

Қазақстан Республикасының Білім және ғылым министрлігі

Әл-Фараби атындағы Қазақ ұлттық университеті

Бердалы Айдана Көбейқызы

ТАПСЫРЫС ЖАСАУ ПРОЦЕСІН АВТОМАТТАНДЫРУҒА АРНАЛҒАН
МОБИЛЬДІ ҚОСЫМШАНЫ ЖАСАУ

ДИПЛОМДЫҚ ЖҰМЫС

5B070400-«Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы, 2021

Қазақстан Республикасы Білім және ғылым министрлігі

әл-Фараби атындағы Қазақ ұлттық университеті

Ақпараттық технологиялар факультеті

Информатика кафедрасы

ДИПЛОМДЫҚ ЖҰМЫС

тақырыбы: «ТАПСЫРЫС ЖАСАУ ПРОЦЕСІН АВТОМАТТАНДЫРУҒА
АРНАЛҒАН МОБИЛЬДІ ҚОСЫМШАНЫ ЖАСАУ»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Орындаған

(қолы)

Бердалы А.К.

Ғылыми жетекші,
PhD, доцент м.а.

(қолы)

Бектемесов А.Т.

Қорғауға жіберілді:

Хаттама №____, «____» _____ 2021 ж.

Кафедра меңгерушісі

(қолы және мөрі)

Дарибаев Б.С.

Норма бақылаушы

(қолы)

Түркен Г.

Алматы, 2021

АҢДАТПА

Дипломдық жұмыс кіріспе, негізгі бөлім, қорытынды, пайдаланылған әдебиеттер тізімі мен қосымшаладан тұрады. Дипломдық жұмыста 49 бет, 37 сурет, 20 әдебиет, 1 қосымша бар.

Кілттік сөздер: МОБИЛЬДІ ҚОСЫМША, ANDROID STUDIO, FIREBASE CONSOLE, АВТОМАТТАНДЫРУ ЖҮЙЕСІ, ДИЗАЙН, ИНТЕРФЕЙС.

Зерттеу нысаны - Android Studio платформасы арқылы тапсырыс жасау процесін автоматтандыруға арналған мобильді қосымшаны жасау.

Жұмыстың мақсаты: онлайн тапсырыс беретін дүкендер желісінің автоматты жүйесін жасау және дамытып, қолданысқа енгізу.

Қосымшаны жасау барысында Android Studio платформасын таңдау арқылы қосымша тек андроид қолданушыларына ғана қолжетімді болатындай жасалды, қолданушыларға арналған деректер базасы құрылды, қосымша жобалау процестерінен өтті. Қосымшаның толық интерфейсінің дизайні жасалып болған соң, java тілі арқылы қосымша интерфейсі құрылды. Құрылған интерфейстерді деректер базасымен байланыстырылды.

Жұмыстың нәтижесінде мобильді қосымша толығымен өңдеуден өтті. Тестілеу нәтижесінде қателіксіз сұраныстарды автоматтандырып отыратын, онлайн саудаға мүмкіндік бере алатын мобильді қосымша жасалды. Мобильді құрылғыларда да тестілеуден өткізілді.

РЕФЕРАТ

Дипломная работа состоит из введения, основной части, заключения, списка использованной литературы и приложения. Дипломная работа содержит 49 страницу, 37 рисунок, 20 литературу, 1 приложение.

Ключевые слова: МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ANDROID STUDIO, FIREBASE CONSOLE, СИСТЕМА АВТОМАТИЗАЦИИ, ДИЗАЙН, ИНТЕРФЕЙС.

Объектом исследования является разработка мобильного приложения для автоматизации процесса создания заказов через платформу Android Studio.

Цель работы: разработка и внедрение в эксплуатацию автоматической системы сети магазинов онлайн-заказов.

В процессе разработки приложения, выбрав платформу Android Studio, приложение было разработано таким образом, чтобы оно было доступно только пользователям Android, создана база данных для пользователей, приложение прошло процессы проектирования. После того, как был разработан дизайн полного интерфейса приложения, был создан дополнительный интерфейс через язык java. Создаваемые интерфейсы были связаны с базой данных.

В результате работы мобильное приложение прошло полную обработку. В результате тестирования было разработано мобильное приложение, которое автоматизирует запросы без ошибок, позволяет торговать онлайн. Тестировался на мобильных устройствах.

ABSTRACT

The thesis consists of an introduction, a main part, a conclusion, a list of references and an appendix. The thesis contains 49 pages, 37 drawing, 20 literatures, 1 appendix.

Keywords: MOBILE APPLICATION, ANDROID STUDIO, FIREBASE CONSOLE, AUTOMATION SYSTEM, DESIGN, INTERFACE.

The object of the research is the development of a mobile application for automating the process of creating orders through the Android Studio platform.

The purpose of the work: development and commissioning of an automatic system of online order stores.

In the process of developing the application, choosing the Android Studio platform, the application was developed in such a way that it was available only to Android users, a database was created for users, and the application went through the design processes. After the design of the full application interface was developed, an additional interface was created through the java language. The interfaces that were created were linked to the database.

As a result, the mobile application has been fully processed. As a result of testing, a mobile application was developed that automates requests without errors and allows you to trade online. It was also tested on mobile devices.

МАЗМҰНЫ

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТЫЛҒАН СӨЗДЕР	6
АНЫҚТАМАЛАР	7
КІРІСПЕ	8
1. ТАПСЫРЫСТАРДЫ АВТОМАТТАНДЫРУ ПРОЦЕССИ БОЙЫНША ТӘЖІРБИЕЛЕРГЕ ШОЛУ	10
1.1. Қазақстандағы тәжірбиелер.....	10
1.2. Әлемдік тәжірбиелер.....	13
1.3. Тәжірбиені онлайн сатылым процесін автоматтандырумен байланыстыру	14
1.4. Мобильді қосымшаны әзірлеу кезеңдері	15
2. МОБИЛЬДІ ҚОСЫМШАНЫ ӘЗІРЛЕУ.....	19
2.1. Мобильді қосымшаға қойылатын талаптарды әзірлеу	19
2.2. Қолданушы интерфейсіне талдау	22
2.3. Қолданған технология және таңдалу себебі	23
2.4. Мобильді қосымша архитектурасы	25
3. МОБИЛЬДІ ҚОСЫМШАҒА ШОЛУ	26
3.1. Қосымша навигациясы.....	26
3.2. Мобильді қосымшаны құрастыру қадамдары	36
3.3. Мобильді қосымшаның деректер қорымен байланысы	42
ҚОРЫТЫНДЫ.....	47
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	49
ҚОСЫМША А.....	50

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТЫЛҒАН СӨЗДЕР

CRM – Customer relationship management. Клиенттермен қарым-қатынасты басқару, бақылау, зерттеу және болашақта сатылым мөлшерін арттыруға негізделген технология.

UI – User interface. Қолданушы интерфейсі машина мен программалық жабдықтамалар үшін жобалау.

UX –User experience. Қолданушы интерфейсіндегі дизайн мен аналитика ортасындағы бағыт.

RPA – Robotic process automation. Өртүрлі бизнес-тапсырмаларды программалық робот көмегімен автоматтандыру әдісі.

API – Application programming interface. Бағдарламалауда қолданылатын мобильді және веб қосымшалардың интерфейсі.

AVD – Android virtual device. Android эмуляторында тестілеуге болатын, эмулятор ретінде қолданылатын виртуалды құрылғы.

Android SDK – Software development kit. Программалық жабдықтама құру барысындағы жүктелетін құралдар пакеті.

XML – extensible markup language. Кеңейтілген белгілеу тілі.

АНЫҚТАМАЛАР

Аутентификация – пайдаланушы енгізген құпия сөз бен пайдаланушының кіру дерекқорында сақталған құпия сөзбен салыстыру процессі.

Интерфейс – пайдаланушы мен компьютердің қарым қатынасы кезіндегі өзара әрекеттесу жазықтығы немесе ортасы.

Прототиптеу – жүйенің тұтастай жұмысын талдау үшін болашақ өнімнің негізгі функцияларын тез арада кескіндеу әдісі.

Автоматтандыру – адамды энергияны, материалды, ақпаратты алу, түрлендіру немесе еңбек ету процестеріне қатысуды босату немесе қатысу мөлшерін азайту мақсатында техникалық құралдар мен математикалық әдістерді пайдаланатын техникалық прогресс бағыттары.

Фронтенд – қызметтің бағдарламалық-аппараттық бөлігіне қолданушы интерфейсінің клиенттік жағы.

Мобильді қосымшаны тестілеу – мобильді қосымшаның жұмыс жасауының функционалдығын, ыңғайлылығы мен дәйектігін тексеру процесі.

КІРІСПЕ

Жеткізілетін өнімнің сапасын едәуір немесе белгілі бір дәрежеде жақсартуға және онлайн тапсырыстар жасау арқылы уақыт тапшылығының орнын толтыруға, әлемдік өндірістің дамуына ықпал етуге арналған қазіргі әлемдік өндірістің дамуына ықпал етуге арналған қазіргі әлемдік экономикалық жүйе нарықта мол әр түрлі өндірушілерді өз кәсібін дамытуға және бәсекеге қабілеттілігін сақтаудың жаңа тәсілдерін табуға мәжбүр етеді.

Қазіргі жағдайда әлем бойынша ішкі өндіріс процестерін оңтайландыру мәселесі автоматтандырумен тығыз байланысты. Қазіргі заманғы технологияның жетістіктерін іс жүзінде қолдану көбінесе өндіріс уақытын қысқартып қана қоймай, процесте артық элементтерді жою арқылы шығындарды азайтуға көмектеседі. Қызмет көрсету саласы үшін бұл өндірістің тиімділігін арттырудың ең қолжетімді және жиі мүмкін болатын тәсілі болғандықтан, ішкі процестерді автоматтандыру осы саланың кәсіпорындары үшін атап айтқанда сату өндірісімен айналысатын компаниялар үшін ерекше қызығушылық тудырды. Негізгі міндеті оңтайлы бағыттар бойынша бір нүктеден екінші нүктеге ұйымдастыру болып табылатын бұл сала, мұнда оптималдылық қашықтық, уақыт және материалдық шығындар сияқты объективті параметрлер бойынша бағаланады, автоматтандыру технологияларын қолдануға жақсы келеді. Атап айтқанда, тапсырыстарды сұраныс арқылы жедел автоматтандыруға болады.

Бұл бітіру жұмысының мақсаты өнім сатылымымен айналысатын түрлі компаниялардың тапсырыс жүйесін автоматтандыру үшін мобильді қосымша әзірлеу болып табылады. Осы мақсатқа жету үшін жұмыс барысында келесі міндеттер дәйекті түрде шешілуі керек:

- Пәндік салаға және қолданыстағы шешімдерге талдау жасау;
- Мобильді қосымшаның тұжырымдамасын жасау;
- Мобильді қосымшаның функционалын анықтау;
- Мобильді қосымшада қолданылатын технологияларды анықтау;
- Мобильді қосымшаны әзірлеу.

Жұмыстың нәтижесінде өндіріске енгізуге дайын мобильді қосымша әзірлену.

Жұмыс негізінен мынадай маңыздылықты білдіреді: мұндай мобильді қосымша тапсырыстарды қабылдау және өңдеу, жөнелту процесінде адами қызметтерді азайту арқылы өндіріс жұмысын оңтайлауға және автоматтандыруға мүмкіндік береді. Адам, субъективті жүйе ретінде барлық процесс үшін ықтимал қауіп төндіреді. Ал автоматтандырылған мобильді қосымша жалпы ақаулардың ықтималдылығын азайтады және адами факторға байланысты ақауды жояды. Аталған дипломдық жұмыстың келесі артышылығы платформаны таңдаумен тікелей байланысты: ұтқырлық ақпаратты әлемнің кез-келген нүктесінен қол жетімді етеді, бұл тапсырыс бойынша ақпаратты жедел жіберуге және алға мүмкіндік береді. Онлайн түрде тапсырыс беру жүйесін автоматтандырудың тағы бір айқын артықшылығы-жұмыс жылдамдығын

арттыру, бұл адамның процеске қатысуын азайту арқылы сатушының еңбек уақыты мен өнімділігіе оң әсер етеді. Өндіріс процесіне автоматтандыру құралын енгізу – бұл кәсіпорынның бәсекеге қабілеттілігін арттыру және оның жұмыс жылдамдыған арттыру, нәтижесінде өндіріс кірісінің өсуі.

Бірінші тарауда онлайн дүкендер саласына жататын кәсіптердің құрылымдылық ерекшеліктері қысқыртылады, қолданыстағы шешімдерге талдау жасалады және мобильді қосымшаның жалпы тұжырымдамасы анықталады. Екінші тарауда қосымшаның функционалдылығы және оны жүзеге асыруды қолданылатын технологиялар анықталады. Үшінші тарауды қосымшаның пайдаланушы интерфейсін дамытуға арналған. Төртінші тарау серверлік сценарийлер мен мобильді қосымшаларды тікелей дамытуға арналған.

1. ТАПСЫРЫСТАРДЫ АВТОМАТТАНДЫРУ ПРОЦЕССИ БОЙЫНША ТӘЖІРБИЕЛЕРГЕ ШОЛУ

1.1. Қазақстандағы тәжірбиелер

Клиенттердің тапсырыстарын басқару жүйесінің міндеттерінің бірі – тапсырыстарға тез және тиімді қызмет көрсету, тауарлар ағынын бақылау, бухгалтерлік және қоймалық есепті толтыру, әр кезеңдегі профестерді оңтайландыру. Тапсырыс беру процесінің басқарылуын автоматтандыру сатылымды оңтайландыруға мүмкіндік береді. Процессті автоматтандыру арқылы сатылым жүйесін мамандандырылған бағдарламалық жасақтамаға тапсыруға және уақыт пен адами шығындардың көп бөлігін үнемдеуге мүмкіндік бар. Тағы бір пайдасы клиент өз тапсырсын бақылап отыра алады. Бұл жүйе B2B және B2C сауда сегменттерінде де қолдануға ыңғайлы.

Қазіргі таңда қызмет көрсету нарығында клиенттің таңдауы бәсекелестіктің туындауына және тапсырыстарды орындау процессінің оңтауландырылуына әкеліп отыр. Сол себептен сапа, тапсырыстарды жылдамдату, шығындарды азайту секілді көрсеткіштер бизнестің дамып, бәсекеге қабілетті болуына әсер етеді. Тапсырыстарды басқару мен автоматтандыру бизнес–процесстеріне тікелей әсер ететіндіктен, жұмыс процестерін дұрыс ұйымдастыру, пайдалы автоматтандырылған бағдарламалық жасақтаманы құру сапалық көрсеткіштерді арттыруы мүмкін.

Сонымен, тапсырыс беру процессінің өмірлік циклі клиент тапсырыс бергеннен басталып, клиент тапсырысын қолына алғаннан кейін аяқталады. Ал тапсырыстардың автоматтандырылған жүйесі осы өмірлік циклды бірыңғай цифрлық ортада бүкіл процесті автоматты түрде бақылауға және үйлестіруге мүмкіндік береді. Бұл белгілі бір уақыт режимінде тапсырыстың түсуін, қорлардың бөлінуін, тапсырыстың орындалуын және сатудан кейінгі қолдануды бақылауға мүмкіндік береді.

CRM(басқару жүйесінің клиентпен қарым қатынасы) - процестерді автоматтандыру көмегімен сатып алушымен тиімді диалог құруға, жұмыста қателіктер жібермеуге және соңында сатушыға көбірек сатуға көмектеседі. CRM-нің басты көрсеткіштерінің бірі ол – ашықтық. Ол кәсіпорынға қажетті барлық ақпараттар мен көрсеткіштерді көрсету және клиентке қажетті тапсырысының қозғалысын көрсетумен айналысады [1].

Көбінесе компаниялар экономикалық пайданы ойлай отырып, тапсырыстарды бақылау деңгейі мен ашықтыған төмендететін жабдықтау қызметтерін немесе компоненттерді, буып-түю қызметтерін ұсынатын үшінші тарап ұйымдарын тартады. Бұл автоматтандырылған процестер көбінесе шығындарға, қателіктерге және сәйкесінше шығындарға әкеледі. CRM қолмен жасалатын процестерді автоматтандыруға, қателерді азайтуға, шығындарды бақылауға арналған. Сонымен қатар автоматтандырылған үшінші тарапты тарту мекеменің беделіне және клиенттердің имиджін қабылдауына кері әсері тиеді. Клиенттерге интернет арқылы бірыңғай көп арналы ақпараттық ортадан

тапсырыс беру оңай және онда тапсырыстың кез-келген уақытта орындалу барысын бақылай алады немесе туындаған сұрақтарға оператормен байланысу арқылы, кері байланыстан жауап ала алады.

Клиентпен болған барлық байланыс нүктелері клиенттерді ұстап қалу пайызын жоғарылататын, жалпы кіріс табысының өсу қарқынына оң әсер етеді.

Тапсырысты автоматтандырудың басқару жүйесі мынандай критерийлерді қамтамасыз етеді:

- Қызметкерлердің тапсырмаларды орындау уақыты мен өнімділігін бақылау;
- Тапсырыстардың жеткізу бағытының ашықтығы;
- Мекеме бөлімдері арасындағы өзара тиімді іс-қимыл;
- Тапсырысты орындау кезеңдерінде барлық құжаттарды автоматтандырылған түрінде өңдеуден өткізу;
- Тапсырыстардағы өзгерту мен жоюларды дәл уақытында, автоматты түрде басқарып отыру;
- Клиенттер үшін тапсырыс күйін тапсырыстың жеткізілу маршрутын қадағалау мүмкіндігі.

Тапсырысты автоматтандырылған басқару жүйесінің арқасында жеткізу тапсырыс үшін өте ыңғайлы болады, қателер саны азаяды, жалпы жұмыс тиімділігін арттырады. Осындай жүйелердің мобильді нұсқасы бар, оның IOS және Android жүйелерінде жұмыс істейтін протативті құрылғыларда дәл осылай орналастыруға болады.

Бүгінгі таңда із жүзінде кез-келген, тіпті кішігірім компанияда, әр түрлі мекемелердің сату бөлімдері CRM жүйелерінде жұмыс істейді. Сату бөліміндегі тапсырысты жеткізу саласын автоматтандыру оның жоқтығынан үнмдеуден гөрі көп пайда әкеледі, бұл айқын факт. Кәсіпкер әрқашан осы жүйенің арқасында бос уақытты не үшін жұмсау керетігін түсінеді. Өйткені, бұл ер адамның өмірінде, әсіресе бизнес ортада баға жетпес ресурс.

Сандық технологиялар дамыған заманда біздің елімізде мұндай дамулардан қалыс қалған жоқ. Қазақстанда да тапсырысты орындауды автоматтандыруға арналған көптеген ұйымдар мен жеткізу қызметін ұсынатын арнайы мекемелер бар. Соның ішінде көш бастап тұрғаны: «1С:Қазақстан үшін кешенді автоматтандыру» жобасы.

«1С:Қазақстан үшін кешенді автоматтандыру» - коммерциялық ауқымы кең кәсіпорындар үшін тапсырысты басқару мен есепке алуды кешенді автоматтандыру міндеттерін шешетін жоба болып табылады [2].

Жұмыс жасау принципі бойынша ақпараттар бірыңғай базада бір және бірнеше коммерциялық ұйымдардың басқарушылық, есептік және салықтық есептер жүргізіледі. Мұндай жол ақпараттың жалпы массивтерін пайдалану арқылы бухгалтерлік есептердің күрделілігін айтарлықтай төмендетеді. Мекеменің тапсырысы бойынша басқару есебі кез-келген валютада жүргізіледі. Әрбір мекеме үшін регламент қойылып, сол бойынша есептік бөлік қалыптастырылады.

«1С:Қазақстан үшін кешенді автоматтандыру 8» қолданбалы шешімі кәсіпорында тапсырысты басқару мен есепке алудың негізгі міндеттерін қамтитын ақпараттық жүйелерді ұйымдастыруға мүмкіндік береді:

- Сатылымды, тауар қорларын және сатып алуды басқару;
- Клиент пен кәсіпорын арасындағы қарым-қатынасты басқару;
- Бағаны қою мен қалыптастыру;
- Жедел түрде ресурсты жоспарлау – тапсырысты орындағаннан кейін қажеттіліктерді қамтамасыз етуді басқару;
- Қазынашылық – ақша шығындары мен кірістерін және өзара есеп айырысуларды басқару;
- Өндірістегі есеп;
- Негізгі құрал-жабдықтарды басқару;
- Бухгалгерлік, салықтық және регламенттік есептер;
- Жалақыны есеп бойынша есептеу және персоналдарды басқару;
- Басшылар үшін тиімділікті мониторингілеу және талдау.

«1С:Қазақстан үшін кешенді автоматтандыру 8» жобасы жоғары икемділікті, масштабталуды, өнімділікті және қолданбалы шешімдердің эргономикасын қамтамасыз ететін «1С:Кәсіпорын 8»технологиялық платформасында құрылған.

Қолданбалы шешім құралдарын пайдалану ұйым ішінде де, сыртқы ортада да (клиенттер, жеткізушілер және бәсекелестер) бөлімшелердің үйлесімді жұмысын қамтамасыз етуге мүмкіндік береді [3].

Сонымен «1С:Қазақстан үшін кешенді автоматтандыру 8»бағдарламасы сауда-өндірістік кәсіпорындар үшін мінсіз шешім. Ұсынылған құрал менеджерге қажетті шешімдер қабылдау үшін нақты ақпарат алуға мүмкіндік береді.

Сауда кәсіпорындарына «1С:Қазақстан үшін сауданы басқару» базасында бірыңғай шешімді ұсынуға болады. Бағдарлама жеткізілімдерді, қоймаларды, сату мен бағаны басқаруға, бірден бірнеше қоймаларда тауарлардың есебін жүргізуге көмектеседі. Жоспарлау сен сатыа алу және сату мүмкіндігін ұсынады. Ол екі нұсқада келеді: шағын және ірі бизнес үшін.

Бөлшек сауда дүкендері мен сауда желілерінің жұмысын «1С:Қазақстан үшін бөлшек сауда» автоматтандыруға мүмкіндік береді. Тауарларды қабылдаудан бастап сатуға дейінгі барлық қозғалыстар тіркелген. Сауда жабдықтарымен жұмыс жүргізілуде.

Қойманы автоматтандыру шығындарды азайтуға және оның жұмысын оңтайландыруға мүмкіндік береді. Қызметкерлерге жүктеме азаяды, өйткені енді күнделікті процесстерді орындаудың қажеті жоқ. Сонымен қатар, жаңа технологияларды енгізу жабдық пен бағдарламалық жасақтамаға үлкен шығындарды қажет етпейді. Ал модернизацияға салынған қаражат қызмет көрсету шығындарын азайту арқылы тез өтіледі [4].

«1С:Қазақстан үшін кешенді автоматтандыру 8» және «1С:Кәсіпорын 8» технологиялық платформаларының жаңа нұсқасында әзірленеді, ол:

- Жүйенің жоғары сенімділігін, өнімділігін және масштабталуын қамтамасыз ету;
- Жүйемен жұмысты интернет арқылы, жұқа клиент немесе веб-клиент режимінде (әдеттегі интернет-барузер арқылы), оның ішінді «бұлтты» режимінде ұйымдастыру;
- IOS немесе Android-пен жұмыс істейтін планшеттер мен смартфондарды пайдалана отырып, мобильді жұмыс орындарын құру;
- Пайдаланушының рөлін, оның кіру құқықтарын және жеке параметрлерін ескере отырып, белгілі бір пайдаланушы немесе сатушы үшін тобы үшін интерфейсті құрастыру.

