



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

Аэротехника және аппаратив технологиялар факультеті  
Есептеу техникасы және бағдарламалық жүйелер мамандығы  
Компьютерлік технологиялар кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Күрсінбея Төлеби Асқарұлы  
(аты - жөні)

Жоба тақырыбы Android платформасында бағдарламалау тәсілдеріне арналған электрондық оқу материалдары құрастыру  
ректордың «10» наурыз 2016 ж. №21 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «8» 06 2016 ж.

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

Android операциялық жүйесіне негізделген қосымша құралдар. Бұл қосымша бағдарламалау тәсілдерінің үйренісі келетін адамдарға оқу үрдісін жеңілдету үшін арналған.

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

- мобильді қосымша құралға қажетті техникалық, бағдарламалау құралдар
- қолданушы интерфейсімен, қосымша құралдар
- жүйе арқындағы тиімділік қауіпсіздігі;
- техникалық-экономикалық негізделуі;





Диплом жобасын дайындау

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Негізгі бөлім	21.01.16	
1.1	Android операциялық жүйесі	22.01.16	
1.2	Android қосымшалары негізгі түрлері	25.01.16	
2.	Әңгілеу құралдары	20.03.16	
2.1	Қосымша графикалық қабаттасқан әңгілеу құралдары	20.03.16	
2.2	Қосымша кодты әңгілеу құралдары	20.03.16	
3	Қосымшаны құру сатысы	16.04.16	
3.1	Проекттеу	16.04.16	
3.2	Қосымшаны құру	16.04.16	
4	Тіршілік қауіпсіздігі	16.04.16	
4.1	Программалау өңірінің жұмыс жағдайына талдау	03.05.16	
5.	Технико-экономикалық негізделуі	03.05.16	
5.1	Күлдетім мақсаты	03.05.16	
5.2	Еңбектен жұмыста пайдаланатын қабаттар	12.05.16	
5.3	Күлдетім қолданатын кодтар	22.05.16	
5.4	Пайдаланатын бағдарламалар	24.05.16	
5.5	Көлемді талдау	25.05.16	
5.6	Дәулеттің салық шығындарын есептеу	25.05.16	
5.7	Амортизацияны есептеу	26.05.16	

Тапсырманың берілген уақыты « 15 » 02 2016 ж.

Кафедра меңгерушісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Жоба жетекшісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Орындалатын тапсырманы қабылдаған студент

(қолы)

(аты-жөні)

## **Аннотация**

В данной дипломной работе рассмотрена разработка мобильного приложения для операционной системы Android, приложение «Language Helper» предназначена для познания языков программирования. В дипломной работе предусмотрена создания базы данных и интерфейса мобильного приложения. Условия и требования по отношению к функционалу и интерфейсу приложения были разделены.

В разделе безопасности жизнедеятельности было проанализировано условия работы, план эвакуации персонала и освещение комнаты.

Также составлено экономическое обоснование проекта, подтверждающее его экономическую целесообразность.

## **Annotation**

In this diploma work examined the development of mobile applications for the operating system Android, the app «Language Helper» designed for learning programming languages. The research paper provides a database and mobile application interface. The conditions and requirements with respect to functionality and interface of the application have been divided.

In terms of health and safety section was analyzed the work plan for the evacuation of staff and room lighting.

Also composed the economic justification of the project, confirming its economic viability.

## **Аңдатпа**

Бұл дипломдық жұмыста Android операциялық жүйесінде жасалынған, бағдарламалау тілдеріне арналған анықтама «Language Helper» мобильдік қосымшасы жобаланған. Дипломдық жұмыста, қосымшада қолданылатын деректер қоры мен қосымша интерфейсін құру жолдары қарастырылған. Функционалдық сипаттамалар мен техникалық құралдарға тиісті қойылатын шарттар мен талаптарға бөлінген.

Тіршілік қауіпсіздігін қамтамасыз ету бөлімінде еңбек ету жағдайына талдау жасалынып, жарықтандыру және эвакуациялану жағдайлары қарастырылған.

Сондай-ақ, жобаның экономикалық тиімділігін бекітетін экономикалық негіздемесі құрастырылды.

## Мазмұны

Кіріспе	7
1 Негізгі бөлім	9
1.1 Андроид операциялық жүйесі	9
1.1.1 Android құрамдас қосымшалары	10
1.1.2 Интерфейсті әзірлеу	12
1.1.3 Android программалау ортасының негізгі сипаты	19
1.1.4 Әзірлеуші фреймворкі	20
1.2 Android қосымшалары неден тұрады	20
1.2.1 Android қоралары	22
1.2.2 AndroidManifest.xml дегеніміз не?	24
1.2.3 Android қолданушылар интерфейсі	25
2 Өңдеу құралдары	29
2.1 Қосымша графикалық қабықшасын өңдеу құралдары	29
2.2 Қосымшаның кодын өңдеу құралдары	37
3 Қосымшаны құру сатысы	45
3.1 Проектілеу	45
3.2 Қосымшаны құру	49
4 Тіршілік қауіпсіздігі	53
4.1 Программалау офисінің жұмыс жағдайының талдауы	53
4.1.1 Өрт қауіпсіздігі	55
4.1.2 Эвакуацияның уақытының есептеуі	58
4.1.3 Есептелуі	59
5 Техничко-экономикалық негізделуі	62
5.1 Жұмыстың мақсаты	62
5.2 Еңбектің жұмыста пайдаланатын қамбаларының	62
5.3 Жұмыста қолданылатын жабдықтар	62
5.4 Пайданылатын бағдарламалар	63
5.5 Жалақы төлемі	65
5.6 Әлеуметтік салық шығындарын есептеу	68
5.7 Амортизацияны есептеуі	68
5.8 Энергия шығындарын есептеу	69
5.9 Үстемені санау	70
5.9.1 Жобаның жалпы құны	70
5.9.2 Жобаның бағасы	71
Қорытынды	73
Қолданылған әдебиеттер тізімі	74
А қосымшасы	75
Б қосымшасы	83

## Кіріспе

Қазіргі заманда мобильді қосымшаларды әзірлеуші бағдарламаушылар ретінде біздің алдымызда көптеген, алдын болмаған мүмкіндіктер пайда болып жатыр. Осыдан 7-8 жыл бұрын ешкім көз алдына, мобильді қосымшаларды әзірлеушінің өз қосымшаларын бүкіл әлемдегі миллиондаған және миллиардаған адамдарға сату мүмкіндігін елестете алмаған. Адамдардың бұл қосымшаларды ұнатып оның пайдалығын ескертіп сатып алуында мүмкін деп ойламаған. Әр түрлі ақпараттар көздерінің есептеулері бойынша 2011 жылы әлемдік нарықта мобильді қосымшалар 9 – 12 миллиард АҚШ долларын құраған. Бұл көрсеткіш қазіргі заманға дейін 5 есе өсіп 45 – 60 миллиард АҚШ доллары шамасында болып жатыр. Бұл мобильді қосымшалардың қарқынды өсу себебінің бірі смартфондарға операциялық жүйе жасаушылар Microsoft – Windows Phone платформасымен, Apple – IOS және Google – Android платформалары арасындағы үлкен бәсекелестік болып табылады. Осыған орай жоғарыда айтылған платформалардың қарқынды дамуына әкеледі, ал бұл жақын арада мобильді қосымшалар әзірлеушілердің қажеттілігін 4 – 5 есе арттырады.

Android платформасының қандай ерекшеліктері бар? Google – дың ең басты, негізгі ойының бірі бұл компания өз операциялық жүйесінің кодына қолжетімділікті ашып қойған. Өте ыңғайлы аспаптар жинағы және SDK құжаттар жинағы жақын арада бұл Android платформасында көптеген мобильді қосымшалардың пайда болуына түрткі болады. Android платформасы мобильдік телефондар нарығын қарқынды жаулап алу үстінде, Windows phone және IOS сияқты үлкен корпорацияларды артында қалдырып бәсекелесті жеңу үстінде. Android тек қана мобильді телефондарда емес сондай-ақ планшеттер, нетбук, теледидар және электронды қол сағаттарына да адаптация жасалынған. Көптеген бағдарламалаушылар бұл операциялық жүйені автокөліктерге енгізуге жұмыс жасау үстінде.

Google Android-тың даму үстіндегі ең үлкен қадамдарының бірі 2008 жылдың қазан айында Play Market онлайн магазин қосымшасының пайда болуы болып табылады. Play Market-те ойындар, анықтамалар, аударма және басқа да мыңдаған, миллиондаған қосымшаларды өз мобильді телефоныңызға жүктеп алуыңызға болады. Осыдан басқа қосымша әзірлеушілерге өз қосымшаларын ақылы қылып қоюына және жарнама орнатуына мүмкіндік бар, бұл жағдай қосымша әзірлеушілерге бұл платформаны тіпті қызықты етіп көрсетеді.

Алдын, Android жүйесінде crash жүйесімен, қате оптимизациялау және тағыда басқа көптеген қиыншылықтар туындайтын еді. Бағдарламалау android V.2 шыққанға дейін өте қиын болатын, бірақ android V.2 шығарылғаннан бері өте үлкен өзгерістер болып қателер мен қиыншылықтар алдындағылармен салыстырғанда «жоқ» болды деп айтуға болады. Осы уақыттан бастап

Android-тың әр жаңа операциялық жүйесінің шығуы Google-дың өз платформасын максималды жақсартқысы келетіндігін көрсетті.



## 1. Негізгі бөлім

### 1.1 Android операциялық жүйесі

Android – жаңа дәуірдегі операциялық жүйелердің бірі, заманауи мобильді телефондармен, аппараттық жүйелерімен жұмыс жасауға арналған. Бүгінгі күнде Windows mobile, Apple iPhone және Palm Pre сияқты компаниялар қарапайым және тез жұмыс жасайтын мобильді қосымшаларды жасайтын программалау ортасын ұсынуда. Бірақ, Android-тан қарағанда бұл патентталған операциялық жүйе, бұл жерде басқа бағдарламаушылардың қосымшаларына қарағанда операциялық жүйеде орнатылған қосымшаларға көп мән бөледі. Одан басқа бұл операциялық жүйелер, қосымшалардың мобильдік телефондағы деректермен жұмыс жасауын шектеп, осы платформаға әзірленген қосымшалардың таратылуын қадағалап, бақылап отырады.

Android, мобильді қосымшаларға жаңа мүмкіндіктерді қолжетімді Linux ядросында құрастырылған ашық бағдарламалау ортасы ретінде беріп отыр. Бұл жердегі барлық қосымшалардың құрылғы аппараттары жайлы деректер ала алады, осы үшін арнайы API-сериялы кітапхана қолданылады. Одан басқа бұл арада қосымшалардың толық және қадағаланатын қатынас түрі қарастырылған.

Android платформасында барлық бағдарламалар бірдей мәртебеге ие. Шеткі қолданбалар да, басынан орнатылған қосымшалар сияқты сол API-да жазылған, оның үстіне барлық бағдарламалардың орындалу уақыты бірегей болады. Қолданушылар бастапқы орнатылған бағдарламаларды жоюына немесе өз қалауындағы бағдарламаға өзгертуіне болады, ол тіпті нөмір теретін айна болсын немесе жұмыс тақтасы болса да.

Жалпақ тілмен айтқанда Android-ты үш компоненттің комбинациясы деп елестетуге болады:

1. Қолжетімді операциялық жүйе, ашық бастапқы кодымен;
2. Мобильдік қосымшаларды әзірлеу ортасы, өзінің ашық бастапқы кодымен;
3. Android проекциялық жүйесі орнатылған мобильді телефондар және оларға арнап әзірленген мобильді қосымшалар.

Android өзіне бірнеше қажетті және өзара байланысты элементтерді қосады:

- Операциялық жүйемен үйлесімділігін кепілдеу үшін, аппараттың қамтамасының референс-дизайны мен мобильдік құрылғыға қажеттілік тізімі;
- Мобильді құрылғылармен жұмыс жасауға оптимизацияланған Linux операциялық жүйесінің ядросы өзімен қатар төмен деңгейдегі интерфейс осы аппаратты қамтаманы, жадыны және үрдістерді бақару элементі;

- Қосымшаларды құру үшін SQLite, WebKit, OpenGL және медиа-менеджер кітапханаларының ашық алғашқы коды;
  - Android функционалына жауап беретін қосымшалардың орындалу ортасы өзіне Dalvik виртуалды машинасы және кітапхана ядросын қосады;
  - Орындалу ортасы өзінің шағым өлшемімен ерекшеленеді. Бұл мобильді қосымшаларды пайдалы қолдануына жол береді;
  - Қосымша деңгейінде жүйелік қызметке қолжетімділікті қамтамасыз ететін бағдарламалық компонент жиынтығы; бұның арасында айна менеджері және орналасу менеджері, контент-провайдер, телефония мен сенсорлық дисплеймен жұмыс жасау мүмкіндігі;
  - Қосымшаларды орналастыру және қосу үшін қолданушылар интерфейсінің компоненттер жинағы;
  - Жалпы бағдарламалық жиында жеткізілетін алдын орнатылған қосымшалар;
  - Өзіне аспап, плагин және анықтама құжаттарын қосатын мобильді қосымшаларды әзірлеуге арналған бағдарламалар комплекті;
- Мән бөлетін жайдың бірі Android ашық архитектурсы қолданушылар интерфейсі мен орнатылған дизайн қосымшаларының кез-келген қателерін түзеуге болады. Android мобильді қосымшаларға арналған өз интерфейсін жасап шығару мүмкіндігін ұсынады және де өз талабыңызды қанағаттандыратын функционалы және дизайны бар қосымша мүмкіндіктерін ұсына алады.

### **1.1.1 Android құрамдас қосымшалары**

Android жүйесіндегі телефондар алдын орнатылған бағдарламалар жиынымен қамтамасыздандырылған, олар Android Open Source Project (AOSP) проект аясында жасап шығарылған.

Ең бастыларын атап өтейік:

- E-mail клиент;
- SMS пен жұмыс жасау қосымшасы;
- Жеке деректермен, күн тізбе және мекенжай кітабымен басқару үшін толық аспаптар жинағы;
- WebKit қорындағы браузер;
- Музыкалық ойнатқыш және фотогалерея;
- Калькулятор;
- Жұмыс тақтасы;
- Оятқыш(будильник);

Сондай-ақ көп жағдайларда Android келесі келтірілген Google-дың лицензионды операциялық қамтамаларын өз енішіне енгізеді:

- Android патформасына жасап шығарылған басқа да қосымшаларды жүктеуге арналған Play Market қосымшасы;

- Толыққанды Google Maps қосымшасы, өз енішіне Street View (Көшені қарай), Driving Directions(Жүру жолын көрсету), маршрутизацияланатын навигация, спутникалық карта және жол кептелісі жайлы деректер;
  - Поштамен жұмыс жасау қосымшасы Gmail;
  - Тезарада хабар алмасу қосымшасы Google Talk;
  - Youtube сервисімен жұмыс жасау видеоплеері.
- 1.1 Суретте ОС Android архитектурасы көрсетілген.



Сурет 1.1 – ОС Android архитектурасы

1.1 Суретке қарай отырып ОС Android-тың келесі деңгейлерді бақылауға болады:

- Қосымшалар деңгейі (Applications) – Android құрамына базалық қосымшалар кіреді: SMS, күнтізбе, карталар, браузерлер және тағы да басқа;
- Қосымша қаркасы деңгейі (Application Framework) – Android API-дың толық күшін қолдануға мүмкіндік береді;
- Орындау орта деңгейі (Android Runtime) – Android құрамына ядро кітапханасының жиыны кіреді, олар Java тілінің функционалының үлкен бөлігін ұсынады;
- Linux ядро деңгейі (Linux Kernel) – Android ОС Linux 2.6 да құрастырылған сол себепті оған ядроның жүйелік қызметі, жадыны басқару және үрдісті басқарулар ашық болады;

Қосымша әзірлеушілерге келесі кітапханалар қолжетімді:

- Bionic – стандартты функция кітапханасы;
- SSL – шифрлау;
- Media Framework (MPEG4, H.264, AAC, PNG, AMR, JPG, OpenCORE, MP3, PacketVideo).
- Surface Manager;
- LibWebCore (WebKit базасында);
- SGL – 2D графика;
- OpenGL ES – 3D-кітапхана;
- FreeType – шрифттер;
- SQLite – жеңіл ДҚБЖ;

Қарапайым Linux қосымшаларымен салыстырғанда Android қосымшалары кейбір қосымша ережелерге бағынады:

- Content Providers – қосымшалар арасындағы деректер алмасу;
- Resource Manager – XML, JPEG, PNG сияқты файлдарға қолжетімділік;
- Notification Manager – қатар күйіне қолжетімділік;
- Activity Manager – активтілікті қосымшаларды басқару;

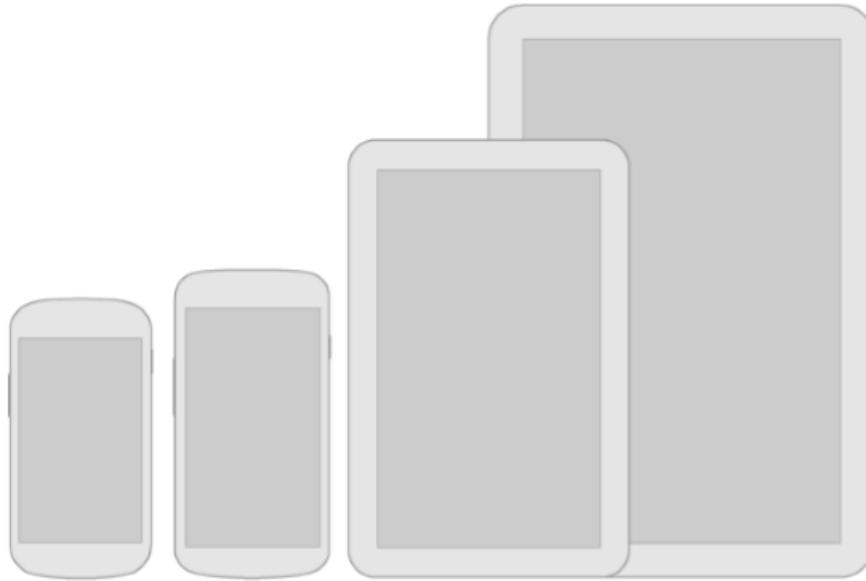
### **1.1.2 Интерфейсті әзірлеу**

Интерфейсті бекіту

Android операциялық жүйесінде жүз миллиондаған телефон, планшет және тағыда басқа алуан түрлі экран пішінімен және формаларымен ерекшеленетін құрылғылар жұмыс жасайды. Android операциялық жүйесінің иілгіштігіне байланысты әзірлеушілер қосымшаларды әр түрлі пішінді экрандарда бірдей әдемі көрінетін қосымшалар құруына болады. 1.2 суретте формалар мен пішіндер факторлары көрсетілген.

Бекіту темалары – қосымшаларды бірегей стильге келтіру үшін Android механизмі. Стиль элементтердің визуалды қасиеттерін анықтайды, олар өзіне қатар қолданушы интерфейсін құрайды, бұлар түс, шрифт пішіні мен үлкендігі. Бірегей стильді қатыру үшін Google Android платформасына екі жүйелік тема ұсынады оны қосымшаны жасар алдында таңдауға болады:

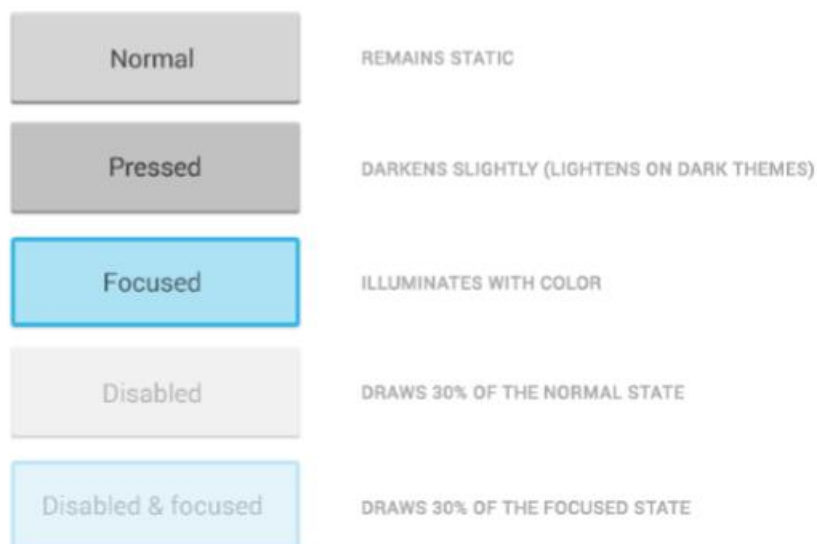
- Holo Light;
- Holo Dark;



Сурет 1.2 – Құрылғылардың форм-факторы

Өз қалауыңызға және өз қосымшаңызға сай келетін дизайн темасын таңдаңыз. Егер де өз қосымшаңызға оригиналды сұр жасағыңыз келсе, ондай болса жүйедегі бір теманы таңап ары қарай өз талғамыңызға өзгертсеңіз болады.

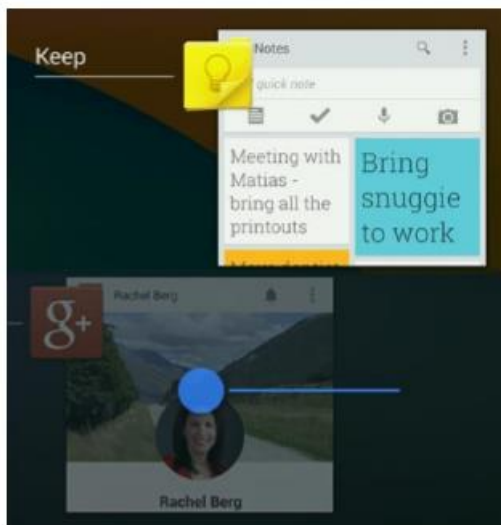
Өшрілген немесе басылатын батырмалардың активностін бейнелеу үшін әртүрлі түс, жарықтандыру және қараңғылық эффектілерін қолдануға болады. 1.2 суретте батырмалардың стандартты шаблону көрсетілген.



