

3. Железко, Ю.С. Стратегия снижения потерь и повышения качества электроэнергии в электрических сетях / Ю.С. Железко // Электричество. – 1992. – № 5. – С. 6-12.
4. Воротницкий, В.Э. Расчет, нормирование и снижение потерь электроэнергии в электрических сетях / В.Э Воротницкий, М.А. Калинкина. – М.: 2000. – 64 с.
5. Белей, В.Ф. Стандарты в области качества электроэнергии: проблемы и тенденции/ В.Ф. Белей, М.С. Харитонов // Информационные ресурсы России. – 2016. – № 1. – С. 10-12.
6. Белей, В.Ф. Результаты экспериментальных исследований светодиодных ламп сетевого напряжения / В.Ф. Белей, М.С. Харитонов // Промышленная энергетика. – 2012. – № 9. – С. 9-14.
7. Белей, В.Ф. Некоторые рекомендации для систем освещения на основе энергосберегающих ламп / В.Ф. Белей, М.С. Харитонов // Промышленная энергетика. – 2014. – № 6. – С. 41-47.
8. Белей, В.Ф. Компактные люминесцентные лампы: электрические характеристики, проблемы электромагнитной совместимости / Электротехника. – 2002. – № 7. – С. 48-51.
9. Artyukhov, I.I. Adaptive Control of Energy Flows in the Hybrid Power Supply System / I.I. Artyukhov, S.F. Stepanov, G.N. Tulepova, E.T. Erbaev, K.K. Tulegenov // 2018 International Conference on Actual Problems of Electron Devices Engineering, APEDE 2018, 2018, S. 355–361, 8542260.
10. Artyukhov, I.I. Energy Efficiency Analysis of Control Algorithms for Fan Electric Drives in Gas Air-Cooling Plants / I.I. Artyukhov, A.M. Abakumov, A.I. Zemtsov, Ye.T. Yerbayev, V.P. Zakharov // Lecture Notes in Civil Engineeringthis link is disabled, 2022, 190, S. 46–55.

РЕЗЮМЕ

Оптимальные меры по снижению затрат на электроэнергию и ее потерь на промышленном предприятии, такие как использование напряжения 10 (35) кВ в качестве основного, энергосберегающих трансформаторов, светодиодных ламп, замена электроприводов систем водоснабжения и т.д. на вариаторы с асинхронным двигателем.

RESUME

Optimal measures to reduce expenditures on electricity and its losses in an industrial plant, such as the use of voltage 10 (35) kV as the primary, energy-saving transformers, led lamps, replacement of electric drives of water supply systems, etc. on variable speed drives with asynchronous motor.

ӘОЖ:004.3

Білім алушы: Абутолыбов А., магистрант

Ғылыми жетекші: Касымова А.Х., п.м.а., п.ғ.к, жетекші

«Жәңгір хан атындағы Батыс Қазақстан аграрлық-техникалық университеті» КеАҚ, Орал қ.

ІТ МАМАНДАРДЫҢ ҚЫЗМЕТ КӨРСЕТУГЕ АРНАЛҒАН МОБИЛЬДІ ҚОСЫМШАСЫН ӘЗІРЛЕУ

АННОТАЦИЯ

Қазіргі заманғы ақпараттық технологияның дамуы, кез келген уақытта байланыста жүру, қажетті ақпаратқа қол жеткізу, жалпы қоғамды ақпараттандыру үшін мобильдік құрылғылар арқылы мобильді қосымшаларды пайдалану мен оны дамыту өте маңызды болып табылып отыр. Мобильді құрылғыларға арналған ақпараттық технологияларды дамыту белсенді түрде орындалуда. Мақалада мобильді қосымшалардың

пайдалану, оларды құру процессері және олардың құрылымы бойынша жіктелуі қарастырылған. Сонымен қатар, мақалада мобильді құрылғылар мен қосымшалардың маңыздылығы атап көрсетілген. Мобильдік қосымшаларды жасаудың кейбір құралдары талданады, оларды қолдану мысалдары келтірілген. Мобильді қосымшаларды әзірлеу процесі айқындалған. Мобильді қосымшаларды бағдарламау кезінде ұсынылған. Мобильді қосымшаны тестілеудің төрт негізгі түрлері сипатталынған. Сонымен қатар, мақалада мобильдік қосымшаларды әзірлеуге арналған қолданыстағы құралдарды талдау берілген.

Кілт сөздер: *ақпараттық технология, мобильді құрылғылар, қосымша, мобильді қосымшалар тестілеу, мобильді қосымшаларды бағдарламау, дизайн, интерфейс, платформа.*

Кіріспе. Қазіргі таңда технологияның дамыған уақытында смартфондарды қолданудың өсуімен мобильді қосымшалар біздің күнделікті өміріміздің ажырамас бөлігіне айналды. Осылайша, әртүрлі функционалдығы бар сапалы мобильді қосымшаларға сұраныс жоғары.

Мобильді қосымшаны құру көп күш пен берілгендікті қажет етеді. Бұл бағдарламалық жасақтаманы әзірлеудегі ең қызықты тапсырмалардың бірі.

Мобильді қосымшаны әзірлеу жоспарлауды, жобалауды, кодтауды және тестілеуді қажет етеді. Бірінші қадам-қосымшаның мақсатын анықтау және оның мақсатты аудиториясын анықтау. Содан кейін қосымшаның функциялары анықталып, пайдаланушы интерфейсі жасалуы керек[1]. Пайдаланушы интерфейсі пайдаланушыларға тартымды болу үшін қарапайым және пайдаланушыға ыңғайлы болуы керек.

