

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

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

(қолы)

« _____ » _____ 2014 ж.

ДИПЛОМДЫҚ ЖОБА

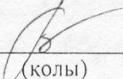
Тақырыбы: «Android операциялық жүйесі негізінде саяхатшыларға арналған “КСЕТ” мобильдік бағдарламасын құрастыру»
5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша

Орындаған Тулубаев Аяз Берикович тобы: ВТк-10-2

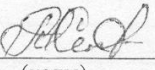
Жетекші аға оқытушы Рахимжанова З.М.

Кеңесшілер :

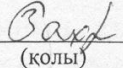
Экономикалық бөлім бойынша :


(қолы) доцент Боканова Г.Ш.
« 29 » 05 2014ж.

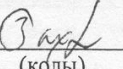
Өмір тіршілігі қауіпсіздігі бойынша:


(қолы) т.ғ.к., аға оқытушы Муташева Г.С.
« 20 » 05 2014 ж.

Есептеу техникасын қолдану бойынша :


(қолы) аға оқытушы Рахимжанова З.М.
« 23 » 05 2014 ж.

Мөлшер бақылаушы:


(қолы) аға оқытушы Рахимжанова З.М.
« 26 » 05 2014 ж.

Пікір жазушы :

ҚазҰТУ, РЭЖТ кафедрасының аға оқытушысы: Усембаева С.Г.

« _____ » _____ 2014 ж.
(қолы)

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

Тулубаев Аяз Берикович

Жоба тақырыбы: Android операциялық жүйесі негізінде саяхатшыларға арналған “КСЕТ” мобильдік бағдарламасын құрастыру.
ректордың «24» қыркүйек №115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «29» маусым 2014 ж.

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

Android операциялық жүйесіне негізделген бағдарламаны құрастыру. Бұл бағдарлама саяхатшыларға, саяхатқа шығу кезінде ақпараттық қолдануға және кезес беруге арналған.

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

- ақпараттық жүйені құруға қажетті техникалық бағдарламалық құралдар;
- дүние жүзілік нарықтағы аналогты өнімді зерттеу;
- қолданушы интерфейсін, қосымшаны құрастыру;
- техникалық-экономикалық негіздеуі;
- жұмыс орнындағы тиімділік мақсатты;


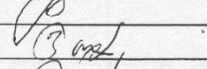
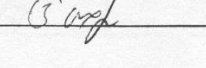

Сызба материалдарынын (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

- Жасаған бағдарламалық қосымшаның жұмыс істеу қабілеттілігі мен функционалда көрсетілген графикалық материал;
- жобаламадан қосымшаның сызбасы.

Негізгі ұсынылатын әдебиеттер:

1. Голощапов А. Google Android программирование для мобильных устройств. Санк-Петербург 2011-438 с.
2. Брайан Х.,Филлипс Б. Программирование под Android. Питер 2014-592 с.
3. Амелин К.С., Граничин О.Н., Кияев В.И, Корякко А.В. Введение в разработку приложений для мобильных платформ. Санкт-Петербург 2011-505 с.
4. Лорен Д., Кондор Ш., Android программирование приложений под операционную систему Google 2011 – 464 с.
5. Бурнет Э., Привет, Андроид! Разработка мобильных приложений. Питер, 2012 – 256 с.

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Рахимжанова З.М.	20.05.2014	
Тіршілік қауіпсіздігі	Муташева Г.С.	15.05.2014	
Экономикалық бөлім	Боканова Г.Ш.	20.05.2014	
Норма бақылаушы	Рахимжанова З.М.	28.05.2014	
Есептеу техникасын қолдану	Рахимжанова З.М.	23.05.2014	

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Негізгі бөлім	9.09.2013	
1.1	Android операциялық жүйесіндегі бағдарлама жасау өзектілігі.	16.09.2013 23.09.2013 30.09.2013	
1.2	Мақсаттама талайындау және негізгі мүлкіліктер жобалау	1.10.2013 8.10.2013	
1.3	Бағдарламаның талақ архитектурасы	15.10.2013	
2	Арнаулы бөлім	25.10.2013	
2.1	Зерттеудің нәтиже облысының анализі	28.10.2013	
2.2	Google maps навигаторға Android API v2 бағдарлама сәйкесіне	10.12.2013	
2.3	Деректер моделінің анықтамасы	24.12.2013	
3	Техникалық бөлім		
3.1	Бағдарлама архитектурасы	11.03.2014	
3.2	Жұданбама интерфейсін зерттеуі	18.03.2014	
4	Тіршілік қауіпсіздігі	8.04.2014	
4.2	Есептеу бөлімі	15.04.2014	
4.3	Тіршілік қауіпсіздігі бойынша қорытқыда.	22.04.2014	
5	Техникалық экономикалық негіздері	1.04.2014	
5.5	Өзіндік құн нәтижесінің кестесі	14.04.2014	

Тапсырманың берілген уақыты « 10 » қыркүйек 2014 ж.

Кафедра меңгерушісі _____
(қолы)

ф.-м.ғ.д., проф. Құралбаев З.Қ

Жоба жетекшісі _____
(қолы)

аға оқытушы Рахимжанова З.М.

Орындалаты тапсырманы қабылдаған студент _____
(қолы)

Тулубаев Аяз Берикович

Андатпа

Бұл дипломдық жобада саяхатшыларға арналған «КСЕТ» мобильдік бағдарламасы жүзеге асырылды. Дипломдық жобада қолданбаның серверлік және навигациялық 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.

Аннотация

В данном дипломном проекте разработано приложение для туристов «КСЕТ». В работе предусмотрены требования к серверной и навигационной 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.

Annotation

In the given Thesis Project tourists oriented application «КСЕТ» was developed. The paper involves requirements for the server-based and navigation 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

Кіріспе	8
1 Негізгі бөлім	9
1.1 Android операциялық жүйесіндегі бағдарлама жасау өзгектілігі.....	9
1.1.1 Android операциялық жүйесінің платформасы.....	10
1.1.2 Мобильдік бағдарламаның нарықтық анализі.....	12
1.2 Мақсатты тағайындау және негізгі мүмкіндіктер жобалау.....	13
1.3 Бағдарламаның толық архетиктурасы.....	14
1.4 Сервер бөлімінің бағдарламаға талаптары.....	14
1.4.1 Операциялық жүйе.....	17
1.4.1.1 Қазіргі операциялық жүйесіндегі желінің шолуы	18
1.4.1.2 Талаптарға байланысты операциялық жүйені таңдау	21
1.4.2 Дерект қорын басқару жүйесі.....	21
1.4.3 Android бағдарламасының жалпы жұмыс істеу сұлбасы.....	26
1.4.4 Java құрылым және өзгешеліктері.....	28
1.4.5. XML тілі ерекшеліктері.....	31
2 Арнаулы бөлім	35
2.1 Зерттелетін жүйе облысының анализі.....	35
2.2 Google maps навигаторы Android API v2 бағдарламасынына	38
2.3 Деректер моделінің анықтамасы.....	41
2.3.1 Даталогикалық проектилеу.....	42
2.3.2 Физикалық проектилеу	43
3 Техникалық бөлім	45
3.1 Бағдарлама архитектура	45
3.2 Қолданбалы интерфейстің зерттемесі.....	46
3.2.1 Навигация.....	46
3.2.2 Аудармашы беті	47
3.2.3 Саяхатшыларға көмек беру беті.....	50
3.2.4 Google Maps саяхатшының орнын анықтау	52
3.2.5 Саяхатшының қажетті мәліметін сақтау беті	52
4 Тіршілік қауіпсіздігі.....	54
4.1 Еңбек шартынның анализы	54
4.2 Есептеу бөлімі	55

4.2.1 Кондиционерлеу жүйесін есептеу.....	55
4.2.2 Табиғи жарықты есептеу.....	59
4.3. Тіршілік қауіпсіздігі бөлімі бойынша қорытынды.....	62
5 Техникалық экономикалық негізделуі.....	63
5.1 Жоба сипаттамасы	63
5.2 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі.....	64
5.3 Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі	66
5.4 Бағдарламаның шығындар статьясы	67
5.5 Өзіндік құн нәтижесінің кестесі	69
5.6 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу	72
5.7 Қолдану аясындағы жылдық бірмезгілдегі шығындарды есептеу	73
5.8 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	73
5.9 Пайда мен шығындарды есептеу	74
5.10 Экономикалық тиімділікті есептеу	75
Қорытынды	78
Қолданылған әдебиеттер тізімі.....	79
Бағдарламасы А.....	80

Кіріспе

Бүгінгі таңда Android мобильдік бағдарлама әзірлеу, қарқынды дамып келе жатқан бағдарламалардың бірі болып танылады. Android операциялық жүйесі көптеген смартфондарда орнатылған, ондай смартфондарға: Samsung, Motorola, Sony, LG, Lenovo және т.б. Қазіргі заманда өзекті мәселелердің бірі ғаламтор-бағдарламалары тез дамып жатыр. Соған байланысты ғаламтор екпінді көтеріле бастады және пайдаланушылардың сұранысы бойынша, бағдарламалар тұрақты жаңартылып отыр. Платформалардың арасында жетекші орындары Android және IOS платформалары алады.

Дипломдық жұмыстың мақсаты - Android операциялық жүйесі негізінде саяхатшыларға арналған “КСЕТ” мобильдік бағдарлама құрастыру. Жасалып отырған жоба саяхатшыларға арналған. Бағдарлама көнегімен саяхатшылар өзінің уақытын үнемдейді және аудармашыны қажет етпейді. Зерттеулер бойынша, таңдаған тематикамаға байланысты Қазақстанда бағдарлама жасалмаған және ешқандай бәсекелістік жоқ. Сол себепті ең басты

қиыншылықтардың бірі көп қолданушыларды табу және интернет желісіне сенімге кіру.

Android – операциялық жүйесі коммуникаторларға, планшеттарға, цифрлық ойнатқыштарға, цифрлық фоторамкаларға, қолға киетін сағаттарға, нетбук және смартбуктарға, Linux ядросында жасалған операциялық жүйе орнатылған. Бағдарламаларды өңдеудегі негізгі тілдердің бірі Java, алайда C++ тілінде бағдарламаларды өңдеу кітапханалары бар. Бағдарламаны жасау үшін Eclipse ортасын қолданамыз, AndroidSDK мобильдік құрылысы SDK эмуляторы арқылы іске асады. Java объектісіне негізделген тіл, оның өзінің көптеген ішкі кітапханалары бар. Бірінші орында ол қолданушыға жеңіл әрекеттестікті көрсетеді.

Жасалатын бағдарлама Андроидтың 2.2 версияларынан бастап 4.4 версияларында жұмыс істейді. Бағдарламаны жазу кезінде басты характеристикалардың бірі сервисті процесстерді фондық режимінде қолдану болып табылады. Қойылған мақсатымызға жету үшін келесі талаптарды орындауымыз қажет:

- веб-бағдарламаның компоненттік сәулетін құрастыру, өмірлік үрдістің қалыптасуын құрастыру;
- веб-бағдарламаларды жобалау және жүзеге асыру әдістемесін зерттеу;
- жобаның талдауын зерттеу әдістерін және ыңғайын қарастыру.

1 Негізгі бөлім

Қарастырылатын бағдарлама саяхатшыларға арналған. Дипломдық жұмыстың басты тапсырмасы Android операциялық жүйесінде саяхатшыларға арналған бағдарлама жасау, бағдарламаның негізгі функциялары: сөздерді аудару, оны дауыстап айту, қолданушыға туризмге шығу кезінде нұсқау беру, қолданушының GPS арқылы орнын анықтау және өзіне керекті мағлұматтарды жазбаға сақтауға мүмкіндік береді.

1.1 Android операциялық жүйесіндегі бағдарлама жасау өзгектілігі

Android, негізінде бірегей, өзгеше операциялық жүйе. Нәтижелерді жақсы көрсету үшін бағдарлама әзірлеуші операциялық жүйенің өзектілігін және өзгешілігін толығымен көрсетіп өткен. Бағдарлама әзірлеу кезінде, бірнеше қиындықтарды ескеру қажет. Оларға тоқталып өтейік:

- бағдарламаны қондыру кезінде, екі есе немесе төрт есе орын алады, яғни ол бағдарламаның түпнұсқалық орнын қажет етеді;
- орнатылған флеш-картамен жұмыс істеу кезінде файылдың жылдамдығы он есе төмендейді, егер бос орын аз болса;
- әр процесс кезінде жедел жадыдан 16Мб (кейде 24 Мб) қолданылады.

Android, Linux операциялық жүйесі негізінде жасалған. Бағдарламаның және ядроның арасында API қабаты және нативті кодтағы кітапхана орналасқан. Бағдарлама виртуалды машина Java (Dalvik Virtual Machine) орындалады.

Android -та көптеген бағдарламаларды біруақытта ашу мүмкіндігі бар. Брак, бір бағдарлама толығымен бүкіл экранды алады. Ағымдағы бағдарламаны қолдана отырып басқа бағдарламаны қосуға немесе жаңасын ашуға болады. Оның тарихын көруге болатын браузерге ұқсайды.

Қолданушының интерфейсі әр экранда Activity класс кодындағы көріністі көрсетеді. Процессте әр түрлі Activity тұтынады немесе қолданылады. Процесстің көбі Activity көбірек өмір сүреді, яғни, ол дегеніміз Activity қолдануы көбірек. Activity жұмыс істеу кезінде тоқтатылып және жаңадан қосылып бүкіл қажетті информацияны сақтап қосылады.

Android арнаулы механизмді қолданып әрекеттің сипаттамалары бас Intent негізде жүргізіледі. Белгілі әрекеттер істелген кезде (қоңырау соғу, хабарлама жіберу, терезені көрсету), Intent шақырылады.

Тағы да, Android демоға ұқсас Linux серверларын ұстайды, олардың атқару қызметі керекті әрекетті фондық режимде (мысалы, өленнің ойналуы). Бағдарламада деректерді айырбастау үшін Content providers (провайдері қолданылады).

Айтылмыш жұмыс үшін деректер провайдерін пайдаланып, қолданушының құрылғысына орнату керек [1].

1.1.1 Android операция жүйесінің платформасы

Android Linux жүйесіндегі ең бірінші мобильдік жүйесі жасалған платформа және ол Люникс және бағдарлама жасаушылардың ішінде, көптеген қолданушылардың арасында мобильдік бағдарламаның атағын көтерген. Себебі, Linux файлдық қауіпсіздіктің саясат концепциясы кез-келген жүйе файлы үш категория қолданушылардан тұрады деп пайымдайды: файлдың өз иесі немесе жасаушысы, қандай болса да қолданушылар тобы, оған көп жағдайда файл иесі және тағы басқалар кіреді. Осылайша бәрі рұқсат етілген қолданушы немесе файл иесі, тек ол ғана қол жеткізу құқығын өзгерту мүмкіндігіне ие болғандықтан, файлдық жүйе қауіпсіздігінің саясатын құрып, файл иесі үшін жеке құқығын анықтайды, қолданушы топтар үшін де, қалған жүйе қолданушылар үшін де.

Заңды түрде Android платформасы 2003 жылдың 23 қыркүйегінде құрастырылды. Ең бірінші версиясы операциялық жүйенің индексімен 1.0 Apple Pie атымен шықты. Android операциялық жүйесінің ядросы Linux-та жасалған және ол өзін ашық мобильдік платформа ретінде көрсетті.

Көптеген гаджет бағдарламасын жасаушылар «Жасыл роботты» тегін қолдана алады және жаңа Android операциялық жүйесін. Көлемді түрде бағдарламаны тарату үшін, Google компаниясы Open Handset Alliance организациясын негіздейді. Бұл организацияға тез арада көптеген үлкен дүниежүзілік смартфон өндірушілер қосылады, оларға жиынтықталады, тағыда мобильдік операторлар қосылады.

Үш айдан кейінгі Android 1.0 презентациясы, дүниежүзілік нарыққа бірінші рет смартфондық операциялық жүйені енгізді. Құрылғының аты –HTC Dream (T-mobile G1). Осыған дейін аппарат қомақты техникалық характеристикасын көрсетті. Оларға тоқталып өтсек: дисплейдің рұқсаты 3,2 дюйм, тактілік жиілігі 525 МГц процессор, жедел жадтың көлемі 192 мегабайт және камера матрицасының рұқсаты 3,2 МП.

Логотипқа қатысты жасалу тарихы өтте қызықты. Бағдарламаның логотипін құрушы 2007 жылы ұзақ уақыт бойы лайық бейнесін таптай жүрді, көптеген нұсқаларды қарастырды: фильмдерден, ойыншық дүкендерінен және интернет желісінен. Нәтижесіне орай, прототип ретінде туалет кабинкасындағы отырған адам алынған. Қарапайым көрініс басындағы екі антенасымен. Бұл мобильдік платформаны көрсетеді, ол ашық және жеңіл болуы қажет.

Бір жылдың ішінде жаңа платформаны үш рет жаңартты. Бірінші рет - Android 1.1, кодталған аты Banana Bread, екінші - Android 1.5, кодталған аты Cupcake және үшінші - Android 1.6 кодталған аты Donut. Бұл жаңартулар құрылғыда көптеген көлемді жаңалықтарды ашты. Оларға тоқталып өтсек: экранның биік рұқсатын қолдайды, анимациялық интерфейс, мультитілді дауыс іздеуші, видео жазушы және тағы басқа. 2008 жылы қазан айында онлайн-дүкен ашылған, бұл дүкеннен Андроид операциялық жүйесіне актуалды бағдарламаны жазып алса болады. Оның аты Android Market. Дүкеннің табысына байланысты 70% пайдасы бағдарлама жасаушыларға беріледі, ал

мобильдік байланысшыларға 30% бөлінеді. Ең бірінші онлайн-дүкенін Sony Ericson компаниясы қосты. Бағдарлама жазып алу саны 2011 жылға байланысты 1 млрд-қа жетті. Кейінірек бүкіл версиялары қосылды («Android Market» «Кітапхана» және «Өлендер») оның аты Google Play. Ол 190 мемлекетте таратылған бүкіл дүние жүзі бойынша оның ішінде 700 мың бағдарламма бар. Кезекті бағдарламаның жаңартылуы 2.0 версиясына жетті, ол операциялық жүйені өте көркем етіп көрсетеді, дүкеннің ішінде бағдарламаны қызықты етіп көрсетеді. Келесі версиясы 2.1 өзінде “тәтті” код атын көрсетті. Жаңартулардан кейін бағдарлама HTML 5 және бірнеше Google аккаунтын қолдануға рұқсат берді. Бірінші версияның қарлығашы: HTC Magic және HTC Hero, Motorola Droid және Samsung Galaxy. 2010 жыл үлкен жарыс жылы болды, ол үлкен гигагерцті мобильдік процессорларды жасай бастады. Осы уақытта бірінші рет Google смартфонны ойлап табылды, осы бағдарлама жайлы талқылаулар көп жүрді. Google Nexus One - ақылды телефон 1ГГц процессорымен. Бірінші жеткізу және соңғы жеткізуші HTC компаниясы болды. Кейінірек осыған ұқсас процесстерді Motorola және Samsung аппараттарына қолданды. 2010 жылда <<Жасыл робот>> 2,2 индексіні алды, ал кодталған аты – Froyo. Бастапқы қабылдайтын заттары: Adobe Flash қолдайды, Jit компиляциясын қолдайды, ал ол жұмыс істеуді тезірек арттырады, тағы да бағдарламаның өсуіне әкелді. Бұл жаңарту 1ГГц қуатты процессорға ақпарат орналастырды. 2010 жылдың соңында Android 2.3 версиясы қуып жетті, оның кодты аты баяғыдай - Gingerbread. Бұл версия төрт жыл бойы дүкенде бірінші орынды алып тұрды. Оның басты ерекшеліктері басқа версиялардан: HD рұқсатын қолдайды, жаңа файылдық системаны функциясын көрсетеді, интерфейсі өте көркем және көптеген функциялары қосылды. Осыған орай тағы да жаңа флагман шықты ол Google - Nexus S. Келесі брендты смартфонды Google бағдарламаны Gingerbread компаниясымен ойлап тапты. Осы жолы партнер ретінде Samsung компаниясы болды, ал Nexus S смартфонны Galaxy S смартфонна жарады, көрсету бойынша видеоөзгерістер Google Nexus S кіргізілді. Сол жылдары дүкенге LG екі ядролы смартфонды Optimus 2x атымен шығарды. Кейінірек екі ядролы чиппен Samsung Galaxy S II, Motorola Droid X2, HTC Sensation қамтамасыз етілді. Galaxy S смартфоннан кейін компания ғибадатты құрылғы шығарды. Ол Galaxy Tab-қа қарағанда өте танымал болды. 2011 жылдың басында Android көркем, қызықты болды және Google компаниясы планшеттарға арнап Android 3.0 Honeycomb атты версиясын шығарды. Смартфонның интерфейсі Gingerbread, кішкене созылған болып, ал Honeycomb қалып отырды. Ол компанияға аса табыс алып келмеді. Планшеттерге арнап көптеп бағдарламаларды қазір де шығарады, табысы аз болса да. Шынайы, күшті әсерді Samsung компаниясы өзінің көрінісінде IFA 2011 ұсынды. Ол Galaxy Note-ке қарап жобаланды. Бірінші 5 дюймдық смартфон, Корей компаниясы шығарған форм-фактор смартфон, өзінің компаниясына үлкен табыс алып келді. 2011 жылғы ең жақсы соңғы жаңалықтардың бірі, бұл күмәнсіз, екі Android версияның қосылуы болды: планшет пен смартфонның. Екі класс та құрылғының жаңартылған 4.0 Ice Cream Sandwich версиясында

жақсы жұмыс істеді. Бұл жүйе PC Magazine басылымның сыйлығын жеңіп алды, осы платформа көптеген өзгерістер алып келіп смартфонды толығымен өзгертті. Тағы да айта кететін болсақ, 2012 жылы осы операциялық жүйе, жылдық көріністе бірінші орынға ие болды, ең жақсы платформа ретінде. Тағы бір ерекшелігі және жаңалығы, осы жылғы ICS екі архитектуралы процессорды қолданды Intel x86 және MIPS, олар ARM қосылған болып келді[1].

