O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI
VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
NUKUS FILIALI

TELEKOMMUNIKATSIYA TEXNOLOGIYALARI VA KASBIY
TA’LIM FAKULTETI

5330600 - Dasturiy injiniring yo'nalishi
bitiruvchisi Yakubov Kuvonchbek Ergashevichning

“Reklama agentliklarining interaktiv mobil dasturini yaratish
(Nukus shahridagi bizler.uz nomli reklama agentligi misolida)”
mavzusidagi

BITIRUV MALAKAVIY ISHI

Kafedra mudiri:	prof. N.Uteuliev
Ilmiy rahbar:	Sh.Yadgarov

Bitiruv malakaviy ishi kafedraning dastlabki __ sonli bayonnomasi
«_____» _________________ 2019-yil himoyasidan o'tdi

Nukus-2019

O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
NUKUS FILIALI

TELEKOMMUNIKATSIYA TEXNOLOGIYALARI VA KASBIY TA’LIM FAKULTETI

DASTURIY INJINIRING KAFEDRASI

TASDIQLAYMAN
Kafedra mudiri___________
«____»__________2019 y.

Yakubov Kuvonchbek Ergashevichning
“Reklama agentliklarining interaktiv mobil dasturini yaratish
(Nukus shahridagi bizler.uz nomli reklama agentligi misolida)” mavzusidagi
bitiruv malakaviy ishiga

TOPSHIRIQ

Bitiruv malakaviy ishining mavzusi TATU NF ning «____» __________ 2019 y. buyrug’i bilan tasdiqlandi.

BMI ishini topshirish muddati: «____» _____________ 2019-yil.

BMIni bajarishga tegishli ma’lumotlar: o'quv materiallari, maruza matn materiallari, ilmiy adabiyotlar va Internet resurs materiallari.

Bitiruv malakaviy ishining tarkibi:

Kirish

1. Android Studio va Java/Kotlin dasturlash tili va uning imkoniyatlari

2. Mobil ilovalar yaratishda asosiy komponentalar va freymvorklar

3. Java dasturlash tilida reklama agentliklarining interaktiv mobil dasturini yaratish

Xulosa

Foydalanilgan adabiyotlar ro'yxati

Ilovalar

Topshiriq berilgan sana «____» __________ 2019-yil.

Ilmiy rahbar __________

Topshiriqni oldi __________

Bitiruv malakaviy ishining paragraflari bo'yicha maslahatchilar.

Paragraflar	Ilmiy rahbar (maslahatchi)	Imzo, sa’na	
		Topshiriqni berdi	Topshiriqni oldi
1	SH. Yadgarov		
2	SH. Yadgarov		
3	SH. Yadgarov		

BMI ishini bajarish rejasi:

№	Bob (paragraf) nomi	Bajarish muddati	Ilmiy rahbar (maslahatchi imzosi)
1.	Android Studio va Java/Kotlin dasturlash tili va uning imkoniyatlari	10.02.2019	
2.	Mobil ilovalar yaratishda asosiy komponentalar va freymvorklar	12.04.2019	
3.	Java dasturlash tilida reklama agentliklarining interaktiv mobil dasturini yaratish	10.05.2019	
4.	MS POWER POINT dasturida taqdimot tayyorlash	04.06.2019	

Bitiruvchi _________________ «____»___________2019-yil.

Ilmiy rahbar _________________ «____»___________2019-yil.

MUNDARIJA

Kirish

Ma`lumki, dastur mashina kodlarining qanday ketma-ketligi bo'lib, aniq bir hisoblash vositasini amal qilishini boshqaradi. Dastur vositasini yaratish jarayoni osonlashtirishishi uchun yuzlab dasturlash tillari yaratilgan. Barcha dasturlash tillarini ikki toifaga ajratish mumkin:

- quyi darajadagi dasturlash tillari
- yuqori darajadagi dasturlash tillari

Quyi darajadagi dasturlash tillariga Assambler tillari kiradi. Bu tillar nisbatan qisqa va tezkor bajariluvchi kodlarni yaratish imkonini beradi. Lekin, Assambler tilida dastur tuzish qiyin, nistaban uzoq davom etadigan jarayondir. Bunga qarama-qarshi ravishda yuqori bosqich tillari yaratilganki, ularda tabiiy tillarning (ingliz tilining) cheklangan ko'rinishidan foydalangan holda dastur tuziladi. Yuqori bosqich tillari operatorlar, berilganlarning turlari, o'zgaruchilar va dastur yozishning turli usullari tilning ifodalash imkoniyatini oshiradi va dasturni "o'qimishli" bo'lishini taminlaydi. Yuqori bosqich tillariga Fortran, PL/1, Prolog, Lisp, Basic, Pascal, C va boshqa tillarni misol keltirish mumkin. Kompyuter arxitekturasi takomillashuvi, kompyuter tarmog'ining rivojlanishiga mos ravishda yuqori darajali tillarni yangi variantlarini yuzaga kelishiga, yangi tillarni paydo bo'lishiga, ayrim tillarni yo'qolib ketishiga olib keldi. Hozirda keng tarqalgan tillarga Object Pascal, C++, C#, PHP, Java, ASP tillari hisoblanadi. Xususan, C++ tilining takomillashgan varianti sifatida Java tilini olishimiz mumkin. Java dasturlash tilining yaxshilangan va mukammalash alternativ varianti sifatida esa Kotlin dasturlash tilini tanlashimiz mumkin.

Muammoning dolzarbligi va o'rganilganlik darajasi. Insoniyat hayotining yaxshilanib borishi, bu muammolarning avtomatlashgan tartibda tizimli hal qilinishi bilan uzviy bog’liq bo’lib, jismoniy mehnat talab qiladigan ishlarning o’rnini asta-sekin texnologiyalar egallamoqda. Bu kabi texnologiyarning boshqaruvini esa albatta dasturiy vositalar va algoritmlar bajaradi.

Ishning maqsadi va vazifalari. Ishning maqsadi ilovalar ishlab chiqish orqali foydalanuvchilarga maksimal darajada qulaylik va soddalikni taqdim qilish va muammolarini qisqa vaqt ichida hal qiladigan dasturiy maxsulotlarni ishlab chiqishdan iborat. Shu o’rinda, dasturiy maxsulotning sifati va unumdorligidan tashqari uning ko’rinishi ya`ni dizayniga ham katta e`tibor qaratish lozim. Chiroyli ko’rinishni yaratish uchun animatsiyalar va zamonaviy dizayn texnologiyalaridan foydalanish talab qilinadi.

I. BOB. ANDROID STUDIO VA JAVA/KOTLIN DASTURLASH TILI VA UNING IMKONIYATLARI

1.1 Android va Java dasturlash tili va uning hozirgi kundagi o’rni

Android operatsion tizimi Google kompaniyasi tomonidan yaratilgan mobil OS hisoblanadi. Bu tizim Linux kernel ning modifikatsiyalangan versiyasi asosida qurilgan bo’lib, asosan touchscreen ya’ni sensorli ekranda boshqarish uchun ishlab chiqilgan. Masalan smartfonlar, planshetlar, aqlli soatlar va hokazo. Shuningdek, Google korporatsiyasi, Android tizimini Android TV lar, Android Auto avtomobillari, Wear OS soatlar va boshqa shu kabi qurilmalar uchun moslashtirilgan interfeysga ega tizim versiyalarini yaratib kelmoqda. Android turli xil variantlari o’yin konsollari, raqamli video kameralar, kompyuterlar va boshqa elektron qurilmalar uchun ham mavjud.

Dastlab, ushbu operatsion tizim Android Inc. firmasi tomonidan yaratilgan bo’lib, so’ngra uni Google 2005-yilda sotib olgan. Birinchi marta Android 2007-yil ommaga taqdim qilindi va 2008-yil sentyabr oyida birinchi kommersial Android qurilma sotuvga chiqarildi. Hozirgi vaqtgacha tizimning ko’plab relizlari ishlab chiqildi, joriy versiya esa 9 “Pie” 2018-yil avgust oyida paydo bo’ldi. Birinchi marotaba 2019-yil 13-mart kuni Google Androidning “Android Q beta” sinov (beta) versiyasi relizini o’zining Pixel smartfonlarida o’rnatishni boshladi. Android tizimining yadrosi AOSP (Android Open Source Project) bilan mashxur va Apache License siga ega.

Shu bilan birga Android tizimi Google tomonidan yaratilgan xususiy dasturiy to’plamga ega. Ular Google Mobile Services (GMS) deb nomlanadi. Bu paket dasturlar to’plami juda ko’p qurilmalarga o’rnatilgan bo’lib, Google Chrome web brouzeri, Google Search qidiruv ilovasi, Gmail pochta ilovasi va shuningdek ilovalar do’koni va raqamli distributiv platforma Google Play ni o’z ichida oladi.

Android 2011-yildan beri dunyoda eng yaxshi sotiladigan smartfonlar va 2013-yildan boshlab eng ko’p sotilgan planshetlar operatsion tizimi hisoblanadi. 2017-yil may oyiga ko’ra oylik aktiv foydalanuvchilar soni ikki mlrd. dan ziyod operatsion tizim va 2018-yil dekabr oyidan Google Play ilovalar do’konida 2.6 milliondan ortiq ilovalar dasturchilar tomonidan joylashtirilgan.

Tizim interfeysi asosi to’g’ridan-to’g’ri boshqaruv (direct manipulation) asosida tegish harakatlaridan (touch actions) foydalangan holda, real harakatlarni amalgan oshirish imkonini beradi. Masalan, surish, chertish, tegish, ushlab turish orqali obyektlarni boshqarish, virtual klaviatura va shu kabilarni o’ch ichiga oladi. Boshqaruv interfeysi foydalanuvchiga tez javob qaytarish va bazi hollarda tebranish orqali foydalanuvchiga boshqaruv haqida xabar berish imkoniyatiga ega. Ichki qurilmalar, yani accelerometer lar, gidroskoplar, yaqinlik sensorlari va boshqa ko’plab datchiklar foydalanuvchi harakatlariga javob qaytarish uchun bazi ilovalar tomonidan ishlatiladi. Masalan, displeyning portrait yoki lanscape rejimlarini to’g’irlash yoki o’yinlarda obyektning harakatlarini boshqarish va boshqalar. Ekranning pastki qismida navigatsiya paneli, yuqorida status bar joylashgan. Status barda vaqt, quvvat, telefon signallari va ilovalar bildirishnomalari haqida indicator ko’rinishida namoyon bo’ladi.

Ilovalar qurilma imkoniyatlarini kengaytiradi, ular Android Software Development Kit (SKD) va Java dasturlash tili yordamida yoziladi. Shuningdek tizim Go dasturlash tilini ham qo’llab-quvvatlaydi, garchi ilovalar interfeyslariga cheklovlar qo’yilgan bo’lsa ham. 2017-yil may oyida Google Android ilovalarni ishlab chiqish uchun Kotlin dasturlash tilini qo’llab-quvvatlashini e`lon qildi.

Android Studio

Android Studio JetBrains ning IntelliJ IDEA dasturiy loyihasi bo’lib, Google ning Android operatsion tizimi uchun dasturiy taminot yaratish uchun mo’ljallangan rasmiy IDE (Integrated Development Environment) si hisoblanadi. Ushbu dasturlash muhiti macOs, Windows va Linux operatsion tizimlari uchun mo’ljallangan versiyalari mavjud. Android uchun avvalgi nativ dasturiy muhit Eclipse Android Development Tools (ADT) o’rnini xozirgi kunga kelib aynan Android Studio egallagan.

Android Studio 2013-yil, 16-mayda Google I/O konferensiyasida e`lon qilingan. Eng birinchi versiyasi 0.1 dan boshlangan va 2014-yil iyunda 0.8-beta versiyasi chiqarilgan. Muqobil 1.0 versiyasi 2014-yil dekabr oyida taqdim qilingan.