Сурет 1.3 - Батырма шаблондары

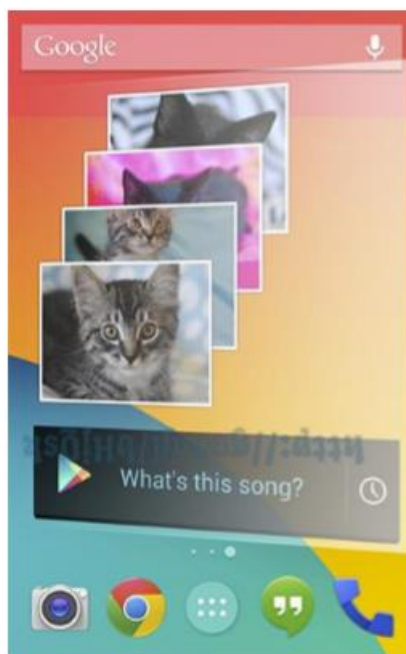


Қосымшада навигация немесе кез-келген іс-әрекет жасау үшін кез-келген күрделі қимылды (жест) қолдануыңызға болады. 1.4 суретте қимыл мысалы келтірілген.



Сурет 1.4 – Қимыл шаблоны

Интерфейсте айналдыратын элементтерді қолданған кезінде, экран шегінің эффектісін қолдануға болады. 1.5 суретте экран шегінің шаблоны көрсетілген.



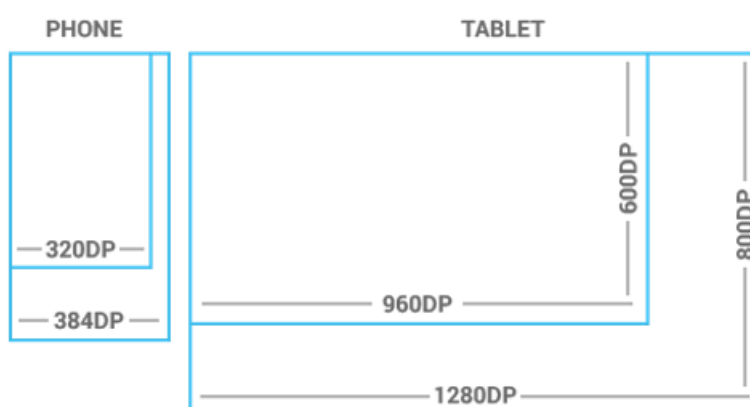
Сурет 1.5 – Экран шегінің шаблоны

Құрылғылар тек физикалық экран пішінімен ғана емес пиксельдер тығыздығымен (DPI) де айырылады. Дизайнерлердің жұмысын жеңілдету үшін алдын-ала дайындалған телефон мен планшеттерге арналған бірнеше шаблондар бар, осының арқасында әр түрлі құрылғыларда дизайнның бірдей болып шығады.

DPI әртүрлі пішіндерде болады, оларды оңай ажырату үшін олар өз келесі аттармен аталған LDPI, MDPI, HDPI, XHDPI, XXHDPI және XXXHDPI, әр прошивкада әр құрылғының өз пиксель тығыздығы деңгейі жазылған.

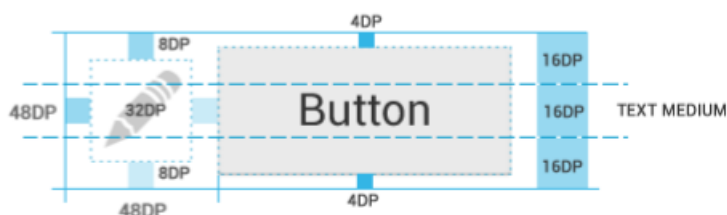
Әрдайым экран макетінің берілгендерін ескеру керек, қарсы жағдайда қосымша айғыз болып, инерфейсі құртылуы мүмкін.

1.6 суретте экран тығыздығының мүмкін нұсқалары көрсетілген.

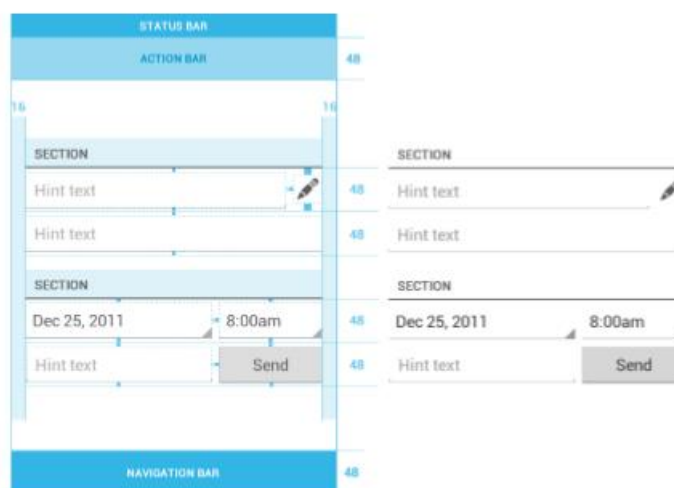


Сурет 1.6 – Экран тығыздығының макеті

Интерфейстің барлық элементтері 48x48 dp-ден бастап жоғары өлшемдерде орындалады.



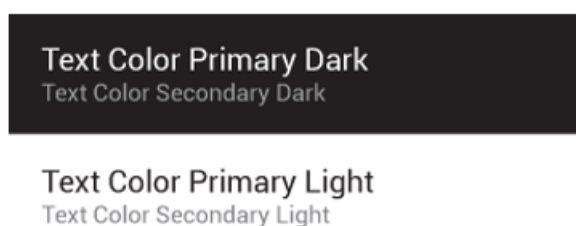
Сурет 1.7 – Белгі мен батырма мысалы



Сурет 1.8 – Интерфейстері 48dp өлшемді элемент мысалы

Android +4 версияларында жүйелік шрифт ретінде әртүрлі Roboto шрифті қолданылады (Roboto Regular, Roboto Medium, Roboto Black, Roboto Condensed, Roboto Light, Roboto Thin, Roboto Bold, Roboto Condensed Light, Roboto Condensed Bold). Android 4.4-тен бастап Roboto Condensed және оның басқа түрлер модификациясы қолданылады.

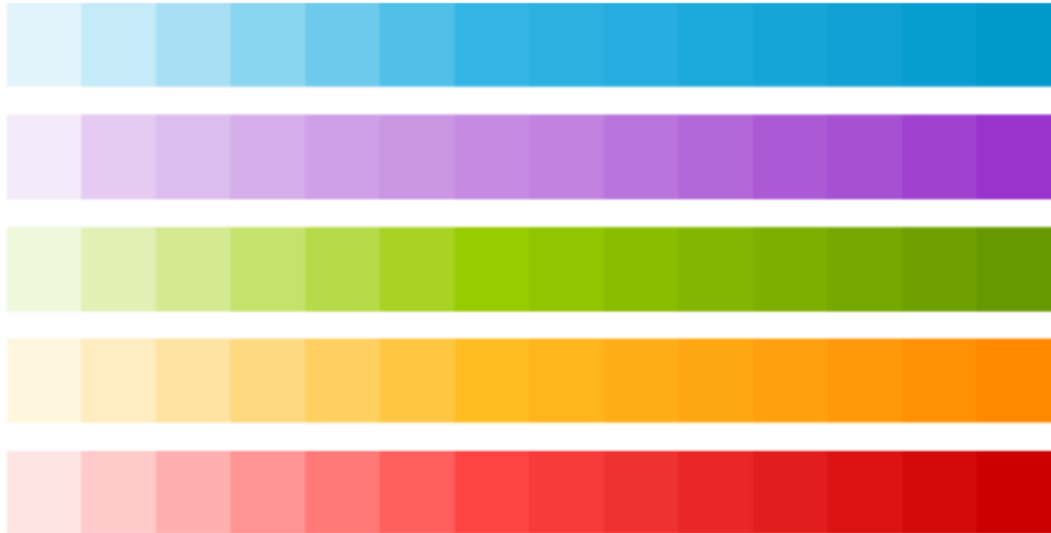
Android UI келесі түс стильдерін қолданады: `textColorPrimary` және `textColorSecondary`. Жеңіл темалар үшін `textColorPrimaryInverse` және `textColorSecondaryInverse`. Текст айналасында қолданылатын қабырғалар жарық және қара темаларды қабылдай алады, және басылған кезде өзгере алады. 1.9 суретте жарық және қара текст темалары көрсетілген.



Сурет 1.9 – Жарық және қара текст мысалдары

Түспен жұмыс жасағанда «Ноло-түстер» деп аталатын түстермен жұмыс жасаған дұрыс. Ол 1.10 суретте бейнеленген.

Қосымша белгісі – қосымшалар мәзірінде немесе жұмыс тақта экранының біріне алып шығарылған дербес графикалық ярлық, ол қосымшаны қосу үшін және оның графикалық бейнесін сипаттау қызметін атқарады. Белгіні проектилеген кезде оның әртүрлі экрандарда, әртүрлі тығыздығы бар экрандарда және Play Market желісінде орналасатынын ескеру қажет.



Сурет 1.10 – Түс палитрасы

Сол үшін экран тығыздығының макетін қолдана отырып әртүрлі пішіндегі бірнеше белгіні әртүрлі экран үшін әзірлеу керек. Бастапқы нүкте ретінде MDPI тығыздықты экранды алған дұрыс, әрі қарай пропорциясын 1,5х- ке үлкейтіп HDPI үшін, 2х XHDPI үшін, 3х XXHDPI үшін, 4х XXXHDPI үшін. Бастапқы пішін 48dp ға тең болуы қажет. Сонымен экранға MDPI тығыздығында пішіні 48x48 болады, HDPI экраны үшін – 72x72 болады, XHDPI үшін – 96x96 болады және т.б. Play Market-те жариялау үшін 512x512 пішінді сурет қажет. 1.11 қосымша белгі пішіні келтірілген

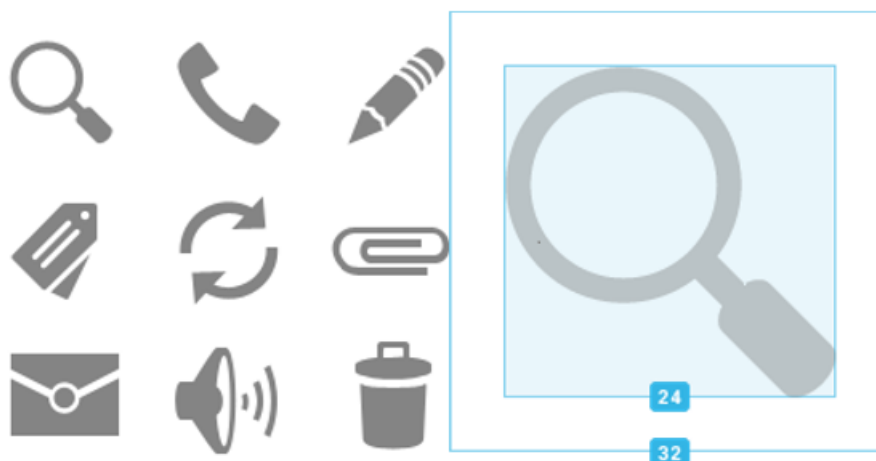


Сурет 1.11 – Белгі пішіні мысалы

Қосымша ішіндегі әртүрлі іс-әрекеттер үшін Google компаниясының өзі ұсынған Ноло - белгішелерін қолданған абзал. Бұл элементтердің физикалық пішіні 32x32 нүкте, ал көрінетін бөлігі 24x24.

Түстердің де стандартты түрін қолданған дұрыс. Ашық түс (#444444) қара фон үшін және тағыда сол сияқты пропорциялар жасаған дұрыс.

1.12 суретте белгішелердің пішіні мен олардың атқаратын әрекеттеріне байланысты көрсетілген.



Сурет 1.12 – Әрекет белгішелер мысалы

Контексті белгішеларды қолданған кезде, физикалық пішіні 16x16-ға тең болатын және визуалды аймағы 12x12 қолданған дұрыс. 1.13 суретте көрсетілгендей.



Сурет 1.13 – Контексті белгіше мысалы

Егер сіздің қосымшаңыз хабарландыру жүйесін қолданатын болса, хабарландыру орнында бейнелеу үшін 24x24 белгі қолданылуы керек.

Интерфейс элементтерін жасап жатқанда мүмкін жерлердің барлығында бейнелердің орнына векторлық элементтерді қолданған дұрыс. Бұл әртүрлі



экрандар пішіні мен пиксель тығыздығымен оптимизация кезінде қателіктерден құтылуды қамтамасыздандырады.

Файлдарға атау бергенде префикстерді қолданған абзал, сонда белгішелер директория ішінде топтасып оларды іздеу оңай болады.

Программа кодында интерфейс элементтерінің барлығын бөлек топтарға бөлген дұрыс болып саналады. Қателіктерді дұрыстағанда уақыт аз кетуіне себепші болады.

Мысалы:

```
res /...
    drawable - ldpi /...
        finished_asset . png
    drawable - mdpi /...
        finished_asset . png
    drawable - hdpi /...
        finished_asset . png
    drawable - xhdpi /...
        finished_asset . png
```

Интерфейста текст қолданған жағдайда элементтерін сипаттағанда оның көлемін максималды түрде қысқартуға және оны түсінуге оңай етуге тырысыңыз. Артық сөздерді қоспай кейбір жерлерді цифрлар көмегімен түсіндіргеніңіз жақсы.

### **1.1.3 Android программалау ортасының негізгі сипаты**

Android-тың бағдарламалау ортасы ретінде басты қазыналарының бірі оның API болды.

Android қосымшаларға нейтралды платформа ретінде сол телефонның айнытпас бір бөлігі болатындай бағдарламаларды жамап шығару мүмкіндігін береді.

Келесі тізім Android-тың негізгі қасиеттерін иллюстрациялайды:

- Лицензияны қолданудағы, қосымшаларды тарату және әзірлеудегі шығынның жоқтығы;
- Wi-Fi құрылғыларына ашық қолжетімділік;
- Телефония және деректер алмасуға арналған GSM, 3G, 4G, EDGE желілерде қоңырау және SMS қабылдауға, жібіруге болады.
- Комплексті API навигациялық қызметтермен жұмыс жасау үшін, мысалы GPS;
- Мультимедиалық құрылғылар үстінен толық бақылау, оған қоса камера мен микрофоннан алынған информацияны жазу және ойнату;
- Сенсорлық құрылғылармен жұмыс жасау үшін API, мысалы акселерометр мен компас;
- Bluetooth-бен жұмыс жасау кітапханалары сонымен бірге p2p протоколы бойынша деректер алмасу;

- IPС-хабарламаларды жіберу;
- Ортақ қорлар үшін қойма;
- Фондық қосымшалар мен үрдістер;
- Жұмыс тақтасы үшін виджеттер, тірі каталогтар (Live Folders) және тірі обойлар (Live Wallpaper);
- Қосымшаны іздеу нәтижесін жүйелік іздеуге интеграциялау мүмкіндігі;
- WebKit базасындағы кіріктірілген ашық қолжетімді коды бар және HTML5-ті қолдайтын браузер;
- Қолданушылар интерфейсінде карталармен жұмыс жасау функционалын қолданатын қосымшалар;
- Видеофайлдар, аудио мен суреттерді ойнататын мультимедиалық кітапханалар;
- Динамикалық қорлармен жұмыс жасау үшін аспаптар көмегімен локализациялау;
- Бағдарламалық компоненттердің компоненттерді қайта пайдалануға қолдану және үйлестірілген қосымшалардың орнын ауыстыру жиыны;

#### **1.1.4 Әзірлеуші фреймворкі**

Android платформасына мобильді қосымшаларды бағдарламалау тілі Java. Алайда бұл Java-ның классикалық виртуалды машинасы Java VM – да орындалмайды, бұл арнайы Dalvik виртуалды машинасында орыдалады.

Android-тағы әрбір қосымша өз Dalvik машинасы экземпляр ішінде бөлек үрдіс болып орындалады. Жадыны және үрдістерді басқару жауапкершілігі толық Android-қа тәуелді, ол өзімен қатар ресурстарды босату қажеттілігі болса үрдістерді тоқтатады немесе толықтай жояды.

Dalvik және Android Linux ядросының жоғарғы сатысында орналасқан, ол аппараттық қамтамасыздандырумен төмен дәрежеде қарым-қатынас құрады, оған қоса жадыны басқару мен драйвер жұмысымен айналысады. Осыған қоса кіріктірілген API жиыны барлық қызметтер түріне, функционалға және аппаратты салындыларға қолжетімділік алуға мүмкіндік береді.

#### **1.2 Android қосымшалары неден тұрады**

Android қосымшалары өзара әлсіз байланысқан компоненттерден тұрады, бұл компоненттер бағдарламалық манифест көмегімен бірге жинақталады. Манифест – файл, бұл қосымшаның барлық компоненттерін, олардың байланыс түрін және метаберілгендерді оған қоса платформаға деген талаптарын және аппарат конфигурациясын бейнелейді.

Келесі келтірілген компоненттер қосымшалар құралатын кірпіштер деп айтуға болады.

*Активтілік.* Қабылдау дәрежесі. Әр қосымша экраны Activity классынан мұраланады. Активтіліктер қолданушылармен өзара әрекеттесетін және хабарламалар шығаратын графикалық интерфейсін құру үшін қолданылады. Платформа қызметін дамыту тұрғысынан Активтілік – Форма эквиваленті. Түсінікті тілмен айтқанда Активтілік графикалық сызбалар, текстті хабарламалар және т.б. хабарлау, ақпарат беру түрлері орналасатын аймақ.

*Қызмет көрсету.* Қосымшаңыздың көрінбейтін қозғалтқышы. Қызмет көрсету компоненттері фондық режимде жұмыс жасайды, хабарламаларды қосып, деректер қорын және көрінетін Активтілікті жаңартып отырады. Жүйелі түрдегі операциялар үшін қолданылады және сіздің Активтілігіңіз алдыңғы орында болмаса да жұмысын тоқтатпауы керек.

*Деректер көзі.* Деректер қоймасы. Берілген компоненттер бір қосымша шегіндегі деректер қорын басқару үшін және оған сырттан қолжетімділікті қамтамасыздандырады. Деректер көзі әртүрлі қосымшалар арасында информация алмасуда қолданысқа ие болады. Яғни бұл өзіңіздің жеке ContentProvider объектісін баптауыңызға болады және оларға басқа да қосымшалардан қолжетімділікті қамтамасыз етеді. Оған қоса сіз үшін сыртқы бағдарламалар ашқан қосымшалардың деректерін қолдану және жұмыс жасау үшін қолжетімділік. Android басқару аясында жұмыс жасайтын бұйымдар бірнеше стандартты Көзерден (Источник) тұрады. Бұл өз алдына пайдалы деректер қорына қолжетімділікті қамтамасыз етіп оған қоса мультимедиялық файлдар мен қонақты информацияларға қолжетімділікті қамтамасыздандырады.

*Ниеттілік.* Қосымшалар арасындағы хабарлама алмасу жүйесі. Ниеттілікті қолдана отырып сіз арнайы Активтілік немесе Қызмет көрсетулерге немесе жүйелік дәрежеде хабарламалар трансляцияларын жүргізу мүмкіндігі бар. Осымен берілген тапсырмаларды орындау маңыздылығы айтылып отырады. Осыдан соң Android келген тапсырыстардың орындау керек компоненттерін өзі анықтап алады.

*Виджеттер.* Басты экранға қосуға болатын визуалды бағдарламалық компоненттер. Осы айтылған Виджеттер динамикалық, интерактивтік компоненттер бола алады, бұл виджеттерді басты экранға қойып телефонды қолдану ыңғайлығын көбейтуге болады. Мысалы: сағат виджеті, ауа райы виджеті және т.б.

*Хабарландыру.* Қолданушыға хабарлау жүйесі. Орындалып тұрған Активтіліктің өзіне мән аудартпай немесе жұмысын тоқтатпай қолданушыға кез-келген хабарламаны жеткізуді қамтамасыздандырады. Хабарландыру механизмі қызмет көрсетуде қолданған өте ыңғайлы себебі қолданушы назарын өзіне аударады. Мысалы: кіріс қоңырауды немесе хабарламаларды қабылдай отырып, құрылғы қолданушыға дауыс шығарып, светодиодтары

жанып-өшіп, белгілер көрсетіп, хабарлама көрсетіп немесе вибрацияланып сізді хабардар етеді.

### 1.2.1 Android қорлары

Қорлар – жұмыс кезінде ішіне енгізілетін және қосымшаға сіздің кодыңызбен компиляцияланған сыртқы файлдар (код емес). Андроид көптеген түрлі ресурс файлдар түрін XML, PNG мен JPEG қоса қолдайды. Файлы XML файлдары олар нені сипаттайтынына қарай әр түрлі форматтары болады. Қорлар бастапқы кодта сипатталған, және XML файлдары тез және эффективті компиляция үшін екілік кодта берілген. Жолдар жадыны экономдайтын түрге сығылған.

