

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ
«Компьютерлік технологиялар» кафедрасы

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

_____ « _____ » _____ 2014 ж.
(КОЛЫ)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «Дистанционды оқытудағы тестілеу жүйесінің дамуы.
Кроссплатформалық мобильдік қосымша»

5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету
мамандығы бойынша

Орындаған: ВТк-10-2 Оспанов Д.С.

Жетекші: ф.-м.ғ.д., проф. Құралбаев З.Қ.

Кеңесшілер :

Экономикалық бөлім бойынша :

_____ « 02 » _____ 2014 ж.
(КОЛЫ) доцент Боканова Г.Ш.

Өмір тіршілігі қауіпсіздігі бойынша:

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

Есептеу техникасын қолдану бойынша :

_____ « _____ » _____ 2014 ж.
(КОЛЫ) ф.-м.ғ.д., профессор Құралбаев З.Қ.

Мөлшер бақылаушы:

_____ « _____ » _____ 2014 ж.
(КОЛЫ) аға оқытушы Рахимжанова З.М.

Пікір жазушы :

_____ « _____ » _____ 2014 ж.
(КОЛЫ) ҚазҰТУ, п.ғ.д., проф. Сатыбалдиев О.С.

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Ақпараттық технологиялар» факультеті
«Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы
«Компьютерлік технологиялар» кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Оспанов Дүйсен

Жоба тақырыбы «Дистанционды оқытудағы тестілеу жүйесінің дамуы. Кроссплатформалық мобильдік қосымша»
ректордың « ____ » _____ № ____ бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: « ____ » _____ 2014 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері):

MS Visual Studio 2012 бағдарламалау ортасы мен PhoneGap фреймворкы көмегімен HTML5, CSS3, JavaScript тілдерін пайдалану арқылы дистанционды оқытуды дамыту барысына арналған кроссплатформалық мобильдік қосымша құрылды.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

- Клиент пен сервер арасындағы байланысты анықтау.
- Мобильдік қосымшаны құрастыру барысында құрылғы платформаларымен танысу.
- Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, ER моделін жобалау, дерекқорды құру, пайдаланушы интерфейсін жобалау, дерекқормен байланысын орнату.

- Техникалық-экономикалық негізделуі.
- Өндірістегі еңбек қорғау мен қауіпсіздік шараларын ұйымдастыру.

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

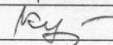

- Дерекқордың ER диаграммасы.

– Жасалған бағдарламалық қосымшаның жұмыс істеу принципі мен функционалдығын көрсететін графикалық материал.

Негізгі ұсынылатын әдебиеттер

- 1 Бондарь А.Г. MS SQL Server 2012. Создание баз данных и разработка программ. –БХВ-Петербург, 2013. – 608с.
- 2 Д. Петкович Microsoft SQL Server 2012. Руководство для начинающих. –БХВ-Петербург, 2013. – 816с.
- 3 Джо Майо: Самоучитель Microsoft Visual Studio 2010. – BHV, 2011г.– 464с.
- 4 Мамаев Е. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: BHV, 2004. – 416с.
- 5 Марк Шпеник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.

Жоба бойынша бөлімшелерге қатысты белгіленген кеңесшілер

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Құралбаев З.Қ.		
Тіршілік қауіпсіздігі	Муташева Г.С.		
Экономикалық бөлім	Боканова Г.Ш.		
Норма бақылаушы	Рахимжанова З.М.		
Есептеу техникасын қолдану	Құралбаев З.Қ.		

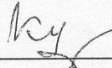
диплом жобасын дайындау

К Е С Т Е С І

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Зерттелу жүйе облысының анализі . (құрылымы, классификациясы)		
2	Деректер моделінің анықтамасы, дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.		
3	Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, ER моделін жобалау.		
4	Концептуалды проектилеу. Сервер мен клиент арасындағы байланысты анықтау		
5	Жүйенің техникалық бөлімін кескіндеу, қосымшаны тестілеуден өткізу		

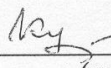
Тапсырманың берілген уақыты « _____ » _____ 2014ж.

Кафедра меңгерушісі


(қолы)

ф.-м.ғ.д., проф. Құралбаев З.Қ.

Жоба жетекшісі


(қолы)

ф.-м.ғ.д., проф. Құралбаев З.Қ.

Орындалатын тапсырманы қабылдаған студент _____

(қолы)

Оспанов Д.С.

Аннотация

В данной дипломной работе разработано приложение для всех желающих дистанционно обучаться по мобильным устройствам. В работе предусмотрены требования к серверной и образовательным частям приложения, условия эксплуатации и осуществление. Было уделено необходимое внимание функциональным характеристикам и техническим средствам.

В разделе обеспечения безопасности жизнедеятельности проведен анализ условий труда, сделан расчет освещения и кондиционирования.

В экономической части составлен бизнес план проекта, подтвердивший его экономическую целесообразность.

Андатпа

Бұл дипломдық жұмысы бойынша кез келген білім алушыға арналған дистанционды түрде білім алуға мүмкіндік беретін мобильдік қосымшаны құрастыру жүзеге асырылды. Дипломдық жұмыста қосымшаның қолданбалы серверлік және тест тапсыратын бөлімдеріне қойылған талаптар, пайдалану шарттары мен оны жүзеге асыру қарастырылған. Техникалық құралдар мен функционалдық сипаттамаларға тиісті шарттарға бөлінген.

Өміртіршілік қауіпсіздігін қамтамасыз ету бөлімінде еңбек ету жағдайына талдау жасалған, жарық беру және желдету есептелген.

Экономикалық бөлімде, жобаның экономикалық тиімділігін дәлелдеп бизнес-жоспарын жасалды.

Abstract

In the given Thesis Project an application for those who want to learn at a distance mobile devices.. The paper involves requirements for the server-based and educational parts of the application, operating conditions and implementation. Necessary attention was paid to the functional characteristics and facilities.

In the section of living safety provision working conditions were analyzed , as well as the calculation of lighting and air conditioning was provided.

In the economic part of the project was compiled a business plan, which confirmed its economic practicability.

Мазмұны

1	Негізгі бөлім	9
1.1	Кроссплатформалық қосымша жасау өзгектілігі	9
1.1.1	IOS, Android операциялық жүйелерінің платформалары.....	10
1.1.2	Мобильдік бағдарламаның нарықтық анализі.....	12
1.2	Мақсатты тағайындау және негізгі мүмкіндіктерді жобалау	13
1.3	Қосымшаның толық архитектурасы.....	13
1.4	Сервер бөлімінің қосымшаға талаптары.....	Ошибка! Закладка не определена.
1.4.1	Операциялық жүйе	16
1.4.2	HTML5 тілінің ерекшеліктері, артықшылығы.....	18
1.4.3	ASP.NET MVC4 құрылымы.....	19
1.4.4	SQL серверімен жұмыс істеу.....	20
1.4.5	CSS технологиясының қасиеттері.....	21
1.4.6	HTML тілінің синтаксисі мен құрылымы.....	25
2	Арнаулы бөлім.....	30
2.1	Зерттелетін жүйе облысының анализі.....	30
2.2	Деректер моделінің анықтамасы.....	34
2.3	Концептуалды проектилеу.....	35
3	Техникалық бөлім	37
3.1	Қосымша архитектурасы	37
3.2	Колданбалы интерфейстің зерттемесі	38
3.2.1	Сервер интерфейсі.....	38
3.2.2	Қолданушы беті.....	40
4	Тіршілік қауіпсіздігі.....	43
4.1	Еңбек шартының анализі.....	43
4.2	Есептеу бөлімі	44
4.2.1	Кондиционерлеу жүйесін есептеу.....	44
4.2.2	Табиғи жарықты есептеу.....	49
4.3	Тіршілік қауіпсіздігі бөлімі бойынша қорытынды.....	51
5	Техника-экономикалық негізделуі	52
5.1	Жоба сипаттамасы.....	53
5.2	Қосымшаның өңделуінің еңбек өнімділігінің есептелуі.....	53
5.3	Қосымшаны өңдеуге кеткен шығынның есептелуі	55
5.4	Қосымша шығындар статьясы	56
5.5	Өзіндік құн нәтижесінің кестесі	59
5.6	Қосымшаны иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу	61
5.7	Қолдану саласындағы жылдық бірмезгілдегі шығындарды есептеу	62
5.8	Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	62
5.9	Пайда мен шығындарды есептеу	63
5.10	Экономикалық тиімділікті	65
	Қорытынды	67
	Пайдаланылған әдебиеттер тізімі	68

А қосымшасы.....	69
Ә қосымшасы.....	85

Кіріспе

Қазіргі уақытта дистанционды білім алу жеделді түрде дамып келеді. Сонымен қатар жеке компания қызметкерлерінің, кәсіпорын жұмысшыларының біліктілігін арттыру үшін дистанционды оқыту курстары өткізілуде. Осыған байланысты дистанционды оқыту жүйесіне арналған мобильдік қосымша ұсынып отырмын.

Осы заманда Android, Windows Phone, IOS платформаларын қолдайтын ұялы телефон екінің бірінде бар десем қателеспеймін. Бұл платформалар ұялы телефон шығаратын компанияларда жоғары сұранысқа ие. Айта кетсек, IOS платформасында Iphone, Android платформасында HTC, Samsung, Windows Phone платформасында Nokia сияқты алдыңғы қатарлы ұялы телефон маркалары жұмыс істейді.

Дипломдық жұмысымның мақсаты - барлық операциялық жүйелерге арналған дистанционды түрде тестілеуден өтуге болатын кроссплатформалы мобильдік қосымша құрастыру. Жасалып отырған жоба кез-келген білім алушыға (студент, біліктілігін асырғысы келетін қызметкер) арналған. Қосымшаның көмегімен білім алушы өзінің уақытын үнемдеп, ыңғайлы әрі жеңіл түрде ұялы телефонының көмегімен дистанционды түрде тест тапсыруға мүмкіндік алады. Зерттеулер бойынша, таңдалған жобаға сәйкес Қазақстанда қосымша жасалынбаған және бәсекелестік мүлде жоқ десе де болады. Сондықтан, ең бастысы интернет желісіне байланысқа шығу мен дистанционды түрде білім беретін ЖОО табу болып табылады.

Android, IOS, Windows Phone платформалары - коммуникаторларға, планшеттарға, цифрлық ойнатқыштарға, цифрлық фоторамкаларға, қолға киетін сағаттарға, смартбуктарға, Linux (Unix) ядросында жасалған операциялық жүйеге орнатылған. Мобильдік қосымшаны жасау кезінде Phonegap фреймворкы арқылы мобильдік бағдарламалаудың негізгі тілдерін қолданбай-ақ, Javascript, HTML5, CSS3 тілдерін қолдана отырып жасадым. Ал мобильдік құрылғыға ақпаратты жіберіп, өңдеп және нәтижелерді тіркеп отыратын серверді құру үшін Visual Studio 2012, ASP NET MVC4, MsSQL бағдарламалық орталарын пайдаландым.

Жасалатын қосымша IOS, Android, Windows Phone платформалы кез келген мобильді құрылғыға арналған. Қосымшаны құруда ең басты шартымыздың бірі - жүйе жадында көп орын алмайтын, мобильді құрылғыда жедел түрде жүйеге қатысты жауар беретін қосымша ойлап табу болды. Міндеттелген мақсатымызға жету үшін мына талаптарды жүзеге асырумыз қажет:

- веб-қосымшаның компоненттік бөліктерін құрастыру, өмірлік үрдістке сай болуын қадағалау;
- веб-қосымшаны жобалап, жүзеге асыру әдістемесін зерттеу;
- жобаны талдау, сараптау және жеке бөліктерін зерттеу.

1 Негізгі бөлім

Жасалынып отырған қосымша білім алушыларға арналған. Дипломдық жұмыстың басты тапсырмасы дистанционды түрде тестілеуден өтуге болатын кроссплатформалы мобильдік қосымша құрастыру, қосымшаның негізгі қызметтері: тест тапсыру, тестілеуге дайындалу, тест нәтижелерін сараптау болып табылады.

1.1 Кроссплатформалық қосымша жасау өзгектілігі

Нарықтағы мобильдік платформалардың қатаң бәсекелестігі тек қана бір операциялық жүйеге бағытталған қосымша шығаруға мүмкіндік бермейді. Бұл жағдай құрастырушыны қиын таңдау жасауға мәжбүрлейді. Олар әр операциялық жүйе үшін өзіндік жеке қосымша құрастыру (әр платформа үшін жеке туған тілінде қосымша құрастыру), сайт түріндегі мобильдік қосымшаны HTML5 тілінде құрастыру немесе фреймворк арқылы кроссплатформалы қосымша жасау арасында таңдаулары керек болды.

Бизнес өзінің клиенттерімен мобильдік құрылғылар арқылы коммуникациялық байланыста болғандықтан, олардың мобильдік құрылғыларының сан алуандығына қолдаумен қамтамасыз етіп отыруы керек. Себебі, қолданушыға смартфонның белгілі бір моделін алуға мәжбүрлеу, мысалға алғанда белгілі бір банктің мобильдік қосымшасын сай келетін смартфонды алдыруға болмайды. Өйткені, қолданушы IOS-тан Android-ке көшуден бұрын банкті ауыстыруы мүмкін.

Мобильдік қосымша құрастырудың заманға сай дәуірі Iphone маркасының пайда болуынан бері есепке алу қабылданған. Егер нақтырақ айтып кетсек, мобильдік қосымшалардың тарихы 2008 жылдың 28 шілдесінде Iphone 3G шығуы мен AppStore дүкенінің ашылуынан бастау алады. Сол күні статистикаға сүйенсек 500 қосымша жаңа шыққан құрылғыға жүктелінуге мүмкін болған. Арада 5 жыл өтіп, 2013 жылдың маусымындағы Apple фирмасының бас директор Тим Куктың хабарлауы бойынша App Store онлайн-каталогындағы iPhone мен Ipad құрылғыларына арналған қосымшалардың саны 900 мыңнан асып түседі екен. Тағы да қосып кеткені Apple мобильдік дүкені 575 миллионнан астам аккаунтпен байланыста, және соңғы 5 жылдың ішінде компания қосымша құрастырушыларға 10 миллиард долларға жуық ақша төлеп, қолданушылардың сол қосымшалардың 50 миллиардқа жуық көшірмесін жүктеуіне себепкер болған. Және бұл, аналитиктардың бағалауы бойынша, IOS платформасы нарықтың жеке көшбасшысы емес. Себебі, Android нарықта екпінді түрде өсуде. Android-құрылғыларының қолданушылары Google Play дүкенінен 48 миллиардқа жуық қосымша көшірмелерін жүктегенін 2013 жылдың 15 мамырындағы компанияның құрастырушы топ-менеджерлері жиналған баспасөз жиымында Хьюго Барра хабарлады [1].

1.1.1 IOS, Android, Windows Mobile операциялық жүйелерінің платформалары

IOS - Apple компаниясы құрастырып, шығаратын мобильдік операциялық жүйесі. 2007 жылдан бастап iPhone мен iPod touch үшін, кейіннен iPad пен Apple TV құрылғыларына арналып шығарылды. Windows Phone мен Google Android операциялық жүйелеріне қарағанда, бұл платформа тек қана Apple фирмасының құрылғыларына ғана құрастырылады. IOS қолданушы интерфейсі тікелей қимыл концепциясына негізделіп, мультитач белгілерін қолданған. IOS OS X негізінде құрастырылып, және тура сондай Darwin негізгі компоненттерін қолданып, POSIX стандартына сай жасалынады. iOS-тың қолданушы интерфейсі мультитач қозғалыстары арқылы тікелей басқару концепциясында құрылған. Интерфейсті басқару элементтері батырмалардан, сүйрегіштер мен тетіктерден құрылған.

2007 жылдың 27 маусымында Apple фирмасы бірінші рет IOS пен бірінші iPhone ұялы телефонын жарыққа шығарды. Ең алғашқы операциялық жүйелер 1.x серияларымен жүріп отырды. Ал, келесі жылы IOS 2.0 операциялық жүйесі шығып 2.2.1 түріне дейін жаңартылды. 2008 жылы 10 маусымында WWDC 2008 баспасөз жиылымында бағдарламалық және аппараттық кемшіліктері жөнделген, телефонның саудаға түскеннен бері үшінші моделі - iPhone 3G таныстырылды. Бұл модель операциялық жүйенің жаңа нұсқасы iPhone OS 2.0 платформасында жұмыс істейді. 2009 жылдың 17 маусымында жарыққа жаңа iPhone ұялы телефонының шығарылуына байланысты OS 3.0 пайда болды. Бұл операциялық жүйесінің түрі 3.1.1 версиясымен шектелді.

Қазіргі таңда IOS 7.x сериялы жаңа операциялық жүйелер жарыққа шығуда. Бұл операциялық жүйеде кемшіліктерді барынша азайтып, қолданушыға жаңа интерфейс ұсынылған. Және қолданушының жеке ақпараттарының конфедициалдығына аса қатты көңіл бөлген. Сонымен қатар iFace, iCloud және т.б. қосымшалар жаңартылып, жүйенің жаңа замандық талапқа сай болуына себепкер болды.

Android - бұл Linux-қа негізделген мобильді платформа, Open Handset Alliance (ОНА) көмегімен құрастырылып, Google жетекшілігіне алынған ең алғашқы операциялық жүйе. Бұл платформа Google кітапханаларының басқарылуымен Java-қосымшаларын жасауға мүмкіндік береді. Сонымен қатар, Си тілі мен басқа тілдермен де Android Native Development көмегімен қосымша жазуға мүмкіндік бар.

Android платформасын қолданған ең алғаш құрылғы HTC компаниясы құрастырған T-Mobile G1 смартфонна болды. Google компаниясы Android платформасының басқа платформалардан өзгешелейтін бірнеше басымдылықтарды атады:

- Ашықтық - Android стандартты API шақырылымы көмегімен мобильдік құрылғының негізгі функцияларына рұқсат береді.

- Шектеулерді жою - Android үшін телефонның негізгі қосымшасы мен шеткі қосымшаның арасында айырмашылық жоқ. Android платформасында номерді теретін бағдарламаны немесе экран суретін де өзгертуге де болады.

- Жеңіл әрі ыңғайлы қолдану - SDK-да бәрі бар, Android қосымшасын жасау немесе іске қосу үшін керекті функциялар, шынайы құрылғының имитатора мен кеңейтілген дұрыстап, қалпына келтіру құралдары да бар.

Тағы айта кетеріміз, Android басқа да функционалды мүмкіндіктерге ие. Мысалы қосымшаны орындау үшін жадының аз ғана бөлігін пайдаланатын виртуалды Dalvik Java-машинасы қолданылады. Dalvik бірнеше қосымшамен бірге жұмыс істеуге мүмкіндік беріп, файлдарды арнайы мобильді құрылғылармен оптимизацияланған dex форматында ашуға рұқсат береді.

Android 2D/3D кескінін қолдайды. Деректерді сақтау үшін танымал аз көлемді СУБД SQLite қолданылады. Платформада LibWebCore (WebKit) кітапханасы бар, бұл өз кезегінде Android web-браузерінің қозғалтқышы болып табылады.

Бір сөзбен айтқанда, Android - мобильдік құрылғыларға арналған бағдарламалық платформа. Бұл өз тарапынан, операциялық жүйені, аралық қабаттың бағдарламалық қамтамасыздығын (middleware), және қолданушының негізгі қосымшаларын (e-mail клиент, күнтізбе, карта, браузер, байланыстар және т.б.) қамтиды.

Windows Phone - 2010 жылдың 11 қазанында жарық көрген Microsoft компаниясының басшылығымен құрастырылған мобильді операциялық жүйе. Сол жылдың 21 қазаннан бастап жаңа платформада құралған алғашқы мобильді құрылғылар сатылымға түсе бастады. Windows Phone ұқсас болмаса да Windows Mobile операциялық жүйесінің толықтай жаңартылған интерфейсі бар және алғашқы рет Microsoft-тың Xbox Лайв және Zune медиаплеері қызметі интеграцияланған жолындағызбасары болып есептеледі. Жүйенің тұсаукесері Испания мемлекеті Барселона қаласында өткен Mobile World Congress 2010 шарасында жарыққа шықты.

2011 жылдың ақпан айында Mobile World Congress 2011 барысында бірінші рет Windows Phone операциялық жүйесінің жаңартылған түрін ұсынды. Ал сәуір айында Лас-Вегас қаласындағы MIX'2011 конференциясында бұл операциялық жүйесі туралы толық ақпарат беріп, Mango деген атқа ие болды. Ал одан кейін, бұл операциялық жүйенің ресми аты 7.5 болып қабылданды. Жаңа нұсқадағы операциялық жүйеде 500-ден аса жетілдіру мен толықтырулар енгізілді. Жүйеде Explorer Mobile браузері жаңартылып, HTML5 пен Canvas тілдерінің аппараттық жылдамдықтары арттырылды.

Ал, 2012 жылдың ақпан айында Windows Phone ресми блогында құрастырушылар 2013 жылдың басында 7.8 версиялы, Windows Phone 8 платформасының біршама жаңалықтарымен қосатын жаңа операциялық жүйе шығатындығы жайлы хабарлады. 2013 жылдың 30 ақпанынан бастап қолданушылар өз құрылғыларындағы платформаны 7.8 версиясына дейін жаңартуға мүмкіндің алды.

2012 жылдың 20 маусымында Microsoft компаниясы Windows Phone 8 платформасына шолу жасады. Windows Phone 8 жаңалығы болып қауіп төндіретін сайттарды анықтайтын функциясы бар Internet Explorer Mobile 10 браузері таныстырылды. Windows Phone 8 ең басты басымдылығы болып -

планшет, смартфон, дербес компьютерлерге ортақ «экожүйе» ойлап табылды. Бұл жүйе бойынша құрастырушыларға қолайлы шарттар жасалынды. Жаңа ұрпақ платформасы NT ядросында негізделген және құрылғылардың максималды рұқсат етілу көрінісі 1280x768 немесе 1280x720 болып қабылданды. Айта кетейік, бұрынғы Windows Phone 7.8 операциялық жүйесі CE ядросында негізделіп, максималды 800x480 көрінісін ғана ұсына алған еді.

1.1.2 Мобильдік бағдарламаның нарықтық анализі

Мобильдік құрылғының дамыған құрастырушылары күннен күнге басымдылығын арттыруда. Қазір екі қазақстандықтың бірінде ұялы телефон бар, көпшілігінде смартфондар мен планшеттер бар. Олардың әрқайсысында өзіндік қосымшалары бар. Одан бөлек, құрылғы шығаратын компанияға тәуелсіз құрастырушылар қосымшалар базасын толықтыруда. Міне осы жерден миллиондаған қолданушылар құрылғыдағы онлайн дүкеннен мобильдік қосымшаларды жүктейді. Орташа алғанда, бір қосымшаға iPhone қолданушылары - 2.15 доллар, Android қолданушылары - 4.57 доллар, ал BlackBerry қолданушылары - 5.61 АҚШ долларын төлейді.

Бүгінгі күнде мобильді қосымшалар нарығында үш компанияның басымдылығын байқауға болады. Егер оларды танымалдылығы жағынан санап айтатын болсақ, әлем бойынша бірінші орында Apple7, өзінің IOS платформасымен алады (iTunes, App Store). Екінші орында Android платформасы (Android Market) және үшінші орынды Windows Phone (Windows marketplace) меншіктейді. Тағы да Symbian, BlackBerry, HP webOS деген платформалары қолданыста бар.

IDC анықтауы бойынша 2013 жылдың бірінші ширегіндегі IOS пен Android платформаларының нарықтағы үлесі 92.3 %, ал қалған бөліктерді Windows пен BlackBerry өзара бөлісуде.

Қазір құрастырушылар Microsoft жасаған Windows Phone 8 платформасы мен Nokia компаниясымен серіктестігінен үлкен үміт күтіп отыр. Nokia қызметі мен сатылымының оңтүстік СНГ-дағы менеджері Елена Каришманның сөзі бойынша, соңғы жылдары Қазақстанда Nokia мобильдік құрылғысына арналған қосымшалардың жүктелуі 10 еседей өсті. Жалпы жүктелу саны қазір 10 миллионды құрайды және әлі де өсу барысында.

Шетел эксперттері бұрындары мобильдік қосымшаларға деген сұранысты болжап қойып еді. 2009 жылдардың өзінде маркетингтік-аналитикалық зерттеулер агенттігі IDC хабарлауы бойынша 2013 жылға дейін әлемдегі 1.19 миллиард маман (бұл әлем бойынша жұмыс күшінің 34.9% құрайды) мобильдік технологияны қолданатындығын болжаған.

Gartner зерттеуі бойынша, мобильдік шешімнің өсімі ақпараттық технология саласында бұлттық есептеуден кейінгі номер екінші әлемдік тренд болып табылды. Эксперттер сенімді, 2014 жылы 79.6 миллиард мобильдік қосымша жүктеліп, ал мобильдік қосымша индустриясы 35 миллиард АҚШ долларын құрайтын болатындығына.

Әлемдік тәжірибеде сәтті қосымшаның сатылымы немесе таратылымы кезінде бизнесіесіне қыруар ақша алып келеді. Болжау бойынша 2014 жылдың аяғына дейін мобильдік қосымшалар нарығы 15 миллиард АҚШ долларынан жоғары пайданы құрастырушылар мен компанияларға табыс ретінде әкеледі.

Қазастандық қолданушыларында көбірек қажеттілікті шетел ойын қосымшалары тудырады. Екінші орында навигациялық, ақпаратты-танымдылық және қоғамдық қосымшалар иемденеді. Үшінші орында мобильдік банкинг қызметі мен қызметтерді төлеу жүзеге асыра алатын Отандық бизнес қосымшалар алады [7].

1.2 Мақсатты тағайындау және негізгі мүмкіндіктерді жобалау

Бағдарлама дистанционды түрде білім алушыларға тестілеуден өтуге мүмкіндік береді.