Xususiyatlari

- Gradle-based qurishni qo’llab-quvvatlash
- Android-maxsus kodni refaktrolash va tuzatish
- Lint-tools unumdorlikni oshirish, barqarorlashtirish, versiyalar nazorati va boshqa muammolarni yechish
- ProGuard integratsiyasi va ilovani (app-signing) identifikatsiya qilish
- Template-based umumiy Android dizaynlarni va komponentlarni yaratish
- Drag and drop orqali komponentlarni boshqarish imkoniga ega bo’lgan va ko’p ekranlarda ilovaning ko’rinishini ko’rish qobiliyatiga ega layout tahrirlovchi.
- Android Wear uchun ilovalarni yaratishni qo’llab-quvvatlash

- Google Could Platform uchun qurishni qo’llab-quvvatlash, Firebase Could Messaging va Google App Engine larni integratsiya qilish imkoniyati
- Android Virtual Device (AVD Emulator) ilovalarni virtual debug qilish

Android Studio IntelliJ ning deyarli barcha dasturlash tillarida ishlay oladi. Masalan, Java, C++ va boshqa kengaytirilgan tillar, Go kabilar. Android ning 3.0 versiyasidan boshlab Kotlin dasturlash tili kiritildi. Shuningdek, Java 7, Java 8 va Java 9 ning bazi xususiyatlaridan foydalanadi.

Java

Java dasturlash tili yuqori darajali obyektga yo’naltirilgan til bo’lib, asosini klasslar tashkil etadi. Ba’zan uni klasslar tili deb ham atashadi. Dasturchilar orasida “bir marta yoz, hamma joyda ishga tushir (write once, run anywhere - WORA)” iborasi mavjud. Chunki, Javada yozilgan kod barcha qurilmalarda va platformalarda ishlay oladi. Odatda, Java kodlarini Java Virtual Mashina (JVM) baytkodga aylantiradi va ishga tushiradi. Java dasturlash tilining sintaksisi C va C++ dasturlash tillariga juda o’xshashdir. 2018-yildan boshlab Java GitHub platformasida eng ko’p foydalaniladigan tillardan biri hisoblanadi. Xususan, client-server web ilovalarda 9 milliondan ortiq dasturchilar aynan Java dasturlash tilidan foydalanadi.

Java James Gosling tomonidan Sun Microsystems (hozirda Oracle) kompaniyasida yaratilgan. 1995-yilda ilk relizi Sun Microsystems ning komponenti sifatida Java Platform nomi bilan taqdim qilingan.

Eng oxirgi Java SE 12 versiyasi 2019-yil mart oyida chiqarildi.

Yaratilish prinsplari:

1. U sodda, qulay va obyetga yo’naltirilgan bo’lishi kerak
2. U mustahkam va xafsiz bo’lishi kerak
3. U betaraf yani arxitektura yoki qurilma saylamasligi kerak
4. U yuqori samaradorlik ega bo’lgan holda ishlashi kerak
5. U interpretatsiyalangan, oqimlarni bilan ishlovchi va dinamik bo’lishi kerak.

Nashrlari (Editions):

1. Java Card - smart kartalar uchun
2. Java Platform, Micro Edition (Java ME) – cheklangan resursli muhitlar uchun
3. Java Platform, Standart Edition (Java SE) – ishchi muhitlari uchun
4. Java Platform, Enterprise Edition (Java EE) – keng distributivli korxona yoki internet muhitlari uchun

Sintaksisi (“Salom Dunyo” misolida):

```
public class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Salom Dunyo!"); // Ushbu matn konsol ekranida chiqadi.
    }
}
```

Bu klassning fayl nomi klass nomi bilan bir xil bo'lishi shart va .java kengaytmasida bo'lishi zarur. Masalan: HelloWroldApp.java.

1.2. Kotlin dasturlash tili

Kotlin – kross-platformali, statik ravishda yozilgan, yuqori darajali dasturlash tili. Kotlin Java dasturlash tili bilan to'liq o'zaro aloqa va faoliyatni amalga oshiruvchi imkoniyatga ega qilib loyihalashtirilgan va JVM versiyalarining standart kutubxonalaridan ham foydalanadi. Lekin Kotlin dasturlash tili sintaksisi Javaga nisbatan ancha qisqartirilgan va soddalashtirilgan. Kotlin JVM asosida ishlaydi, lekin shuningdek, JavaScript va nativ kodga ham kompilyatsiya qila oladi. JetBrains kompaniyasi Kotlinga homiylik qiladi va Google tomonidan Kotlin Foundation tuzilgan.

Kotlin dasturlash tili Google ning Android operatsion tizimi uchun ilovalar ishlab chiqishda rasmiy maqomga ega. 2017-yil oktabr oyidan Android Studio ning 3.0 versiyasidan boshlab, Java kompilyatorning alternativ standartiga kiritildi. Android Kotlin foydalanuvchiga Java 6 yoki Java 8 oraqlig'idagi mos baytkodga aylantirishga ruxsat beradi.

Kotlinning bosh maqsadi bugungi kunda Java foydalaniladigan barcha joylarda, Javaning o'rnini bosa oladigan qisqaroq, samaraliroq, ishonchliroq muqobil bilan ta'minlashdir. Java nihoyatda mashhur til hisoblanadi va u keng miqyosdagi muhitda: Smart card dan tortib Google, Twitter, LinkedIn va boshqa internet sohasidagi kompaniyalar tomonidan yuritiladigan eng katta ma'lumot bazalarida ham foydalaniladi. Ushbu joylarning ko'pchiligida, Kotlindan foydalanish dasturchilarga kamroq asabiylashish va kamroq kod yozishlar bilan o'z maqsadlariga erishishida ko'maklashadi.

Kotlindan foydalanadigan eng ko’p tarqalgan dasturlash sohalar, bular:

- **Server-side** (odatda Web dasturning Back-end lari);
- **Android OS** ilovalari;

Bundan tashqari

- Intel multi-OS Engine da **iOS** ilovalari
- JavaFX orqali **Desktop** dasturlari

Kotlinning kelajak versiyalarida boshqa platformalarga ham moslashtirish ham ko’zda tutilgan. Kotlinning maqsadi anchayin keng. Kotlin faqatgina bir soha muammolariga yoxud, bugungi kunda dasturchilar duch keladigan ma’lum turdagi qiyinchilikni hal qilishga qaratilmagan. Aksincha, u dasturlash jarayonida yuzaga keladigan barcha holatlarda samaradorlikni oshirishga yo’naltiriligan.

Xuddi Java dasturlash tili singari, Kotlin dasturlash tili ham statik dasturlash tillaridan biridir. Bu shuni anglatadiki, dasturdagi har bir ifoda turi kompelyatsiya vaqtida ma’lum bo’ladi, va kompilyator siz foydalaniyotgan funksiya va o’zgaruvchilarni siz ishlatayotgan obyektlarda mavjudligini tekshiradi. Bu boshqa JVMda ishlovchi dasturlash tillari bo’lgan Groovy va JRuby kabi dinamik dasturlash tillaridan farq qiladi. Chunki dinamik dasturlash tillarda o’zgaruvchi va funksiyalar qaytaradigan turlari dastur ishlashi davomida aniqlanadi. Bu esa kamroq kod va ma’lumotlar strukturasini yaratishdagi ko’proq moslashuvchanlikka imkon beradi. Biroq yomon tomoni shundaki, nomlarni noto’g’ri yozish kabi muammolar kompilyatsiya vaqtida aniqlanmaydi va bu dastur ishlashi davomida xatoliklar yuzaga kelishiga sabab bo’ladi.

Boshqa tomondan qaraganda, Javadan farqli o’laroq, Kotlin sizning kodingizdagi har bir o’zgaruvchining turini batafsil aniqlashni talab qilmaydi. Ko’p holatlarda, o’zgaruvchilarni turi vaziyatdan kelib avtomatik tarzda aniqlanishi mumkin va bu sizga turini e’lon qilishni tushirib qoldirishga imkon yaratadi.

Misol:

```
// BMI, Ishi! misol
fun main() {
  val scope = "BMI"
  println("BMI, $scope!")
}

fun main(args: Array<String>) {
  for (arg in args) {
    println(arg)
  }
}
```

Kotlin o’zgaruvchining int turida ekanligini avtomatik tarzda aniqlaydi. Kompilyatorning turlarni aniqlay olish qobiliyati type inference deb ataladi.

Quyida statik dasturlash tillarning ba’zi foydali jihatlari berilgan:

- **Effektlilik**—metodlarni nomlash tezroq sodir bo’ladi chunki dastur ishlashi vaqtida metodlarni nomlashga ehtiyoj bo’lmaydi;

- **Ishonchlilik**—kompilyator dasturning to’g’riligini tasdiqlaydi, shu sababli dastur ishlashi vaqtida nosozliklar sodir bo`lish imkoni juda kam;
- **Qo’llab-quvvatlash**—notanish kod bilan ishlash osonroq, chunki siz kod qanday obektlar bilan ishlayotganini ko’ra olasiz;
- **Uskuna ta’minoti**—statik yozish qayta foydalanishga, kodni aniq-ravon tugatishga va boshqa IDE xususiyatlari uchun imkon beradi.

Kotlinning turni avtomatik aniqay olishi sababli, koddan ortiqcha keraksiz so’zlarining ko’pchiligi yo’qolib ketadi, chunki o’zgaruchilarni e’lon qilishga ehtiyoj qolmaydi.

Agarda Kotlin tizimining aniqligiga e`tibor berilsa, ko’plab o’xshash tushunchalarni ko’rish mumkin. Turlar, interfeyslar, “generic”lar Javaga o’xshash usulda ishlaydi, shuning uchun Javaga oid bilimlar orqali Kotlinni osongina o’rganish mumkin. Shunga qaramasdan, ba’zi narsalar yangidir. Ulardan eng muhimi Kotlinning NullPointerException qaytarishi mumkin bo’lgan holatlarni kompilyatsiya vaqtida aniqlab, nullable turlar yordamida ishonchliroq dasturlarni yozishga imkon beradi.

Kotlin tili, shu jumladan kompilyator, kutubxonalar va barcha tegishli vositalar butunlay ochiq kodli va har qanday maqsadda foydalanish uchun bepul. Bu Apache 2 litsenziyasiga ega va GitHub da ochiq bo’lib, barcha dasturchilar o’z hissalarini qo’shishlari mumkin. Kotlin ilovalarini ishlab chiqish uchun 3ta ochiq kodli IDE ni tanlash mumkin: 1. IntelliJ IDEA Community Edition (va IntelliJ IDEA Ultimate), 2. Android Studio, 3 Eclipse.

II. BOB. MOBIL ILOVALAR YARATISHDA ASOSIY KOMPONENTALAR VA FREYMVORKLAR

2.1 Eng ko’p foydalaniladigan freymvork va komponentlar

Androidning asosiy va qo’shimcha komponentlari

Asosiy komponentlar:	**Qo’shimcha komponentlar:**
• Activitylar • Serviceslar • Broadcast Receiverslar • Content Provayderlar	• Fragmentlar • Viewlar • Layoutlar • Resourcelar • Manifest

Activitylar	Har bir ilovada mavjud bo’lishi shart
Viewlar / Foydalanuvchi interfeysi	
Intentlar	
Servislar	Har bir ilovada mavjud bo’la oladi
Content Providerlar	
Broadcast Receiverlar	

Ilovalar komponentlari Android ilovalarni qurishning ajralmas bo’lagi hisoblanadi. Komponentlar ilovaning Android Manifest.xml fayli erkin e`lon qilingan va izohlangan, ularning xususiyatlari keltirilgan bo’lishi kerak.

Componentlar	Izoh
Activitylar	Ular foydalanuvchi interfeysini hosil qiladi va foydalanuvchining smartfon displeyidagi harakatlarini ushlaydi
Serviceslar	Ular ilovalarning orqa fon rejimida ishlayotgan birlashgan jarayonlarning ishlashini ta`minlaydi
Broadcast Receiverlar	Android OS ilovalari o’rtasidagi kommunikatsiyani ta`minlash vazifasini bajaradi
Content Providerlar	Tizimdagi ma’lumotlar bazasi va muammolarini boshqarish vazifasini bajaradi

Ilovalar va APK fayllar

Android ilovalar APK faylga (Android application package) o’ralgan (packaged) yoki yig’ilgan va arxivlangan fayllar to’plamidir. APK fayllar Java kodlar va boshqa resurslarni ya`ni matnlar va rasm kabilarni o’z ichiga oladi.

