

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «LISBOOK» МОВОЮ
ПРОГРАМУВАННЯ JAVA»**

Виконав: студент 4 курсу, групи ПД–41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Лях О.А.

(прізвище та ініціали)

Керівник Негоденко О.В.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ –2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ ____ ” _____ 2023 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ЛЯХ ОЛЬГИ АНДРІЇВНИ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка мобільного додатку «LisBook» мовою програмування Java»

Керівник роботи: Негоденко О.В., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «24» лютого 2023 року №26.

2. Строк подання студентом роботи «1» червня 2023 року

3. Вхідні дані до роботи

3.1 Мова програмування Java;

3.2 Науково-технічна література з питань, пов'язаних з розробкою мобільних додатків під платформу Android;

3.3 Система управління базами даних SQLite.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Аналіз предметної області

4.2 Аналіз та дослідження існуючих аналогів

4.3 Аналіз засобів розробки

4.4 Формування вимог до мобільного додатку

4.5 Розробка програмного забезпечення

4.6 Тестування розробленого мобільного додатку

5. Перелік демонстраційного матеріалу (назва основних слайдів)

5.1 Мета, об'єкт та предмет дослідження

5.2 Аналоги

5.3 Технічне завдання

5.4 Програмні засоби реалізації

5.5 Розроблений мобільний додаток для обліку книг

5.6 Висновки

6. Дата видачі завдання «25» лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23-27.02.23	Виконано
2	Дослідження та аналіз існуючих аналогів	28.02.23-01.03.23	Виконано
3	Дослідження програмних засобів	02.03.23-04.03.23	Виконано
4	Моделювання інтерфейсу та внутрішньої структури мобільного додатку	05.03.23-01.04.23	Виконано
5	Розробка мобільного додатку	02.04.23-01.05.23	Виконано
6	Вступ, висновки реферат	02.05.23-15.05.23	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	16.05.23-21.05.23	Виконано
8	Попередній захист роботи	22.05.23	
9	Здача роботи	01.06.23	

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 55с., 31 рис., 27 джерел.

JAVA, ANDROID STUDIO, FIGMA, SQLITE, МОБІЛЬНИЙ ДОДАТОК, ЦИФРОВА БІБЛІОТЕКА, ANDROID OS.

Об'єкт дослідження – процес обліку книг в персональній бібліотеці.

Предмет дослідження – мобільний додаток для обліку книг та ведення персональної бібліотеки.

Мета роботи – покращення процесу обліку книг та створення персональної бібліотеки за допомогою мобільного додатку мовою програмування JAVA.

Методи дослідження – методи проектування та розробки програмного забезпечення, методи проектування користувацького інтерфейсу, методи опрацювання та аналізу отриманих даних, аналіз існуючих додатків, методи тестування програмного забезпечення.

Для реалізації мети потрібно вирішити наступні проблеми:

1. Провести аналіз існуючих мобільних додатків та обрати інтегроване середовище розробки Android Studio для програмування мовою Java;
2. Створити візуалізацію роботи додатка за допомогою засобів UML;
3. Спроекувати модель користувацького user-friendly інтерфейсу в онлайн-сервісі Figma;
4. Побудувати структуру бази даних на основі SQLite;
5. Розробити мобільний додаток для ведення персональної цифрової бібліотеки, що спрощує облік книг користувача за допомогою обраних засобів;
6. Провести мануальне тестування додатку.

Галузі використання – освіта, наукова діяльність, розваги.

ЗМІСТ

ВСТУП	10
1 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ	12
1.1 Turn - Reading Tracker & Timer.....	13
1.2 Goodreads	14
1.3 my Record Record Book Tracker або My Reading Record	15
2 ЗАСОБИ РОЗРОБКИ ДОДАТКУ	17
2.1 Сервіс з розробки інтерфейсів Figma	17
2.2 Мова програмування Java	17
2.3 Середовище розробки Android Studio	18
2.3.1 Мова розмітки та стилів XML	18
2.3.2 Структура проекту в Android Studio.....	18
2.4 SQLite – компактна СУБД	21
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	23
3.1 Діаграма прецедентів	24
3.2 Діаграма діяльності	25
3.3 Діаграма пакетів	26
3.4 Діаграма класів	28
3.5 Модель бази даних	29
3.6 Проектування користувацького інтерфейсу	30
4 ПРОГРАМНА РЕАЛІЗАЦІЯ	31
4.1 Створення frontend частини додатку	31
4.1.1 Ініціалізація кольорів	31
4.1.2 Визначення рядків та локалізація	32
4.1.3 Створення тем та стилів.....	33
4.1.4 Заповнення папки drawable.....	34
4.1.5 Створення Activity та їхніх елементів інтерфейсу	34
4.2 Створення бази даних	41
4.2.1 Модель Book.....	42

4.2.2 Адаптер для бази даних	43
4.2.3 Запис даних у csv файл	44
4.3 Програмування backend-частини.....	44
4.3.1 RecyclerViewAdapter	45
4.3.2 Створення меню та сторінок із показом списку книг за одним параметром	46
4.3.3 CRUD-сторінки	48
4.3.4 Сторінка налаштувань.....	50
4.4 AndroidManifest файл	51
4.5 Тестування	53
ВИСНОВКИ	55
ВИКОРИСТАНІ ДЖЕРЕЛА	56
ДОДАТОК А	59
ДОДАТОК Б	67
ДОДАТОК В	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

APK – Android Package Kit

UI – User Interface

UX – User Experience

UML – Unified Modeling Language

SDK – Software Development Kit

ОС – Операційна Система

ПЗ – Програмне Забезпечення

СУБД – Система Управління Базами Даних

ВСТУП

Обґрунтування вибору теми та її актуальність: Книги стимулюють розумову діяльність людини, розкривають творчий потенціал та покращують пам'ять. Книги є одвічним супутником людини із самого народження і до глибокої старості: від казок до класичної літератури та статей у періодичних виданнях. Із розвитком інформаційних та цифрових технологій літературні видавництва, бібліотеки, книгарні та вільні автори все більше починають викладати твори в електронному вигляді. Старі архівні документи, креслення найперших механізмів та багато іншого, що має історичну цінність, також оцифровують і надають доступ до них через інтернет мережу. На сьогодні поціновувачі книг зіштовхнулись із проблемою їхнього обліку у персональному користуванні. Сьогодні виникає потреба у персональних цифрових бібліотеках та щоденниках. Потреба додатку, який був би простим у користуванні та функціональним водночас.

Об'єкт дослідження – процес обліку книг в персональній бібліотеці.

Предмет дослідження – мобільний додаток для обліку книг та ведення персональної бібліотеки.

Мета роботи – покращення процесу обліку книг та створення персональної бібліотеки за допомогою мобільного додатку мовою програмування JAVA.

Методи дослідження – методи проектування та розробки програмного забезпечення, методи проектування користувацького інтерфейсу, методи опрацювання та аналізу отриманих даних, аналіз існуючих додатків, методи тестування програмного забезпечення.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Провести аналіз існуючих мобільних додатків та обрати інтегроване середовище розробки Android Studio для програмування мовою Java;
2. Створити візуалізацію роботи додатка за допомогою засобів UML;

3. Спроекувати модель користувацького user-friendly інтерфейсу в онлайн-сервісі Figma;

4. Побудувати структуру бази даних на основі SQLite;

5. Розробити мобільний додаток для ведення персональної цифрової бібліотеки, що спрощує облік книг користувача за допомогою обраних засобів;

6. Провести мануальне тестування додатку.

Практична значущість результатів: Такий додаток цифрової бібліотеки спростить облік книг, групування корисних статей та посилань. Користувач зможе зберігати нотатки до твору, бачити свій прогрес у читанні, виставляти книзі рейтинг, тощо. В свою чергу, така платформа дає можливість книгарням рекламувати новинки та підвищувати інтерес користувача до читання, а отже і саморозвитку. Це розширює коло потенційних користувачів.

1 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ

Поціновувачі книг завжди бажають мати паперовий примірник книги, але є випадки, коли людина просто не може собі цього дозволити через певні обставини.

На інфографіці (рисунок 1.1) згідно дослідження, що було проведено проектом «Ukrainian Reading and Publishing Data 2018» за підтримки Українського інституту книги, можна побачити, що 35.7% з опитаних не мають місця де зберігати книжки, щоб створити домашню бібліотеку. А 13.9% людей вважають, що зручніше придбати або завантажити електронну версію книжки.[1]

Чому Ви не збираєте домашню бібліотеку?

% тих, хто читає книжки і не збирає домашньої бібліотеки

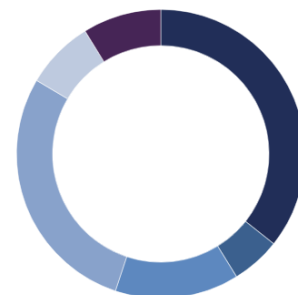


Рисунок 1.1 Інфографіка відповіді на питання «Чому Ви не збираєте домашню бібліотеку?»

Однак потреба у веденні обліку прочитаних книг залишається. Саме тут на допомогу приходять сучасні технології, адже, згідно того ж опитування, найдоступнішим пристроєм для користування є смартфон – 70%.

Які з наведених пристроїв є доступними для вашого користування?

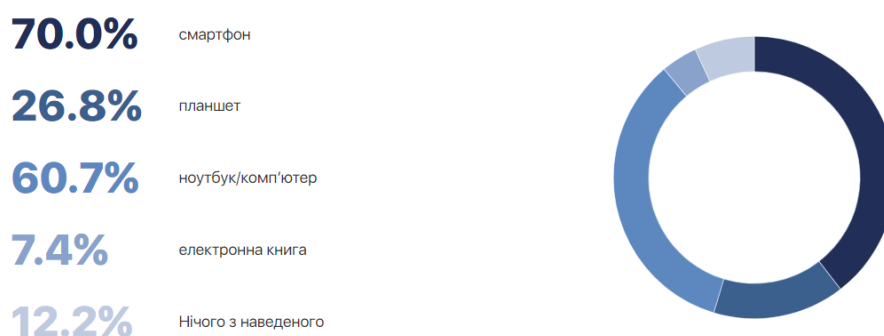


Рисунок 1.2 Інфографіка відповіді на питання «Які з наведених пристроїв є доступними для вашого користування?»

1.1 Turn - Reading Tracker & Timer

Turn - Reading Tracker & Timer – розроблений, як трекер книг та бібліотеки, щоб допомогти користувачу слідкувати за звичкою до читання та легко організувати свою книжкову колекцію під час читання.[2]

Функції додатку:

- Відслідковування сеансів читання;
- Збереження нотаток та цитат;
- Додавання книги в бібліотеку за допомогою штрих-коду або вручну;
- Сортування за категоріями/полицями;
- Аналітика читання;
- Можливість поставити оцінку книзі;
- Імпорт та експорт даних.

Переваги:

- Можна додати обкладинку;
- Можна відсортувати книги за датою додавання;
- Можна відсканувати ISBN код паперової книги;
- Зручна система цілей.

Недоліки:

- Не зручне розділення за категоріям;

- Відсутня українська мова;
- Відсутня можливість додати посилання на книгу.

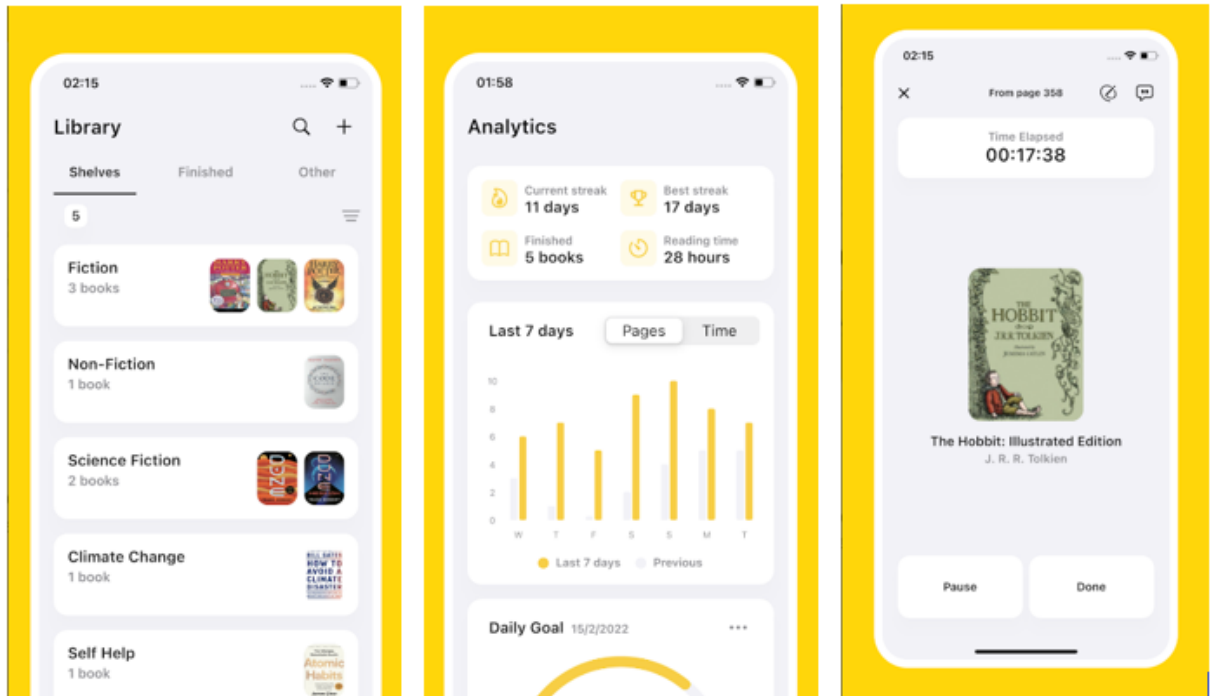


Рисунок 1.3 Приклади екранних форм додатку «Turn - Reading Tracker & Timer»

1.2 Goodreads

Goodreads – додаток, що базується на веб-сайті для читачів і книжкових рекомендацій. Більше схожий на соціальну мережу для любителів книг. Користувач може залишати відгуки про книжки, оцінювати їх та бачити «книжкові полиці» своїх друзів.[3]

Функції додатку:

- Сканування книг за обкладинкою;
- Персоналізовані рекомендації на основі вибору користувача;
- Система друзів та перегляд їхніх полиць;
- Система купівлі електронних книг;
- Трекінг статусу книг.

Переваги:

- Зручний пошук по обкладинці;
- Мережа друзів та система підписок і коментарів;
- Пропозиції книжок на основі користувацьких вподобань.

Недоліки:

- Відсутня можливість експорту даних;
- Відсутня можливість зміни теми;
- Обов'язкова реєстрація;
- Відсутня українська мова.

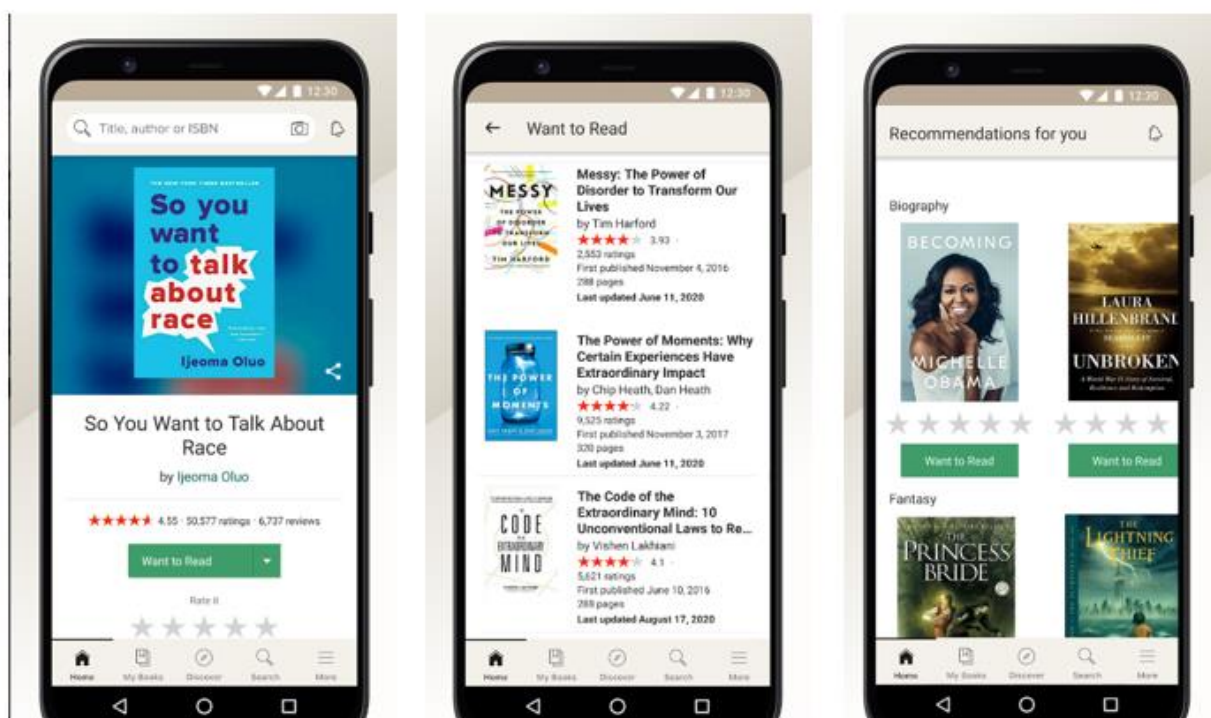


Рисунок 1.4 Приклади екранних форм додатку «Goodreads»

1.3 my Record Record Book Tracker або My Reading Record

My Reading Record – програма для відстеження читання, щоб допомагати відстежувати та розділяти за каталогами поточні та минулі книги, електронні та аудіо версії.[4]

Функції додатку:

- Додавання книги в бібліотеку за допомогою штрих-коду або вручну;
- Створення цілей прочитання на день, місяць та рік;
- Ведення статистики читання;
- Розділення книг на полиці: хочу прочитати, прочитано.

Переваги:

- Зручний пошук книг в інтернеті;
- Можливість встановити денну/місячну/річну ціль;
- Зручна система статистики.

Недоліки:

- Не зручне розділення за категоріям;
- Відсутня можливість експорту;
- Відсутня українська мова.



Рисунок 3.2 Приклади екранних форм додатку «My Reading Record»

2 ЗАСОБИ РОЗРОБКИ ДОДАТКУ

Для розробки мобільного додатку персональної бібліотеки «LisBook» потрібно спроектувати користувацький інтерфейс, структуру бази даних, а також розробити сам застосунок. Платформою та операційною системою додатку було обрано Android, середовищем розробки Android Studio, яка підтримує базу даних SQLite та мову програмування Java, а для проектування інтерфейсу обрано онлайн-сервіс Figma.

2.1 Сервіс з розробки інтерфейсів Figma

Figma – це зручний та сучасний безкоштовний онлайн-сервіс з проектування користувацьких інтерфейсів.[5] Сервіс має дві версії: браузерну та як клієнтський додаток на пристрої користувача. Figma має багатий набір функцій для прототипування веб-сайтів, мобільних додатків або окремих елементів інтерфейсу. Також, зберігає користувацькі версії файлів і надає можливість спільного редагування та перегляду проекту.

2.2 Мова програмування Java

Java – одна з найпопулярніших об'єктно-орієнтованих високорівневих мов програмування на сьогодні.[6] Зараз Java це не тільки мова програмування, але і ціла платформа, окрема «еко-система», яка має широкий спектр застосування: розробка веб-порталів та сервісів, створення програмного забезпечення, ігор та ігрових модулів, мобільних додатків, тощо.

Однією з особливостей мови є автоматична збірка «сміття» за допомогою Garbage Collector та динамічна робота із пам'яттю, що значно полегшують роботу розробника.

2.3 Середовище розробки Android Studio

Android Studio – інтегроване середовище розробки, яке дозволяє створювати програми для ОС Android.[7] Містить редактор коду та багато інструментів для розробки не тільки backend, але і frontend частини, інтегровані засоби для роботи із системами контролю версій.

Ще одна особливість та зручність середовища – емулятор. Емулятор – це симуляція, своєрідне відтворення потенційного пристрою користувача. Розробник може обрати модель, розміри екрану не лише смартфона чи планшету, але і, наприклад, телевізора, системи Android Auto або Smart Watch. Це значно покращує процес тестування та створення UI.

2.3.1 Мова розмітки та стилів XML

В Android Studio XML код зберігається в файлах з розширенням .xml, які є ресурсами та визначенням візуального представлення інтерфейсу.[8] Оголошення користувацького інтерфейсу в XML дозволяє відокремити екранні форми програми від коду, який керує її поведінкою та функціями. Мова має гнучкі засоби розробки на основі яких можна створити динамічний UI під різні розміри та положення екрану.

2.3.2 Структура проекту в Android Studio

Система збірки Android Studio базується на Gradle – вдосконаленому наборі інструментів збірки програми, для автоматизації процесу збирання та керування ним, дозволяючи визначити гнучкі користувацькі конфігурації збірки.[9] Система збірки Android компілює ресурси програми та вихідний код і пакує їх у файли APK або Android App Bundle, які можна тестувати, розгортати на пристроях та розповсюджувати. Усі файли збірки розташовані в розділі «Gradle Scripts».

Тека «manifests» – містить AndroidManifest.xml файл. Кожен проект програми повинен мати файл саме із такою назвою в корені вихідного набору

проекту. Файл «маніфесту» описує основну інформацію про програму для інструментів збірки Android, ОС Android і сервісу Google Play. В цьому ж файлі зберігається назва додатку та шлях до його іконки.[10]

У файлі AndroidManifest потрібно декларувати:

- Компоненти програми, включно з усіма «Activity». Кожен компонент повинен визначати основні властивості, такі як ім'я класу Kotlin або Java. Він також може декларувати можливості, такі як конфігурації пристроїв, які додаток може обробляти, і фільтри намірів, які описують, як можна запустити компонент;

- Дозволи, що потрібні програмі для доступу до захищених частин системи, сервісів та інших програм. Тут також оголошуються будь-які дозволи, які інші програми повинні мати, щоб отримати доступ до вмісту цієї програми;

- Функції апаратного та програмного забезпечення, необхідні програмі, що впливає на те, на яких пристроях можна встановити програму з сервісу Google Play.

В теці «java» – зберігаються файли із розширенням .java, де ми і пишемо логіку роботи нашої програми. Вони називаються «Activity» і за замовчуванням при створенні такого класу в ньому створюється метод onCreate.[11] Цей метод є основним, з якого починається створення «Activity» і де викликаються усі функції та ініціалізуються усі змінні, як взяті із бази даних, так і елементи UI. Так само в цій теці зберігаються і інші програмні класи, різні адаптери для роботи з UI або базою даних та моделі об'єктів. Також тут є пакет тестування – для написання методів та класів для тестування додатку.

У папці «res» (від слова – «resources») – лежать файли, що відносяться до ресурсів і усі вони розподілені по своїх теках: drawable, font, layout, mipmap, values, xml.[12] Доступ до всіх елементів всередині цих тек здійснюється за шаблоном «@назва_теки/назва_файлу». Розглянемо деякі з них.

Тека «drawable» або ресурс із можливістю малювання – це загальна концепція графіки, яку можна намалювати на екрані та яку можна отримати за допомогою API. Існує кілька різних типів малюнків, котрі підтримує Android Studio, але для створення додатку потрібні будуть лише файли растрового зображення із розширенням .jpg або .png.

Тека «font» – визначає спеціальні шрифти, які можна використовувати у програмі. Шрифти можуть бути окремими файлами або колекцією файлів, відомих, як «сімейство шрифтів» («font-family») і визначених у XML.

Тека «layout» – визначає архітектуру користувацького інтерфейсу для «Activity» або компоненти UI. Саме тут зберігаються файли із розширенням .xml та компоненти з яких вони складаються.

Тека «values» зберігає файл colors.xml та дві стандартні папки – strings та themes. Файл colors.xml – містить значення кольору у шістнадцятиричній (hexadecimal) колірній системі. Тека «themes» – містить файли із світлою/денною та темною/нічною темою додатка. Саме тут визначаються кольори для окремих UI елементів, а система стилів визначає зовнішній вигляд UI моделі при тій, чи іншій активованій темі. Така система дозволяє відокремити дизайн від структури та поведінки інтерфейсу. Тека «strings» – містить текстові рядки для програми з додатковим стилем і форматуванням тексту. Кожному рядку надається ім'я, через яке його можна викликати програмно у java або у xml файлах. Також саме тут міститься локалізація – переклад рядків для UI/UX.

Існує три типи рядків:

- String – один рядок;
- String Array – масив рядків;
- Quantity Strings (Plural) – різні рядки для множини.