#### *Қорлар тізімі*

Қорлар типі және оның орналасуы:

- Суреттер – “app/res/drawable/”;
- Layout – файлдары “app/res/layout/”;
- Қосымша белгішесы – “app/res/mipmap/ic\_launcher/”;
- Анимация – “app/res/anim/”;
- Стильдер, жолдар мен массивтер – “app/res/value/”;

Атаулары ерекшеленбеу керек:

- “colors.xml” – түстерді анықтау үшін;
- “arrays.xml” – массивтерді анықтау үшін;
- “styles.xml” – объекттердің стилін анықтау үшін;
- “dimens.xml” – мөлшер анықталуы үшін;
- “strings.xml” – жолдарды анықтау үшін;
- Mp3 немесе видеоларға ұқсас өңделмей қалған файлдар – “res/raw/”;

#### *Кодта қорларды қолдану*

Қорларды қолдану үшін кодта тек қордың толық ID мен ресурс қандай объектте компиляцияланғанын білу керек. Бұл қорға айналу синтаксис мысалы:

*R.resource\_type.resource\_name*

немесе

*android.R.resource\_type.resource\_name*

Resource\_type – нақты қор типі бар R подклассы.

Resource\_name – нақты XML файлардағы қорлар атрибуты немесе қор үшін файл аты. Әрбір қор типі оның типіне байланысты R подклассына қосылады.

Қосымшада компиляцияланған қорлар пакет атысыз қолданыла береді. (R.resource\_type.resource\_name сияқты). Android экран стилі мен түйме фоны сияқты стандартты ресурстары бар. Кодта оған android.R.resource\_type.resource\_name мысалды атауға болады.

```
android.R.drawable.button_background
```

### *Қорларға сілтеме*

Атрибут немесе қордағы мағына басқа қорға сілтеме бола алады. Бұл layout файлдарда жолдарды сақтау мен суреттер (басқа файлдарда тұратын), дегенмен сілтеме басқа қор типінде қолданылады.

Мысалы, егер бізде түс қоры боса, оны layout файлында жазамыз.

```
<EditText  
android:layout_width="fill_parent"  
android:layout_height="fill_parent"  
android:textColor="@color/opaque_red"  
android:text="Hello, World!" />
```

"@" префиксі ол қорға сілтеме болып табылатының көрсетеді. Мысалда біз пакетті анықтамаймыз, өйткені өзінің пакетіндегі қорға сілтейміз. Жүйелік қорға сілтеме жасау үшін:

```
<EditText  
android:layout_width="fill_parent"  
android:layout_height="fill_parent"  
android:textColor="@android:color/opaque_red"  
android:text="Hello, World!" />
```

### *Альтернативті қорлар мен оқшаулану*

Альтернативті қорлар мен оқшаулану – Android-та жақсы шешілген, маңызды мәселе. Әдетте жиі UI-ді проектилеу керек болады, ол мүмкін емес әрбір экран анықтығын бірдей алады.

Қосымша қорларды қосу үшін, қорлармен параллель папка құру керек және сызықша арқылы әрбір мәтінге, ол папка қайда орналасқанын қосу керек. Мысалы, ағылшын және неміс оқшаулануы.

```
MyApp/
```

```
Res/
```



*Values-en/  
Strings.xml  
Values-de/  
Strings.xml*

Android әрқайсысына әр түрлі мағынасы бар бірнеше спецификаторды қолдайды. Мысалы, drawable нақты конфигурация үшін қоры бар папка.

*MyApp/  
res/  
drawable-en-rUS-port-92dpi-finger-keyshidden-12key-dpad-480x320/*

Бұдан басқа, бірнеше нақты конфигурация операциясын анықтауға болады.

*MyApp/  
res/  
drawable-en-rUS-finger/  
drawable-port/  
drawable-port-160dpi/  
drawable-qwerty/*

### **1.2.2 AndroidManifest.xml дегеніміз не?**

Android-та жасалатын кез-келген қосымшада, манифест файлы болады. Ол жобаның тамырлы каталогында сақталады. Манифест қосымшаның құрылым мен таңбалы деректерін сипаттауға мүмкіндік береді.

Манифест әрбір компонентке (Белсенділік, Сервис, Дереккөздері және қабылдағыштың кең таралуы) түйінді қосады, олардан қосымша тұрады және ниеттер фильтрі (Intent Filters) көмегімен өкілеттікті анықтайды.

Манифестте таңбалы деректерді көрсету қарастырылған. Айтып ескертсек, жоғарға деңгейдің қосымша түйіндерін қауіпсіздік жөндеуін, модульді тесттерін, аппараттық және жүйелік талап етуді сипаттауға болады.

Манифест жоба пакетіне аударатын <manifest> түбір тегі package атрибутымен сипатталады.

Типтік тег <manifest> кодтың фрагментінде көрсетілген.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  package="com.example.ee.languagehelper">
```

```
<application
```

```

    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".Main2Activity"
        android:label="@string/title_activity_main2"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".Main3Activity"
        android:label="@string/title_activity_main3"
        android:theme="@style/AppTheme.NoActionBar"></activity>
</application>

</manifest>

```

Тег <manifest> тегі өзіне түйіндерді қосады. Олар қосымшада тұратын бағдарлама компоненттерін, қауіпсіздік жөндеуін, тестілеу мен талап ету класстарын сипаттайды.

### 1.2.3 Android қолданушылар интерфейсі

Қолданушы интерфейсі (UI) Android-та екі жолмен құруға болады, XML-код арқылы немесе java-коды арқылы. Қолданушы интерфейсін XML кодында құрған дұрыс болып есептеледі, себебі Қолданушы интерфейсі (UI) бағдарлама логикасынан әрқашан да бөлектеніп тұруы керек. Оған қоса бір пішінді экран түрінен екінші пішінді экранға өту бейімделулігі оңай болады. UI – ді XML-де жасау HTML-ге өте ұқсас болып келеді.

```

<html>
<head>
<tittle> Tittle </tittle>
</head>
Content.
</body>

```

```
</html>
```

Android-та барлығы жақсы структураланған және ағаш тәрідес структуралармен баяндала алады.

```
<?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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"/ >
</LinearLayout>
```

### *Экран элементтер иерархиясы*

Android қосымшаларының негізгі функционалдық модулі – Activity android.app.activity объект классы. Activity көп нәрселер жасай алады, бірақ жалғыз ол экранда мән мен мәтіндерді көрсете алмайды. Activity-ге мән мен мәтіндерді шығару мүмкіндігін беру үшін және UI-ті құру үшін, View және Viewgroups еденицаларымен жұмыс жасау қажет.

### *Views*

View – объект, android.view.view базалық классын ұлғайтады. View объектісі өзі ұсынатын жазықтықты орындайды, оның орналасу схемасын, суретті, центрдің өзгеруін, айналдыруды және батырмаларды орындайды. View классы барлық графикалық фрагменттер үшін базалық класс қызметін орындайды. Графикалық фрагменттер өз жазықтығы мен суреттерін орындайды, осыған орай сіз өз UI-іңізді тез арада құруыңызға болады. Қолжетімді графикалық фрагменттер тізімі: TextView, EditText, Button, CheckBox, ScrollView, RadioButton, LargeText, SmallText, MediumText, PlainText, Button, SmallButton, Switch, ImageButton, ImageView, ProgressBar, ToggleButton және т.б.

### *Viewgroups*

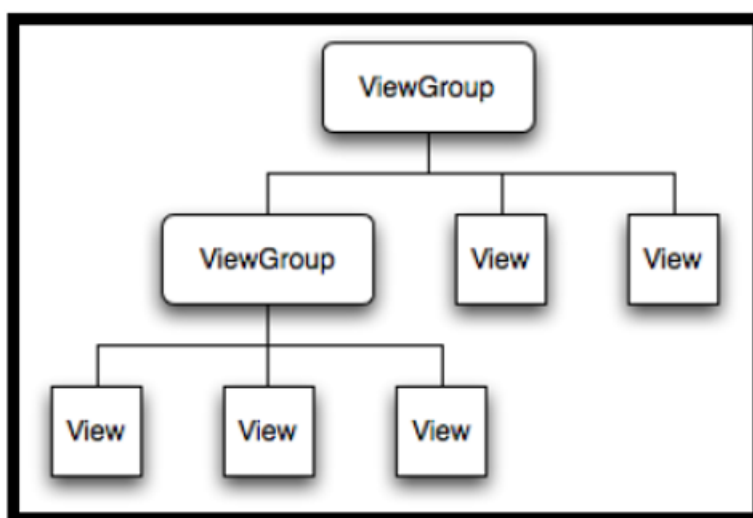
Viewgroup – android.view.viewgroup класс объектісі. Viewgroup – View объектісінің ерекше типі, оның функциясы – View және Viewgroup жиындарын орналастыру және онымен басқару. Viewgroups сізге өзіңіздің UI-ға структуралар қосуға және экранның күрделі элементтерін құруға мүмкіндік береді, оларға жалғыз объектіге қатынасқандай қатынасуға болады. Viewgroup

классы Layouts-қа базалық класс ретінде қызмет атқарады. Layouts View қатарына структура кіріктірудің мүмкіндігін береді.

### *UI ағаштәріздес структурасымен*

Android платформасында View және ViewGroup ағаштарын қолдануға UI Activity-ді анықтайсыз, төмендегі диаграммада көрсетілгендей. Ағаш сіз жасап шығарғаныңызға байланысты қарапайым немесе күрделі болуы мүмкін. Оларды графикалық фрагменттер жинын және Layouts Android, немесе View тапсырысты типтерін қолдана отырып өзіңіз жасауыңызға болады. 1.14 суретте қолданушы интерфейсінің ағаштәріздес түрі көрсетілген.

Ағашты экранға тіркеу үшін және оны есетеу үшін, сіздің Activity-іңіз өзінің setContentView() методын шақырады және тамырлы түйін объектісіне ақпараттарды жібереді. Android жүйесі ақпараттарды тамырлы түйін жүйесіне ала салысымен ол түйінмен оның ағашын есептеп өлшеу үшін жұмыс істей бастайды. Сіздің Activity-іңіз активті болып басымдылық алған соң, жүйе сіздің Activity-іңізді тіркейді және тамырлы түйінді өлшеп есептеу үшін сұраныс жібереді.



Сурет 1.14 – UI ОС Android – ағаштәріздес структура

### *Android-тың UI элементтерін UI Swing элементтерімен салыстыру*

Кейбір мобильді қосымшаларды әзірлеушілер Swing-тың UI мен ұқсас болуы мүмкін дейді, келесі тізімде Android пен Swing арасындағы ұқсас жерлері келтірілген:

- Android-тағы Activity – Swing-тегі (J) Frame-ге ұқсас;
- Android-тағы View – (J) Swing-тегі Component;
- Android-тағы TextView – (J) Swing-тегі TextField;
- Android-тағы EditText – (J) Swing-тегі TextField;

- Android-тағы Button – (J) Swing-тегі Button;

View-ға тінтуір тыңдатқышын орнату Swing-тегіге өте ұқсас.

