеріңізбен қосымша жобасын тексеруге көмегін тигізеді. Олар негізінен қосымшаның интерфейсін телефон арқылы сынақтан өткізу үшін шынайы тестілеу әдісі болып табылады. Мысалы, тұтынушылар жай телефон арқылы қосымша ашылған кезде олардың шеңберінен шығуға мүмкіндік беретін сілтеме алады. Қосымша қазірде ешқандай функционалдық болмаса да, олар қосымшаның әр бетін шертуге және бағдарламаның шарлауын бастайды. Бұл қадамда қиындықтар туындағанда қосымшаның сұлбасына өзгертулер енгізу керек.

Стиль нұсқаулықтары - бұл бағдарламаның дизайнының құрылыстық блоктарына негізделген. Дыбыс стилі нұсқаулығы қолданбаны қолдануда өте пайдалы болады. Бірыңғай дизайн тілінің арқасында пайдаланушылар қосымшаны пайдалануда өзін ыңғайлы сезінеді. Өйткені мобильді

қосымшаның «өмір сүру» ұзақтығына да оның жасалған дизайны әсерін тигізеді.

Жоғары деңгейдегі техникалық жобалау. Ұялы қосымшаны жасау үшін қолданылатын көптеген тәсілдер, технологиялар және бағдарламалау тілдері бар. Олардың әрқайсысы өзінің күшті және әлсіз жақтары бар. Олардың кейбіреулері пайдалануға арзан болуы мүмкін, бірақ өнімділігі төмен, ал басқалары көп уақытты қажет етеді және сапалы, кәсіби түрде болады. Қосымшаны құрудағы ең нашар мүмкіндік - сенімсіз технологиялық стәкті пайдалану, қазіргі таңда мұндай қызмет түрі өте көп. Яғни ешқандай бағдарламалық кодтаусыз-ақ дайын шаблондар арқылы әртүрлі тақырыпта қосымшалар құруға болады. Алайда мұндай дайын шаблондары бар программалармен құрылған қосымшалардың кері тұстары көп: толыққанды басқарудың жоқтығы, берілген дизайнға тәуелділілік, шектеулі интерфейстер және т.б.

Сонымен қатар, мобильді қосымшаларды бағдарламау кезінде Front-end және Back-end жобалау жүйесі бар.

Front-end жобалау - бұл тұтынушы бөлігінің қосымшасын жобалау. Басқаша айтқанда, бұл- пайдаланушы мен сервердің қосымша арасындағы интерфейсін құрастыру. Ол пайдаланушының қандай да бір деректерді енгізуін, сондай-ақ оның бастапқы өңдеуін және тиісті API (application programming interface-қосымшаны бағдарламалау интерфейсі) арқылы серверге жіберуді жүзеге асырады.

Back-end жобалау - бұл пайдаланушылардың немесе ресурстардың арасында деректерді бере үшін жауап беретін қосымшаның сервер жағының дамуы. Төменде кодты жазуды бастамас бұрын ескеру қажет бірнеше нәрселер бар:

- Программалау тілдері - API жасау үшін қолдануға болатын ондаған тілдер бар. Ең жиі қолданылатын тілдер Java, C#, Javascript, PHP және Python.

- Мәліметтер қоры - қазіргі заманғы дерекқорлардың екі негізгі түрі бар. SQL және noSQL. SQL барлық жағдайларда дәстүрлі және жақсы таңдау ретінде саналады. Жалпы SQL нұсқаларын MSSQL, MYSQL және PostgreSQL қамтиды [11].

- Хостинг ортасы (Инфрақұрылым) - бұл кадамда API және дерекқордың қайда және қалай орналастырылатынын шешу қажет. Мұнда қабылданған шешімдер хостинг шығындарын, масштабталуын, орындалуын және өтінімінің сенімділігін анықтауға көмектеседі. Жалпы хостингпровайдерлерге Amazon AWS және Rackspace кіреді. Провайдерді таңдаудан тыс, жүйенің қаншалықты кеңейтілетінін жоспарлау қажет. Бұлтқа негізделген шешімдер ресурстарды пайдалы деп санап, қажет болғанда жоғары және төмен қарай ауқымды түрде төлеуге мүмкіндік береді. Олар сондай-ақ дерекқор сақтық көшірмелерін, сервердің жұмыс уақытын және операциялық жүйенің жаңартуларын жасауға көмектеседі.

Түпнұсқалық (native) қосымшалар - белгілі бір платформаға арналған бағдарламалау тілдерінде жазылған және операциялық жүйеге кіріктірілген, тез және дұрыс жұмыс жасайды және басқа мобильді қосымшалардың функционалдығы мен жылдамдығының артықшылығына ие. Олар осы платформа үшін ең қарапайым түрде бағдарламаның интерфейсі мен жалпы іс-әрекетін жүзеге асыруға мүмкіндік береді. Сонымен қатар, қосымшалар аппараттық құралдарға қол жетімді: бейне камера, микрофон, акселерометр, телефон кітапшасы және т.б. Әрине бұлар бағдарламаны жазу кезінде көп уақытты қажет етеді, мұндай кезде бағдарламаушының құрастыру ортасында арнайы білімі болуы қажет, сондай-ақ әр платформаның өзінің жеке бағдарламалары болғандықтан үлкен бағаны да талап етеді.

Мұндай қосымшаларда өзіне тән құралдары мен бағдарламалау тілдері бар. Мысалы, Android операциялық жүйесіне арналған бағдарлама жазу үшін көбіне Android Studio, Java тілі қолданылады, ал iOS үшін Xcode және Objective-C, сонымен қатар соңғы кезде көп танымалдыққа ие болып келе жатқан Swift қолданылады, Windows Phone үшін Visual Studio және C# программалау тілдері қолданылады.

Кейде бірнеше платформада жұмыс жасауға мүмкіндігі бар, native қосымшалардан бөлек қысқа мерзімде қосымша құруға тура келетін жағдайлар болады. Мұндай кезде, гибриді немесе вебқосымшалар таңдалынады, ал құрастыру үшін кроссплатформалы мобильді фреймворктар қолданылады.

Веб-қосымшаларды сайттың мобильді нұсқасы деп атауға болады, мұндай қосымшалардың кеңейтілген интерфейсі бар. Бұлар арнайы дүкендерде орналаспайды, тек браузер жұмысы үшін қолданылады. Осындай қосымшалардың жұмыс жылдамдығы ғаламтор байланысына тәуелді, сонымен қатар, жылдам құрастырылуы мен төмен бағаға ие. Кроссплатформалы болып саналатын стандартты мынадай веб-технологиялар қолданылады: HTML5, JavaScript және CSS.

Қосымшаның негізгі мақсаты - дүкендегі тауарлардың есебі мен қозғалысын ішінара автоматтандыру. Сатудың жоғары деңгейі бар мұндай жүйе компаниядағы тауарларды есепке алуды оңтайландыруға және жеңілдетуге, олардың компаниядағы қозғалысын бақылауға және ыңғайлы Статистика жүргізуге қабілетті [12].

Қосымшада тауарлар бойынша жалғыз пайдаланушы - менеджер үшін қол жеткізу құқығы қамтамасыз етілген.

Жалпы, жүйе барлық пайдаланушылар үшін келесі мүмкіндіктерді ұсынуы керек:

- бар тауарларды қарау;
- әр түрлі өлшемдер бойынша өнімдерді сүзу және іздеу;
- фирманың клиенттерін тіркеу;
- жеткізушінің деректерін өзгерту мүмкіндігі;
- жеткізушіні жою мүмкіндігі;

- фирманың клиентін өзгерту мүмкіндігі;
- фирма клиентін жою мүмкіндігі;
- өнім санатын қосу;
- тауарлар санатының өзгеруі;
- тауарлар санатын алып тастау;
- барлық кеңестер бар ыңғайлы, интуитивті интерфейс.

Деректерді енгізу және редакциялау кезінде қосымша енгізілетін ақпараттың дұрыстығын бақылауы және пайдаланушылар тарапынан қате іс-әрекеттерге әкелуі мүмкін жағдайларды мүмкіндігінше болдырмауы тиіс [10].

Тауарларды есепке алу және қозғалысы жөніндегі қосымшаны қолдау мен сүйемелдеуді жеңілдету, сондай-ақ оны одан әрі кеңейту мақсатында ол бағдарламалық жүйелерді құрудың үш деңгейлі архитектурасы негізінде жобалануға және құрылуға, клиенттік және серверлік бөліктері болуға тиіс.

Сервер бөлігі деректерге қол жеткізу үшін жауап беруі керек және қосымшаның іскери логикасын қамтуы керек. Клиент бөлігі өз кезегінде қосымшаның пайдаланушы интерфейсін жүзеге асыруы керек.

2 **Логикалық модельдеу**

2.1 **Модельдеу әдістері мен құралдарын таңдау**

Проблемалық аймақты визуалды модельдеу үшін Rational Rose rational Software компаниясына артықшылық берілді. Бұл құрал интернет-шешімдерді қоса алғанда, бағдарламалық жасақтаманы әзірлеу үшін қарапайым және толық интеграцияланған шешім болып табылады. Rational Rose-бұл қосымшаларды жобалау құралдары арасындағы дефакто стандарты. Ешқандай басқа CASE құралы Rational платформасы сияқты шешімдердің кеңдігі мен тереңдігін ұсынбайды. Rational Rose көмегімен модельді визуализациялауға, өзгертуге және сынауға болады.

Rational Rose – дің сөзсіз артықшылықтарының бірі-бұл кері дизайн, өйткені әзірлеуші мен дизайнерге қалыпты графикалық көріністе жұмыс істеп тұрған жүйені өзгертпес бұрын көру өте маңызды. Әдетте, визуалды-графикалық серия техникалық тапсырмалар мен бағдарламалық мәтіндерді қарап шығуға қарағанда әлдеқайда көп әсер етеді. Сонымен қатар, кері жобалаудан өткен жоба пысықталып, қайта құрылуы мүмкін (және кейіннен құрастырылуы мүмкін). Rational Rose бұл үшін барлық қажетті құралдарды ұсынады.

Rational Rose визуалды модельдеудің жетекші құралы болып табылады, өйткені ол барлық қажетті мүмкіндіктерге ие - UML қолдауы, көп тілді итеративті қолдау, командалық дамуды толық қолдау, жетекші архитектуралар мен WinDNA және J2EE/SE/ME сияқты компоненттік модельдерді қолдайтын компоненттік-негізделген даму, қолдану жеңілдігі, оңтайландырылған Интеграция және басқалары [13].

Деректерді жобалау және модельдеу үшін Computer Associates компаниясының AllFusion ERwin Data Modeler (ERwin) құралдары қолданылды. ERwin мәліметтер базасын, деректер қоймаларын және деректер сөрелерін (data marts) жобалауға, құжаттауға және сүйемелдеуге мүмкіндік береді. Осы құралды пайдалану пайдасына БҚ әзірлеушілері үшін негізгі дәлелдер мен фактілер:

- тікелей (модель негізінде мәліметтер базасын құру) және кері (қол жетімді мәліметтер базасында модель құру) ДҚБЖ-нің 20 түріне арналған дизайнға қолдау көрсетіледі;
- ыңғайлы интерфейс пен күнделікті Процедураларды автоматтандыру арқасында еңбек өнімділігін арттырады;
- ERwin-де-факто стандарты;
- бұрын жасалған модельдердің компоненттерін қайта пайдалануға, сондай-ақ басқа әзірлеушілердің жетістіктерін пайдалануға мүмкіндік береді;
- ДБ құрылымын бір типті ДҚБЖ-дан екінші ДҚБЖ-ға ауыстыруға мүмкіндік береді;

- деректер базасының құрылымын құжаттауға мүмкіндік береді;
- өнімді Дерекқордың өмірлік циклінің барлық кезеңдерінде пайдалануға болады;
- деректер қайда сақталатыны және оған қалай қол жеткізуге болатыны туралы нақты және көрнекі ақпарат алуға мүмкіндік береді;
- көрнекі құралдарды қолдана отырып, мәліметтер базасының құрылымын сипаттауға, содан кейін кез-келген ДҚБЖ үшін деректер файлдарын автоматты түрде құруға мүмкіндік береді.

2.2 Пайдалану диаграммаларын әзірлеу

Пайдалану диаграммалары жүйенің функционалды мақсатын сипаттайды, яғни жүйенің жұмыс істеу процесінде не істейтіні және оны жобалау мен әзірлеу процесінде жүйенің бастапқы тұжырымдамалық моделі болып табылады.

Пайдалану жағдайының диаграммасын жасау келесі мақсаттарды көздейді:

- жүйені жобалаудың бастапқы кезеңдерінде модельделген пәндік аймақтың жалпы шекаралары мен контекстін анықтаңыз, сонымен қатар жобаланған жүйенің функционалды мінез-құлқына жалпы талаптарды тұжырымдаңыз;
- логикалық және физикалық модельдер түрінде одан әрі нақтылау үшін жүйенің бастапқы тұжырымдамалық моделін жасау;
- жүйе әзірлеушілерінің өз тапсырыс берушілерімен және пайдаланушыларымен өзара іс-қимылы үшін бастапқы құжаттаманы дайындау.

