**М.Б.**  **Қайыржан, Ә.Қ. Инаматов,**

**А.Ғ. Бақтыгереев, Л.А. Аманкусова,**

**Ж.Е. Валериева**

**(Атырау, Қазақстан)**

**МОБИЛЬДІҚОСЫМШАЛАРДЫҚҰРУЖӘНЕҚОЛДАНУПРОЦЕСІНТАЛДАУ**

**Кіріспе**. Ақпараттық ресурстар, жүйелер және заманауи технологиялар адам қызметінің ажырамас, қарқынды дамып келе жатқан саласы болып табылады.1997 жылы ұялы байланыс нарығында WAP (Wireless Application Protocol) технологиясы пайда болды, ол компьютерге қосылу үшін кабельді пайдаланбай ұялы телефондарға Интернеттен қосымшаларды тікелей орнатуға мүмкіндік берді.Содан бері қоғамды «жұмылдыру» процесі басталды.2000 жылдардың басында үлкен сенсорлық экраны бар телефондары бар мобильді қосымшалардың дамуы сапалы байланысты қамтамасыз ететін мобильді қосымшалардың жаңа деңгейін құруға мүмкіндік берді. 2010 жылдан бастап мобильді құрылғылар қосымшаларды әзірлеу үшін заманауи ақпараттық технологияларды пайдалануға мүмкіндік беретін жоғары өнімді процессорлармен жабдықталған.Көптеген бағдарламашылар кәсіби қызметтің жаңа бағыттарын зерттей бастады және нарықтық үрдістерге бейімделді.

Қазіргі уақытта мобильді қосымшаларды әзірлеу ақпараттық технологиялар саласындағы ең танымал бағыттардың бірі болып табылады. Мобильді қосымшаларды әзірлеу тұтынушылардың қажеттіліктерін алдын ала білетін, пайдаланушыға белгісіз алгоритмдер мен тапсырмаларға негізделген .

Мобильді қосымша тұтынушылардың белгілі бір тобына арналған және кез келген мәселелер мен қиындықтарды шешуге бағытталған. Мобильді қосымша – белгілі бір әрекеттерді орындай алатын белгілі бір платформада орнатылған функционалды қосымша [1]. Бұл әртүрлі ақпаратпен өзара әрекеттесуге көмектесетін жүйе. Осыған байланысты мобильді қосымшалар келесі түрлерге бөлінеді:

* қосымша оқиға: спорттық, мәдени, білім беру және ғылыми іс-шаралар сияқты әртүрлі іс-шараларды көрсетуге мүмкіндік береді;
* сервистік қосымша: ұйымдарға қызмет көрсететін сайттарға ұқсас сервистік қосымша;
* оқытуға және дамытуға арналған әртүрлі ойындарға арналған қосымшалар;
* интернет-дүкендер;
* әртүрлі брендтерді жарнамалау үшін қолданылатын жарнамалық қосымшалар;
* бизнес-қосымшалар: ұйымдық процесті оңтайландырады, бизнес ақпаратына қол жеткізуді қамтамасыз етеді және оны мәліметтер қорымен біріктіреді;
* жүйелік қолданба: телефон мен оның бағдарламалық құралының кеңейтілген параметрлері мен опцияларын пайдалану;
* телефонды толыққанды навигатор ретінде пайдалануға мүмкіндік беретін GPS модулі арқылы навигация және іздеу қызметтері бар қосымша;
* бейне және дыбыстық ақпаратпен жұмыс істеу кезінде телефонның мүмкіндіктерін кеңейтетін мультимедиялық бағдарламалардағы қосымшалар;
* әлеуметтік желілер, онлайн байланыс қызметтері, ақпаратты тарату және әлеуметтік желі қосымшалары;
* қолданбаның мазмұны және т.б.

Келесі қадам ​​мәселенің неліктен пайда болғанын түсіну және мәселені шешу жолы туралы ақпаратты табу. Проблемалық аймақта мүмкіндігінше көп зерттеулер жүргізу қажет. Мәселе анықталғаннан кейін бұл мәселені шешуде мобильді қосымшаның әсерін бағалауды бастау керек.

