**Қазақстан Республикасы Ғылым және жоғары білім министрлігі**

**Қожа Ахмет Ясауи атындағы Халықаралық қазақ-түрік университеті**

**«Қорғауға жіберілді»**

«Компьютерлік инженерия»

кафедра меңгерушісі

\_\_\_\_\_\_\_\_\_\_\_\_\_ техн.ғ.к., доцент м.а. **Казбекова Г.Н.**

**ДИПЛОМДЫҚ ЖОБА**

**Тақырыбы: «Денсаулық сақтау саласында консультация берудің Android жүйесін әзірлеу» кафедрасының автоматтандырылған ақпараттық жүйесін әзірлеу және жобалау**

**6В06151 – «Ақпараттық жүйелер»**

Орындағандар: **Тохтамуратов Д.Ш.,**

**Бектембаева С.Ф.**

Ғылыми жетекші:

PhD, аға оқытушы **Жунисов Н.М.**

**Кентау 2023**

**Аннотация**

Проект по разработке системы Android для консультирования по вопросам здравоохранения представляет собой приложение, созданное с целью предоставления пользователю доступа к медицинским консультациям и информации о здоровье. Приложение обладает функционалом онлайн-чата с медицинскими специалистами, возможностью записи на прием к врачу, доступом к полезным статьям и советам по уходу за здоровьем. Основные возможности включают в себя удобный интерфейс, защищенную передачу данных и персонализированный подход к каждому пользователю. Проект направлен на улучшение доступности медицинских услуг и повышение здоровья населения через использование современных технологий мобильных приложений.

**Annotation**

The Android project development for healthcare consultation is an application designed to provide users with access to medical consultations and health information. The app features an online chat function with medical specialists, appointment booking with doctors, access to useful articles and health tips. Key features include a user-friendly interface, secure data transmission, and a personalized approach to each user. The project aims to improve the accessibility of medical services and enhance public health through the use of modern mobile application technologies.

**Özet**

Sağlık danışmanlığı için Android projesi geliştirme, kullanıcılara tıbbi danışmanlıklara ve sağlık bilgilerine erişim sağlamak amacıyla tasarlanmış bir uygulamadır. Uygulama, tıbbi uzmanlarla çevrimiçi sohbet özelliği, doktor randevusu alma, yararlı makalelere ve sağlık ipuçlarına erişim sunar. Temel özellikler arasında kullanıcı dostu arayüz, güvenli veri iletimi ve her kullanıcıya özgü bir yaklaşım bulunmaktadır. Proje, modern mobil uygulama teknolojileri kullanılarak tıbbi hizmetlere erişimin iyileştirilmesini ve halk sağlığının artırılmasını amaçlamaktadır.

**МАЗМҰНЫ**

|  |  |  |
| --- | --- | --- |
|  | **КІРІСПЕ** | 4 |
|  |  |  |
| **1** | **МОБИЛЬДІ ҚОСЫМШАНЫ ҚҰРУДЫҢ ТЕОРИЯЛЫҚ НЕГІЗДЕРІ** | 6 |
| 1.1 | Мобильді қосымшалардың түсінігі және жіктелуі | 6 |
| 1.2 | Бағдарламалық құралдарды таңдау | 8 |
| 1.3 | Android бағдарламаларын жасауға арналған бағдарлама тілдері | 13 |
| 1.4 | Android бағдарламаларын жасау үшін қолданылатын шеңберлер | 16 |
|  |  |  |
| **2** | **ЕМХАНАНЫҢ МОБИЛЬДІ ҚОСЫМШАСЫН ЖОБАЛАУ ЖӘНЕ ӘЗІРЛЕУ** | 22 |
| 2.1 | Қосымшаның қызметін жобалау | 22 |
| 2.2 | Қолданба құрылымын және мәліметтер базасын әзірлеу | 24 |
| 2.3 | Денсаулық сақтау мәселелері бойынша кеңес беру үшін мобильді қосымшаны әзірлеу | 29 |
| 2.4 | Мобильді қосымшаны құруға арналған бағдарламалық жасақтама | 37 |
|  |  |  |
|  | **ҚОРЫТЫНДЫ** | 43 |
|  |  |  |
|  | **ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ** | 44 |
|  |  |  |
|  |
|  |  |  |

**КІРІСПЕ**

Бүгінгі таңда мобильді қосымшалар цифрлық дәуірдің ажырамас бөлігіне айналды, адамдардың күнделікті өмірінде маңызды рөл атқарды. Олар жеке пайдалану, ойын-сауық, бизнес және Мемлекеттік қызметтерді қоса алғанда, әртүрлі салаларда қолданылады.

Мобильді қосымшалардың негізгі қосымшаларына ақпарат іздеу, сауда жасау, жаңалықтар оқу, оқу және жұмыс іздеу кіреді. Іскери қосымшалар компанияларға жаңа клиенттерді тартуға, имиджін жақсартуға және тиімді жарнама жасауға көмектеседі. Олар сондай-ақ мемлекеттік қызметтерге және қызметкерлер туралы ақпаратқа қол жеткізуді қамтамасыз етеді. Мобильді қосымшалар байланыс пен ақпаратқа қол жеткізудің маңызды құралына айналды, бұл оларды біздің күнделікті өміріміздің ажырамас бөлігіне айналдырды.

Бұл зерттеудің өзектілігі Android платформасында дамудың соңғы технологияларын қолдана отырып, денсаулық сақтау мәселелері бойынша кеңес беру үшін мобильді қосымшаны құру болып табылады. Бұл қосымша медициналық мекемелермен өзара әрекеттесуді жақсартуға бағытталған, бұл пайдаланушыларға мәліметтер базасына ыңғайлы қол жетімділік пен кеңес алуға мүмкіндік береді.

**Зерттеудің мақсаты мен міндеттері:** Android Studio мүмкіндіктерін пайдалана отырып, денсаулық сақтау бойынша кеңес беру үшін мобильді қосымшаны әзірлеу. Міндеттерге Денсаулық сақтау мекемелері туралы ақпарат жинау, қосымшаның функционалды құрылымын анықтау, Android Studio платформасында даму мүмкіндіктерін жақсарту және Мобильді қосымшаларды құру тәжірибесін арттыру кіреді.

**Зерттеу нысаны:** медициналық мекемелер және олардың мобильді қосымша арқылы пациенттермен өзара әрекеттесуі.

**Зерттеу міндеттерінің қысқаша сипаттамасы:** Android Studio платформасында озық технологияларды пайдалана отырып, денсаулық сақтау мәселелері бойынша кеңес беру үшін мобильді қосымшаны әзірлеу және жобалау. Негізгі назар қосымшаның ыңғайлылығына, мәліметтер базасына қол жетімділікке және медициналық мекемелермен өзара әрекеттесу мүмкіндігіне аударылады.

**Зерттеудің әдістері:**

1. *Зерттеудің негізгі мақсаттарын анықтау:* қосымша арқылы медициналық мекемелермен өзара әрекеттесуді жақсарту.
2. *Аналогтарды талдау:* Android жүйесінде бар денсаулық сақтау қолданбаларын зерттеу. Олардың функционалдығын, дизайнын, пайдаланушылардың пікірлерін және басқа сипаттамаларын талдау.
3. *Нәтижелерді талдау және қорытындыларды тұжырымдау:* Алынған деректерді жүйелеу және талдау. Пайдаланушылардың негізгі проблемалары мен қажеттіліктерін анықтау. Алынған нәтижелер негізінде қосымша арқылы медициналық мекемелермен өзара әрекеттесуді жақсарту бойынша ұсыныстарды тұжырымдау.
4. *Жақсартуларды енгізу және тиімділікті бағалау:* Анықталған ұсынымдарға сәйкес қосымшаға өзгерістер әзірлеу және енгізу. Пайдаланушылардың пікірлерін бақылау, пайдалану статистикасы және т. б. арқылы енгізілген жақсартулардың тиімділігін бағалау*.*

**Дипломдық жобаның құрылымы мен мазмұны:**Дипломдық жоба кіріспе, екі негізгі бөлімнен, қорытынды және пайдаланылған әдебиеттер тізімін қамтиды.

**1 МОБИЛЬДІ ҚОСЫМШАНЫ ҚҰРУДЫҢ ТЕОРИЯЛЫҚ НЕГІЗДЕРІ**

* 1. **Мобильді қосымшалардың түсінігі және жіктелуі**

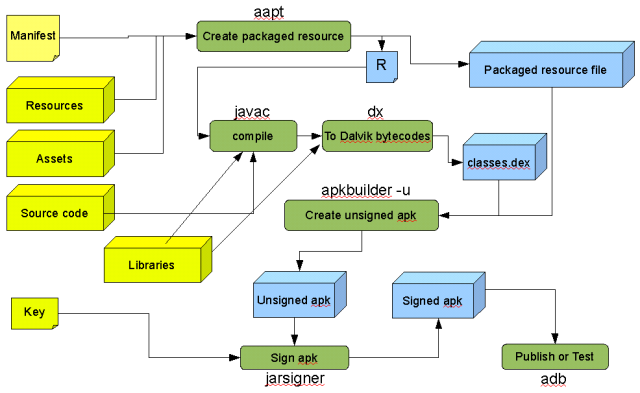
Бүгінгі таңда ұйымда мобильді қосымшаның болуы таңқаларлық емес. Сауда фирмалары мен өнеркәсіптік кәсіпорындардан бастап ауруханаларға дейінгі барлық дерлік компаниялардың өздерінің мобильді қосымшалары бар. Мобильді қосымшаны құру және оны қолданбалар дүкендеріне орналастыру туралы көбірек адамдар ойлануда. Біреу мұны тек қызығушылық үшін қызықтырады, біреуге өзін-өзі таныстыру немесе байланыс қосымшасы қажет, ал біреу бастықтардан тапсырма алады: компания немесе бөлім үшін мобильді қосымшаны құру немесе жаңарту. Көптеген онлайн қызметтер бар, бірақ ең көп таралған деректер қолданбалар дүкендерінен алынған.

Олар пайдаланушыға ыңғайлы кез келген уақытта қол жетімді мәтіндік, графикалық және бейне ақпарат. Мобильді қосымша белгілі бір тақырыпқа арналуы немесе жинақтың бөлігі болуы мүмкін. Ол бірнеше экрандардан тұруы мүмкін және белсенді қолданылатын қосымшалардың бөлігі бола береді.

Мобильді қосымша-бұл бір атаумен біріктірілген қабаттар жиынтығы. Әр қабатта мәтін, суреттер, аудио және бейне файлдар, сондай-ақ түймелер, сілтемелер және кері байланыс формалары сияқты интерактивті элементтер болуы мүмкін. Мобильді қосымшаның құрылымы қабаттардан, қолданба жолынан, кодтан және файл атауынан тұрады. Мобильді қосымшаның негізгі экраны үй болып саналады және оның атауы бар негізгі әрекет.

Мобильді қосымшаны құру кезінде қолданбаға қажетті деректер мен ақпаратты, сондай-ақ мақсатты аудиторияны және олардың қалауын анықтау маңызды. Қолданба экрандарындағы ақпарат жүйеленіп, белгілі бір ережелерге сәйкес келуі керек. Қосымшаны жасағаннан кейін емле мен техникалық қателерді мұқият тексеріп, оның әртүрлі құрылғылармен және экран ажыратымдылықтарымен үйлесімділігін қамтамасыз ету қажет. Қолданбаны үнемі жаңартып отыру да оны қолдаудың маңызды аспектісі болып табылады.

Мобильді қосымшаны қолданбалар дүкеніне орналастыру үшін белгілі бір Виртуалды кеңістік қажет.



1.1-сурет Android қосымшасының негізгі құрылымы

Құрастыруды іске қосқан кезде, ең алдымен, AndroidManifest талданады.пакет сияқты негізгі параметрлерді қамтитын xml (мысалы, com.example.app) және SDK мақсатты нұсқасы.

Содан кейін AndroidManifest жіберетін aapt (Android Asset Packaging Tool) бағдарламасы шақырылады.xml, RES/, assets / ресурстары бар каталог, android жолы.қажетті мақсатты нұсқаның jar. aapt ресурстарды тексереді және сыныпты құра отырып, оларды құрастырады R.java, ресурс идентификаторларын, сондай-ақ ресурстар файлын қамтиды.xml ресурстары және олардың атрибуттары туралы ақпаратты қамтитын arsc.

Әрі қарай, жобада қолданылатын барлық кітапханалар жүктеледі және Java - javac компиляторы іске қосылады. Алынған class файлдары dx бағдарламасына жіберіледі, ол оларды Dalvik dex форматына түрлендіреді. Сонымен қатар, оңтайландыру үшін дайын кітапханалар бөлек, ал жоба сыныптары бөлек дексирленеді (бұл дексирленген кітапханаларды кэштеуге мүмкіндік береді). Егер бірнеше dex файлдары жасалса, олардың барлығы Dex Merger құралы арқылы біріктіріледі. Нәтижесінде бір classes файлы пайда болады.dex (немесе бірнеше, егер multidex қолданылса).

Енді барлық қажетті компоненттер бар және сіз apk құрастыруды бастауға болады. Бұл барлық файлдарды zip мұрағатына орау ғана, бірақ ол үшін арнайы apkbuilder бағдарламасы қолданылады. Осы процесті аяқтағаннан кейін біз қол қойылмаған apk файлын аламыз, яғни ішінде meta-INF каталогы жоқ.

Соңғы кезең-apk-ге қол қою. Ол үшін қол қойылмаған apk файлымен бірге jarsigner-ге жіберілетін алдын-ала жасалған кілт қолданылады. Нәтижесінде орнатуға болатын дайын қосымша пайда болады.

Мобильді қосымшалар әртүрлі мақсаттарға қызмет ете алады және әртүрлі функцияларды орындай алады: ақпарат беру, тауарлар мен қызметтерді сату, пайдаланушылармен өзара әрекеттесу, файлдарды бөлісу және т.б. Олар сондай-ақ әкімші араласуынсыз мазмұн өзгермеген кезде статикалық және пайдаланушы әрекеттері немесе басқа деректер көздері негізінде мазмұн автоматты түрде қалыптасып, өзгерген кезде динамикалық болып бөлінуі мүмкін.

Мобильді қосымшалардың бірнеше түрі бар, оларды келесідей жіктеуге болады:

1-кесте Мобильді қосымшалардың жіктелуі

|  |  |
| --- | --- |
| Коммерциялық қосымшалар | Коммерциялық қосымшалар-бұл тауарларға немесе қызметтерге тапсырыс беруге және төлем жасауға болатын қосымшалар. Олар клиенттерді тарту және сатылымды арттыру үшін компаниялар арасында танымал. |
| Әлеуметтік платформалар | Әлеуметтік платформалар немесе басқаша айтқанда, әлеуметтік медиа негізінен мобильді құрылғылар арқылы байланысуға арналған. |
| Мазмұн қосымшалары | Мазмұн қолданбалары пайдаланушыларға ағымдағы деректерге жылдам қол жеткізуге мүмкіндік береді. Онда сіз басылымдардағы соңғы жаңалықтар туралы ақпаратты, валюта бағамдары туралы деректерді немесе тиісті аудиторияға арналған пайдалы блогтарды таба аласыз. |
| Ойын қосымшалары | Ойын смартфондар мен планшеттер сияқты мобильді құрылғыларға арналған қосымшалар деп аталады. Олардың басты мақсаты – интерактивті ойындар арқылы пайдаланушының көңілін көтеру. |
| Блогтар мен форумдар | Пайдаланушыларға пікір және тәжірибе алмасуға, мақалалар мен жазбаларды жариялауға мүмкіндік беретін сайттар. |

Мобильді қосымшалардың әртүрлі түрлерін түсіну сіздің мақсатыңызға жету үшін сайттың қай түрі жақсы екенін анықтауға көмектеседі.