Тақырыптың өзектілігі. Компьютерлік техниканы қолдану саласының жылдам кеңеюі көптеген күрделі қолданбалы міндеттерді тудырады. Бұл міндеттерді кешенді шешу бір ғана технология шеңберіне жатқызылмайды, сонымен қатар әр түрлі ақпараттық модельдерді бірлесіп қолдануды және әр түрлі кластағы қосымшалар арасындағы өзара іс-қимылды талап етеді. Сондықтан қазіргі уақытта әр түрлі қосымшаларды интеграциялауды, сондай-ақ әр түрлі технологияларды гибридизациялауды қамтитын ақпараттық технологияларды қолдану өзекті болып табылады.

Бұл ретте оларды оңдау, қателіктерлі болдырмау, дер кезінде көмек көрсету сынды міселелендінің маңыздылығы арта түседі.

Kotlin және Android studio бағдарламалары арқылы осы талаптарға сай мобильді қосымша жасау маңызды болып табылады. Деректер базасын енгізу, әр қолданушы мен маман жайында деректерді, атқарылған жұмыс жайында мәліметтерді сақтау маңызды.

Зерттеу мақсаты: Үйден шықпай-ақ заманауи технологияларды жөндеу процессін тез әрі жеңілдетуге арналған мобильді қосымша құру

Зерттеу міндеттері: Ақпараттық құрылғылармен қолдану кезінде өзекті әрі жиі кездесетін қиыншылықтардың тізімін даярлау;

- жүйені пайдаланушының мүмкін болатын операцияларын автоматтандыру есебінен ақпаратты өңдеу жылдамдығын арттыру;
- есеп беру нысандарын қалыптастыру;
- жүйенің кеңеюі (автоматтандырылған жүйеге қойылатын талаптардың жоғарылауы жағдайында оны нақтылау мүмкіндігі);
- пайдаланушының ыңғайлы интерфейсі.

Жаңашылдығы: Маман іздеу процессінің қарапайымдылығы мен баға жөнінде алдын ала келісу және пайдаланушыға да, маманға да ыңғайлы интерфейс даярлау.

Мобильді қосымшалардың сиқыры-олардың қарапайымдылығы, бір рет түрту әрекеттері және лезде жауап беру. Дизайн осы сипаттамаларды көрсетуі керек. Навигация интуитивті және пайдалану оңай болуы керек. Түстер мен қаріптер бүкіл қолданбада бірдей болуы керек, ал суреттер мен графика мобильді құрылғылар үшін оңтайландырылуы керек.

Дизайн аяқталғаннан кейін кодтау процесі басталады. Мобильді қосымшаларды құру үшін қолданылатын бағдарламалау тілі қолданылатын платформаға байланысты өзгереді. Android Java немесе Kotlin қолданады, ал IOS Swift немесе Objective C қолданады, үйлесімділікті қамтамасыз ету үшін қосымшаны әзірлеу кезінде қолданылатын платформаны ескеру қажет[2].

Тестілеу мобильді қосымшаны құру процесінің ажырамас бөлігі болып табылады. Тестілеу қолданбадағы кез келген ақауларды немесе қателерді анықтауға және жоюға көмектеседі. Жақсы тестілеу стратегиясы әртүрлі орталарда, құрылғыларда және пайдаланушы сценарийлерінде тестілеуді қамтиды. Пайдаланушылардың кері байланысы тестілеу кезеңінде де маңызды, өйткені ол қосымшаны көпшілікке шығарар алдында жақсарту бағыттарын анықтауға көмектеседі[3].

Әдістемелер, нәтижелер: Мобильді қосымшаларды әзірлеу дегеніміз не? Смартфондар және басқа портативті құрылғылар сияқты шағын сымсыз есептеу құрылғыларына арналған бағдарламалық жасақтаманы жазумен байланысты процестер мен процедуралар Мобильді қосымшаларды әзірлеу деп аталады.

Мобильді қосымшаларды әзірлеу, веб-қосымшаларды әзірлеу сияқты, дәстүрлі бағдарламалық жасақтамадан туындайды. Бір маңызды айырмашылық-мобильді қосымшалар көбінесе белгілі бір мобильді құрылғының бірегей мүмкіндіктерін пайдалану үшін арнайы жазылады. Мысалы, iPhone акселерометрінің артықшылықтарын пайдаланатын ойын қолданбасын немесе смарт сағаттардағы температура сенсорының артықшылықтарын пайдаланатын мобильді денсаулық қолданбасын жазуға болады[4].

Бүгінгі таңда ең танымал екі мобильді платформа-Apple iOS және Google Android. Apple телефондары мен планшеттері толық веб-шолғыш және Apple App Store сияқты алдын ала орнатылған негізгі қолданбалармен бірге келеді.[5]

Мобильді қосымшалардың алғашқы күндерінде кез-келген құрылғыда қосымшаның оңтайлы жұмыс істеуін қамтамасыз етудің жалғыз әдісі оның алғашқы дамуы болды. Бұл әр құрылғыдағы процессор үшін арнайы жаңа код жазылуы керек дегенді білдірді. Бүгінгі таңда жасалған мобильді қосымшалардың көпшілігі құрылғыға тәуелді емес.

Қолданба кросс-платформа болуы және бірнеше операциялық жүйелерде (OSes) іске қосылуы керек болса, бастапқы даму жобасынан қайта пайдалануға болатын код аз болды. Әрбір құрылғы өзінің кодтық базасы бар мобильді қосымшаны әзірлеудің жеке жобасын қажет етті. Заманауи кросс-платформалық құралдар C# және JavaScript сияқты жалпы тілдерді қолдана отырып, әртүрлі жобаларда кодты бөліседі; ең бастысы, олар Jenkins сияқты қосымшалардың өмірлік циклін басқару құралдарымен жақсы үйлеседі.