Activitylar

Activitylar yolg’iz screenni foydalanuvchi interfeysi orqali tasvirlaydi. Android activitylar ilovalarning foydalanuvchi interfeysi bir ko’rinishli screeni hisoblanadi. Foydalanuvchi interfeysini bu tarzda tasvirlash xuddi Windows ning ishchi oynalariga o’xshab ketadi. Ilovalarda bir yoki ko’plab activity ko’rinishlari bo’lishi mumkin, bu ko’p ko’rinishli yoki bir ko’rinishli bo’lishi mumkinligidan dalolat beradi. Ilovalar ishga tushganda birinchi bo’lib bosh activitydan boshlanadi va unga yana qo’shimcha activitylar qo’shilishi mumkin.

Servicelar

Android Servicelar orqa fondagi yani fon rejimidagi jarayonlarni ishga tushirish va ma`lum operatsiyalarni bajarish uchun mo’ljallangan. Xattoki ilova ko’rinmayotgan bo’lsa ham. Servicelar uchun foydalanuvchi interfeysi zarur emas. Servicelar belgilangan vaqt intervalida, masalan har soatda, serverdagi malumotlarni tekshirishi, ulardan backup ko’chirishi, malumot yuborishi va hokazo amallari bajarish imkoniyatiga ega. Misol uchun, foydalanuvchi musiqani orqa fonda eshitib, boshqa ilovalarda ishlashi yoki foydalanuvchi activityni bloklashiga qaramasdan serverdan malumot qabul qilishi mumkin.

Broadcast Receiverlar

Radioeshittirish qabul qiluvchilari – bu komponent ilovalar o’rtasida yoki tizim bilan ilova o’rtasidagi aloqani taminlash va xabarlar almashish imkonini beradi. Masalan, biror ilova internetdan fayl ko’chirib olsa, u bu haqda boshqa ilovani ogohlantirishi mumkin. Aynan manashu narsa broadcast deb ataladi. Broadcast Receiverlar BroadcastReceiver deb nomlanuvchi subklassni implement qiladi. Har bir broadcast Intent obyekti sifatida uzatiladi.

Content Providerlar

Content taminlovchisi boshqa ilovalarga so’rov yuborib ularning malumotlarini qabul qilishi mumkin. Bu kabi so’rovlarni Content Resolver klassi metodlari amalga oshiradi. Ma`lumotlar fayl tizimida, malumotlar bazasida yoki boshqa biror joyda saqlangan bo’lishi mumkin. Content provider klassini implement qiladi va o’tkazmalarni yani malumotlar almashishni bajarish uchun boshqa ilovalarning API lari ishga tushirilgan bo’lishi kerak bo’ladi. Masalan, mobil raqamlar saqlangan bazani Content provider yordamida boshqa ilovalar ko’rishi va foydalanishi mumkin. Yoki shunga o’xshash SMS xabarlarni ko’rishi mumkin.

Qo’shimcha komponentlar

Fragmentlar

Androidda fragmentlar butunlay foydalanuvchi interfeysi hisoblanadi. Odatda, faqat ekranning malum bir bo’laklarini yoki butunlay yuzasini egallashi mumkin. Fragmentlar activitylar ichida foydalaniladi. Shuningdek, ular turli xil activitylarda qayta ishlatilishi mumkin. Fragment odatda ko’rinishlar (View) va ko’rinish guruhlari (ViewGroup) lar bilan deyarli bir xil bo’ladi.

Layout XML Fayllar

Activitylar, fragmentlar va bazi ko’rinish guruhlari o’zlarining layuotlari va kontentlarida XML fayllarni ishlata oladi. XML fayl komponentlar xuddi GUI (Graphics User Interface) lar, fragmentlar va activitylar kabidir. Shunday qilib, bu komponent foydalanuvchi interfeysini qurishda ishlatiladi (o’lchamlar, chegaralar, oraliqlar va h.k).

Intentlar

Android intentlar kichik obyektlar bo’lib, activitylar ularni Android operatsion tizimiga uzatadi va foydalanuvchi harakatlariga kerakli javoblarni qaytaradi. Misol uchun, foydalanuvchi biror rasmni yuborishi kerak bo’lsa, ilova yuborilish haqidagi so’rovchi tizimga yuboradi va tizim unga mos ravishda javob qaytaradi, masalan qaysi xizmatlar orqali yuborishini so’rashi mumkin(bluethooth, mms va h.k). Ko’pincha intentlar “ulashish harakatini” bildiradi. Boshqa ilovalar qanday “ulashish harakati” qandayligi belgilaydi va kerakli buyruqni bajaradi.

Widgetlar

Android widgetlar activitydan tashqarida ko’rina olish xususiyatiga ega bo’lgan foydalanuvchi grafik interfeysi komponentidir. Masalan, ob-havo malumotini ko’rsatuvchi widgetlar Androidning bosh ekranlarida joylashishi mumkin. Widgetlar ilovaning implement qilingan va birlashtirilgan bo’lagi bo’lib hisoblanadi.

Ba`zan ko’rinishlar(View lar) “widget”lar deb nomlangan. Masalan, ko’pchilik foydalanuvchi grafik interfeysi komponentlari (View subklass lari) Java paketlari ichida joylashgan android.widget deb nomlanadi. Lekin, GUI komponentlari widget lar bilan bir xil emas, ya`ni ular Android qurilmaning bosh ekranida joylashga olmaydi.

View va View Group lar

Android GUI elementlari uchta toifaga bo'lingan: Activity lar, ko'rinishlar (View) va ko’rinish guruhlari (View Groups). Activity lar ekranlar/oynalar. Ko’rinishlar, matnni ko'rsatadigan matn ko'rinishi, foydalanuvchilarni bosishi mumkin bo'lgan tugma kabi boshqa GUI elementlarini o’z ichiga oladi. Ko'rinish guruhlar ko'rish uchun mo’ljallangan konteynerlardir. View Group guruhlari birlashuvlar to'plamini hosil qiladi. Ko'rinishlar va ko'rinish guruhlarini bir faoliyat yoki ichki qismda (bir faoliyat doirasida qayta kiritilgan) ichiga joylashtirish mumkin.

Resurslar

Androiddagi resurslar sizning ilovangizdan foydalanadigan tasvirlar, strings va boshqa materiallar kabi narsalarga ishora qiladi, lekin ba'zi dasturiy tillar manba kodi ko'rinishida emas. Foydalanuvchi interfeysi resurslarning boshqa bir turidir. Layoutlarni IDE ning strukturasi va drag-and-drop GUI ishlab chiqaruvchisi yoki XML shaklida qo'l bilan tuzilgan bir vosita yordamida tuzish mumkin. Ba'zan tuzilgan foydalanuvchi interfeysi turli xil qurilmalarda ishlaydi: telefonlar, planshetlar, televizorlar va hk. Ba'zan yaratilgan interfeys turli muhitlar uchun moslashtirilishi kerak.

AnroidManifest.xml fayli

AndroidManifest.xml fayli sizning to'plamingiz, jumladan, dasturning tarkibiy qismi, faoliyati, xizmat ko'rsatuvchi, eshittirish qabul qiluvchilar, kontent provayderlari va hokazolar kabi ma'lumotlarni o'z ichiga oladi. U boshqa vazifalarni ham bajaradi, jumladan:

- Ruxsatlar bilan himoyalangan qismlarga kirish uchun ilovani himoya qilishga javobgardir.
- Bundan tashqari, Android api-ni dasturda ishlatishini e'lon qiladi.
- Asboblar sinflarni ro'yxatini ko'rsatadi. Asboblar sinflarini, profil va boshqa ma'lumotlarni beradi. Ushbu ma'lumotlar dastur nashr etilguncha va undan keyin o’chirib yuborilishi mumkin.

Quyidagi misolda barcha android ilovalarda bo’lishi zarur bo’lgan manifest xml faylining root direktoriyasi keltirilgan:

```
<manifest xmlns:android=http://schemas.android.com/apk/res/android
          package=”com.kuvonchbek.yakubov.graduation.qualification.work.test”
          android:versionCode=”1”
```

```
      android:versionName=”1.0” >

      <uses-sdk
          android:minSdkVersion=”8”
          android:targetSdkVersion=”15” />

    <application
        android:icon=”@drawable/ic_launcher”
        android:label=”@string/app_name”
        android:theme=”@style/AppTheme” >
        <activity
            android:name=”.MainActivity”
            android:label=”@string/title_activity_main” >
            <intent-filter>
                <action android:name=”android.intent.action.MAIN” />

                <category android:name=”android.intent.category.LAUNCHER” />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Freymvorklar

1. GSON

Gson Java kutubxonasi bo'lib java obyektlarni JSON ichida tartiblash va ularni boshqarish imkonini beradi. APIlar orqali tez-tez xabar almashish kerak bo'lib qolsa, odatda JSONdan foydalaniladi. Chunki, u juda yengil va XML ga qaraganda sodda tuzilgan.

```
// Serialize
String userJSON = new Gson().toJson(user);

// Deserialize
User user = new Gson().fromJson(userJSON, User.class);
```

2. Retrofit

Retrofit web saytlardan REST API larni java interfeyslar ichiga qaytaruvchi kutubxona bo'lib, loyihaning API chaqiruvlarini tashkil qiluvchi ajoyib yechimdir. Toza va oddiy kod orqali, URL ga bog'liq bo'lgan metodlar va bir qancha parametrlarni o'z ichiga oladi.

Parametrlar orqali APIga murojaat qilish va URLni osonlik bilan boshqarish yoki so'rovlarni amalga oshirish mumkin.

Ushbu kutubxona ikki turdagi metodni qo'llab-quvvatlaydi. Bular sinxron va asinxron. Quyidagi misolda ular keltirilgan:

```
public interface RetrofitInterface {

    // asinxron callback
    @GET("/api/user_foydalanuvchi")
    User getUser(@Query("user_id") int userId, Callback<User>
callback);

    // sinxron post metodi
    @POST("/api/user/register")
    User registerUser(@Body User user);
}


// misol
RetrofitInterface retrofitInterface = new RestAdapter.Builder()

.setEndpoint(API.API_URL).build().create(RetrofitInterface.class
);

// 2048-id orqali foydalanuvchi malumotlarini olish
retrofitInterface.getUser(2048, new Callback<User>() {
    @Override
    public void success(User user, Response response) {

    }

    @Override
    public void failure(RetrofitError retrofitError) {

    }
});
```

Retrofit odatda Gson formatida malumotlarni qaytaradi. Shuning uchun, unga alohida o’zgartirilgan metodlar orqali amalga oshirish shart emas.

3. EventBus

EventBus kutubxonasi turli xil ilovalar o'rtasidagi aloqa va xabar almashishlarni oddiy va sodda usulda amalga oshirish imkonini beradi. Masalan, Activity dan turib ishlab turgan servisga biror malumot yuborish, yoki fragmentlar o'rtasida oson munosabatlarni yo'lga qo'yish (interaction). Quyidagi misolda, Internet bilan aloqa uzilganda qanday qilib activityni ogohlantirishda ishlatiladigan kod keltirilgan:

```
public class NetworkStateReceiver extends BroadcastReceiver {

    // internet bilan aloqa yo’qolganda post qiluvchi hodisa
    public void onReceive(Context context, Intent intent) {
        super.onReceive(context, intent);
        if(intent.getExtras()!=null) {
            NetworkInfo ni=(NetworkInfo)
intent.getExtras().get(ConnectivityManager.EXTRA_NETWORK_INFO);
            if(ni!=null && ni.getState()==NetworkInfo.State.CONNECTED) {
                // there is Internet connection
            } else if(intent

.getBooleanExtra(ConnectivityManager.EXTRA_NO_CONNECTIVITY,Boolean.FALSE)) {
                // no Internet connection, tarmoq holati o’zgarganini
yuborish
                EventBus.getDefault().post(new NetworkStateChanged(false));
            }
}