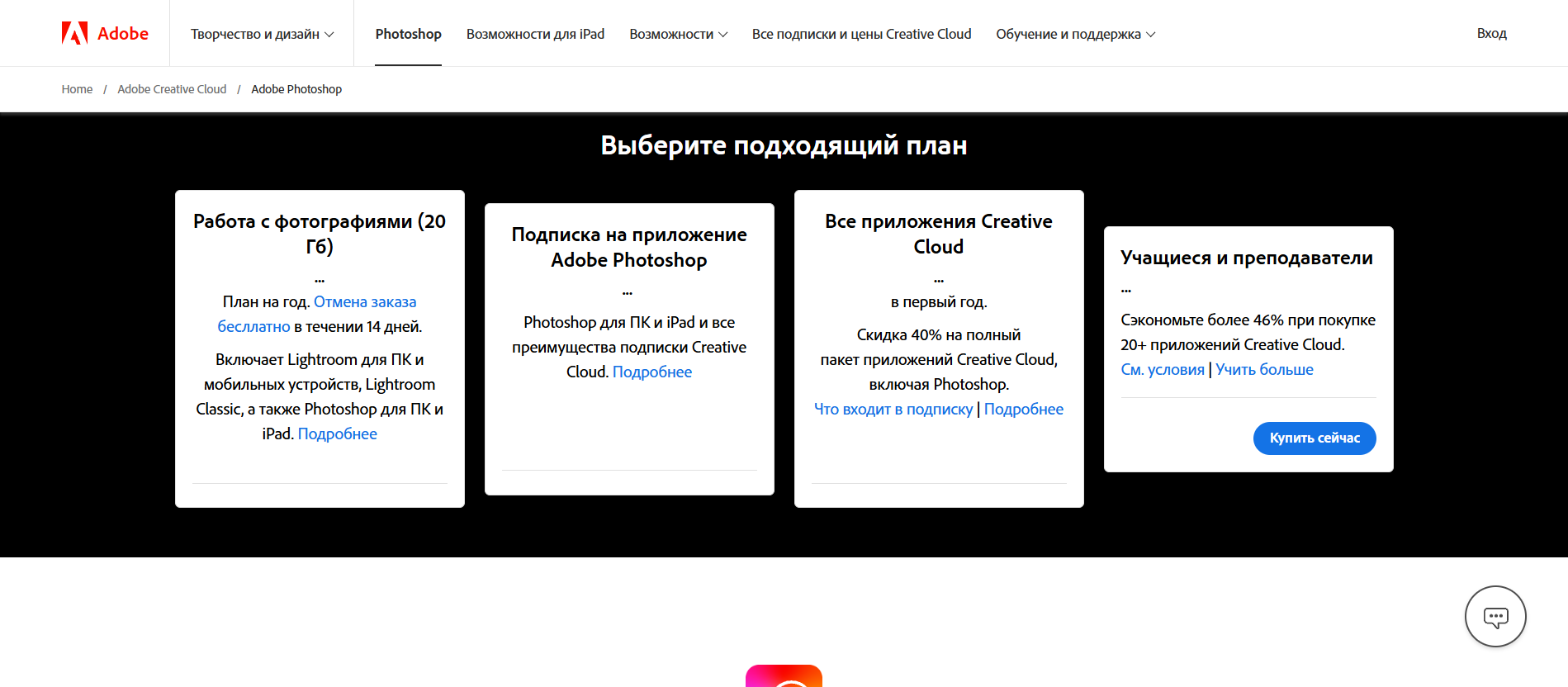
**1.2 Бағдарламалық құралдарды таңдау**

Мобильді қосымшаны әзірлеу кезінде негізгі қадам-жоспар құру, ал оны жүзеге асырудың бірінші кезеңі-қосымшаның дизайнын жасау. Осы мақсатқа арналған көптеген бағдарламалар бар. Кейбіреулері графикалық элементтерді жасауға маманданған, ал басқалары макеттерді жобалауға маманданған, ал кейбіреулері осы екі функцияны біріктіреді. Міне, олардың кейбіреулері:

1. *Adobe Photoshop* – кескіндерді жасау және өңдеу үшін қолданылатын ең танымал графикалық редакторлардың бірі.
2. *Adobe Illustrator* – векторлық форматта графикалық элементтер мен логотиптер жасауға мүмкіндік беретін векторлық графикалық редактор.
3. *Sketch* – қолданбалар мен веб-сайттарды жобалау үшін қолданылатын қуатты интерфейстер мен орналасу құралы.
4. *Figma* – нақты уақыт режимінде макеттерді жасауға және бірлесіп жұмыс істеуге мүмкіндік беретін интерфейс дизайны платформасы.
5. *Adobe XD* – веб-сайттар мен мобильді қосымшаларға арналған интерактивті макеттерді жобалау және құру құралы.
6. *Affinity Designer* – векторлық және растрлық графиканы құруға арналған Photoshop және Illustrator мүмкіндіктерін біріктіретін графикалық редактор.

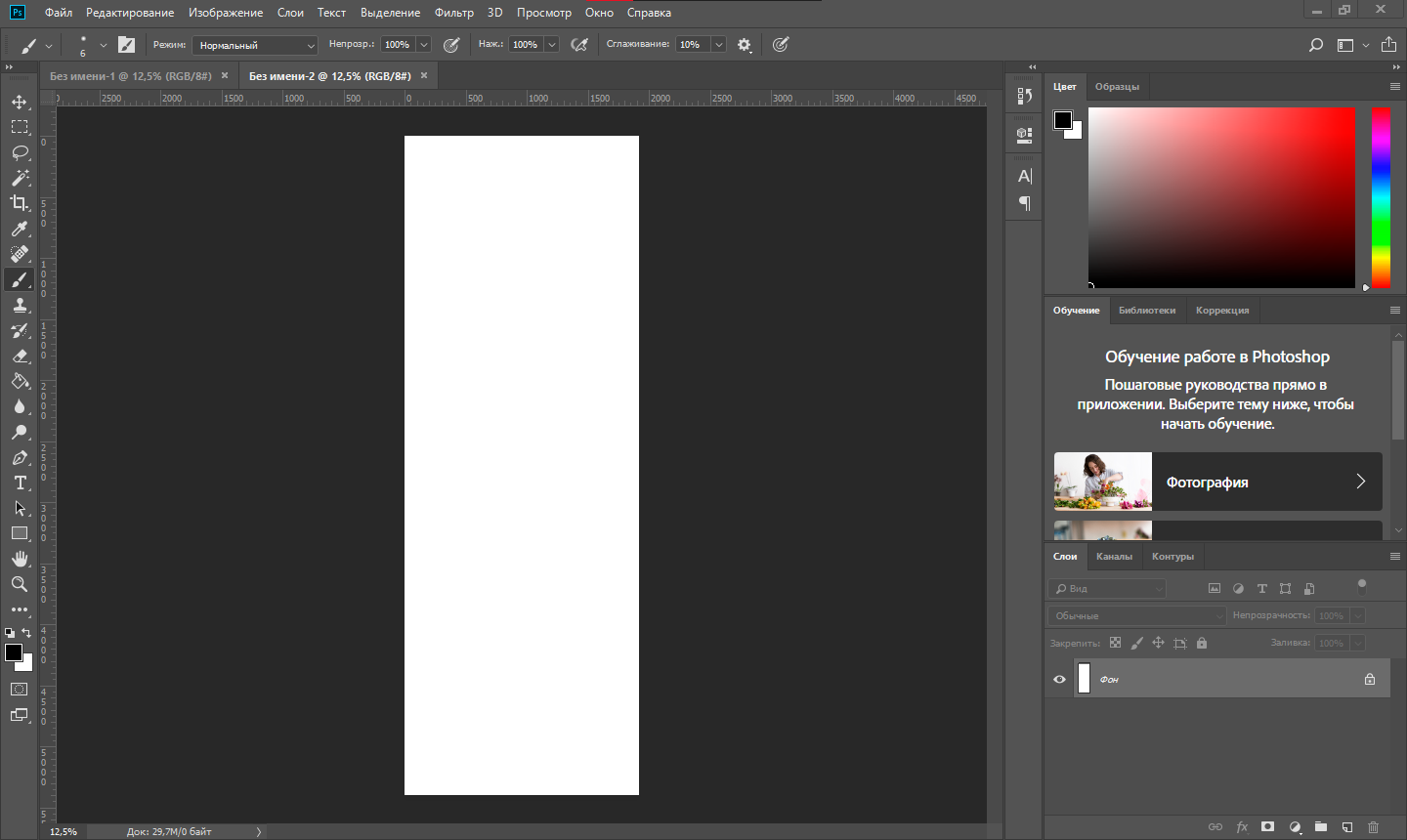
Бұл мобильді қосымшаның дизайнын жасауға арналған қол жетімді бағдарламалардың аз ғана бөлігі. Белгілі бір бағдарламаны таңдау пайдаланушының жеке қажеттіліктері мен қалауына байланысты. Мобильді қосымшаларды жобалау үшін ең көп қолданылатын бағдарламалардың бірі-Adobe Photoshop.

Adobe Photoshop – бұл кескіндерді жасауға және өңдеуге арналған кәсіби графикалық бағдарлама. Ол дизайнерлерге жоғары сапалы графикалық элементтерді, пайдаланушы тәжірибесінің интерфейстерін және мобильді қосымшалардың макеттерін жасауға мүмкіндік беретін көптеген құралдар мен мүмкіндіктерге ие. Photoshop әртүрлі қабаттармен, сүзгілермен, эффектілермен және сурет салу құралдарымен жұмыс істеуге мүмкіндік береді, бұл оны мобильді қосымшалардың дизайнын жасаудың қуатты құралы етеді. Бағдарлама Adobe ресми сайтында жазылу арқылы қол жетімді <https://www.adobe.com/ru/products/photoshop.html> сурет (сурет. 1.2).



1.2-сурет Adobe компаниясының ресми сайты

Бағдарлама интерфейсі туралы. Сол жақта құралдар тақтасы орналасқан. Онда қажетті мәзір, ресурстар, құралдар бар. Қылқалам параметрлері жоғарғы жағында. Ал оң жақта қабаттарды шарлау тақтасы және түстер палитрасы. Орталықта жаңа бет бар және ол сайттың дизайнын жасайды (сурет.1.3).



1.3-сурет Adobe Photoshop бағдарламасының интерфейсі

Мобильді қосымшаның дизайны аяқталғаннан кейін, осы дизайнға сүйене отырып, код жазуды бастайды. Осы мақсатта Android Studio жиі қолданылады, ол мобильді қосымшаларды әзірлеудің барлық қажетті құралдарын, соның ішінде код редакторын, жөндеу және құрастыру құралдарын және басқа да көптеген мүмкіндіктерді ұсынады.

Android қосымшаларын әзірлеуге арналған бағдарламалық құралдарды таңдау шектеулі болуы мүмкін, бірақ кейбір баламалар бар. Олардың кейбіреулері мыналарды қамтиды:

1. *Eclipse:* Eclipse Java үшін интеграцияланған даму ортасы ретінде танымал болғанымен, ол сонымен қатар тиісті плагиндермен Android қосымшаларын әзірлеуді қолдайды.
2. *IntelliJ IDEA:* бұл тағы бір танымал интеграцияланған даму ортасы, ол Android-ті дамытуға қолдау көрсетеді.
3. *Xamarin:* бұл құрал әзірлеушілерге C# бағдарламалау тілі мен жалпы кодтық базаны қолдана отырып, әр түрлі платформалар үшін, соның ішінде Android үшін мобильді қосымшалар жасауға мүмкіндік береді.
4. *React Native:* React Native сізге JavaScript бағдарламалау тілін және платформалар арасында кодты қайта пайдалану арқылы мобильді қосымшалар жасауға мүмкіндік береді*.*

Android studuio бағдарламалық құралының негізгі артықшылықтарын қарастырыңыз:

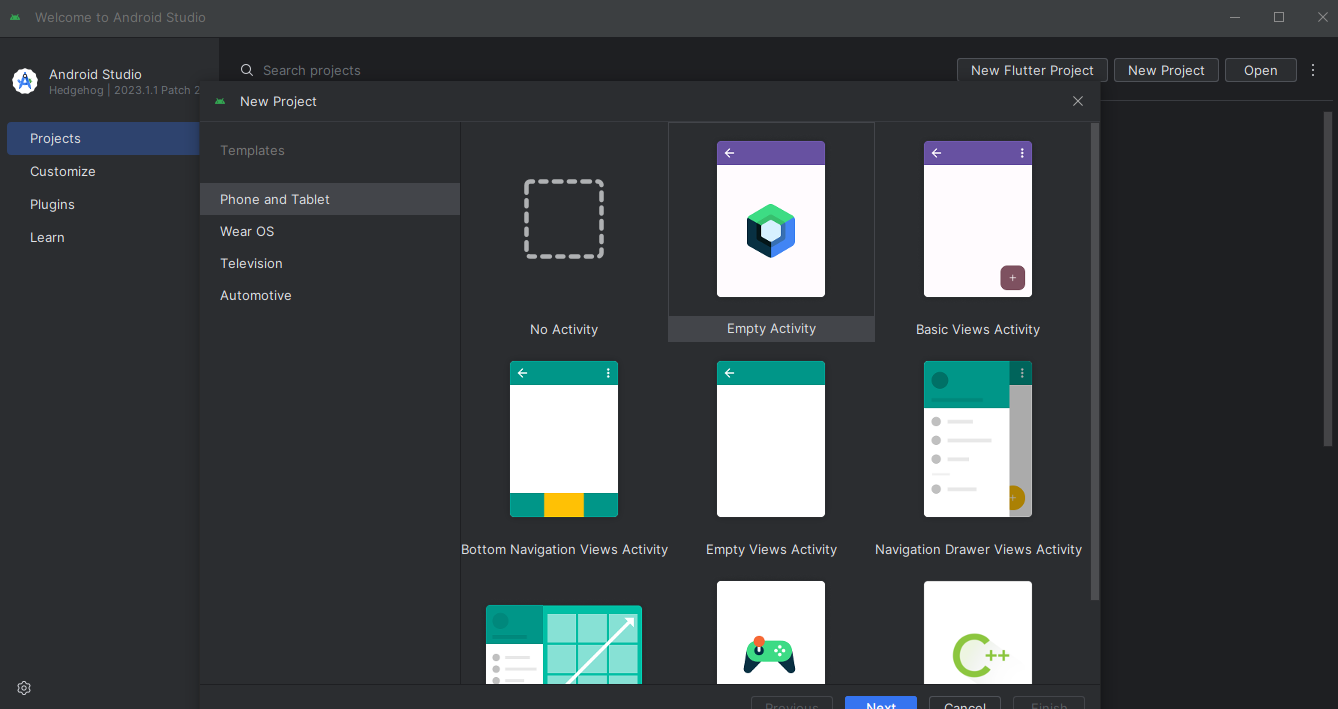
***Артықшылықтары:***

* *Жүктеп алу Тегін:* Android studio тегін таратылады және Windows, MacOS және Linux сияқты барлық операциялық жүйелерде жүктеп алуға болады.
* *Интеграцияланған даму ортасы (IDE):* Android Studio IntelliJ IDEA негізіндегі қуатты және ыңғайлы даму ортасын ұсынады. Бұған синтаксисті бөлектейтін код редакторы, автотолтыру, интеграцияланған отладчик және әзірлеушінің жұмысын жақсарту үшін басқа да көптеген мүмкіндіктер кіреді.
* *Gradle Build System:* Gradle құрастыру жүйесі Android Studio бағдарламасындағы жобалардың негізі болып табылады. Ол икемді жоба конфигурациясын, тәуелділікті басқаруды және қолданбаны тиімді құрастыруды қамтамасыз етеді.
* *Android Emulator:* Android Studio құрамында Android эмуляторы бар, ол Android-тің әр түрлі нұсқалары мен әр түрлі сипаттамалары бар виртуалды құрылғыларда қосымшаларды іске қосуға және тексеруге мүмкіндік береді.
* *Layout Editor:* бұл құрал әзірлеушілерге UI макеттерін визуалды түрде жасауға мүмкіндік береді. Ол элементтерді экранда оңай орналастыруға және олардың қасиеттерін реттеуге арналған көптеген мүмкіндіктерге ие.
* *Өнімділік мониторы (Profiler):* бұл құрал қолданбаның нақты уақыттағы өнімділігін талдауға, жадты, CPU және басқа ресурстарды пайдалануды бақылауға, жадтың ағып кетуін анықтауға және өнімділікті оңтайландыруға мүмкіндік береді.
* *Көптеген плагиндер мен кеңейтімдер:* әзірлеушілер қауымдастығы оның функционалдығын кеңейтетін және кодты талдау, кодты құру және т. б. сияқты жаңа мүмкіндіктерді қосатын көптеген Android Studio плагиндерін жасады.