```
// Android  
Button.setOnClickListener(new OnClickListener){  
// батырманы басқанда орындалатын функциялар  
}  
// Swing  
myButton2.addActionListener(new ActionListener){  
// батырманы басқанда орындалатын функциялар  
}
```



## 2. Өңдеу құралдары

### 2.1 Қосымша графикалық қабықшасын өңдеу құралдары

Қосымша проектісінің графикалық элементтерін құру үшін келесі келтірілген бағдарламалар қолданылады, Adobe компаниясымен құрастырылған суреттерді өзгерту және олармен басқа да операциялар жасау үшін шығарылған Adobe Photoshop CS6 бағдарламасы. Бұл бағдарламаны жүктеу үшін келесі сілтемеге өтіп жүктеуге болады.

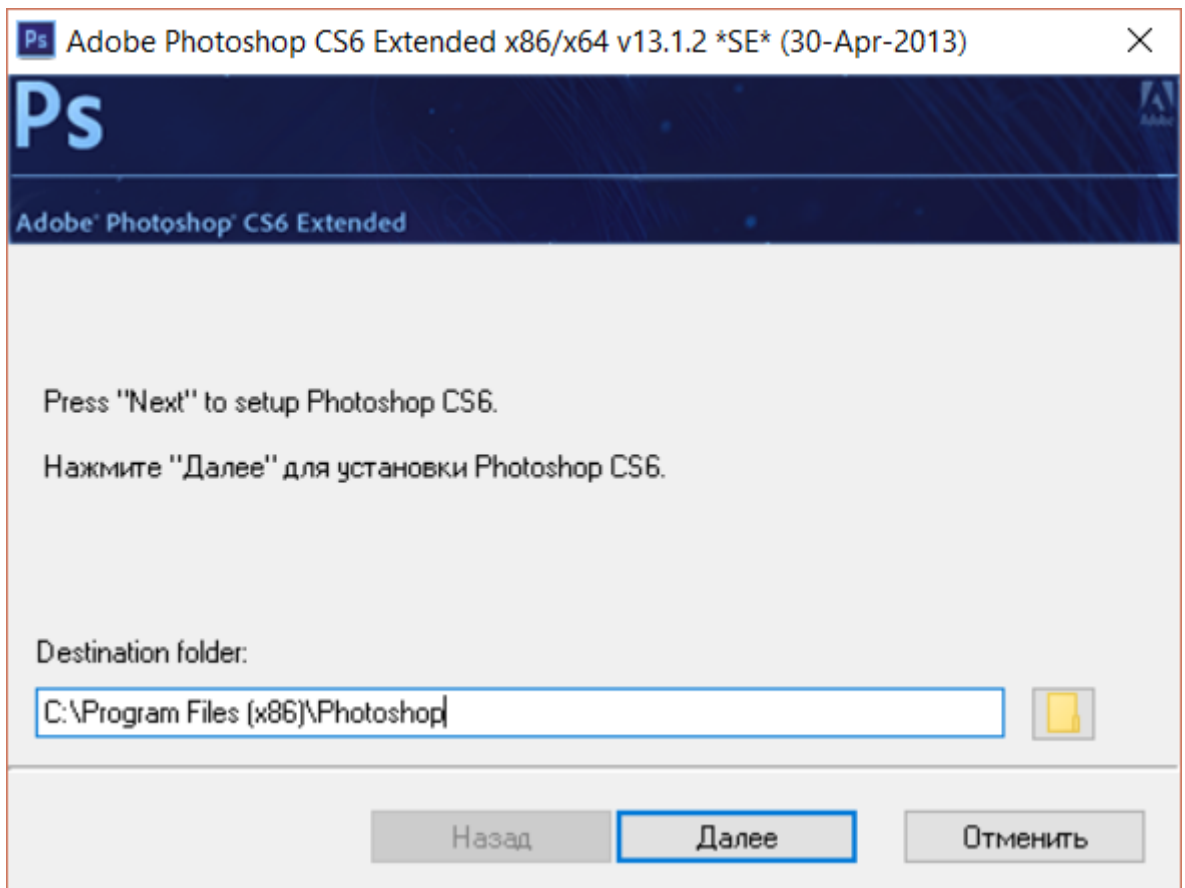
<https://www.adobe.com/support/downloads/thankyou.jsp?ftpID=5677&fileID=5701>

және бұл бағдарламаға бір жылдық жазылуға жазылып, жүктеп алуыңызға болады. Одан басқа арнайы дүкендерден орнатуға арналған DVD дискісін сатып алу мүмкіндігі бар. Өзінің ең атақты және ең мықты графикалық қалталарын біріктіру арқасында Adobe компаниясы жаңадан Creative Suite қысқашасы CS деп аталатын графикалық қалтасын ұсынды. Бұл құрамға Photoshop бағдарламасымен бірге Acrobat, GoLive, InDesign, Illustrator және Image ready бағдарламалары бар.

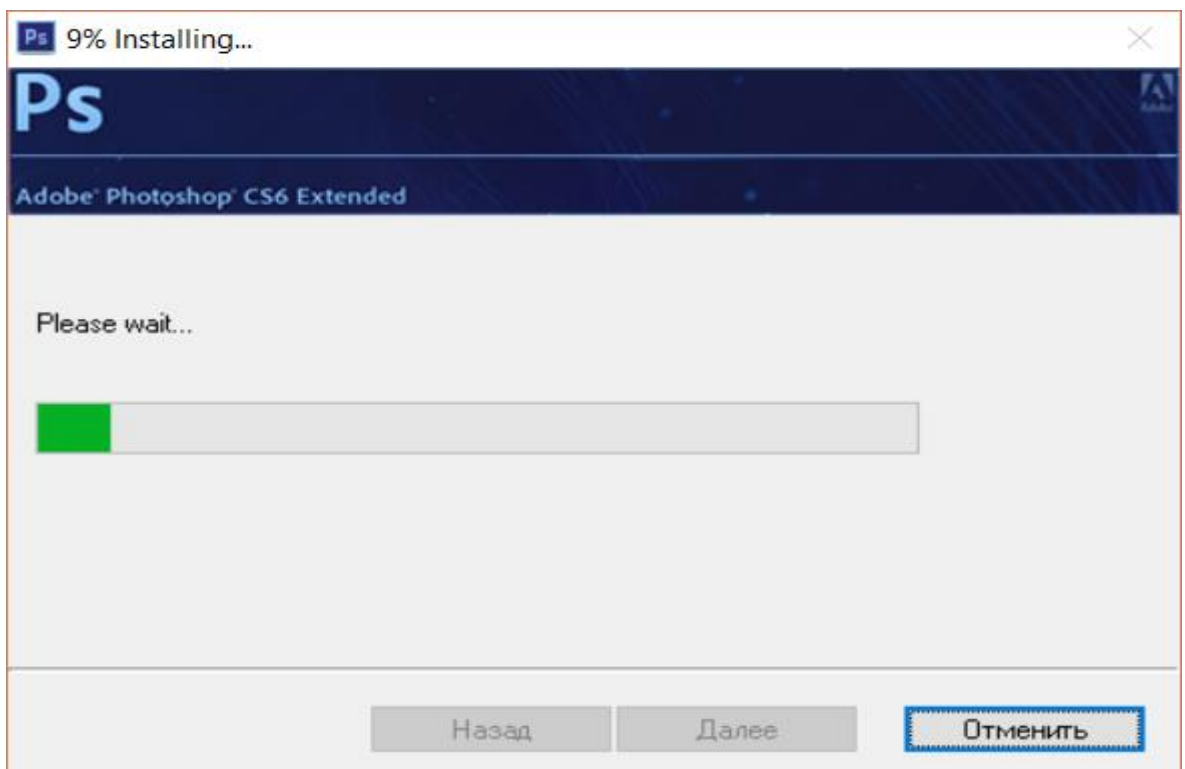
Өзінің жаңа өзгертілістерімен Photoshop бағдарламасы графикалық суреттер мен кескінді оңай, ұтымды өзгерістер енгізуге және өңдеуіне мүмкіндігін береді. Бұл нағыз ақпараттардың цифрлық өңдеулер жүргізуіне мүмкіндік беретін қосымша қалтасы. Photoshop CS-тың басты ерекшеліктерінің бірі бұл бағдарламамен кез-келген адам жұмыс жасай алады, жай қолданушыдан бастап білікті кәсіпкерлерге дейін жылдам әрі оңай орындай алады.

Adobe Photoshop бағдарламасы графикалық объектіні өңдеуге, өзгертуге, сақтауға қолданылады. Adobe Photoshop – палитралармен жұмыс жасағанда, калибровка жасауды, сканерлеуде, импорттауды және экспорттауды, контурды бөлуді, салып өзгертуді, түстер, қабаттар, каналдарды және масканы, таңдауда және т.б. әрекеттер жасау мүмкіндіктерін береді.

Жұмыс үрдісінің басы 2.1 суретте көрсетілген.



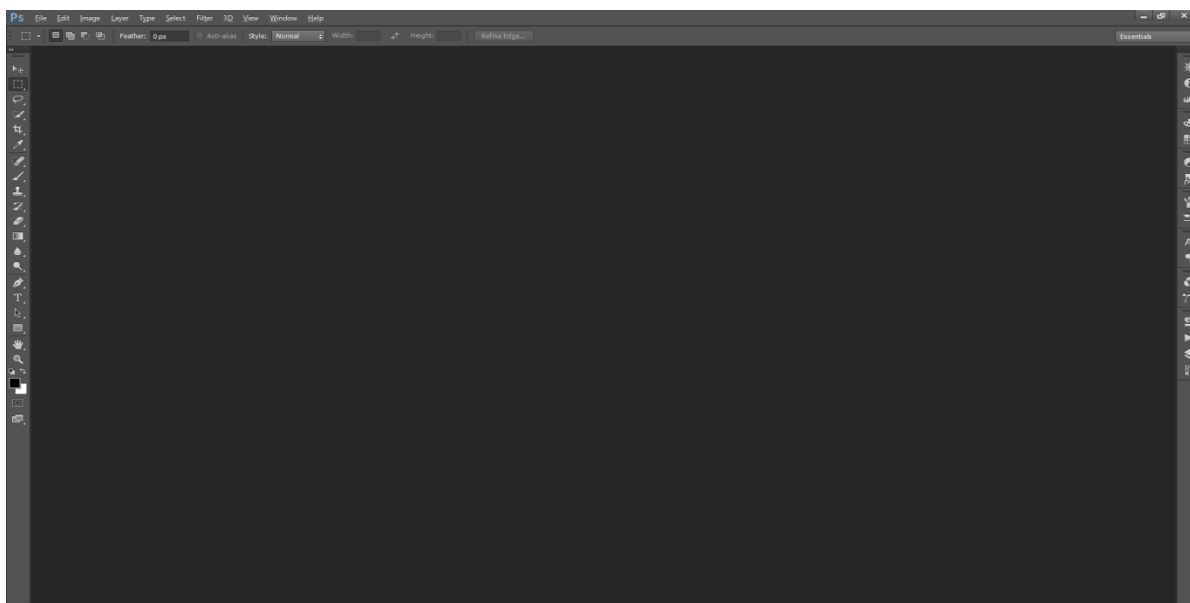
Сурет 2.1 – Бағдарламаны орнату жолын көрсету айнасы



Сурет 2.2 – Орнатылу үрдісі

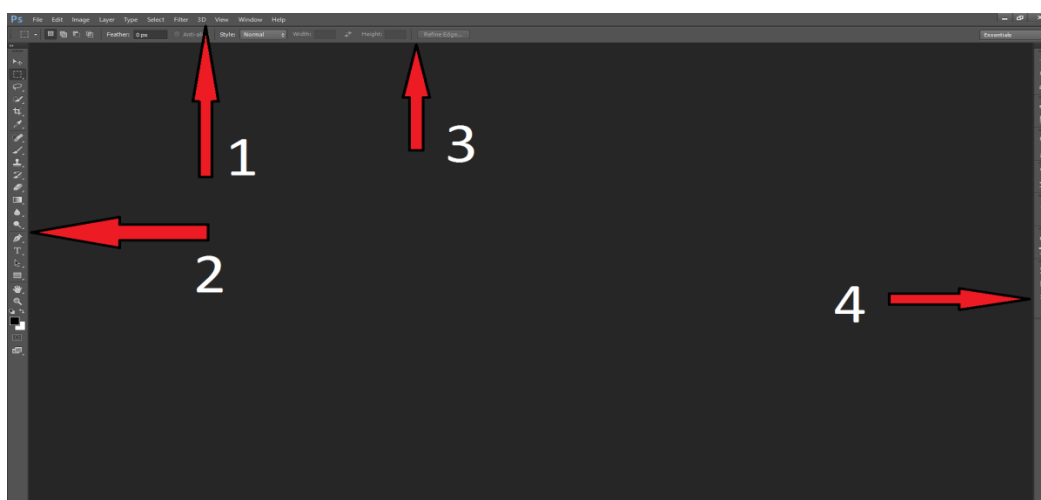
Adobe Photoshop-тың CS5, CS6, CC және т.б. түрлері бар, оған қоса олар x86 және x64 разрядты болып бөлінеді. Өз компьютеріңіздің ОЖ-сына қарай керекті бағдарламаны орындатқаныңыз дұрыс.

Бағдарлама орнатылып болғаннан кейін бағдарламаны қосуыңызға болады.



Сурет 2.3 – Қосылған Adobe Photoshop CS6 бағдарламасы

Adobe Photoshop CS6 – бұл фото редактор бағдарламалар арасындағы ең функционалы көп, ең жаңа бағдарламалардың бірі. Бұл бағдарлама дизайнерлерге көптеген мүкіндіктерді ашады және бар аспаптарды жақсартады.



Сурет 2.4 – Басты экран инструменнтер сипаты

1.

Мәзір жолағы

Мәзір жолағы терезенің төменгі бетінде орналасқан. Оның ішіне редактордың басты мәзір атаулары кіреді. Кез-келген мәзір бұйрығын таңдау үшін оның бетінен тінтуірмен басуға болады.

Егер көп нүктелері бар батырмасын басқан кезде диалог терезесі ашылады. Сол жерде кейбір керекті атрибуттар көрсетіледі. Мысалы, Файл мәзірінің Ашу командасын басқан кезде құжаттары бар диалогты терезе ашылады және қажетті құжаттарды таңдап, бағдарламағы жүктеледі.

Әртүрлі мәзірлер арқасында көптеген қиыншылықтард шешіледі:

Файл (File) – файлмен жұмыс істеу үшін қажет: құжатты құруға, ашуға және сақтауға; баспасына жібереді және ұйымдастырады; құжат импортталады және экспортталады; құжаттар туралы ақпараттарды енгізу; шығу редакторы; автоматты түрде бейнелер өңделеді;

Правка (Edit) – редакторлау: белгілеп алған облысты көшіреді, өшіреді және орнатады; қимылдарды артқа қайтарады; белгіленген жерлерді трансформациялайды;

Изображение (Image) – бейнелермен жұмыс істеу: дубляждарды жасау, холст өлшемін және бейнесін өзгертеді; оны бұру, түрлі-түсті немесе ақ-қара түсті коррекциялауға болады, түсті моделін өзгерту, түстік қақпанды құру және т.б.;

Слой (Layer) – қабатпен жұмыс жасау: құрылады, өшіріледі, дубляждалады, қабат эффектілері, қабатты топтайды, байланысқан қабатты объектілерін реттейді, қабат маскасы және т.б.;

Выбор (Select) – ерекшелік аумағымен жұмыс жасау: бейне толықтай ерекшеленеді, ерекшелер алынады және қайтарылады, ерекшелік аумағын инвенторлау және трансформаторлау, қысылу, жиіліктілік мүмкіндігі, ерекшеліктер сақталады және жүктеледі, берілген түстің аймағын ерекшелеп алу;

Фильтр (Filter) –Photoshop фильтрлары;

Вид (View) – редактордың сыртқы бетімен жұмыс жасау: құжат көрсету режимі және масштаб; бағыттауышты, торды, координатталған сызықтарды көрсету; бағыттауыштарды құрады, торға және бағыттауышқа байланыстырады;

Окно (Window) – әр түрлі құжаттармен бір уақытта жұмыс немесе әр терезеде бір құжатпен жұмыс жасау мүмкіндігі; экранға құрал панелі мен өзгеретін палитраларды шақыру;

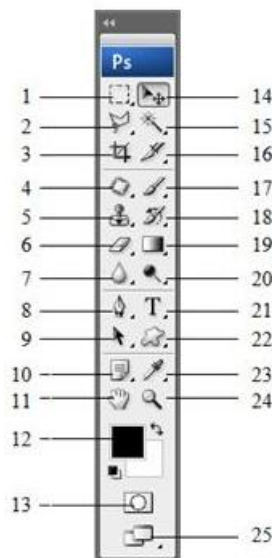
Помощь (Help) – қолданушыға үшін көмекші құжатты ашу.

2.

Құралдар панелі

Құралдар панелінің көмегімен біз сурет саламыз, бояймыз, бейненің керекті

облыстарын бөліп аламыз, оларды жылжытып өзгертеміз, мәтін енгіземіз және т.б. амалдарды орындаймыз. Жоғарғы жақта редактор терезесінің қалпын өзгертетін батырма орналасқан.



Сурет 2.5 – Құралдар панелі

1. Ағымдағы форма облыстарын ерекшелеп алуда қолданылатын құралдар :
  - Тікбұрышты облыс (Rectangular Marquee), Shift+M;
  - Эллипс типті облыс (Elliptical Marquee), Shift+M;
  - Жалғыз жол (Single Row Marquee), Shift+M;
  - Жалғыз баған (Single Column Marquee), Shift+M.
2. Облыстарды ерекшелеп алуда қолданылатын құралдар:
  - Лассо (Lasso), Shift+L;
  - Көпбұрышты лассо (Polygonal Lasso), Shift+L;
  - Магнитті лассо (Magnetic Lasso), Shift+L.
3. Шеттер бойынша қиып алу құралдары: Кадрлау (Crop), Shift+C.
4. Ретуш операциясына қажетті құралдар:
  - Қайта келтіру кисть (Healing Brush), Shift+J;
  - Заплата (Patch), Shift+J.
5. Бейне бөліктерін көшіру құралдары:
  - Клондаушы штамп (Clone Stamp), Shift+S;
  - Өрнек штампы (Pattern Stamp), Shift+S.

6. Өшіруге арналған құралдар:  
Өшіргіш (Eraser), Shift+E;  
Фонды өшіргіш (Background Eraser), Shift+E;  
Сиқырлы өшіргіш (Magic Eraser), Shift+E.
7. Дәлдікті басқару құралдары:  
Жуу (Blur), Shift+R;  
Дәлдік (Sharpen), Shift+R;  
Саусақ (Smudge), Shift+R.
8. Векторлық контурларды салу және өңдеу үшін құралдар:  
Перо (Pen), Shift+P;  
Бос форма (Freeform Pen), Shift+P;  
Тіректі нүктені қою (Add Anchor Point), Shift++;  
Тіректі нүктені жою (Delete Anchor Point), Shift+-;  
Нүктені түрлендіру (Convert Point).
9. Векторлық контурларды бөліп алу құралдары:  
Контурды бөліп алу (Path Selection), Shift+A;  
Тік бөліп алу (Direct Selection), Shift+A;
10. Комментарий құруға арналған құралдар:  
Комментарий (Notes), Shift+N;  
Аудиоаннотация (Audio annotation), Shift+N.
11. Қол (Hand) терезесінде бейнелердің орнын ауыстыру үшін құралдар, Shift+H.
12. Активті түстерді басқару үшін қолданылатын құралдар
13. Жылдам маскілеу режимінде редакторлайтын батырманы қосу (edit in quick Mask Mode), Shift+Q  
Өңдеудің стандартты режимін қосып қою құралы. (Edit in standard Mode), Shift+Q.
14. Бөлінген облыстардың орнын ауыстыру (Move) үшін қолданылатын құралдар, Shift+V.
15. Бір түсті облыстарды бөліп алуда қолданылатын құрал (Magic Wand), Shift+W.

16. Бейнені фрагментке кесуде қолданылатын құралдар:  
Фрагмент (Slice), Shift+K;  
Фрагментті бөліп алу (Slice Select), Shift+K.
17. Сурет салуда қолданылатын құралдар:  
Кисть (Brush), Shift+B;  
Карандаш (Pencil), Shift+B.
18. Бейнелердің алдында болған жағдайына келтіретін құралдар:  
Қалпына келтіретін кисть (History Brush), Shift+Y;
19. Бөлініп алынған облыстарды бояу құралдары:  
Градиент (Gradient), Shift+G;  
Бояуы бар шелек (Paint Bucket), Shift+G.
20. Жарықтылықты басқару құралдары:  
Жарықтағыш (Dodge), Shift+O;  
Қараңғылатқыш (Burn), Shift+O;  
Губка (Sponge), Shift+O.
21. Мәтін енгізу үшін қолданылатын құралдар:  
Горизонтальды мәтін (Horizontal type), Shift+T;  
Вертикалды мәтін (Vertical Type), Shift+T;  
Горизонталды мәтін маскасы (Horizontal Type Mask), Shift+T;  
Вертикалды мәтін маскасы (Vertical Type Mask), Shift+T.
22. Векторлық фигураларды салуда қолданылатын құралдар:  
Тікбұрыш (Rectangle), Shift+U;  
Дөңгелектенген тікбұрыш (Rounded Rectangle), Shift+U;  
Эллипс (ellipse), Shift+U;  
Көпбұрыш (Polygon), Shift+U;  
Түзу (Line), Shift+U;  
Қолданушы формасы (Custom Shape), Shift+U.
23. Қашықтық пен бұрыштарды өлшеуде, түстерді өлшеу және көшірмесін алатын құралдар:  
Пипетка (Eyedropper), Shift+I;  
Түстік пробалар (Color Sampler), Shift+I;  
Өлшеуіш (Measure), Shift+I.



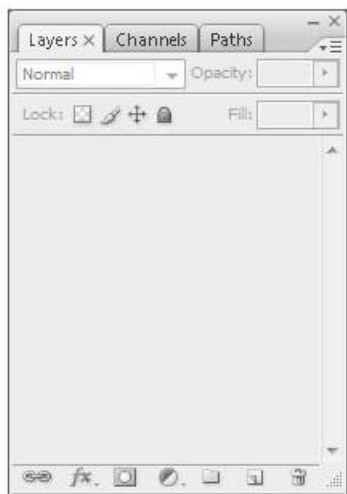
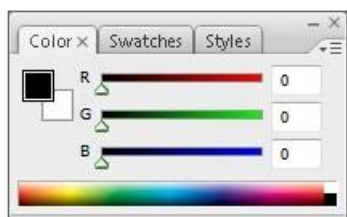
24. Бейнені әртүрлі масштабта қарап шығу құралы Лупа (Zoom), Shift+Z.

3. Қасиеттер панелі

Ағымдағы құралды баптауға арналған панель

4.

Бейнелермен жасалатын операциялардың барлығы палитралар көмегімен іске асады. Олардың барлық саны 21. Бірге қолданылатын палитралар «бір шатырдың астында» орналасады.



Сурет 2.6 – Палитралар

## 2.2 Қосымшаның кодын өңдеу құралдары

Бұл жобада бағдарламалық қосымшаның кодын жазу үшін Android Studio әзірлеу ортасы пайдаланылады, ал жазатын тіл ретінде Java және XML тілдері қолданылады. Android Studio-мен жұмыс жасау үшін Java SE Development Kit (JDK) қолданылатын жүйеде орнатылуы тиіс. JDK түрі 7-ден кем болмауы керек. Бұл JDK-ді келесі сілтеме бойынша жүктеуге болады:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Java SE Development Kit бағдарламалар әзірлеуіне тегін аспаптар беріледі және тегін таратылады. Осымен қатар жоғарыда көрсетілген сілтеме бойынша өтіп сіз операциялық жүйеңіздің версиясын таңдауыңыз қажет Windows(x86), Windows(x64), Linux немесе Mac OS X. Айтылған файлды жүктеп болғаннан кейін оны нұсқауға сәйкес қосып, осы аспаптар жинағын орнату қажет.