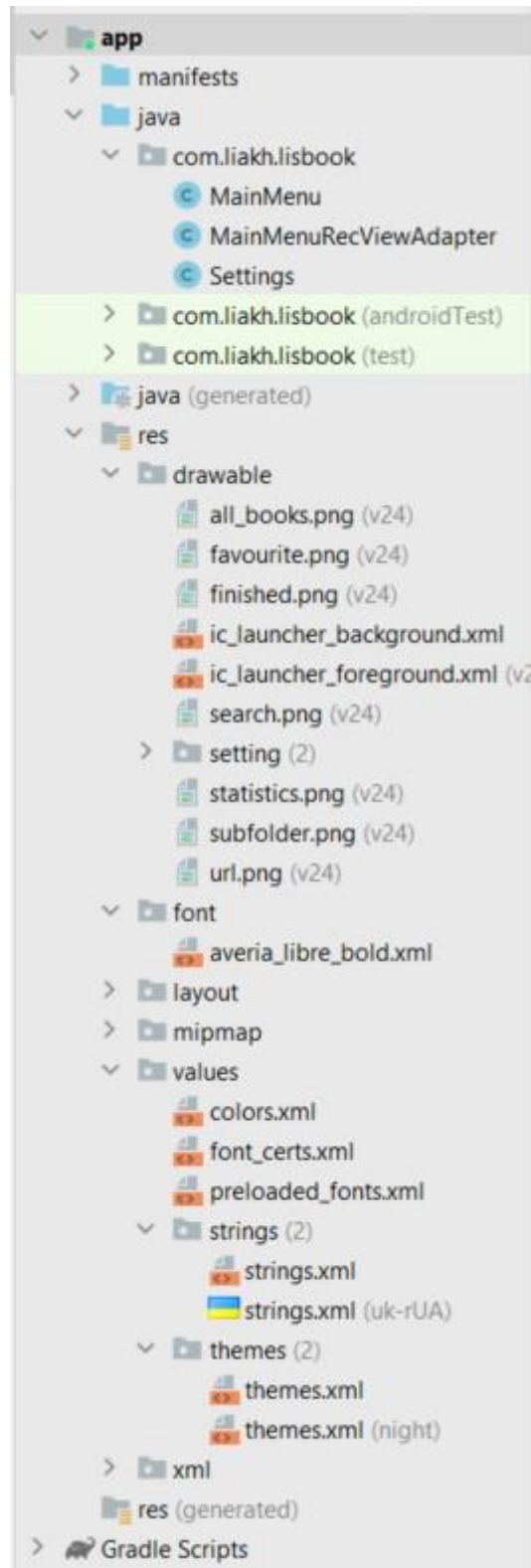


Рисунок 2.1 Структура проекту в інтеграційному середовищі розробки
Android Studio

2.4 SQLite – компактна СУБД

SQLite – це стабільна кросплатформна бібліотека із відкритим вихідним кодом написана мовою C, що реалізує невелику, швидку, функціональну, самодостатню реляційну систему баз даних на основі структурованої мови запитів SQL.[13] SQLite є найбільш використовуваним механізмом баз даних у світі.

API, необхідні для роботи із локальною базою даних на ОС Android, доступні в пакеті `android.database.sqlite`.

Така бібліотека підходить для функціонального використання і збереження локальної користувацької бази даних, що не буде займати багато місця на пристрої.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

З проведеного аналізу та огляду аналогів можна сформувавши основні функціональні вимоги до мобільного додатку персональної бібліотеки «LisBook». Функціональні вимоги – описують поведінку системи та внутрішню роботу і основні функції з якими взаємодіє кожен користувач системи.[14]

Функціональні вимоги додатку:

- Користувач повинен мати можливість створити «книгу» із усіма її параметрами, а саме: назва, категорія, посилання, нотатки;

- Користувач повинен мати можливість побачити деталі книги, а саме: назву, тег, категорію, перейти за посиланням, дата створення, нотатки, рейтинг;

- Користувач повинен мати можливість змінити дані, а саме: назву, тег, категорію, посилання, нотатки та рейтинг;

- Користувач повинен мати можливість видалити книгу;

- Користувач повинен мати можливість перейти за посиланням, яке він зберіг;

- Користувацький інтерфейс має відображатись згідно мови користувацького пристрою: англійська, українська;

- Користувач повинен мати можливість перемикає теми: світлу, темну;

- Користувач повинен мати можливість переглянути дані за одним із параметрів: тег – «Завершено», за категоріями, улюблені – якщо рейтинг рівний 5;

- Користувач повинен мати можливість перегляду усіх записів з бази даних;

- Користувач повинен мати можливість видалити усі записи з бази даних;

- Система повинна показувати користувачу рекламу при ввімкненому інтернет з'єднанні;

- Користувач повинен мати можливість експорту бази даних на свій пристрій.

3.1 Діаграма прецедентів

Використовуючи UML, зі сформованих функціональних вимог можемо створити Діаграму варіантів використання.[15] Система надає актору (користувачу) можливі сценарії використання:

- CRUD-операції над книгами (назва, категорія, тег, посилання, дата створення, нотатки, рейтинг);

- можливість зміни мови (англійська, українська);

- можливість зміни теми (світла, темна);

- розподілення за параметрами;

- очищення бази даних;

- експорт бази даних (формату .csv).

В свою чергу від системи очікується, що вона буде показувати актору (користувачу) рекламу і він зможе перейти за її посиланнями, натиснувши на неї.

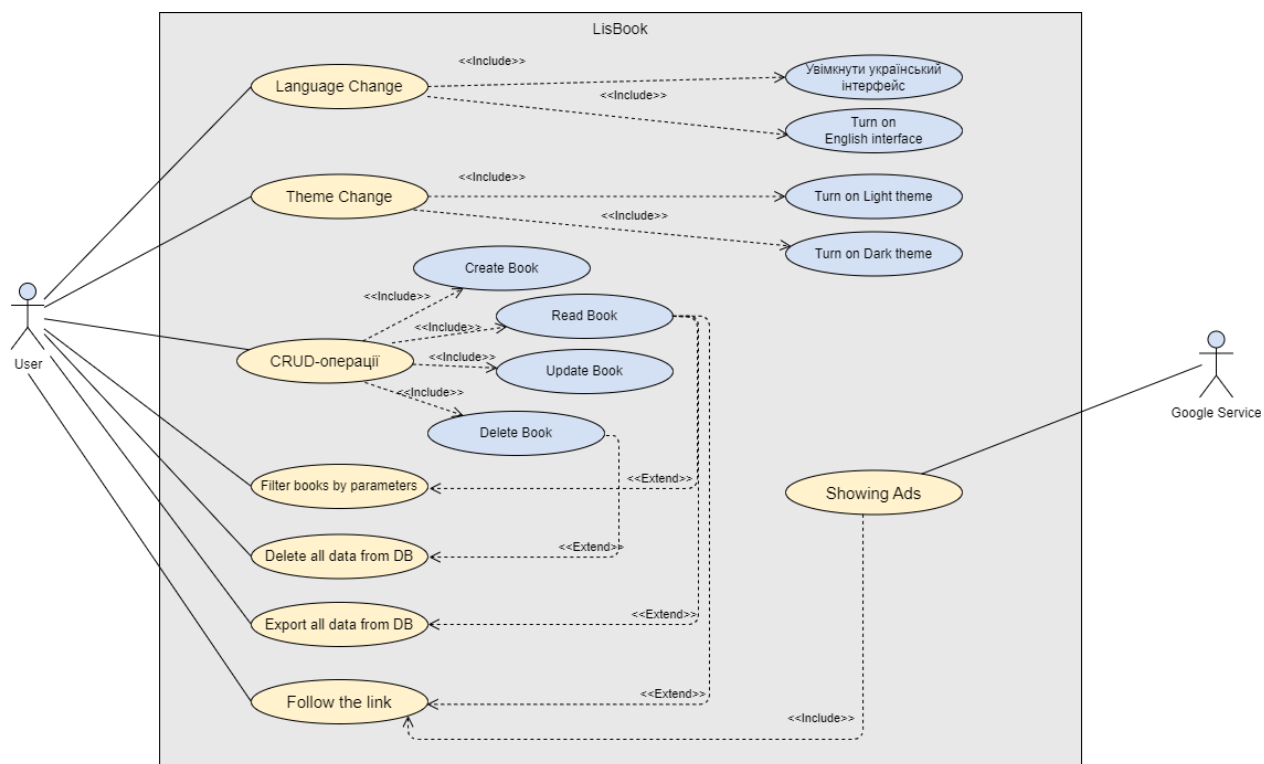


Рисунок 3.1 Use case діаграма

3.2 Діаграма діяльності

Діаграма діяльності прецеденту експорт бази даних описує дії, які необхідно виконати для запису даних із бази у файл формату csv на пристрій користувача. Спочатку необхідно перевірити чи наданий дозвіл для керування пам'яттю. Якщо дозвіл не надано необхідно запитати його в користувача, інакше запитати про підтвердження про експорт даних і перейти до його безпосереднього виконання. Якщо користувач відхиляє дозвіл його необхідно попередити про неможливість експорту бази даних та закрити діалогове вікно. Користувач зможе в будь-який момент дати дозвіл системно через налаштування пристрою або ж додаток знову спитає його про це при новій спробі експортувати дані.

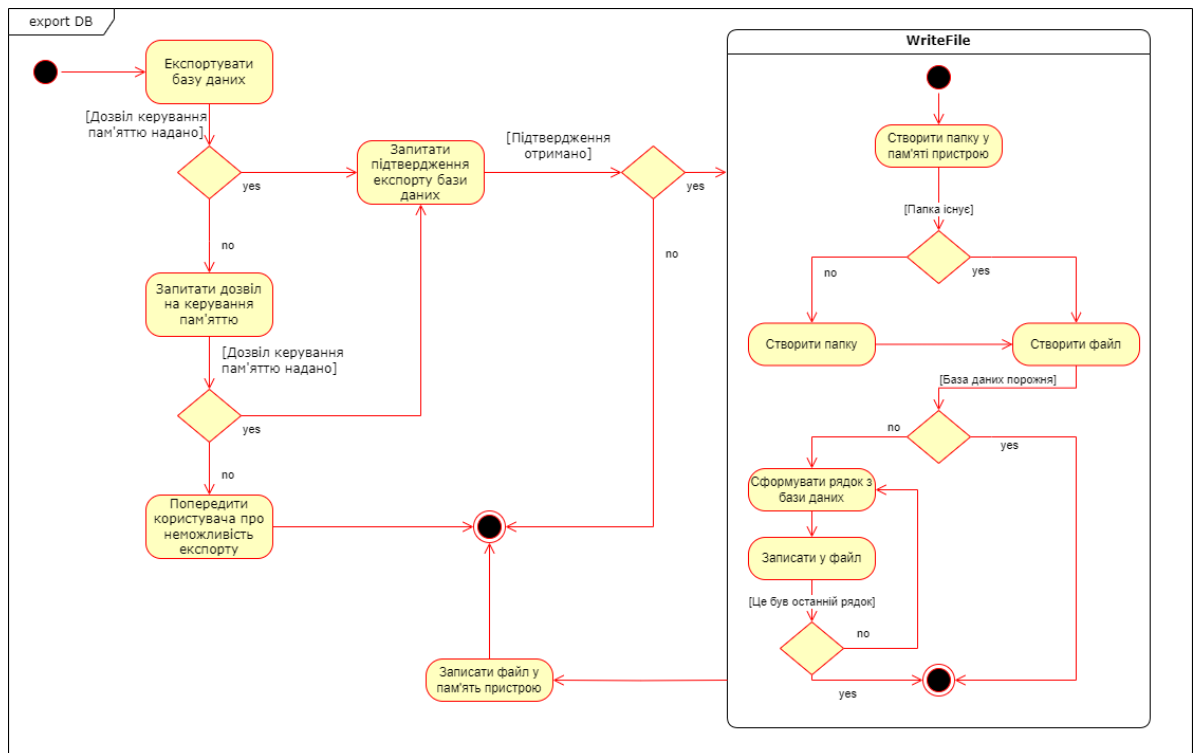


Рисунок 3.2 Діаграма діяльності для прецеденту експорт бази даних

3.3 Діаграма пакетів

При створенні основного елементу Activity окрім програмного класу java до нього створюється xml файл, що є відображенням інтерфейсу із яким безпосередньо взаємодіє користувач. Ці файли знаходяться в основному пакеті «res» (resources) у теці «layout» і викликаються програмно з відповідного ім класу з методу onCreate. Зв'язок між цими пакетами – композиція. Без програмного класу – клас відображення не буде ініціалізований і показаний користувачу.

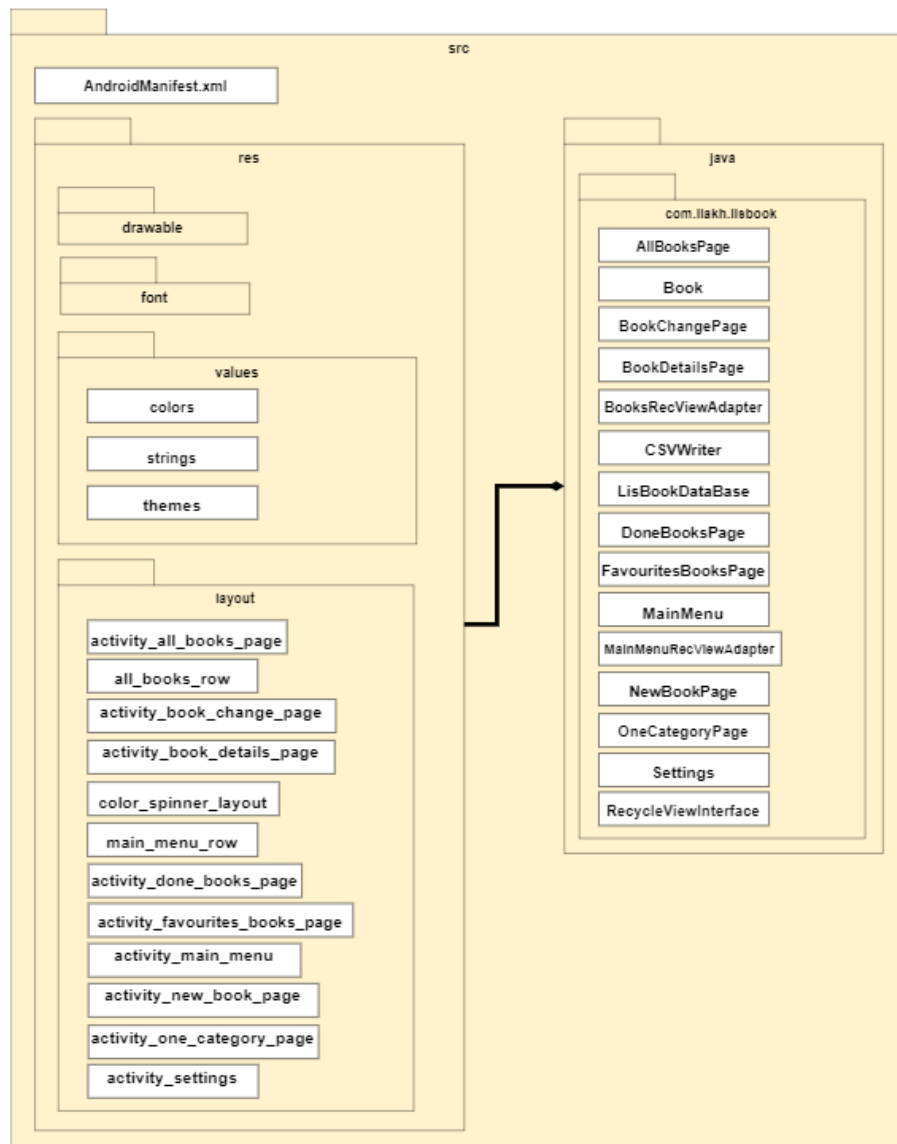


Рисунок 3.3 Діаграма пакетів

В свою чергу між теками в пакеті «res» також є свої зв'язки. Файли теки «layout» композиційно зв'язані із теками «values» та «drawable». З їхнього вмісту беруться значення для відтворення інтерфейсу. А тека «font» є не обов'язковою і створюється у випадку, коли розробник використовує шрифти із сторонніх ресурсів, тому вона утворює зв'язок – агрегації.

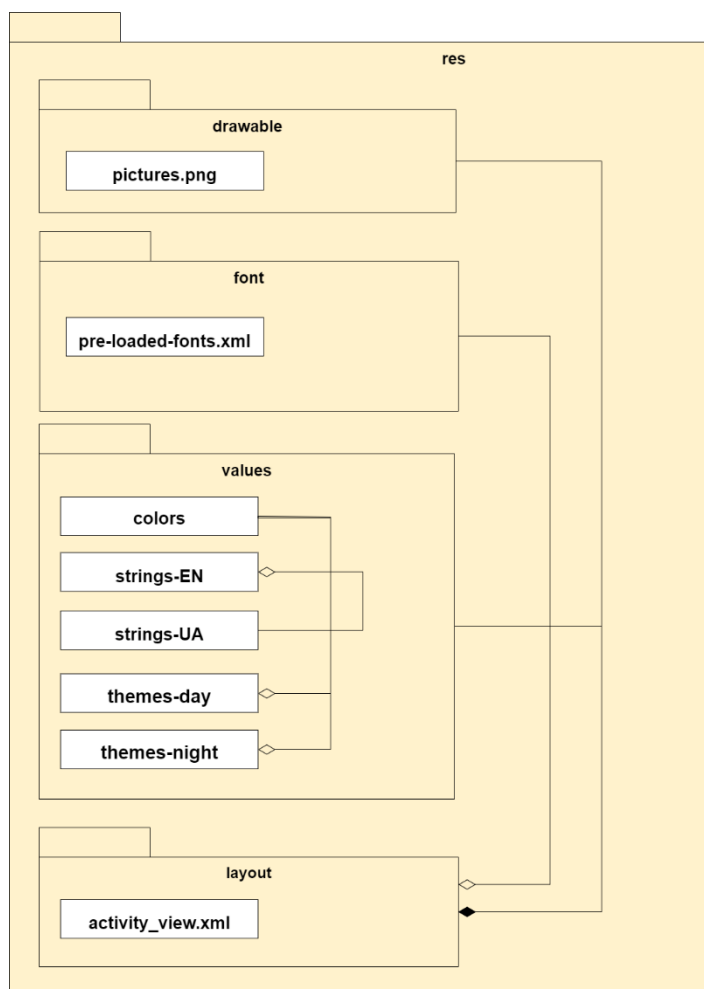


Рисунок 3.4 Деталізована частина діаграми пакетів – пакету «res»

3.4 Діаграма класів

Розглянемо детальніше діаграму класів пакету «com.liakh.lisbook». Класи `MainMenu.java`, `CategoriesPage.java`, `AllBooksPage.java`, `DoneBooksPage.java`, `FavouritesBooksPage.java` та `OneCategoryPage.java` реалізують в собі метод із інтерфейсу `RecyclerViewInterface.java`. Отже їхній зв'язок – реалізація.

Клас-частина `MainMenuRecyclerViewAdapter.java` та його контейнери `MainMenu.java` і `CategoriesPage.java` – утворюють зв'язок агрегації.

Клас-частина `BooksRecyclerViewAdapter.java` та його контейнери `AllBooksPage.java`, `DoneBooksPage.java`, `FavouritesBooksPage.java` і `OneCategoryPage.java` – також утворюють зв'язок агрегації.

Клас `LisBookDataBase.java` є композитом для моделі `Book.java`, і є класом контейнером для `NewBookPage.java`, `BookDetailsPage.java`, `BookChangePage.java`, `Settings.java` та `CSVWriter.java` – класи-частини. [16]

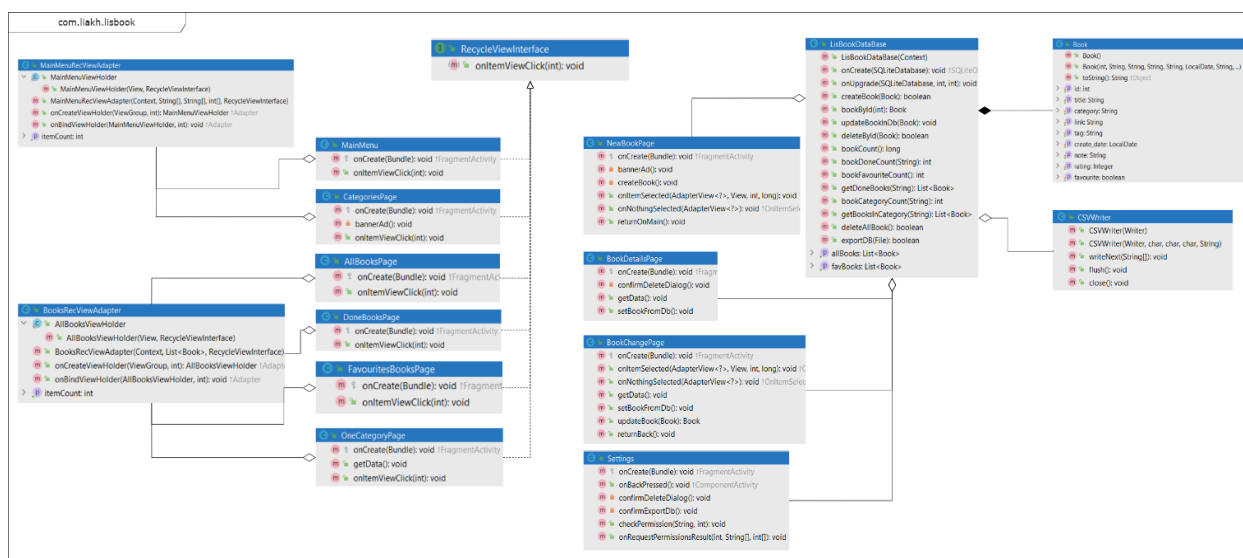


Рисунок 3.5 Діаграма класів пакету «com.liakh.lisbook»

3.5 Модель бази даних

База даних наповнюється і складається з моделі класу `Book` з типами даних і полями: `INTEGER PRIMARY KEY AUTOINCREMENT` - `_id`, `TEXT` – `title`, `TEXT` – `category`, `TEXT` – `link`, `TEXT` – `tag`, `TEXT` – `create_date`, `TEXT` – `note`, `INTEGER` – `rating`.

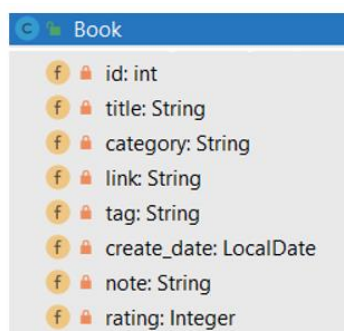


Рисунок 3.6 Модель бази даних

3.6 Проектування користувацького інтерфейсу

При проектуванні користувацького інтерфейсу у сервісі Figma слід враховувати положення екрану в якому додаток буде працювати. Для LisBook – це вертикальна орієнтація екрану.

Оскільки нам потрібно дві теми – має бути дві палітри кольорів: світлі – для світлої, темені – для темної. При виборі слід враховувати, що текст має бути зворотній: чорний чи сірий – для світлої, білий – для темної.

Згідно останніх світових трендів мінімалізм є фаворитом – монохромність, простота, однаковість форм та обмеження строкатості.

Додавання картинок, що інтуїтивно підкажуть користувачу для чого та чи інша кнопка, або що означає той, чи інший напис.

Написи на кнопках мають відповідати їхнім функціями або ж посиланням на які вони переадресовують.

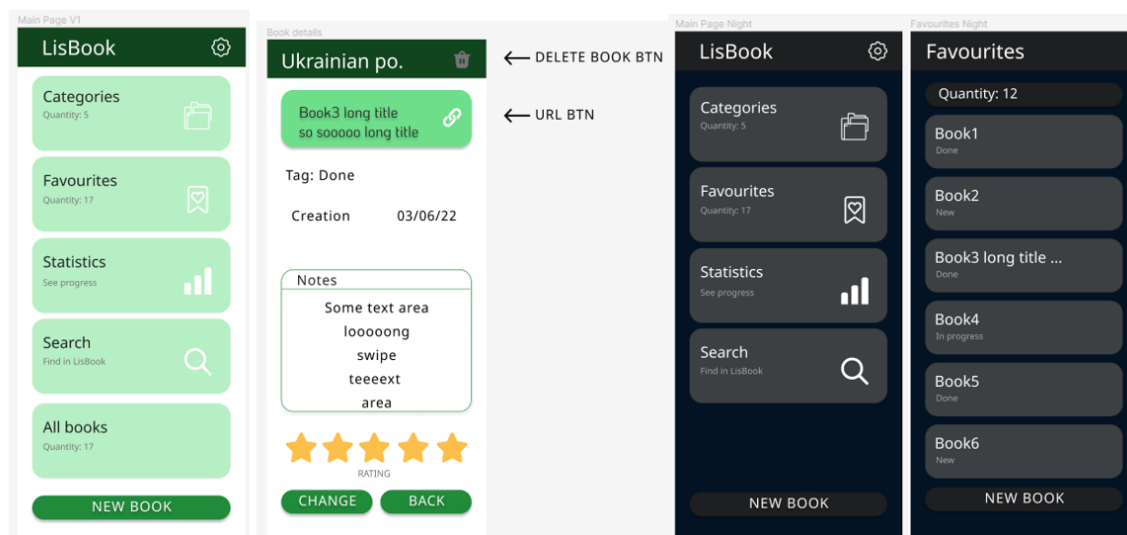


Рисунок 3.7 Приклади екранних форм додатку LisBook розроблених в сервісі Figma

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

Для реалізації поставленої мети потрібно створити екранні форми та їхню програмну частину, а також створити базу даних із можливістю запису усіх даних в окремий файл на пристрої користувача.

4.1 Створення frontend частини додатку

Розробку почнемо із користувацького інтерфейсу. Для цього потрібно: визначити кольори, рядки (написи) англійською та українською мовами, створити світлу та темну теми, створити усі Activity та їхні елементи інтерфейсу.