1.2. Әлемдік тәжірбиелер

Сату тапсырыстарын автоматтандыру әлемдік деңгейде дамыған. Автоматтандырылған басқару әдетте шет елдерде клиенттерге қызмет көрсету, сату, өндіріс, логистика, дебиторлық қаражатты және қаржы секілді түрлі тараптар мен бөлімдерден тұрады. Қосымша бөлімдері ретінде мекеме ресурстарын жоспарлау(ERM), клиенттермен қарым-қатынасты басқару(CRM), бухгалтерлік есеп-қисаптар және бизнес-процесстерді басқару туралы шешімдер кіруі мүмкін. Сонымен қатар, әт түрлі онлайн порталдардан тапсырыс алып, оны өңдеу үшін тапсырыстарды ERP жүйесінде құрады. Осылайша көптеген күрделі процестерден өтетін тапсырысты жеткізу процесі автоматтандарған пайдалы болып келеді [5].

Қолмен тапсырысты жеткізу процедураларына сүйенетін компаниялар мен ұйымдар APQC ашық стандарттар бойынша шет елдік салыстыру бойынша автоматтандырылған тәсілге көшкен мекемелерге қарағанда әлдеқайда шығындарға көп жолғады, нақтылай есептейтін болса сегіз есе жоғары шығындайды.

Әлемдік тәжірбиедегі жасанды интеллект (AI) кіретін роботтандырылған процесстерді автоматтандыру(RPA) бизнеске тиімді көмек беріп, адами ақаулардың алдын алып, операциялық шығынды арттыратын қолмен, қайталанатын тапсырмалардың алдын алады. Бұл жаңа технология қолданушылар порталын немесе электрондық пошта жәшіктеріндегі тапсырыстарға бақылау жасайды, тапсырысты өңдеу жылдамдығы жоғары және алдын-ала анықталған бизнес қағидаларына сәйкес ақпаратпен хабарлап, ақпаратты растай алады.

RPA боты қатесіз құрылымдалған деректерді қабылдап, оны электронды түрде тиісті тұлғаларға автоматты түрде таратып немесе басқару ұйымдарына жібереді және мекеменің ERP қосымшасына жүктейді.

Robotic Process Automation(RPA) – бағдарламалық роботтарды (software robots) және жасанды интеллектті пайдалануға негізделген бизнес-процесстерді автоматтандыру технологиясының бір түрі. Бағдарламалық робот жүйесінің интерфейсімен өзара әрекеттесу арқылы адамның әрекетін ойнатады.

RPA артықшылықтары:

- Адамзатты күнделікті және іш-пыстырарлық міндеттерден құтқарады;
- Бизнесің өтімділігіне тікелей оң әсер ете отырып, операциялық шығыстарды қысқарта отырып, адами ресурстарды босатуына мүмкіндік береді;
- Робот тәулік бойы жұмыс істейді, шаршамайды, демалысқа кетпейді;
- Роботтың көңіл күйі болжамды, барлық қадамдары бағдарланған;
- Әдетте тезірек жұмыс істейді және тапсырмаларды қатесіз орындайды;
- Өз әрекетінің толық журналын оңай қамтамасыз етеді және операциялық өнімділікті икемді басқаруға мүмкіндік береді.

RPA және когнитивті автоматтандырумен тапсырыс енгізуді автоматтандыру бизнесіне бес жолмен жақсартады:

- Операциялық шығандарды төмендетеді;
- Тұтынушының тәжірбиесін арттырады;
- Кіріс көзін көбейтеді;
- Құндылығы төмен артық әрекеттерді азайтады;
- Ұйымдық ептілік пен шешім қабылдауды жақсартады.

1.3. Тәжірбиені онлайн сатылым процесін автоматтандырумен байланыстыру

Мобильді қосымшаны әзірлеу барысында бірінші кезекте зерттеуді талап етеді. Бұл кезеңде нақты саланы қарастырумен, бәсекелестікті ескеру мен мобильді қосымшаның аналогтарын ескере отырып мобильді қосымшаның жоспарын жасау. Бітіру жұмысындағы мен алған сала онлайн сатумен айналысатын дүкендер желісі топтастырылған мобильді қосымша ретінде қарастырсақ болады.

Мобильді қосымша пайдаланушыларға онлайн, өз парақшасында және мобильді қосымша арқылы сатып алуға, тапсырыс жасауға, тапсырысты рәсімдеуге және жеткізу тәсілін таңдауға, тапсырысты төлеуге мүмкіндік береді. Бұл ретте тауарларды сату қашықтақтан жүзеге асырылады және екі тараптың да уақытын үнемдейді.

Тауар сатылымын арттыру, уақыт үнемділігін ескеру және клиенттер тарапсынан сенімділікке ие болу үшін адами ақаулардың алдын ала отырып мобильді қосымшаны автоматтандырылған басқару процессіне қосқан дұрыс.

1.4. Мобильді қосымшаны әзірлеу кезеңдері



1.1.-сурет.Мобильді қосымша әзірлеу кезеңдері

Мобильді қосымшаны әзірлеу 1.1 – суретте көрсетілгендей 6 кезеңнен тұрады:

- Стратегия. Мобильді қосымшаны әзірлеудің бірінші кезеңі идеяны дамыту мақсатында бағдалама үшін стратегия құру. Қосымшалардың мақсаттары басқалардан өзгеше болуы мүмкін. Даму барысында осыны ескеріп мобильді қосымшаға тән, жеке ұтқырлық стратегиясын ойластырған жөн. Бұл кезеңде анықтау керек:

- Мобильді қосымшаның тұтынушыларын;
- Бәсекелестікті;
- Мобильді қосымша мақсаттары мен міндеттері;
- Мобильді қосымша үшін платформа таңдау.

Бұл кезеңдегі анықтау керек мақсаттарға сүйене отырып жүргізілген зерттеулер бойынша:

- Мобильді қосымша пайдаланушылары 16–45 жас аралығындағы мобильді қосымшалардан хабары бар кез–келген қолданушылар болуы мүмкін;
- Бәсекелестікті зерттегенде тұтынушы ретінде, онлайн сатылыммен айналысатын бәсекелес мобильді қосымшаларды зерттеп, тиімді тұстарын ескеріп және ұнамаған тұстарын да жазып алынып, болашақта ескерілетін болды;
- Мобильді қосымша мақсаттары: тұтынушысы мен сатушыны тікелей, қашықтықтан байланыстыру, қолданушылар уақытын үнемдеу, атылымды арттыру, онлайн тапсырыстар рәсімдеу, сатушы салатын өнімдердің тұтынушыға тәулік бойы қолжетімділігі және жүйеге енген өнімдерге

кез-келген уақытта түзетулер енгізу.

- Қосымша жасалатын платформа Android Studio.
- Аналитика және жоспар. Осы кезде мобильді қосымшаның идеясы қалыптаса бастайды және нақты жобаға айналады. Талдау және жоспарлау пайдалану жағдайларын анықтаудан және егжей-тегжейлі функционалдық талаптарды жинаудан басталады. Ұялы байланыс қызметтеріндегі мобильді қосымша үшін аналитиканың келесі түрлері шешуші болып табылады:

- Маркетинг – дамыту арналарын талдау және қайсысы тиімді екенін анықтау;

- Өнім – қосымшадағы клиенттердің мінез құлқын анықтау. Өнімділікті арттыру және конверсияны арттыру үшін тағы не ойлап керек деген сұраққа жауап табу;

- Қолданбалы дүкендер – сіздің мобильді қосымшаңыздың ұқсас платформалардың пайдаланушылар арасында қаншалықты танымал екендігін, жүктелгендер саны қаншау екенін бақылау.

- UI / UX мобильді қосымшасының дизайны. Бағдарлама дизайнының мақсаты – мінсіз сыртқы келбеті бар және жеңіл пайдаланушы интерфейсімен қамтамасыз ету. Ең алдымен визуалды түрде пайдаланушылардың көңілінен шығу. Мобильді қосымшаның сәтті қолданысқа енуі тұтынушылардың оның барлық функциялар қаншалықты игеріп, қолданылатынына байланысты. UI / UX мобильді қосымшасының дизайнының мақсаты –бағдарламаны интерактивті, интуитивті және ыңғайлы ету арқылы керемет пайдаланушы интерфейсін құру. Жақсартылған UI дизайны іске асырудың алғашқы кезеңінде көмектеседі, бірақ пайдаланушының қызығушылығын сақтау үшін құрастырушы бағдарламасында интуитивті пайдаланушы интерфейсі болуы керек.

- Мобильді қосымшаны әзірлеу. Жоспарлау мобильді қосымшаны әзірлеу процессінде осы кезеңнің ажырамас бөлігі болып табылады. Бағдарламалауды бастамас бұрын немесе әзірлемес бұрын қажет:

- Техникалық архитектураны таңдау;
- Технология стегін таңдау;
- Даму кезеңдерін талдау.

Мобильді қосымшаның типтік жобасы үш компоненттен тұрады:

- Серверлік технология;
- Қолданбалы бағдарламалау интерфейсі;
- Мобильді қосымшаның интерфейсі.

Back-End/серверлік технология. Бұл бөлімде мобильді қосымшаның функцияларына арналған қажетті мәліметтер базасы мен серверлік нысандар бар. Мұнда визуалды түрде клиентке көрінбегенімен, барлық процестерді іске асыру жүргізіледі[6].

Қолданбалы бағдарламалау интерфейсі(API) – бұл бағдарлама мен ішкі мәліметтер базасы арасындағыбайланысты орнату әдісі. Осы жылдар ішінде «API» ұғымы көбінесе қолданбаға қосылудың жалпы интерфейсін сипаттады

[7]. Алайда, соңғы уақытта қазіргі API оларды өте құнды және пайдалы ететін кейбір сипаттамаларға ие болды:

- Қазіргі заманғы API-лер стандарттарға сәйкес келеді (әдетте HTTP және REST), олар әзірлеушілерге ыңғайлы, оңай қол жетімді және кең мағынада түсінікті;
- Олар кодқа қарағанда өнімдерге көбірек қарайды. Олар белгілі бір аудиторияда (мысалы, мобильді әзірлеушілер) пайдалануға арналған, олар құжатталған және пайдаланушылар оның қызметі мен өмірлік цикліне қатысты белгілі бір үміттерге ие болатындай нұсқалары бар;
- Олар әлдеқайда стандартталған болғандықтан, олар қауіпсіздік пен басқаруға қатысты әлдеқайда қатаң тәртіпке ие, сонымен қатар өнімділік пен масштабтауды қамтамасыз ету үшін оларды бақылау және басқаруды да қадағалайды;
- Өндірілген бағдарламалық жасақтаманың кез-келген бөлігі сияқты, қазіргі заманғы API-де жобалау, тестілеу, құрастыру, басқару және нұсқаларды басқаруды қамтитын бағдарламалық жасақтаманы әзірлеудің өзіндік өмірлік циклі (SDLC) бар. Сонымен қатар, қазіргі заманғы API интерфейстерін пайдалану және басқару үшін жақсы құжатталған.

Мобильді қосымшаның интерфейсі/Front-End – бұл пайдаланушы көретін, мобильді қосымшаның жалпы көрінісі.

Көптеген жағдайларда мобильді қосымшалар ақпаратты басқару үшін API және артқы жағын пайдаланатын интерактивті интерфейстерден тұрады. Кейбір жағдайларда, қолданба пайдаланушыларға Интернетсіз жұмыс істеуге мүмкіндік беруі керек болса, қолданба жергілікті деректер қоймасын қолдана алады. Ол кезде кез-келген дерлік веб-бағдарламалау тілі мен мәліметтер базасын қолдана алады. Жергілікті мобильді қосымшалар үшін әр мобильді ОЖ платформасы үшін қажет технологиялық стекті таңдау керек. Android қосымшалары негізінен Java немесе Котлин көмегімен жасалады.

Мобильді қосымшаларды құруға арналған бірнеше бағдарламалау тілі мен технологиялық стек бар – мобильді қосымшаға жақсы сәйкес келетін технологиялық стекті таңдаудың кілті. Мобильді технологиялар мобильді платформалардың жаңа нұсқаларының пайда болуымен тезірек дамып келеді.

Сонымен қатар, бірнеше ай сайын жаңа мобильді құрылғылар шығарылады. Платформалар мен құрылғылар тез өзгеретіндіктен, уақыт пен бюджет аясында мобильді қосымшаларды құру үшін икемділік қажет. Егер нарыққа шығу уақыты басымдыққа ие болса, дамуға икемді тәсіл қолдану қажет. Бұл тәсіл толық функционалдығы бар бағдарламалық жасақтаманың жиі шығарылуын қолдайды.

Дамудың икемді жоспары шеңберіндегі даму кезеңдерін анықтау мобильді қосымшаны итерацияда дамытуды қолдайды. Әрбір даму кезеңі аяқталғаннан кейін ол тексеру үшін қосымшаларды тестілеу тобына беріледі.

• Тестілеуден өткізу. Мобильді қосымшаларды әзірлеу барысында қосымша сапасын мұқият, толық тексеру қосымшаларды тұрақты, ыңғайлы және қауіпсіз етеді. Қосымшаның сапасын тексеру үшін алдымен қосымшаны тестілеудің жан-жақты барлық аспектілеріне қатысты сынау жағдайларынан өткізу қажет. Тест жағдайлары тестілеу қадамдарын орындауға, бағдарламалық жасақтаманың сапасын бағалау үшін тестілеу нәтижелерін жазуға және қайта тестілеуге арналған түзетулерді бақылауға арналған. Бағдарламаның функционалды талаптары мен мақсаттарымен танысу нақты сынақ мысалдарын жасауға көмектеседі. Ұтқырлықтың сапалы шешімін қамтамасыз ету үшін сіздің қосымшаңыз келесі тестілеу әдістерінен өтуі керек:

- Пайдаланушылардың тәжірбиесін тестілеу;
- Функционалдық тестілеу;
- Өнімділікті тексеру;
- Құрылғылар мен платформаларды тексеру.

• Қолданысқа енгізу. Жеке мобильді қосымшаны шығару мобильді қосымшаны қолданбалар дүкендеріне, iOS қосымшалары үшін Apple App Store және Android қосымшалары үшін Google Play дүкендеріне жіберуді талап етеді. Алайда, мобильді қосымшаны іске қоспас бұрын Apple App Store және Google Play Store-да әзірлеуші есептік жазбасы қажет болады.

Android қосымшаларында тексеру процесі жоқ және олар жіберілгеннен кейін бірнеше сағат ішінде App Store-да қол жетімді болады.

Бағдарлама қолданбалар дүкенінде қолжетімді болғаннан кейін, оны мобильді аналитикалық платформаларын басқарады және бағдарламаның сәттілігін өлшеу үшін негізгі көрсеткіштерді бақылайды. Пайдаланушылар хабарланған сәтсіздің туралы есептерді немесе басқа мәселелерді жиі тексеру қажет. Пайдаланушыларды мобильді қосымшаға бағдарлама бойынша пікірлер мен ұсыныстар беруге шақыру қажет.

Соңғы пайдаланушыларға жедел қолдау көрсету және жиі жақсартылатын қосымшаларды түзету пайдаланушының белсенділігін сақтау үшін өте маңызды болады.

2. МОБИЛЬДІ ҚОСЫМШАНЫ ӘЗІРЛЕУ

2.1. Мобильді қосымшаға қойылатын талаптарды әзірлеу

Онлайн сату мобильді қосымша құрылымында басқару блогын қоспағанда екі блоктан тұрады:

- Клиенттік аймақ;
- Сатушылар аймағы.

Бұл бөлімде автоматтандыруды қолдану өте ыңғайлы, өйткені оның процессі стандартты формаларды толтырумен және рәсімдеумен тығыз байланысты, сондықтан құжаттаманың бір бөлігі компант қолданатын модель бойынша автоматты түрде жасалуы мүмкін. Сонымен қатар, бұл фаза шағын функционалды қосалқы станцияларға бөлінген динамикалық процесс болып табылады, бұл процесс динамикасын бақылау жүйесін енгізуге және тапсырыс берушілерге немесе процеске қатысушыларға белгілі бір тәртіппен орындалатын жұмыстардың жай-күйі туралы хабарлауға мүмкіндік береді.

Жоспарлау мен әзірлеуге дайындықтың келесі кезеңі – бар шешімдер іздеу мен қарау. Айта кету керек, бүгінде тауарлар мен тауарларға тапсырыс беруге арналған мобильді қосымшалар өзін-өзі шешудің немесе қолданыстағы интернет-шешімнің қосымша құралы ретінде танымал бола бастады. Бұл, әрине, мобильді құрылғылардың танымалдылығын сөзсіз өсуіне байланысты. Бұл топтың қосымшаларына ресми және бейресми клиенттер кіреді, ал соңғларының болуы мұндай қосымшаларға қажеттілік бар екенін көрсетеді. Барлық қосымшалардың ортақ ерекшелігі – олардың барлығы белгілі бір компания немесе қызмет провайдері аясында жүзеге асырылады.

Мобильді қосымшаның негізгі мақсаты жүк қызметіне сұранысты автоматты түрде алу болып табылады, бұл клиенттер үшін де, қызметкерлер үшін де тапсырыстың орындалу мәртебесін динамикалық бақылауға мүмкіндік береді. Сондықтан, жоғарыда айтылғандарды ескере отырып, өтінімді екі бөлікке бөлген жөн: тапсырыс беруші және өндіріс

Процестің екі жағы да әртүрлі қызығушылықтарға ие және әр түрлі рөл атқарады, қосымшаның функционалдылығы әр түрлі. Қосымшаны пішімдегеннен кейін клиент тек іске асыру күйін бақылауға мүдделі. Атап айтқанда, тасымалдаушылар көліктің барлық түрлері мен жеткізу процессіне қатысатын қосалқы жеткізушілер туралы хабардар болуы тиіс.

Бағдарламалар барлық көшірмелерге қол жеткізе алатын қашықтықтағы дерекқорды қажет етеді. Сонымен қатар, мобильді қосымшаның жұмысын ұйымдастыру үшін мобильді қосымша мен мәліметтер базасы арасында делдал ретінде әрекет ететін, дерекқордан сұралатын деректерді мобильді қосымша қолдана алатын пішінге айналдыратын фон қажет.

Қосымша тапсырма ретінде пайдалану ыңғайлылығын арттыру үшін веб-интерфейсті жасауға болады.

Қосымшаның және оның компоненттерінің қосымша функционалдылық мүмкіндіктері келесі тарауда қарастырылады.

Мобильді қосымшаны пайдалану артықшылықтары:

- Үйден шығып, басқа сатып алушылармен кездесудің қажеті жоқ. Бұл пайдаланушыда уақыт болмаған кезде немесе тұмау маусымы басталған кезде және тағы бір қоғамдық ортаға шыққысы келмеген кезде үлкен плюс;
- Бірнеше минут ішінде ондаған дүкендерді «аралауға» болады. Дүкенге бір сағат бойы бару және соңында пайдаланушыға қажет нәрсе жоқ екенін білу секілді оқиғалармен жолықпайды. Мобильді қосымшада бірден басқа сатушылар парақшасына кіре алады немесе алдын-ала тапсырыс бере алады;
- Ақшаны үнемдеу. Мобильді қосымшада кеңсені жалға алу үшін ақы төлеудің қажеті жоқ, сонымен қатар көптеген кеңесшілер мен кассирлерді жалдамаудың арқасында тауарлар әлдеқайда арзан сатылады;
- Кең ассортимент Дәл қазір дүкен клиентке қажет нәрсені таппаса да, сол тауарды дүкен әкімшілігі арнайы тапсырыс бере алады;
- Клиенттердің пікірлері. Әдетте, өнімнің сипаттамасында қолданушыларға дұрыс таңдау жасауға көмектесетін басқа сатып алушылардың пікірлерін бірден оқи алады.

Мобильді қосымшаларды жасау үшін нақты қай бағытта қозғалатынымызды түсіну үшін әрі қарай, қосымшаға қойылатын талаптардың тізімін жасау керек. Құжат түріндегі талаптардың сипаттамалары әзірлеушіге жасалуы керек мобильді қосымша туралы нақты түсінік береді.

Техникалық талаптар:

- Ойластырылған дизайнға сай келу;
- Қателіксіз жұмыс жасау;
- Android Studio 4.3 версиясын қолдану;
- Көрнекті кідіріссіз жұмыс жасау.

Функционалдық талаптар.

Қолданушы мобильді қосымшаға қандай да бір авторизация арқылы кіреді, бағдарлама қолданушы енгізген деректерді жүктеп, дұрыс болған жағдайда автоматты түрде жаңалықтар бөлімінен көрсетілуі керек.

Басқа бөлімге өту үшін сол жақ жоғарғы жағында дизайн бойынша батырма және сол жақтағы экранда саусақ қимылыменашу мүмкіндігі болуы керек.

Ашылған мәзірде мәзір тақырыбында қолданушы фотосурет және электрондық поштасы секілді жеке бас ақпараттары болуы тиіс, ал төменгі бөлігінде бөлімдер тізімі, бөлімдер атауларына сәйкес суреттер және бөлім атаулары болуы керек.

Жаңалықтар бөлімінде айналдыру мүмкіндігі болуы керек. Жүктелген жаңа тауарлар тізімдері жаңалықтар бөлімінен шығып, ол жаңалық аяқталса шектеулі жаңалықтар саны жүктелуі керек, егер жаңалықтар жоқ болса, жүктеу ары қарай жүргізілмеу керек.

Менің сатып алуларым бөлімінде пайдаланушы төлем жасаған тауарлар туралы ақпарат орналасуы қажет. Төлем жасалынған тауар жеткізілуіне қарай әр-бір күйді көрсетіп тұру керек (жеткізуде, жеткізілді немесе тауарлар қоймада жиналуды).

Іздеу батырмасында тауарлардың индексімен, атымен немесе категориясына сайы іздеу мүмкіндігі болуы керек.

Категориялар бөлімінде тауарларды типіне қарай ажыратып, іздеуге ыңғайлы болатындай барлық тауарларды орналастыру қажет.

Баптаулар арқылы жеке парақшасындағы мәліметтерді кез-келген уақытта өзгертіп, қайта баптай алады. Сонымен қатар бұл бөлімде пайдаланушылар атынан қауіпсіздікті арттыру мақсатында арнайы сұрақтар қойылып, сол сұрақтарға жауап бергенде ғана құпия пароль қолданушы тіркеген телефон номеріне жіберілетін функция болуы тиіс.

Пайдаланушылардан бөлек бізде сатушылар немесе админдер парақшасы болуы керек. Сатушылар парақшасы пайдаланушылармен тікелей қарым қатынаста болады. Сатушыларда да басты парақшада бір ғана функция болады. Ол категориялар таңдап, сол бойынша тауарды енгізу. Тауарларды енгізіп болғаннан кейін арнайы тауарлар туралы ақпаратты өзгертуге болатын батырма пайда болуы тиіс. Басты бетте тауарларға келіп түскен тапсырыс туралы ақпараттар орналасатын болады, сол бойынша тауарларды тапсырыстар бойынша автоматтардырып отыруы керек.

Жоғарыда аталған талаптар бойынша, біз мобильді қосымшамыз үшін талдау жүргізіп, осы талдаулар бойынша тиімді, тез және қажетсіз қаржылық және адами шығындарсыз қосымшаны жан-жақты қамтамасыз етуге көмектеседі.

Бағдарламаны оңай кеңейтуге, қолдауға, оңай оқытылатын бағдарламалық кодқа салуға болады, бағдарламалық кодты қолданған адам түсініп, болашақта кеңейтулер мен қолдаулар жасай алуға мүмкіндік туады.