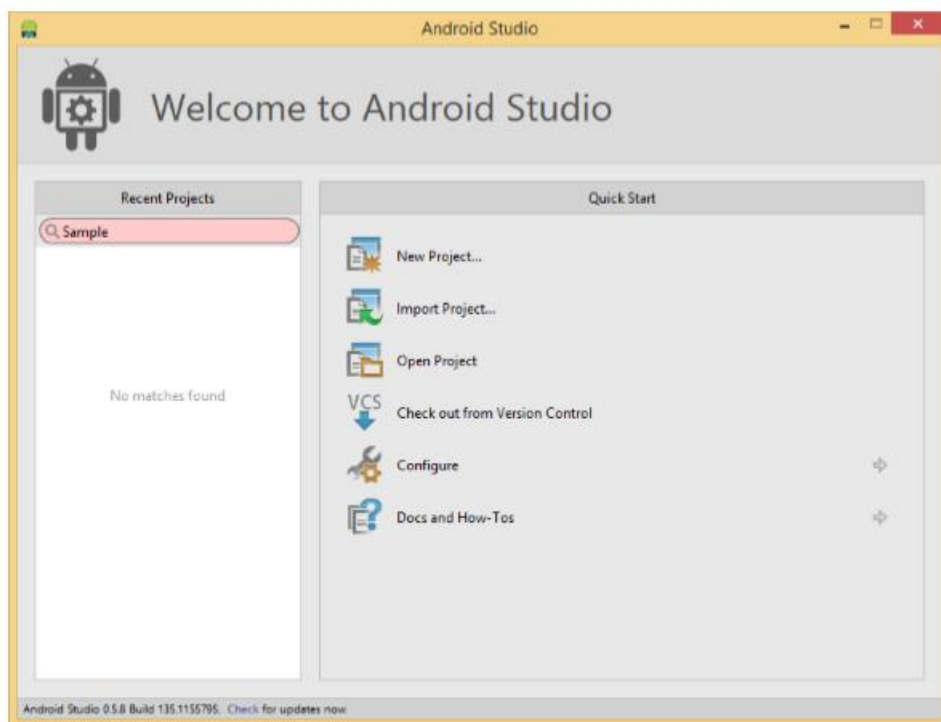
Осыдан соң Android Studio қойылым(установочный) файлы Android-тың әзірлеушілерге арналған ресми сайтынан жүктеу керек. Келесі сілтеме арқылы жүктеуге болады:

<http://developer.android.com/intl/ru/sdk/installing/studio.html>

Бұл жерде де керекті нұсқаны таңдау қажет, сіздің операциялық жүйеңізге сәйкес келетін болуы тиіс. Жүктеу аяқталған соң файлды ашып нұсқауларға сәйкес орнату қажет. Осы жерде мән бөлетін жердің бірі, Android Studio-да жұмыс жасау үшін Android NDK-ны жүктеу қажет емес, себебі қажетті утилиттердің барлығы қосымшаны жеткізу комплектісінде қолжетімді.

Қосымшаны қосқаннан кейін сізге 2.7 суретте көрсетілгендей айна пайда болады. Мұнда сіз:

- Жаңа проект құра аласыз;
- Бар проектті импорттай аласыз (мысалы, егер сіз алдын Eclipse-та жұмыс жасаған болсаңыз);
- Бар проектті ашуыңызға болады;
- Кез-келген жақында ашылған проектін ашуыңызға болады және ол проектiлер арасында бiр проектiнi iздеуге болады;
- Бағдарламалық қамтаманың версиясын тексеруге және қажет болса жаңартуға болады;
- Android Studio баптауларын өзгертуге болады;
- Анықтамаларға қолжетімділік ала аласыз;

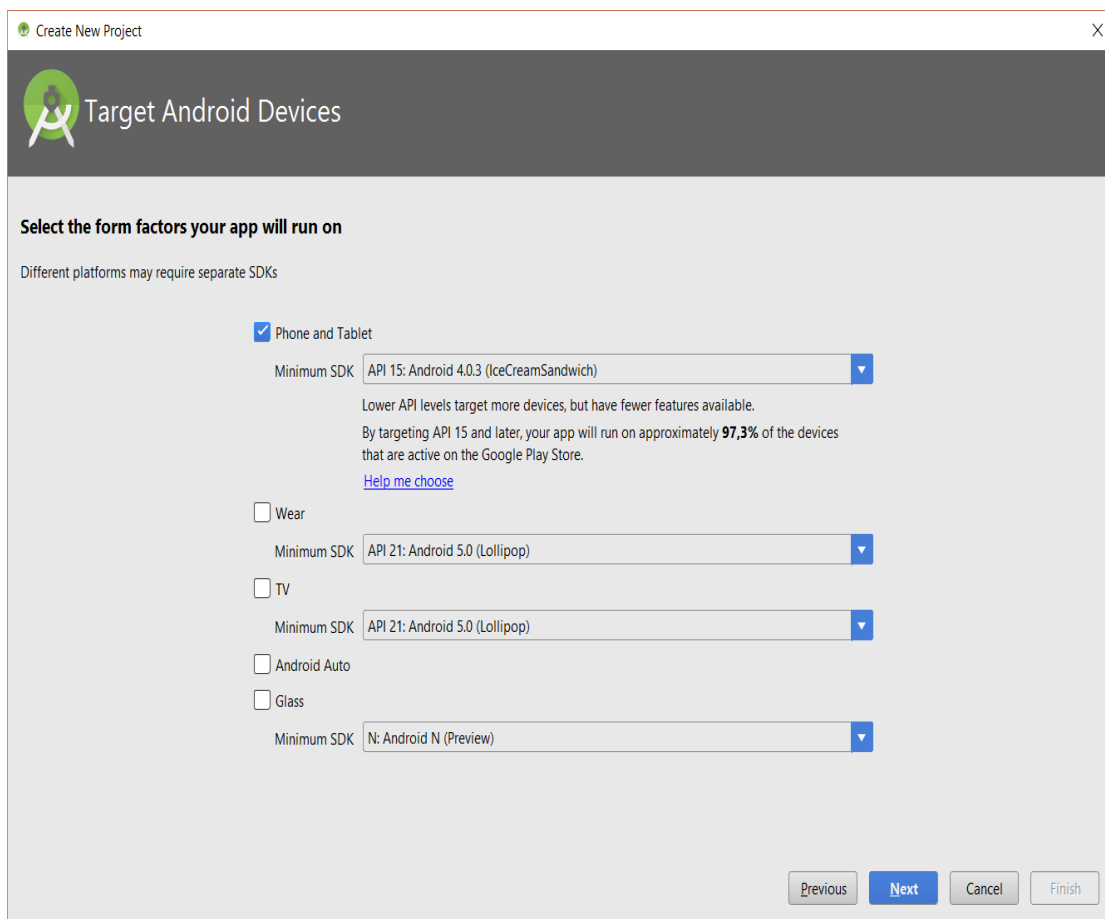


Сурет 2.7 – Android Studio бастапқы айнасы

Жаңа проектті құру үшін «New Project» батырмасын басу қажет, одан соң 2.7 суретте көрсетілгендей проектті құру айнасы пайда болады.

Келесі айнада өз қолданбаңызды қандай құрылғыға жасап шығаратыныңызды және оның ең төмен ОЖ-сін таңдауыңызға болады. 2.8 Суретте көрсетілген.

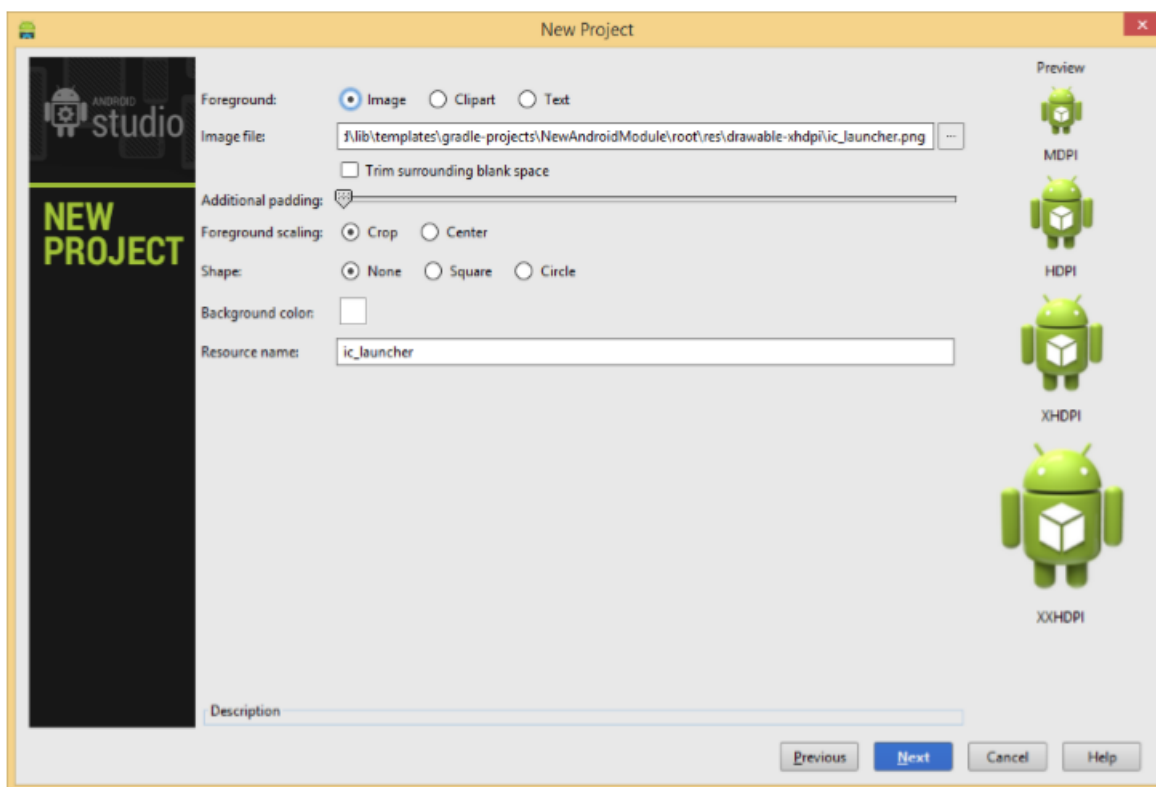
Бұл айнада сіз қосымша атын, модуль атын, қосымшаға файл атын және осы қосымшаның орналасу каталогін беруіңізге болады. Осыған қоса қосымшаның орындалатын құрылғының ең минималды операциялық жүйесін таңдауға да болады. Осыған қоса тіл дәрежесін таңдау нұсқасы бар. Қосымшаның базалық темасын таңдау мүмкіндігі бірден пайда болады. Егер қажет болса «Create custom launcher icon» боксына маркер қою арқылы өз қосымшаңыздың белгішесін Android Studio ішінде жасап шығаруыңызға болады. Және «Support Mode» таңдау мүмкіндігіңіз бар.



Сурет 2.8 – қандай құрылғыға қосымшаны жасайтыныңызды таңдау

Егер «Create custom launcher icon» айнасында таңдалған болса, онда «next» батырмасын басқаннан кейін 2.9 суретте көрсетілгендей айна пайда болады. Мұнда сіз өз қосымшаңызға тез әрі сапалы белгіні жасап шығаруыңызға болды. Бұл функция көптеген бағдарлама әзірлеушілерге ыңғайлы болып саналады.

Бұл айнада сіз қосымша белгі ретінде нақты не қолдану керек екенін таңдай аласыз: дайын сурет, оларды сіз өз компьютеріңізден жүктей аласыз; клипарт немесе текст. Оған қоса бұл жерде белгінің пішінін өзгерту функциясы қолданылған.

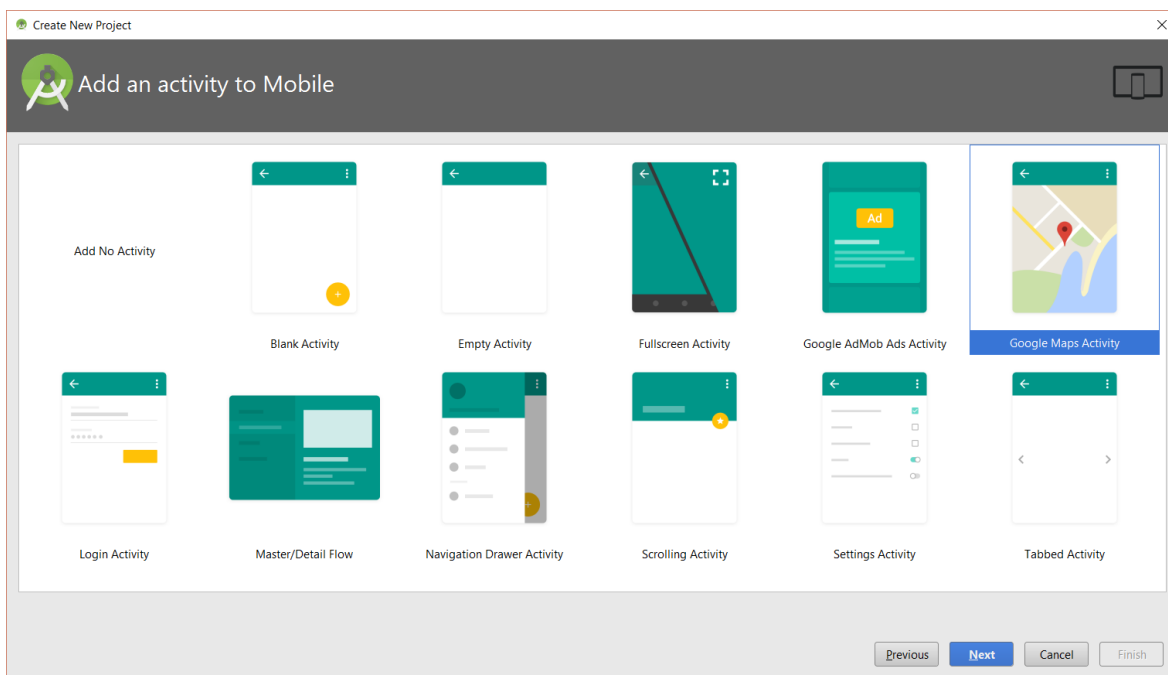


Сурет 2.9 – Қосымшаның белгісін құру айнасы

Белгіні құру айналымынан өткен соң, сіз базалық активтілікті таңдау activity-іне өтесіз, ол 2.10 суретте көрсетілген.

Сізде 12 түрлі activity-ді таңдау мүмкіндігі болады:

- Empty Activity;
- Blank Activity;
- Fullscreen Activity;
- Google AdMob Ads Activity;
- Google Maps Activity;
- Login Activity;
- Master/Detail Flow;
- Navigation Drawer Activity;
- Scrolling Activity;
- Settings Activity;
- Tabbed Activity;



Сурет 2.10 – Базалық Activity таңдау айнасы

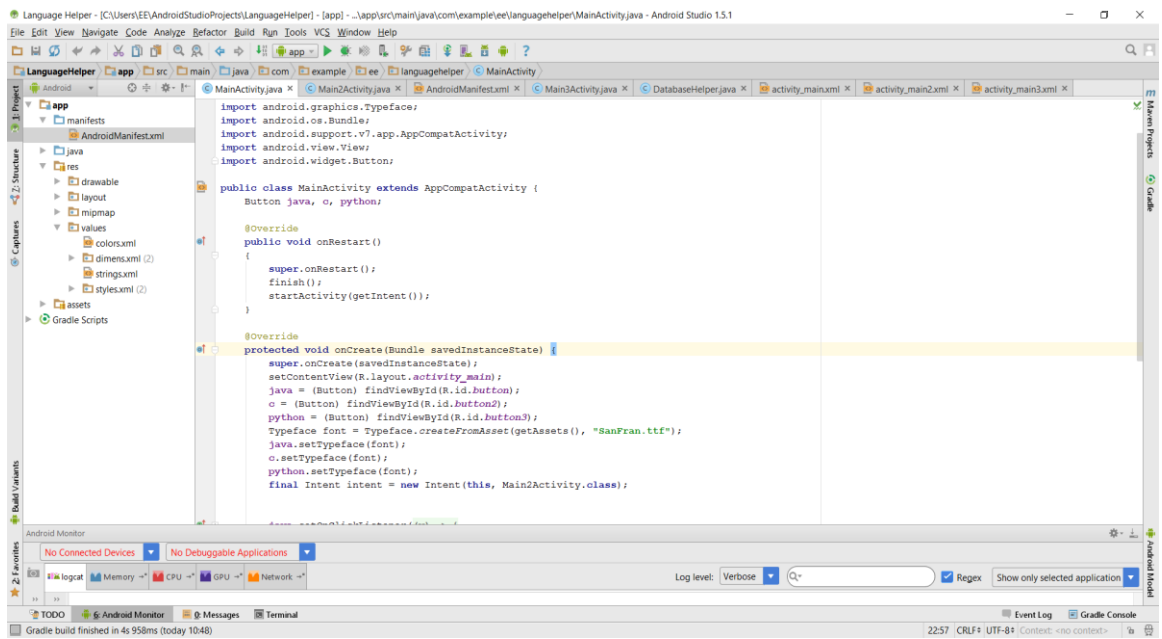
Одан соң біздің бастапқы Activity және оның layout атын береміз, оған қоса бастапқы фрагмент атын, себебі біз проектiнi құру кезінде фрагмент қолдауын таңдадық. Activity бойынша навигация типін таңдауға да болады.

Осыдан кейін бізді IDE айнасына өткізеді. Ең басты мән беретін жағдай ол проект құрылымы. Ол Eclipse-қа мүлдем ұқсамайды.

Үйреншікті src және res папкалары бар, бірақ res папкасы src папка ішінде жатады, java папкасымен бір деңгейде. Мұндай проект құрылымының себебі, Android Studio Gradle жаңа құру жүйе проектiсiн қолданған. Ол бізге, біздің проектiдегі тәуелділік пен басқа да сыртқы кітапханаларды қосу мүмкіндігін ашады.

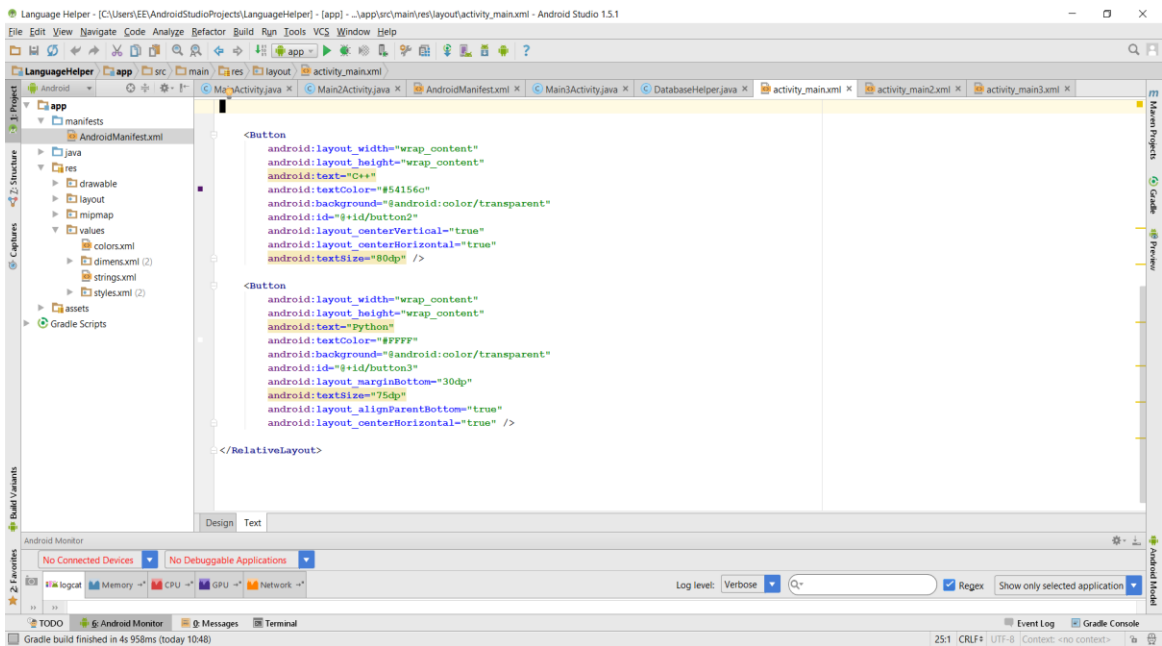
Еске түсіретін нәрсенің бірі қарапайым құрылымды проекттерді импорттаған жағдайда бәрі де жұмыс істеп тұра береді.

2.11 суретте негізгі жұмыс жасау айнасы көрсетілген.

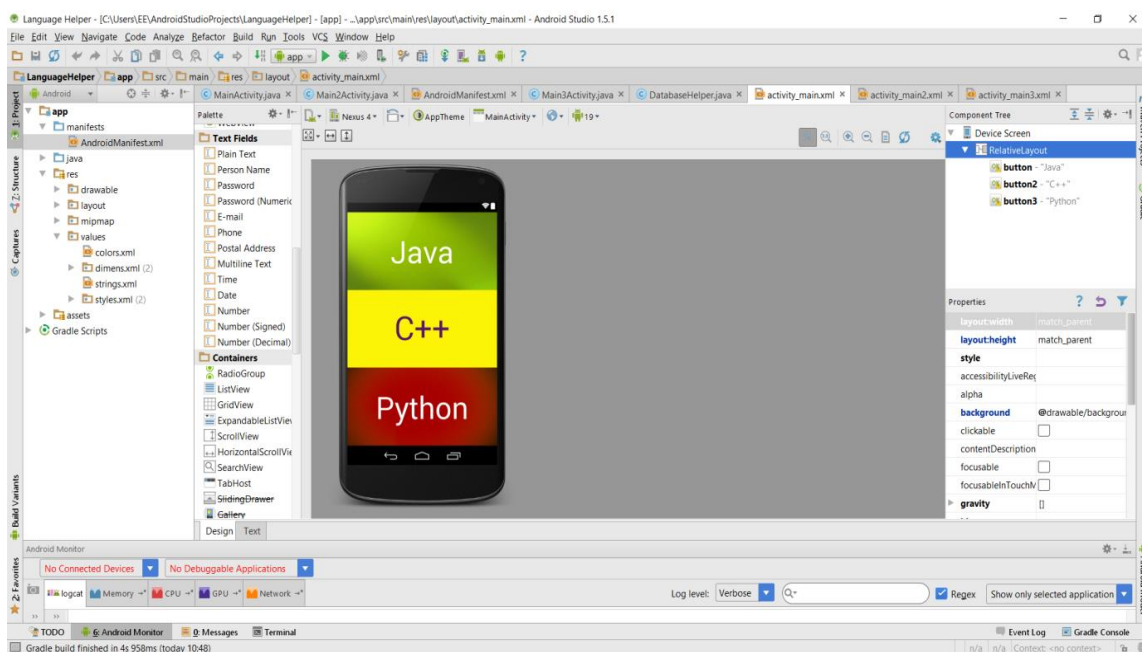


Сурет 2.11– IDE негізгі айнасы

## XML редактор 2.12 – суретте көрсетілген



Сурет 2.12 – XML редактор айнасы №1



Сурет 2.13 – XML кодының визуалды редактор айнасы №2

2.12 мен 2.13 Суреттерді көріп айта кететін жерлер:

- xml кодты текстілік режимда редакторлағанда превью-і де бар;
- layout-та қолданған түстер мен суреттер оның шегінде көрсетіліп, сіздің нақты қандай объектілерді қолданып жатқаныңызды оңай түсінуге көмектеседі;
- Қорды таңдаған жағдайда оның құрамы жүзетін JavaDoc айнасында көрсетіледі. Мысалы: `android:background="@drawable/background`.
- Dimens-тегі қорлар автоматты түрде мәнмен көрсетіледі, ал оған келтіргенде дәл қандай қор қолданатыныңызды көруге болады;

Android компоненттерімен жақсыланған интеграция.

Жаңа классты қосып көрейік. Классты орналастырғымыз келген пакет ішіне орналасып, `alt-insert` батырмаларын басамыз. Айта кететін жағдай Android Studio-да көп ыстық батырмаларды жаттау қиынға түседі, Eclipse-та бұл жағы кішкене оңай болғанымен мұнда қиын және серпімді функционал болады.

Студия бізге құруға бірнеше объектіні таңдауға береді.

Әрқайсысы жайлы қысқаша:

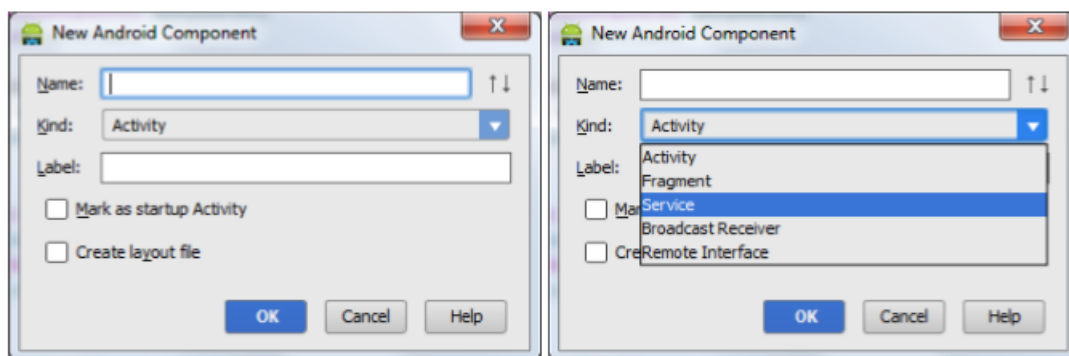
- Java Class - негізінде Java Component. Java-ның келесі басты компоненттерінің бірін құруға мүмкіндік береді: Class, Interface, Enum, Annotation және Singleton;
- Module - модульді құру. Модуль – негізі Android Studio-ның қосалқы проектісі. Проектідегі модульдер барлық сыртқы кітапханалар (мысалы, ActionBarSherlock немесе Facebook SDK);
- File - қарапайым файл барлық кеңеюлерде (txt, json, xml және т.б.);



- Package – қосымшамыздың пакеті.

Android компоненттерін құру

2.14 сурет қосымшаның жаңа компонентін құруды көрсетеді.



Сурет 2.14 – Жаңа компонент құру айнасы

Activity – Activity-ді дайын бір шаблондар көмегімен құрып оны Манифестке тіркейді.

Android Component – универсалды зат, кез-келген негізгі Android компонентін құрай алады, оған дереу керек жағдайда layout құрып және activity-ді бастапқы етіп қояды. Әдепкі қалып бойынша компонентті Манифестке тіркейді, қажет жағдайда.

Package-info.java – пакет жайлы информацияны бейнелеу файлы.

HTML File – html файлды құрады.

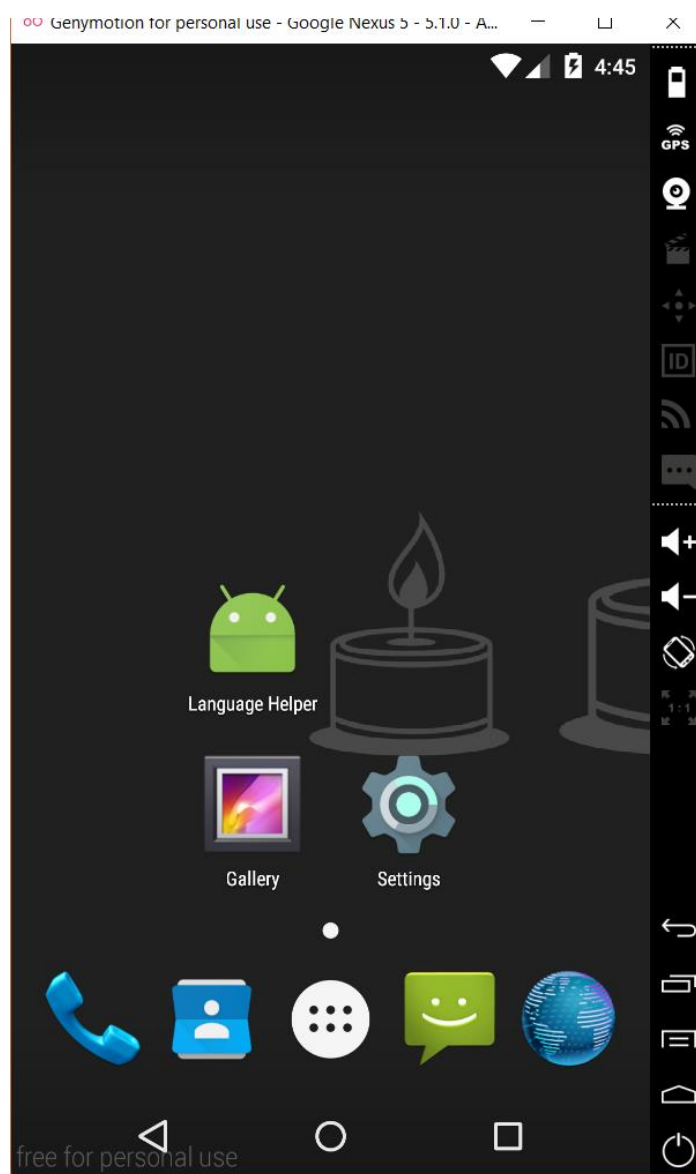
Android Studio құрамында іргелі текст редакторы бар, олар келесі артықшылықтарға ие:

- If, while, switch конструкция немесе методтың басы мен аяғын жалғайды;
- Код анализы интеллектуалдырақ;
- Android Sources-қа қосылу кіріктірілген;
- Класстан мұралану мүмкіндігі, немесе оған тексттерді 2 тінтіп құру;

### 3. Қосымшаны құру сатысы

#### 3.1 Проектілеу

Language Helper – қолданбасы Android платформасына арналған қосымша, анықтамалық класс қосымшасына жатады. Бұл қосымшаның интерфейсі өте қарапайым және қолдануға оңай. Қосымшаның басты мақсаты қолданушыға бұл қолданбаны қолдану оңайлығы және ыңғайлығы. Осы қосымша арқылы қолданушы бағдарламалау тілдерін меңгеру үрдісін әлде қайда оңай қылуға болады. 3.1 суретте қолданбаның сыртықы келбеті келтірілген.



Сурет 3.1 – Қолданбаның сыртқы келбеті

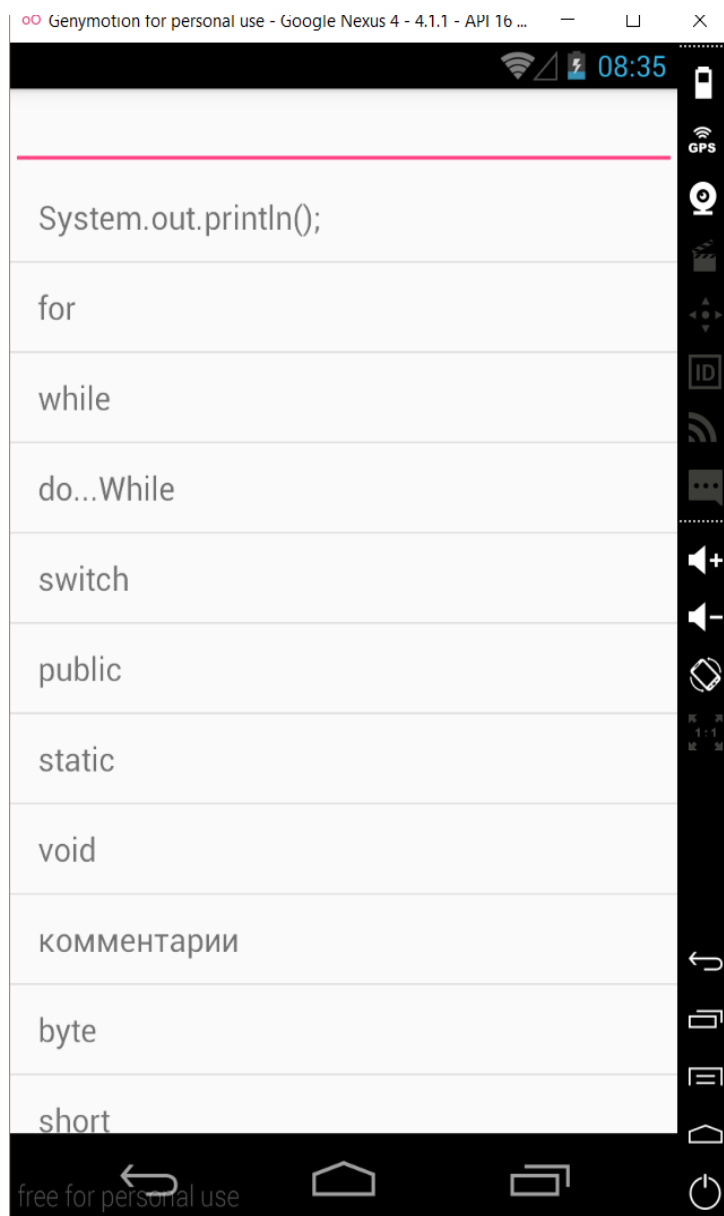
Қолданбаның басты беті қарапайым жасалған. Үш түрлі түс қолданбаның артқы фоны оның бетінде үш батырма орналасқан JAVA, C++,

PYTHON. Әр батырманы басқан кезде сол бағдарламалау тіліне байланысты деректер қорынан сол тілдің деректерін алып шығады. 3.2 суретте қолданбаның басты беті көрсетілген.



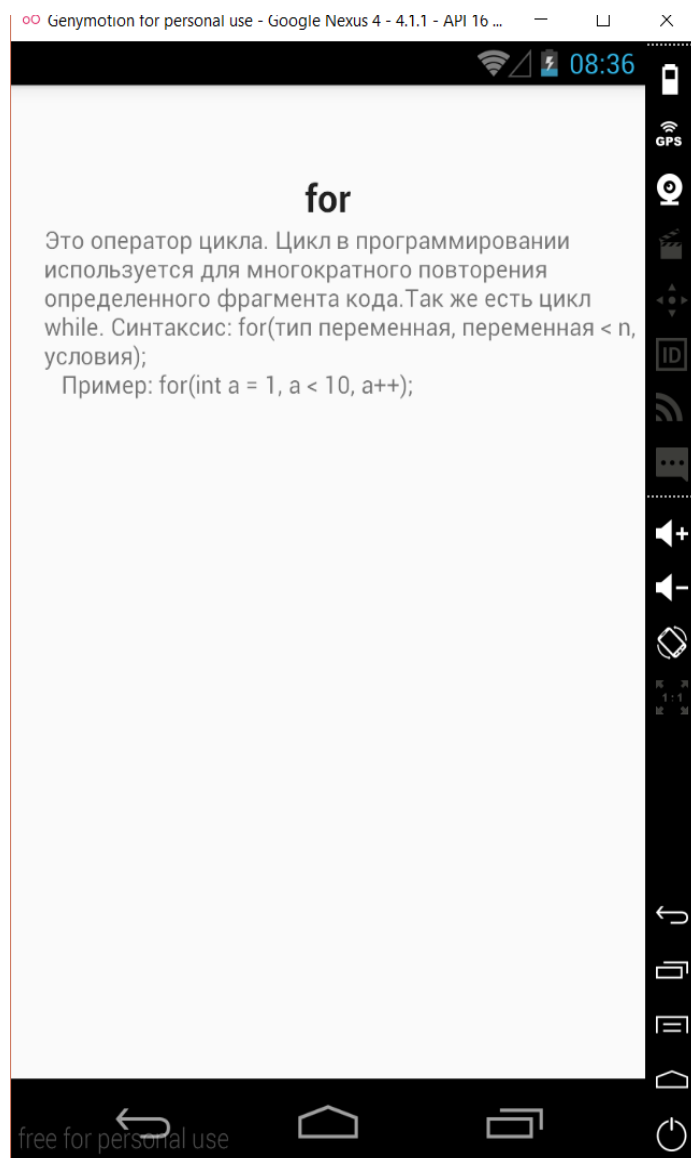
Сурет 3.2 – Қолданбаның басты беті

Қолданбаның екінші бетінде таңдалған бағдарламалау тілдеріне байланысты командалар арасынан өзіңізге керектісін іздеуге болды. Ол бетте EditText пен ListView орналасқан. EditText өзіне текстті қабылдап сол текстке сүйене отырып командалар арасынан іздейді. ListView деректер қорындағы командалар мен олардың түсініктемесін дұрыстап, ұқыпты және түсінікті етіп шығаруына қолданылады. 3.3 суретте қосымшаның екінші беті көрсетілген.



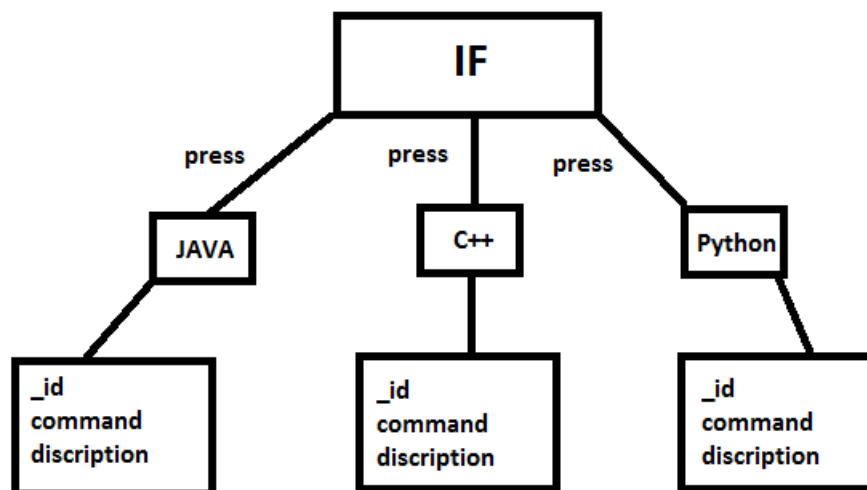
Сурет 3.3 – Қосымшаның екінші беті

Қосымшаның үшінші бетінде таңдалынған команданың толық түсініктемесі мен оның мысалдары және синтаксистері түсіндіріп көрсетіледі. Ол бетте LargeText пен MediumText қолданылған. LargeText команда атын шығаруға, ал MediumText команданы сипаттауға арналған. 3.4 суретте қосымшаның үшінші беті көрсетілген.



Сурет 3.4 – Комманда түсініктемесі және сипаттамасы

Қосымшаның деректер қорысының құрылым сұлбасы 3.5 суретте көрсетілген.



Сурет 3.5 – Деректер қоры сұлбасы

### 3.2 Қосымшаны құру

Проект бірнеше папка мен файлдардан тұрады. Ең бастылары:

- Src – бастапқы коды Java тілінде. Мұнда жұмысқа керек негізгі файл орналасқан. Осында жаңа класстар да орналасады;
- Gen – Java-ның өзі генерациялаған файл;
- Res – қорлар файлы;

Басты экрандағы бағдарламалау тілін таңдау үшін activity\_main.xml файлы жарияланған, res/layout каталогінде. Мұнда үш батырма мен қолданылатын шрифттер, батырма стильдері анықталған.