***Кемшіліктері:***

* *Кейбір мүмкіндіктер жаңадан бастаушылар үшін интуитивті болмауы мүмкін:* Visual Studio Code интерфейсі өте интуитивті болғанымен, бағдарламалауды жаңадан бастағандар үшін жөндеу сияқты кейбір мүмкіндіктер түсініксіз болуы мүмкін.
* *Ол толыққанды даму ортасын алмастыра алмайды:* Visual Studio Code веб-қосымшаларды құруға өте ыңғайлы, бірақ ол үлкен жобаларда жұмыс істеу үшін жеткілікті тиімді болмауы мүмкін және толыққанды даму ортасын алмастыра алмайды.
* *Кейбір қондырмалар тұрақсыз болуы мүмкін:* плагиндер мен кеңейтімдердің кең таңдауы болғанымен, кейбіреулері тұрақсыз болуы мүмкін және Visual Studio Code пайдалану кезінде қиындықтар тудыруы мүмкін.

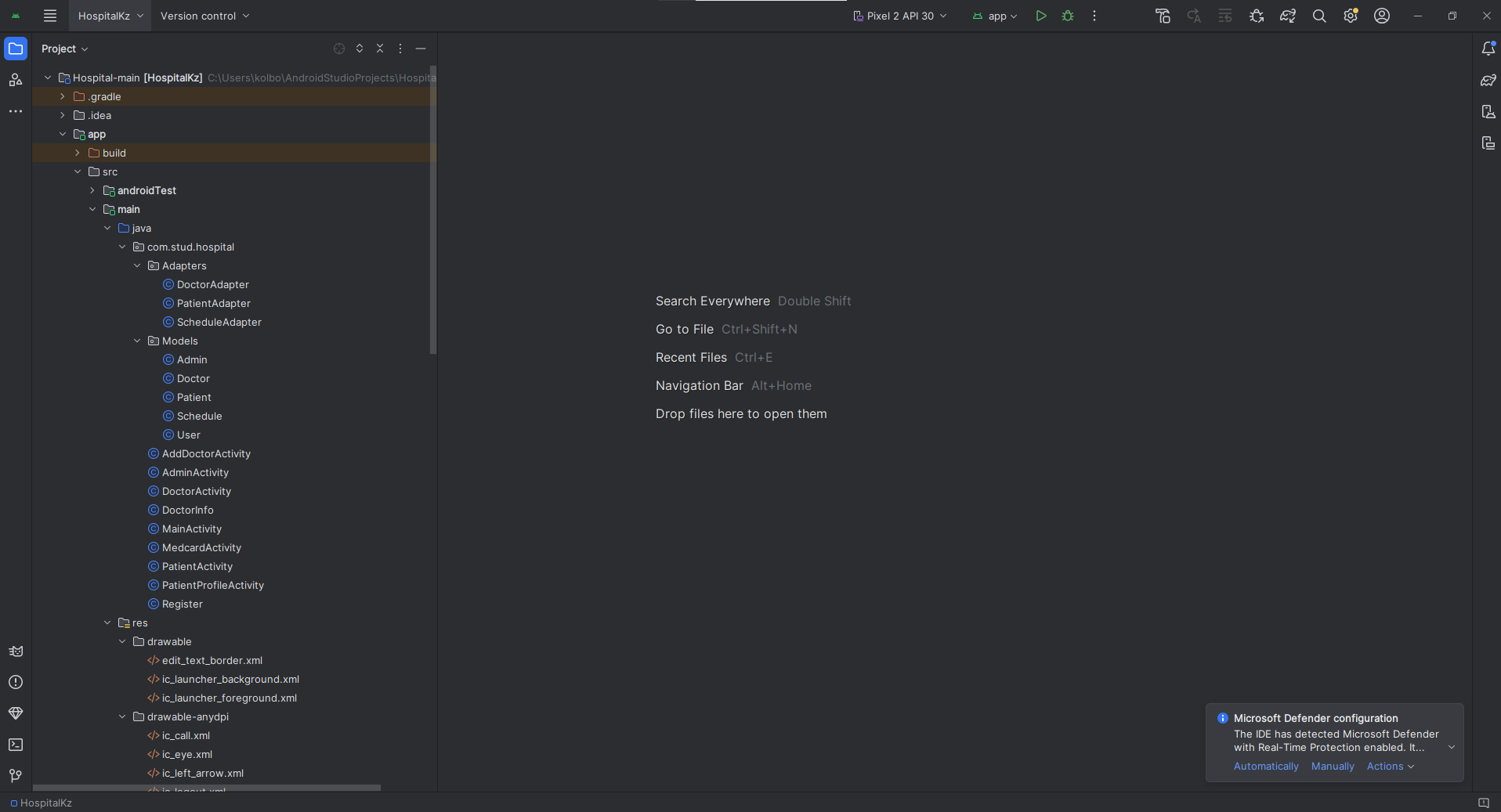
Android Studio-бұл Android операциялық жүйесіне арналған қосымшаларды құруға арналған интеграцияланған даму ортасы (IDE). Бағдарлама Java, Kotlin және C.сияқты әртүрлі бағдарламалау тілдерін қолдайды. Сіз Android Studio-ны ресми сайттан жүктей аласыз (сурет. 1.4) сілтеме бойынша <https://developer.android.com/studio>. бұл Android-пен жұмыс істейтін мобильді құрылғыларға арналған қосымшаларды құруға, тексеруге және жөндеуге арналған көптеген құралдарды ұсынатын қуатты даму құралы.



Сурет 1.4 Android Studio басты беті

Android Studio интерфейсі төрт негізгі компоненттен тұрады (сурет. 1.5):

1. *Әрекеттер тақтасы ("А" әрпімен белгіленген) -* бұл экранның жоғарғы жағында орналасқан, әртүрлі жұмыс режимдері арасында ауысу функцияларын ұсынатын панель.
2. *Редакторлар ("B" әрпімен белгіленеді) -* мұнда жобаның коды мен ресурстарын өңдеуге ашық әр түрлі редакторлар көрсетіледі. Мұнда бір уақытта бірнеше файлды ашуға болады.
3. *Бүйірлік тақта ("С" әрпімен белгіленеді) -* файл менеджері, пайдаланушы интерфейсінің компоненттері, ресурстар және жобаның басқа элементтері сияқты маңызды құралдар бар.
4. *Күй жолағы ("D" әрпімен көрсетілген) -* ағымдағы жоба туралы ақпаратты, белсенді қойындыны және құрастыру күйі және жөндеу құрылғысына қосылу сияқты басқа мәліметтерді көрсетеді.



1.5-сурет VS Code бағдарламасының интерфейсі

**1.3 Android бағдарламаларын жасауға арналған бағдарлама тілдері**

Мобильді қосымшаларды әзірлеу үшін әртүрлі бағдарламалау тілдері қолданылады. Олардың әрқайсысы ерекше ерекшеліктерге, функционалдылыққа ие және нақты мәселелерді шешуге арналған. Android Studio-да мобильді қосымшаларды құру үшін жиі қолданылатын бағдарламалау тілдерінің ішінде мыналарды атап өтуге болады: Java, Kotlin және C..

Java-бұл Android қосымшаларын жасау үшін кеңінен қолданылатын объектіге бағытталған бағдарламалау тілі. Ол кодтың жоғары тасымалдануын және көптеген кітапханалар мен құрылымдарды қолдауды қамтамасыз етеді.

Kotlin-бұл Android қосымшаларын жасаушылар арасында танымал бола бастаған Заманауи бағдарламалау тілі. Ол Jам және мәнерлі синтаксисті ұсынады, сонымен қатар Java кодымен өзара әрекеттесу мүмкіндігі бар.

C++ - бұл құрылғының төменгі деңгейлі ресурстарымен жұмыс істеуді қажет ететін ойындар мен қосымшаларды қоса алғанда, жоғары деңгейлі қосымшаларды әзірлеу үшін қолданылатын төмен деңгейлі бағдарламалау тілі.

Android Studio әзірлеушілерге осы бағдарламалау тілдеріне негізделген мобильді қосымшаларды құруға және тексеруге кең мүмкіндіктер ұсынады, сонымен қатар Android API және басқа Даму құралдарымен ыңғайлы өзара әрекеттесуге мүмкіндік беретін интеграцияланған даму ортасы бар.HTML гипермәтінінің құрылымы тегтерденде тұрады.

2-кесте негізгі Java тегтері

|  |  |
| --- | --- |
| **public:** | Элементке ашық қол жетімділік |
| **static:** | Объектіге емес, сыныпқа жатады |
| **void:** | Мәнді қайтармайды |
| **int:** | Бүтін деректер түрі |
| **String:** | мәтіндік деректермен жұмыс |

Java тілінің негізгі ерекшеліктері:

* Көп ағынды: Java көп ағынды бағдарламалауды қолдайды, бұл бір уақытта бірнеше тапсырманы орындауға қабілетті қосымшалар жасауға мүмкіндік береді.
* Платформаның тәуелсіздігі: Java-дағы кодты бір рет жазуға және әртүрлі операциялық жүйелер мен архитектураларда іске қосуға болады, өйткені Java виртуалды машинасы (JVM) платформаның тәуелсіздігін қамтамасыз етеді.
* Ерекшеліктерді өңдеу: Java бағдарламадағы қателер мен ерекше жағдайларды икемді басқаруға мүмкіндік беретін ерекшеліктерді өңдеу механизмін ұсынады.
* Кең стандартты кітапхана: Java файлдармен, желімен, графикамен және т. б. жұмыс істеу сияқты қолданбаларды әзірлеудің әртүрлі аспектілерімен жұмыс істеуге арналған көптеген сыныптар мен әдістерді қамтитын кең стандартты кітапханамен бірге келеді.

*Kotlin-*JetBrains әзірлеген статикалық терілген бағдарламалау тілі, ол Java байт кодына жинақталған және Java виртуалды машинасында (JVM) жұмыс істей алады. Ол әзірлеушілерге қосымшаларды әзірлеу үшін заманауи, мәнерлі және қауіпсіз балама тіл ұсыну мақсатында жасалған. Kotlin сонымен қатар бір жобаның бөлігі ретінде Kotlin және Java-да кодты пайдалануға мүмкіндік беретін 100% Java үйлесімділігін қамтамасыз етеді. Ол функционалды бағдарламалауды қолдайды, айқын және қысқа синтаксиске ие және Google-дің ресми қолдауының арқасында Android қосымшаларын жасау үшін белсенді қолданылады.

Java, C#, C немесе C++ сияқты тілдермен таныс тәжірибелі бағдарламашылардың JavaScript-пен жұмыс істеу қажет болатын жағдайға тап болуы сирек емес. Пайдаланушы интерфейстері вебте көбірек орналастырылуда, ал JavaScript-барлық браузерлер қолдайтын тіл. Electron жақтауы бұл мүмкіндікті байытылған клиенттік қосымшаларға таратады және мобильді JavaScript қосымшаларын құруға арналған бірнеше шешімдер бар. Сонымен қатар, JavaScript серверлік бағдарламалауға көбірек енеді.

Көптеген жылдар бұрын JavaScript «қарапайым бағдарламалау» тілі ретінде ойластырылған, оған енгізілген құралдар жиынтығы шатастырып, үлкен бағдарламаларда қателіктер тудыруы мүмкін. Алайда, стандарттау бойынша күш-жігер мен жасалған құралдар алғашқы қарапайым идеялардан әлдеқайда асып түсті.

Өкінішке орай, ескі нұсқалардың батпағына батпастан заманауи JavaScript-ті үйрену өте қиын. Көптеген кітаптардың, курстардың және блог мақалаларының мақсаты-JavaScript-тің бұрынғы нұсқаларынан қазіргі заманғы нұсқаларға көшу, бұл басқа тілдерден келімсектерге онша пайдалы емес.

Менің ойымша, оқырман тармақталу, цикл, функция, деректер құрылымы деген не екенін түсінетін және объектіге бағытталған бағдарламалаудың негіздерімен таныс білетін бағдарламашы. Мен қазіргі JavaScript-те өнімді бағдарламашы болу нені білдіретінін түсіндіремін, тек жақшада өткен құралдар туралы айтамын. Сіз ежелгі аулау шұңқырларынан аулақ бола отырып, заманауи JavaScript-ті қызметке қалай қою керектігін білесіз.

Java мінсіз болмауы мүмкін, бірақ тәжірибе көрсеткендей, бұл әртүрлі қосымшаларды, соның ішінде мобильді, веб-қосымшаларды және көптеген серверлік тапсырмаларды бағдарламалау үшін өте жақсы. Джефф Этвуд атап өткендей,"Java-да жазуға болатын кез-келген қосымша Java-да жазылады".

Сіз Java-ны веб-беттеріңізге ендіре аласыз және пайдаланушылардан алған ақпаратқа сәйкес теңшелетін беттер жасай аласыз. Бұл веб-беттерді қызықты және динамикалық етеді. Java веб-қосымшаларды құру үшін қолданылады, өйткені оның соңғы нәтижесін HTML түрінде көрсетуге болады. Бұл сервер жағында тиімді, кросс-платформа (әр түрлі операциялық жүйелерде жұмыс істейді) және C немесе Perl сияқты басқа бағдарламалау тілдерімен салыстырғанда салыстырмалы түрде оңай.

Java-да жоғары қуат және кең қолдану аясы бар. Мысалы, көптеген ірі веб-қызметтер, соның ішінде танымал әлеуметтік желілер мен ақпараттық ресурстар Java-ны өз жұмысында пайдаланады. Бұл оның үлкен және масштабталатын жүйелерді құрудағы тиімділігі мен сенімділігін растайды. Сонымен қатар, Java-да қиын сәттерде көмек пен қолдау көрсетуге дайын белсенді әзірлеушілер қауымдастығы бар, бұл оны веб-әзірлеудің тартымды нұсқасына айналдырады.

Әрине, мұнда жеті пайдалы Java мүмкіндігі бар:

* Көп ағынды: Java параллель тапсырмалар жасауға және Қолданба өнімділігін жақсартуға мүмкіндік беретін қуатты көп ағынды құралдарды ұсынады.
* Стандартты кітапхана: Java файлдармен, желімен, коллекциялармен және т. б. жұмыс істеу сияқты бағдарламалаудың әртүрлі аспектілерімен жұмыс істеуге арналған көптеген сыныптар мен әдістерді қамтитын кең стандартты кітапханамен бірге келеді.
* Ерекшеліктерді өңдеу: Java-дағы ерекшеліктерді өңдеу механизмі бағдарламадағы қателер мен ерекше жағдайларды икемді басқаруға мүмкіндік береді, оның сенімділігін арттырады.
* GUI: Java JavaFX және Swing сияқты кітапханалар арқылы графикалық пайдаланушы интерфейстерін құруға арналған құралдарды ұсынады, бұл ыңғайлы интерфейсі бар қосымшаларды әзірлеуді жеңілдетеді.
* Сериялау: Java нысандарды файлдарға сақтауға немесе оларды желі арқылы тасымалдауға мүмкіндік беретін нысандарды сериялау механизмін қолдайды, бұл қолданба күйін сақтау үшін пайдалы.
* Жинақтар: Java тізімдер, массивтер, карталар және т.б. сияқты бай коллекциялар жиынтығын ұсынады, бұл деректермен жұмыс істеуді және оларды ұйымдастыруды жеңілдетеді.

Веб-әзірлеудің ыңғайлылығы: Java EE (Enterprise Edition) веб-қызметтерді, сервлеттерді, JSP және т.б. қоса алғанда, веб-қосымшаларды әзірлеудің қуатты құралдарын ұсынады, бұл Java-ны таратылған және вебке бағытталған қосымшаларды құрудың танымал таңдауына айналдырады.

**1.4 Android бағдарламаларын жасау үшін қолданылатын шеңберлер**

Java Шеңбері-бұл Java тілінде веб-сайттар мен қосымшаларды әзірлеуді жеңілдететін құралдар, кітапханалар және бағдарламалау стандарттарының жиынтығы. Бұл бағдарламашыларға мәліметтер базасымен жұмыс істеу, шаблондар жасау және басқа күнделікті тапсырмаларды орындау үшін дайын шешімдерді қолдану арқылы мәселелерді шешуге көмектеседі. Қазіргі уақытта Java үшін көптеген өзекті құрылымдар бар, соның ішінде:

* Java-да қосымшаларды әзірлеуге арналған стандартталғанқұралдар мен кітапханалар жиынтығы;
* Деректер базасымен және дамудың басқа аспектілерімен жұмыс істеуді жеңілдету;
* Дамуды жеделдету үшін бағдарламалау стандарттары мен архитектуралық шаблондарды қолдау;
* Күнделікті бағдарламалау тапсырмалары үшін дайын шешімдер мен кодты тиімді пайдалану.