Бұл диаграмманың мәні келесідей: жобаланған жүйе пайдалану жағдайлары деп аталатын жүйемен өзара әрекеттесетін көптеген нысандар немесе актерлер түрінде ұсынылған. Бұл жағдайда актер (actor) сыртқы жүйемен өзара әрекеттесетін кез-келген субъект деп аталады. Өз кезегінде, пайдалану жағдайы (use case) жүйе актерге ұсынатын қызметтерді сипаттауға қызмет етеді, яғни әр пайдалану жағдайы актермен диалог кезінде жүйенің кейбір әрекеттер жиынтығын анықтайды. Бұл жағдайда актерлердің жүйемен өзара әрекеттесуі қалай жүзеге асырылатыны туралы ештеңе айтылмайды [14].

Әрекет етуші тұлғалар

Жүйенің жұмысын талдау кезінде келесі кейіпкерлер анықталды:

- Менеджер.

Пайдалану жағдайлары

Келесі пайдалану жағдайлары анықталды:

- әрбір каталог бойынша тиісті ақпаратты қарау (оған: тауарлар, контрагенттер, сатып алушылар кіреді);

- деректермен жұмыс істеу (мыналарды қамтиды: өнімдерді қосу, жаңарту және жою, сонымен қатар статистиканы шығару);
- дерекқорды жаңарту және қолдау (тиісті деректерді қосуды, жаңартуды және жоюды қамтиды).

Пайдалану жағдайларының диаграммасы

Пайдалану жағдайларының диаграммалары талаптарды талдауда, итеративті дамуды жоспарлауда және басқаруда қажетті құрал болып табылады. Пайдалану опцияларымен жұмыс жасау нақтылау кезеңіндегі ең маңыздыларының бірі болып табылады. Әрбір пайдалану жағдайы жүйеге қойылатын элеуетті талап болып табылады және ол анықталмайынша, оны іске асыруды жоспарлау мүмкін емес.

Ақпараттық - есепке алу жүйесін әзірлеу кезінде қойылған міндеттер мен талаптарды талдау кезінде. Пайдаланушылар санатты таңдауға және ол туралы тиісті ақпаратты көруге мүмкіндік алады. Пайдаланушылар жүйеде тиісті санат бойынша ақпаратпен жұмыс істей алады (енгізу, өзгерту, жою), сондай-ақ статистиканы шығара алады [15].

Бұл диаграммада әзірленген жүйеге қажетті негізгі пайдалану жағдайлары көрсетілген. Сонымен, жүйемен жұмыс жасамас бұрын тіркелуіңіз керек, содан кейін пайдаланушыларды анықтау үшін аутентификациядан өтуіңіз керек. Жүйеге кіру үшін деректер жоғалған жағдайда ақпаратты қалпына келтіру мүмкіндігі қарастырылған. Пайдаланушы-әкімшілер жүйеде қарастырылған барлық мүмкіндіктерге ие.

Пайдалану жағдайларының сипаттамасы

"Тиісті санат бойынша деректермен жұмыс істеу" пайдалану жағдайы:

- мақсаты: тиісті санат бойынша ақпаратты қосу, жою және жаңарту үшін қызмет етеді.

- оқиғалардың негізгі ағымы:" пайдаланушы " жүйемен жұмыс істей бастағанда орындалады:

1) жүйе қажетті әрекетті сұрайды (Тиісті санат бойынша ақпаратты қосу, жою, жаңарту);

2)" пайдаланушы " әрекетті таңдағаннан кейін бағыныңқы ағындардың бірі іске қосылады: санат бойынша деректерді қосу, жаңарту және жою.

Санат бойынша деректерді қосу:

- жүйе деректерді қосу үшін тиісті бөлімді сұрайды (өнім, контрагент, сатып алушы);

- "Пайдаланушы" тиісті бөлімді таңдайды;

- жүйе енгізу үшін тиісті деректерді сұрайды;

- "Пайдаланушы" деректерді енгізеді;

- жүйе енгізілген деректердің дұрыстығын тексереді;

- жүйе дерекқорға тиісті бөлім бойынша деректерді сақтайды.

Пән бойынша деректерді жаңарту:

1) жүйе деректерді жаңарту үшін тиісті бөлімді сұрайды (тауар, контрагент, сатып алушы);

- 2) "пайдаланушы" тиісті бөлімді таңдайды;
- 3) жүйе жаңарту үшін тиісті деректерді сұрайды;
- 4) "пайдаланушы" тиісті деректерді көрсетеді;
- 5) жүйе жаңарту үшін тиісті деректерді шығарады;
- 6) "пайдаланушы" деректерді жаңартады;
- 7) жүйе енгізілген деректердің дұрыстығын тексереді және оларды деректер базасында сақтайды.

Санат бойынша деректерді жою:

- 1) жүйе деректерді жою үшін тиісті бөлімді сұрайды (тауар, контрагент, сатып алушы);
- 2) "пайдаланушы" тиісті бөлімді таңдайды;
- 3) жүйе жою үшін тиісті деректерді сұрайды;
- 4) "пайдаланушы" тиісті деректерді көрсетеді;
- 5) жүйе деректерді жояды.

- Балама ағындар: егер қандай да бір себептермен ақпаратты қосу, жою, жаңарту мүмкін болмаса, онда қатенің себебі мен оны шешу мүмкіндіктері бар тиісті хабарлама көрсетіледі:

1) санат бойынша деректерді қосу: егер қосу үшін деректер қате болса, жүйе қате деректер туралы ескерту хабарламасын көрсетеді;

2) деректерді қосу тоқтатылды: егер "пайдаланушы" санаты бойынша деректерді қосу " бағыныңқы ағындарын орындау кезінде оны жою туралы шешім қабылданса, онда қосу жойылады және негізгі ағын алдымен басталады;

3) санат бойынша деректерді жаңарту: егер жаңарту үшін деректер қате болса, жүйе деректердің дұрыс еместігі туралы ескерту хабарламасын шығарады;

4) Деректерді жаңарту тоқтатылды: егер "пайдаланушы" санаты бойынша деректерді жаңарту " бағыныңқы ағындарын орындау кезінде оны жою туралы шешім қабылданса, жаңарту жойылады және негізгі ағын алдымен басталады;

- алғышарттар: осы пайдалану жағдайын орындауды бастамас бұрын "пайдаланушы" жүйеге кіруі керек;

- кейінгі шарттар: егер пайдалану жағдайы сәтті аяқталса, дерекқордағы деректер сәйкесінше жаңартылады. Егер орындау барысында қателер орын алса, дерекқордағы жаңартулар сақталмайды.

"Тиісті ақпаратты қарау" пайдалану жағдайы:

- мақсаты: мәліметтер базасындағы тиісті пән бойынша барлық ақпаратты қарауды сипаттайды;

- оқиғалардың негізгі ағымы: " пайдаланушы " жүйе ұсынған пән бойынша оны қызықтыратын ақпаратты көргісі келгенде қолданылады:

1) "пайдаланушы" жұмыс үшін тиісті санатты таңдайды, жүйе таңдалған санаттың тиісті бөлімдерін көрсетеді;

2) "Пайдаланушы" көру үшін тиісті бөлімді таңдайды;

3) жүйе басты бетте көрсетілуі тиіс тиісті бөлімнің ақпаратын іздеуді жүзеге асырады, сондай-ақ ресурс бойынша ыңғайлы навигацияны ұсынады;

4) "пайдаланушы" өзін қызықтыратын ақпараттың тиісті сілтемесін таңдайды;

5) жүйе тиісті ақпараттың деректер базасынан оқуды жүзеге асырады, тиісті бетті ашады және пайдаланушы қарауы үшін қызықтыратын ақпаратты шығарады;

6) балама ағын: жүйе дерекқордағы "пайдаланушы" сұранысы бойынша тиісті ақпаратты таба алмаған жағдайда, жүйе тиісті хабарламаны шығарады (пайдалану жағдайын орындау жалғасуда).

- алғы сөз: осы пайдалану жағдайын бастамас бұрын "пайдаланушы" жүйеге кіруі керек;

- постшарттар: егер пайдалану нұсқасы сәтті орындалса, "пайдаланушыға" таңдалған пән бойынша берілген бөлімнің барлық ақпаратын қарау үшін ұсынылады.

"Дерекқорды жаңарту және қолдау" пайдалану жағдайы:

- мақсаты: жүйенің жұмыс істеуі үшін қажет ақпаратты қосу, жою және жаңарту үшін қызмет етеді.

- оқиғалардың негізгі ағымы: "пайдаланушы" жүйемен жұмыс істей бастағанда орындалады:

1) жүйе қажетті әрекетті сұрайды (Тиісті санат бойынша ақпаратты қосу, жою, жаңарту);

2) "пайдаланушы" әрекетті таңдағаннан кейін, бағынышты ағындардың бірі жұмыс істей бастайды: деректерді қосу, жаңарту және жою.

Деректерді қосу:

1) жүйе деректерді қосу үшін тиісті бөлімді сұрайды (өнім, контрагент, сатып алушы);

2) "пайдаланушы" тиісті бөлімді таңдайды;

3) жүйе енгізу үшін тиісті деректерді сұрайды;

4) "пайдаланушы" деректерді енгізеді;

5) жүйе енгізілген деректердің дұрыстығын тексереді;

6) жүйе дерекқорға тиісті бөлім бойынша деректерді сақтайды.

Деректерді жаңарту:

1) жүйе деректерді жаңарту үшін тиісті бөлімді сұрайды (тауар, контрагент, сатып алушы);

2) "пайдаланушы" тиісті бөлімді таңдайды;

3) жүйе жаңарту үшін тиісті деректерді сұрайды;

4) "пайдаланушы" тиісті деректерді көрсетеді;

5) жүйе жаңарту үшін тиісті деректерді шығарады;

6) "пайдаланушы" деректерді жаңартады;

7) жүйе деректердің дұрыстығын тексереді және оларды деректер базасында сақтайды.

Деректерді жою:

1) жүйе деректерді жою үшін тиісті бөлімді сұрайды (тауар, контрагент, сатып алушы)

2) "пайдаланушы" тиісті бөлімді таңдайды;

3) жүйе жою үшін тиісті деректерді сұрайды;

4) "пайдаланушы" тиісті деректерді көрсетеді;

5) жүйе деректерді жояды.

- балама ағындар: егер қандай да бір себептермен ақпаратты қосу, жою, жаңарту мүмкін болмаса, онда қатенің себебі мен оны шешу мүмкіндіктері бар тиісті хабарлама көрсетіледі:

1) деректерді қосу: егер қосу деректері дұрыс болмаса, жүйе қате деректер туралы ескерту хабарламасын шығарады;

2) деректерді қосу тоқтатылды: егер "санат бойынша деректерді қосу", "пайдаланушы" бағынышты ағындарды орындау кезінде оны жоюды шешсе, онда қосу жойылады және негізгі ағын алдымен басталады;

3) Деректерді жаңарту: егер жаңарту үшін деректер қате болса, жүйе деректердің дұрыс еместігі туралы ескерту хабарламасын шығарады;

4) Деректерді жаңарту тоқтатылды: егер "пән бойынша деректерді жаңарту", "пайдаланушы" бағыныңқы ағындарын орындау кезінде оны жоюды шешсе, жаңарту жойылады және негізгі ағын алдымен басталады;

5) Деректерді жою: егер жойылатын деректердің басқа деректермен байланысы болса, онда жүйе деректерді жою мүмкін еместігі туралы ескерту хабарламасын шығарады.

- алғышарттар: осы пайдалану жағдайын орындауды бастамас бұрын "пайдаланушы" жүйеге кіруі керек;

- кейінгі шарттар: егер пайдалану жағдайы сәтті аяқталса, дерекқордағы деректер сәйкесінше жаңартылады. Егер орындау барысында қателер орын алса, онда дерекқордағы жаңартулар сақталмайды.

2.3 Талдау сыныптарын сәйкестендіру

Нысанға бағытталған талдау және дизайн әдіснамасында басты орын-класс диаграммасы түрінде жүйенің логикалық моделін жасау. Сыныптар жүйенің логикалық көрінісін жасауға мүмкіндік береді, оның негізінде сипатталған сыныптардың бастапқы коды жасалады [16].