**Модельдеу**. Бұл кезеңде қосымшаның ақпараттық архитектурасы жасалуы керек. Ақпараттық архитектура – ​​бұл қолданбада қандай ақпарат пен функционалдылықты көрсету керектігін және бұл деректер мен функционалдылықтың қалай ұйымдастырылғанын анықтау процесі. Әдетте, бұл процесс бағдарламаны іске қосумен байланысты функциялар тізімінен басталады және оны қолданбада көрсету жолын анықтайды [3].Осылайша, бұл әрекет қолданбаны модельдеу кезінде модельдің негізгі құрылымдық блогы болып табылады. Әрі қарай, бағдарлама терезесін құруды және әрбір функция мен деректер орнын тағайындауды бастау керек. Бұл жағдайда әрбір нысанның өз орны бар екеніне сенімді бола аласыз (Cурет 1). Пайдаланушылар қолданбаны пайдалану кезінде туындауы мүмкін кез келген қолайсыздықты алдын ала қарастыруы керек.

Идея

Модель

Бағдарламалау

Түзетуенгізу

Тексеру

*Сурет 1. Мобильді қосымшаларды әзірлеу процесі*

Қолданба негізгі бетте қандай нысандар орналасқанын, мәзірде әрекетті орындау үшін қанша рет басу керектігін және кез келген бетте әрекетті орындау үшін нысанды қанша рет басу керектігін анықтайды.

Мұнда басу объектіні басу дегенді білдіреді. Әрбір басудың интуитивті екенін тексеріп отыру қажет. Әрекетті орындау үшін бір рет басу жеткілікті болса, бұл пайдаланушыға ыңғайлы, бірақ басқа тапсырмалар үшін нұсқаны қайталап басу ұсынылмайды.

Келесі қадам үлгіні басып, ауыстыру болып табылады. «Басу және ауыстыру» үлгісі қолданбаны іске қосу кезінде тексеруге көмектеседі. Бұл негізінен ұялы телефондағы қолданба интерфейсін тексерудің арнайы әдісі [4]. Мысалы, тұтынушылар телефон арқылы қолданбаны ашқан кезде оны одан әрі апаратын сілтеме алады. Қолданбаның қазіргі уақытта ешқандай функционалдығы болмаса да, олар қолданбаның әрбір бетін басып, қолданбаны шарлай бастайды. Мәселелер туындаса, қолданба жоспарына өзгертулер енгізу қажет.

Бағдарламаның құрылымдық блоктары стилистикалық нұсқауларға негізделген. Бағдарламаны пайдалану кезінде дыбыс мәнері нұсқаулығына мұқият назар аудару пайдалы. Жеке дизайн тілінің арқасында қолданба пайдаланушыға ыңғайлы болады. Осылайша, мобильді қосымшаның «өмір сүру ұзақтығы» оның дизайнына тікелей байланысты [5].

Бағдарламалау – техникалық жобалаудың жоғары деңгейі. Мобильді қосымшаны жасау үшін көптеген әдістер, технологиялар және бағдарламалау тілдері қолданылады. Олардың әрқайсысының өз артықшылықтары мен кемшіліктері бар. Кейбіреулерін пайдалану арзанырақ, бірақ өнімділігі төмен, ал басқалары көбірек уақыт пен жоғары деңгейдегі тәжірибені қажет етеді. Қолданбаны құрудың ең оңай жолы – жалпы сенімді технологияны қолдану, яғни бағдарламалық қамтамасыз етуді кодтаусыз, дайын шаблондар арқылы әртүрлі тақырыптар бойынша қосымшалар жасауға болады [6].

Дегенмен, мұндай дайын үлгілер арқылы жасалған қолданбалардың көптеген кемшіліктері бар, мысалы, толық бақылаудың жоқтығы, бар дизайнға тәуелділік, шектеулі интерфейс және т.б.

Мобильді қосымшаларды бағдарламалау кезінде Front-end және Back-end дизайн жүйелерін пайдалануға болады.

Алдыңғы дизайн, Front-end дизайны клиенттің қолданбалы жобасы болып табыладыБасқаша айтқанда, бұл пайдаланушы мен қолданбалы сервер арасындағы интерфейстің дизайны. Бұл пайдаланушыға кейбір деректерді енгізуге, сонымен қатар оны бастапқы өңдеуге және тиісті API (Application Programming Interface) арқылы серверге жіберуге мүмкіндік береді.