Даму принциптерін тереңірек түсіну үшін біз қандай бағдарламалық платформаны түсінуіміз керек және қай платформада қосымшаны әзірлейтінімізді түсіну керекпіз. Ол үшін ең алдымен Android операциялық жүйесімен жақынырақ танысуымыз керек.

2014 жылдың екінші тоқсанында сатылған смартфондардың 86%-ында Android операциялық жүйесі орнатылды. Сонымен қатар, 2014 жыл бойы Android құрылғылары бір миллиардтан астам сатылды.

Android платформасына жаңа әзірлеушілерді тартуға күш салуды жалғастыра отырып, Google әртүрлі жарыстар өткізе бастады. Олардың біріншісі, Android Developer Challenge (ADC) деп аталады, 2008 жылы өтті. Жоба жеңімпаздарына ақшалай қомақты сыйлықтар уәде етілді [8]. ADC келесі жылы да өткізіліп, оған көптеген үміткерлер қатысты. 2010 жылы ADC жүргізілмеген, оны Android-тің қажетті әзірлеушілер базасын жинағандығымен түсіндіруге болады, сондықтан жаңа мүшелерді тарту қажеттілігі жойылды. Сонымен қатар Google 2010 жылдың басында құрылғыларды тарту бағдарламасын бастады.

2.2. Қолданушы интерфейсіне талдау

UI – бұл пайдаланушы интерфейсінің аббревиатурасы. UI дизайны адам мен компьютердің өзара әрекеттесуінің жалпы дизайнын, жұмыс логикасын және әдемі бағдарламалық интерфейссті білдіреді. Пайдаланушы интерфейсінің жақсы дизайны бағдарламалық жасақтаманы жеке және талғампаз етіп қана қоймайды, сонымен қатар бағдарламалық жасақтаманың жұмысын ыңғайлы, қарапайым, еркін етеді және бағдарламалық жасақтаманың орналасуы мен сипаттамаларын толық көрсетеді.

Адам мен машинаның өзара әрекеттесуінің маңызды аспектілерінің бірі-біз интерфейс деп атаймыз.

Пайдаланушы интерфейсінің дизайны экран өнімдерінің маңызды бөлігі болып табылады. Әдетте интерфейссті екі деңгейге бөлуге болады: қабылдау (көру, жанасу, есту және т.б.) және эмоциялар. Интерфейс дизайны-бұл үлкен кәсіби білімді қажет ететін жұмыс. Мұнда танымдық психология, дизайн, лингвистика және т.б. маңызды рөл атқарады.

UI дизайны – бағдарламалық өнім сапасының үш негізгі индикаторына әсер ететін фактор болып табылады:

- Функционалдығы;
- Эстетикасы ;
- Өнімділігі.

Функционалдылық – мобильді қосымша жасаушылардың жиі әрі басты назар аударатын фактор болып табылады. Қосымша жасаушылар қосымшаны пайдаланушылардың ыңғайына қарай, өз міндеттерін орындай алатындай етіп, қолданысқа ыңғайлы әрі түсінікті етіп жасауға тырысады. Функционалдылық маңызды, бірақ бұл даму барысында ескерілуі керек жалғыз көрсеткіш емес.

Қосымшаның эстетикалық көрінісі және оны пайдалануға ұсыну әдісі тұтынушы тарапынан визуалды түрде оң пікір қалыптастыруға мүмкіндік береді. Бірақ эстетикалық сипаттамалар өте субъективті және оларды сандық сипаттау әлдеқайда қиын функционалдық талаптар немесе өнімділік көрсеткіштері. Кез-келген қосымшаның эстетикасы көбіне қарапайым таңдаулардан тұрады. Олар:

- Қолданылатын түстердің бір-бірімен байланысы;
- Интерфейс элементтері, олардың мақсаты мен ұсынылған операциялардың мағынасын білдіруі;
- Адам белгілі бір басқару элементтерін қолданғанда не сезінеді және оларды қаншалықты сәтті қолданады.

Өнімділік, сондай-ақ сенімділік бағдарламаны қолдану перспективасына да әсер етеді. Егер бағдарлама жақсы көрінсе, қарапайым және ыңғайлы жұмыс істесе, бірақ, мысалы, экрандарды баяу қозғалтса, үнемі он немесе одан да көп секундқа кешігу болады. Осындай кемшіліктері көрініп, жұмыс өнімділігін арттырмаса пайдаланушылар тарапынан сұраныс азайып, қосымша туралы теріс пікір қалыптасады . Өз кезегінде, қосымшаның тез және тұрақты жұмыс

істеуі оның ең стильді емес дизайнын немесе кейбір екінші функциялардың болмауының орнын өтей алады.

Қолданушының сәтті жұмысын қамтамасыз ету үшін интерфейс дизайнерінен қосымшаны әзірлеудің бүкіл өмірлік циклі бойында жоғарыда аталған факторлар арасындағы тепе-теңдікті сақтау қажет.

Бұл нәтижеге пайдаланушы интерфейсін дамытудың әр кезеңінде интерактивті өзара әрекеттесудің егжей-тегжейін дәйекті және мұқият зерттеу арқылы қол жеткізіледі:

- Жобалау

Функционалдық талаптар:

- Даму мақсатын анықтау;
- Бастапқы талаптар.

Пайдаланушыларды талдау:

- Пайдаланушылардың қажеттіліктерін анықтау;
- Сценарийлерді әзірлеу;
- Сценарийлердің пайдаланушылардың күтулеріне сәйкестігін бағалау.

Тұжырымдамалық жобалау:

- Қосымша әзірленуіндегі процессті жобалау;

Логикалық жобалау:

- Қосымшадағы ақпараттық ағындарды анықтау;

Физикалық дизайн:

- Жоба мен даму құралдары іске асырылатын платформаны таңдау;

- Жүзеге асыру

Прототиптеу–экрандық формалардың қағаз немесе интерактивті пакеттерін жасау, оның дизайнын өзгерту мүмкіндігін ескере отырып, қосымшаны құру.

Қосымшаны жасаушылар көбінесе тақырыптың түстерін компанияның бренд стиліне сәйкес келтіреді. Егер сізге тақырыптың түстерін өзгерту қажет болса, Google-дің түстерді пайдалану туралы материалдық дизайн бойынша ұсыныстары негізгі түс (үш реңктен аспайтын) және екпін түсінен тұратын түстер палитрасын таңдауға кеңес береді.

2.3. Қолданған технология және таңдалу себебі

Android Studio – Android платформасымен жұмыс істеуге арналған интеграцияланған даму ортасы(IDE), 2013 жылғы 16 мамырда Google I/O конференциясында жарияланды.

Бұл IDE 2013 жылдың мамыр айында жарияланған 0.1 нұсқасынан бастап еркін қол жетімді болды, содан кейін 2014 жылдың маусым айында шыққан 0.8 нұсқасынан бастап бета-тестілеу кезеңіне өтті. 1.0-тің алғашқы тұрақты

нұсқасы 2014 жылдың желтоқсанында шығарылды, содан кейін Eclipse үшін Android дамыту құралдары (ADT) плагинін қолдау тоқтатылды [9].

JetBrains компаниясының IntelliJ IDEA бағдарламалық жасақтамасына негізделген Android Studio – Android қосымшаларын әзірлеудің ресми құралы. Бұл даму ортасы Windows және macOS және Linux үшін қол жетімді [10]. 17 мамыр 2017 жылы жыл сайынғы Google I/O конференциясында, Google Android студиясында қолданылатын Котлин тілін қолдауды жариялады, Android платформасы үшін ресми бағдарламалау тілі ретінде, Java және C++ тілдері белгіленді.

Android Studio платформасын пайдалану себептері:

- Жетілдірілген макет редакторы: WYSIWYG, UI компоненттерімен Drag-and-Drop көмегімен жұмыс істеу мүмкіндігі;
- Экран конфигурацияларының бірнеше жерде орналасуын алдын ала қарау мүмкіндігі;
- Қосымшаларды gradle негізінде құрастыру;
- Көптеген .apk файлдардың құрастыру және генерациялау түрлері;
- Кодты қайта жобалау;
- Негізгі макеттер мен Android компоненттерінің үлгілері;
- UI компоненттерін, бірнеше экран конфигурацияларындағы алдын-ала орналасуларды және басқа компоненттерді кодсыз сүйреуге мүмкіндік беретін бай орналасу редакторы;
- Өнімділікті, пайдалану ыңғайлылығын, нұсқалардың үйлесімділігін және басқа мәселелерді бақылауға арналған Lint құралдары [11].

Бұл мобильді қосымша құру бағдарламасы бірнеше бағдарламалау тілдерін, соның ішінде C/C++ және Java тілдерін қолдайды, кіріктірілген эмулятор және барлық шаблондар мен компоненттері бар үлкен кітапханасы бар, бұл қосымшаларды әзірлеу процесін едәуір жеңілдетеді және тездетеді.

Android Studio-да Android – N-нің соңғы нұсқасы үшін мобильді қосымша жасай алады, ал жасалған қосымшаны дереу қателіктерге тексеріп, бағдарламаның барлық элементтерін әртүрлі құралдармен тексеріп, оның жұмысындағы барлық кемшіліктерді алдын-ала анықтауға болады [12]. Мобильді қосымшаны меңгеру қиын болса немесе тек алғашқы қадамдарды ғана жасай алатын болса, қолданушыға Android Studio-ның ресми сайтынан нақты нұқаулықтарды меңгеру мүмкіндігі де қарастырылған [13].

Кіріктірілген эмулятордың арқасында әр түрлі жүйелерде жасалған қосымшалардың өнімділігі мен дұрыстығын тексеруге болады, қажет болған жағдайда оңтайландыруды жүргізуге болады.

Мобильді қосымшаны құру үшін бағдарламалау тілі ретінде Java тілі алынды. Java Sun Microsystems 1995 жылы шығарған есептеу және бағдарламалау платформасы болып табылады [14]. Бұл тілдің ең маңызды технологиялары GPL лицензиясы бойынша талатылады. Ал сатылым маркасының құқығы Oracle корпорациясына тиесілі [15].

2.4. Мобильді қосымша архитектурасы

Мобильді қосымша архитектурасы 2.1 – суретте көрсетілгендей тармақтардан тұрады. Қосымшаға тіркелген тұста қолданушы таңдау жасайды. Таңдауға қарай:

- Пайдаланушы;
- Сатушы.

Әрі қарай 2.1 – сурет арқылы әрбір парақшаларға сай өзіндік процесстер жүріп отыратынын көреміз. Сондай-ақ деректер қорымен тікелей байланысатын әрекеттер де бар. Барлық процесстер сызбада көрсетілген сілтемелер бойынша бір-бірімен әрекеттеседі.



2.1.–сурет. Мобильді қосымша архитектурасы

3. МОБИЛЬДІ ҚОСЫШАҒА ШОЛУ

3.1. Қосымша навигациясы

UI дизайны қолданушылар тарапынан сұранысқа ену мақсатында нарықта көп қолданылатын арнайы шаблондарды қолданады. Жаңа және сәнді үлгілерді түсіну және тану мобильді қосымшаны әзірлеу барысында айтарлықтай прогреске әкеледі және жасалған қосымша заманауи талаптарға сай және бәсекеге қабілеттілікті арттырады.

Дипломдық жұмыстың тақырыбы бойынша мобильді қосымша құрылған болатын. Мобильді қосымша интерфейсіне толық шолу жасайық.

Кез-келген қосымша секілді қосымша идентификациядан басталады. Бұл қосымшаға қол жеткізуді бөлісуге, қолданушының жеке ақпараттарын деректер базасында сақтауға және қолданушы тарапына басқа құрылғылар тарапынан шабуылдар болдырмауға, қауіпсіздікті қамтамасыз етуге мүмкіндік береді.

Мобильді қосымша ашылғаннан кейін 3.1 – суретте көрсетілгендей пайдаланушы алдынан аутентификация беті пайда болады. Пайдаланушы тарапынан таңдау мүмкіндіктері беріледі. Қосымшаға алғаш кіргендер үшін арнайы тіркелу батырмасы болады.



3.1 -сурет. Тіркелу батырмасы

Бұл батырманы басу арқылы қолданушы мобильді қосымшаға кірген кезде 3.2 – суретте көрсетілгендей пайдаланушы енгізетін барлық ақпараттар сұралады. Осы ақпараттарды енгізе салысымен сізде мобильді қосымшаны пайдалану мүмкіндігі пайда болады. Енгізілген ақпараттар мобильді қосымшамен байланысқан деректер базасында сақталады.

3.2 -сурет. Тіркелу шаблоны

Пайдаланушы деректер базасына өзі туралы жеке ақпараттарды енгізгеннен кейін мобильді қосымшаға кіру мүмкіндігін беретін 3.3 – суретте көрсетілгендейбатырма пайда болады.



3.3 -сурет. Қосымшаға кіру бастырмасы

Батырма басылғанда 3.4 – суретте көрсетілгендейбасқа бетке өтеміз.Бұл бетте арнайы шаблон арқылы құрылған қосымшаға кіруге болатын бет ашылады. Осы жердегі мәліметтерді толтыру арқылы қосымшаның басты бетіне жіберіледі.

3.4 -сурет. Қосымшаға кіру шаблоны

Телефон номері пайдаланушыны анықтауға, яғни аутентификацияның негізгі функциясын орындауға мүмкіндік береді.

Пароль рұқсатсыз кірудің алдын алады, яғни ақпараттық қауіпсіздіктің негізгі міндетін шешеді.

Парольді сақтау – сессия идентификаторы cookie-де сақталады. AFAIK, пароль немесе hash сақталмайды. Сеанс қолданушы кірген кезде серверде жасалады. Егер қолданушы «Запомнить» тексеруімен кірсе, сервер сеанс идентификаторы бар cookie файлы жібереді (немесе шифрланған сеанс идентификаторы немесе пайдаланушының сеансын анықтайтын нәрсе) және сеанстың бар-жоғын тексереді, содан кейін сервер қолданушының кірген идентификаторы бар деп анықтайды және сайтқа кіргізеді. Пайдаланушы кіруге арналған құпия сөзін ұмытқан жағдайда арнайы батырманы басу арқылы қайта құпия сөзге сұраныс жіберіледі. Сұраныс жіберілгеннен кейін арнайы бет ашылады. Бұл бетте 3.5 – суретте көрсетілгендейқолданушы алғаш тіркелген кезде жазған мәліметтерге сәйкес сұрақтарға жауап берсе, дұрыстығын базадан тексеріп, жазылған ақпараттар дұрыс болған жағдайда тіркелген номерге жасырын хабарлама жіберіледі.

3.5 -сурет. Қосымшаның құпия паролыне қайта сұраныс жасау шаблону

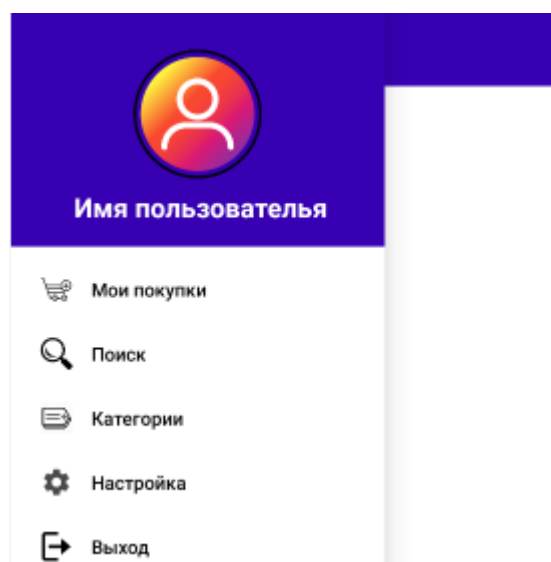
Әрі қарай қосымшаға кіргенде 3.6 – суретте көрсетілгендейсол жақ мәзірді байқаймыз, бұл функция қосымшадағы басты функция ретінде жұмыс жасайды. Оларды жасау үшін макетке Toolbar деп аталатын функция қосылады.



3.6 -сурет. Қолданушы парақшасының басты беті

Toolbar – әрекет тақталары деп те аталады, бұл біздің қосымшаның қызметіндегі маңызды дизайн элементтерінің бірі, өйткені ол пайдаланушыларға таныс визуалды құрылым мен интерактивті элементтерді ұсынады.

Әрі қарай басты беттегі toolbar-ды саусақ қимылымен сырғытқанда 3.7 – суретте көрсетілгендеймәзір пайда болады. Google дизайнерлерінің ұсыныстарымен сәйкес мәзірде белгішелері мен жазулары бар тізім түріндегі элементтер болуы керек, сонымен қатар қажетті функционалдылығы бар тақырып болуы керек. Қосымшада навигация жүзеге асыру үшін арнайы Navigation Drawer деп аталатын шаблонды пайдаланып мәзір құрдық.



3.7 -сурет. Қосымша мәзірі

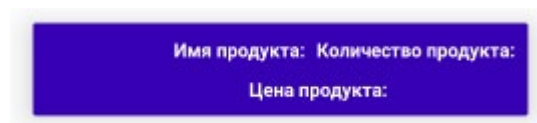
Прототиптелген Navigation Drawer шаблону Android үшін Google бекіткен Material Design талаптарына сай келеді және мобильді қосымша пайдаланушысына қосымша бөлімдері арасындағы қадамдар кезінде ыңғайлылықты қамтамасыз етеді. Сонымен қатар, бөлімдерден бөлек пайдаланушы деректерін, суретерін қамтиды. Мәзір өз кезегінде екі топқа бөлініп жасалынған. Жоғарғы бөлігін пайдаланушы өзгерте алса, төменгі бөлігі өзгертуге келмейді.

Ендігі кезекте мәзірдегі әрбір бөлімге шолу жасайтын болсақ, мәзір бес бөлімнен тұрады. Клиент ұнатқан тауарларын 3.8 – суретте көрсетілгендейсебетке жіберу арқылы сақтай алады, осы сақталған тауарлар мен сатып алынған тауарлар санат бойынша «Мои покупки» бөлімінде орналасады.



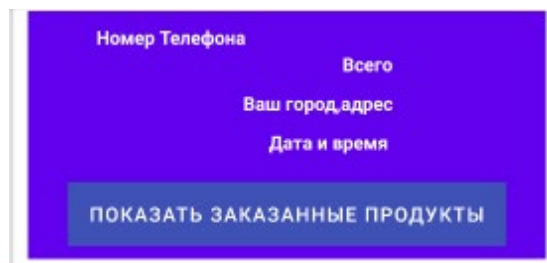
3.8 -сурет. Ұнаған және сатып алынған тауарлар бөлімі

Тауарларға арналған арнайы 3.9 – суретте көрсетілгендей шаблон түрінде берілген item-дер жасалынған. Осы шаблонда таңдаған тауардың жеке ақпараты шығып тұрады.



3.9 -сурет. Ұнаған және сатып алынған тауарлар бөліміндегі тауарлар шаблону

Ұнатқан тауарларына тапсырыс бергеннен кейін «Мои покупки» бөлімінде тауар жайды ақпарат жазылып тұрады. Арнайы батырманы басқан кезде 3.10 – суретте көрсетілгендей тауарлар туралы жеке-жеке ақпараттар шығатын болады.



3.10 -сурет. Ұнаған және сатып алынған тауарлар бөліміндегі тапсырыс берген тауарлар шаблоны

Тапсырыс берілген тауарларға арнайы ұяшықтар ашылып, әр тауар жеке шығып тұрады. Тауарларға арналған ұяшықтар көптеген қосымшаларда қолданатын карта дизайны қолданылған, Android-те олар CardView деп аталады.

3.11 – суретте көрсетілгендей тауарды тапсырыс жасағанда толтырылатын арнайы шаблон.

Подтвердите детали

Ф.И.О

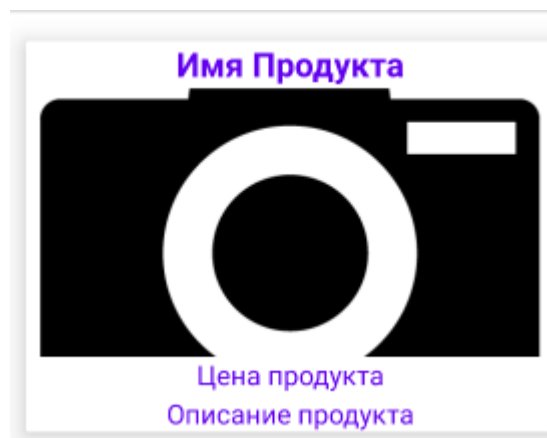
Номер телефона

Ваш адрес

Город

3.11 -сурет. Тауарды тапсырыс жасауға қолданылатын шаблон

Жаңа 3.12 – суретте көрсетілгендей cardview компоненті Android Lollipop – та пайда болды, бірақ үйлесімділік кітапханасының арқасында ескі құрылғылар үшін де қол жетімді. Бұл компонент көлеңке мен дөңгелек бұрыштары бар әдемі картаны жасауға мүмкіндік береді, ол басқа компоненттер үшін контейнер ретінде қызмет етеді.



3.12 -сурет. Жүктелген тауар бейнесі

Қолданушы басты бетте отырғанда немесе басқа беттерде жүргенде тапсырыстарына кіруге арналған тез қол жетімділік батырмасы да жасалған.

FloatingActionButton(FAB) – бұл бағдарламаның пайдаланушы интерфейсіндегі негізгі әрекетін іске қосатын дөңгелек батырма. Бұл беттегі орналасуға 3.13 – суретте көрсетілгендей FAB қосу, оның сыртқы түрін реттеу және батырмаларды басуға қалай жауап беру керектігі көрсетілген.



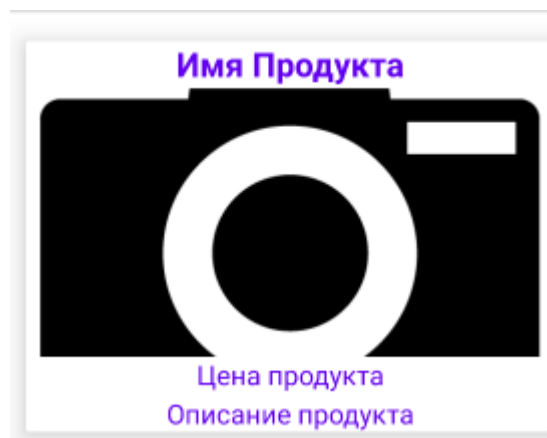
3.13 -сурет. FloatingActionButton

Нақты бір нәрсені қажет ететін пайдаланушылар 3.14 – суретте көрсетілгендей сұранысты қайда енгізу керектігін бірден іздейді. Пайдаланушылар көбінесе қолданбадағы іздеу жүйесіне сүйенеді және оның Google-ден гөрі жақсы жұмыс істегенін қалайды.



3.14 -сурет. Іздеу бөлімі

Іздеу процессі сәтті аяқталса, іздеген мәліметтер 3.15 – суретте көрсетілгендей item ретінде шығарылатын болады. Item (элемент) параметрі мәзір элементіне жауап береді.