Сыныптар мен объектілерді сәйкестендіру үшін:

– классикалық тәсілдер (классикалық категориялауға сүйенеді және пәндік саланың талаптарына сәйкес келеді);

– мінез-құлықты талдау (объектілер мен сыныптардың түпкі себебі ретінде динамикалық мінез-құлыққа назар аударады);

– пәндік саланы талдау (пәндік сала сарапшылары маңызды деп санайтын операциялар мен байланыс объектілерін бөлу);

- опцияларды талдау (пайдалану үлгілері, Жүйе пайдаланушысы операцияны немесе операциялар тізбегін модельдеуден басталатын сценарийлер);
- CRC-карточкалары (компонент–жауапкершілік–қатысушылар: карточкада компоненттің атауын төменнен сол жақ жартысында іздейді – ол не үшін жауап береді, ал оң жақта-кіммен ынтымақтасады);
- бейресми сипаттама (тапсырманы қарапайым ағылшын тілінде сипаттайды, содан кейін зат есімдер сыныптарға үміткерлер ретінде, ал етістіктер операция атауларына жатады);
- құрылымдық талдау (құрылымдық талдаудың ең жоғарғы деңгейін орындайды, содан кейін жіктеуді орындауға ауысады).

2.4 Даму тақырыбының сипаттамасы

Жобаланған немесе талданған жүйенің сипаттамасын модельдеу кезінде оның күйлерінің өзгеру процесін елестету ғана емес, сонымен қатар жүйенің орындайтын операцияларының алгоритмдік және логикалық іске асырылуының ерекшеліктерін нақтылау қажет. Дәстүр бойынша, бұл үшін блок-схемалар немесе алгоритмдердің құрылымдық схемалары қолданылды. Әрбір осындай схема белгілі бір әрекеттерді немесе қарапайым операцияларды орындау дәйектілігіне назар аударады, олар бірге қажетті нәтижеге әкеледі.

Даму тақырыбын сынып ретінде түсіндіре отырып, жүйе объектісінің әрекеті 2-суретте көрсетілген қызмет диаграммасы түрінде ұсынылуы мүмкін.

Қызмет диаграммалары мемлекеттік диаграммалардың ерекше жағдайы болып саналады. Бұл UML тілінде ішкі әрекеттер мен әрекеттердің аяқталуына байланысты процедуралық және синхронды басқарудың ерекшеліктерін жүзеге асыруға мүмкіндік береді.

Қызмет диаграммаларын қолданудың негізгі бағыты-оларды орындау алгоритмдерін көрсету қажет болған кезде сынып операцияларын орындау ерекшеліктерін визуализациялау. Сонымен қатар, әрбір күй жүйенің ішкі оқиғаларына реакцияларды сипаттау үшін қызмет диаграммаларын пайдалануға мүмкіндік беретін белгілі бір сыныптың немесе оның бір бөлігінің жұмысын орындай алады [17].

Қызмет диаграммасында бүкіл жүйенің жалпы жұмысы, сонымен қатар бір әрекеттен екіншісіне ауысудың логикасы мен реттілігі көрсетілген, ал назар іс-әрекеттің нәтижесінде белгіленеді.

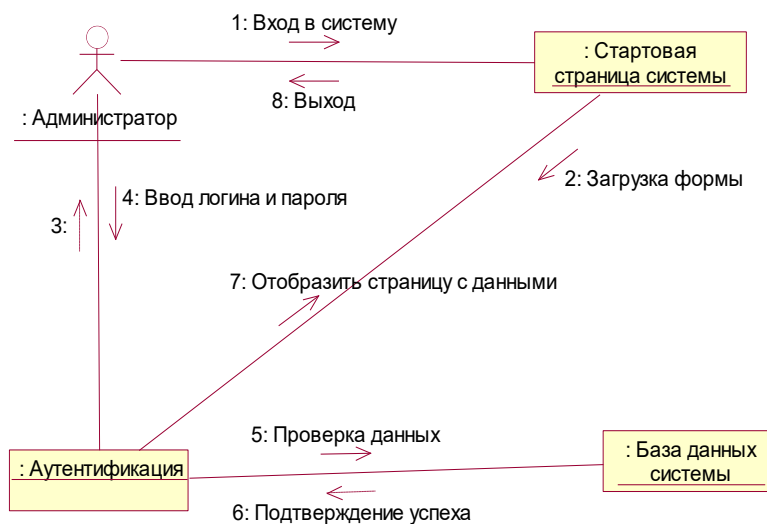
2.5 Нысандардың өзара әрекеттесуі және экрандық нысандар

2.5.1 Аутентификация пайдалану жағдайы

2.1-суретте келтірілген "аутентификация" реттілігінің диаграммасында жүйеге кіруді жүзеге асыру кезіндегі пайдаланушы әрекеттерінің реттілігі көрсетілген.



2.1-сурет – "Аутентификация" реттілігінің диаграммасы



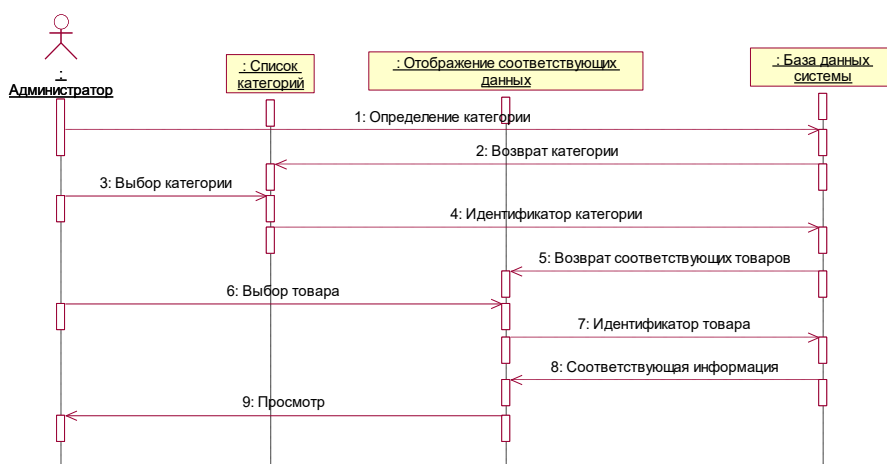
2.2-сурет – "Аутентификация" кооперациясының диаграммасы

2.2-сурет пайдалану жағдайы пайдаланушының жұмысына арналған жүйенің бір бөлігінің мінез-құлқының жалпы сипаттамаларын анықтауға

қызмет етеді. Ол пән бойынша тиісті ақпаратты қарау және оны жүктеу әрекеттерінің реттілігін анықтайды, оны жобаланатын жүйе оны қолданушы актермен өзара әрекеттесу кезінде орындауы керек.

2.3-суретте келтірілген "санат бойынша тиісті ақпаратты қарау" реттілігінің диаграммасында жүйеге кіруді жүзеге асыру кезіндегі пайдаланушының іс-қимылдарының реттілігі көрсетілген.

2.4-суретте келтірілген осы пайдалану жағдайының ынтымақтастық диаграммасы Тиісті санат бойынша деректерді қарауды жүзеге асыру кезінде пайдаланушы мен тиісті сыныптардың өзара іс-қимылының құрылымдық аспектілерін көрсетеді.



2.3-сурет – "Санат бойынша тиісті ақпаратты қарау" реттілік диаграммасы

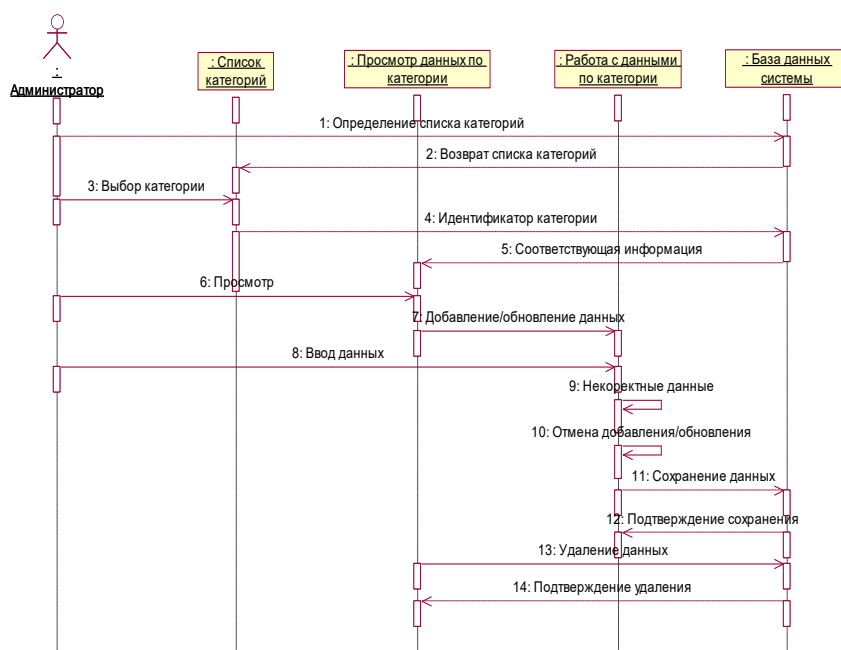


2.4-сурет – "Ақпаратты қарау" кооперациясының диаграммасы

2.5.2 Пайдалану жағдайы «тауарлар санатының деректерімен жұмыс»

Бұл пайдалану жағдайы мәліметтер базасының күйін өзгерту кезінде жүйенің мінез-құлқының ерекшеліктерін сипаттауға қызмет етеді. Ол актер-оқытушымен өзара әрекеттесу кезінде жобаланған жүйенің тиісті пәні бойынша деректерді қосу, жаңарту және жою операцияларын орындау үшін орындалуы керек әрекеттер тізбегін анықтайды.

2.5-суретте келтірілген «тиісті санат бойынша деректермен жұмыс істеу» реттілігінің диаграммасында Тиісті санат бойынша деректермен операцияларды жүзеге асыру кезіндегі оқытушының іс-қимылдарының дәйектілігі көрсетілген [19].



2.5-сурет – "Тиісті санат бойынша деректермен жұмыс істеу" реттілік диаграммасы



2.6-сурет – "Тиісті санат бойынша деректермен жұмыс істеу" кооперация диаграммасы»

2.6-суретте келтірілген "тиісті санаттағы деректермен жұмыс істеу" экран пішінінің орналасуы өте дерексіз көрініске ие. Жүйені іске асыру барысында, әрине, нақты формалар, ең алдымен, пайдаланушы интерфейсі тұрғысынан жобаланғаннан айтарлықтай ерекшеленеді. Төменде 2.7-суретте экрандық нысанының макеті көрсетілген.

ADMIN PAGE

Name

Amount

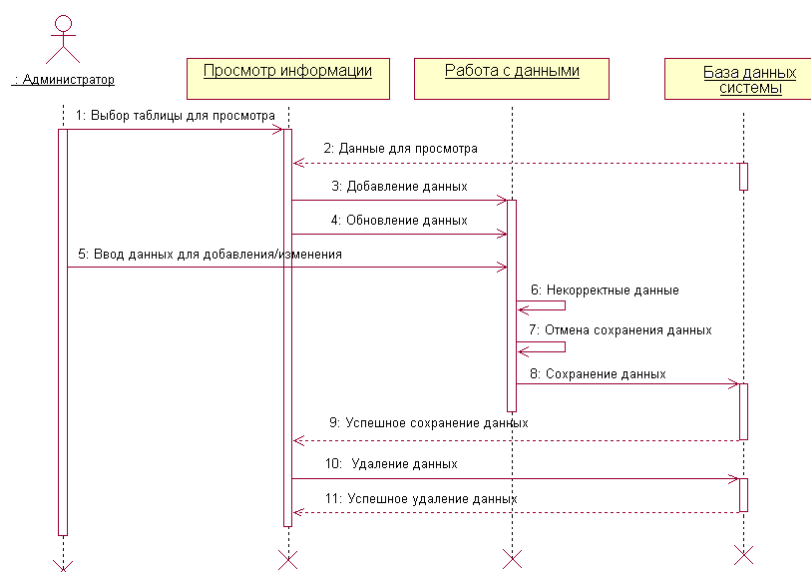
Price

2.7-сурет – "Тиісті санат бойынша деректермен жұмыс істеу" экрандық нысанының макеті

2.5.3 Пайдалану жағдайы «деректер базасын жаңарту және қолдау»

Бұл пайдалану жағдайы тұтастай алғанда мәліметтер базасының күйін өзгерту кезінде жүйенің мінез-құлқының ерекшеліктерін сипаттауға қызмет етеді. Ол тиісті әкімші актерімен өзара әрекеттесу кезінде жобаланған жүйенің тиісті деректерін қосу, жаңарту және жою операцияларын орындау үшін орындалатын әрекеттер тізбегін анықтайды.

2.8-суретте келтірілген "ДБ жаңарту және қолдау" реттілік диаграммасында тиісті деректерді қосу, жаңарту және жою операцияларын жүзеге асыру кезінде әкімші әрекеттерінің реттілігі көрсетілген.



2.8-сурет – "Дерекқорды жаңарту және қолдау" реттілік диаграммасы

2.9-суретте келтірілген "ДБ жаңарту және қолдау" экран пішінінің интерфейсінің орналасуы болашақ интерфейсін прототипі болып табылады, ол жобаланған интерфейсден айтарлықтай ерекшеленеді.