Android 4.0 бағдарламасын жасау уақытында, Google жұмыс берушісі бұл жасап жатқан платформаны бренд ретінде жарялады. Ол Galaxy Nexus моделі болып танылды. Атын әйгілі етіп тұрған бағдарлама жасаушы Samsung компаниясы болды. 2012 жылдың басындағы жаңалықтың бірі, бұл NVIDIA компаниясының төрт ядролы ARM процессоры Tegra 3 атымен әйгілі болды. Екіншіден Qualcomm компаниясы кейірек осыған ұқсан чипсетті шығарды. Тағы да осы жылдың басында екі смартфон шығады деп жарияланды, ол HTC One X және Optimus 4X HD. Бірақ, екі смартфонның атағын Samsung компаниясының Galaxy S III аты смартфонды алды. Планшет дүкенге байланысты айта кетерлік жай, ең көп сатылған планшеттердің бірі болып ASUS компаниясының Nexus 7 атты моделі танылды. Оның бағасы 200 доллар және планшет жаңартылған Android 4,1 операциялық жүйесімен. Project Butter технологиясы Google компаниясының тоқтап қалатын операциялық жүйесінен құтылып, өзінің жаңа операциялық жүйесін іске асырды. Бес жылдан кейін, яғни бұл 2012 жылдың соңына қарай Google компаниясы тағыда екі брендтық девайстарды шығарды, ол планшет Samsung Nexus 10 және смартфон LG Nexus 4. Айта кететін жай 2013 жылғы смартфондар үлкен дисплейға және өте тез жұмыс атқаратын процессорға ие болды.

1.1.2 Мобильдік бағдарламаның нарықтық анализі

Қазір, Қазақстанның мобильдік нарығында, мобильдік бағдарлама жасау өте тез өзгеруде және осыған баға бере отырып, динамикалық тұрғыда мобильдік бағдарлама жасау арту процессінде. Бағдарлама сатылу және жасау өсіп жатыр. 2012 жыл периодында Қазақстанда 200 астам мобильдік бағдарлама жасау және оны қамтамасыз ету компаниялары тіркелген. Басты веб-бағдарламашылардың артуының себебі, ол интернетпен қолданушылардың саны өсіп жатыр, соған орай бағдарлама жасаушылардың саны өсіп жатыр. Қазір интернет желісінде көптеген келісімдер жүргізіледі, тағы да желіде электронды валюталар бар (Яндекс.Деньги, Webmoney және т.б), банктің шотымен оңай түрде қолданушыға жұмыс істеуге мүмкіндік береді. Күнделікті миллиондаған қолданушылар іздеу жүйесін қолданады, электродны кітаптарды, блогтарды, жаңалықтарды және тағы басқа. Мобильдік бағдарламалар, ақпараттық технологиялар саласында даму, арту трендісіне ие болды. Егер 2008 жылы, App store ашылу кезінде, мобильдік нарық енді даму барысында болса, ал қазір нарық активті түрде дамып жатыр. Мобильдік бағдарлама – бұл тек мобильдік құрылғыға арналған өнім. Телефондық бағдарламалар арнайы телефондарға арналған дүкендер арқылы таратылады.

Олар App Store, Google Play, Windows Phone Store и BlackBerry App World және тағы басқа. Мобильдік бағдарламалар көптеген қолданбалы жағдайды шешеді: мобильді картографияны және электронды поштаны қабылдауға оның ішінде жеке функцияларды шешеді. Олар қолданушының өмірлік циклін жеңілдетуге арналған. Мобильдік бағдарлама нарығында әр облыста өзінің ерекшеліктері бар, мысалы ақша төлеу және операциялық жүйе айырмашылығында. J'son & Partners Consulting компанияларының деректері бойынша, 2013 жылы, мобильдік бағдарлама жасау нарықта дүние жүзі бойынша \$7,83 млрд құрайды. Осыған орай 2016 жылға дейін, ол \$65,79 млрд құрайды деп қарастырылып отыр. Мобильдік нарықтың өсуіне қатысты келесі мағлұматтарды айта кетсе болады, олар әлеуметтік желі, жаңа мобильдік платформалар шығуы, Windows 8, планшетті компьютерлердің өсуі, мобильді төлеулердің артуы, мобильді банкингтер және тағы басқа [1].

Анализ бойынша нарықта ең басты дамып және көтеріліп келе жатқан дүкен ретінде Apps Store және Google Play Market саналады. Бұл дүкендегі бір бағдарламаның орташа құны \$7 iPad-тарға құрайды, ал телефонға \$3,5 доллар. Осыған орай бағдарлама жасаушылардың бизнесі аудиториядағы бағдарламаны сатып алушыларға байланысты [7].

1.2 Мақсатты тағайындау және негізгі мүмкіндіктерді жобалау

Бағдарлама «КСЕТ» саяхатшыларға арналған және саяхатшылардың жағдайын жеңілдетеді, сөйлесіге мүмкіндіктер береді.

Саяхатшы бағдарламасын жобалаудың негізгісі болып келесі жағдайларды қарастырамыз:

- жүйе жеңіл және интерфейсін ыңғайлы;
- қолданушыға әр түрлі сөздерді аударуға мүмкіндік береді, аударлатын тілдер: ағылшын, неміс, француз және итальян тілдері;
- қолданушыға GPRS арқылы тұрған жерін көрсетеді;
- қолданушыға туризмге қатысты бірнеше нұсқауларды көрсетеді;
- қолданушы өзінің қойылған мақсаттарын, бағдарламаға сақтап қоя алады.

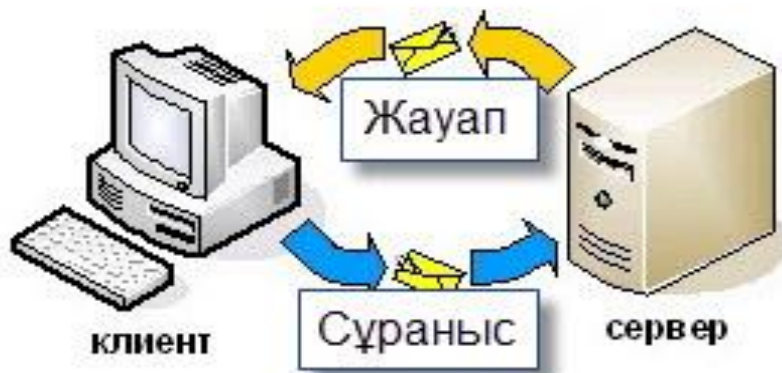
Жасалып отырған жобамыз Play Market нүктесіне потенциалды кіруі мүмкін. Ол үшін жасалып отырған жобамыз бүкіл технологиялық және техникалық мағлұматтарға ие.

1.3 Бағдарламаның толық архитектурасы

Келесі кезеңде мағлұматтарды кезекті орындап, оны итерациялық тілге негіздеуді жүзеге асырамыз. Оның ішінде тұрақты қиындату және бастапқы құрылымы детальді қарастырылады.

Негізгі жасалатын бағдарламаның архитектуралық жүйесі «қолданушы-сервер» арасында орындалады. Қарастырылатын мағлұматтың бірі желінің жеңілділігі және тез арада қолдануы. Сервер деп отырғанымыз желідегі

сақталған деректерімізге қол жеткізу. Клиентпен сапалы қарым-қатынас жасау үшін интернетті қолданамыз.



Сурет 1.1 – Бағдарламаның толық архетиктурасы

Архитектураның негізгі сипаттамасы және клиент-сервер өзгешеліктері:

- Көп жағдайдағы мүмкіндіктер, есептеуіш жүйесінің функцияларның, дербес компьютерлермен арасындағы желілер арқылы байланысын толық қарастырады. Осы мүмкіндіктер қолданушыға және бағдарлама жасаушыға үлкен жеңілдіктер береді. Егер бағдарлама толығымен жұмыс істемеген жағдайда, бағдарлама жасаушы серверге кірмей-ақ бағдарламаны толығымен өзгерте алады, сонда клиентке жаңадан бағдарламаны жаңарту туралы хабар немесе мағлұмат келеді.

- Толығымен бүкіл деректер серверде сақталады, осылай деректеріміз толығымен қорғалады және тез қол жетімді болады. Серверде деректерді қадағалауды өте жеңілдетеді.

- Әр түрлі клиенттерді біріктіруге рұқсат береді. Бір сервердегі деректерді әр түрлі платформадағы клиенттер, бір уақытта қолдана алады және әр түрлі операция жүйелерімен жұмыс істей алады [2].

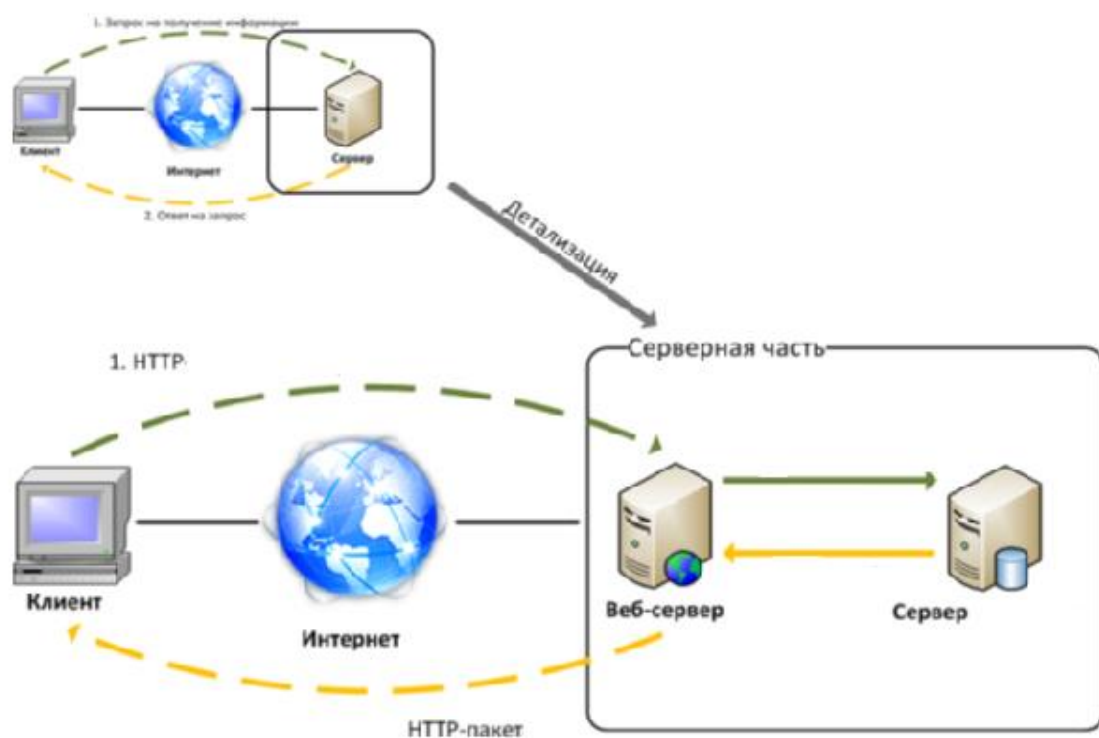
1.4 Сервер бөлімінің бағдарламаға талаптары

Басты мақсатымыз интернет желісі арқылы орындалатын болғандықтан, бізге интернет жайлы қысқаша мағлұматтар беріп өтуіміз қажет. Интернет арқылы бағдарламаны серверге қосуымыз қажет. Сонда HTTP- сұранысы арқылы протоколымыз беріледі және жүзеге асады. Мобильдік бағдарламаның өзінің IP адресі болады, сол арқылы платформадағы операциялық жүйенің қандай екенін білуге болады.

Қарастырылып отырған бағдарламамыз көптеген санды операцияларды оқиды. 1.2-сурет процесстің бастапқы архетиктурасын және толығымен оның

серверлік құрылымын көрсетеді. Серверлік бөлім өзіне веб-серверлік және серверлік деректі кіргізеді. Веб-сервер есебіне келесілер кіреді:

- HTTP сұранысқа жауаптарды қабылдау;
- сұранысты басқа деректерге, басқа домендерге және төменгі домендерге бағыттайды;
- қарастырылып отырған бағдарламамызға басқа модульдар қол жетімді, оларда деректер қорын өңдеуге болады;
- деректер қорын күту;
- деректер қорын толықтау.



Сурет 1.2 – Бағдарламаның сервер бөлімінің детализациясы

HTML тілін ағылшын ғалымы Тим Бернерс-Ли 1989—1991жж.. Женевадағы (Швейцария) Жарылыс зерттеулері бойынша Еуропалық кеңес қабырғаларында өңдеді. HTML тілі алғашында беттеу аймағында маман болып табылмайтын адамдарға ғылыми және техникалық құжаттармен алмасу үшін құрылды. HTML тілі құрылымдық және семантикалық элементтер— дескрипторлар жиынтығын анықтау жолы арқылы SGML мәселелерін жақсы шешті. Дескрипторлар өте жиі «тегтер» деп аталады. HTML тілі көмегімен салыстырмалы түрде қарапайым, бірақ әдемі рәсімделген құжатты оңай құруға болады. HTML тілінде құжаттың қарапайым құрылымынан басқа гипермәтіндік қолдау бар. Мультимедиялық мүмкіндіктер кейін қосылды.

Алғашында HTML тілі құжаттарды ойнату (көрсету) құралдарына байланыстырмай құрылымдау және пішімдеу құралы ретінде ойластырып

табылды. Идеалды түрде HTML белгісі бар мәтін құрылғыда әр түрлі техникалық жабдықталулармен стилистикалық және құрылымдық бұрмалаусыз өндірілуі керек (дербес компьютердің түрлі түсті экраны, органайзердің монохромды экраны, өлшемі бойынша шектеулі ұялы телефонның немесе мәтіндерді дыбыстық ойнату бағдарламасының және құрылғының экрандары). Бірақ қазіргі кездегі HTML тілінің қолданылуы өзінің бастапқы қолданысынан әлдеқайда алысырақ. Мысалы, бірнеше рет беттерді пішімдеуге қолданған <TABLE> тегі құжаттарда қарапайым кесеттер құру үшін арналған. Уақыт өте келе HTML тілінің платформаға тәуелсіздіктің негізгі идеясы мультимедиялық және графикалық рәсімдеу қажеттіліктеріне 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.

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:

 Екі тег арасындағы мәтін — ашылушы және жабылушы.

 мұнда элементтің href. атрибуты бар [3].

1.4.1 Операциялық жүйе

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



Сурет 1.3 – Операциялық жүйенің компьютердегі логикалық құрылымы

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 Қазіргі операциялық жүйесіндегі желінің шолуы

Қазіргі операциялық жүйені үлкен екі топқа бөлсе болады : Windows-және Unix-жүйесіне.

Серверлердің шешім қабылдауының ішінде Windows 7 келесі операциялық жүйелерді қарастырады:



– Windows Server 2008 R2 Foundation — бұл қымбат емес және бизнеске арналған экономикалық техникалық шешім. Бұл редакция жекелеңген кішкентай компанияларға арналған және Ақпараттық технологиялар мамандықтарына арналған, оларды қамтамасыз етуге арналғандарға. Бұл қымбат емес, ыңғайлы ашуға және сенімді платформалы, осы платформада бизнес бағдарламаларды жұмысқа асыруға болады.



– Windows Server 2008 R2 Standard — бұл ең сенімді операциялық жүйе, қазіргі уақыттағы Windows Server. Бұл жүйеде веб-сервері және виртуалдау мүмкіндігі бар. Ол сенімділікті артырады және сервердің иілу инфраструктурасын қарастырады, егер шығынымыз төмендесе және үнемдеу уақыты азайса. Қуатты инструментіміз сервермен ыңғайлы басқаруды қанағаттандырады. Сенімді қауіпсіздікті ең алдымен бағдарламаның өзі көрсетеді.



– Windows Web Server 2008 R2 веб-бағдарламаға және веб-қызметке арналған қуатты бағдарламма. Бұл редакцияда қызметтер Internet Information Services (IIS) 7.5 және интернет-серверіне арналған; осында администрлеуге

және диагностикалық тестілеуге әрекет береді және басқа бағдарламамен жұмыс істеу өте ыңғайлы. Бұл платформа веб-сервердің, DNS-сервердің рөлін қолдайды, жоғарғы қорғанысты және әр түрлі орталарға ие болады.

Сонда, ең лайық операциялық жүйе Windows платформасындағы Windows Web Serv-er 2008 R2 болып тағайындалады. Лицензияның бағасы 88449,9 тенгені құрайды.

Unix операциялық жүйенің ортасы, келесі екі операциялық жүйені бөледі:

- MacOS X Server;
- Linux операциялық жүйесіне негізделген;
- MacOS X Server тек қана Apple құрылғысында ғана жұмыс істейді.

Сервер құрылғысы 600 000 теңге тұрады.

Операциялық жүйе арасында, Linux ядросында жасалғандардағы келесі түрлерін бөлуге болады:

- сервер Альт Линукс Ковчег. ALT Linux 5.0 Ark Server россиялық зерттемемен болып табылады және толық бағаланған шешімді шоғырлану үшін корпоративтік желілерге бағдарланған [4].

Негізгі функционалдық мүмкіндіктері:

- кәсіпорынның желісін ұйымдастыру;
- пайдаланушының және сервистің орталықты авторландыруы;
- ұйымдық аумақты (VPN) офистерге бөлу;
- орталық жаңартылған серверлер және жұмыс стнацияларына бөлу;
- жұмыс бекетінің желілік қондырғысы;
- көшірменің резервті копиясын жасау және қалпына келтіру (оған қоса жеке файлдарды);
- біртұтас файл қойма үлестемелеуді және квоталау;
- автоматикалық тұрғыда RAID жасау оны екі немесе одан басқа қатты дискісіне ескеру;
- ыңғайлы администрациялауға жасалған, қолданушыға терең зеріттеулерді қажет етпейді;
- құны — ақысыз.

Ubuntu (ubuntu-адамшылық) - Linux ядросын қолданатын және Debian-ға негізделген операциялық жүйе. Негізгі жасаушысы және демеушісі Canonical компаниясы болып табылады. Қазіргі уақытта жоба белсенді түрде дамуда және ашық бірлестік тарапынан қолдау көруде.Ubuntu-ды 21 миллионнан астам қолданушы пайдаланады, бұл оны Openstat санақ ғылымы бойынша десктоптарға арналған Linux-тың ең танымал таратушысы етеді. Веб-серверлерге арналған ең танымал ОЖ-дің тізімінде ол 4-орында тұр.Әдетте дистрибутивтің жаңа нұсқасы әрбір жарты жыл сайын шығады және оған бір жарым жыл бойы қауіпсіздік жаңаруларымен қолдау көрсетіледі, ал 2 жылда бір рет шығатын LTS нұсқасына 3 жыл бойы көмек көрсетіледі (серверлік нұсқасы-5 жыл бойы). Сонымен қатар енді LTS -тың келесі шығарылымдарына нұсқаға тәуелсіз 5 жыл бойы қолдау көрсетілетіні туралы хабарланды[5].

– құны — ақысыз.

Бүгінгі күні Linux-тің бір ядросы бар және оның жетілдіруін Линус Торвальдс (Linus Torvalds) және Алан Кокс (Alan Koks) координациялайды. Linux дистрибутивтар жиыны (20-30-дан кем емес), олардың айырмашылығы: функционалдық тағайындауында және дистрибутивке енген бағдарламалық қамтамасыз ету құрамымен. Кейбір дистрибутивтер ондаған компакт-дисктен, ал кейбіреулері бір-екі дискетадан тұрады.

Дистрибутив үшін құжаттар бөлек беріледі. Бірақ оған көптеген ережелер, анықтамалық файлдар және HOWTO кіреді, сондықтан сауатты қолданушы құжатты қажет етпейді. Linux-ті мықты құрастырушылар: Red Hat Inc., және Mandrake S.A. өздерінің нұсқаларын дистрибутивке енгізеді, бірақ бұл OS жиынының бағасының көтерілуіне әкеліп соғады. Linux дистрибутивтері көптеген әр түрлі компаниялармен шығарылады. Бүгінгі күні Linux-тің 90 түрі бар, бірақ олардың барлығы Unix стандарттарымен ұқсас. Батыстағы Linux-тің негізгі құрастырушылары:

– Red Hat, Inc. – Red Hat Linux-тің үш түрін: Home, Standard және Professional шығарады;

– Caldera, Inc. – Caldera Open Linux түрін шығарады;

– Debian Group – Linux Debian;

– Corel, Inc. – әмбебап саудаға қондырылуы ыңғайлы Corel Linux шығарады, ол Linux Debian қорында сақталады;

– SUSE Group – S.U.S.E Linux-ті шығарады, әсіресе еуропалық тілдер үшін локальданған.

– Келесі дистрибутивтер ТМД территориясында өте танымал:

– “Процедура” қоғамынан Red Hat Linux. Бұл компания ТМД саудасы үшін ыңғайлы бағамен сатылатын Red Hat дистрибутивтерінің көшірмелерін шығарады.

– Ресей-француз “Mandrake S.A.” компаниясынан Mandrake Linux дистрибутиві. Linux-тің бұл түрі жақсы орыстілдік локализациямен, көлемді құжаттарымен және арзан бағамен танымал.

– Black Cat Group, BC Linux дистрибутивімен танымал. Бұл Red Hat дистрибутивінің түрі, бірақ ол Украинаға ғана локальданған.

Техникалық аспектісінде бұл барлық дистрибутивтерінің айырмашылығы өте бос қарастырылады (айырмашылықтар динамикалық кітапханаларда, аудармаларда және қондырушы бағдарламалардың түрлерінде бар). Таңдаудың негізгі критеріі тек дистрибутивтің қателерінің, аударманың және бағаның болмауы болып табылады[5].

Талаптарға байланысты операциялық жүйені таңдау.

Ең оптималды вариант ретінде Ubuntu 8.04 LTS операциялық жүйесін аламыз. Факторларға негізделген, шешімдеріне әсер еткен:

– ақысыз шешім;

– операциялық жүйеге, хостинг-провайдерге әрекет жасайды;

– толық документациялау, орыс тілінде жүргізілген;

- тематикалық одақтың үлкен саны бары;
- LTS версиясы (англ. Long Term Support; сөзбе сөз «үлкен уақыт аралығында қамтамасыз ету») білдіреді;
- «пакетті» тарату операциялық жүйелерге жүреді, пакетті қондыру үшін бізден тек қана бір пакетті талап етеді, қалған файылдар операциялық жүйеде бар, егер жоқ болған жағдайда, ол интернет желісінен жазылып алынады;
- көптеген санды серверларды және деректер қорын, осы дистрибутте қолдануға болады.

1.4.2 Дерект қорын басқару жүйесі

Деректер қорын басқару жүйесі (ДҚБЖ) бағдарламалық қамсыздандыруды ұсынады, барлық дерекқорды басқарады және рұқсат береді [4].

Деректер қорын басқару жүйесінің критикалық моменті ол, сенімділігі және өндірімділігі. Себебі деректер қорында өтте үлкен мәліметрет сақталуы мүмкін, тағыда көп қолданушылардың деректер қорны манипулярлы түрде қолдануын ескеру қажет.