// hodisa
public class NetworkStateChanged {

    private mIsInternetConnected;

    public NetworkStateChanged(boolean isInternetConnected) {
        this.mIsInternetConnected = isInternetConnected;
    }

    public boolean isInternetConnected() {
        return this.mIsInternetConnected;
    }
}

public class HomeActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EventBus.getDefault().register(this); // register EventBus
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        EventBus.getDefault().unregister(this); // unregister EventBus
    }

    // method that will be called when someone posts an event
NetworkStateChanged
    public void onEventMainThread(NetworkStateChanged event) {
        if (!event.isInternetConnected()) {
            Toast.makeText(this, "Aloqa uzilgan!",
Toast.LENGTH_SHORT).show();
        }
    } }
```

4. ActiveAndroid

ActiveAndroid Android tizimi uchun ORM bo'lib, SQLite dan alohida, qurilmadagi malumotlar bazasi bilan SQL so'rovlar yozmasdan kommunikatsiya qilish imkoniga ega. Obyektni malumotlar omborida saqlash quyidagicha amalga oshiriladi:

```
user.save();
```

u quyidagi SQL sorov bo’lishi mumkin edi:

```
INSERT INTO Users (Nickname, Name, Address, City, PostalCode, Country)
VALUES ('Batman','Bruce W','Palisades 21','Gotham','40000','USA');
```

Ushbu kod esa barcha foydalanuvchilarni qabul qiladi:

```
List<User> users = new Select().from(User.class).execute();
```

uning SQL so’rovi quyidagicha bo’ladi:

```
SELECT Nickname, Name, Address, City, PostalCode, Country FROM
Users;
```

ActiveAdroid da malumolar bazasi bilan ishlashda ko’pgina qulaylik va ortiqcha kod xalos bo’lishda muhim ahamiyat kasb etadi. Shuningdek, unga o’xshash boshqa yana bir nechta ochiq kodli kutubxonalar ham mavjud. Masalan, GreenDAO, ORMLite va hokazo.

5. Universal Image Loader

UIL asinxron tarzda rasmlarni yuklash va saqlash xususiyatiga ega bo'lgan kutubxona. U juda ham oddiy va foydalanish uchun qulay:

```
imageLoader.displayImage(imageUri, imageView);
```

Shu bilan birga, rasmlarni yuklash va saqlash uchun Picasso ham yaxshi kutubxona. Lekin u to’liq o’zgartirish kiritish va boshqarishni taminlamaydi. UIL esa deyarli barcha konfiguratsiya va parametrlarini o’zgartirishga ruxsat beradi (ayniqsa, katta o’lchamdagi rasmlarni yuklash va saqlashda).

Ko’plab ochiq kodli kutubxonalar ilovalar yaratishni osonlashtiribgina qolmay, ularni qisqa vaqtda tayyorlashda ham katta yordam beradi. Ommabop kutubxonalar tez-tez yangilanib va testdan o’tkazilib turiladi. Android Studio da ularni Maven orqali osongina loyihaga import qilish mumkin. Quyidagi misolda build.gradle faylida kutubxonalarni import qilish ko’rsatilgan:

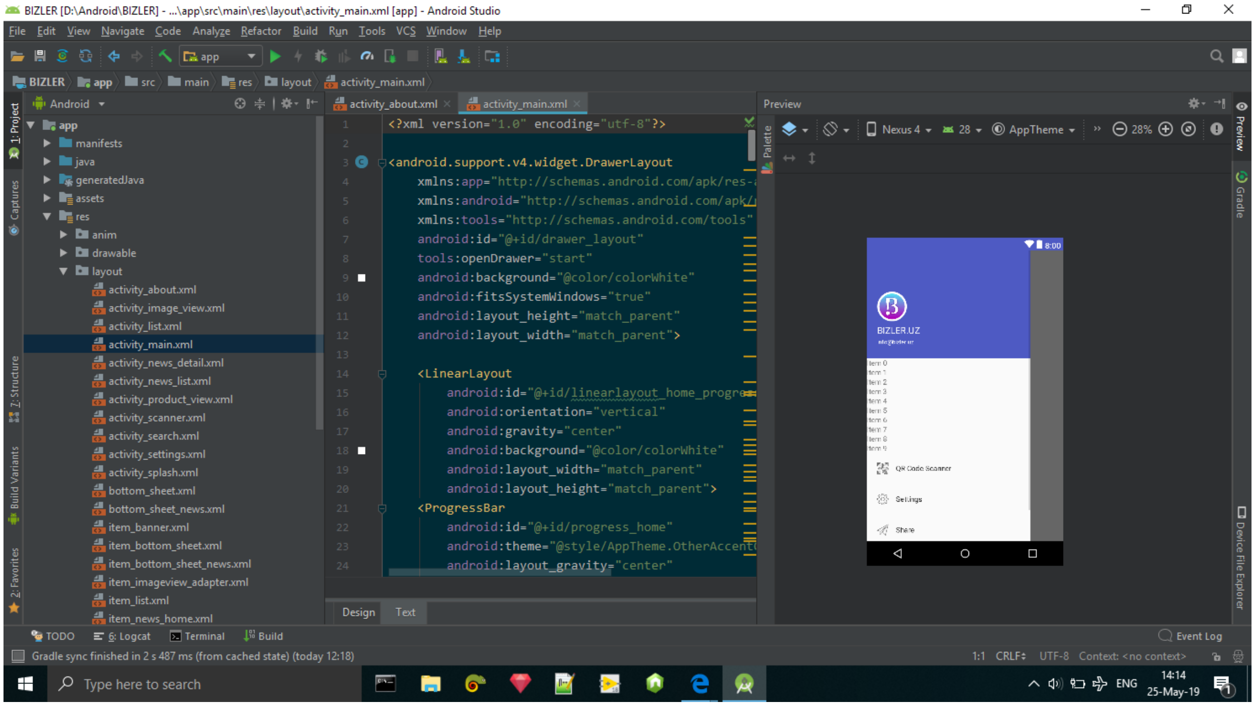
```
dependencies {
    implementation 'com.google.code.gson:gson:2.2.4'
    implementation 'com.squareup.okhttp:okhttp:1.3.0'
    implementation 'com.squareup.retrofit:retrofit:1.3.0'
    implementation 'de.greenrobot:eventbus:2.2.+'
    implementation 'com.nostra13.universalimageloader:universal-
image-loader:1.9.1'
}
```

Gradle ni sinxronizatsiya qilgandan keyin ushbu kutubxonalarni loyihda implement qilish mumkin bo’ladi.

III. BOB. JAVA DASTURLASH TILIDA REKLAMA AGENTLIKLARINING INTERAKTIV MOBIL DASTURINI YARATISH

3.1. Android Studio Java dasturlash tilida ilovaning foydalanuvchi interfeysini (UI) ishlab chirish va ma’lumotlar bazasiga bog’lash

Dasturning foydalanuvchi interfeysini ishlab chiqishning bir nechta yo’llari bor, masalan ko’rinishlarni drop-and-drag qilib layoutga joylashtirish yoki XML kodlar yordamida boshidan yozib chiqish va yana bir usuli ko’rinishni dinamik java kodlar yordamiga yaratish. Ushbu diplom ishida interfeysni yaratish uchun XML kodlardan foydalanildi. Malumotlar bazasiga ulanish uchun esa Retrofit freymvorki qo’llanildi. Asosan, malumotlar serverdan qabul qilanadi va qurilmaning malumotlar bazasidan foydalanilmaydi.



BIZLER [D:\Android\BIZLER] - ...\app\src\main\res\layout\activity_main.xml [app] - Android Studio

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

<android.support.v4.widget.DrawerLayout
    xmlns:app="http://schemas.android.com/apk/res-a
    xmlns:android="http://schemas.android.com/apk/r
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    tools:openDrawer="start"
    android:background="@color/colorWhite"
    android:fitsSystemWindows="true"
    android:layout_height="match_parent"
    android:layout_width="match_parent">

    <LinearLayout
        android:id="@+id/linearlayout_home_progre
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <ProgressBar
        android:id="@+id/progress_home"
        android:theme="@style/AppTheme.OtherAccent
        android:layout_gravity="center"
```

Gradle sync finished in 2 s 487 ms (from cached state) (today 12:18)

activity_main.xml fayli:

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

```
<android.support.v4.widget.DrawerLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    tools:openDrawer="start"
    android:background="@color/colorWhite"
    android:fitsSystemWindows="true"
    android:layout_height="match_parent"
    android:layout_width="match_parent">
    <LinearLayout
        android:id="@+id/linearlayout_home_progressbar"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@color/colorWhite"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <ProgressBar
        android:id="@+id/progress_home"
        android:theme="@style/AppTheme.OtherAccentColor"
        android:layout_gravity="center"
        android:visibility="visible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    </LinearLayout>
    <LinearLayout
        android:id="@+id/linearlayout_home"
        android:gravity="center"
        android:layout_centerInParent="true"
        android:visibility="invisible"
        android:orientation="vertical"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content">
        <ImageView
            android:layout_height="80dp"
            android:layout_width="80dp"
            android:src="@drawable/fail"/>
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="@string/please_check_your_internet_connection"
            android:textSize="22sp"
            android:gravity="center"
            android:layout_gravity="center_horizontal"
            android:padding="10dp"/>
        <Button
            android:id="@+id/refresh_button_home"
            android:text="@string/retry"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
<android.support.v4.widget.NestedScrollView
    android:id="@+id/nested_scroll"
    android:visibility="invisible"
    android:layout_height="match_parent"
    android:layout_width="match_parent">

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
android:background="#ffffff"
tools:context=".MainActivity">

<ImageView
    android:id="@+id/imageview_home"
    android:src="@color/colorPrimary"
    android:layout_width="match_parent"
    android:layout_height="220dp" />

<android.support.design.card.MaterialCardView
    android:id="@+id/cardview_home"
    android:layout_centerHorizontal="true"
    android:clickable="true"
    android:foreground="?attr/selectableItemBackground"
    app:cardBackgroundColor="#999999"
    app:cardCornerRadius="2dp"
    app:cardElevation="0dp"
    android:layout_marginTop="130dp"
    android:layout_width="340dp"
    android:layout_height="220dp"
    android:focusable="true">

    <com.glide.slider.library.SliderLayout
        android:id="@+id/slider"
        android:gravity="center"
        app:auto_cycle="true"
        android:layout_gravity="center"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</android.support.design.card.MaterialCardView>
```

```xml
<ImageView
    android:id="@+id/imageview_logo"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:layout_alignParentTop="true"
    android:layout_width="150dp"
    android:layout_height="35dp" />
<LinearLayout
    android:id="@+id/linearlayouthome1"
    android:orientation="horizontal"
    android:layout_marginTop="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/imageview_logo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
<ImageView
    app:srcCompat="@drawable/ic_location"
    android:layout_gravity="center"
    android:layout_width="14dp"
    android:layout_height="14dp" />
<TextView
    android:id="@+id/textview_address"
    android:textColor="@color/colorWhite"
    android:layout_marginLeft="5dp"
    android:layout_gravity="center_vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>

<LinearLayout
```

```
    android:id="@+id/linearlayouthome2"
    android:orientation="horizontal"
    android:layout_marginTop="5dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/linearlayouthome1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <ImageView
        app:srcCompat="@drawable/ic_phone"
        android:layout_gravity="center"
        android:layout_width="14dp"
        android:layout_height="14dp" />

    <TextView
        android:id="@+id/textview_contactphone"
        android:textColor="@color/colorWhite"
        android:layout_gravity="center_vertical"
        android:autoLink="phone"
        android:textColorLink="@color/colorWhite"
        android:layout_marginLeft="5dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
<android.support.design.card.MaterialCardView
    android:id="@+id/cardview_search"
    android:clickable="true"
    android:layout_margin="15dp"
    android:layout_alignParentLeft="true"
    app:cardBackgroundColor="@android:color/transparent"
    app:cardCornerRadius="0dp"
    app:cardElevation="0dp"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:focusable="true">
    <ImageView
        android:layout_gravity="center"
        app:srcCompat="@drawable/ic_menu_white"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</android.support.design.card.MaterialCardView>
<android.support.design.card.MaterialCardView
    android:id="@+id/cardview_info"
    android:clickable="true"
    android:layout_margin="15dp"
    android:layout_alignParentRight="true"
    app:cardBackgroundColor="@android:color/transparent"
    app:cardCornerRadius="8dp"
    app:cardElevation="0dp"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:focusable="true">
    <ImageView
        app:srcCompat="@drawable/ic_search_actionbar"
        android:textSize="20sp"
        android:textColor="@color/colorPrimary"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</android.support.design.card.MaterialCardView>
```

```
    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerview_home"
        android:layout_below="@id/cardview_home"
        android:layout_marginTop="5dp"
        android:background="@android:color/transparent"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <include layout="@layout/layout_for_news"
        android:layout_marginTop="25dp"
        android:layout_below="@+id/recyclerview_home"
        android:layout_width="match_parent"
        android:layout_marginBottom="20dp"
        android:layout_height="wrap_content" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="100dp"/>
</RelativeLayout>
</android.support.v4.widget.NestedScrollView>


    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        >


        <ScrollView
```

```xml
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">


    <RelativeLayout
        android:id="@+id/nav_header_imageview"
        android:background="@color/colorPrimary"
        android:layout_height="210dp"
        android:layout_width="match_parent">

        <LinearLayout
            android:orientation="vertical"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="80dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/circle_imageview"
        android:src="@drawable/logotip_img"
        app:civ_border_width="4dp"
        app:civ_border_color="@color/colorWhite"
        android:layout_width="58dp"
        android:layout_height="58dp" />

        <TextView
            android:id="@+id/textview_logotip_navmenu"
            android:text="BIZLER.UZ"
            android:layout_marginTop="5dp"
            android:textSize="18sp"
            android:autoLink="web"
            android:textColorLink="@color/colorWhite"
            android:textColor="@color/colorWhite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
            <TextView
                android:text="info@bizler.uz"
                android:layout_marginTop="3dp"
                android:autoLink="email"
                android:textSize="11sp"
                android:textColorLink="@color/colorWhite"
                android:textColor="@color/colorWhite"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </LinearLayout>
    </RelativeLayout>
    <android.support.v7.widget.RecyclerView
    android:id="@+id/expandablerecyclerview"
    android:clickable="false"
    android:nestedScrollingEnabled="false"
    android:indicatorLeft="250dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            app:srcCompat="@drawable/ic_qrcode"
            android:layout_gravity="center_vertical"
            android:layout_marginLeft="20dp"
            android:layout_width="22dp"
            android:layout_height="22dp" />
        <TextView
            android:id="@+id/textview_qrcode"
            android:text="@string/codescanner"
            android:clickable="true"
            android:focusable="true"
            android:gravity="center_vertical"
            android:layout_marginLeft="15dp"
            android:textColor="@color/colorBlack"
            android:layout_width="match_parent"
            android:layout_height="60dp" />
    </LinearLayout>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
       <ImageView
            app:srcCompat="@drawable/ic_cog"
            android:layout_gravity="center_vertical"
            android:layout_marginLeft="20dp"
            android:layout_width="22dp"
            android:layout_height="22dp" />
        <TextView
            android:id="@+id/textview_settings"
            android:text="@string/settings"
            android:clickable="true"
```