Дистанционды түрде тестілеу бағдарламасын жобалаудың негізі болып келесі талаптарды қарастырамыз:

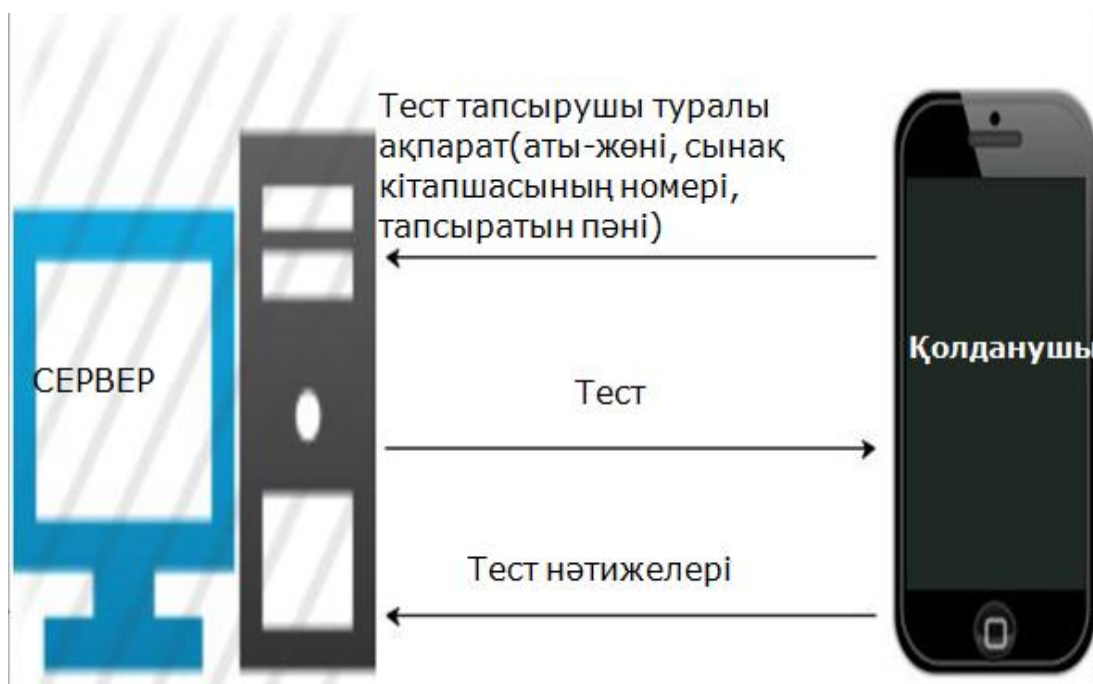
- жүйе жеңіл және қолданушыға түсінікті болуы қажет;
- қолданушы өзіне ыңғайлы мобильдік құрылғымен тестілеуді өтуі керек;
- қолданушы тест тапсырмастан бұрын, тестілеуге дайындалуға мүмкіндігі болуы керек;
- тест аяқталысымен қолданушы өз нәтижесін сараптауға болатындығы.

Құрастырып отырған қосымша Play Market, App Store онлайн қосымша дүкендеріне кіруге мүмкіндігі бар. Ол үшін жасалып отырған қосымшамыз барлық технологиялық және техникалық мағлұматтарға қамтулы.

1.3 Қосымшаның толық архитектурасы

Келесі сатыда мағлұматтарды кезекпен орындап, оны итерациялық тілге негіздеуді іске асырамыз.

Сурет 1.1-де негізгі жасалатын қосымшаның архитектуралық құрылымы «қолданушы-сервер» аясында орындалады. Қарастырып отырған мағлұмат желінің жеделділігі жылдам қолдануы. Сервер дегеніміз желідегі сақталып отырған деректерге қол жеткізу. Клиентпен сапалы түрде қарым-қатынасты іске асыру үшін ғаламторды пайдаланамыз.



Сурет 1.1 - Қосымшаның толық архитектуралық сипаттамасы

Архитектураның басты сипаттамасы және клиент-сервер айырмашылықтары:

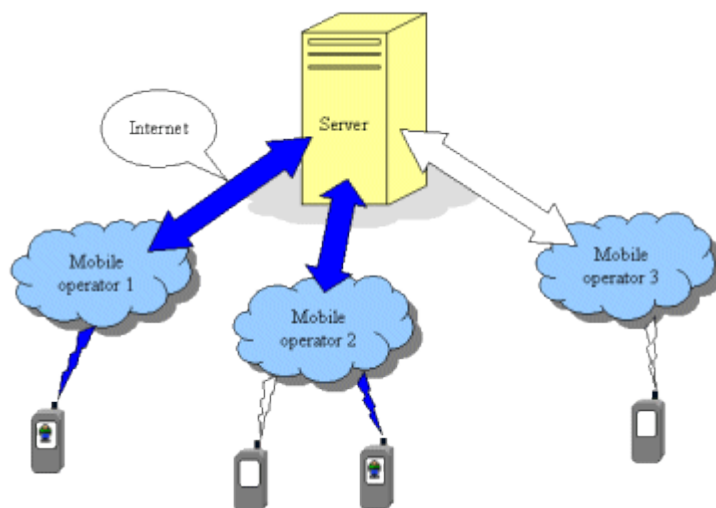
- Көптеген қолданыстағы мүмкіндіктер, есептік жүйенің функцияларының, жеке компьютерлер арасындағы желілер арқылы байланысын толығымен қарастырады. Бұл мүмкіндіктер қолданушыға және қосымша құрастырушыға үлкен жеңілдіктер ұсынады. Қосымша толықтай жұмыс жасамаған кезде, қосымша құрастырушы серверге кірмей, қосымшаны толығымен өзгерте алады, сонда клиентке жаңадан қосымшаны жаңарту туралы ақпарат жіберіледі.

- Бүкіл ақпараттар серверде сақталады, осылай деректеріміз толығымен қауіпсіз жерде сақталып, қол жетімді жерде орналасады. Ал серверде деректерді өзгерту, жаңарту және қадағалау жеңіл түрде өтеді [2].

1.4 Сервер бөлімінің қосымшаға талаптары

Көздеген мақсатымыз ғаламтор желісі арқылы орындалатын болғандықтан, ғаламтор тақырыбына қысқаша тоқталып өтуіміз керек. Ғаламтор арқылы қосымшаны серверге қосуымыз қрек. Сол кезде ғана HTTP-сұранысы арқылы протоколымыз беріліп, жүзеге асырылады. Мобильдік қосымшаның жеке IP мекен-жайы болады, сол арқылы платформадағы ОЖ-ның түрін анықаймыз. Кез келген құрылғы өзіндік IP мекен-жайды өзіне меншіктейді.

Жобаланып жатқан қосымшамыз көлемді санды операцияларды оқиды. Сурет 1.2-де процестің бастапқы құрылымы мен толығымен оның серверлік құрлымын бейнелейді.



Сурет 1.2 - қосымшаның сервер мен клиент жұмысы

Алдыңғы қатарлы дамыған интернет-технологиялардың да негізі HTML болып табылады. HTML тілі ғаламтордағы құжаттарды белгілеу мен көріктендіру үшін қолданылады. HTML тілінің туындауы алыстағы 1986 жылға әкеліп соғады. Сол жылы бірінші рет Халықаралық стандарттау ұйымы (ISO) көмегімен ISO-8879 стандарты қабылданып, «Standard Generalized Markup Language» деген атпен бекітілді.

Ең басында HTML тілі құжаттарды көрсету құралдарына байланыссыз қылып, тек қана редакциялық түрде пайдаланылды. Арнаулы түрдегі HTML белгісі бар текст құрылғыда әр алуан техникалық қамтамасыздандырумен стилистикалық, құрылымдық бұрмалаусыз, кедергісіз өндірілуі керек (жеке компьютердің түрлі түсті көрінісі, монохромды экраны органайзердің, көрініс өлшемі бойынша шектеуі бар ұялы телефон құрылғысының немесе текстерді дыбыстап ойнату бағдарламаларының және құрылғылардың көрінісі). Бірақ, осы таңда HTML тілінің қолданылу аясы өзінің алғашқы қолданысынан әлдеқайда шеттелген. Мысалы, бірнеше мәрте беттерді форматтау қолдансақ <TABLE> тегі құжаттарда қарапайым кесетелер жасауға арналған. Ал бертін келе HTML тілінің платформаға тәуелсіздігінің басты идеясы мультимедиалық және графиктік рәсімдеу мүмкіндіктеріне берілген.

Браузерлер. HTML белгісі бар мәтіндік құжаттар (бұл құжаттардың кеңейтілуі .html немесе .htm), пішімделген түрде көрсетілетін арнайы бағдарламалармен өңделеді. «Браузерлер» немесе «интернет-шолушылар», деп аталатын мұндай бағдарламалар әдетте қолданушыларға веб-беттер сұранысы, оларды қарауға және (басқа сыртқы құрылғыларға шығару үшін) және қажеттілік жағдайында қолданушылармен енгізілген мәліметтерді серверге жіберу үшін ыңғайлы интерфейсті ұсынады. Қазіргі кезде ең танымал браузерлер Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome және Opera болып табылады.

Нұсқалары HTML 1.0 тілінің ресми түрдегі спецификациясы жоқ. 1995 жылға дейін HTML тілінің көптеген бейресми түрдегі стандарттары болды. Ресми түрдегі стандарттарды олардан айыру үшін оған екінші нөмірді меншіктеді.

1995 ж наурызында 3 нұсқасы бүкіләлемдік тордың Консорциумымен ұсынылды (W3C) және көптеген кестелерді құру, мәтіндердің суреттерді «айналып өтуі», күрделі математикалық формулаларды көрсету сияқты жаңа мүмкіндіктерді қамтамасыз етті. Және бұл стандарт екінші нұсқамен сәйкес болған кезде де сол кездегі браузерлер үшін оны жүзеге асыру қиын болды. 3.1 нұсқасы ресми түрде ешқашан ұсынылған жоқ және HTML стандартының нұсқасы 3.2 болды. Бұл нұсқаға Netscape Navigator және Mosaic браузерлері қолдайтын стандартты емес элементтер қосылған.

HTML 4.0 нұсқасында стандартта кейбір «тазартылулар» болды. Көптеген элементтер ескірген және ұсынылмаған (ағ. deprecated) болып белгіленді. Әдетте шрифт қасиетін өзгертуге қолданылатын font элементі ескірген деп белгіленді (оның орнына CSS каскадты кестелер стилі қолдану ұсынылды).

1998 ж. бүкіләлемдік тордың Консорциумы HTML 4 тіліне негізделген, бірақ XML синтаксисіне сәйкес белгісі бар жаңа тілмен жұмыс істей бастады. Соңында атауы XHTML болды [4].

WHATWG бірлестігімен (ағ. Web Hypertext Application Technology Working Group) 2004 ж. бастап, HTML тілін кеңейтетін, жиі бейресми «HTML 5» деп аталатын және әр түрлі типтік беттердің семантикасын жақсы көрсету үшін, мысалы форумдар, аукциондар сайты, іздеу жүйелері, онлайн-дүкендер және т.б. үшін Web Applications 1.0 спецификациясы өңделіп жатыр.

HTML-құжаттың құрылымы. HTML тілі өз алдына элементтер жиынтығын келтіреді және онда әр элементтің басы мен соңы арнайы белгілермен — тегтермен белгіленеді. Элементтер бос болуы мүмкін, яғни ешқандай мәтін мен басқа да мәліметтер болмауы мүмкін (мысалы, жолды ауыстыру тегі
). Мұндай жағдайда әдетте жабушы тег көрсетілмейді. Одан басқа, элементтерде қандай-да бір қасиеттерді анықтайтын атрибуттар болуы мүмкін, (мысалы, font элементі үшін шрифт өлшемі). Атрибуттар ашылушы тегте көрсетіледі. HTML-құжат фрагментінің мысалы:

```
<strong> Екі тег арасындағы мәтін - ашылушы және жабылушы  
</strong>;
```

```
<a href="http://www.example.com"> мұнда элементтің href.</a>  
атрибуты бар [3].
```

1.4.1 Операциялық жүйе

Операциялық жүйе — компьютерді қосқан кезде ең бірінші жүктелетін және басқа бағдарламаларды қамтамасыз ететін жүйе. Жүйелік бағдарламаларды байланыстыратын жиыны, оның мағынасы компьютерді

қолданушымен байланысын ұйымдастырады және барлық басқа бағдарламаларды қамтамасыз етеді. Операциялық жүйе көбінесе компьютердің сыртқы жадында, дискіде сақталады. Компьютерді қосқан кезде, ол дисктік жадтан тексеріледі де, ОЕС-да (ОЗУ) орналасады. Осы процесті операциялық жүйенің жіктелуі деп атайды. Операциялық жүйенің қызметіне кіреді:

- қолданушымен сұқбатты жүзеге асыру;
- енгізу-шығару және деректермен басқару;
- бағдарламалардың өңдеу процесін жоспарлау және ұйымдастыру;
- ресурстарды орналастыру (оперативтік жадпен кэш жадты, процессорді, сыртқы құрылымды);
- бағдарламаларды орындалуға енгізу;
- бірнеше көмекші операцияларды қамтамасыз ету;
- әр түрлі сыртқы құрылымдар арасында ақпараттарды беру;
- перифериялық құрылымдарды бағдарламалық қолдау жұмыстары (дисплей, пернетақта, диск жинақтауыш, принтер және т.б.).

Операциялық жүйені компьютердің құрылымдарың басқару бағдарламасының жалғасы дейміз. ОЖ төрт негізгі түрі бар: Бір пайдаланушылық бір есептік, олар бір пернетақтаны қолдайды және тек бір есеппен жұмыс істей алады; Бір пайдаланушылық бір есептік фондық мөрімен, негізгі есеппен бірге қосымша есепті еңгізе алады және ақпаратты мөрге бағыттайды. Бұл көп көлемді ақпараттарды мөрге жіберу жұмысың тездетеді.; Бір пайдаланушылық көп есептік ,бір қолданушыға көп есептермен жұмыс жасап,өңдей алады. Мысалы, бір компьютерге бірнеше принтерлер қосуға болады және әр қайсысы өз жұмысын атқарады; Көп пайдаланушылық көп есептік, бір компьютерде бірнеше қолданушылармен бірнеше есептер еңгізе алады. Осы ОЖ өте күрделі және бірталай машиналық ресурстарды керек етеді.

Операциялық жүйе, ОЖ- комплексті бағдарлама, бастапқы функционалды мағлұматтарды компьютерге немесе серверге беретін жүйе немесе дербес компьютер ретінде қарастырса болады.

Операциялық жүйеге байланысты,бағдарламаның серверлік бөліміндегі қолданылатын негізгі талаптары:

- спектрлік желілік технологияға толық (желілік құрылығы, протокол, жойылып кеткен бағдарламаға жол ашу қызметі және тағы басқа) сүйеніш жасау;
- көпесептілік;
- көпағындылық;
- көпқолданбалы режим;
- әр түрлі платформадағы бағдарлама (64 биттік процессор, қатты дискке сақтау жүйесі және тағы басқа);
- кеңейтімділік;
- қауіпсіздікті қамсыздандыру деңгейі;
- лицензиялық көшірменің төмен құны [4].

1.4.2 HTML5 тілінің ерекшеліктері, артықшылығы

HTML5 - дүниежүзілік ғаламтордың құрылымы мен мазмұнының және оның қолдану аясының тілі болып табылады. HTML - дің бесінші нұсқасы үлгі ретінде 2012 жылдың сәуіріндегі өңдеу сатысындағы жағдайда қалып отыр. Оның негізгі мақсаты тілді жақсарту, жаңа мультимедиялық ұсыныстардың жұмыстарын қолдайтын, сонымен қатар адамдар үшін кодты оңай оқу және компьютерлер үшін орындаудың анықтығы мен HTML5-дің HTML4-пен қоса қалыптасуын қамтамасыз етеді.

Өзінің ізімен келе жатқан ізбасарлары ретінде HTML4.01 және XHTML 1.1, HTML5 дүниежүзілік торда негізінен қолданылатын HTML мен XHTML-дің талаптарына жауап бере алады. Сонымен қатар HTML5 болашақтағы әртүрлі мамандандырылған құрылымының негізі ретінде енгізілген бағдарламамен қамтамасыз етуге арналған жалпы практикада қолданылатын веб-браузер, сонымен қатар веб-құжаттарда кездесетін көптеген синтаксистік қателерді түзеу болып табылады. Сонымен қатар HTML5 HTML-да сонымен қатар HTML-да жазылуы мүмкін белгілерді анықтайтын әрі синтаксистік жағынан түзететін бірыңғай тіл мүмкіндігіне ие болады. Өзіне қосымша ретінде процессордың өзара әрекеттесуін ұзақ сақтау үшін үлгілердің бөлшектерін өңдеуге жауап бере алады: ол кеңітеді, жақсартады, және құжаттарға қажетті белгілерді тиімді түрде бөлшектейді әрі күрделі веб-ұсыныстарға арналған API мен белгілерді шығарады. HTML5 көптеген синтаксистік ерекшеліктер пайда болады. Мысалы, video, audio және canvas элементтері, сонымен қатар SVG пайдалану мүмкіндігі бар. Бұндай жаңалықтар API мен жеке плагииндарды қажет етпейтін мультимедиялық және графикалық басқару мен оны енгізу үшін жүзеге асырылады. Басқа қанда `<article>`, `<header>` және `<nav>` сияқты жаңа элементтер құжаттардың семантикалық мазмұнын байытуға байланысты шығарылған. Бір қатар элементтер мен атрибуттардың алынып тасталғанына қарамастан, осы мақсатта жаңа атрибуттар енгізілді. Бірнеше элементтер мысалы, `<a> menu` және `<cite>` өзгертілді, қайтадан нақтыланды немесе қалыпқа келтірілді.

API және DOM HTML5 -дің спецификациясының негізі бөлігі болып табылады. HTML5 сонымен қатар рұқсат етілмейтін құжаттардың бір қатар бөліктерін анықтай алады, сондықтан синтаксистік қателер агенттерді пайдаланушылар мен бейімделген браузерлердің барлығына бірдей қаралады.

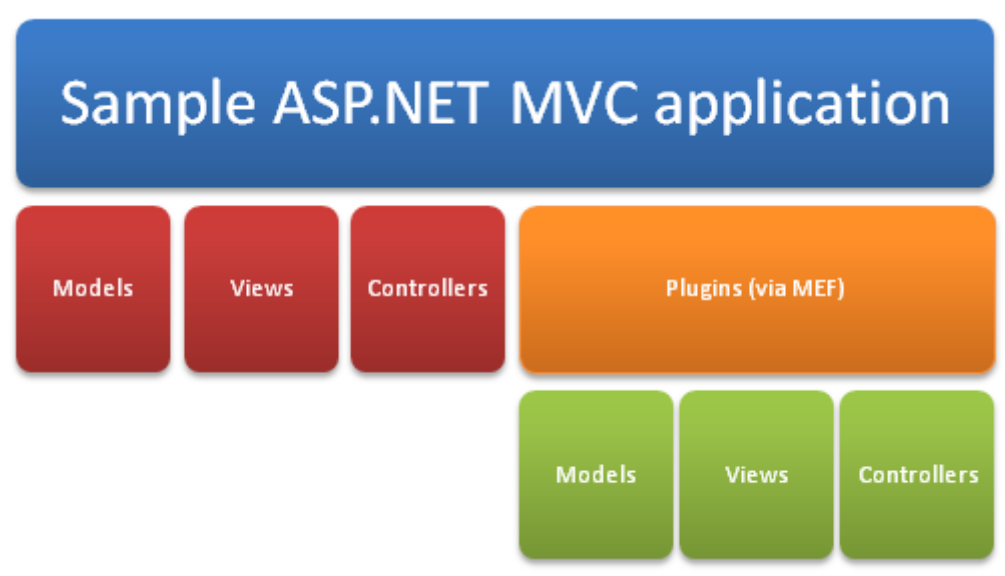
Вебтағы (WHATWG) Шағын мәтіннің Қолданбалы Технологиясының жұмыс тобы жаңа үлгіні жүргізу жұмыстарын World Wide Web Consortium (W3C) XHTML 2.0 жобаларын анықтауға арнады, өйткені 2.0, а HTML 4.01 2000 жылдан бері өзгермеген еді. 2009 жылы W3C XHTML 2.0-дың жұмысының уақыты өткенін мойындап, оны жаңалау туралы шешімге келді. Қазіргі уақытта W3C мен WHATWG HTML5-тің жобасымен бірге жұмыс жасайды.

HTML5-тің веб-құрастырушылардың арасында бірнеше жылдан бері жақсы таныс болғандығына қарамастан, 2010 жылдың сәуірінен ғана БАҚ-ның

негізгі тақырыбына айналды. Осыдан кейін ғана Apple Inc Стив Джобс компаниясы көпшілікке хат жазды, оның кіріспе бөлімінде былай делінген: « Flash-қа қатысты ойлар », оның HTML5 жасасқан жобаларына қатысты бейнежазбаларды Adobe Flash көмегі арқылы көру немесе пайдаланудың қажеттілігі жоқ екенін атап өтті. Сондықтан да осы мәселеге қатысты веб-құрастырушылардың арасында пікірталастар орын алды, онда HTML5 және кеңейтілген қызметтер түрімен қамтамасыз еткенмен құрастырушылар браузерлердің өзгешелігі мен стандарттың әртүрлі бөліктерін қолдаудың қажеттігін HTML5 мен Flash-тің арасындағы қызметтік өзгешеліктермен қатар назарда ұстау керек.

1.4.3 ASP.NET MVC4 құрылымы

Model-View-Controller (MVC) құрылысы сызбасының қосымшасы үш негізгі бөлікке жіктеледі: модель, қойылым және контроллер. ASP.NET MVC платформасы веб-қосымша құрастыруда ASP.NET веб-үлгі сызбасына балама бола алады. ASP.NET MVC платформасы кең көлемде сынақ жүргізу көрінісінің жеңіл салмақтағы платформасы болып табылады және сурет 1.4-те көрсетілген. Сонымен қатар веб-үлгі негізі, ASP.NET қолданып жүрген қызметтерін жандандырады, мысалы басты бет және мүшелік негізінің дұрыстығын тексереді.



Сурет 1.4 - MVC құрылымының сызбасы

MVC стандартты шаблон жобасы бойынша көптеген мамандарға таныс. Веб-қосымшалардың бірнеше түрлері MVC платформасын құруда артықшылыққа ие. Басқаларға ASP.NET веб-үлгі негізінде және кері байланыста дәстүрлі қосымшаларын пайдалану тиімді болуы мүмкін. Кейбір жағдайларда екі жүрісті үйлестіру: бір сызбаны қолдану екіншісін пайдалануға кедергі келтірмейді.

MVC платформасына мынадай компоненттер кіреді.

Модельдер. Модельдің нысаны болып, аталған қосымшаның домені логикасын жүзеге асыратын қосымшаның өзі болып табылады. Модельдің нысаны үнемі дерекқорды қабылдап, ондағы модельдердің жағдайын сақтайды. Мысалы, Product нысаны дерекқорынан ақпарат алуы мүмкін, онымен жұмыс жасап, содан соң жаңартылған деректерді Products кестесіндегі SQL Server дерекқорына жазып алады.

Шағын қосымшаларда бұл модель заңды бөліктерге емес, концептуальды түрде бөлінеді. Мысалы, егер қосымша деректі тек қана жинап алып, көрініске жіберетін болса, онда модельдердің класқа байланысты заңды қабаты болмайды. Мұндай жағдайда деректерді жинау модельдің нысаны ролін атқарады.

Қойылым. Қойылым интерфейс қосымшасын пайдаланушыға қызмет етеді. Пайдаланушы интерфейсін әдетте берілген модельдердің негізінде құрылады. Үлгі ретінде қызмет ететін мәтіндік жолдарды құрайтын, жалаушалар мен тізімдерді бояйтын Products кестесін редакциялау қойылымы Product нысанының маңызы ағымдағы жағдайына негізделген.

MVC сызбасының күрделі құрылымын нығайтуды толықтыру сонымен қатар қосымшаны тестілеуді веб-үлгі негізіндегі ASP.NET қарағанда жеңілдетеді. Мысалы, ASP.NET веб-қосымшасындағы веб-үлгінің негізінде бір класс нәтижені көрсетуге және нәтижеге қолданушының жауабына пайдаланылады. ASP.NET веб-үлгісі негізіндегі қосымшаларға арналған автоматты түрде тестілеуді құруда қиындықтар туындауы мүмкін, өйткені жеке парақтарды тестілеу үшін парақтардың кластық топтамасын барлық басқарудың серіктес қосымшалар класына қарамастан құрастыру керек болады.

1.4.4 SQL серверімен жұмыс істеу

SQL Server Enterprise Manager тұтынушы идентификаторын құрудың жай ғана амалы. Ол үшін SQL Server 7.0 программалық топтан SQL Server Enterprise Manager активтендіріледі және қолданушы идентификаторын қосатын серверді белгілейді. Осыдан кейін тышқанның оң жақ пернесімен Login папкасын басып, контекстік меню New Login командасын таңдаймыз. General қабықшасында жаңа қолданушы идентификаторын енгізіп жанама түрде мүмкіндік алатын идентификаторының тілі мен мәліметтер базасын таңдаймыз.

Бұл жерде кім кіру құқығын тексеретіндігі маңызды. Егер Windows NT authentication қосылғышын таңдасақ және ашшылатын Domain тізімінде домен SQL Server қолданушының жүйеге кіргеннен кейін қосымша тексеруін қажет етпейді деп санайды.

Талап бойынша тексеруді SQL Server-ге де жүктеуге болады. On SQL Server authentication қосылғышын орнатқаннан кейін. Бұл параметрлер анықталғаннан кейін келесі Server Roles-ке көшуге болады. SQL Server 7.0 орналастырғанда бірнеше стандартты рольдер құрылады. Олардың ішіне өзіңіздің құқығыңызға сәйкес рольдерді таңдайсыз:

- мәліметтер базасының кез-келген рольдері мен басқа да қосымша және конфигурациялау рұқсатын шешетін қимылдарды орындайды;
- мәліметтер базасындағы қолданушылар тізіміне топтар мен Windows NT, SQL Server қолданушыларын қосуға;
- кез-келген мәліметтер базасы кестесінен ақпараттар алуға;
- кез-келген мәліметтер базасы кестесінен қосуға, жаңалауға, алып тастауға;
- мәліметтер базасында нысандарды қосуға, жаңалауға, алып тастауға;
- рольдер мен SQL Server рольдер мүшелері және мәселені шешу құқы және мәліметтер база нысандарына кіру мүмкіндіктерін басқару;
- мәліметтер базасын резервтеу операциясын орындауға;
- мәліметтер базасы схемасын модификациялауға, бірақ оны оқымауға;
- мәліметтер базасындағы ақпараттарды жаңалауға (модификациялауға) рұқсат емес.

Идентификаторды құрудың "стандартты" түрінен басқа арнайы мастер-программасын қолдануға болады. Мастердің "сұрақтары" жоғарыда қарастырғаннан аз ғана айырмашылығы бар. Сондықтан мұны қысқаша түрде қарастырамыз:

- SQL Server's программалық тобынан SQL Server Enterprise Manager-ді қосамыз. Сонымен қатар үйренген Master Create Login Wizard-ды іске қосамыз;
- бірінші терезе жалпы мәліметтер болғандықтан Ok - басамыз және екі терезеге көшеміз. Мұнда қолданушы идентификаторын кім тексеретінін таңдау қажет.