```

<?xml version="1.0" encoding="utf-8"?>
<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:orientation = "horizontal"
    android:background="@drawable/background"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ee.lhelper_10.MainActivity">

<Button
    android:layout_width="wrap_content"
  
```

```
android:layout_height="wrap_content"
android:text="Java"
android:textColor="#FFFF"
android:background="@android:color/transparent"
android:id="@+id/button"
android:layout_marginTop="38dp"
android:textSize="75dp"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true" />
```

<Button

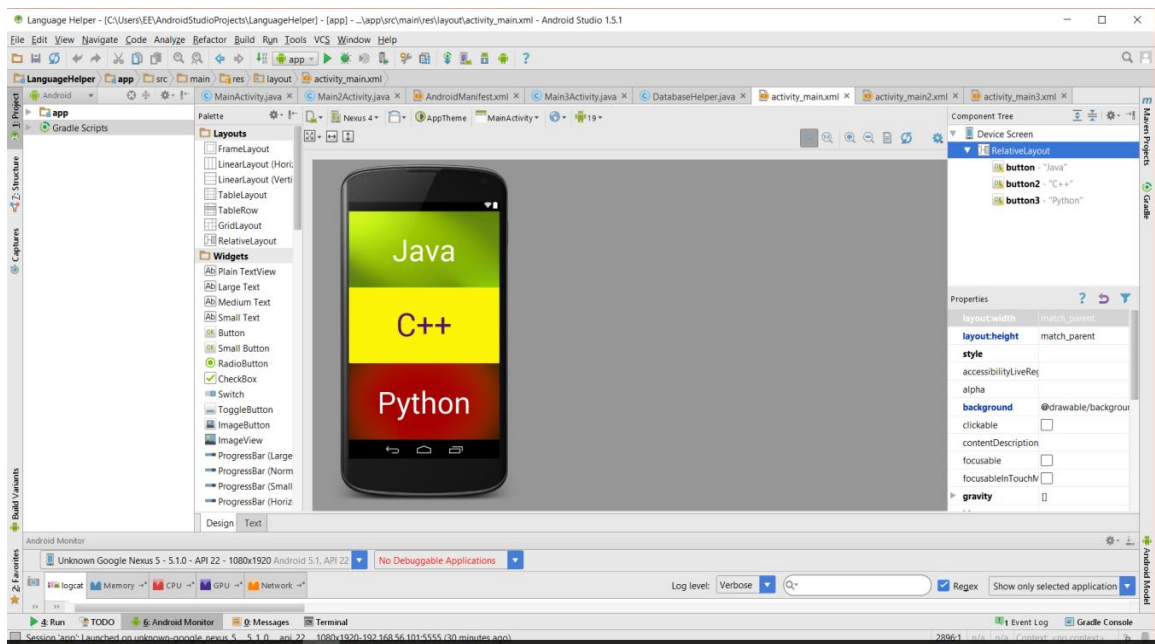
```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="C++"
android:textColor="#54156c"
android:background="@android:color/transparent"
android:id="@+id/button2"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:textSize="80dp" />
```

<Button

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Python"
android:textColor="#FFFF"
android:background="@android:color/transparent"
android:id="@+id/button3"
android:layout_marginBottom="30dp"
android:textSize="75dp"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true" />
```

</RelativeLayout>

Бағдарламалық тілді таңдау бетінің Android Studio бағдарламасындағы суреті 3.6 суретте көрсетілген.



Сурет 3.6 – Android Studio-дағы бірінші бет дизайны

Қосымшадағы деректер қорынан деректерді алып шығару екінші бетте реализацияланған, ол үшін activity\_main2.xml файлы арналған.

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

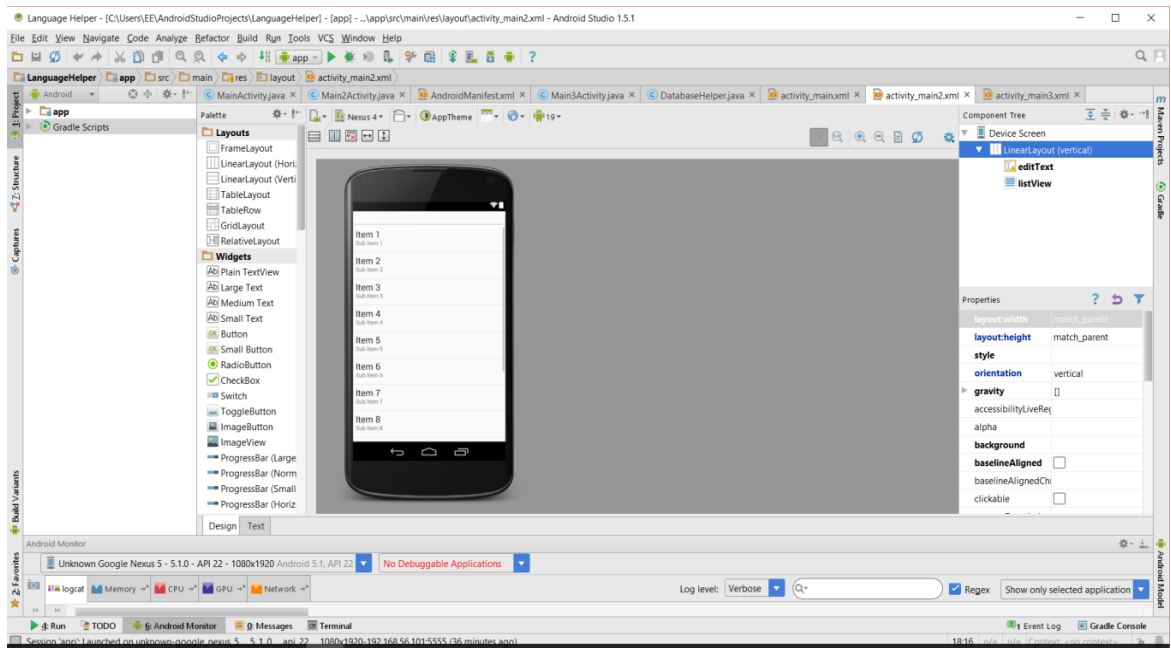
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:layout_gravity="center_horizontal" />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_gravity="center_horizontal" />