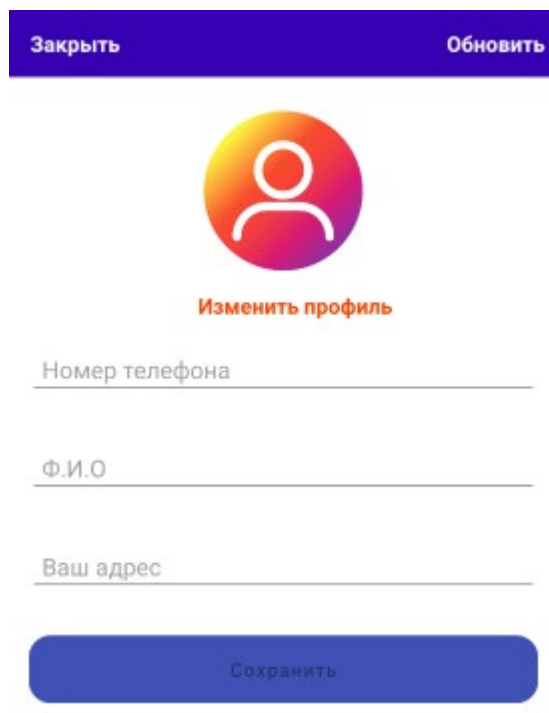
3.15 -сурет. Іздеу бөлімінен пайда болатын тауарлар бейнесі

Категориялар бөлімі 3.16 – суретте көрсетілгендей қолданушылар үшін ыңғайлы, іздеген затын тез арада табуы үшін жасалған бөлім. Бұл бөлім сұраныстарға ие он екі категориядан тұрады. Әр категорияда сатушы саслуларына сәйкес шексіз тауарлар орналастырылады.



3.16 -сурет. Категориялар бөлімі

Категориялар бөлігінен кейін 3.17 – суретте көрсетілгендей баптаулар бөлігі келеді. Бұл бөлікте пайдаланушы жеке бас ақпараттарына ауыстыру мен түзетулер енгізе алады. Пайдаланушының парақшасында жазылған жеке ақпараттары тапсырысты жеткізу кезінде немесе әртүрлі комиссияларды орындау кезінде қажет болуы мүмкін. Сол себептен қолданушының жеке ақпараттары тұрмыстық жағдайда кез-келген кезде өзгеруі мүмкін болғандықтан, парақшаны өзгерту қадамы мобильді қосымшада қолжетімді болғаны жөн.



3.17 -сурет. Қолданушы парақшасы

Айта кететін болсақ, құрастырып отырған мобильді қосымшада қолданушы парақшасынан бөлек, 3.18 – суретте көрсетілгендейсатушы ретінде де қосымшаны пайдалануға мүмкіндік бар. Ол үшін қосымшаға кіру кезінде қосымшаның ең астыңғы бөлігіндегі «Я админ» батырмасын басса жеткілікті. Сол кезде қосымшаға сатушы ретінде өз тауарларын салса болады. Бірақ әрбір сатушы ретінде тіркелген пайдаланушы деректер базасы арқылы арнайы тексерістен өтіп, арнайы рұқсат алған соң ғана қосымшаны пайдалануға мүмкіндік алады.

Я админ!

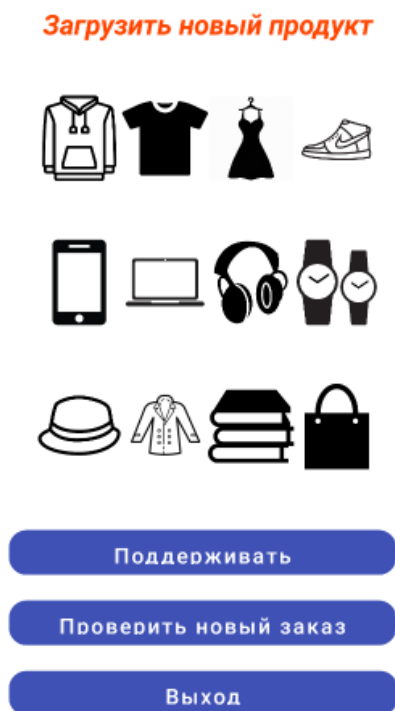
3.18 -сурет. Сатушы қосымшаға кіруіне арналған батырма

Сатушы парақшасының мүмкіндігі қолданушы парақшасына қарағанда шектеулі. Қолданушы парақшасы қолданушыларды бизнесті жүргізуге тарту үшін жасалса, сатушы парақшасы 3.19 – суретте көрсетілгендейтек өз тауарымен бөлісіп, тауардың күйін бақылауға арналған парақша ретінде қарастырса болады.



3.19 -сурет. Жүктелген тауарлар тізімі

Сатушылар парақшасының басты бетінде 3.20 – суретте көрсетілгендеймобильді қосымшаға кіре салысымен басты бетте бірнеше мүмкіндіктер болады. Соның ішінде рауарларды жүктеуге арналған категориялар бөлімі. Категориялар ең керекті деген он екі топқа жіктелген. Категориялар бойынша сататын тауар категориясын таңдайды.



3.20 -сурет. Стаушыға парақшасының басты беті

Қосымшаға сатушылар өз тауарларын 3.21 – суретте көрсетілгендейенгізген кезде арнайы бет ашылады. Бұл жерде тауарға берілетін сипаттамалар көрсетілген. Шарт бойынша барлық мәліметтер толғанда ғана батырма басылып, тауар мәліметтер базасына жүктелуі тиіс.



3.21 -сурет. Тауарды жүктеу бөлімі

Сонымен қатар, 3.22 – суретте көрсетілгендей жүктелген тауар сипаттамасы, құны немесе басқа да ақпараттары үнемі өзгеріп отыру ықтималдылығы жоғары болғандықтан, тауарларды жүктеу бөлімінде тауарларға арналған арнайы өзгерту батырмасы да орналастырылған. Батырманы басу арқылы ақпаратты өзгерете алады немесе тауарды сатылымнан өшіру арқылы алып тастай алады.



3.22 -сурет. Жүктелген тауарларға түзетулер енгізу бөлімі

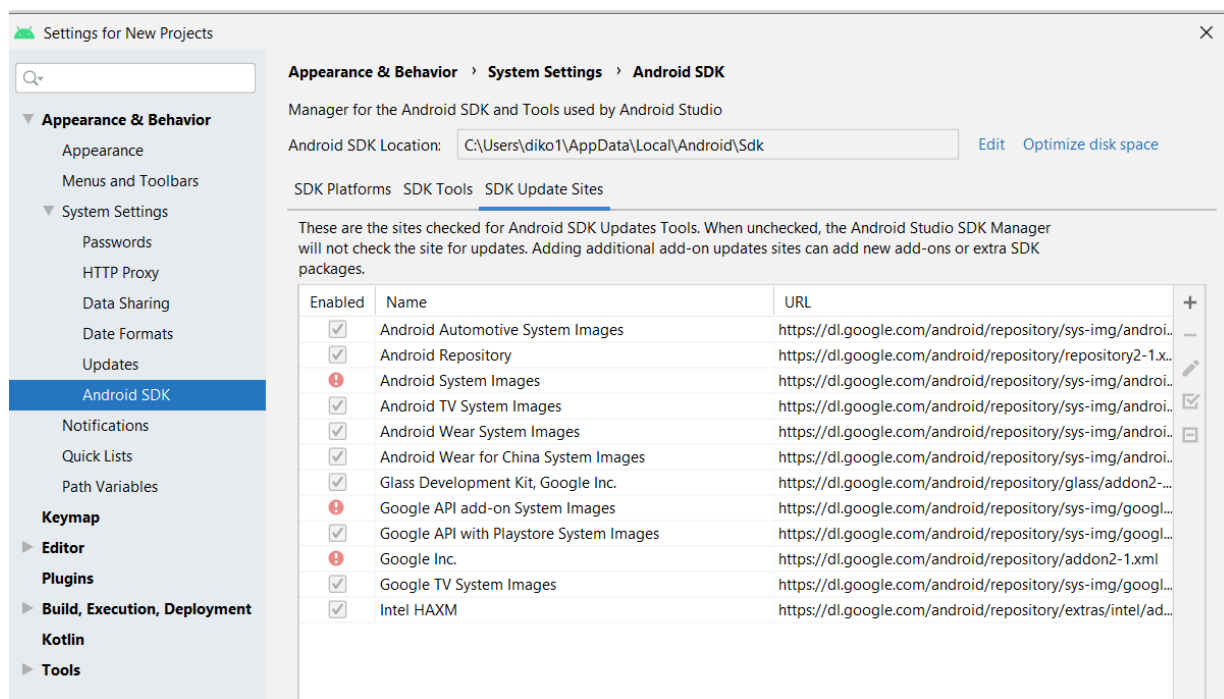
3.2. Мобильді қосымшаны құрастыру қадамдары

Android негізгі құрылғыларды сатудың қарқынды өсуі Android қосымшаларын жасаушыларға керемет мүмкіндіктер ашады. Қазір Android платформасында смартфондар, планшеттер, электронды кітаптар, роботтар, рекативті қозғалтқыштар, теледидар, камералар, медициналық құрылғылар, NASA спутниктері, ойын консольдері, тоназытқыштар, теледидарлар, камералар, ақылды сағаттар, автомобиль ақпараттық жүйелері (радио, GPS, телефондар, термостаттар және т.б. басқару үшін) және басқа да көптеген құрылғылар жұмыс істейді.

Android платформасының басты артықшылықтарының бірі – оның ашықтығы. Android операциялық жүйесі ашық бастапқы код негізінде құрылған және еркін таратылады. Бұл қызметкерлерге Android бастапқы кодына қол жеткізуге және олардың осылайша қосымшалардың әдістері, қасиеттері мен функцияларының жүзеге асырылуын зерттеуге мүмкіндік береді. Бұл процесске кез-келген қолданушы қатыса алады, онда қолданушы Android бастапқы кодын ала алады, ашық операциялық жүйенің негізі болып табылатын идеология туралы біле алады және лицензиялық ақпаратты аза алады. Платформаның ашықтығы тез жаңаруға ықпал етеді.

Android ОЖ-ге арналған қосымшаларды жасау үшін ең алдымен Android Studio платформасын орнату керек.

Android SDK орнату. Android SDK–Android ОЖ-ге орналған мобильді қосымша құрудағы әмбебап құрал. Android SDK міндеттеріне түрлі жаму құралдары жатады, нақты автоматты түзеткіш, кітапханалар жинағы, телефон эмуляторы, құжаттама жинағы, тағы қосымшалар мен нұсқаулықтар жатады. Яғни Android SDK бізге қажетті бөлік болып саналады [16]. Android SDK дамыту құралдарын 3.23 – суретте көрсетілгендейәзірлеушілер сайтында жүктеуге болады. Орнату кезінде сіз платформаны және Android SDK элементтерін тандауға болады [17].



3.23 -сурет. Android SDK элементтері

Android қосымшаларын жасау кезінде көбіне Java қолданылады. Ол ең таралған бағдарламалау тілдерінің бірі болып табылады. Java қолдану Android платформасы үшін логикалық таңдау болады, өйткені ол қуатты, миллиондаған әзірлеушілерге белгілі еркін әрі ашық тіл. Тәжірбиелі Java бағдарламашылары Google Android Api интерфейсін (Application Programming Interface) және тәуелсіз фирмалардың басқа әзірлемелерін қолдана алады [16]. Java тілі объектіге бағытталған, әзірлеушілерге қосымшаларды әзірлеуді тездететін қуатты сынып кітапханаларына қол жеткізуге мүмкіндік береді.

Графикалық пайдаланушы интерфейсін бағдарламалауды экранды түрту сияқты пайдаланушылар бастаған оқиғаларға жауап беретін оқиғалар басқарады. Басқа бағдарлама кодын тікелей жазу арқылы әзірлеуші мыналарды пайдалана алады: қосымша әзірлеуге мүмкіндік беретін Eclipse және Android Studio дамыту орталары, түймелер және мәтін сияқты дайын нысандардың графикалық интерфейсі, оларды экранның белгілі бір жерлеріне апарып, қолтаңбалар қосып және өлшемдерін өзгерте отырып қолдану.

Eclipse және Android Studio мобильді қосымша әзірлеу ортасы тез әрі ыңғайлы орта құрастырып, тестілеу мен түзетулер енгізе алады.

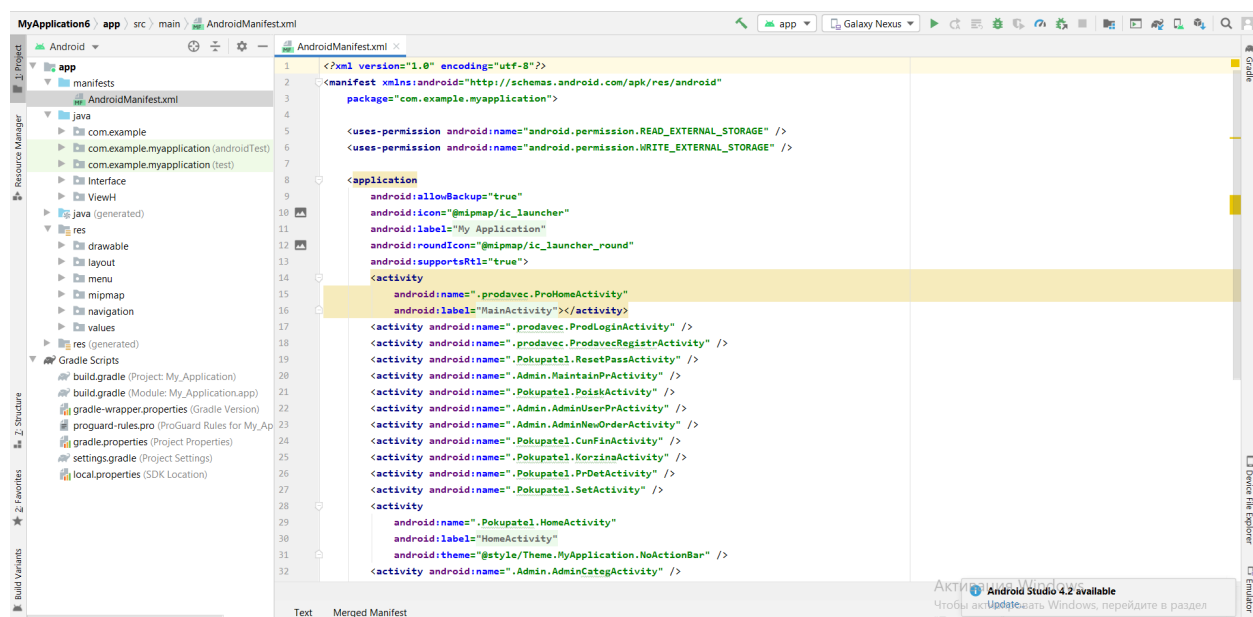
Android Studio-да GUI компонентері көріністер (view) деп аталады. LinerLayout тік көрінісі мәтін мен графиканы орналастыру үшін қолданылады, сондықтан әр LinerLayout көрінісі тік кеңістігінің жартысын алады. LinerLayout компоненті сонымен қатар көріністерді көлденең орналастыруға мүмкіндік береді. Бағдарламада мәтін шығару үшін TextView компоненті қолданылады, ал графика ImageView компонентінде көрсетіледі. Әдепкі бағдарлама үшін жасалған графикалық интерфейс TextView компонентін қамтиды. Оның әртүрлі параметрлері компоненттері — мәтін, қаріп өлшемі, мәтін түсі,

LinearLayout-тағы салыстырмалы өлшем және т.б. imageView компоненттері – даму ортасының қасиеттері терезесінде бапталған. Содан кейін біз imageView компонентін палитрадан GUI макетіне апарамыз және оның қасиеттерін, соның ішінде графикалық деректер көзі мен LinearLayout позициясына баптаулар жүргіземіз.

XML тілі(eXtensible Markup Language, яғни кеңейтілетін белгілеу тілі) графикалық интерфейстерді сипаттаудың табиғи әдісі болып табылады. XML белгілеуін адам да, компьютер де жақсы оқиды. Android контекстінде ол қолданылатын компоненттер мен олардың атрибуттарының орналасуын сипаттау үшін қолданылады: өлшем, позиция, түс, мәтін өлшемдері, өрістер және шегіністер.

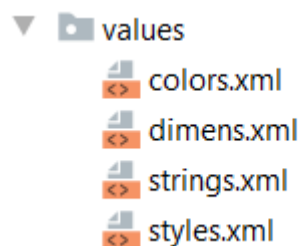
Android Studio орналасуды орналасу редакторында көрсету және орындалу сатысында графикалық интерфейсті құрайтын Java кодын құру үшін XML белгілеуін талдайды. Сондай-ақ, XML файлдары қосымшаның мынандай ресурстарын сақтау үшін қолданылады: жолдар, сандар, түстер және т.б.

Әрбір қосымшада біз қолданылатын стандартты компоненттердің дизайнын анықтайтын тақырып бар. Қосымшаның тақырыбы 3.24 – суретте көрсетілгендейAndroidManifest.xml файлында көрсетіледі.



3.24 -сурет. AndroidManifest.xml файлы

Ондағы styles файлындағы ресурстарды анықтау арқылы қолдаушы жобаның әртүрлі аспектілерін(мысалы, түс схемасының құрамдас бөліктері) реттей алады. Бұл XML қолданбаның 3.25 – суретте көрсетілгендейres/values қалтасында орналасқан.

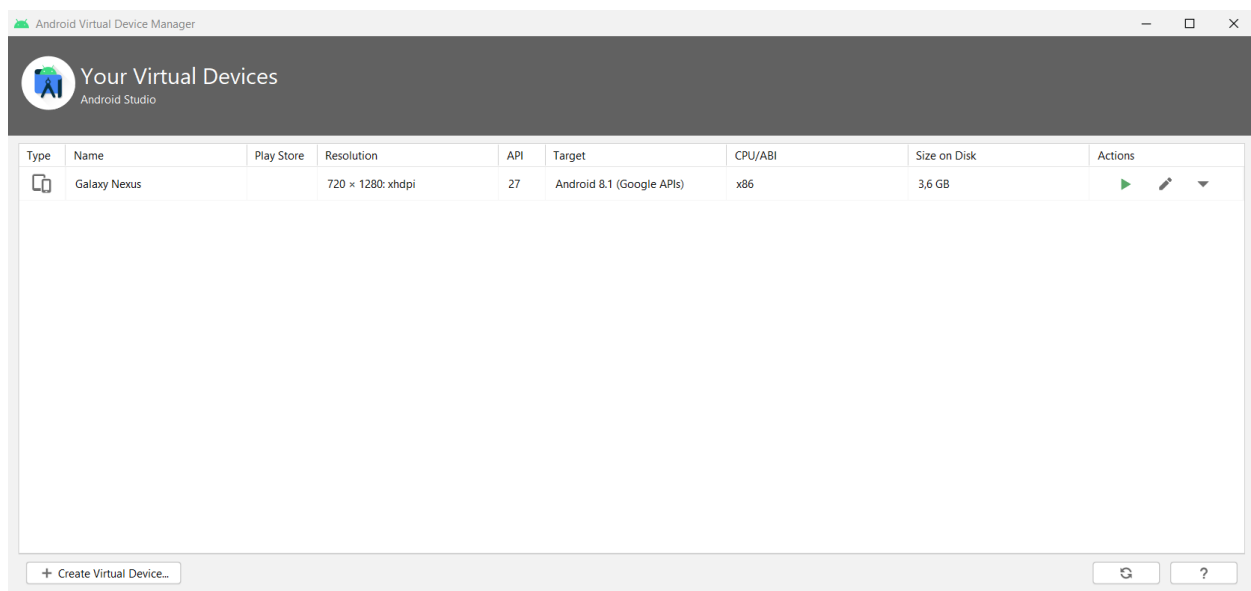


3.25 -сурет. res/values қалтасында орналасқан xml файлдар

Styles.xml ресурстық файлы «AppTheme»деп аталатын стильден тұрады, оның сілтемесі AndroidManifest.xml файлына қосылып қосымшаларға арналған стиль құрылады. Бұл стиль сонымен қатар Java-да суперклассқа ұқсас деп санауға болатын мұрагерлік тақырыбын анықтайды. Бұл жаңа стиль мұрагерлік тақырыбының атрибуттары мен олардың әдепкі мәндерін мұра етеді. Java-дағыдай, стиль мұрагерлік атрибуттарын белгілі бір қосымшаларға бейімделген мәндермен қайта анықтай алады. Біз бұл ойды мобильді қосымша тақырыбында қолданылатын үш түсті баптау үшін қолданамыз.

Жоғарыда айтылғандай, Android Studio қосымшаларының шаблондарына енді платформаның ескі нұсқаларында жаңа Android мүмкіндіктерін пайдалануға мүмкіндік беретін AppCompatActivity кітапханаларынан қолдау алады. Әдепкі бойынша, Android Studio AppCompatActivity.Light.DarkActionBar мұрагерлік тақырыбын таңдайды. AppCompatActivity кітапханасындағы бірнеше стандартты тақырыптардың бірі – осы тақырыпты қолданатын қосымшалар ашық фонда көрсетіледі, ал қосымшаның жоғары жағында қосымшаның қараңғы тақтасы орналасқан. AppCompatActivity-тың барлық тақырыптары графикалық интерфейстерді жобалау үшін Google-дің материалдық дизайнының ұсыныстарын пайдаланады.

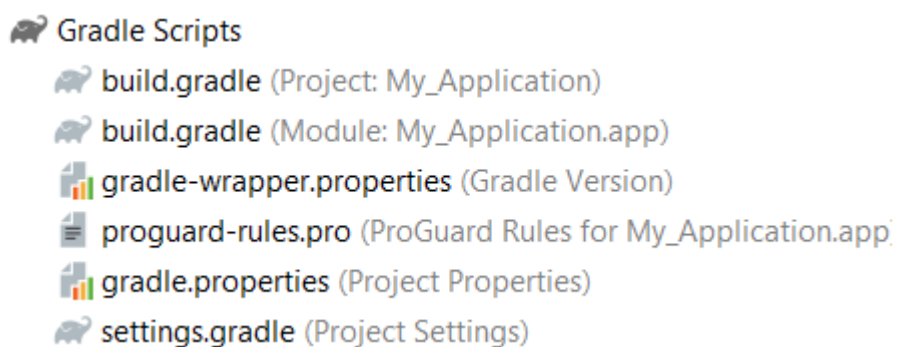
Android виртуалды құрылғысын құру. Android құралдары 3.26 – суретте көрсетілгендей«Android Virtual Device»(AVD) эмуляторын қамтиды. AVD эмуляторы виртуалды мобильді құрылғыда Android ОЖ-сі көмегімен қосымшаларды тексеруге мүмкіндік береді. Эмулятор әртүрлі конфигурациялары бар бірнеше виртуалды құрылғаларды жасауға мүмкіндік береді.



3.26 -сурет. Android виртуалды құрылғысы

Әдепкі бойынша жасалатын Android ОЖ-ге аралған мобильді қосымша жобасының құрылымын қарастырайық.

Жоба әртүрлі модульді қамтуы керек. Барлық модульдер gradle.setting файлымен сипатталады. Егер жобаның құрылымын қарастыратын болсақ, онда барлық маңызды код – интерфейс файлдары, java сыныптары және т.б. әдепкі app модульінде орналасады. Build.gradle файлы проектке қолданылатын барлық ақпараттан тұрады. Әр модульде 3.27 – суретте көрсетілгендей өзінің build.gradle файлы бар, ол осы модульге тән жоба құрылысының конфигурациясын анықтайды.