4.1.1 Ініціалізація кольорів

Оскільки кольори були обрані при проектуванні інтерфейсу потрібно перенести їх у файл «colors.xml» у шістнадцятковій системі. Викликати їхні значення потрібно за схемою: @color/name, де ім'я – визначене розробником. Усього в проєкті п'ятнадцять кольорів:



```
3 ■ <color name="dark_green">#10451D</color>
4 ■ <color name="light_green">#208B3A</color>
5 ■ <color name="dark_turquoise">#6EDE8A</color>
6 ■ <color name="turquoise">#B7EFC5</color>
7 ■ <color name="dark_grey">#1E1F20</color>
8 ■ <color name="light_grey">#747575</color>
9 ■ <color name="yellow">#FFBA08</color>
10 ■ <color name="stars">#FCBF49</color>
11 ■ <color name="blue">#168AAD</color>
12 ■ <color name="black">#FF000000</color>
13 ■ <color name="white">#FFFFFF</color>
14 ■ <color name="deep_blue">#041324</color>
15 ■ <color name="dark_grey_night">#3D4042</color>
16 ■ <color name="light_grey_night">#A8A8A8</color>
17 ■ <color name="red">#EE082E</color>
```

Рисунок 4.1 Файл із кольорами

4.1.2 Визначення рядків та локалізація

У файлі «string.xml» визначаємо увесь текст, який буде використовуватись в додатку – саме цей файл призначений для англійської мови і завантажується за замовчуванням, якщо інших мов не визначено. Ім'я для рядка розробник визначає самостійно. Також в цьому файлі формується і масиви рядків (string-array) із вже доданого тексту.

Для проекту нам потрібно чотири масиви рядків: для головного тексту головного меню, для маленького тексту головного меню, для випадаючого списку тегів, для меню категорій.

Рядки головного меню: категорії, улюблені, статистика, пошук, усі книжки, завершені.

Рядки тегів: нова, в процесі, завершено.

Рядки категорій: немає, фантастика, детективи, вітчизняна література, зарубіжна література, технічна література, придбати.

Визначивши всі рядки англійською мовою, тепер створюємо файл з такою ж самою назвою для української мови, але його параметром локалізації буде – uk-rUA. За допомогою вбудованого засобу Translations Editor робимо переклад рядків українською. У чекбокс-полі «Untranslatable» рядок з назвою додатку «LisBook» відмічаємо галочкою – це означає, що назва додатку не потребує перекладу.

Тепер мова додатку буде змінюватись в залежності від системних налаштувань пристрою користувача.

Key	Resource Folder	Untranslatable	Default Value	Ukrainian (uk) in Ukraine (UA)
app_name	app/src/main/res	<input checked="" type="checkbox"/>	LisBook	LisBook
categories	app/src/main/res	<input type="checkbox"/>	Categories	Категорії
cat_item_none	app/src/main/res	<input type="checkbox"/>	None	Немає
cat_item_fantasy	app/src/main/res	<input type="checkbox"/>	Fantasy	Фантастика
cat_item_detectives	app/src/main/res	<input type="checkbox"/>	Detectives	Детективи
cat_item_technical	app/src/main/res	<input type="checkbox"/>	Technical literature	Технічна література
cat_item_to_buy	app/src/main/res	<input type="checkbox"/>	To buy	Придбати
fav	app/src/main/res	<input type="checkbox"/>	Favourites	Улюблені
settings	app/src/main/res	<input type="checkbox"/>	Settings	Налаштування
statistics	app/src/main/res	<input type="checkbox"/>	Statistics	Статистика
search	app/src/main/res	<input type="checkbox"/>	Search	Пошук
all_b	app/src/main/res	<input type="checkbox"/>	All Books	Усі книжки
new_b	app/src/main/res	<input type="checkbox"/>	New Book	Нова Книга
new_c	app/src/main/res	<input type="checkbox"/>	New Category	Нова Категорія
change	app/src/main/res	<input type="checkbox"/>	Change	Змінити

Рисунок 4.2 Приклад перекладу рядків

4.1.3 Створення тем та стилів

Для того, щоб була можливість персоналізувати такі елементи, як кнопки, задній фон, підказки, форми та інше потрібно визначати для них стилі.

В середині файлу «themes.xml» спочатку перевизначаємо головні та акцентні кольори за замовчуванням із палітри, яку створили раніше. Потім в додаткових тегах «style» визначаємо ім'я для елемента на якому цей стиль буде доданий. В середині тегу визначаємо параметри: колір заднього фону, колір тексту та його розмір, кольори підказок та їхню прозорість, тощо. Це будуть стилі для світлої теми.

Щоб створити стилі для темної теми потрібно перенести вже створені стилі у файл з таким самим ім'ям - «themes.xml», але його параметр буде – «night». Тепер в кожному стилі перевизначаємо кольори для темної теми.

```

values\themes.xml
1 <resources xmlns:tools="http://schemas.android.com/tools">
2 <!-- Base application theme. -->
3 <style name="Theme.LisBook" parent="Theme.MaterialComponents.Light.NoActionBar">
4 <!-- Primary brand color. -->
5 <item name="colorPrimary">@color/dark_green</item>
6 <item name="colorPrimaryVariant">@color/light_green</item>
7 <item name="colorOnPrimary">@color/white</item>
8 <!-- Secondary brand color. -->
9 <item name="colorSecondary">@color/dark_turquoise</item>
10 <item name="colorSecondaryVariant">@color/turquoise</item>
11 <item name="colorOnSecondary">@color/black</item>
12 <!-- Status bar color. -->
13 <item name="android:statusBarColor">?attr/colorPrimary</item>
14 <!-- Customize your theme here. -->
15 <item name="colorAccent">@color/light_grey</item>
16 <item name="colorPrimaryDark">@color/dark_grey</item>
17 </style>
resources

night\themes.xml
1 <resources xmlns:tools="http://schemas.android.com/tools">
2 <!-- Base application theme. -->
3 <style name="Theme.LisBookt" parent="ThemeOverlay.MaterialComponents.Dark">
4 <!-- Primary brand color. -->
5 <item name="colorPrimary">@color/dark_grey</item>
6 <item name="colorPrimaryVariant">@color/light_grey</item>
7 <item name="colorOnPrimary">@color/black</item>
8 <!-- Secondary brand color. -->
9 <item name="colorSecondary">@color/dark_grey_night</item>
10 <item name="colorSecondaryVariant">@color/light_grey_night</item>
11 <item name="colorOnSecondary">@color/deep_blue</item>
12 <!-- Customize your theme here. -->
13 <item name="colorAccent">@color/white</item>
14 </style>

```

Рисунок 4.3 Файли із темами

4.1.4 Заповнення папки drawable

В теку drawable копіюємо заздалегідь намальовані растрові зображення формату png для головного меню, меню категорій, іконки налаштування, видалення та посилання.

4.1.5 Створення Activity та їхніх елементів інтерфейсу

На основі шаблону «Empty Activity» починаємо створювати файли з якими буде взаємодіяти користувач. Це файли для показу всіх книг, головного меню, створення книги, перегляд деталей книги, зміни даних, налаштування, перегляду завершених книг, улюблених книг, меню категорій та перегляду книг з однієї категорії.

Для кожної екранної форми буде використовуватись `ConstraintLayout` контейнер, який дає можливість створення масштабованого і гнучкого графічного інтерфейсу.[17] Його властивості або якорі, які дозволяють розташовувати елементи на сторінці чи в блоках відносно одне одного: `Top_toTopOf`, `Bottom_toBottomOf`, `End_toEndOf`, `Start_toStartOf` – це основні властивості, які комбінуються між собою, наприклад: `Top_toBottomOf`. За допомогою них можемо прив'язати елемент до батьківського положення (`parent`) або відносно конкретного елемента, вказавши його унікальний `id`. Задаємо ширину відносно батька – тобто елемент буде автоматично визначати розміри екрану пристрою користувача і підлаштовуватись під них. В основний `ConstraintLayout` контейнер додаємо тег `style`, де вказуємо значення «`Screen`» – це ім'я стилю, який було створено раніше у файлі з темами. Увесь вміст буде із зовнішніми відступами (`marginHorizontal`) від країв – `40dp`. Також всередину додаємо ще один `ConstraintLayout`, `RecyclerView` та кнопку.

Новому контейнеру даємо `id` – `toolbar`, прив'язуємо його по ширині та верхньому краю до батьківського контейнеру, а висоту задаємо статичну – `60dp`, задаємо стиль – `Toolbar`. В середину додаємо `TextView` та `ImageButton`. В `TextView` задаємо текст із задалегідь визначеного файлу з рядками – назву додатку, та прив'язуємо його по правому краю до `ImageButton`. В `ImageButton` також додаємо свій стиль, а ще посилання на картинку – `setting.png`, centruємо зображення по середині та встановлюємо зовнішні та внутрішні відступи.

`RecyclerView` – це один з елементів управління, який має вигляд автоматично масштабованого списку.[18] Коли верхній елемент списку заходить за межі екрану він очищується і перестає бути видимим, при цьому з'являється нижній елемент, який і є «очищеним» елементом зверху. Це дозволяє зменшити використання ресурсів пристрою. Надаємо елементу `id`, якорі та відступи.

На кнопку додаємо стиль, текст, відступи, якорі.

За допомогою засобів `Android Studio` перемикаємо світлу тему на темну і перевіряємо чи все працює.

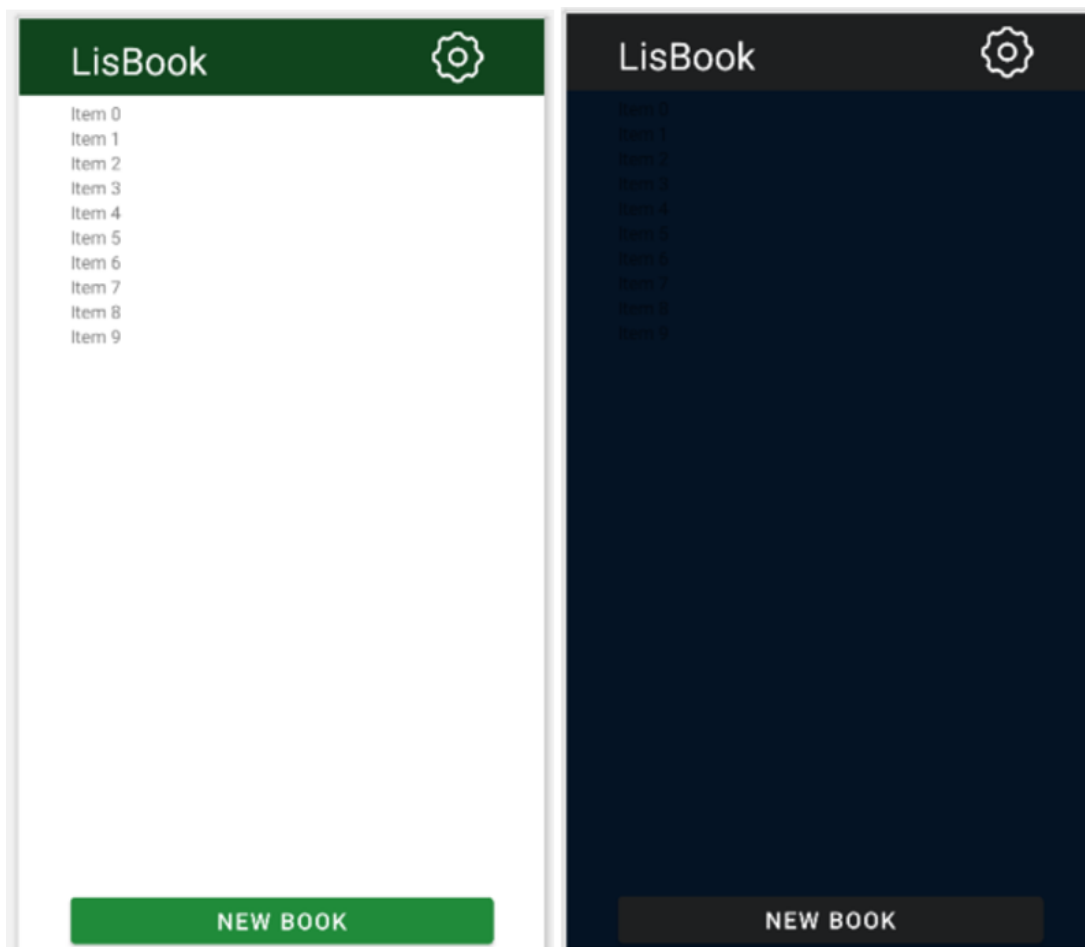


Рисунок 4.4 Основа екранної форми головного меню

Це основа для таких екранних форм: головного меню, меню категорій, сторінка усіх книг, сторінка улюблених книг, із тегом «Завершено» і за одною категорією. Міняться в середині них буде лише текст в `ToolBar` та форма списку в `RecyclerView`.

Створимо вигляд списку для `RecyclerView` для головного меню та меню категорій. Для цього створюємо новий файл – `main_menu_row.xml`. В основний `ConstraintLayout` контейнер додаємо `CardView` – карточку, яка буде відображати кожен елемент списку за заданим в ній шаблоном.[19] Додаємо в середину два `TextView` та один `ImageView`. Обгортаємо їх в ще один `ConstraintLayout` для легшого масштабування. Задаємо стилі, якорі, відступи і ставимо простий текст, щоб подивитись, як виглядає наша карточка. Цей текст

не буде відображатись користувачеві. Він буде змінений на той, який вже визначений і який задамо програмно пізніше, так само, як і картинка.

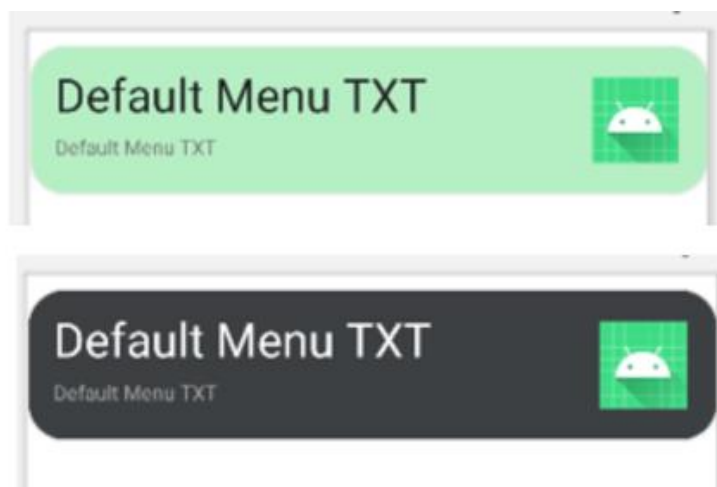


Рисунок 4.5 CardView для головного меню та меню категорій

Для сторінки усіх книг, улюблених книг, із тегом «Завершено» і за однією категорією створим такий самий CardView, тільки без картинки.

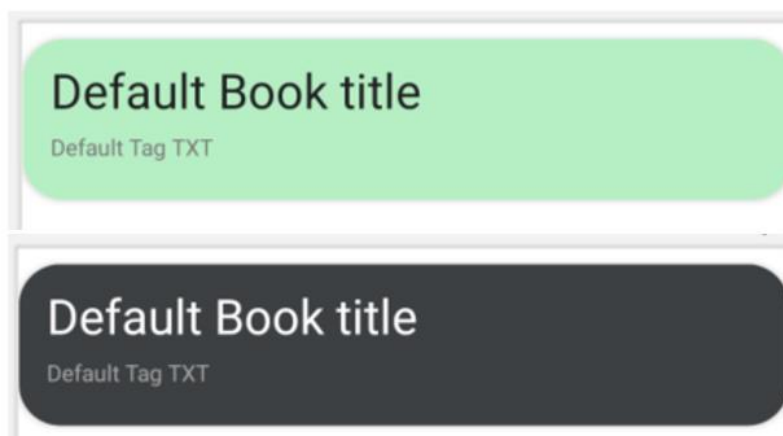


Рисунок 4.6 CardView для сторінки усіх книг, улюблених книг, із тегом «Завершено» і за однією категорією

Далі створимо сторінку для нової книги. Для цього потрібні: TextView – назва сторінки, два EditText – назва та посилання, Spinner – для категорій, ImageView – для картинки посилання, TextInputLayout з вкладеним

TextInputEditText – для нотаток, AdView – для показу реклами, та дві кнопки – «Зберегти» та «Назад».

Елементи TextInputLayout, TextInputEditText та EditText – дають можливість користувачу ввести свій текст, який після натиснення кнопки «Зберегти» буде збережений у базі даних. На всі елементи додаємо власні стилі, текст, підказки, посилання на картинку та якорі відносно одне одного.

В рекламний тег додаємо id – в цьому випадку тестовий.[26] Справжній id користувача видається при викладенні додатка на платформу Google Play. Також задаємо розмір за допомогою унікальних тегів визначених для реклами.

На нотатки вводимо обмеження у вигляді трьохсот символів: параметр android:maxLength="300" не дасть користувачу ввести більше заданої кількості символів, вмикаємо лічильник та додаємо підказку про обмеження і дозволяємо прокрутку вниз та багаторядковий ввід.

Для елементу Spinner (випадаючий список) потрібно створити власне відображення.[20] Створюємо файл color_spinner_layout.xml в якому визначаємо відступи, розміри тексту та головне – стилі, щоб текст було видно при переключенні на темну тему.

Перевіряємо вигляд у двох темах:

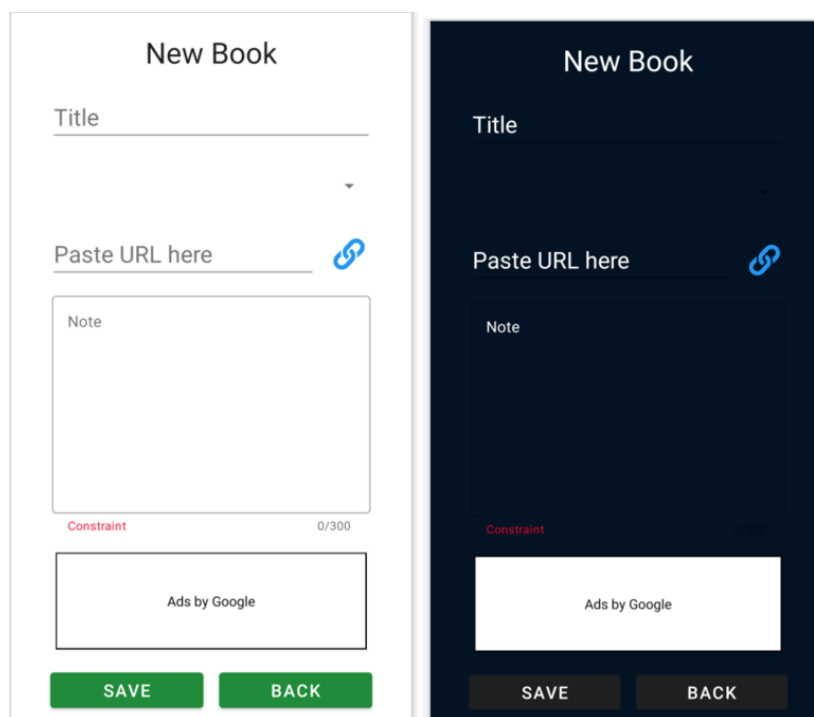


Рисунок 4.7 Екранна форма сторінки «Нова книга»

Сторінка для перегляду деталей книги виглядає схожим чином, але додається дата створення, обрана категорія в текст у Toolbar, кнопка видалити, кнопка посилання, тег та рейтинг, а також кнопка «Змінити». Блокуємо можливість вводу тексту в нотатки. В текстових елементах виставляємо два додаткові параметри: `android:maxLines`, `android:ellipsize="end"`. Перший параметр вказує скільки рядків буде виводитись на екран, а другий вказує на те, як рядок буде закінчуватись і у цьому випадку трьома крапками. Це буде вказувати користувачу на те, що рядок довгий і в ньому є продовження.

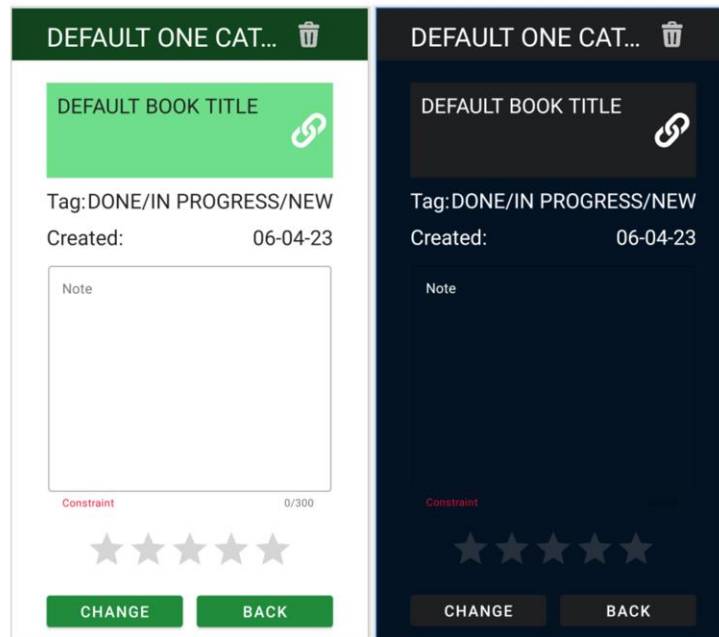


Рисунок 4.8 Екранна форма сторінки «Деталі книги»

Деякі з полів книги при її створенні будуть виставлені за замовчуванням: дата, тег, рейтинг. Щоб це можна було змінити створюємо іншу сторінку. Змінити можна тег, рейтинг, категорію, посилання та нотатки.

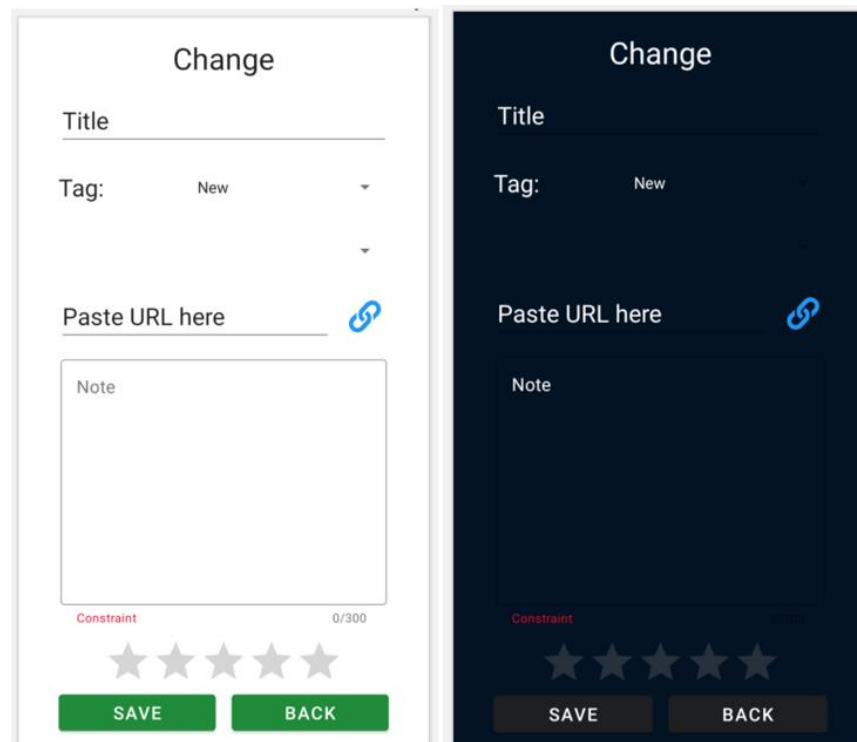


Рисунок 4.9 Екранна форма «Змінити книгу»

Для можливості перемикання теми, очищення та експорту бази даних у csv файл – створюємо сторінку налаштувань. Створюємо елемент перемикання (Switcher), кнопки експорту та очищення, до кожного додаємо відповідний їм текст у TextView.