2.9-сурет – "ДБ жаңарту және қолдау" кооперация диаграммасы

2.6 Логикалық деректер моделі

2.6.1 Нысанның негізгі түрлерін анықтау

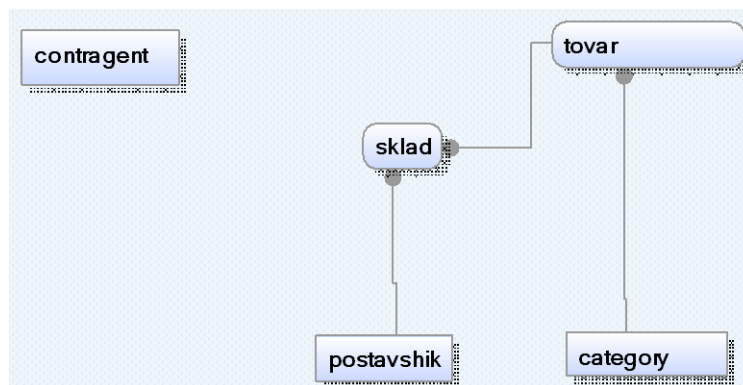
Пәндік аймақты және ТО-ВЕ моделінің диаграммасын талдау нәтижесінде, сондай-ақ бұрын таңдалған нысандарды қажетті қалыпқа келтіргеннен кейін (3 қалыпты пішінге дейін), олардан қажетсіз элементтерді алып тастап, алынған модельдерді жергілікті логикалық деректер модельдеріне түрлендіру (m:N типті байланыстарды, күрделі байланыстарды, рекурсивті байланыстарды, бірнеше атрибуттарды, байланыстарды жою атрибуттармен және артық байланыстармен, 1-1 типті Байланыстарды қайта тексеру), 2.1-кестеде көрсетілген нысан түрлері түпкілікті анықталды, ал нысан атрибуттарының анықтамасы мен сипаттамасы "А қосымшасында" келтірілген [20].

2.1-кесте – Нысандардың негізгі түрлерін және олардың атрибуттарын анықтау

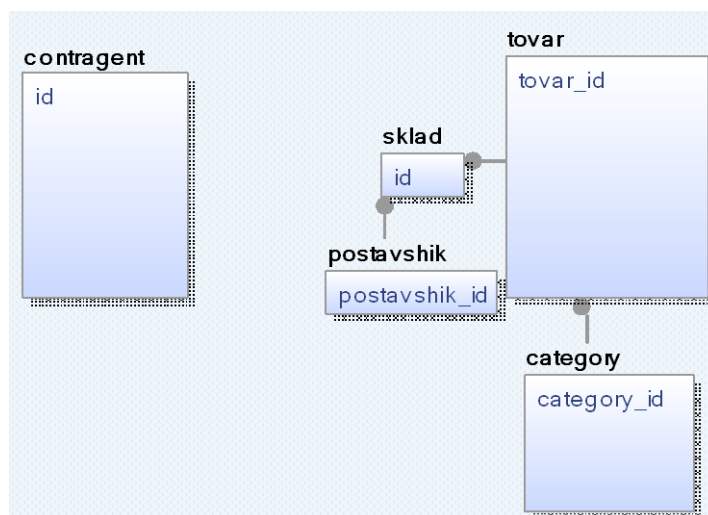
Аты мәні	Сипаттамасы	Лақап	Пайдалану ерекшеліктері
Контрагент	Шарт жасасу кезінде азаматтық-құқықтық қатынастарда тарап болып табылатын жеке немесе заңды тұлға	клиент, сатып алушы	Ол өнімді сатып алатын клиент ретінде қолданылады.
Қойма	Біздің тауарлар сақталатын орын.	Қойма	Тауарды сақтау, сондай-ақ оны жеткізушіден қосу және тауарды клиентке сату үшін қажет.
Тауар	Құны бар және қоғамда алмасу, сатып алу - сату арқылы бөлінетін еңбек өнімі	өнім	
Категория	Белгілі бір жалпы белгілермен байланысты тауарлар	топ, сынып	Белгілері бойынша тауарларды байланыстырады.
Жеткізуші	Жеке немесе заңды тұлға.	импорттаушы	Тауарды сатып алатын тұлға.

– ERwin құралының көмегімен нысандар мен олардың атрибуттарына жүргізілген талдау негізінде IDEF1X (integration DEFinition for Information Modeling) жазбасында төменде әртүрлі деңгейлерде ұсынылған деректердің логикалық моделі жасалды.

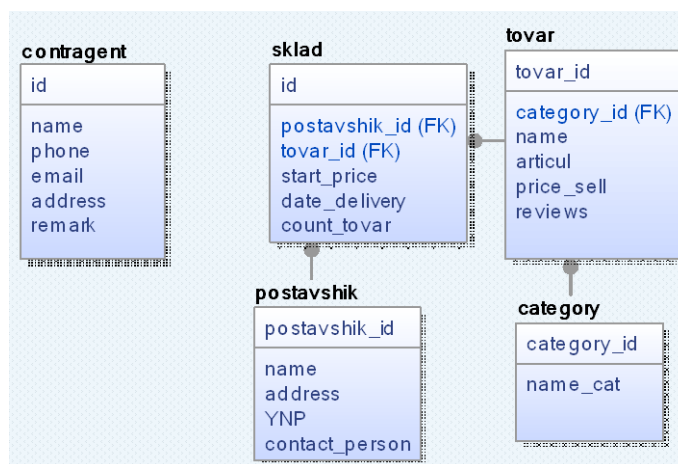
- 2.10-суретте нысан деңгейіндегі деректер моделі келтіріледі (олардың ішкі құрылымын ашпай);
- 2.11-суретте бастапқы кілттер деңгейіндегі мәліметтер моделі;
- 2.12-суретте атрибуттар деңгейіндегі мәліметтер моделі (субъектінің ішкі құрылымын және олардың арасындағы қатынастарды ашумен) [21].



2.10-сурет – Нысан деңгейіндегі деректердің логикалық моделі



2.11-сурет – Бастапқы кілттер деңгейіндегі деректердің логикалық моделі

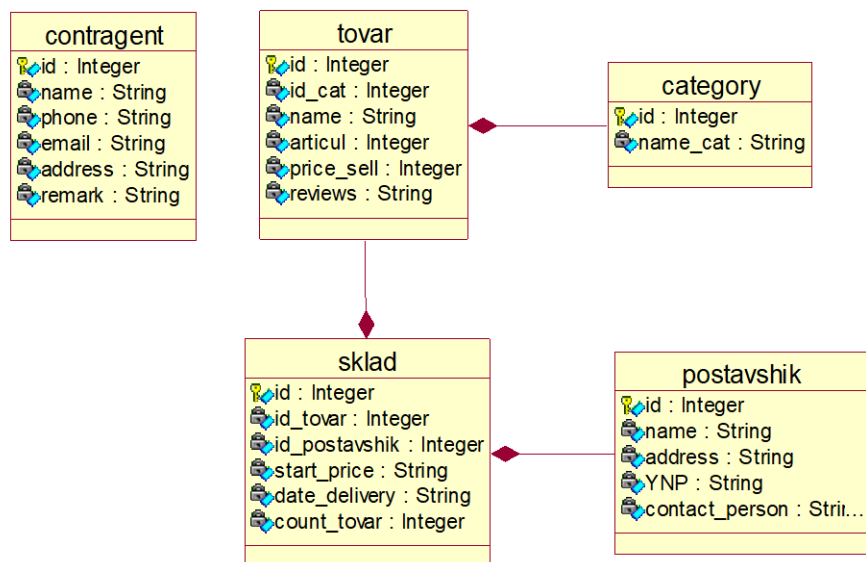


2.12-сурет – Атрибуттар деңгейіндегі деректердің логикалық моделі

2.7 Даму тақырыбының статикалық моделі

2.7.1 Қолданбалы архитектураның статикалық моделі

2.13-суретте келтірілген сынып диаграммасында жүйенің мәліметтер базасындағы мәліметтермен жұмыс істеуге қызмет ететін сыныптар көрсетілген. Олар әзірлеушіге нақты деректер көзінен абстракциялауға мүмкіндік беретін нысандардың прототиптері [22].



2.13-сурет – Мәліметтер базасының класс диаграммасы

3 Жобалау бөлімі

3.1 Даму ортасын, бағдарламалау тілін және даму құралдарын таңдау

Компанияда тауарларды есепке алу үшін Android қосымшасы Java бағдарламалау тілін қолдану арқылы жүзеге асырылады. Құрал ретінде IntelliJ Idea 15 ортасы таңдалған бағдарламалық жасақтаманы іске асырудың таңдалған технологияларына қуатты аспаптық және функционалды қолдау көрсетеді.

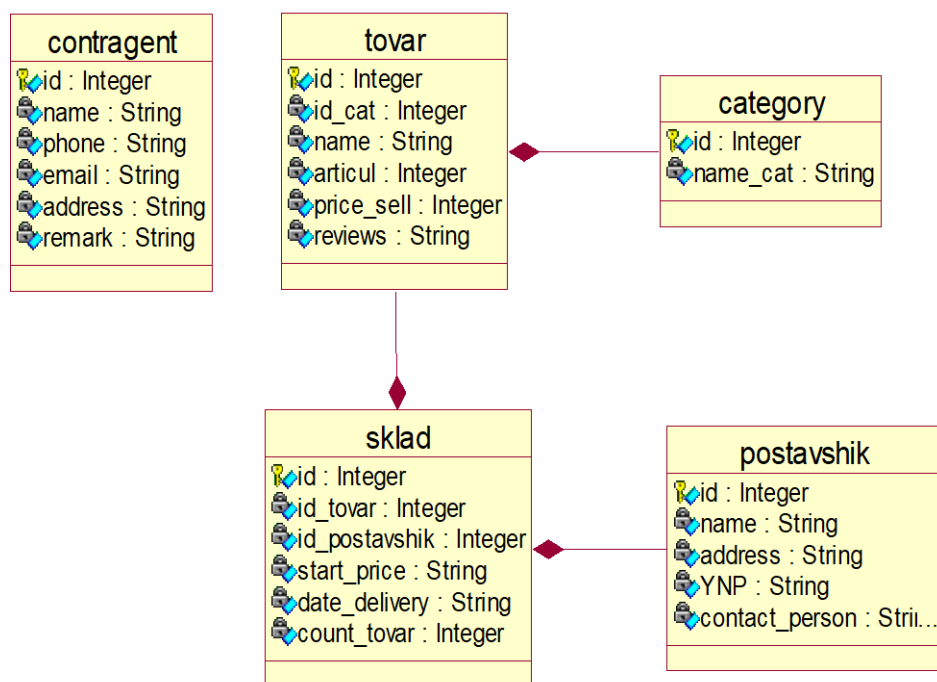
MySQL серверінің ДҚБЖ дерекқорды дамыту ортасы ретінде таңдалды.

MySQL Server-бұл кеңейтілген электрондық коммерция шешімдерін, бизнес қосымшаларын және деректер қоймаларын тез құруға арналған мәліметтер базасының толық ұсынысы. Бұл осы шешімдердің нарыққа шығу уақытын едәуір қысқартуға мүмкіндік береді, сонымен бірге ең жоғары талаптарға жауап беретін масштабталуды қамтамасыз етеді. Серверге XML тілі мен HTTP протоколына қолдау, жүктемені бөлуге және үздіксіз жұмыс істеуге мүмкіндік беретін өнімділік пен қол жетімділікті арттыру құралдары, басқару мен конфигурацияны жақсарту функциялары кіреді. Сондай-ақ ДҚБЖ HTTP хаттамасы бойынша Web арқылы деректер базасына ыңғайлы қолжетімділікті, ДБ-да және құжаттарда сақталатын деректерде жылдам әрекет ететін кіріктірілген толық мәтінді іздеуді ұсынады [23].

3.2 Деректердің физикалық моделі

3.2.1 Мәліметтер базасының кестелерін жобалау және құру

Деректер базасының логикалық моделіне сүйене отырып, 3.1-суретте "FIRM" жобаланған мәліметтер базасының физикалық моделі берілген, ол бастапқы және сыртқы кілттер деңгейінде көрсетілген, ал "В қосымшасы" Mysql Workbench көмегімен жүзеге асырылған кестелердің физикалық құрылымын және олардың "FIRM" мәліметтер базасының байланысын ұсынады.



3.1-сурет – Атрибуттар деңгейіндегі деректердің физикалық моделі

3.2.2 Сақталатын процедураларды әзірлеу

Деректердің физикалық және логикалық тәуелсіздігі ұғымы тұтастықтың барлық шектеулерін қосымшаның логикасы емес, мәліметтер базасының деңгейінде анықтау керек деп түсіну керек. Осы технологияны қолдана отырып, Дерекқордың тиімді жұмысының логикасын және жетіспейтін функциялардың орнын толтыруды жүзеге асыру үшін серверде тиісті сақталған процедуралар жасалды, олар 3.1-кестеде келтірілген, ал "В қосымшасында" олардың кейбіреулерінің тізімі берілген [24].