Байқағанымыздай бұл мүмкіндіктер бойынша шектелген. Сондықтан менің ойымша бірінші тәсілді қолдану керек [7].

1.4.5 CSS технологиясының қасиеттері

Web-құжаттың құрылымы мен сипаттауын бөлектеуді CSS (Cascade Style Sheets - каскадты стильдер кестесі) технологиясын қолдана отырып орындаймыз. CSS технологиясы Web-құжатты сипаттауда қосымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі.

Стильді анықтау құжаттың түрін стиль қасиеттерін енгізу арқалы өзгертетеді. Стиль қасиеттерінің мәні ретінде кілттік сөздер алынады. Кілттік сөздер арасында кездесетін | символы, қасиеттің осы немесе келесі мәнді қабылдайтынын білдіреді. Төменде осы технологияның негізгі қасиеттерін қарастырайық.

Font-family: [шрифттер тізімі] – Шрифттер немесе олардың типтері үтірмен бөлетеніп көрсетіледі. Браузер шрифттерді көрсетілген ретпен компьютерде кездескенше іздеп, табылған шрифтті пайдаланады. Қасиеттің шрифт типтері түрінде төмендегі мәндер алынуы мүмкін:

- serif - түртпесі бар шрифттер, мысалы: Times;
- sans-serif - түртпесі жоқ шрифттер, мысалы: Helvetica;
- cursive - курсивті шрифттер, мысалы: Zarf-Chencery;

`fantasy` - декоратив шрифттер, мысалы: `Western`;
`monospace` - моношрифттер, мысалы `Courier`.

Егер шрифт типі бірнеше сөзден тұрса, онда шрифт типін тырнақша ішінде жазу қажет, мысалы: “`Times New Roman Cyr`”.

Егер бірнеше шрифт | белгісімен көрсетілсе, браузер олардың біреуін үндеместен алу принципі бойынша таңдайды.

`Font-style: normal | italic | oblique` - шрифт стилдерін көрсетеді. Үндеместен `normal` алынады. Мұндағы:

`normal` - қалыпты;

`italic` - курсив;

`oblique` - қиғаш.

`Font-size: [абсолют] | [салыстырмалы] | [нақты] | [пайыз]` - шрифт өлшемін анықтайды. Қасиеттің мәні ретінде абсолют, салыстырмалы, нақты өлшемдер немесе пайыз көрсетіледі. Үндеместен `normal` алынады.

`Font-weight: normal | bold | bolder | lighter` - шрифттің енін анықтайды. Үндеместен `normal` алынады. Мұндағы:

`normal` - қалыпты;

`bold` - жартылай жуан;

`bolder` - жуан;

`lighter` - жіңішке.

`Font-variant: normal | small-caps` - шрифттің бас қаріпінің өлшемін анықтайды. Үндеместен `normal` алынады. Мұндағы:

`normal` - қалыпты;

`small-caps` - кіші бас қаріп.

`Font: [font-style] || [font-variant] || [font-weight] || [font-size] || [font-family]` - жоғарыда сипатталған қасиеттердің жалпыламасы, яғни шрифт қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

```
h1 {
font-family: Arial;
font-size: 30 pt;
font-weight: bold;
font-style: italic
}
p {
font: bold italic large Times New Roman, serif
}
word-spacing:
```

`normal` | [ұзындық] - сөздер арасындағы аралық беріледі. Үндеместен `normal` алынады.

Letter-spacing: normal | [длина] - қаріптер арасындағы аралық анықталады. Үндеместен normal алынады.

Text-decoration: underline | overline | line-through | blink | none - Қаріп гарнитурасы көрсетіледі. Мұндағы:

underline - астын сызу;

overline - төбесін сызу;

line-through - үстін сызу;

blink - жанып-өшу;

none - алдыңғы орнатылған мәндерді жою.

Text-transform: capitalize | uppercase | lowercase | none – мәтінді бейнелеуде регистрді түрлендіруді анықтайды. Үндеместен none алынады. Мұндағы:

capitalize - әрбір сөздің бірінші қаріпі бас қаріпке түрленеді;

uppercase - барлық әріптер бас қаріпке түрленеді;

lowercase - барлық әріптер кіші қаріпке түрленеді;

none - алдыңғы орнатылған мәндерді жояды.

Text-align: left | right | center | justify - мәтінді терезе шеттеріне теңестіреді.

Мұндағы:

left - солға;

right - оңға;

center - ортаға;

justify - екі шеттен.

Text-indent: [абсолют бірліктер] | [пайыз] – мәтіннің шегінуін көрсетеді. Әдетте шегіну абзацтың бірінші қатарының сол жағына ғана тиісті. Егер бірлік мәні сол таңбалы болса, онда салыстырмалы түрде солға қарай жылжыйды. Абсолют бірліктер төмендегідей анықталады:

px - логикалық пиксельдер;

pt - логикалық пункттер;

em - шрифт өлшемінің бірліктері;

cm - сантиметрлер;

mm - миллиметрлер.

Мысалы:

```
text-indent: -3em.
```

Vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | [пайыз] - элементтің тік орналысуын анықтайды. Үндеместен baseline алынады. Мұндағы:

baseline - элементті негізгі элементтен төменге орналастырады;

middle - элементтің орта сызығын «негізгі + жартылай биіктікке» теңестіреді;

sub - элементті қатар астына теңестіреді (әдетте font-size: smaller қасиетімен бірге беріледі);

`super` - элементті қатар төбесіне теңестіреді (әдетте `font-size: smaller` қасиетімен бірге беріледі);

`text-top` - элементтің жоғарғы шетін негізгі элементтің төбесімен теңестіреді;

`text-bottom` - элементтің төменгі шетін негізгі элементтің астымен теңестіреді;

`top` - элементтің төбесін негізгі элементтің ең жоғарғы шетімен теңестіреді;

`bottom` - элементтің астын негізгі элементтің ең төменгі шетімен теңестіреді;

`line-height: normal | [сан] | [ұзындық] | [пайыз]` - қатар биіктігін көрсетеді. Үндеместен `normal` алынады;

`color:[түс]` - құжат мәтінінің түсін көрсетеді. Үндеместен құжатта анықталған түсті алады. Түстің мәні бірнеше тәсілмен анықталады:

- түс аттары бойынша, мысалы: `red` (қызыл), `blue` (жасыл);
- он алтылық сан түрінде, мысалы: `#000000` (қара);
- нөлдері алып тасталған он алтылық сан түрінде, мысалы: `#fff` (ақ);
- RGB моделі бойынша, мәндері 0-ден 255-дейін, мысалы: `rgb(255,255,255)` (ақ);
- RGB моделі бойынша пайызбен берілген, мысалы: `rgb(100%, 0%, 0%)` (қызыл).

Егер түс атымен берілсе, онда браузерлерде әр түрлі бейнеленуі мүмкін.

- `background-color: [түс] | transparent` - құжаттың фон түсін анықтайды. Фонның түсін анықтаумен қатар, `transparent` арқалы фон асты түсін де көрсетуге болады. Үндеместен `transparent` алынады.

- `background-image: url ([url]) | none` - фон ретінде суретті файлдан алады. Мысалы:

- `background-image: url ("foto.jpg"). none` - фондық сурет берілмейді.

- `background-repeat: repeat | repeat-x | repeat-y | no-repeat` - фондық суретті қайталайды. Үндеместен `repeat` алынады. Мұндағы:

`repeat-x` - фондық сурет көлделенеңінен қайталанады;

`repeat-y` - фондық сурет тігінен қайталанады;

`no-repeat-y` - фондық сурет бір-ақ рет көрсетіледі, қайталанбайды;

`repeat` - фондық сурет көлделенеңінен де, тігінен де қайталанады;

`background-attachment: scroll | fixed` - фонды мәтінге байланыстырады. Үндеместен `scroll` алынады. Мұндағы:

`scroll` - егер мәтін терезеге симаса, онда фон мәтінмен қоса жүреді;

`fixed` - фон орнында тұрады, ал мәтін оның үстінде қозғалады;

`background-position:`

`top | center | bottom | left | center | right [пайыз] [ұзындық]` - фондық суреттің басталу координаталарын анықтайды. Фондық сурет координаталары кілттік сөздермен тігінен және көлденеңінен беріледі.

Фондық суреттің тігінен басталу координаталары төмендегіше беріледі:

top - жоғарыдан;

center - ортадан;

bottom - төменнен.

Фондық суреттің көлденеңінен басталуы төмендегіше беріледі:

left - солдан;

center - ортадан;

right - оңнан.

Координаталар тігінен бастап, сонан соң көлденеңінен көрсетіледі.

Мысалы: Суретті жоғарғы оң жақтағы бұрышқа орналастырайық, онда `background-position: top right`.

Координаталар пиксельде - (px), сантиметрде - (cm) немесе пайызда - (%) берілуі де мүмкін. Мысалы:

`background-position:50px 50px` немесе `background-position: 50%50%`;

`background:[background-color] || [background-image] || [background-repeat] || [background-attachment] || [background-position]` - жоғарыда сипатталған қасиеттердің жалпыламасы, яғни фондық сурет қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

```
body (background-image:url("suret.jpg");
background-repeat: repeat-y; background-attachment: fixed;
background-position: 1004 1004 );
p (background: url(suret.gif) red 50% repeat fixed).
```

1.4.6 HTML тілінің синтаксисі мен құрылымы

Әрбір тілдің өзіндік құрылымдық логикалық ерекшеліктері болатыны сияқты HTML тілінің де өзіндік ерекшеліктері бар. Алайда ол ең қарапайым, қатаң логикалық ережелерге онша бағына бермейтін тәуелсіз тіл. Дегенмен оның да өз синтаксисі, басқарушы құрылымдары бар. Тілдің негізгі басқарушы құрылымы дескриптор (кейде элемент дщеп те аталады). Дескриптор – белгілі-бір әрекетті анықтайтын HTML элементі. Дескрипторлардың мүмкіндіктері мәндерге ие бола алатын атрибуттарды қолдану арқылы толығымен ашылады [3].

HTML дескрипторлары дербес ережелерге бағынумен қатар, жалпы ережелерге де бағынады. Бірінші ереже бойынша барлық дескрипторлар “кіші” (<) және “үлкен” (>) символдарының арасына жазылады, мысалы:

<HTML>

Келесі ереже дескрипторды ашу және жабу компоненттерін қолдануға арналған. Дескрипторды жабу “кіші” (<) және “үлкен” (>) символдарының арасына қисық сызық (/) символын қою арқылы іске асырылады:

```
</HTML>
```

Бұл ашылған дескрипторды белгілі-бір жерде жабу керектігін білдіреді. Бірақ көптеген дескрипторлар бұл ережеге бағынбайды. Мысалы, абзацтар үшін қолданылатын <P> дескрипторы, тек ашылады, оны жабудың қажеті жоқ.

Жалпы ережеге дескрипторларды төменгі не жоғарғы регистрде жазу тәсілі де кіреді. Әдетте түсінікті болу үшін дескрипторлар жоғарғы регистрде жазылады. Бірақ кодтаудың жаңа әдіснамасы – XHTML-да (кеңейтілген HTML) дескрипторларды төменгі регистрде жазу қалыптасқан.

Дескрипторларды жазудағы тағы бір маңызды мәселе бос орындар (пробел) қалдыру. HTML кодтарының кейбір тұстарына бос орын қалдыру міндетті болса, кейде артық қалдырылған бос орын браузерді қатты шатастыратыны сонша, ол ешқандай да информация бере алмай қалады.

Интернетте электрондық құжат құру үшін HTML атаулы арнайы т3л пайдаланылады. Қазіргі кезде HTML тілін терең білмей –ақ, Microsoft Front page программасын пайдаланып, Web-беттерді құруға болады. Web- түйін біртұтас информациялық блок болып табылатын, өзара байланысқан Web беттер қорларынан тұратын жиынтық.

Web түйін құру үшін One Page Web шаблону пайдаланылады. Берілген шаблонда Web түйіндер құрылымы жасалып, кейіннен информациямен толықтыратын бір таза бет қосылады. Web беттің негізгі мақсаты бұл

Жоғарыда ескертіп кеткеніміздей атрибуттар дескрипторлардың әрекетін анықтайды. Көп жағдайларда дескрипторлар атрибутсыз жұмыс жасайды. Алайда кейде міндетті түрде атрибуты болуы тиіс дескрипторлар кездеседі.

Атрибуттар дескриптордың атауынан кейін бір бос орын тасталып, мән меншіктеліп жазылуы керек. Дескрипторда бірнеше атрибут болса, олар бос орын арқылы ажыратылады. Мысалы:

```
<BODY bgcolor=#FFFFFF text=#000000>
```

мұндағы bgcolor – фон түсі, text – мәтін түсі.

HTML-де дескрипторға атрибуттар тіркеліп, атрибуттарға мәндер меншіктеліп жазылған сөйлем жол деп аталады.

Атрибуттарға міндетті түрде мәндер меншіктелуі тиіс. Бұлай болмаған жағдайда HTML дескрипторды қабылдамайды. Матрибуттарға меншіктелетін мәндер түрліше болуы мүмкін: сандық, мәтіндік, пайыздық, т.с.с. Мысалы, <DIV> - бөлім дескрипторының align атрибутына left (сол жақ шеті), right (оң жақ шеті), center (ортасы), justify (ені бойынша) мәндердің бірін меншіктеу жолы:

```
<DIV align= "right">
```


DIV міндетті түрде жабылуы тиіс дескриптор болғандықтан оны </DIV> арқылы жабу керек.

Сол сияқты (шрифт) дескрипторының size (өлшем атрибуты 1-ден 7-ге дейін мәндер қабылдайды, мұндағы 1 ең кіші шрифт өлшемі, ал 7 ең үлкен шрифт өлшемі) атрибутына мән меншіктеу жолы:

```
<FONT size= "5">
```

Бұл кезде браузердегі мәтіннің шрифт өлшемі дескрипторы жабылғанша 5 өлшемін қабылдап тұрады - .

Сандық мәндердің келесі бір типі онақтылық код түрінде беріледі. Бұл HTML-де пайдалануға мүмкін түстердің онақтылық санау жүйесінде алфавиттік-цифрлық коды. Мұндай мысал жоғарыда келтірілген болатын. Мысалдағы, FFFFFFF – ақ түсті фон, ал 000000 - кара мәтін түсін береді. Онақтылық мәндердің алдына # символы қойылады.

HTML-де сілтеме дескрипторы - <A> арқылы Web-адресі шығаруға болады:

```
<A href= "http://www.yahoo.com/"> </A>
```

Осы сияқты каталогты және кескін файлдары дескрипторын қолдануға болады:

```
<IMG src= "suretter/aspan.gif/">
```

Өздеріңіз байқағандай барлық атрибуттардың мәндерінің алдына = (теңдік белгісі) таңбасы қойылады және онақтылық мәндерден басқа типтегі мәндер тырнақшаға (“ ”) алынып жазылады. Бұл HTML-да жол жазудың жалпы ережелерінің бірі.

Арнайы символдардың көмегімен HTML бетке HTML кодын шығаруға болады. Ол үшін “кіші” және “үлкен” символдарының орнына & символын пайдалану керек, мысалы:

```
&lt; IMG src= "aspan.gif" &gt;
```

HTML қабықшасы. HTML қабықша – беттің тақырыбы (HEAD) және негізгі дескрипторлардан тұратын денесі (BODY) орналасқан қарапайым дескриптор. Тақырыбы және денесі бар HTML дескриптордың жазылу үлгісі:

```
<HTML>  
<HEAD>  
<TITLE> </TITLE>  
</HEAD>  
<BODY>  
</BODY>
```

</HTML>

мұндағы, <TITLE> </TITLE> жолы құжат тақырыбын береді. Құжат тақырыбы келесі мәліметтерді қамтиды:

Беттің аталуы. <TITLE> және </TITLE> дескрипторлары беттің аталуын анықтайды. Ол HTML құжаттың денесінің құрамына кірмейді.

Стиль. HTML құжатты басқарудың элементтері мен стилдерін тақырыпқа енгізу қажет болған жағдайда қолданылады.

Метаинформация. META дескрипторы құжат авторы, кілттік сөздер, арнайы әрекеттер қажет болған жағдайда қолданылады.

Кез келген құжатты құрғанда оның басқару элементтері мен стильдері, авторы, кілттік сөздер жайлы мәліметтер онша қажет бола бермейтіндіктен аталған дескрипторлардың барлығы бірдей жиі қолданыста емес.

Дегенмен әрбір құжаттың атауы болуы тиіс, сондықтан <TITLE> дескрипторы жиі қолданылады.

Құжат денесінің құрылымы. HTML құжаттың денесі орналасатын негізгі бөлігі <BODY> және </BODY> дескрипторлары арқылы іске асырылады. Құжат денесіне енгізілетін информация келесідей болуы мүмкін:

Мәтін. Құжат құрамында кездесетін мәтіндер. Олар оқылуы жеңіл әрі түсінікті болу үшін форматталуы да мүмкін.

Графикалық кескін. Құжатта жылжуға немесе информацияны көрнекі түрде беру үшін енгізілетін графикалық информация.

Сілтеме. Web түйіннің ішінде оңай жылжуға және желінің басқа да тұстарына жылдам өтуге арналған сілтемелер.

Мультимедиа және арнайы программалар. Видеороликтер, Java апплеттерімен, Flash, Shockwave программаларымен жұмысты басқаратын дескрипторлар құжат денесінде орналасуы тиіс.

Web каталогтары кітапхананың жүйелік каталогтары тәрізді жасалып, олар «спорт», «компьютерлер», «ойындар» сияқты рубрикаларға жіктелген. Каталогтан керекті тақырыпты тауып алып, оның ішінен Web –тегі түйінді сөздер тізімінен іздеуді бастауға болады.

Көптеген серверлердегі тізімдерге әрбір түйінді тақырып қысқаша сипатталып отырады да, одан негізгі тақырыпқа өту жолы көрсетіліп тұрады.

Түсініктемелер. Құжатта қолданылып жатқан дескрипторлар мен әрекеттерге берілетін түсініктемені құжат денесінің кез келген тұсына енгізуге болады. Түсініктеме үлкен HTML-беттер құрғанда өте тиімді. Түсініктеме <!-- --> дескрипторы арқылы жазылады. Форматтау элементтері. Мәтінді форматтау – HTML тілінің ең қарапайым және ең қуатты құралы. Себебі Web-бет форматталған мәтіннен тұрғанда ғана түсінікті әрі әсем болып көрінеді. Мәтінді форматтауда келесі дескрипторлар қолданылады:

<h1>..<h1>-<h6>..<h1> дескрипторлар диапазоны тақырыптар мен тақырып мазмұнына қолданылады;

. Каретканы қайтару дескрипторы;

<p>..</p>. Абзацты ашу/жабу дескрипторлары;

... Мәтінді жартылай қалың ету;
<i>..</i>. Мәтінді көлбеу ету;
<u>..</u>. Мәтін астын сызу.

2 Арнаулы бөлім

Бұл бөлімде қосымшаны жобалау және мобильдік қосымшаға деректер қорын құру қарастырылады.

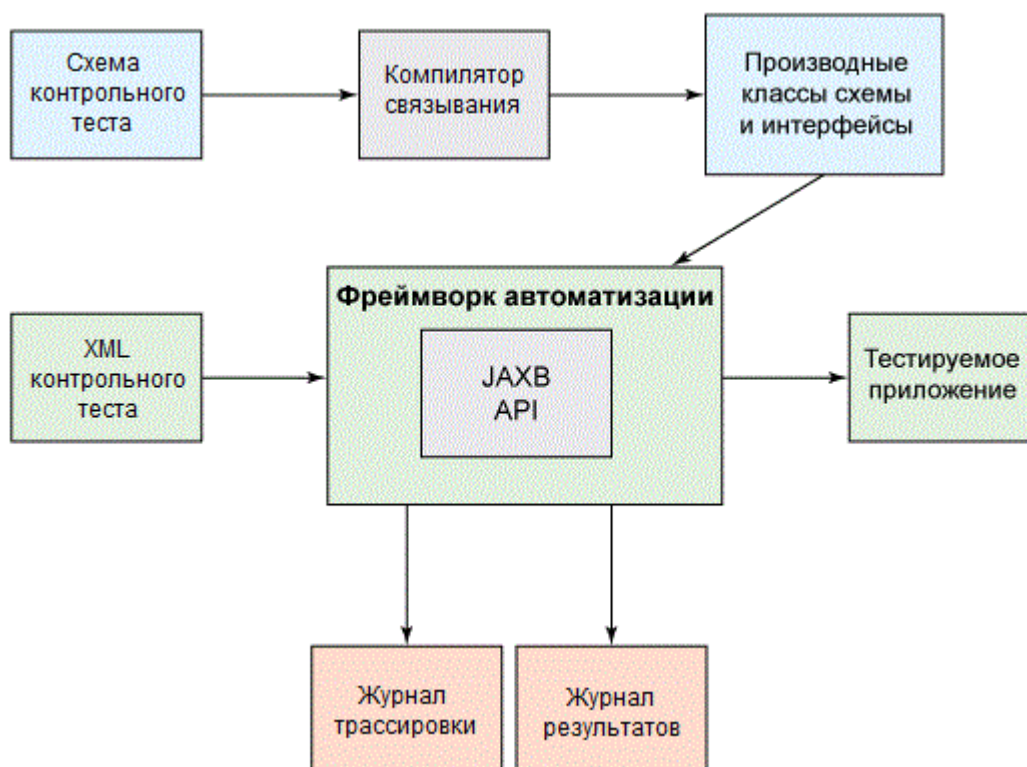
2.1 Зерттелетін жүйе облысының анализі

Мобильдік тестілеу қосымшасы – тест тапсыратын қолданушыларға кез келген уақытта, ғаламтор желісін пайдаланып, өзінің мобильдік құрылғысы арқылы тестілеуден өтуге мүмкіндік береді. Мобильдік қосымшаны қолданушылар тестіге дайын болмаған жағдайда, дайындық кезеңінен өтуіне болады.

Қосымшаны жасау негізінде тестілеу жүйесі жатыр. Тест құрастырудың бірнеше этаптары болады:

- бірінші этапта тестілеу мақсаты мен тапсырмалары анықталады. Қате жауап мысалдары талданады;
- екінші этапта тестік пішімдегі тапсырмалар құрастырылады;
- үшінші этапта статистикалық әдістерді қолданып, жауап беру мүмкіндігін бағалайды.

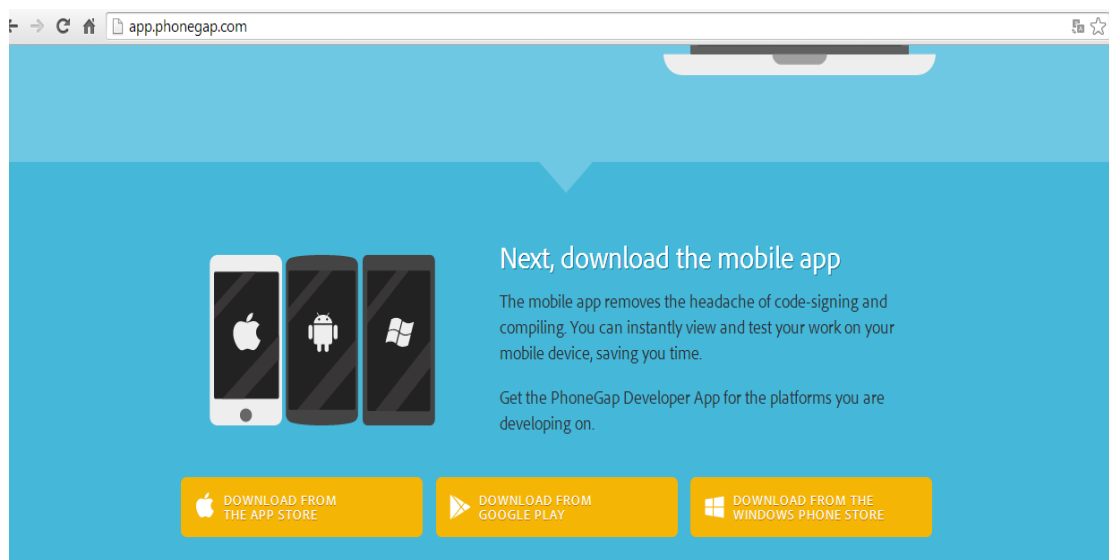
Сурет 2.1-де тестілеу жүйесін іске асыру архитектуралық схемасы көрсетілген. Бұл схеманың жұмыс істеу принципі HTTP сұраныстары көмегімен орындалады.



Сурет 2.1- Тестілеу принципінің архитектурасы

Мобильдік қосымша құрастыруда маңызды орынды PhoneGap платформасы алды. PhoneGap – бұл OpenSource платформасы, HTML, Javascript және CSS тілдерін ғана қолданып әр түрлі платформаларға арнайы қосымша жасауға мүмкіндік береді. Мысалы: IOS, Android, Blackberry, WebOS және Windows Mobile платформаларын атап өтсек болады. PhoneGap жаңалығы сол, ол ешқай дәлме-дәл платформаға құрастырушы бәләмән қажет етпей-ақ қолдануға болады. Сіз өзіңіз қосымшаны Javascript тілімен жазып, HTML мен CSS-ті үшін де қолдансаңыз болады. Сіз қосымшаны кәдімгі сайт немесе вэб-қызмет ретінде жазатын боласыз.

PhoneGap қозғалтқышы API браузерлерін кеңейтіп, келесі мүмкіндіктерді қосады: акселометрге рұқсат алу, камераны қолдануға рұқсат, компусты қолдануға рұқсат, контакт тізіміне және аудиожазбаларға, файлдық жүйе мен HTML5 әр түрлі localStorage, Web SQL және т.б. сақтау қорын қолдануға мүмкіндік туғызады. Тағы да еш ауыртпашылықсыз кез келген кросс-домендік мекен-жайларға бет бұруға болады. Келесі сурет 2.2-де PhoneGap фреймфоркының үй парақшасы көрсетілген. Бұл парақшадан PhoneGap бағдарламасын жүктеп, өзіңе керек платформаны таңдап қосымша құрастыруға мүмкіндік аласың.