ДҚБЖ негізгі талаптары:

- сыртқы жадтағы деректерді басқару (дисктерде);
- дискті кэшти қолдана отырып, жедел жадтағы деректерді басқару;
- өзгеріс журналы, дерекқор қалпына келтіру;
- реляциялық моделді деректер қорын сақтау;
- клиент пен сервер арасындағы әрекеттестікті сақтау;
- сыртқы утилиті бары басқарма үшін ДҚБЖ;
- құны төмен.

Қазіргі ДҚБЖ шолу.

Дүние жүзінде көптеген ДҚБЖ бар, осылардың бәрін дипломдық жұмыстың қоршауына кірмейді. Тек қана ең танымалы ДҚБЖ қарастырамыз және олардың ерте замандағы тарихын жазып өтеміз.

Oracle Database 11g Standard Edition.

Oracle Database 11g ДҚБЖ бірден-бір болып табылады, есептеуіштерді орнатуға немесе бөлуге арналған (Grid). Oracle Database 11g шығара, Oracle компаниясы қолданушыларға басқару мүмкіншілігін жеңіл жолын көрсетті, соған байланысты клиентер өздерінің бизнесін іске асырып инновацияларды еңгізді. Oracle Database 11g өндірімділіктің ең асқақ деңгейлерін қамсыздандырады, масштабтандырады, қолжетімділік, қауіпсіздік және басқарманың желідегі жайлылығын стандартты сервер және сақтау жүйесімен танымалы. Oracle Database 11g ДҚБЖ тиімді ұңғылау үшін арналған және әр түрлі типтегі базалармен және құрылғылармен жұмыс істеуге ыңғайлы. Бұл өнім автоматизацияланғандағы басқару функциясын ұсынады және эффективті эксплуатациялауды ұсынады.

Шешімнің құны — 500 000 тенгеден жоғары .

Firebird.

Firebird (FirebirdSQL) — компактi, кросс-платформасында, басқарманың еркін жүйесіндегі дерекқорлар (ДҚБЖ), GNU/Linux, Microsoft Windows жұмыс істейтін жіне әр түрлі Unix платформаларында.

Firebird ерекшілігі ретінде көп версияланған, шұғыл және талдағыш сұраныстың қамсыздандыруын қарастырады. Компакті (дистрибутив 5Mb), жоғарғы эффективті және мықты тілдік сүйенішті сақталмалы процедураларға және триггерларға арналған.

Жаман жағы арасында: сұраныстан шығатын кәштың жоғы, толық тексті индекс.

Құны—ақысыз.

MySQL.

Microsoft SQL Server – деректер базасын басқарудың реляционды жүйесі (ДББЖ). Реляционды деректер базасында мәліметтер кестелерде сақталады. Өзара байланысты мәліметтер кестелерге топталуы мүмкін, сонымен қатар кестелер арасында да байланыс орнатылуы мүмкін. Осыдан реляционды деген атаудың негізі шығады, ағылшын сөзі relational (туыстық, байланысқан). Қолданушылар деректерге бағдарламалар арқылы қол жеткізеді, ал администраторлар серверге тікелей қол жеткізеді.

MySQL Enterprise бір жылдық лицензияның құны — 600 у.е., немесе 90 000 тенге.

MySQL Community edition лицензияның құны — ақысыз.

Талаптарға сәйкес ДҚБЖ таңдау.

Oracle деректер қорын таңдау шешімді емес, себебі лицензияның бағасы өтте жоғары болып келеді және мобильдік телефонда деректер қорының түрі еңгізілмеген.

FireBird және MySQL ДҚБЖ салыстырса олар функционалды ұқсас, бірақ FireBird кешті түрде сұраныстарды қолдамайды, бұл жұмыста жылдамдығы өтте төмен болғанға байланысты деректерді оқу жылдамдығы азаяды. Тағыда FireBird жайлы деректер өтте аз, соған байданысты интернет желісінде мәліметтер аз. Ол біздің жұмысымыздың тез арада шешілуіне бөгет және шешімдерді іздеу кезінде қиыншылыққа соқтырады. (сурет 1.4).

MySQL толығымен желіге тіркеішілігі бар ОЖ Ubuntu Server 8.04. MySQL 5.0.x тармақтарын ұстайды, ең тұрақты на айтылмыш кезді болып табылады. MySQL өзінде екі лицензиясы бар. MySQL GPL лицензиясының шарттарымен тарау береді. Бірақ GPL шартына байланысты, егер бір бағдарлама бастапқы кодты қосатын болса MySQL, онда олда GPL лицензиясымен таралуы қажет болдады. Сол әзірлеушінің бағдарламма жасаушылармен жоспарларлануы қажет, өзінің бағдарламасының ішкі кодын ашқысы келмесе Осындай байланысты коммерциялық лицензиялар қарастырылған, осылар нарықты сервисті сүйенішті қамсыздандырады.

Ол білдіреді, бағдарлама шегінде ақысыз MySQL Community Edition 5.0.10 бағдарламасын қолданамыз.

MySQL де API ,Delphi тіліне арналған C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk және Tcl, кітапханалар платформа тілдері үшін.NET, тағыда ODBC арқылы ODBC-драйвері MyODBC.

Scale is based on the average worldwide traffic of **firebird** in all years. [Learn more](#)



Сурет 1.4 – Дүниежүзілік «mysql» және «firebird» ізденістің трафигін салыстыру

SQL - Structured Query Language-дің қысқартылуы (сұраныстардың құрылымдық тілі). SQL реляционды мәліметтер базасымен жұмыс істеу үшін жасалған. Ол қолданушыға мәліметтер базасымен жұмыс істеуге (қарау, іздеу, қосуға және ақпаратты басқаруға) мүмкіндік береді. MySQL ANSI 92 SQL спецификасына сәйкес келеді.

MySQL – интернеттегі ең кең таралған және ең атақты ақпарат базасын басқару жүйесі. Ол үлкен көлемді ақпаратпен жұмыс істеуге арналмаған, бірақ үлкен, кіші интернет сайттар үшін өте ыңғайлы жүйе.

MySQL үлкен және орташа бағдарламалармен жұмыс істеуге өте ыңғайлы. Сервердің негіздері көптеген платформаларда компиляция жасайды. Сервердің толығырақ мүмкіндіктері өнімділікті арттыратын көпағымдылықтың қолдауы бар Unix-серверлерде байқалады.

MySQL жұмыстың жоғары жылдамдылығымен, сенімділігімен, ыңғайлылығымен ерекшеленіледі. Онымен жұмыс істеу үлкен қиындықтар тудырмайды.Тегінділігі де үлкен роль атқарады. MySQL ортақ GNU (GPL, GNU Public License) лицензиясы негізінде таратылады. Бұрын ақпаратты ұзақ уақыт сақтау үшін файлдармен жұмыс істелетін: бірнеше жолдар санына енгізіп, келесі жұмыстар үшін алынатын. Ақпаратты ұзақ уақытқа сақтау мәселесі Web – бағдарламаларын программалауда жиі кездеседі:

қолданушыларды санау, хабарламаларды форумда сақтау, сайттағы ақпаратты арақашықтықтан басқару т.б.

Файлдармен маманданған түрде жұмыс істеу үлкен еңбекті талап етеді, оларға ақпаратты енгізуді, олардың сортталуын, алынуын қадағалау керек, сонымен қатар бұл істердің бәрі Unix нұсқасының бірі тұратын хост – провайдердің серверінде жүзеге асатынын ұмытпау керек, яғни файл құқығы мен оларды орналастыруды да ойластыру керек. Сонымен қатар код көлемі елеулі түрде өседі және программада қате жіберу өте оңай. Осы мәселелердің барлығын мәліметтер базасын қолдану арқылы шешеді. Мәліметтер базасы ақпараттың қауіпсіздігі мен сортталуын өздері қадағалайды және ақпаратты алу мен орнатуды бір жолдың көмегімен жүзеге асыруға мүмкіндік береді. Мәліметтер базасын қолданумен код жинақы болып шығады, және оны әлдеқайда жеңілдірек жөндейді. Сонымен қатар, жылдамдықты да ұмытпау керек – ақпаратты мәліметтер базасынан алған файлдан алғанға қарағанда әлдеқайда тезірек жүзеге асады. MySQL мәліметтер базасының ақпаратын сақтауда қолданылатын бағдарлама файлдарда салынған бағдарламаларға қарағанда әрқашан тезірек істейді. Өйткені мәліметтер базасы C++ тілінде жазылған және қатқыл дискте мәліметтер базасына қарағанда нәтижелі жұмыс істейтін PHP программасына жазу шешілмейтін мәселе, өйткені PHP программалары C++ программаларынан баяуырақ жұмыс істейді, себебі PHP – интерпретатор, ал C++ компилятор.

Осылайша, мәліметтер базасының басты артықшылығы қатты дискпен жұмыстың бәрін өз мойнына алады және өте жоғары деңгейде жүзеге асырады. MySQL мүмкіншіліктерінің қысқаша тізімі:

- 1) бір уақытта мәліметтер базасымен жұмыс істейтін тұтынушылардың шексіз санына қызмет көрсетіледі;
- 2) кестедегі жолдар саны 50 млн. дейін жетуі мүмкін;
- 3) бұйрықтардың тез орындалуы. MySQL серверлердің ішіндегі ең жылдамы болуы да мүмкін;
- 4) қауіпсіздіктің қарапайым және нәтижелі жүйесі;

MySQL негізінде өте жылдам сервер, бірақ бұған жету үшін өңдеушілер МБЖ-нің реляциялы талаптарынан бас тартуына тура келді. MySQL-де:

- 1) `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)` секілді енгізілген сұраныстар жұмыс істемейді. Мұндай мүмкіндік 3.23 версиясында болады деп айтылуда;
- 2) транзакцияны қолдау жүзеге аспаған. Орнына `LOCK/UNLOCK TABLE` қолдану ұсынылады;
- 3) сыртқы(`foreign`) кілттер жұмыс істемейді;
- 4) триггер және сақталатын үдерістер істемейді;
- 5) көрсетілім (`VIEW`) жұмыс істемейді. 3.23 версиясында көрсетілім мүмкіндігі іске асады деп жоспарлануда.

Құрастырушылар сөзі бойынша дәл 2-4 пункттер жоғары жылдамдыққа қол жеткізуге мүмкіндік берді. Оларды жүзеге асыру сервердің жылдамдығын елеулі түрде төмендетеді. Бұл мүмкіндіктер Web – бағдарламаларды жасағанда қиын емес және осыған орай жылдамдық пен бағасының арзандығына

байланысты серверге үлкен танымалдылық әкелді. Төменде MySQL программалық жабдықтаманың маңызды сипаттамалары көрсетілген:

Ішкі сипаттамалар және тасымалдануы:

- түрлі платформаларда жұмыс істейді. Тасымалдануды қамтамасыз етуде GNU Automake, Autoconf және Libtool қолданылады;

- C, C++, Eiffel, Java, Perl, PHP, Python, Ruby және Tcl-ге арналған API;

- ядро ағымын пайдалана отырып толығымен, көпағымды. Бұл дегеніміз, мұндай мүмкіншілік қамтамасыз етілсе бірнеше процессормен жұмысты оңай ұйымдастыруға болады;

- бұтақ негізінде индексті сығумен өте тез дисктік кестелер;

- көлемді тарату жүйесінің ағымындағы өте тез негіздеуші;

- біржолды мультибайланыстардың (one-sweep multi-join) оптималды әдісін қолданатын өте тез байланыс;

- уақытша кестелер ретінде қолданылатын көлемдегі хеш – кестелер

- жақсы оптимизацияланған класс кітапханалары көмегімен жүзеге асқан SQL-функциялары. Сондықтан олар мүмкіншілігінше тез жұмыс істейді. Әдетте сұранысты инициализациялағаннан кейін көлемді тарату мүлдем жасалынбайды.

MySQL – код Purify, Valgrind және GPL – құралдарының бірін қолдану арқылы тестілеуден өткен.

Бағана түрлері:

- үлкен сан: белгімен/белгісіз бүтінсандық, ұзындығы 1,2,3,4 және 8 байт;

- бекітілген және айнымалы ұзындықты жазбалармен;

- барлық бағаналардың үндеместен мағынасы бар. INSERT көмегімен кес тенің көп бағаналарын қоюға болады; мағына берілмеген бағаналарға мағыналар үндеместен беріледі;

- операторлар мен SELECT және WHERE-ді сауал жөнінде толық қолдау көрсету;

- SQL GROUP BY және ORDER BY операторларына толық қолдау көрсету.

- LEFT OUTER JOIN және RIGHT OUTER JOIN -ді ANSI SQL және ODBC синтаксисімен;

- SQL92 стандартына сәйкес кестелер мен бағаналарға жалған аттар рұқсат етілген;

- DELETE, INSERT, REPLACE, және UPDATE өзгертілген бағана сандарын қайтарады. Оның орнына сәйкес келген бағаналарды беруге болады. Ол үшін сервермен байланыс кезінде жолақ орнату қажет;

- MySQL үшін арнайы болып табылатын SHOW командасын мәліметтер базасы, кесте және индекс туралы ақпарат алу үшін қолдануға болады. Оптимизатордың сұранысты қалай орындайтынын анықтау үшін EXPLAIN командасын қолдануға болады;

– функция аттары бағандар мен кестелер аттарымен дауласпайды. Мысалы, ABS бағананың дұрыс аты болып табылады. Функцияны шақыру үшін тек бір ғана шектеу бар: функция аты мен келесі ашылатын жақша арасында бос орын болмауы керек.

Қауіпсіздік:

– жүйе құпия сөз бен жеңілдіктер негізінде құрылған, соған орай қауіпсіздік пен ыңғайлылық қамтамасыз етіледі. Құпия сөздер қорғалған, өйткені желі арқылы берілгенде сервермен байланысқанда шешіледі.

Көлемділік және шектеулер:

– өте үлкен мәліметтер базасын басқару. MySQL AB компаниясы MySQL - ді кейбірі 50 миллион жазбалар бар бірнеше мәліметтер базасымен жұмыс істеу үшін қолданады. Сонымен қатар, MySQL-ді 60 000 кестемен, 5000000000 жолмен жұмыс істеу үшін қолданатын тұтынушыларды білеміз;

– әр кестенің 32 индексі болуы мүмкін. Әр индексте 1-ден 16 баған не баған бөлшектері болуы мүмкін. Индекстің максималды кеңдігі 500 бит (бұл көрсетілім MySQL-ді компиляциялағанда өзгеруі мүмкін). Индекс үшін CHAR немесе VARCHAR префикс полялары қолдануы мүмкін.

Жергіліктендіру:

– сервер клиенттер үшін қателіктер туралы хабарламаны түрлі тілдерде қамтамасыз ете алады;

– ISO-8859-1 (Latin1), немісше, big5, ujis тағы басқа түрлі кодировкаларды жұмыс істетуді;

– барлық ақпаратты сақтау үшін таңдалған белгілер жинағы қолданылады. Қалыпты жолдары бар бағаналар үшін барлық салыстырулар белгілер регистрінің есептелуі бойынша жүргізіледі;

– сорттау таңдалған алфавит бойынша жүргізіледі (үндеместен шведтік қолданылады). Бұл орнатылымды MySQL серверді қосқан кезде өзгертуге болады. Өте жақсы жасалған сортталу мысалымен танысу үшін чех тілі үшін арналған сорттау кодын қолдануға болады. MySQL көптеген басқа кодтауларға қолдау көрсетеді.

Клиенттер және инструментарий:

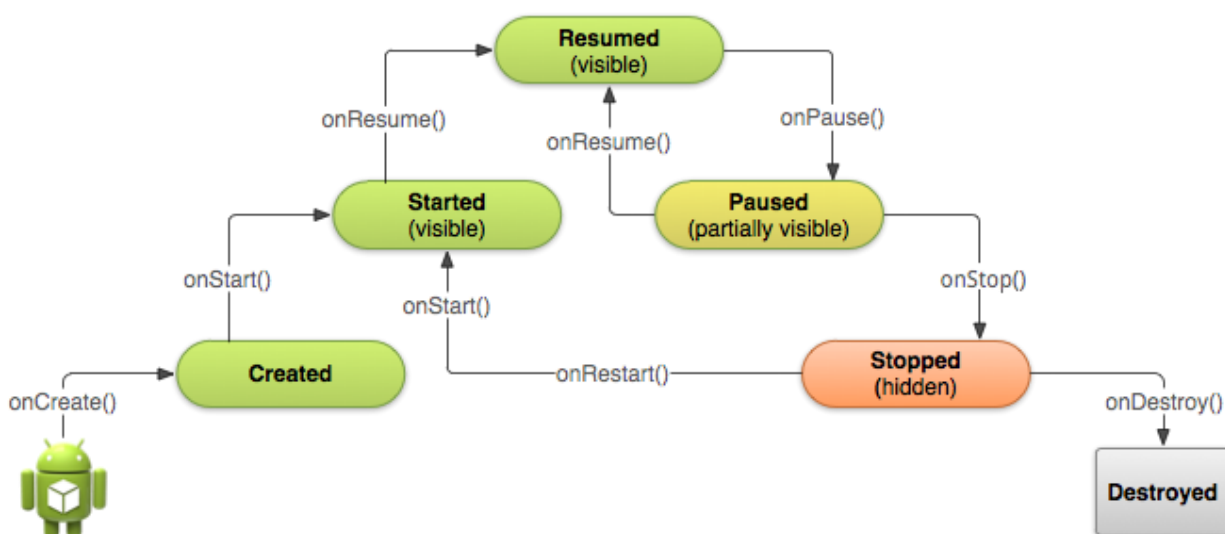
– құрамында myisamchk, кестелерді қайта келтіру, оптимизация және тексеру утилитасы. Барлық функционалды мүмкіншіліктері myisamchk SQL-интерфейс арқылы да қол жетерлік;

– барлық MySQL – программаларды көмек алу үшін help опциялары арқылы қосуға болады.

1.4.3 Android бағдарламасының жалпы жұмыс істеу сұлбасы

Android-пен жұмыс істейтін бағдарлама терезені қолданады (Windows тәрізді), бірақ жоғарыда көрсетілген система Activity атын қолданады. Windows сияқты, әр терезе өзінің өмірлік циклы және жеке ерекшеліктері бар. Жаңа терезе құру кезінде onCreate() тәсілі шақырылады, тағыда бағдарламаны

жасау кезінде, бұл тәсілде бағдарлама және компонентары инициализацияланады. Одан кейін onStart() және onResume() тәсілі шақырылады. Екі тәсілде терезені бейнелеудің алдында көрсетеді немесе қалпына келтірі кезінде (ауыстырып қосу немесе басқа бағдарлама азу кезінде тағы басқа). Бұрғылау кезінде onPause() және onStop() тәсілі шақырылады. Бағдарламаны жабу кезінде және onDestroy() терезесі шақырылады, осы тәсілде қолданушының мағлұматын сақтауға және параметірін қалпында ұстауға болады. Толығымен бейнелеу және тізбектей шақыру тәсілін ресми сайттан табуға болады. Android бағдарламасының толығымен өмірлік цикл 1.5-ші суретте көрсетілген.



Сурет 1.5 – Android операциялық жүйесінің жұмыс істеуінің өмірлік циклы

Android негізінде бірегей, өзгеше операциялық жүйе. Жақсы нәтижеге жету үшін бағдарлама әзірлеуші операциялық жүйенің өзектілігін және өзгешілігін білуі қажет. Бағдарлама әзірлеу кезінде бірнеше қиындықтарды ескеру қажет. Оларға тоқталып өтейік:

- бағдарламаны қондыру кезінде , бағдарлама екі есе немесе төрт есе орын алады, яғни ол бағдарламаның түпнұсқалық орның қажет етеді;
- орнатылған флеш-картамен жұмыс істеу кезінде файылдың жылдамдығы он есе төмендейді, егер бос орын аз болса;
- әр процесс кезінде жедел жақтан 16Мб (кейде 24 Мб) қолданылады.

Android негізінде Linux операциялық жүйесінде жасалған. Бағдарламаның және ядроның арасында API қабаты және нативті кодтағы кітапхана орналасқан. Бағдарлама виртуалды машинада Java (Dalvik Virtual Machine) орындалады.

Android -та көп теген бағдарламалар қосуға болады. Брақ бір бағдарлама толығымен бүкіл экранды алады. Ағымдағы бағдарламаны қолданып отырып басқа бағдарламаны қосуға немесе жаңасын ашуға болады. Ол тарихын қоруге болатын браузерге ұқсайды.

Қолданушының интерфейсі әр экранда Activity класс кодындағы көріністі көрсетеді. Процесста әр түрлі Activity тұтынады немесе қолданылады. Процесстаң көрі Activity көбірек өмір сүреді, яғни ол дегеніміз Activity қолдануы көбірек. Activity жұмыс істеу кезінде тоқтатылып және жаңадан қосылып бүкіл қажетті информацияны сақтап қосылады.

Android арнаулы механизмді қолданып әрекеттің сипаттамалары бас Intent негізде жүргізіледі. Белгілі әрекеттер істелген кезде (қоңырау соғу, хабарлама жіберу, терезені көрсету), Intent шақырылады.

Тағы да Android демоға ұқсас Linux серверларын ұстайды, олардың атқару қызметі керекті әрекетті фондық режимде (мысалы, өленнің ойналуы). Бағдарламада деректерді айырбастау үшін Content providers (провайдері қолданылады).

Айтылмыш жұмыс үшін деректер провайдерін пайдаланып, қолданушының құрылғысына орнату керек.

1.4.4 Java құрылымы және өзгешеліктері

Java – бағдарламашылардың (программистердің) арасында кең таралған, танымал ОБ(объектіге бағытталған) бағдарламалау тілдердің бірі. Sun Microsystems 1995 жылы шығарған Java 1.0 версиясы бағдарламалауда жаңа стандарттар орнатып, революция жасады. Нәтижесінде Web нағыз интерактивті ортаға айналды. Әрине ғаламдық тордың дамуы есептеуіш жүйелердің қалыптасуына үлкен әсерін тигізді. Web пайда болмаған кезде көпшілік қолданушыларда бір-бірімен байланыспаған дербес компьютерлер болған. Ал қазіргі кезде компьютерлердің барлығына жуығы Интернетке қосылған, ал ол, өз кезегінде есептеуіш жүйелердің жаңадан қалыптасуына және ақпарты бірлесіп қолдануға мүмкіндік береді. Сол жаңа жүйелердің қалыптасуы мен Интернеттің дамуы бағдарламалауда Java-ға негізделген жаңа көз-қарас, шешім алып келді. Бұл тіл тек қана Web-бағдарламалауды жеңілдеткен жоқ, оған қоса желілік бағдарламалаудағы жаңа типке, апплетке, жол ашты.

Java-да “апплет” ұғымы web бетте орындалатын кішкентай бағдарлама ретінде қарастырылады. Қолданушының компьютеріне жүктелген соң, ол браузер арқылы қолданушымен өзара әрекет жасап, белгілі бір тапсырмаларды орындайды.