Java тілінде веб-әзірлеуге арналған екі танымал құрылым-бұл көктем және күту.

Көктем-бұл Java-да веб-қосымшаларды құрудың қуатты құралы. Ол веб-қызметтермен жұмыс істеу, тәуелділікті басқару және басқару инверсиясын қоса алғанда, функционалдылықтың кең ауқымын қамтамасыз етеді. Spring әзірлеу процесін жеңілдетуге және қолданбалардың жоғары өнімділігін қамтамасыз етуге арналған.

Екінші жағынан, Hibernate-бұл Java қосымшаларында мәліметтер базасымен жұмыс істеуге арналған құрылым. Ол дерекқордағы деректермен жұмыс істеу үшін Java нысандарын пайдалануға мүмкіндік беретін объектілі-реляциялық дисплеймен (ORM) жұмыс істеуге ыңғайлы құралдарды ұсынады. Hibernate SQL сұрауларын қолмен жазу қажеттілігін азайтады және деректермен жұмыс істеуді жеңілдетеді.

Бұл құрылымдардың екеуі де, Spring және Hibernate, Java-да веб-қосымшаларды әзірлеу кезінде функционалдығы, ыңғайлылығы және тиімділігі арқасында Java әзірлеушілер қауымдастығында өте танымал.

/



1.6-сурет spring ресми сайты

Bootstrap файлдарының структурасы төмендегідей:

- src/

- main/

- java/

- com/

- example/

- myapplication/

- controller/ (контроллеры)

- service/ (сервисы)

- repository/ (репозитории)

- model/ (модели данных)

- resources/

- static/ (статические ресурсы)

- templates/ (шаблоны)

- application.properties (конфигурационные файлы)

- test/ (тесты)

- java/

- com/

- example/

- myapplication/

- controller/

- service/

- repository/

- model/

- resources/

Spring сонымен қатар басқаруды инверсиялауды (IoC) және тәуелділікті енгізуді (DI) қолдайды, бұл кодты икемді, модульдік және оңай тексеруге мүмкіндік береді.Мұнда Spring Boot көмегімен қарапайым веб-қосымшаны құруға арналған шағын кодтың мысалы келтірілген:

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

@SpringBootApplication

public class MyApplication {

@GetMapping("/")

public String home() {

return "Привет, мир!";

}

public static void main(String[] args) {

SpringApplication.run(MyApplication.class, args);

}

}

Hibernate-Java нысандары арқылы дерекқорлармен қарапайым және тиімді жұмыс істеуді қамтамасыз ететін Java тіліне арналған қуатты ORM (object-Relational Mapping) құрылымы. Ол әзірлеушілерге SQL сұрауларынан гөрі объект деңгейіндегі деректермен жұмыс істеуге мүмкіндік беретін Дерекқордың реляциялық моделіне объект моделін көрсетуге арналған құралдарды ұсынады.

Hibernate сонымен қатар деректерді кэштеу, транзакцияларды басқару, мұрагерлікті қолдау және нысандар арасындағы байланыстар сияқты көптеген мүмкіндіктерді ұсынады. Оның икемділігі мен кеңеюі оны дерекқорға қол жетімді күрделі қосымшаларды жасаумен айналысатын әзірлеушілер үшін танымал таңдау етеді.

Hibernate және оның функциялары туралы толық ақпаратты оның ресми сайтынан табуға болады https://hibernate.org/orm/documentation/5.6/ . Hibernate әртүрлі дерекқорлармен және операциялық жүйелермен үйлесімді, бұл оны Java қосымшаларында деректермен жұмыс істеудің әмбебап құралы етеді.

Hibernate бағдарламалау әлемінде 2000 жылдардың басында басталды. Бұл ORM (object-Relational Mapping) шеңберін Кевин Кинг 2001 жылдың ақпанында шыққан Hibernate ORM жобасы аясында жасаған. Hibernate идеясы Объектілік модельді реляциялық модельге көрсетудің қуатты құралын ұсына отырып, Java әзірлеушілері үшін дерекқорлармен өзара әрекеттесуді жеңілдету болды.Бүгінгі күні Hibernate әзірлеушілер қауымдастығының белсенді қатысуымен дамуды және жетілдіруді жалғастыруда. Оның ресми сайты мен құжаттамасы бағдарламалық жасақтаманы әзірлеудегі заманауи мәселелерге жаңа мүмкіндіктер мен шешімдер ұсыну үшін үнемі жаңартылып отырады. Hibernate ORM-дің жетекші құрылымдарының бірі болып қала береді және Java қосымшаларында деректермен жұмыс істеудің сенімді құралы ретінде қызмет етеді.

Күту күйі қалай жұмыс істейді?

Hibernate Java қолданбасы мен дерекқор арасындағы қабат ретінде жұмыс істейді. Міне, бұл қалай болады:

Кестелердегі нысандарды көрсету: Hibernate әзірлеушілерге дерекқорда сақталатын нысандарды (Entity) білдіретін Java нысандарымен жұмыс істеуге мүмкіндік береді. Аннотациялар немесе картаға түсіру файлдары (XML) арқылы дерекқордағы нысандар мен кестелер арасындағы сәйкестікті көрсетуге болады.

Нысандардың өмірлік циклін басқару: Hibernate нысандарды құру, Оқу, жаңарту және жоюды қоса алғанда, нысандардың өмірлік циклін ыңғайлы басқаруды қамтамасыз етеді. Ол объектілердің күйін бақылайды және қажет болған жағдайда оларды дерекқормен автоматты түрде синхрондайды.

SQL сұрауларын генерациялау: Hibernate автоматты түрде SQL сұрауларын нысан операциялары негізінде жасайды, бұл әзірлеушілерге SQL мәліметтері туралы алаңдамай объектілермен жұмыс істеуге мүмкіндік береді.

Деректерді кэштеу: Hibernate өнімділікті жақсарту үшін деректерді кэштеу мүмкіндіктерін ұсынады. Ол дерекқордағы қоңыраулар санын азайтуға мүмкіндік беретін нысандар, сұраулар және аймақтар деңгейінде кэштеуді қолдайды.

Транзакцияны басқару: Hibernate транзакция менеджерімен біріктіріледі (мысалы, JTA немесе Spring), бұл транзакцияны сенімді басқаруға және ACID қасиеттерін қолдауға мүмкіндік береді.

Hql және criteria API: Hibernate пайдалану объектілер мен олардың қасиеттерін пайдалана отырып, күрделі дерекқор сұрауларын орындау үшін HQL сұрау тілін (Hibernate Query Language) және Criteria API ұсынады.

Тұтастай алғанда, Hibernate Java нысан деңгейіндегі дерекқорды абстракциялауды қамтамасыз етеді, деректермен жұмыс істеуді жеңілдетеді және қолданбаларды әзірлеу кезінде жоғары өнімділік пен икемділікті қамтамасыз етеді.

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HelloWorld {

public static void main(String[] args) {

// Создаем конфигурацию Hibernate

Configuration configuration = new Configuration();

configuration.configure("hibernate.cfg.xml");

// Создаем фабрику сессий

SessionFactory sessionFactory = configuration.buildSessionFactory();

// Получаем сессию

try (Session session = sessionFactory.openSession()) {

// Начинаем транзакцию

session.beginTransaction();

// Создаем объект

HelloWorldEntity helloWorldEntity = new HelloWorldEntity();

helloWorldEntity.setMessage("Hello, World!");

// Сохраняем объект в базе данных

session.save(helloWorldEntity);

// Фиксируем транзакцию

session.getTransaction().commit();

// Выводим сообщение

System.out.println("Message saved successfully!");

} catch (Exception e) {

System.err.println("Error: " + e.getMessage());

} finally {

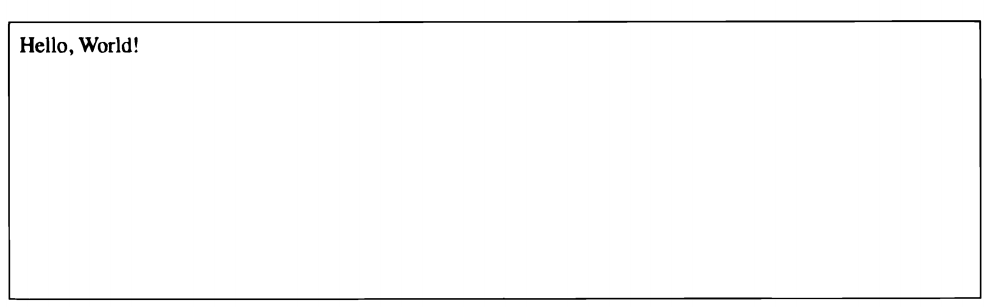
// Закрываем фабрику сессий

sessionFactory.close();

}

}

}



Сурет 1.7 Hello, World жолын қайтару! Hibernate көмегімен

Күту Режимінің Артықшылықтары:

* ORM шеңбері: Hibernate Java нысандары арқылы дерекқорлармен ыңғайлы жұмыс істеуге мүмкіндік береді, бұл SQL сұрауларын қолмен жазу қажеттілігін азайтады.
* Нысандардың өмірлік циклін басқару: Hibernate нысандардың күйін автоматты түрде бақылайды және басқарады, бұл оларды құруды, өзгертуді және жоюды жеңілдетеді.
* Икемділік: Hibernate нысандар арасындағы байланыстарды көрсетудің әртүрлі түрлерін қолдайды (One-to-One, m-to-Many, Many-to-One, many-to-Many), бұл оны әртүрлі сценарийлерге икемді етеді.
* Деректерді кэштеу: Hibernate қолданбаның өнімділігін жақсартатын нысан деңгейіндегі және сұрау деңгейіндегі Деректерді кэштеу механизмдерін ұсынады.
* JPA интеграциясы: Hibernate-Java Persistence API (JPA) спецификациясының бірі, ол әр түрлі ORM шеңберлері арасында стандарттау мен кодтың тасымалдануын қамтамасыз етеді.
* Осы артықшылықтардың арқасында Laravel веб-әзірлеушілерге веб-сайттарды басқа фреймворктарға қарағанда тезірек, оңай жасауға мүмкіндік береді.

Көктемнің Артықшылықтары:

* Басқару инверсиясы (IoC): Spring қолданба құрамдастары арасындағы тәуелділіктерді басқаруға және байланысты азайтуға мүмкіндік беретін IoC механизмдерін қамтамасыз етеді.
* Тәуелділікті енгізу (DI): көктем компоненттер арасындағы тәуелділікті енгізуді жеңілдетеді, бұл тестілеуді жеңілдетеді және кодтың икемділігін арттырады.
* Модульдік және кеңею: Spring модульдік архитектураны қамтамасыз етеді, тек қажетті модульдерді (мысалы, Spring MVC, Spring Security) пайдалануға мүмкіндік береді, бұл артықтықты азайтады және өнімділікті арттырады.
* Аспектке бағытталған бағдарламалау (AOP): Spring API-ді қолдайды, бұл аспектілерді (мысалы, транзакциялар, логинг) негізгі кодтан шығаруға мүмкіндік береді, оның оқылуы мен сақталуын жақсартады.
* Басқа технологиялармен Интеграция: Spring әртүрлі технологиялар мен құрылымдармен біріктірілген (мысалы, Hibernate, JPA, RESTful қызметтер), қосымшаларды әзірлеу үшін бірыңғай және ыңғайлы технологиялар стегін қамтамасыз етеді.

Екі шеңбердің де артықшылықтары бар және оларды әртүрлі қосымшаларды әзірлеу сценарийлерінде тиімді пайдалануға болады.

**2 ЕМХАНАНЫҢ МОБИЛЬДІ ҚОСЫМШАСЫН ЖОБАЛАУ ЖӘНЕ ӘЗІРЛЕУ**

**2.1 Қосымшаның қызметін жобалау**

Медициналық мекеме жұмысының бағдарламасын әзірлеу кезінде халықтың медициналық көмекке деген қажеттілігін ескеру қажет, ол медициналық мекеме тұрғындарының жыл ішінде бір тұрғынға келу санында көрсетілген келуімен анықталады. Ауруларға барудан басқа, профилактикалық сапарларды да ескеру қажет, бұл жалпы сапарлардың шамамен 30-35% құрайды. Алайда, медициналық тексерулер мен диспансерлеу бағдарламаларының кеңеюіне байланысты олардың саны біртіндеп артады.

Медициналық мекеменің қуаты оның бір ауысымда келушілерге қызмет көрсету қабілетімен анықталады. Әдетте бұл көрсеткіш бір ауысымда 250-ден 1200-ге дейін немесе одан да көп болады, ал қызмет көрсетілетін халықтың саны 15 - тен 100-120 мың адамға дейін өзгереді (1-кестені қараңыз).

Медициналық мекеме мен оның Штаттарының жұмысын жоспарлау мекеме қызмет көрсететін халық санына, оның жасына және жынысына, сондай-ақ аурудың деңгейі мен ерекшелігіне негізделуі керек.

3-кесте Санаттар бойынша емханаларды топтастыру

|  |  |  |
| --- | --- | --- |
| Санаттар | Дәрігерлердің саны  ауысымға бару | Саны  қызмет көрсетілетін  халық саны |
| 1 топ  (санат) | 1200-ден астам | 80 мыңнан астам |
| 2 топ  (санат) | 751 – 1200 | 50-79 мың |
| 3 топ  (санат) | 501-750 | 33-44 мың |
| 4 топ  (санат) | 251-500 | 17-32 мың. |
| 5 топ  (санат) | 250 | 17 мыңға дейін. |

Емханалар негізгі Халыққа қызмет көрсетуден басқа, өз аумағында орналасқан және жеке медициналық мекемелері жоқ өнеркәсіптік кәсіпорындардың қызметкерлеріне медициналық көмек көрсетуге міндетті. Учаскелер дәрігерлерінің тең жұмыс жағдайларын қамтамасыз ету үшін терапевтік учаскелерді ұйымдастыру кезінде учаскенің ұзындығы, оның құрылыс түрі, емханаға дейінгі қашықтық, автокөліктің болуы және басқа факторлар ескеріледі.

Қалаларда емханалар сатылық және медициналық-санитарлық аймақтарға бөлу қағидатын ескере отырып орналастырылады.

*Медициналық көмектің үш негізгі деңгейін қолдану ұсынылады:*

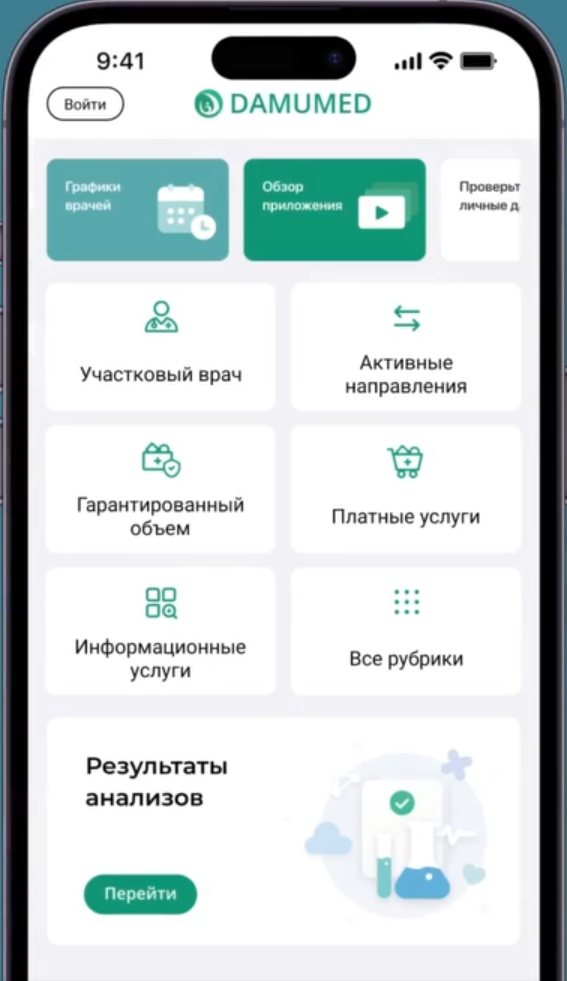
1. Қаланың барлық тұрғындарына жоғары мамандандырылған медициналық көмек көрсетуге арналған жалпы қалалық деңгей. Бұл деңгей медициналық университеттердің консультативтік емханалары, ғылыми-зерттеу институттары және қалалық Диспансерлер сияқты қалалық деңгейдегі емханаларды қамтиды.
2. Ірі қалалар үшін ұсынылатын және бірнеше әкімшілік аудандардың тұрғындарына мамандандырылған көмек көрсетуге арналған медициналық-санитариялық аймақтардың деңгейі. Бұл деңгейде көпсалалы ауруханаларға жататын консультативтік емханалар жұмыс істейді.
3. Аудандық деңгей бір әкімшілік ауданның тұрғындарына негізгі медициналық мамандықтар бойынша медициналық көмек көрсетуге арналған.