Әзірлеушілер енді Apple iOS, Google Android және progressive Web apps (PWAs) үшін бірыңғай кодтық базаны қолдана алады. PWA соңғы пайдаланушыдан app store дүкеніне кіруді, Бағдарламалық құралды сатып алуды және оны жергілікті жүктеп алуды талап етпей, мобильді құрылғылардың кірістірілген мүмкіндіктерінің артықшылықтарын пайдалануға арналған. Оның орнына PWA-ны іздеу жүйесін сұрау арқылы табуға болады және тікелей браузер арқылы қол жеткізуге болады, бұл Электрондық коммерция сатушыларын бірнеше мобильді операциялық жүйелерге арналған жеке қосымшаларды жасау қажеттілігінен құтқарады. YouTube бейнелері сияқты PWA мазмұны кезең-кезеңімен жүктеледі, бұл соңғы пайдаланушыға жауап беретін дизайны бар дәстүрлі веб-сайтқа қарағанда жақсы пайдаланушы тәжірибесін ұсынады. Прогрессивті веб-қосымшалар жедел мобильді қосымшалар ретінде де белгілі. Сіз оны әзірлеуді бастамас бұрын алдымен қандай қолданба түрін жасайтыныңызды шешуіңіз керек. Мұнда әртүрлі түрлердің қысқаша сипаттамасы берілген[6].

Жергілікті қосымшалар. Бұл қосымшалар интеграцияланған даму орталары (IDE) және Apple iOS немесе Google Android сияқты мобильді операциялық жүйелерге арналған Тілдер арқылы жасалады. Жергілікті қолданбалар қажетті мүмкіндіктерді реттеуге мүмкіндік береді, бірақ олар басқа технологияларға қарағанда қымбатырақ.

Гибриді қосымшалар. Бұл жергілікті қосымшаларға ұқсас жұмыс істейтін веб-қосымшалар. Олар HTML технологиялары, JavaScript және каскадты стиль кестелері (CSS) арқылы жасалады. Гибриді қосымшаларды әзірлеу жергілікті қосымшаларға қарағанда арзанырақ және оларды тезірек жасауға болады, бірақ оларда жергілікті қосымшалардың мүмкіндіктері жоқ.

Прогрессивті веб-қосымшалар, прогрессивті веб-бағдарлама (PWA) - бұл мобильді қосымшаға ұқсайтын және әрекет ететін веб-сайт. Бұл қосымшалар Facebook React сияқты веб-технологияларды қолдану арқылы жасалады.

Инкапсуляцияланған қосымшалар. Microsoft Power App drag-and-drop app creation tool сияқты өнімдер тәжірибесі аз әзірлеушілерге Мобильді қосымшаларды жылдам жасауға мүмкіндік береді. Дегенмен, негізгі операциялық жүйеден оқшаулаудың болмауы, операциялық жүйені құлыптау және салыстырмалы жаңалық қиындықтар тудыруы мүмкін[7].

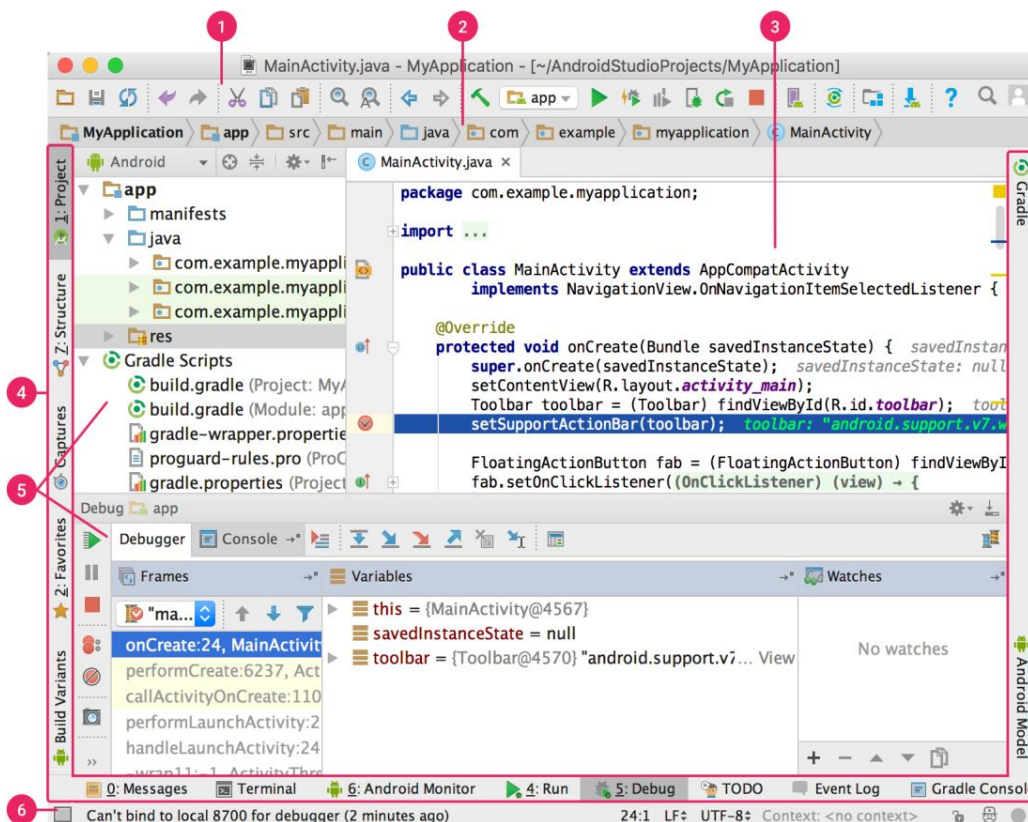
Құрылымдар мен кітапханалар. Мобильді қосымшаның дамуын жеделдету үшін басқа біреудің қайта пайдаланылатын кодын пайдалануға болады.