Ішкі дизайн, Back-end дизайн – пайдаланушылар мен ресурстар арасында мәліметтерді тасымалдауға жауап беретін қолданбаның серверлік бөлігін әзірлеу [7]. Кодты жазбас бұрын бірнеше нәрсені орындау керек. Бағдарламалау тілдері. API құру үшін ең жиі қолданылатын тілдер Java , C#, Javascript , PHP және Python болып табылады.

Мәліметтер базасы SQL және noSQL екі негізгі деректер базасына негізделген. Дәстүрлі SQL түрлері MSSQL және MYSQL болып табылады.

Инфрақұрылымды енгізу барысында осы кезеңде API және дерекқорды орналастыру мәселелері шешілетін болады. Мұнда қабылданған шешімдер хостинг құнын, масштабталатындығын, орындалуын және қолданбаның сенімділігін анықтауға көмектеседі. Жалпы хостинг провайдерлеріне Amazon AWS және Rackspace кіреді. Провайдерді таңдаудан басқа, жүйені кеңейту көлемін жоспарлау қажет.

Бұлттық шешімдер ресурстарды пайдалы деп қарауға және қажетінше пайдалануға мүмкіндік береді. Олар сонымен қатар дерекқордың сақтық көшірмелерін жасауға, сервердің жұмыс уақытын және операциялық жүйені жаңартуға көмектеседі [8].

Қолданбалар - арнайы платформаға арналған бағдарламалау тілдерінде жазылған және операциялық жүйеге біріктірілген, тез және дұрыс жұмыс істейді, сонымен қатар басқа мобильді қосымшалардың функционалдығы мен жылдамдығын пайдаланады. Олар бұл платформаға бағдарлама интерфейсімен қарапайым әрекеттесулерді қарапайым түрде орындауға мүмкіндік береді [9]. Сонымен қатар, аппараттық қосымшалар бар: бейнекамера, микрофон, акселерометр, телефон кітапшасы және т.б. Әрине, мұндай бағдарламаны жазу көп уақытты қажет етеді, сондықтан бағдарламаны әзірлеу ортасын пайдалану үшін арнайы біліміңіз болуы керек, сонымен қатар әрбір платформаның өз бағдарламалары бар екендігі, бұл бағдарламаларға сұранысты арттырады.

Мұндай қолданбалардың өз құралдары мен бағдарламалау тілдері бар. Мысалы, Java Android Studio үшін жиі пайдаланылады, және де Android операциялық жүйесіне арналған бағдарламаларды жазу үшін, сондай-ақ iOS үшін Xcode және Objective-C, сонымен қатар жақында танымал Swit, Visual Studioбағдарламалау тілдері Windows Phone және C# үшін тиімді жұмыс істейді [8].

Кейде сізге бірнеше платформаларда жұмыс істей алатын, жергілікті қолданбалардан басқа қосымшаларды қысқа мерзімде жасауға мүмкіндік беретін бағдарламалар қажет. Бұл жағдайда гибридті немесе веб-қосымшалар таңдалады, ал өңдеу үшін кросс-платформалық мобильді платформалар қолданылады [4].

**Мобильді қосымшаларды жобалау негіздері.** Бүгінгі таңда мобильді қосымшаларды жасауға көмектесетін көптеген құралдар мен фреймворктар бар. Мысалдарға IONIC, Mobile Angular UI, Intel XDK, Titanium Devices, Sencha Touch, Kendo UI, PhoneGap.AppceleratorTitanium жатады - компьютерлерде де, ұялы телефондарда да қосымшаларды жазуға арналған платформалар.

Бұл платформада iOS, Android, BlackBerry сияқты операциялық жүйесі бар телефондарға арналған қосымшаларды жасауға болады. Бағдарламаны жазу үшін JavaScript бағдарламау тілін білу керек. Бағдарлама арнайы жүйелері мен әдістері бар объектілерден тұрады. Объектілердің үлкен тізімі операциялық жүйенің барлық мүмкіндіктерін пайдалануға мүмкіндік береді. JavaScript қолданбасы Appcelerator Titanium API интерфейсімен байланысады. Жақтау мәтін өрістері, түймелер, тізімдер сияқты элементтерді жасау үшін әртүрлі UI құрамдастарын пайдалануға болады. Бұл нысандардың дәлдігі мобильді платформаны дұрыс басқару арқылы қамтамасыз етіледі. Көп жағдайда бір платформада жазылған код басқа платформада ешбір өзгеріссіз жұмыс істей алады.