Java-да “апплет” ұғымы web бетте орындалатын кішкентай бағдарлама ретінде қарастырылады. Қолданушының компьютеріне жүктелген соң, ол браузер арқылы қолданушымен өзара әрекет жасап, белгілі бір тапсырмаларды орындайды.

Java шынында да желілік ортада құнды құрал болып табылады. Бірақ ол тек онымен шектеліп қалмайды. Java тілі әртүрлі бағдарлама жасау үшін қолданылатын универсалды тіл. Оған қоса Java платформаға тәуелсіз проектер жасау үшін қолданылады. Бағдарлаудың қарапайымдылығы және қауіпсіздік амалдары тез және түсінікті код жазуға мүмкіндік береді. Басқа бағдарлау тілдерінде кеңінен тараған қателер Java бағдарламаларында болмайды. Десекте,

барлық ішкі әдіс-амалдарына қарамастан Java бағдарлау тілі қарапайым, үйренуге оңай тіл.

Артықшылықтары:

- архитектураға тәуелсіз;
- интернет бағдарламалар үшін ақысыз;
- бір рет жазылған код басқа жерлерде орындалады;
- көп ағымды;
- бағдарламалар желі арқылы тапсырманы орындай алады;
- динамикалы;
- код қажет болған жағдайда ғана қолданылады;
- ООП-ның бар болуы;
- Java бағдарлау тілі қазіргі ООП негізіне сәйкес бағдарламалар жазу

үшін қолданылады;

- жад басқарылмалы;
- жад автоматты түрде бөлінеді;
- “Garbage collector” қолданылмайтын жадты өшіріп отырады;
- сенімді;
- қатты типтелген;
- көрсеткіштер жоқ (C бағдарлау тілін өткенде түсінесіздер :-));
- қарапайым;
- бір-бірімен өте жақсы қиысқан тілдің элементтері оның қолданысын

жеңілдетеді.

– Java бағдарлау тілі – ол JDK (Java Development Kit). Яғни, бағдарлау тілі бір-бірімен топтасқан көптеген кішкентай бөлшектерден, класстардан, тұрады. Демек, Java-да жазылған бағдарламалар, негізінде, осы класстардың жиынтығын, пакеттерді, қолданады. Сонымен, Java-да бағдарлау кезінде қолданылады:

- Java-ның стандартты пакеттеріндегі класстар;
- өзіміз жаңадан жазған класстар;
- басқалардың көпшілік үшін жазған класстары;
- Java-да бағдарламаның жазу процесі;
- жазу .java файлын жазу арқылы өзіміздің классты анықтаймыз;
- компиляция жазылған .java файлды Java компиляторына жіберіп .class

файлды аламыз;

– орындау: пайда болған .class файлды Java интерпретаторына кодты орындау үшін жібереміз.

Java компиляторы – жазылған кодты орындау үшін дайындайды және байт-коды бар .class файлды шығарады. Егер компиляция кезінде қателер болса, онда кодтағы қателерді түзетіп қайтадан компиляциядан өткізу керек.

Байт-код – Java виртуалды машинасында орындалатын оңтайландырылған нұсқаулардың жиынтығы. Яғни, JVM байт-кодтың интерпретаторы. Бұл әдіс басында әдеттегіден тыс болып көрінуі мүмкін, өйткені қазіргі кездегі

бағдарлау тілдерінің компиляторлары байт-кодты шығармай бірден орындалатын кодты шығарады (.exe). Бірақ бұндай әдістің өзіндік артықшылықтары бар және жылдамдық ерекшеліктері зор емес.

Java интерпретаторы (JVM – Java Virtual Machine) – .class файлды оқып оны компьютерге түсінікті тілге аударды, яғни бағдарламаны орындайды.

JavaScript – прототипті-бағытталған скриптті бағдарламалау тілі.

ECMAScript тілінің диалектісі болып табылады. JavaScript әдетте бағдарламалар объектісіне бағдарламалық рұқсатнама ретіндегі құрылған тіл болып пайдаланылады. Браузерлерде веб-беттерге интерактивтілік беру тілі болып кең қолданыс табады.

Негізгі құрылымдық сипаттамалары: динамикалық типтеу, әлсіз типтеу, жадымен автоматты түрде басқару, прототипті бағдарламалау, бірінші класс объектісі сияқты функциялар.

JavaScript тіліне көптеген тілдер әсер етті. JavaScript тілі қандай-да бір компанияға жатпайды. «JavaScript» атауы Oracle Corporation компаниясының тауарлық белгісімен тіркелген.

JavaScript тілінің танымалдылығы web-бет элементтерін бір-бірімен байланысу және қайта жүктеудің кең мүмкіндіктеріне байланысты. Дизайн бөліктерін көрсетуге, жасыруға, көшіруге және безендіруге көмектеседі. Осындай әрекеттер презентациялық эффектілерді жасауға, форма мәліметтерін өңдеуге және құрамын басқаруға септігін тигізеді.

Бірінің үстіне бірін орналастыруға болатын HTML бөліктері (пиксельге дейінгі дәлдікпен):

- пішіндермен жұмыс. Кез келген мәліметтерді скрипттер арқылы өңдеу ыңғайлы, ол ақпаратты серверге жіберу алдында дұрысытығына тексереді. Егер мәтіндік алаңға қандай да бір санды енгізуін сұраса, әріптік символдардың енгізілу мүмкіндігін ескеру қажет;

- суреттермен жұмыс. Скриптер арқылы суреттің алдын ала жүктелуін жасауға болады. Суреттерді тікелей көру алдында браузер өзінің жадысында сақтайды да, қажет болған жағдайда жылдамырақ бейнелейді. Ені, биіктігі, графикалық файлдың адресі сияқты суреттің негізгі параметрлерін динамикалық өзгертуге болады. Ол тінтуірдің әрекетіне байланысты жауап береді, ерекшеленгенде сурет өлшемі өзгереді. Ұқсас әдіспен суреттер галереясы жасалады суреттің өзгеруі бір бет шекарасында жүргізіледі;

- уақыт пен күнді айқындау. Көптеген дизайнерлер өз жобаларында ағымдағы күн мен уақытты көп қолданады. Арнайы күнмен жұмыс істеуге арналған Date объектісі пайдаланылады;

- cookies файлдарымен жұмыс. JavaScript cookie файлдарымен толық жұмысты қамтамасыз етеді. Cookie- техникалық ақпарат сақталатын локальды компьютердегі мәтіндік файл. Бұл файлды оқырманның соңғы кіру күнін, парольдерді, сонымен қатар сайттағы басқа да әрекеттері туралы ақпарат сақталады. Аталған әрекет сайтты жекешелеп кірушілерге ыңғайлы етіп жасайды;

– оқиғаларды бақылау. Оқиға ретінде пайдаланушының белгілі бір әрекеті немесе құжат күйінің өзгеруі айтылады. JavaScript көптеген оқиғаларды бақылап отырады да оған жасалған әрекеттерді анықтауға мүмкіндік береді. Мысалы, web-бетті жүктегенде onLoad оқиғасы жүреді. Егер скриптті құжат жүктелгеннен кейін жүзеге асуы керек болса, онда осы оқиға басталғандағы болатын функцияны анықтау қажет;

– жаңа терезелерді құру. Скриптер жаңа терезелерді құруға, оларға сыртқы пішінін және ішкі құрамын өңдеуге мүмкіндік береді. Жасалған терезелердің адрестік жолын алып тастауға, айналдыру жолағын жасыруға және экранда орналасу орнын беруге болады;

– браузерді тексеру. Түрлі браузерлерде объектілік модельдерді бейнелеу стандартталмаған, соңына дейін барлығы қолдамайды. Сол себепті универсалды құжатты жасау үшін, ағымдағы уақытта қандай браузер қолданыста екендігіне байланысты сәйкес кодты беру керек;

– математикалық функциялар. JavaScript қажетті арифметикалық операцияларды қамтиды, барлық стандартты математикалық функцияларды, бүтін мен үтірі жылжымалы сандарды, қолдайды;

– құжат құрамын басқару. Write() әдісімен кез келген ақпаратты жасалған құжатқа ендіруге болады, мысалы, бүгінгі күнді. Жаңа құжатты динамикалық құрастыруға болады. Бұл түрлі браузерлердің, операциялық жүйелердің ерекшеліктерін ескере, жеке мәтіндерді шығаруға мүмкіндік береді.

– әрине, мұнда JavaScript тілінің барлық мүмкіншіліктері жазылмаған. Дегенмен, айтылған мысалдардан web-беттің мүмкіндіктерін арттыратын универсалды және ыңғайлы құрал екендігін көруге болады.

1.4.5 XML тілі ерекшеліктері

Android платформасы мобильдік құрылғыға, бағдарлама жасау үшін ашық кодты көрсеті. Оның көмегімен құрылымның барлық компоненттарын көріге болады, осы операциялық жүйенің көмегімен, төменгі деңгейдегі бағдарламма жасаумен және камера орнатумен, графиканы программалауға болады. Android-та көптеген қызықты заттар бар, осыдан сұрақ туындайды: XML жайлы не үшін ойлаймыз? Алайда мүдде бұның бәрін XML ұсынбайды, неше сыртқы нысандарымен әрекеттестік көмегімен жүзеге асады. Жеке XML бұл интернет желісінде ақпараттарды айырбастау немесе жібері үшін қолданады, сол себептен ықтималдыққа байланысты ол бізгеде қажетті болуы мүмкін. Басқа да, XML деректерді беріліс үшін қолданылады, мысалы, Web-сервисі үшін. Басқа сөздермен, егер сізге керек болса, сіздің бағдарламаңыз интернет желісі арқылы жұмыс істеу үшін, сізге XML файлымен жұмыс істеу қажет. Тағыда Android-та XML файлдарымен жұмыс істедудің көптеген варианттары бар.

XML (Extensible Markup Language, белгілеудің кеңейтілген тілі) World Wide Web Consortium (W3C) консорциумы – HTML тілін жасаушылардың 2000 жылы ұсынған жаңа стандарты болып табылады. Web-парақ жасайтын бұл жа-

на технология және оның соңғы ерекшеліктері бүгінгі таңда әлі де әзірлену үстінде. XML құрылымы бойынша гипермәтін тілді емес, төменгі деңгейдегі басқа тілдерді суреттеуге арналған метатілді көрсетеді. Өз кезегінде XHTML (Extensible Hyper Text Markup Language, гипермәтінді белгілеуді кеңейту тілі) XML мен HTML 4.0 арасындағы аралық нұсқаны көрсетеді. Оның W3C кеңінен қолданылуы есебінен, бір стандарттан екіншісіне ақырындап көшіруді жүзеге асыру жоспарланады. XHTML тілін ерекшелендіруге кері сәйкестендіру принципі салынған. Басқа сөзбен айтқанда, HTML 4.0 технологиясы немесе басқа да ертеректегі версиялар бойынша орындалған парақтардың web-сайт иелері, Интернетте жарияланған құжаттардың форматына қандай да бір өзгеріс енгізбейді: себебі, жаңа тілді кеңейту мен барлық толықтырулар алдыңғы стандарттардың барлығын өзіне қосып алған.

XML мен XHTML HTML мүмкіндіктерін әлдеқайда кеңейтеді және web-мастерге қазіргі Интернеттің бар мүмкіншілігін жүз пайызға пайдалануға мүмкіндік тудырады, әсіресе, электрондық коммерция саласында. XHTML электрондық құжаттың модульдік архитектуралық құрылысын көрсететін болғандықтан, бұл стандарт Интернет үшін әдеттен тыс құрылғылармен қарым-қатынасының механизмін жасауға мүмкіндік береді. Ондай құрылғылар: факс, ұялы телефон мен телевизорлар. Талдаушылардың пікірі бойынша жарқын болашақта XML әлемдік желінің негізгі стандарты болады.

XML ақпараттарды толықтай сипаттайды. XML оны грамматиканы және т.б тілдерді, құжаттарды жасауда дұрыстығын тексеру және жоғары сапасы болуын қамтамасыз етеді. Команданың авторы ақпараттың және оған байланысты элементтерді көру және құжатты талдау үшін керек. XML құжатының т.б ерекшелігі мәліметтерді сақтау және оны жіктеу сияқты қасиетке ие.

XML сонымен қатар, ақпараттардың қатесіз болуын қадағалау және олардың бәріне бірдей стандарт қою үшін қажет. XML тағы бір ерекшелігі ретке келтіру программалары. XML құжаттары күрделі емес оны бүгінде еркін пайдаланып барлық программалық өнім ретінде тұтынуға болады. XML бүгінде Microsoft Internet Explorer 5/0 оның Netscape Communicator, СУБД Oracle DB-2 MS-Office-те пайдалану қарастырылуда. Бұның барлығы да жақын арада XML ақпарат алмасу мен ақпарат жүйесінен HTML программасының орнын басады. Web бойынша саяхатты сіз тек статистикалық парақ тексттік HTML форматындағы файлдан ала аласыз, Web-серверден сіздің компьютеріңізге жіберіледі. Web жүйесінен және парақты қолмен немесе ақпараттар мәліметтерінен статистикалық парақтарды генерлеуге болады. HTML парақтарының орнын динамикалықта жасауға болады. Ол үшін сіз деректер қорынан ақпараттарды алып браузердің сұранысына жауап береді, сіздің ұсынысыңызбен бірге гипертексте беріледі. Бұндай жағдай үнемі ақпаратты өзгерткенде қажет. Web-сервердің CGI және әр түрлі АРТ Web-серверінің негізгі протоколы бар. CGI стандартты протокол Web-серверінен браузер арасындағы байланысына әсер етеді. Протокол браузерге мәліметтерді сұраныс жасауға, жіберуге және шығару мен енгізу сияқты командалық парақты пайдаланады. Сервер параққа сұранысты алғанда CGI қатысы бар жағдайда ол сол кезде сіздің компьютеріңізге

стандартты программаны шығарады. Андроид бағдарламасындағы қолданылған xml кодымыз.

```
public class RssHandler extends DefaultHandler{
    private List<Message> messages;
    private Message currentMessage;
    private StringBuilder builder;

    public List<Message> getMessages(){
        return this.messages;
    }
    @Override
    public void characters(char[] ch, int start, int length)
        throws SAXException {
        super.characters(ch, start, length);
        builder.append(ch, start, length);
    }

    @Override
    public void endElement(String uri, String localName, String name)
        throws SAXException {
        super.endElement(uri, localName, name);
        if (this.currentMessage != null){
            if (localName.equalsIgnoreCase(TITLE)){
                currentMessage.setTitle(builder.toString());
            } else if (localName.equalsIgnoreCase(LINK)){
                currentMessage.setLink(builder.toString());
            } else if (localName.equalsIgnoreCase(DESCRIPTION)){
                currentMessage.setDescription(builder.toString());
            } else if (localName.equalsIgnoreCase(PUB_DATE)){
                currentMessage.setDate(builder.toString());
            } else if (localName.equalsIgnoreCase(ITEM)){
                messages.add(currentMessage);
            }
            builder.setLength(0);
        }
    }

    @Override
    public void startDocument() throws SAXException {
        super.startDocument();
        messages = new ArrayList<Message>();
        builder = new StringBuilder();
    }
}
```

```
@Override
public void startElement(String uri, String localName, String name,
    Attributes attributes) throws SAXException {
    super.startElement(uri, localName, name, attributes);
    if (localName.equalsIgnoreCase("ITEM")){
        this.currentMessage = new Message();
    }
}
}
```

2 Арнаулы бөлім

Бұл бөлімде бағдарламаны жобалау және мобильдік бағдарламаға деректер қорын құру қарастырылады.

2.1 Зерттелетін жүйе облысының анализі

Саяхатшы бағдарламасы «КСЕТ» - саяхатқа шыққан саяхатшыларға әр уақытта өзінің қай жерде келе жатқанын көрсетеді және аудармашыны қажет етпейді. Қолданушыға саяхатқа шығатын кезде бірнеше мағлұматтарды беріп кетеді және өзімен ең басты қандай заттарды алу керектігін айтып кетеді.

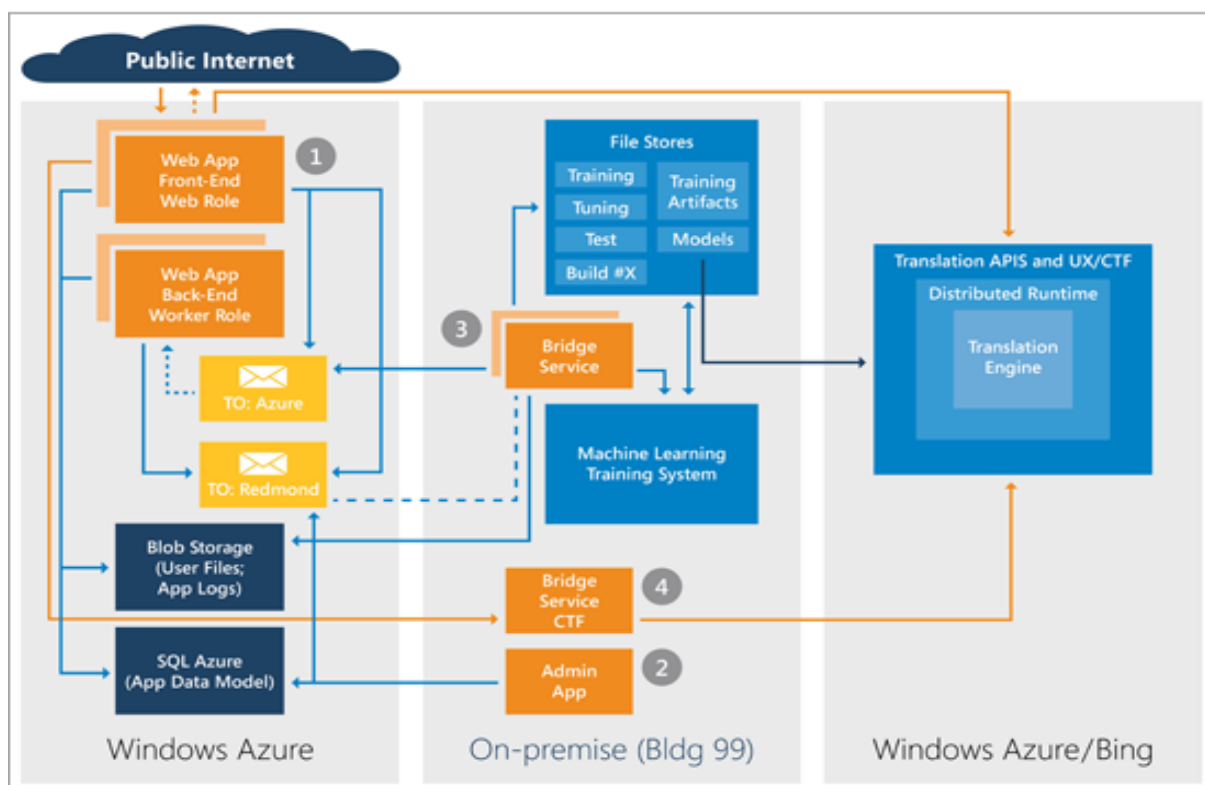
Бағдарлама жасау негізінде ең алдымен бірнеше тілдерге аударушы, туризмге шығу кезіндегі анықтама беру, GPRS навигациясы және блокнот.

Аудармашы ретінде Microsoft компаниясының Microsoft Translator API аудармасын қолданамын. Microsoft компаниясының бағдарламасының функциясын қолдану үшін ең алдымен бірнеше талаптарды орындау қажет және Microsoft компаниясымен келісім шартқа отыру қажет. Сол уақытта компанияның өнімін қолдануға болады. Microsoft компаниясының аудармашысы ақысыз тек қана бір жыл аралығына беріледі және оны басынан орнатуға мүмкіндіктер береді.

Параллельді документ қолдану, аудармашы Microsoft бірнеше шараларды ескереді:

- сөздерді және сөйлемдерді заңдарға байланысты аударды;
- контексті қалай сәндеуін түсіндіреді;
- етістікті және зат есімді сөйлемді жіктейді.

Microsoft Translator API жұмыс істеу схемасы және архитетурасы.



Сурет 2.1- Microsoft аудармашысының архитектурасы

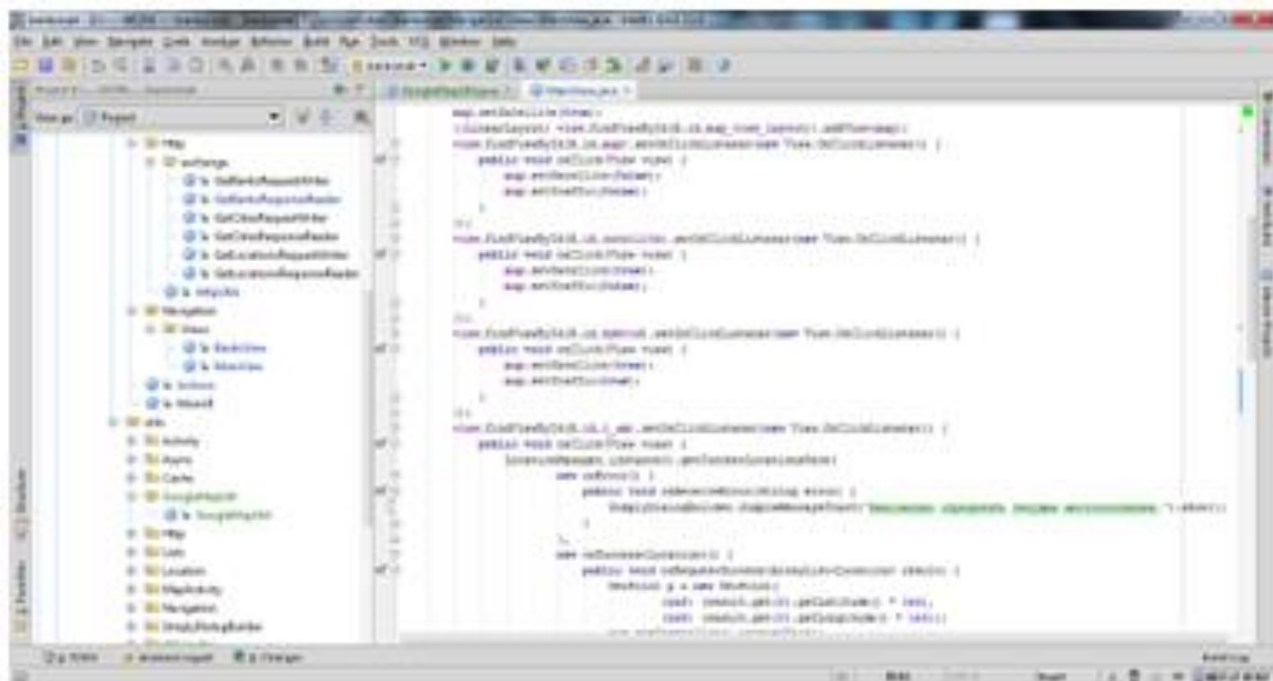
Өзгешеліктері және артықшылықтары:

- әр түрлі екі тіл арасында интернет арқылы жұмыс орнында аудару;
- қолданушы жүйесін жаңарту, сөйлемді нақты аудару;
- модель тек қана Microsoft қолданушыларына жетімді және өзінің шектеулері бар;
- организациялар аудармашы арқылы жүйелерді жасай алады және диалекті сөздерді аударуға мүмкіндіктерді береді;
- аударылған сөзді дауытап айтуға мүмкіндік береді және сөзді аударылған тілде айтады;
- қолданушы интерфейсі бағдарлама жасаушының өзіне байланысты;
- қолданушы бірнеше тілді бір уақытта аударуға мүмкіндік береді.