Қолданбаны әзірлеу құны қолданбаның түріне және оның мақсатты пайдаланылуына байланысты іс жүзінде ештеңеден миллиондаған долларға дейін болуы мүмкін. Төменде мобильді қосымшаны әзірлеуге арналған баға диапазоңдарының бөлінуі келтірілген.

Кодсыз қосымшаларды жасаушылар. Егер қолданбаға тек негізгі мүмкіндіктер қажет болса, әзірлеушіге кодты қалай жазу керектігін білудің қажеті жоқ. GoodBarber сияқты тегін құралдар, Appery.io , Shoutem, Appy Pie және BuildFire сізге Swift немесе басқа бағдарламалау тілдерін үйренбестен қосымшалар жасауға мүмкіндік береді. Бұл құралдардың функционалдығы шектеулі және оларды кодсыз қосымшалармен ойын құру үшін пайдалану мүмкін болмаса да, кодсыз тәсіл көптеген ұйымдардың қажеттіліктерін қанағаттандырады[8].

Мысалға, Кәсіпорын қосымшаларын қарастыратын болсақ, Amazon, Mendix және Microsoft Power Suite's HoneyCode сияқты құралдардың арқасында кез келген адам мобильді қосымшаны жасай алатын азаматтық әзірлеуші тұжырымдамасы қарқынды дамып келеді. Бұл құрылғыларға деректер көздеріне қосылу және мазмұн ағынын басқару үшін апарып тастау интерфейстері кіреді. Құны әдетте ай сайынғы жазылыммен байланысты болып келеді.

Мобильді оңтайландырылған Веб-сайт. Жұмыс үстеліне де, мобильді құрылғыларға да веб-сайттар жасау практикалық болғанымен, сіз қолданатын веб-сайттың мазмұнын басқару құралында веб-сайтыңызды мобильді құрылғыларға оңтайландыратын плагиндер болуы мүмкін.



Сурет 1- Android Studio Интерфейсі

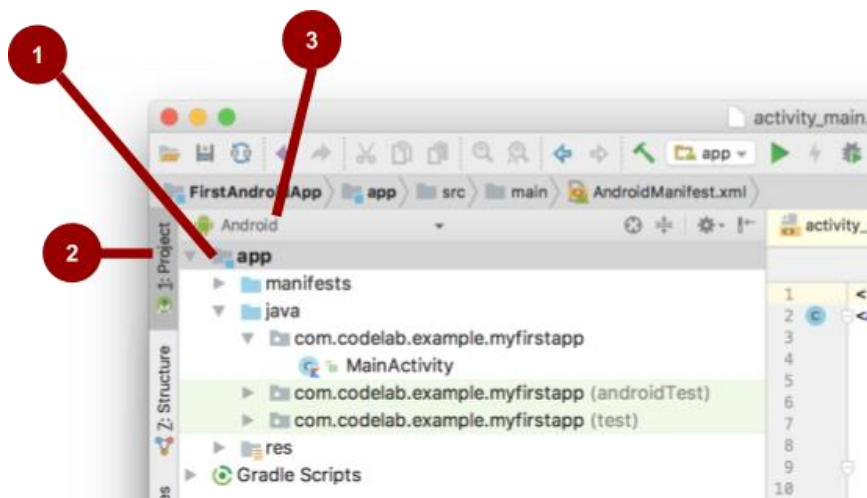
Күрделі қосымшалар. 3D, ойын немесе күрделі жасанды интеллект (AI) сияқты мүмкіндіктерге жергілікті қолданба қажет болуы мүмкін. Күрделі қосымшаның құны әдетте жоғары болуы мүмкін. Құны мобильді әзірлеушілердің тапшылығына тікелей пропорционалды[9].

Жобаны жасау кезінде көрсетілген жерде дискіде Android Studio жобасы үшін қалта жасайды. Содан кейін жобаның құрылысы жүреді (бұл бірнеше минутты алуы мүмкін). Android Studio Gradle-ді құрылыс жүйесі ретінде пайдаланады. Кейде құрастыру жүйесі хабарламалар мен ескертулер береді, мысалы қазір жетіспейтін SDK компонентін қайта орнату керек.

Ағымдағы жобаны көрсететін код редакторы терезесін ашады. Android Studio терезесі келесі 1-суреттегідей болуы керек. Android Studio интерфейсін қарастырсақ, құралдар тақтасы қолданбаны іске қосу немесе SDK менеджерін ашу сияқты ең көп сұралған командаларға жылдам қол жеткізуге мүмкіндік береді. Navigation bar жоба бойынша шарлауға және өңдеуге арналған файлдарды ашуға көмектеседі. Ол project терезесінде қамтамасыз көрінетін құрылымның ақтам көрінісін етеді. Редактор терезесі кодты жасауға және өңдеуге мүмкіндік береді. Ашық файл түріне байланысты редактордың көрінісі өзгеруі мүмкін. Мысалы, орналасу файлын қарау кезінде редактор орналасу редакторын көрсетеді.