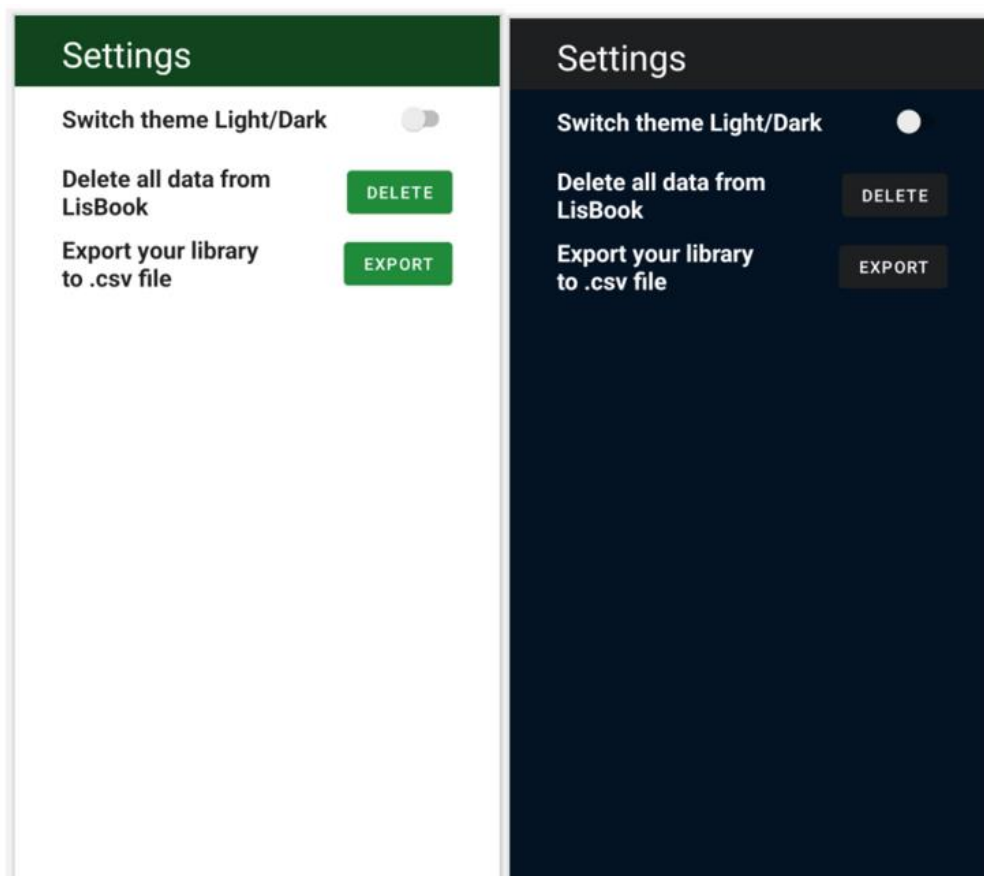


Рисунок 4.10 Екранна форма сторінки «Налаштування»

4.2 Створення бази даних

База даних SQLite зберігається локально у теці «database» з усіма файлами додатку. Типи даних, які використовує SQLite: INTEGER – цілі числа, REAL – з плаваючою комою, TEXT – текст не обмежений кількістю символів, BLOB – для збереження зображень, аудіо та відео.

4.2.1 Модель Book

Для того щоб працювати з даними користувача потрібно створити модель або сутність (entity) за допомогою полів і методів якої можна передавати дані в базу даних і витягати їх з неї. Така структура забезпечує високу гнучкість та зрозумілість програми, адже кожна модель, по суті, є об'єктом із реального світу, який легко описати.

Отже, створюємо клас Book із приватними полями: id, title, category, link, tag, create_date, note, rating. Для доступу до приватних полів з інших класів та пакетів нам необхідно створити методи Getter і Setter. В методі setRating робимо перевірку на вхідні данні: якщо значення від 0 до 5 включно, то встановлюємо вхідне значення, в іншому випадку – встановлюємо 0. Також потрібні два конструктори: один порожній, інший з усіма полями. Гарною практикою є перевизначення метода toString, що виводить інформацію в текстовому вигляді.

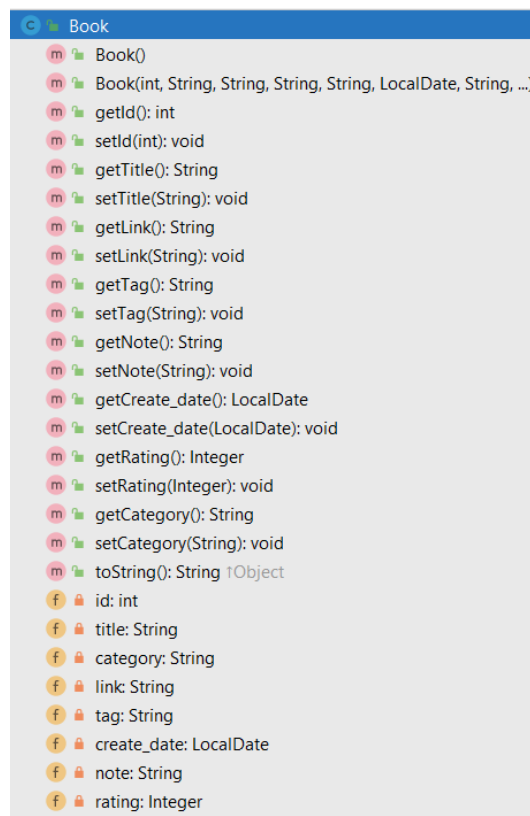


Рисунок 4.11 Модель Book

4.2.2 Адаптер для бази даних

Для створення бази даних в Android Studio потрібно розширити створений клас LisBookDataBase базовим – SQLiteOpenHelper класом та перевизначити реалізацію методів onCreate та onUpgrade.

Метод onCreate – основний метод, який створює базу даних при першому відкритті додатку. Тут рядком-запитом прописані колонки, їхні назва та типи даних, які вони приймають. Вкінці на екземплярі бази даних викликається вбудований метод execSQL - куди передається рядок створення бази даних для її створення.[21]

```
@Override
public void onCreate(SQLiteDatabase db) {
    String query = "CREATE TABLE " + TABLE_NAME + " (" + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        COLUMN_TITLE + " TEXT, " +
        COLUMN_CATEGORY + " TEXT, " +
        COLUMN_LINK + " TEXT, " +
        COLUMN_TAG + " TEXT, " +
        COLUMN_CR_DATE + " TEXT, " +
        COLUMN_NOTE + " TEXT, " +
        COLUMN_RATING + " REAL);";
    db.execSQL(query);
}
```

Рисунок 4.12 Метод створення бази даних

Метод onUpgrade – це метод для оновлення структури бази даних. Приймає версію задану розробником і нову базу даних.

Власний метод createBook – приймає на вхід модель, відкриває можливість запису даних у базу, виставляє параметри моделі відповідно до колонок і повертає 1 – якщо успішно, 0 – якщо помилка.

Власний метод bookById – приймає на вхід Id книги, відкриває можливість читання з бази даних, знаходить Id, який співпадає з вхідним і виставляє усі значення полів з бази у відповідні змінні моделі та повертає її.

Власний метод getAllBooks – повертає усі записи з бази даних, формуючи з них список.

Власний метод `updateBookInDb` - приймає на вхід модель, відкриває можливість запису в базу даних, знаходить `Id`, який співпадає з вхідним з моделі і виставляє усі значення полів з бази на нові взяті з моделі, зберігає нові значення.

Власний метод `deleteById` - приймає на вхід `Id` книги, відкриває можливість зміни записів у базі даних, знаходить `Id`, який співпадає з вхідним і видаляє запис, повертає 1 – якщо успішно, 0 – якщо помилка.

Власні методи підрахунку `allBookCount`, `doneBookCount`, `bookFavouriteCount`, `bookCategoryCount` відповідно рахують записи у базі за параметрами: всі записи, кількість книг із тегом «Завершено», всі книги із рейтингом 5, кількість книг у відповідній категорії.

Методи `getBooksInCategory`, `getFavouriteBooks`, `getDoneBooks` – повертають усі книги відповідно до категорії, із рейтингом 5, із тегом «Завершено».

Метод `deleteAllBook` – видаляє усі записи з бази даних.

4.2.3 Запис даних у csv файл

Метод `exportDB` приймає на вхід файл, та використовує клас `CSVWriter`, щоб сформувати потрібний вигляд із користувацькими даними.

Клас `CSVWriter` містить в собі розділювач – кому, дужки відкриття, дужки закриття. Використовуючи клас `StringBuffer`, рядок формується в буфері, а потім записується у файл і знову формується новий рядок. Такий спосіб економить ресурси пристрою і є швидким та ефективним, ніж сформувати увесь вміст в буфері, а потім записати його у файл.

При використанні класів, що працюють із базою даних – записують та читають дані, їх обов'язково треба закривати вкінці кожного методу.

4.3 Програмування backend-частини

Спочатку створюємо інтерфейс `RecyclerViewInterface` із методом `onItemClickListener`, який приймає на вхід позицію у списку. Цей метод буде призначений для переходу на задану сторінку зі списку або на сторінку із деталями книги за позицією у списку.

Також в кожному класі знаходимо за унікальними `id` елементи інтерфейсу: кнопки, `RecyclerView`, перемикачі, випадаючі списки, тощо. Ініціалізуємо їх у змінні.

4.3.1 `RecyclerViewAdapter`

`RecyclerViewAdapter` потрібен для виводу даних у тому вигляді, який хоче розробник. Він обробляє та зв'язує вхідні дані зі списком та виглядом, відслідковує зміни. В класі є три методи: `onCreateViewHolder` – повертає об'єкт, який буде зберігати дані по одній моделі (`CardView`), `onBindViewHolder` – виконує прив'язку об'єкта `ViewHolder` до моделі по позиції, `getItemCount` – повертає кількість об'єктів в списку.[22]

Також він має вкладений клас `ViewHolder` на основі якого визначаються елементи інтерфейсу в одній моделі – це створені в попередніх кроках `CardView`.

Для проекту знадобиться два таких `RecyclerViewAdapter` – для головного меню і меню категорій та сторінок з відображення списку книг. Головна відмінність – передані `CardView` та їхні елементи. Для головного меню та меню категорій використовується масив картинок і створений файл `main_menu_row`, а для списку з книг – ні і файл `all_books_row`.

Також встановлюємо на кожному `CardView` слухач події – `setOnClickListener`, де при натисканні на елемент списку передаємо його позицію в створений раніше метод інтерфейсу.

```

public class MainMenuRecyclerViewAdapter extends RecyclerView.Adapter<MainMenuRecyclerViewAdapter.MainMenuViewHolder> {

    private final RecyclerViewInterface recycleViewInterface;
    String[] default_main_menu, small_title_main_menu;
    int[] default_images;
    Context context;

    public MainMenuRecyclerViewAdapter(Context ct, String[] d_m_m, String[] s_t_m_m, int[] img,
                                       RecyclerViewInterface recycleViewInterface) {

        this.context = ct;
        this.default_main_menu = d_m_m;
        this.small_title_main_menu = s_t_m_m;
        this.default_images = img;
        this.recycleViewInterface = recycleViewInterface;
    }

    @NonNull
    @Override
    public MainMenuViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(R.layout.main_menu_row, parent, attachToRoot false);
        return new MainMenuViewHolder(view, recycleViewInterface);
    }

    @Override
    public void onBindViewHolder(@NonNull MainMenuViewHolder holder, int position) {
        holder.default_main_menu_name.setText(default_main_menu[position]);
        holder.small_title_main_menu.setText(small_title_main_menu[position]);
        holder.default_img_main_menu.setImageResource(default_images[position]);
    }

    @Override
    public int getItemCount() { return default_main_menu.length; }
}

```

Рисунок 4.13 Основні методи класу MainMenuRecyclerViewAdapter

4.3.2 Створення меню та сторінок із показом списку книг за одним параметром

Попередньо проініціалізувавши усі елементи інтерфейсу, починаємо заповнювати масив для картинок, які будуть викликатись відповідно їхніх назв, а записуватись, як індекс: R.drawable.subfolder, R.drawable.favourite, R.drawable.statistics, R.drawable.search, R.drawable.all_books, R.drawable.finished.

Заповнюємо два масиви рядків з файлу з рядками для назв в головному меню та для підписів. Важливо, щоб рядки були розташовані в тому ж порядку, що і картинки, щоб співпадати із назвами.

Для підписів створюємо нові рядки: через цикл викликаємо методи підрахунку кількості книг в базі даних, яку ініціалізували раніше.

Після цього створюємо та встановлюємо адаптер: контекст – поточна Activity, заголовки, підзаголовки, картинки, RecyclerViewInterface.

Перевизначаємо метод з інтерфейсу: при натисненні на один з пунктів меню буде відкриватись нове вікно відповідно до позиції. Важливо викликати функцію закриття попередньої Activity, щоб не використовувати багато ресурсів та оперативної пам'яті.

Виставляємо на кнопки слухачі подій. При натисненні на них будуть відкриватись нові сторінки: кнопка settings – налаштування, newBookBtn – сторінка створення книги. Важливо що спочатку вказується сторінка з якої здійснюється перехід, а потім на яку.

Все те саме робимо для класу CategoriesPage. Завдяки відділенню логіки одного класу MainMenuRecyclerViewAdapter від основних класів, його можна використовувати з різних частин програми.

Для сторінок показу списку книг за параметром створюємо подібний адаптер, але без масиву картинок. Також в адаптер передаємо список книг витягнутих з бази за одним із параметрів: за тегом «Завершено», рейтингом 5, за однією категорією або усі книги.

```

@Override
public void onItemClick(int position) {
    Intent intent;
    switch (position) {
        case 0:
            intent = new Intent( packageContext: MainMenu.this, CategoriesPage.class);
            startActivity(intent);
            finish();
            break;
        case 1:
            intent = new Intent( packageContext: MainMenu.this, FavouritesBooksPage.class);
            startActivity(intent);
            finish();
            break;
        case 4:
            intent = new Intent( packageContext: MainMenu.this, AllBooksPage.class);
            startActivity(intent);
            finish();
            break;
        case 5:
            intent = new Intent( packageContext: MainMenu.this, DoneBooksPage.class);
            startActivity(intent);
            finish();
            break;
    }
}

```

Рисунок 4.14 Перевизначений метод onItemClick для пунктів

ГОЛОВНОГО МЕНЮ

4.3.3 CRUD-сторінки

Спочатку для кожної сторінки `NewBookPage` (create), `BookDetailsPage` (read), `BookChangePage` (update) – ініціалізуємо елементи інтерфейсу.

`NewBookPage` – сторінка, на якій дані користувача зберігаються в модель і передаються в базу даних. Важливим тут є перевірка на ввід назви: якщо введено порожній рядок при натисканні на кнопку «Створити» спливе підказка про помилку. В разі правильності введення назви, в модель додаються усі параметри задані користувачем. Тег, дата, рейтинг виставляються за замовчуванням.

```

createBook.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    public void onClick(View v) {
        String tmp = titleInput.getText().toString().trim();
        if(tmp != null && !tmp.equals("")) {
            createBook();
        } else {
            Toast.makeText(context: NewBookPage.this, "Error creating!", Toast.LENGTH_SHORT).show();
        }
    }
});

```

Рисунок 4.15 Кнопка створення книги та перевірка на правильність вводу даних

Обов'язково виставляємо створений шар випадяючого списку для категорій та ставимо відповідний масив рядків і слухач події за позицією у списку.

Створюємо метод показу реклами, який викликаємо вгорі при створенні сторінки. Виставляємо кнопку повернення назад – `backBtn`.

При правильному вводі даних користувачеві буде показана підказка про успішність додавання книжки до бібліотеки.

Дані створеної книги він може подивитись натиснувши на неї і запустивши сторінку `BookDetailsPage`. На цій сторінці вже відображається повна інформація про книгу з раніше встановленими значеннями за замовчуванням. Натиснувши на іконку з посиланням, користувач переходить

у браузер за посиланням, яке він вказав при створенні книги. Також на цій сторінці користувач може видалити книгу. Щоб видалити книгу потрібно натиснути на кнопку із картинкою корзини і підтвердити свій вибір у діалоговому вікні.

```
private void confirmDeleteDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
    builder.setTitle("Delete");
    builder.setMessage("Are you sure you want to delete" + " " + book.getTitle() + "?");
    builder.setPositiveButton("YES", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            try {
                dataBase.deleteById(book);
                Intent intent = new Intent( packageContext: BookDetailsPage.this, AllBooksPage.class);
                Toast.makeText( context: BookDetailsPage.this, "Successfully deleted!", Toast.LENGTH_SHORT).show();
                startActivity(intent);
                finish();
            } catch (Exception e) {
                Toast.makeText( context: BookDetailsPage.this, "Error deleting!", Toast.LENGTH_SHORT).show();
                Log.e( tag: "error", e.getMessage());
            }
        }
    });
    builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.dismiss();
        }
    });
    builder.create();
    builder.show();
}
```

Рисунок 4.16 Функція виклику діалогу для видалення книги

Також на цій сторінці є кнопка «Змінити», яка веде на сторінку, де можна оновити дані про книгу.

На сторінці BookChangePage відображається рейтинг-бар у вигляді зірочок, випадаючі списки з категоріями та тегами, введені нотатки та назва, посилання.

З попередньої сторінки ми отримали id книги і виставили її значення. При натисканні кнопки «Змінити» існуючі дані замінюються новими і записуються в базу даних або залишаються такими самими, якщо користувач повернувся назад. Тут важливо викликати функції в правильному порядку: отримати дані та встановити на сторінці, взяти дані зі сторінки та замінити на нові в базі.

4.3.4 Сторінка налаштувань

На сторінці налаштувань ініціалізуємо перемикач (Switch). За допомогою нього користувач зможе перемикати теми світлу та темну. При запуску сторінки перевіряємо тему телефону користувача і відштовхуємось від неї у слухачі подій на переключення теми.

Ініціалізуємо кнопку «Видалити» при натисканні якої буде відкриватись діалогове вікно запиту на видалення усіх даних з бібліотеки.[23] Перевіряємо користувацьку відповідь: так – видаляємо, ні – закриваємо діалогове вікно.

Ініціалізуємо кнопку «Експорт» при натисканні якої спочатку відкривається діалог для запиту на дозвіл до пам'яті пристрою користувача. Якщо користувач не дає дозвіл – діалогове вікно закривається. Якщо користувач погоджується надати дозвіл – система може створити файл, запитати чи дійсно користувач хоче експортувати дані і перевірити його відповідь. Тоді, якщо папка ще не існує, створити теку з вказаним ім'ям «LisBook», створити файл «lisbook.csv» та викликати метод з бази даних. Дозвіл для доступу до пам'яті потрібно надавати один раз. Якщо користувач системно вимкне дозвіл, додаток запитає його знову.

```

private void confirmExportDb() {
    AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
    builder.setTitle("Export");
    builder.setMessage("Are you sure you want to export your library to .csv?");
    builder.setPositiveButton("YES", new DialogInterface.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.R)
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            boolean isSuccess;
            try {
                File exportDir = new File(Environment.getExternalStorageDirectory(), child: "LisBook");
                if (!exportDir.exists()) {
                    exportDir.mkdirs();
                }
                File file = new File(exportDir, child: "lisbook.csv");
                isSuccess = dataBase.exportDB(file);
                if (isSuccess) {
                    file.createNewFile();
                    Toast.makeText( context: Settings.this, "Successfully exported!", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: Settings.this, MainMenu.class);
                    startActivity(intent);
                    finish();
                }
            } catch (Exception e) {
                Toast.makeText( context: Settings.this, "Error exporting!", Toast.LENGTH_SHORT).show();
                Log.e( tag: "error", e.getMessage());
            }
        }
    });

    builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.dismiss();
        }
    });
    builder.create();
    builder.show();
}
}

```

Рисунок 4.17 Функція діалогового вікна для експорту бази даних

4.4 AndroidManifest файл

Це основний програмний файл з якого починається збірка додатку. У файлі AndroidManifest треба задекларувати основні дозволи, назву додатку, іконку, мета-дані реклами, теми, версію SDK та усі Activity.

В тегу «Application» визначаємо іконку, назву додатку та основну тему.

Для кожної Activity визначаємо орієнтацію екрану – портретну, щоб екранні форми не перевертались. Для NewBookPage та BookChangePage виставляємо параметр `android:windowSoftInputMode="adjustPan"`, що дозволить клавіатурі з'являтися понад кнопками. MainMenu визначаємо як LAUNCHER – тобто головна екранна форма з якої починається запуск додатку.

Даємо дозволи на перевірку інтернет з'єднання, запис в пам'ять пристрою та керування пам'яттю.[24] До всіх тегів додаємо прибирання максимальної версії SDK, щоб на всіх пристроях запити працювали коректно.

В тезі з мета-даними визначаємо тестові значення реклами від сервісів Google. А у файлі build.gradle для модуля LisBook в залежності додаємо рядок, який підключає модулі реклами. Тепер, якщо в користувача увімкнене інтернет з'єднання йому буде показуватись реклама.[25]

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
  android:name="android.permission.WRITE_EXTERNAL_STORAGE"
  tools:remove="android:maxSdkVersion" />
<uses-permission
  android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
  tools:remove="android:maxSdkVersion" />

<application
  android:allowBackup="true"
  android:dataExtractionRules="@xml/data_extraction_rules"
  android:fullBackupContent="@xml/backup_rules"
  android:icon="@drawable/bookstack"
  android:label="LisBook"
  android:roundIcon="@drawable/bookstack"
  android:supportsRtl="true"
  android:theme="@style/Theme.LisBook"
  tools:targetApi="31">
  <activity
    android:name=".FavouritesBooksPage"
    android:exported="false"
    android:screenOrientation="portrait" />

  <meta-data
    android:name="com.google.android.gms.ads.APPLICATION_ID"
    android:value="ca-app-pub-3940256099942544~3347511713" />