Біздің жасап отырған бағдарламамыз тек орыс тілден басқа ағылшын, неміс, итальян француз тіліне аударады.

Бағдарламаны жасау ортасы.

Бағдарлама жасау ортасы ретінде JetBrains компаниясының IntelliJ Idea қолданамыз. Бағдарламаны жасайтын орта көп болып келеді, бірақ біздің орта талай программистардың таңдауы және пірікі, графикалық интерфейс бойынша таңдалған. Келесі 2.2 суретте- бағдарлама жасау ортасы көрсетілген.



Суретте 2.2 - Бағдарлама жасау ортасы көрсетілген

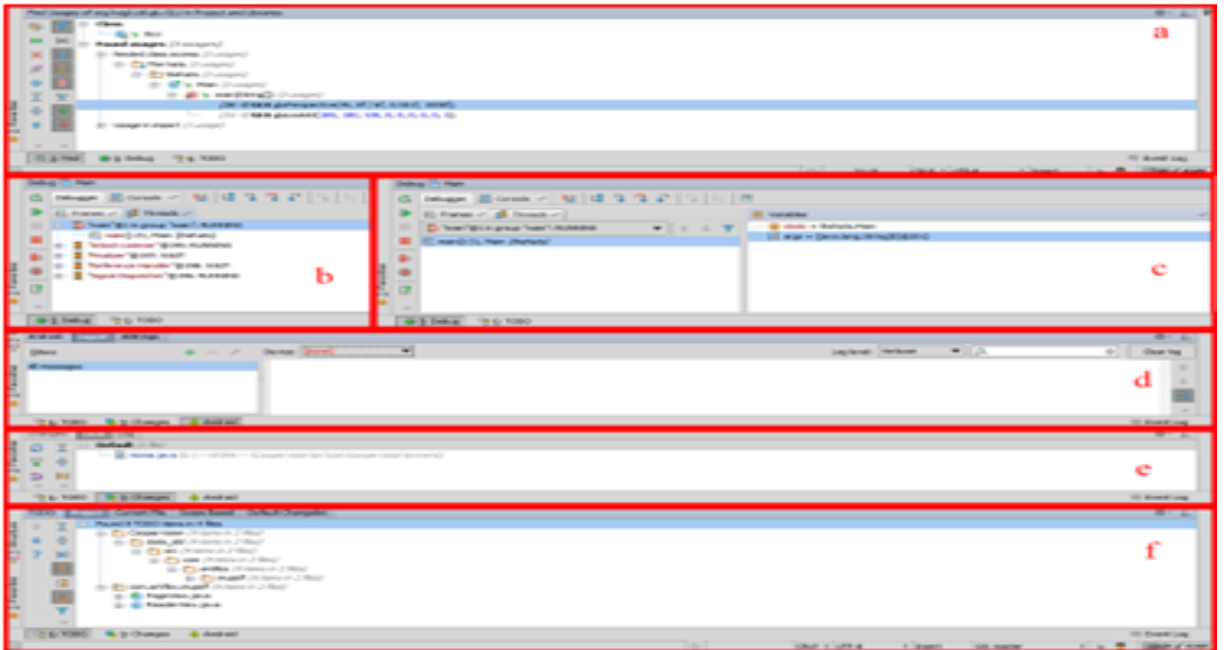
Жоғарғы 2.2-суретте базақыл түрде бағдарлама жасау ортасы көрсетілген. Суретте көрсетілгендей, бағдарлама жасау ортасы екіге бөлінген, сол жақтағы бағанда проекттің менеджер файлы және жылдам көруге болатын класстарды, ресурстарды және бағдарлама жасау навигациясын жеңілдетеді. Оң жақтағы блокта файлды редақтырлауға арналған ортасы көрсетілген.

Тағыда бағдарлама ортасы «Code injection» системасын қамтамасыз етеді, бұл система бағдарламаның фрагментарын редакторлауға және басқа тілдерге байланыстырады. Мысалға: бағдарламада деректер қорын қолдансақ, онда бұл жағдайда «SQL» тілін проектқа қоса аламыз және сұраныстарды редактірлеуге, java кодында жол ретінде сақтай аламыз, sql дағы сұраныс тағыдай. Html фрагменті тәрізді редақтырлауға және тағы басқа фаргемтерді орындаса болады .

Егерде бағдарламаны кеңейтілген түрде қолдансақ, онда біз тағыда бағдарлама панелің толықтаймыз. Осы панельдің ішіне мыналар кіреді:

- бағдарламаны ретке келтіру панель жабдығы;
- бағдарлама ағының бақылау панелі;
- консоль;
- іздеу нәтижесі «элементің қолдану»;
- лог панелі (лог бағдарламасы көрсетіледі);
- панель системасын бақылау версиясы;
- системалық хаббарлама;
- панель элементері;

Төменгі суретте аталып өткен панельдар көрсетілген.



2.3-сурет – Кеңейтілген режимде UI бағдарлама жасау ортасы көрсетілген

2.2 Google maps навигаторы Android API v2 бағдарламасына

Мобильдік бағдарлама жасаушыларға Google компаниясы өзінің Google maps картасын ұсынады. Ол бағдарлама жасаушыға картамен жұмыс істеуге өтте үлкен және ауқымды көрсеткіштерді көрсетеді. Картамен қолдану үшін ең алдымен Google компаниясының шарттарын орындау қажет, одна кейін жұмыс істеуге болады. Google компаниясы картаның бірнеше түрлерін көрсетеді.

API Google Картасында келесі карталарға қол жетімді:

- MapTypeId.ROADMAP – жол картасы, бағдарлама қойғанда өзі автоматты түрде орнатылады;
- MapTypeId.SATELLITE – Google ғарыш жерінің суреті, сурет спутник арқылы жасалады;
- MapTypeId.HYBRID – жай картаның комбинациясы және спутниктан жасалатын сурет;
- MapTypeId.TERRAIN – физикалық карта, ландшафтағы ақпараттар.

Картаның типі Map объектісімен қолданылады, оны өзгерту үшін MapTypeId. қасиеттерін өзгертеміз. Ол конструктордың үшіндегі Map options немесе setMapTypeId () арқылы арқылы карта өзгертіледі. Картаның типтеріне байланысты google.maps.MapTypeId. SATELLITE және google.maps.MapTypeId.HYBRID типтеріне бөлінеді.



Сурет 2.4- Google map 45° градустағы түсірілген сурет

Олардың айырмашылығы картаны 45° градуста көрсетуінде. Егер 45° градустағы картаның масштабын кішрейтетін болсақ онда картаның түрі өзгереді және төменгі шарттарды орындайды:

- барлық бар элементтер басқармасына панорамалы компас қосылады, навигациядағы элементтерді ұстап алады. Осы компас арқылы 45° градустағы суреттерді көрсетіге болады және панараманың өзгерісіне байланысты картадағы суретіміз өзгереді.

- басқарма элементтері панорамалық және масштабты арасындағы айналмалы басқарма пайда болады. Соның көмегімен картаның айналуын бақылауға және өзгертіге мүмкіндік көрсетеді.

- спутникті және гибриді 45° градустағы суретке өзгертіледі центр ортасына байланысты. Бағдарлама басынан солтүстіктен бастайды. Масштабтың үлкен болуына қарай картаның режимі бастапқы күйге өзгеріледі.

- басқару элементі картаның типіне қарай, басқару элементінің бағдарлама 45° градустағы сурет түсіруге болатын шарты қосылады.

- масштабтың өсуіне байланысты 45° градустағы картадағы суретіміз түп нүктесіне жеткенде тоқтап қалады және карта бастапқы күйге келеді. Google күнделікті 45° градустағы суреттерді еңгізеді және жаңа қалаларды көрсетіп отырады. Ол ақпаратты толығымен білгіні келсе Google maps арқылы біле аласыз.

Нақты болған 45° градустағы суреттер, олар суреттердің жиынтығын құрайды (солтүстік, оңтүстік, шығыс, батыс). Егер сіз суретке қарайтын болсаңыз сіз 45° градустағы суреттің қай бағыта түсіргенін анықтай аласыз себеді күннің түсуіне байланысты көрінеді.

Анализден өткізілген карталардың типтері.

JavaScript API Google Карт (версия 3) қазір өзгертілген карталардың барлық типтерін және оларды басқара отырып, өзіңіз түсірген суреттерді қабылдай береді.

API (версия 3) картаның типтеріне қарай реализацияға картаның бірнеше түрдерін көрсетеді:

- стандарт фрагментіне сай, суреттері бар, оларды геофикалық картаға сай етіп формироваксын жүргізеді. Осы фрагметтер базалық типті фрагметтерге жатқызуға болады. Осындай картаның типтері ROADMAP, SATELLITE, HYBRID және TERRAIN қанағаттандырады. Массивті mapTypes типіне толығымен анализден өткен типтің түрін беріп өтеміз.

- фрагменттерге суреттерді белгілеу үшін, базалық типтегі картаны белгілеуіміз қажет. Заңға сүйене отырып осындай типтегі карталар басқа немесе бағдарлама суреттерді қабылдай алады. Тағыда осы фрагметтер түссіз болуы мүмкін, сол уақытта объектіні картаға еңгізуімізге болады.

- картаның типтері суретсіз, болған жағдайда оны ең төменгі деңгейде көрсетеді.

- осы шарттарды біз тек қана класстың көмегі арқылы іске асыра аламыз, ос клас түрі mapTypes. Соған орай тағыда ImageMapType өзінің үшінде бірнеше орнатылған функциялар түрін көрсетеді. Осының көмегі арқылы сізге суреттерді жеңіл және тез арада көруіңізге мүмкіндіктер береді.

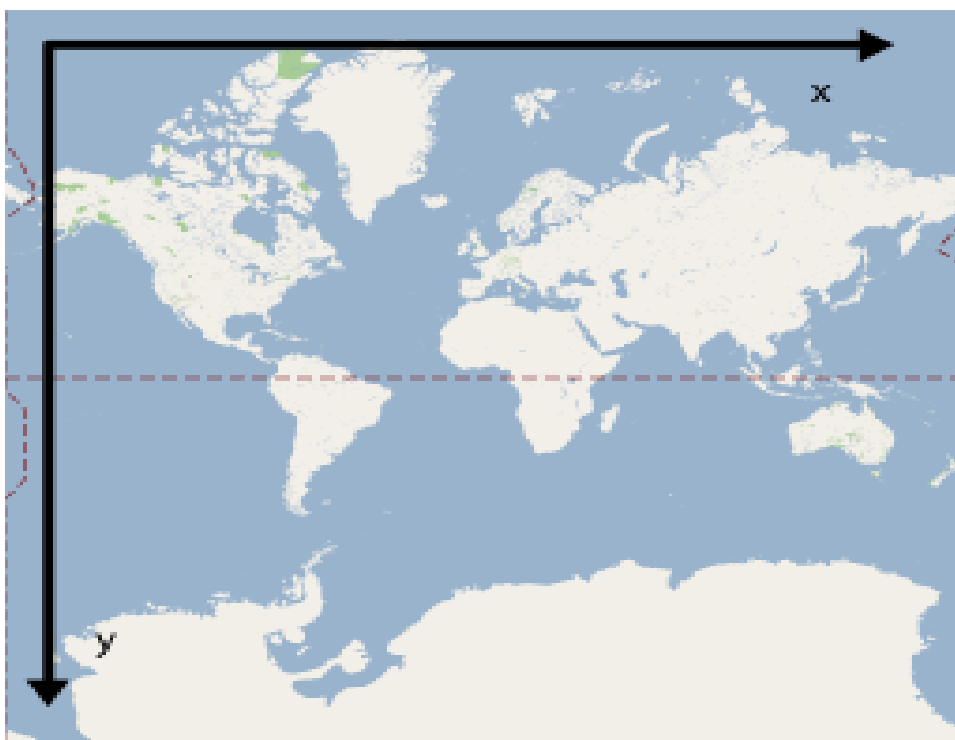
Картадағы координаталар.

API Google картасында келесі системалық координаталарды қолданады:

- кеңдіктің және ұзақтықтың мағыналарын жербетіндегі нүктелер арқылы анықтайды. (Google WGS84 координат дүниежүзілік геодезиялық жүйесін пайдаланады.)

- дүние жүзілік координаталар, картадағы нүктелер арқылы анықталады.

- координаталар, картадағы нақты фрагметтерді масштабтың дейгейіне байланысты анықтайды.



Сурет 2.5- Google map дүние жүзілік суреті

2.3 Деректер моделінің анықтамасы

Жасалатын бағдарламамыз ең басты жеңіл интерфейстан және сақтауды тез орындауы қажет. Қолданушыға өзінің мәліметтерін деректерге сақтауға және оны жойуға мүмкіндік беруі қажет.

Android бағдарламасы SQLite деректер қорын көруге болатын және SQLiteOpenHelper классы деректер қорын басқаруға мүмкіндік береді. Ол абстрактылық класс, оларға біз төменгі класс onCreate () және onUpgrade () методын қолдануымыз қажет.

Деректер қорының SQLite фактілері.

SQLite фактілері:

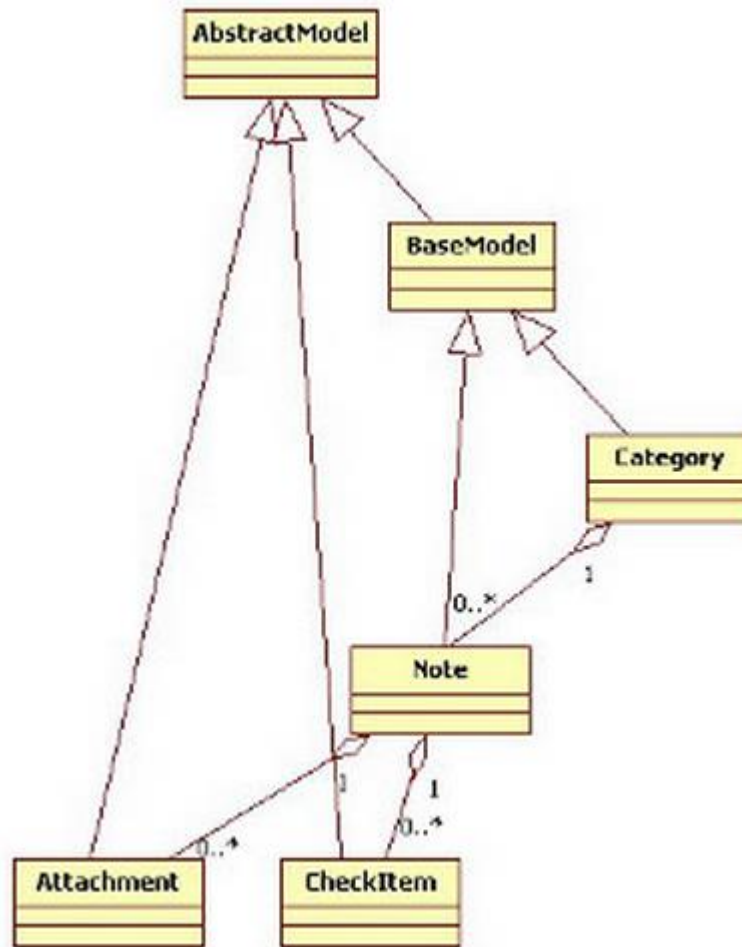
- SQLite аққысыз және деректер қорының ядросы ашық кодты SQL;
- SQLite оқылады және жазылады, ең қарапайым дискілерде;
- деректер қорының форматы кросс-платформалы болып келеді;
- ол стандартты деректер қорының функцияларын, синтаксисті қоса отырып, мәміле және баптаулы өтініштерді қанағаттандырады.

Деректер қорының структурасы.

Біздің деректер қорымызда тек бірғана кестеміз болады және келесі бағандарымыз заметканы сақтап отырады:

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

Осы класс-схемасы суретте көрсетеді.



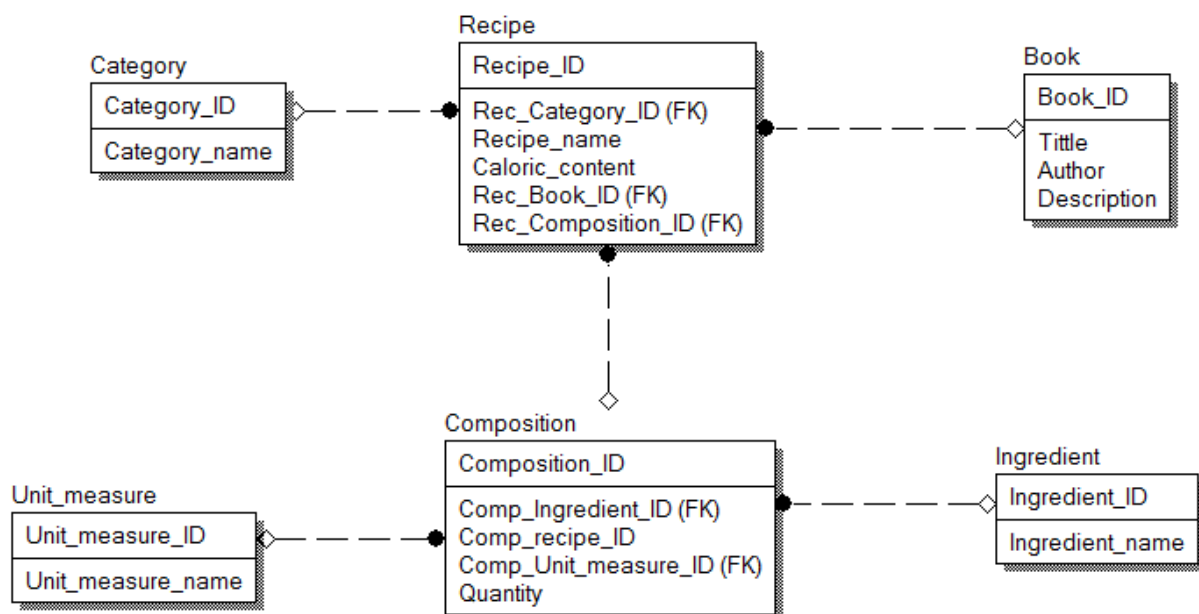
Сурет 2.6- Деректер қорының жұмыс істеу принципі

2.3.1 Даталогикалық проектилеу

Даталогикалық немесе логикалық деректер қорын проектилеу, - зертелген деректер қорының терминында және даталогикалық деректер қорының моделінде көрсетілген.

Даталогикалық модель ДҚБЖ есебімен жүзеге асатын құралдар, стандартты ұстап тұрады SQL:1999 [3]. SQL стандарты жаңартылған, оның ескі стандартын қолданудан алып тасталды.

2.5-суретте ER – диаграммасының Microsoft Visio 2007 нотациясы көрсетілген. PK — primary key —бастапқы кілт, бір идентификацияны белгілеу үшін қолданылады.



Сурет 2.7 – Толық ER – диаграммасы

2.3.2 Физикалық проектилеу

Бұл физикалық кезеңде деректер қорын проектилеу, негізінде деректер қорына байланысты бағдарлама жасауға бағытталған. Индекс – деректер қорының объектісі, сұраныстарды артыруға көмектеседі және сапалығын арттырады. Кестедегі деректер қоры көптеген бағандары болуы мүмкін, оларға белгілі шартта сақталуы мүмкін. Индекс негізінде бір немесе бірнеше бағандар кестесінен және кестеге көрсететін бағандардан тұруы мүмкін.

Деректер қорының кестесінде көптеп қолданылатын движок MyISAM (сақтау жүйесі). MyISAM кестесі WWW арналған осы аралықта тез жұмыс істейді және басқа орталарда, сұраныстарды оқиды. MyISAM типтегі кесте сұраныс кезінде өтте үлкен жылдамдықты көрсетеді. Көбінен ол транзакцияның жоқтығына байланысты және ішкі кілттерге. Осы сақтау жүйесін көптеген ДҚБЖ қабылдайды және бағдарлама жасауға оптималды жүйе. 2.6-сурет сұраныстардың процессі деректер қорына ұсынылған жолы көрсетілген. Осы процессте тағы операциялық жүйе қосылған, ол «көпір» ретінде жұмыс атқарады.