Tool buttons панелі ide терезесінің периметрі бойынша орналасқан және жеке құралдар терезелерін үлкейтуге немесе кішірейтуге мүмкіндік беретін түймелерден тұрады. Құралдар терезелері жобаларды басқару, іздеу, нұсқаларды басқару және т.б. сияқты белгілі бір тапсырмаларға қол жеткізуге мүмкіндік береді. Оларды ашуға және орауға болады. Әрі қарай, біз оларды толығырақ қарастырамыз. Status bar жоба күйлерін және Ide-нің өзін, сондай-ақ әртүрлі ескертулерді немесе хабарламаларды көрсетеді. Негізгі терезені құралдар тақталары мен құралдар терезелерін жасыру немесе жылжыту арқылы экранда көбірек орын беру үшін ұйымдастыруға болады. Көптеген ide мүмкіндіктеріне қол жеткізу үшін пернелер тіркесімдерін де пайдалануға болады. Кез келген уақытта Shift пернесін екі рет басу немесе Android Studio терезесінің жоғарғы оң

жақ бұрышындағы үлкейткіш әйнекті басу арқылы бастапқы кодты, дерекқорларды, пәрмендерді, пайдаланушы интерфейсінің элементтерін және т.б. іздеуге болады. Егер сіз, мысалы, іске қосуды ұмытып кеткен белгілі бір ide пәрменін табуға тырыссаңыз, бұл өте пайдалы болуы мүмкін. Жоба құрылымы терезесі 2-суретте көрсетілгендей.



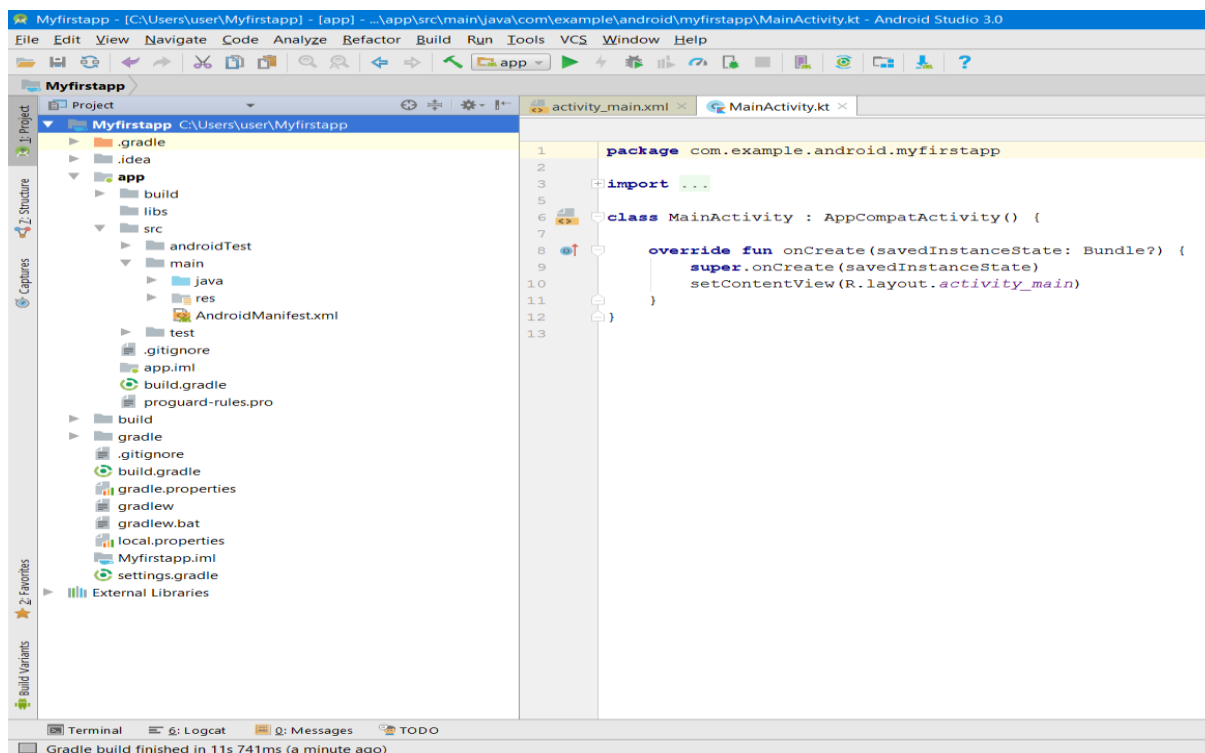
Сурет 2 - Android Studio жобасының құрылымы терезесі.

Жобаның құрылымы мен орналасуын зерттеу, қолданба файлдарының иерархиясын бірнеше жолмен көруге болады.

Жоба құрылымы терезесі (1) файл ағашын көрсетеді. (сурет-1 қараңыз)

Project (2) түймесі жоба құрылымы терезесін жасырады және көрсетеді. Мәзір элементін қосыңыз көру > құралдар түймелері осы батырманы көру үшін. Project > Android жобасының түрін ағымдағы таңдау. Басыңыз Android ашылмалы тізімі (3) жобаның басқа қол жетімді түрлерін көру үшін.