  <activity
    android:name=".OneCategoryPage"
    android:exported="false"
    android:screenOrientation="portrait" />
  <activity
    android:name=".DoneBooksPage"
    android:exported="false"
    android:screenOrientation="portrait" />
  <activity
    android:name=".AllBooksPage"
    android:exported="false"
    android:screenOrientation="portrait" />

```

Рисунок 4.18 Файл маніфесту

4.5 Тестування

Щоб перевірити якість створеного програмного забезпечення і головне – відповідність вимогам, на кожному проекті обов'язково проводять тестування продукту. Під час тестування програмного продукту можна визначити головні проблеми, знайти помилки і усунути їх до того, як продукт буде представлений клієнту та випущений на ринок.

Тести можуть бути отримані з вимог і специфікацій, дизайнерських артефактів або вихідного коду. Різні рівні тестування супроводжують кожну окрему діяльність з розробки ПЗ:

- Acceptance testing – оцінка програмного забезпечення відповідно до вимог;

- Системне тестування – оцінка програмного забезпечення з огляду на архітектурний дизайн;

- Інтеграційне тестування – оцінка програмного забезпечення з огляду на дизайн підсистем;

- Module Testing – оцінка програмного забезпечення з огляду на детальний дизайн;

- Unit Testing – оцінка програмного забезпечення з огляду на реалізацію.

Black-box тестування – це отримання тестів та результатів лише із описів програмного забезпечення, специфікацій, вимог та дизайну. Тестувальник не має інформації про внутрішню архітектуру та роботу програмного забезпечення. Тестування «чорного ящика» – є тестуванням високого рівня, яке зосереджується виключно на поведінці програмного забезпечення. Це передбачає тестування з точки зору зовнішнього або кінцевого користувача. Тестування «чорного ящика» можна застосовувати практично до всіх рівнів тестування програмного забезпечення.

Параметри Black-box тестування:

- ґрунтується на основі зовнішніх очікувань, коли внутрішня поведінка програми невідома;

- перевірка функціональних можливостей системи, які тестуються;
- заснований на методі проб і помилок;
- менш вичерпний і трудомісткий;
- добре підходить та є ефективним для об'ємних сегментів коду.

White-box тестування – це отримання тестів та результатів із вихідного коду, внутрішнього програмного забезпечення, зокрема, включаючи гілки, окремі умови та структури продукту. Тестування White-Box вважається тестуванням низького рівня. Його також називають скляною коробкою або прозорою коробкою, або тестуванням бази коду. Метод тестування «білого ящика» передбачає, що шлях логіки в модулі чи програмі відомий.[27]

Параметри White-box тестування:

- внутрішня структура заздалегідь відома тестувальнику;
- найкраще підходить для нижчого рівня тестування, наприклад модульного тестування, інтеграційного тестування;
- легко автоматизувати;
- можна розпочати після підготовки робочої документації;
- вичерпний і трудомісткий метод;
- дозволяє видалити зайві рядки коду, які можуть нести приховані дефекти.

ВИСНОВКИ

В результаті виконання дипломної роботи було розроблено мобільний додаток персональної бібліотеки LisBook мовою програмування Java. Актуальність теми зумовлена необхідністю україномовного додатку для спрощення обліку книг. В ході досягнення поставлено мети було:

1. Проаналізовано та обрано технічні засоби, що використовувались для розробки мобільного додатку: Figma – для прототипування інтерфейсу, Java, Android Studio, SQLite – для гнучкої розробки під ОС Android;

2. Розроблено вимоги для додатку на основі проведеного аналізу переваг та недоліків існуючих додатків та встановлено, що головними недоліками є відсутність української мови, можливості зберегти посилання, експорту бази даних;

3. Спроектовано та розроблено додаток на основі аналізу потреб користувачів та визначених вимог. Основними є CRUD-операції над книгами, реалізація україномовного інтерфейсу та двох тем, збереження посилання, експорт бази даних на пристрій користувача;

4. Пройдено апробацію на науково-технічних конференціях;

5. Проведено тестування мобільного додатку методами White-box та Black-box.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Читання в Україні [Електронний ресурс] – 2018р. Режим доступу до ресурсу: <http://data.chytomo.com/chytannya-v-ukrayini/>.
2. Turn - Reading Tracker & Timer [Електронний ресурс]. Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=com.jackwradford.readingtracker>
3. Goodreads [Електронний ресурс]. Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=com.goodreads>
4. My Reading Record [Електронний ресурс]. Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=myrr.auto1.myreadingrecord1>
5. Figma [Електронний ресурс]. Режим доступу до ресурсу: <https://www.figma.com/about/>
6. What is Java [Електронний ресурс]. Режим доступу до ресурсу: https://www.java.com/en/download/help/whatis_java.html
7. Android Studio [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/about>
8. XML Declaring layout [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/develop/ui/views/layout/declaring-layout>
9. Dependency management [Електронний ресурс]. Режим доступу до ресурсу: https://docs.gradle.org/current/userguide/dependency_management.html
10. Manifest [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/guide/topics/manifest/manifest-intro>
11. Activity [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/reference/android/app/Activity>
12. Providing Resources [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/guide/topics/resources/providing-resources>
13. SQLite [Електронний ресурс]. Режим доступу до ресурсу: <https://sqlite.org/index.html>

14. Що таке функціональні вимоги: приклади, визначення, повний посібник. Visure Solutions. URL: <https://visuresolutions.com/uk/blog/functional-requirements/>

15. What is Use Case Diagram?. Visual Paradigm. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

16. What is Class Diagram?. Visual Paradigm. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

17. Build a responsive UI with ConstraintLayout. [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/develop/ui/views/layout/constraint-layout>

18. RecyclerView. [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/jetpack/androidx/releases/recyclerview>

19. Create a card-based layout [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/develop/ui/views/layout/cardview>

20. Spinners [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/develop/ui/views/components/spinner>

21. SQLiteOpenHelper [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper>

22. RecyclerView.Adapter [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView.Adapter>

23. AlertDialog [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/reference/android/app/AlertDialog>

24. Request runtime permissions [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/training/permissions/requesting>

25. Add ads to your instant app [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.android.com/topic/google-play-instant/guides/advertising>

26. Enabling test ads [Электронный ресурс]. Режим доступа до ресурсу:
<https://developers.google.com/admob/android/test-ads>

27. Differences between Black Box Testing vs White Box Testing [Электронный ресурс]. Режим доступа до ресурсу:
<https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



«Розробка мобільного додатку LisBook мовою Java»

Виконала студентка 4 курсу

Групи ПД-41

Лях Ольга Андріївна

Керівник роботи

К.т.н, доцент, завідувач кафедри ІПЗ Негоденко Олена Василівна

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи - покращення процесу обліку книг та створення персональної бібліотеки за допомогою мобільного додатку, розробленого мовою програмування JAVA.

Об'єкт дослідження – процес обліку книг в персональній бібліотеці.

Предмет дослідження – мобільний додаток для обліку книг та ведення персональної бібліотеки.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз існуючих мобільних додатків та обрати інтегроване середовище розробки Android Studio для програмування мовою Java.
2. Створити візуалізацію роботи додатка за допомогою засобів UML.
3. Спроекувати модель користувацького user-friendly інтерфейсу в онлайн-сервісі Figma.
4. Побудувати структуру бази даних на основі SQLite.
5. Розробити мобільний додаток для ведення персональної цифрової бібліотеки, що спрощує облік книг користувача за допомогою обраних засобів.
6. Провести мануальне тестування додатку.

3

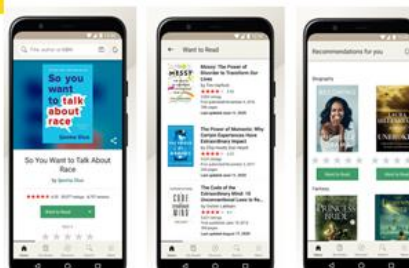
АНАЛІЗ АНАЛОГІВ



Приклади екранних форм додатку
«Turn - Reading Tracker & Timer»



Приклади екранних форм додатку
«My Reading Record»



Приклади екранних форм додатку
«Goodreads»

4

АНАЛІЗ АНАЛОГІВ

Назва Характеристики	Turn - Reading Tracker & Timer	Goodreads	My Reading Record	LisBook
Зміна теми (світла/темна)	+	-	+	+
Збереження посилання на книгу	-	-	-	+
Україномовний інтерфейс	-	-	-	+
Експорт бази даних	+	-	-	+
Розділення за категоріями	+	+	+	+
Ведення статистики читання	+	+	+	-

5

ВИМОГИ ДО МОБІЛЬНОГО ДОДАТКУ

Функціональні вимоги

Користувач повинен мати можливість:

1. Створити «книгу» із усіма її параметрами, а саме: назва, категорія, посилання, нотатки.
2. Побачити деталі книги, а саме: назву, тег, категорію, перейти за посиланням, дата створення, нотатки, рейтинг.
3. Змінити дані, а саме: назву, тег, категорію, посилання, нотатки та рейтинг.
4. Видалити книгу.
5. Перейти за посиланням, яке він зберіг.
6. Переглянути дані за одним із параметрів: тег – «Завершено», за категоріями, улюблені – якщо рейтинг рівний 5.
7. Видалити усі записи з бази даних.
8. Перегляду усіх записів з бази даних.
9. Перемикати теми: світлу, темну.
10. Експорту бази даних на свій пристрій у форматі csv.
11. Система повинна показувати користувачу рекламу при ввімкненому інтернет з'єднанні.

Нефункціональні вимоги

1. Користувацький інтерфейс має відобразитись згідно мови користувацького пристрою: англійська, українська.
2. Написи мають бути короткі та зрозумілі.
3. Текстові написи мають бути читабельними в двох темах.
4. Кількість реклами має бути помірною: показ на 3-4 сторінках.

6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



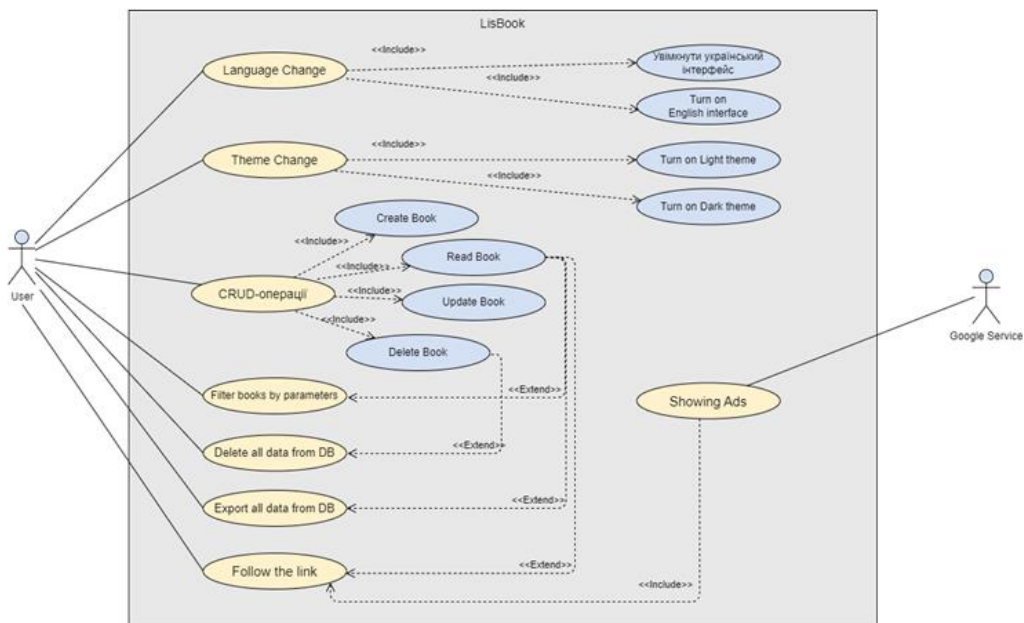
Figma

Android
Studio

SQLite

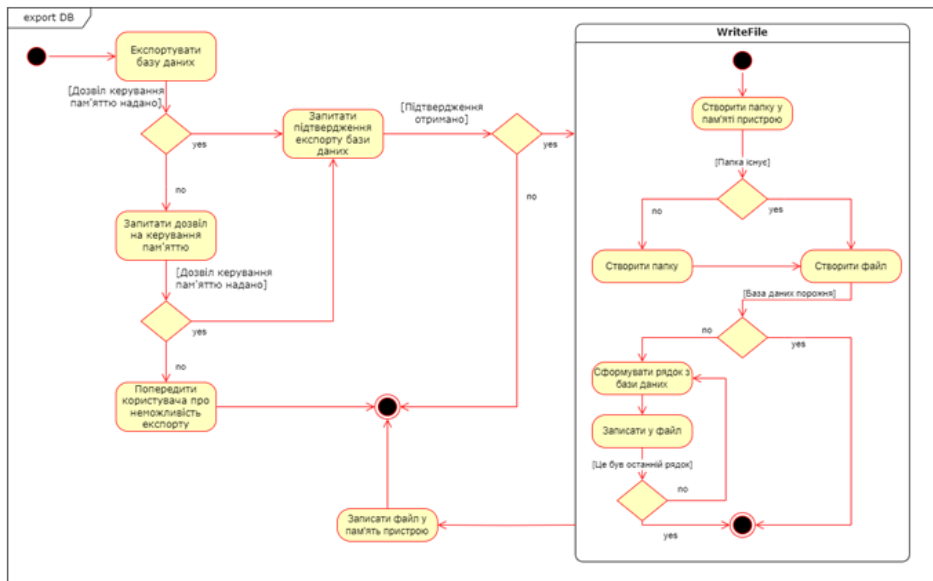
7

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



8

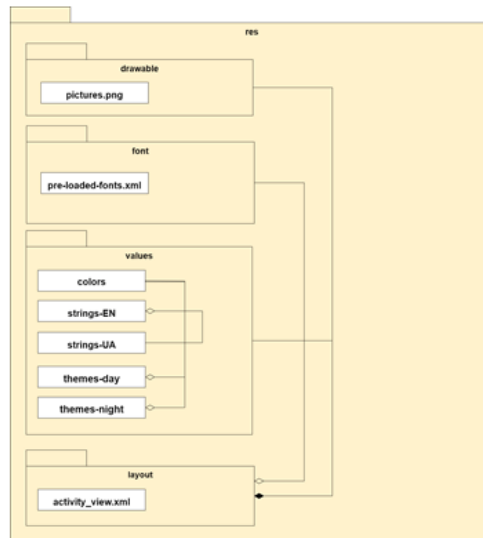
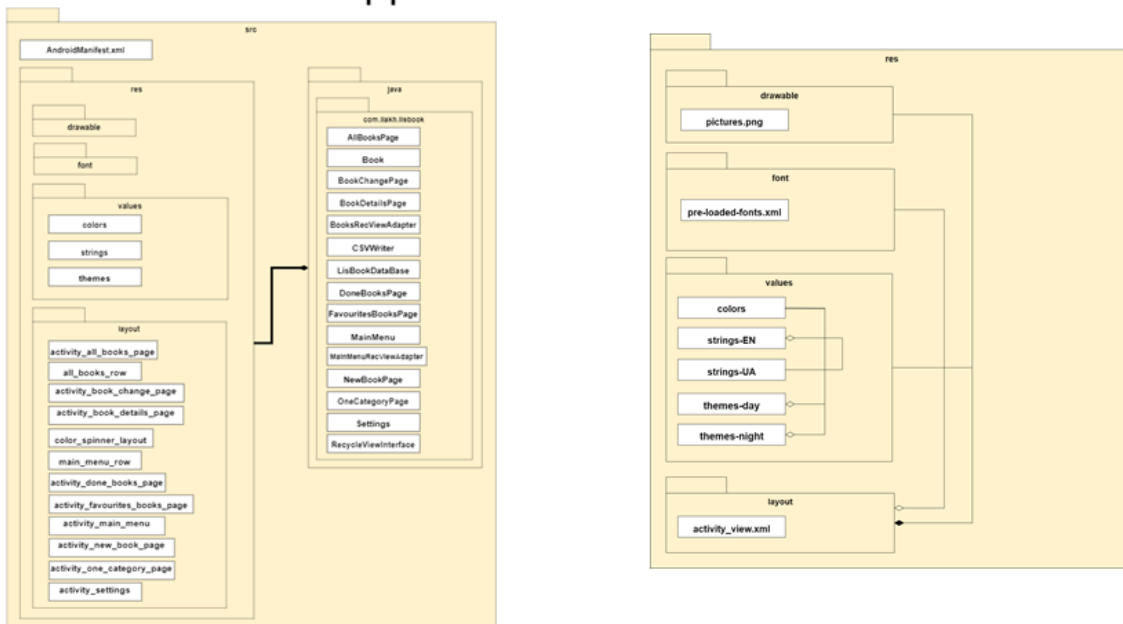
ДІАГРАМА ДІЯЛЬНОСТІ



Прецедент експорт бази даних

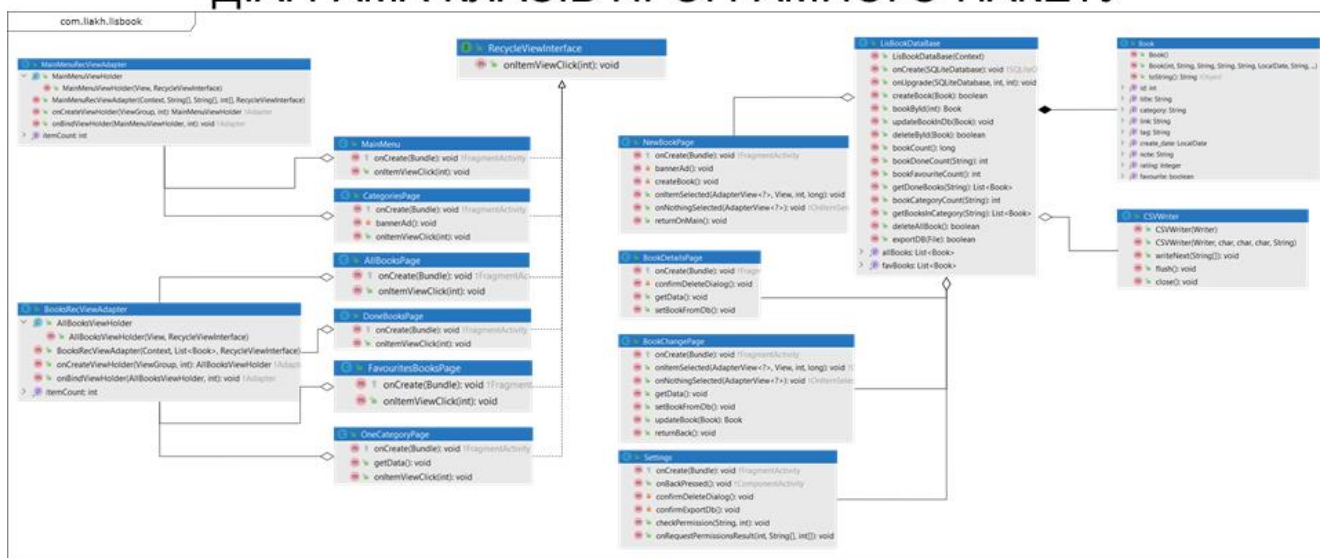
9

ДІАГРАМА ПАКЕТІВ



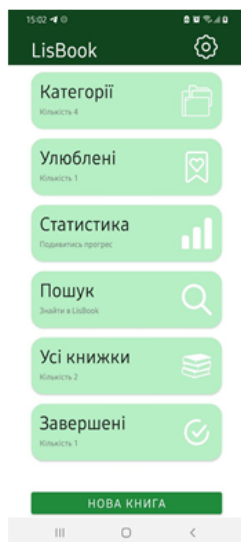
10

ДІАГРАМА КЛАСІВ ПРОГРАМНОГО ПАКЕТУ

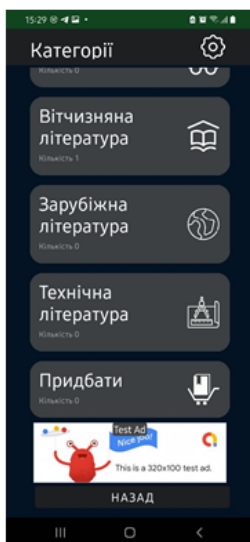


11

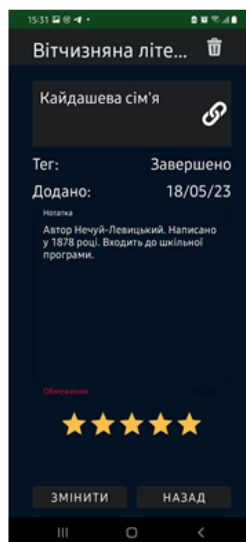
ЕКРАННІ ФОРМИ ДОДАТКУ LISBOOK



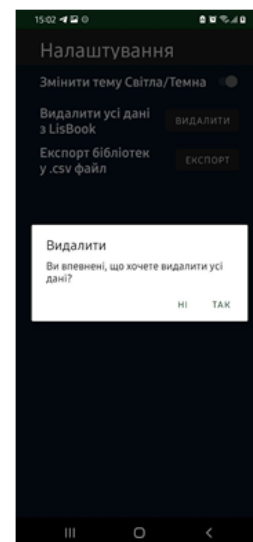
Головний екран додатку



Екран перегляду категорій



Екран перегляду книги



Вікно діалогу в налаштуваннях

12

АРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Лях О.А. English learning Applications / Andrii Ahashkov, Olga Lyakh // «Integration of New Knowledge, Research and Innovation Across Europe: матеріали I міжнародної науково-практичної конференції (м. Київ, 23-25 квітня 2020 року)». – 2020 – С. 91
2. Лях О.А Розробка мобільного додатку «LisBook» - бібліотека в смартфоні // III Всеукраїнській науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». Збірник тез. 07.04.2023, ДУТ, м. Київ – С. 108

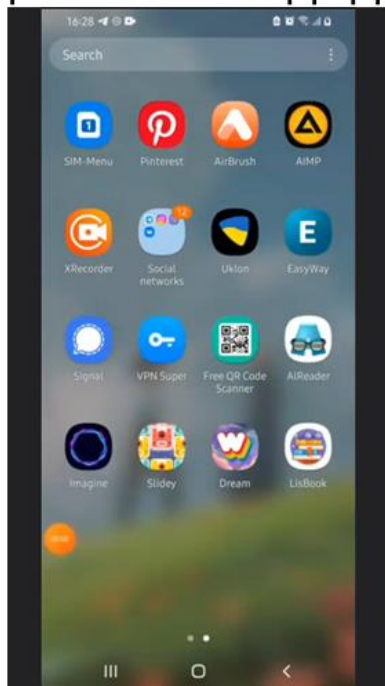
13

ВИСНОВКИ

1. Проаналізовано та обрано технічні засоби, що використовувались для розробки мобільного додатку: Figma – для прототипування інтерфейсу, Java, Android Studio, SQLite – для гнучкої розробки під ОС Android.
2. Розроблено вимоги для додатку на основі проведеного аналізу переваг та недоліків існуючих додатків та встановлено, що головними недоліками є відсутність української мови, можливості зберегти посилання, експорту бази даних.
3. Спроектовано та розроблено додаток на основі аналізу потреб користувачів та визначених вимог. Основними є CRUD-операції над книгами, реалізація україномовного інтерфейсу та двох тем, збереження посилання, експорт бази даних на пристрій користувача.
4. Пройдено апробацію на науково-технічних конференціях.
5. Проведено тестування мобільного додатку методами White-box та Black-box.

14

ВІДЕО РОБОТИ ДОДАТКУ



15

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК Б

Публікації

1. Лях О.А. English learning Applications / Andrii Ahashkov, Olga Lyakh // «Integration of New Knowledge, Research and Innovation Across Europe: матеріали І міжнародної науково-практичної конференції (м. Київ, 23-25 квітня 2020 року)». – 2020 – С. 91
2. Лях О.А Розробка мобільного додатку «LisBook» - бібліотека в смартфоні // ІІІ Всеукраїнській науково-технічній конференції «Сучасний стан та перспективи розвитку ІоТ». Збірник тез. 07.04.2023, ДУТ, м. Київ – С. 108

ДОДАТОК В

Вихідний код мобільного додатку «LisBook»

```

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

  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    tools:remove="android:maxSdkVersion" />
  <uses-permission
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:remove="android:maxSdkVersion" />

  <application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@drawable/bookstack"
    android:label="@string/app_name"
    android:roundIcon="@drawable/bookstack"
    android:supportsRtl="true"
    android:theme="@style/Theme.LisBook"
    tools:targetApi="31">
    <activity
      android:name=".FavouritesBooksPage"
      android:exported="false"
      android:screenOrientation="portrait" />

    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="ca-app-pub-3940256099942544~3347511713" />

    <activity
      android:name=".OneCategoryPage"
      android:exported="false"
      android:screenOrientation="portrait" />
    <activity
      android:name=".DoneBooksPage"
      android:exported="false"
      android:screenOrientation="portrait" />
    <activity
      android:name=".AllBooksPage"
      android:exported="false"
      android:screenOrientation="portrait" />
    <activity

```

```

        android:name=".BookDetailsPage"
        android:exported="false"
        android:screenOrientation="portrait" />
    <activity
        android:name=".BookChangePage"
        android:exported="false"
        android:screenOrientation="portrait"
        android:windowSoftInputMode="adjustPan" />
    <activity
        android:name=".NewBookPage"
        android:exported="false"
        android:screenOrientation="portrait"
        android:windowSoftInputMode="adjustPan" />
    <activity
        android:name=".CategoriesPage"
        android:exported="false"
        android:screenOrientation="portrait" />
    <activity
        android:name=".Settings"
        android:exported="false"
        android:screenOrientation="portrait" />
    <activity
        android:name=".MainMenu"
        android:exported="true"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

    <meta-data
        android:name="preloaded_fonts"
        android:resource="@array/preloaded_fonts" />
</application>

</manifest>

```

```
package com.liakh.lisbook;
```

```
import java.time.LocalDate;
```

```
public class Book {
```

```
    private int id;
```

```
    private String title;
```

```
    private String category;
```

```
    private String link;
```

```
    private String tag;
```

```
    private LocalDate create_date;
```

```
private String note;
private Integer rating;

public Book() {
}

public Book(int id, String title, String category, String link, String tag, LocalDate create_date,
String note, Integer rating) {
    this.id = id;
    this.title = title;
    this.category = category;
    this.link = link;
    this.tag = tag;
    this.note = note;
    this.create_date = create_date;
    this.rating = rating;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getLink() {
    return link;
}

public void setLink(String link) {
    this.link = link;
}

public String getTag() {
    return tag;
}

public void setTag(String tag) {
    this.tag = tag;
}

public String getNote() {
    return note;
}
```

```

    }

    public void setNote(String note) {
        this.note = note;
    }

    public LocalDate getCreate_date() {
        return create_date;
    }

    public void setCreate_date(LocalDate create_date) {
        this.create_date = create_date;
    }

    public Integer getRating() {
        return rating;
    }

    public void setRating(Integer rating) {
        if (rating <= 5 && rating >= 0) {
            this.rating = rating;
        } else {
            this.rating = 0;
        }
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    @Override
    public String toString() {
        return "Book{" +
            "id=" + id + "\" +
            ", title=" + title + "\" +
            ", link=" + link + "\" +
            ", tag=" + tag + "\" +
            ", note=" + note + "\" +
            ", create_date=" + create_date + "\" +
            ", rating=" + rating + "\" +
            '";
    }
}

package com.liakh.lisbook;

import android.content.ContentValues;
import android.content.Context;

```

```

import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Build;
import android.os.Environment;
import android.util.Log;
import android.widget.Toast;

```

```

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;

```

```

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

```

```

public class LisBookDataBase extends SQLiteOpenHelper {

```

```

    private final Context context;
    private static final String DATABASE_NAME = "LisBook.db";
    private static final int DATABASE_VERSION = 1;

```

```

    private static final String TABLE_NAME = "library";
    private static final String COLUMN_ID = "_id";
    private static final String COLUMN_TITLE = "title";
    private static final String COLUMN_LINK = "link";
    private static final String COLUMN_TAG = "tag";
    private static final String COLUMN_NOTE = "note";
    private static final String COLUMN_CR_DATE = "create_date";
    private static final String COLUMN_RATING = "rating";
    private static final String COLUMN_CATEGORY = "category";

```

```

    public LisBookDataBase(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context = context;
    }

```

```

    @Override

```

```

    public void onCreate(SQLiteDatabase db) {
        String query = "CREATE TABLE " + TABLE_NAME + " (" + COLUMN_ID + " INTEGER
PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_TITLE + " TEXT, " +
            COLUMN_CATEGORY + " TEXT, " +
            COLUMN_LINK + " TEXT, " +
            COLUMN_TAG + " TEXT, " +
            COLUMN_CR_DATE + " TEXT, " +
            COLUMN_NOTE + " TEXT, " +
            COLUMN_RATING + " REAL);";
    }

```



```

        db.execSQL(query);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME + "");
        onCreate(db);
    }

    public boolean createBook(Book book) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(COLUMN_TITLE, book.getTitle());
        cv.put(COLUMN_CATEGORY, book.getCategory());
        cv.put(COLUMN_LINK, book.getLink());
        cv.put(COLUMN_TAG, book.getTag());
        cv.put(COLUMN_CR_DATE, book.getCreate_date().toString());
        cv.put(COLUMN_NOTE, book.getNote());
        cv.put(COLUMN_RATING, book.getRating());

        long insert = db.insert(TABLE_NAME, null, cv);
        if (insert == -1) {
            return false;
        } else {
            return true;
        }
    }

    @RequiresApi(api = Build.VERSION_CODES.O)
    public Book bookById(int id) {
        String queryString = "SELECT * FROM " + TABLE_NAME + " WHERE _id = " + id;
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(queryString, null);
        Book bookFromDb = null;
        if (cursor.moveToFirst()) {
            int bookId = cursor.getInt(0);
            String bookTitle = cursor.getString(1);
            String bookCategory = cursor.getString(2);
            String bookLink = cursor.getString(3);
            String bookTag = cursor.getString(4);
            String bookDate = cursor.getString(5);
            String bookNote = cursor.getString(6);
            int bookRating = cursor.getInt(7);

            bookFromDb = new Book(bookId, bookTitle, bookCategory, bookLink, bookTag,
                LocalDate.parse(bookDate), bookNote, bookRating);
        }
        db.close();
        cursor.close();
        return bookFromDb;
    }
}

```