3.27 -сурет. Build.gradle файлы

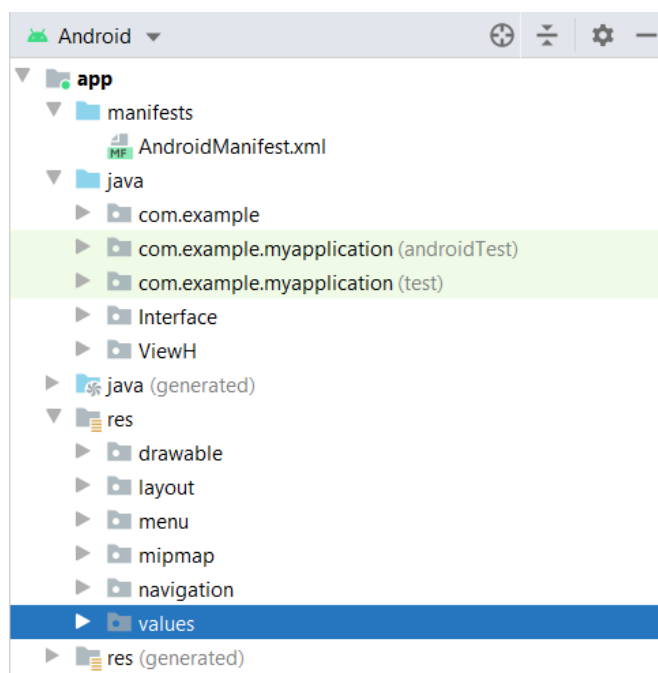
Сонымен, егер біз app модульінің мазмұнын қарастыратын болсақ, онда build.gradle файлын табуға болады. Бастапқы кезеңде бұл файлдар соншалықты маңызды, тек олардың не жұмыс атқаратынын түсіну керек. Әдепкі бойынша, әр жоба бір app модульін қамтиды. Шын мәнінде, біз жұмыс істейтін барлық код осы модуль ішінде орналасқан. Бұл модульде 3.28 – суретте көрсетілгендей

біз бірнеше қалталар мен файлдарды көре аламыз, олар біз үшін ең маңызды болып табылады:

Libs каталогы – қосымшада пайдаланған кітапханаларды сақтауға арналған.

Src каталогы – бастапқы кодты сақтауға арналған. Оның құрамында бірқатар ішкі каталогтар бар.

Негізгі қалта үлкен құрылымға ие:



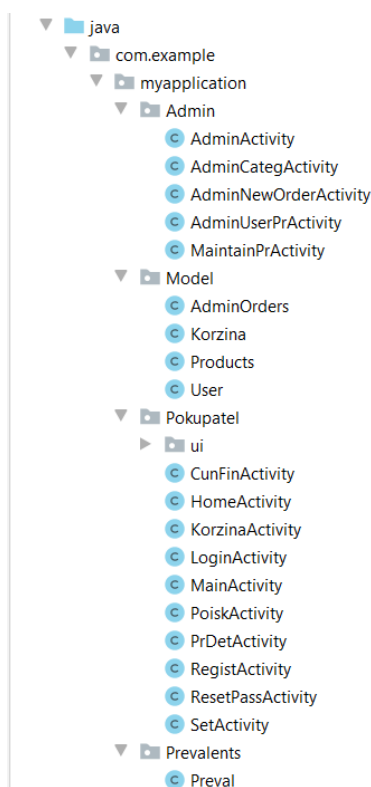
3.28 -сурет. Мобильді қосымшаның толық құрылымы

Біз интерфейстер редакторында қарапайым графикалық интерфейсті жасадық және құрылымдар сипаттамаларын сипаттама терезесінде баптадық. XML белгілеу файлында әдепкі бойынша қолданылатын RelativeLayout компоненті LinearLayout компоненттеріне бөлініп, содан кейін LinearLayout көріністерінің тік орналасуы үшін конфигурацияланды. Қосымшадағы мәтінді компонентте TextView көрметеді, ал суреттер ImageView компонентінде. Мәтін ортасында, үлкейтілген қаріппен және стандартты тақырыптың түстерінің бірінде туралануы үшін біз графикалық интерфейстің TextView компонентін өзгерттік. ImageView компоненттері тінтуірден компоненттерінің палитрасы редакторы арқылы сүйрелеу арқылы орналастырылады. Барлық жолдар мен сандық мәндер жобаның RES қалтасындағы ресурстық файлдарына анықталады.

MainActivity сыныбында Java тілінде объектіге бағытталған бағдарламалаудың көптеген құралдары, соның ішінде сыныптар, нысандар, интерфейстер, анонимді ішкі сыныптар мен мұрагерлік сыныптар да бар. Сондай-ақ, графикалық интерфейсті толтыру тұжырымдамасы ұсынылады, яғни XML файлының мазмұнын оның экран көрінісіне түрлендіруі. Біз Android

Activity класымен және өмірлік циклмен таныстық. Атап айтқанда, біз іске қосу кезінде қосымшаны іске қосу үшін onCreate әдісін қайта анықтадық. onCreate әдісінде Activity класындағы findViewById әдісі бағдарлама бағдарламалық жасақтама деңгейінде өзара әрекеттесетін визуалды компоненттерге сілтемелер алу үшін қолданылады. Соңында біз AndroidManifest.xml файлын өңдедік. Бұл MainActivity тек портреттік бағдарды қолдайтындығын және MainActivity класы әрқашан виртуалды пернетақтаны көрсетуі қажет.

Сонымен қатар, жобаны құру кезінде манифестке Android Studio қосылған 3.29 – суретте көрсетілгендей басқа элементтер енгізіледі. Біз Configuration нысанын қалай пайдалану керектігін көрсеттік, бағдарлама планшетте ландшафт бағдарында орындалды ма. Сондай-ақ, бұл ырауда assets қалтасындағы кіріктірілген қалталарды қолдана отырып, көптеген графикалық ресурстарды қалай басқаруға болатындығы және AssetManager арқылы осы ресурстарға қалай қол жеткізуге болатындығы көрсетілген. Сондай-ақ, қосымшаның RES қалтасынан басқа қалталар қарастырылады. Олар menu ресурстарының файлдарын сақтауға арналған мәзір, xml белгіленген файлдарды сақтауға арналған XML файлы.



3.29 -сурет. Мобильді қосымшаның толық құрылымы

3.3. Мобильді қосымшаның деректер қорымен байланысы

Firebase – бұл Google мобильді және веб-қосымшаларды құруға арналған платформа. Firebase бастапқыда 2011 жылы құрылған тәуелсіз компания

болды[18]. 2014 жылы Google платформаны сатып алды және қазір бұл Google-де қосымшаларды әзірлеуге арналған басты платформа.

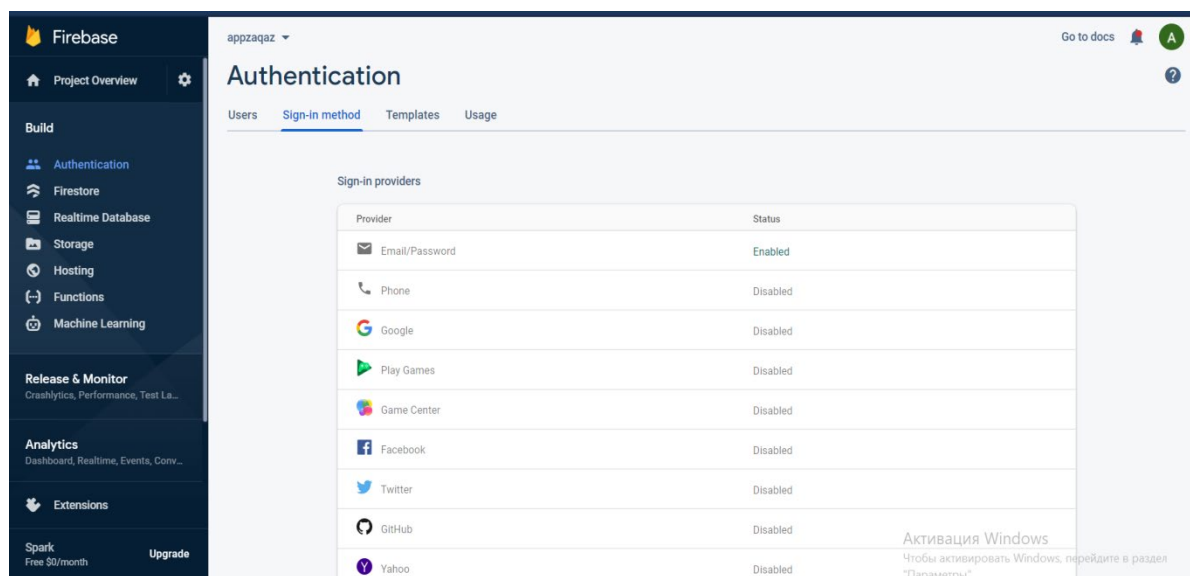
Firebase платформасы үш топқа бөлінген 18 өнімнен тұрады:

- Даму;
- Сапа;
- Өсу.

Firebase пайдаланушыларға сақталған ақпаратты сақтауға және алуға мүмкіндік беретін, сонымен қатар онымен өзара әрекеттесудің ыңғайлы құралдары мен әдістері бар бұлтты мәліметтер базасы.

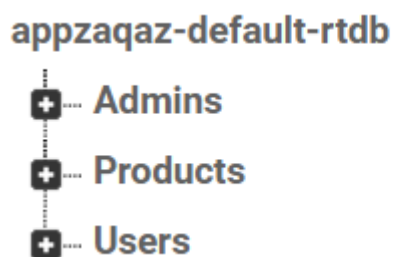
Firebase мәтіндік деректерді JSON форматында сақтайды және деректерді оқу, жаңарту және алудың ыңғайлы әдістерін ұсынады. Сондай-ақ, Firebase пайдаланушыларды тіркеуге және авторизациялауға, сессияларды (уәкілетті пайдаланушылар) сақтауға, Storage cloud арқасында оңай қол жеткізуге мүмкіндік беретін медиа файлдарға көмектеседі.

Пошта және пароль арқылы тіркеуді қосамыз. Қаласақ, поштаға келген сілтеме бойынша авторизацияны қоса аламыз. Толық ақпаратты 3.30 – суретте көрсетілгендей «электрондық пошта мекенжайы мен құпия сөзді» басып, қажетті параметрлерді көресіз.



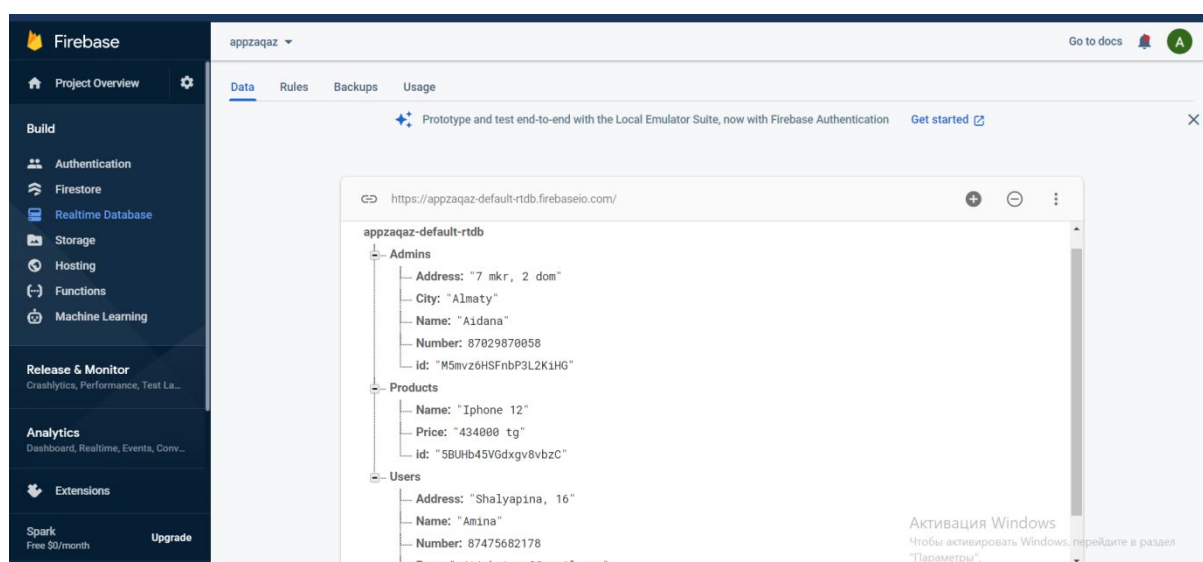
3.30 -сурет. Қосымшадағы электрондық пошта мекенжайы мен құпия сөзді іске қосу

Firebase Realtime дерекқоры – бұл Firebase бұлтты NoSQL дерекқоры. Деректер 3.31 - суретте көрсетілгендей JSON форматында сақталады және әр қосылған клиентпен нақты уақыт режимінде синхрондалады [19]. Android және iOS операциялық жүйелері үшін қосымшалармен интеграциялау ерекшеліктері қолдау тапты, JavaScript, Java, Objective-C және Node.js қосымшаларына арналған API іске асырылды, сонымен қатар AngularJS, React, Vue.js, Ember.js және Backbone.js сияқты бірқатар JavaScript шеңберлерінен rest стиліндегі дерекқормен тікелей жұмыс істеуге болады.



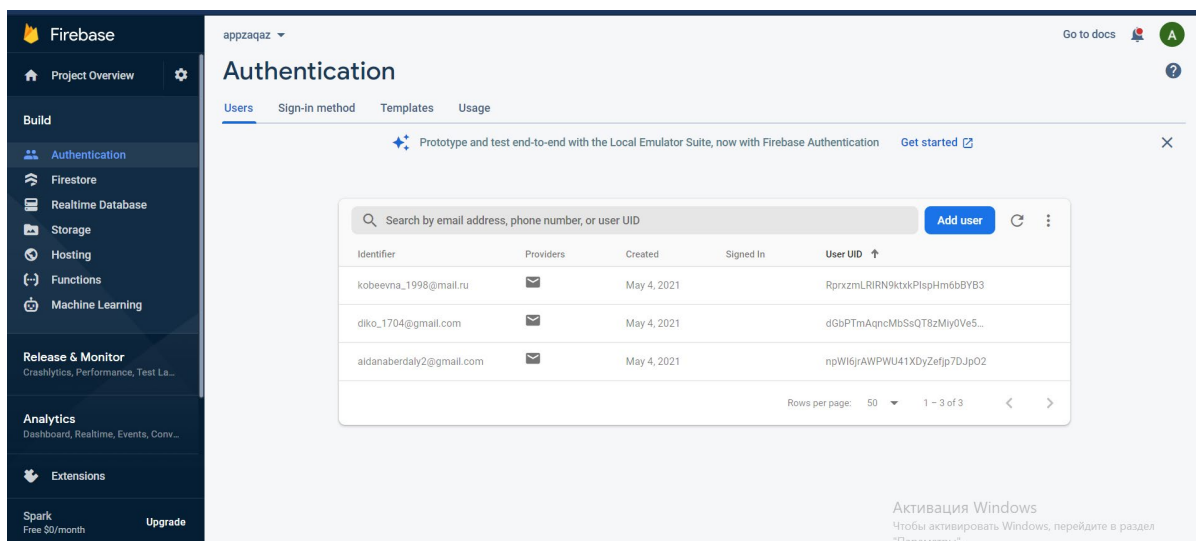
3.31 -сурет. Firebase Realtime дерекқоры

Мобильді қосымша үшін Firebase консолінде 3.32 – суретте көрсетілгендей деректер қоры құрылған болатын. Осы деректер қоры бойынша қосымша базасы үш бөлімнен тұрады. Осы үш бөлім бойынша пайдаанушылар қосымшаға тіркелген кезде және парақшасындағы жеке бас ақпараттарын өзгерткен кездегі деректер жаңартылып, сақталып отырады.



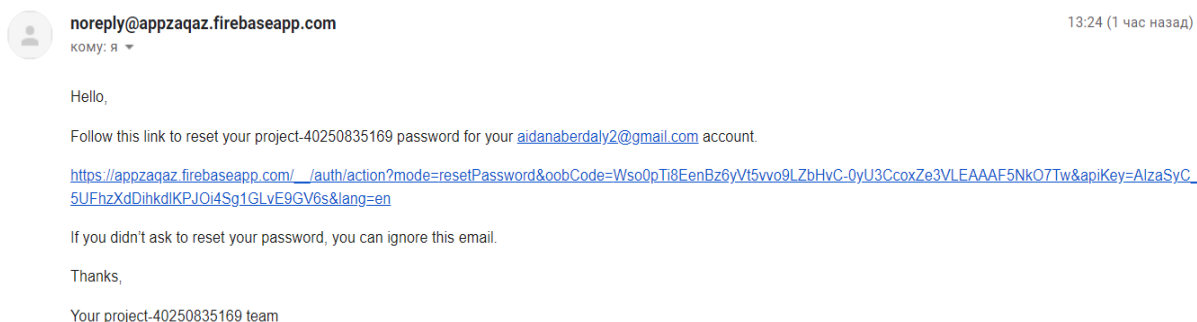
3.32 -кесте. Firebase Realtime дерекқоры

Аутентификация – бұл бір нәрсенің түпнұсқалығын тексеру процесі. Бұл термин көбінесе ақпараттық технологиялар ортасында қолданылады. Аутентификацияның мысалы пайдаланушы енгізген парольді сервер дерекқорында сақталған парольмен салыстыру болуы мүмкін [20]. Біздің жағдайда 3.33 – суретте көрсетілгендей пайдаланушының почтасы мен құпия кілт сөзі аутентификация бөлімінде сақталады. Кілт сөзді ұмытып қалу секілді жағдайлар орын алса осы жерден құпия сөзге қайта сұраныс жасаса болады.



3.33 -сурет. Аутентификация

Электронды почта арқылы 3.34 – суретте көрсетілгендей парольді ауыстыруға ұсыныс жасалған кезде поштаға келген арнайы хабарлама.



3.34 -сурет. Парольді почталық адрес арқылы қалпына келтіру процессі

Жалпы сатып алушылар мен сатушылар арасындағы байланыс 3.35 – суретте көрсетілгендей деректер базасы арқылы жүзеге асатын болады. Сатушылар өз парақшасынан тауарды жүктейді. Жүктелген тауар деректер базасында сақталады. Сақталып болған тауарлар критерий бойынша тасымалданып, кей тауарлар клиент парақшасының басты бетінде жаңалықтар ретінде де шығады.



3.35 -сурет. Сатып алушылар мен сатушылар арасындағы байланыс

ҚОРЫТЫНДЫ

Дипломдық жұмыстың мақсаты – мобильді қосымша құру арқылы жаңа клиенттерді тиімді тарту, өнім сатылымымен айналысатын түрлі компаниялардың тапсырыс жүйесін автоматтандыру үшін мобильді қосымшаны әзірлеу болып табылады, сонда-ақ, Android Studio әзірлеу ортасында Android ОЖ үшін практикалық бағдарламалау дағдыларын бекіту.

Осы мақсатқа жету үшін жұмыс барысында келесі міндеттер дәйекті түрде шешілді:

- Пәндік салаға және қолданыстағы шешімдерге талдау жасау;
- Мобильді қосымшаның тұжырымдамасын жасау;
- Мобильді қосымшаның функционалын анықтау;
- Мобильді қосымшада қолданылатын технологияларды анықтау;
- Мобильді қосымшаны әзірлеу;

Дипломдық жұмыстың бірінші тарауында мобильді қосымшаны құр тиімділігінің экономикалық көрсеткіштерін есептеу және мобильді қосымшаның көмегімен тұтынушылар ағынын арттыру жолдарын іздеу және талдау қарастырылды.

Мәселені шешу кезінде бағдарламалық жобаның экономикалық тиімділігін, атап айтқанда мобильді даму саласына қатысты есептеудің теориялық және практикалық әдістері қарастырылды.

Мобильді қосымшаның көмегімен тұтынушылар ағынын арттыру жолдары табылды, талданды және ұсынылды.

Екінші тарауда мобильді қосымшаға талдау жасалынып, талаптар қойылды, пайдаланушы интерфейсінің прототипі жасалған интерфейске қойылатын функционалдық талаптар талданды. Оның көмегімен болашақта мобильді технологиялар саласындағы жаңа технологияларды қолдана отырып, Android ОЖ үшін мобильді қосымшаның минималды нұсқасы жасалды. Мәселені шешу кезінде бағдарламалық жасақтамаға, пайдаланушы интерфейсіне қойылатын талаптарды ұсыну және талдау әдістері теориялық тұрғыдан зерттелді. Android Studio интерфейсімен жұмыс істеудің практикалық дағдысы алынды, онда стандарттар талаптарына сәйкес келетін пайдаланушы интерфейсі жасалды. Дипломдық жұмыстың талаптарына сәйкес келетін және сипатталған талаптарға сәйкес жұмыс істейтін қосымша жасалынды. Сондай-ақ, Android Studio дамыту ортасында жаңа жұмыс дағдылары бекітілді және оқытылды.

Дипломдық жұмыс аясында төменде көрсетілген нәтижелерге қол жеткізілді:

- Google Play-де қол жетімді және онлайн сатылым жүйесіне қатысы бар мобильді қосымшалар қарастырылды (AliExpress, Amazon және т.б.);
- Келесі функцияларды жүзеге асыратын қосымша құрылды:
 - Пайдаланушы үйден шықпай ақ онлайн тауарларға тапсырыс бере алатын;

- Интерфейс жағынан артық оқытуды талап етпейтін, қолданысқа ыңғайлы және түсінікті;
- Таңдаулы қосымшаларды пайдаланып таңдалған нысанға навигация.
- Қосымша Android SDK-дан алынған стандартты эмуляторда да, Android платформасындағы нақты құрылғыларда (планшеттік компьютер мен смартфон) сынақтан өткізіліп, жұмыс қабілеттілігін көрсетті.