Сурет 2.2 - PhoneGap үй парақшасы көрсетілген

Мобильдік қосымша құрастырып отырғанна кейін, біз jQuery Mobile вэб фреймворкын қолдандық. Осы қолданыстың айғағы ретінде сурет 2.3-те көре аласыз. Осыған орай jQuery Mobile ерекшеліктерін айта кетейік:

- jQuery негізге алынған түсінікті синтаксис;
- барлық негізгі платформалармен жарасымдығы – IOS, Android, Blackberry, Windows Mobile және т.б.;
- файл өлшемінің аздығы (12 кб функционалды қосылымы барлық мобильді құрылғыға)

- HTML5 белгілеуімен басқарулы парақша конфигурациясы мен жылдам даму нәтижесіне минималды талап;
- дамытылған әдіс ұсыну. Барлық мобильді құрылғының негізгі мазмұны мен функционалдығын күшейту, электронды кітап, Desktop Platform сынды қызметтер мобильдік құрылғыда оңай пайдаланылады;
- қолдануға мүмкін стандартты интернет-қосымшалардың, мысалы айтсақ WAI-ARIA, өте ыңғайлы көрініспен оқылады (мысалы VoiceOver IOS-та);
- Қарапайым API, Touch процесі көмегінің рационалы;
- Сенсорлық басқарылудың кеңейтілген жаңа модулдер;
- Қуатты визуализация көріністері қосымшаны мүмкіндігінше қарапайым құрылым ретінде интерфейспен ұсынады.

```

$(document).ready(function() {
    var predmet;
    var question;
    var correct;
    var incorrect1;
    var incorrect2;
    var incorrect3;
    var sourcequestion;
    var dataset;
    var numofQuestions;
    var num = 0;
    var id=1;

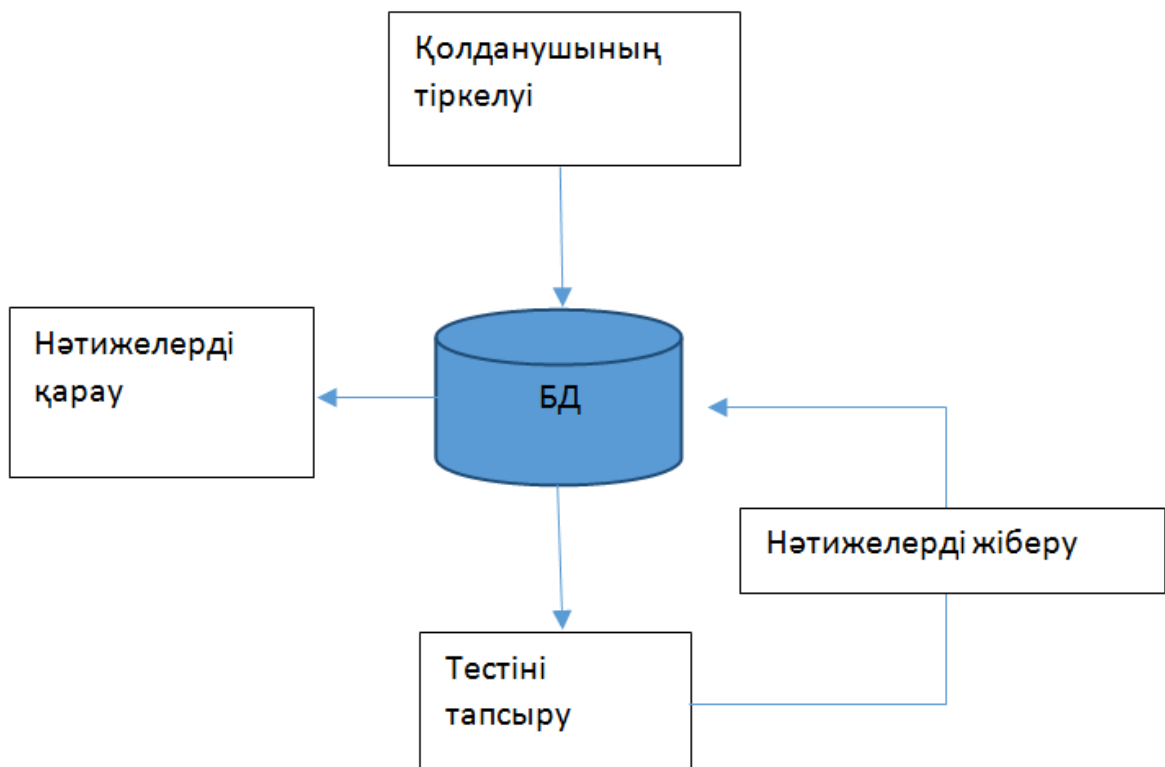
    var allQuestions = [];
    var questionArray = [];
    var correctArray = [];
    var incorrectArray1= [];
    var incorrectArray2 = [];
    var incorrectArray3 = [];

    $('#result').hide();
    var db = window.openDatabase("TestStudy", "1.0",
    "TestStudy", 5*1024*1024);
    db.transaction(populateDB, errorCallback);

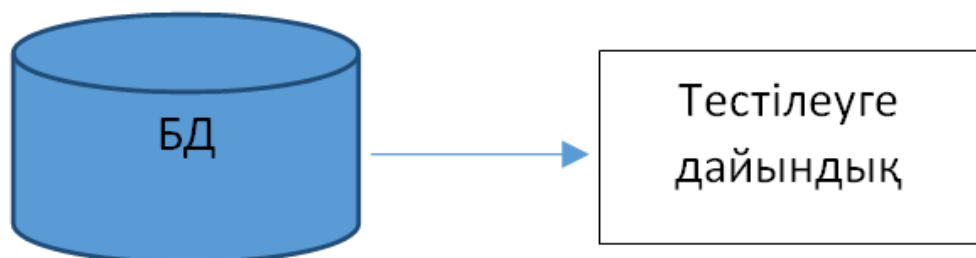
```

Сурет 2.3 – jQuery Mobile вэб-фреймворкында терілген кодтардың бір бөлігі

Дистанционды оқытуды дамыту қосымшасы 3 түрлі маңызды функцияны өзіне қосады: тест тапсыру, тестіге дайындық, нәтижелерді сараптау. Бұл қосымшаның өзектілігі болып кез келген тестілеу жүйесіне енгізуге болады. Себебі, бұл қосымша вэб-технологиялар көмегімен құрастылырылған және JavaScript арқылы кез келген алыстағы серверге ақпарат беруге және қабылдауға болады. Төменде тестілеу (Сурет 2.4) және дайындық кезеңі (Сурет 2.5) көрсетілген.



Сурет 2.4 – Тест тапсырудың жұмыс істеу принципі



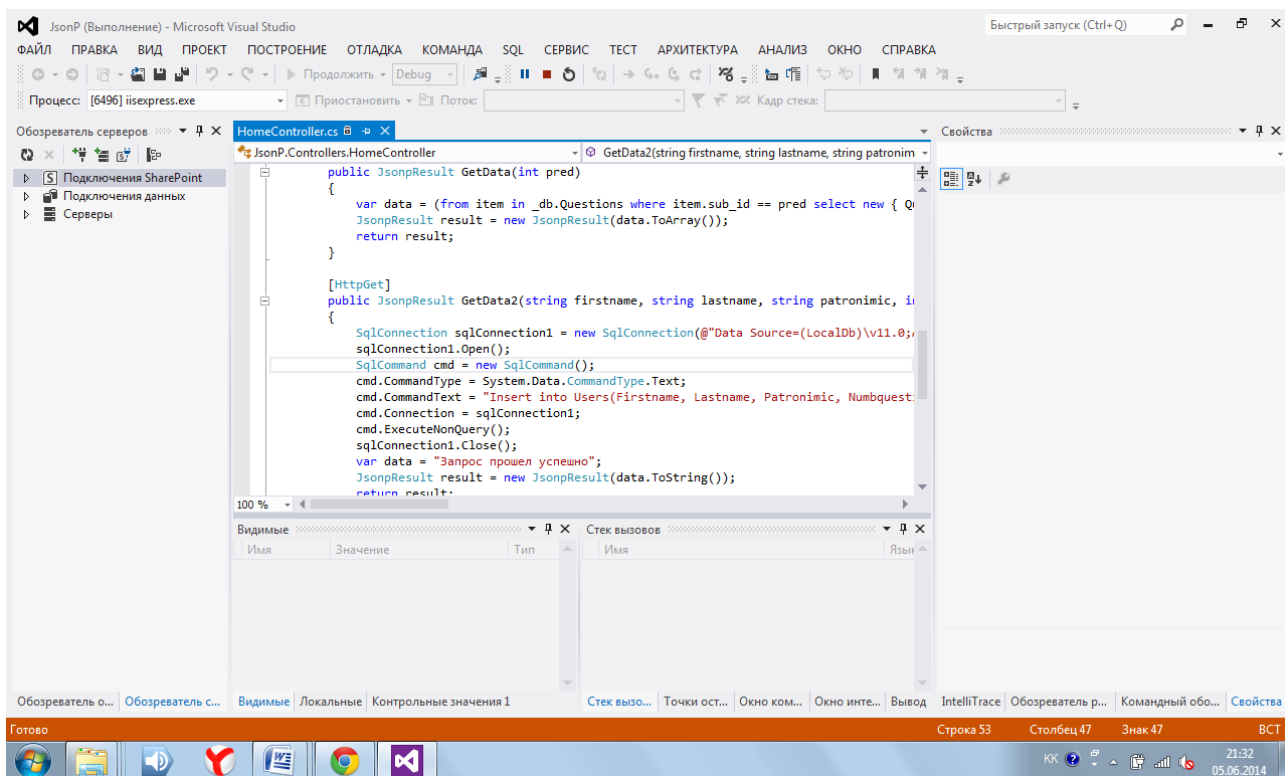
Сурет 2.5 - Тестілеуге дайындық кезеңі

Егерде қосымшаны кеңейтілген түрде қолдансақ, онда біз тағыда бағдарлама панелін толықтырамыз. Панелерге мыналар кіреді:

- бағдарламаны ретке келтіру панелі;
- бағдарлама ағынын бақылау панелі;
- консоль режимі;
- ізденіс нәтижесі «элементің қолдану»;
- панел логі (лог бағдарламасы көріністе);

- панель жүйесін бақылау версиясы;
- панель элементтері.

Төменгі сурет 2.3-те аталған панел жабдықтары көрсетілген.



Сурет 2.3 - Кеңейтілген режимдегі студиядағы жұмыс

2.2 Деректер моделінің анықтамасы

Құрастырылып отырған қосымшамыз ең алдымен ыңғайлы интерфейстан және жылдам орындалатын операцияларды қолдауы қажет. Қолданушыға оңай қолданылып, керек кезінде тестіні бастап кетуге мүмкіндік тудыруы керек.

IOS, Android, Windows Phone платформалары SQLite деректер қорын көруге болатын және SQLiteOpenHelper классы деректер қорын басқаратын қызметке рұқсат береді. Ол абстрактылық класс, оларға біз төменгі класс OnCreate () және onUpgrade () әдісін қолдануымыз керек.

SQLite басымдылықтары:

- SQLite қарапайым дискілерде оқылады және жазылады;
- SQLite аққысыз және деректер қорының ядросы ашық кодты SQL;
- деректер қорының піші кроссплатформалы;
- толығымен тәуелсіз лицензия;
- СУБД серверін баптаудың қажеті жоқ;
- SQLite стандартты деректер қорының қызметтерін, синтаксисін қоса отырып, мәміле және баптаулы анықтауды толығымен қуаттайды.

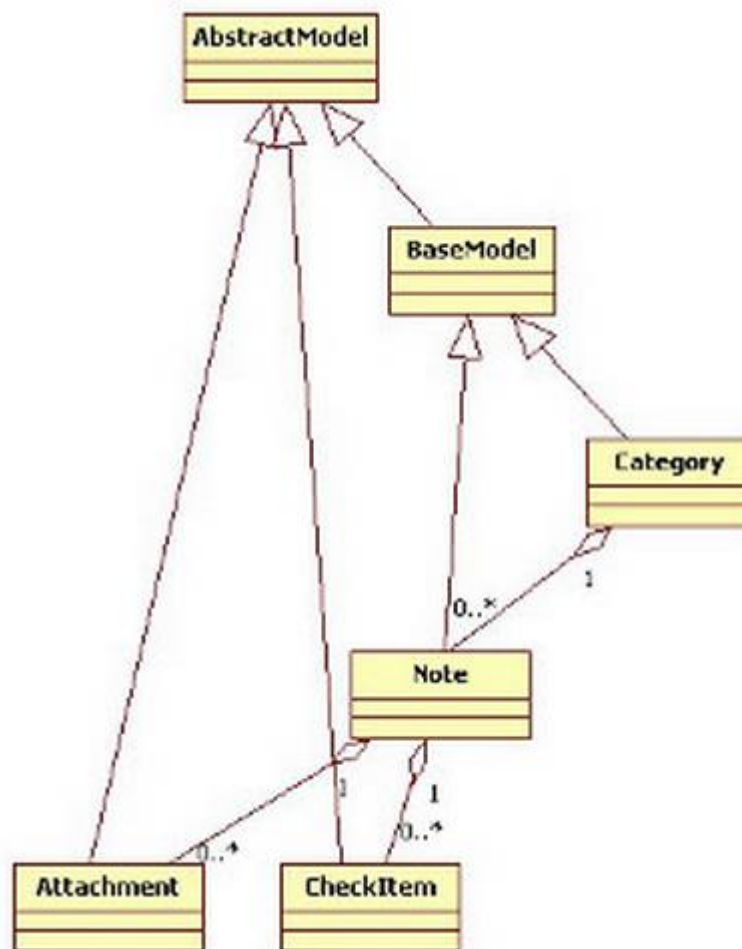
Деректер қорының структурасы:

PHP5+Apache+Linux (id записи:1)
SQLite - классная вещь (id записи:2)
sqlite.org (id записи:3).

Біздің дерек қорымызда үш бірдей кестеміз болады, келесі бағандарымыз еске алманы сақтап отырады:

- title -- text (note title);
- body -- text (note body text);
- _id -- integer, autoincrement (the id of the note).

Осы класс-схемасы сурет 2.6-да кескінделген.



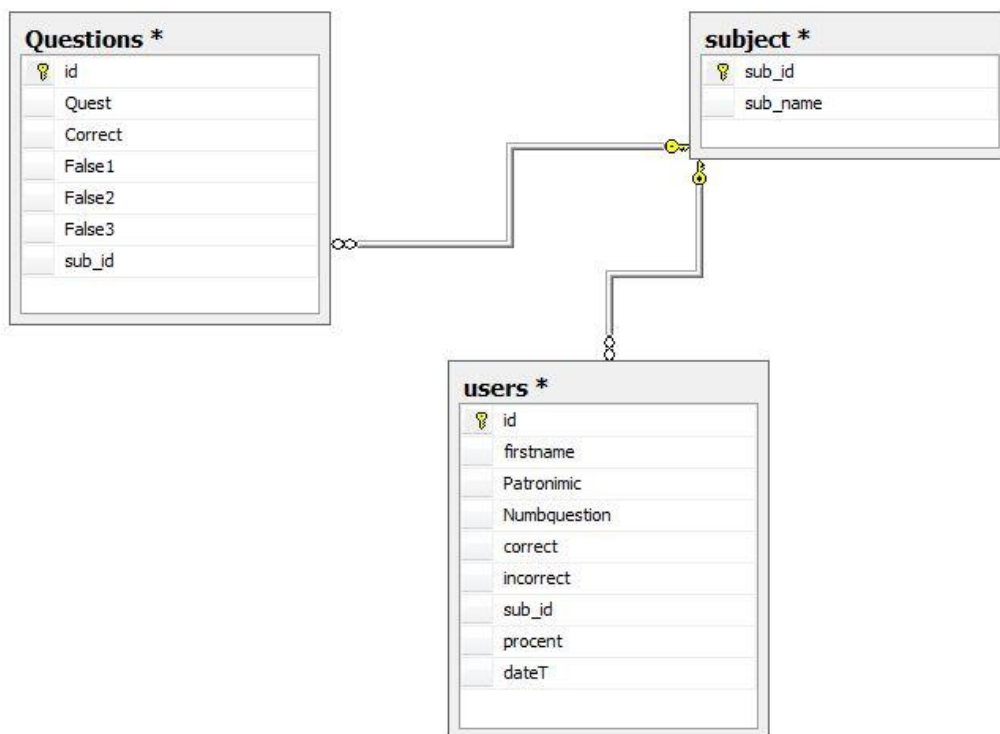
Сурет 2.6 - Дерекқордың жұмыс істеу тәртібі

2.3 Концептуалды проектилеу

Концептуалды проектилеу - зертелген деректер қорының терминында және концептуалды деректер қорының моделінде бейнеленген.

Концептуалды модель белгілі бір нақты модельге жасалынбайды. Бұл барлық мәліметтер облысын қамтитын, жоғары абстракциялық беделде жасалынатын ақпаратты модель.

Сурет 2.7-де ER – диаграммасы кескінделген. Бұл сурет бойынша байқайтынымыз қосымша сұрақтары үшін үш кесте алдық.(Пәндер - Subject, Сұрақтар - Questions, Қолданушылар - Users).



Сурет 2.7 – ER диаграммасы

3 Техникалық бөлім

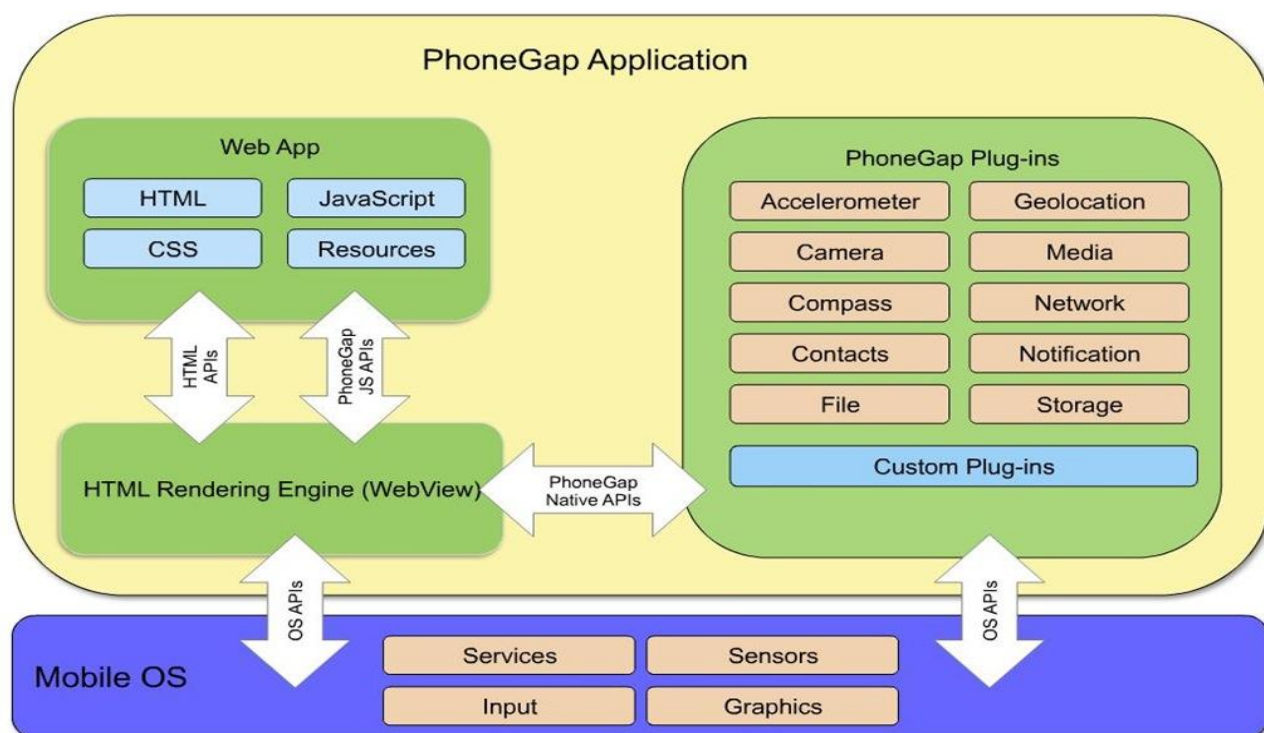
Бұл бөлімде қосымша құрастыру сипаттамасы келтірілген: бағдарлама архетиктурасының сипаттамасы және шаблонды проектилеудің қысқаша түсініктемесі мен құрастырылатын қосымша моделі, жасалатын интерфейс ті сипаттау және жасалатын процесті қысқаша сипаттауы бар.

3.1 Қосымша архитектурасы

Қосымша архитектурасы ретінде біршама кескіндерді көре аламыз. Кроссплатформалық мобильдік қосымша жасау кезінде PhoneGap фреймворкы көмегіне жүктелдік. Қосымша құру кезінде тек қана мобильдік қосымшаға ғана емес, оны толық ақпаратпен қамтамасыз ететін серверге де интерфейс тік жоба бердік. Ол үшін Microsoft Visual Studio 2012 ортасын алдық. Және бұл бағдарламалау ортасы арқылы сервердің интерфейс тік бөлігін стандартқа сәйкес жобалануын көздедік.

PhoneGap фреймворкының архитектурасын сурет 3.1-ден көре аласыздар. Мобильдік қосымшаны PhoneGap-пен жүзеге асыру үшін, алдымен HTML5, CSS3 және javascript тілдерімен қосымшаның бағдарламалық бөлігін жазып алдық.

PhoneGap Architecture

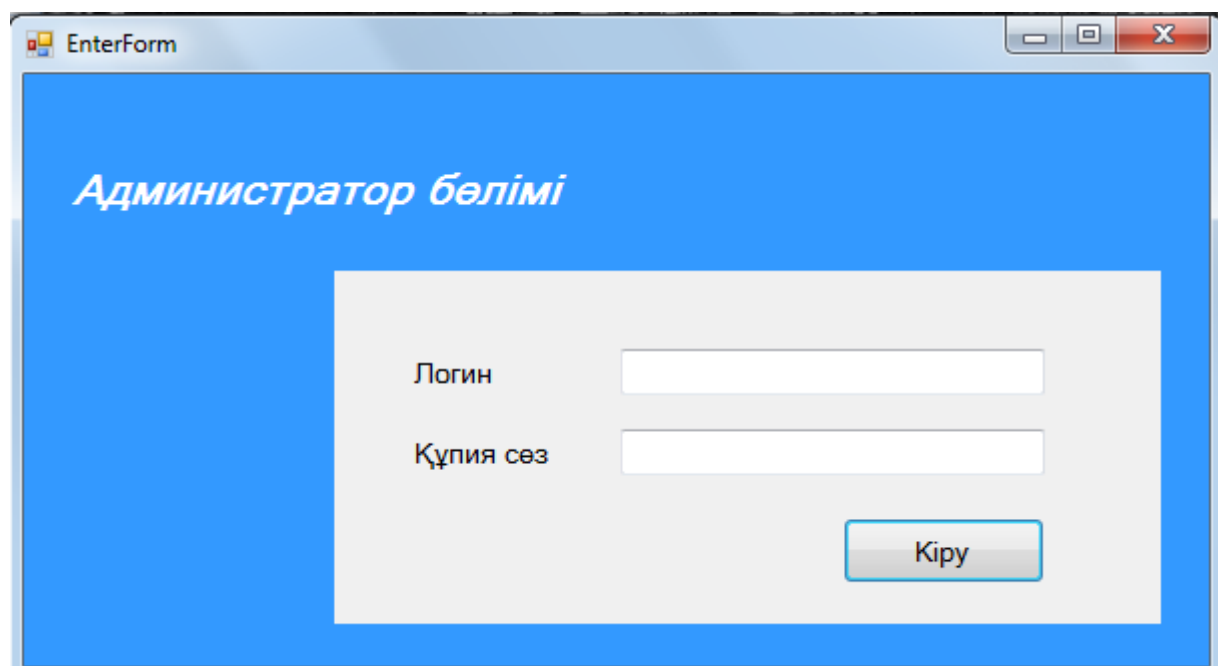


Сурет 3.1 – PhoneGap архитектурасы

3.2 Қолданбалы интерфейстің зерттемесі

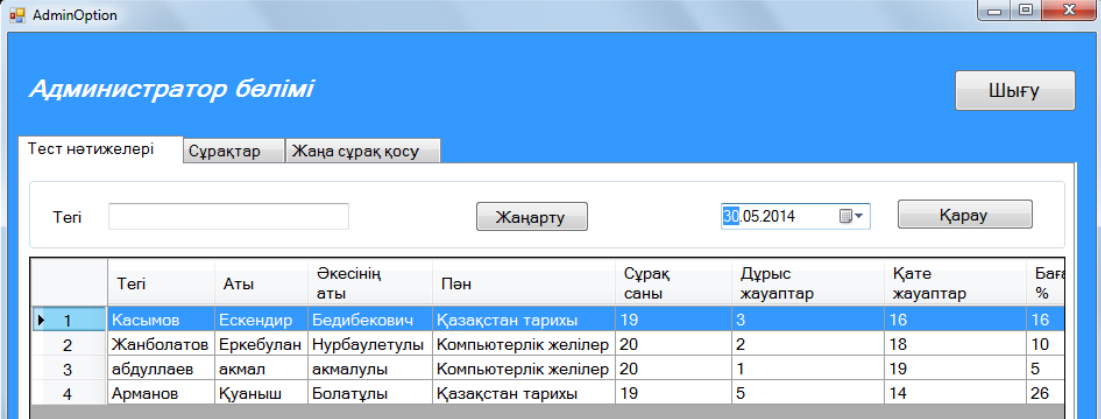
3.2.1 Сервер интерфейсі

Қолданушыға тест мәліметтерін беріп, олардың нәтижелерін тіркеп отыратын қосымшаның маңызды қосалқы бөлігі. Администрация басты бетінің көрінісі сурет 3.2-де бейнеленген.