</LinearLayout>
```

Бұл беттің Android Studio бағдарламасындағы дизайны 3.7 суретте көрсетілген.





Сурет 3.7 – Қосымшаның деректер қоры орындалатын, екінші беті

## 4 Тіршілік қауіпсіздігі

### 4.1 Программалау офисінің жұмыс жағдайының талдауы

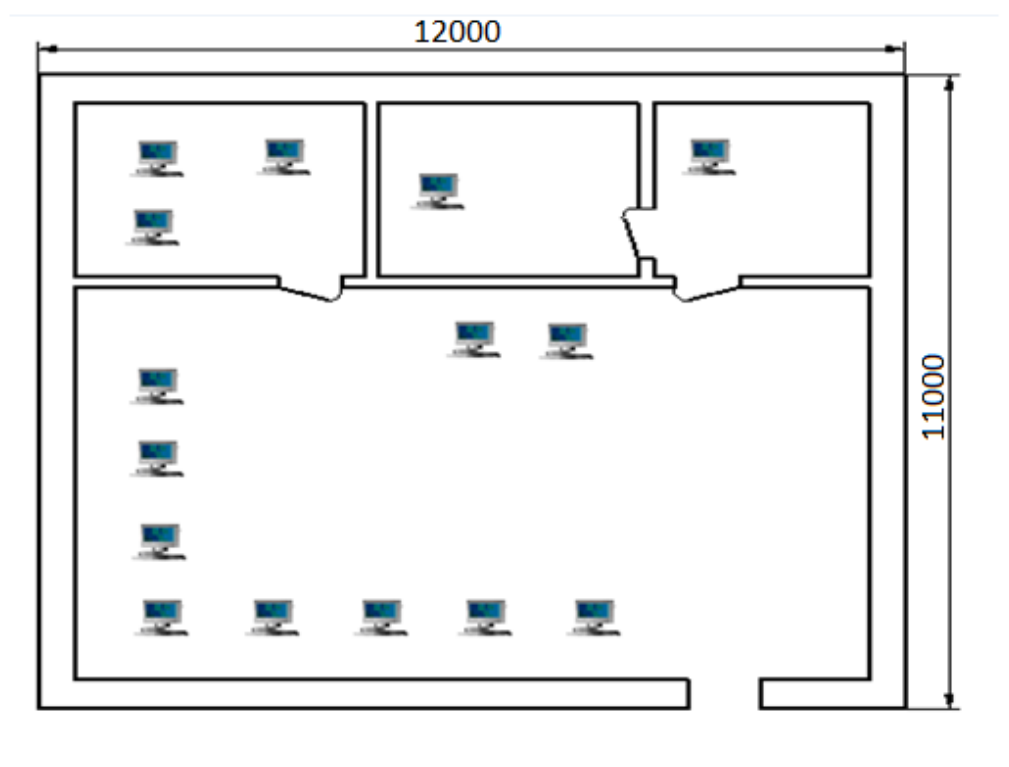
Бұл дипломдық жобаның тақырыбы –«Android платформасында бағдарламалау тілдеріне электрондық анықтаманы құрастыру». Компьютердің көмегімен ұялы телефондарға арналған қолданбалар құру. Осының бәрі дербес компьютерде орнатылған арнайы бағдарламалардың көмегімен жасалынады, сондықтан қызметкерлер көп уақытын компьютердің алдында өткізеді. Сол себепті жұмыс орнын ұйымдастыруды және еңбек қауіпсіздігін қамтамасыз ету көзқарасымен компьютермен жұмыс істеудің ережелерін қарастыру қажет. Желі құрылымы «Компьютерлер көмегімен қолданбалар жасау» параметрлерге сәйкес келеді:

- информациялық қауіпсіздігі;
- жұмысқа реттеу жеңілдігі;
- жобаны ауқымды ету жеңілдігі;
- шығынның аз болуы;

Мобильді қолданба құру кезінде келесі бағдарламалық қамтамасыздандырулар қолданылады:

SQLite, Xml, Java тілдерін қолданады.

Ғимараттың схемасы төмендегі 4.1-суретте көрсетілген.



4.1 сурет – Жобаланатын бөлме

Компьютермен жұмыс істеу кезінде оператордың ақыл ойына және жүйке жүйесіне, көру қабілетіне, пернетақтамен істеген кезде қол

бұлшықеттерінің шаршауына әсер етеді. Сондықтан операторлардың жұмыс орындары дұрыс жерде орналасуы, стандартқа сәйкес болуы өте маңызды.

Компьютермен жұмыс істеген кезде дұрыс жұмыс және демалыс кестесін сақтау керек.

Құрылғы аз шу шығарады. Жоғары электромагниттік сәулелену жоқ, өйткені LCD мониторлар қолданылады.

Дұрыс жобаланған және орындалған өндірістік жарықтандыру көру жұмысын жақсартады, әлсіздікті төмендетеді, жұмыс өндірісін арттырады, жұмыс қауіпсіздігін жоғарылатады.

Жарықтың жеткіліксіз болуы адам көзінің көру қабілетін төмендетіп, оның бұзылуына алып келуі мүмкін. Шамадан көп жарықтану көздің ашып ауруына алып келуі мүмкін. Дұрыс емес бағытталған жарық жұмыс орнында колеңке пайда болады. Осы барлық себептер аурулардың пайда болуына әкеледі. Сондықтан жарықтандыруды дұрыс есептеу өте маңызды.

Бөлмеде екі терезе орналасқан және табиғи жарықтану нормаға сәйкес келеді. Табиғи жарықтандыру қажетті жарықты толық жұмыс уақытында қамтамасыз ете алмайды, әсіресе қыстық жұмыс күндерінде. Сондықтан жұмыс бөлмесінде жасанды жарықтандыру ескерілген.

Жұмыс бөлмесінде орнатылған компьютерлер жылудың боліну көзі болып табылмайды. Қондырғылардың өте аз мөлшерде жылу бөлуі жұмыс орынның микроклиматына әсер етпейді.

Бөлмедегі жұмыс шартын қарастырайық. Бөлменің параметрлері келесідей: ұзындығы  $L = 12$  м, ені  $B = 11$  м, биіктігі  $H = 3$  м. Бөлменің ауданы  $S = L * B = 132$  м<sup>2</sup>. жұмыс бөлмесінде 15 адам жұмыс істейді. Нормаға сай әр адамға кем дегенде 6 м<sup>2</sup> бөліну керек. Бұл жерде әр адамға 8,8 м<sup>2</sup> сәйкес келеді. Яғни талап етілген нормаға сай келеді.

Бөлмеде қауіпті және зиянды факторлардың деңгейі қойылған нормалардан аспайды. Бөлме жарық, құрғақ және таза, яғни гигиеналық нормаға сәйкес келеді.

Қарастырылып жатқан бөлмеде адамдарды оттың түтінінен қорғау үшін вентиляциялық каналдан тұратын түтінге қарсы қорғаныс қолданылады. Түтінге қарсы қорғаныс автоматты түрде немесе батырма көмегімен қосылады.

Жұмыс орындарын ұйымдастыру кезінде ерекше талаптар қойылады:

- жұмыс орындарының бір-біріне бүйір жақ аралығы 1,2 метрден кем болмау керек. Тізбектей орналасу кезінде 2 метрден кем емес, бетпе-бет орналасса – 0,6 метр, ал оператордың ең аз жұмыс орны 6 м<sup>2</sup> сәйкес болады;

- компьютерді экранымен терезеге қаратып орналастыруға болмайды, өйткені монитор бетінде көлеңкелер пайда болады;

- егер операторлар арқа мен арқа болып отырса, олар бір-біріне кедергі жасайды.

Оператордың жұмыс орны орындалатын жұмыстың сипатына қарай максимал түрде бейімделген. Электроқауіпсіздік үшін жерлеу қолданылады.

Алынған мәліметтердің негізіне сүйене отырып орындалатын жұмыс түрін анықтауға болады.

Эвакуация кезіндегі жұмыс түрі схемасы төмендегі 4.1-кесте көрсетілген.

4.1 кесте – Жұмыс түрі (ГОСТ 12.2.032-78)

Жұмыс түрі	Жұмыс сыныбы	Жұмысшының жынысы	Жұмыс жазықтығының биіктігі	Орындық биіктігі
Жеңіл	Бір а	Ер, әйел	725 мм	420 мм

Жұмыс орнына баға берейік:

- орындалатын жұмыс жеңіл физикалық категорияға жатады. 138 Ваттан төмен, отырып істелінетін жұмыс және физикалық күш қажет емес;
- жұмыс істеу жазықтығының биіктігі 725 мм, орындық биіктігі 420 мм (ГОСТ 12.2.032-78);
- жұмысшылар саны – 15 адам.

#### 4.1.1 Өрт қауіпсіздігі

Өрттің пайда болу себептері:

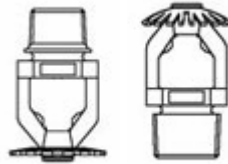
- аппараттың элементтерінің жануы;
- жөнделмеген ажыратқыштардан, розеткалардан, бөлімдегі материалдармен тікелей қатынаста болатын желдету жүйелерінен бөлім материалдарының жануы;
- жабдықты эксплуатациялау режимдерін сақтамау, қызметшінің дұрыс емес әрекеті.

Өрт болған жағдайда тек бөлме ғана зиян шекпейді, сонымен бірге қымбат тұратын аппараттар, адам өлімі болуы болуы мүмкін. Сондықтан өртті ертерек тауып және сөндіру бойынша шара қолдану қажет. Бөлмені қорғау үшін дренчерлік қондырғы негізіндегі автоматтық өрт сөндіргішті таңдаймыз.

Өрт сөндіретін дренчерлік жүйе.

Дренчерлік жүйе сумен толтырылмаған және жапқыштары жоқ және су шығаратын тесіктері ашық дренчерлік бастармен жабдықталған су өткізетін құбырлар жүйесінен тұрады. Дренчерлік жүйеге өрт көздерін сөндіру үшін су айналадағы ауаның температурасы жоғарылағанда ашылатын клапандар арқылы автоматты түрде жіберіледі.

Дренчерлік бастар схемасы төмендегі 4.2-суретте көрсетілген.



4.2 сурет - Дренчерлік бастар

Дренчерлік қондырғылар бүкіл ғимаратқа су шаша алатын кран. Бұл кран өрт болатынын хабарлайтын құрылғы көмегімен іске қосылады.

Өрт хабарлайтын құрылғы әр түрлі – түтіндік, жылулық, жарықтық болып бөлінеді.

Дренчерлік суландырғыш ретінде ДВН10 суландырғышты 4.3-сурет таңдаймыз.



4.3 сурет - ДВН10 дренчерлік суландырғыш

Автоматты өрт сөндіру қондырғыларында суды шашу және суды қорғалатын аудан аралығында өрт ошақтарын сөндіру немесе оларды жою мақсатымен тарату, сонымен бірге сулы бүркеулер құру үшін қолданылады. Кез келген бөлмелерде қолдануға болады, кеңсе, кітапхана, театр, өндірістік және қойма бөлмелерінде де қолданылады.

Техникалық сипаттамалары:

- біріктіру оймасының диаметрі  $R\frac{1}{2}$ ;
- шығыстық тесіктің шартты диаметрі 10 мм;
- шашу алдындағы минимал жұмыстық қысым 0,05 МПа;
- қорғалатын аудан  $12 \text{ м}^2$ ;
- суландырудың орташа өнімділігі  $0,04 \text{ л/сТм}^2$ ;
- шығын коэффициенті 0,9;

Суландыру біркелкілігінің коэффициенті 0,5 кем емес;

Қондырғыны есептеу үшін алғашқы деректер төмендегі параметрлер болып табылады:

Бөлме тобы - 1.

Суландыру өнімділігі -  $0,04 \text{ л/с*Тм}^2$ .

Қорғалатын аудан -  $12 \text{ м}^2$ .

Су шығынын есептеу үшін қажетті аудан -  $110 \text{ м}^2$ .

Сумен өрт сөндіру қондырғысының жұмыс істеу ұзақтылығы - 30 мин.

Дренчерлік суландырғыштардың аралығы - 4 м.

Суқұбырларындағы судың қозғалыс жылдамдығы - 10 м/с.

Басқару түйініндегі қысым - 1,0 МПа.

Суландырғыш арқылы су шығынын  $Q_d$ , л/с төмендегі формуламен есептейміз

$$Q_d = k \cdot \sqrt{H} \quad (4.1)$$

мұндағы  $k$  - суландырғыштың өнімділік коэффициенті,  $k = 0,31$ ;

$H$  - суландырғыштың (генератордың) алдындағы бос күш (генератор).

Минималды бос күш - 5 м.

Максималды рұқсат етілген күш - 100 м.

Сулы дренчерлік шығыс тесігінің диаметрі - 10 мм (4.10) формула бойынша суландырғыш арқылы су шығынын есептейміз:

$$Q_{d\min} = 0.31 \cdot \sqrt{5} = 0.7 \text{ л/с}$$

$$Q_{d\max} = 0.31 \cdot \sqrt{100} = 3.1 \text{ л/с}$$

Құбырлар – болаттан жасалған сулыгаздысымды (ГОСТ 3262x75)

Шартты өтудің диаметрі - 25 мм.

Сыртқы диаметрі - 32 мм.

Қабырғаларының қалыңдығы – 2,2 мм ,  $k_1 = 3,44$ .

Құбырлардың есептік бөліктеріндегі күш шығыны  $H_1$  төмендегі формуламен анықталады

$$H_1 = \frac{Q^2}{B} \quad (4.2)$$

мұндағы  $B$  құбырдың сипаттамасы, төмендегі формуламен анықталады

$$B = \frac{k_1}{l} \quad (4.3)$$

мұндағы  $k_1$  коэффициент, біз 3,44-ке тең деп аламыз;

$l$  - құбырдың есептік бөлігінің ұзындығы, м.

4.2 және 4.3 формулаларға тиісті мәндерді қойып есептейміз:

$$B = \frac{3.44}{6} = 0.57$$

$$H_1 = \frac{3.1^2}{0.57} = 16.85 \text{ м}$$

Қондырғының басқару түйіндеріндегі күш шығыны  $H_2$ , төмендегі формуламен анықталады

$$H_2 = \varepsilon \cdot Q^2 \quad (4.4)$$

Мұндағы  $\varepsilon$  басқару түйініндегі күш шығынының коэффициенті.

Басқару түйіндері – дренчерлік қондырғылары.

Клапан түрі – КТПА.

Клапанның диаметрі - 100 мм.

Күш шығынының коэффициенті  $\varepsilon = 2,27 \cdot 10^{-3}$  4.4 формулаға тиісті мәндерді қойып  $H_2$ -ні есептейміз:

$$H_2 = 2,27 \cdot 10^{-3} \cdot 3.1^2 = 21.8 \cdot 10^{-3}$$

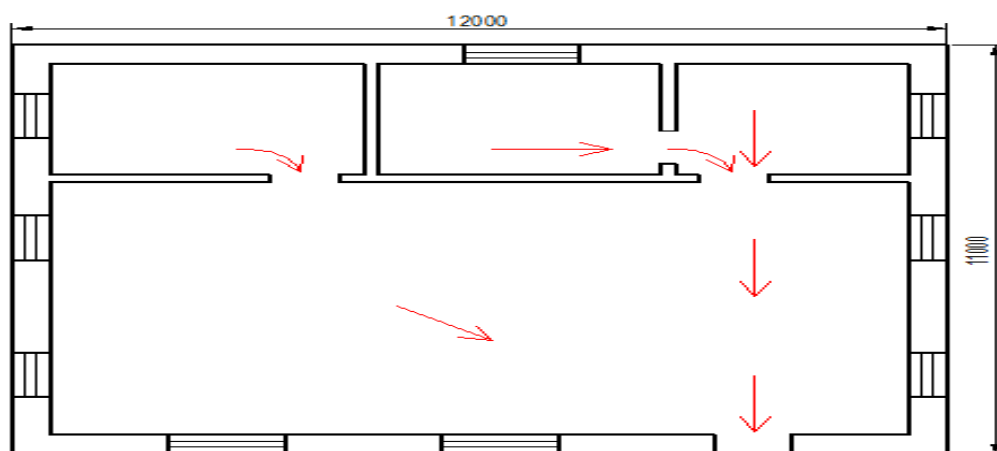
Бөлменің ауданы  $110 \text{ м}^2$  болғандықтан 10 дренчерлік суландырғыш қажет.

#### 4.1.2 Эвакуацияның уақытының есептеуі

Қорғаудың негізгі әдістерінің қатты әсер ететін факторларынан бірі қауіпті аудандар және зілзала аймақтарының объекттердің қызметшілерін және тұрғындарын дер кезінде эвакуациялау болып табылады.

Кәсіпорынның қызметкерлерін кабинет өртінің туу жағдайында ғимаратқа эвакуациялық уақыттын анықтау керек. «Компьютерлік техниканы сатуға арналған интернет дүкенін» құру ғимараты 1-этажды және жобада  $12 \times 11 \text{ м}$  болып келеді. Бөлмелердің аудандарының қосындысы 110 шаршы метрді алып жатыр. Бұл мекемеде 15 адам саны жұмыс жасайды. Эвакуациялық шығуларға арналған есіктер ені-1метр және ғимараттан шығатын есіктерді саны – 1.

Эвакуация кезіндегі ғимараттың схемасы төмендегі 4.2-суретте көрсетілген.



4.2 сурет – Бағдарламалау офис қызметкерлерінің эвакуациялау схемасы

### 4.1.3 Есептелуі

Эвакуациялық жолмен қозғалған адамның ағындары жол-жөнекейімен екі есік ойықтары арқылы өтеді. Олардың тиісті ені:

$$\beta = N/165$$

(4.5)

мұндағы  $N$  - адамдардың жиынтық саны, адам ;

165 - I және II отқа төзімділік дәрежелері үшін ғимараттардың ені 1 м есіктің алдынан адамдарының жиналып қалмай, жіберілетін нормативтік адамдардың саны.

$$N=15/1=15 \text{ адам.}$$

Бір эвакуациялық шығу үшін :

$$\beta = 15/165=0,09 \text{ м.}$$

Эвакуациялық есіктердің нақты өлшемдері 1 м.

Қорытынды: көрермен залынан эвакуациялық шығуларын ең - жобалаудың нормаларының талаптарына жауап береді.

Эвакуацияның уақытының есептеуі.

Ғимараттан шығуға дейін адамдарды эвакуацияның ұзақтығы сыртқы есіктер және сатылар арқылы эвакуацияның созылымдық және өткізгіштік қабілет бойымен анықтайды.

$T$  - жол дәліз бойымен адам ағын қозғалыс уақыты мына формула бойынша есептейді:

$$t_1 = \frac{L_1}{V_1} = 5/8.7 = 0,57$$

$L$  жолдың дәліз ұзындығы.

$V$  - көлбеу жол бойымен адам ағынды қозғалыс жылдамдығының мәні, салыстырмалы тығыздық  $D$  м<sup>2</sup>/м<sup>2</sup> тәуелділікте анықталады.

$D$  - адам ағынының тығыздығы, формула бойынша есептейді:

$$D_1 = \frac{N_1 \cdot f}{L_1 \cdot b_1} = 15 \cdot 0.1 / 5 \cdot 1 = 0.3 \text{ м}^2/\text{м}^2.$$

$N_1$  - адам саны;

$f$  - адамның көлбеу проекциясының орташа ауданы;

$L_1$  и  $b_1$  - ұзындық және жолдың бірінші бөлімшенің ені;

Ойықта қозғалыс уақыты формула бойымен анықталады:

$$t_2 = \frac{N \cdot f}{q \cdot b} = \frac{15 \cdot 0,1}{8,7 \cdot 1} = 0,17 \text{ мин}$$

Жалпы ғимараттан эвакуациялау уақыты:

$$T = t_1 + t_2 = 0,17 + 0,57 = 1,14 \text{ мин.}$$

Өрт жарылыс қауіпі бойынша бөлмелердің классификациясы 4.1-



кестеде көрсетілен.

#### 4.1 кесте - Өрт жарылыс қауіпі бойынша бөлмелердің классификациясы

Бөлменің категориясы	Заттардың сипаттамасы
А(өрт-жарылыс қауіпті)	Жарылыс қауіпті газдар, $<28^{\circ}$ С тотығу температурасы бар жанғыш сұйықтықтар
Б (өрт-жарылыс қауіпті)	Жарылыс қауіпті шаңдар және $T \sim 28-61^{\circ}$ С жанғыш сұйықтықтар
В (өрт қауіпті)	Қатты, жанғыш, сұйықтықтар (ағаш өңдейтін цехтар)
Г	Қыздырылған немесе балқытылған түрдегі жанбайтын заттар
Д	Суық түрдегі жанбайтын заттар

Уақыт эвакуацияның кідіріс уақыты 4.2 - кестеде көрсетілен.

#### 4.2 кесте . Уақыт эвакуацияның кідіріс уақыты

Ғимарат түрі	Эвакуациялық кідіріс			
	W1	W2	W3	W4
Өндірістік, сауда саттық ғимараттар	<1	3	>4	<4
Музейлер, демалыс орындары	<2	3	>6	<6
Жатақхана, интернаттар	<2	4	>5	<5
Қонақ үйлер	<2	4	>6	<5
Госпитальдар, ауруханалар	<3	5	>8	<8
Ескерту: Хабарлау жүйесі сипаттамасы W1 - оператордың эвакуациясымен хабарлау және басқару; W2 - жазып алған алдын ала бір үлгідегі сөйлемдер және ақпараттық көрсеткіш тақталардың қолдануы;				

Адамның проекциясының ауданы 4.3 - кестеде көрсетілен.

#### 4.3 кесте - Адамның проекциясының ауданы

Қозғалатын адамды сипаттамасы	Мәні, м <sup>2</sup> адам
Үй киімінде ересек адам	0,1
Қысқы киімде ересек адам	0,125
Қолдарында ересегі	0,26
Ересегі қолдорбамен	0,16
Ересегі чемоданмен	0,35
Жас өспірім	0,07

Адам ағынды тығыздықтан жылдамдық және жүріс қарқындылығының тәуелділігі 4.4 кестеде көрсетілен.

4.4 кесте - Адам ағынды тығыздықтан жылдамдық және жүріс қарқындылығының тәуелділігі

Ағын тығыздығы $D, \text{ м}^2/\text{м}^2$	Горизонталь жол		Есік қуысы
	$V, \text{ м/мин}$	$q > \text{ м/мин}$	$q, \text{ м/мин}$
0,01	100	1,0	1,0
0,05	100	5,0	5,0
0,1	80	8,0	8,7
0,2	60	12,0	13,4
0,3	47	14,1	15,6
0,4	40	16,0	18,4
0,5	33	16,5	19,6
0,6	27	16,2	19,0
0,7	23	16,1	18,5
0,8	19	15,2	17,3
0,9 және көп	15	13,5	8,5

## 5 Техико-экономикалық негізделуі

### 5.1 Жұмыстың мақсаты

Бұл қарастырылып отырған дипломдық жұмыстың мақсаты Android операциялық жүйесіне арналған, бағдарламалау тілдеріне қатысты электронды анықтама жасап шығару. Бұл қолданба арқылы 3 тіл бойынша анықтама алуға болады Java, Python, C++. Тіл операторларымен сипаттамаларын сипаттап береді. Бағдарлама Android Studio бағдарламалау ортасында Java, Xml, SQLite тілінде жазылған Android OS операциялық жүйесіне арналған болып табылады.

### 5.2 Еңбектің жұмыста пайдаланатын қамбаларының

Дипломдық жұмыс орындалу барысында:

- дизайншы – интерфейстің сұрын жасап шығарушы;
- программист – бағдрламаның алгоритмін және жалпы қолданбаны құрушы.

5.1 Кестеде қызметкерлердің саны мен олардың айлық саны көрсетілген

Кесте 5.1 – қызметкерлер деректері

Міндеттілік	Қызметкер саны	Қызметкерлер айлығы
Программист	1	120000
Дизайыншы	1	100000
Барлығы	2	220000

### 5.3 Жұмыста қолданылатын жабдықтар

Жұмыс барысында қолдатынлатын құрал-жабдықтар, компьютер, телефон және тағыда басқа төмендегі 5.2-кестеде көрсетілген.

К е с т е 5.2 – жабдықтар.

Жабдықтың аты	Сипаттамалар	Саны	Құны, тг
Ноутбук Asus Zenbook UX305	Intel Core I7 6700K, 8 GB, DDR3, 1000 GB HDD, Intel HD Graphics 5300	1	250000
Ноутбук HP Envy 15- j011sg	Intel Core I7 – 4702MQ, 6 GB, DDR3, 1000GB HDD,	1	180000

	GeForce GT 740M		
Смартфон LG Bello D335	ROM 4Gb, 1 Gb RAM, MediaTek MT6582, 4 ядра	1	50000
Барлығы		3	480000

#### 5.4 Пайданылатын бағдарламалар

Программаны дайындау барысында келесі бағдарламалармен жұмыс жасадық:

- Windows 10 - операциялық жүйесі;
- Adobe Photoshop CS6 – фоторедактор бағдарламасы;
- Android Studio – Android-қа бағдарлама құру платформасы;
- Java Development Kit - қосымшаларды әзірлеушілер жиынтығы.

К е с т е 5.3 – пайдаланылатын бағдарламалық қамтамасыз ету

Аты	Саны	Құны
Windows 10	2	Ноутбук бағасына кіріктірілген
Adobe Photoshop CS6	1	40000
Android Studio	1	Тегін
Java Development Kit	1	тегін
Барлығы		40000

Жобалау процесін ұзақтығы және бағдарламаны әзірлеу бес кезеңнен тұрады:

- Мәселені қою және деректер жинау;
- Қолдану интерфейсін жобалау, дамыту;
- Өтінімдерді әзірлеу;
- Тестілеу операция;
- Тіркеу және баяндама ұсыну.

К е с т е 5.4 – жоба

Жұмыс тізімі	Апталық жұмыс уақыты							
	1	2	3	4	5	6	7	8
1	Мәлімдеме құру							
	Жетілдіру ортасын тандау							
	Әдебиет зерттеу							
	«директивалар зерттеу»							
2	Интерфейсті құру							
	Қажетті графикалық элементтерді жасау							
3	Өтінімдерді әзірлеу							
4	Бағдарламаны тестілеу							
	Қателерді жөндеу							
5	Баяндаманы тіркеу және ұсыну							

Шығындар мен жоба мобильді қосымшалар әзірлеу бойынша жұмыс құны 4.6 есептеу жұмыстарын орындау қызметкерлерінің интеллектуалдық құны, сондай-ақ оны іске асыру үшін қажетті техникалық құралдар көп талап етеді. Барлық осы жобаның қорытынды құнын негізінде есептеледі қаржы инвестициялар, талап етеді. Осы қолдану даму шығындар формула бойынша есептеледі.

$$C = \text{ФОТ} + C_{\text{Н}} + A + \text{Э} + H \quad (5.1)$$

мұндағы ФОТ – жалақы қоры;  
 $C_{\text{Н}}$  – әлеуметтік салық;  
 $A$  – амортизациялық аударымдар;  
 $\text{Э}$  – энергия шығындары;  
 $C_{\text{пр}}$  – басқа шығындар;  
 $H$  – үстеме құны.

## 5.5 Жалақы төлемі

ЕТҚ қызметкерлердің негізгі және қосымша жалақыдан тұрады, есептеу мына формула бойынша қарастырылады:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}} \quad (5.2)$$

осында  $Z_{\text{осн}}$  – негізгі жалақы;  
 $Z_{\text{доп}}$  – қосымша жалақы.

Негізгі жалақы құнын есептеу үшін орташа тәуліктік пайданы және әрбір қызметкердің нақты уақыт туралы деректер пайдаланылады.

Орташа тәуліктік жалақы:

$$D = 3P_m * D_p \quad (5.3)$$

осында  $3P_m$  – айлық жалақысы;  
 $D_p$  – бір айлық жұмыс күндер саны;

Дизайнер:

$$D = 100000/21 = 4761 \text{ теңге/күн};$$

Программист:

$$D = 120000/21 = 5714 \text{ теңге/күн};$$

Қызметкердің сағыттық жалықысы келесі келтірілген формуламен есептелінеді:

$$H = D / \text{Чр} \quad (5.4)$$

осында  $D$  - Орта тәуліктік жалақы қызметкері;

$\text{Чр}$  - жұмыс күні сағат саны (8 сағат).

Дизайнер:

$$H = 4761/8 = 595 \text{ теңге/сағ};$$

Программист:

$$H = 5714/8 = 714 \text{ теңге/сағ};$$

Жұмыстың әрбір күн сайын орындалатын цикл уақытын есептейтін формула:

$$t = T/q_n * Z * K \quad (5.4)$$

осында  $T$  - еңбек кіріс кезеңі, жұмыс нормасы сағат;

$q_n$  - жұмыс орындаушылардың саны;  
 $Z$  - жұмыс уақыты,  $Z = 8$  сағат;  
 $K$  - уақыт нормаларын орындау коэффициенті,  $K = 1,1$ .

Нәтиже мәнін ең жақын бүтін күнге дейін дөңгелетеміз:

$T1 = 16/1 * 8 * 1.1 \approx 2$  күн ; Программист, проблеманы тұжырымдау;

$T2 = 16/1 * 8 * 1.1 \approx 2$  күн ; Дизайнер, әдебиеттерді қарастыру;

$T3 = 16/1 * 8 * 1.1 \approx 2$  күн ; Программист, әдебиеттерді қарастыру;

$T4 = 8/1 * 8 * 1.1 \approx 1$  күн ; Программист, жобалауды талдау;

$T5 = 24/1 * 8 * 1.1 \approx 3$  күн ; Дизайнер, «директивалар» зерттеу;

$T6 = 48/1 * 8 * 1.1 \approx 6$  күн ; Дизайнер, интерфейсті дамыту;

$T7 = 56/1 * 8 * 1.1 \approx 7$  күн ; Дизайнер, қажетті графикалық элементтерін құру;

$T8 = 96/1 * 8 * 1.1 \approx 12$  күн ; Программист, өтінімдерді әзірлеу;

$T9 = 16/1 * 8 * 1.1 \approx 2$  күн ; Программист, бағдарламаны тестілеу;

$T10 = 16/1 * 8 * 1.1 \approx 2$  күн ; Программист, бағдарламаны өңдеу;

$T11 = 8/1 * 8 * 1.1 \approx 1$  күн ; Программист, баяндаманы тіркеу.

5.5 – кесте қызметкерлердің басты жалақы құнын есептеу, нәтижелер қорытындылары

Жұмыс сатысы	Орындаушы	Еңбек саны		Цикл және күн ұзақтығы	Бір сағатта төленетін жалақы	Жалақының саны
		Сағат нормасы	% барлық еңбек санынан			
Мәселені тұжырымдау	Программист	16	5	2	714	11424
Әдебиет тізімін талдау	Дизайнер	16	5	2	595	9520
Әдебиет тізімін талдау	Программист	16	5	2	714	11424
Жобалауды талдау	Программист	8	2,5	1	714	5712
«директивалар зерттеу»	Дизайнер	24	7,5	3	595	14280
Интерфейсті дамыту	Дизайнер	48	15	6	595	28560
Қажетті графикалық элементтерін құру	Дизайнер	56	17,5	7	595	33320
Өтінімдерді әзірлеу	Программист	96	30	12	714	68544
Бағдарламаны тестілеу	Программист	16	5	2	714	11424



Бағдарламаны өңдеу	Программист	16	5	2	714	11424
Баяндаманы тіркеу	Программист	8	2,5	1	714	5712
Барлығы		320	100	40		21134,4

Осылайша, негізгі жалақы өндірілген бағалауы бойынша 211344 теңге шықты.

Жалпы жалақы негізгі жалақысынан 10% құрайды және мына формула бойынша есептеледі:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \quad (5.6)$$

$$Z_{\text{доп}} = 211344 * 0,1 = 21134,4 \text{ теңге.}$$

5.2 формула, жалпы жалақы қоры бойынша есептеуге болады:

$$\text{ФОТ} = 211344 + 21134,4 = 232478,4 \text{ теңге}$$

### 5.6 Әлеуметтік салық шығындарын есептеу

Әлеуметтік салық қызметкердің табысын 11% болып табылады және мынадай формула бойынша есептеледі:

$$C_{\text{Н}} = (\text{ФОТ} - \text{ПО}) * 11\% \quad (5.7)$$

мұндағы - формула бойынша есептеледі ЕТҚ 10% құрайды, және әлеуметтік салық салуға жатпайды зейнетақы жарналары:

$$\text{ПО} = \text{ФОТ} * 10\% \quad (5.8)$$

$$\text{ПО} = 232478,4 * 0,1 = 23247,84 \text{ теңге.}$$

Осылайша, әлеуметтік салық осыны құрайды:

$$C_{\text{Н}} = (232478,4 - 23247,84) * 0,11 = 23015,36 \text{ теңге.}$$

### 5.7 Амортизацияны есептеуі

Амортизация осы формуламен есептеледі:

$$A_i = (N_a * C_{\text{пер}} * N) / (100 * 12 * n) \quad (5.9)$$

осында  $N_a$  - амортизация нормасы;  
 $C_{\text{пер}}$  - жабдықтардың бастапқы құны;

N- жұмыстарды орындауға күндер саны;  
n - бір ай жұмыс күндерінің саны.

Афторизациялық шығарымы пайдаланылатын жабдықтар мен бағдарламалық қамтамасыз етуді 4.9 формуласымен сәйкес келеді:

Жабдықта:

$$A1 = 40 * 480000 * 40 / 100 * 12 * 21 = 30476 \text{ теңге}$$

Бағдарламалық қамтамасыз етуде:

$$A2 = 40 * 40000 * 40 / 100 * 12 * 21 = 2539 \text{ теңге}$$

Амортизация үшін жалпы шығындар осы формула бойынша есептеледі:

$$A = A1 + A2 \quad (5.10)$$

$$A = 30476 + 2539 = 33015 \text{ теңге}$$

### 5.8 Энергия шығындарын есептеу

Жобаның барысында техникалық құрал-жабдықтары пайдаланатып , ол деректер жабдықтарды тұтынатын электр энергиясының құнын есептеу қажет.

Пайдаланылатын электр энергиясының құнын есептеу үшін Формула 5.11 қолданамыз:

$$\mathcal{E} = \mathcal{Z}_{\text{эл.эн.об}} + \mathcal{Z}_{\text{доп}} \quad (5.11)$$

осында  $\mathcal{Z}_{\text{эл.эн.об}}$  - жабдықтар үшін энергия шығындары;

$\mathcal{Z}_{\text{доп.}}$  - қосымша қажеттіліктері үшін электр энергиясын тұтыну.

Жабдық үшін электр қуатын тұтыну 5.12 формуласын пайдалана отырып есептеледі:

$$\mathcal{Z}_{\text{эл.эн.об}} = W * T * S * K_{\text{исп}} \quad (5.12)$$

осында W- жабдықтардың қуатты тұтыну, КВт;

T- жұмыс уақыты, сағат;

S - (1кВт/сағ = 18 теңге);

$K_{\text{исп}}$  - қолдану коэффициенті (0,9);

Тұтынатын қуаты HP Envy 15-j011sg 100 Вт.  
Тұтынатын қуаты Asus Zenbook UX305 80 Вт.

Жылдан бастап сіздің смартфон және планшет қадағалаусыз зарядтау үшін тұтынылатын қуат адаптері үнемі ноутбук HP Envy 15 - j001sg қосылған, ал олар тестілеу және жөндеу қосымшалар үшін ғана пайдаланылады.

Уақыт жұмыс күні және тәулігіне жұмыс сағаттарының санына негізінде есептеледі.

Жабдықтардың электрдің осы жалпы құны:

$$Z_{эл.эн.об} = (0,1 + 0,08) * (40 * 8) * 18 * 0,9 = 933,12 \text{ теңге}$$

Қосымша қажеттіліктері құны жабдықтарды құнының 5% мөлшерінде алынады және формула бойынша есептеледі:

$$Z_{доп.} = Z_{эл.эн.об.} * 5\%$$

Осыны құрайды:

$$Z_{доп.} = 933,12 * 0,05 = 46,656 \text{ теңге}$$

Барлығы энергия шығындар:

$$\mathcal{E} = 933,12 + 46,656 = 979,776 \text{ теңге}$$

## 5.9 Үстемені санау

Үстеме шығыстардың жалпы құнының 50% деңгейінде есептеледі:

$$H = (\text{ФОТ} + O_c + A + \mathcal{E}) * 50\% \quad (5.14)$$

$$H = (232478,4 + 23015,36 + 33015 + 979,776) * 0,5 = 144744,2 \text{ теңге}$$

### 5.9.1 Жобаның жалпы құны

Осылайша бағдарламаның өзінің құны, 5.1 формуламен сәйкес:

$$C = 232478,4 + 23015,36 + 23015,36 + 979,776 + 144744,2 = 434231,7 \text{ теңге}$$

Қосымшаларды әзірлеу және оның құрылымын құнын есептеу қысқаша қорытындылары кестеде 5.6 көрсетілген:

Кесте 5.6 - Қосымшаларды әзірлеу және оның құрылымын құнын есептеу қысқаша қорытындылары

Аты	Құны, теңге
ФОТ	232478,4
Әлеуметтік салық	23015,36
Амортизациялық аударымдар	33015
Энергетикалық шығындар	979,776
Үстеме шығындар	144744,2
Барлығы	434231,7



Сурет 5.1 – Шығындар диаграммасы

### 5.9.2 Жобаның бағасы

Жобаның бағасы құны және қалаған таза табыстың тұрады:

$$Ц_{п} = С + П$$

осында С – бағдарламаның құны.

Рентабельділік бастапқы бағасы пайдалану үшін қажетті деңгейін анықтайды. Осы сала үшін 25% құрайды:

$$Ц_{п} = С * (1 + P/100) \quad (5.16)$$

осында Р – рентабельділік.

$$Ц_{п} = 434231,7 * (1+25/100) = 651347.55 \text{ теңге}$$

ҚҚС - қосылған құн салығы.

ҚҚС Қазақстан Республикасының Салық кодексіне сәйкес, бұл жағдайда, яғни 12%, болып табылады.

$$ҚҚС = 78161,64 \text{ тг.}$$

Нәтижесінде, біз жобаны тең бағасын алу

$$Ц_{р} = 651347.55 + 78161,64 = 729509,19 \text{ теңге}$$

## **Қорытынды**

Осы дипломдық жұмыстың орындалу барысында мобильдік қосымша дамуының негізгі принциптері және оларға қойылатын талаптар зерттелді. Android операциялық жүйесіне мобильді қосымшаны құру және онымен жұмыс жасау ережелері зерттелді. Сонымен қатар, осы зерттелген тақырыптың заманауи тұрғыда, мобильді қосымшалардың өзектілігі өте жоғары екені орындалды.

Аталынып отырған қосымша платформаның барлық талаптары мен стандарттарына сай келеді. Құрылғымен ыңғайлы жұмыс жасауға және керекті деректерді аз уақыт ішінде табуға септігін тигізеді.

Өнімнің жалпы сома бағасы есептелінді, оған қосымшаның жасалу барысында шығындалуы мүмкін әртүрлі шығындар соммасы көмегімен есептелінді.

Тіршілік қауіпсіздігі бөлімінде қызметкерлердің жұмыс орнының барлық қажетті жағдайлары ескеріліп, бұл жұмыс орнының барлық стандарттарға сай келетіні анықталды. Оған қоса, жұмыс орнының өрт немесе т.б. жағдайлардағы эвакуация сұлбасы сызылды.

## Қолданылған әдебиеттер тізімі

1. Голощапов А. Google Android программирование для мобильных устройств. Санкт-Петербург 2011-438 б.
2. Брайан Х., Филлипс Б. Программирование под Android. Питер 2014592 б.
3. Амелин К.С., Граничин О.Н., Кияев В.И, Корявко А.В. Введение в разработку приложений для мобильных платформ. Санкт-Петербург 2011-505 б.
4. Лорен Д., Кондор Ш., Android программирование приложений под операционную систему Google 2011 – 464 б.
5. Бурнет Э., Привет, Андроид! Разработка мобильных приложений. Питер, 2012 – 256 б.
6. Блог на хабре о разработке под Андроид [Электронный ресурс] // URL: [http://habrahabr.ru/blogs/android\\_development/](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 б.
12. Сайт <https://www.youtube.com/user/freshgamer10/videos>
13. Сайт <http://startandroid.ru/ru/>
14. Сайт <http://developer.android.com/intl/ru/index.html>
15. Бадагуев Б.Т. Документация по охране труда в организации. – М.: Альфа-пресс, 2010.
16. Рето М., Android 2. Программирование приложений для планшетных компьютеров и смартфонов. – М. Эксмо, 2011.
17. Дюсебаев М.К., Бегимбетова А.С. Методические указания к выпускной работе (для студентов всех форм обучения специальностей 050719 – Радиотехника электроника и телекоммуникации, 050704 – Вычислительная техника и программное обеспечение). – Алматы: АИЭС, 2008.
18. СНиП РК 2.04-05-2002 «Естественное и искусственное освещение. Общие требования. – Астана, 2002.

19. Коробко, В.И. Охрана труда: Учебное пособие для студентов вузов / В.И. Коробко. – М.: ЮНИТИ-ДАНА, 2013. – 239 б.
20. Абдимуратов Ж. С., Маманбаева С. Е. Расчет производственного помещения. – Алматы: АУЭС, 2009.
21. Айдарханова М., Основы экономической теории. – М.: Фолиант, 2010.
22. Махотина М.В., Симоненко В.И. Экономика в схемах: Учебное пособие. – М.: Эксмо, 2011.
23. Носова С.С., Экономическая теория. – М.: Кнорус, 2010.
24. Артамонова В.С., Иванова С.А. Экономическая теория. – СПб.: Питер, 2010.
25. Камаева В.Д., Лобачевой Е.И. Экономическая теория: Учебник.– М.: Юрайт, 2010.
26. Выпускная работа бакалавров. Экономический раздел: методичка/ Под ред. Бабич А.А., Казыкен Б.Б., Сагира А.А. – Алматы: АУЭС, 2009
27. Сайт <http://www.fandroid.info/>
28. Сайт <https://developer.android.com/index.html>
29. Сайт <http://stackoverflow.com/>
30. Сайт <https://www.youtube.com/user/vitaxafication>



## Қосымша А

### *MainActivity*