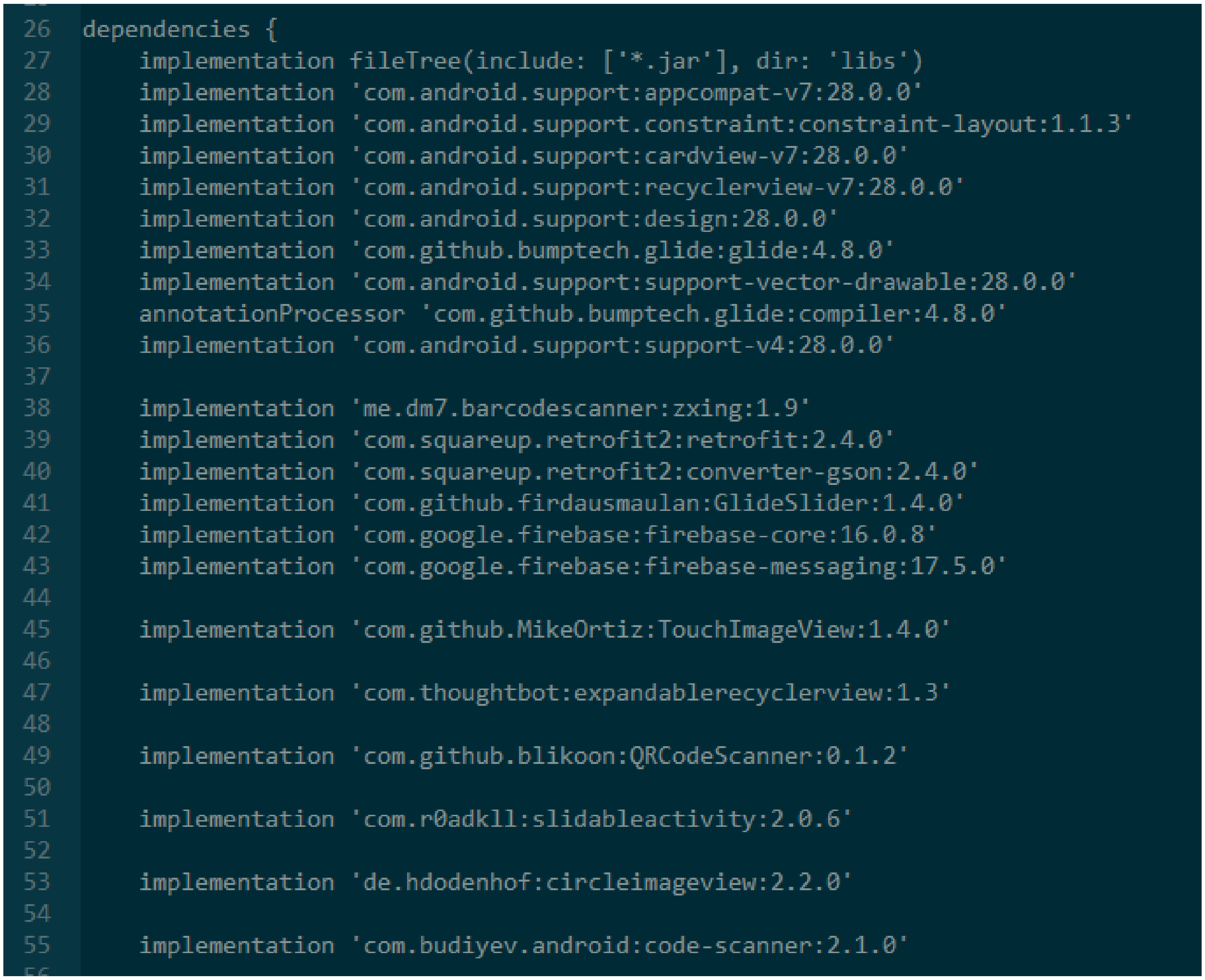
```
         android:focusable="true"
         android:gravity="center_vertical"
         android:layout_marginLeft="15dp"
         android:textColor="@color/colorBlack"
         android:layout_width="match_parent"
         android:layout_height="60dp" />
 </LinearLayout>
<LinearLayout
     android:orientation="horizontal"
     android:layout_width="match_parent"
     android:layout_height="wrap_content">
     <ImageView
         app:srcCompat="@drawable/ic_paper_plane"
         android:layout_gravity="center_vertical"
         android:layout_marginLeft="20dp"
         android:layout_width="22dp"
         android:layout_height="22dp" />
     <TextView
         android:id="@+id/textview_share"
         android:text="@string/share"
         android:clickable="true"
         android:focusable="true"
         android:gravity="center_vertical"
         android:layout_marginLeft="15dp"
         android:textColor="@color/colorBlack"
         android:layout_width="match_parent"
         android:layout_height="60dp" />
 </LinearLayout>
 <LinearLayout
     android:orientation="horizontal"
     android:layout_width="match_parent"
```

```
                android:layout_height="wrap_content">
                <ImageView
                    app:srcCompat="@drawable/ic_info_circle"
                    android:layout_gravity="center_vertical"
                    android:layout_marginLeft="20dp"
                    android:layout_width="22dp"
                    android:layout_height="22dp" />
                <TextView
                    android:id="@+id/textview_about"
                    android:text="@string/about"

                    android:clickable="true"
                    android:focusable="true"
                    android:gravity="center_vertical"
                    android:layout_marginLeft="15dp"
                    android:textColor="@color/colorBlack"
                    android:layout_width="match_parent"
                    android:layout_height="60dp" />
            </LinearLayout>
        </LinearLayout>
        </ScrollView>
    </android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>
```

3.2. Java dasturlash tilida Nukus shahridagi bizler.uz nomli reklama agentligining mobil ilovasini yaratish

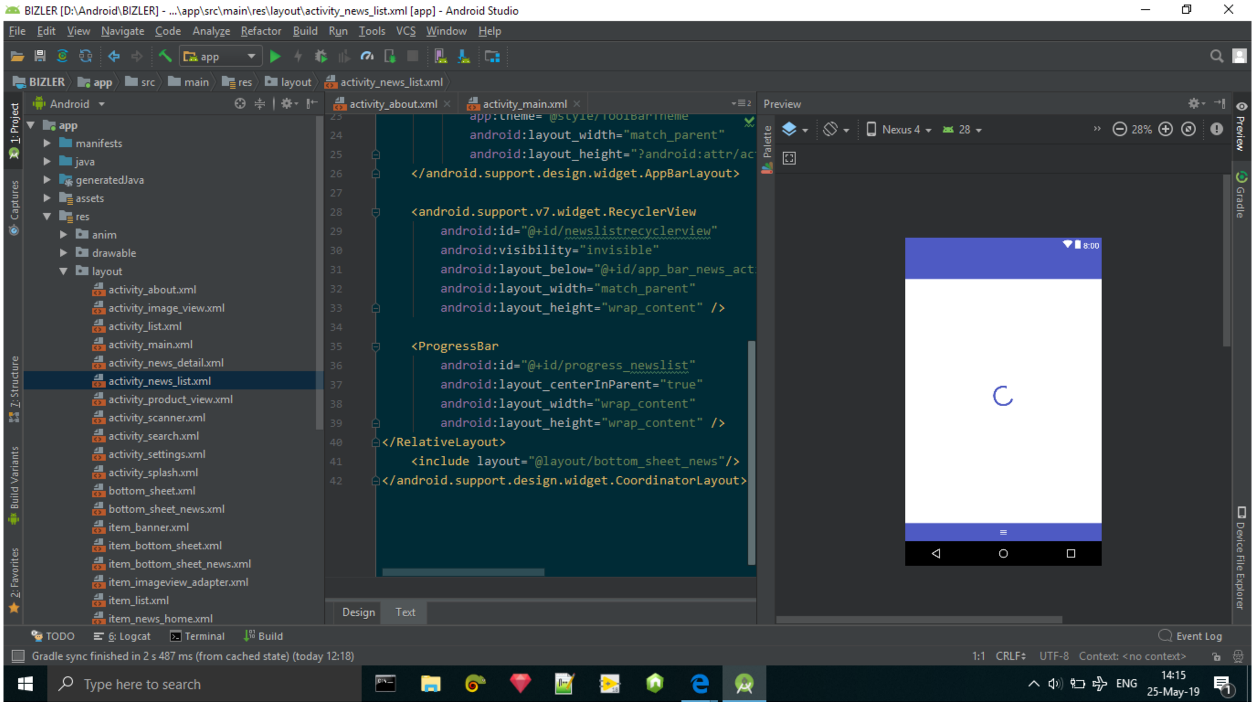
Web saytlarning Android ilovasini yaratishda eng asosiy komponent va freymvork bu Retrofit hisoblanadi. Aynan manashu komponent malumotlarni boshqarishni taminlovchi vositadir.