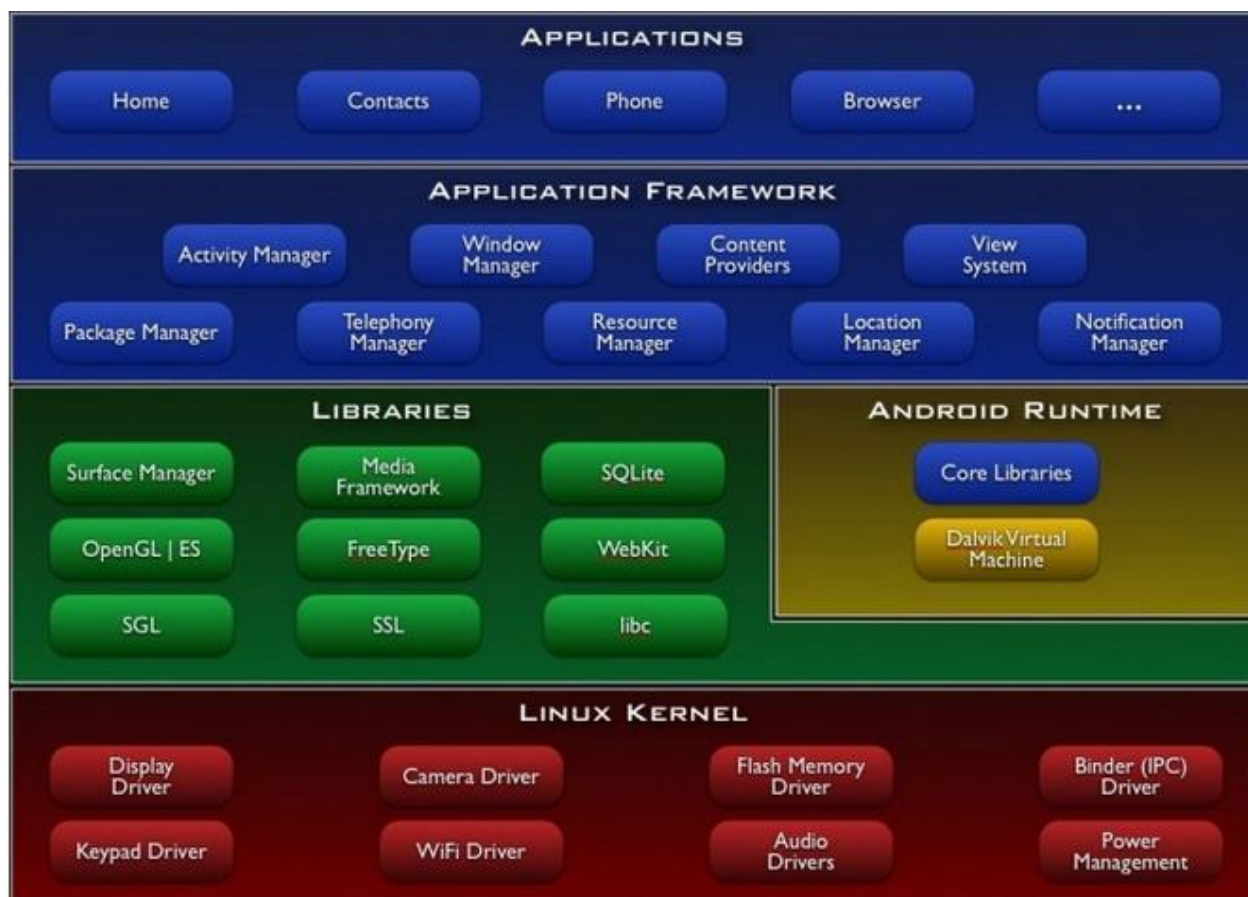
Сурет 2.8 - Сұраныстардың процесі деректер қорына ұсынылған жолы көрсетілген

3 Техникалық бөлім

Осы бөлімде бағдарлама жасау сипаттамасы келтірілген: бағдарлама архитектурасының сипаттамасы және шаблонды проектилеудің қысқаша түсініктемесімен, жасалатын бағдарлама моделінің, жасалатын интерфейсін сипаттау және жасалатын процессті қысқаша сипаттау.

3.1 Бағдарлама архитектура

Бағдарлама архитектурасы ретінде бірнеше көріністерді көріге болады. `hide` көрінісі алдында Android операциялық жүйесінің архитектурасы стек ретінде көрінеді. Android бағдарламасының құралы болып (Android SDK) бірнеше құралдары бар және интерфейсі бағдарлама жазуға негізделген (API), Android платформасына бағдарлама жазуға қажетті, Java тілін қолданады. Бағдарлама жазу кезінде Eclipse қабықшасын қолданамыз. Eclipse қабықшасының ішінде Android SDK, SDK Manager және AVD Manager стандартты бағдарламасы бар.

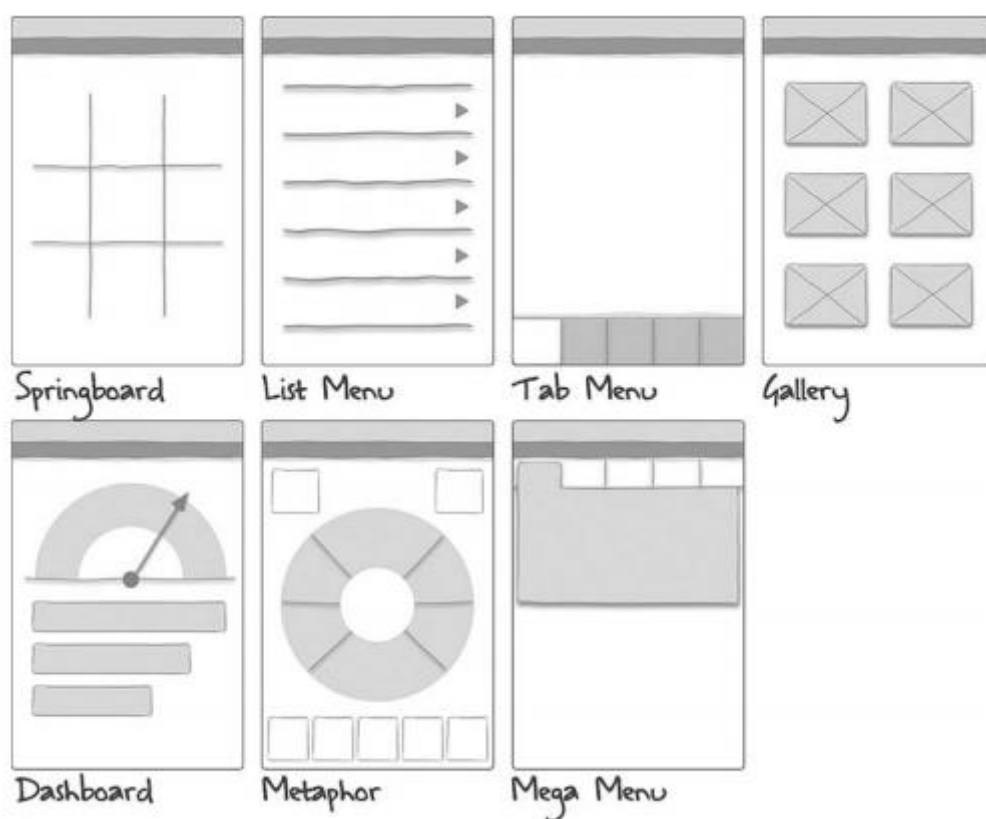


Сурет 3.1 – Android негізгі компоненттары

3.2 Қолданбалы интерфейсін зерттемесі

3.2.1 Навигация

Алғашқы навигацияның қалыптары. Жақсы навигация, ол проектилеудің ең басты жолы және көрінісі. Бағдарламаның навигациясы жақсы болса, онда оны интуитивті жеңіл түсініуге болады. Егер бағдарламаның интерфейсі жеңіл әре қызықты, түсінікті болсан, онда қолданушыға жұмыс істеу жеңілрек болады.



Сурет 3.2 – Навигацияның бастапқы шаблонны

Әр операциондық жүйеде вкладкалардың орналу режимі бар, соған байланысты мен өзімнің бағдарламанда Gallery атты вкладканы қолдандым. Ортасындағы вкладкалар әр класстарға жауап береді, әр класстың өзінің вкладкалары бар. Классқа байланысты мен бағдарламада бес вкладканы қолдандым, яғни класстың саны бес тең көп болады. Android операциондық жүйесі тек қана сенсорлы экранда тараған. Егер экранымыз сенсорлы болатын болса, онда сізге жұмыстеуге ыңғайлы болады. Сенсорлы экранды біз әр түрлі бағытта айналдыра аламыз вертикальды және горизонтальды бағыттарда айналдыра аламыз. Менім бағдарламанда сіз тек қана вертикальды түрде қозғалта аласыз.

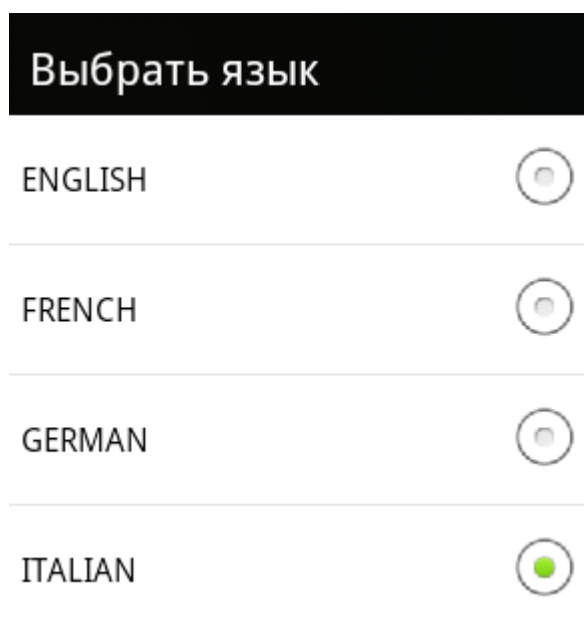
3.3- суретте бағдарламаның бастапқы беті көрсетілген.



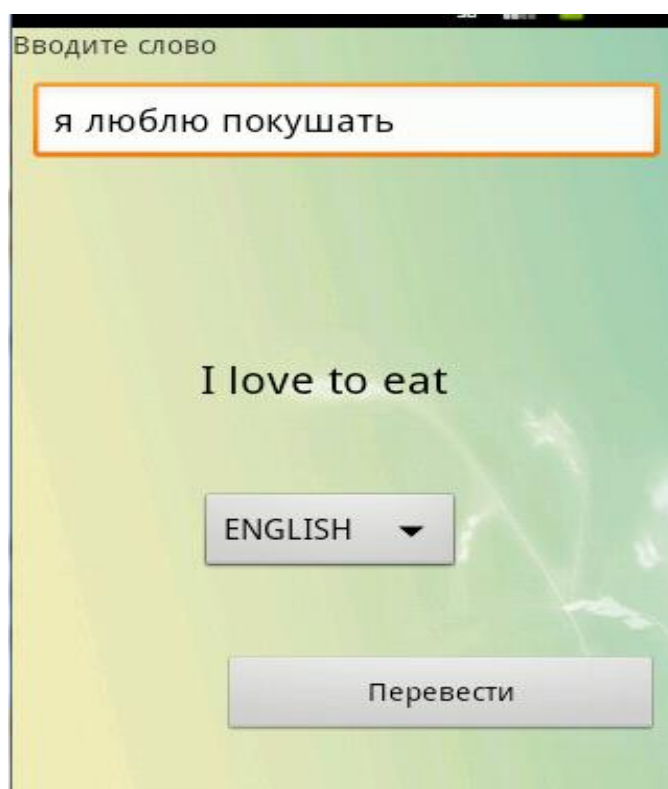
Сурет 3.3 –Бағдарламаның басты беті

3.2.2 Аудармашы беті

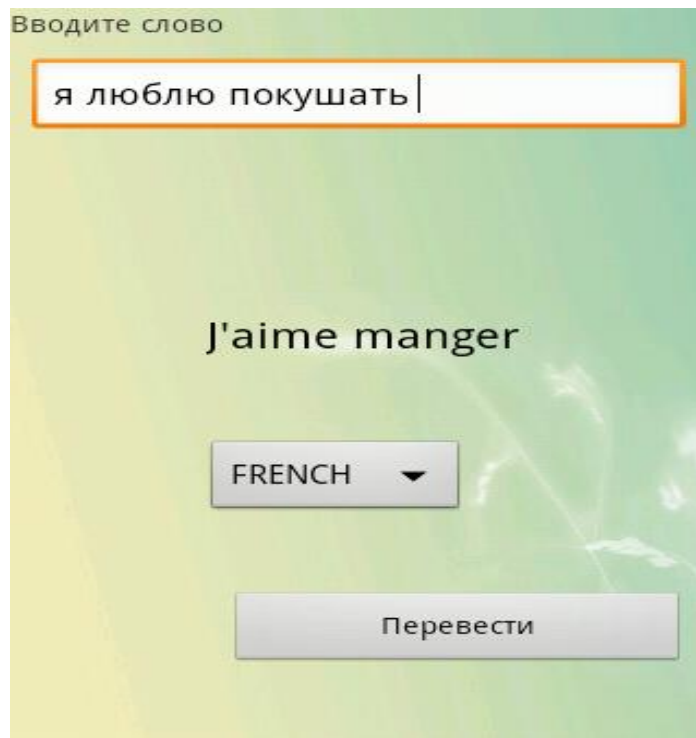
Аудармашы беті ең басты беттердің бірі. Осы бетте сіз орыс тіліндегі сөзді ағылшын тіліне, француз тіліне, неміс тіліне және итальян тіліне аударуыңызға болады. Сөзді және сөйлемдерді аударуға жеңіл және тез. Бағдарламаның бетінде, сіз төрт тілдің біреуін таңдай аласыз.



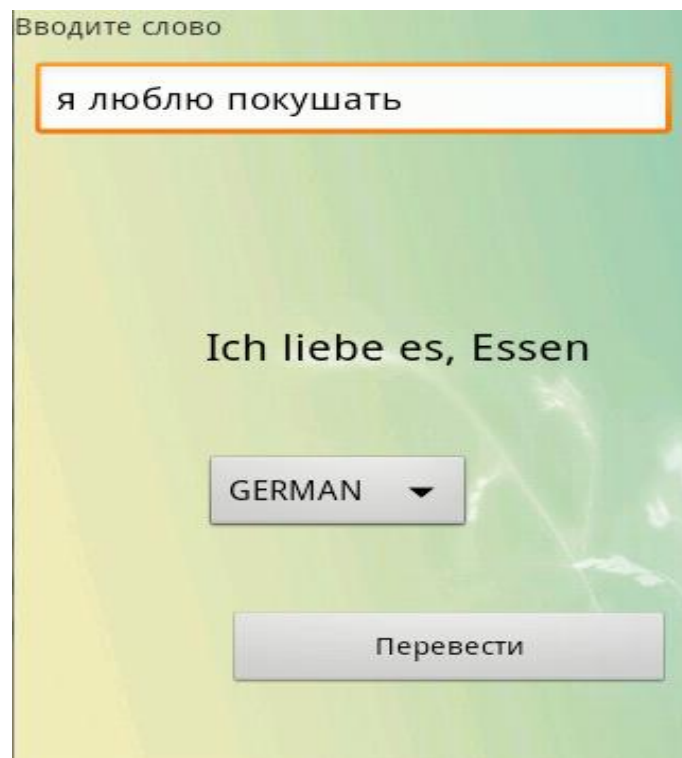
Сурет 3.4 – Тіл таңдау беті



Сурет 3.5 – Ағылшын тіліне аударылған сөз



Сурет 3.6 – Француз тіліне аударылған сөз



Сурет 3.7 – Неміс тіліне аударылған сөз



Сурет 3.8 –Итальян тіліне аударылған сөз

3.2.3 Саяхатшыларға көмек беру беті

Қазіргі уақытта саяхатқае шығатын адамдардың саны жылдан жылға көбейіп келе жатыр, соған байланысты саяхатшылар, туризімге шығар алдында бірнеше түсініктеме беріп өту қажет. Ол саяхатшының өзімен қандай құжаттары болуынан бастап аэропорт, отель, медицина, дүкен, төл-құжат және қоғамдық орында өзін ұстау тәртібін білуі қажет.

В отеле

Медицина

Магазины

Паспорт

Куда нужно обращаться и
какие действия
предпринимать

Как себя вести в
непредвиденных ситуациях

Правила поведения на воде

Сурет 3.9 –Туристке көмек беру беті

Медицина

Перед поездкой необходимо оформить международную медицинскую страховку. При наступлении страхового случая необходимо в первую очередь связаться с офисом страховой компании, указанном в полисе. Оператор направит к вам необходимого специалиста или назовет адрес ближайшего медицинского учреждения. Сертификат о прививках не требуется.

Сурет 3.10 –Медициналық бөлімі



Сурет 3.11 –Паспорт жайлы мәліметтер бөлімі

3.2.4 Google Maps саяхатшының орнын анықтау

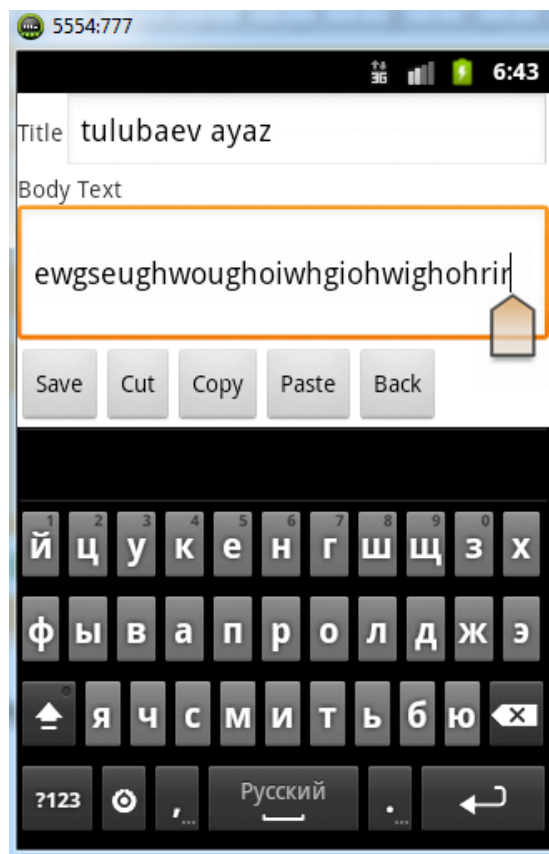
Мобильдік бағдарлама жасаушыларға Google компаниясы өзінің Google maps картасын ұсынады. Google maps картасының көмегімен сіз қай жерде келе жатқаныңызды тез анықтауға мүмкіндік береді және қай көшеде келе жатқаныңызды көрсетіп тұрады. Тағыда осының көмегімен сіз бір орыннан басқа орынға GPRS ретінде қолдануыңызға болады.



Сурет 3.12 – Google maps навигациясы

3.2.5 Саяхатшының қажетті мәліметін сақтау беті

Саяхатқа шыққан кезде әр адам өзінің қай жерге баратының анықтайды, одан кейін баратын жерлерін, сатып алатын заттарын бір жерге жазып алады. Соған байланысты жасалатын бағдарламада мен заметка жасап қойдым, соның көмегімен саяхатшының өзінің керекті мәліметтерін сақтап қоюға мүмкіндік береді.



Сурет 3.13 – Мәлімет сақтау бет

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.

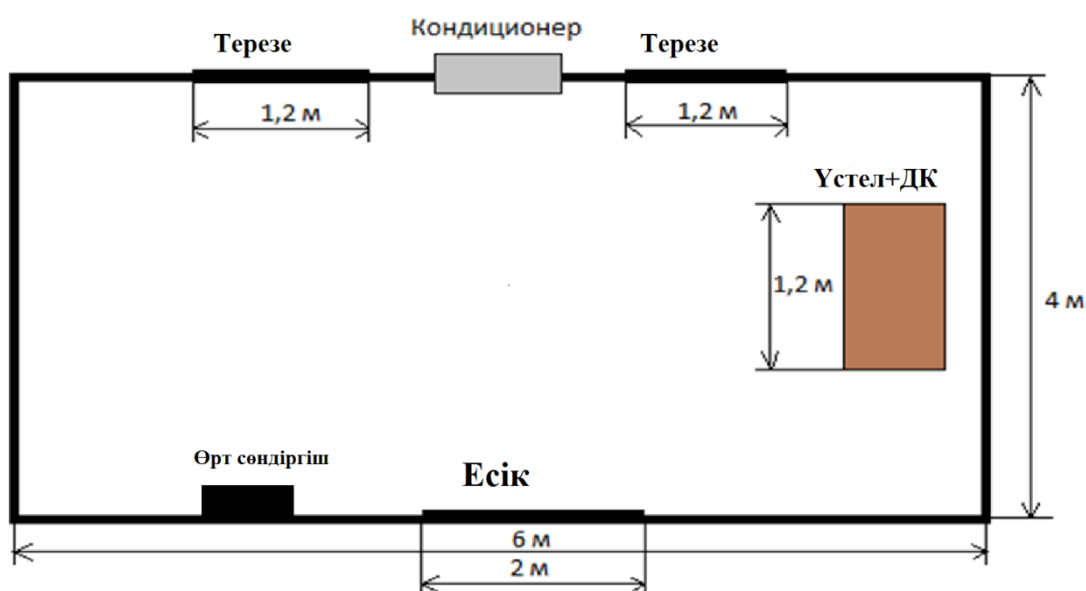
Бағдарлама жасау кезінде дербес компьютерді және электронды құрылғыларды қолданамыз. Қарастырылып отырған бөлмеде 1 оператор жұмыс атқарады, оның өзінің жеке жұмыс істеу орны бар.

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 5.1-суретте көрсетілген.



Сурет 5.1 – 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;
- бөлmemіз 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 лампы;
- 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 бөлmemіз 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:

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

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

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

p_{np} -бөлмеге келетін ауаның тығыздығы, тең $1,2 \frac{кг}{M^3}$;

$t_{\text{выт}}$ - 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 (кондиционер), 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 артықшылығына $5 - 7 \text{ } ^{\circ}\text{C}$ 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, жылынған 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_q - 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 100^0 C	егер 350^0 C	егер 100^0 C	егер 350^0 C
Жеңіл	180^0 C	145^0 C	150^0 C	5^0 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 коэффициенті, тең болса $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^0\text{C}}$;

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

k_n – қабатқа жылу берілу коэффициенті, $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^0\text{C}}$.

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 6 - 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 $140 \frac{\text{Вт}}{\text{м}^2 \cdot ^0\text{C}}$. 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_{yx} , кВт есектен және қабырғадан, терезені формула арқылы бағалайды:

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

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

S – аудан, м^2 ;

δ – қабырғаның қалыңдығы, м.

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

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

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:

$$Q_{\text{изб}} = 1120,3 \text{ кВт.}$$

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 райдың 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 им/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 $L=6$ м, ені $B=4$ м, биіктігі $H=3$ м. Жұмыс бетінің еден деңгейінен биіктігі $h_{pp}, h_{pp}=0,725$ м, терезелер $h_{но}, h_{но}=0,8$ м биіктігінен басталады, терезелердің биіктігі $h_o, h_o=1,5$ 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, m^2 , осы формуламен анықтаймыз:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_H \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3, \quad (4.10)$$

$$S_0 = \frac{S_n \cdot e_H \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3 \quad (4.11)$$

мұндағы S_n – бөлменің ауданы m^2 ; $S_n = 24 m^2$;

e_H – Ортақ жарықтандыру жарық көздері 4.3- кестесінен тандаймыз.

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

m_N - климаттың жарықтанукоэффициент, 4.3- кестесінен тандаймыз.

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

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

$m = 0,7$;

$c = 0,75$ (қабырғаның сыртқы жағы);

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

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

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 :

$$l = B - 1,$$

$$l = 4 - 1 = 3 \text{ м},$$

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

$$h_{\text{расч}} = h_{\text{но}} + h_o - h_{\text{рп}} \quad (4.13)$$

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

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

$\eta_0 = 10$ 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 $\eta_1 = 1,05$.

$k_{\text{зд}}$ – 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 $S_{\text{сн}} = 8,16 \text{ м}^2$.