3.1-кесте – Әзірленген сақталатын процедуралардың тізімі

№	Сақталған процедураның атауы	Мақсаты
1.	deleteProduct	Өнімді жояды
2.	deleteCategory	Санатты жояды
3.	listProducts	Тауарлар тізімін қайтарады
4.	GetStore	Тауарлар қоймасын қайтарады
5.	GetProvidersList	Жеткізушілер тізімін қайтарады
6.	InsertCategory	Жаңа санатты қосу үшін қолданылады
7.	addProduct	Жаңа өнімді қосу үшін қолданылады
8.	InsertStore	Жаңа қойманы қосу үшін қолданылады
9.	UpdateCategory	Қолданыстағы санатты жаңарту үшін қолданылады
10.	UpdateStore	Қолданыстағы қойманы жаңарту үшін қолданылады
11.	UpdateProduct	Қолданыстағы өнімді жаңарту үшін қолданылады
12.	GetProviderById	Жеткізушіні идентификатор бойынша қайтарады
13.	GetContagentList	Клиенттердің тізімін қайтарады
14.	GetContagentById	Клиентті идентификатор арқылы қайтарады
15.	GetCategoriesList	Санаттар тізімін қайтарады
16.	GetCategoryById	Идентификатор бойынша санатты қайтарады
17.	GetProductById	Өнімді идентификатор арқылы қайтарады

3.3 Әзірлеу пәнінің компоненттері

Құрамдас Диаграмма келесі мақсаттарға арналған:

бағдарламалық жүйенің бастапқы кодының жалпы құрылымын визуализациялау;

бағдарламалық жүйенің орындалатын нұсқасының ерекшеліктері;

бағдарламалық кодтың жеке бөліктерін бірнеше рет пайдалануды қамтамасыз ету.

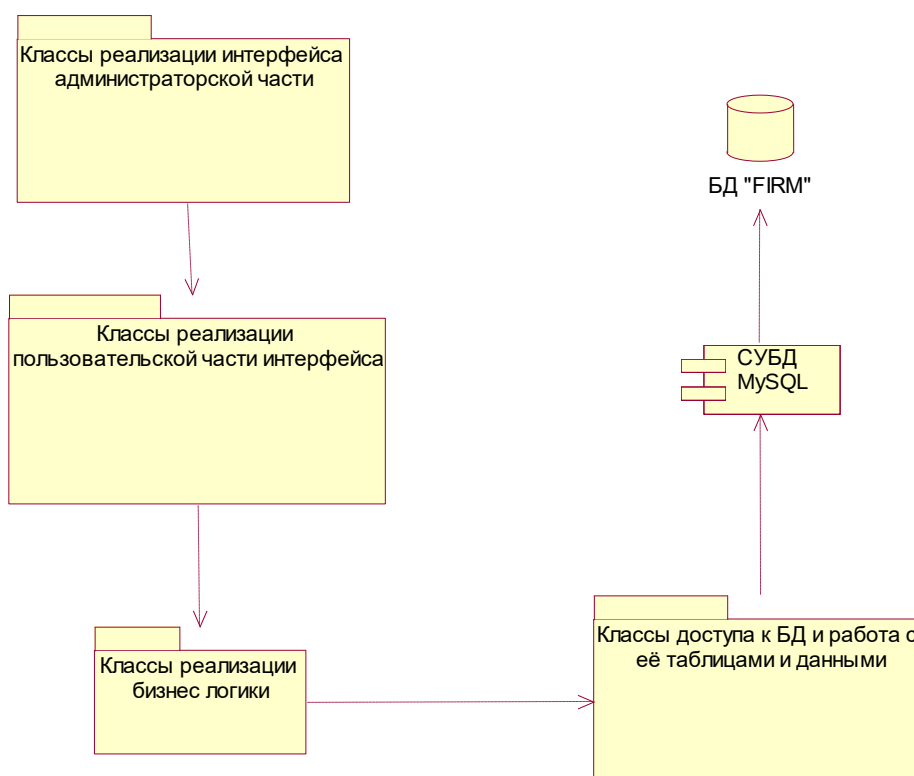
Пайдаланушы деңгейінде әзірленген жүйе үш бөліктен тұрады: әкімші, оқытушы және студент, ал Физикалық деңгейде – мәліметтер базасынан (ақпараттық ресурстың серверінде орналасқан және ДҚБЖ компоненттерінің бірі болып табылады) және Android қосымшасынан (смартфонда көрсетілетін activity түріндегі клиенттік қосымшадан) [25].

Қосымшаның клиенттік бөлігінің құрамдас бөліктерінің диаграммасы
3.2-суретте келтірілген.



3.2-сурет – Қосымшаның клиенттік бөлігі компоненттерінің диаграммасы

Қосымшаның серверлік бөлігі компоненттерінің диаграммасы
3.3-суретте келтірілген.



3.3-сурет – Қосымшаның серверлік компоненттерінің диаграммасы

3.4 Әзірлеу пәнін орналастыру

- Жүйенің элементтері мен компоненттерін визуализациялау үшін оны орындау сатысында ғана бар, 3.4-суретте жүйе болуы керек элементтердің бейнесі бар орналастыру диаграммасы көрсетілген. Диаграммада тек компоненттер көрсетілген-орындалатын файлдар немесе динамикалық кітапханалар болып табылатын бағдарлама даналары.

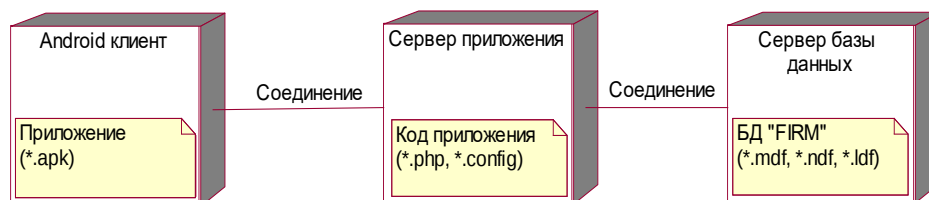
- Логикалық бейнелеу диаграммаларынан айырмашылығы, орналастыру диаграммасы тұтастай алғанда жүйе үшін бірдей, өйткені ол оны жүзеге асырудың ерекшеліктерін толығымен көрсетеді. Бұл диаграмма берілген бағдарламалық жүйе үшін объектіге бағытталған талдау мен жобалау процесін аяқтайды және оны жасау модель сипаттамасының соңғы кезеңі болып табылады.

- Диаграмманы әзірлеу кезінде алға қойылған мақсаттарды тізімдейміз:

- жүйе компоненттерінің оның физикалық түйіндеріне таралуын анықтаңыз;

- оны орындау сатысында жүйенің барлық түйіндері арасындағы физикалық байланыстарды көрсету;

- жүйенің қиындықтарын анықтаңыз және қажетті өнімділікке қол жеткізу үшін оның топологиясын қайта конфигурациялаңыз.[26]

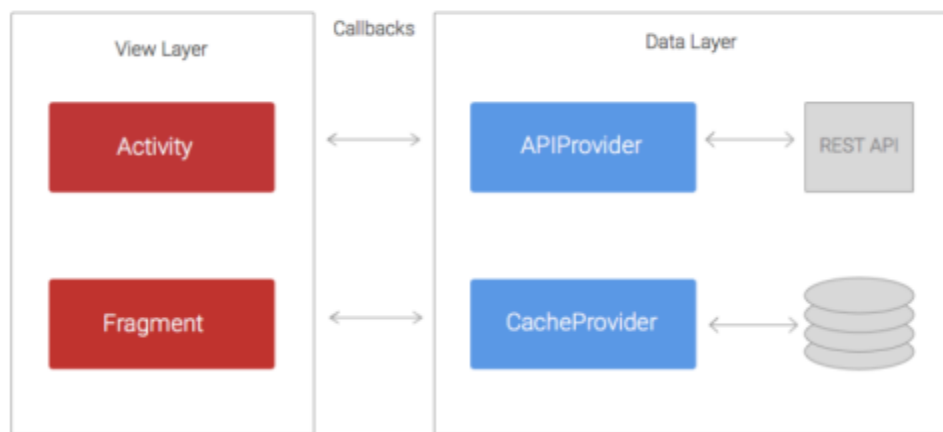


3.4-сурет – Жүйені орналастыру диаграммасы

Web-қосымшаның сервері php-беттердің көмегімен іске асырылады, олар өз кезегінде әзірленген сыныптарға негізделеді (негізделеді) (файлдарда сақталады *.php) және master-бірдей бет интерфейсін қамтамасыз ететін беттер.

Серверлік станцияда сақталатын және ДҚБЖ компоненті болып табылатын мәліметтер базасы *деректер файлдарынан тұрады.mdf және*.NDF, сондай-ақ транзакциялар журналы*.ldf.

Осы бөлімнің соңында 3.5-суретте бағдарламалық-әдістемелік кешеннің болашақ физикалық (аппараттық) қанат жаюының жалпы құрылымы келтірілген.



3.5-сурет – Қолданбаны аппараттық орналастыру құрылымы

4 Бағдарламалық жасақтаманы іске асыру және тестілеу

4.1 Бағдарламалық жасақтама компоненттерінің мақсаты мен сипаттамасы

Әзірленген қосымша бағдарламалық өнімнің ажырамас бөлігі болып табылады. Ол дерекқордан қажетті ақпаратты алу үшін пайдаланушы интерфейсінің орындалуын қамтамасыз етеді. Қосымша әртүрлі статистикалық ақпаратты қосуға мүмкіндік береді, сонымен қатар оны қарау және өзгерту құралы ретінде қызмет етеді. Осы мүмкіндіктердің барлығын жүзеге асыру үшін қосымша Android қосымшаларын әзірлеу технологияларын қолдана отырып, Android қосымшасы түрінде жұқа клиент ретінде жасалды [27].

Төменде дерекқорда сақталған мәліметтермен транзакцияларды қосу, қол жеткізу, көрсету және орындау үшін жасалған сыныптардың, сондай-ақ тікелей деректер сыныптарының сипаттамасы келтірілген:

- Server – базалық класс ол бағалармен жұмыс істеуге жауап береді;
- Main – негізгі класс, бағаны алу үшін;
- LoginMenu – негізгі класс ол сатып алушылармен жұмыс істеуге жауап береді;
- AdminPage2 – негізгі класс, сатып алушы туралы ақпаратты көрсету үшін;
- ServerSomething – базалық класс, қосымшадағы барлық процестерді жүзеге асыруға жауап береді;
- Connect – деректер базасына негізгі транзакцияларды (операцияларды) қосу және орындау үшін базалық класс;
- WarehouseList – қойма сыныбы;
- MainFrame – ұйым класы;
- WindowWithTheProduct – кіріс тауарлар класы;
- inputStream – жеткізушінің класы;
- outputStream – тауарлардың келу сыныбы.

Файл құрылымы AndroidManifest.xml

Элемент <manifest> файлдың түбір элементі болып табылады.

```
<manifest package="com.example.mogilevtrans2"
```

```
android:versionCode="1"
```

```
android:versionName="1.0"
```

```
xmlns:android="http://schemas.android.com/apk/res/android">
```

<Uses-sdk > элементі Android платформасының көрсетілген нұсқасымен (немесе API-нің жаңа нұсқаларымен) қосымшаның үйлесімділігін жариялауға

мүмкіндік береді. Бағдарлама жариялаған әрі деңгейі осы бағдарлама орнатылған мобильді құрылғы жүйесінің API деңгейімен салыстырылады.

```
<uses-sdk android:minsdkversion="11" />
```

Sdk-тің 11 нұсқасы қосымшаның android 3.0 кітапханаларын қолдана отырып құрастырылатынын білдіреді.

<Uses-permission > элементі қолданбаға оның қалыпты жұмыс істеуі үшін жүйе беретін рұқсаттарды сұрайды.

Интернетке қол жетімділікті пайдалану:

```
<uses-permission android:name="android.permission.internet" />
```

Geolocation класы арқылы алынған wi-fi желісі немесе ұялы желі ұсынатын орын деректеріне қол жеткізуге рұқсат беру:

```
<uses-permission  
android:name="android.permission.access_coarse_location" />
```

Қолданбаға geolocation класы арқылы gps деректеріне қол жеткізуге мүмкіндік береді.

```
<uses-permission android:name="android.permission.access_fine_location"  
/>
```

<Application > элементі-пакетте қол жетімді бағдарлама компоненттерінің сипаттамасын қамтитын манифест элементі. Бұл элементте қосымшаның құрамына кіретін компоненттердің әрқайсысын жариялайтын балалар элементтері бар [28].

Элемент <activity> әрекетті жариялайды. Барлық әрекеттер манифест файлында <activity> жеке элементтерімен анық көрсетілуі керек.