Bizler.uz nomli web saytning mobil ilovasini ishlab chiqish uchun quyidagi kutubxonalardan foydalanildi:



```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.github.bumptech.glide:glide:4.8.0'
    implementation 'com.android.support:support-vector-drawable:28.0.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.8.0'
    implementation 'com.android.support:support-v4:28.0.0'

    implementation 'me.dm7.barcodescanner:zxing:1.9'
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.github.firdausmaulan:GlideSlider:1.4.0'
    implementation 'com.google.firebase:firebase-core:16.0.8'
    implementation 'com.google.firebase:firebase-messaging:17.5.0'

    implementation 'com.github.MikeOrtiz:TouchImageView:1.4.0'

    implementation 'com.thoughtbot:expandablerecyclerview:1.3'

    implementation 'com.github.blikoon:QRCodeScanner:0.1.2'

    implementation 'com.r0adkll:slidableactivity:2.0.6'

    implementation 'de.hdodenhof:circleimageview:2.2.0'

    implementation 'com.budiyev.android:code-scanner:2.1.0'
```

Bildirishnoma (Notification) uchun esa, Google servicesi FCM Firebase Could Messaging dan foydalanildi. Dizayn uchun esa, asosan, CardView va RecyclerViewlardan foydalanildi. Ushbu ilova android API 16 dan yuqori versiyalarda ishlaydi (Android 4.1 JellyBean +). Androidning 6-versiyalaridan boshlab tizim resurslaridan foydalanishda ruxsatnomalarni olish uchun alohida bo’lgan joylarga shart operatorlaridan foydalangan holda, resursdan foydalanishga ruxsat so’rovlari yuborildi. Shuningdek, AndroidManifest.xml fayli ichida foydalaniladigan ruxsatnomalar elon qilindi (uses-permissions).

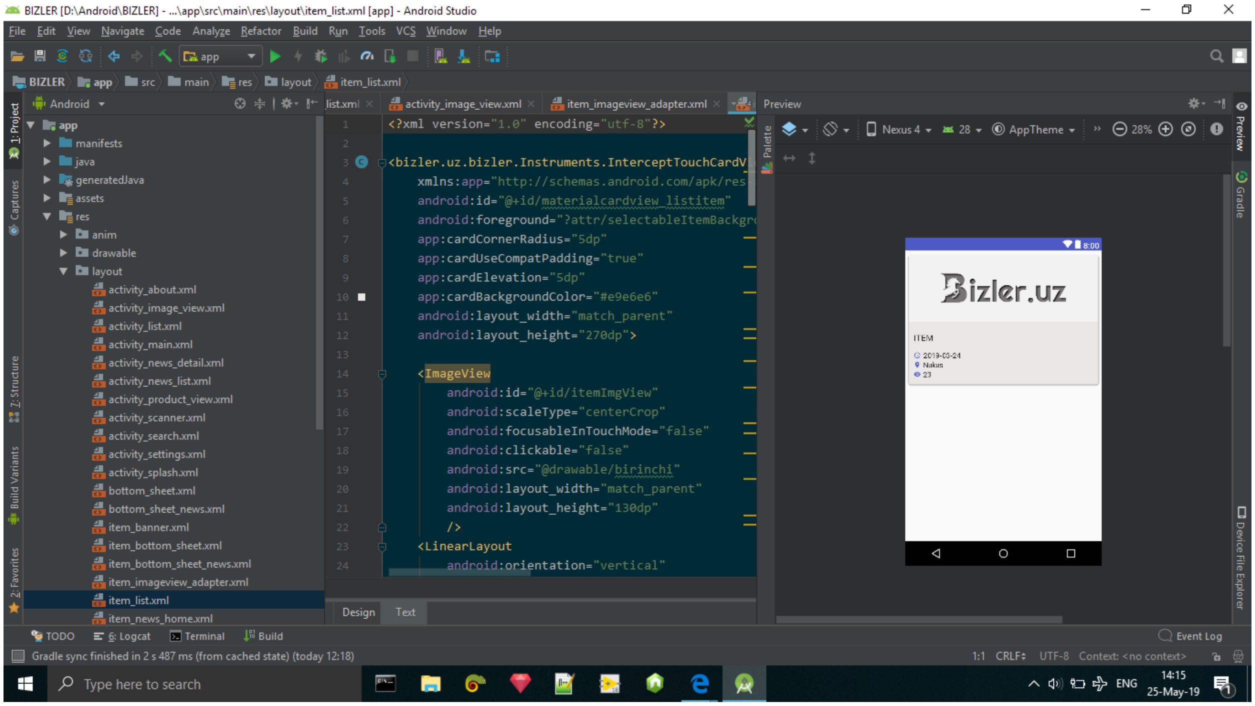


BIZLER [D:\Android\BIZLER] - ...\app\src\main\res\layout\activity_news_list.xml [app] - Android Studio

```
            app:theme="@style/ToolBarTheme"
            android:layout_width="match_parent"
            android:layout_height="?android:attr/ac
    </android.support.design.widget.AppBarLayout>

    <android.support.v7.widget.RecyclerView
        android:id="@+id/newslistrecyclerview"
        android:visibility="invisible"
        android:layout_below="@+id/app_bar_news_act
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ProgressBar
        android:id="@+id/progress_newslist"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
    <include layout="@layout/bottom_sheet_news"/>
</android.support.design.widget.CoordinatorLayout>
```

Gradle sync finished in 2 s 487 ms (from cached state) (today 12:18)

Ushbu rasmda yangiliklarning ro’yxatini chiqaruvchi ilovaning interfeysi namoyish qilingan. Unga ko’ra avval ProgressBar ya`ni bu yerda, malumotlar yuklanayotganini foydalanuvchiga bildirish uchun ishlatildi. Malumot yuklab bo’lingandan so’ng, RecyclerView va uning itemlari yani yangiliklar ko’rinadi. ProgressBar esa o’z ko’rinishini yashiradi. (progressbar.setVisibility(View.GONE)



BIZLER [D:\Android\BIZLER] - ...\app\src\main\res\layout\item_list.xml [app] - Android Studio

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

<bizler.uz.bizler.Instruments.InterceptTouchCardV
    xmlns:app="http://schemas.android.com/apk/res
    android:id="@+id/materialcardview_listitem"
    android:foreground="?attr/selectableItemBackgr
    app:cardCornerRadius="5dp"
    app:cardUseCompatPadding="true"
    app:cardElevation="5dp"
    app:cardBackgroundColor="#e9e6e6"
    android:layout_width="match_parent"
    android:layout_height="270dp">

    <ImageView
        android:id="@+id/itemImgView"
        android:scaleType="centerCrop"
        android:focusableInTouchMode="false"
        android:clickable="false"
        android:src="@drawable/birinchi"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        />
    <LinearLayout
        android:orientation="vertical"
```

Gradle sync finished in 2 s 487 ms (from cached state) (today 12:18)

Bu rasmda esa, recyclerViewning item larining namunasi keltirilgan, ushbu ko’rinish har bir malumotlar bazasidagi har bir itemning malumotini olib ko’rsatib beradi. Va uning ustiga bosganda maxsus ID raqam orqali kerakli malumot filtrlab olinadi va foydalanuvchiga taqdim qilinadi.

MainActivity.java fayli:

```
public class MainActivity extends AppCompatActivity implements
BaseSliderView.OnSliderClickListener, ViewPagerEx.OnPageChangeListener{

    List<Banner> list;

    RecyclerView recyclerview;

    List<CategoriesMenu> menuList = new ArrayList<>();

    RecyclerAdapterHome adapter;


    CardView infoBtnCardView, searchBtnCardView;


    SliderLayout slider;


    SharedPref sharedpref;
    NestedScrollView nestedScrollView;

    ProgressBar progressbar;

    LinearLayout linearlayoutHome, linearlayoutProgressbar;

    Button refreshBtn;


    TextView textviewContactPhone, textviewAddress;

    DrawerLayout drawerLayout;

    RecyclerView exRecyclerview;

    ExpandableAdapter exAdapter;

    TextView textviewQrcode, textviewSettings, textviewShare, textviewAbout;

    JsonPlaceHolderApi jsonPlaceHolderApi;

    private static final int REQUEST_CODE_QR_SCAN = 101;

    RecyclerView newsRecyclerview;
    RecyclerAdapterHomeNews adapterHomeNews;

    ImageView logotipImageView;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        final SharedPref sharedPref;

        super.onCreate(savedInstanceState);


        sharedPref = new SharedPref(this);

        LanguageHelper.setAppLocale(sharedPref.getLang(), this);

        setContentView(R.layout.activity_main);

        exRecyclerview =
(RecyclerView)findViewById(R.id.expandablerecyclerview);

        logotipImageView = (ImageView)findViewById(R.id.imageview_logo);

        drawerLayout = (DrawerLayout)findViewById(R.id.drawer_layout);

        FirebaseMessaging.getInstance().subscribeToTopic("all");
        NavigationView navigationView =
(NavigationView)findViewById(R.id.nav_view);
        if(navigationView != null){
            setupDrawerContent(navigationView);
        }
        //DrawerLayout swipe length
        try{
            Field mDragger = drawerLayout.getClass().getDeclaredField(
                    "mLeftDragger");//mRightDragger or mLeftDragger based on
Drawer Gravity
            mDragger.setAccessible(true);
            ViewDragHelper draggerObj = (ViewDragHelper) mDragger
                    .get(drawerLayout);

            Field mEdgeSize = draggerObj.getClass().getDeclaredField(
                    "mEdgeSize");
            mEdgeSize.setAccessible(true);
            int edge = mEdgeSize.getInt(draggerObj);

            mEdgeSize.setInt(draggerObj, edge * 6);
        }catch (Exception e){
            e.printStackTrace();
        }
        //nav menu custom textview items
        textviewQrcode = (TextView)findViewById(R.id.textview_qrcode);
        textviewSettings = (TextView)findViewById(R.id.textview_settings);
        textviewShare = (TextView)findViewById(R.id.textview_share);
        textviewAbout = (TextView)findViewById(R.id.textview_about);
        textviewQrcode.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT)
                {
                    startActivity(new Intent(MainActivity.this,
ScannerActivity.class));
                    drawerLayout.closeDrawer(GravityCompat.START);
                }else{
                    Intent intent = new Intent(MainActivity.this,
QrCodeActivity.class);
                    startActivityForResult(intent, REQUEST_CODE_QR_SCAN);
                    drawerLayout.closeDrawer(GravityCompat.START);
                }
            }
        });
        textviewSettings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this,
SettingsActivity.class));
                drawerLayout.closeDrawer(GravityCompat.START);
            }
        });
        textviewShare.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent shareIntent = new Intent();
                shareIntent.setAction(Intent.ACTION_SEND);
                shareIntent.setType("text/plain");
                shareIntent.putExtra(Intent.EXTRA_TEXT,
"https://play.google.com/store/apps/details?id=" + getPackageName() +
                        "  BIZLER");
                startActivity(shareIntent.createChooser(shareIntent,
"Share"));
                drawerLayout.closeDrawer(GravityCompat.START);
            }
        });
        textviewAbout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this, 
AboutActivity.class));
                drawerLayout.closeDrawer(GravityCompat.START);
            }
        });
        newsRecyclerview = 
(RecyclerView)findViewById(R.id.recyclerview_home_news);
        list = new ArrayList<>();
        sharedpref = new SharedPref(this);
        progressbar = (ProgressBar)findViewById(R.id.progress_home);
        progressbar.setVisibility(View.VISIBLE);
        linearlayoutProgressbar = 
(LinearLayout)findViewById(R.id.linearlayout_home_progressbar);
        nestedScrollView = 
(NestedScrollView)findViewById(R.id.nested_scroll);
        linearlayoutHome = 
(LinearLayout)findViewById(R.id.linearlayout_home);
        refreshBtn = (Button)findViewById(R.id.refresh_button_home);
        //Here used FONTAWESOME
        textviewContactPhone = 
(TextView)findViewById(R.id.textview_contactphone);
        textviewAddress = (TextView)findViewById(R.id.textview_address);
        slider = (SliderLayout)findViewById(R.id.slider);
        infoBtnCardView = (CardView)findViewById(R.id.cardview_info);
        searchBtnCardView = (CardView)findViewById(R.id.cardview_search);
        infoBtnCardView.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this, 
SearchActivity.class));
```