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 c 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 1,5x1,2м 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 максимальды 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

КСЕТ – бұл саяхатшыларға арналған мобильдік бағдарлама. Жасалып отырған проекттің мақсаты туристерге арналған бағдарлама жасау. Сонда саяхатшылардың қолында аудармашы және саяхатшының орнын анықтайтын навигациясы болады. Зертеулер бойынша, таңдаған тематикама байланысты бағдарлама жасалмаған және еш қандай бәсекелістік жоқ. Мобильді бағдарлама алғашқы уақытта 4 фирмаға сатылады деп жоспарланады, себебі мобильді бағдарламаның тиімділігін және аз уақыт ішінде өзін-өзі ақтайтына байланысты сұранысқа ие болады деп ойластырылуда.

Тұтас сегмент – бұл қазіргі мобильдік прогрестің мүмкіндігін толық пайдаланып жұмыс істеуді қажет ететін бағдарлама жүйесі.

К е с т е 5.1 – Қызметті SWOT-талдау

Артықшылықтары	Кемшіліктері
Бұл бағдарламаны саяхатшылар компаниясына әлі еңгізбеген;	Саяғатшылар компаниясының шектеулі болуы
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.

Айта кету керек, стратегия тек компанияның белгілі бір мақсатта нарықтың кез келген сегментінде қозғалуына арналған ұмтылысы және стратегия жалпы, бөлінбейтін сипатқа ие. Жағдайды басқару мен оны назардан тыс шығармау үшін тек нақты орындалу уақыты анықталған жиындарды іске асыру керек, осы жиындарға жауапты адамдарды, қажетті ресурстарды (қаржылық, адами және т.б.) басқару қажет. іске асыру қажет. Міне сондықтан таңдалған стратегия үшін маркетинг-микс деп аталатын арнайы жиындар іске асырылады. Осы маркетингтің тактикалық деңгейі болып табылады.

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.

К е с т е 5.2– Маркетинг стратегиясының кестесі

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Product	
Саяхатшыларға арналған мобильді бағдарлама қызметі	Жалпы тапсырманы шешу келесі бөлімдерден тұрады: 1)Пайдаланушыға қолайлы интерфейс жүйесін ұйымдастыру. 2) Пайдаланушы арнайы жасалған мобильді бағдарламаға кіріп өзіне қажетті ақпаратты ала алады.
Price	
Базалық бағасы	Саяхатшы компаниясының ерекшеліктерін ескере отырып ақысы алынады.
Promotion	
	Басында уақытша жарнамасыз жұмыс істеу, яғни тікелей сату
Place	
Фирма 1, фирма 2, фирма 3, фирма 4	Алматы, Астана қалаларындағы саяхатшы компаниялармен келіссөз жүргізу

5.2 Бағдарламалық қамтаманың 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 1800-ге тең, [12] таңдап алуға болады.; c-бағдарлама күрделілігінің коэффициенті, ол менде 1 ге тең, себебі менің жобамның бағдарламалық тілі жоғарғы тілге жатады және күрделілік тобы үшінші топ, ал жаңалық дәрежесі бойынша В тобына,яғни типтік шешім қолданған бағдарламаларды дайындау тобына жатқандықтан осы мәндердің қиылысуынан арнайы кестеден алынды; Осы әдіспен базалық көрсеткішті табамыз:

$$Q = 1800 * 1 = 1800$$

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

$$T_{PO} = 24 \text{ адам / сағ.}$$

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

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

$$T_O = 1800 * 1,5 / 50 * 0,8 = 67,5 \text{ адам/сағ.}$$

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:

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

$$T_A = 1800 / (50 * 0,8) = 45 \text{ адам/сағ.}$$

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

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

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

$$T_H = 1800 * 1,5 / (50 * 0,8) = 67,5 \text{ адам/сағ.}$$

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

$$T_{II} = Q / 50 = 36 \text{ адам/сағ.} \quad (5.5)$$

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

$$T_{OT} = Q \times 4,2/50 \times K = 1800*4,2/50*0,8 = 189 \text{ адам/сағ} . \quad (5.6)$$

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:

$$T = T_{ПО} + T_{ГО} + T_A + T_{БС} + T_H + T_{П} + T_{OT} + T_D, \quad (5.7)$$

$$T = 24 + 67.5 + 45 + 45 + 67.5 + 36 + 189 + 24 = 498 \text{ адам/сағ.}$$

5.3 Бағдарламалық қамтаманы өңдеуге 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 498–ге тең.

Осылайша негізгі еңбек ақы:

$$З_{осн} = 498 * 2495,75 / 21 = 59179 \text{ теңге.}$$

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

$$З_{доп} = 0,2 * З_{осн} = 11835.8 \text{ теңге.} \quad (5.9)$$

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

$$ФОТ = З_{доп} + З_{осн} = 11835.8 + 59179 = 71014.8 \text{ теңге} . \quad (10)$$

5.4 Бағдарламаның 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 $C_{обор} = 194900$ 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:

$$C_{МжК} = 0,015 \times C_{обор} = 0,015 * 194900 = 2923,5 \text{ теңге}, \quad (5.11)$$

$$C_{ТО} = 0,025 \times C_{обор} = 0,025 * 194900 = 4872,5 \text{ теңге}.$$

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:

$$N = T/24 = 498/24 = 20.75 \text{ күн}.$$

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

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі:

$$t = T_A + T_{BC} + T_H + T_{II} + T_{OT} = 45 + 45 + 67.5 + 36 + 189 = 382.5 \text{ адам/ сағ}.$$

Амортизация нормасы (5.13) формуласы бойынша анықталады:

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% . \quad (5.13)$$

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

$$C_{ликв} = 0,05 \cdot C_{обор} = 0,05 \cdot 194900 = 9745 \text{ теңге}. \quad (5.14)$$

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

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% = \frac{194900 - 9745}{4 \cdot 194900} \cdot 100\% = 23,75\% ,$$

$$A = \frac{C_{обор} \times H_A \times N}{100 \times 12 \times t} = 194900 \cdot 23,75 \cdot 20,75 / 100 \cdot 12 \cdot 382,5 = 209,25 \text{ теңге}.$$

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

$$C_{ЭЭ} = M \cdot k_3 \cdot F_{эф} \cdot C_{кВт-ч} . \quad (5.15)$$

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

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны, оның мәні ҚР – да 14,65 теңгеге тең;

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

$$F_{эф} = D_{ном} \cdot d \cdot (1 - f / 100) = 258 \cdot 8 \cdot (1 - 2,5 / 100) = 2012. \quad (5.16)$$

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

$d = 8$ жұмыс күнінің ұзақтығы [сағ];

$f = 2,5$ % – ЭВМ –ды жөндеуге кеткен жоспарланған уақыт.

Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны: $C_{ЭЭ} = 0,45 \cdot 0,8 \cdot 2012 \cdot 14,65 = 10611$ теңге.

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

$$\mathcal{E}_3 = t_{\Sigma} \cdot C_3 / F_{эф}, \text{ теңге}. \quad (5.17)$$

мұндағы $C_{\Sigma} = C_{\Sigma\Sigma} + C_{TO} + A_{жыл}$ – ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$C_{\Sigma} = C_{\Sigma\Sigma} + C_{TO} + A_{жыл} = 1061 + 4872,5 + 209,25 = 15693 \text{ теңге.}$$

мұндағы $F_{эф}$ – жұмыс уақытының эффективті фонды, (16) формуласы бойынша анықталды, $t_{\Sigma'}$ – ЭЕМ-нің (7) формуласы бойынша анықталған есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған:

$$t_{\Sigma'} = t_{пр} + t_{отл} + t_d = 45 + 189 + 24 = 258 \text{ сағ}$$

Сәйкесінше, ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$\mathcal{E}_3 = 258 * 15693 / 2012 = 2012 \text{ теңге}$$

ал уақыт коэффициенті мына формула бойынша есептеледі (5.18):

$$w = \frac{\mathcal{E}_3}{C_{\Sigma}} = \frac{2012}{15693} = 0,13 \quad (5.18)$$

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

$$\text{Электроэнергияға шығын: } C_{\Sigma\Sigma}^* = C_{\Sigma\Sigma} \cdot w = 10611 * 0,13 = 1379,4 \text{ теңге;}$$

$$\text{Амортизационды аударым: } A_{жыл}^* = A_{жыл} \cdot w = 209,25 * 0,13 = 27,2 \text{ теңге.}$$

Сонымен қатар, (5.19) формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар. Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], (5.19) формуласы бойынша анықталады:

$$CC = (\text{ФОТ} - 0,1 * \text{ФОТ}) * 0,11 = (71014,8 - 0,1 * 71014,8) * 0,11 = 7030 \text{ теңге} \quad (5.19)$$

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, (5.20) формуласы бойынша анықталады:

$$C_{накл} = 0,5 * \text{ФОТ} = 0,5 * 71014,8 = 35507,4 \text{ теңге} \quad (5.20)$$

5.5 Өзіндік құн нәтижесінің кестесі

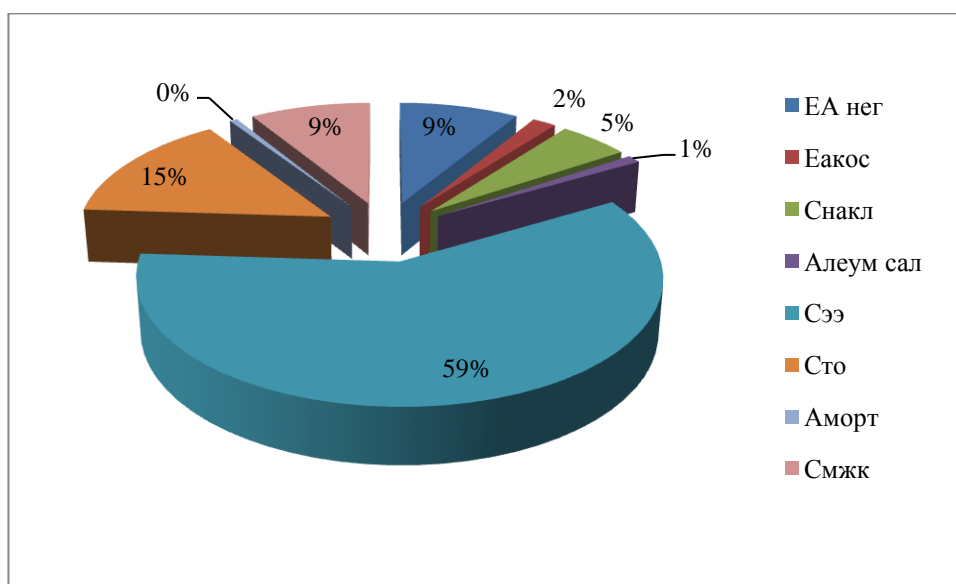
Бағдарламалық қамтаманы өңдеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру,

жүктеме шығындар мен материал және құрамалардың қосындысына тең.

К е с т е 5.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сумма, теңге	Жалпы суммадан процентпен
ФЗП	$Z_{осн}$	59179	48,2
	$Z_{доп}$	11835.8	9,64
Жүктеме шығындар, $C_{накл}$		35507.4	28,93
Әлеуметтік сақтандыру, $СС$		7030	5,72
Эксплуатационды Шығындар	$C_{ээ}^*$	1379.4	1,1
	$C_{ТО}$	4872.5	3,96
	$A_{жыл}^*$	27.2	0,022
Материалдар мен комплектілер, $C_{мик}$		2923,5	2,38
Барлығы:		122754.8	100

Өзіндік құнының диаграммасы:



Сурет 5.1 – Өнімнің өзіндік құнының диаграммасы

Бағдарламаның бағасына есептеу.

Өнімнің өзіндік құны 122754.8 теңге (жоғарыда кесте 1) болды, онда бір енгізу объектісіне арналған жүйенің толық өзіндік құны (C_{nc}):

$$C_{nc} = 122754.8 / 4 = 30688.7 \text{ теңге}$$

Минималды бағаны, кірісті және пайданы есептеу.

Өнімнің минималды бағасы келесі формуламен есептеледі:

$$C_{\min} = (C_{\text{пс}} + C_{\text{мд}} + C_3)(1+r/100) = (30688.7 + 200 + 150) * 1,21 = 37556.8 \text{ теңге.} \quad (5.21)$$

мұндағы $C_{\text{мд}}$ - магнитті диск бағасы, қазіргі нарықтық бағасын қарастырып, 200 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%):

$$C_{\min_{\text{НДС}}} = C_{\min} + N_{\text{ндс}} C_{\min} = 37556.8 + 0,12 * 37556.8 = 42063.6 \text{ теңге.} \quad (5.22)$$

Кіріс келесі формуламен есептеледі:

$$V_{\text{реал}} = C_{\min_{\text{НДС}}} * V_{\text{объем}} = 42063.6 * 4 = 168254.4 \text{ теңге.} \quad (5.23)$$

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

Жалпы пайда келесі формуламен есептеледі:

$$П = V_{\text{реал}} - V_{\text{реал}} * N_{\text{ндс}} - C_{\text{разраб}} = 168254.4 - 20190.52 - 122754.8 = 25309.1 \text{ теңге.} \quad (5.24)$$

мұндағы НДС – қосылған құн салығы (12%).

Таза пайда келесі формуламен есептеледі:

$$П_{\text{чист}} = П - N_{\text{прибыль}} = 25309.1 - 0,2 * 25309.1 = 20247.28 \text{ теңге.} \quad (5.25)$$

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

5.6 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу

Кәсіпорындардың бірмезгілдегі шығындары келесі бөлімдерден тұрады:

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

Жүйе құны 42063.6 теңге (22 формуласы).

Көлік шығындары жүйе құнының 20% құрайды:

$$42063.6 * 0,2 = 8412.72 \text{ теңге.}$$

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 2500-3000 теңгедей болып табылады [15].

Соған сәйкес оқу бағасы:

$$2500*2=5000 \text{ теңге.}$$

Кестесе 5.4 – кәсіпорынның бірмезгілдегі шығындарын есептеулерінің нәтижесі көрсетілген.

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.

Есептеу жылына бір рет саяхатшы гидіне 2500 теңге/сағ (жоғарыда берілген) сағаттық мөлшерде бір сағат көлемінде жүргізіледі.

$$2500*1=2500 \text{ теңге.}$$

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

$$(2500-2500*0,1)*0,11=247,5 \text{ теңге.}$$

Электрэнергияға кететін шығындар:

$$0,8*14,65= 11,72 \text{ теңге.}$$

мұндағы 0,8 – жұмсалған қуат мөлшері, кВт-сағ;
14,65 – ҚР 1 кВт-с электрэнергияның орташа құны, теңге.
Есептеулерді 5.5 – кестеге енгіземіз.

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

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	2500
Әлеуметтік салыққа аударымдар	247,5

К е с т е 5.5 –Жалғасы

Электроэнергияға төлем	11,72
Қорытынды:	2759,2

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.

Саяхатшы гидіне келесі шығындар түрлері есептеледі:

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

Саяхатшы гидінің еңбек өнімділігін талдау үшін 2560 теңге/сағ (ұйымның қойып отырған талабы) мөлшерінде жұмыс істейтін саяхатшы гиді қажет. Өткізу қабілетін есептеу үшін саяхатшы гиді күніне 8 сағаттан, 5 күн жұмсайды. Есептеу барысында еңбектің жылдық төлем қоры:

$$\text{ФОТ} = 8 * 5 * 2560 = 102400 \text{ теңге.}$$

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

$$(102400 - 102400 * 0,1) * 0,11 = 10137 \text{ теңге.}$$

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

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

Еңбекке төленетін қор	102400
Әлеуметтік салыққа аударымдар	10137
Қорытынды:	112537

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 2759,2 теңге болды. Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығын 112537 теңге болды.

Осыған сәйкес, шығындардың шартты үнемделуі мынаған тең болады:

$$C_{ЭК} = 112537 - 2759,2 = 109777,8 \text{ теңге.}$$

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	109777,8
Ағымдағы шығындар, теңге	2759,2
Шартты пайда, теңге	107018.6
Пайдаға салынатын салық, теңге	21403.72
Таза пайда, теңге	85614.88

Ақшалай құралдардың қозғалысы.

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

Аты	Жылдар		
	2014	2015	2016
Бірмезгілдік шығындар	-55476,32		
Жеке кәсіпкерліктен пайда		85614.88	85614.88
Шығындарды үнемдеу		109777,8	109777,8
Ағымдағы шығындар		2759,2	2759,2
Пайдаға салынатын салық		21955,56	21955,56

Таза табыс	-55476,32	87822,24	87822,24
------------	-----------	----------	----------

К е с т е 5.8 – Жалғасы

Дисконттау коэффициенті (25%) мөлшерінде)	1	0,8	0.64
Таза дисконтталған табыс (ТДТ)	-55476,32	70257,792	56206,2336
ТДТ нәтижесімен өспелі	-55476,32	14781,472	70987,7056

$$K_{д2} = 1/(1+\alpha)^1 = 1/(1+0,25)^1 = 0,8,$$

$$K_{д3} = 1/(1+\alpha)^2 = 1/(1+0,25)^2 = 0.64.$$

мұндағы α – инфляция коэффициенті:

$K_{д}$ - дисконттау коэффициенті.

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

Таза дисконтталған табыс:

$$(ТДТ) = 87822,24 * 0,8 = 70257,792 \text{ теңге,}$$

$$87822,24 * 0,64 = 56206,2336 \text{ теңге.}$$

$$ТДТ_2 \text{ (өспелі нәтижесімен)} = ТДТ_1 + \text{Таза дисконтталған табыс (ТДТ)}_2$$

5.10 Экономикалық тиімділікті есептеу

Таза ағымдағы құндылықты есептеу (Net present value, NPV).

Шығындары біремізгілде тек жобаның басында күрделі салымдар (C_0) ретінде іске асатын жобалар үшін NPV келесі формуламен есептеледі:

$$NPV = \sum_{i=1}^n \frac{B_i}{(1+r)^i} - C_0. \quad (5.26)$$

мұндағы B_i – i -ші жылдағы жобадан алынатын пайда; -55476,32;

r – дисконттеу мөлшері.

$NPV(25\%) = 70257,792 - 55476 = 14781,472$ теңге.

$NPV(25\%) > 0$ сондықтан ұсынылып отырған жоба табысты. Сол себептен жобаны қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

Пайда индексін есептеу (Profitability index, PI).

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

$$PI = \frac{NPV}{C_0} \quad (5.26)$$

$$PI = (14781,472 / 122754.8) * 100\% = 0.12 * 100\% = 12\%.$$

Табыстың ішкі нормасын есептеу (Internal rate of return, IRR).

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.

$$IRR = r_1 + \frac{f(r_1)}{f(r_1) - f(r_2)} * (r_2 - r_1) \quad (5.27)$$

$r_b = 25\%$:

$$PV_1 = 87822,24 / (1 + 0,25) = 70257,792 \text{ теңге,}$$

$$NPV(25,0\%) = (70257,792) - 55476,32 = 14781,472 \text{ теңге,}$$

$r_1 = 0.2$

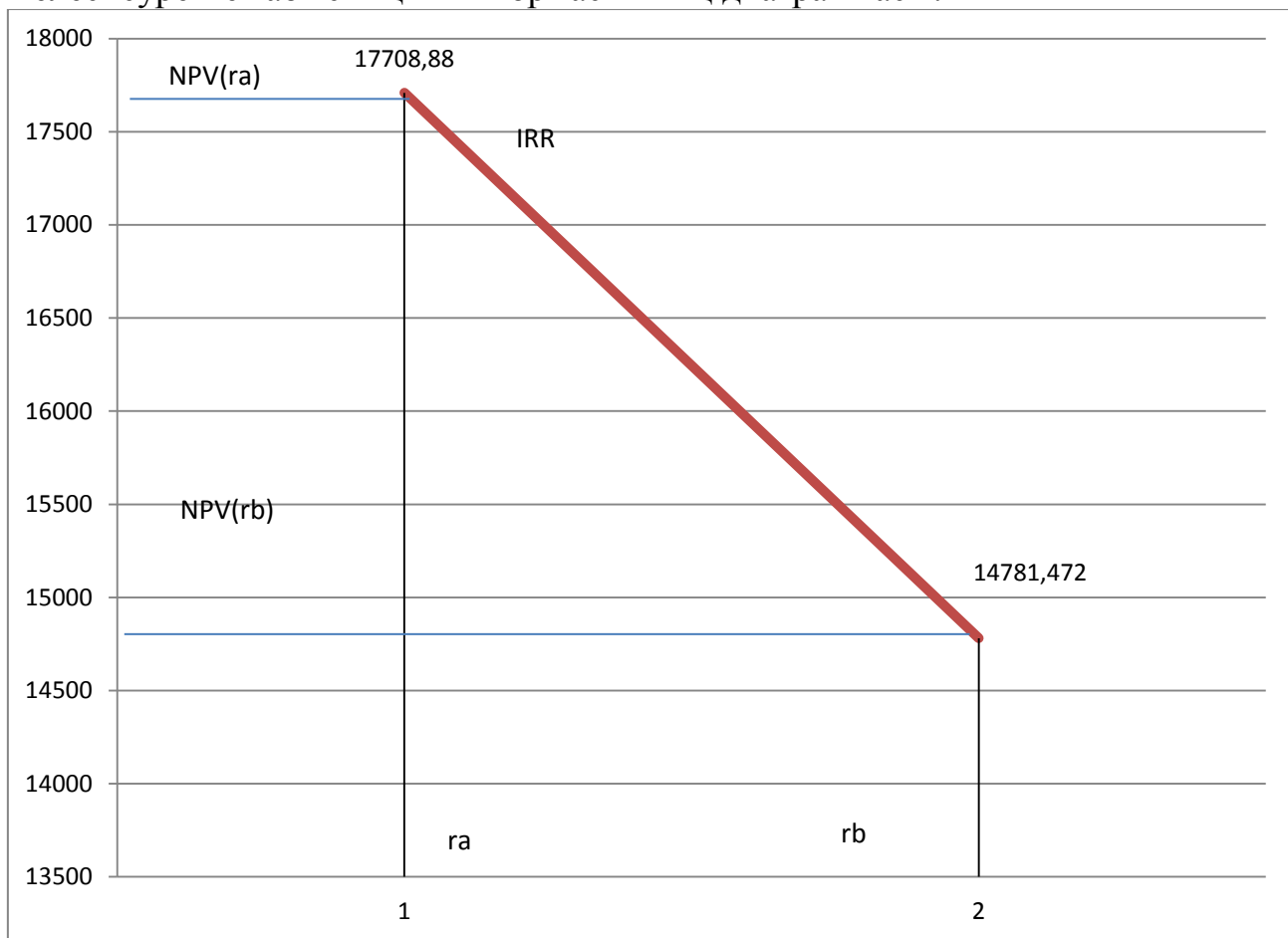
$r_2 = 0.25$

$NPV(20\%) = 17708,88$ теңге

$NPV(25\%) = 14781,472$ теңге

$$IRR = \left(0.2 + \frac{17708.88}{17708.88 - (14781.472)} * (0.25 - 0.2) \right) * 100\% = 50.2\%$$

Келесі суретте табыстың ішкі нормасынның диаграммасы.