Project > Project дисплейі ретінде сіз жоба қалталарының толық құрылымын көресіз, ол дискіде қалай көрінеді. Мұнда көптеген файлдар мен қалталар бар, ал жаңадан келген адам бұл жерде шатасуы мүмкін. Ең бастысы, назар аудару керек: жоба модульдік құрылымға ие және біздің қосымшаның негізгі модулі – app. Сурет-3 те көрсетілген. Егер сіз app қалтасын ашсаңыз, біз осындай қалталарды көреміз: build - мұнда компиляция процесінде жүйе жасаған файлдар ондағы ештеңені өзгертпегені дұрыс libs-жобаға қосылған үшінші тарап кітапханаларына арналған қалта src-бастапқы код пен ресурстарға арналған қалта src ішінде main қалтасы бар-бұл біз айналысатын негізгі жұмыс қалтасы main ішінде java және res қалталары бар-бұл код пен ресурстарға арналған қалталар, app Модулінің қалтасында build құрастыру жүйесінің файлы бар.gradle, біз оны кейінірек қарастырамыз. Егер сіз Project > Android жобасының көрінісіне ауыссаңыз, сіз тек жеңілдетілген жоба құрылымы бар app Модулінің қалтасын көресіз.

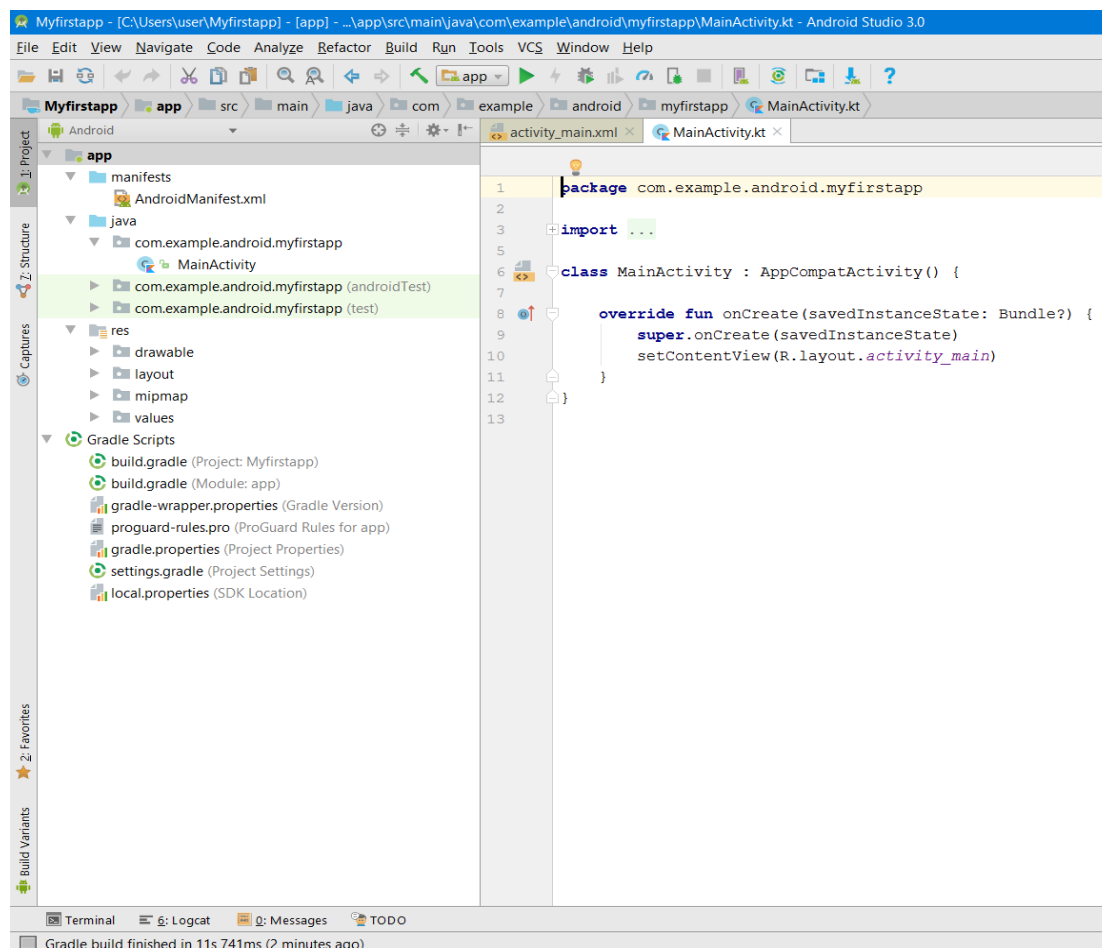


Сурет 3 - Жоба> жоба түріндегі Android Studio жобасының құрылымы терезесі.

Бұл көрініс ыңғайлы, өйткені мұнда біз тікелей жұмыс істейтін файлдар ғана сүзіледі. Бірақ бұл формада жоба қалталарының құрылымы дискіде сақталғанға сәйкес келмейтінін есте ұстаған жөн. Мысалы, жоба қалтасындағы дискіде manifests қалтасын таба алмайсыз. Бұл режимдегі қалта ағашында src және main қалталары көрсетілмейді. 4-суретте көрсетілген. Project > Android түрінде сіз app қалтасында келесі қалталарды көресіз: manifests, java, және res. Manifests қалтасын ашыңыз. Бұл қалтада AndroidManifest файлы бар.xml. Бұл файл Android қосымшасының барлық компоненттерін сипаттайды және Қолданбаны іске қосқан кезде android жұмыс уақыты жүйесі оқиды. Java қалтасын ашыңыз. Котлин және Java тілдеріндегі барлық Код файлдары осында ұйымдастырылған. Java қалтасында үш ішкі қалта бар: com.example.android.myfirstapp (немесе көрсетілген домен атауы): бұл қалтада сіздің бағдарламаңызға арналған Котлин және Java бастапқы код файлдары бар. Бұл қалтамен біз көп жағдайда жұмыс істейтін боламыз, өйткені бұл жобаның негізгі пакеті.

com.example.android.myfirstapp (androidTest): бұл қалта аспаптық сынақтарды орналастыруға арналған. com.example.android.myfirstapp (test): бұл қалта модульдік сынақтарды орналастыруға арналған.