Тапсырма үлгісі қызметі:

```
<activity android:name=".activities.mapsactivity"  
android:label="map" />  
<activity android:name=".activities.objectseditoractivity"  
android:label="objects">  
</activity>
```

Кіші каталог res/layout/

Бұл қалтаға белгілеу файлдары орналастырылған.

Белгілеу-бұл әрекетті білдіретін белгілі бір терезе үшін пайдаланушы интерфейсі элементтерінің орналасу архитектурасы. Белгілеуді анықтаудың

және көрініс иерархиясын құрудың ең жалпы тәсілі-XML белгілеу файлында. Пайдаланушы интерфейсін xml файлында жариялаудың артықшылығы-бұл қосымшаның әрекетін басқаратын бағдарламалық кодтан бағдарламаның көрінісін анықтауға мүмкіндік береді. Xml файлының жалпы құрылымы-xml элементтерінің ағашы, онда әр түйін view класының атауын білдіреді.

Activity_home файлы.xml

Activity_home файлы.xml-негізгі активити үшін белгілеу файлы. Бұл файлдың түбірлік элементі <RelativeLayout>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent" >

Бұл элементтің 3 еншілес элементі бар <ImageButton>

<ImageButton

android:id="@+id/ButStops"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignParentLeft="true"

android:layout_alignParentTop="true"

android:layout_marginLeft="28dp"

android:layout_marginTop="74dp"

android:src="@drawable/stops" />

Осы элементтердің әрқайсысында берілген элементтің қасиеттерін анықтайтын атрибуттар бар. Мысалы:

android:id="@+id/ButStops – бағдарлама кодындағы view объектісімен белгілеу файлынан объект байланыстыратын объектінің идентификаторын анықтайды. + (@+Id) таңбасы бұл файлдағы ресурстарға жасалуы және қосылуы керек жаңа ресурс атауы екенін білдіреді r.java, android ортасы жоба үшін автоматты түрде жасалады.

Map файлдары.xml, object_info.xml, object_menu.xml, object_view.xml, objects.xml, search_stage.xml, search_menu.xml, search.xml ұқсас түрде орналастырылған [29].

Әрекетті бастаған кезде жүйе мобильді құрылғы экранында графикалық интерфейсін салу үшін қолданылатын белгілеу ағашының түбірлік Түйініне сілтеме алуы керек. Ол үшін активтендіру таңбасы келесідей жүктеледі[20]

```
Public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_home);  
    Кіші каталог res / drawable/
```

Бұл каталогта қосымшада қолданылатын барлық графикалық файлдар орналасқан: белгішелер, объект маркерлері, түймелер және т. б.

Кіші каталог `res / values/`

Бұл қалта бүкіл қосымшаның жалпы тұрақтыларын сақтайды: басқару элементтері қолданатын мәтін. Мысалы:

```
<string name="app_name">TMap</string>
```

Әрекет файлдары

Әрбір қызмет файлында осы қызметте қосымшаның логикасын жүзеге асыратын код қосылатын сынып бар. Осы сыныптардың әрқайсысында `onCreate ()` кері шақыру әдісі міндетті түрде анықталады, оны құрылғы экранында қызмет терезесін салу үшін жүйе шақырады.

Бұл қосымша одан әрі кеңейту (жаңа бет-модульдермен толықтыру) және модификациялау мүмкіндігі бар модульдік құрылымға ие. Демек, бұл жүйе болашақта сүйемелдеу үшін де, кеңейту үшін де оңай.

Бұл қосымша одан әрі кеңейту (жаңа бет-модульдермен толықтыру) және модификациялау мүмкіндігі бар модульдік құрылымға ие. Демек, бұл жүйе болашақта сүйемелдеу үшін де, кеңейту үшін де оңай.

4.2 Бағдарламалық жасақтаманы тестілеу

Жүйе қолданушыға дерекқорға қарама-қайшы немесе дұрыс емес ақпаратты қосуға мүмкіндік бермейтіндей етіп жүзеге асырылады, бұл жүйенің басқа аналогтарға қарағанда артықшылықтарының бірі болып табылады.

Қосымшаны тестілеу Web-қосымшаның барлық модульдерінің дұрыс жұмыс істеуін тексеру және мәлімделген функционалдылықтың болуын тексеру мақсатында орындалады. Дипломдық жобаны әзірлеу талаптарына сәйкес, деректермен барлық операциялар, олардың дұрыстығы мен дәйектілігі, есептеулерді орындау, анықтамалық тұтастық пен деректердің тұтастығын бақылау серверде шоғырланған. Клиент бөлігі тек серверде орындалған транзакциялардың нәтижелерін, серверде болған қателер туралы хабарламаны, сондай-ақ серверге қосу, жанарту және жою үшін деректерді беру үшін көрсетеді. Осылайша, біз қосымшаны әзірлеу архитектурасын ұстанамыз-қалың сервер, жұқа клиент [30].

Сыни тестілеу-бұл стандартты жұмыс кезінде бағдарламадағы қателерді табу процесі (іс-қимылдардың дұрыс реттілігімен, өрістерді дұрыс толтырумен және т.б.).

Тереңдетілген (кеңейтілген) тестілеу-бұл стандартты емес, күтпеген жағдайларда (қате енгізілген мәліметтермен) бағдарламадағы қателерді табу процесі.

5 Пайдаланушы нұсқаулығы

5.1 Бағдарламаны орнату және конфигурациялау бойынша нұсқаулық

Қосымшаның дұрыс жұмыс істеуі үшін Android құрылғысында келесі бағдарламалық жасақтама орнатылуы керек:

Android 4.4 және одан жоғары операциялық жүйе.

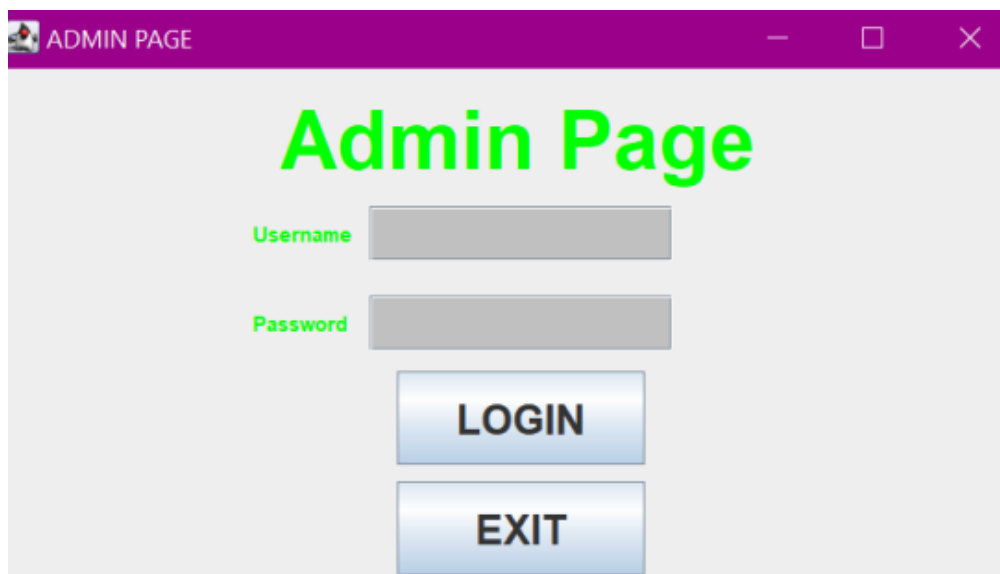
Кешенді Android құрылғысына орнату үшін келесі әрекеттерді орындау қажет:

- 1)Android құрылғысына apk файлын көшіру;
- 2)apk файлын іске қосыңыз және орнату.[32]

5.2 Кез келген пайдаланушы рөлінде жұмыс істеу

5.2.1 Қосымшаға кіру

Бағдарламаға кіру үшін оны android құрылғысында іске қосу керек. Нәтижесінде қосымшаның негізгі терезесі пайда болады, онда пайдаланушыға компаниямен жұмыс істеу үшін мәзір элементтерін таңдау ұсынылады.(5.1-сурет)



5.1-сурет – Қосымшаның негізгі терезесі

5.2.2 Контрагенттермен жұмыс

Контрагенттермен жұмыс істеу үшін компания менеджері қосымшаның негізгі терезесіндегі "Контрагент" тармағын таңдау керек. Нәтижесінде жаңа контрагенттер терезесі көрсетіледі (5.2-сурет), онда контрагенттер туралы ақпарат қол жетімді болады, сонымен қатар жаңаларын қосу мүмкіндігі қол жетімді болады. "Қосу" батырмасын басқаннан кейін жаңа контрагентті қосу терезесіне өту орын алады (5.3-сурет).

The image shows a login interface on a light gray background. At the top, the word 'USERNAME' is in bold black text above a white rectangular input field. Below this, 'PASSWORD' is in bold black text above another white rectangular input field. Under the password field is a blue button with the word 'LOGIN' in white. At the bottom, there are three blue buttons: 'FORGOT' on the left, 'SIGN UP' on the right, and 'EXIT' centered below them. All buttons have white text.

5.2-сурет – Контрагенттермен жұмыс істеу терезесі

The image shows a registration form on a light gray background. At the top, a blue button labeled 'REGISTRATION FORM' in white text is centered. Below it are four input fields: 'YOUR NAME' (a simple text box), 'PASSWORD' (a simple text box), 'SECRET QUESTION' (a dropdown menu with 'Your lovely country' selected), and 'YOUT ANSWER' (a simple text box). At the bottom, there are two blue buttons: 'BACK' on the left and 'CREATE' on the right, both with white text.

5.3-сурет – Контрагент қосу

Барлық өрістерді енгізіп, "Қосу" түймесін басқаннан кейін "контрагенттер"терезесінде жаңа контрагент қосылады және көрсетіледі.

Контрагенттермен бірге терезені қосуды және қайтаруды болдырмау үшін Android құрылғысындағы "Артқа" түймесін басу керек [33].

5.2.3 Жеткізушілермен жұмыс

Жеткізушілермен жұмыс істеу үшін компания менеджері қосымшаның негізгі терезесінде "жеткізуші" тармағын таңдау керек . Нәтижесінде жаңа жеткізушілер терезесі көрсетіледі , онда жеткізушілер туралы ақпарат қол жетімді болады, сонымен қатар жаңаларын қосу мүмкіндігі қол жетімді болады. "Қосу" батырмасын басқаннан кейін жаңа жеткізушіні қосу терезесіне өту орын алады (5.4-сурет).



NAME		SEARCH
QUESTION		
ANSWER		RETRIEVE
PASSWORD		
BACK		

5.4-сурет – Жеткізушілер терезесі

5.2.4 Қоймамен жұмыс

Қоймамен жұмыс істеу үшін компания менеджері қосымшаның негізгі терезесіндегі "қойма" тармағын таңдау керек . Нәтижесінде жаңа қойма терезесі көрсетіледі, онда тауарлар бар қойма туралы ақпарат қол жетімді болады, сонымен қатар қоймаға жаңа тауарларды қосу мүмкіндігі қол жетімді болады. "Қосу" батырмасын басқаннан кейін жаңа жеткізушіні қосу терезесіне өту орын алады (5.5-сурет).[34]

Name

Amount

Price

ADD

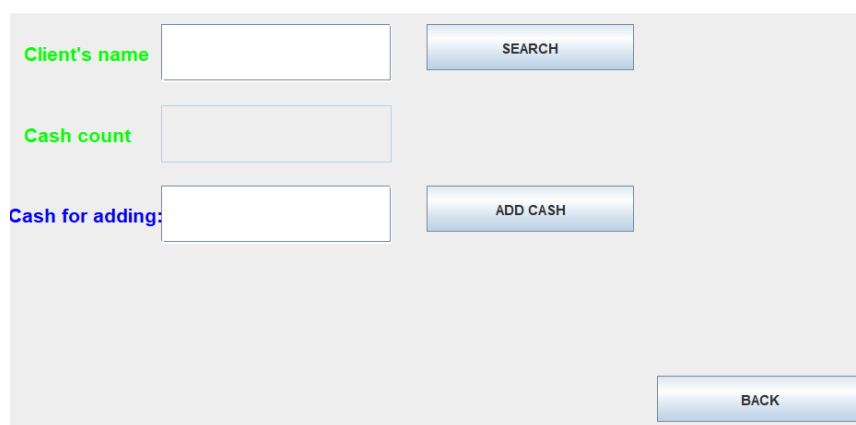
BACK

5.5-сурет – Тауарды қоймаға қосу

Барлық өрістерді енгізіп, "Қосу" түймесін басқаннан кейін жаңа өнім қоймаға қосылып, "қойма" терезесінде көрсетіледі.

