

## ВСТУП

Однією з характеристик культури ХХ ст. полягає у розширенні сфер її впливу на людей за допомогою засобів масової інформації, таких як радіо, кіно, телебачення, які були продуктами наукової революції. Їх значення в посиленні обміну інформацією не переоцінити.

Якщо на рубежі 19-го і 20-го століть з'явилося нове мистецтво - засвітився екран кінотеатру - виявляється, це було потрібно не «комусь», а тим самим людям, які були зібрані перед цим екраном.

Аудіовізуальних пристроїв вводу-виводу в наші дні вдосталь, як і фільмів та серіалів. Усі дивляться відео, будь то в телевізорі, соціальних мережах або кінотеатрах. Кіноіндустрія, безперечно, є одним із найефективніших секторів сучасного суспільства.

Ситками та комедійні шоу нас розсмішують, психологічні трилери допомагають бачити світ із покращеної точки зору, а історичні фільми допомагають зрозуміти, звідки ми люди. Кожне відео та кожен фільм відображають суспільство та трансформують думки. Беззаперечна необроблена сила фільмів, сила аудіовізуальних матеріалів проявлялась і використовувалась політично, соціально та економічно протягом історії.

Фільми нас надихають. Хороший фільм розважатиме, навчатиме та надихатиме глядача різними способами. Подумайте, наприклад, про вплив пісень на людей. Вони змушують нас замислитися. Вони роблять нас співчутливими. Вони надихають нас допомагати іншим і робити добро людству та для нього. Романтичні фільми, навпаки, нагадують нам, чому любов важлива і чому за неї варто боротися. Вони змушують нас плакати і сміятися з наших власних романтичних вад, отже, допомагаючи нам більше розуміти наших партнерів та членів сім'ї. Вони роблять життя вартим життя - вони змушують нас відчувати себе живими.

Крім того, доступні послуги перекладу полегшили людям перегляд і розуміння фільмів, що навчають медитації та уважності, більшість з яких

написані на азіатських мовах. Є багато фільмів, сюжети яких дають нам підставу щоранку підніматися та вирушати у світ із надією та оптимізмом. Вони заохочують нас перемагати особисті болі та позитивно впливати на життя інших людей. Фільми можуть сприяти поінформованості про різні аспекти життя. Людям потрібно нагадувати про важливість формальної освіти, а також спільних навчальних занять, таких як мистецтво та спорт. Шкільні фільми підкреслюють це значення та дають зацікавленим сторонам освіту ідеї щодо вдосконалення освітніх систем у різних частинах світу. Крім того, фільм дає нам зрозуміти негативні наслідки зловживання наркотиками, алкоголем та наркотиками. Телепрограми про злочини та дії також попереджають нас про небезпеку злочинної діяльності, тероризму та війни.

Говорячи про війну, фільми допомагають людям зрозуміти жорстокість бездомних та жалюгідних таборів біженців. Фільми пробуджують почуття співпереживання у людей, які ніколи не переживали громадянської війни з перших вуст. Вони допомагають нам відчувати відповідальність за своїх братів і сестер, які живуть у країнах, що постраждали від війни. Це те, що сприяло зростанню такої кількості благодійних організацій та цільових фондів.

Фільми користуються великою популярністю, а отже і сучасні програмні засоби, які полегшують життя людей, надають можливість перегляду фільмів, слідування за новинками кіно, ведення статистики, категоризації та іншого.

В такий період реформації та покращення обізнаності й вимогливості споживачів щодо того, якої якості повинен бути продукт, критичним є впровадження новітніх технологічних рішень. На піку розвитку знаходить сфера мобільних додатків та різноманітних сервісів. Відкриваються нові можливості для підприємців і, взагалі, інформація стає все більш доступною. Перелік послуг постійно розширюється.

Виникає потреба в дослідженні питання використання додатків та принципів розробки й впровадження.

Метою даної магістерської роботи є розробка інформаційної системи та бази даних художнього контенту на базі Android, яка призначена для

забезпечення доступу до бази даних усіх шоу, фільмів, серіалів. Відомостей про дату виходу, жанр, тривалість, середню оцінку від глядачів, кількості та назв сезонів та епізодів конкретного продукту, різноманітної довідкової інформації.

Відповідно до зазначеної мети вирішуються такі завдання:

- дослідити, що таке мобільний додаток;
- надати характеристику системам, засобам розробки мобільних додатків для платформи Android;
- визначити, як можуть використовуватися додатки, яка їх роль в полегшенні життя сучасної людини;
- зробити аналіз характеристик та властивостей додатків, що є лідерами ринку;
- розробити власний мобільний додаток.

Об'єктом дослідження є використання додатку для доступу до інтернет бази даних.

Предметом дослідження є сукупність теоретичних, та практичних засад створення й впровадження мобільного додатку.

Ідея роботи полягає в описі різноманітних технологій розробки сучасного мобільного додатку, покращення методів для доступу до даних на мобільних пристроях, надання статистики та ін.

Наукова новизна отриманих результатів полягає в реалізації та використанні власної системи доступу до даних серверних технологій в мобільному додатку, розробці мобільного додатка. Додаток не має аналогів з описаними функціями.

При вирішенні поставленого завдання використовувалися досягнення в області розробки програмних систем.

Особлива увага приділена огляду принципів роботи, основних можливостей і переваг використаних інструментів і технологій.

Результата та деякі з положень були обговорені на студентській науковій конференції.

# 1 АНАЛІТИЧНА ЧАСТИНА

## 1.1 Призначення й область застосування

Програмний додаток використовуватиметься користувачами у яких є мобільний пристрій або планшетний комп'ютер на операційній системі Android та доступ до мережі інтернет. Призначення додатку - надання користувачам отримувати повну інформацію про новинки в галузі кіно, слідкувати за виходом нових епізодів чи цілих картин. Вести власну історію переглядів, керувати різноманітними списками, чартами. А також переглядати статистику на власному мобільному пристрої.

Функціональне призначення додатка - отримання детальної інформації про всі шоу, серіали, фільми. Бути в курсі всього, що відбувається в індустрії кіно. Мати доступ до хмарної бази даних. Здійснювати пошук, робити помітки, керувати списками, добірками. Користувач може переглянути статистичні дані та поділитися з друзями або в соцмережах. Переглядати всю необхідну інформацію завдяки зручному інтерфейсу та продуманій навігації.

Експлуатаційним призначенням додатка є економія часу на пошук та отримання даних, а також зручність. Підтримуються всі актуальні версії операційної системи. Користування додатком не потребує особливих навичок.

Потреба в розробці даного програмного продукту полягає в тому, щоб надавати людині легкий доступ до глобальної бази даних кіно та використання цієї можливості для проведення власного дозвілля чи іншої активності.

## 1.2 Постановка задачі

Спроекувати та розробити Android додаток для доступу та опрацювання великих об'ємів даних. Їх завантаження, здійснення пошуку, фільтрації, сортування. Відображення для користувача системи. В програмі повинні бути функції створення авторських підбірок, реалізована також локальна база даних.

Повинна бути можливість відзначити ту чи іншу картину, як «переглянуту» або «пропустити», а також додати у свою колекцію. Крім того, на основі дій користувача буде вестися статистика, яка включатиме кількість переглянутих серій, загальний час витрачений, кількість фільмів в списках переглянутих, списку спостереження та в колекції. Програма буде виконана у формі мобільного клієнта. В якості джерела даних буде використовуватися онлайн сервіс «TheTVDB», а також «TMDB». Додаток повинен мати інтерактивний, зручний інтерфейс для перегляду даних з серверу, відображення інформації.

Додаток повинен пропонувати користувачу наступні функції:

- відображення великої кількості даних, отриманих від запиту на сервер;
- наявність функціонального блоку «Серіали», «Списки», «Фільми» та «Статистика»;
- можливість перегляду інформації про найближчі епізоди та фільми;
- можливість перегляду списку нещодавно переглянутих картин;
- наявність фільтру та функції сторткування;
- пошук за допомогою запиту на сервер;
- керування користувацькими списками, збереження локальних даних на пристрої;
- перегляд цифрових видань, видань на диску та популярних фільмів;
- управління власною колекцією;
- ведення статистики з можливістю поширити.

Користувач може зберігати отримані результати, додавати до них власні помітки, обмінюватись результатами з друзями через популярні соціальні мережі.

### **1.3 Аналіз предметної області**

Кількість використовуваних смартфонів на планеті Земля вже перевищило два мільярди. На кожному з цих смартфонів встановлені мобільні

додатки - від годин і погоди до цікавих ігор і калькуляторів. Всі вони важливі і потрібні.

Починаючи з 2017 року свій увійшов у «мобільну» еру. Саме з того часу кількість користувачів мобільних пристроїв перевищила кількість користувачів персональних комп'ютерів на планеті. Велика частина з сервісів, яким користуються люди по всьому світу мають власні, призначені для конкретної цілі, мобільні додатки – від кар'ярні до салону краси. Такий підхід дозволяє компанії здійснювати меркетинг так ефективно, як ніколи раніше.

Статистика ж показує, що середньостатистична людина проводить більше двох годин у день за мобільним пристроєм. І бути в полі зору потенційних клієнтів для компанії, сервісу чи бізнесу стає дуже важливо. Мобільні додатки працюють в півтора рази ефективніше, ніж веб-сайт. Вони зберігають локальні дані на пристрої, що дозволяє швидше видавати інформацію користувачеві. А за потреби навіть користуватися сервісом за відсутності інтернет-з'єднання.

Переломний момент настав, коли з простих телефонів з простими функціями ми отримали складні смартфони з додатками «телефонів», «повідомлень», «годин», «калькуляторів» і багато-багато чого іншого. Так було покладено початок розвитку великого ринку. Безліч розробників роблять свої додатки і завантажують їх у Google Play або App Store. Виключно завдяки їм, такі як ми можемо завантажити те, що нам треба безкоштовно або за невеликі гроші. Сказати, що мобільних додатків багато - нічого не сказати. І нам це безумовно подобається.

У сучасному світі важко уявити собі мобільний пристрій, на якому б не стояло жодного додатка. Вони міцно увійшли в наше життя практично одночасно з планшетами та смартфонами. Тому даний напрямок так стрімко розвивається і захоплює ринок. Все більше підприємців усвідомлюють необхідність розробки мобільного застосування.

Мобільний додаток являє собою розроблену програму для планшетів і смартфонів, яка встановлюється на ту чи іншу платформу і має певний функціонал. Простіше кажучи, виконує певні дії і вирішує поставлене коло

питань. Мобільні додатки зовсім недавно представляли собою тільки гри. Але дуже швидко підприємці зрозуміли, що додаток може допомогти вести бізнес. А також стати сильним маркетинговим інструментом, за допомогою якого можна підвищити впізнаваність і довіра до свого бренду, проводити рекламні кампанії, спростити зворотний зв'язок з клієнтами.

Головною перевагою мобільних телефонів є портативність, ви можете носити їх у кишенях, куди б ви не пішли. Смартфони дали нове обличчя сектору послуг завдяки винаходу мобільних додатків. Мобільні програми - це програми, призначені для виконання певного завдання. Споживання часу зменшилось із популярністю мобільних додатків, і тепер товари та послуги знаходяться на кінчиках пальців. Послуг, які надають мобільні програми та смартфони, безліч.

Інтернет-покупки - одна з найважливіших переваг мобільних додатків, за допомогою яких ви можете отримати улюблений одяг, гаджети, аксесуари та сировину в Інтернеті, не відвідуючи торгові центри. Клієнти можуть переглядати, шукати та купувати ваш улюблений товар за розміром, формою та кольором, який їм більше подобається. Інтернет-покупки - це вид бізнесу, де ви виступаєте як продавцем, так і покупцем. Сьогодні багато людей заробляють на існування за допомогою інтернет-покупок. Інтернет-магазини .

Якщо у вас немає часу готувати їжу вдома, не хвилюйтеся! Якщо у вас встановлено додаток, ви отримаєте з улюбленого готелю приготовану їжу, яку доставлять до вашого порогу за розумними цінами. Сьогодні багато програм надають своїм клієнтам послуги таксі за розумними тарифами. .

Банківські послуги, які мобільні програми надають своїм клієнтам, численні. Клієнти можуть переглядати дебети та кредити свого особистого рахунку, переказувати гроші без бар'єрів часу та відстані, переглядати баланс поточного рахунку тощо. Сьогодні центральний уряд та уряд штату намагаються максимально побудувати безготівкову економіку по всій країні, разом із банками вони запропонували кілька мобільних додатків для досягнення своєї мети побудови безготівкової економіки. Клієнти можуть оплатити всі свої

рахунки в Інтернеті, особливо рахунки за електроенергію, рахунки за телефон, воду та ін., Не стоячи в довгих чергах. Банківські програми.

Не стоячи в довгих чергах, ви можете перевірити наявність місць і забронювати квитки на свій фільм, автобус, поїзд та літак. Сьогодні у нас є окремі програми, визначені для кожної послуги, і вам більше не потрібно мати паперові квитки. Ваші квитки будуть у формі електронного листа або текстового повідомлення. Електронні квитки.

Сьогодні люди перебувають у світі напруги та стресу. Розважальні програми доставляють надзвичайне задоволення і знімають стрес від напружених умов праці та стресового життя. Перегляд фільмів, відео або прослуховування пісень зменшить стрес і напругу та дасть змогу відпочити. Це освіжило б ваш розум і зберегло ваше здоров'я. Розважальні програми.

Програми для читання та навчання. Через напружений графік у вас може не бути часу на щоденне читання газет. Але якщо у вас на телефонах встановлена програма для читання новин, вона буде в курсі останніх новин. Багато людей люблять читати книги у вільний час, але придбати їх може бути не по кишені, а перевезти їх на великі відстані складно. Але тепер ви можете купувати книги в Інтернеті за допомогою мобільних додатків і переносити їх на великі відстані. Навчальні програми пропонують своїм клієнтам онлайн-класи з певних тем та проводять тести залежно від теми.