Қажет болған жағдайда базалық емханалардағы науқастар аймақтық көпсалалы ауруханалардың мамандандырылған орталықтарына, сондай-ақ ғылыми-зерттеу институттары мен жоғары оқу орындарының консультативтік емханаларына консультацияға жіберілуі мүмкін. Алайда, барлық мамандандырылған көмек бірінші кезекте жалпы практика дәрігерлерінің, учаскелік дәрігерлердің диагностикалық және емдеу практикасындағы жұмысын қолдауға бағытталуы керек.

Амбулаториялық-емханалық мекемелерде халыққа медициналық көмек қабылдау кезінде де, үйде де көрсетіледі. Бұл көмектің көлемі мекеменің түріне және дәрігердің мамандығына байланысты бөлінеді.

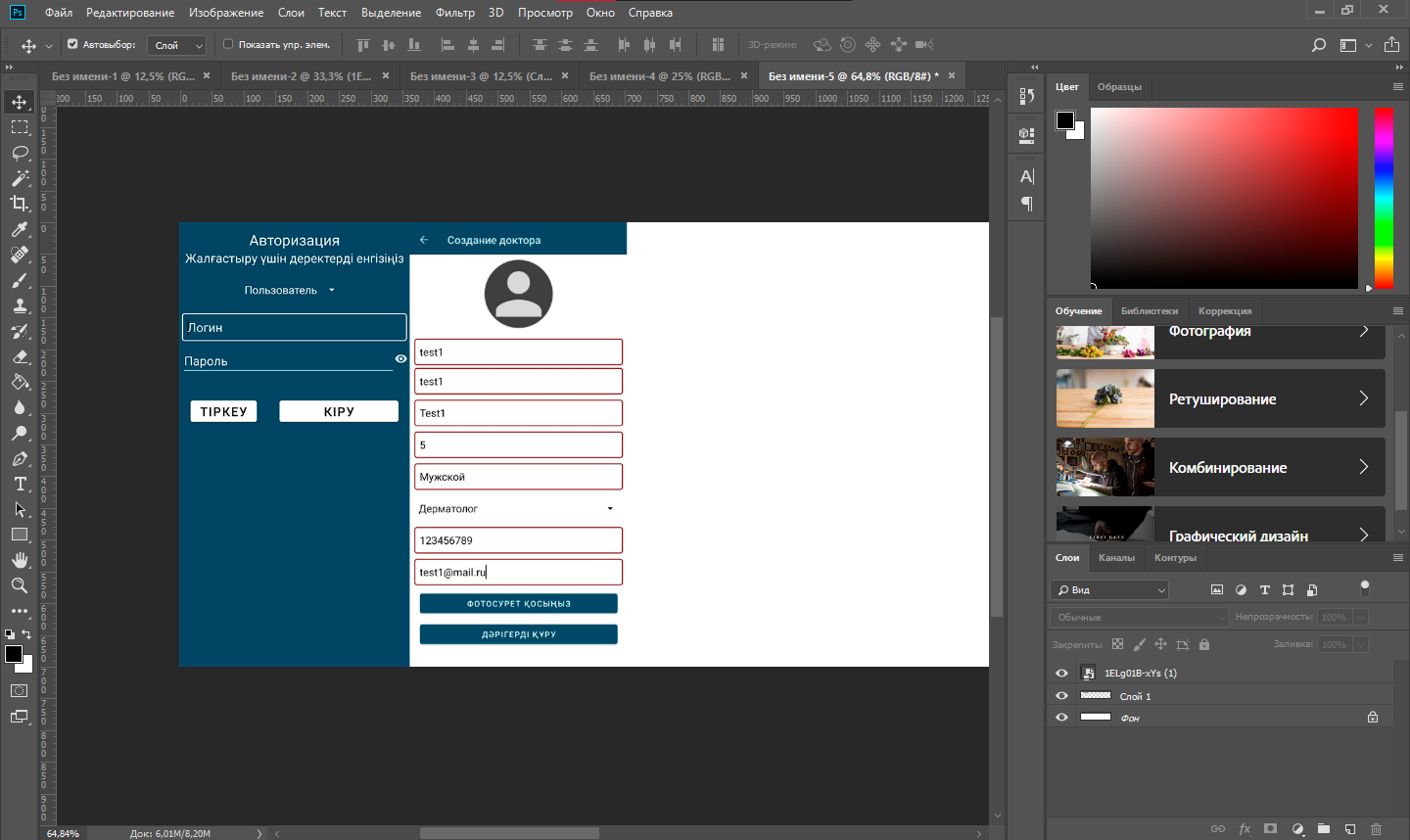
**2.2 Қолданба құрылымын және мәліметтер базасын әзірлеу**

Денсаулық сақтау саласындағы мобильді қосымшаны жасамас бұрын біз басқа бөлімдердің мобильді қосымшаларының үлгілерін алдық. Мысалы, біз damamed қосымшасын қарап шықтық және маңызды сәттерді атап өттік бізге ұнады (сурет.2.1).



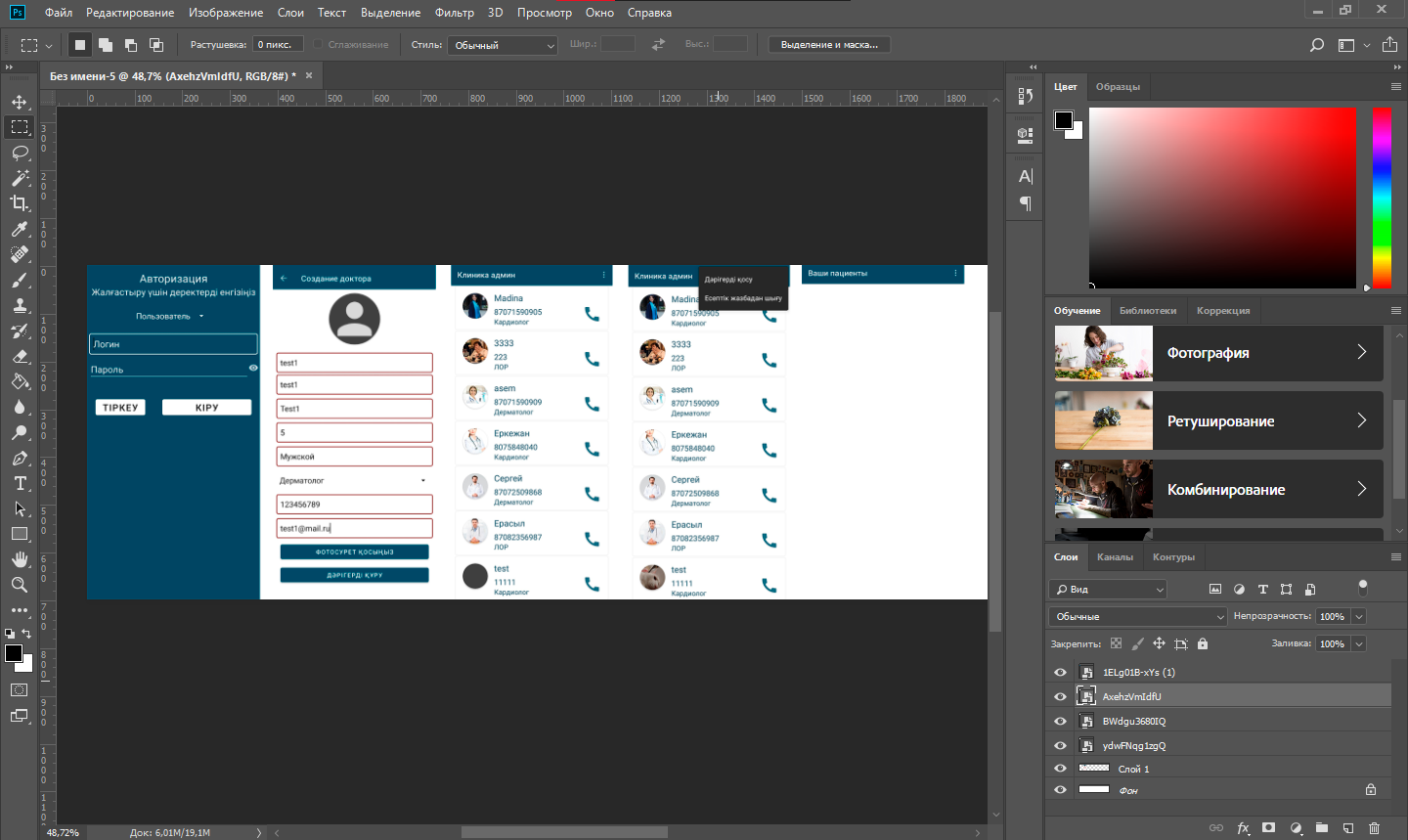
2.1-сурет damumed қосымшасы

Осындай мобильді қосымшаларды мұқият тексергеннен кейін біз денсаулық сақтау мәселелері бойынша мобильді қосымшаның дизайнын жасай бастадық. Біз Adobe Photoshop бағдарламасында Дизайн жасадық. Adobe Photoshop бағдарламасы басқа бағдарламаларға қарағанда қарапайым және интерфейс түсінікті болғандықтан, біз бұл бағдарламаны қолдандық (сурет.2.2).



2.2-сурет Мобильді қосымшаның дизайнын жасау процесі

Біз интерфейске қатысты барлық нәрсені жасадық: авторизация беті, рөлдерді бөлу, дәрігерлер тізімі, қабылдау уақыты және профиль беті (сурет.2.3).



2.3-сурет Adobe Photoshop бағдарламасындағы мобильді қосымшаның дизайны

Біз мобильді қосымшаның дизайнын жасадық. Енді бізге мәліметтер базасы қажет, бұл тапсырма үшін android studio бағдарламалық жасақтама пакетімен, атап айтқанда firebase-пен бірге берілген мәліметтер базасы өте қолайлы.

Дұрыс жұмыс істеу үшін бізге email авторизациясы және пайдаланушылардың аватарларын сақтау мүмкіндігі қажет. Firebase тапсырманы өте жақсы орындайды. (сурет.2.4).



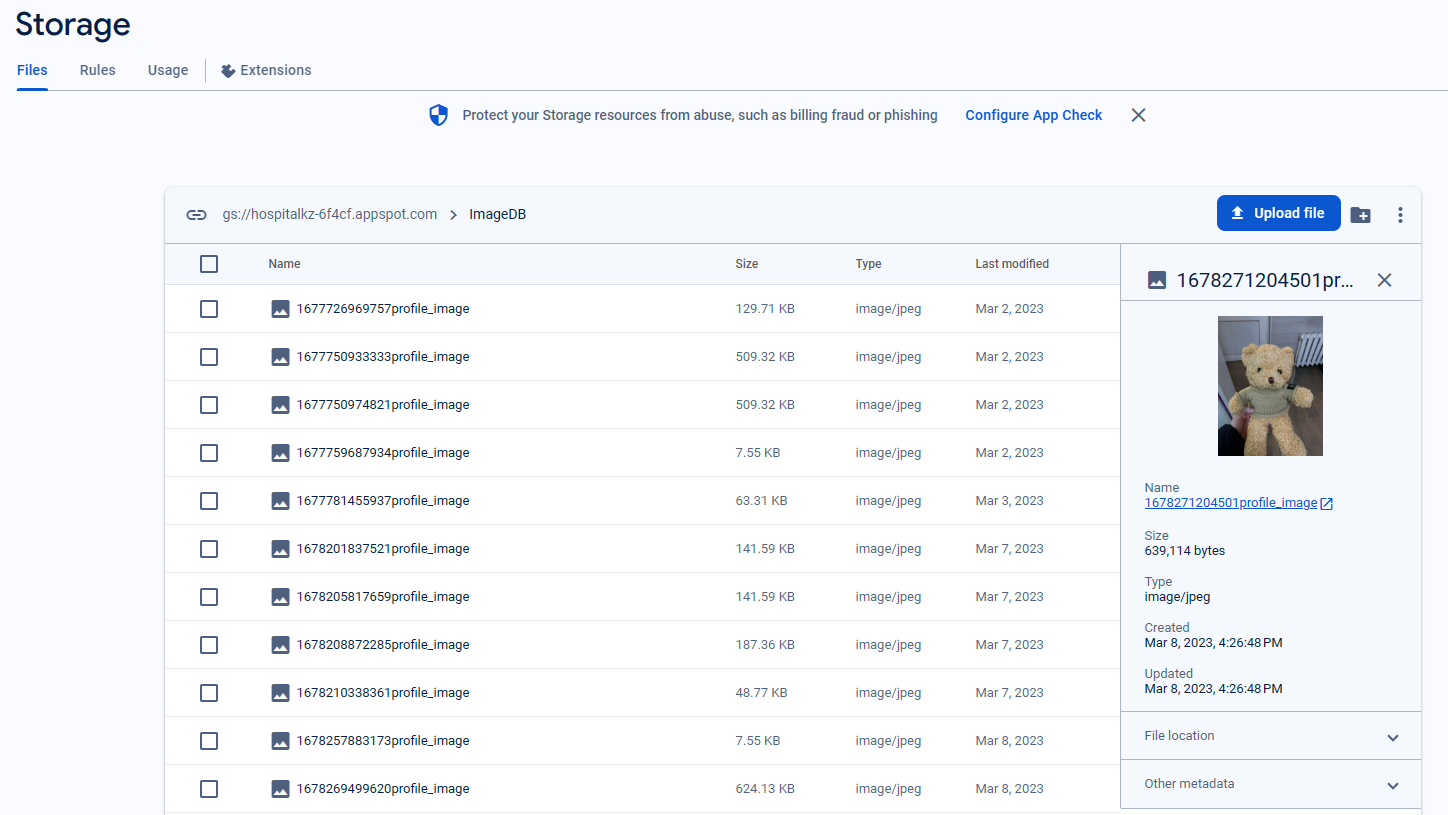
2.4-сурет Мобильді қосымшаның мәліметтер базасы

Сондай-ақ, біздің қосымшаны іске асыру үшін бізге рөлдерді нақты бөлу қажет, біздің жағдайда олар 4, пайдаланушы, дәрігер, кесте және әкімшілер болады

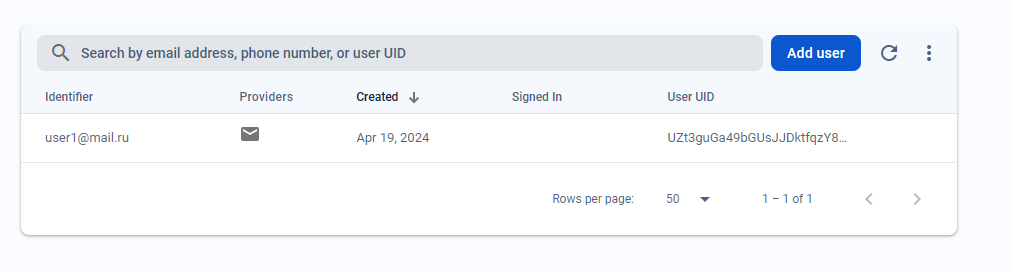
4-кесте Рөлдердің сипаттамасы

|  |  |
| --- | --- |
| **Рөлі** | **Сипаттама** |
| Администратор | Емхананың мобильді қосымшасының әкімшісі дәрігерлердің кестесін жаңартуға және басқаруға жауапты. |
| Дәрігер | Дәрігер мобильді қосымшаны қолдана отырып, кестесін қарап шығады, кездесулерге жазылуды қабылдайды немесе жояды, сонымен қатар кездесуге дайындалу үшін пациенттер туралы жазбалар жазады. |
| Кесте | Емхананың мобильді қосымшасындағы белгілі бір дәрігермен байланысты кесте пациенттерге кездесуге арналған қол жетімді слоттарды көруге, ыңғайлы уақытты таңдауға және дәрігермен кездесуге тапсырыс беруге мүмкіндік береді. |
| пайдаланушы | Емхананың мобильді қосымшасын пайдаланушы дәрігерлердің кестесін көре алады, қабылдауға ыңғайлы уақытты таңдай алады, кездесуге жазыла алады, жазбаларынан бас тарта немесе өзгерте алады. |

Аватарларды жүктеу үшін біз Firebase-те сақтау атрибутын қолдандық (сурет.2.5).