```
package com.example.ee.lhelper_10;

import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button java, c, python;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        java = (Button) findViewById(R.id.button);
        /*ImageButton ingbtn = (ImageButton) findViewById(R.id.imageButton);

ingbtn.setImageDrawable(getResources().getDrawable(R.drawable.ic_info_black_24dp));*/
        c = (Button) findViewById(R.id.button2);
        python = (Button) findViewById(R.id.button3);
        Typeface font = Typeface.createFromAsset(getAssets(), "SanFran.ttf");
        java.setTypeface(font);
        c.setTypeface(font);
        python.setTypeface(font);
        final Intent intent = new Intent(this, Main2Activity.class);

        java.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String d = "JAVA";
                intent.putExtra("variableJava", d);
                startActivity(intent);
            }
        });
    }
}
```

```

    }
});
c.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String c = "c";
        intent.putExtra("variableC", c);
        startActivity(intent);
    }
});
python.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String p = "Python";
        intent.putExtra("variablePyth", p);
        startActivity(intent);
    }
});
}
}

```

### *Main2Activity*

```

package com.example.ee.lhelper_10;

import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;

```

```

import java.util.ArrayList;

public class Main2Activity extends AppCompatActivity {

    public ArrayAdapter<String> adapter;
    private DatabaseHelper mDatabaseHelper;
    public SQLiteDatabase mSqliteDatabase;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        EditText inputsearch = (EditText)findViewById(R.id.editText);
        ListView l1 = (ListView)findViewById(R.id.listView);
        final ArrayList<String> arr1 = new ArrayList<String>();
        final ArrayList<String> arr2 = new ArrayList<String>();
        adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, arr1);
        l1.setAdapter(adapter);
        inputsearch.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int
after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count)
{
                Main2Activity.this.adapter.getFilter().filter(s);
            }

            @Override
            public void afterTextChanged(Editable s) {

            }
        }
    }
}

```

```

});

final Intent intent = getIntent();
String CLanguageName = intent.getStringExtra("variableC");
String JAVALanguageName = intent.getStringExtra("variableJava");
String PythonLanguageName = intent.getStringExtra("variablePyth");
mDatabaseHelper = new DatabaseHelper(this, "my_new_database", null,1);
SQLiteDatabase sdp;
sdp = mDatabaseHelper.getWritableDatabase();
/*ContentValues values2 = new ContentValues();
values2.put(DatabaseHelper.COMMAND_NAME, "System.out.println()");
values2.put(DatabaseHelper.COMMAND_DISCRIPTION, "Это команда
ВЫВОДА значения на консоль.\n" +
    "Example: System.out.println>Hello World!!!");
sdp.insert("Java", null, values2);

ContentValues values3 = new ContentValues();
values3.put(DatabaseHelper.COMMAND_NAME, "for");
values3.put(DatabaseHelper.COMMAND_DISCRIPTION, "Это цикл. Так
же есть цикл while. Синтаксис: for(тип переменная, переменная < n,
условия);\n"+
    "Пример: for(int a = 1, a < 10, a++);");
sdp.insert("Java", null, values3);
Cursor cursor3 = sdp.query("Java", new String[]
{DatabaseHelper.COMMAND_NAME,
    DatabaseHelper.COMMAND_DISCRIPTION},
    null, null, null,
    null, null);
ContentValues values4 = new ContentValues();
values4.put(DatabaseHelper.COMMAND_NAME, "while");
values4.put(DatabaseHelper.COMMAND_DISCRIPTION, "Это цикл. Так
же имеется do...while\t Синтаксис: while(условие){ Тело; }\n" +
    "Пример: While(a < 20) { System.out.println(a);}");
sdp.insert("Java", null, values4);
Cursor cursor4 = sdp.query("Java", new String[]
{DatabaseHelper.COMMAND_NAME,
    DatabaseHelper.COMMAND_DISCRIPTION},

```

```

        null, null, null,
        null, null);
    ContentValues values5 = new ContentValues();
    values5.put(DatabaseHelper.COMMAND_NAME, "do..while");
    values5.put(DatabaseHelper.COMMAND_DISCRIPTION, "Это цикл. Так же имеется while\t Синтаксис: do { Тело; } while(условие)\n"+
        "Пример:    do    {    System.out.println(it's
do...while!!!);} \n"+
        "while(a < 10)");
    sdp.insert("Java", null, values5);
    Cursor cursor5 = sdp.query("Java", new String[]
{DatabaseHelper.COMMAND_NAME,
    DatabaseHelper.COMMAND_DISCRIPTION},
    null, null, null,
    null, null);
    ContentValues values6 = new ContentValues();
    values6.put(DatabaseHelper.COMMAND_NAME, "switch");
    values6.put(DatabaseHelper.COMMAND_DISCRIPTION, "Это оператор условия.\t Синтаксис: switch(переменная) case значение: Тело; break; \n"+
        "Пример: switch (month) {\n"+
        "case 1: monthString = January;\n"+
        "break;\n"+
        "case 2: monthString = February;\n"+
        "break;});\n");
    sdp.insert("Java", null, values6);
    Cursor cursor6 = sdp.query("Java", new String[]
{DatabaseHelper.COMMAND_NAME,
    DatabaseHelper.COMMAND_DISCRIPTION},
    null, null, null,
    null, null);*/
    Cursor cursor2 = sdp.query(DatabaseHelper.DATABASE_TABLE_Java, new
String[]{DatabaseHelper.COMMAND_NAME,
    DatabaseHelper.COMMAND_DISCRIPTION},
    null, null, null,
    null, null);

while(cursor2.moveToNext()){

```

```

        String comitself =
cursor2.getString(cursor2.getColumnIndex(DatabaseHelper.COMMAND_NAME))
;
        arr1.add(comitself);
        String discriptionitself =
cursor2.getString(cursor2.getColumnIndex(DatabaseHelper.COMMAND_DISCRI
PTION));
        arr2.add(discriptionitself);
    }
    final Intent intent3 = new Intent(this, Main3Activity.class);

    11.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            intent3.putExtra("name",arr1.get(position).toString());
            intent3.putExtra("desc", arr2.get(position).toString());
            startActivity(intent3);
        }
    });

}

}

Main3Activity
package com.example.ee.lhelper_10;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

public class Main3Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.activity_main3);
TextView name = (TextView) findViewById(R.id.textView);
TextView discr = (TextView) findViewById(R.id.textView2);
Intent intent = getIntent();
name.setText(intent.getStringExtra("name"));
discr.setText(intent.getStringExtra("desc"));
}
}
```

## Қосымша Б

### *Activity\_main*

```
<?xml version="1.0" encoding="utf-8"?>
<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:orientation = "horizontal"
    android:background="@drawable/background"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ee.lhelper_10.MainActivity">
```

### <Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Java"
    android:textColor="#FFFFF"
    android:background="@android:color/transparent"
    android:id="@+id/button"
    android:layout_marginTop="38dp"
    android:textSize="75dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />
```

### <Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="C++"
    android:textColor="#54156c"
    android:background="@android:color/transparent"
    android:id="@+id/button2"
```



```
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:textSize="80dp" />
```

```
<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Python"
    android:textColor="#FFFF"
    android:background="@android:color/transparent"
    android:id="@+id/button3"
    android:layout_marginBottom="30dp"
    android:textSize="75dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true" />
```

```
</RelativeLayout>
```

```
Acitivity_main2
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_gravity="center_horizontal" />
```

```
<ListView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/listView"
    android:layout_gravity="center_horizontal" />
```

```
</LinearLayout>
```

### *Activity\_main3*

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/textView"
        android:textStyle="bold"
        android:layout_marginTop="88dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/textView2"
        android:layout_below="@+id/textView"
        android:layout_alignLeft="@+id/textView"
        android:layout_alignStart="@+id/textView"
        android:layout_marginTop="103dp" />
</RelativeLayout>
```