Додатки соціальних медіа та додатки для відеодзвінків. Додатки соціальних медіа є важливим джерелом для побудови міцних стосунків між людьми, і їх поява дало нове обличчя спілкуванню. Програми соціальних медіа допомагають створювати та підтримувати міцні зв'язки між людьми. Ці програми допомагають спілкуватися в чаті, обмінюватися відео та фотографіями. Люди воліють використовувати ці програми в мобільних телефонах, ніж на настільних комп'ютерах. Соціальні програми, встановлені на мобільних телефонах, матимуть більше доступу, ніж на настільних комп'ютерах, і, отже, спілкування та стосунки будуть міцнішими без будь-яких прогалин. У наш час також поширені програми для відеодзвінків, коли ви



можете побачити своїх дорогих, коли завгодно, без будь-яких бар'єрів на відстані. Програми для соціальних медіа Сьогодні мобільні додатки - це хороша платформа для бізнесу, банківської справи, освіти, читання тощо ... Клієнти можуть отримати свої товари та послуги за порогом за розумними цінами без бар'єрів відстані та часу. Розробка та розробка нових мобільних додатків для конкретних завдань ще більше зробить послуги більш зручними для користувачів. Компанії, що розробляють мобільні додатки, пропонують новітні технічні та зручні мобільні програми.

Зростаюча популярність смартфонів обумовлена наявністю великою кількістю переваг над стаціонарними персональними комп'ютерами. Вони дозволяють виконувати значну частку задач користувача, таких як функції календаря, калькулятора, годинника, будильника та мультимедійного пристрою, при цьому залишаючись портативними.

Додатки надають миттєвий доступ до інформації в один «клік». Дозволяють споживати контент швидко, завдяки збереженню віртуальних даних, які також можуть бути доступними «офлайн». Частина ж додатків, такі, як банкінг, роздрібна торгівля, фінанси, ігри та новини, працюють в обох режимах: «онлайн» та «офлайн». Таким чином завантаження статті чи мобільної гри може проводитися за відсутності інтернет-зв'язку. Можливість роботи в двох режимах – фундаментальна відмінність між додатком та веб-сайтом.

Великою перевагою є використання сенсорів мобільного пристрою, як от камери, QR-код сканерів чи BAR-код сканерів, NFC для здійснення оплати, GPS для роботи з картами, акселерометр, гірометр і компас, чому, в свою чергу, можна знайти безліч застосувань.

Пуш-сповіщення, що забезпечують миттєве донесення інформації до клієнтів – серйозна причина, чому бізнес повинен розглянути розробку додатку. Сповіщення бувають двох типів: ті, що можуть бути отриманими лише, коли додаток відкрито і ті, що отримуються девайсом безумовно. Це допомагає просувати продукти та сервіси, регулярно збільшуючи продажі та дохід. Крім

того, у користувача завжди є можливість поділитися фото з друзями, щоб розповісти про свої покупки, що теж здійснює позитивний вплив на загальну прибутковість.

Мобільні програми сформували наш спосіб життя сьогодні. Будь-який додаток для всього - від банківських операцій до замовлення продуктів через Інтернет. Через швидкість, з якою ми можемо щось зробити зараз, світ значно змінився завдяки мобільним додаткам. Чи змінилося це на краще чи гірше, можна дискутувати. Розглянемо 4 способи, за допомогою яких мобільні програми змінили спосіб нашого повсякденного життя.

### 1. Негайна потреба в інформації.

Через мобільні програми, які дозволяють шукати по суті будь-що, будь-коли та в будь-якому місці, ми всі відчуваємо невідкладність у пошуку інформації. Ще 20 років тому людям, які сиділи за обіднім столом, раптом потрібно було знати ім'я актриси, яка грає на ерані, просто тут і зараз.. Але сьогодні такий сценарій є реальністю, і ми щиро відчуваємо необхідність щось пошукати на місці, щоб отримати відповіді на наші, часто безглузді, запити. Через це, відчуття, коли інформація недоступна відразу, розчаровує.

### 2. Миттєвий зв'язок та зв'язок.

За допомогою мобільних додатків для обміну повідомленнями, соціальних мереж та відео-чатів спілкуватися зараз простіше, ніж будь-коли раніше. Можна залишатися на зв'язку з коханими, які живуть у різних країнах, стежити за старими однокласниками в соціальних мережах та вести відеочат із собакою, поки ви перебуваєте у відпустці.

Постійне спілкування у нас під рукою відкрило нам двері, які раніше ніколи не були можливими, але залишатися на зв'язку завжди є свої підводні камені. Наприклад, коли ми комусь надсилаємо SMS і не отримуємо відповіді відразу, це викликає у нас нетерпіння і роздратування. До створення мобільних додатків для спілкування не було незвичним проходити дні, навіть тижні, не чуючи друзів чи коханої людини. Зараз це сприймається як незручність, якщо хтось не надішле вам повідомлення протягом декількох годин.

### 3. Підвищення продуктивності праці.

Мобільні програми дозволяють постійно підключатися до роботи. Це чудово для продуктивності, але, можливо, не настільки для соціального життя. Через можливість постійно бути «підключеними», багато людей працюють довше, ніж будь-коли раніше. Однак якщо ви багато подорожуєте у справах, можливість отримати доступ до роботи зі свого телефону - це знахідка. Тільки вся справа в балансі.

### 4. Спрощення щоденних завдань.

Щоденні завдання, про які люди зазвичай не думають, стають дедалі простішими та зручнішими завдяки певним мобільним додаткам. Такі речі, як придбання квитків у кіно, отримання продуктів та здача чеків стали зручнішими завдяки мобільним додаткам. Ці програми усунули необхідність фізично відвідувати магазин або банк, замість того, щоб дозволяти виконувати завдання, перебуваючи в дорозі, лише за допомогою телефону та кінчиків пальців. Це величезна економія часу, тому полегшує життя в цілому. Ми проводимо стільки часу на роботі, виконуємо доручення та виконуємо домашні справи, що будь-яка дрібниця, яка може допомогти нам заощадити час, дуже сильно впливає на краще.

Світ, безумовно, змінився завдяки мобільним додаткам, і я можу лише уявити, як додатки будуть і надалі змінювати наше життя та формувати майбутнє.

У світі мобільних додатків не всі програми створені однаково. Нативні програми, веб-програми та гібридні програми багато в чому відрізняються.

Перш ніж порівнювати відмінності, давайте спочатку розглянемо, як відрізняються кожен із цих типів програм, а потім переваги та недоліки для кожного.

Нативні програми - це програми для смартфонів, спеціально розроблені для певної операційної системи - iOS або Android. Це найбільше приходить на думку, коли ми думаємо про мобільні програми. Вони завантажуються з App Store або Google Play і встановлюються на пристрій.

Що відрізняє нативні програми від мобільних веб- та гібридних програм, це те, що вони розроблені для певних пристроїв. Наприклад, програми для Android написані на Java, а програми для iPhone - на Objective-C або Swift. Перевага вибору нативного додатка полягає в тому, що він найшвидший і найнадійніший, коли справа стосується користувацького досвіду. Такі програми створюються з використанням SDK операційної системи, а також можуть взаємодіяти з усіма функціями пристрою, такими як мікрофон, камера, GPS, пристрій для зберігання пристроїв тощо. Однак недолік пов'язаний із витратами на розробку та обслуговування. Потрібен більший бюджет, якщо ви хочете створити свій додаток для кількох платформ (наприклад, iPhone та Android), який також стосується будь-яких поточних оновлень, необхідних для оновлення вашої рідної програми.

Веб-програми - це веб-сайти, оптимізовані для мобільних браузерів. Вони розроблені виключно для доступу через веб-браузер. Вони можуть працювати в декількох браузерах, таких як Chrome або Safari, і написані на JavaScript та HTML5.

Ці програми є, по суті, веб-сайтами з інтерактивними функціями, завдяки яким вони відчують себе схожими на мобільні програми. Однак перевага полягає в тому, що ці програми часто дешевші у розробці.

Якщо бюджет невеликий і не потрібні складні функціональні можливості або доступ до функцій операційної системи, то створення веб-програми може бути найменш дорогим варіантом. Недоліком є те, що веб-програми можуть бути повільнішими, менш інтуїтивними та недоступними через магазини додатків.

Гібридні програми поєднують функції як нативних, так і веб-програм. До них можна отримати доступ через веб-браузер та завантажити їх із магазинів додатків. Вони написані на HTML5 та JavaScript, як веб-програми. Здебільшого це веб-сторінки, загорнуті в мобільний додаток за допомогою WebView. Однак вони також мають доступ до вбудованих можливостей пристрою. Вони

побудовані з використанням платформних фреймворків, таких як React, Ionic, Sencha та Xamarin.

Перевага гібридних додатків полягає в тому, що їх, як правило, легше і швидше розробляти, ніж нативні програми. Вони також вимагають менше обслуговування. Однак швидкість гібридного додатка буде повністю залежати від швидкості браузера користувача. Зрештою це означає, що гібридні програми майже ніколи не працюватимуть так швидко, як рідна програма.

Давно минули часи, коли вдома у людей була величезна колекція фільмів. Більшість країн світу перейшли на цифрову колекцію, але навіть це вимагає величезного простору. Зрештою, фільм може зайняти до 10 ГБ місця для зберігання. Для вирішення цієї проблеми вже існує чимало рішень на ринку. Одне з найпопулярніших – «Sony Crackle». Воно надає можливість для потокового перегляду фільмів. У ньому є не тільки величезна колекція, але й безкоштовний додаток для фільмів майже для будь-якої платформи та пристрою.

У всіх фільмах, що виходять на Crackle, є підписки для людей із вадами слуху, і для перегляду безкоштовних фільмів на платформі зазвичай не потрібна авторизація. Недоліком є те, що фільми підтримуються завдяки рекламі, а отже глядач змушений буде її переглядати. Інтерфейс системи «Crackle» наведено на рисунку 1.1.

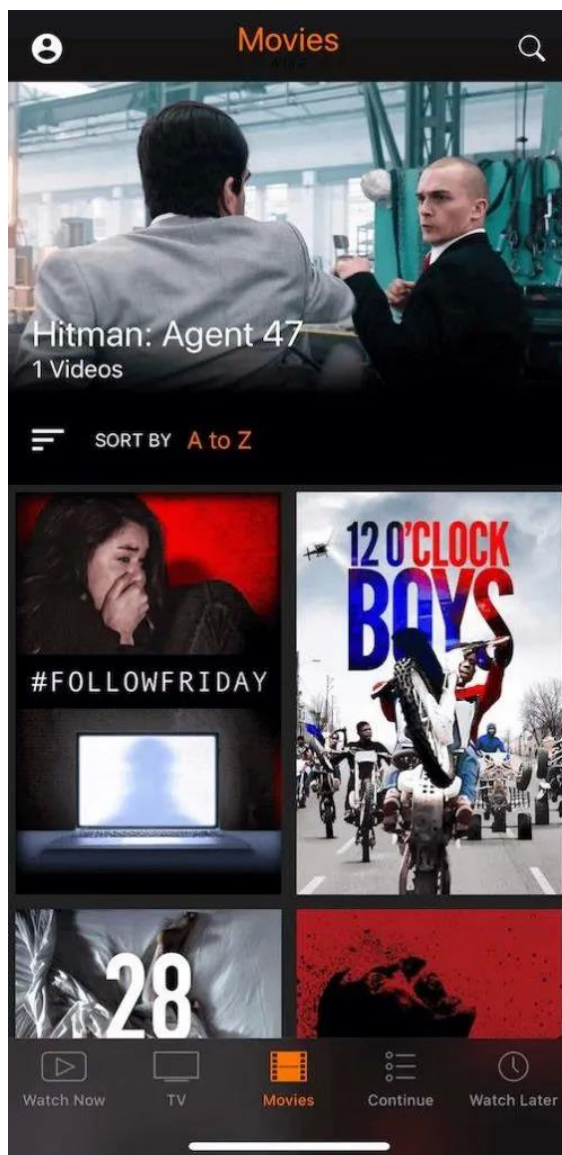


Рисунок 1.1 – Інтерфейс сервісу «Crackle»

Ще одним вагомим гравцем на ринку є система «Popcornflix» - популярний варіант перегляду фільмів безкоштовно. Додаток розділяє фільми, телешоу та вірусні відео на три різні розділи для зручного перегляду. Далі розділ фільму поділяється на такі категорії, як найпопулярніші, підбір персоналу, драма, бойовик тощо. Хоч Popcornflix транслює рекламу, більшість оголошень є короткими, тому вони не погіршують загального враження від користування. Popcornflix доступний для iPhone, Android, Roku, Apple TV тощо. Приклад інтерфейсу «Popcornflix» наведено на рисунку 1.2.

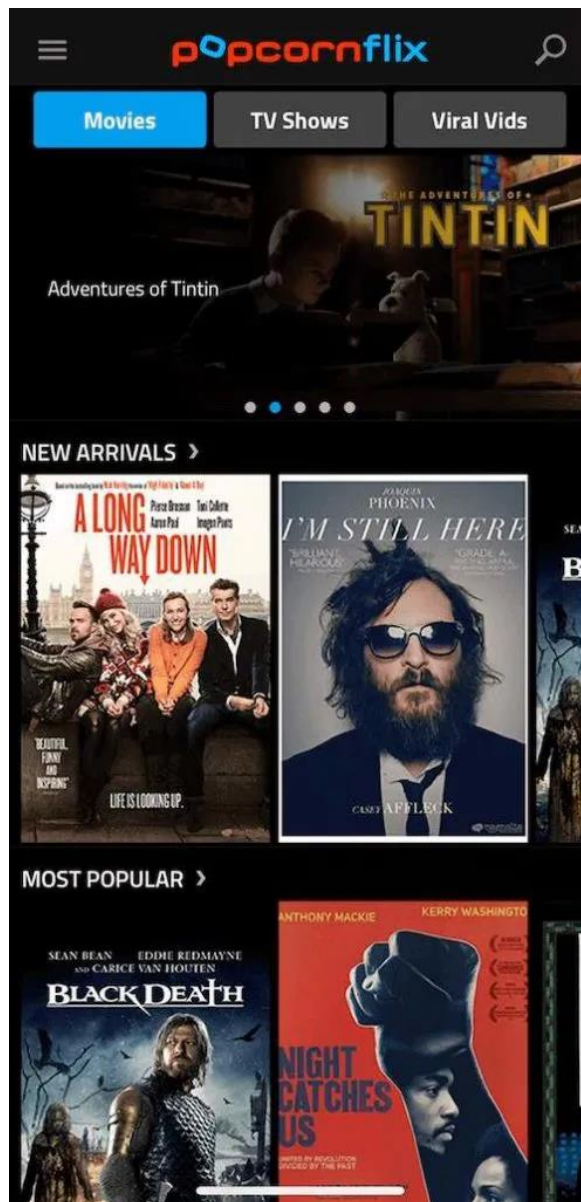


Рисунок 1.2 – Приклад інтерфейсу «Popcornflix»

Останнім з трійки лідерів є сервіс Vudu. Не зважаючи, що система більш відома завдяки прокату фільмів удома, є також сотні фільмів, які ви можна переглянути повністю безкоштовно.