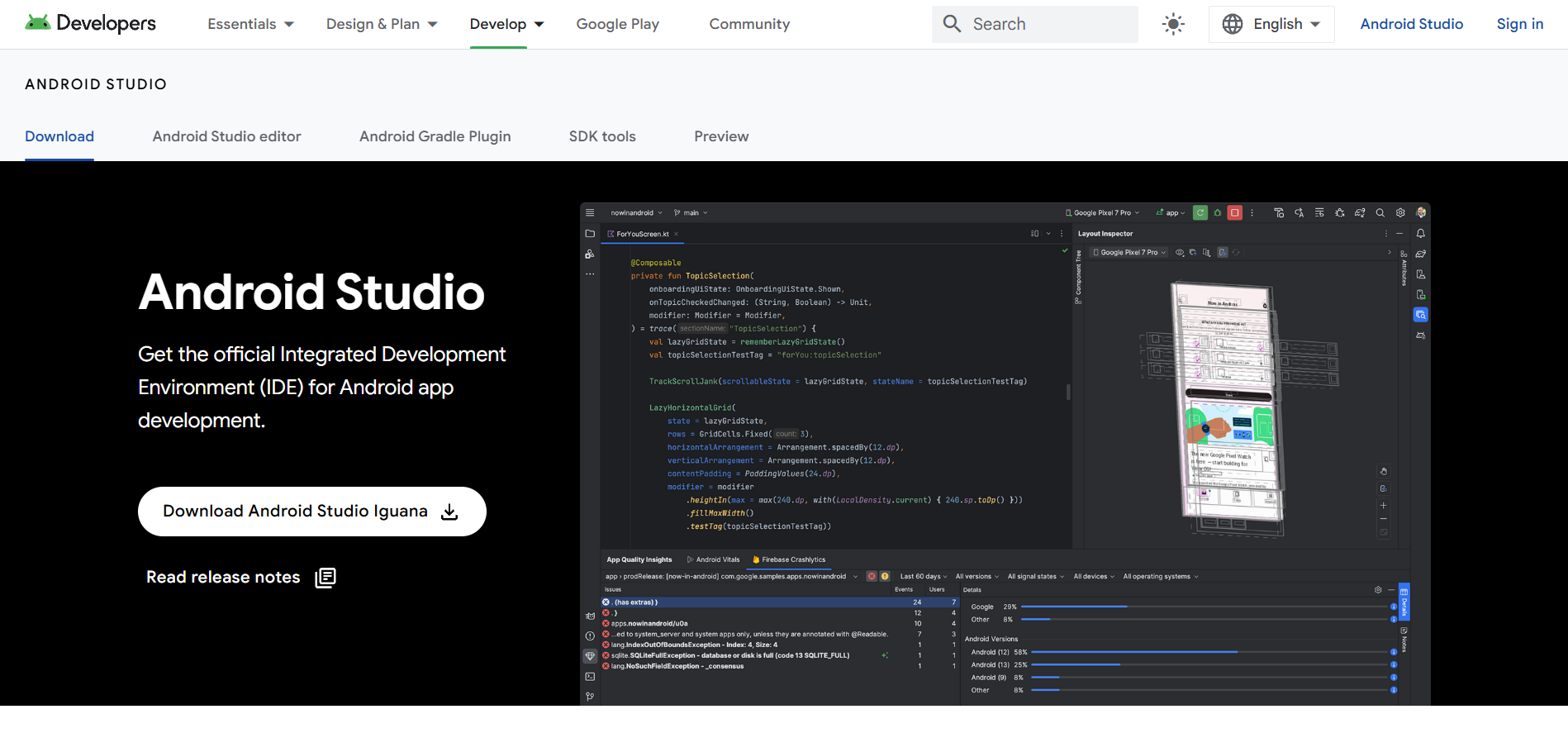
2.5-сурет Firebase-те аватарларды сақтау

Авторизация үшін Firebase-те кіріктірілген функция қолданылады. Firebase авторизация немесе тіркеу кезінде поштаның пайдаланылғанын автоматты түрде тексереді, Егер пошта көрсетілмесе, қолданбада авторизация қатесі орын алады 

2.6-сурет Firebase-те пошта арқылы тіркелу мысалы

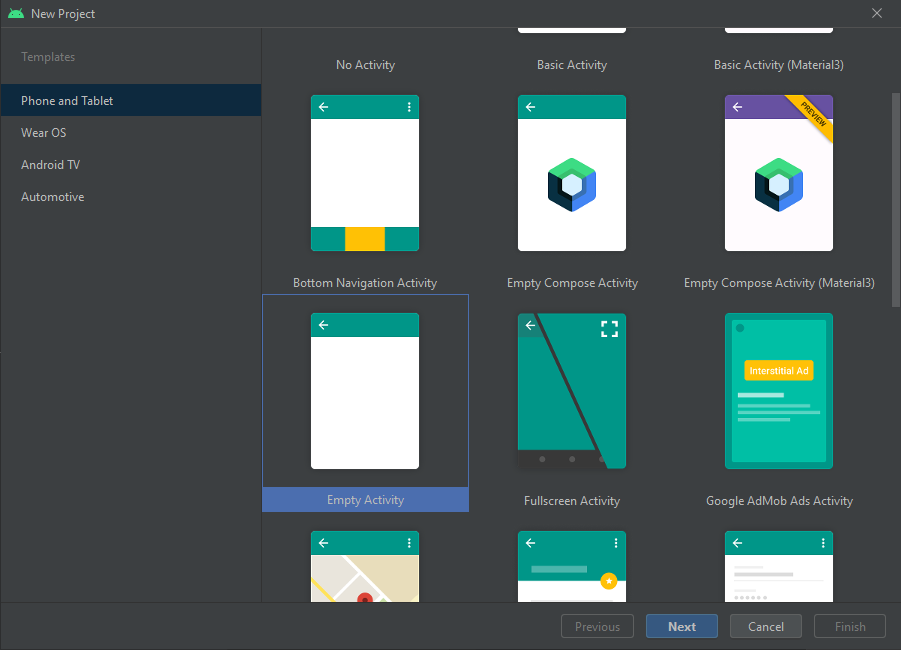
**2.3 Денсаулық сақтау мәселелері бойынша кеңес беру үшін мобильді қосымшаны әзірлеу**

Мобильді қосымшаның соңғы дизайны мен мәліметтер базасы дайын болған кезде, біз кодты қолдана отырып, мобильді қосымшаны құра бастаймыз. Ол үшін біз соңғы технологияларды қолдана отырып қосымшаларды жасауға мүмкіндік беретін Android Studio бағдарламалық жасақтамасын қолданамыз. Android Studio-да жобаны бастамас бұрын, біз оны алдымен интернеттен тікелей жүктейміз (сурет.2.7).



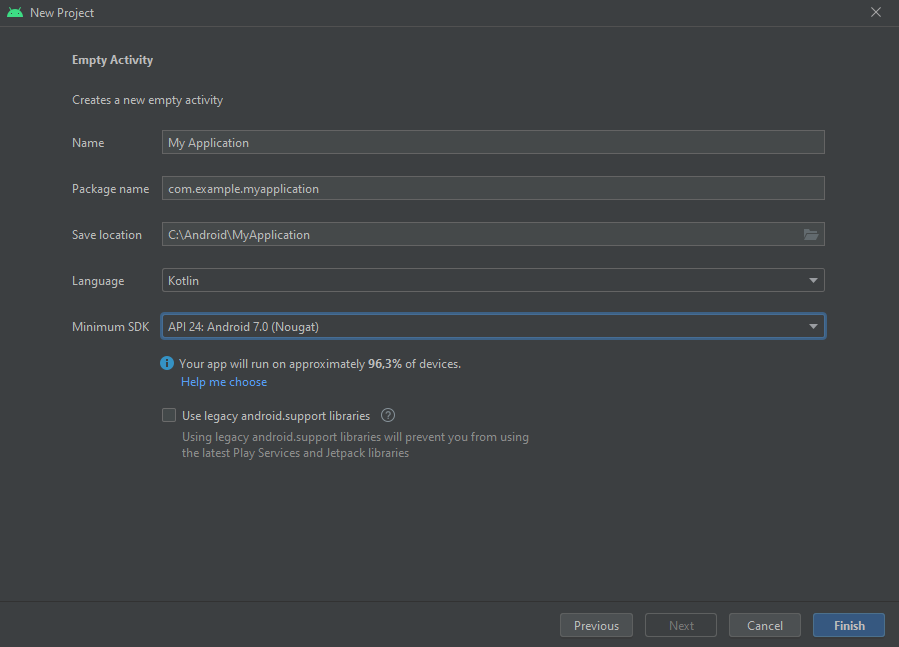
2.7-сурет Android studio бағдарламасын ресми сайттан жүктеп алыңыз

Орнатушыны ашып, жүктелген бағдарламаны компьютерге орнатыңыз. Бағдарламаны орнатқаннан кейін оны іске қосыңыз, "File" -> "New" -> "New Project" бөліміне өтіп , Empty activity таңдап, "Next" батырмасын басыңыз (сурет.2.8).



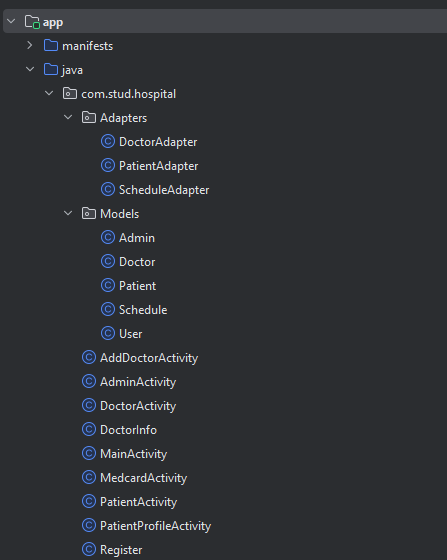
2.8-сурет Open Server-ден консольді ашу

Келесі бетте біз жобаны құру терезесін күтеміз, атауын, бағдарламалау тілін және API нұсқасын көрсетеміз. Біздің жағдайда біз API 24: Android 7.0 (Nougat) қолданамыз. Бұл ең жақсы нұсқа, өйткені бұл API android ОЖ смартфондарының 96% - дан астамын қолдайды және сонымен бірге барлық қажетті функционалдылыққа ие (сурет.2.9).



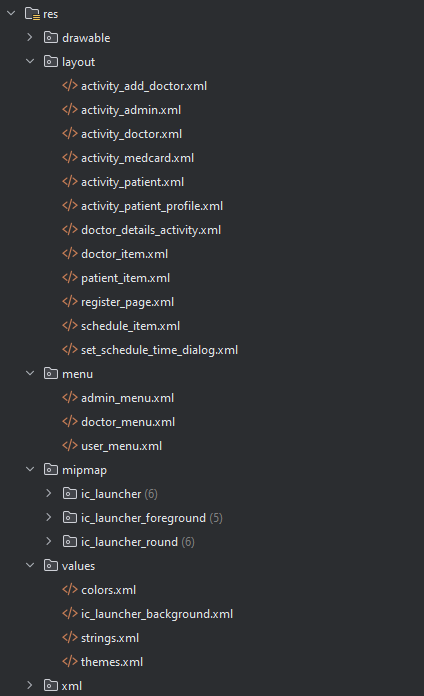
2.9-сурет Android studio-да қосымшаны құру

"Java" қалтасында "com.stud.hospital" біздің activity үшін адаптерлер, модульдер мен кодты қамтитын құрылымға біздің қосымшамыздың негізгі коды болады (сурет.2.10).



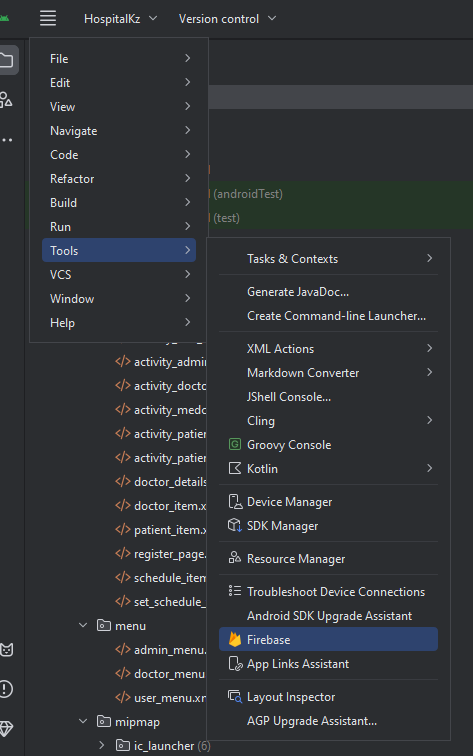
2.10-сурет Vue фреймворкын Laravel-ге орнату

"Res" қалтасында жобаның барлық ресурстары бар белгішелер, қабаттар, мәтінге арналған стильдер, түстер және қаріптер қосымшаны жылдам әрі ыңғайлы дамыту үшін (2.11-сурет).



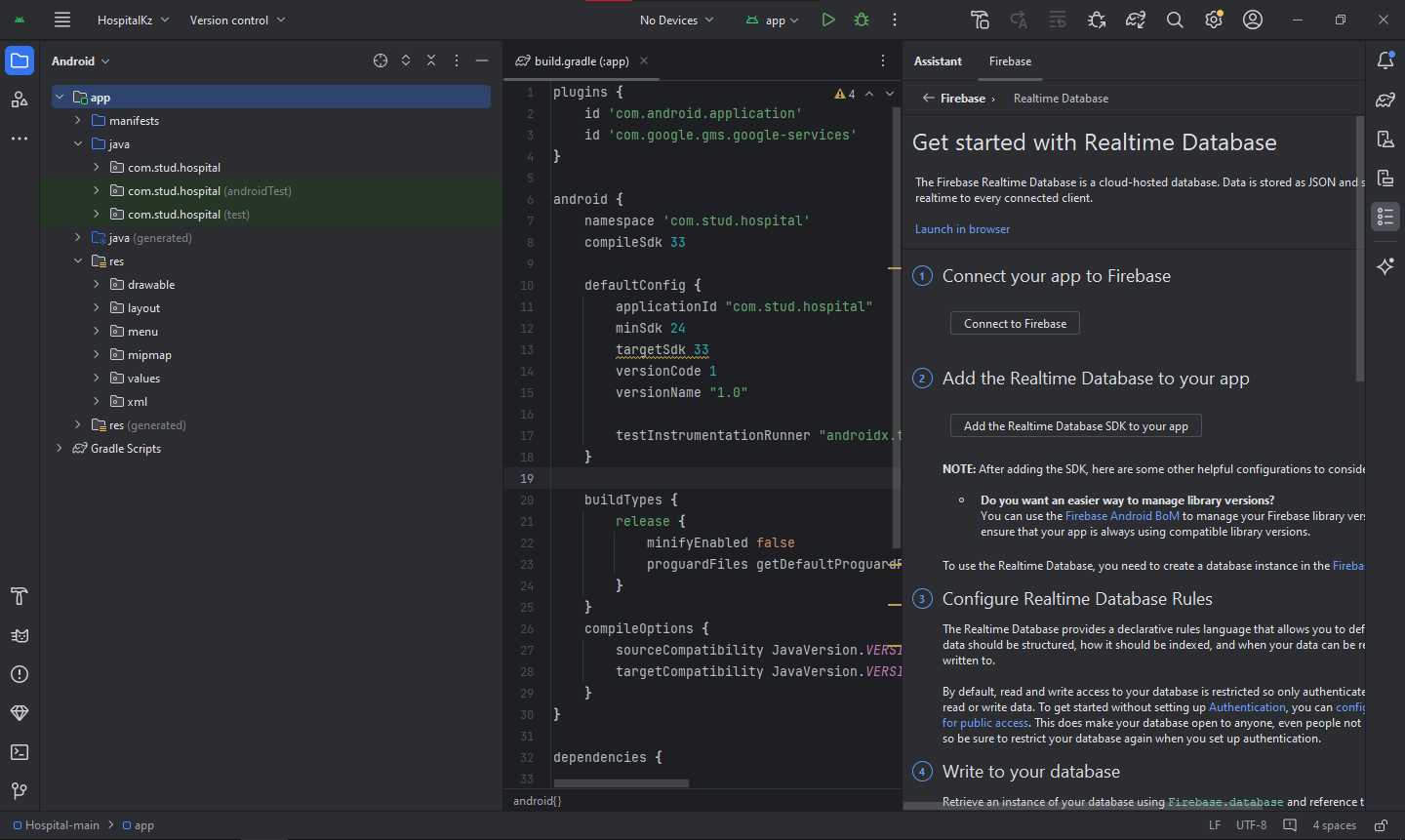
2.11-сурет Res қалта құрылымы

Әрі қарай, Негізгі мәзірді нұқыңыз жоғарғы оң жақта android-де "құралдар" - > "firebase" тармағын таңдаңыз (2.12-сурет).