Осылайша, міндеттер толық көлемде шешілді, мақсатқа қол жеткізілді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Кудинова А., Сорокин М., CRM: практика эффективного бизнеса, 2012. – 38 с.
2. Itset.kz [Itset.kzкомпаниясының жеке веб-сайты]. – [05.11.2017] – Режим доступа:<https://www.itset.kz/catalog/1C8/KA/>
3. Solutions.1c.ru[1Скомпаниясының жеке веб-сайты]. – [2019] – Режим доступа:<https://solutions.1c.ru/catalog/crm-prof-20-kz/features>
4. Михаил Гордеев-Бургвиц, Системы автоматического управления взаимосвязанными электроприводами мощных техник, 2017. – 106 с.
5. Радионов А.А., Карандаев А.С., Достижения в автоматизации, 2019. – 23 с.
6. М. Федотенко. Разработка мобильных приложений. Первые шаги, 2019. – 52 с.
7. Лоре Арно, Проектирование веб – API, 2020. – 39 с.
8. Википедия[общедоступная многоязычная универсальная интернет-энциклопедия]. – [17.02.2021] – Режим доступа:https://ru.wikipedia.org/wiki/Android_Studio
9. Ян Ф. Дарвин, Android. Сборник рецептов, 2018. – 182 с.
10. Мельникова О.М, Смартфоны на Android. Издательство Эксмо, 2013. – 252 с.
11. Дэвид Гриффитес, Дон Гриффитес, Head first. Программирование для Android, 2018. – 98 с.
12. Зеев Фрейман, Создание приложений для смартфонов и планшетов под ОС Android, 2018. – 71 с.
13. Bill Phillips, Brain Hardy. Android Programming: The Big Nerd Ranch Guide, 2013. – 403 p.
14. Брюсь Эккель, Философия Java, 2016. – 92 с.
15. Хорстманн Кей. Java. Библиотека профессионала.Том 1. Основы, 2018. – 53 с.
16. Википедия[общедоступная многоязычная универсальная интернет-энциклопедия]. – [21.09.2020] – Режим доступа:https://ru.wikipedia.org/wiki/Android_SDK
17. Fandroid.info[сайт уроки по разработке андроид приложений]. – [2020] – Режим доступа:<http://www.fandroid.info/dobavlenie-paketov-android-sdk-manager/>
18. Нил Смит, Firebase Essential – Android Edition, 2017. – 66 p.
19. Ashok Kumar S, Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase, 2018. – 115 p.
20. Coderlessons.com [Электронный ресурс]. – [13.01.2019] – Режим доступа:<https://coderlessons.com/tutorials/web-razrabotka/izuchite-firebase/uchebnik-po-firebase>

ҚОСЫМША AndroidManifest.xml

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

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true">
        <activity
            android:name=".prodavec.ProHomeActivity"
            android:label="@string/title_activity_pro_home"></activity>
        <activity android:name=".prodavec.ProdLoginActivity" />
        <activity android:name=".prodavec.ProdavecRegistrActivity" />
        <activity android:name=".Pokupatel.ResetPassActivity" />
        <activity android:name=".Admin.MaintainPrActivity" />
        <activity android:name=".Pokupatel.PoiskActivity" />
        <activity android:name=".Admin.AdminUserPrActivity" />
        <activity android:name=".Admin.AdminNewOrderActivity" />
        <activity android:name=".Pokupatel.CunFinActivity" />
        <activity android:name=".Pokupatel.KorzinaActivity" />
        <activity android:name=".Pokupatel.PrDetActivity" />
        <activity android:name=".Pokupatel.SetActivity" />
        <activity
            android:name=".Pokupatel.HomeActivity"
            android:label="@string/title_activity_home"
            android:theme="@style/Theme.MyApplication.NoActionBar" />
        <activity android:name=".Admin.AdminCategActivity" />
        <activity android:name=".Admin.AdminActivity" />
        <activity android:name=".Pokupatel.RegistActivity" />
        <activity android:name=".Pokupatel.LoginActivity" />
        <activity android:name=".Pokupatel.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.theartofdev.edmodo.cropper.CropImageActivity"
            android:theme="@style/Base.Theme.AppCompat" />
    </application>

</manifest>
```

AdminActivity

```
package com.example.myapplication.Admin;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
```

```

import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.example.myapplication.R;
import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;

public class AdminActivity extends AppCompatActivity {
    private String CatName,Opisan,Pricee,ProdN,SaveCurD,SaveCurT;
    private ImageView InputImage;
    private Button AddProduct;
    private EditText InputPrName,Opisanie,Price;
    private static final int GalleryPick=1;
    private Uri ImageUri;
    private String PrRandKl,downIm;
    private StorageReference PrImageRef;
    private DatabaseReference ProdRef;
    private ProgressDialog loadingBaar;

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

        CatName=getIntent().getExtras().get("category").toString();
        PrImageRef= FirebaseStorage.getInstance().getReference().child("Products im");
        ProdRef=FirebaseDatabase.getInstance().getReference().child("Products");
        AddProduct=(Button) findViewById(R.id.add_product);
        InputImage=(ImageView) findViewById(R.id.product_src);
        InputPrName=(EditText) findViewById(R.id.Pro_name);
        Opisanie=(EditText) findViewById(R.id.opisan_prod);
        Price=(EditText) findViewById(R.id.price_prod);
        loadingBaar=new ProgressDialog(this);

        InputImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                OpenGallery();
            }
        });

        AddProduct.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ValidateProdData();
            }
        }

```

```

    });
}
private void OpenGallery(){
    Intent galleryInt =new Intent();
    galleryInt.setAction(Intent.ACTION_GET_CONTENT);
    galleryInt.setType("image/*");
    startActivityForResult(galleryInt, GalleryPick); }

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode==GalleryPick && resultCode==RESULT_OK && data!=null){

        ImageUri=data.getData();
        InputImage.setImageURI(ImageUri);
    }
}
private void ValidateProdData(){
    Opisan=Opisanie.getText().toString();
    Pricee=Price.getText().toString();
    ProdN=InputPrName.getText().toString();

    if (ImageUri == null) {
        Toast.makeText(this, "Изображение продукта обязательно!", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(Opisan)){
        Toast.makeText(this, "Напишите описание о продукте!", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(Pricee)){
        Toast.makeText(this, "Напишите цену продукта!", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(ProdN)){
        Toast.makeText(this, "Напишите название продукта!", Toast.LENGTH_SHORT).show();
    }
    else{
        StorePrInf();
    }
}
private void StorePrInf() {
    loadingBaar.setTitle("Загрузка нового продукта.");
    loadingBaar.setMessage("Подождите, мы загружаем ваши продукты...");
    loadingBaar.setCanceledOnTouchOutside(false);
    loadingBaar.show();
    Calendar calendar=Calendar.getInstance();
    SimpleDateFormat curDate=new SimpleDateFormat("мм,дд,гг");
    SaveCurD = curDate.format(calendar.getTime());
    SimpleDateFormat curTime=new SimpleDateFormat("чч,мм,сс");
    SaveCurT = curTime.format(calendar.getTime());
    PrRandKl=SaveCurD+SaveCurT;
    StorageReference fileP=PrImageRef.child(ImageUri.getLastPathSegment()+PrRandKl+".jpg");
    final UploadTask uploadTask=fileP.putFile(ImageUri);
    uploadTask.addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            String message=e.toString();
            Toast.makeText(AdminActivity.this, "Ошибка"+message, Toast.LENGTH_SHORT).show();
            loadingBaar.dismiss();
        }
    }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override

```

```

        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            Toast.makeText(AdminActivity.this, "Изображение успешно загружено!",
                Toast.LENGTH_SHORT).show();
            Task<Uri> uriTask=uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot,
                Task<Uri>>() {
                    @Override
                    public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {
                        if(!task.isSuccessful()){
                            throw task.getException();
                        }
                        downIm=fileP.getDownloadUrl().toString();
                        return fileP.getDownloadUrl();
                    }
                }).addOnCompleteListener(new OnCompleteListener<Uri>() {
                    @Override
                    public void onComplete(@NonNull Task<Uri> task) {
                        if (task.isSuccessful()) {
                            downIm=task.getResult().toString();
                            Toast.makeText(AdminActivity.this, "Изображение продукта успешно сохранено в базе...",
                                Toast.LENGTH_SHORT).show();
                            SavePrInfonDB();
                        }
                    }
                });
        }
    }

    private void SavePrInfonDB() {
        HashMap<String,Object> productMap=new HashMap<>();
        productMap.put("pid",PrRandKI);
        productMap.put("date",SaveCurD);
        productMap.put("vremya",SaveCurT);
        productMap.put("opisanie",Opisan);
        productMap.put("image",downIm);
        productMap.put("category",CatName);
        productMap.put("price",Pricee);
        productMap.put("NameP",ProdN);

        ProdRef.child(PrRandKI).updateChildren(productMap)
            .addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if(task.isSuccessful()){
                        Intent intent=new Intent(AdminActivity.this, AdminCategActivity.class);
                        startActivity(intent);
                        Toast.makeText(AdminActivity.this, "Продукт успешно загружено",
                            Toast.LENGTH_SHORT).show();
                        loadingBaar.dismiss();
                    }
                    else {
                        loadingBaar.dismiss();
                        String message= task.getException().toString();
                        Toast.makeText(AdminActivity.this, "Ошибка"+message, Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
}

```

AdminCategActivity

```

package com.example.myapplication.Admin;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

import com.example.myapplication.Pokupatel.HomeActivity;
import com.example.myapplication.Pokupatel.MainActivity;
import com.example.myapplication.R;

public class AdminCategActivity extends AppCompatActivity {
    private ImageView tolstovka,futbolka,dress,shoes;
    private ImageView mobile,laptop,nauwnik,watch;
    private ImageView hat,coat,books,bag;
    private Button Logout_btn,check_btn,maintain_btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_categ);

        Logout_btn=(Button) findViewById(R.id.logout_order_btn);
        check_btn=(Button) findViewById(R.id.check_btn);
        maintain_btn=(Button) findViewById(R.id.maintain_btn);

        maintain_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(AdminCategActivity.this, HomeActivity.class);
                intent.putExtra("com/example/myapplication/Admin", "com/example/myapplication/Admin");
                startActivity(intent);
            }
        });

        Logout_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(AdminCategActivity.this, AdminNewOrderActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK| Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
                finish();
            }
        });

        check_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(AdminCategActivity.this, MainActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK| Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
                finish();
            }
        });

        tolstovka=(ImageView) findViewById(R.id.tolstovka);
        futbolka=(ImageView) findViewById(R.id.futbolka);
        dress=(ImageView) findViewById(R.id.dress);
        shoes=(ImageView) findViewById(R.id.shoes);
        mobile=(ImageView) findViewById(R.id.mobile);

```

```

laptop=(ImageView) findViewById(R.id.laptop);
nauwnik=(ImageView) findViewById(R.id.nauwnik);
watch=(ImageView) findViewById(R.id.watch);
hat=(ImageView) findViewById(R.id.hat);
coat=(ImageView) findViewById(R.id.coat);
books=(ImageView) findViewById(R.id.books);
bag=(ImageView) findViewById(R.id.bag);

tolstovka.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "tolstovka");
        startActivity(intent);
    }
});
futbolka.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "futbolka");
        startActivity(intent);
    }
});
dress.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "dress");
        startActivity(intent);
    }
});
shoes.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "Shoes");
        startActivity(intent);
    }
});
mobile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "mobile");
        startActivity(intent);
    }
});
laptop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "laptop");
        startActivity(intent);
    }
});
nauwnik.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent=new Intent(AdminCategActivity.this, AdminActivity.class);
        intent.putExtra("category", "nauwnik");
        startActivity(intent);
    }
});
watch.setOnClickListener(new View.OnClickListener() {

```



```

        @Override
        public void onClick(View view) {
            Intent intent=new Intent(AdminCategActivity.this,AdminActivity.class);
            intent.putExtra("category","watch");
            startActivity(intent);
        }
    });
    hat.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(AdminCategActivity.this,AdminActivity.class);
            intent.putExtra("category","hat");
            startActivity(intent);
        }
    });
    coat.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(AdminCategActivity.this,AdminActivity.class);
            intent.putExtra("category","coat");
            startActivity(intent);
        }
    });
    books.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(AdminCategActivity.this,AdminActivity.class);
            intent.putExtra("category","books");
            startActivity(intent);
        }
    });
    bag.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(AdminCategActivity.this,AdminActivity.class);
            intent.putExtra("category","bag");
            startActivity(intent);
        }
    });
}
}
}

```

AdminUserPrActivity

```

package com.example.myapplication.Admin;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.example.myapplication.Model.Korzina;
import com.example.myapplication.R;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import ViewH.KorzineViH;
public class AdminUserPrActivity extends AppCompatActivity {
    private RecyclerView productL;
    RecyclerView.LayoutManager layoutManager;
    private DatabaseReference korlistR;

```

```

private String userID=" ";
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_admin_user_pr);
    userID=getIntent().getStringExtra("uid");
    productL=findViewById(R.id.product_list);
    productL.setHasFixedSize(true);
    layoutManager=new LinearLayoutManager(this);
    productL.setLayoutManager(layoutManager);
    korlistR= FirebaseDatabase.getInstance().getReference().child("Cart
list").child("com.example.myapplication.Admin view").child(userID).child("Products");
}
@Override
protected void onStart() {
    super.onStart();
    FirebaseRecyclerOptions<Korzina> options=new
    FirebaseRecyclerOptions.Builder<Korzina>().setQuery(korlistR,Korzina.class).build();
    FirebaseRecyclerAdapter<Korzina, KorzineViH>adapter=new FirebaseRecyclerAdapter<Korzina,
    KorzineViH>(options) {
        @Override
        protected void onBindViewHolder(@NonNull KorzineViH korzineViH, int i, @NonNull Korzina korzina) {
            korzineViH.txtPrQu.setText("Количество="+korzina.getKol());
            korzineViH.txtPrPr.setText("Цена="+korzina.getPrice());
            korzineViH.txtPrN.setText("Название="+korzina.getPname());
        }
        @NonNull
        @Override
        public KorzineViH onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
            View view= LayoutInflater.from(parent.getContext()).inflate(R.layout.c_item_layout,parent,false);
            KorzineViH korzineViH=new KorzineViH(view);
            return korzineViH;
        }
    };
    productL.setAdapter(adapter);
    adapter.startListening();
}
}

```

MaintainPrActivity

```

package com.example.myapplication.Admin;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.example.myapplication.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
import java.util.HashMap;
public class MaintainPrActivity extends AppCompatActivity {
    private Button apply_btn,delete_btn;
    private EditText name,price,opisanie;

```

```

private ImageView imageView;
private String proID="";
private DatabaseReference proRef;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maintain_pr);
    proID=getIntent().getStringExtra("pid");
    proRef= FirebaseDatabase.getInstance().getReference().child("Products").child(proID);
    apply_btn=findViewById(R.id.apply_btn);
    name=findViewById(R.id.Pro_name_main);
    price=findViewById(R.id.Pro_cena_main);
    opisanie=findViewById(R.id.Pro_opis_main);
    imageView=findViewById(R.id.pro_image_main);
    delete_btn=findViewById(R.id.delete_btn);
    displaySpPrInfo();
    apply_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            apply_change();
        }
    });
    delete_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DeleteProd();
        }
    });
}

private void DeleteProd() {
    proRef.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            Intent intent=new Intent(MaintainPrActivity.this, AdminCategActivity.class);
            startActivity(intent);
            finish();
            Toast.makeText(MaintainPrActivity.this, "Этот продукт успешно удалено!",
Toast.LENGTH_SHORT).show();

        }
    });
}

private void apply_change() {
    String pName=name.getText().toString();
    String pCena=price.getText().toString();
    String pOpis=opisanie.getText().toString();

    if(pName.equals("")){
        Toast.makeText(this, "Напишите имя продукта", Toast.LENGTH_SHORT).show();
    }
    else if(pCena.equals("")){
        Toast.makeText(this, "Напишите цену продукта", Toast.LENGTH_SHORT).show();
    }
    else if(pOpis.equals("")){
        Toast.makeText(this, "Напишите описание продукта", Toast.LENGTH_SHORT).show();
    }
    else {
        HashMap<String,Object> productMap=new HashMap<>();
        productMap.put("pid",proID);

```

```

        productMap.put("opisanie",pOpis);
        productMap.put("price",pCena);
        productMap.put("NameP",pName);

        proRef.updateChildren(productMap).addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if(task.isSuccessful()){
                    Toast.makeText(MaintainPrActivity.this, "Изменение успешно сохранено!",
Toast.LENGTH_SHORT).show();
                    Intent intent=new Intent(MaintainPrActivity.this, AdminCategActivity.class);
                    startActivity(intent);
                    finish();
                }
            }
        });
    }

    private void displaySpPrInfo() {
        proRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()){
                    String pName=snapshot.child("NameP").getValue().toString();
                    String pCena=snapshot.child("price").getValue().toString();
                    String pOpis=snapshot.child("opisanie").getValue().toString();
                    String pImage=snapshot.child("image").getValue().toString();
                    name.setText(pName);
                    price.setText(pCena);
                    opisanie.setText(pOpis);
                    Picasso.get().load(pImage).into(imageView);
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });
    }
}

```

Korzina

```
package com.example.myapplication.Model;
```

```

public class Korzina {
    private String pid,pname,price,kol,dicount;

    public Korzina(){
    }
    public Korzina(String pid, String pname, String price, String kol, String dicount) {
        this.pid = pid;
        this.pname = pname;
        this.price = price;
        this.kol = kol;
        this.dicount = dicount;
    }

    public String getPid() {
        return pid;
    }
    public void setPid(String pid) {
        this.pid = pid;
    }
    public String getPname() {

```

```

        return pname;
    }
    public void setName(String pname) {
        this.pname = pname;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
    public String getKol() {
        return kol;
    }
    public void setKol(String kol) {
        this.kol = kol;
    }
    public String getDicount() {
        return dicount;
    }
    public void setDicount(String dicount) {
        this.dicount = dicount;
    }
}

```

Products

```
package com.example.myapplication.Model;
```

```
import android.widget.ImageView;
```

```

public class Products {
    public ImageView imageView;
    private String pname,opisan,price,img,categor,pid,date,time;

    public Products(String pname, String opisan, String price, String img, String categor, String pid, String date, String
time) {
        this.pname = pname;
        this.opisan = opisan;
        this.price = price;
        this.img = img;
        this.categor = categor;
        this.pid = pid;
        this.date = date;
        this.time = time;
    }

    public String getPname() {
        return pname;
    }
    public void setName(String pname) {
        this.pname = pname;
    }
    public String getOpisan() {
        return opisan;
    }
    public void setOpisan(String opisan) {
        this.opisan = opisan;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
}

```

```

    }
    public String getImg() {
        return img;
    }
    public void setImg(String img) {
        this.img = img;
    }
    public String getCategor() {
        return categor;
    }
    public void setCategor(String categor) {
        this.categor = categor;
    }
    public String getPid() {
        return pid;
    }
    public void setPid(String pid) {
        this.pid = pid;
    }
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public String getTime() {
        return time;
    }
    public void setTime(String time) {
        this.time = time;
    }
}

```

User

```

package com.example.myapplication.Model;
public class User {
    private String name,phone,password,img,address;
    public User(){

    }

    public User(String name, String phone, String password, String img, String address) {
        this.name = name;
        this.phone = phone;
        this.password = password;
        this.img = img;
        this.address = address;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}

```

```

    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getImg() {
        return img;
    }
    public void setImg(String img) {
        this.img = img;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}

```

HomeFragment

```

package com.example.myapplication.Pokupatel.ui.home;

```

```

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProvider;

```

```

import com.example.myapplication.R;

```

```

public class HomeFragment extends Fragment {

```

```

    private HomeViewModel homeViewModel;

```

```

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        homeViewModel =
            new ViewModelProvider(this).get(HomeViewModel.class);
        View view = inflater.inflate(R.layout.fragment_home, null, false);
        final TextView textView = view.findViewById(R.id.text_home);
        homeViewModel.getText().observe(getViewLifecycleOwner(), new Observer<String>() {
            @Override
            public void onChanged(@Nullable String s) {
                textView.setText(s);
            }
        });
        return view;
    }
}

```

HomeViewModel

```

package com.example.myapplication.Pokupatel.ui.home;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
public class HomeViewModel extends ViewModel {
    private MutableLiveData<String> mText;

```

```

public HomeViewModel() {
    mText = new MutableLiveData<>();
    mText.setValue("This is home fragment");
}
public LiveData<String> getText() {
    return mText;
}
}

```

CunFinActivity

```

package com.example.myapplication.Pokupatel;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;

public class CunFinActivity extends AppCompatActivity {
    private EditText nEdTxt,pEdTxt,AEdTxt,cEdTxt;
    private Button confBtn;
    private String totalAmount="";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cun_fin);

        totalAmount=getIntent().getStringExtra("Bcero=");
        Toast.makeText(this, "Bcero= tr"+totalAmount, Toast.LENGTH_SHORT).show();

        confBtn=(Button) findViewById(R.id.confirm_btn);
        nEdTxt=(EditText)findViewById(R.id.y_name);
        pEdTxt=(EditText)findViewById(R.id.y_phone);
        AEdTxt=(EditText)findViewById(R.id.y_address);
        cEdTxt=(EditText)findViewById(R.id.y_city);

        confBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Chek();
            }
        });
    }

    private void Chek() {
        if(TextUtils.isEmpty(nEdTxt.getText().toString())){
            Toast.makeText(this, "Пожалуйста напишите полное имя...", Toast.LENGTH_SHORT).show();
        }
        else if(TextUtils.isEmpty(pEdTxt.getText().toString())){
            Toast.makeText(this, "Пожалуйста напишите свой номер...", Toast.LENGTH_SHORT).show();
        }
    }
}

```



```

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.appcompat.widget.Toolbar;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.myapplication.Admin.MaintainPrActivity;
import com.example.myapplication.Model.Products;
import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.rey.material.widget.FloatingActionButton;
import com.squareup.picasso.Picasso;

import ViewH.ProViewH;
import de.hdodenhof.circleimageview.CircleImageView;
import io.paperdb.Paper;

public class HomeActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    private DatabaseReference ProductsRef;
    private RecyclerView recyclerView;
    RecyclerView.LayoutManager layoutManager;

    private String type=" ";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        Intent intent=getIntent();
        Bundle bundle=intent.getExtras();
        if(bundle!=null){
            type=getIntent().getExtras().get("com/example/myapplication/Admin").toString();
        }

        ProductsRef = FirebaseDatabase.getInstance().getReference().child("Products");
        Paper.init(this);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        toolbar.setTitle("Home");
        setContentView(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);

```

```

fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(type.equals("com/example/myapplication/Admin")){
            Intent intent=new Intent(HomeActivity.this, KorzinaActivity.class);
            startActivity(intent);
        }
    }
});

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, null, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
View headerView = navigationView.getHeaderView(0);
TextView userNameTextView = headerView.findViewById(R.id.user_pr_name);
CircleImageView profileImageView = headerView.findViewById(R.id.pro_image);
if(!type.equals("com/example/myapplication/Admin")){
    userNameTextView.setText(Preval.curronlineUser.getName());
    Picasso.get().load(Preval.curronlineUser.getName()).placeholder(R.drawable.ava).into(profileImageView);
}
recyclerView = findViewById(R.id.recycle_menu);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);
}
@Override
protected void onStart()
{
    super.onStart();
    FirebaseRecyclerOptions<Products> options =
        new FirebaseRecyclerOptions.Builder<Products>()
            .setQuery(ProductsRef, Products.class)
            .build();
    FirebaseRecyclerAdapter<Products, ProViewH> adapter =
        new FirebaseRecyclerAdapter<Products, ProViewH>(options) {
            @Override
            protected void onBindViewHolder(@NonNull ProViewH holder, int position, @NonNull Products
model)
            {
                holder.textPrN.setText(model.getPname());
                holder.textPrOp.setText(model.getOpisan());
                holder.txtPrPrica.setText("Price = " + model.getPrice() + "$");
                Picasso.get().load(model.getImg()).into(holder.imageView);

                holder.itemView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        if(type.equals("com/example/myapplication/Admin")){
                            Intent intent=new Intent( HomeActivity.this, MaintainPrActivity.class);
                            intent.putExtra("put",model.getPid());
                            startActivity(intent);
                        }
                        else{
                            Intent intent=new Intent( HomeActivity.this, PrDetActivity.class);

```

```

        intent.putExtra("put",model.getPid());
        startActivity(intent);
    }
}
});
}
@NonNull
@Override
public ProViewH onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
{
    View root = LayoutInflater.from(parent.getContext()).inflate(R.layout.prod_item, null, false);
    ProViewH holder = new ProViewH(root);
    return holder;
}
};
recyclerView.setAdapter(adapter);
adapter.startListening();
}
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.home, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int id = item.getItemId();
    return super.onOptionsItemSelected(item);
}
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item)
{
    int id = item.getItemId();

    if (id == R.id.nav_car)
    {
        if(type.equals("com/example/myapplication/Admin")){
            Intent intent=new Intent(HomeActivity.this,KorzinaActivity.class);
            startActivity(intent);
        }

    }
    else if (id == R.id.nav_poisk)
    {
        if(type.equals("com/example/myapplication/Admin")){
            Intent intent=new Intent(HomeActivity.this, PoiskActivity.class);
            startActivity(intent);
        }
    }
    else if (id == R.id.nav_categ)
    {
        if(type.equals("com/example/myapplication/Admin")){

```

```

    }
}
else if (id == R.id.nav_setting)
{
    if(type.equals("com/example/myapplication/Admin")){
        Intent intent = new Intent(HomeActivity.this, SetActivity.class);
        startActivity(intent);
    }
}
else if (id == R.id.vyhod)
{
    if(type.equals("com/example/myapplication/Admin")){
        Paper.book().destroy();
        Intent intent = new Intent(HomeActivity.this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }
}
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}
}
}

```

LoginActivity