```
            }
        });
        searchBtnCardView.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                //startActivity(new Intent(MainActivity.this,
SearchActivity.class));
                drawerLayout.openDrawer(GravityCompat.START);
            }
        });
        recyclerview = (RecyclerView)findViewById(R.id.recyclerview_home);
        recyclerview.setHasFixedSize(true);

        if(isNetworkConnected()){
            linearlayoutHome.setVisibility(View.GONE);
            loadData();
        }else{
            progressbar.setVisibility(View.INVISIBLE);
            linearlayoutProgressbar.setVisibility(View.INVISIBLE);
            linearlayoutHome.setVisibility(View.VISIBLE);
            nestedScrollView.setVisibility(View.INVISIBLE);
        }
        refreshBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(isNetworkConnected()){
                    loadData();
                    linearlayoutHome.setVisibility(View.GONE);
                }else{
                    progressbar.setVisibility(View.INVISIBLE);
                    linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                    linearlayoutHome.setVisibility(View.VISIBLE);
                    nestedScrollView.setVisibility(View.INVISIBLE);
                }
            }
        });
   }

   private void setupDrawerContent(NavigationView navigationView) {
        navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem
menuItem) {
                menuItem.setChecked(true);
                drawerLayout.closeDrawers();
                return true;
            }
        });
    }
    private void loadData(){
        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl("http://bizler.uz/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        jsonPlaceHolderApi = retrofit.create(JsonPlaceHolderApi.class);
        String lang = "kr";
        if(sharedpref.getLang().equals("kr"))
            lang = "kr";
        else if(sharedpref.getLang().equals("uz"))
            lang = "uz";
        else if(sharedpref.getLang().equals("ru"))
            lang = "ru";
        Call<List<Banner>> call = jsonPlaceHolderApi.getBanners(lang);

        Call<List<CategoriesMenu>> callCategories = 
jsonPlaceHolderApi.getCategories(lang);

        Call<AboutModel> callAbout = jsonPlaceHolderApi.getAbout();

        call.enqueue(new Callback<List<Banner>>() {

            @Override

            public void onResponse(Call<List<Banner>> call, 
Response<List<Banner>> response) {

                progressbar.setVisibility(View.VISIBLE);

                linearlayoutProgressbar.setVisibility(View.VISIBLE);

                nestedScrollView.setVisibility(View.INVISIBLE);

                if(!response.isSuccessful()){

                    progressbar.setVisibility(View.INVISIBLE);

                    linearlayoutProgressbar.setVisibility(View.INVISIBLE);

                    Toast.makeText(MainActivity.this, ""+response.code(), 
Toast.LENGTH_SHORT).show();

                    return;

                }

                list = response.body();

                final RequestOptions requestOptions = new RequestOptions();

                requestOptions.centerCrop();

                for (int i = 0; i < list.size(); i++) {

                    TextSliderView sliderView = new 
TextSliderView(MainActivity.this);

                    // if you want show image only / without description text 
use DefaultSliderView instea

                    // initialize SliderLayout

                    try{

                        sliderView

 .image("http://bizler.uz"+list.get(i).getImages()[0])

                                //.description(list.get(i).getTitle())

                                .setRequestOption(requestOptions)

                                .setProgressBarVisible(true)
                                .setOnSliderClickListener(MainActivity.this);
                    }catch (Exception e){
                        e.printStackTrace();
                    }
                    //add your extra information
                    sliderView.bundle(new Bundle());
                    sliderView.getBundle().putInt("extra",
list.get(i).getId());
                    slider.addSlider(sliderView);
               }
slider.setPresetTransformer(SliderLayout.Transformer.ZoomOutSlide);
slider.setPresetIndicator(SliderLayout.PresetIndicators.Center_Bottom);
                slider.setCustomAnimation(new DescriptionAnimation());
                slider.setDuration(3000);
                slider.addOnPageChangeListener(MainActivity.this);
//                Toast.makeText(MainActivity.this, ""+list.size(),
Toast.LENGTH_SHORT).show();
//                Toast.makeText(MainActivity.this,
""+response.body().get(0).getImages()[0], Toast.LENGTH_SHORT).show();
//          //      dialog.dismiss();
                progressbar.setVisibility(View.INVISIBLE);
                linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                nestedScrollView.setVisibility(View.VISIBLE);
            }
            @Override
            public void onFailure(Call<List<Banner>> call, Throwable t) {
                progressbar.setVisibility(View.INVISIBLE);
                linearlayoutProgressbar.setVisibility(View.INVISIBLE);
            }
        });
        final GridLayoutManager gridLayoutManager = new
GridLayoutManager(this, 3);
```

```
        callCategories.enqueue(new Callback<List<CategoriesMenu>>() {
            @Override
            public void onResponse(Call<List<CategoriesMenu>> call,
Response<List<CategoriesMenu>> response) {
                progressbar.setVisibility(View.VISIBLE);
                linearlayoutProgressbar.setVisibility(View.VISIBLE);
                nestedScrollView.setVisibility(View.INVISIBLE);
                if(!response.isSuccessful()){
                    progressbar.setVisibility(View.INVISIBLE);
                    linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                    Toast.makeText(MainActivity.this, ""+response.code(),
Toast.LENGTH_SHORT).show();
                    return;
                }

//                Log.d("FOR EXAMPLE", response.body().get(2).getIcon());
                menuList = response.body();
                final int totalSize = menuList.size();
                gridLayoutManager.setSpanSizeLookup(new
GridLayoutManager.SpanSizeLookup() {
                    @Override
                    public int getSpanSize(int i) {
                        int span;
                        span = totalSize % 3;
                        if(totalSize < 3){
                            return 2;
                        }else if(span == 0 || (i <= ((totalSize - 1) -
span))){
                            return 1;
                        }else if(span == 1){
                            return 2;
                        }else{
                            return 1;
                        }
                    }
                });
                adapter = new RecyclerAdapterHome(MainActivity.this, 
menuList);
                recyclerview.setLayoutManager(gridLayoutManager);
                recyclerview.setAdapter(adapter);



                progressbar.setVisibility(View.INVISIBLE);
                linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                nestedScrollView.setVisibility(View.VISIBLE);
            }
            @Override
            public void onFailure(Call<List<CategoriesMenu>> call, Throwable 
t) {
                progressbar.setVisibility(View.INVISIBLE);
                linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                Toast.makeText(MainActivity.this, "Error connecting to 
server", Toast.LENGTH_SHORT).show();
            }
        });
        callAbout.enqueue(new Callback<AboutModel>() {
            @Override
            public void onResponse(Call<AboutModel> call, 
Response<AboutModel> response) {
                progressbar.setVisibility(View.VISIBLE);
                linearlayoutProgressbar.setVisibility(View.VISIBLE);
                nestedScrollView.setVisibility(View.INVISIBLE);
                if(!response.isSuccessful()){
                    Toast.makeText(MainActivity.this, ""+response.message() + 
" " + response.code(), Toast.LENGTH_SHORT).show();
```

```
                }
                AboutModel model = response.body();
                textviewAddress.setText(model.getAddress());
                textviewContactPhone.setText(model.getContactPhone());
                try{
                    Glide.with(MainActivity.this)
                            .load("http://bizler.uz/" + model.getLogo())
                            .apply(new
RequestOptions().placeholder(R.drawable.birinchi).error(R.drawable.birinchi))
                            .into(logotipImageView);
                }catch (Exception e){
                    e.printStackTrace();
                }
                progressbar.setVisibility(View.INVISIBLE);
                linearlayoutProgressbar.setVisibility(View.INVISIBLE);
                nestedScrollView.setVisibility(View.VISIBLE);
            }

            @Override
            public void onFailure(Call<AboutModel> call, Throwable t) {

            }
        });

        final GridLayoutManager glm = new GridLayoutManager(this, 2);
        newsRecyclerview.setHasFixedSize(true);

        Call<List<Subnews>> callNewsCat =
jsonPlaceHolderApi.getNewsCategory(lang);

        callNewsCat.enqueue(new Callback<List<Subnews>>() {
            @Override
            public void onResponse(Call<List<Subnews>> call,
Response<List<Subnews>> response) {

                List<News> templist = new ArrayList<>();


                //Log.d("EXAMP ", response.body().toString());

                templist.add(new News(getString(R.string.news),
response.body(), R.drawable.ic_newspaper));
                exAdapter = new ExpandableAdapter(templist,
MainActivity.this, drawerLayout);

                exRecyclerview.setLayoutManager(new
LinearLayoutManager(MainActivity.this));
                exRecyclerview.setAdapter(exAdapter);
            }
            @Override
            public void onFailure(Call<List<Subnews>> call, Throwable t) {
            }
        });
        Call<List<HomeNewsModel>> callhomenews =
jsonPlaceHolderApi.getNewsCategoryHome(lang);
        callhomenews.enqueue(new Callback<List<HomeNewsModel>>() {
            @Override
            public void onResponse(Call<List<HomeNewsModel>> call,
Response<List<HomeNewsModel>> response) {
                List<HomeNewsModel> listSubnews = new ArrayList<>();
                listSubnews = response.body();
                final int totalSize = listSubnews.size();
                glm.setSpanSizeLookup(new GridLayoutManager.SpanSizeLookup()
{
                    @Override
                    public int getSpanSize(int i) {
                        int span;
                        span = totalSize % 2;
                        if(totalSize < 3){
                            return 2;
                        }else if(span == 0 || (i <= ((totalSize - 1) -
span))){
                            return 1;
                        }else if(span == 1){
                            return 2;
                        }else{
                            return 1;
                        }
                    }
                });
                newsRecyclerview.setLayoutManager(glm);
                adapterHomeNews = new
RecyclerAdapterHomeNews(MainActivity.this, listSubnews);
                newsRecyclerview.setAdapter(adapterHomeNews);
            }
            @Override
            public void onFailure(Call<List<HomeNewsModel>> call, Throwable
t) {
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        if(resultCode != Activity.RESULT_OK)
        {
            if(data == null)
                return;
            String result =
data.getStringExtra("com.blikoon.qrcodescanner.error_decoding_image");
            if(result != null)
            {
                AlertDialog alertDialog = new
AlertDialog.Builder(MainActivity.this).create();
                alertDialog.setTitle("Scan error");
                alertDialog.setMessage("QR Code could not be scanned");
                alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                    }
                });
                alertDialog.show();
            }
            return;
        }
        if(requestCode == REQUEST_CODE_QR_SCAN)
        {
            if(data == null)
                return;
            final String result =
data.getStringExtra("com.blikoon.qrcodescanner.got_qr_scan_relult");


            AlertDialog alertDialog = new
AlertDialog.Builder(MainActivity.this).create();
            alertDialog.setTitle("Scan result");
            alertDialog.setMessage(result);
            alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                }
            });
            alertDialog.setButton(AlertDialog.BUTTON_POSITIVE,
getString(R.string.openwithbrowser), new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    String url = result;
                    try{
                        if(!url.startsWith("https://") &&
!url.startsWith("http://")){
                            String query = URLEncoder.encode(url, "utf-8");
                            url = "http://www.google.com/search?q=" + query;
                            Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse(url));
                            startActivity(browserIntent);
                        }else{
                            Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse(url));
                            startActivity(browserIntent);
                        }
                    }catch (Exception e){
                        e.printStackTrace();
                    }
                }
            });
            alertDialog.show();
        }
    }