2.12-сурет Firebase жүйесіне кіру

Firebase-ке өткеннен кейін кітапханаларды қосуға арналған панель ашылады, Realtime Database таңдаңыз және базаға қосылыңыз (2.13-сурет).



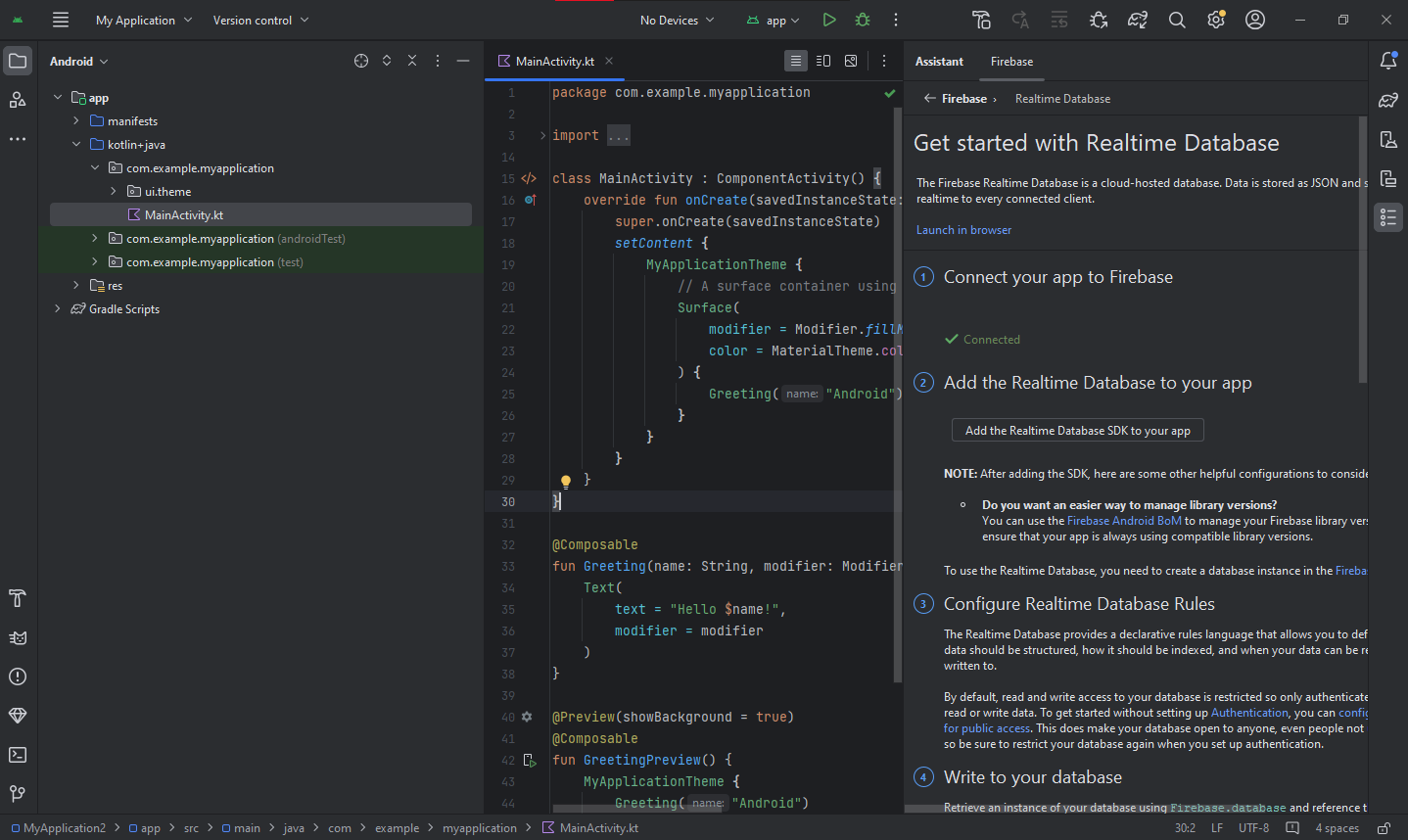
2.13-сурет Firebase Кітапханаларын Қосу

Біз Firebase-ке қосылуды басамыз және жаңа дерекқорды авторизациялау және құру үшін браузерге лақтырамыз, оған атау береміз, растаймыз және жаңа дерекқорды құрудың аяқталуын күтеміз (2.14-сурет).



2.14-сурет Мәліметтер базасын құру

Дерекқор жасалғаннан кейін біз оны android studio-мен байланыстырамыз және база қосылады (2.15-сурет)

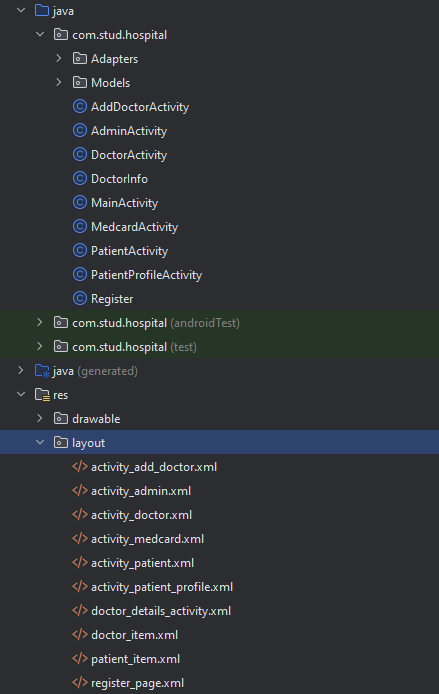


2.15-сурет Қосылған мәліметтер базасы

**2.4 Мобильді қосымшаны құруға арналған бағдарламалық жасақтама**

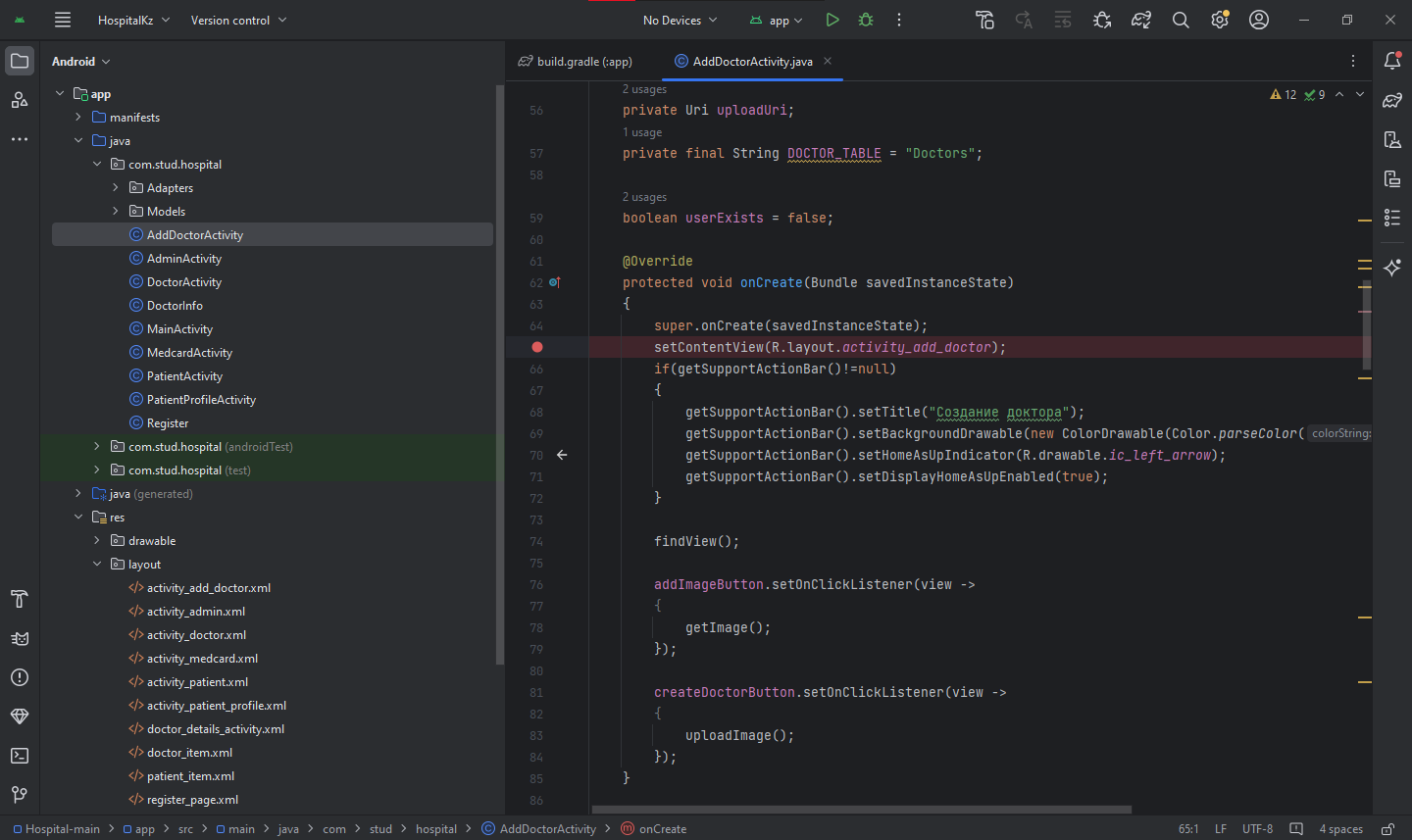
Android studio-да "res" -> "layout" қалтасында біз барлық қажетті әрекеттерді жасаймыз және қосымшаның визуалды бөлігін жасау үшін xml кодын жазамыз.

1. Layout қалтасын тінтуірдің оң жақ түймешігімен басып, layout resource file жасаңыз.
2. Қолданбаның барлық қажетті қабаттарын жасаңыз.
3. Содан кейін "java" -> "com.stud.hospital" папкасында қажетті модульдерді жасаймыз.



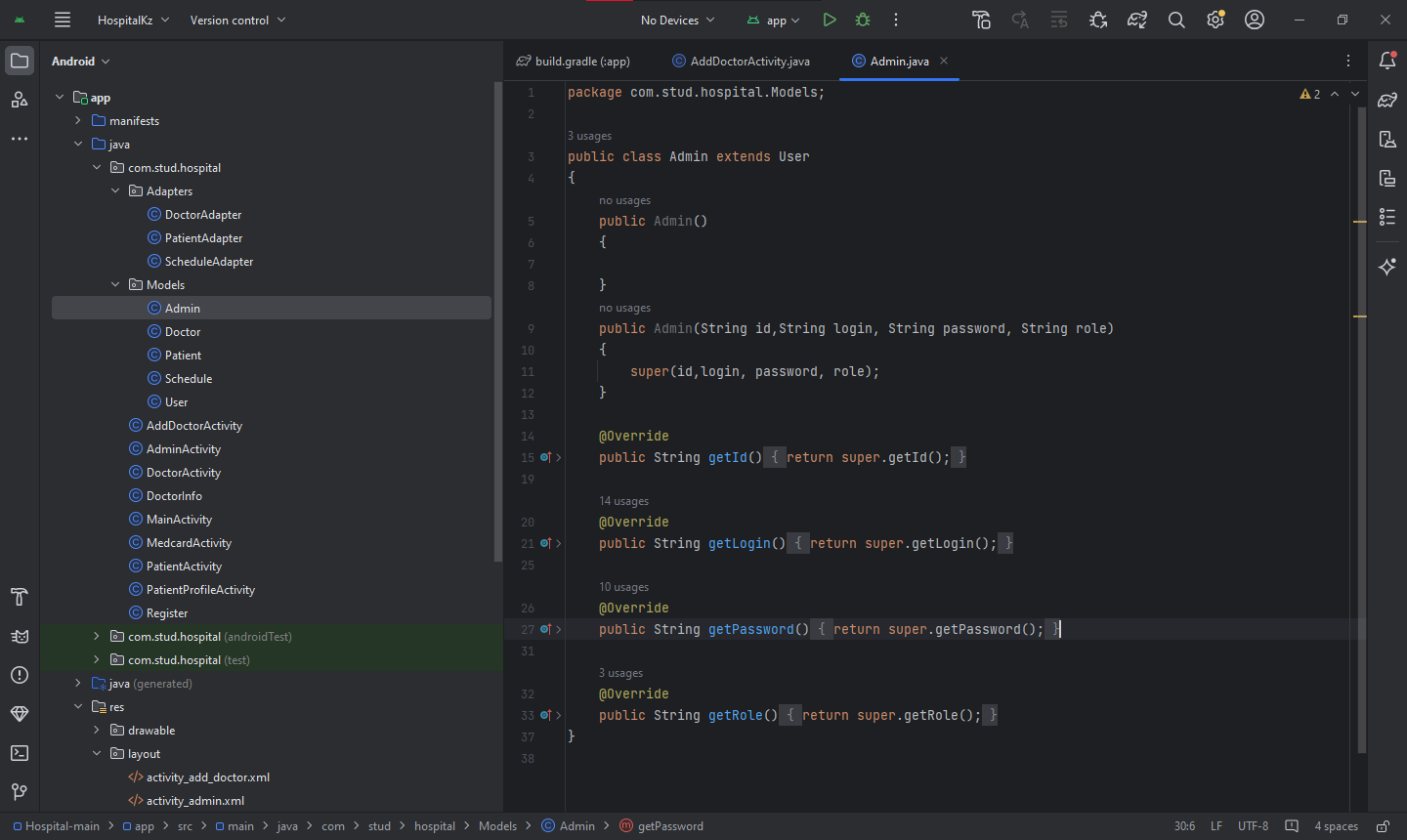
2.16-сурет Қажетті қабаттар мен модульдерді құру

Негізгі логика сәйкес әрекет атауларының сыныптарында жазылады, мысалы, дәрігерді қосу коды onCreate функциясындағы setContentView(R.layout.activity\_add\_doctor) әдісін қолданатын activity\_add\_doctor.xml қабатымен және AddDoctorActivity сыныбымен байланысты (2.17-сурет).