```

package com.example.myapplication.Pokupatel;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.myapplication.Admin.AdminCategActivity;
import com.example.myapplication.Model.User;
import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.rey.material.widget.CheckBox;

import io.paperdb.Paper;

public class LoginActivity extends AppCompatActivity {
    private EditText InputNumber, InputPassword;
    private Button LoginButton;
    private ProgressDialog loadingBaar;
    private String parentDatabName="User";
    private CheckBox chBoxRem;
    private TextView AdminLink,NotAdminLink,ForgPass;
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    LoginButton=(Button) findViewById(R.id.log_btn);
    InputPassword=(EditText) findViewById(R.id.password_input);
    InputNumber=(EditText) findViewById(R.id.number_input);
    AdminLink=(TextView) findViewById(R.id.admin_im);
    NotAdminLink=(TextView) findViewById(R.id.admin_not_im);
    ForgPass=findViewById(R.id.forg_password);
    loadingBaar=new ProgressDialog(this);

    chBoxRem=(CheckBox) findViewById(R.id.remember);
    Paper.init(this );
    LoginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            LoginUser();
        }
    });
    ForgPass.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(LoginActivity.this, ResetPassActivity.class);
            intent.putExtra("check","login");
            startActivity(intent);
        }
    });

    AdminLink.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            LoginButton.setText("Login com.example.myapplication.Admin");
            AdminLink.setVisibility(View.INVISIBLE);
            NotAdminLink.setVisibility(View.INVISIBLE);
            parentDatabName= "com/example/myapplication/Admin";
        }
    });
    NotAdminLink.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            LoginButton.setText("Login ");
            AdminLink.setVisibility(View.VISIBLE);
            NotAdminLink.setVisibility(View.VISIBLE);
            parentDatabName="User";
        }
    });
}

private void LoginUser() {
    String phone= InputNumber.getText().toString();
    String password= InputPassword.getText().toString();

    if (TextUtils.isEmpty(phone)){
        Toast.makeText(this,"Напишите свой номер",Toast.LENGTH_SHORT).show();
    }
    else if (TextUtils.isEmpty(password)){
        Toast.makeText(this,"Напишите свой пароль",Toast.LENGTH_SHORT).show();
    }
    else{
        loadingBaar.setTitle("Вход");
        loadingBaar.setMessage("Подождите, мы проверяем ваши данные...");
        loadingBaar.setCanceledOnTouchOutside(false);
        loadingBaar.show();
    }
}

```

```

        AllowAccessToAccount(phone,password );    }
    }

    private void AllowAccessToAccount(String phone, String password) {
        if(chBoxRem.isChecked()){
            Paper.book().write(Preval.UserPhoneKey,phone);
            Paper.book().write(Preval.UserPasswordKey,password);
        }
        final DatabaseReference RootRef;
        RootRef= FirebaseDatabase.getInstance().getReference();
        RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.child(parentDatabName).child(phone).exists()){
                    User userData=snapshot.child(parentDatabName).child(phone).getValue(User.class);
                    if(userData.getPhone().equals(phone)){
                        if(userData.getPassword().equals(password)){
                            if(parentDatabName.equals("com/example/myapplication/Admin")){
                                Toast.makeText(LoginActivity.this, "Вход в систему успешно!",
                                    Toast.LENGTH_SHORT).show();
                                loadingBaar.dismiss();
                                Intent intent=new Intent(LoginActivity.this, AdminCategActivity.class);
                                startActivity(intent);
                            }
                            else if(parentDatabName.equals("User")){
                                Toast.makeText(LoginActivity.this, "Вход в систему успешно!",
                                    Toast.LENGTH_SHORT).show();
                                loadingBaar.dismiss();

                                Intent intent=new Intent(LoginActivity.this, HomeActivity.class);
                                Preval.curronlineUser=userData;
                                startActivity(intent);
                            }
                        }
                    }
                    else{
                        loadingBaar.dismiss();
                        Toast.makeText(LoginActivity.this, "Пароль неверно!", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
        else{
            Toast.makeText(LoginActivity.this, phone+"не существует!", Toast.LENGTH_SHORT).show();
            loadingBaar.dismiss();
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {

    }
}
});
}
}
}

```

MainActivity

```

package com.example.myapplication.Pokupatel;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.example.myapplication.Model.User;
import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.example.myapplication.prodavec.ProdLoginActivity;
import com.example.myapplication.prodavec.ProdavecRegistrActivity;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import io.paperdb.Paper;

public class MainActivity extends AppCompatActivity {
    private Button nowButton, LoginButton;
    private ProgressDialog loadingBaar;
    private TextView sellerPro;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        nowButton=(Button) findViewById(R.id.regist_btn);
        LoginButton=(Button) findViewById(R.id.login_btn);
        sellerPro=(TextView) findViewById(R.id.seller_pro);
        loadingBaar=new ProgressDialog(this);
        Paper.init(this);
        LoginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        });
        sellerPro.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(MainActivity.this, ProdavecRegistrActivity.class);
                startActivity(intent);
            }
        });
        nowButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(MainActivity.this, RegistActivity.class);
                startActivity(intent);
            }
        });
        String UserPhoneKey=Paper.book().read(Preval.UserPhoneKey);
        String UserPasswordKey=Paper.book().read(Preval.UserPasswordKey);

        if(UserPhoneKey!=""&& UserPasswordKey!=""){
            if(!TextUtils.isEmpty(UserPhoneKey)&& !TextUtils.isEmpty(UserPasswordKey)){
                AllAccess(UserPhoneKey,UserPasswordKey);
                loadingBaar.setTitle("Вы уже вошли");
                loadingBaar.setMessage("Подождите, мы проверяем ваши данные...");
            }
        }
    }
}

```



```

        loadingBaar.setCanceledOnTouchOutside(false);
        loadingBaar.show();
    }
}
}

private void AllAccess(String phone, String password) {
    final DatabaseReference RootRef;
    RootRef= FirebaseDatabase.getInstance().getReference();

    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(snapshot.child("User").child(phone).exists()){
                User userData=snapshot.child("User").child(phone).getValue(User.class);
                if(userData.getPhone().equals(phone)){
                    if(userData.getPassword().equals(password)){
                        Toast.makeText(MainActivity.this, "Вы уже вошли ", Toast.LENGTH_SHORT).show();
                        loadingBaar.dismiss();

                        Intent intent=new Intent(MainActivity.this, HomeActivity.class);
                        startActivity(intent);
                    }
                    else{
                        loadingBaar.dismiss();
                        Toast.makeText(MainActivity.this, "Пароль неверно!", Toast.LENGTH_SHORT).show();
                    }
                }
            }
            else{
                Toast.makeText(MainActivity.this, phone+"не существует!", Toast.LENGTH_SHORT).show();
                loadingBaar.dismiss();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
}
}

```

PoiskActivity

```

package com.example.myapplication.Pokupatel;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;

import com.example.myapplication.Model.Products;
import com.example.myapplication.R;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

```

```

import com.squareup.picasso.Picasso;

import ViewH.ProViewH;

public class PoiskActivity extends AppCompatActivity {

    private Button poisk_btn;
    private EditText inputText;
    private RecyclerView poiskList;
    private String PoiskInput;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_poisk);

        inputText=findViewById(R.id.poisk_pr);
        poisk_btn=findViewById(R.id.poisk_btn);
        poiskList=findViewById(R.id.poisk_list);
        poiskList.setLayoutManager(new LinearLayoutManager(PoiskActivity.this));

        poisk_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PoiskInput=inputText.getText().toString();
                onStart();
            }
        });
    }
    @Override
    protected void onStart() {
        super.onStart();
        DatabaseReference reference= FirebaseDatabase.getInstance().getReference().child("Products");
        FirebaseRecyclerOptions<Products> options=new
        FirebaseRecyclerOptions.Builder<Products>().setQuery(reference.orderByChild("pname").startAt(PoiskInput),Produ
        cts.class).build();
        FirebaseRecyclerAdapter<Products, ProViewH>adapter=new FirebaseRecyclerAdapter<Products,
        ProViewH>(options) {
            @Override
            protected void onBindViewHolder(@NonNull ProViewH proViewH, int i, @NonNull Products products) {
                proViewH.textPrN.setText(products.getPname());
                proViewH.textPrOp.setText(products.getOpisan());
                proViewH.txtPrPrica.setText("Price = " + products.getPrice() + "$");
                Picasso.get().load(products.getImg()).into(products.imageView);
                proViewH.itemView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Intent intent=new Intent( PoiskActivity.this, PrDetActivity.class);
                        intent.putExtra("put",products.getPid());
                        startActivity(intent);
                    }
                });
            }
            @NonNull
            @Override
            public ProViewH onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
                View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.prod_item, parent, false);
                ProViewH proViewH = new ProViewH(view);
                return proViewH;
            }
        };
        poiskList.setAdapter(adapter);
        adapter.startListening();
    }
}

```

```

    }
}

```

RegistActivity

```

package com.example.myapplication.Pokupatel;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.example.myapplication.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.HashMap;

public class RegistActivity extends AppCompatActivity {
    private Button CreateAccountButton;
    private EditText InputName, InputPhoneNumber, InputPassword;
    private ProgressDialog loadingBaar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_regist);
        CreateAccountButton=(Button) findViewById(R.id.reg_btn);
        InputName=(EditText) findViewById(R.id.reg_name_input);
        InputPassword=(EditText) findViewById(R.id.reg_password_input);
        InputPhoneNumber=(EditText) findViewById(R.id.reg_number_input);
        loadingBaar=new ProgressDialog(this);
        CreateAccountButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CreateAccount();
            }
        });
    }

    private void CreateAccount(){
        String name= InputName.getText().toString();
        String phone= InputPhoneNumber.getText().toString();
        String password= InputPassword.getText().toString();

        if (TextUtils.isEmpty(name)){
            Toast.makeText(this,"Напишите свое имя",Toast.LENGTH_SHORT).show();
        }
        else if (TextUtils.isEmpty(phone)){
            Toast.makeText(this,"Напишите свой номер",Toast.LENGTH_SHORT).show();
        }
        else if (TextUtils.isEmpty(password)){
            Toast.makeText(this,"Напишите свой пароль",Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
    else {
        loadingBaar.setTitle("Регистрация");
        loadingBaar.setMessage("Подождите, мы проверяем ваши данные...");
        loadingBaar.setCanceledOnTouchOutside(false);
        loadingBaar.show();

        ValidatephoneNumber(name, phone, password);
    }
}

private void ValidatephoneNumber(String name, String phone, String password) {
    final DatabaseReference RootRef;
    RootRef= FirebaseDatabase.getInstance().getReference();

    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot datasnapshot) {
            if (!(datasnapshot.child("User").child(phone).exists())) {
                HashMap<String, Object> userdataMap = new HashMap<>();
                userdataMap.put("phone", phone);
                userdataMap.put("password", password);
                userdataMap.put("name", name);

                RootRef.child("User").child(phone).updateChildren(userdataMap)
                    .addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                Toast.makeText(RegistActivity.this, "Поздравляем, ваш аккаунт успешно
зарегистрирован!!!", Toast.LENGTH_SHORT).show();
                                loadingBaar.dismiss();

                                Intent intent = new Intent(RegistActivity.this, LoginActivity.class);
                                startActivity(intent);
                            } else {
                                loadingBaar.dismiss();
                                Toast.makeText(RegistActivity.this, "Соединение прервано!",
Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
            } else {
                Toast.makeText(RegistActivity.this, "Номер" + phone + "уже существует",
Toast.LENGTH_SHORT).show();
                loadingBaar.dismiss();
                Toast.makeText(RegistActivity.this, "Попробуйте ввести другой номер!",
Toast.LENGTH_SHORT).show();

                Intent intent = new Intent(RegistActivity.this, MainActivity.class);
                startActivity(intent);
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError databaseerror) {

        }
    });
}
}

```

```

package com.example.myapplication.Pokupatel;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.HashMap;

public class ResetPassActivity extends AppCompatActivity {
    private String check="";
    private TextView pageTitle,quesTitle;
    private EditText phoneN,quest1,quest2;
    private Button verify_btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reset_pass);
        check=getIntent().getStringExtra("check");
        pageTitle=findViewById(R.id.title_p);
        quesTitle=findViewById(R.id.title_ques);
        phoneN=findViewById(R.id.phone_reset);
        quest1=findViewById(R.id.ques_1);
        quest2=findViewById(R.id.ques_2);
        verify_btn=findViewById(R.id.verification);
    }
    @Override
    protected void onStart() {
        super.onStart();
        phoneN.setVisibility(View.GONE);
        if(check.equals("settings")){
            pageTitle.setText("set question");
            quesTitle.setText("Ответьте на следующие вопросы?");
            verify_btn.setText("Set");
            dispPreviousAnswers();

            verify_btn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    setAnswer();
                }
            });
        }
    }
}

```

```

else if(check.equals("login")){
    phoneN.setVisibility(View.VISIBLE);
    verify_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            verifyUser();
        }
    });
}
}

private void setAnswer(){
    String otv1=quest1.getText().toString().toLowerCase();
    String otv2=quest1.getText().toString().toLowerCase();

    if(quest1.equals("") && quest2.equals("")){
        Toast.makeText(ResetPassActivity.this, "Ответьте на следующие вопросы?",
        Toast.LENGTH_SHORT).show();
    }
    else {
        DatabaseReference reference =
        FirebaseDatabase.getInstance().getReference().child("User").child(Preval.curronlineUser.getPhone());
        HashMap<String, Object> userdataMap = new HashMap<>();
        userdataMap.put("otvet1", otv1);
        userdataMap.put("otvet2", otv2);
        reference.child("Вопросы безопасности").updateChildren(userdataMap).addOnCompleteListener(new
        OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(ResetPassActivity.this, "Ваши ответы на вопросы безопасности приняты",
                    Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(ResetPassActivity.this, HomeActivity.class);
                    startActivity(intent);
                }
            }
        });
    }
}

private void dispPreviousAnswers(){
    DatabaseReference reference =
    FirebaseDatabase.getInstance().getReference().child("User").child(Preval.curronlineUser.getPhone());
    reference.child("Вопросы безопасности").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.exists()){
                String otv1=snapshot.child("otvet1").getValue().toString();
                String otv2=snapshot.child("otvet2").getValue().toString();
                quest1.setText(otv1);
                quest2.setText(otv2);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}

private void verifyUser(){
    final String phone =phoneN.getText().toString();
    final String otv1=quest1.getText().toString().toLowerCase();
    final String otv2=quest1.getText().toString().toLowerCase();

```

```

        if(!phone.equals("") && otv1.equals("") && !otv2.equals("")){
            final DatabaseReference reference =
                FirebaseDatabase.getInstance().getReference().child("User").child(phone);

            reference.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    if(snapshot.exists()){
                        String mPhone=snapshot.child("phone").getValue().toString();

                        if(snapshot.hasChild("Вопросы безопасности")){
                            String otv1=snapshot.child("Вопросы Безопасности").child("otvet1").getValue().toString();
                            String otv2=snapshot.child("Вопросы Безопасности").child("otvet2").getValue().toString();

                            if(!otv1.equals(otv1)){
                                Toast.makeText(ResetPassActivity.this, "Первый ответ неправильно",
                                    Toast.LENGTH_SHORT).show();
                            }
                            else if(!otv2.equals(otv2)){
                                Toast.makeText(ResetPassActivity.this, "Второй ответ неправильно",
                                    Toast.LENGTH_SHORT).show();
                            }
                            else {
                                AlertDialog.Builder builder=new AlertDialog.Builder(ResetPassActivity.this);
                                builder.setTitle("Новый пароль");
                                final EditText newPass=new EditText(ResetPassActivity.this);
                                newPass.setHint("Напишите новый пароль");
                                builder.setView(newPass);

                                builder.setPositiveButton("Change", new DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface dialogInterface, int i) {
                                        if(!newPass.getText().toString().equals("")){

                                            reference.child("password").setValue(newPass.getText().toString()).addOnCompleteListener(new
                                                OnCompleteListener<Void>() {
                                                    @Override
                                                    public void onComplete(@NonNull Task<Void> task) {
                                                        if(task.isSuccessful()){
                                                            Toast.makeText(ResetPassActivity.this, "Пароль успешно изменено",
                                                                Toast.LENGTH_SHORT).show();
                                                            Intent intent = new Intent(ResetPassActivity.this, LoginActivity.class);
                                                            startActivity(intent);
                                                        }
                                                    }
                                                });
                                                    }
                                                }
                                            });
                                builder.setPositiveButton("cancel", new DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface dialogInterface, int i) {
                                        dialogInterface.cancel();
                                    }
                                });
                                builder.show();
                            }
                        }
                    }
                }
            });
        }
    }
}
else {
    Toast.makeText(ResetPassActivity.this, "Ошибка, данные не совпадают",

```

```

Toast.LENGTH_SHORT).show();
    }

    }
    else {
        Toast.makeText(ResetPassActivity.this, "Номер неправильно", Toast.LENGTH_SHORT).show();
    }

}

@Override
public void onCancelled(@NonNull DatabaseError error) {

}

});
}
else{
    Toast.makeText(this, "Заполните форму", Toast.LENGTH_SHORT).show();
}

}
}

```

SetActivity

```

package com.example.myapplication.Pokupatel;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.example.myapplication.Prevalents.Preval;
import com.example.myapplication.R;
import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.StorageTask;
import com.squareup.picasso.Picasso;
import com.theartofdev.edmodo.cropper.CropImage;

import java.util.HashMap;

import de.hdodenhof.circleimageview.CircleImageView;

public class SetActivity extends AppCompatActivity {
    private CircleImageView ava_set;

```



```

private EditText name,phone,address;
private TextView change_btn,close_btn,sav_btn;
private Button security_btn;
private Uri imgUri;
private String myUrl="";
private StorageTask uploadTask;
private StorageReference storageReferencePr;
private String chec="";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_set);
    storageReferencePr= FirebaseStorage.getInstance().getReference().child("Фото профиля");
    ava_set=(CircleImageView) findViewById(R.id.ava_set);
    name=(EditText) findViewById(R.id.name_settings);
    phone=(EditText) findViewById(R.id.phone_settings);
    address=(EditText) findViewById(R.id.adres_settings);
    change_btn=(TextView) findViewById(R.id.prof_izmen);
    close_btn=(TextView) findViewById(R.id.close_set);
    sav_btn=(TextView) findViewById(R.id.update_set);
    security_btn=findViewById(R.id.security_settings);
    UserInfoD(ava_set,name,phone,address);
    close_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });
    security_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(SetActivity.this, ResetPassActivity.class);
            intent.putExtra("check","settings");
            startActivity(intent);
        }
    });
    sav_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(chec.equals("click")){
                UserInfoSaved();
            }
            else {
                updateO();
            }
        }
    });
    change_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            chec="click";
            CropImage.activity(imgUri)
                .setAspectRatio(1,1)
                .start(SetActivity.this);
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

```

```

        if(requestCode==CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE &&
resultCode==RESULT_OK&& data!=null){
            CropImage.ActivityResult result=CropImage.getActivityResult(data);
            imgUri=result.getUri();
            ava_set.setImageURI(imgUri);
        }
        else {
            Toast.makeText(this, "Ошибка,попробуйте позже", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(SetActivity.this,SetActivity.class));
            finish();
        }
    }
}

private void updateOO() {
    DatabaseReference reference=FirebaseDatabase.getInstance().getReference().child("User");
    HashMap<String,Object> userMap= new HashMap<>();
    userMap.put("name",name.getText().toString());
    userMap.put("address",address.getText().toString());
    userMap.put("phone",phone.getText().toString());
    reference.child(Preval.curronlineUser.getPhone()).updateChildren(userMap);
    startActivity(new Intent(SetActivity.this, MainActivity.class));
    Toast.makeText(SetActivity.this, "Информация успешно обновлено!", Toast.LENGTH_SHORT).show();
    finish();
}

private void UserInfoSaved() {
    if(TextUtils.isEmpty(name.getText().toString())){
        Toast.makeText(this, "Напишите имя обязательно", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(address.getText().toString())){
        Toast.makeText(this, "Напишите адрес обязательно", Toast.LENGTH_SHORT).show();
    }
    else if(TextUtils.isEmpty(phone.getText().toString())){
        Toast.makeText(this, "Напишите номер обязательно", Toast.LENGTH_SHORT).show();
    }
    else if(chec.equals("click")){
        uploadImage();
    }
}

private void uploadImage() {
    final ProgressDialog progressDialog=new ProgressDialog(this);
    progressDialog.setTitle("Обновить аккаунт");
    progressDialog.setMessage("Подождите,мы обновляем ваш аккаунт.");
    progressDialog.setCanceledOnTouchOutside(false);
    progressDialog.show();
    if (imgUri!=null){
        final StorageReference fileR=storageReferencePr.child(Preval.curronlineUser.getPhone()+".jpg");
        uploadTask=fileR.putFile(imgUri);
        uploadTask.continueWithTask(new Continuation() {
            @Override
            public Object then(@NonNull Task task) throws Exception {
                if(!task.isSuccessful()){
                    throw task.getException();
                }
                return fileR.getDownloadUrl();
            }
        })
        addOnCompleteListener(new OnCompleteListener<Uri>() {
            @Override
            public void onComplete(@NonNull Task<Uri> task) {
                if(task.isSuccessful()){
                    Uri downloadUrl=task.getResult();
                    DatabaseReference reference=FirebaseDatabase.getInstance().getReference().child("User");
                    HashMap<String,Object> userMap= new HashMap<>();
                    userMap.put("name",name.getText().toString());

```

```

        userMap.put("address",address.getText().toString());
        userMap.put("phone",phone.getText().toString());
        userMap.put("img",myUrl);
        reference.child(Preval.currronlineUser.getPhone()).updateChildren(userMap);
        progressDialog.dismiss();
        startActivity(new Intent(SetActivity.this, HomeActivity.class));
        Toast.makeText(SetActivity.this, "Информация успешно обновлено!",
        Toast.LENGTH_SHORT).show();
        finish();
    }
    else {
        progressDialog.dismiss();
        Toast.makeText(SetActivity.this, "Ошибка...", Toast.LENGTH_SHORT).show();
    }
}
});
}
else {
    Toast.makeText(this, "Не удалось загрузить фото...", Toast.LENGTH_SHORT).show();
}
}
private void UserInfoD(CircleImageView ava_set, EditText name, EditText phone, EditText address) {
    DatabaseReference UserP =
    FirebaseDatabase.getInstance().getReference().child("User").child(Preval.currronlineUser.getPhone());
    UserP.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(snapshot.exists()){
                if(snapshot.child("img").exists()){
                    String img=snapshot.child("img").getValue().toString();
                    String namee=snapshot.child("name").getValue().toString();
                    String phonee=snapshot.child("key").getValue().toString();
                    String adress=snapshot.child("adres").getValue().toString();
                    Picasso.get().load("img").into(ava_set);
                    name.setText(namee);
                    phone.setText(phonee);
                    address.setText(adress);
                }
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
}
}

```

activity_admin

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="com.example.myapplication.Admin.AdminActivity">
    <ImageView
        android:id="@+id/product_src"
        android:layout_width="350dp"
        android:layout_height="200dp"
        android:cropToPadding="false"
        android:scaleType="centerCrop"
        android:src="@drawable/camera"
        android:background="@drawable/input_dizain"
    >

```

```

        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"/>

<EditText
    android:id="@+id/Pro_name"
    android:layout_width="350dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/product_src"
    android:layout_marginLeft="35dp"
    android:layout_marginTop="21dp"
    android:layout_marginRight="35dp"
    android:background="@drawable/input_dizain"
    android:hint="Имя продукта"
    android:inputType="textMultiLine"
    android:padding="20dp"
/>

<EditText
    android:id="@+id/opisan_prod"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/Pro_name"
    android:layout_marginLeft="35dp"
    android:layout_marginTop="21dp"
    android:layout_marginRight="35dp"
    android:background="@drawable/input_dizain"
    android:hint="Описание продукта"
    android:inputType="textMultiLine"
    android:padding="20dp"
/>

<EditText
    android:id="@+id/price_prod"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/opisan_prod"
    android:layout_marginLeft="35dp"
    android:layout_marginTop="21dp"
    android:layout_marginRight="35dp"
    android:background="@drawable/input_dizain"
    android:hint="Цена продукта"
    android:inputType="textMultiLine"
    android:padding="20dp"
/>

<Button
    android:id="@+id/add_product"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/price_prod"
    android:layout_marginLeft="35dp"
    android:layout_marginTop="21dp"
    android:layout_marginRight="35dp"
    android:background="@drawable/input_dizain"
    android:padding="10dp"
    android:text="Загрузить"
    android:textAllCaps="false"
    android:textSize="18sp"/>
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

activity_admin_categ