Сурет 5.2 – Табыстың ішкі нормасынның диаграммасы

Қорытынды

Операциондық жүйені жұмысқа қажетті етіп, тек қана бағдарлама іске асырады. Бағдарлама түрлері интернетқа шығатын, веб-парақтарын көруге және тағы басқа заттарды көруге, соған сүйене мобильдік бағдарламаны көптеген функциялары бар, кішкентай компьютер қолдануға ыңғайлы. Бағдарламалардың бәрі әр түрлі платформаларға жасалуы мүмкін, яғни кросс платформалар арнайы жүйеде ғана жұмыс істейді. Алынған практикадан, өзімнің индивидуалды бағдарлама жасауым және жаңа мәліметтерді игеруі көп уақытты алады. Жасалған дипломдық жұмысымда клиенттік-серверлерге бағдарлама жасау және жоғары дәрежеде кодттау тәжірбиесін меңгердім.

Дипломдық жұмыстың мақсаты - Android операция жүйесіне негізделген саяхатшыларға арналған «КСЕТ» мобильдік бағдарламасын құрастыру. Жасалып отырған проектіде саяхатшыларға арналған бағдарлама жасадым. Сонда саяхатшылар өзінің уақытын үнемдейді және аудармашыны қажет етпейді. Зертеулер бойынша, таңдаған тематикама байланысты бағдарлама жасалмаған және еш қандай бәсекелістік жоқ.

Осы проектіге бағдарлама жасау процессінде Android операциялық жүйесіне жаңа анализдер жүргізілді, олар жаңа және тез жұмыс істейтін техниканы қажет етпейді. Алынған деректерге сүйене, сонында біз қандай технологиямен жұмыс істейтінімізді таңдаймыз. Қиындалған және маңызды проектілерді шеше отырып мен өзімнің бағдарламаның қажеттілігін және IT аумағына жаңа идеяларды еңгіздім іске асырдым.

Дипломдық жұмысқа берілген есебім толығымен толық көлемде орындалы.

Қолданылған әдебиеттер тізімі

1. Голощапов А. Google Android программирование для мобильных устройств. Санкт-Петербург 2011-438 с.
2. Брайан Х.,Филлипс Б. Программирование под Android. Питер 2014-592 с.
3. Амелин К.С., Граничин О.Н., Кияев В.И, Корявко А.В. Введение в разработку приложений для мобильных платформ. Санкт-Петербург 2011-505 с.
4. Лорен Д., Кондор Ш., Android программирование приложений под операционную систему Google 2011 – 464 с.
5. Бурнет Э., Привет, Андроид! Разработка мобильных приложений. Питер, 2012 – 256 с
6. Блог на хабре о разработке под Андроид [Электронный ресурс]
// URL: http://habrahabr.ru/blogs/android_development/.
7. Официальная справка для Android разработчиков [Электронный ресурс]
// URL <http://developer.android.com/index.html>
8. Официальная справка по среде программирования [Электронный ресурс]
// URL: <http://www.jetbrains.com>
9. Форум о программировании для Android [Электронный ресурс]
// URL: <http://www.cyberforum.ru/android-dev/>
10. 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.
11. Ф.Р. Жандаулетова. Безопасность жизнедеятельности. Методические указания к выполнению раздела выпускной работы бакалавров для студентов всех форм обучения специальностей. – Алматы: АИЭС, 2008. – 10 с.
12. Иванов И.Н. Экономика промышленного предприятия: Учебник. ИНФРА–М, 2011. – 305 с.
13. Рофе А.И. Экономика труда: Учебник. – М.: КноРус, 2010. – 400 с.
14. Остапенко Ю.М. Экономика труда: Учеб. пособие. – М.: ИЦ РИОР, 2010. – 160 с.
15. Базылов Қ.Б., Алибаева С.А., Нурмагамбетова С.С. Бітіруші жұмысының экономикалық бөлімі үшін әдістемелік нұсқаулар. – Алматы: АИЭС, 2009. – 25 б. Козырев В.М. Основы современной экономики: учебник-3-е изд. –М.: Изд-во Финансы и статистика, 2008. – 98 с.
16. Манилов, Р. Г. Бизнес-план: Методические материалы / Под ред. Р.Г. Маниловского. – М.: Изд-во Финансы и статистика, 2008. – 124с.

Бағдарлама А

Класстар құруға арналған скрипт

```
package com.ayaz.kcet; //googlemap
import android.app.Dialog;
import android.location.Location;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.view.Menu;
import android.widget.TextView;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnMyLocationChangeListener;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
public class ActivityFour extends FragmentActivity implements
OnMyLocationChangeListener {
    GoogleMap googleMap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Getting Google Play availability status
        int status =
        GooglePlayServicesUtil.isGooglePlayServicesAvailable(getBaseContext());
        // Showing status
        if(status!=ConnectionResult.SUCCESS){ // Google Play Services are not available
            int requestCode = 10;
            Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status, this, requestCode);
            dialog.show();
        }else { // Google Play Services are available
            // Getting reference to the SupportMapFragment of activity_main.xml
            SupportMapFragment fm = (SupportMapFragment)
            getSupportFragmentManager().findFragmentById(R.id.map);
            // Getting GoogleMap object from the fragment
            googleMap = fm.getMap();
            // Enabling MyLocation Layer of Google Map
            googleMap.setMyLocationEnabled(true)
            // Setting event handler for location change
            googleMap.setOnMyLocationChangeListener(this);
        }
    }
    @Override
```



```

public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.layout.googlemap, menu);
return true;}
@Override
public void onMyLocationChange(Location location) {
TextView tvLocation = (TextView) findViewById(R.id.tv_location);

```

А бағдарламасының жалғасы

```

/ Getting longitude of the current location
double longitude = location.getLongitude();
// Creating a LatLng object for the current location
LatLng latLng = new LatLng(latitude, longitude);
// Showing the current location in Google Map
googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
// Zoom in the Google Map
googleMap.animateCamera(CameraUpdateFactory.zoomTo(15));
// Setting latitude and longitude in the TextView tv_location
tvLocation.setText("Latitude:" + latitude + ", Longitude:" + longitude );
}}
package com.ayaz.kcet;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;
import android.webkit.WebView;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
public class Vieww extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
requestWindowFeature(Window.FEATURE_NO_TITLE); //скрываем заголовок
setContentView(R.layout.view);
//скрываем статусбар:
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
Bundle bundle = getIntent().getExtras();

```

```

String itemname = "n" + bundle.getString("defStrID"); //получаем строку и
формируем имя ресурса
Context context = getBaseContext(); //получаем контекст
//читаем текстовый файл из ресурсов по имени
String text = readRawTextFile(context, getResources().getIdentifier(itemname,
"raw", "com.ayaz.kcet"));
WebView myWebView = (WebView) findViewById(R.id.webview);
String summary = "<html><body>" + text + "</body></html>";
myWebView.loadDataWithBaseURL(null,text, "text/html", "utf-8",null);
//загружаем текст в webview myWebView.loadData(summary, "text/html", "utf-
8"); //загружаем текст в webview}
public static String readRawTextFile(Context ctx, int resId) //читаем текст из raw -
аргументы контекст и идентификатор ресурса {
InputStream inputStream = ctx.getResources().openRawResource(resId);

```

А бағдарламасының жалғасы

```

StringBuilder text = new StringBuilder();
try {

while (( line = buffreader.readLine()) != null) {
text.append(line);
text.append('\n');
}} catch (IOException e) {
return null;
}return text.toString();
}}
package com.ayaz.kcet;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;
import android.webkit.WebView;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
public class Vieww extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
requestWindowFeature(Window.FEATURE_NO_TITLE); //скрываем заголовок
setContentView(R.layout.view);
//скрываем статусбар:
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
Bundle bundle = getIntent().getExtras();
String itemname = "n" + bundle.getString("defStrID"); //получаем строку и
формируем имя ресурса
Context context = getBaseContext(); //получаем контекст
//читаем текстовый файл из ресурсов по имени
String text = readRawTextFile(context, getResources().getIdentifier(itemname,
"raw", "com.ayaz.kcet"));
WebView myWebView = (WebView) findViewById(R.id.webview);
String summary = "<html><body>" + text + "</body></html>";
myWebView.loadDataWithBaseURL(null,text, "text/html", "utf-8",null);
//загружаем текст в webview myWebView.loadData(summary, "text/html", "utf-
8"); //загружаем текст в webview}
public static String readRawTextFile(Context ctx, int resId) //читаем текст из raw -
аргументы контекст и идентификатор ресурса {
InputStream inputStream = ctx.getResources().openRawResource(resId);
InputStreamReader inputreader = new InputStreamReader(inputStream);
BufferedReader buffreader = new BufferedReader(inputreader);
String line;
StringBuilder text = new StringBuilder();
try {

```

А бағдарламасының жалғасы

```

while (( line = buffreader.readLine()) != null) {
text.append(line);
text.append('\n');
}} catch (IOException e) {
return null;}
return text.toString();}}
package com.ayaz.kcet;
import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ContextMenu.ContextMenuInfo;

```

```

import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.AdapterView.AdapterContextMenuInfo;
public class ActivityThree extends ListActivity {
private static final int ACTIVITY_CREATE=0;
private static final int ACTIVITY_EDIT=1;
private static final int INSERT_ID = Menu.FIRST;
private static final int DELETE_ID = Menu.FIRST + 1;
private NotesDbAdapter mDbHelper;
private Cursor mNotesCursor;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.mmmmain);
mDbHelper = new NotesDbAdapter(this);
mDbHelper.open();
fillData();
registerForContextMenu(getListView());
}
public void onBackPressed() {
moveTaskToBack(true);
this.finish();
return;}
private void fillData() {
// Get all of the rows from the database and create the item list
mNotesCursor = mDbHelper.fetchAllNotes();
startManagingCursor(mNotesCursor);
// Create an array to specify the fields we want to display in the list (only TITLE)
String[] from = new String[]{NotesDbAdapter.KEY_TITLE};
// and an array of the fields we want to bind those fields to (in this case just text1)
int[] to = new int[]{R.id.text1};

```

А бағдарламасының жалғасы

```

SimpleCursorAdapter notes = new SimpleCursorAdapter(this, R.layout.notes_row,
mNotesCursor, from, to);
setListAdapter(notes);}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
super.onCreateOptionsMenu(menu);
menu.add(0, INSERT_ID, 0, "Создать новую заметку");
return true;}
@Override

```

```

public boolean onOptionsItemSelected(int featureId, MenuItem item) {
    switch(item.getItemId()) {
        case INSERT_ID:
            createNote();
            return true;
    }return super.onOptionsItemSelected(featureId, item);}
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo
menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(0, DELETE_ID, 0, "Удалить");
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch(item.getItemId()) {
        case DELETE_ID:
            AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
            mDbHelper.deleteNote(info.id);
            fillData();
            return true;
    }return super.onOptionsItemSelected(item);
}
private void createNote() {
    Intent i = new Intent(this, NewText.class);
    startActivityForResult(i, ACTIVITY_CREATE);
}
@Override
protected void onItemClick(AdapterView l, View v, int position, long id) {
    super.onItemClick(l, v, position, id);
    Cursor c = mNotesCursor;
    c.moveToPosition(position);
    Intent i = new Intent(this, NewText.class);
    i.putExtra(NotesDbAdapter.KEY_ROWID, id);
    i.putExtra(NotesDbAdapter.KEY_TITLE,
    c.getString(c.getColumnIndexOrThrow(NotesDbAdapter.KEY_TITLE)));
    i.putExtra(NotesDbAdapter.KEY_BODY,
    c.getString(c.getColumnIndexOrThrow(NotesDbAdapter.KEY_BODY)));
    startActivityForResult(i, ACTIVITY_EDIT);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {

```

А бағдарламасының жалғасы

```

witch(requestCode) {
case ACTIVITY_CREATE:
String title = extras.getString(NotesDbAdapter.KEY_TITLE);
String body = extras.getString(NotesDbAdapter.KEY_BODY);
mDbHelper.createNote(title, body);
fillData();
break;
case ACTIVITY_EDIT:
Long rowId = extras.getLong(NotesDbAdapter.KEY_ROWID);
if (rowId != null) {
String editTitle = extras.getString(NotesDbAdapter.KEY_TITLE);
                А бағдарламасының жалғасы
String editBody = extras.getString(NotesDbAdapter.KEY_BODY);
mDbHelper.updateNote(rowId, editTitle, editBody);}
fillData();
break;
}}}
package com.ayaz.kcet;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;
import android.webkit.WebView;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
public class Vieww extends Activity {
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
requestWindowFeature(Window.FEATURE_NO_TITLE); //скрываем заголовок
setContentView(R.layout.view);
//скрываем статусбар:
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
Bundle bundle = getIntent().getExtras();
String itemname = "n" + bundle.getString("defStrID"); //получаем строку и
формируем имя ресурса
Context context = getBaseContext(); //получаем контекст
//читаем текстовый файл из ресурсов по имени
String text = readRawTextFile(context, getResources().getIdentifier(itemname,
"raw", "com.ayaz.kcet"));

```

```

WebView myWebView = (WebView) findViewById(R.id.webview);
String summary = "<html><body>" + text + "</body></html>";
myWebView.loadDataWithBaseURL(null,text, "text/html", "utf-8",null);
//загружаем текст в webview myWebView.loadData(summary, "text/html", "utf-
8"); //загружаем текст в webview

```

А бағдарламасының жалғасы

```

public static String readRawTextFile(Context ctx, int resId) //читаем текст из raw -
аргументы контекст и идентификатор ресурса

```

```

{
InputStream inputStream = ctx.getResources().openRawResource(resId);
InputStreamReader inputreader = new InputStreamReader(inputStream);
BufferedReader buffreader = new BufferedReader(inputreader);
String line;
StringBuilder text = new StringBuilder();

```

```

try {
while (( line = buffreader.readLine()) != null) {
text.append(line);
text.append("\n");
}} catch (IOException e) {
return null;
}return text.toString();
}}

```

```

package com.ayaz.kcet;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView; ///android.app.Activity
public class ActivityTwo extends Activity {
private ListView lv1; //даем название listview - lv1
private static final int STOPSPLASH = 0;
private static final long SPLASHTIME = 1800; //Время показа Splash картинки
private ImageView splash;
//Создаем массив из тем:
private String lv_arr[]={"Документы"," В аэропорту","В
отеле","Медицина","Магазины", "Паспорт","Куда нужно обращаться и какие

```

```

действия предпринимать", "Как себя вести в непредвиденных
ситуация", "Правила поведения на воде"},};
private Handler splashHandler = new Handler() { //создаем новый хэндлер
@Override
public void handleMessage(Message msg) {
switch (msg.what) {
case STOPSPLASH:
//убираем Splash картинку - меняем видимость
splash.setVisibility(View.GONE);
break;
}
super.handleMessage(msg);

```

А бағдарламасының жалғасы

```

public void onCreate(Bundle icle)
{
super.onCreate(icle);
setContentView(R.layout.mmain); //ставим слой
splash = (ImageView) findViewById(R.id.splashscreen); //получаем
идентификатор ImageView с Splash картинкой
Message msg = new Message();
msg.what = STOPSPLASH;
splashHandler.sendMessageDelayed(msg, SPLASHTIME);

lv1=(ListView)findViewById(R.id.lister); //получаем идентификатор ListView
lv1.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1 ,
lv_arr)); //устанавливаем массив в ListView
lv1.setTextFilterEnabled(true);
//Обрабатываем клик по пункту:
lv1.setOnItemClickListener(new OnItemClickListener() {
public void onItemClick(AdapterView<?> a, View v, int position, long id) {
//Позиция элемента, по которому кликнули
String itemname = new Integer(position).toString();
//Создаем новый intent
Intent intent = new Intent();
intent.setClass(ActivityTwo.this, Vieww.class);
Bundle b = new Bundle();
b.putString("defStrID", itemname); //defStrID - уникальная строка, отправим
itemname в другое Activity
intent.putExtras(b);
startActivity(intent); //запускаем intent
}});}}
package com.ayaz.kcet;

```



```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainActivity extends Activity implements OnClickListener {
    Button btnActTwo;
    Button btnActOne;
    Button btnActThree;
    Button btnActFour;
    Button btnActFive;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btnActTwo = (Button) findViewById(R.id.btnActTwo);
        btnActOne = (Button) findViewById(R.id.btnActOne);
        btnActThree = (Button) findViewById(R.id.btnActThree);
        А бағдарламасының жалғасы

        btnActFive = (Button) findViewById(R.id.btnActFive);
        btnActTwo.setOnClickListener(this);
        btnActOne.setOnClickListener(this);
        btnActFour.setOnClickListener(this);
        btnActFive.setOnClickListener(this);
        btnActThree.setOnClickListener(this); //btnActThree btnActThree
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btnActTwo:
                Intent intent = new Intent(this, ActivityTwo.class);
                startActivity(intent); // TODO Call second activity
                break;
            case R.id.btnActOne:
                Intent intent1 = new Intent(this, ActivityOne.class);
                startActivity(intent1); // TODO Call second activity
                break;
            case R.id.btnActThree:
                Intent intent2 = new Intent(this, ActivityThree.class);
                startActivity(intent2); // TODO Call second activity
                break;
        }
    }
}

```

```

case R.id.btnActFour:
Intent intent3 = new Intent(this, ActivityFour.class);
startActivity(intent3);// TODO Call second activity
break;
case R.id.btnActFive:
Intent intent4 = new Intent(this, ActivityFive.class);
startActivity(intent4);// TODO Call second activity
break;
default:
break;
}}
}

```

```

package com.ayaz.kcet;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
public class NotesDbAdapter
{
public static final String KEY_TITLE = "title";
public static final String KEY_BODY = "body";
public static final String KEY_ROWID = "_id";
private static final String TAG = "NotesDbAdapter";
private DatabaseHelper mDbHelper;

```

А бағдарламасының жалғасы

```

private static final String DATABASE_CREATE = "create table notes (_id integer
primary key autoincrement, " + "title text not null, body text not null);";
private static final String DATABASE_NAME = "data";
private static final String DATABASE_TABLE = "notes";
private static final int DATABASE_VERSION = 2;
private final Context mContext;
private static class DatabaseHelper extends SQLiteOpenHelper {
DatabaseHelper(Context context) {
super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
db.execSQL(DATABASE_CREATE);
}
@Override

```

```

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
Log.w(TAG, "Upgrading database from version " + oldVersion + " to " +
newVersion + ", which will destroy all old data");
db.execSQL("DROP TABLE IF EXISTS notes");
onCreate(db);
}
}
public NotesDbAdapter(Context ctx) {
this.mCtx = ctx;
}
public NotesDbAdapter open() throws SQLException {
mDbHelper = new DatabaseHelper(mCtx);
mDb = mDbHelper.getWritableDatabase();
return this;
}
public void close() {
mDbHelper.close();
}
public long createNote(String title, String body) {
ContentValues initialValues = new ContentValues();
initialValues.put(KEY_TITLE, title);
initialValues.put(KEY_BODY, body);
return mDb.insert(DATABASE_TABLE, null, initialValues);
}
public boolean deleteNote(long rowId) {
return mDb.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId, null) > 0;
}
public void deleteAllNote() {
mDb.delete(DATABASE_TABLE, null, null);
}
public Cursor fetchAllNotes() {
return mDb.query(DATABASE_TABLE, new String[] {KEY_ROWID,
KEY_TITLE, KEY_BODY}, null, null, null, null, null);
}

```

А бағдарламасының жалғасы

```

Cursor mCursor = mDb.query(true, DATABASE_TABLE, new String[]
{KEY_ROWID, KEY_TITLE, KEY_BODY}, KEY_ROWID + "=" + rowId, null,
null, null, null, null);
if (mCursor != null) {
mCursor.moveToFirst();
}
return mCursor;
}

```

```

public boolean updateNote(long rowId, String title, String body) {
    ContentValues args = new ContentValues();
    args.put(KEY_TITLE, title);
    args.put(KEY_BODY, body);
    return mDb.update(DATABASE_TABLE, args, KEY_ROWID + "=" + rowId, null)
    > 0;
}
}

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/in" >
    <EditText
        android:id="@+id/text"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="31dp"
        android:ems="10"
        android:text="Привет!!!" >
        <requestFocus />
    </EditText>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true"
        android:orientation="vertical" >
        <TextView
            android:id="@+id/textView1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Вводите слово" />
        <TextView
            android:id="@+id/translatedtext"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/etUserText"
            android:layout_centerHorizontal="true"
            android:layout_marginLeft="90dp"
            android:layout_marginTop="168dp"
            android:text=""

```

А бағдарламасының жалғасы

```

android:textAppearance="?android:attr/textAppearanceLarge" />
<Spinner
  android:id="@+id/selectLanguage"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginLeft="90dp"
  android:layout_marginTop="50dp"
  android:entries="@array/lang"
  android:prompt="@string/choose_lang" />
<Button
  android:id="@+id/say"
  android:layout_width="210dp"
  android:layout_height="wrap_content"
  android:layout_marginLeft="100dp"
  android:layout_marginTop="46dp"
  android:text="Пepeвecmу" />
</LinearLayout>
</RelativeLayout>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity" >
  <TextView
    android:id="@+id/tv_location"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <fragment
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/tv_location"
    class="com.google.android.gms.maps.SupportMapFragment" />
</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  ><WebView
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"

```

```

/></LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"

```

А бағдарламасының соңы

```

    android:id="@+id/splashscreen"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:src="@drawable/splash"
    android:layout_gravity="center">
</ImageView>
<ListView
    android:id="@+id/lister"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
</ListView>
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content" android:layout_height="wrap_content">
<ListView android:id="@+id/android:list" android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No New Notes...Press on Menu to Add New Note"/>
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<LinearLayout android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="wrap_content">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Title" />
<EditText
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_weight="1" />
</LinearLayout>
<TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Body Text" />
<EditText android:id="@+id/insertdata" android:layout_width="fill_parent"

```

```
android:layout_height="wrap_content" android:layout_weight="1"
android:scrollbars="vertical" />
<LinearLayout android:orientation="horizontal" android:layout_width="fill_parent"
android:layout_height="wrap_content">
<Button          android:id="@+id/save"          android:text="Save"
android:layout_width="wrap_content" android:layout_height="wrap_content" />
<Button          android:text="Cut"          android:id="@+id/cut"
android:layout_width="wrap_content" android:layout_height="wrap_content" />
<Button          android:text="Copy"          android:id="@+id/copy"
android:layout_width="wrap_content" android:layout_height="wrap_content" />
<Button          android:text="Paste"          android:id="@+id/paste"
android:layout_width="wrap_content" android:layout_height="wrap_content" />
```