Тесттер туралы толығырақ Android қосымшаларының архитектурасы курсынан біле аласыз. Res қалтасын ашыңыз. Бұл қалтада қолданбаға арналған барлық ресурстар, соның ішінде суреттер, экран орналасу файлдары, жол ресурстары, белгішелер мен белгішелер, түстер мен безендіру стильдері бар.



Сурет-4. Project>Android түріндегі Android Studio жобасының құрылымы терезесі.

Оған келесі ішкі қалталар кіреді: drawable: қолданбаңызға арналған барлық суреттер сол қалтада сақталады.layout: бұл қалтада қолданбаның белсенді экрандарына арналған орналасу файлдары бар. Қазіргі уақытта сіздің қосымшаңызда activity_main орналасу файлы бар бір активити бар.xml.mipmap: бұл қалтада қолданбаны іске қосу белгішелері бар-бұл қолданбаны орнатқаннан кейін Android құрылғысында көрсетілетін белгішелер.мәндер: қолданбада қолданылатын жолдар мен түстер сияқты ресурстарды қамтиды.Сондай-ақ, жоба құрылымы терезесінде Gradle Scripts қалтасы бар. Оны ашыңыз. Тізімнің басында бірдей build атаулары бар екі файл бар.gradle. Бұл қолданбалар мен кітапханаларды құрастыру, құру және орау үшін қолданылатын Gradle жүйесіне арналған құрастыру файлдары. Файл атауынан кейінгі жақшада файлдың app модуліне немесе бүкіл жобаға тиесілі екендігі көрсетілген. Жоба деңгейіндегі құрастыру файлы бүкіл жобаның параметрлерін, ал модуль деңгейіндегі құрастыру файлы модульдің параметрлерін қамтиды. Көбінесе біз модуль деңгейіндегі құрастыру файлымен жұмыс істейтін боламыз. Онда келесі негізгі бөлімдер бар: android {...}, мұнда әзірлеу құралдарының нұсқалары, API-дің минималды қолдау көрсетілетін нұсқасы, Google Play идентификаторы және Қолданба нұсқасы және басқа опциялар көрсетілген жобаға қосылатын кітапханалардың тізімін қамтитын тәуелсіздіктер {...}. ергілікті (libs қалтасына орналастырылған) және қашықтан сақталған кітапханалар қосылуы мүмкін.Android қосымшасын іске қосу.

Әзірлеу процесінде қосымшаның жұмысын тексерудің ең айқын әдісі-оны Android құрылғысына орнату және іске қосу. Қолданбаны Android Studio әзірлеу ортасынан іске қосу үшін біз 2 әдісті қолдана аламыз:Android жүйесі жұмыс істейтін компьютерге қосылған смартфон немесе планшет Android эмуляторы Бұл әдістердің екеуінің де артықшылықтары мен кемшіліктері бар және қайсысын пайдалану керектігін өзіңіз

таңдайсыз. Жаңадан бастаушы үшін кез-келген әдіс қолайлы. Android смартфонды дегеніміз не, сіз қосымшаларды әзірлеу тақырыбына қызығушылық танытқаныңызды білесіз. Ал Android эмуляторы дегеніміз не? Бұл дербес виртуалды машинада компьютерде Android жүйесін іске қосу. Көптеген Android эмуляторлары бар, мысалы BlueStacks, Droid4X, Nox APP Player, Genymotion, және басқалар. Олардың көпшілігі ойынға арналған және Genymotion-тан басқа қолданбаларды сынау үшін өте қолайлы емес. Android Studio-да Nexus және Pixel смартфондары, сондай-ақ ең көп таралған планшеттер сияқты әртүрлі сипаттамалары бар виртуалды құрылғылардың кескіндерін басқара алатын өзінің эмуляторы бар.

Виртуалды android құрылғысын (эмулятор) құру Android Studio-да Android құрылғысының белгілі бір түрінің конфигурациясын модельдейтін виртуалды құрылғыны (эмулятор деп те аталады) жасауға арналған Android Virtual Device (AVD) manager утилитасы бар.

Бірінші қадам-виртуалды құрылғыны сипаттайтын конфигурация жасау.

Android Studio негізгі мәзірінде Құралдар > Android > AVD Manager тармағын таңдаңыз немесе тулбардағы AVD Manager белгішесін басыңыз тулбардағы AVD Manager белгішесін басыңыз + Create Virtual Device түймесін басыңыз. (Егер виртуалды құрылғы бұрын жасалған болса, терезеде барлық бар құрылғылар көрсетіледі және +Create Virtual Device түймесі төменгі жағында орналасқан.) Select Hardware тілқатысу терезесінде аппараттық құрылғылардың алдын ала конфигурацияланған түрлерінің тізімі көрсетіледі.

Nexus 5 сияқты құрылғыны таңдап, Келесі түймесін басыңыз.

System Image терезесінде, Recommended қойындысында қажетті нұсқаны таңдаңыз. Егер жүктеу сілтемесі нұсқаның жанында көрсетілсе, онда ол әлі орнатылмаған және оны жүктеу қажет. Қажет болса, жүктеуді бастау үшін сілтемені нұқыңыз және жүктеу аяқталғаннан кейін Келесі түймесін басыңыз.