```

@RequiresApi(api = Build.VERSION_CODES.O)
public List<Book> getAllBooks() {
    List<Book> returnBooksList = new ArrayList<>();
    String queryString = "SELECT * FROM " + TABLE_NAME;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);

    if (cursor.moveToFirst()) {

        do {
            int bookId = cursor.getInt(0);
            String bookTitle = cursor.getString(1);
            String bookCategory = cursor.getString(2);
            String bookLink = cursor.getString(3);
            String bookTag = cursor.getString(4);
            String bookDate = cursor.getString(5);
            String bookNote = cursor.getString(6);
            int bookRating = cursor.getInt(7);

            Book book = new Book(bookId, bookTitle, bookCategory, bookLink, bookTag,
LocalDate.parse(bookDate), bookNote, bookRating);
            returnBooksList.add(book);
        } while (cursor.moveToNext());

    } else {
        //do not anything
    }
    cursor.close();
    db.close();
    return returnBooksList;
}

public void updateBookInDb(Book book) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_TITLE, book.getTitle());
    cv.put(COLUMN_LINK, book.getLink());
    cv.put(COLUMN_CATEGORY, book.getCategory());
    cv.put(COLUMN_TAG, book.getTag());
    cv.put(COLUMN_NOTE, book.getNote());
    cv.put(COLUMN_RATING, book.getRating());

    db.update(TABLE_NAME, cv, "_id=?", new String[]{String.valueOf(book.getId())});
    db.close();
}

public boolean deleteById(Book book) {
    SQLiteDatabase db = this.getWritableDatabase();
    String query = "DELETE FROM " + TABLE_NAME + " WHERE " + COLUMN_ID + " = " +
book.getId();
    Cursor cursor = db.rawQuery(query, null);

```

```

    if (cursor.moveToFirst()) {
        return true;
    } else {
        return false;
    }
}

public long allBookCount() {
    SQLiteDatabase db = this.getReadableDatabase();
    long count = DatabaseUtils.queryNumEntries(db, TABLE_NAME);
    db.close();
    return count;
}

public int bookDoneCount(String tag) {
    String queryString = "SELECT COUNT (*) FROM " + TABLE_NAME + " WHERE " +
    COLUMN_TAG + "=" + "'" + tag + "'";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);
    int count = 0;
    if (cursor != null && cursor.moveToFirst()) {
        do {
            cursor.moveToFirst();
            count = cursor.getInt(0);
        } while (cursor.moveToNext());
        cursor.close();
    }
    db.close();
    return count;
}

public int bookFavouriteCount() {
    String queryString = "SELECT COUNT (*) FROM " + TABLE_NAME + " WHERE " +
    COLUMN_RATING + "=5";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);
    int count = 0;
    if (cursor != null && cursor.moveToFirst()) {
        do {
            cursor.moveToFirst();
            count = cursor.getInt(0);
        } while (cursor.moveToNext());
        cursor.close();
    }
    db.close();
    return count;
}

@RequiresApi(api = Build.VERSION_CODES.O)
public List<Book> getDoneBooks(String tag) {
    List<Book> returnBooksList = new ArrayList<>();

```

```

String queryString = "SELECT * FROM " + TABLE_NAME + " WHERE " + COLUMN_TAG
+ "=" + "'" + tag + "'";
SQLiteDatabase db = this.getReadableDatabase();
Cursor cursor = db.rawQuery(queryString, null);

if (cursor.moveToFirst()) {

    do {
        int bookId = cursor.getInt(0);
        String bookTitle = cursor.getString(1);
        String bookCategory = cursor.getString(2);
        String bookLink = cursor.getString(3);
        String bookTag = cursor.getString(4);
        String bookDate = cursor.getString(5);
        String bookNote = cursor.getString(6);
        int bookRating = cursor.getInt(7);

        Book book = new Book(bookId, bookTitle, bookCategory, bookLink, bookTag,
        LocalDate.parse(bookDate), bookNote, bookRating);
        returnBooksList.add(book);
    } while (cursor.moveToNext());

} else {
    //do not anything
}
cursor.close();
db.close();
return returnBooksList;
}

@RequiresApi(api = Build.VERSION_CODES.O)
public List<Book> getFavBooks() {
    List<Book> returnBooksList = new ArrayList<>();
    String queryString = "SELECT * FROM " + TABLE_NAME + " WHERE " +
COLUMN_RATING + "=5";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);

    if (cursor.moveToFirst()) {

        do {
            int bookId = cursor.getInt(0);
            String bookTitle = cursor.getString(1);
            String bookCategory = cursor.getString(2);
            String bookLink = cursor.getString(3);
            String bookTag = cursor.getString(4);
            String bookDate = cursor.getString(5);
            String bookNote = cursor.getString(6);
            int bookRating = cursor.getInt(7);

            Book book = new Book(bookId, bookTitle, bookCategory, bookLink, bookTag,
            LocalDate.parse(bookDate), bookNote, bookRating);

```

```

        returnBooksList.add(book);
    } while (cursor.moveToNext());

} else {
    //do not anything
}
cursor.close();
db.close();
return returnBooksList;
}

public int bookCategoryCount(String category) {
    String queryString = "SELECT COUNT (*) FROM " + TABLE_NAME + " WHERE " +
COLUMN_CATEGORY + "=" +
        "" + category + "";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);
    int count = 0;
    if (cursor != null && cursor.moveToFirst()) {
        do {
            cursor.moveToFirst();
            count = cursor.getInt(0);
        } while (cursor.moveToNext());
        cursor.close();
    }
    db.close();
    return count;
}

@RequiresApi(api = Build.VERSION_CODES.O)
public List<Book> getBooksInCategory(String category) {
    List<Book> returnBooksList = new ArrayList<>();
    String queryString = "SELECT * FROM " + TABLE_NAME + " WHERE " +
COLUMN_CATEGORY + "=" + "" + category + "";
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);

    if (cursor.moveToFirst()) {
        do {
            int bookId = cursor.getInt(0);
            String bookTitle = cursor.getString(1);
            String bookCategory = cursor.getString(2);
            String bookLink = cursor.getString(3);
            String bookTag = cursor.getString(4);
            String bookDate = cursor.getString(5);
            String bookNote = cursor.getString(6);
            int bookRating = cursor.getInt(7);

            Book book = new Book(bookId, bookTitle, bookCategory, bookLink, bookTag,
LocalDate.parse(bookDate), bookNote, bookRating);
            returnBooksList.add(book);

```

```

    } while (cursor.moveToNext());

    } else {
        //do not anything
    }
    cursor.close();
    db.close();
    return returnBooksList;
}

public boolean deleteAllBook() {
    SQLiteDatabase db = this.getWritableDatabase();
    String query = "DELETE FROM " + TABLE_NAME;
    Cursor cursor = db.rawQuery(query, null);

    if (!cursor.moveToFirst()) {
        return true;
    } else {
        return false;
    }
}

public boolean exportDB(File file) throws IOException {
    LisBookDataBase lisBookDataBase = new LisBookDataBase(context(getApplicationContext()));
    CSVWriter csvWrite = new CSVWriter(new FileWriter(file));
    SQLiteDatabase db = lisBookDataBase.getReadableDatabase();
    Cursor curCSV = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    csvWrite.writeNext(curCSV.getColumnNames());
    if (curCSV.moveToFirst()) {
        while (curCSV.moveToNext()) {
            String[] arrStr = {curCSV.getString(0), curCSV.getString(1), curCSV.getString(2),
                curCSV.getString(3), curCSV.getString(4), curCSV.getString(5), curCSV.getString(6),
                curCSV.getString(7)};
            csvWrite.writeNext(arrStr);
        }
        csvWrite.close();
        curCSV.close();
        return true;
    } else {
        return false;
    }
}
}

package com.liakh.lisbook;

import java.io.IOException;
import java.io.PrintWriter;
import java.io.Writer;

public class CSVWriter {

```

```

private PrintWriter pw;

private char separator;

private char quotechar;

private char escapechar;

private String lineEnd;

public static final char DEFAULT_ESCAPE_CHARACTER = '\\';

public static final char DEFAULT_SEPARATOR = ',';

public static final char DEFAULT_QUOTE_CHARACTER = '"';

public static final char NO_QUOTE_CHARACTER = '\u0000';

public static final char NO_ESCAPE_CHARACTER = '\u0000';

public static final String DEFAULT_LINE_END = "\n";

public CSVWriter(Writer writer) {
    this(writer, DEFAULT_SEPARATOR, DEFAULT_QUOTE_CHARACTER,
        DEFAULT_ESCAPE_CHARACTER, DEFAULT_LINE_END);
}

public CSVWriter(Writer writer, char separator, char quotechar, char escapechar, String lineEnd)
{
    this.pw = new PrintWriter(writer);
    this.separator = separator;
    this.quotechar = quotechar;
    this.escapechar = escapechar;
    this.lineEnd = lineEnd;
}

public void writeNext(String[] nextLine) {

    if (nextLine == null)
        return;

    StringBuffer sb = new StringBuffer();
    for (int i = 0; i < nextLine.length; i++) {

        if (i != 0) {
            sb.append(separator);
        }

        String nextElement = nextLine[i];
        if (nextElement == null)
            continue;
    }
}

```

```

    if (quotechar != NO_QUOTE_CHARACTER)
        sb.append(quotechar);
    for (int j = 0; j < nextElement.length(); j++) {
        char nextChar = nextElement.charAt(j);
        if (escapechar != NO_ESCAPE_CHARACTER && nextChar == quotechar) {
            sb.append(escapechar).append(nextChar);
        } else if (escapechar != NO_ESCAPE_CHARACTER && nextChar == escapechar) {
            sb.append(escapechar).append(nextChar);
        } else {
            sb.append(nextChar);
        }
    }
    if (quotechar != NO_QUOTE_CHARACTER)
        sb.append(quotechar);
}

sb.append(lineEnd);
pw.write(sb.toString());
}

public void flush() throws IOException {
    pw.flush();
}

public void close() throws IOException {
    pw.flush();
    pw.close();
}
}

package com.liakh.lisbook;

public interface RecyclerViewInterface {
    void onItemClick(int position);
}

package com.liakh.lisbook;

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

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

public class MainMenu extends AppCompatActivity implements RecyclerViewInterface {

    RecyclerView recyclerViewMainMenu;

```



```

LisBookDataBase dataBase;
String[] default_main_menu, small_title_main_menu;
int[] default_images = {R.drawable.subfolder, R.drawable.favourite, R.drawable.statistics,
    R.drawable.search, R.drawable.all_books, R.drawable.finished};

```

```

ImageButton settings;
Button newBookBtn;

```

```
@Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main_menu);

    dataBase = new LisBookDataBase(this);

    settings = findViewById(R.id.settings_img);
    newBookBtn = findViewById(R.id.new_book_btn);
    recyclerViewMainMenu = findViewById(R.id.recycle_view_main_menu);

    default_main_menu = getResources().getStringArray(R.array.default_main_menu);
    small_title_main_menu = getResources().getStringArray(R.array.small_title_main_menu);

    for (int i = 0; i < small_title_main_menu.length; i++) {
        String tmp = small_title_main_menu[i];
        switch (i) {
            case 0:
                tmp = tmp + " 4";
                small_title_main_menu[0] = tmp;
                break;
            case 1:
                tmp = tmp + " " + dataBase.bookFavouriteCount();
                small_title_main_menu[1] = tmp;
                break;
            case 4:
                tmp = tmp + " " + dataBase.allBookCount();
                small_title_main_menu[4] = tmp;
                break;
            case 5:
                tmp = tmp + " " + dataBase.bookDoneCount(getString(R.string.tag_done));
                small_title_main_menu[5] = tmp;
                break;
        }
    }

    MainMenuRecyclerViewAdapter mainMenuRecyclerViewAdapter =
        new MainMenuRecyclerViewAdapter(this, default_main_menu, small_title_main_menu,
    default_images, this);
    recyclerViewMainMenu.setAdapter(mainMenuRecyclerViewAdapter);
    recyclerViewMainMenu.setLayoutManager(new LinearLayoutManager(this));

    settings.setOnClickListener(new View.OnClickListener() {

```

```

    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainMenu.this, Settings.class);
        startActivity(intent);
    }
});

newBookBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainMenu.this, NewBookPage.class);
        startActivity(intent);
    }
});
}

@Override
public void onItemClickClick(int position) {
    Intent intent;
    switch (position) {
        case 0:
            intent = new Intent(MainMenu.this, CategoriesPage.class);
            startActivity(intent);
            finish();
            break;
        case 1:
            intent = new Intent(MainMenu.this, FavouritesBooksPage.class);
            startActivity(intent);
            finish();
            break;
        case 4:
            intent = new Intent(MainMenu.this, AllBooksPage.class);
            startActivity(intent);
            finish();
            break;
        case 5:
            intent = new Intent(MainMenu.this, DoneBooksPage.class);
            startActivity(intent);
            finish();
            break;
    }
}
}
}

package com.liakh.lisbook;

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

```

```

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

import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdView;

import java.util.ArrayList;
import java.util.Arrays;

public class CategoriesPage extends AppCompatActivity implements RecyclerViewInterface {

    RecyclerView recyclerViewCategories;
    LisBookDataBase dataBase;
    private String[] default_main_menu_categories, small_title_main_menu;
    int[] default_images = {R.drawable.hat, R.drawable.spy, R.drawable.domestic,
R.drawable.globe,R.drawable.technical,
        R.drawable.buying};

    ImageButton settings;
    Button backBtn, newBookBtn;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_categories_page);

        dataBase = new LisBookDataBase(this);

        settings = findViewById(R.id.settings_img);
        newBookBtn = findViewById(R.id.new_book_btn);
        backBtn = findViewById(R.id.back_btn);
        recyclerViewCategories = findViewById(R.id.recycle_view_main_menu);
        bannerAd();

        ArrayList<String> tmpArr = new
ArrayList<String>(Arrays.asList(getResources().getStringArray(R.array.cat_spinner_items)));
        tmpArr.remove(0);
        default_main_menu_categories = tmpArr.toArray(new String[tmpArr.size()]);
        small_title_main_menu = new String[tmpArr.size()];
        String tmp = getString(R.string.quantity);

        for (int i = 0; i < default_main_menu_categories.length; i++) {
            switch (i) {
                case 0:
                    small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_fantasy));
                    break;

```

```

        case 1:
            small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_detectives));
            break;
        case 2:
            small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_domestic_literature));
            break;
        case 3:
            small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_world_literature));
            break;
        case 4:
            small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_technical));
            break;
        case 5:
            small_title_main_menu[i] = tmp + " " +
dataBase.bookCategoryCount(getString(R.string.cat_item_to_buy));
            break;
    }
}

MainMenuRecyclerViewAdapter mainMenuRecyclerViewAdapter =
    new MainMenuRecyclerViewAdapter(this, default_main_menu_categories,
small_title_main_menu, default_images, this);
recyclerViewCategories.setAdapter(mainMenuRecyclerViewAdapter);
recyclerViewCategories.setLayoutManager(new LinearLayoutManager(this));

settings.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(CategoriesPage.this, Settings.class);
        startActivity(intent);
        finish();
    }
});

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(CategoriesPage.this, MainMenu.class);
        startActivity(intent);
        finish();
    }
});
}

private void bannerAd() {
    AdView mAdView = findViewById(R.id.ad_view);
    AdRequest.Builder adRequestBuilder = new AdRequest.Builder();
    mAdView.loadAd(adRequestBuilder.build());
}

```

```

    }

    @Override
    public void onItemClick(int position) {
        Intent intent;
        intent = new Intent(CategoriesPage.this, OneCategoryPage.class);
        intent.putExtra("category", position);
        startActivity(intent);
        finish();
    }
}

package com.liakh.lisbook;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class MainMenuRecyclerViewAdapter extends
RecyclerView.Adapter<MainMenuRecyclerViewAdapter.MainMenuViewHolder> {

    private final RecyclerViewInterface recycleViewInterface;
    String[] default_main_menu, small_title_main_menu;
    int[] default_images;
    Context context;

    public MainMenuRecyclerViewAdapter(Context ct, String[] d_m_m, String[] s_t_m_m, int[] img,
        RecyclerViewInterface recycleViewInterface) {
        this.context = ct;
        this.default_main_menu = d_m_m;
        this.small_title_main_menu = s_t_m_m;
        this.default_images = img;
        this.recycleViewInterface = recycleViewInterface;
    }

    @NonNull
    @Override
    public MainMenuViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
    {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(R.layout.main_menu_row, parent, false);
        return new MainMenuViewHolder(view, recycleViewInterface);
    }

    @Override

```

```

public void onBindViewHolder(@NonNull MainMenuViewHolder holder, int position) {
    holder.default_main_menu_name.setText(default_main_menu[position]);
    holder.small_title_main_menu.setText(small_title_main_menu[position]);
    holder.default_img_main_menu.setImageResource(default_images[position]);
}

@Override
public int getItemCount() {
    return default_main_menu.length;
}

public static class MainMenuViewHolder extends RecyclerView.ViewHolder {

    TextView default_main_menu_name, small_title_main_menu;
    ImageView default_img_main_menu;

    public MainMenuViewHolder(@NonNull View itemView, RecyclerViewInterface
recycleViewInterface) {
        super(itemView);

        default_main_menu_name = itemView.findViewById(R.id.default_main_menu_name);
        small_title_main_menu = itemView.findViewById(R.id.small_title_main_menu);
        default_img_main_menu = itemView.findViewById(R.id.default_img_main_menu);

        itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (recycleViewInterface != null) {
                    int position = getAdapterPosition();

                    if (position != RecyclerView.NO_POSITION) {
                        recycleViewInterface.onItemViewClick(position);
                    }
                }
            }
        });
    }
}

package com.liakh.lisbook;

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

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

import java.util.List;

public class AllBooksPage extends AppCompatActivity implements RecyclerViewInterface {

    LisBookDataBase dataBase;
    List<Book> allBooks;
    RecyclerView recyclerViewDone;
    BooksRecyclerViewAdapter allBooksRecyclerViewAdapter;
    Button backBtn, newBookBtn;
    ImageButton settings;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_all_books_page);

        recyclerViewDone = findViewById(R.id.recycle_done_books);
        dataBase = new LisBookDataBase(this);
        allBooks = dataBase.getAllBooks();
        backBtn = findViewById(R.id.back_btn);
        newBookBtn = findViewById(R.id.new_book_btn);
        settings = findViewById(R.id.settings_img);

        allBooksRecyclerViewAdapter = new BooksRecyclerViewAdapter(AllBooksPage.this, allBooks, this);
        recyclerViewDone.setAdapter(allBooksRecyclerViewAdapter);
        recyclerViewDone.setLayoutManager(new LinearLayoutManager(this));

        newBookBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(AllBooksPage.this, NewBookPage.class);
                startActivity(intent);
                finish();
            }
        });

        settings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(AllBooksPage.this, Settings.class);
                startActivity(intent);
                finish();
            }
        });

        backBtn.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View view) {
            Intent intent = new Intent(AllBooksPage.this, MainMenu.class);
            startActivity(intent);
            finish();
        }
    });
}

@Override
public void onItemClick(int position) {
    Intent intent = new Intent(AllBooksPage.this, BookDetailsPage.class);

    intent.putExtra("bookId", allBooks.get(position).getId());
    startActivity(intent);
    finish();
}
}

package com.liakh.lisbook;

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

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

import java.util.List;

public class DoneBooksPage extends AppCompatActivity implements RecyclerViewInterface {

    LisBookDataBase dataBase;
    List<Book> doneBooks;
    RecyclerView recyclerView;
    BooksRecyclerViewAdapter doneBooksRecyclerViewAdapter;
    Button backBtn, newBookBtn;
    ImageButton settings;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_done_book_page);

        recyclerView = findViewById(R.id.recycle_done_books);
        dataBase = new LisBookDataBase(this);

```



```

doneBooks = dataBase.getDoneBooks(getString(R.string.tag_done));
backBtn = findViewById(R.id.back_btn);
newBookBtn = findViewById(R.id.new_book_btn);
settings = findViewById(R.id.settings_img);

doneBooksRecyclerViewAdapter = new BooksRecyclerViewAdapter(DoneBooksPage.this, doneBooks,
this);
recyclerView.setAdapter(doneBooksRecyclerViewAdapter);
recyclerView.setLayoutManager(new LinearLayoutManager(this));

newBookBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(DoneBooksPage.this, NewBookPage.class);
        startActivity(intent);
        finish();
    }
});

settings.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(DoneBooksPage.this, Settings.class);
        startActivity(intent);
        finish();
    }
});

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(DoneBooksPage.this, MainMenu.class);
        startActivity(intent);
        finish();
    }
});

@Override
public void onItemClick(int position) {
    Intent intent = new Intent(DoneBooksPage.this, BookDetailsPage.class);

    intent.putExtra("bookId", doneBooks.get(position).getId());
    startActivity(intent);
    finish();
}
}

package com.liakh.lisbook;

import androidx.annotation.RequiresApi;

```

```

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

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

import java.util.List;

public class FavouritesBooksPage extends AppCompatActivity implements RecyclerViewInterface {

    LisBookDataBase dataBase;
    List<Book> favBooks;
    RecyclerView recyclerView;
    BooksRecyclerViewAdapter favBooksRecyclerViewAdapter;
    Button backBtn, newBookBtn;
    ImageButton settings;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_favourites_book_page);

        recyclerView = findViewById(R.id.recycle_done_books);
        dataBase = new LisBookDataBase(this);
        favBooks = dataBase.getFavBooks();
        backBtn = findViewById(R.id.back_btn);
        newBookBtn = findViewById(R.id.new_book_btn);
        settings = findViewById(R.id.settings_img);

        favBooksRecyclerViewAdapter = new BooksRecyclerViewAdapter(FavouritesBooksPage.this, favBooks,
this);
        recyclerView.setAdapter(favBooksRecyclerViewAdapter);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        newBookBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(FavouritesBooksPage.this, NewBookPage.class);
                startActivity(intent);
                finish();
            }
        });

        settings.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        Intent intent = new Intent(FavouritesBooksPage.this, Settings.class);
        startActivity(intent);
        finish();
    }
});

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(FavouritesBooksPage.this, MainMenu.class);
        startActivity(intent);
        finish();
    }
});
}

@Override
public void onItemClick(int position) {
    Intent intent = new Intent(FavouritesBooksPage.this, BookDetailsPage.class);
    intent.putExtra("bookId", favBooks.get(position).getId());
    startActivity(intent);
    finish();
}
}

package com.liakh.lisbook;

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

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class OneCategoryPage extends AppCompatActivity implements RecyclerViewInterface {

    LisBookDataBase dataBase;
    List<Book> oneCatBooks;
    ArrayList<String> tmpArr;

```

```

TextView categoryFromArr;
BooksRecyclerViewAdapter categoriesBooksRecyclerViewAdapter;
RecyclerView recyclerViewCat;
Button backBtn, newBookBtn;
ImageButton settings;

@RequiresApi(api = Build.VERSION_CODES.O)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_one_category_page);

    tmpArr = new
ArrayList<String>(Arrays.asList(getResources().getStringArray(R.array.cat_spinner_items)));
    tmpArr.remove(0);
    dataBase = new LisBookDataBase(this);

    categoryFromArr = findViewById(R.id.toolbar_app_name);
    recyclerViewCat = findViewById(R.id.recycle_view_one_category_books);
    newBookBtn = findViewById(R.id.new_book_btn);
    backBtn = findViewById(R.id.back_btn);
    settings = findViewById(R.id.settings_img);

    getData();
    categoriesBooksRecyclerViewAdapter = new BooksRecyclerViewAdapter(OneCategoryPage.this,
oneCatBooks, this);
    recyclerViewCat.setAdapter(categoriesBooksRecyclerViewAdapter);
    recyclerViewCat.setLayoutManager(new LinearLayoutManager(this));

    newBookBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(OneCategoryPage.this, NewBookPage.class);
            startActivity(intent);
            finish();
        }
    });

    settings.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(OneCategoryPage.this, Settings.class);
            startActivity(intent);
            finish();
        }
    });

    backBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(OneCategoryPage.this, CategoriesPage.class);
            startActivity(intent);
        }
    });
}

```

```

        finish();
    }
});
}

@RequiresApi(api = Build.VERSION_CODES.O)
public void getData() {
    int position;
    if ((position = getIntent().getIntExtra("category", -1)) != -1) {
        categoryFromArr.setText(tmpArr.get(position));
        oneCatBooks = dataBase.getBooksInCategory(categoryFromArr.getText().toString());
        System.out.println(oneCatBooks.toString());
    } else {
        Toast.makeText(this, "Something get wrong!", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onItemClick(int position) {
    Intent intent = new Intent(OneCategoryPage.this, BookDetailsPage.class);
    intent.putExtra("bookId", oneCatBooks.get(position).getId());
    startActivity(intent);
    finish();
}
}

package com.liakh.lisbook;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class BooksRecyclerViewAdapter extends
RecyclerView.Adapter<BooksRecyclerViewAdapter.AllBooksViewHolder> {

    private final RecyclerViewInterface recycleViewInterface;
    List<Book> books;
    Context context;

    public BooksRecyclerViewAdapter(Context context, List<Book> books, RecyclerViewInterface
recycleViewInterface) {
        this.books = books;
        this.context = context;
        this.recycleViewInterface = recycleViewInterface;
}
}

```