У сервісу є кілька мінусів, але вони врівноважуються. Основна проблема полягає в тому, що для перегляду будь-яких фільмів потрібно увійти в систему. З іншого боку, кількість якісних фільмів робить цей сервіс вартим для того, аби спробувати. Vudu також регулярно додає нові фільми. Інтерфейс мобільного додатку наведено на рисунку 1.3.

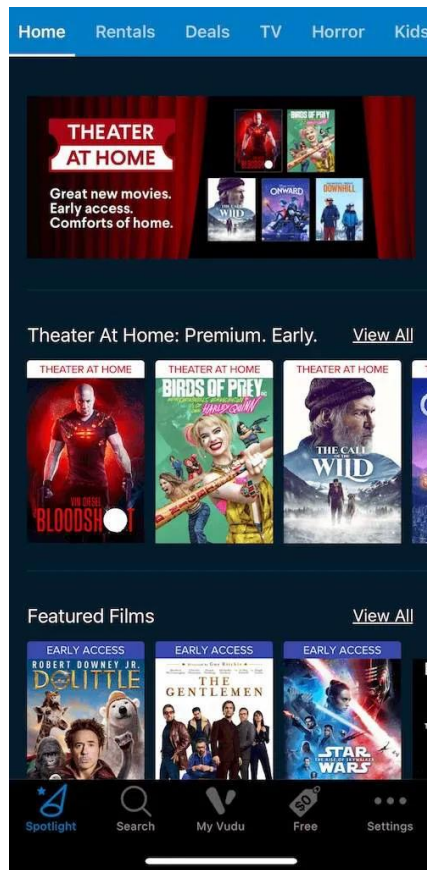


Рисунок 1.3 - Інтерфейс додатку системи «Vudu»

Таким чином на ринку доволі багато рішень, які задовольняються потребу перегляду, та конкурують між собою, але не так багато якісних рішень для доступу до повної бази даних усіх новинок кіно.

Найпродуктивнішим та найшвидшим каналом передачі інформації у світі є глобальна мережа Інтернет, що дає змогу ефективно використовувати її для впровадження інновацій у розробці мобільних додатків.

Метою роботи є аналіз форм застосування мобільних технологій та сфери розробки мобільних додатків та серверів, а також дослідження досвіду їх запровадження у вітчизняній та закордонній практиці.



## 2 ТЕОРЕТИЧНА ЧАСТИНА

### 2.1 Мета дослідження

Мета даного дослідження – розробка системи відстежування кінокартин, що дозволить знайти шоу та фільми для перегляду, відстежувати переглянуті епізоди та фільми, отримувати сповіщення про майбутні епізоди та створювати резервні копії та синхронізувати їх на різних пристроях.

У системі можна буде знайти шоу з новими епізодами, які популярні або схожі на ті, що ви переглядаються. Буде присутнім функціонал дослідження фільмів, які зараз перебувають у кінотеатрах, випускаються цифровим шляхом або на диску. Також резервне копіювання та синхронізація.

Додаток повинен пропонувати користувачу наступні функції:

- відображення великої кількості даних, отриманих від запиту на сервер;
- наявність функціонального блоку «Серіали», «Списки», «Фільми» та «Статистика»;
- можливість перегляду інформації про найближчі епізоди та фільми;
- можливість перегляду списку нещодавно переглянутих картин;
- наявність фільтру та функції сторткування;
- пошук за допомогою запиту на сервер;
- керування користувацькими списками, збереження локальних даних на пристрої;
- перегляд цифрових видань, видань на диску та популярних фільмів;
- управління власною колекцією;
- ведення статистики з можливістю поширити.

Користувач може зберігати отримані результати, додавати до них власні помітки, обмінюватись результатами з друзями через популярні соціальні мережі.

## 2.2 Інструменти розробки

Мобільні додатки, як такі, широко розповсюдженні ви використовуються для найрізноманітніших потреб. Розробка додатків це недешева справа, і крім того, потребує багато часу. Зараз на ринку лідирують дві операційні системи: Android та iOS. Порівняльна характеристика інструментів розробки додатків для обох платформ наведена в таблиці 2.1.

Таблиця 2.1 – порівняльна характеристика інструментів розробки мобільних додатків для різних платформ

	<b>Apple iOS</b>	<b>Android</b>
Мовб програмування	Objective-C, C, C++, Swift	Java, Kotlin, інші
Засоби розробки	Xcode	Android Studio, Android SDK
Формат	.app	.apk
Магазин	Apple App Store	Google Play Store

Така ситуація змушує різні використовувати різні команди розробників для розробки додатку під кожну платформу, що, звичайно збільшує загальну дороговизну Вартість базована вимогах до конкретного розроблюваного продукту, використанні складних функцій системи.

Якщо клієнти використовують пристрої iOS та Android, то є три варіанти:

- вибрати одну платформу для свого першого додатка, а пізніше додати підтримку для іншої,
- створити дві програми відразу,
- створити крос-платформний або гібридний додаток.

Якщо обрано одну платформу для початку, різниця у вартості здебільшого залежатиме від того, скільки часу триває розробка, оскільки ставки

на розробку iOS та Android більш-менш схожі. У Південній Америці та Індії розробка iOS коштує дорожче, хоча різниця не дуже велика.

Створення мобільного додатка для Android, як правило, займає більше часу, однією з причин є те, що вашій команді доведеться тестувати на більш широкому діапазоні пристроїв.

Тип програми робить найбільшу різницю в оцінці вартості. Перш за все, що це за типи програм?

Нативні мобільні програми створені спеціально для однієї мобільної платформи. Для звичайного користувача неможливо запустити файл APK (виконуваний для Android) на iPhone або IPA (виконуваний для iOS) на пристрої Android.

Android та iOS використовують різні мови програмування та фреймворки: Java та Kotlin для Android, Swift та Objective-C для iOS. Коли ваша цільова аудиторія використовує пристрої Android та iOS, і ви вирішили підтримувати обидві платформи за допомогою власних програм, це означає, що вам потрібно буде створити дві окремі програми: одну для Android та іншу для iOS.

Однак існують варіанти створення однієї програми для обох платформ (крім веб-програми або адаптивного веб-сайту):

Кросплатформна програма може бути написана на C # із Xamarin або JavaScript з React Native, а в гібридних додатках використовується AngularJS та його Ionic фреймворк. Будь-який із цих варіантів буде працювати як на пристроях Android, так і на iOS.

Звідси виникає питання: чому я буду робити програми для кожної платформи, коли я можу зробити один додаток для обох?

Справді, швидше та дешевше створити гібридний додаток, ніж власний, не кажучи вже про два. Однак вснують і недоліки міжплатформенних та гібридних додатків. Порівняння нативних та кросплатформних рішень наведено в таблиці 2.2:

Таблиця 2.2 – Порівняння нативних та кросплатформних рішень

	Нативні	Крос-платформні	Гібридні
«+»	Найкраща швидкодія, Проста інтеграція з системою, Доступ до функцій платформи, Безпека, Хороший UX	Середня ціна, Один додаток для двох платформ, Легкість оновлення	Дешевизна, Легкість маніпуляцій, Один додаток на дві платформи
«-»	Дорога вартість розробки, Великий час щоб розробити	Лімітована гнучкість, Відсутній доступ до функцій платформи, Погана швидкодія, Проблеми з безпекою	Низька швидкодія, Поганий UX, Промлеми з безпекою, Лімітований доступ до апаратної частини

Незважаючи на всі недоліки, все ще існує ряд популярних кросплатформних та гібридних додатків, включаючи Skype, Slack та Instagram. Фокус полягає у вирішенні, який тип програми потрібен вашому бізнесу. Можливо, мінуси не будуть мати значення для конкретного додатка. Але, можливо, вони це зроблять.

Чудові візуальні ефекти є наріжним каменем хорошого додатка, тому навряд чи є несподіванкою, що дизайн може вплинути на вартість створення додатка. Спеціальні значки та елементи дизайну - екрани, логотипи, кнопки - займуть багато часу та коштуватимуть багато. І кожен екран програми слід намалювати окремо. Створення власних анімацій - це зовсім інша проблема, але вони часто виграшні.

Втім, чудовий дизайн не завжди означає складні візуальні ефекти. Іноді простота є найкращою відповіддю: залежно від того, чим займається ваш додаток, можливо, можна буде використовувати стандартні елементи, що надаються ОС, для складання яких не потрібно багато часу та зусиль.

Звичайно, дизайн є особливо важливою частиною ціни на розробку мобільних додатків, коли мова йде про ігри.

Вартість розробки програми в основному залежить від кількості функцій, але це ще не все. Складність цих особливостей іноді є більш впливовою. Деякі функції використовують лише стандартні інструменти та API; інші вимагають інтеграції сторонніх API; для третього типу потрібні побудовані на замовлення алгоритми.

Структура частки на ринку найпопулярніших ОС наведена на рисунку 2.1.

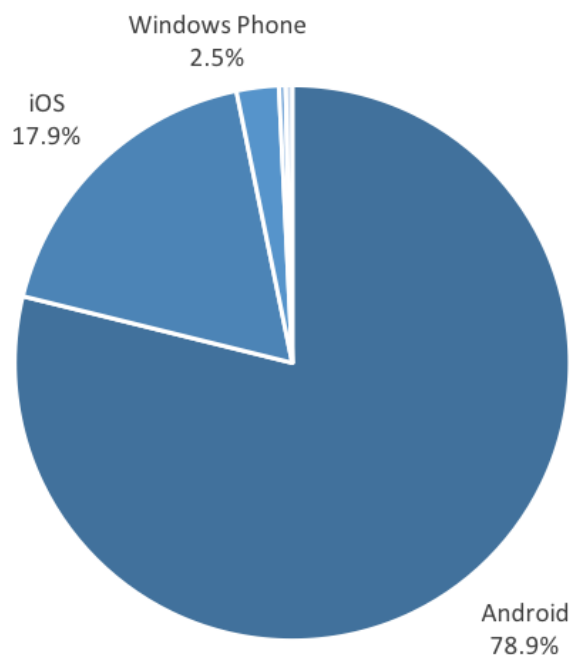


Рисунок 2.1. Структура частки на ринку найпопулярніших ОС

Важливим аспектом розробки кросплатформних мобільних додатків є вибір платформи. Статистика пошукових запитів найбільш популярних кросплатформних рішень наведена на рисунку 2.2.

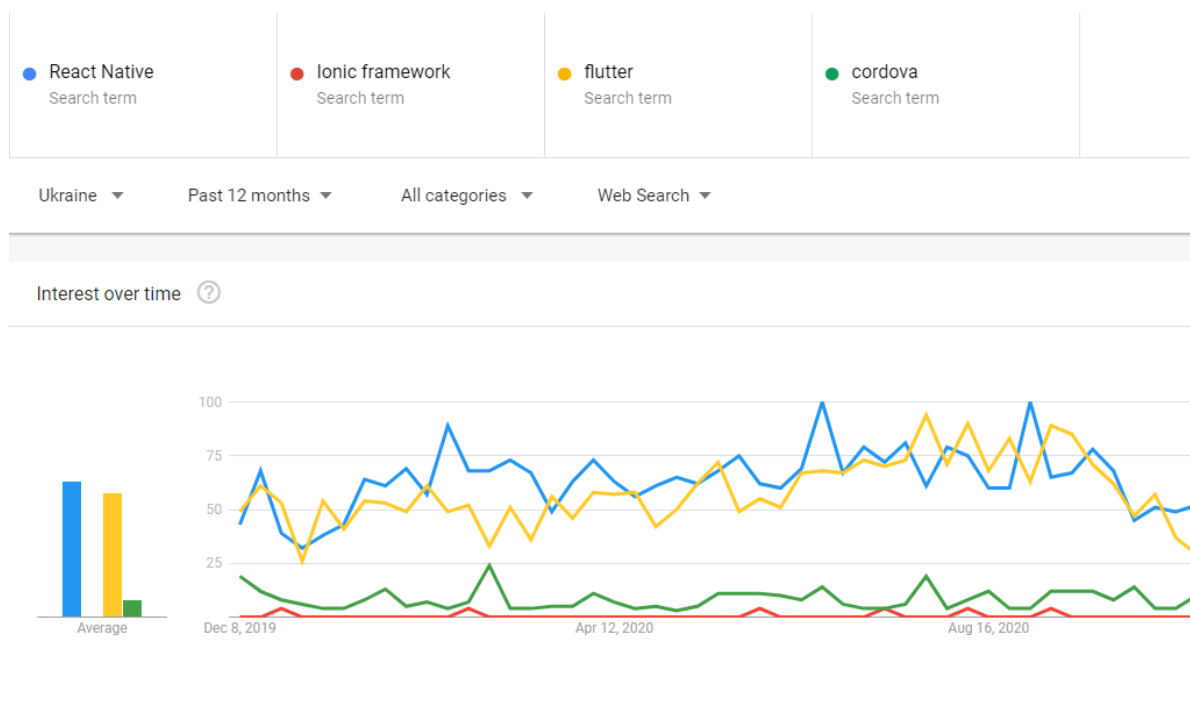


Рисунок 2.2 - Статистика пошукових запитів найбільш популярних кросплатформних рішень у 2020 році

Зрозуміло, що розробка додатків сама по собі найбільше впливає на вартість програми. Досвідчені розробники, які успішно реалізували безліч проектів, все ще є дефіцитним ресурсом. Ось чому на кодування програми зазвичай припадає 50–70 відсотків загальних витрат на розробку програми.