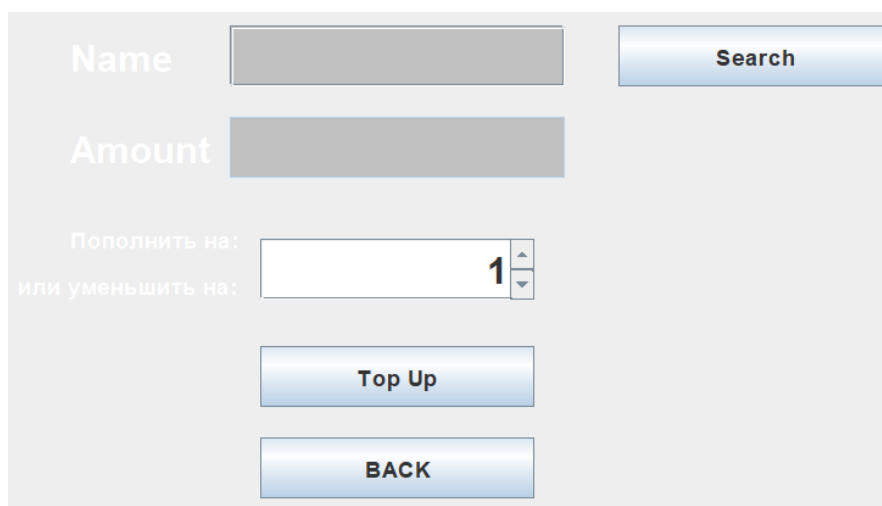
5.2.5 Тапсырыстармен жұмыс

Тапсырыстармен жұмыс істеу үшін компания менеджері қосымшаның негізгі терезесіндегі "тапсырыстар" тармағын таңдау керек. Нәтижесінде жаңа тапсырыстар терезесі көрсетіледі, онда тапсырыс туралы ақпарат қол жетімді болады, сонымен қатар жаңаларын қосу мүмкіндігі қол жетімді болады. "Қосу" батырмасын басқаннан кейін жаңа тапсырысты қосу терезесіне өту орын алады (5.6-сурет [34]).



The screenshot shows a web form for adding cash. It has three input fields on the left: 'Client's name' (with green text), 'Cash count' (with green text), and 'Cash for adding:' (with blue text). To the right of the first two fields is a 'SEARCH' button. To the right of the third field is an 'ADD CASH' button. At the bottom right, there is a 'BACK' button.

5.6-сурет – Тапсырысты қосу



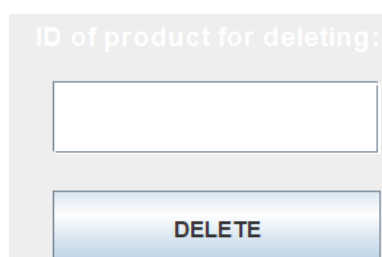
The screenshot shows a web form for topping up. It has two input fields at the top: 'Name' and 'Amount'. To the right of the 'Name' field is a 'Search' button. Below these fields is a section with the text 'Пополнить на:' and 'или уменьшить на:'. To the right of this text is a numeric input field with the value '1' and up/down arrows. Below this section are two buttons: 'Top Up' and 'BACK'.

5.7-сурет – Тапсырысты қосу

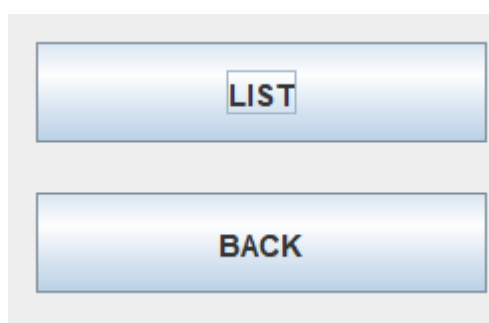
Барлық өрістерді енгізіп, "Қосу" түймесін басқаннан кейін "тапсырыстар" терезесінде жаңа тапсырыс қосылады және көрсетіледі. (5.7-сурет)

5.2.6 Тауарлармен жұмыс

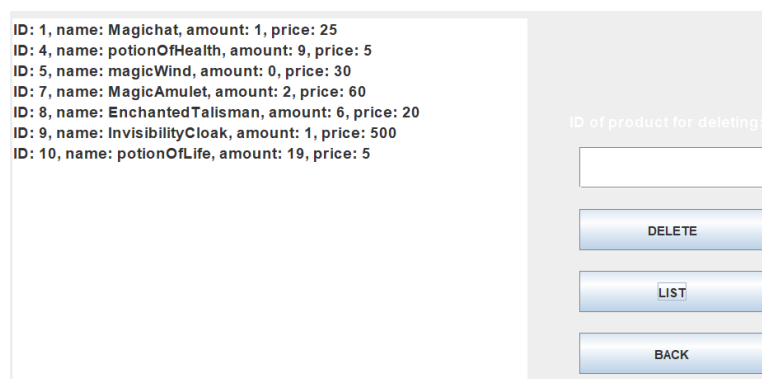
Тауарлармен жұмыс істеу үшін компания менеджері қосымшаның негізгі терезесіндегі "тауарлар" тармағын таңдау керек.(5.8-сурет) Нәтижесінде "тауарлар" жаңа терезесі көрсетіледі, онда тауарлар туралы ақпарат қолжетімді болады, сондай-ақ жаңаларын қосу мүмкіндігі қолжетімді болады. "Өнімді қосу" түймесін басқаннан кейін жаңа өнімді қосу терезесіне өтіңіз. Сондай-ақ, өнімнің жаңа санатын қосу мүмкіндігі бар, "санатты қосу" түймесін басқаннан кейін жаңа санатты қосу терезесіне өту керек (5.9-сурет) [35].



5.8-сурет – Тауарлармен жұмыс істеу терезесі



5.9-сурет – Тауар санатын қосу



5.10-сурет – Өнімді қосу

Барлық өрістерді енгізіп, "өнімді қосу" түймесін басқаннан кейін "өнімдер" терезесінде жаңа өнім қосылады және көрсетіледі.(5.10-сурет)

5.2.7 Деректерді жою

Тауарлар, тапсырыстар, контрагенттер мен жеткізушілер туралы жазбаларды жою бірдей түрде жүреді: керісінше, әр жазбада чек жәшігі бар және оны жою үшін қажетті мәліметтерге қарама-қарсы басқан кезде, деректер тізімінің үстінде "Жою" батырмасы пайда болады, оны басқан кезде жойылады [36].

ҚОРЫТЫНДЫ

Бұл дипломдық жоба Android платформасына арналған компанияда тауарларды есепке алу және жылжыту үшін мобильді бағдарламалық өнімді жасады. Ол үшін тақырып бойынша әдебиеттерге шолу жасалды, ұқсас бағдарламалық өнімдер талданды, деректердің логикалық және физикалық модельдері жасалды, мәліметтер базасы, бағдарламалық кешеннің серверлік және клиенттік бөліктері жобаланып, іске асырылды, экономикалық тиімділік есептелді.

Өзірленген жүйе келесі мүмкіндіктерді ұсынады:

- 1) мобильді қосымшаға фирма менеджерлерінің қол жеткізуі;
- 2) тауарлар, тапсырыс берушілер, жеткізушілер туралы тиісті деректерді қарап шығу. Өнім, тапсырыс берушілер, жеткізушілер туралы жазбаларды қосу, жою, өзгерту;
- 3) пайдаланушыға ыңғайлы және интуитивті интерфейс;
- 4) енгізілетін деректердің тұтастығын, дұрыстығын және сәйкестігін бақылау, сондай-ақ қате деректерді енгізу әрекетін болдырмау мүмкіндігі;
- 5) үш деңгейлі дизайн архитектурасының арқасында функционалдылықты оңай кеңейту және кеңейту мүмкіндігі.[37]

Тестілеу барысында бағдарламалық өнім айтарлықтай тұрақты жұмыс нәтижелерін көрсетті. Жүйе стандартты жұмыс режимінде, сондай-ақ Пайдаланушының әртүрлі рұқсатсыз әрекеттерінде сәтті жұмыс істейтіні анықталды. Осылайша, қойылған міндет толық көлемде орындалды.

Компанияда заманауи есепке алу жүйелерін қолдану:

- жеңілдету қызметін персонал фирманың;
- қызметкерлердің нәтижелілігін арттыру;
- тауарларды есепке алу және фирма қызметкерлері мен басшылық статистикасын бақылау бойынша тиімді және уақтылы жаңартылатын құралмен қамтамасыз ету;
- бүкіл тауарды есепке алу кезінде ықтимал қателіктердің ықтималдығын азайту;
- жұмыс орнынан тыс жерде де жұмыстың үздіксіздігін қамтамасыз ету;
- барлық қызметкерлердің тиімділігін арттыру.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Дон Гриффитс, Дэвид Гриффитс бірінші Android бағдарламалау, 2-ші басылым.: Ағылшын тілінен Пер. - "Петр", 2016. – 694 Б.: ил.
- 2 п.Дейтель, х. дейтель бағдарламашыларға арналған Android. Қосымшалар жасаңыз, 2-ші басылым.: Ағылшын тілінен Пер. - "Петр", 2016. – 694 Б.: ил.
- 3 Дон Гриффитс, Дэвид Гриффитс бірінші Android бағдарламалау, 2-ші басылым.: Ағылшын тілінен Пер. - "Петр", 2016. – 694 Б.: ил.
- 4 Barry Burd Android Application Development All-In-One For Dummies®; Свердловск: Клип-Москва, 2011. - 816 с.
- 5 Daniel A. Begun Таңғажайып Android Apps For Dummies®; Вильнюс: Вага - Мәскеу, 2011. - 256 с.
- 6 Donn Felker Android Application Development For Dummies; Wiley Publishing, Inc-Мәскеу, 2010. - 364 с.
- 7 Ed Brunette Hello Android 3e; жер-Мәскеу, 2010. - 300 с.
- 8 Ed Burnette Hello, Android; Басқа жағалаулар-Мәскеу, 2009. - 228 с.
- 9 Ed Burnette Hello, Android 2e; Минск: Артаал-Пресс - Мәскеу, 2009. - 250 с.
- 10 Jonathan Simon Head First Android Development; Ресей Суретшілер қауымдастығы АХР-Мәскеу, 2011. - 608 с.
- 11 Ken Ford Android Epistemology ; Car Tours-Мәскеу, 1995. - 334 с.
- 12 Onur Cinar Pro Android C++ with the NDK; Полиграф-Ақпарат - Мәскеу, 2012. - 300 с.
- 13 Rajiv Ramnath Android ?X? SDK Programming For Dummies®; М.: АСТ, сөз - Мәскеу, 2011. - 456 с.
- 14 Charles Bell Expert MySQL; Alma Classics-Мәскеу, 2012. - 550 с.
- 15 Paul DuBois MySQL; Addison-Wesley Professional-Мәскеу, 2008. - 506 с.
- 16 Philip J. Pratt, Mary Z. last a Guide to MySQL; БХВ-Петербург, Арлит-Мәскеу, 2012. - 304 с.
- 17 Robert Sheldon Beginning MySQL®; М.: Гарант - Москва, 2005. - 864 с.
- 18 How To do Everything With Php And Mysql; Мәскеу педагогикалық мемлекеттік университеті-Мәскеу, 2011. - 400 с.
- 19 Дронов Владимир PHP 5/6, MySQL 5/6 және Dreamweaver CS4. Интерактивті веб-сайттарды әзірлеу; БХВ-Петербург-Мәскеу, 2009. - 544 с.
- 20 РБ ДСМ 28.06.2013 ж. № 59 қаулысымен бекітілген "бейнедисплейлі терминалдармен және электронды-есептеуіш машиналармен жұмыс істеу кезіндегі нормаланатын параметрлердің шекті рұқсат етілген деңгейлері" санитарлық нормалары мен ережелері және "бейнедисплейлі терминалдармен

және электронды-есептеуіш машиналармен жұмыс істеу кезіндегі талаптар" гигиеналық нормативі.

21 РБ ДСМ 30.04.2013 ж. № 33 қаулысымен бекітілген "өндірістік және офистік үй-жайлардағы жұмыс орындарының микроклиматына қойылатын талаптар" санитариялық нормалары мен қағидалары және "өндірістік және офистік үй-жайлардың микроклимат көрсеткіштері" гигиеналық нормативі.

22 "Жұмыс аймағының ауасында регламенттелген зиянды заттардың тізбесі" санитариялық нормалары, қағидалары және гигиеналық нормативтері. – Мн.: Беларусь Республикасының Денсаулық сақтау министрлігі, 2009.

23 СНБ 4.02.01-03. Жылыту, желдету және ауаны баптау.

24 ТКП 45-2.04-153-2009. Табиғи және жасанды жарықтандыру. – Мн.: Беларусь Республикасының Құрылыс Министрлігі, 2010. – 104 б.

25 Санитарлық нормалар, ережелер және гигиеналық нормативтер. Жұмыс орындарындағы, көлік құралдарындағы, тұрғын үй, қоғамдық ғимараттардың үй-жайларындағы және тұрғын үй құрылысы аумағындағы Шу. – Мн.: Беларусь Республикасының Денсаулық сақтау министрлігі, 2011.