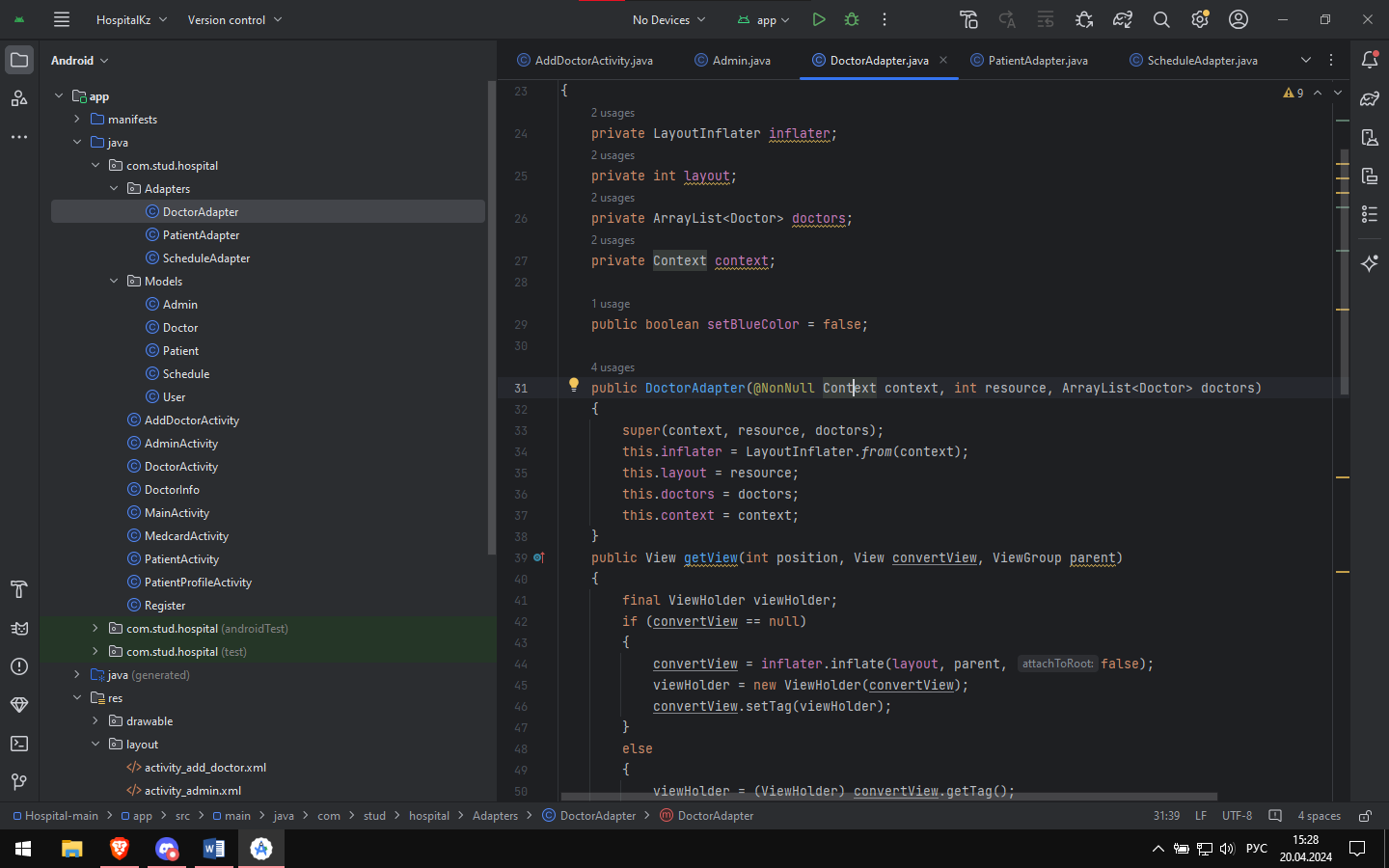
2.17-сурет AddDoctorActivity файлындағы onCreate әдісі

Models қалтасында негізгі рөлдер мен олардың көмекші әдістері бар, кодта рөл алу үшін көмекші әдістер қажет, мысалы, кейбір функциялардың жұмысын шектеу (2.18-сурет).



2.18-сурет Models қалтасындағы admin сыныбы

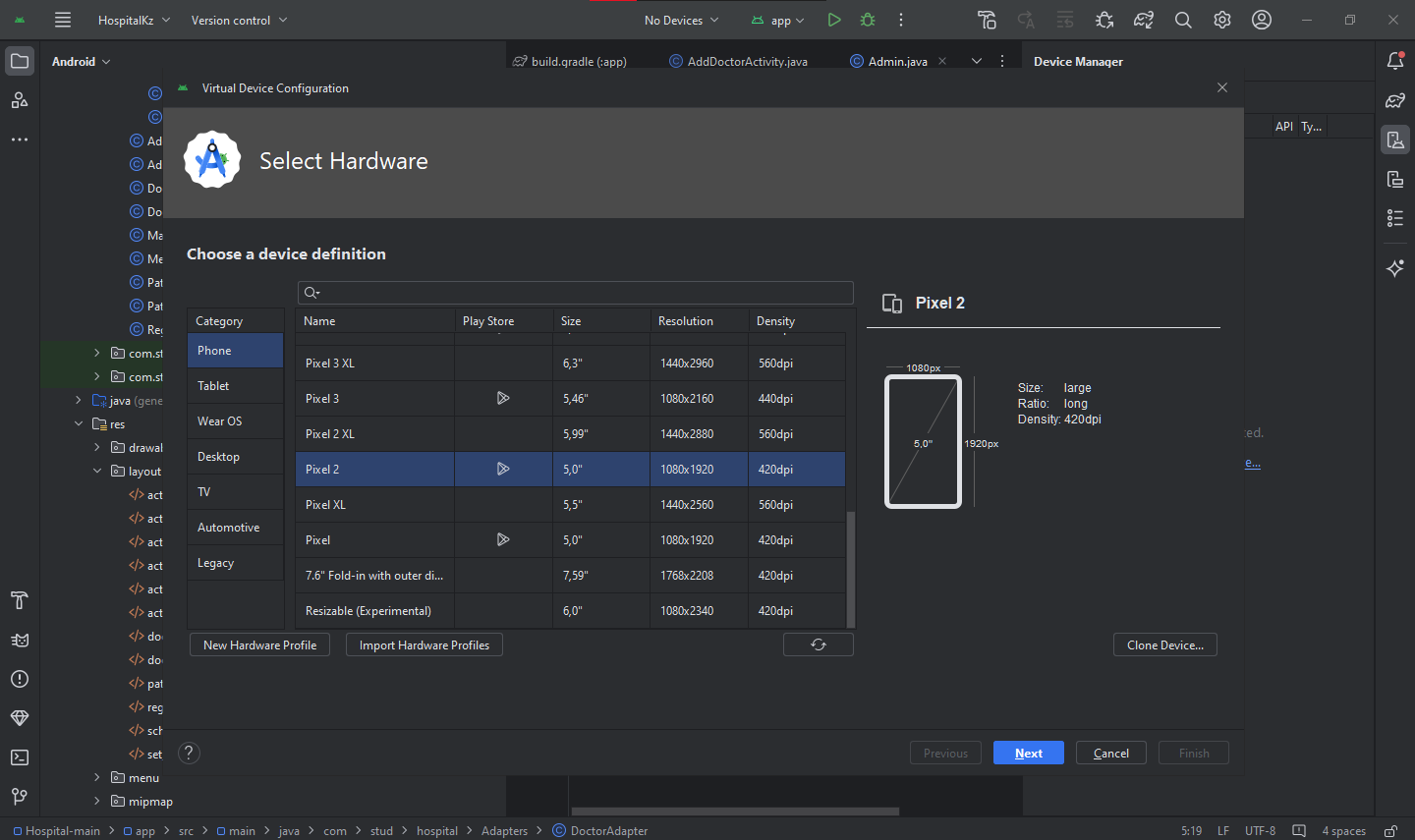
Дәрігер, пациент және кесте рөлдері үшін біз адаптерлер жасаймыз, олар қосымшаның жұмысы кезінде жасалған кез-келген іс-әрекеттер туралы ақпарат алу үшін қажет, біз адаптерді әкімші рөліне әзірлемедік, өйткені оның міндеті тек дәрігерлерді құру болып табылады. Адаптер әкімшіге қажет емес, ол тек кодты шамадан тыс жүктейді, нәтижесінде қосымшаның өнімділігі нашарлайды (2.19-сурет).



2.19-сурет DoctorAdapter функциясы

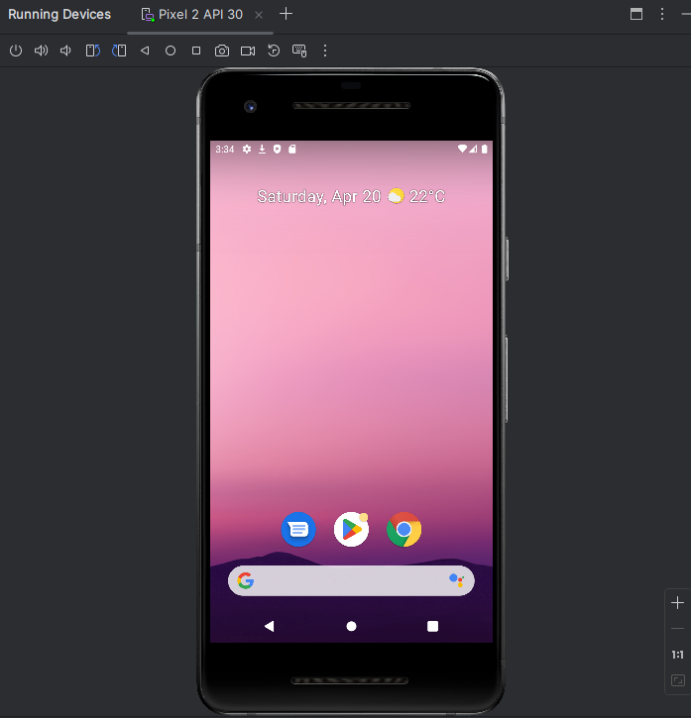
Барлық қажетті кодты жазғаннан кейін біз қосымшамызды тексеруіміз керек, ол үшін android studio-да қосымшаны тексеруге арналған эмуляторлар бар, біз олардың ең қарапайымын орнатамыз

Эмуляторды орнату үшін оң жақтағы Device Manager түймесін басыңыз (2.20-сурет).



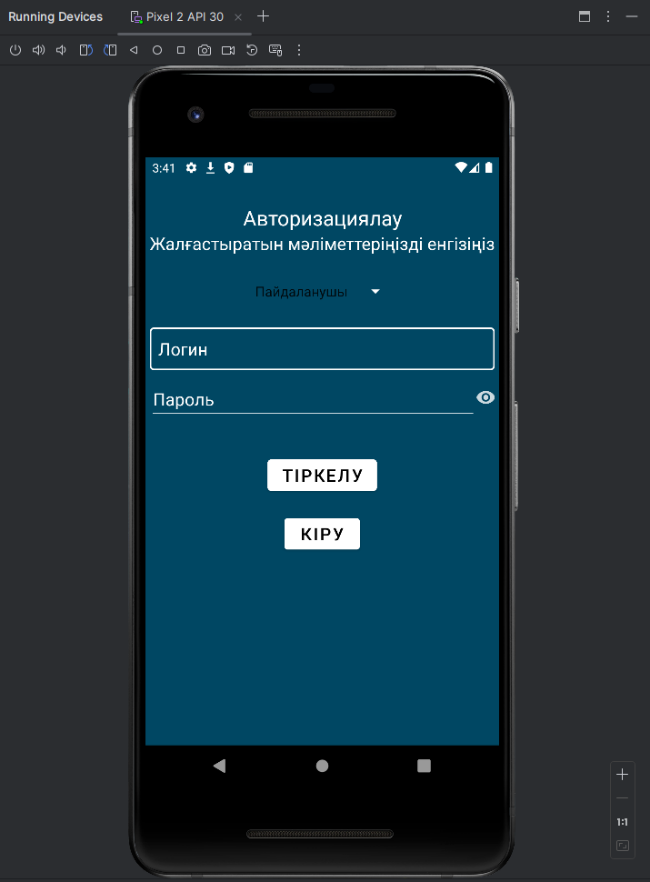
2.20-сурет Pixel 2 виртуалды құрылғысын жасаңыз

Виртуалды құрылғыны жасағаннан кейін орнатылған құрылғының терезесі пайда болады, біздің жағдайда бұл pixel 2 (2.21-сурет).



2.21-сурет Pixel 2 виртуалды құрылғысы бар терезе

Pixel 2 сәтті іске қосылғаннан кейін түймесін басыңыз Іске қосу қолданбасы және біздің қолданба іске қосу (2.22-сурет).



2.22-сурет Қолданбаны Pixel 2 виртуалды құрылғысында іске қосыңыз

**ҚОРЫТЫНДЫ**

Бүгінгі таңда заманауи технологиялар дәуірінде қоршаған ортаны түсінуге және жақсартуға көмектесетін көптеген инновациялық құралдар бар. Осындай құралдардың бірі-Android үшін мобильді қосымша.

Денсаулық сақтау саласындағы мобильді қосымша денсаулық мәселелері бойынша консультациялар мен қолдауды қамтамасыз ететін Интернет желісі арқылы қолжетімді бағдарламалық шешімдер кешені болып табылады. Тіпті мобильді қосымшаны жасаушылар сіздің кеңесшіңіз бола алады. Қазіргі уақытта денсаулық сақтау үшін мобильді қосымшаларды пайдалану өзекті бола түсуде.

Адамдар әртүрлі мақсаттармен Android үшін мобильді қосымшалар жасайды: медициналық ақпарат беру, Дәрігерлермен кеңесу, денсаулықты бақылау, медициналық қызметтерге қол жеткізу және т. б.

Әрбір мобильді қосымша серверде жұмыс істейді, бұл оның пайдаланушыларға тұрақты қол жетімділігін қамтамасыз етеді. Соңғы жылдары мобильді дамудың жаңа технологияларының пайда болуымен көптеген бағдарламалау тілдері мен бағдарламалық құралдар жасалды. Атап айтқанда: Java, Kotlin, Swift, React Native және басқалары.

Денсаулық сақтау саласындағы Android мобильді қосымшалары Дәрігерлермен кеңесу, денсаулық көрсеткіштерін қадағалау, медициналық деректерге қол жеткізу және т.б. сияқты көптеген мүмкіндіктерді ұсынады.

Денсаулық сақтау саласындағы сапалы Android мобильді қосымшасы мұқият жоспарлауды, әзірлеуді және тестілеуді қажет етеді. Бағдарламалау тілдерінен фреймворктерге дейін SDK-ге дейін мобильді қосымшаларды құруға арналған көптеген технологиялар мен құралдар бар.

Android үшін мобильді қосымшаларды әзірлеуге арналған танымал құрылымдардың бірі-Android SDK. Оның артықшылықтарының ішінде мыналарды атап өтуге болады: интуитивті және қарапайым синтаксис, үлкен және белсенді әзірлеушілер қауымдастығы, пайдаланушы интерфейсін құруға арналған қуатты құралдар, кіріктірілген қауіпсіздік жүйесі және т. б.

Осылайша, Android мобильді қосымшалары Денсаулық сақтау саласында маңызды рөл атқаруды жалғастыруда және оларды құру жаңа технологиялар мен құралдардың арқасында қол жетімді және қарапайым бола бастады.

**ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ**

1. Head First. Программирование для Android. Гриффитс Д. Питер, 2016, ISBN: 978-5-496-02171-5.
2. Android NDK. Руководство для начинающих. Сильвен Ретабоуил. ДМК Пресс, 2016, ISBN: 978-5-97060-394-9.
3. Android. Программирование для профессионалов. Билл Филлипс, К. Стюарт, Кристин Марсикано. Питер, 2017, ISBN: 978-5-4461-0413-0.
4. Эффективное использование потоков в операционной системе Android. Технологии асинхронной обработки данных. Андерс Эрландссон. ДМК Пресс, 2015, ISBN: 978-5-97060-168-6.
5. Android Security Internals: An In-Depth Guide to Android's Security Architecture. Николай Еленков. No Starch Press, 2014, ISBN: 978-1-5932-7581-5.
6. Android Security Cookbook. Keith Makan, Scott Alexander-Bown, Packt Publishing Ltd., 2013, ISBN: 978-1-7821-6716-7.
7. Android Hacker's Handbook. Joshua J. Drake, Zach Lanier, Collin Mulliner, Pau Oliva Fora, Stephen A. Ridley, Georg Wicherski. Wiley, 2014, ISBN: 978-1-1186-0864-7.
8. Bulletproof Android: Practical Advice for Building Secure Apps (Developer's Library). Godfrey Nolan. Addison-Wesley Professional, 2014, ISBN: 978-0-1339-9332-5.
9. Android User Interface Design: Implementing Material Design for Developers (2nd Edition) (Usability). Ian G. Clifton. Addison-Wesley Professional, 2015. ISBN: 978-0-1341-9140-9.