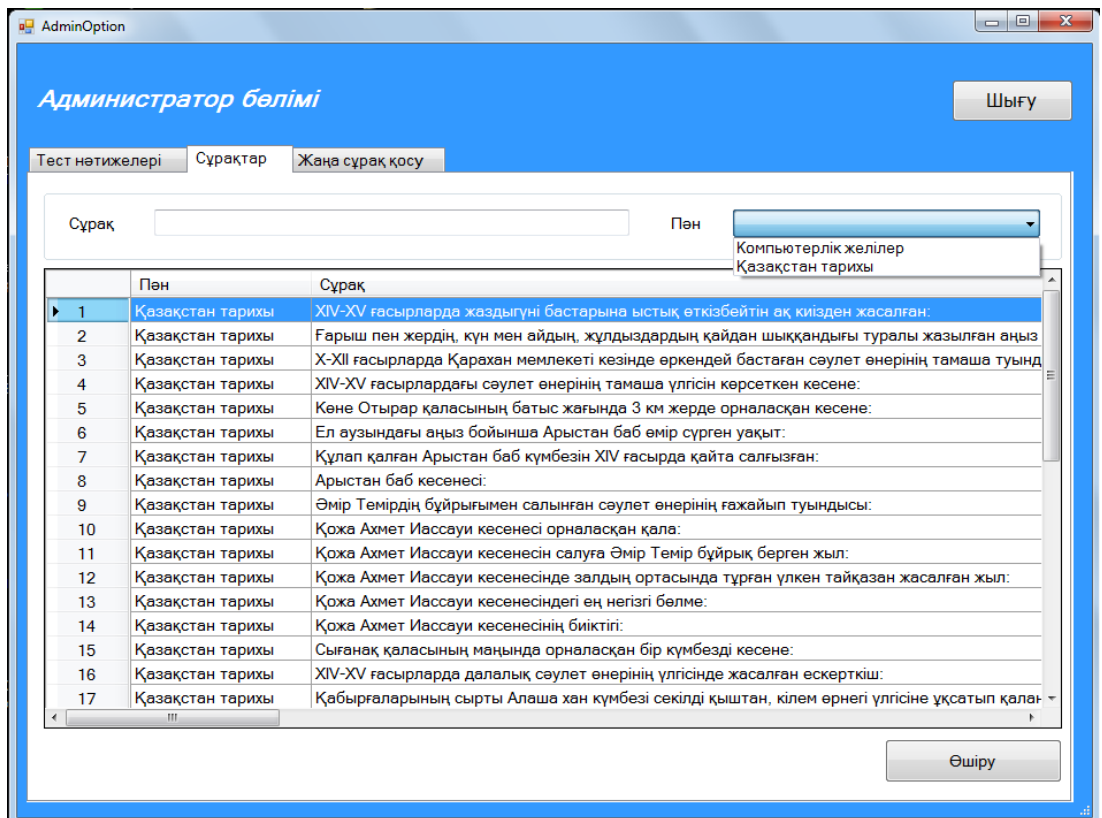
Сурет 3.2 – Администрация басты беті

Бұл қосымшада серверге рұқсат тек қана администраторда бар. Администратор өз логині мен құпия сөзін теріп көптеген мүмкіншіліктерге ие болады. Мысалы, жаңа сұрақтар қосу, сұрақтарды өзгерту, тест нәтижелерін бақылау және т.б. Администратордың жеке беті(Сурет 3.3, Сурет 3.4, Сурет 3,5) тест нәтижелері, сұрақтар және жаңа сұрақтарды енгізу бөлімінен тұрады.

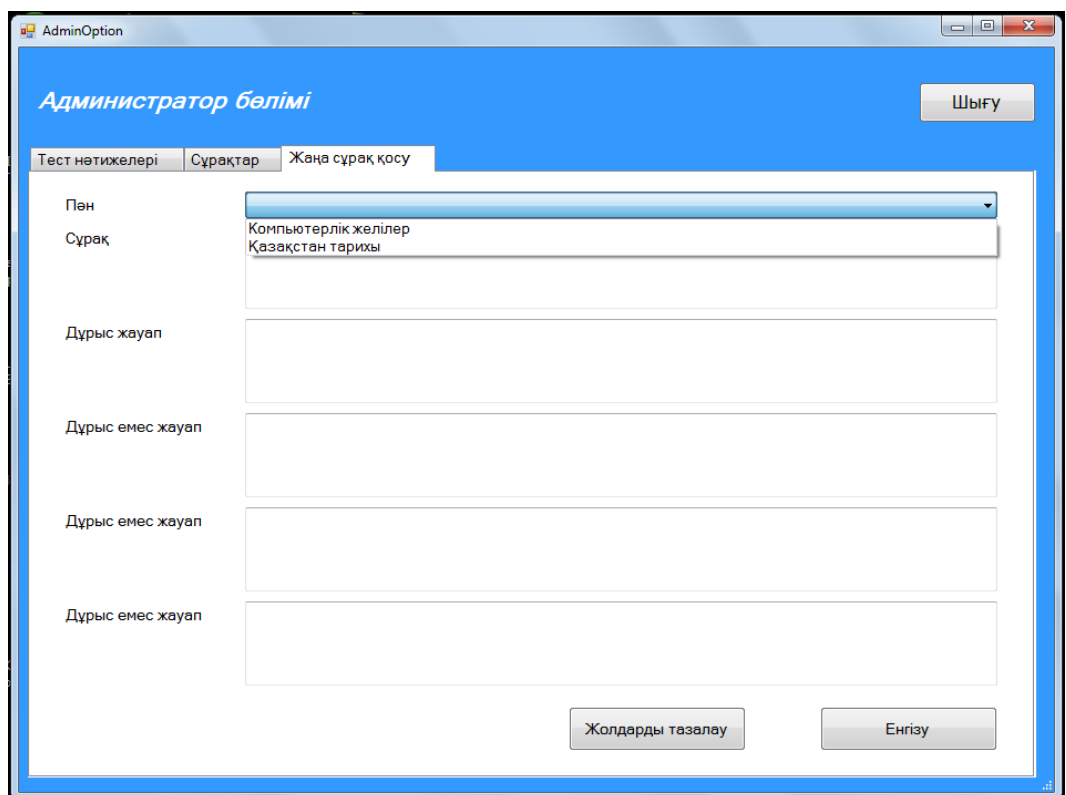


	Тегі	Аты	Әкесінің аты	Пән	Сұрақ саны	Дұрыс жауаптар	Қате жауаптар	Бағ %
1	Касымов	Ескендир	Бедибекович	Қазақстан тарихы	19	3	16	16
2	Жанболатов	Еркебулан	Нурбаулетулы	Компьютерлік желілер	20	2	18	10
3	абдуллаев	акмал	акмалулы	Компьютерлік желілер	20	1	19	5
4	Арманов	Қуаныш	Болатұлы	Қазақстан тарихы	19	5	14	26

Сурет 3.2 – Тест нәтижелерін көру (Администратор)



Сурет 3.3 – Сұрақтарды дерекқордан көру



Сурет 3.4 – Дерекқорға жаңадан сұрақ енгізу (Администратор)

3.2.2 Қолданушы беті

Қолданушы қосымшаны қосып, тест тапсыруға бірден кірісуге мүмкіндік алады. Ол үшін тест тапсырушы өзінің аты-жөнін, сынақ кітапшасының номерін және тест тапсыратын пәнді көрсетуі керек. Қолданушының тіркелу бөлімі сурет 3.4-те бейнеленген.

ТЕСТ

ДАЙЫНДЫҚ

НӘТИЖЕЛЕР

Тегі

Аты

Өкесінің аты

Есеп кітапшасының номері

ПӘН ТАНДАУ:

КОМПЬЮТЕРЛІК ЖЕЛЛЕР

ҚАЗАҚСТАН ТАРИХЫ

Тестілеуді бастау

Сурет 3.4 – Қолданушының тіркелу бөлімі

Қолданушы қажетті мәліметтерді енгізгеннен кейін тест тапсыруға мүмкіндік алады. Тест тапсыру уақытын 30 минут деп, сұрақты 4 нұсқамен бердім. Тест тапсыру мезетінде қолданушы сұрақтан сұраққа жауап бермей-ақ ауысуға мүмкіндік алады. Тест тапсырушы тест бастаған мезеті сурет 3.5-те бейнеленген, ал егер тест тапсырушы берілген сұрақтарға толығымен жауап бермей тестілеуді аяқтайтын болса, сурет 3.6-да көрсетілгендей оған жауап берілмеген сұрақтары нұсқалған ескертпе келеді.

ТЕСТ

ДАЙЫНДЫҚ

НӘТИЖЕЛЕР

СҰРАҚТАР: 2

АЯҚТАЛУҒА ҚАЛҒАН УАҚЫТ: 00:29:29

Ғарыш пен жердің, күн мен айдың, жұлдыздардың қайдан шыққандығы туралы жазылған аңыз

«ЖОЯМЕРГЕН»

«ЖЕРДЕН ШЫҚҚАН ЖЕЛІМ БАТЫР»

«ЕЛ АЙЫРЫЛҒАН»

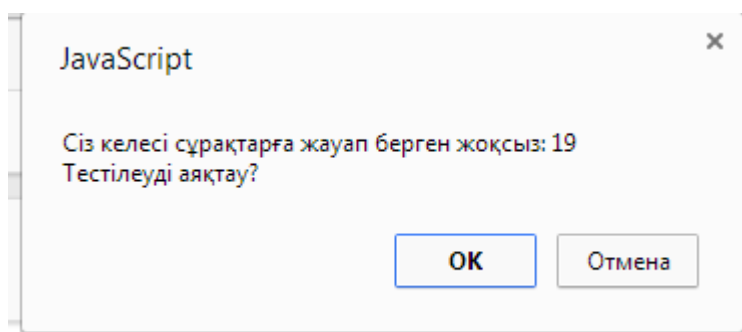
«АҚСАҚ ҚҰЛАН»

АРТҚА

КЕЛЕСІ

Тестілеуді аяқтау

Сурет 3.5 – Қолданушының тест тапсыру барысы көрсетілген



Сурет 3.6 – Тест сұрақтарына толығымен жауап бермегенде шығаты ескертпе

Тест нәтижесін тестілеуді аяқтауды көрсеткеннен кейін көруге болады. Сурет 3.7-де көрсетіліп отырғандай, тест нәтижесі дұрыс және қате жауаптар санымен қоса бағаны пайыздық мөлшермен есептеп береді.

ТЕСТ	
АРМАНОВ ҚУАНЫШ БОЛАТҰЛЫ	
СҰРАҚТАР САНЫ:	19
ДҰРЫС ЖАУАПТАР:	5
ҚАТЕ ЖАУАПТАР:	14
БАҒАСЫ:	26%

Сурет 3.7 – Тест нәтижесімен танысу

Егер қолданушы өз біліміне сенімді болмаса, ол біз қарастырып қойған дайындық кезеңінен өтсе болады. Дайындық кезінде тестілеуде келетін сұрақтар дұрыс жауабымен көрсетілген. Дайындық кезеңі қолданушыға кез келген уақытта қолжетімді, тест тапсыру кезінен басқа. Қолданушы дайындық кезеңін бастауы керекті пәнді таңдап (Сурет 3.8), дайындыққа кірісуіне болады.

ТЕСТ	ДАЙЫНДЫҚ	НӘТИЖЕЛЕР
<p>СҰРАҚТАР: 1</p> <p>Этапанды OSI моделінің қандай деңгейіне желілік адаптер жатады?</p> <p><input checked="" type="radio"/> АРНАЛЫҚҚА</p> <p><input type="radio"/> ФИЗИКАЛЫҚҚА</p> <p><input type="radio"/> КӨЛІКТІККЕ</p> <p><input type="radio"/> КӨРСЕТІМДІК ДЕҢГЕЙІНЕ</p> <p><input type="button" value="← АРТҚА"/> <input type="button" value="КЕЛЕСІ →"/></p>		

Сурет 3.8 – Қосымшаның дайындық бөлімі

Қосымшаның нәтижелер бөліміне келетін болсақ, мобильдік құрылғыдан осы қосымша арқылы тест тапсырған барлық адамның тізімі шартты түрде шығады. Сурет 3.9-да бейнеленген тест нәтижесі, әр тест тапсырған сайын жаңарып отырады.

ТЕСТ	ДАЙЫНДЫҚ	НӘТИЖЕЛЕР
<p>АТЫ-ЖӨНІ: ҚАСЫМОВ ЕСКЕНДИР БЕДИБЕКОВИЧ ПӘН АТЫ: ҚАЗАҚСТАН ТАРИХЫ БАҒАСЫ: NULL% СҰРАҚТАР САНЫ: 19 ДҰРЫС ЖАУАПТАР: 3 ҚАТЕ ЖАУАПТАР: NULL ТЕСТ ПЛЕУ УАҚЫТЫ: 2014-05-24 18:08:02</p>		
<p>АТЫ-ЖӨНІ: ЖАНБОЛАТОВ ЕРКЕБУЛАН НУРБАУЛЕТУЛЫ ПӘН АТЫ: КОМПЬЮТЕРЛІК ЖЕЛІЛЕР БАҒАСЫ: NULL% СҰРАҚТАР САНЫ: 20 ДҰРЫС ЖАУАПТАР: 2 ҚАТЕ ЖАУАПТАР: NULL ТЕСТ ПЛЕУ УАҚЫТЫ: 2014-05-25 21:00:51</p>		
<p>АТЫ-ЖӨНІ: АБДУЛЛАЕВ АҚМАЛ АҚМАЛУЛЫ ПӘН АТЫ: КОМПЬЮТЕРЛІК ЖЕЛІЛЕР БАҒАСЫ: 5% СҰРАҚТАР САНЫ: 20 ДҰРЫС ЖАУАПТАР: 1</p>		

Сурет 3.9 – Қосымшаның тест нәтижелері бөлімі

4 Тіршілік қауіпсіздігі

4.1 Еңбек шартынның анализы

Оператордың жұмыс орыны деп іскердің өндірістік процесті жүргізу және бақылау мақсаты үшін жабдықталған кеңістік бөлігін айтады. Сонымен қатар, жұмыс орыны болып, оның еңбек қызметін іске асыратын және басқару органы мен бағдарлама жабдықпен, ақпаратты көрсету құралдарымен жабдықталған жүйедегі орыны саналады. Оператордың жұмыс орынын құрастырғанда келесі шарттардың маңызы зор.

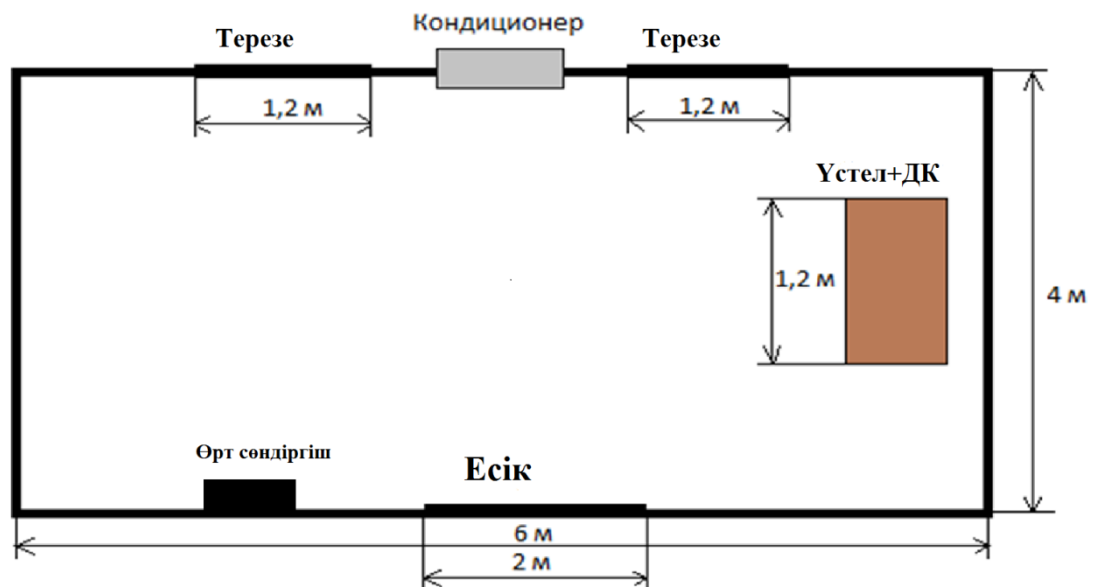
Бағдарлама жасау кезінде дербес компьютерді және электронды құрылғыларды қолданамыз. Қарастырылып отырған бөлмеде 1 оператор жұмыс атқарады, оның өзінің жеке жұмыс істеу орны бар.

Бағдарламаны қамтамасыз ету және жасау кезінде, басты рөлді жұмыс істеу орнына бөледі, себебі оператор аптасына 5 күн жұмыс істейді, 8 сағаттан, яғни ол компьютердің алдында аптасына 40 сағат жұмыс істейді, оған табиғи жарықты және жасанды жарықты түсу нормаларын ескеру қажет.

Жұмыс орынының мінездемесі:

Қарастырылып отырған жұмыс орындағы бөлмеміз, ғимаратта орналастырылған және ғимарат теміржол магистралінде немесе жүкті автомагистралінде, аэропорт және тағы басқа орындарда орналаспаған. Соған байланысты жұмыс істеу процессіне, сыртқы шудың көзі жоқ.

Бөлменің жоспары 4.1-суретте көрсетілген.



Сурет 4.1 – Бөлме жоспары

Бөлме келесі параметрлерге ие:

- бір қабатты ғимаратта орналасқан, бірінші қабатында;
- бөлменің өлшемдері (бөлме): ұзындығы 6 м, ені 4 м, биіктігі 3 м;
- жарық өткізетін материалдың түрі –шыны парағы, екіқабатты;
- тоқыманың түрі - екіқабатты стальден жасалған;
- күннен қорғанатын жабдықтар – жиналатын жазюли және шторлар;
- екі терезе 1,5*1,2;
- қабырғаның ішкі түрі – ақшыл;
- бөлмеміз жұмыс үстеу шарты бойынша, жеңіл жұмыс категориясына жатады (физикалық жеңіл, категориясы Ia, жұмысты отырып істейді және физикалық ауыр жұмыс атқармайды), осы стандарт (ГОСТ 12.2.032-78);

- жасанды жарық көзі – 2 шамдал екі люминесцентты шамы.

Жұмыс істеу кезіндегі қолданылатын құрылғының мінездемесі:

- Intel(R) core 2Duo CPU E8400 @ 3.0 GHz 3.0 GHz, 4 ГБ RAM / HDD 1000 Gb;
- монитор Samsung SyncMaster 932 BF, диагональ 19”;
- қоректену: айнымалы кернеу 220-250 В, жиілігі 50 Гц. Қуат 400 Вт;
- 2 шамдал, 4 люминесцентты лампасы;
- қоректену: айнымалы кернеу 220-250 В, жиілігі 50 Гц, шамдалдың қуатты 2x28 Вт.

Электротехникалық жабдық, ең бірінші өрт көзі болып табылады және қауып тудырады. Жұмыс уақыты біткеннен кейін барлық электротехникалық жабдықтарды өшіруіміз қажет. Құралдарымыз аз шуылдайтын және зиянды ара сапалы шуы болмайды.

Берілген бөлмеміз барлық электр қауіпсіздігі қанағаттандырады, яғни бөлме ылғал емес 60% аспайды, тақтай еденнен жасалған. Жерге тұйықтау және нөлдеу сымдарының қимасы, олардың қосылыстары мен жалғанымдарының бүтіндігі және беріктігі тексерілген. Аппаратты жерге тұйықтау контурымен қосатын жерге тұйықтау сымдарда үзілу және көзге көрінетін ақаулар болмауы тиісі ескерілген. Дәнекерлеу жұмысының сенімділігі балғамен соғып тексерілген.

4.2 Есептеу бөлімі

4.2.1 Кондиционерлеу жүйесін есептеу

Жұмыс бөлмесінде белсенді вентиляция жүйесі бар. Жұмыс бөлмесін талдау барысында жұмыс аумағының қалыпты микроклиматтық шарттарын міндетті түрде қарастыру қажет. Мұндай әмбебап жүйе ретінде автономдық кондиционерлер болып табылады. Кондиционерді орнату алдында арнайы есептеулер жасау керек. Сол есептеулердің қорытындысымен сипаттамасы және талаптарына сай кондиционер таңдалады. Бөлмені кондиционерлеу төменде келтірілген.

Микроклиматтың күйін қадағалау оператор залында қолайлы жағдайды орнатуға көмектеседі. Ал жұмыс орнының қолайлы жағдайларында адамдардың жұмыс істеу қабілеттері жақсарады, жұмысшылардың ауруға шалдығуы азаяды сонымен қатар, еңбек өндірісі артады [10]. Жұмыс орнындағы нормаланған климаттық параметрлер 4.1-кестеде келтірілген.

К е с т е 4.1 – Жұмыс орнындағы нормаланған климаттық параметрлер

Жыл мезгілі	Жұмыс категориясы	Ауа температурасы, °С	Салыстырмалы ауа ылғалдылығы, %	Ауа қозғалысының жылдамдығы, м/с
Суық	Жеңіл - 1а	22-24	40-60	0,1
	Жеңіл – 1б	21-23	40-60	0,1
Жылы	Жеңіл - 1а	23-25	40-60	0,1
	Жеңіл – 1б	22-24	40-60	0,2

Офис бөлмесін вентеляциялауға арнайы вентеляция каналдарын қолданады, бұл каналды ғимаратты салып жатқанда жасайды және жазғы уақытта терезе ашық болғанын ескереді. Жылдың жылы мезгілінде, офистағы температураның нормадан артуы 4.1-кестесінде көрсетілген, ал оптимальді микроклиматты ұстау үшін, бұл жағдайда кондиционерді қолданамыз. Әдеттегі офистағы микроклимат оператордың жұмыс істеу қабілетіне әсер етеді, егер офистағы микроклиматымыз жақсы болса, онда оператордың жұмыс істеу қабілеті арта түседі. Соған орай, микроклиматты жақсы дәрежеде ұстау үшін, біз кондиционер оратуымыз қажет.

Төменде жұмыс бөлмесіндегі, кондиционерлеу системасының есептеуі көрсетілген. Кондиционерлеу бөлмедегі лайықты климатты ұстап тұрады және берілген нормаларды қанағаттандырады.

Бөлмеге келетін ауаны $L_k, \frac{M^3}{c}$ формуласымен анықтаймыз:

$$L_{np} = \frac{Q_{ашж}}{c \cdot \rho_{np} \cdot (t_{вйт} - t_{np})} \quad (4.1)$$

мұндағы $Q_{ашж}$ артық шығындалатын жылу, $\frac{кДж}{c}$;

c - ауаның үлесті жылусиымдылығы, тұрақты түпкілікті қысым тең $c=1 \frac{кДж}{кг \cdot ^0C}$;

$\rho_{\text{пр}}$ - бөлмеге келетін ауаның тығыздығы, тең $1,2 \frac{\text{кг}}{\text{м}^3}$;

$t_{\text{выт}}$ - бөлмеден ауа температурасының шығуы және жұмыс істеу зонасынан, $^{\circ}\text{C}$;

$t_{\text{пр}}$ - келетін ауа температурасы, $^{\circ}\text{C}$.

Бөлмеден шығатын ауаның температурсы $t_{\text{шат}}$, $^{\circ}\text{C}$, осы формуламен анықталады:

$$t_{\text{шат}} = t_{\text{жз}} + \Delta t \cdot (h_{\text{ст}} - z) \quad (4.2)$$

мұндағы $t_{\text{жз}}$ - жұмыс істеу зонасындағы температура, осы берілген нормадан аспауы ($t_{\text{жз}} \leq t_{\text{доп}}$), $^{\circ}\text{C}$;

$h_{\text{ст}}$ - еденмен центрге дейінгі сорып алатын тесігі (кондиционера), м;

H - жұмыс зонасының биіктігі, м.

Осыған орай есептелу жылу мезгіліне арналған, сонда температура

$$t_{\text{жз}} = 22 \text{ } ^{\circ}\text{C}$$

Кондиционердің үшкі жағы, осындай биіктікте $h_{\text{ст}} = 2,5$ м:

$$t_{\text{выт}} = 22 + 1,2 \cdot (2,5 - 3) \text{ } ^{\circ}\text{C}$$

Келетін ауаның температурасы $t_{\text{пр}}$ бар жылудың артықшылығыа $5 - 7 \text{ } ^{\circ}\text{C}$ төменде жұмыс істеу орнының:

$$t_{\text{пр}} = 22 - 7 = 15 \text{ } ^{\circ}\text{C}$$

Артық бөлінетін және шығатын жылу $Q_{\text{ашж}}$ табамыз, бөлмедегі жылу балансына сүйене отырып осы:

$$Q_{\text{ашж}} = \sum Q - \sum Q_{\text{шы}} \quad (4.3)$$

мұндағы $\sum Q$ - бөлмеге келетін жылудың суммарлы саны;

$\sum Q_{\text{шы}}$ - бөлмеден шығатын суммарлы жылудың саны (есептеу арқылы жылудың жоғалуы, жылыңған жылудың бөлмеге келуі және т.б.).

Басты жылудың бөліну көзі болып, шамдал есептеледі және адамдар т.б. Тағыда, осыған орай есептеуде біз күн радиациясынан түсетін жылуды ескеруіміз қажет. Осы бөлмедегі дербес компьютерлерді және электр құрылғыларын ескермеуімізге болады. Осыдан біз бөлмеге бөлінетін жасанды жарықты, адамдардан, күн радиациясы арқылы келу жылу саннын есептейміз және терезеден келетін жылуды [10].

Жасанды жарықтан бөлінетін жылуымыз Q_2 , есептейміз, практикалық күйдегі шығатын энергияны, сонғы есепте төменде көрсетілген формуламен есептейміз

$$Q_2 = 1000 \cdot N \quad (4.4)$$

мұнда N – шамдалдан шығатын қуат, кВт

$$Q_2 = 1000 \cdot 0,28 \cdot 4 = 1120 \text{ кВт}$$

Адамдардан шығатын жылу Q_3 төмендегі формуламен анықтаймыз:

$$Q_3 = n \cdot q_{\text{ч}} \quad (4.5)$$

мұнда n – жұмыс атқаратын адамдар саны;

$q_{\text{ч}}$ - жылу саны, бір адамнан бөлінетін, 4.2- кестесінде көрсетілген.

К е с т е 4.2 – Жылу саны, бір адамнан бөлінетін, жұмыс істеу категориясына байланысты және осы жұмыс зонадағы температураға байланысты

Жұмыс категориясы	Жылу, Вт			
	Толық		Ашық	
	егер 100°C	егер 350°C	егер 100°C	егер 350°C
Жеңіл	180°C	145°C	150°C	5°C

$$Q_3 = 1 \cdot 145 = 145 \text{ Вт}$$

Жылу саны, күн радиациясынан бөлмеге келетін $Q_{\text{КУН,РАД}}$, осы формуламен анықталады:

$$Q_{\text{КУН,РАД}} = F_{\text{ост}} \cdot q_{\text{ост}} \cdot A_{\text{ост}} \quad (4.6)$$

Жабылған жағдай үшін:

$$Q_{\text{Ж,РАД}} = F_n \cdot q_n \cdot k_n \quad (4.7)$$

мұндағы $F_{\text{ост}}$ және F_n -жабылатын беттің ауданы, м^2 ;

$q_{\text{ост}}$ және q_n - жылу келетін 1 м^2 беткі шыныдан және беткі қабаттан, егер жылу өткізу коэффициент, тең болса $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^{\circ} \text{C}}$;

$A_{\text{ост}}$ – шыналу коэффициенті;

k_n – қабатқа жылу берілу коэффициенті, $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^{\circ} \text{C}}$.