Від того, чи створюється програма з нуля, чи використовує наявну кодову базу, залежить вартість.

Існує багато причин, за якими над існуючими проектами легше чи складніше працювати, ніж робити «з нуля», але врешті-решт, все зводиться до того, що існуючі проекти, природно, мають вплив своїх початкових творців.

Дана інформаційна система використовуватиметься для відстежування кінокартин, що дозволить знайти шоу та фільми для перегляду, відстежувати

переглянути епізоди та фільми, отримувати сповіщення про майбутні епізоди та створювати резервні копії та синхронізувати їх на різних пристроях.

Чесно кажучи, іноді виявляється, що код просто погано написаний. Зараз кодування - це вид мистецтва, і іноді те, що виглядає як поганий код, просто не те, до чого звикли розробники.

Незалежно від того, чи є програма для планування зустрічей з пацієнтами чи, відстеження симптомів захворювання, чи просто показу вашого блогу у всьому світі, йому, ймовірно, знадобиться база даних для зберігання інформації про товар, платіжний процесор для обробки платежів або CMS для управління вмістом. Іншими словами, якщо додаток не є настільки простим, наскільки це може бути, то варто подумати про калькулятори або основні додаткові програми.

Чим складніші функції програми, тим дорожче буде її розробка.

Складний додаток часто передбачає такі речі, як інтеграція сторонніх розробників, логіка на стороні сервера, адміністративні панелі або використання мобільного обладнання, такого як Bluetooth або GPS. Кожен із них має власні витрати як на час розробника, так і на потенційні витрати на підписку у випадку сторонніх послуг.

Існує маса технологій, які входять до більш використовуваних наборів функцій: бази даних, бібліотеки машинного навчання, інтерфейсні фреймворки, і т.п. Кожен шар нової технології додає проекту складності, часу та необхідних знань, а отже, і більших витрат.

Таким чином галузь розробки мобільних додатків є доволі складною і багатогранною. Її розуміння на хорошому рівні потрібне задля досягнення успіху в розробці власного додатку.

У розроблюваній системі можна буде знайти шоу з новими епізодами, які популярні або схожі на ті, що ви переглядаються. Буде присутнім функціонал дослідження фільмів, які зараз перебувають у кінотеатрах, випускаються цифровим шляхом або на диску. Також резервне копіювання та синхронізація.

## 2.3 Засоби для розробки додатку

Для розробки додатку слід опанувати такі навички та засоби:

Мови програмування Java, Kotlin та Groovy, мова XML, система збірки Android, а також середовище для розробки Android Studio. Засоби розробки нативного додатку зображено на рисунку 2.3



Рисунок 2.3 – Засоби розробки нативного додатку

У Android розробці використовуються три мови програмування та одна мова розмітки.

Java є офіційною мовою для розробки Android і однією з найпопулярніших мов програмування у світі. Класи Java працюють на Android Runtime (ART), спеціалізованій віртуальній машині. Приклад із файлу Android MainActivity.java. зображено на рисунку 2.4.

```
1 public class MainActivity extends AppCompatActivity {
2
3     @Override
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_main);
7
8         TextView mainTextView = findViewById(R.id.mainTextView);
9         mainTextView.setText("Hello educative.io");
10    }
11 }
```

Рисунок 2.4 - Приклад головного файлу Android MainActivity.java

Kotlin є другою офіційною мовою для розробки Android з 2017 року. Котлін, відомий тим, що є набагато лаконічнішим та виразнішим, допомагає



усунути деякі недоліки Java. Приклад того ж самого коду наведеного вище, але написаного на Kotlin зображено на рисунку 2.5.

```
1 public class MainActivity : AppCompatActivity {
2
3     override fun onCreate(savedInstanceState: Bundle) {
4         super.onCreate(savedInstanceState)
5         setContentView(R.layout.activity_main)
6
7         val mainTextView = findViewById<TextView>(R.id.mainTextView)
8         mainTextView.text = "Hello educative.io"
9     }
10 }
```