Дегенмен, барлық айырмашылықтары платформада дұрыс көрсетілмейді. Бұл жағдайда әрбір нақты платформа үшін жеке код парағын жазу керек. Шағын қолданбаны жасау үшін сізге Titanium.UI сайтынан бірнеше нысандарды білу қажет.

Міне, бір мысал: Titanium.UI.Window — әртүрлі қасиеттері мен әдістері бар ең маңызды нысандардың бірі.

Фондық суретті, тақырыпты, нысанды орналастыруды және т.б. көрсетуге болады. Нысан үшін мыналарды көрсетуге болады:

var windows = Titanium.UI.createWindow ({backgroundColor: 'red', title: 'My window'})орнату ; windows.open (); itTitanium.UI.

Мұнда батырма белгілі бір әрекеттерді орындау үшін пайдаланылады

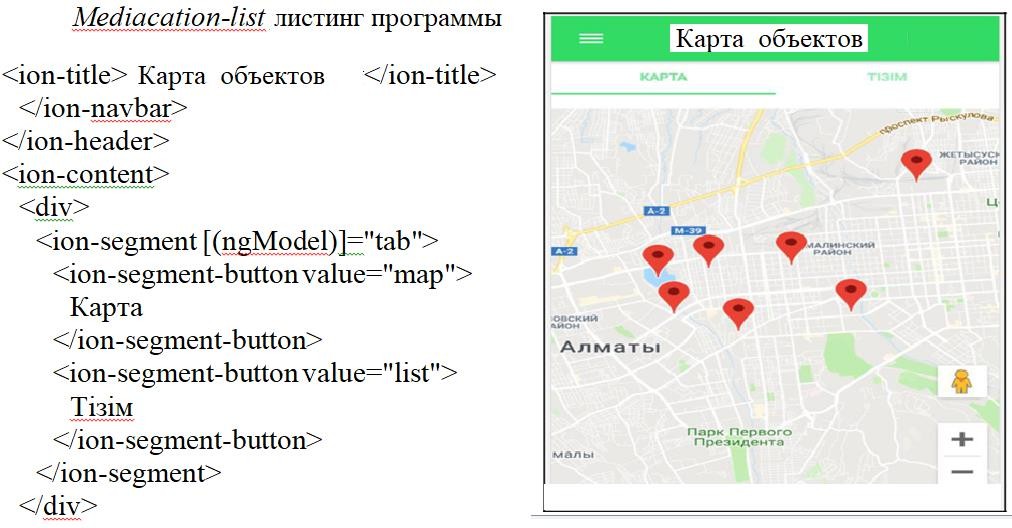
Varbutton=Titanium.UI.createButton({title:'Hello'});button.addEventListener('click',function(e)

{Titanium.API.info("button");});

itTitanium.UI.ImageView — кескіндікөрсетугеарналғанэлемент. var image = Titanium.UI. createImageView ({изображение: 'myimg.png', ширина: 'авто', высота: 'авто'}); view.add (Сурет 2).

Android көп бөлікті Linux ядросына негізделген. Ол жүйені функцияларға бөлуді қамтамасыз етеді және қауіпсіздікке, жадқа, қуат пен процесті басқаруға жауап береді және желілік стекпен драйвер үлгісін пайдаланадыЯдроның «жоғарғы» деңгейінде бағдарламаның негізгі негізгі функцияларына арналған кітапханалар жинағы, бағдарламалық және аппараттық құрал болып табылады.

Осылайша, бұл деңгей жоғары деңгейлер үшін дайындалған алгоритмдерді көрсетеді, файл пішімдерін қолдайды, ақпаратты кодтайды, декодтауды орындайды, графиканы жасайды және т.б. Кітапханалар арнайы жабдық негізінде C/C++ тілінде жасалады.