    @Override
    public void onBackPressed() {


        if(drawerLayout.isDrawerOpen(GravityCompat.START)){
            drawerLayout.closeDrawer(GravityCompat.START);
        }else{
            moveTaskToBack(true);
            MainActivity.this.finish();
      }    }
    @Override
    public void onSliderClick(BaseSliderView slider) {
        int val = (int) slider.getBundle().get("extra");
        Intent intent = new Intent(this, ProductViewActivity.class);
        intent.putExtra("id", val);
      //  CommonModel.CATEGORY_SELECTED = list.get(val-1).getTitle();
        startActivity(intent);
    }
    @Override
    public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels) {
    }
    @Override
    public void onPageSelected(int position) {
    }
    @Override
    public void onPageScrollStateChanged(int state) {
    }
    @Override
    protected void onStop() {
        //slider.stopAutoCycle();
        super.onStop();
    }
    @Override
    protected void onStart() {
        slider.startAutoCycle();
        super.onStart();
    }
    public boolean isNetworkConnected(){
        ConnectivityManager cm = 
(ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
        return cm.getActiveNetworkInfo() != null;
    }
    public int getPixelFromDips(float pixels){
        final float scale = getResources().getDisplayMetrics().density;
        return (int)(pixels * scale + 0.5f);
    }
    //ExpandableRecyclerView resources
    public List<News> makeGenres() {
        return Arrays.asList(makeRockGenre());
    }
    public News makeRockGenre() {
        return new News(getString(R.string.news), makeRockArtists(), 
R.drawable.ic_newspaper);
    }
    public List<Subnews> makeRockArtists() {
        Subnews queen = new Subnews(9, "Shou biznes");
        Subnews styx = new Subnews(7, "Texnologiya");
        Subnews reoSpeedwagon = new Subnews(39,"Dunyo");
        Subnews boston = new Subnews(47, "Mahalliy");
        return Arrays.asList(queen, styx, reoSpeedwagon, boston);
    }
}
```

Yuqoridagi kod va rasmlar (Screenshotlar) dasturning ishlanishidan namuna sifatida keltirildi. Ilovada foydalanilgan barcha kodlar va resurslarning ko’pligini inobatga olgan holda, to’liq kod va fayllarni ko’rish uchun loyihaning arxiv faylidan foydalanish tavsiya qilinadi.

XULOSA

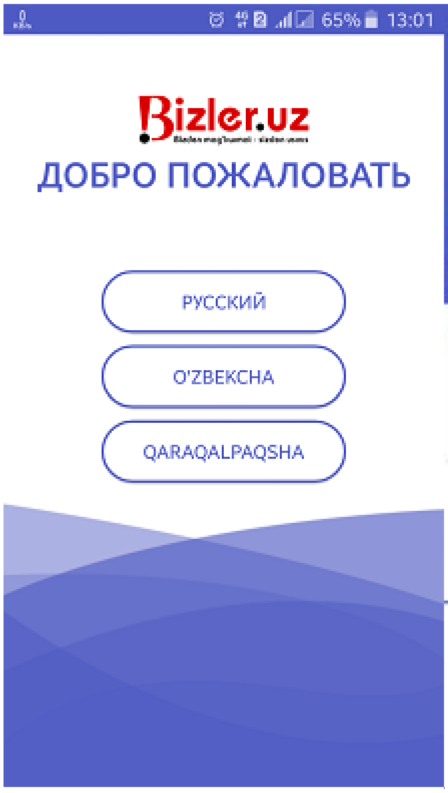
Mobil qurilmalar sonining hozirgi kundagi keng suratlarda o’sib borishi, ayniqsa, Android operatsion tizimida ishlovchi smartfon va qurilmalar jahon bozorining katta qismini egallayotgani, mobil ilovalarga bo’lgan talabga ham proportsionaldir. Shuningdek, odamlar o’z muammolarini smartfonlarda hal qilishga o’rganmoqda, masalan onlayn haridlardan tortib, virtual o’yinlar va tabriklargacha.

Shu o’rinda savol paydo bo’lishi mumkin, “Web saytlarning mobil ilovasini ishlab chiqish shartmi? Axir xoxlagan saytga brauzer orqali murojaat qilib, kerakli malumotlarni olish mumkinku”. Ilovalar o’zlarining navit xususiyatlari bilan saytlardan ajralib turadi. Ya`ni qurilmalar o’zlarining barcha huquqlarini web saytlar va brauzerlarga ruxsat bermaydilar. Aynan shuning uchun ham mobil ilovalarning ahamiyati saqlanib qolgan. Shu bilan birga, web saytlarga qaraganda ilovalar ancha tezkor hisoblanadi.

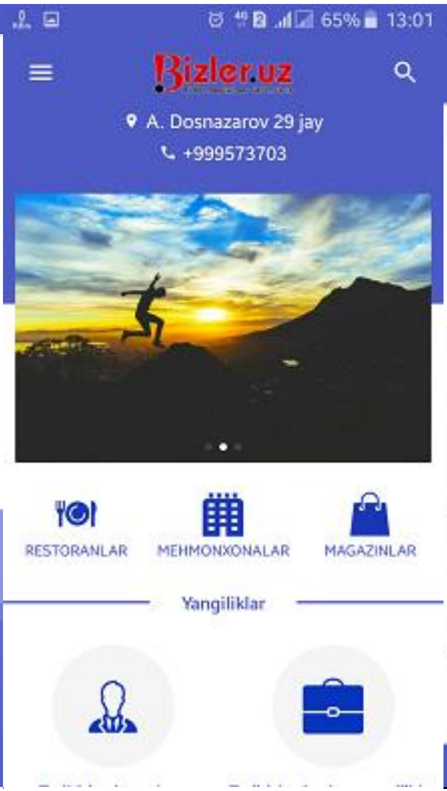
FOYDALANILGAN ADABIYOTLAR

1. https://github.com/JetBrains/kotlin/releases/latest
2. "kotlin-stdlib". kotlinlang.org. JetBrains. Retrieved April 20, 2018. Shafirov, Maxim (May 17, 2017).
3. "Kotlin on Android. Now official". Today, at the Google I/O keynote, the Android team announced first-class support for Kotlin.
4. "Kotlin FAQ". Kotlin lets you choose between generating Java 6 and Java 8 compatible bytecode. More optimal byte code may be generated for higher versions of the platform.
5. "Kotlin is now Google's preferred language for Android app development". TechCrunch. Retrieved 2019-05-08. Krill, Paul (Jul 22, 2011).
6. "JetBrains readies JVM language Kotlin". infoworld.com. InfoWorld. Retrieved February 2, 2014. Waters, John (February 22, 2012).
7. Андрей Бреслав — Kotlin для Android: коротко и ясно, retrieved 2017-05-28
8. "Basic Syntax". Kotlin. Jetbrains. Retrieved 19 January 2018.
9. https://medium.com/@kadiroff/kotlinning-asosiy-xususiyatlari-eb910b569fb1
10. http://www.techplayon.com/applications-component/
11. https://infinum.co/the-capsized-eight/top-5-android-libraries-every-android-developer-should-know-about

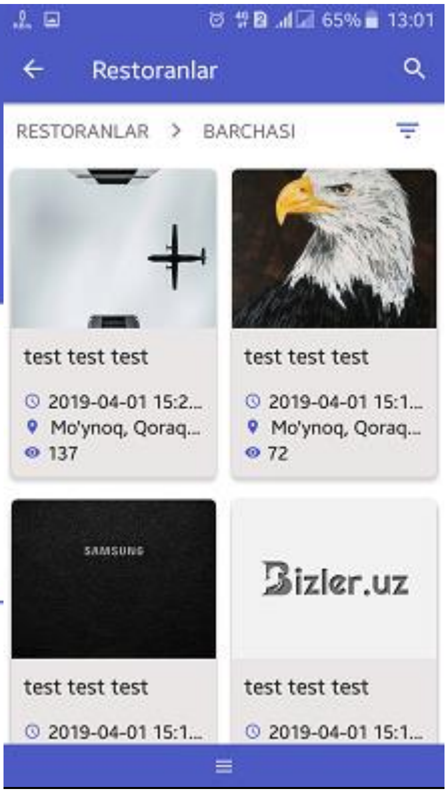
ILOVALAR



ДОБРО ПОЖАЛОВАТЬ



A. Dosnazarov 29 jay
+999573703

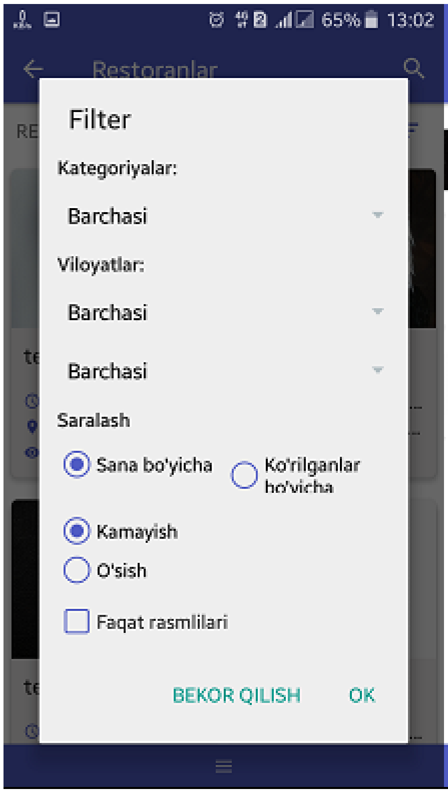


test test test
2019-04-01 15:2...
Mo'ynoq, Qoraq...
137

test test test
2019-04-01 15:1...
Mo'ynoq, Qoraq...
72

test test test
2019-04-01 15:1...

test test test
2019-04-01 15:1...



Filter

Kategoriyalar:
Barchasi

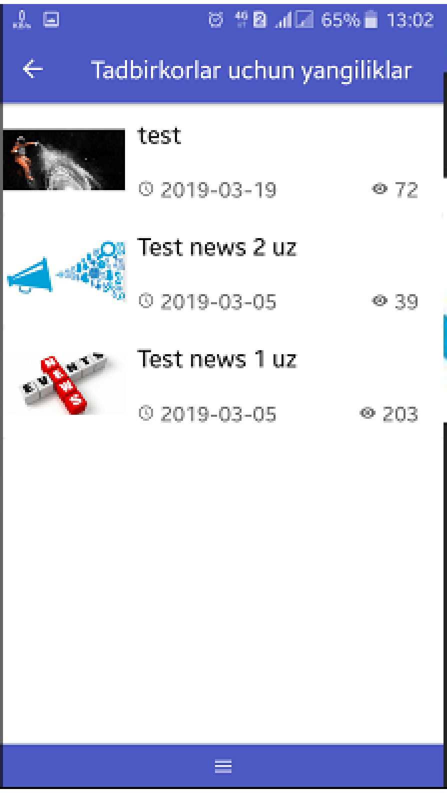
Viloyatlar:
Barchasi

Barchasi

Saralash
● Sana bo'yicha ○ Ko'rilganlar bo'yicha

● Kamayish
○ O'sish

□ Faqat rasmlilari

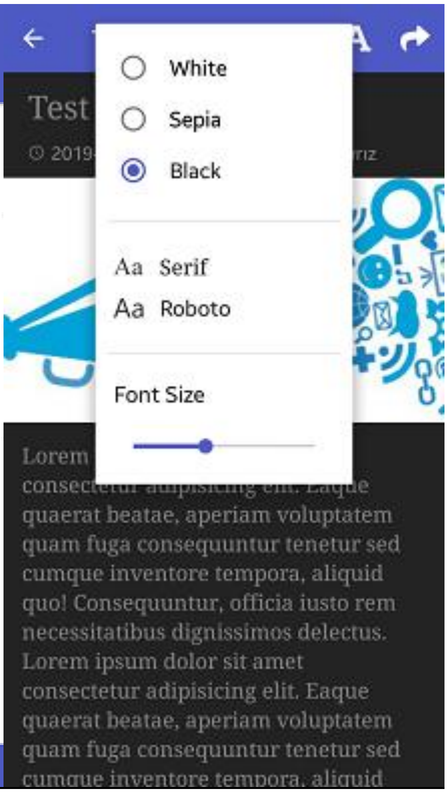


Tadbirkorlar uchun yangiliklar

test
2019-03-19 72

Test news 2 uz
2019-03-05 39

Test news 1 uz
2019-03-05 203



Test
2019

○ White
○ Sepia
● Black

Aa Serif
Aa Roboto

Font Size

Lorem
consectetur adipisicing elit. Eaque quaerat beatae, aperiam voluptatem quam fuga consequuntur tenetur sed cumque inventore tempora, aliquid quo! Consequuntur, officia iusto rem necessitatibus dignissimos delectus. Lorem ipsum dolor sit amet consectetur adipisicing elit. Eaque quaerat beatae, aperiam voluptatem quam fuga consequuntur tenetur sed cumque inventore tempora, aliquid