Келесі тілқатысу терезесінде әдепкі мәндерді қабылдап, Аяқтау түймесін басыңыз. Егер сіздің Virtual Devices AVD manager терезесі әлі ашық болса, оны жабыңыз. Мүмкін проблемалар егер сіздің компьютеріңізде Windows амалдық жүйесі және AMD процессоры болса, онда сіз виртуалды құрылғыға арналған жабдықты таңдау терезесінде осындай ескертуді көресіз. X86 типті виртуалды құрылғылардың суреттері Windows + AMD байламында жұмыс істемейді. Сонымен қатар, Linux + AMD байламында мұндай проблемалар жоқ. Intel процессорлары кез-келген операциялық жүйеде x86 кескіндерін қолдайды. Windows + AMD үшін arm типті кескіндерді қолдануға болады, бірақ олар әлдеқайда баяу. Шығу нақты құрылғыны – Android смартфонын немесе компьютерге қосылған планшетті пайдалану болады.

Тағы бір жиі кездесетін мәселе-процессордың виртуализациясы өшірілген. Виртуализацияны биоста қосуға болады, интернетте сіз бұл туралы көптеген нұсқауларды таба аласыз. Сондай-ақ, SDK Tools қойындысындағы SDK менеджерінде жақсы өнімділік пен жұмыс жылдамдығы үшін Intel x86 Emulator Accelerator бағдарламасын орнату қажет. Тек Intel процессорлары үшін. Егер жүйелік кескінді таңдау терезесінде Орнату үшін белсенді кескіндер болмаса, SDK менеджеріне қажетті Android нұсқасының кескінін жүктеңіз. SDK manager терезесін ашып, төменгі оң жақтағы SDK Platforms қойындысында платформаның барлық компоненттерін көрсету үшін Show package details жалаушасын орнатыңыз (4-суретті қараңыз). Жүктеу үшін қажетті System Image файлы белгілеңіз. Жүктелгеннен кейін таңдалған кескін AVD Manager терезесінде қол жетімді болады.

Жоғарыда айтылғандар, IT мамандарының қызмет көрсетуге арналған мобильді қосымшаны әзірлеуге болатын жолдары көрсетілген. Бағдарламалармен жұмыс жасай отырып, Android жүйесінде қалай оындау барысы зерттеліп, орындалып ұсынылып отыр.

Қорытындылай келе, мобильді қосымшаны құру оның сәтті болуын қамтамасыз ету үшін мұқият жоспарлауды, жобалауды, кодтауды және тестілеуді қажет етеді. Бұл процесс күрделі болуы мүмкін, бірақ сонымен бірге өте пайдалы. Табысты мобильді қосымша өз аудиториясымен қарым-қатынас жасаудың жаңа және инновациялық тәсілдерін ұсына

отырып, бизнеске, ұйымдарға және жеке тұлғаларға үлкен пайда әкелуі мүмкін. яғни IT мамандарының қызметтерінің сұранысы жылма-жыл артып келеді. Сонымен қызмет көрсету орталығы барлық маман иелеріңе қажеттілігі туады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 [https:// www.kaznu.kz](https://www.kaznu.kz)
- 2 <https://stud.kz> > referat > show
- 3 <https://www.travelpayouts.com/blog/how-to-create-a-successful-mobile-application/>
- 4 <https://www.techtarget.com>
- 5 <https://www.nilebits.com/blog/2022/07/mobile-application-development/>
- 6 <https://maddevs.io/blog/developing-a-mobile-application-from-a-to-z/>
- 7 <https://www.nomtek.com/blog/mobile-app-development-process>
- 8 <https://medium.com/nerd-for-tech/an-ultimate-guide-for-mobile-app-development-ff3c17ece87a>
- 9 <https://polygant.net/mobiledev/mobile-application-development/>
- 10 <https://blog.tubikstudio.com/the-ultimate-guide-to-creating-a-mobile-application/>
- 11 <https://cyberleninka.ru/article/n/social-features-of-mobile-application-development>
- 12 <https://cyberleninka.ru/article/n/social-features-of-mobile-application-development>
- 13 [https://cloud3.college.edu.kz/uploads/ 741118401756/f6075040871a2e0.pdf](https://cloud3.college.edu.kz/uploads/741118401756/f6075040871a2e0.pdf)

RESUME

The development of modern information technology, the use and development of mobile applications through mobile devices for Communication at any time, access to the necessary information, and informing society as a whole is becoming very important. The development of information technologies for mobile devices is being actively implemented. The article discusses the classification of mobile applications by use, the processes of their creation and their structure. In addition, the article emphasizes the importance of mobile devices and applications. Some tools for creating mobile applications are analyzed, examples of their use are given. The process of developing mobile applications is defined. Recommended when programming mobile applications. Four main types of mobile application testing are described. In addition, the article provides an analysis of existing tools for developing mobile applications.

РЕЗЮМЕ

Развитие современной информационной технологии, поддержание связи в любое время, доступ к необходимой информации, использование и развитие мобильных приложений с помощью мобильных устройств для информатизации общества в целом являются очень важными. Активно реализуется разработка информационных технологий для мобильных устройств. В статье рассмотрена классификация мобильных приложений по использованию, процессам их создания и их структуре. Кроме того, в статье подчеркивается важность мобильных устройств и приложений. Проанализированы некоторые инструменты разработки мобильных приложений, приведены примеры их использования. Определен процесс разработки мобильных приложений. Рекомендуется при программировании мобильных приложений. Описаны четыре основных типа тестирования мобильных приложений. Кроме того, в статье представлен анализ существующих инструментов для разработки мобильных приложений.