```

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"
tools:context="com.example.myapplication.Admin.AdminCategActivity">
<TextView
    android:id="@+id/slogan_cat"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="27dp"
    android:text="Загрузить новый продукт"
    android:textAlignment="center"
    android:textSize="26sp"
    android:textColor="@color/orange"
    android:textStyle="bold|italic"
/>

<LinearLayout
    android:id="@+id/ll_categ"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/slogan_cat"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="45dp"
>

<ImageView
    android:id="@+id/tolstovka"
    android:layout_width="80dp"
    android:layout_height="90dp"
    android:src="@drawable/tolst"></ImageView>

<ImageView
    android:id="@+id/futbolka"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:src="@drawable/footbolka"></ImageView>

<ImageView
    android:id="@+id/dress"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:src="@drawable/dress"></ImageView>
<ImageView
    android:id="@+id/shoes"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:src="@drawable/shoes"></ImageView>

</LinearLayout>
<LinearLayout
    android:id="@+id/ll_categ2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ll_categ"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:gravity="center"

```

```

        android:layout_marginTop="45dp"
    >

    <ImageView
        android:id="@+id/mobile"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:src="@drawable/mobile"></ImageView>

    <ImageView
        android:id="@+id/laptop"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/laptop"></ImageView>

    <ImageView
        android:id="@+id/nauwnik"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/headph"></ImageView>
    <ImageView
        android:id="@+id/watch"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/watch"></ImageView>

</LinearLayout>
<LinearLayout
    android:id="@+id/ll_categ3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ll_categ2"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="45dp"
    >

    <ImageView
        android:id="@+id/hat"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:src="@drawable/hat"></ImageView>

    <ImageView
        android:id="@+id/coat"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/coat"></ImageView>

    <ImageView
        android:id="@+id/books"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/books"></ImageView>
    <ImageView
        android:id="@+id/bag"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/bag"></ImageView>

</LinearLayout>

```

```

<Button
    android:id="@+id/maintain_btn"
    android:layout_width="match_parent"
    android:layout_height="42dp"
    android:layout_below="@id/ll_categ3"

    android:layout_marginStart="20dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="30dp"
    android:background="@drawable/input_dizain"
    android:text="Поддерживать"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="20dp" />

```

```

<Button
    android:id="@+id/check_btn"
    android:layout_width="match_parent"
    android:layout_height="42dp"
    android:layout_above="@+id/maintain_btn"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="98dp"
    android:layout_marginEnd="30dp"
    android:layout_marginBottom="105dp"
    android:background="@drawable/input_dizain"
    android:text="Проверить новый заказ"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="20dp" />

```

```

<Button
    android:id="@+id/logout_order_btn"
    android:layout_width="match_parent"
    android:layout_height="42dp"
    android:layout_below="@id/maintain_btn"
    android:background="@drawable/input_dizain"
    android:layout_marginStart="20dp"
    android:layout_marginTop="90dp"
    android:layout_marginEnd="29dp"
    android:layout_marginBottom="20dp"
    android:text="Выход"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="20dp" />

```

```

</RelativeLayout>

```

activity_admin_new_order

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="com.example.myapplication.Admin.AdminNewOrderActivity">

```

```

<RelativeLayout
    android:id="@+id/re_ley2"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_700"
        android:layout_alignParentTop="true">
        <TextView

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Новый заказ"
            android:textSize="25dp"
            android:textColor="@color/white"
            android:layout_centerVertical="true"
            android:layout_marginTop="10dp"/>
    </RelativeLayout>

```

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/order_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/re_ley2"
    android:layout_alignParentTop="true"/>

```

```

</RelativeLayout>

```

activity_admin_user_pr

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="com.example.myapplication.Admin.AdminUserPrActivity">

```

```

    <RelativeLayout
        android:id="@+id/re_ley3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_700"
        android:layout_alignParentTop="true">
        <TextView

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Продукты"
            android:textSize="25dp"
            android:textColor="@color/white"
            android:layout_centerVertical="true"
            android:layout_marginTop="10dp"/>
    </RelativeLayout>

```

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/product_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/re_ley3"
    android:layout_alignParentTop="true"/>

```

```

</RelativeLayout>

```

activity_cun_fin


```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".Pokupatel.CunFinActivity">
    <TextView
        android:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Подтвердите детали"
        android:textSize="25dp"
        android:textColor="@color/orange"
        android:layout_margin="25dp"
        android:textAlignment="center"
        android:textStyle="bold"/>
    <EditText
        android:id="@+id/y_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:hint="Ф.И.О"/>

    <EditText
        android:id="@+id/y_phone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/y_name"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:hint="Номер телефона"/>
    <EditText
        android:id="@+id/y_address"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/y_phone"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:hint="Ваш адрес"/>
    <EditText
        android:id="@+id/y_city"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/y_address"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:hint="Город"/>

    <Button
        android:id="@+id/confirm_btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_margin="30dp"
        android:background="@drawable/input_dizain"

```

```

        android:text="Подтвердить"
        android:textSize="20dp"/>

```

```

</RelativeLayout>

```

activity_home

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout 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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <include
            layout="@layout/app_bar_main"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <com.google.android.material.navigation.NavigationView
            android:id="@+id/nav_view"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:fitsSystemWindows="true"
            app:headerLayout="@layout/nav_header_main"
            app:menu="@menu/activity_main_drawer" />
    </FrameLayout>
</androidx.drawerlayout.widget.DrawerLayout>

```

activity_korzina

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".Pokupatel.KorzinaActivity">
    <RelativeLayout
        android:id="@+id/re_ley"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_700"
        android:layout_alignParentTop="true">
        <TextView
            android:id="@+id/title_p"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Корзина"
            android:textColor="@color/white"
            android:textAlignment="center" />
    </RelativeLayout>

```

```

<TextView
    android:id="@+id/message1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/purple_500"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    android:textAlignment="center"
    android:layout_centerVertical="true"
    android:text="Поздравляем,ваш окончательный заказ был успешно выполнен. Скоро вы получите свой заказ...."
    android:textSize="20dp"
    android:visibility="gone"
/>

```

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/c_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/re_ley"
    android:layout_above="@+id/next"
    android:layout_alignParentTop="true"/>
<Button
    android:id="@+id/next"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="@color/purple_500"
    android:text="Следующий"
    android:textColor="@color/white"
    android:textSize="20dp"
    android:layout_margin="10dp"/>

```

```

</RelativeLayout>

```

activity_login

```

<?xml version="1.0" encoding="utf-8"?>
<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="@color/white"
    tools:context=".Pokupatel.LoginActivity">

    <ImageView
        android:id="@+id/app_logo"
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="200dp"
        android:src="@drawable/logo"
    />

    <EditText
        android:id="@+id/number_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/app_logo"
        android:background="@drawable/input_dizain"

```

```

        android:padding="20dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:layout_marginTop="20dp"
        android:hint="Номер телефона"
        android:inputType="number"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold"></EditText>
<EditText
    android:id="@+id/password_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/number_input"
    android:background="@drawable/input_dizain"
    android:padding="20dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="6dp"
    android:hint="Пароль"
    android:inputType="textPassword"
    android:textColor="@color/black"
    android:textSize="17sp"
    android:textStyle="bold"></EditText>
<LinearLayout
    android:id="@+id/ll_1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_below="@+id/password_input"
    android:layout_marginTop="5dp">
    <com.rey.material.widget.CheckBox
        android:id="@+id/remember"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.CompoundButton.CheckBox"
        android:text="Запомнить"
        android:textColor="@color/orange"
        app:cbd_strokeColor="@color/orange"
        android:gravity="center_vertical"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginStart="17dp"/>

    <TextView
        android:id="@+id/forg_password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Забыли пароль?"
        android:textColor="@color/orange"
        android:textStyle="bold"
        android:layout_marginLeft="120dp"

    />

</LinearLayout>
<Button
    android:id="@+id/log_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ll_1"
    android:layout_marginLeft="15dp"

```

```

        android:layout_marginRight="15dp"
        android:background="@drawable/input_dizain"
        android:padding="20dp"
        android:textAllCaps="false"
        android:textSize="18sp"
        android:text="Вход">

</Button>

<TextView
    android:id="@+id/admin_im"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Я админ!"
    android:layout_alignParentEnd="true"
    android:layout_below="@+id/log_btn"
    android:textColor="@color/orange"
    android:textStyle="bold"
    android:layout_marginLeft="120dp"
    android:layout_marginEnd="23dp"
    android:textSize="20sp"

/>
<TextView
    android:id="@+id/admin_not_im"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Я не админ!"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/log_btn"
    android:textColor="@color/black"
    android:textStyle="bold"
    android:layout_marginLeft="120dp"
    android:layout_marginEnd="23dp"
    android:textSize="20sp"
    android:layout_marginStart="35dp"
    android:visibility="invisible"
/>
</RelativeLayout>

```

activity_main

```

<?xml version="1.0" encoding="utf-8"?>
<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="@color/white"
    tools:context=".Pokupatel.MainActivity"
    >

<ImageView
    android:id="@+id/app_logo"
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="60dp"
    android:src="@drawable/logo"
    />
<TextView
    android:id="@+id/app_slogan"
    android:layout_width="220dp"

```

```

        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="180dp"
        android:layout_marginEnd="88dp"
        android:text="Добро пожаловать!"
        android:textAlignment="center"
        android:textColor="@color/purple_700"
        android:textSize="22sp"
        android:textStyle="bold|italic"></TextView>
<Button
    android:id="@+id/login_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@id/seller_pro"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:background="@drawable/buttonss"
    android:padding="20dp"
    android:textAllCaps="false"
    android:textSize="18sp"
    android:text="Уже есть аккаунт? Вход">

</Button>
<Button
    android:id="@+id/regist_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="false"
    android:layout_above="@+id/login_btn"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:background="@drawable/input_dizain"
    android:padding="20dp"
    android:textAllCaps="false"
    android:textSize="18sp"
    android:text="Нет аккаунт? Регистрация"/>

<TextView
    android:id="@+id/seller_pro"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Хочешь стать продавцом?"
    android:textSize="20dp"
    android:textColor="@color/orange"
    android:textStyle="bold"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_marginRight="18dp"
    android:layout_marginBottom="10dp"/>

</RelativeLayout>

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

```

activity_poisk

```

android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Pokupatel.PoiskActivity">

```

```

<RelativeLayout
    android:id="@+id/re_ley5"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@color/purple_500"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true">

```

```

<com.rey.material.widget.EditText
    android:id="@+id/poisk_pr"
    android:layout_width="260dp"
    android:layout_height="wrap_content"
    android:hint="Поиск"
    android:textColorHint="@color/purple_200"
    android:textSize="16dp"
    android:background="@color/white"
    android:padding="9dp"
    android:textStyle="bold"
    android:layout_marginStart="5dp"
    android:layout_centerVertical="true"/>

```

```

<Button
    android:id="@+id/poisk_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_marginLeft="49dp"
    android:layout_marginEnd="14dp"

    android:layout_toRightOf="@+id/poisk_pr"
    android:text="Поиск"
    android:textSize="13dp"
    android:textStyle="bold"
    android:background="@drawable/input_dizain"
/>

```

```

</RelativeLayout>
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/poisk_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/re_ley5"
></androidx.recyclerview.widget.RecyclerView>

```

```

</RelativeLayout>

```

activity_regist

```

<?xml version="1.0" encoding="utf-8"?>
<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="@color/white"
    tools:context=".Pokupatel.RegistActivity">
    <ImageView
        android:id="@+id/reg_app_logo"

```

```

        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="190dp"
        android:src="@drawable/logo"
    />
    <EditText
        android:id="@+id/reg_name_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/reg_app_logo"
        android:background="@drawable/input_dizain"
        android:padding="20dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:layout_marginTop="10dp"
        android:hint="Имя"
        android:inputType="text"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold"></EditText>

    <EditText
        android:id="@+id/reg_number_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/reg_name_input"
        android:background="@drawable/input_dizain"
        android:padding="20dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:layout_marginTop="10dp"
        android:hint="Номер телефона"
        android:inputType="number"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold"></EditText>
    <EditText
        android:id="@+id/reg_password_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/reg_number_input"
        android:background="@drawable/input_dizain"
        android:padding="20dp"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:layout_marginTop="10dp"
        android:hint="Пароль"
        android:inputType="textPassword"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold"></EditText>

    <Button
        android:id="@+id/reg_btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/reg_password_input"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/input_dizain"

```



```

        android:padding="20dp"
        android:textAllCaps="false"
        android:textSize="18sp"
        android:text="Регистрация">

</Button>
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".Pokupatel.ResetPassActivity">

    <RelativeLayout
        android:id="@+id/re_ley6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_700"
        android:layout_alignParentTop="true">
        <TextView
            android:id="@+id/title_p"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Сборсить пароль"
            android:textColor="@color/white"
            android:gravity="center"
            android:textStyle="bold"
            android:textSize="20sp"/>
    </RelativeLayout>

    <EditText
        android:id="@+id/phone_reset"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Напишите свой номер"
        android:background="@drawable/input_dizain"
        android:layout_below="@id/re_ley6"
        android:padding="14dp"
        android:layout_margin="15dp"/>
    <TextView
        android:id="@+id/title_ques"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ответьте на следующие вопросы?"
        android:textColor="@color/orange"
        android:layout_below="@id/phone_reset"
        android:gravity="center"
        android:textSize="30dp"
        android:layout_marginTop="15dp"/>
    <EditText
        android:id="@+id/ques_1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Напишите свой адрес"
        android:background="@drawable/input_dizain"
        android:layout_below="@id/title_ques"
        android:padding="16dp"
        android:layout_margin="15dp"/>

```

activity_reset_pass

```

<EditText
    android:id="@+id/ques_2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Ф.И.О на странице"
    android:background="@drawable/input_dizain"
    android:layout_below="@id/ques_1"
    android:padding="16dp"
    android:layout_margin="15dp"/>
<Button
    android:id="@+id/verification"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/input_dizain"
    android:layout_below="@id/ques_2"
    android:padding="14dp"
    android:layout_margin="15dp"
    android:text="Подтвердить"
    android:textStyle="bold"
    android:textSize="20dp"/>

```

```

</RelativeLayout>

```

activity_set

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".Pokupatel.SetActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appbar_set"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="@color/purple_700"
        >
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar_set"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <TextView
                    android:id="@+id/close_set"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Закрыть"
                    android:textColor="@color/white"
                    android:textStyle="bold"
                    android:textSize="16sp"
                    android:layout_alignParentTop="true"
                    />
                <TextView
                    android:id="@+id/update_set"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Обновить"
                    android:textColor="@color/white"
                    android:textStyle="bold"

```

```

        android:layout_marginRight="10dp"
        android:textSize="16sp"
        android:layout_alignParentEnd="true"
    />
</RelativeLayout>
</androidx.appcompat.widget.Toolbar>
</com.google.android.material.appbar.AppBarLayout>

```

```

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/ava_set"
    android:layout_width="130dp"
    android:layout_height="130dp"
    android:src="@drawable/prof"
    android:layout_below="@id/appbar_set"
    android:layout_marginTop="20dp"
    android:layout_centerHorizontal="true"
/>

```

```

<TextView
    android:id="@+id/prof_izmen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/ava_set"
    android:layout_alignParentEnd="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="124dp"
    android:layout_marginRight="10dp"
    android:text="Изменить профиль"
    android:textColor="@color/orange"
    android:textSize="16sp"
    android:textStyle="bold"
/>

```

```

<EditText
    android:id="@+id/phone_settings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/prof_izmen"
    android:hint="Номер телефона"
    android:layout_margin="15dp"
    android:padding="10dp"
/>

```

```

<EditText
    android:id="@+id/name_settings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/phone_settings"
    android:hint="Ф.И.О"
    android:layout_margin="15dp"
    android:padding="10dp"
/>

```

```

<EditText
    android:id="@+id/adres_settings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/name_settings"
    android:hint="Ваш адрес"
    android:layout_margin="15dp"
    android:padding="10dp"
/>

```

```

<Button

```

```

        android:id="@+id/security_settings"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/adres_settings"
        android:hint="Сохранить"
        android:background="@drawable/input_dizain"
        android:layout_margin="15dp"
        android:padding="10dp"
        android:textColor="@color/white"
    />
</RelativeLayout>

```

c_item_layout

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:cardElevation="20dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/purple_700"
        android:padding="10dp">

        <TextView
            android:id="@+id/item_prN"
            android:layout_width="160dp"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_marginLeft="5dp"
            android:text="Имя продукта:"
            android:textColor="@color/white"
            android:textSize="14dp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/item_prQ"
            android:layout_width="160dp"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="5dp"

            android:layout_marginRight="5dp"
            android:text="Количество продукта:"
            android:textAlignment="viewEnd"
            android:textColor="@color/white"
            android:textSize="14dp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/item_prP"
            android:layout_width="160dp"
            android:layout_height="wrap_content"
            android:layout_below="@+id/item_prN"
            android:layout_centerHorizontal="true"
            android:layout_marginLeft="5dp"
            android:layout_marginTop="10dp"
            android:text="Цена продукта:"
            android:textAlignment="viewEnd"
            android:textColor="@color/white"

```

```

        android:textSize="14dp"
        android:textStyle="bold" />
    </RelativeLayout>

</androidx.cardview.widget.CardView>

```

Category

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="com.example.myapplication.Admin.AdminCategActivity">
    <RelativeLayout
        android:id="@+id/re_ley"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple_700"
        android:layout_alignParentTop="true">
        <TextView
            android:id="@+id/title_p"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Категории"
            android:textSize="19dp"
            android:textColor="@color/white"
            android:textAlignment="center" />
        </RelativeLayout>

    <LinearLayout
        android:id="@+id/ll_categ"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/slogan_cat"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:gravity="center"
        android:layout_marginTop="45dp"
    >

    <ImageView
        android:id="@+id/tolstovka"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:src="@drawable/tolst"></ImageView>

    <ImageView
        android:id="@+id/futbolka"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/footballka"></ImageView>

    <ImageView
        android:id="@+id/dress"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/dress"></ImageView>
    <ImageView
        android:id="@+id/shoes"
        android:layout_width="80dp"

```

```

        android:layout_height="80dp"
        android:src="@drawable/shoes"></ImageView>

</LinearLayout>
<LinearLayout
    android:id="@+id/ll_categ2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ll_categ"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="45dp"
    >

    <ImageView
        android:id="@+id/mobile"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:src="@drawable/mobile"></ImageView>

    <ImageView
        android:id="@+id/laptop"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/laptop"></ImageView>

    <ImageView
        android:id="@+id/nauwnik"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/headph"></ImageView>
    <ImageView
        android:id="@+id/watch"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/watch"></ImageView>
</LinearLayout>
<LinearLayout
    android:id="@+id/ll_categ3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ll_categ2"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="45dp"
    >

    <ImageView
        android:id="@+id/hat"
        android:layout_width="80dp"
        android:layout_height="90dp"
        android:src="@drawable/hat"></ImageView>
    <ImageView
        android:id="@+id/coat"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/coat"></ImageView>
    <ImageView
        android:id="@+id/books"
        android:layout_width="80dp"
        android:layout_height="80dp"

```

```

        android:src="@drawable/books"></ImageView>
    <ImageView
        android:id="@+id/bag"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/bag"></ImageView>
    </LinearLayout>
</RelativeLayout>

```

content_main

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main">

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycle_menu"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical"
    />
</RelativeLayout>

```

fragment_home

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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"
    tools:context=".Pokupatel.ui.home.HomeFragment">

    <TextView
        android:id="@+id/text_home"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@color/purple_700"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <de.hdodenhof.circleimageview.CircleImageView
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/profile_image"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:src="@drawable/prof"
        app:civ_border_width="2dp"
        android:layout_gravity="center"
        app:civ_border_color="#FF000000"/>
    <TextView
        android:id="@+id/user_pr_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:textSize="20sp"
        android:layout_marginLeft="4dp"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        android:textStyle="bold"
        android:gravity="center"
        android:text="Имя пользователя"/>
</LinearLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_margin="10dp"
        android:background="@color/purple_500">
        <TextView
            android:id="@+id/number_o"
            android:layout_width="160dp"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginStart="5dp"
            android:layout_marginEnd="217dp"
            android:layout_marginRight="5dp"
            android:text="Номер Телефона"
            android:textAlignment="viewEnd"
            android:textColor="@color/white"
            android:textSize="14dp"

```



```

        android:textStyle="bold" />
<TextView
    android:id="@+id/price_total_o"
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:text="Бцero"
    android:textColor="@color/white"
    android:textAlignment="viewEnd"
    android:textSize="14dp"
    android:layout_below="@+id/item_prN"
    android:layout_marginLeft="5dp"
    android:textStyle="bold"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"/>
<TextView
    android:id="@+id/address_o"
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:text="Ваш город, адрес"
    android:textColor="@color/white"
    android:textAlignment="viewEnd"
    android:textSize="14dp"
    android:layout_below="@+id/price_total_o"
    android:layout_marginLeft="5dp"
    android:textStyle="bold"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"/>
<TextView
    android:id="@+id/Data_time_o"
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:text="Дата и время"
    android:textColor="@color/white"
    android:textAlignment="viewEnd"
    android:textSize="14dp"
    android:layout_below="@+id/address_o"
    android:layout_marginLeft="5dp"
    android:textStyle="bold"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"/>

<Button
    android:id="@+id/show_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/Data_time_o"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:textSize="15dp"
    android:text="Показать заказанные продукты"
    android:background="@color/black"
    />
</RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>

```

Colors

```

    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="orange">#FF4500</color>
</resources>

```

Styles

```

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

    <!-- Base application theme. -->
    <style name="AppTheme" parent="@style/Theme.MyApplication">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>

```

build.gradle

```

buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:4.1.1"
        classpath 'com.google.gms:google-services:4.3.5'
    }
}

```

```

allprojects {
    repositories {
        google()
        jcenter()
    }
}

```

```

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

build.gradle(App)

```

plugins {
    id 'com.android.application'
}

```

```

android {
    compileSdkVersion 28

```

```

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 24
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"

```

```

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

```

```

    buildTypes {

```

```

        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation platform('com.google.firebase:firebase-bom:26.4.0')
    implementation 'com.firebaseui:firebase-ui-database:3.2.2'
    implementation 'com.google.firebase:firebase-core:11.8.0'
    implementation 'com.google.firebase:firebase-database:11.8.0'
    implementation 'com.google.firebase:firebase-auth:11.8.0'
    implementation 'com.google.firebase:firebase-storage:11.8.0'
    implementation 'com.squareup.picasso:picasso:2.71828'
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'com.github.rey5137:material:1.2.5'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'io.paperdb:paperdb:2.6'
    implementation 'com.theartofdev.edmodo:android-image-cropper:2.8.+'
    implementation 'com.cepheuen.elegant-number-button:lib:1.0.2'
    implementation 'com.android.support:design:28.0.0'
    implementation 'androidx.navigation:navigation-fragment:2.3.2'
    implementation 'androidx.navigation:navigation-ui:2.3.2'
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.2.0'
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.2.0'
    implementation 'de.hdodenhof:circleimageview:2.2.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
}

apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

```