Осы мән $q_{\text{ост}}$ тәуелділікке және географиялық қабаттың орнына терезенің және шамның 70–210 аралығындағы мәндерді қабылдайды, ал коэффициент $A_{\text{ост}}$ тәуекел шынының түріне және оның күнге қарсы қорғаныс құрлымына байланысты – осы шекте 0,25–1,25, жылу келудің орташа мәні, ол күннен келетін радиацияға, қандай қабаттардан өтетінін ескеріп отырылады б - 24.

$$F_{\text{ост}} = 1,5 \cdot 1,2 \cdot 2 = 3,6 \text{ м}^2.$$

Жұмыс істеу орнының терезесі солтүстікке қарай бағытталған, соған орай оның мәні $q_{\text{ост}}$ тең $140 \frac{\text{Вт}}{\text{м}^2 \cdot \text{C}}$. Осыдан $A_{\text{ост}} = 0,35$,

$$Q_{\text{ост.рад}} = 3,6 \cdot 140 \cdot 0,35 = 176,4 \text{ Вт}.$$

Жылу келудің орташа мәні географиялық ұзындығын ескеріп және қабатын қарастырсақ тең $Q_{\text{п.рад}} = 18 \text{ Вт}$.

Бөлмедегі жылудың жоғалуы $Q_{\text{ух}}$, кВт есектен және қабырғадан, терезені формула арқылы бағалайды:

$$Q_{\text{ух}} = \frac{\lambda \cdot S \cdot (t_{\text{выт}} - t_{\text{пр}})}{\delta} \quad (4.8)$$

мұндағы λ - қабырғаның жылу өткіздігішігі, $\frac{\text{Вт}}{\text{м} \cdot \text{C}}$;

S – аудан, м^2 ;

δ – қабырғаның қалыңдығы, м.

Жұмыс істеу орнының қабырғасы ауыр бетоннан жасалған М600, жылу өткізгішігі тең $12 \frac{\text{Вт}}{\text{м} \cdot \text{C}}$. Қабырға қалыңдығы $\delta = 0,5 \text{ м}$.

$$Q_{\text{ух}} = \frac{1,2 \cdot 24 \cdot (21,4 - 15)}{0,5} = 368,64 \text{ Вт}$$

Бөлмеге келетін жылудың суммарлы санын есептейік:

$$\sum Q = Q_2 + Q_3 + Q_{\text{ост.рад}} + Q_{\text{п.р.р}} \quad (4.9)$$

$$\sum Q = 1120145 + 176,4 + 18 = 1120,3 \text{ кВт}$$

Сонда есептеуіміз жаздың жылу мезгіліне келеді және осыдан бөлінетін жазғы жылуымыз:

$$O_{\text{изб}} = 1120,3 \text{ кВт}$$

Бөлмеге келетін ауаны есептейік:

$$L_{\text{пр}} = \frac{1120,3}{1 \cdot 1,2 \cdot (21,4 - 15)} = 145,9 \frac{\text{м}^3}{\text{ч}}$$

Ауа шығының қамсыздандыру үшін $L=145,9 \text{ м}^3/\text{ч}$, бізге тек1 кондиционер фирмасы Samsung сериясы НА 85 с ауаның максималды шығыны $195 \text{ м}^3/\text{ч}$, моделі R22

Модельдің қысқаша сипаттамасы:

- тұрпаты: индустриялық моноблокты кондиционер;
- номиналды суық өндірілімдігі Вт 2350;
- номиналды қажет ететін қуаты Вт 9400;
- ауаның шығыны (min-max) $\text{м}^3/\text{сағ}$ 100-195.

Бұл ақиқаттыққа сәйкеседі және жайлы микроклиматтың қамсыздандыруы үшін жетерлік болып табылады.

4.2.2 Табиғи жарықты есептеу

Табиғи жарық толығымен жұмыс істеу уақытысын да жарықты қанағаттандырмайды, тағыда ауа райдын өзгеруін, кешкі уақыттағы жұмысты ескеру қажет. Соған байланысты жасанды жарықты бөлмеде қолданамыз, ол люминесцентты шамдардан тұрады. Жарық көзінің нормативы төмендегі 4.3-кестеде көрсетілген.

Кесте 4.3 – Ортақ жарықтандыру жүйесіндегі ұсынылған жарық көздері

Түсті ажыратудағы талаптарға сәйкес көру жұмысының сипаттамасы	Жарықтандыру, лк	Күн көзінің түсу температурасының түс диапазоны $T_c, ^\circ K$	Қолданылатын шамдар түрі
Түсті ажырату талаптары көп емес объектілердің түстеріндегі ерекшелігі	300, 400	3500 - 5500	ЛД, ЛДЦ, ЛБ,
	150, 200	3000 - 4500	ЛБ, (ЛХБ), НЛВД+МТЛ, ДРЛ

Операторлардың жұмыс орындарының бөлмесін жасанды жарықтандыру үшін люминесцентті шамдарды қолданамыз, олардың жарық бергіштігі жоғары ($75 \text{ им}/\text{Вт}$ -қа дейін және одан жоғары), жарамдылық мерзімінің ұзақ ($10\ 000$ сағатқа дейін), шағылысу бетінің жарықтығы аз және жақсы жарық беруді қамтамасыз ететін сәулелендіргіш жарықтың табиғиға жақын спектральді құрамы [11].

Қолдануға ең ыңғайлы шамдар люминесцентті ЛБ (ақ түсті) және ЛТБ (жылы ақ түсті) қуаты 20,40 немесе 80 Вт шамдар.

Операторлық залды жарықтандыру үшін екі ЛТБ-40 люминесцентті шамдары бар УСП-35 типті шамшырақтар қарастырылған.

Бөлменің өлшемдері: ұзындығы $L=6 \text{ м}$, ені $B=4 \text{ м}$, биіктігі $H=3 \text{ м}$. Жұмыс бетінің еден деңгейінен биіктігі $h_{\text{рп}}$, $h_{\text{рп}}=0,725 \text{ м}$, терезелер

$h_{\text{HO}}, h_{\text{HO}} = 0,8\text{м}$ биіктігінен басталады, , терезелердің биіктігі h_o , $h_o = 1,5\text{м}$. Жұмыс орының бөлмесі IV сағат белдеуі – в г. Алматы (жарықтану белдеуінің климаты- IV 50^0 солтүстік ұзындығы және оңтүстік (Алматы, Караганда)).

Терезенің толық ауданы S_0 , м^2 , осы формуламен анықтаймыз:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_H \cdot \eta_0}{\tau_0 \cdot \tau_1} \cdot k_{\text{зд}} \cdot k_3 \quad (4.10)$$

$$S_0 = \frac{S_n \cdot e_H \cdot \eta_0}{100 \cdot \tau_0 \cdot \tau_1} \cdot k_{\text{зд}} \cdot k_3 \quad (4.11)$$

мұндағы S_n – бөлменің ауданы м^2 ; $S_n = 24 \text{м}^2$;

e_H – Ортақ жарықтандыру жарық көздері 4.3- кестесінен тандаймыз.

Көру жұмыстың биік дәлме-дәлділігі үшін $e_H = 1,2$ қабылдаймыз.

m_N - климаттың жарықтану коэффициенті, 4.3- кестесінен тандаймыз.

Жарықтану белдеуін ескере отырып, анықтауымыз : $m_N = 0,9$.

$$e_H^{\text{IV}} = e_H \cdot m \cdot c \quad (4.12)$$

$m = 0,7$;

$c = 0,75$ (қабырғаның сыртқы жағы);

$e_H = 1,2$ жұстағы үлкен дәлдіктер III разрядтағы көрініс жұмысы;

$e_H^{\text{IV}} = 1,2 \cdot 0,7 \cdot 0,75 = 0,63$;

Бөлменің типын ескере, осыдан табамыз коэффициенті $k_3 = 1,2$ (жұмыс істеу орнын қарастырылады):

τ_0 - жарық өткізудің толық коэффициенті тең $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$;

$\tau_1 = 0,5$ (бос орынды шынылы блок);

$\tau_2 = 0,6$ (екі қабатты ағаштан жасалған тоқыма);

$\tau_3 = 0,8$ (темірбетонды аркалар);

$\tau_4 = 1$ (жиналатын жалюзилер және шторлар);

$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24$;

η_0 -терезенің жарықтану мнездемесі.

Анықтаймыз η_0 :

$l = B - 1$

$l = 4 - 1 = 3\text{м}$

$\frac{L}{l} = \frac{L}{B-1} = \frac{6}{3} = 2$,

$$h_{\text{расч}} = h_{\text{HO}} + h_o - h_{\text{рп}}$$

$$h_{\text{расч}} = 0,8 + 1,5 - 0,72 = 1,57$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{2,3} = 1,7$$

$\eta_0 = 10$ кесте алынған.

Бөлмедегі орташа шағылу коэффициент $\rho_{\text{СР}} = 0,5$, тек бір жағындағы жарықты қарастырамыз.

$$\frac{l_{\text{рт}}}{B} = \frac{0,5}{4} = 0,125$$

мұндағы $r_1 = 1,05$;

$k_{\text{зд}}$ – іске асыру кезіндегі жарық көздерінің шаңдануы мен жарамсыз болуын ескеретін қор коэффициенті $k_{\text{зд}} = 1$.

Терезелердің толық ауданын есептейміз:

$$S_0 = \frac{24 \cdot 1,35 \cdot 10 \cdot 1 \cdot 0,63}{100 \cdot 0,24 \cdot 1,05} = 8,16 \text{ м}^2$$

Жарық өтуінің ауданы тең $S_{\text{ст}} = 8,16 \text{ м}^2$.

Осыған орай жарық өту ауданы ($1,5 \cdot 1,2 \cdot 2 = 3,6 < 8,16$) жұмыс орындағы қажетті еңбек жағдайларын қамтамасыз етпейді. Осы мақсатта еңбектің ең қолайлы с параметрі $6 \times 4 \times 3$ қамтасамыз ететін, $e_N = 0,84$ көру жұмысының III мінездемесі, сонда жұмыс орнында табиғи жарықты және жасанды жарықты қолданады.

4.3 Тіршілік қауіпсіздігі бөлімі бойынша қорытынды

Бұл бөлімде офистік компаниядағы жұмыс шарты және микроклимат параметрін есептеуді орындау, жағдайын тексеру мақсатында анализ жасалды. Еңбек шарты деңгейі дұрыс және есептеу бойынша алынған есептеулер, тіршілік қауіпсіздік бойынша стандарттарды қанағаттандырады.

Нәтиже бойынша, екі терезе $1,5 \times 1,2 \text{ м}$ өлшемді жұмыс орнындағы табиғи жарығы нормаларды қанағатандырмайды. Жұмыс орнында жарықты қамтамасыз ету үшін жасадны жарықты қолданамыз.

Есептеу нәтижесі бойынша, бөлмеге қажетті ауа мөлшерін қамтамасыз ету үшін $L = 145,9 \text{ м}^3/\text{сағ}$ байланысты біз тек 1 кондиционер қойамыз, ол Samsung сериясы HA 85 максималды ауаның шығыны $195 \text{ м}^3/\text{сағ}$, модель R22. Модельдің қысқаша характеристикасы:

- тұрпаты: индустриялық моноблокты кондиционер;
- номиналды суық өндірілімдігі Вт 2350;
- номиналды қажет ететін қуаты Вт 9400;
- ауаның шығыны (min-max) $\text{м}^3/\text{сағ}$ 100-195.

5 Техникалық экономикалық негізделуі

5.1 Жоба сипаттамасы

«Дистанционды оқытудағы тестілеу жүйесінің дамуы. Кроссплатформалық мобильдік қосымша» бұл мобильді құрылғыларға арналған дистанционды тестілеу жүйесіндегі ең алғашқы қосымша. Бұл арнайы бағдарлама Visual Studio 2012, ASP.NET MVC4, Phonegap бағдарламасымен Microsoft SQL деректер қорын байланыстыру арқылы жасалады.

К е с т е 5.1 – Қызметті SWOT-талдау

Артықшылықтары	Кемшіліктері
Бұл бағдарламаны жоғарғы оқу орындарына енгізілмеген.	Тек қана дистанционды оқу жүйесі бар оқу орындары ғана пайдалана алады.
Мүмкіншілігі	Қауіпі
Бағдарлама табыс көзі	Бағдарлама рұқсатсыз көшірілуі мүмкін

Маркетинг-микс қызметтері.

Маркетинг-микс – Маркетингтік құралдардың жиынтығы, бұл арқылы компания тауарлар немесе қызметтер сұранысына ықпал етуге тырысады. Маркетинг-микс – бұл белгілі бір тауарға анықталған баға бойынша белгілі бір орында қозғалыс арқылы сұранысты қалыптастыратын жиындардың (стратегия) жүйесі.

Ереже бойынша, стандартты маркетинг-микс жиындарды төрт құраушы тактикалық іс бойынша сипаттайды: Маркетинг жиынтығы (marketing-mix) – бұл фирма біртұтас нарықтың оң жауабын алу үшін қолданатын басқаруға мүмкіндік беретін маркетингтік факторлардың жиыны. Маркетинг-микстің міндеті – біртұтас нарықта потенциалды тұтынушылардың қажеттерін өтеумен ғана шектелмейтін, сонымен қоса ұйымның әсерін жоғарлатуға тырысатын жиынтықтарды (mix) құрастыру.

Өнім (Product) — бұл фирманың тұтас нарыққа ұсынатын «бұйымдар мен тауарлар» жиынтығы.

Баға (Price) — тұтынушылардың тауарды алу үшін төлеуі қажет ақша суммасы.

Дистрибуция (Place) — тауар барлық бірыңғай тұтынушыларға қолжетімді болу үшін жүзеге асырылатын мүмкіндігі көп іс.

Қозғалу, даму (Promotion) — фирманың өзінің тауарлары туралы ақпарат таратуы мен тұтынушыларды осы тауарларды сатып алуға үгіттейтін мүмкіндігі зор іс-шаралар.

Жобама байланысты ЖОО аралап, мобильдік қосымшалардың орташа құнын есептедім. Мысалы СДУ ЖОО-дағы мобильдік қосымшаның құны 200000 теңгені құрады. Ал, менің жобама сәйкес дәлме-дәл аналог табылмады.

К е с т е 5.2– Маркетинг стратегиясының кестесі

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Product	
Бағдарламаның қызметі	Жалпы тапсырманы шешу келесі бөлімдерден тұрады: 1) Оқу орнынан тыс жерде дистанционды түрде тестке дайындалы, тапсыруға мүмкіндік алады. 2) Бұл бағдарламаны сатып алған жоғарғы оқу орындары жылдық шығынды және тесттер құрастыруға кеткен уақытты үнемдейді. Егер бағдарламаға уақыт өте келе сұраныс артатын болса, бұл қызмет түрі міндетті түрде ары қарай дамытылады.
Price	
Promotion	
	Басында уақытша жарнамасыз жұмыс істеу, яғни тікелей сату.
Place	
АТУ, КазЭУ, КазАТК	Алматы қаласындағы жоғарғы оқу орындарының админстраторларын хабарландыру, келіссөз жүргізу.

5.2 Қосымшаның өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері мына формула бойынша есептеледі:

$$Q=q*c \quad (5.1)$$

Мұндағы q-бағдарламалық өнімдегі (бастапқы команда) операторлар саны, менің жұмысымда операторлар саны 3000-ға тең, [12] таңдап алуға болады.; c-бағдарлама күрделілігінің коэффициенті, ол менде 1.26-ға тең, себебі менің жобамның қосымшалық тілі жоғарғы тілдер қатары және күрделілік тобы жоғары. Осы әдіспен базалық көрсеткішті табамыз:

$$Q = 3000 * 1.26 = 3780$$

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек. Уақыт адам-сағатпен есептеледі, ал T_d нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты Q командасының шартты санына байланысты есептік жолмен анықталады.

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтаймыз:

1) $T_{ПО}$ (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3-тен 5 күнге дейін, 8 сағаттан):

$$T_{ПО} = 24 \text{ адам / сағ}$$

2) T_O (мақсат сипаттамасы уақыты) келесі формуламен анықталады:

$$T_O = Q \times B / (50 \times K) \quad (5.2)$$

$$T_O = 3780 * 1.2 / 50 * 0.8 = 113.4 \text{ адам/сағ}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті, B коэффициенті мақсат күрделігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін [15];

K – бағдарлама жасаушы біліктілігін ескеретін коэффициент, менің берілген салада жұмыс істеу өтілім 2 жылға дейінгі уақытты құрағандықтан, K коэффициенті мәні – 0,8-ге тең болады.

3) T_A (алгоритм құруға кеткен уақыт) мына формуламен есептейміз:

$$T_A = Q / (50 \times K) \quad (5.3)$$

$$T_A = 3780 / (50 * 0,8) = 94.5 \text{ адам/сағ}$$

4) T_{BC} (блок – сұлба құруға кеткен уақыт) T_A сияқты 3 формуламен есептелінеді.

5) T_H (бағдарлама теру уақыты) келесі формуламен анықталады:

$$T_H = Q \times 1,5 / (50 \times K) \quad (5.4)$$

$$T_H = 3780 * 1,5 / (50 * 0,8) = 141.75 \text{ адам/сағ.}$$

T_{II} (бағдарлама теру уақыты) келесі формуламен анықталады:

$$T_{II} = Q / 50 \quad (5.5)$$

$$T_{II} = 3780 / 50 = 75.6 \text{ адам/сағ}$$

6) T_{OT} (бағдарламаны реттеу және тестілеу уақыты) келесі формуламен анықталады:

$$T_{OT} = Q \times 4,2 / 50 \times K \quad (5.6)$$

$$T_{OT} = 1800 * 4,2 / 50 * 0,8 = 189 \text{ адам/сағ}$$

7) T_D (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3-тен 5 күнге дейін, күніне 8 сағат)

$$T_D = 24 \text{ адам / сағ}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде (5.7) формуламен есептеледі:

$$T = T_{ПО} + T_{ТО} + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D \quad (5.7)$$

$$T = 24 + 113.4 + 94.5 + 94.5 + 141.75 + 75.6 + 396.9 + 24 = 964.65 \text{ адам/сағ}$$

5.3 Қосымшаны өндеуге кеткен шығынның есептелуі

Еңбек құны екі құрамнан құралады: негізгі еңбек құны және бағдарлама еңбек құны.

Негізгі еңбек құны (5.8):

$$З_{осн} = t_{\Sigma} * TC / (t_{cp} * 8), \text{ теңге} \quad (5.8)$$

мұндағы t_{Σ} – (5.7) формуласы бойынша есептелетін жалпы еңбек шығыны;

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

Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы 01.01.2014ж бастап ҚР-да 19966 теңгені құрады.

TC – тарифтік қойылым = $19966 / 8 = 2495,5$ теңгеге тең болды.

t_{Σ} – (5.7) формуласы бойынша 964.65–ге тең.

Осылайша негізгі еңбек ақы:

$$З_{осн} = 964.65 * 2495,75 / 21 = 114644 \text{ теңге}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%-ын құрайды. (5.9) формуласы бойынша анықталады:

$$З_{доп} = 0,2 * З_{осн} = 22928.8 \text{ теңге} \quad (5.9)$$

Жалпы еңбек ақысы (еңбекті төлеу фонды) негізгі және бағдарлама еңбек ақының қосындысы ретінде анықталады :

$$ФОТ = З_{доп} + З_{осн} \quad (10)$$

$$ФОТ = 22928.8 + 114644 = 137572.8 \text{ теңге}$$

5.4 Қосымша шығындар статьясы

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

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, бағдарлама шығындар статьясындағы есептеулерде қолданылады. ЭЕМ - на бағдарлама жазу кезінде құрылғы ретінде дербес компьютер қолданылады. Құны $C_{обор} = 189000$ теңге (дербес компьютердің қазіргі уақыттағы нарықтағы теңгемен берілген құны алынды) [15].

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ($C_{МжК}$), сонымен қатар техникалық қызмет көрсету шығыны ($C_{ТО}$), жабдықтың құнынан 1.5% және 2.5% құрайды және мына формулалар мен есептеледі:

$$C_{МжК} = 0,015 \times C_{обор} \quad (5.11)$$

$$C_{МжК} = 0.015 * 189000 = 2835 \text{ теңге}$$

Амортизациялық аударым өнімнің физикалық және моральдық жарамсыздығына байланысты сол өнімді толық қалпына келтіру үшін жиналатын ақша қоры болып есептеледі. Амортизациялық аударым амортизация нормасы бойынша орнатылады, (2.14) формуласы бойынша есептеледі:

$$A = \frac{C_{обор} \times H_A \times N}{100 \times 12 \times t} \quad (5.12)$$

мұндағы $C_{обор}$ – компьютер құны; H_A – амортизация нормасы, N – жұмыс орындалуына кеткен күннің саны.

Дербес компьютерді қолдануға кеткен жалпы уақыт:

$$N = T/24 \quad (5.13)$$

$$N = 964.25/24 = 40.19 \text{ күн}$$

бұл жерде T – еңбек шығынына тең, оның мәні – 498 адам/сағ тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі:

$$t = T_A + T_{BC} + T_H + T_{II} + T_{OT} \quad (5.14)$$

$$t = 94.5 + 94.5 + 141.75 + 75.6 + 396.9 = 803.25 \text{ адам/ сағ}$$

Амортизация нормасы (5.13) формуласы бойынша анықталады:

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% \quad (5.15)$$

мұндағы $C_{ликв}$ – таратылым құны, жабдықтың құнынан 5% құрайды:

$$C_{ликв} = 0.05 \cdot C_{обор} \quad (5.16)$$

$$C_{ликв} = 0.05 \cdot 189000 = 9450 \text{ теңге}$$

$T_{норм}$ – нормативті қызмет ету өтілі (дербес компьютер үшін ҚР - да $T_{норм}=4$ жыл). Осыдан:

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% \quad (5.17)$$

$$H_A = \frac{189000 - 9450}{4 \cdot 189000} \cdot 100\% = 23,75\%$$

$$A = \frac{C_{обор} \times H_A \times N}{100 \times 12 \times t} \quad (5.18)$$

$$A = 189000 \cdot 23,75 \cdot 40,19 / 100 \cdot 12 \cdot 803,25 = 18,71 \text{ теңге}$$

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

$$C_{ЭЭ} = M \cdot k_3 \cdot F_{эф} \cdot C_{кВт-ч} \quad (5.19)$$

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

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны, оның мәні Алматы қаласында 14,36 теңгеге тең;

$F_{эф}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі (2.17):

$$F_{эф} = D_{ном} \cdot d \cdot (1 - f / 100) \quad (5.20)$$

$$F_{эф} = 258 \cdot 8 \cdot (1 - 2,6 / 100) = 2010$$

мұндағы $D_{ном} = 258$ – бір жылдағы жұмыс күнінің номиналды саны;

$d = 8$ жұмыс күнінің ұзақтығы [сағ];

$f = 2.6\%$ – ЭВМ –ды жөндеуге кеткен жоспарланған уақыт.

Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны: $C_{ЭЭ} = 0,45 * 0,8 * 2010 * 14,36 = 10611$ теңге.

Дегенмен, алынған амортизациялық аударым мен электроэнергияға шығын – жылдық шығынның мәні, оны жалпы жылдық эксплуатациондық шығынмен анықталатын уақыт коэффициентімен коррекциялау (дұрыстау) керек(17):

$$\mathcal{E}_3 = t_{\Sigma'} * C_{ЭЭ} / F_{эф}, \text{ теңге} \quad (5.20)$$

мұндағы $C_{ЭЭ} = C_{ЭЭ} + C_{ТО} + A_{жыл}$ – ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$C_{ЭЭ} = C_{ЭЭ} + C_{ТО} + A_{жыл} \quad (5.21)$$

$$C_{ЭЭ} = 1061 + 4872,5 + 209,25 = 15693 \text{ теңге}$$

мұндағы $F_{эф}$ – жұмыс уақытының эффективті фонды, (5.20) формуласы бойынша анықталды, $t_{\Sigma'}$ – ЭЕМ-нің (5.21) формуласы бойынша анықталған есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған:

$$t_{\Sigma'} = t_{пр} + t_{отл} + t_{д} \quad (5.22)$$

$$t_{\Sigma} = 94.5 + 396.9 + 24 = 515.4 \text{ сағ}$$

Сәйкесінше, ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$\mathcal{E}_3 = 515.4 * 60003.3 / 2010 = 15385.9 \text{ теңге}$$

ал уақыт коэффициенті мына формула бойынша есептеледі (5.22):

$$w = \frac{\mathcal{E}_3}{C_{ЭЭ}} \quad (5.23)$$

$$w = 15385.9 / 60003.3 = 0.25$$

Осылайша, жалпы эксплуатационды шығынды уақыт коэффициентін ескеріп коррекциялаймыз(дұрыстаймыз):

$$\text{Электрэнергияға шығын: } C_{ЭЭ}^* = C_{ЭЭ} \cdot w;$$

$$C_{ЭЭ}^* = 10390.8 * 0.25 = 2597.7 \text{ теңге.}$$

$$\text{Амортизационды аударым: } A_{жыл}^* = A_{жыл} \cdot w;$$

$$A_{жыл}^* = 44887.5 * 0.25 = 11221.8 \text{ теңге.}$$

Сонымен қатар, (5.23) формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар. Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], (5.23) формуласы бойынша анықталады:

$$CC = (\text{ФОТ} - 0,1 * \text{ФОТ}) * 0,11; \quad (5.23)$$

$$CC = (137572.8 - 0,1 * 137572.8) * 0,11 = 13620 \text{ теңге.}$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін бағдарлама шығындар болып табылады, еңбек жалақысы фондының 50% құрайды, (5.24) формуласы бойынша анықталады:

$$C_{\text{накл}} = 0,5 * \text{ФОТ}; \quad (5.24)$$

$$C_{\text{накл}} = 0,5 * 137572.8 = 68786.4 \text{ теңге.}$$

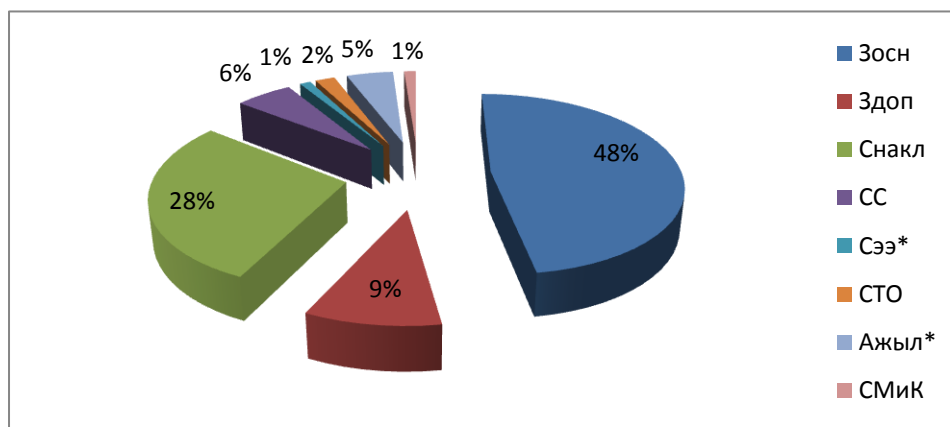
5.5 Өзіндік құн нәтижесінің кестесі

Бағдарламалық қамтаманы өндеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру, жүктеме шығындар мен материал және құрамалардың қосындысына тең.