Рисунок 2.5 - Приклад головного файлу Android MainActivity.kt

XML, мова розмітки, зазвичай використовується при розробці Android для оголошення макета інтерфейсу користувача (UI), розмірів та рядків. Приклад макета, який відображає текст посередині екрана користувача наведено на рисунку 2.6.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <TextView
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Hello World!"
12        app:layout_constraintBottom_toBottomOf="parent"
13        app:layout_constraintLeft_toLeftOf="parent"
14        app:layout_constraintRight_toRightOf="parent"
15        app:layout_constraintTop_toTopOf="parent" />
16
17 </androidx.constraintlayout.widget.ConstraintLayout>
```

Рисунок 2.6 - Приклад макета, що відображає текст посередині екрана користувача

Система збірки Android - Gradle використовує систему автоматизації збірки Android і розширює концепції Apache Maven та Apache Ant, представляючи мову, що базується на Groovy, специфічну для домену. Groovy - це додатково введена динамічна мова з можливостями статичного набору тексту. Це допомагає підвищити продуктивність і швидкість завдяки інтеграції з програмою на основі Java. За допомогою простоти Groovy та зрілої екосистеми Gradle можна автоматизувати своє програмне забезпечення.

IDE Android Studio – найпотужніша система для розробки Android. Базується на основі IntelliJ IDEA це офіційне середовище розробки операційної системи Google. Він поставляється з чудовими інструментами для Android, щоб задовольнити всі потреби. Використання цієї IDE пришвидшить ваш час розробки. Середовище оснащено такими функціями: профайлер та аналізатор у режимі реального часу, гнучка система збірки проекту, інтелектуальний редактор коду, редактор візуального макету, швидкий емулятор.

Android SDK - це офіційний набір для розробки додатків для Android. Він складається з модульних пакетів, які можна завантажити окремо з Android SDK Manager, включаючи інструмент SDK, Google API, підтримку Android, Bridge Debug Bridge (ADB) та багато іншого. Як і IDE, Android SDK постійно оновлюється.

Java є офіційною мовою для розробки програм для Android і підтримує всі інструменти Android. Знання цієї мови значно полегшить досвід розробників.

Java була обрана для розробки Android, оскільки вона добре відома, добре підтримується засобами розробки і вже широко поширена у галузі мобільних телефонів. Крім того, Java працює у віртуальній машині, тому її не потрібно перекомпілювати.

Для того, аби розпочати роботу слід встановити Android Studio. Вигляд основного вікна Android Studio зображено на рисунку 2.7.

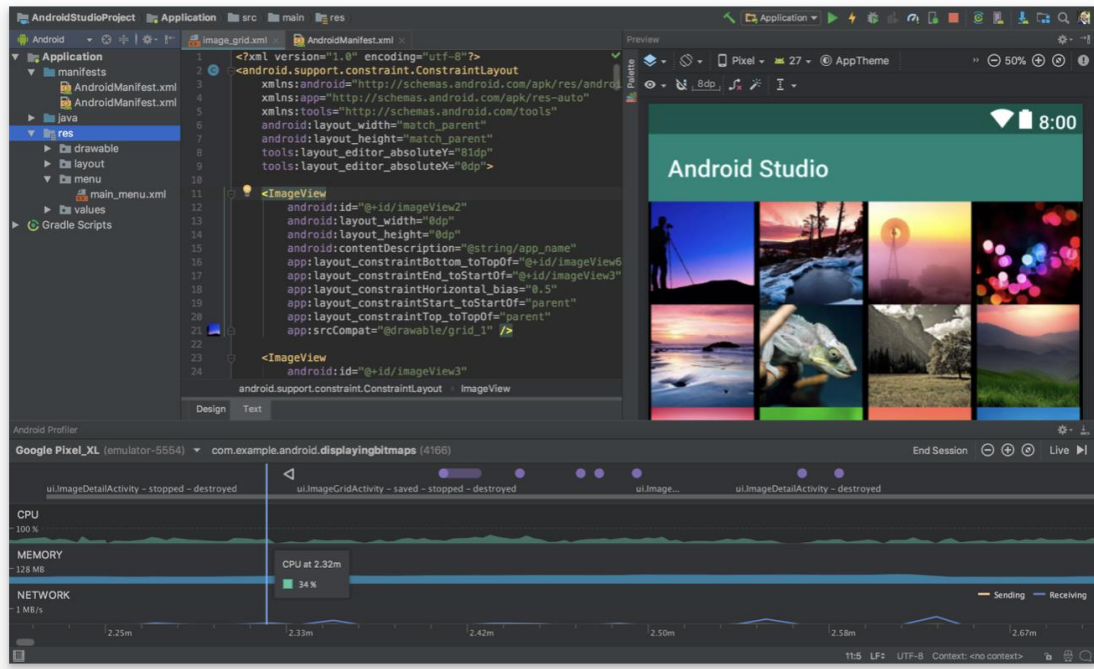


Рисунок 2.7. Вигляд основного вікна Android Studio

Він спеціально розроблений для високоякісних програм для Android. Цей IDE пришвидшить час розробки та зробить програми набагато надійнішими та простішими в оновленні, коли з'являться нові функції.

Щоб встановити Android Studio на Windows, слід виконати такі дії.

Перейти на офіційний сайт, щоб отримати останню версію Android Studio. Після цього завантажити IDE як файл .exe або .zip. Для файлу .exe двічі клацнути файл, щоб запустити його. Для файлу .zip розпакувати ZIP і скопіювати папку android-studio у файли програми. Це запропонує відкрити та запустити android-studio> binfolder.

Після запиту дотримуйтесь майстра налаштування Android Studio, де слід вибрати свої пакети SDK.

Щоб встановити Android Studio на Mac, слід виконати такі дії.

Перейти за посиланням, щоб отримати останню версію Android Studio. Після завантаження, запустити файл DMG і перетягнути його у папку Програми.

Запустити Android Studio. Звідси можна або розпочати новий проект, або імпортувати попередні налаштування.

## 2.4 Опис системи контролю версій

Система контролю версій - це система, що записує зміни у файл або набір файлів протягом деякого часу, так що присутня можливість повернутися до однієї з версій згодом.

Це безкоштовна та розподілена система контролю версій з відкритим кодом. Мета Git - відстежувати проекти та файли, оскільки вони змінюються з часом за допомогою маніпуляцій різних користувачів. Git зберігає інформацію про хід проекту у сховищі. У сховищі є коміти до проекту або набір посилань на коміти, які називаються head. Вся ця інформація зберігається в тій самій папці, що і проект, у підпапці під назвою .git, і в основному вона буде прихована за замовчуванням у більшості систем.

Git відстежує зміни, які декілька людей робить в одному проекті, а потім об'єднує код, де люди працювали над різними частинами, в один проект. Таким чином, коли хтось вводить помилку, можна відстежити код, який дав помилку, переглядаючи коміти. Потрібно зробити коміти в проекті, щоб повідомити git, що Ви задоволені внесеними змінами та хочете внести зміни в основну гілку, яка за замовчуванням називається master.

Потім є можливість завантажити код на GitHub або BitBucket, де авторизовані користувачі можуть переглядати, тягнути код або натискати зміни.

## 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Функціональні модулі розробленої системи

Основними функціональними завданнями системи є забезпечення доступу працівників до бази даних усіх шоу, фільмів, серіалів. Усіх відомостей про дату виходу, жанр, тривалість, середню оцінку від глядачів, кількості та назв сезонів та епізодів конкретного продукту, різноманітної довідкової інформації. Разом з тим, додаток може застосовуватись для організації систематизації даних, для запису та збереження прогресу перегляду серіалу (можливість стежити за власним прогресом). А також як інформаційна база даних в зручному форматі застосунку, яка завжди доступна під рукою.

Система реалізована у формі довідника з можливістю пошуку необхідних даних. Структура складається з 4 функціональних модулів (рис.3.1): шоу, списки, фільми та статистика. В процесі розробки додатку найбільшої уваги було приділено інтерфейсу користувача, що забезпечує репрезентацію даних користувачеві та взаємодію з ним. Графічний дизайн виконано в інформативному, мінімалістичному стилі з великою увагою до деталей та зручності використання.

Також було приділено увагу прикладному рівню, що передбачає створення логічних зв'язків між компонентами, рівню управління даними, що дозволяє зберігати дані на внутрішньому накопичувачі мобільного пристрою та оновлювати їх отримуючи актуальну інформацію з серверів та онлайн баз даних, які й служать джерелом даних для мобільної системи.

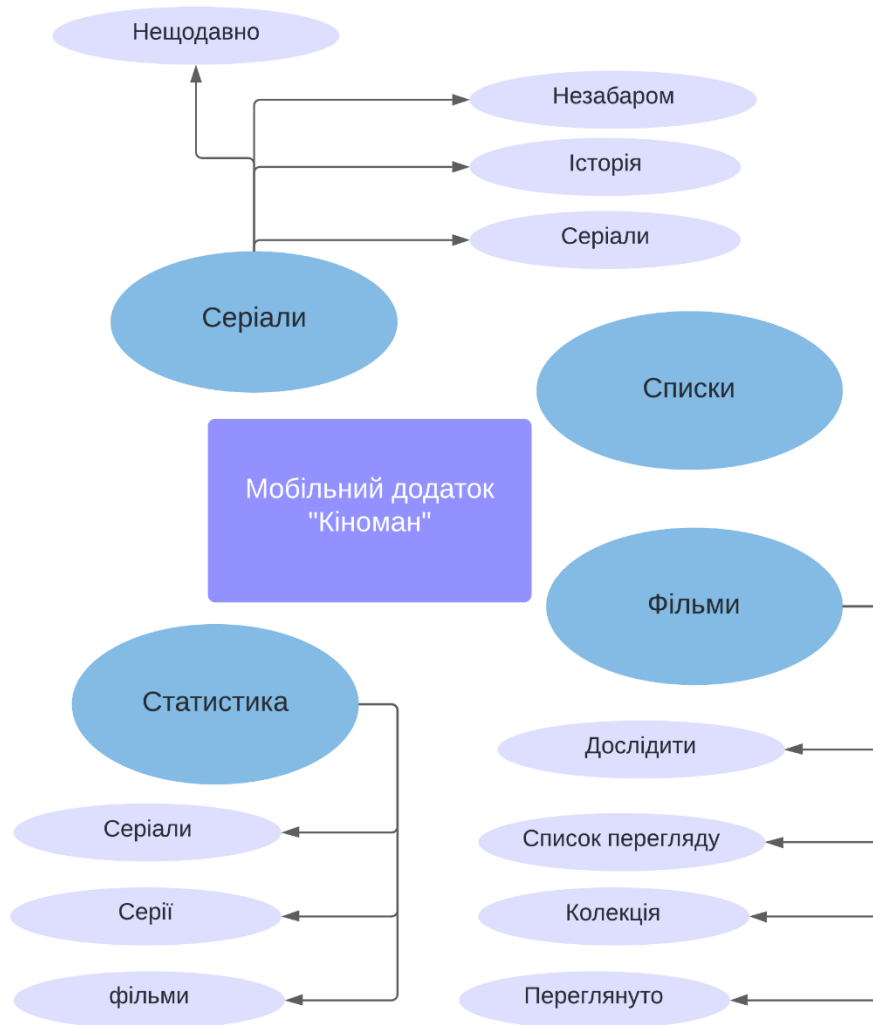


Рисунок 3.1 – Структура функціональних модулів

Шаблон мобільного додатку містить блок з заголовком активного розділу зверху сторінки та внизу панель навігації для переходів між розділами: «Серіали», «Списки», «Фільми» та «Статистика».

Меню сторінки «Серіали» представлено у вигляді списку зі збереженими серіалами. При натисканні виборі конкретного рядка відкривається нова сторінка, яка містить огляд картини. В розділі «Серіали» доступно 4 підрозділи: «Серіали», «Історія», «Незабаром» та «Нещодавно». В пункті «Історія» показані усі картини, які користувач відзначив, як нещодавно переглянуті, з вказаним часом, коли кожен потрапив до цього розділу. Наступна за чергою вкладка «Незабаром». Її призначення в тому, щоб давати інформацію про усі найближчі серії, що виходитимуть, якщо такі є. Вибірка проводиться тільки серед збережених серіалів. Остання ж сторінка надає інформацію про всі

нещодавні випуски. За замовчуванням сортування відбувається за датою виходу. Самі ж серіали на головній сторінці можна сортувати як за назвою, так і за найновішою або найстарішою серією, а також за кількістю, що залишилося переглянути. Дизайн основних розділів стартової сторінки та список варіантів для сортування наведено на рисунку 3.2.

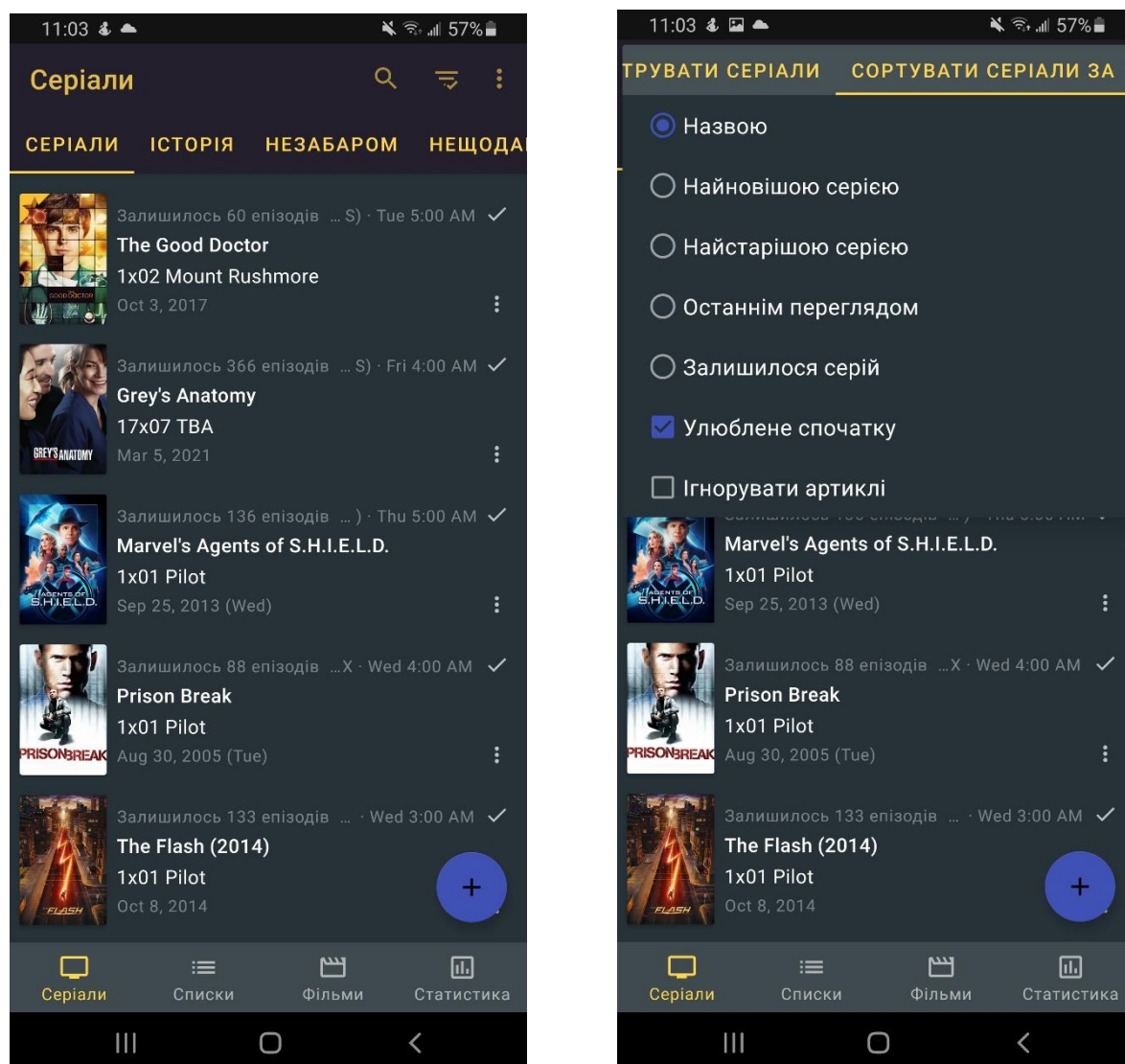


Рисунок 3.2 - Дизайн основних розділів стартової сторінки та список варіантів для сортування

Для кожного елементу присутня можливість відкрити сторінку з усіма подробицями, як от тривалість, опис, список акторів, жанр, рейтинг та дата

релізу. Присутня інформація стосовно кожної серії. Її можна позначити переглянутою, додати в свою колекцію або пропустити.

Наступною після головної розташований функціональний модуль «Списки». Його призначення в тому, щоб надати можливість додавання, серіалів, сезонів та серій до цього списку. Крім того кількість таких списків необмежена. Користувачеві надається повна свобода управління елементами. Він може давати назву кожному, видаляти та сортувати. Даний модуль служить чудовою платформою для різноманітних швидких заміток а також складання власних рейтингів, класифікації, організації. Варіантів застосування такого функціоналу безліч. Дизайн сторінки з подробицями а також модуля «Списки» наведено на рисунку 3.3.

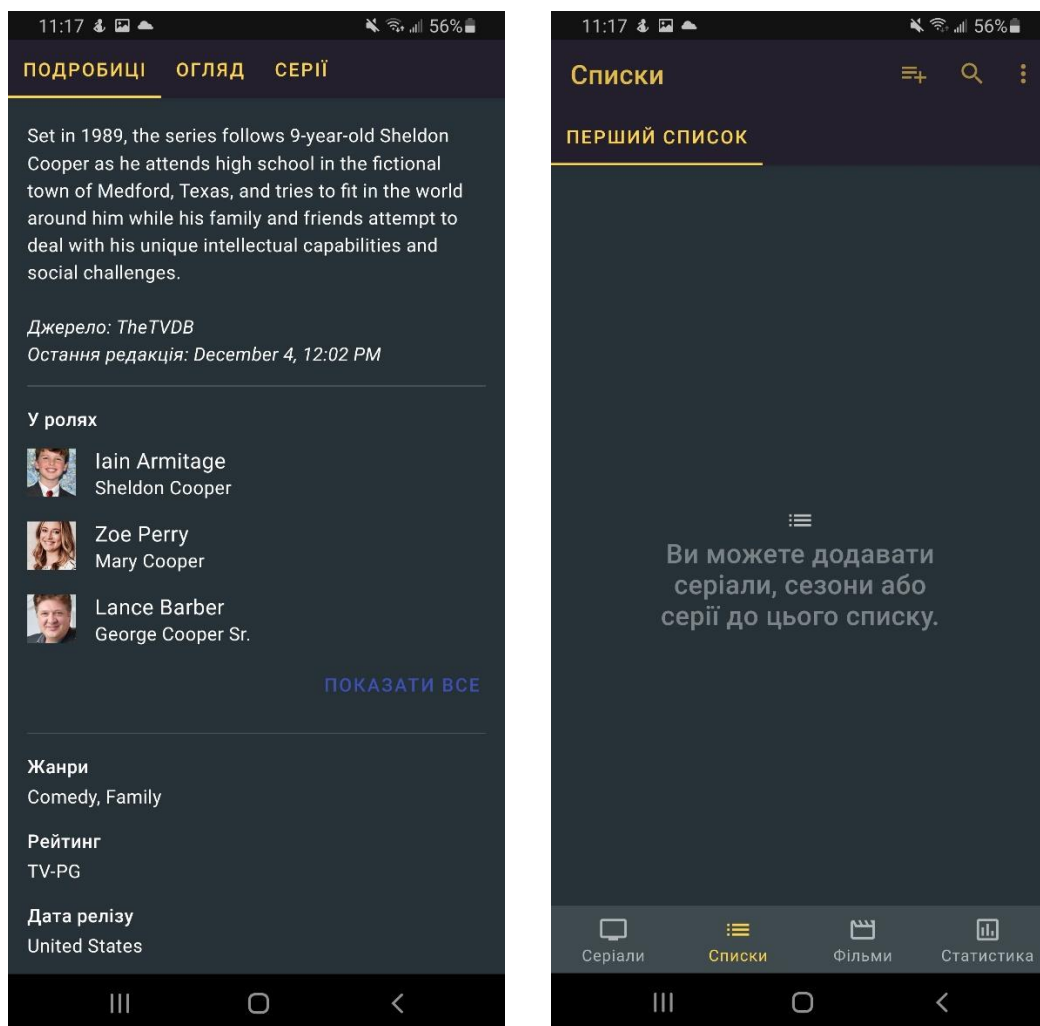


Рисунок 3.3 - Дизайн сторінки з подробицями та модуля «Списки»



Функціональний модуль «Фільми» пропонує повну колекцію найближчих кінокартин. Як цифрових, так і тих, що виходять на диску. Для полегшення доступу до необхідних даних в модулі реалізовано пошук. Всього є чотири підрозділи. «Список перегляду» – сюди потрапляють усі збережені фільми, що незабаром будуть презентовані.

Кожен з них можна позначити переглянутим, видалити зі списку перегляду чи додати в колекцію. «Колекція» служить списком улюблених картин, до яких глядачу захочеться повернутися знову і знову. В останній підрозділ «Переглянуто» потрапляє усе, що було відзначено відповідним чином. По кожному фільму також доступна детальна інформація, рейтинг, опис, дата виходу на екрани, список акторів а також знімальної групи. Дизайн розділу «Дослідити» та «Список перегляду» наведено на рисунку 3.4.

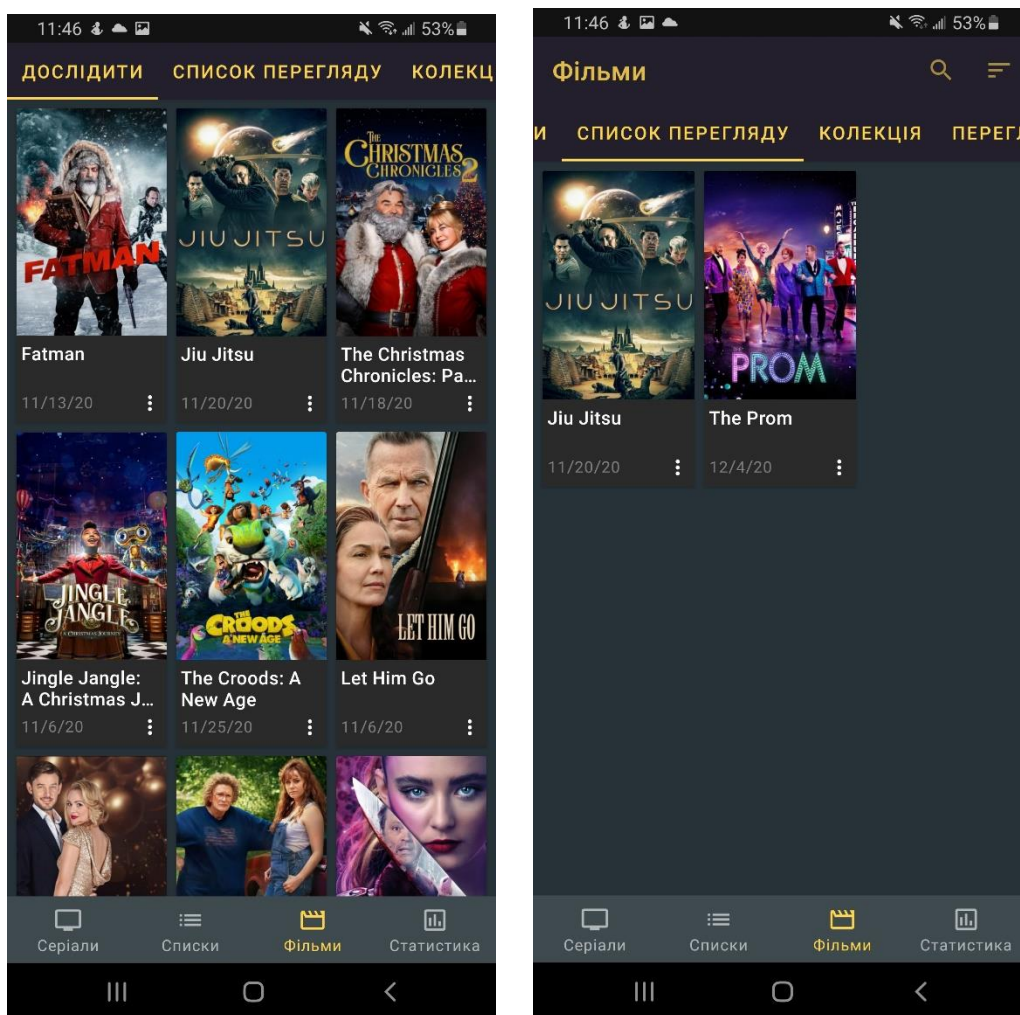


Рисунок 3.4 - Дизайн розділу «Дослідити» та «Список перегляду»

Функціональний модуль «Статистика» збирає та агрегує всі дані про кількість серіалів, фільмів та серій, що були переглянуті користувачем. Присутня інформація про загальну тривалість в годинах та хвилинах. Статистика поділена на також на переглянуті, ті, що в колекції і ті, що в списку спостереження. Також присутня функціональність для того, що поділитися звітом такої статистики в соціальних мережах або надіслати друзям. Дизайн розділу «Статистика» та меню «Поділитися» зображено на рисунку 3.5.

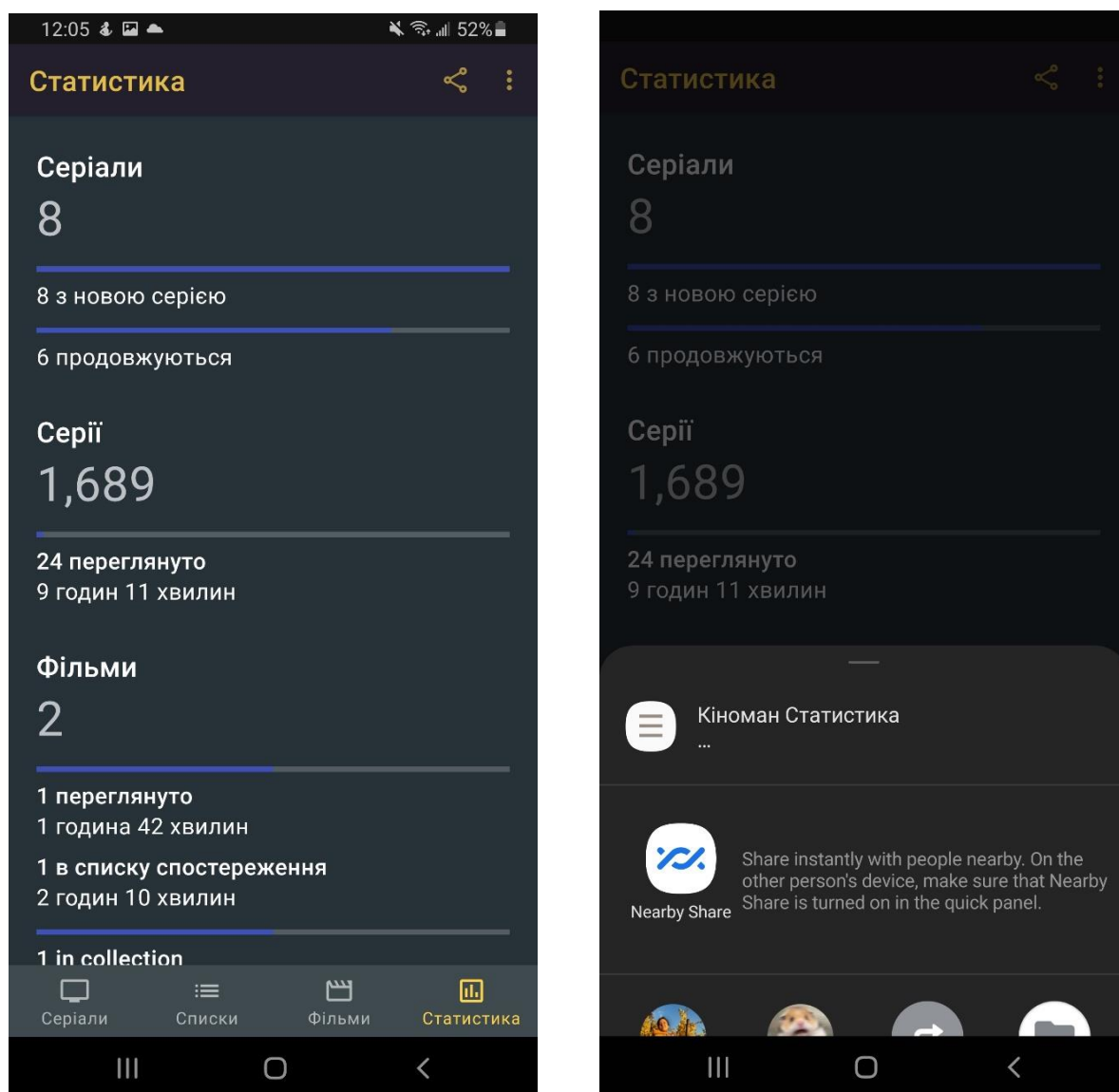


Рисунок 3.5 - Дизайн розділу «Статистика» та меню «Поділитися»

### 3.2 API сторонніх ресурсів

Додаток отримує актуальну інформацію за допомогою API сторонніх ресурсів. Один з таких – це сервіс «TheTVDB», глобальний постачальник метаданих. Заснована в 2006 році, «TheTVDB» є однією з найбільш тривалих баз даних про телебачення та фільми, що керуються громадою. Завдяки метаданим вмісту, доступним для 115 000+ телевізійних серіалів та 141 000+ фільмів, TheTVDB є повноцінним та точним, але доступним рішенням метаданих розваг.