```

}

@NonNull
@Override
public AllBooksViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    LayoutInflater inflater = LayoutInflater.from(context);
    View view = inflater.inflate(R.layout.all_books_row, parent, false);
    return new AllBooksViewHolder(view, recycleViewInterface);
}

@Override
public void onBindViewHolder(@NonNull AllBooksViewHolder holder, int position) {
    holder.book_title.setText(books.get(position).getTitle());
    holder.small_book_tag.setText(books.get(position).getTag());
}

@Override
public int getItemCount() {
    return books.size();
}

public static class AllBooksViewHolder extends RecyclerView.ViewHolder {

    TextView book_title, small_book_tag;

    public AllBooksViewHolder(@NonNull View itemView, RecyclerViewInterface
recycleViewInterface) {
        super(itemView);

        book_title = itemView.findViewById(R.id.book_title);
        small_book_tag = itemView.findViewById(R.id.small_book_tag);

        itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(recycleViewInterface != null){
                    int position = getAdapterPosition();

                    if(position!=RecyclerView.NO_POSITION){
                        recycleViewInterface.onItemViewClick(position);
                    }
                }
            }
        });
    }
}

}

package com.liakh.lisbook;

import androidx.annotation.RequiresApi;

```

```

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdView;
import com.google.android.gms.ads.MobileAds;
import com.google.android.material.textfield.TextInputEditText;

import java.time.LocalDate;
import java.time.ZoneId;
import java.util.Objects;

public class NewBookPage extends AppCompatActivity implements
AdapterView.OnItemClickListener {

    LisBookDataBase dataBase;
    Spinner chooseCategories;
    EditText titleInput, linkInput;
    TextInputEditText noteInput;
    Button createBook, backBtn;

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

        MobileAds.initialize(this);
        bannerAd();

        dataBase = new LisBookDataBase(this);

        backBtn = findViewById(R.id.back_btn);
        createBook = findViewById(R.id.save_btn);
        titleInput = findViewById(R.id.input_title_txt);
        linkInput = findViewById(R.id.input_link);
        noteInput = (TextInputEditText) findViewById(R.id.note_input_txt);
        chooseCategories = findViewById(R.id.cat_spinner_change);

        ArrayAdapter adapterCategory = ArrayAdapter.createFromResource(this,
R.array.cat_spinner_items, R.layout.color_spinner_layout);
        adapterCategory.setDropDownViewResource(R.layout.color_spinner_layout);
        chooseCategories.setAdapter(adapterCategory);
    }
}

```

```

chooseCategories.setOnItemClickListener(this);

createBook.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    public void onClick(View v) {
        String tmp = titleInput.getText().toString().trim();
        if(tmp != null && !tmp.equals("")) {
            createBook();
        } else {
            Toast.makeText(NewBookPage.this, R.string.error_create,
Toast.LENGTH_SHORT).show();
        }
    }
});

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(NewBookPage.this, MainMenu.class);
        startActivity(intent);
        finish();
    }
});
}

private void bannerAd() {
    AdView mAdView = findViewById(R.id.ad_view);
    AdRequest.Builder adRequestBuilder = new AdRequest.Builder();
    mAdView.loadAd(adRequestBuilder.build());
}

@RequiresApi(api = Build.VERSION_CODES.O)
private void createBook(){
    String errorAdded = getResources().getString(R.string.error_create),
        successAdded = getResources().getString(R.string.success_create);
    String[] tags = getResources().getStringArray(R.array.tag_spinner_items);
    Book book;
    try {
        book = new Book(1, titleInput.getText().toString().trim(),
chooseCategories.getSelectedItem().toString(),
        linkInput.getText().toString(), tags[0], LocalDate.now(ZoneId.of("Europe/Kiev")),
        Objects.requireNonNull(noteInput.getText()).toString(), 0);
        boolean success = dataBase.createBook(book);

        Toast.makeText(NewBookPage.this, successAdded, Toast.LENGTH_SHORT).show();
        returnOnMain();
    } catch (Exception e) {
        Toast.makeText(NewBookPage.this, errorAdded, Toast.LENGTH_SHORT).show();
    }
}
}

```



```

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
    Toast.makeText(this, adapterView.getSelectedItem().toString(),
Toast.LENGTH_LONG).show();
}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {

}

public void returnOnMain() {
    Intent intent = new Intent(this, MainMenu.class);
    startActivity(intent);
    finish();
}
}

```

```

package com.liakh.lisbook;

```

```

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

```

```

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.RatingBar;
import android.widget.TextView;
import android.widget.Toast;

```

```

import com.google.android.material.textfield.TextInputEditText;

```

```

import java.time.format.DateTimeFormatter;

```

```

public class BookDetailsPage extends AppCompatActivity {

```

```

    LisBookDataBase dataBase;
    Book book;
    ImageButton linkBtn, deleteBtn;
    Button changeBtn, backBtn;
    TextView bookTitle, bookTag, creationDate, bookCategory;
    TextInputEditText bookNote;
    RatingBar ratingBar;

```

```
int bookId;
```

```
@RequiresApi(api = Build.VERSION_CODES.O)
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_book_details_page);
    dataBase = new LisBookDataBase(this);
    getData();

    bookCategory = findViewById(R.id.toolbar_one_book);
    bookTitle = findViewById(R.id.book_title_txt);
    bookTag = findViewById(R.id.tag_txt);
    creationDate = findViewById(R.id.creation_date);
    bookNote = (TextInputEditText) findViewById(R.id.note_input_txt);
    ratingBar = findViewById(R.id.rating_stars);
    backBtn = findViewById(R.id.back_btn);
    deleteBtn = findViewById(R.id.settings_img);

```

```
setBookFromDb();
```

```
linkBtn = findViewById(R.id.link_img_btn);
changeBtn = findViewById(R.id.change_btn);
```

```
linkBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse(book.getLink()));
        startActivity(intent);
    }
});
```

```
deleteBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        confirmDeleteDialog();
    }
});
```

```
changeBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(BookDetailsPage.this, BookChangePage.class);
        intent.putExtra("bookId", book.getId());

        startActivity(intent);
        finish();
    }
});
```

```

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(BookDetailsPage.this, MainMenu.class);

        startActivity(intent);
        finish();
    }
});
}

private void confirmDeleteDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(R.string.delete);
    builder.setMessage(getString(R.string.dialog_delete_txt) + ": " + book.getTitle() + "?");
    builder.setPositiveButton(R.string.dialog_yes, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            try {
                dataBase.deleteById(book);
                Intent intent = new Intent(BookDetailsPage.this, AllBooksPage.class);
                Toast.makeText(BookDetailsPage.this, R.string.success_delete,
Toast.LENGTH_SHORT).show();
                startActivity(intent);
                finish();
            } catch (Exception e) {
                Toast.makeText(BookDetailsPage.this, R.string.error_delete,
Toast.LENGTH_SHORT).show();
                Log.e("error", e.getMessage());
            }
        }
    });
    builder.setNegativeButton(R.string.dialog_no, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.dismiss();
        }
    });
    builder.create();
    builder.show();
}

@RequiresApi(api = Build.VERSION_CODES.O)
public void getData() {
    int bookId;
    if ((bookId = getIntent().getIntExtra("bookId", -1)) != -1) {
        book = dataBase.bookById(bookId);
    } else {
        Toast.makeText(this, "Something get wrong!", Toast.LENGTH_SHORT).show();
    }
}
}

```

```

@RequiresApi(api = Build.VERSION_CODES.O)
@SuppressLint("SimpleDateFormat")
public void setBookFromDb() {
    bookId = book.getId();
    bookCategory.setText(book.getCategory());
    bookTitle.setText(book.getTitle());
    bookTag.setText(book.getTag());

    creationDate.setText(DateTimeFormatter.ofPattern("dd/MM/yy").format(book.getCreate_date()));
    bookNote.setText(book.getNote());
    ratingBar.setRating(book.getRating());
}

}

package com.liakh.lisbook;

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

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RatingBar;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;

public class BookChangePage extends AppCompatActivity implements
AdapterView.OnItemClickListener {

    Spinner chooseTag, chooseCategory;
    LisBookDataBase dataBase;
    Book book;
    Button updateBtn, backBtn;
    EditText bookTitle, bookLink;
    TextInputEditText bookNote;
    TextInputLayout noteInputContainer;
    RatingBar ratingBar;
    int bookId;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_book_change_page);
dataBase = new LisBookDataBase(this);

getData();

bookTitle = findViewById(R.id.input_new_title_txt);
chooseCategory = findViewById(R.id.cat_spinner_change);
bookLink = findViewById(R.id.input_new_link);
chooseTag = findViewById(R.id.tag_spinner);
noteInputContainer = findViewById(R.id.note_input_view);
bookNote = findViewById(R.id.note_input_txt);
ratingBar = findViewById(R.id.rating_stars);
updateBtn = findViewById(R.id.save_btn);
backBtn = findViewById(R.id.back_btn);
ArrayAdapter adapterTag = ArrayAdapter.createFromResource(this, R.array.tag_spinner_items,
R.layout.color_spinner_layout);
adapterTag.setDropDownViewResource(R.layout.color_spinner_layout);
chooseTag.setAdapter(adapterTag);
chooseTag.setSelection(adapterTag.getPosition(book.getTag()));
chooseTag.setOnItemClickListener(this);

ArrayAdapter adapterCategory = ArrayAdapter.createFromResource(this,
R.array.cat_spinner_items, R.layout.color_spinner_layout);
adapterCategory.setDropDownViewResource(R.layout.color_spinner_layout);
chooseCategory.setAdapter(adapterCategory);
chooseCategory.setSelection(adapterCategory.getPosition(book.getCategory()));
chooseCategory.setOnItemClickListener(this);

setBookFromDb();

updateBtn.setOnClickListener(v -> {
    dataBase.updateBookInDb(updateBook(book));
    Toast.makeText(BookChangePage.this, R.string.success_update,
Toast.LENGTH_SHORT).show();
    returnBack();
});

backBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(BookChangePage.this, BookDetailsPage.class);
        intent.putExtra("bookId", book.getId());
        startActivity(intent);
        finish();
    }
});
}

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

```

```

}

@Override
public void onNothingSelected(AdapterView<?> adapterView) {

}

@RequiresApi(api = Build.VERSION_CODES.O)
public void getData() {
    int bookId;
    if ((bookId = getIntent().getIntExtra("bookId", -1)) != -1) {
        book = dataBase.bookById(bookId);
    } else {
        Toast.makeText(this, "Something get wrong!", Toast.LENGTH_SHORT).show();
    }
}

@RequiresApi(api = Build.VERSION_CODES.O)
@SuppressLint("SimpleDateFormat")
public void setBookFromDb() {
    bookId = book.getId();
    bookTitle.setText(book.getTitle());
    bookLink.setText(book.getLink());
    bookNote.setText(book.getNote());
    ratingBar.setRating(book.getRating());
}

public Book updateBook(Book bookToDb){
    bookToDb.setId(bookId);
    bookToDb.setCategory(chooseCategory.getSelectedItem().toString());
    bookToDb.setTitle(bookTitle.getText().toString());
    bookToDb.setLink(bookLink.getText().toString());
    bookToDb.setTag(chooseTag.getSelectedItem().toString());
    bookToDb.setNote(bookNote.getText().toString());
    bookToDb.setRating(((int) ratingBar.getRating()));
    return bookToDb;
}

public void returnBack(){
    Intent intent = new Intent(this, AllBooksPage.class);
    startActivity(intent);
    finish();
}
}

package com.liakh.lisbook;

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

```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

```

```

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.widget.Button;
import android.widget.Switch;
import android.widget.Toast;

```

```

import java.io.File;

```

```

public class Settings extends AppCompatActivity {

```

```

    @SuppressWarnings("UseSwitchCompatOrMaterialCode")
    Switch themeSwitcher;
    boolean nightMode;
    Button deleteBtn, exportBdBtn;
    SharedPreferences sharedPreferences;
    SharedPreferences.Editor editor;
    LisBookDataBase dataBase;
    private static final int STORAGE_PERMISSION_CODE = 100;
    private static final int MANAGE_PERMISSION_CODE = 101;

```

```

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_settings);
    dataBase = new LisBookDataBase(this);

    themeSwitcher = findViewById(R.id.theme_switcher);
    deleteBtn = findViewById(R.id.delete_all_btn);
    exportBdBtn = findViewById(R.id.export_btn);
    sharedPreferences = getSharedPreferences("MODE", Context.MODE_PRIVATE);
    nightMode = sharedPreferences.getBoolean("night", false);

    if (nightMode) {
        themeSwitcher.setChecked(true);
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
    }
}

```

```

themeSwitcher.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (nightMode) {
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);
            editor = sharedPreferences.edit();
            editor.putBoolean("night", false);
        } else {
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
            editor = sharedPreferences.edit();
            editor.putBoolean("night", true);
        }
        editor.apply();
    }
});

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

exportBdBtn.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.R)
    @Override
    public void onClick(View view) {
        checkPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE,
STORAGE_PERMISSION_CODE);
        checkPermission(Manifest.permission.MANAGE_EXTERNAL_STORAGE,
MANAGE_PERMISSION_CODE);
        confirmExportDb();
    }
});

if (Build.VERSION.SDK_INT >= 30) {
    if (!Environment.isExternalStorageManager()) {
        Intent getpermission = new Intent();

getpermission.setAction(android.provider.Settings.ACTION_MANAGE_ALL_FILES_ACCESS_PER
MISSION);
        startActivity(getpermission);
    }
}

@Override
public void onBackPressed() {
    Settings.super.onBackPressed();
}

private void confirmDeleteDialog() {

```



```

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle(R.string.delete);
builder.setMessage(getString(R.string.dialog_delete_all_txt));
builder.setPositiveButton(R.string.dialog_yes, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        try {
            if (dataBase.deleteAllBook()) {
                Intent intent = new Intent(Settings.this, MainMenu.class);
                Toast.makeText(Settings.this, R.string.success_delete,
Toast.LENGTH_SHORT).show();
                startActivity(intent);
            }
            finish();
        } catch (Exception e) {
            Toast.makeText(Settings.this, R.string.error_delete, Toast.LENGTH_SHORT).show();
            Log.e("error", e.getMessage());
        }
    }
});
builder.setNegativeButton(R.string.dialog_no, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        dialogInterface.dismiss();
    }
});
builder.create();
builder.show();
}

private void confirmExportDb() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(R.string.export);
    builder.setMessage(getString(R.string.dialog_export_txt));
    builder.setPositiveButton(R.string.dialog_yes, new DialogInterface.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.R)
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            boolean isSuccess;
            try {
                File exportDir = new File(Environment.getExternalStorageDirectory(), "LisBook");
                if (!exportDir.exists()) {
                    exportDir.mkdirs();
                }
                File file = new File(exportDir, "lisbook.csv");
                isSuccess = dataBase.exportDB(file);
                if (isSuccess) {
                    file.createNewFile();
                    Toast.makeText(Settings.this, R.string.success_export,
Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(Settings.this, MainMenu.class);
                    startActivity(intent);
                }
            }
        }
    }
}

```

```

        finish();
    }
} catch (Exception e) {
    Toast.makeText(Settings.this, R.string.error_export, Toast.LENGTH_SHORT).show();
    Log.e("error", e.getMessage());
}
}
});

builder.setNegativeButton(R.string.dialog_no, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        dialogInterface.dismiss();
    }
});
builder.create();
builder.show();
}

@RequiresApi(api = Build.VERSION_CODES.R)
public void checkPermission(String permission, int requestCode) {
    // Checking if permission is not granted
    if (ContextCompat.checkSelfPermission(Settings.this, permission) ==
PackageManager.PERMISSION_DENIED) {
        ActivityCompat.requestPermissions(Settings.this, new String[]{permission}, requestCode);
    } else {
        Toast.makeText(Settings.this, R.string.permission_has, Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onRequestPermissionsResult(int requestCode,
                                     @NonNull String[] permissions,
                                     @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == STORAGE_PERMISSION_CODE) {
        if (grantResults.length > 0
            && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(Settings.this, R.string.permission_granted,
Toast.LENGTH_SHORT).show();//stringiiiiii
        } else {
            Toast.makeText(Settings.this, R.string.permission_denied,
Toast.LENGTH_SHORT).show();
        }
    }
}
}
}

```

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

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
style="@style/Screen"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".AllBooksPage">

```

<androidx.constraintlayout.widget.ConstraintLayout

```

    android:id="@+id/toolbar"
    style="@style/ToolBar"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    app:layout_constraintTop_toTopOf="parent">

```

<TextView

```

    android:id="@+id/toolbar_all_books"
    style="@style/ToolBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:font="@font/averia_libre_bold"
    android:paddingStart="40dp"
    android:paddingTop="12dp"
    android:paddingEnd="0dp"
    android:paddingBottom="10dp"
    android:text="@string/all_b"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/settings_img"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

```

</TextView>

<ImageButton

```

    android:id="@+id/settings_img"
    style="@style/ImgButton"
    android:layout_width="56dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="40dp"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/toolbar_all_books"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/setting"
    tools:ignore="SpeakableTextPresentCheck" /> <!--WTF IS SPEAKABLE TEXT?-->

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.recyclerview.widget.RecyclerView
```

```
    android:id="@+id/recycle_done_books"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="4dp"
    app:layout_constraintBottom_toTopOf="@id/button_block"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.797"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/toolbar" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/button_block"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginBottom="8dp"
    android:fitsSystemWindows="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">
```

```
<Button
```

```
    android:id="@+id/new_book_btn"
    style="@style/Button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="2dp"
    android:text="@string/add"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/back_btn"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<Button
```

```
    android:id="@+id/back_btn"
    style="@style/Button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginBottom="2dp"
    android:text="@string/back"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/new_book_btn" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

</androidx.constraintlayout.widget.ConstraintLayout>

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".BookChangePage"
style="@style/Screen">
```

<TextView

```
android:id="@+id/book_change_txt"
style="@style/AnyMainText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginVertical="20dp"
android:font="@font/averia_libre_bold"
android:text="@string/change"
android:textSize="30sp"
app:layout_constraintBottom_toTopOf="@id/input_new_title_txt"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<EditText

```
android:id="@+id/input_new_title_txt"
style="@style/AnyMainText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="16dp"
android:ems="10"
android:importantForAutofill="no"
android:inputType="text"
android:minHeight="48dp"
android:text="@string/title"
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/book_change_txt" />
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/tag_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="16dp"
android:minHeight="48dp"
```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/input_new_title_txt">

```

<TextView

```

    android:id="@+id/tag_txt"
    style="@style/AnyMainText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:font="@font/averia_libre_bold"
    android:text="@string/tag"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/tag_spinner"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

<Spinner

```

    android:id="@+id/tag_spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:entries="@array/tag_spinner_items"
    android:contentDescription="@string/tag"
    android:minWidth="200dp"
    android:minHeight="48dp"
    android:theme="@style/mySpinnerItemStyle"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/tag_txt"
    app:layout_constraintTop_toTopOf="parent" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

```

    android:id="@+id/category_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="16dp"
    android:maxHeight="52dp"
    android:minHeight="48dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/tag_block">

```

<Spinner

```

    android:id="@+id/cat_spinner_change"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:contentDescription="@string/categories"
    android:ellipsize="end"
    android:maxLines="1"
    android:minHeight="48dp"

```

```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintEnd_toEndOf="parent"/>

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/link_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="16dp"
    android:maxHeight="52dp"
    android:minHeight="48dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/category_layout">

```

<EditText

```

    android:id="@+id/input_new_link"
    style="@style/AnyMainText"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginEnd="10dp"
    android:ems="10"
    android:importantForAutofill="no"
    android:inputType="textWebEmailAddress"
    android:minHeight="48dp"
    android:text="@string/paste_url"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/link_img_btn"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

<ImageView

```

    android:id="@+id/link_img_btn"
    android:layout_width="48dp"
    android:layout_height="0dp"
    android:contentDescription="@string/paste_url"
    android:paddingStart="8dp"
    android:paddingEnd="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/input_new_link"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/url" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

<com.google.android.material.textfield.TextInputLayout

```

android:id="@+id/note_input_view"
style="@style/Custom.OutlinedBox"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginVertical="10dp"
android:padding="2dp"
android:hint="@string/note"
app:counterEnabled="true"
app:counterMaxLength="300"
app:helperText="@string/letters_const"
app:helperTextTextColor="@color/red"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/link_layout">

```

<com.google.android.material.textfield.TextInputEditText

```

android:id="@+id/note_input_txt"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="top"
android:textSize="18sp"
style="@style/AnyMainText"
android:inputType="textFilter|textMultiLine"
android:maxLength="300"
android:lines="10"
android:overScrollMode="always"
android:scrollbarStyle="insideInset"
android:scrollbars="vertical"
android:scrollHorizontally="false"/>

```

</com.google.android.material.textfield.TextInputLayout>

<!--android:inputType="textMultiLine"-->

<RatingBar

```

android:id="@+id/rating_stars"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="10dp"
android:contentDescription="@string/rating"
android:numStars="5"
android:stepSize="1"
android:theme="@style/RatingBar"
app:layout_constraintTop_toBottomOf="@id/note_input_view"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"/>

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/button_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:layout_constraintEnd_toEndOf="parent"

```



```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true">

```

<Button

```

android:id="@+id/save_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginEnd="8dp"
android:text="@string/save"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/back_btn"
app:layout_constraintStart_toStartOf="parent" />

```

<Button

```

android:id="@+id/back_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginStart="8dp"
android:text="@string/back"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/save_btn" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
style="@style/Screen"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".BookDetailsPage">

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/toolbar_book_details"
style="@style/ToolBar"
android:layout_width="match_parent"
android:layout_height="60dp"
app:layout_constraintTop_toTopOf="parent">

```

<TextView

```

    android:id="@+id/toolbar_one_book"
    style="@style/ToolBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:ellipsize="end"
    android:font="@font/averia_libre_bold"
    android:maxLines="1"
    android:paddingStart="40dp"
    android:paddingTop="12dp"
    android:paddingEnd="0dp"
    android:paddingBottom="10dp"
    android:text="DEFAULT ONE CATEGORY"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/settings_img"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

```

</TextView>**<ImageButton**

```

    android:id="@+id/settings_img"
    style="@style/ImgButton"
    android:layout_width="56dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="40dp"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/toolbar_one_book"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/delete"
    tools:ignore="SpeakableTextPresentCheck" />

```

</androidx.constraintlayout.widget.ConstraintLayout>**<androidx.constraintlayout.widget.ConstraintLayout**

```

    android:id="@+id/book_title_block"
    style="@style/ConstraintLayoutBookTitle"
    android:layout_width="match_parent"
    android:layout_height="110dp"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="26dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

```

```
app:layout_constraintTop_toBottomOf="@id/toolbar_book_details">
```

<TextView

```
    android:id="@+id/book_title_txt"
    style="@style/AnyMainText"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginHorizontal="12dp"
    android:layout_marginVertical="10dp"
    android:ellipsize="end"
    android:font="@font/averia_libre_bold"
    android:gravity="start"
    android:maxLines="3"
    android:text="DEFAULT BOOK TITLE"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/link_img_btn"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

<ImageButton

```
    android:id="@+id/link_img_btn"
    style="@style/ImgWhiteLinkButton"
    android:layout_width="56dp"
    android:layout_height="wrap_content"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/book_title_txt"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/url_white"
    tools:ignore="SpeakableTextPresentCheck" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

<androidx.constraintlayout.widget.ConstraintLayout

```
    android:id="@+id/book_tag_block"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/book_title_block">
```

<TextView

```
    android:id="@+id/book_tag_txt"
    style="@style/AnyMainText"
    android:layout_width="match_parent"
```

```

android:layout_height="wrap_content"
android:font="@font/averia_libre_bold"
android:gravity="start"
android:text="@string/tag"
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

<TextView

```

android:id="@+id/tag_txt"
style="@style/AnyMainText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:font="@font/averia_libre_bold"
android:gravity="start"
android:text="DONE/IN PROGRESS/NEW"
android:textSize="24sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/date_create_block"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="10dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/book_tag_block">

```

<TextView

```

android:id="@+id/create_date_txt"
style="@style/AnyMainText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:font="@font/averia_libre_bold"
android:gravity="start"
android:text="@string/date_crt"
android:textSize="24sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/creation_date"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

<TextView

```

android:id="@+id/creation_date"
style="@style/AnyMainText"
android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:font="@font/averia_libre_bold"
    android:gravity="start"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:text="@tools:sample/date/ddmmyy" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

<com.google.android.material.textfield.TextInputLayout

```

    android:id="@+id/note_input_view"
    style="@style/Custom.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginVertical="10dp"
    android:hint="@string/note"
    android:padding="2dp"
    app:counterEnabled="true"
    app:counterMaxLength="300"
    app:helperText="@string/letters_const"
    app:helperTextTextColor="@color/red"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/date_create_block">

```

<com.google.android.material.textfield.TextInputEditText

```

    android:id="@+id/note_input_txt"
    style="@style/AnyMainText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:gravity="top"
    android:inputType="textMultiLine"
    android:lines="12"
    android:overScrollMode="always"
    android:scrollbarStyle="insideInset"
    android:scrollbars="vertical"
    android:scrollHorizontally="false" />