Кесте 5.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сумма, теңге	Жалпы суммадан процентпен
ФЗП	$Z_{\text{осн}}$	114644	47.49
	$Z_{\text{доп}}$	22928.8	9.49
Жүктеме шығындар, $C_{\text{накл}}$		68786.4	28,49
Әлеуметтік сақтандыру, CC		13620	5.64
Эксплуатационды Шығындар	$C_{\text{ээ}}^*$	2597.7	1.07
	$C_{\text{ТО}}$	4725	1.95
	$A_{\text{жыл}}^*$	11221.8	4.64
Материалдар мен комплектілер, $C_{\text{мик}}$		2835	1.17
Барлығы:		241358.7	100

Өзіндік құнының диаграммасы:



Сурет 5.1 – Өнімнің өзіндік құнының диаграммасы

Бағдарламаның бағасына есептеу.

Өнімнің өзіндік құны 241358.7 теңге (жоғарыда кесте 1) болды, онда бір енгізу объектісіне арналған жүйенің толық өзіндік құны ($C_{пс}$):

$$C_{пс} = 241358.7 / 3 = 80452.9 \text{ теңге}$$

Минималды бағаны, кірісті және пайданы есептеу.

Өнімнің минималды бағасы келесі формуламен есептеледі:

$$C_{\min} = (C_{пс} + C_{мд} + C_{з}) (1 + r / 100) \quad (5.25)$$

$$C_{\min} = (80452.9 + 180 + 150) * 1,21 = 97747.3 \text{ теңге}$$

мұндағы $C_{мд}$ - магнитті диск бағасы, қазіргі нарықтық бағасын қарастырып, 180 теңге деп алынды.

Өзіндік құнға қатынасымен алынған тиімділіктің жоспарланған пайызы, ол $r = 21\%$. НДС ті есептегендегі минималды баға келесі формуламен есептеледі (ҚР да НДС 12%):

$$C_{\min_{НДС}} = C_{\min} + N_{НДС} C_{\min} \quad (5.26)$$

$$C_{\min_{НДС}} = 97747.3 + 0.12 * 97747.3 = 109476.9 \text{ теңге}$$

Кіріс келесі формуламен есептеледі:

$$V_{\text{реал}} = C_{\min_{НДС}} * V_{\text{объем}} \quad (5.27)$$

$$V_{\text{реал}} = 109476.9 * 3 = 328430.9 \text{ теңге}$$

мұндағы $V_{\text{объем}}$ – бағдарламаны жүзеге асыру көлемі.

Жалпы пайда келесі формуламен есептеледі:

$$П = V_{\text{реал}} - V_{\text{реал}} * N_{\text{ндс}} - C_{\text{разраб}} \quad (5.28)$$

$$П = 168254328430.9 - 328430.9 * 0.12 - 241358.7 \text{ теңге}$$

мұндағы НДС – қосылған құн салығы (12%).

Таза пайда келесі формуламен есептеледі:

$$П_{\text{чист}} = П - N_{\text{прибыль}} \quad (5.29)$$

$$П_{\text{чист}} = 47660.5 - 0.2 * 47660.5 = 38128.4 \text{ теңге}$$

мұндағы $N_{\text{прибыль}}$ – пайдаға салынатын салық (ҚР да $N_{\text{прибыль}} = 20\%$)

5.6 Қосымшаны иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу

Кәсіпорындардың бірмезгілдегі шығындары келесі бөлімдерден тұрады:

- жүйе құны;
- көлік шығындары;
- оқуға кететін төлемдер.

Жүйе құны 109476.9 теңге (5.26 формуласы).

Көлік шығындары жүйе құнының 20% құрайды:

$$109476.9 * 0.2 = 21895.38 \text{ теңге}$$

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 2500 теңгедей болып табылады [15].

Соған сәйкес оқу бағасы:

$$3000 * 2 = 6000 \text{ теңге}$$

Кесте 5.4 – Ақпаратты жүйені енгізуге арналған бірмезгілдегі шығындардың есептелу нәтижесі

Шығындар түрлері	Сумма, теңге
------------------	--------------

Жүйе құны	42063.6
Көлік шығындары	8412.72
Жұмысшыларды оқыту бағасы	5000
Қорытынды:	55476.32

5.7 Қолдану саласындағы жылдық бірмезгілдегі шығындарды есептеу

Негізгі шығындар түрлері:

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

Модератордың жалақысы 42000 теңге.

Бір жылдық жалақысын есептесек 504000 теңге.

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

$$(504000 - 504000 * 0.1) * 0.11 = 49896 \text{ теңге}$$

К е с т е 5.5 – Кәсіпорынның жылдық бірмезгілдегі шығындарын есептеу нәтижесі

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	600000
Әлеуметтік салыққа аударымдар	59400
Қорытынды:	659400

5.8 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу

Ақпараттық өнімді құрастыратын фирма үшін табыс көзі ретінде тапсырыс берушіге бағдарламалық өнімді сату болып табылады. Фирманың шығындары – ол құрастыруға кететін шығындар және жүйенің көбейтілуіне кететін шығындар. Қаржыландыру көзі ретінде құрастырушы-фирманың өзінің ақша қоры алынады.

Тапсырыс беруші-кәсіпорын үшін үнемдеу көзі есептеу уақытын біршама қысқартатын, қол еңбегін алмастыратын машина болып табылады. Кәсіпорын шығындары ақпараттық жүйені иемденудегі бірмезгілдік шығындардан, ақпараттық жүйені тасымалдау және енгізу шығындарынан, және де жүйеге талдау жүргізу мен бақылауға кететін шығындардан тұрады.

Үнемдеу мөлшерін есептейік. Ол үшін өткізу қабілетін есептеуге кететін кәсіпорынның саяхатшы гидіне кеткен шығындарын есептеу керек.

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

Қол еңбегімен келесі шығындар түрлері есептеледі:

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

ПК операторының жылдық жалақысы 600000 теңге

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

$$(600000-600000*0,1)*0,11=59400 \text{ теңге}$$

К е с т е 5.6 – Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығындарды есептеу

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	600000
Әлеуметтік салыққа аударымдар	59400
Қорытынды:	659400

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 3857.98 теңге болды. Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығын 3857.98 теңге болды.

Осыған сәйкес, шығындардың шартты үнемделуі мынаған тең болады

$$С_{ЭК}=659400-553896 =123076.5 \text{ теңге}$$

5.9 Пайда мен шығындарды есептеу

Пайда мен шығындарды есептеу кесте 5.7 те көрсетілген.

Кесте 5.7 – Бір жылдағы пайда мен шығындарды есептеу

Аты	Жыл
Шығындардың шартты үнемделуі, теңге	659411.48
Ағымдағы шығындар, теңге	123076.5
Шартты пайда, теңге	14769.1
Пайдаға салынатын салық, теңге	21403.72
Таза пайда, теңге	108307.3

Ақшалай құралдардың қозғалысы.

К е с т е 5.8 – Ақшалай тәсілдердің қозғалысы, теңге

Аты	Жылдар		
	2014	2015	2016

Бірмезгілдік шығындар	137372,28		
Шығындарды үнемдеу		123076,5	123076,5
Ағымдағы шығындар		659411,48	659411,48
Пайдаға салынатын салық		14769,1	14769,1
Таза табыс	137372,28	108307,3	108307,3
Дисконттау коэффициенті (21% мөлшерінде)	1	0,826	0,683
Таза дисконтталған табыс (ТДТ)	-137372,28	89461,8	73973,8
ТДТ нәтижесімен	-137372,28	-47910,48	26063,32

$$K_{д2} = 1/(1+\alpha)^1 \quad (5.30)$$

$$K_{д2} = 1/(1+0,21)^1 = 0.826$$

$$K_{д3} = 1/(1+\alpha)^2 \quad (5.31)$$

$$K_{д3} = 1/(1+0,21)^2 = 0.683$$

мұндағы α – инфляция коэффициенті:

$K_{д}$ - дисконттау коэффициенті.

Таза дисконтталған табыс (ТДТ) = Таза табыс* Дисконттау коэффициенті.

Таза дисконтталған табыс:

$$(ТДТ) = 108307.3 * 0.826 = 89461.8 \text{ теңге}$$

$$108307.3 * 0.683 = 739973.8 \text{ теңге.}$$

ТДТ₂ (өспелі нәтижесімен) = ТДТ₁ + Таза дисконтталған табыс (ТДТ)₂

5.10 Экономикалық тиімділікті есептеу

Шығындары бізмезгілде тек жобаның басында күрделі салымдар (C₀) ретінде іске асатын жобалар үшін NPV келесі формуламен есептеледі:

$$NPV = \sum_{i=1}^n \frac{B_i}{(1+r)^i} - C_0.$$

Мұндағы B_i –і-ші жылдағы жобадан алынатын пайда;
r – дисконттеу мөлшері

$$NPV (21\%) = \left(\frac{108307,3}{(1+0,21)^1} + \frac{108307,3}{(1+0,21)^2} \right) - 137372,28 = 26118,2 \text{ теңге} > 0$$

сондықтан ұсынылып отырған жоба табысты. Сол себептен жобаны қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

PI жобаның салыстырмалы пайдасын көрсетеді. Бұл көрсеткіш келесі формуламен есептеледі:

$$PI = \frac{NPV}{C_0}$$

$$PI = (163435,5/137372,28) = 1,18\%$$

Табыстың ішкі нормасы (ТИН немесе IRR) r пайыздық мөлшерінің мәні болып табылады, мұндағы NPV = 0. Бұл нүктеде r* қосылған шығындардың дисконтталған ағыны қосылған дисконтталған табыс ағындарына тең. IRR = r, мұндағы NPV = 0.

$$IRR = r1 + \frac{f(r1)}{f(r1) - f(r2)} * (r2 - r1)$$

r_b=31%:

$$PV_1 = 108307,3 / (1 + 0,31) = 79673,3 \text{ теңге}$$

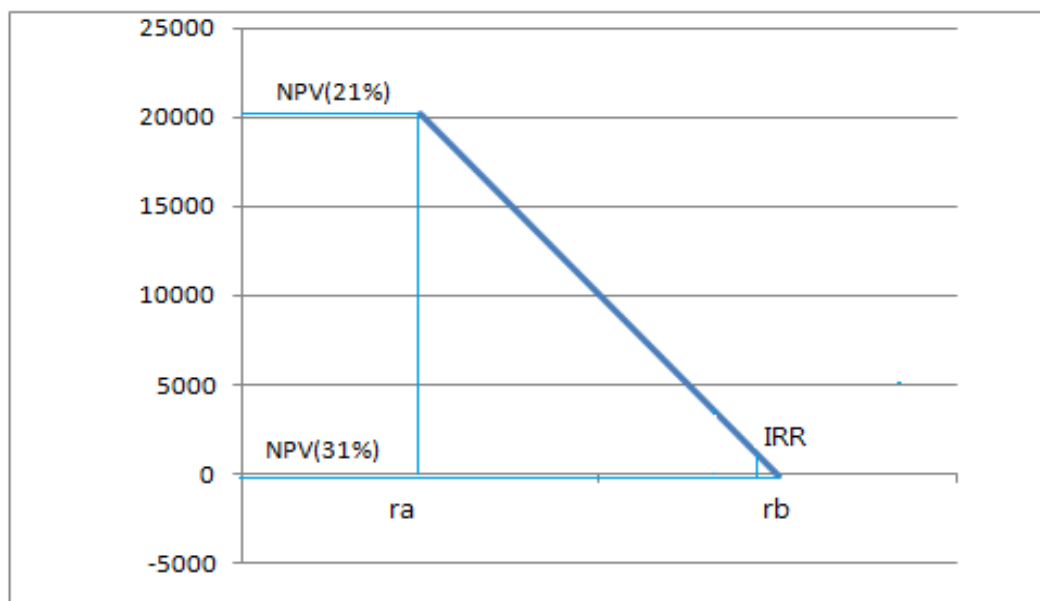
$$PV_2 = 108307,3 / (1 + 0,31)^2 = 57194,9 \text{ теңге}$$

$$NPV(31\%) = (79673,3 + 57194,9) - 137372,28 = -504,08 \text{ т}$$

$$NPV(21\%) = 26118,2 \text{ теңге}$$

$$NPV(31\%) = 504,08 \text{ теңге}$$

$$IRR = r_a + (r_b - r_a) * NPV_a / (NPV_a - NPV_b) = 21 + (31 - 21) * 26118,2 / (26118,2 - (-504,08)) = 30,81\%$$



Өтімділік периоды мына формуламен есептеледі:

$$T_{ок} = t + \frac{C_0 - \sum_{i=1}^n B_i}{B_{i+1}}$$

Мұндағы:

C_0 – жоба басындағы салым;

B_i – i -ші жылдағы жобадан алынатын пайда;

Сонда біздің өтімділік периодымыз:

$$T_{ок} = 1 + \frac{137372,28 - 89461,8}{73973,8} = 1 \text{ жыл } 8 \text{ ай}$$

Бұл жобаның өтімділік периоды 1 жыл 8 айға тең, яғни жобамыз осы уақытта өз-өзін ақтайды.

Қорытынды

Операциондық жүйені жұмысқа қажетті етіп, тек қана қосымша жүзеге асырады. Қосымша ғаламторға шығатын мүмкіндік болған жағдайда, кез келген уақытта дистанционды түрде тестілеуден өтуге болады. Қосымша әр түрлі платформаларға арналып жасалған, яғни кроссплатформалы мобильдік қосымша. Алынған практикадан, өзімнің жеке арнай қосымша жасауым және жаңадан ақпараттарды игеруім көп уақытты алады. Жасалған дипломдық жобамда клиенттік-серверлерге қосымша жасау және жоғары дәрежеде кодтау тәжірбиесін меңгердім.

Дипломдық жобаның мақсаты - Дистанционды оқытудағы тестілеу жүйесінің дамыту. Кроссплатформалық мобильдік қосымша құрастыру. Жасалып отырған проектіде дистанционды түрде білім алушыларға арналған бағдарлама жасадым. Сонда тест тапсырушылар уақытын үнемдейді және ыңғайлы әрі жедел түрде тестілеуден өтуге мүмкіндік алады. Зертеулер бойынша, таңдаған тематикама байланысты бағдарлама жасалмаған және еш қандай бәсекелістік жоқ.

Осы жобаға қосымша құрастыру барысында Android, IOS, Windows Phone операциялық жүйеслеріне жаңа анализдер жүргізілді, олар жаңа және тез жұмыс істейтін техниканы қажет етпейді. Алынған деректерге сүйене, сонында біз қандай технологиямен бірге жұмыс істейтінімізді анықталды. Қиын және маңызды жобаларды шеше отырып мен өзімнің қосымшамның қажеттілігін және IT аумағында жаңа идеяларды еңгіздім, іске асырдым.

Дипломдық жобаға берілген есебім, мақсатым толығымен орындалды.

Қолданылған әдебиеттер тізімі

- 1 Голошапов А. Google Android программирование для мобильных устройств. Санк-Петербург 2011-438 с.
- 2 Соснов А. Основы проектирование информационных систем. – М.: ДМК Пресс, 2002. – 1020 с.
- 3 Марк Шпеник, Оррин Следж. Microsoft SQL Server 2000 DBA Survival Guide. – М.: Вильямс, 2001. – 236 с.
- 4 Мамаев Е. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: BHV, 2004. – 416с.
- 5 Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Вильямс, 2000.
- 6 Глушаков С.В., Ломотько Д.В. Базы данных: учеб. курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
- 7 Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных. МУ к выполнению лабораторных работ. – Алматы: АИЭС, 2004.
- 8 Айтхожаева Е.Ж. Стандартный язык баз данных SQL: учеб. пособие. – Алматы: АИЭС, 2005. - 48 с.
- 9 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург, 2006.
- 10 Кандзюба С.П., Громов В.Н. Delphi 6/7. Базы данных и приложения. – СПб: ООО «ДиаСофт», 2002.
- 11 Архангельский А. Я. Программирование в Delphi 7. – М.: Бином, 2003. – 980 с.
- 12 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат. – Алматы: АИЭС, 2009. – 20 с.
- 13 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
- 14 Голубицкая Е.А., Жигульская Г.М. Экономика связи. – М.: Радио и связь, 1999.
- 15 Резникова Н.П. Маркетинг в телекоммуникациях. – М.:ЭКО-ТРЭНДЗ, 1998.
- 16 Хавронская А.М. Оценка технико-экономической эффективности программных средств. – Алматы: КазНТУ, 2000.
- 17 Линаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика, 1998.

Қосымша А

Класстар құруға арналған скрипт

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

namespace AdminRDO
{
    public partial class AdminOption : Form
    {
        public AdminOption()
        {
            InitializeComponent();

            a1 = "select u.Firstname as 'Тегі', u.Lastname as 'Аты',
u.Patronymic as 'Әкесінің аты', s.sub_name as 'Пән',
u.Numbquestion as 'Сұрақ саны', u.Correct as 'Дұрыс жауаптар',
u.Incorrect as 'Қате жауаптар', u.procent as 'Бағасы %', u.dateT
as 'Тест уақыты' from Users u inner join subject s on s.sub_id =
u.sub_id";
            connect();

            b1 = "select q.Id, s.sub_name as 'Пән', q.Quest as 'Сұрақ',
q.Correct as 'Дұрыс жауап', q.False1 as 'Қате жауап 1', q.False2
as 'Қате жауап 2', q.False3 as 'Қате жауап 3' from Questions q
inner join subject s on s.sub_id = q.sub_id";
            connect2();
            dataGridView2.Columns["Id"].Visible = false;

            var pan = from i in Program.db.subjects select i.sub_name;
            foreach (var i in pan)
            {
                comboBox1.Items.Add(i);
                comboBox2.Items.Add(i);
            }
        }

        private void dataGridView1_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
        {
            int index = e.RowIndex;
            string indexStr = (index + 1).ToString();
        }
    }
}
```

А қосымшасының жалғасы

```
        object header =
this.dataGridView1.Rows[index].HeaderCell.Value;
if (header == null || !header.Equals(indexStr))
this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;
}

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

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

public void connect2()
{
SqlConnection cn = Program.conn;
b = b1;
DataSet ds1 = new DataSet();
SqlDataAdapter da1 = new SqlDataAdapter(b, cn);
da1.Fill(ds1);
dataGridView2.DataSource = ds1.Tables[0];
}
private void button5_Click(object sender, EventArgs e)
{
this.Close();
}

private void button6_Click(object sender, EventArgs e)
{
a1 = "select u.Firstname as 'Тері', u.Lastname as 'Аты',
u.Patronymic as 'Әкесінің аты', s.sub_name as 'Пән',
u.Numbquestion as 'Сұрақ саны', u.Correct as 'Дұрыс жауаптар',
u.Incorrect as 'Қате жауаптар', u.procent as 'Бағасы %', u.dateT
as 'Тест уақыты' from Users u inner join subject s on s.sub_id =
u.sub_id";
```

А қосымшасының жалғасы

```
connect();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    a1 = "select u.Firstname as 'Тегі', u.Lastname as 'Аты',
u.Patronymic as 'Әкесінің аты', s.sub_name as 'Пән',
u.Numbquestion as 'Сұрақ саны', u.Correct as 'Дұрыс жауаптар',
u.Incorrect as 'Қате жауаптар', u.procent as 'Бағасы %', u.dateT
as 'Тест уақыты' from Users u inner join subject s on s.sub_id =
u.sub_id where u.Firstname like N'" + textBox1.Text + "%'";
    connect();
}

private void button1_Click(object sender, EventArgs e)
{
    a1 = "select u.Firstname as 'Тегі', u.Lastname as 'Аты',
u.Patronymic as 'Әкесінің аты', s.sub_name as 'Пән',
u.Numbquestion as 'Сұрақ саны', u.Correct as 'Дұрыс жауаптар',
u.Incorrect as 'Қате жауаптар', u.procent as 'Бағасы %', u.dateT
as 'Тест уақыты' from Users u inner join subject s on s.sub_id =
u.sub_id where (DATEPART(yy, u.dateT) = " +
dateTimePicker1.Value.Year + " and DATEPART(mm, u.dateT) = " +
dateTimePicker1.Value.Month + " and DATEPART(dd, u.dateT) = " +
dateTimePicker1.Value.Day + ")";
    connect();
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    b1 = "select q.Id, s.sub_name as 'Пән', q.Quest as 'Сұрақ',
q.Correct as 'Дұрыс жауап', q.False1 as 'Қате жауап 1', q.False2
as 'Қате жауап 2', q.False3 as 'Қате жауап 3' from Questions q
inner join subject s on s.sub_id = q.sub_id where q.Quest like '"
+ textBox2.Text + "%'";
    connect2();
    dataGridView2.Columns["Id"].Visible = false;
}

private void button2_Click(object sender, EventArgs e)
{
    var idq = dataGridView2.CurrentRow.Cells[0].Value;
    SqlConnection sqlConnection1 = Program.conn;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "delete from Questions where Id = '" + idq
+ "'";
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
}
```

А қосымшасының жалғасы

```
sqlConnection1.Close();
MessageBox.Show("Сұрақ өшірілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
b1 = "select q.Id, s.sub_name as 'Пән', q.Quest as 'Сұрақ',
q.Correct as 'Дұрыс жауап', q.False1 as 'Қате жауап 1', q.False2
as 'Қате жауап 2', q.False3 as 'Қате жауап 3' from Questions q
inner join subject s on s.sub_id = q.sub_id where q.Quest like N'"
+ textBox2.Text + "%'";
connect2();
dataGridView2.Columns["Id"].Visible = false;
}

private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
{
b1 = "select q.Id, s.sub_name as 'Пән', q.Quest as 'Сұрақ',
q.Correct as 'Дұрыс жауап', q.False1 as 'Қате жауап 1', q.False2
as 'Қате жауап 2', q.False3 as 'Қате жауап 3' from Questions q
inner join subject s on s.sub_id = q.sub_id where s.sub_name like
N'" + comboBox2.Text + "%'";
connect2();
dataGridView2.Columns["Id"].Visible = false;
}

private void button4_Click(object sender, EventArgs e)
{
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
textBox6.Clear();
textBox7.Clear();
}

private void button3_Click(object sender, EventArgs e)
{
if (string.IsNullOrEmpty(textBox3.Text) ||
string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox5.Text) ||
string.IsNullOrEmpty(textBox6.Text) ||
string.IsNullOrEmpty(textBox7.Text))
{
MessageBox.Show("Барлақ жолдарды толтырыңыз", "Назар
аударыңыз", MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
else
{
int si = (from i in Program.db.subjects where i.sub_name ==
comboBox1.Text select i.sub_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
```


А қосымшасының жалғасы

```
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Questions(Quest, Correct,
False1, False2, False3, sub_id) values ('" + textBox3.Text + "',
'" + textBox4.Text + "', '" + textBox5.Text + "', '" +
textBox6.Text + "', '" + textBox7.Text + "', '" + si + "')";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Жаңа сұрақ енгізілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
textBox6.Clear();
textBox7.Clear();
}
}
}
```

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;
using System.Data;
using System.Web.Services.Description;
using JsonP.Models;
using Microsoft.Ajax.Utilities;
using Microsoft.Data.Edm.Library.Values;

namespace JsonP.Controllers
{
    public class HomeController : Controller
    {
        QuestionContext _db = new QuestionContext();
        public ActionResult Index()
        {
            var allQuest = _db.Questions.SqlQuery("Select * from
Questions");
            ViewBag.Message = "Modify this template to jump-start your
ASP.NET MVC application.";
            return View(allQuest.ToList());
        }

        public ActionResult About()
        {
```

А қосымшасының жалғасы

```
ViewBag.Message = "Your app description page.";

return View();
}

public ActionResult Contact()
{
    ViewBag.Message = "Your contact page.";

    return View();
}

[HttpGet]
public JsonResult GetData(int pred)
{
    var data = (from item in _db.Questions where item.sub_id ==
pred select new { Quest = item.Quest, Correct = item.Correct,
False1 = item.False1, False2 = item.False2, False3 = item.False3
}).ToArray();
    JsonResult result = new JsonResult(data.ToArray());
    return result;
}

[HttpGet]
public JsonResult GetData2(string firstname, string
lastname, string patronimic, int subject, int numbfOfQuestions, int
countCorrectAnswers, int countInCorrectAnswers, float procent)
{
    SqlConnection sqlConnection1 = new SqlConnection(@"Data
Source=(LocalDb)\v11.0;AttachDbFilename=C:\Users\АpxaT\Desktop\Jso
nP\JsonP\App_Data\Database.mdf;Initial Catalog=aspnet-JsonP-
20140115010050;Integrated Security=True");
    sqlConnection1.Open();
    //SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "Insert into Users(Firstname, Lastname,
Patronimic, Numbquestion, Correct, Incorrect, sub_id, procent,
dateT) values(N'" + firstname + "', N'" + lastname + "', N'" +
patronimic + "', N'" + numbfOfQuestions + "', N'" +
countCorrectAnswers + "', N'" + countInCorrectAnswers + "', N'" +
subject + "', N'" + procent + "', GETDATE())";
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    sqlConnection1.Close();
    var data = "Запрос прошел успешно";
    JsonResult result = new JsonResult(data.ToString());
    return result;
}
}
}
```

А қосымшасының жалғасы

```
$(document).ready(function() {
var userid,firstname, lastname, patronimic, predmet;
var question;
var correct;
var incorrect1;
var incorrect2;
var incorrect3;
var sourcequestion;
var dataset;
var numbfQuestions;
var myAnswer;
var ansText;
var recId;
var num = 0;

var allQuestions = [];
var questionArray = [];
var correctArray = [];
var incorrectArray1= [];
var incorrectArray2 = [];
var incorrectArray3 = [];
var userAnswerArray = [];

var checkedAnswers = [];
var countCorrectAnswers = 0;
var countInCorrectAnswers = 0;
var countSource1 = 0;
var countSource2 = 0;
var countSource3 = 0;
var id;
var procent;

$('#result').hide(); //СІПеСБС<С,СБ P±P»PsPe
СБPµP·СІP»СБС,Р°С,PsPI
var db = window.openDatabase("TestStudy", "1.0", "TestStudy",
5*1024*1024); //СІPSP·PrP°PSPëPµ P±P°P·C<
db.transaction(populatedDB, errorCallback);

function populatedDB(tx) { //СІPSP·PrP°PSPëPµ C,P°P±P»PëC†
tx.executeSql('DROP TABLE IF EXISTS Questions');
tx.executeSql('CREATE TABLE IF NOT EXISTS Questions (id
INTEGER PRIMARY KEY AUTOINCREMENT, question TEXT, correct TEXT,
incorrect1 TEXT, incorrect2 TEXT, incorrect3 TEXT, userAnswer
TEXT) ');
}

function errorCallback(tx, err) { //PIC<PIPsPr PiСБPë PsСëPëP±PePµ
alert("Error processing SQL: "+err);
}
```