Розробники використовують API метаданих цифрових медіа для своїх програм, службових програм та проектів, генеруючи в середньому понад 1 млрд дзвінків API щодня. «TheTVDB» надає великий каталог метаданих цифрових медіа. Надійний набір даних індустрії розваг TheTVDB включає великий каталог глобальних метаданих відео в різних сферах, доступних через API.

TheTVDB може служити джерелом інформації про фільми та серіали для ряду галузей. Незалежно від того, чи хочете ви використовувати TheTVDB як свій основний інформаційний ресурс, чи використовуєте його для заповнення прогалин у наявних даних, TheTVDB пропонує просте у розгортанні рішення для інтеграції метаданих із багатьма платформами.

Додатки, що об'єднують заголовки відео з кількох розважальних джерел, потребують джерела метаданих від надійної третьої сторони. Високоякісні метадані відео можуть поєднувати різні трансляції та цифрові послуги, роблячи великі каталоги вмісту більш зручними для користувачів.

Другий API, що було використано в розробці даної системи, це «TMDB». база даних фільмів та ТБ, побудована спільнотою глядачів. Кожна частина даних була додана починаючи з 2008 року. Крім метаданих про фільми, телевізійні шоу та людей, пропонується також добірка плакатів із високою роздільною здатністю та фанарти. В середньому щодня додається понад 1000 зображень.

### 3.3 Інструменти розробки

В якості інструменту для розробки дизайну та прототипу було обрано програмний засіб Figma.

Figma - це хмарний засіб дизайну, подібний до Sketch за функціональністю та функціями, але з великими відмінностями, які роблять Figma кращим для співпраці в команді. Figma спрощує процес проектування та є ефективнішою за інші програми, допомагаючи дизайнерам та командам ефективно працювати разом.

Коли здійснюємо вибір мови програмування, то беремо до уваги самі ті задачі, які будуть виконуватися. А також сфера застосування додатку і, звісно, платформа. Розробляється нативне рішення, а тому було вирішено використовувати мови Java та Kotlin, так як вони є рідними для цієї платформи, підтримують ООП і дуже зручні в користуванні з точки зору розробника, в особливості Kotlin.

При розробці даного програмного проекту було обрано середовище розробки Android Studio. Середовище розробки підходить для виконання типових завдань, які вирішуються в процесі розробки додатків для платформи Android. Сюди входять інструменти для спрощення тестування програмного забезпечення на сумісність з різними версіями платформи та інструменти для розробки додатків, що працюють на пристроях з різною роздільною здатністю екрана.

На додаток до функцій, присутніх в IntelliJ IDEA, Android Studio має ще кілька додаткових функцій, таких як нова уніфікована підсистема для побудови, тестування та розгортання програм на основі інструментів збірки Gradle та підтримки використання засобів безперервної інтеграції.

Для пришвидшення розробки додатків існує колекція типових елементів інтерфейсу та візуальний редактор їх макету, який забезпечує зручний огляд різних станів інтерфейсу програми (наприклад, можна побачити, як інтерфейс буде виглядати під різні версії Android та різні розміри екрану). Для створення

нестандартних інтерфейсів існує майстер для створення власних елементів дизайну, що підтримують використання шаблонів. Вбудовані функції для завантаження типових зразків коду з GitHub.

Він також включає вдосконалені інструменти рефакторингу, перевірку сумісності з попередніми версіями, проблеми з продуктивністю, моніторинг пам'яті та оцінки зручності використання, які враховують функціональність платформи Android.

У редактор додано режим швидкого редагування. Система виділення, статичного аналізу та виявлення помилок розширена для підтримки Android API. Вбудована підтримка оптимізатора коду ProGuard. Інтегровані засоби створення цифрових підписів. Присутній інтерфейс для підтримки локалізації додатку.

### **3.4 Об'єкто-орієнтоване проектування**

Для розробки програми продукту обрано парадигму об'єкто-орієнтованого програмування..

ООП - це процедурне програмування, яке використовує класи для групування коду та даних для повторного використання та простоти. Розділяючи програму на класи, легше окремо модифікувати частини програми. Кожен клас є шаблоном для типу об'єкта або екземпляра класу. При розміщенні методів у різних класах примірники кожного класу можуть бути легко модифіковані, замінені та використані повторно. Однак зміна екземплярів класу може ускладнитися великими наборами даних, оскільки важче відстежити, де кожен екземпляр був змінений.

Підставами до вибору даної парадигми є:

. 1. Покращена продуктивність розробки програмного забезпечення: об'єкто-орієнтоване програмування є модульним, оскільки забезпечує розподіл обов'язків при розробці об'єктної програми. Воно також розширюване, оскільки об'єкти можуть бути розширені, включаючи нові атрибути та

поведінку. Об'єкти також можуть бути використані повторно в усіх додатках. Тому ці три фактори - модульність, розширюваність та багаторазове використання - забезпечують підвищення продуктивності розробки програмного забезпечення порівняно з традиційними методиками програмування на основі процедур.

2. Покращена придатність до змін програмного забезпечення: із зазначених вище причин, об'єктивно-орієнтоване програмне забезпечення також простіше в обслуговуванні. Оскільки конструкція модульна, частина система може бути оновлена у разі виникнення проблем без необхідності робити масштабні зміни.

3. Швидший розвиток: повторне використання забезпечує швидший розвиток. Об'єктно-орієнтовані мови програмування постачаються з багатьма бібліотеками об'єктів.

4. Менша вартість розробки: Повторне використання програмного забезпечення також знижує вартість. Як правило, більше зусиль докладається до об'єктно-орієнтованого аналізу та дизайну, що знижує загальну вартість розробки.

5. Якісніше програмне забезпечення: Швидший розвиток програмного забезпечення та нижча вартість розробки дозволяє використовувати більше часу та ресурсів для перевірки програмного забезпечення. Хоча якість залежить від досвіду команд, об'єктно-орієнтоване програмування, як правило, призводить до отримання більш якісного програмного забезпечення.

### **3.5 Реалізація інтерфейсу програми**

Для розробки інтерфейсу мобільного додатку було використано концепції Google Material Design – засади дизайну, розроблені компанією Google. Ця система враховує те, як поведуться користувачі, як взаємодіють з інтерфейсом. Такі атрибути, як тіні, краї, розмірність, надають більше візуальної репліки. Використання знайомих тактильних атрибутів допомагає

користувачам швидко зрозуміти поведінку (підказки про те, як об'єкт повинен використовуватися). Матеріальний дизайн класифікує піктограми як два набори, перший - піктограми товарів, які є жирними та простими. Вони повідомляють суть ідеї та намір продукту. Другий - системні піктограми, що представляють команду, файл, пристрій, каталог або загальні дії. Розмір піктограм продукту дорівнює 48 одиниць, а системних піктограм - 24 одиниці. Колекцію системних піктограм зображено на рисунку 3.6.



Рисунок 3.6 - Колекція системних піктограм

Такі елементи, як типографіка, сітки, простір, масштаб та колір, створюють ієрархію, значення та фокус. Користувальницький досвід збільшується через вибір кольорів, наскрізні зображення, а також навмисні пробіли. Анімація - найкраща частина матеріального дизайну, оскільки вона жодним чином не перериває взаємодію користувача. Анімація посилює той факт, що користувач є першочерговим рушієм. Побудова макету ґрунтується на принципах друкуваного дизайну. Він використовує деякі інструменти, такі як базові сітки та структурні шаблони, щоб поліпшити узгодженість середовищ. Ці макети здатні створювати масштабовані програми, оскільки вони можуть підходити до будь-якого розміру екрана. Фреймворк надає безліч компонентів для використання їх в розробці та дизайні власного додатку. Компоненти

матеріального дизайну, що були використані в ході імплементації даного програмного продукту зображено на рисунку 3.7.

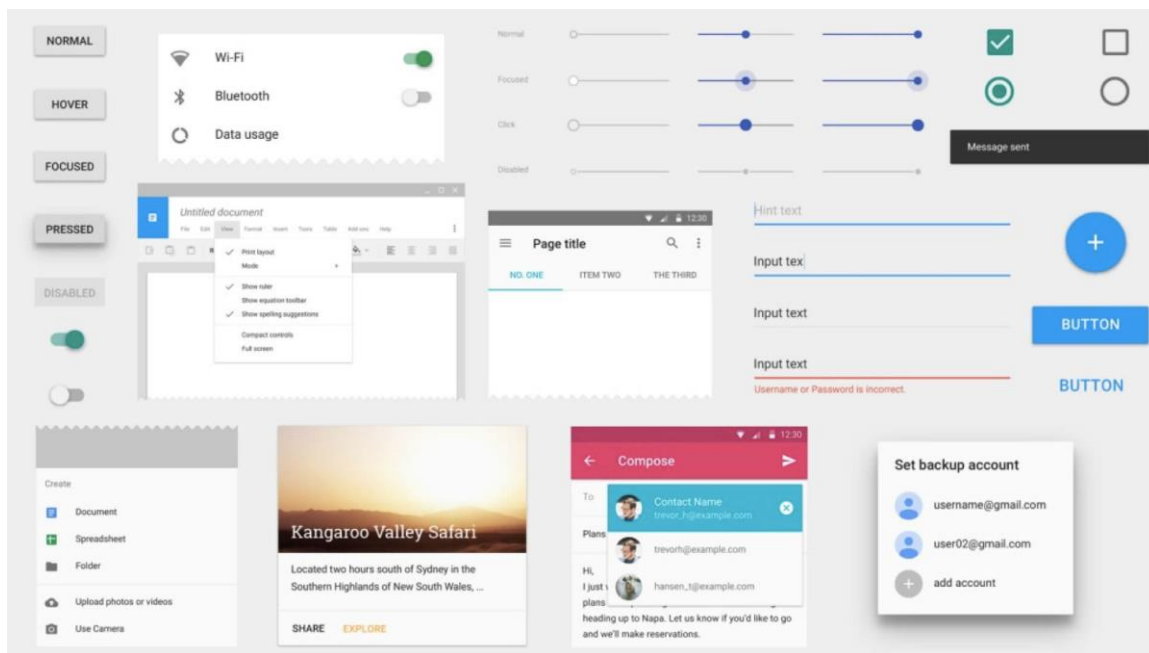


Рисунок 3.7 - Набір компонентів, використаних в програмному продукті

### 3.6 Форма додатку

Додаток виконано у формі нативного. Такі програми завантажуються через магазини програм (App Store, Google Play або їхні філії, магазин додатків Windows тощо) та встановлюються в програмне забезпечення для смартфонів. Важливою відмінністю є те, що нативні програми розробляються спеціально для певної платформи (наприклад, під iOS для iPhone, Android для пристроїв під управлінням Android або Windows для Windows Phone тощо) і вимагають від розробника спеціальних знань та навичок для роботи в певному середовищі розробки (xCode для iPhone, Android Studio для пристроїв Android); більше того, для написання таких програм використовуються лише "рідні" мови програмування. Природно, що сам процес більш трудомісткий.

Нативні програми завжди краще підходять під конкретну ОС і виглядають органічно на смартфоні. Такі програми можуть легко використовувати всі функції програмного забезпечення смартфона (камера, мікрофон, акселерометр,



геолокація, адресна книга, плеєр тощо), і в той же час більш обережно використовувати ресурси телефону (акумулятор, пам'ять). Залежно від мети програми, вони передбачають або не очікують з'єднання з Інтернетом.

Переваги:

- Максимальна функціональність і швидкість роботи
- Для використання не потрібне підключення до Інтернету
- Має доступ до програмного забезпечення для смартфонів (GPS, плеєр, камера)
- Поширення через магазини додатків

Недоліки:

- Вища вартість і довший час розробки
- Потрібен розробник знань конкретного середовища програмування
- Працює лише з однією платформою
- Потрібно випустити оновлення під час змін

Систему розроблено в середовищі Android Studio – найкращому інструменті для створення додатків на всіх типах пристроїв Android.

### **3.7 Взаємодія клієнт-сервер**

Розробка додатку «Колосок» функціонує на основі клієнт-серверної взаємодії. Для отримання інформації з власного серверу та сторонніх API використано HTTP фреймворк Retrofit. Нетворкінг - одна з найважливіших частин програм для Android. Популярною бібліотекою, яка використовується для роботи в мережі, є Retrofit. Це типобезпечний HTTP-клієнт для Android та Java. Використання фреймворку спростило роботу з мережею в додатках Android. Оскільки він має безліч функцій, таких як легке додавання власних заголовків та типів запитів, завантаження файлів, завдяки яким ми можемо зменшити кількість шаблонного коду в програмі та легко підключити веб-сервіс.

Для роботи з бібліотекою нам в основному потрібні наступні три класи:

1. Клас моделі, який використовується як модель JSON.
2. Інтерфейс, в якому визначено всі HTTP операції
3. Клас `Retrofit.Builder`: Екземпляр, якого використовує визначений вище інтерфейс та `API Builder`, щоб дозволити визначення кінцевої точки URL-адреси для HTTP операцій.

Щоб налаштувати роботу з API було виконано наступні кроки:

1. Додано необхідні залежності в файл з назвою `build.gradle`:

```
compile 'com.squareup.retrofit2:retrofit:2.3.0'  
compile 'com.squareup.retrofit2:converter-gson:2.3.0'
```

2. Створено POJO клас або модель даних відповіді з сервера

```
import com.google.gson.annotations.SerializedName  
import java.io.Serializable  
public class ListItem {  
    @SerializedName("list_item_id")  
    public String listItemId;  
    @SerializedName("tvdb_id")  
    public int tvdbId;  
    public String type;  
}
```

3. Створено інтерфейс API, який має методи, що виконують запити HTTP. Надається список анотацій для кожного з методів HTTP, таких як `@GET`, `@POST`, `@PUT`, `@DELETE`, `@PATCH` тощо:

```
import retrofit2.http.*  
interface ApiService {  
    @GET("search/movie")  
    Call<MovieResultsPage> movie(  
        @Query("query") String query,  
        @Query("page") Integer page,  
        @Query("language") String language,  
        @Query("region") String region,  
        @Query("include_adult") Boolean includeAdult,  
        @Query("year") Integer year,  
        @Query("primary_release_year") Integer primaryRelease  
    );  
}
```

4. Створено екземпляр класу `Retrofit.Builder`:

```
val retrofit = Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create())
    .build()
```

5. Створено екземпляр сервісу та викликати метод всередині нього. Це можна зробити в контролері, щоб отримати необхідні дані

```
val shareService = retrofit.create(ApiService.class);
val shareData = shareService.getShareData().execute()
```

Таким чином отримуються дані з сервера дані з сервера, а потім виконуються подальші дії з обробки цих даних.

Додаток не потребує доопрацювання, коли внесені зміни на сервері чи додалася нова інформація.

### 3.8 Етапи розробки та випуску додатка

Доцільно буде розглянути процес розробки додатку «Кіноман» на прикладі платформи Android. Перш за все слід установити обов'язкові елементи розробника для Android: Android Studio, Java Development Kit (JDK); Gradle; Android SDK.

Після встановлення вищеназваних пакетів, слід приступити до розробки графічної складової. Макет було створено за допомогою продукту Figma. А коли прототип готовий, то наступним кроком буде «верстка» дизайну у вигляді XML-файлів. Для кожного модуля чи сторінки рекомендовано створювати окремий файл. Це сприятиме легкості пошуку потрібного елемента в проекті згодом. Середовище розробки надає всі інструменти, щоб реалізувати дизайн у формі дизайн в майбутньому додатку. Приклад реалізованого коду елемента:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="wrap_content">
<TextView
    android:id="@+id/textViewItemListName"
    style="@style/Widget.SeriesGuide.TextView.SingleLine"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ellipsize="end"
```

```
        android:paddingBottom="@dimen/large_padding"  
        android:paddingLeft="40dp"  
        android:paddingTop="@dimen/large_padding"  
        android:textAppearance="  
@style/TextAppearance.SeriesGuide.Subtitle1"  
        tools:ignore="RtlSymmetry"  
        tools:text="Sample List Name" />  
</FrameLayout>
```

Наступним етапом є подубова архітектури проекту, структури директорій. Підключення потрібних модулів та бібліотек. Яке відбувається із допомогою файлу `build.gradle`. В проекті використані такі допоміжні фреймворки, як Google Architecture Components, Gson, Retrofit, Timber, Picasso, Firebase, Room, а також декілька рішень, що полегшують тестування проекту, написання юніт-тестів.

Найважливішою складовою є перелік, так званих, архітектурних компонентів. Вони були створені з метою розробки програм для Android, в яких компоненти будуть не сильно зв'язані, а отже, більш придатні для тестування та підтримки. Для досягнення вищезазначеного компоненти мають на меті вирішити дві найпоширеніші проблеми в додатках:

1. Управління життєвим циклом активності та фрагмента.
2. Збереження даних

Підводячи підсумок, компоненти архітектури - це отримання даних з бази даних чи REST API та оновлення інтерфейсу користувача з цими даними таким чином, щоб брати до уваги життєвий цикл інтерфейсу користувача (Activity / Fragment). Їх складовими архітектури є: Lifecycle, LiveData, ViewModel, Room, Paging Library.

Після того, як архітектура побудована, відбувається налаштування роботи з Rest API та сторонніми сервісами, як TVdb чи MDB та розробка локальної бази даних, яка базується на раніше згаданій бібліотеці Room.

Наступним кроком постає реалізація логіки отримання даних та відображення для користувача. Варто зазначити, увесь програмний код написаний мовами Java та, переважно, Kotlin. Останній вже встиг стати стандартом в галузі завдяки своїм якостям. Kotlin дозволяє писати менше коду.

І крім того зазначає його читабельним. Компанія JetBrains зробила все можливе, щоб зробити мову якомога коротшою, і їм це вдалося.

Менше коду, якщо все зробити правильно, призводить до меншої кількості помилок. Коли ви дозволяється фреймворку піклуватися про певні буденні аспекти кодування, можна зосередитися на більш важливих речах. Kotlin - це все для високої читабельності, простоти та полегшення процесу розробки додатків.

Kotlin походив із галузі, а не академічних кіл. JetBrains потрібен простіший інструмент, ніж Java, для роботи з основним продуктом, який називається IntelliJ IDEA, повністю написаним на ньому.

Вони дослідили альтернативи, залишились незадоволеними та вирішили винайти власну мову. Метою було отримати ефективний інструмент, який можна використовувати разом з Java і працювати скрізь, де Java може.

Отже, Kotlin вирішує проблеми, з якими стикаються програмісти. Наприклад, система типів допомагає уникнути винятків з нульовим покажчиком. Дослідницькі академічні мови, як правило, взагалі не мають нульового значення, але це не приносить користі інженерам програмного забезпечення, що працюють з великими базами кодів та API, які це роблять. Ще однією перевагою є те, що мова програмування Kotlin повністю сумісна з Java. Розробники Kotlin мають доступ до всіх фреймворків і бібліотек Java, пишучи більш чіткий та стислий код.

Отже, інженери Android можуть використовувати обидві мови під час створення мобільного продукту, а також переносити свої програми з Java на Kotlin. JetBrains постійно вдосконалює мову. Він відданий проекту, над ним працює велика та високо компетентна команда, має стабільну бізнес-модель і навіть перетворює частини власного флагманського продукту на його використання.

Написання проекту завершується його тестуванням. Велику роль в цьому відіграють юніт-тести. Код передається для інтеграційного тестування тоді і тільки тоді, коли він проходить усі свої модульні тести. Коли код проходить усі

свої тестові приклади, він називається повним. Як результат, модульне тестування надає багато переваг, таких як ранні пошук програмних помилок, простіший рефакторинг старого коду та як джерело документації.

Юніт-тестування перевіряє найменший фрагмент коду, який можна ізолювати, і перевіряє, чи працює він правильно, передаючи його через деякі тестові кейси. Під час модульного тестування всі залежності цієї частини коду забезпечуються механізмом, таким як Mockito. Написання модульних тестів змушує добре подумати над такими речами, як архітектура та шаблон дизайну коду. В результаті буде написано якісний код, який є елегантним, легшим для читання і головне легшим для рефакторингу в майбутньому.

Модульне тестування є важливим інструментом у наборі інструментів будь-якого серйозного розробника програмного забезпечення. Однак іноді буває досить складно написати хороший модульний тест для певної частини коду. Виникають труднощі з тестуванням власного чи чужого коду, розробники часто думають, що їх боротьба спричинена браком деяких фундаментальних знань тестування або секретних методів модульного тестування.

Модульний тест - це метод, який створює невелику частину нашого додатка та перевіряє його поведінку незалежно від інших частин. Типовий модульний тест містить 3 етапи: спочатку він ініціалізує невелику частину програми, яку хоче протестувати (також відомо як система, що тестується, або SUT), потім застосовує певний стимул до системи, що перевіряється (як правило, викликаючи метод на ньому), і, нарешті, він спостерігає результуючу поведінку. Якщо спостерігається поведінка відповідає очікуванням, модульний тест проходить, інакше він не вдається, вказуючи на те, що десь у системі, що перевіряється, є проблема. Ці три фази одиничного тестування також відомі як упорядкування, дію та затвердження, або просто AAA.

Одиничний тест може перевірити різні поведінкові аспекти тестованої системи, але, швидше за все, він потрапляє в одну з наступних двох категорій: на основі стану або взаємодії. Перевірка того, що система, що перевіряється, дає правильні результати, або що її результуючий стан правильний, називається

модульним тестуванням на основі стану, а перевірка того, що вона правильно викликає певні методи, називається модульним тестуванням на основі взаємодії.