26 СанПиН 2.2.4/2.1.8.9-36-2002. Радиожиілік диапазонының электромагниттік сәулеленуі (ЭМИ РЧ).

27 ТКП 181-2009 "тұтынушылардың электр қондырғыларын техникалық пайдалану ережелері".

28 МЕМСТ 12.1.030-81. ЕҚСЖ. Электр қауіпсіздігі. Қорғаныс жерге қосу, нөлдеу.

29 ТКП 474-2013. Үй-жайларды, ғимараттарды және сыртқы қондырғыларды жарылыс-өрт және өрт қауіптілігі, утв бойынша санаттарға бөлу. Беларусь Республикасы ТЖМ 29.01.2013 ж. № 4 қаулысымен.

30 ТКП 45-2.02-142-2011. Ғимараттар, құрылыс конструкциялары, материалдар мен бұйымдар. Өрт-техникалық сыныптау ережесі.

31 ТКП 45-3. 02-90-2008. Өндірістік ғимараттар. Жобалаудың құрылыс нормалары.

32 ТКП 45-2.02-22-2006. Ғимараттар мен құрылыстар. Эвакуациялық жолдар мен шығу жолдары. Жобалау ережелері.

А қосымшасы (міндетті)

Техникалық тапсырма

Азық-түлік дүкені тауарларын есепке алу үшін мобильді қосымша құру

А.1 Жалпы сипаттама

Android платформасына негізделген азық-түлік дүкені тауарларын есепке алу үшін есеп қосымшасының мобильді нұсқасын жобалау және іске асыру, жүйенің параметрлерін конфигурациялау және оңтайландыру.

А.2 Мақсаты

Қосымшаның негізгі мақсаты - дүкендегі тауарлардың есебі мен қозғалысын ішінара автоматтандыру.

А.3 Функционалдық сипаттамаларына қойылатын талаптар

- бар тауарларды қарау;
- әр түрлі өлшемдер бойынша өнімдерді сүзу және іздеу;
- фирманың клиенттерін тіркеу;
- жеткізушінің деректерін өзгерту мүмкіндігі;
- жеткізушіні жою мүмкіндігі;
- фирманың клиентін өзгерту мүмкіндігі;
- фирма клиентін жою мүмкіндігі;
- өнім санатын қосу;
- тауарлар санатының өзгеруі;
- тауарлар санатын алып тастау;
- барлық кеңестер бар ыңғайлы, интуитивті интерфейс.

А қосымшасының жалғасы

А.4 Сенімділікке талап

Тауарларды есепке алудың ақпараттық жүйесі дұрыс және тұрақты жұмыс жасауы керек. Сол себепті сенімділік келесі талаптарға сай болуы керек:

- Ақпарат қатаң түрде мазмұнына, өріс атауларына сәйкес енгізілуі тиіс;
- Енгізілген ақпараттың тұтастығын қамтамасыз ету;
- Қолданушының қате әрекеттерінен қорғауды қамтамасыз ету.

А.5 Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар

Аппараттық құрал Android және iOS операциялық жүйесі бар мобильді құрылғыларды, сондай-ақ IBM-үйлесімді серверлік компьютерді қамтуы керек, оған мыналар кіреді:

- а) 2,80 ГГц немесе одан жоғары тактілік жиілігі бар Core Duo процессоры;
- б) сыйымдылығы 4 ГБ немесе одан да көп жедел жады;
- в) 100 ГБ немесе одан да көп сыйымдылығы бар қатты диск;
- е) Ethernet портының болуы;
- ғ) енгізу құрылғылары (компьютер тінтуірі мен пернетақта);

А.6 Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар

Қолданбалар мен сервер арасында ақпаратты тасымалдау HTTP(лар) протоколы арқылы жүзеге асырылуы керек.

Оқиғаларға өзгертулер тек әкімшінің, яғни персоналды басқару бөлімінің қызметкерінің қолданбасы арқылы енгізіліп, серверде сақталуы мүмкін. Келісім есептері негізгі ақпаратты қамтуы керек. Өтініш сұранысты барынша қысқа мерзімде орындауы керек

А қосымшасының жалғасы

А.7 Бағдарламалық қамтаманың құжаттамасына қойылатын талаптар

Бағдарламалық жасақтаманың құжаттамасы мыналарды қамтиды:

Техникалық тапсырма. Онда бағдарламалық қамтамасыз етуге қойылатын талаптар, әзірлеудің қажетті кезеңдері мен мерзімдері қамтылады.

Бағдарлама мәтіні – қажетті түсініктемелері бар бағдарлама жазбалары;

Логикалық құрылымы мен жұмысы туралы ақпаратты қамтитын бағдарламаның сипаттамасы.

Б қосымшасы
(міндетті)

Бағдарлама коды

```
package MainBody;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.sql.*;
import java.util.ArrayList;
import java.util.LinkedList;

public class Server {
    public static final int PORT = 8080;
    public static LinkedList<ServerSomthing> serverList = new LinkedList<>();

    public static void main(String[] args) throws IOException{

        ServerSocket server = new ServerSocket(PORT);
        System.out.println("Server started");
        try {
            while (true) {
                System.out.println("Waiting for client");
                Socket socket = server.accept();
                System.out.println("Client accept");
                try {
                    serverList.add(new ServerSomthing(socket));
                } catch (IOException e) {
                    //e.printStackTrace();
                    System.out.println("serverList.add error line 19 (Server)");
                }
            }
        } finally {
            server.close();
        }
    }
}
```

Б қосымшасының жалғасы

```
class ServerSomething extends Thread {

    private Socket socket; // сокет, через который сервер общается с клиентом,
    // кроме него - клиент и сервер никак не связаны
    private ObjectOutputStream outputStream;
    private ObjectInputStream inputStream;

    private static Connection connection = Connect.ConnectDb();
    private ResultSet resultSet;
    private PreparedStatement preparedStatement;
    private Statement statement;

    private Data data; //объект для передачи данных между сервером и
    клиентом

    public ServerSomething(Socket socket) throws IOException {
        this.socket = socket;
        start();
    }
    @Override
    public void run() {
        //int i = 0;
        try {
            outputStream = new ObjectOutputStream(socket.getOutputStream());
            inputStream = new ObjectInputStream(socket.getInputStream());
            data = new Data();
            try {
                while (true) {
                    try {
                        data = (Data) inputStream.readObject();
                        //i++;
                        //System.out.println("принял " + i);
                    } catch (ClassNotFoundException e) {
                        e.printStackTrace();
                    }
                }
            }

            else if (data.getOrder().equalsIgnoreCase("ForgotRetrive")) {
                try {
                    String a2 = data.getUser().getAnswer();
                    String n2 = data.getUser().getName();
                }
            }
        }
    }
}
```

Б қосымшасының жалғасы

```
String query = "select * from Account where answer = '" + a2 + "'  
and login ='" + n2 + "'";  
preparedStatement = connection.prepareStatement(query);  
preparedStatement.executeQuery();  
Statement statement = connection.createStatement();  
resultSet = statement.executeQuery(query);  
if (resultSet.next()) {  
    String pas = resultSet.getString("password");  
    try {  
        data.setChoiceOfServer(true);  
        data.getUser().setPass(pas);  
        outputStream.writeObject(data);  
        outputStream.flush();  
    } catch (IOException ignored) {  
    }  
    resultSet.close();  
} else {  
    try {  
        data.setChoiceOfServer(false);  
        outputStream.writeObject(data);  
        outputStream.flush();  
    } catch (IOException ignored) {  
    }  
    resultSet.close();  
}  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}  
else if (data.getOrder().equalsIgnoreCase("Registration")) {  
    try {  
        String name = data.getUser().getName();  
        String pass = data.getUser().getPass();  
        String sec_q = data.getUser().getSec_q();  
        String answer = data.getUser().getAnswer();  
        String query = "select * from account where login=?";  
        preparedStatement = connection.prepareStatement(query);  
        preparedStatement.setString(1, name);  
        resultSet = preparedStatement.executeQuery();  
        if (resultSet.next()) {
```

Б қосымшасының жалғасы

```
data.setChoiceOfServer(false);
outputStream.writeObject(data);
outputStream.flush();
resultSet.close();
preparedStatement.close();
} else {
    try {
        String query1 = "insert into Account (Login, password,
sec_q, answer) values (?, ?, ?, ?)";
        preparedStatement = connection.prepareStatement(query1);
        preparedStatement.setString(1, name);
        preparedStatement.setString(2, pass);
        preparedStatement.setString(3, sec_q);
        preparedStatement.setString(4, answer);
        preparedStatement.execute();
        data.setChoiceOfServer(true);
        outputStream.writeObject(data);
        outputStream.flush();
        resultSet.close();
        preparedStatement.close();
    } catch (Exception eg) {
        eg.printStackTrace();
    }
}
```

```
package MainBody;
```

```
import com.sun.security.jgss.GSSUtil;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.image.ImageObserver;
```

```
import java.awt.image.ImageProducer;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.io.ObjectInputStream;
```

```
import java.io.ObjectOutputStream;
```

Б қосымшасының жалғасы

```
import java.net.Socket;
import java.net.URL;
import java.sql.Connection;
import java.util.ArrayList;

public class Main {
    public static MainFrame mf;
    public static String ipAddr = "localhost";
    public static int port = 8080;
    public static Socket socket;
    public static ObjectOutputStream outputStream;
    public static ObjectInputStream inputStream;
    public static User userData;

    private static String myLogin;
    private static String myPassword;
    private static Integer myId;

    public static void main(String[] args) {
        connectServer();
        mf = new MainFrame();
        mf.setVisible(true);
    }
    public static void connectServer() {
        try{
            socket = new Socket(ipAddr, port);
            outputStream=new ObjectOutputStream(socket.getOutputStream());
            inputStream=new ObjectInputStream((socket.getInputStream()));
        }catch (Exception e){
            e.printStackTrace();
        }
    }

    public static void listProducts() {
        Data data = new Data();
        data.setOrder("listProducts");
        try {
            outputStream.writeObject(data);
            data = (Data) inputStream.readObject();
            ArrayList<Product> products = data.getProducts();
```

Б қосымшасының жалғасы

```
        mf.windowWithTheProduct.listProducts(products);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void checkCash() {
    Data data = new Data();
    data.setOrder("checkCash");
    data.setUser(new User(null, myLogin, null, null, null, null));
    try {
        outputStream.writeObject(data);
        data = (Data) inputStream.readObject();
        if (data.getChoiceOfServer()) {
            mf.windowWithTheProduct.checkCash(data.getUser().getCash());
        }
        else {
            mf.windowWithTheProduct.checkCash(null);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void buyProduct(Data data) {
    if (data != null) {
        try {
            data.getUser().setName(myLogin);
            data.getUser().setInteger(myId);
            outputStream.writeObject(data);
            data = (Data) inputStream.readObject();
            mf.windowWithTheProduct.dialog1(data.getChoiceOfServer(),
data.getOrder(), data.getUser().getCash());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    else {
        mf.windowWithTheProduct.dialog1(false, null, null);
    }
}
```

Б қосымшасының жалғасы

```
package MainBody;

import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.ArrayList;

public class MainAdmin {
    public static AdminFrame af;
    public static String ipAddr = "localhost";
    public static int port = 8080;
    public static Socket socket;
    public static ObjectOutputStream outputStream;
    public static ObjectInputStream inputStream;
    public static User userData;

    private static String myLogin;
    private static String myPassword;
    public static void main(String[] args) {
        connectServer();
        af = new AdminFrame();
        af.setVisible(true);
    }
    public static void connectServer() {
        try{
            socket = new Socket(ipAddr, port);
            outputStream=new ObjectOutputStream(socket.getOutputStream());
            inputStream=new ObjectInputStream((socket.getInputStream()));
        }catch (Exception e){
            e.printStackTrace();
        }
    }
    public static void addProduct(Data data) {
        try {
            outputStream.writeObject(data);
            data = (Data) inputStream.readObject();
            if (data.getChoiceOfServer()) {
                af.adminAddProduct.dialogError(true);
            }
            else {
                af.adminAddProduct.dialogError(false);
            }
        }
    }
}
```

Б қосымшасының жалғасы

```
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
public static void listProducts() {
    Data data = new Data();
    data.setOrder("listProducts");
    try {
        outputStream.writeObject(data);
        data = (Data) inputStream.readObject();
        ArrayList<Product> products = data.getProducts();
        af.adminDeleteProduct.listProducts(products);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
public static void deleteProduct(Data data) {
    try {
        outputStream.writeObject(data);
        data = (Data) inputStream.readObject();
        if (data.getChoiceOfServer()) {
            af.adminDeleteProduct.deleteProduct(true);
        }
        else {
            af.adminDeleteProduct.deleteProduct(false);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```