*Сурет 2.Mediacation-listлистинг бағдарламасы*

Бұл кодты қолдану арқылы әзірлеуші ​​Java немесе Swift-пен салыстырғанда жылдамдыққа ие болады, бірақ бағдарламаның жылдамдығы төмендеуі мүмкін, себебі бағдарламаларда орындалатын операциялардың саны артады. Осылайша, бағдарламашы алдымен белгілі бір тапсырманы немесе орындалу жылдамдығын таңдауы керек.

Xamarin платформасын мобильді қосымшаларды жасау үшін пайдалануға болады. Xamarin C++ тілін пайдаланады. Бұл жүйені пайдалана отырып, бір қолданбаның логикасын бір уақытта бірнеше платформаларда – Android, iOS, Windows Mobile-де жасауға болады. Бұл қолданба Android, iOS және Windows Phone жүйелерінде жұмыс істейді. Xamarin.Forms технологиясы бұған мүмкіндік береді.Xamarin платформасын пайдалану қолданбада кросс-платформалық кодтың үлкен бөлігі болуы керек фактісіне негізделген. Android құралдары – Android виртуалды құрылғысы (AVD) эмуляторын қамтидыAVD эмуляторы Android OC жұмыс істейтін виртуалды мобильді құрылғыда қолданбаны тексеруге мүмкіндік береді. Эмулятордың әртүрлі конфигурациялары бар бірнеше виртуалды құрылғыларды жасау мүмкіндігі бар. Android виртуалды құрылғысын жасау үшін Терезе → Android SDK және AVD менеджері мәзірінде «Жаңа» батырмасын басыңыз, виртуалды құрылғының атын енгізіңіз және оның сипаттамаларын көрсетіңіз [8].

**Қорытынды**. Қазіргі уақытта мобильді қосымшаларды жобалаудың заманауи технологиялары кеңінен қолданылады, ал мобильді бағдарламалау кеңінен тараған. Мобильді бағдарламалау технологияларындағы жетістіктер мобильді қосымшаларды әзірлеуге, сынауға және орналастыруға мүмкіндік береді.

Мобильді қосымшалар Google Play және Apple App Store дүкендерінде жарияланып, кеңінен қолданылады. Әзірленген қосымшалар, егер олар кәсіби түрде әзірленсе, мобильді қосымшалар адамзат өмірінде ерекше орын алып, олардың күнделікті жұмысын жеңілдетеді.

**ӘДЕБИЕТТЕР ТІЗІМІ:**

1. Куляшова Н.М., Карпюк И.А., Дроздов С.А. Основные принципы построения и классификации информационно- поисковых систем // МатериалыXVI науч.конф.молодых ученых, аспирантов и студентов Мордов.гос.ун-та.им. Н. П.Огарёва: в6ч.-Ч.3. Естественные науки.–Саранск: Изд-во Мордов. ун-та, 2013.–С.66-68.
2. Пантилейкин Н.В. Мобильные приложения и их виды // Научно-методический электронный журнал Концепт. – 2016. – Т. 26. –С. 776-780.
3. Харди Б., Филлипс Б. Программирование под Android. Для профессионалов. — СПб.: Питер, 2014. —592 с.: ил. - (Серия «Для профессионалов»).
4. Варакин М.В. Разработка мобильных приложений под Android–УЦ «Специалист» 2012г. - 592с.
5. Head First. Программирование для Android. —СПб.: Питер 2016.—704с.: ил.— (Серия «Head First O’Reilly»).
6. VogelL. Android Development Tutorial.URL:[http://www.vogella.de/android.html ( дата](http://www.vogella.de/android.html%20(%20дата) обращения: 20.04.2011)
7. Буч Г., РамбоД., Якобсон И.Язык UML. Руководство пользователя. 2-еизд.-М.:ДМК Пресс,2007.
8. Голощапов А.Л. Google Android: Программирование для мобильных устройств. - СПб.: БХВ- Петербург, 2011.
9. Хашими С., Коматинени С., Маклин Д .Разработка приложений для Android. - Питер,2011.

**Ғылыми жетекшісі**

п.ғ.к., профессор Абыканова Бакытгуль Толыбековна.