Як метафору для належного тестування програмного забезпечення, можна уявити божевільного вченого, який хоче побудувати якусь надприродну химеру із жаб'ячими лапами, щупальцями восьминога, пташиними крилами та головою собаки. (Ця метафора досить близька до того, що насправді роблять програмісти на роботі). Як би цей учений переконався, що кожна вибрана ним частина (або одиниця) насправді працює? Що ж, він може взяти, скажімо, одну ногу жаби, застосувати до неї електричний подразник і перевірити на правильне скорочення м'язів. Те, що він робить, - це, по суті, ті самі кроки Arrange-Act-Assert, що й модульний тест; єдина відмінність полягає в тому, що в цьому випадку одиниця посилається на фізичний об'єкт, а не на абстрактний об'єкт, з якого ми будуємо наші програми.

Ще одна важлива річ, яку слід врахувати, - це різниця між модульним тестуванням та інтеграційним тестуванням.

Метою модульного тестування в програмній інженерії є перевірка поведінки відносно невеликого програмного забезпечення, незалежно від інших частин. Модульні тести вузькі за обсягом і дозволяють охопити всі випадки, гарантуючи правильну роботу кожної окремої частини.

З іншого боку, інтеграційні тести демонструють, що різні частини системи працюють разом у реальному середовищі. Вони перевіряють складні сценарії (ми можемо уявляти інтеграційні тести як користувача, який виконує якісь операції високого рівня в нашій системі), і зазвичай вимагають наявності зовнішніх ресурсів, таких як бази даних або веб-сервера.

Повернімось до нашої божевільної вченої метафори і припустимо, що він успішно поєднав усі частини химери. Він хоче провести інтеграційний тест отриманої істоти, переконавшись, що вона може, скажімо, ходити по різних типах місцевості. Перш за все, вчений повинен наслідувати середовище, по якому істота може ходити. Потім він кидає істоту в це середовище і тикає нею

палицею, спостерігаючи, чи вона ходить і рухається, як задумано. Закінчивши випробування, божевільний учений очищає весь бруд, пісок і гірські породи, які зараз розкидані в його милій лабораторії.

Зверніть увагу на суттєву різницю між модульними та інтеграційними тестами: модульний тест перевіряє поведінку невеликої частини програми, ізольованої від навколишнього середовища та інших частин, і досить простий у реалізації, тоді як інтеграційний тест охоплює взаємодію між різними компонентами, в середовище, близьке до реального, і вимагає більших зусиль, включаючи додаткові налаштування та етапи відключення.

Розумна комбінація модульних та інтеграційних тестів гарантує, що кожен окремий блок працює коректно, незалежно від інших, і що всі ці блоки працюють добре в інтеграції, що дає нам високий рівень впевненості в тому, що вся система працює належним чином.

Для тестування додатку було розроблено план дій: відібрані модулі для тестування; тестування має відбуватися всіма методами; тестування відбуватиметься знизу вгору.

Завдяки середовищу Android Studio було створено всі види тестів.

Після написання та проходження усіх тестів, додаток готовий до збірки та ручного тестування. Android вимагає, щоб усі файли .apk мали цифровий підпис сертифікатом, перш ніж вони будуть встановлюватися на пристрої. У разі, якщо використовується Android App Bundle, то потрібно підписати лише свій пакет додатку, перш ніж завантажувати його на Play Console, а Play App Signing подбає про все інше. Однак також є можливість вручну підписати свій додаток для завантаження в Google Play та інші магазини додатків.

За допомогою Play App Signing Google керує та захищає ключ підписання додатка і використовує його для підписання файлів .apk для розповсюдження. Оскільки набори додатків відкладають створення та підписання файлів .apk у Google Play Store, перед завантаженням пакету додатків потрібно увімкнути підписання додатків Play. Це дозволить скористатися наступним:



- Використовувати Android App Bundle і підтримувати розширені режими доставки Google Play. Набір Android App Bundle робить додаток набагато меншим, а випуски простішими, також дозволяє використовувати функціональні модулі.

- Підвищити безпеку свого ключа підписання та зробити можливим використання окремого ключа завантаження для підписання набору програм, який ви завантажуватимете в Google Play.

Увімкнення підпису Play App застосовується протягом усього терміну дії додатка. Щоб забезпечити безпеку, після вибору відсутня можливість отримати копію ключа підпису додатка, а також видалити його з серверів Google, не видаливши програму.

У програмі Play App Signing використовуються два ключі: ключ підписання програми та ключ завантаження. Розробник зберігає ключ завантаження та використовує його для підписання програми для завантаження в Google Play Store. Використовуючи окремий ключ завантаження, він може подати запит на скидання ключа завантаження, якщо ключ коли-небудь буде втрачено або порушено. Для порівняння, якщо розробник не вибрав підписування додатків через Google Play і загубив ключ підписання додатка, то втрачесь можливість оновити програму.

Коли додаток буде готовим до публікації, то можна підписати свій додаток за допомогою Android Studio, завантажити його в Google Play. Ключ, за допомогою якого підписується програма, стає ключем завантаження програми. Google використовує сертифікат завантаження для підтвердження особи та підписує файли .apk із ключами підпису додатка для розповсюдження, як показано на рисунку 3.8.

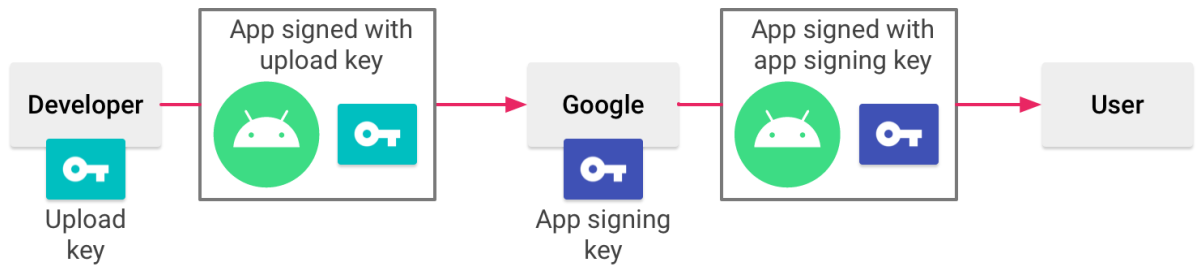


Рисунок 3.8 - Процедура підпису мобільного додатку перед розповсюдженням

Якщо ключ підписання програми відсутній, то його можна створити під час реєстрації. Підписуючи програму, інструмент підписання додає сертифікат до програми. Сертифікат пов'язує файл .apk або пакет додатків із відповідним приватним ключем. Це допомагає Android гарантувати, що будь-які майбутні оновлення додатку будуть справжніми та надходять від оригінального автора. Ключ, який використовується для створення цього сертифіката, називається ключем підписання програми.

Сертифікат для ключа підписання програми та ключа завантаження можна завантажувати зі сторінки підписання програми в Play Console, щоб зареєструвати свої ключі у постачальників API. Сертифікат може бути переданий будь-кому. Він не містить приватного ключа.

Кожен додаток повинен використовувати один і той же сертифікат протягом усього свого життя, щоб користувачі могли встановлювати нові версії як оновлення програми. Коли все готово до опублікації, потрібно підписати додаток та завантажити в магазин програм, наприклад, Google Play. Публікуючи у Google Play, слід також увімкнути підписання додатків Play. В середовищі розробки Android Studio наявні все необхідні інструменти, щоб це зробити. Меню для створення сховища та підпису додатку показано на рисунку 3.9.

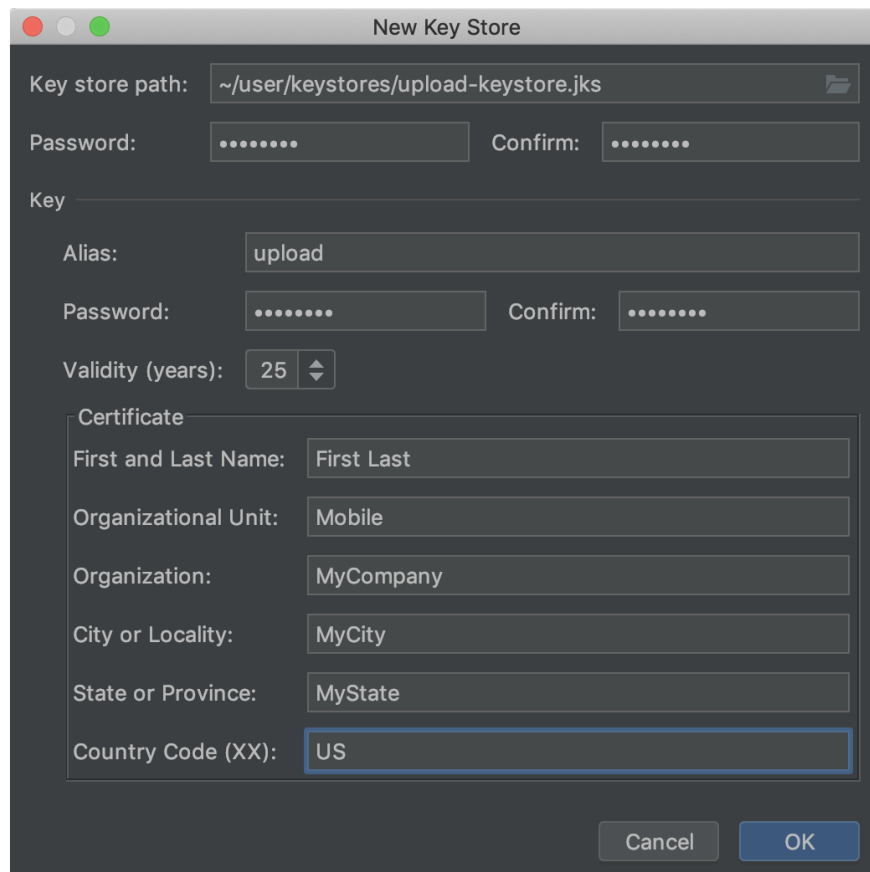


Рисунок 3.9 - Меню створення сховища та підпису додатку

Після того, як Android Studio закінчить створення підписаної програми, її можна знайти або проаналізувати, натиснувши відповідну опцію у спливаючому сповіщенні.

Після виконання усіх перелічених кроків отримано повноцінний додаток, готовий до публікації в інтернет чи на офіційній платформі Google Play.

Використання розробленого додатку в роботі забезпечує своєчасне надходження довідникової інформації, акумуляції корисних даних для користувачів, зниження витрат на пошук інформації та економію часу. Інтерфейсу користувача, забезпечує репрезентацію даних та легку взаємодію. В ході розробки задіяно всі найсучасніші підходи в галузі мобільних сервісів. Такий спосіб концентрованого надання інформації на персональному мобільному пристрої надає найбільш приємний досвід для усіх користувачів, яким небайдуже кіно.

## ВИСНОВКИ

В результаті виконаної роботи, було спроектовано та розроблено Android додаток для забезпечення доступу до бази даних усіх шоу, фільмів, серіалів. Відомостей про дату виходу, жанр, тривалість, середню оцінку від глядачів, кількості та назв сезонів і епізодів конкретного продукту, різноманітної довідкової інформації.

Призначення додатку - це надання кінолюбителям інформації про новинки в галузі кіно, стеження за виходом нових епізодів чи цілих картин. Ведення власної історії переглядів, керувати різноманітними списками, чартами. А також перегляд статистики на власному мобільному пристрої. В програмі є функції створення авторських підбірок, реалізована також локальна база даних. Присутня можливість відзначити ту чи іншу картину, як «переглянуту» або «пропустити», а також додати у свою колекцію. Крім того, на основі дій користувача ведеться статистика, яка включає кількість переглянутих серій, загальний час витрачений, кількість фільмів в списках переглянутих, списку спостереження та в колекції.

Програма виконана у формі мобільного клієнта. В якості джерела даних використовується онлайн сервіс «TheTVDB», а також «TMDB».

Додаток має інтерактивний, зручний інтерфейс для перегляду даних з серверу, відображення інформації.

Було проаналізовано предметну область та показано проблеми в галузі розробки додатків під мобільні пристрої.

Виконано, протестовано, впроваджено програму-додаток для Android.

Для досягнення успіху, було проаналізовано підходи та методології, отримано результати такої роботи.

В роботі виконано аналіз існуючих серверних технологій для розробки мобільних додатків, виявлення їх переваг та недоліків. Спроектовано та розроблено мобільний додаток для операційної системи Android, для надання та зображення отриманих результатів.

В сьогоднішній час, використання додатків повністю поширилося у всіх сферах та галузях, у бізнесі, розвагах та освіті. Вони переймають на себе роботу, яка виконувалася в ручну, надають швидкий доступ до інформації та з чудовим засобом миттєвої комунікації.

Смартфони дали нове обличчя сектору послуг завдяки винаходу мобільних додатків. Мобільні програми - це програми, призначені для виконання певного завдання. Споживання часу зменшилось із популярністю мобільних додатків, і тепер товари та послуги знаходяться на кінчиках пальців. Послуг, які надають мобільні програми та смартфони, безліч.

В ході роботи здійснено теоретичне узагальнення і запропоновано нові практичні рекомендації для створення мобільного додатку.

Таким чином додаток – це така програмна система для мобільних пристроїв, смартфонів, планшетів, що встановлюється через магазини додатків і має на меті вирішення завдань користувача.

У системі можна знайти шоу з новими епізодами, які популярні або схожі на ті, що переглядаються. Присутній функціонал дослідження фільмів, які зараз перебувають у кінотеатрах, випускаються цифровим шляхом або на диску. Також доступне резервне копіювання та синхронізація.

Можна стверджувати, що програмний продукт отримає хорошу репутацію та широку популярність завдяки своїм функціональним можливостям.

Після розробки була написана вся необхідна документація.

Було виконано все заплановане в постановці завдання. Згідно розрахованих показників, даний програмний продукт повинен покрити та забезпечити усі потреби користувачів.