А қосымшасының жалғасы

```
db.transaction(function (tx) {
  tx.executeSql('SELECT * FROM Users ORDER BY id DESC LIMIT 1',
[], function (tx, result) {
  var userSet = result.rows;
  for (var i = 0, itemD = null; i < userSet.length; i++) {
    itemD = userSet.item(i);
    userid = itemD['id'];
    firstname = itemD['firstname'];
    lastname = itemD['lastname'];
    patronimic = itemD['patronimic'];
    predmet = itemD['predmet'];

    if(predmet == "Рыбсыржыпичык,ручур»с-ре ррру»с-р»руч") {
      id = 1;
      $.getJSON("http://localhost:45100/home/GetData?pred=" + id +
"&callback=?", function (data) {
        db.transaction(function (tx) {
          for(var i=0; i <= data.length; i++){
            if(i == data.length){
              tx.executeSql('SELECT * FROM Questions', [], function (tx,
result) {
                numbofQuestions = result.rows.length;
                $('#slider').attr('max', numbofQuestions);
                showRecords();
              })
            }
            else{
              var jsonString = JSON.stringify(data[i]);
              var jsonParse = JSON.parse(jsonString);
              question = jsonParse.Quest;
              correct = jsonParse.Correct;
              incorrect1 = jsonParse.False1;
              incorrect2 = jsonParse.False2;
              incorrect3 = jsonParse.False3;
              tx.executeSql('INSERT INTO Questions (question, correct,
incorrect1, incorrect2, incorrect3) VALUES (" + question + "', " +
+ correct + "', " + incorrect1 + "', " + incorrect2 + "', " +
incorrect3 + "')');
            }
          }
        })
      })
    }

    else if(predmet == "Тыр°р·р°т>сїс,р°рс с,р°сррөс...с<") {
      id = 2;
      $.getJSON("http://localhost:45100/home/GetData?pred=" + id +
"&callback=?", function (data) {
        db.transaction(function (tx) {
```

А қосымшасының жалғасы

```
for(var i=0; i <= data.length; i++){
  if(i == data.length){
    tx.executeSql('SELECT * FROM Questions', [], function (tx,
result) {
    numOfQuestions = result.rows.length;
    $('#slider').attr('max', numOfQuestions);
    showRecords();
  })
  }
  else{
    var jsonString = JSON.stringify(data[i]);
    var jsonParse = JSON.parse(jsonString);
    question = jsonParse.Quest;
    correct = jsonParse.Correct;
    incorrect1 = jsonParse.False1;
    incorrect2 = jsonParse.False2;
    incorrect3 = jsonParse.False3;
    tx.executeSql('INSERT INTO Questions (question, correct,
incorrect1, incorrect2, incorrect3) VALUES (" + question + "', "'
+ correct + "', "' + incorrect1 + "', "' + incorrect2 + "', "' +
incorrect3 + '"')');
  }
}
})
}
}
})
})
})
```

```
function showRecords()//P·P°PiPsP»PSPµPSPëPµ PIPSPiCᵀPsCÍPsPI
{
  db.transaction(function (tx) {
    tx.executeSql('SELECT * FROM Questions', [], function (tx,
result) {
    dataset = result.rows;
    var ranQuest = [];
    for (var i = 0, item = null; i <= numOfQuestions; i++) {
    var rand = Math.round((Math.random()*4)+1|0);
    if( i == numOfQuestions){
    $('#slider').val(1).slider('refresh');
    }
    else{
    item = dataset.item(i);
    questionArray.push(item['question']);
    correctArray.push(item['correct']);
    incorrectArray1.push(item['incorrect1']);
    incorrectArray2.push(item['incorrect2']);
    incorrectArray3.push(item['incorrect3']);
    }
    }
  })
}
```

А қосымшасының жалғасы

```
userAnswerArray.push(000);

if((rand == 1) && (rand != numOfQuestions)){
    ranQuest.push(questionArray[i], correctArray[i],
incorrectArray1[i], incorrectArray2[i], incorrectArray3[i]);
    allQuestions.push(ranQuest);
    ranQuest = [];
}
else if((rand == 2) && (rand != numOfQuestions)){
    ranQuest.push(questionArray[i], incorrectArray1[i],
correctArray[i], incorrectArray2[i], incorrectArray3[i]);
    allQuestions.push(ranQuest);
    ranQuest = [];
}
else if((rand == 3) && (rand != numOfQuestions)){
    ranQuest.push(questionArray[i], incorrectArray1[i],
incorrectArray2[i], correctArray[i], incorrectArray3[i]);
    allQuestions.push(ranQuest);
    ranQuest = [];
}
else if((rand == 4) && (rand != numOfQuestions)){
    ranQuest.push(questionArray[i], incorrectArray1[i],
incorrectArray2[i], incorrectArray3[i], correctArray[i]);
    allQuestions.push(ranQuest);
    ranQuest = [];
}
}
}
}
}
}

function insertForm(IdNum) { //P·P°PіPSP»PSPµPSPĖPµ
СҚР»PµPjPµPSC,PsPI C,,PsCБPjC<
    var item1 = allQuestions[IdNum][0];
    var item2 = allQuestions[IdNum][1];
    var item3 = allQuestions[IdNum][2];
    var item4 = allQuestions[IdNum][3];
    var item5 = allQuestions[IdNum][4];
    $('#textarea-1').val(item1);
    var answer1 = $('label[for=radio-choice-v-6a]').text(item2);
    var answer2 = $('label[for=radio-choice-v-6b]').text(item3);
    var answer3 = $('label[for=radio-choice-v-6c]').text(item4);
    var answer4 = $('label[for=radio-choice-v-6d]').text(item5);

    if(userAnswerArray[IdNum] == 000){
        $('input[type=radio]').prop('checked',
false).checkboxradio('refresh');
    }
    else{
```

А қосымшасының жалғасы

```
var answer1 = $('label[for=radio-choice-v-6a]').html();
var answer2 = $('label[for=radio-choice-v-6b]').html();
var answer3 = $('label[for=radio-choice-v-6c]').html();
var answer4 = $('label[for=radio-choice-v-6d]').html();
if(answer1 == userAnswerArray[IdNum]){
    $('input[type=radio]').prop('checked',
false).checkboxradio('refresh');
    $('#radio-choice-v-6a').prop('checked',
true).checkboxradio('refresh');
}

else if(answer2 == userAnswerArray[IdNum]){
    $('input[type=radio]').prop('checked',
false).checkboxradio('refresh');
    $('#radio-choice-v-6b').prop('checked',
true).checkboxradio('refresh');
}

else if(answer3 == userAnswerArray[IdNum]){
    $('input[type=radio]').prop('checked',
false).checkboxradio('refresh');
    $('#radio-choice-v-6c').prop('checked',
true).checkboxradio('refresh');
}

else if(answer4 == userAnswerArray[IdNum]){
    $('input[type=radio]').prop('checked',
false).checkboxradio('refresh');
    $('#radio-choice-v-6d').prop('checked',
true).checkboxradio('refresh');
}
}
}

function updateRecord(rec)//PëP·PjPµPSPµPSPëPµ PI
C,P°P±P»PëC±Pµ PìPSP»СЪP·PSPIP°C,PµP»PµPN°
{
    recId = num + 1;
    db.transaction(function (tx) {
        tx.executeSql('UPDATE Questions SET userAnswer = "' + rec +
'" WHERE id = ' + recId);
    })
}

function answerCountResult() {//PµPSPrCÍC±PµC, PsC,PIPµC,PsPI
db.transaction(function (tx) {
    tx.executeSql('SELECT * FROM Questions', [], function (tx,
result) {
        var datacheck = result.rows;
```

А қосымшасының жалғасы

```
for (var i = 0, item = null; i < datacheck.length; i++) {
    item = datacheck.item(i);
    checkedAnswers.push(item['userAnswer']);
}

for(var i = 0; i < numbfOfQuestions; i++){
    if(checkedAnswers[i] == correctArray[i]){
        countCorrectAnswers = countCorrectAnswers + 1;
        if(i == (numbfOfQuestions -1)){
            procent =
Math.round((countCorrectAnswers*100)/numbfOfQuestions);
            $.getJSON("http://localhost:45100/home/GetData2?firstname=" +
    firstname + "&lastname=" + lastname + "&patronimic=" + patronimic
    + "&subject=" + id + "&numbfOfQuestions=" + numbfOfQuestions +
    "&countCorrectAnswers=" + countCorrectAnswers +
    "&countInCorrectAnswers=" + countInCorrectAnswers + "&procent=" +
    procent + "&callback=?", function (data) {
                var jsonString = JSON.stringify(data);
                var jsonParse = JSON.parse(jsonString);
                //alert(jsonParse);
            })
            db.transaction(function (tx) {
                tx.executeSql('UPDATE Users SET numberofquestions = ' +
    numbfOfQuestions + ', countcorrect = ' + countCorrectAnswers + ',
    countincorrect = ' + countInCorrectAnswers + ', procent = ' +
    procent + ', datetest = datetime("now","localtime") WHERE id = ' +
    userid);
                $('#fio').prepend(firstname + " " + lastname + " " +
    patronimic);
                $('#allCount').prepend(numbfOfQuestions);
                $('#Cor').prepend(countCorrectAnswers);
                $('#InCor').prepend(countInCorrectAnswers);
                $('#otc').prepend(procent + "%");
                $('#tableResult').prepend();
                $('#page1').hide();
                $('#result').show();
            })
        }
    }
    else{
        countInCorrectAnswers = countInCorrectAnswers + 1;
        if(i == (numbfOfQuestions -1)){
            procent =
Math.round((countCorrectAnswers*100)/numbfOfQuestions);
            $.getJSON("http://localhost:45100/home/GetData2?firstname=" +
    firstname + "&lastname=" + lastname + "&patronimic=" + patronimic
    + "&subject=" + id + "&numbfOfQuestions=" + numbfOfQuestions +
    "&countCorrectAnswers=" + countCorrectAnswers +
    "&countInCorrectAnswers=" + countInCorrectAnswers + "&procent=" +
    procent + "&callback=?", function (data) {
```


А қосымшасының жалғасы

```
var jsonString = JSON.stringify(data);
var jsonParse = JSON.parse(jsonString);
//alert(jsonParse);
})
db.transaction(function (tx) {
    tx.executeSql('UPDATE Users SET numberOfquestions = ' +
numbOfQuestions + ', countcorrect = ' + countCorrectAnswers + ',
countincorrect = ' + countInCorrectAnswers + '&procent = ' +
procent + ', datetest = datetime("now","localtime") WHERE id = ' +
userid);
    $('#fio').prepend(firstname + " " + lastname + " " +
patronimic);
    $('#allCount').prepend(numbOfQuestions);
    $('#Cor').prepend(countCorrectAnswers);
    $('#InCor').prepend(countInCorrectAnswers);
    $('#otc').prepend(procent + "%");
    $('#tableResult').prepend();
    $('#page1').hide();
    $('#result').show();
})
}
}
}
}
}
}

function answerCount(){ //PePsP»PëC#PµCÍC,PIP°
PIPsPiCßPsCÍPsPI Pë PsPiPµCßP°C+PëPë
db.transaction(function (tx) {
var confMas = [];
var conStr = '';
tx.executeSql('SELECT * FROM Questions WHERE userAnswer IS
NULL' , [], function(tx2, result2){
var dataExec = result2.rows;
for (var i = 0, item = null; i < dataExec.length; i++) {
item = dataExec.item(i);
confMas.push(item['id'])
}
if((userAnswerArray.length != dataExec.length) ||
(dataExec.length == userAnswerArray.length)) && (dataExec.length
!= 0)){
for(var i = 0; i < confMas.length; i++){
conStr = conStr + ", " + confMas[i];
}
var usAns = confirm("PŸC-P· PePµP»PµCÍC- CÍT±CßP°T>C,P°CßT`P°
P¶P°CíP°Pi P±PµCßPiPµPS P¶PsT>CÍC<P·: " + confMas.join('; ') +
"\n" + "PŸPµCÍC,C-P»PµCíPrC- P°C¶T>C,P°Cí?");
if(usAns == true){
answerCountResult();
}
```

```

    }
    }
    else{
    answerCountResult();
    }
    })
    })
    }

    $('input[type=radio]').change(function(){//PіCBPsPIPμCBPeP°
PsC,PIPuC,PsPI
    myAnswer = $('input[type=radio]:checked').val();
    if(myAnswer == 'answer1'){
    ansText = $('label[for=radio-choice-v-6a]').text();
    userAnswerArray[num] = ansText;
    updateRecord(ansText);
    }

    else if(myAnswer == 'answer2'){
    ansText = $('label[for=radio-choice-v-6b]').text();
    userAnswerArray[num] = ansText;
    updateRecord(ansText);
    }

    else if(myAnswer == 'answer3'){
    ansText = $('label[for=radio-choice-v-6c]').text();
    userAnswerArray[num] = ansText;
    updateRecord(ansText);
    }

    else if(myAnswer == 'answer4'){
    ansText = $('label[for=radio-choice-v-6d]').text();
    userAnswerArray[num] = ansText;
    updateRecord(ansText);
    }
    })

    $('button[name=submitTest]').click(function(){
    answerCount();
    })

    $('#next').click(function(){ //CÍP»PuPr. PIPsPіCBPsCÍ
    if((num < numbOfQuestions - 1)){
    num = num + 1;
    $('#slider').val(num + 1).slider('refresh');
    }
    else{
    alert("PŸT±CBP°T>C,P°CB P¶PsT>");
    }
    })

```

А қосымшасының жалғасы

```
$('#prev').click(function(){//PіCБPμPrC<PrCfC%PëPN°
PIPsPіCБPsCf
    if((num < numofQuestions) && (num != 0)){
        num = num - 1;
        $('#slider').val(num + 1).slider('refresh');
    }
    else{
        alert("PŸT±CБP°T>C,P°CБ P¶PsT>");
    }
})

$('#slider').change(function(){//CfP»P°PN°PrPμCБ
num = $('#slider').val() - 1;
insertForm(num);
})

function simple_timer(sec, block, direction) {
    var time      = sec;
    direction     = direction || false;

    var hour      = parseInt(time / 3600);
    if ( hour < 1 ) hour = 0;
    time = parseInt(time - hour * 3600);
    if ( hour < 10 ) hour = '0'+hour;

    var minutes = parseInt(time / 60);
    if ( minutes < 1 ) minutes = 0;
    time = parseInt(time - minutes * 60);
    if ( minutes < 10 ) minutes = '0'+minutes;

    var seconds = time;
    if ( seconds < 10 ) seconds = '0'+seconds;

    block.innerHTML = "PђCŸT>C,P°P»CfT`P° T>P°P»T`P°PS
CfP°T>C<C,: " + hour+':'+minutes+':'+seconds;

    if ( direction ) {
        sec++;

        setTimeout(function(){ simple_timer(sec, block, direction);
}, 1000);
    } else {
        sec--;

        if ( sec > 0 ) {
            setTimeout(function(){ simple_timer(sec, block, direction);
}, 1000);
        } else {
            answerCountResult();
        }
    }
}
```

А қосымшасының соңы

```
}
}
}

function start_countdown() {
var block = document.getElementById('sample_countdown');
simple_timer(1800, block);
}

start_countdown();

//P' PSPëP¶PSPµPN° C‡P°CÍC, Pë PICÍPµ C„CíPSC†PëPë PrP»C¶
PSC, P»PSPIP° PSCëPëP±PSPe
$('#hrefIndex').click(function() {
var div = $('#hrefIndex').html();
var mess = confirm("PŸC-P·PrC-TJ PSY™C, PëP¶PµP»PµC¶C-TJC-P·
CÍP°T>C, P°P»PjP°PN°PrC< . P-P°P»T°P°CÍC, C<C¶T°C<TJC<P· PePµP»Pµ
PjP°?", "PќP°P·P°C¶ P°CíPrP°C¶C<TJC<P·!");
if(mess == true){
document.location.href = "index.html";
}
else{
$('#hi').remove();
$('#hrefIndex').prepend(div);
}
})

$('#hrefStudy').click(function() {
var div = $('#hrefStudy').html();
var mess = confirm("PŸC-P·PrC-TJ PSY™C, PëP¶PµP»PµC¶C-TJC-P·
CÍP°T>C, P°P»PjP°PN°PrC< . P-P°P»T°P°CÍC, C<C¶T°C<TJC<P· PePµP»Pµ
PjP°?", "PќP°P·P°C¶ P°CíPrP°C¶C<TJC<P·!");
if(mess == true){
document.location.href = "study_test.html";
}
else{
$('#hs').remove();
$('#hrefStudy').prepend(div);
}
})

$('#hrefResult').click(function() {
var div = $('#hrefResult').html();
var mess = confirm("PŸC-P·PrC-TJ PSY™C, PëP¶PµP»PµC¶C-TJC-P·
CÍP°T>C, P°P»PjP°PN°PrC< . P-P°P»T°P°CÍC, C<C¶T°C<TJC<P· PePµP»Pµ
PjP°?", "PќP°P·P°C¶ P°CíPrP°C¶C<TJC<P·!");
if(mess == true){
document.location.href = "result.html";
}
}
```

Қосымша Ә

Манифест құруға арналған скрипт

```
$(document).ready(function() {
    var predmet;

var question;

var correct;

var incorrect1;

var incorrect2;

var incorrect3;

var sourcequestion;
    var dataset;
    var numbfOfQuestions;
    var num = 0;

var id=1;

    var allQuestions = [];
    var questionArray = [];
    var correctArray = [];
    var incorrectArray1= [];
    var incorrectArray2 = [];
    var incorrectArray3 = [];

    $('#result').hide();

var db = window.openDatabase("TestStudy", "1.0", "TestStudy",
5*1024*1024);

db.transaction(populateDB, errorCB);

    function populateDB(tx) { //CÍPsP·PrP°PSPëPµ C,P°P±P»PëC†
        tx.executeSql('DROP TABLE IF EXISTS Questions');

tx.executeSql('CREATE TABLE IF NOT EXISTS Questions (id INTEGER
PRIMARY KEY AUTOINCREMENT, question TEXT, correct TEXT, incorrect1
TEXT, incorrect2 TEXT, incorrect3 TEXT, userAnswer TEXT)');

    }
}
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
function errorCallback(tx, err) {

alert("Error processing SQL: "+err);

}

db.transaction(function (tx) {

tx.executeSql('SELECT * FROM Study ORDER BY id DESC LIMIT 1', [],
function (tx, result) {
    var userSet = result.rows;
    for (var i = 0, itemD = null; i < userSet.length;
i++) {
        itemD = userSet.item(i);
        predmet = itemD['predmet'];

        if(predmet == "РәпсРјРіСЪСҢС,РµСЪР»С-Рё Р¶РµР»С-
Р»РµСЪ") {
            var id = 1;

$.getJSON("http://localhost:45100/home/GetData?pred=" + id +
"&callback=?", function (data) {

db.transaction(function (tx) {

for(var i=0; i <= data.length; i++){

if(i == data.length){

tx.executeSql('SELECT * FROM Questions', [], function (tx, result)
{
    numbOfQuestions = result.rows.length;
    $('#slider').attr('max',
numbOfQuestions);

    showRecords();
})
})
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
else{
    var jsonString = JSON.stringify(data[i]);

var jsonParse = JSON.parse(jsonString);

question = jsonParse.Question;

correct = jsonParse.Correct;

incorrect1 = jsonParse.False1;

incorrect2 = jsonParse.False2;

incorrect3 = jsonParse.False3;
    tx.executeSql('INSERT INTO Questions
(question, correct, incorrect1, incorrect2, incorrect3) VALUES ('
+ question + ', ' + correct + ', ' + incorrect1 + ', ' +
incorrect2 + ', ' + incorrect3 + ')');
    }
    })
    })
}

else if(predmet == "Тәбиғат ғылымы") {
    var id = 2;

$.getJSON("http://localhost:45100/home/GetData?pred=" + id +
"&callback=?", function (data) {
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
db.transaction(function (tx) {

for(var i=0; i <= data.length; i++){

if(i == data.length){

tx.executeSql('SELECT * FROM Questions', [], function (tx, result)
{
    numbOfQuestions = result.rows.length;
    $('#slider').attr('max',
numbOfQuestions);
    showRecords();
})

}

else{
    var jsonString = JSON.stringify(data[i]);

var jsonParse = JSON.parse(jsonString);

question = jsonParse.Quest;

correct = jsonParse.Correct;

incorrect1 = jsonParse.False1;
```


Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
incorrect2 = jsonParse.False2;

incorrect3 = jsonParse.False3;
        tx.executeSql('INSERT INTO Questions
(question, correct, incorrect1, incorrect2, incorrect3) VALUES ("'
+ question + '", "' + correct + '", "' + incorrect1 + '", "' +
incorrect2 + '", "' + incorrect3 + '"')');
        }
    })
})
    }
})
    })
```

```
function showRecords()
{

db.transaction(function (tx) {

tx.executeSql('SELECT * FROM Questions', [], function (tx, result)
{

dataset = result.rows;
        var ranQuest = [];

for (var i = 0, item = null; i <= numbfOfQuestions; i++) {

var rand = Math.round((Math.random()*4)+1|0);
        if( i == numbfOfQuestions){
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
$('#slider').val(1).slider('refresh');
    }
    else{
        item = dataset.item(i);
        questionArray.push(item['question']);
        correctArray.push(item['correct']);
        incorrectArray1.push(item['incorrect1']);
        incorrectArray2.push(item['incorrect2']);
        incorrectArray3.push(item['incorrect3']);

        if((rand == 1) && (rand !=
numbOfQuestions)){
            ranQuest.push(questionArray[i],
correctArray[i], incorrectArray1[i], incorrectArray2[i],
incorrectArray3[i]);
            allQuestions.push(ranQuest);
            ranQuest = [];
        }
        else if((rand == 2) && (rand !=
numbOfQuestions)){
            ranQuest.push(questionArray[i],
incorrectArray1[i], correctArray[i], incorrectArray2[i],
incorrectArray3[i]);
            allQuestions.push(ranQuest);
            ranQuest = [];
        }
        else if((rand == 3) && (rand !=
numbOfQuestions)){
            ranQuest.push(questionArray[i],
incorrectArray1[i], incorrectArray2[i], correctArray[i],
incorrectArray3[i]);
            allQuestions.push(ranQuest);
            ranQuest = [];
        }
        else if((rand == 4) && (rand !=
numbOfQuestions)){
            ranQuest.push(questionArray[i],
incorrectArray1[i], incorrectArray2[i], incorrectArray3[i],
correctArray[i]);
            allQuestions.push(ranQuest);
            ranQuest = [];
        }
    }
}
})
})
}

function insertForm(IdNum){
    var item1 = allQuestions[IdNum][0];
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
var item2 = allQuestions[IdNum][1];
var item3 = allQuestions[IdNum][2];
var item4 = allQuestions[IdNum][3];
var item5 = allQuestions[IdNum][4];
$('#textarea-1').val(item1);
var answer1 = $('label[for=radio-choice-v-6a]').text(item2);
var answer2 = $('label[for=radio-choice-v-6b]').text(item3);
var answer3 = $('label[for=radio-choice-v-6c]').text(item4);
var answer4 = $('label[for=radio-choice-v-6d]').text(item5);

var answer1 = $('label[for=radio-choice-v-6a]').html();
var answer2 = $('label[for=radio-choice-v-6b]').html();
var answer3 = $('label[for=radio-choice-v-6c]').html();
var answer4 = $('label[for=radio-choice-v-6d]').html();

if(answer1 == correctArray[IdNum]){
    $('input[type=radio]').prop('disabled', false).checkboxradio('refresh');
    $('input[type=radio]').prop('checked', false).checkboxradio('refresh');
    $('#radio-choice-v-6a').prop('checked', true).checkboxradio('refresh');
    $('#radio-choice-v-6b').prop('disabled', true).checkboxradio('refresh');
    $('#radio-choice-v-6c').prop('disabled', true).checkboxradio('refresh');
    $('#radio-choice-v-6d').prop('disabled', true).checkboxradio('refresh');
}

else if(answer2 == correctArray[IdNum]){
    $('input[type=radio]').prop('disabled', false).checkboxradio('refresh');
    $('input[type=radio]').prop('checked', false).checkboxradio('refresh');
    $('#radio-choice-v-6b').prop('checked', true).checkboxradio('refresh');
    $('#radio-choice-v-6a').prop('disabled', true).checkboxradio('refresh');
    $('#radio-choice-v-6c').prop('disabled', true).checkboxradio('refresh');
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
    $('#radio-choice-v-6d').prop('disabled',
true).checkboxradio('refresh');
    }

    else if(answer3 == correctArray[IdNum]){
        $('#input[type=radio]').prop('disabled',
false).checkboxradio('refresh');
        $('#input[type=radio]').prop('checked',
false).checkboxradio('refresh');
        $('#radio-choice-v-6c').prop('checked',
true).checkboxradio('refresh');
        $('#radio-choice-v-6a').prop('disabled',
true).checkboxradio('refresh');
        $('#radio-choice-v-6b').prop('disabled',
true).checkboxradio('refresh');
        $('#radio-choice-v-6d').prop('disabled',
true).checkboxradio('refresh');
    }

    else if(answer4 == correctArray[IdNum]){
        $('#input[type=radio]').prop('disabled',
false).checkboxradio('refresh');
        $('#input[type=radio]').prop('checked',
false).checkboxradio('refresh');
        $('#radio-choice-v-6d').prop('checked',
true).checkboxradio('refresh');
        $('#radio-choice-v-6a').prop('disabled',
true).checkboxradio('refresh');
        $('#radio-choice-v-6b').prop('disabled',
true).checkboxradio('refresh');
        $('#radio-choice-v-6c').prop('disabled',
true).checkboxradio('refresh');
    }
}

$('#next').click(function(){

if((num < numbOfQuestions - 1)){

num = num + 1;
    $('#slider').val(num + 1).slider('refresh');
    }
    else{
        alert("ПҰТІСБЕРТ>C, P°СБ P¶PsT>");
    }
})

$('#prev').click(function(){
if((num < numbOfQuestions) && (num != 0)){
    num = num - 1;
```

Ә ҚОСЫМШАСЫНЫҢ СОҢЫ

```
$('#slider').val(num + 1).slider('refresh');
    }
    else{
        alert("PŸT±CБP°T>C,P°CБ P¶PsT>");
    }
})

$('#slider').change(function(){
    num = $('#slider').val() - 1;
    insertForm(num);
})
})
```