```

</com.google.android.material.textfield.TextInputLayout>

<RatingBar

```

    android:id="@+id/rating_stars"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="20dp"
    android:contentDescription="@string/rating"
    android:numStars="5"
    android:isIndicator="true"
    android:stepSize="1"

```

```

android:theme="@style/RatingBar"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/note_input_view" />

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/button_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent">

```

<Button

```

android:id="@+id/change_btn"
style="@style/Button"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:layout_marginBottom="2dp"
android:text="@string/change"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/back_btn"
app:layout_constraintStart_toStartOf="parent" />

```

<Button

```

android:id="@+id/back_btn"
style="@style/Button"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginBottom="2dp"
android:text="@string/back"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/change_btn" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

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

```

```

android:layout_height="match_parent"
tools:context=".CategoriesPage"
style="@style/Screen">

```

<androidx.constraintlayout.widget.ConstraintLayout

```

    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    app:layout_constraintTop_toTopOf="parent"
    style="@style/ToolBar">

```

<TextView

```

    android:id="@+id/toolbar_app_name"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:font="@font/averia_libre_bold"
    android:paddingStart="40dp"
    android:paddingTop="12dp"
    android:paddingEnd="0dp"
    android:paddingBottom="10dp"
    android:text="@string/categories"
    style="@style/ToolBar"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/settings_img"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

```

</TextView>

<ImageButton

```

    android:id="@+id/settings_img"
    android:layout_width="56dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="40dp"
    style="@style/ImgButton"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/toolbar_app_name"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/setting"
    tools:ignore="SpeakableTextPresentCheck" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.recyclerview.widget.RecyclerView

```

    android:id="@+id/recycle_view_main_menu"

```

```

android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="4dp"
app:layout_constraintBottom_toTopOf="@id/ad_view"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.797"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/toolbar" />

```

<com.google.android.gms.ads.AdView

```

android:id="@+id/ad_view"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
xmlns:ads="http://schemas.android.com/apk/res-auto"
ads:adSize="LARGE_BANNER"
ads:adUnitId="ca-app-pub-3940256099942544/6300978111"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toTopOf="@id/button_block"/>

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/button_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true">

```

<Button

```

android:id="@+id/back_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginStart="8dp"
android:text="@string/back"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"

```



```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".DoneBooksPage"
style="@style/Screen">

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="60dp"
app:layout_constraintTop_toTopOf="parent"
style="@style/ToolBar">

```

<TextView

```

android:id="@+id/toolbar_all_books"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:font="@font/averia_libre_bold"
android:paddingStart="40dp"
android:paddingTop="12dp"
android:paddingEnd="0dp"
android:paddingBottom="10dp"
android:text="@string/tag_done"
style="@style/ToolBar"
android:textSize="30sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/settings_img"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0">

```

</TextView>

<ImageButton

```

android:id="@+id/settings_img"
android:layout_width="56dp"
android:layout_height="wrap_content"
android:layout_marginEnd="40dp"
style="@style/ImageButton"
android:padding="8dp"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toEndOf="@id/toolbar_all_books"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0"
app:srcCompat="@drawable/setting"
tools:ignore="SpeakableTextPresentCheck" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.recyclerview.widget.RecyclerView
```

```
    android:id="@+id/recycle_done_books"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="4dp"
    app:layout_constraintBottom_toTopOf="@id/button_block"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.797"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/toolbar" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/button_block"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginHorizontal="40dp"
    android:layout_marginBottom="8dp"
    android:fitsSystemWindows="true">
```

```
<Button
```

```
    android:id="@+id/new_book_btn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginEnd="8dp"
    android:text="@string/add"
    android:textSize="18sp"
    style="@style/Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/back_btn"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<Button
```

```
    android:id="@+id/back_btn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginStart="8dp"
    android:text="@string/back"
    android:textSize="18sp"
    style="@style/Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/new_book_btn" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

</androidx.constraintlayout.widget.ConstraintLayout>

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".FavouritesBooksPage"
style="@style/Screen">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="60dp"
app:layout_constraintTop_toTopOf="parent"
style="@style/ToolBar">
```

```
<TextView
android:id="@+id/toolbar_all_books"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:font="@font/averia_libre_bold"
android:paddingStart="40dp"
android:paddingTop="12dp"
android:paddingEnd="0dp"
android:paddingBottom="10dp"
android:text="@string/fav"
style="@style/ToolBar"
android:textSize="30sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/settings_img"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0">
```

</TextView>

```
<ImageButton
android:id="@+id/settings_img"
android:layout_width="56dp"
android:layout_height="wrap_content"
android:layout_marginEnd="40dp"
style="@style/ImageButton"
android:padding="8dp"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
```

```

app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toEndOf="@id/toolbar_all_books"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0"
app:srcCompat="@drawable/setting"
tools:ignore="SpeakableTextPresentCheck" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.recyclerview.widget.RecyclerView

```

android:id="@+id/recycle_done_books"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="4dp"
app:layout_constraintBottom_toTopOf="@id/button_block"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.797"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/toolbar" />

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/button_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true">

```

<Button

```

android:id="@+id/new_book_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginEnd="8dp"
android:text="@string/add"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/back_btn"
app:layout_constraintStart_toStartOf="parent" />

```

<Button

```

android:id="@+id/back_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginStart="8dp"
android:text="@string/back"

```

```

        android:textSize="18sp"
        style="@style/Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@id/new_book_btn" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainMenu"
style="@style/Screen">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    app:layout_constraintTop_toTopOf="parent"
    style="@style/ToolBar">

    <TextView
        android:id="@+id/toolbar_app_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:font="@font/averia_libre_bold"
        android:paddingStart="40dp"
        android:paddingTop="12dp"
        android:paddingEnd="0dp"
        android:paddingBottom="10dp"
        android:text="@string/app_name"
        style="@style/ToolBar"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@id/settings_img"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0">

```

```
</TextView>
```

```

<ImageButton
    android:id="@+id/settings_img"
    android:layout_width="56dp"

```

```

    android:layout_height="wrap_content"
    android:layout_marginEnd="40dp"
    style="@style/ImgButton"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/toolbar_app_name"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/setting"
    tools:ignore="SpeakableTextPresentCheck" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recycle_view_main_menu"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="4dp"
    app:layout_constraintBottom_toTopOf="@id/new_book_btn"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.797"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/toolbar" />

```

<Button

```

    android:id="@+id/new_book_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="4dp"
    style="@style/Button"
    android:text="@string/new_b"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/recycle_view_main_menu" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

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

```

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    style="@style/Screen"
    android:layout_width="match_parent"

```

```
android:layout_height="match_parent"
tools:context=".NewBookPage">
```

<TextView

```
android:id="@+id/new_book_txt"
style="@style/AnyMainText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginVertical="20dp"
android:font="@font/averia_libre_bold"
android:text="@string/new_b"
android:textSize="30sp"
app:layout_constraintBottom_toTopOf="@id/input_title_txt"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<EditText

```
android:id="@+id/input_title_txt"
style="@style/HintText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginVertical="20dp"
android:ems="10"
android:importantForAutofill="no"
android:inputType="text"
android:minHeight="48dp"
android:hint="@string/title"
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/new_book_txt" />
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/category_layout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginVertical="20dp"
android:maxHeight="52dp"
android:minHeight="48dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/input_title_txt">
```

<Spinner

```
android:id="@+id/cat_spinner_change"
android:layout_width="0dp"
android:layout_height="match_parent"
android:contentDescription="@string/categories"
```

```

    android:ellipsize="end"
    android:maxLines="1"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/link_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginVertical="20dp"
    android:maxHeight="52dp"
    android:minHeight="48dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/category_layout">

```

<EditText

```

    android:id="@+id/input_link"
    style="@style/HintText"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginEnd="10dp"
    android:ems="10"
    android:importantForAutofill="no"
    android:inputType="textWebEmailAddress"
    android:minHeight="48dp"
    android:hint="@string/paste_url"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/link_img_btn"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

<ImageView

```

    android:id="@+id/link_img_btn"
    android:layout_width="48dp"
    android:layout_height="0dp"
    android:contentDescription="@string/paste_url"
    android:paddingStart="8dp"
    android:paddingEnd="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/input_link"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/url" />

```



```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
```

```
    android:id="@+id/note_input_view"
    style="@style/Custom.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    android:layout_marginVertical="10dp"
    android:padding="2dp"
    android:hint="@string/note"
    app:counterEnabled="true"
    app:counterMaxLength="300"
    app:helperText="@string/letters_const"
    app:helperTextTextColor="@color/red"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/link_layout">
```

```
<com.google.android.material.textfield.TextInputEditText
```

```
    android:id="@+id/note_input_txt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="top"
    style="@style/AnyMainText"
    android:inputType="textFilter|textMultiLine"
    android:maxLength="300"
    android:lines="10"
    android:overScrollMode="always"
    android:scrollbarStyle="insideInset"
    android:scrollbars="vertical"
    android:scrollHorizontally="false" />
```

```
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.gms.ads.AdView
```

```
    android:id="@+id/ad_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    ads:adSize="LARGE_BANNER"
    ads:adUnitId="ca-app-pub-3940256099942544/6300978111"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/note_input_view"
    app:layout_constraintBottom_toTopOf="@id/button_block"/>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/button_block"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
```

```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true">

```

<Button

```

    android:id="@+id/save_btn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginEnd="8dp"
    android:text="@string/save"
    android:textSize="18sp"
    style="@style/Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/back_btn"
    app:layout_constraintStart_toStartOf="parent" />

```

<Button

```

    android:id="@+id/back_btn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginStart="8dp"
    android:text="@string/back"
    android:textSize="18sp"
    style="@style/Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/save_btn" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
style="@style/Screen"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".OneCategoryPage">

```

<androidx.constraintlayout.widget.ConstraintLayout

```

    android:id="@+id/toolbar"
    style="@style/ToolBar"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    app:layout_constraintTop_toTopOf="parent">

```

<TextView

```

    android:id="@+id/toolbar_app_name"
    style="@style/ToolBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:font="@font/averia_libre_bold"
    android:paddingStart="40dp"
    android:paddingTop="12dp"
    android:paddingEnd="0dp"
    android:paddingBottom="10dp"
    android:text="ONE CATEGORY CHOOSE"
    android:maxLines="1"
    android:textSize="30sp"
    android:ellipsize="end"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/settings_img"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

```

</TextView>**<ImageButton**

```

    android:id="@+id/settings_img"
    style="@style/ImgButton"
    android:layout_width="56dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="40dp"
    android:padding="8dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@id/toolbar_app_name"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/setting"
    tools:ignore="SpeakableTextPresentCheck" />

```

</androidx.constraintlayout.widget.ConstraintLayout>**<androidx.recyclerview.widget.RecyclerView**

```

    android:id="@+id/recycle_view_one_category_books"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginHorizontal="40dp"
    android:layout_marginTop="4dp"
    app:layout_constraintBottom_toTopOf="@id/button_block"
    app:layout_constraintEnd_toEndOf="parent"

```

```

app:layout_constraintHorizontal_bias="0.797"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/toolbar" />

```

<androidx.constraintlayout.widget.ConstraintLayout

```

android:id="@+id/button_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginHorizontal="40dp"
android:layout_marginBottom="8dp"
android:fitsSystemWindows="true">

```

<Button

```

android:id="@+id/new_book_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginEnd="8dp"
android:text="@string/add"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/back_btn"
app:layout_constraintStart_toStartOf="parent" />

```

<Button

```

android:id="@+id/back_btn"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="2dp"
android:layout_marginStart="8dp"
android:text="@string/back"
android:textSize="18sp"
style="@style/Button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/new_book_btn" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
style="@style/Screen"
android:layout_width="match_parent"

```

```
android:layout_height="match_parent"
tools:context=".Settings">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/toolbar"
    style="@style/ToolBar"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    app:layout_constraintTop_toTopOf="parent">
```

```
<TextView
```

```
    android:id="@+id/toolbar_settings"
    style="@style/ToolBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:font="@font/averia_libre_bold"
    android:paddingStart="40dp"
    android:paddingTop="12dp"
    android:paddingEnd="0dp"
    android:paddingBottom="10dp"
    android:text="@string/settings"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:id="@+id/constr_theme"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="40dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/toolbar">
```

```
<TextView
```

```
    android:id="@+id/change_theme_txt"
    style="@style/AnyMainText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/switch_theme"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/theme_switcher"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent" />
```

<Switch

```
android:id="@+id/theme_switcher"
android:layout_width="54dp"
android:layout_height="54dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/change_theme_txt"
app:layout_constraintTop_toTopOf="parent"
tools:ignore="UseSwitchCompatOrMaterialXml" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/delete_all_block"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginHorizontal="40dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/constr_theme">
```

<TextView

```
android:id="@+id/delete_all_txt"
style="@style/AnyMainText"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:text="@string/delete_all_txt"
android:textSize="20sp"
android:textStyle="bold"
android:layout_marginEnd="20dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/delete_all_btn"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/delete_all_btn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
style="@style/Button"
android:text="@string/delete"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/delete_all_txt"
app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

<androidx.constraintlayout.widget.ConstraintLayout

```
android:id="@+id/export_block"
```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginHorizontal="40dp"
android:layout_marginTop="10dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/delete_all_block">

```

<TextView

```

android:id="@+id/export_txt"
style="@style/AnyMainText"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:text="@string/export_txt"
android:textSize="20sp"
android:textStyle="bold"
android:layout_marginEnd="20dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/export_btn"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

<Button

```

android:id="@+id/export_btn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
style="@style/Button"
android:text="@string/export"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/export_txt"
app:layout_constraintTop_toTopOf="parent" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

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

<androidx.constraintlayout.widget.ConstraintLayout

```

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"
xmlns:app="http://schemas.android.com/apk/res-auto">

```

<androidx.cardview.widget.CardView

```

android:id="@+id/card_view_main_menu"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginVertical="8dp"
app:cardCornerRadius="22dp"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toBottomOf="parent">

```

```

<androidx.constraintlayout.widget.ConstraintLayout

```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="@style/RecyclerView">

```

```

<TextView

```

```

    android:id="@+id/book_title"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    style="@style/RecyclerView"
    android:font="@font/averia_libre_bold"
    android:paddingStart="15dp"
    android:paddingEnd="15dp"
    android:paddingTop="10dp"
    android:text="Default Book title"
    android:maxLines="1"
    android:ellipsize="end"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="26sp"
    app:layout_constraintBottom_toTopOf="@id/small_book_tag"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

```

<TextView

```

```

    android:id="@+id/small_book_tag"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    style="@style/RecyclerViewTitle"
    android:paddingStart="15dp"
    android:paddingTop="5dp"
    android:paddingBottom="20dp"
    android:text="Default Tag TXT"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/book_title" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

</androidx.cardview.widget.CardView>

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```



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

```
<androidx.cardview.widget.CardView
android:id="@+id/card_view_main_menu"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginVertical="8dp"
app:cardCornerRadius="22dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toBottomOf="parent">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
style="@style/RecyclerView">
```

```
<TextView
android:id="@+id/default_main_menu_name"
android:layout_width="0dp"
android:layout_height="wrap_content"
style="@style/RecyclerView"
android:font="@font/averia_libre_bold"
android:paddingStart="15dp"
android:paddingTop="10dp"
android:text="Default Menu TXT"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
android:textSize="28sp"
app:layout_constraintBottom_toTopOf="@id/small_title_main_menu"
app:layout_constraintEnd_toStartOf="@id/default_img_main_menu"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/small_title_main_menu"
android:layout_width="0dp"
android:layout_height="wrap_content"
style="@style/RecyclerViewTitle"
android:paddingStart="15dp"
android:paddingTop="5dp"
android:paddingBottom="20dp"
android:text="Default Menu TXT"
android:textSize="12sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@id/default_img_main_menu"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/default_main_menu_name" />
```

<ImageView

```
android:id="@+id/default_img_main_menu"
android:layout_width="72dp"
android:layout_height="74dp"
android:maxWidth="80dp"
android:maxHeight="80dp"
android:minWidth="46dp"
android:minHeight="48dp"
android:padding="10dp"
android:layout_marginEnd="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/default_main_menu_name"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@mipmap/ic_launcher" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

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

```
<CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/tag"
android:padding="6dp"
android:layout_marginBottom="4dp"
style="@style/mySpinnerItemStyle"
android:textSize="20sp">
```

```
</CheckedTextView>
```

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

```
<resources>
<color name="dark_green">#10451D</color>
<color name="light_green">#208B3A</color>
<color name="dark_turquoise">#6E8E8A</color>
<color name="turquoise">#B7EFC5</color>
<color name="dark_grey">#1E1F20</color>
<color name="light_grey">#747575</color>
<color name="yellow">#FFBA08</color>
<color name="stars">#FCBF49</color>
<color name="blue">#168AAD</color>
<color name="black">#FF000000</color>
<color name="white">#FFFFFF</color>
<color name="deep_blue">#041324</color>
```

```

<color name="dark_grey_night">#3D4042</color>
<color name="light_grey_night">#A8A8A8</color>
<color name="red">#EE082E</color>
</resources>

<resources>
  <string name="app_name" translatable="false">LisBook</string>
  <string name="categories">Categories</string>
  <string name="cat_item_none">None</string>
  <string name="cat_item_fantasy">Fantasy</string>
  <string name="cat_item_detectives">Detectives</string>
  <string name="cat_item_technical">Technical literature</string>
  <string name="cat_item_domestic_literature">Domestic literature</string>
  <string name="cat_item_world_literature">World literature</string>
  <string name="cat_item_to_buy">To buy</string>
  <string name="fav">Favourites</string>
  <string name="settings">Settings</string>
  <string name="statistics">Statistics</string>
  <string name="search">Search</string>
  <string name="all_b">All Books</string>
  <string name="new_b">New Book</string>
  <string name="new_c">New Category</string>
  <string name="change">Change</string>
  <string name="main">MAIN</string>
  <string name="save">SAVE</string>
  <string name="back">BACK</string>
  <string name="add">ADD</string>
  <string name="tag">Tag:</string>
  <string name="tag_new">New</string>
  <string name="tag_in_prog">In progress</string>
  <string name="tag_done">Done</string>
  <string name="title">Title</string>
  <string name="paste_url">Paste URL here</string>
  <string name="date_crt">Created:</string>
  <string name="date_end">End</string>
  <string name="note">Note</string>
  <string name="rating">Rating</string>
  <string name="quantity">Quantity:</string>
  <string name="find_in">Find in LisBook</string>
  <string name="see_prog">See progress</string>
  <string name="switch_theme">Switch theme Light/Dark</string>
  <string name="finished">Finished</string>
  <string name="letters_const">Constraint</string>
  <string name="category">Category:</string>
  <string name="error_create">Error creating!</string>
  <string name="success_create">Book successfully added!</string>
  <string name="success_update">Book successfully updated!</string>
  <string name="success_delete">Successfully deleted!</string>
  <string name="error_delete">Error deleting!</string>
  <string name="dialog_yes">YES</string>
  <string name="dialog_no">NO</string>

```

```

<string name="delete">Delete</string>
<string name="export">Export</string>
<string name="dialog_delete_txt">Are you sure you want to delete</string>
<string name="delete_all_txt">Delete all data from LisBook</string>
<string name="dialog_delete_all_txt">Are you sure you want to delete all data?</string>
<string name="dialog_export_txt">Are you sure you want to export your library to .csv?</string>
<string name="export_txt">Export your library to .csv file</string>
<string name="permission_granted">Storage Permission Granted</string>
<string name="permission_denied">Storage Permission Denied</string>
<string name="permission_has">Permission already granted</string>
<string name="success_export">Successfully exported!</string>
<string name="error_export">Error exporting!</string>

```

```

<string-array name="default_main_menu">
  <item>@string/categories</item>
  <item>@string/fav</item>
  <item>@string/statistics</item>
  <item>@string/search</item>
  <item>@string/all_b</item>
  <item>@string/finished</item>
</string-array>

```

```

<string-array name="small_title_main_menu">
  <item>@string/quantity</item>
  <item>@string/quantity</item>
  <item>@string/see_prog</item>
  <item>@string/find_in</item>
  <item>@string/quantity</item>
  <item>@string/quantity</item>
</string-array>

```

```

<string-array name="tag_spinner_items">
  <item>@string/tag_new</item>
  <item>@string/tag_in_prog</item>
  <item>@string/tag_done</item>
</string-array>

```

```

<string-array name="cat_spinner_items">
  <item>@string/cat_item_none</item>
  <item>@string/cat_item_fantasy</item>
  <item>@string/cat_item_detectives</item>
  <item>@string/cat_item_domestic_literature</item>
  <item>@string/cat_item_world_literature</item>
  <item>@string/cat_item_technical</item>
  <item>@string/cat_item_to_buy</item>
</string-array>

```

```
</resources>
```

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

```

<string name="app_name">LisBook</string>
<string name="categories" translatable="true">Категорії</string>
<string name="fav">Улюблені</string>
<string name="settings">Налаштування</string>
<string name="statistics">Статистика</string>
<string name="search">Пошук</string>
<string name="all_b">Усі книжки</string>
<string name="new_b">Нова Книга</string>
<string name="new_c">Нова Категорія</string>
<string name="change">Змінити</string>
<string name="main">ГОЛОВНА</string>
<string name="save">ЗБЕРЕГТИ</string>
<string name="back">НАЗАД</string>
<string name="add">ДОДАТИ</string>
<string name="tag">Тег:</string>
<string name="tag_new">Нова</string>
<string name="tag_in_prog">В процесі</string>
<string name="tag_done">Завершено</string>
<string name="title">Назва</string>
<string name="paste_url">Вставте посилання</string>
<string name="date crt">Додано:</string>
<string name="date_end">Закінчено</string>
<string name="note">Нотатка</string>
<string name="rating">Рейтинг</string>
<string name="quantity">Кількість</string>
<string name="find_in">Знайти в LisBook</string>
<string name="see_prog">Подивитись прогрес</string>
<string name="switch_theme">Змінити тему Світла/Темна</string>
<string name="finished">Завершені</string>
<string name="letters_const">Обмеження</string>
<string name="category">Категорія:</string>
<string name="error_create">Помилка додавання!</string>
<string name="success_create">Книга успішно додана!</string>
<string name="success_delete">Успішно видалено!</string>
<string name="error_delete">Помилка видалення!</string>
<string name="dialog_yes">ТАК</string>
<string name="dialog_no">НІ</string>
<string name="delete">Видалити</string>
<string name="dialog_delete_txt">Ви впевнені, що хочете видалити</string>
<string name="dialog_delete_all_txt">Ви впевнені, що хочете видалити усі дані?</string>
<string name="delete_all_txt">Видалити усі дані з LisBook</string>
<string name="cat_item_none">Немає</string>
<string name="cat_item_fantasy">Фантастика</string>
<string name="cat_item_detectives">Детективи</string>
<string name="cat_item_domestic_literature">Вітчизняна література</string>
<string name="cat_item_world_literature">Зарубіжна література</string>
<string name="cat_item_technical">Технічна література</string>
<string name="cat_item_to_buy">Придбати</string>
<string name="success_update">Книга успішно оновлена!</string>
<string name="export">Експорт</string>
<string name="dialog_export_txt">Ви впевнені, що хочете експортувати ваші дані з бібліотеки
у .csv файл?</string>

```

```

<string name="export_txt">Експорт бібліотек у .csv файл</string>
<string name="permission_has">Усі дозволи надано</string>
<string name="permission_granted">Дозвіл на збереження надано</string>
<string name="permission_denied">Дозвіл на збереження не отримано</string>
<string name="success_export">Успішно експортовано!</string>
<string name="error_export">Помилка експорту!</string>

```

```

<string-array name="default_main_menu">
  <item>@string/categories</item>
  <item>@string/fav</item>
  <item>@string/statistics</item>
  <item>@string/search</item>
  <item>@string/all_b</item>
  <item>@string/finished</item>
</string-array>

```

```

<string-array name="small_title_main_menu">
  <item>@string/quantity</item>
  <item>@string/quantity</item>
  <item>@string/see_prog</item>
  <item>@string/find_in</item>
  <item>@string/quantity</item>
  <item>@string/quantity</item>
</string-array>

```

```

<string-array name="tag_spinner_items">
  <item>@string/tag_new</item>
  <item>@string/tag_in_prog</item>
  <item>@string/tag_done</item>
</string-array>

```

```

<string-array name="cat_spinner_items">
  <item>@string/cat_item_none</item>
  <item>@string/cat_item_fantasy</item>
  <item>@string/cat_item_detectives</item>
  <item>@string/cat_item_domestic_literature</item>
  <item>@string/cat_item_world_literature</item>
  <item>@string/cat_item_technical</item>
  <item>@string/cat_item_to_buy</item>
</string-array>

```

```

</resources>

```

```

<resources xmlns:tools="http://schemas.android.com/tools">
  <style name="Theme.LisBook" parent="Theme.MaterialComponents.Light.NoActionBar">
    <item name="colorPrimary">@color/dark_green</item>
    <item name="colorPrimaryVariant">@color/light_green</item>
    <item name="colorOnPrimary">@color/white</item>
    <item name="colorSecondary">@color/dark_turquoise</item>
    <item name="colorSecondaryVariant">@color/turquoise</item>
    <item name="colorOnSecondary">@color/black</item>
  </style>

```

```

    <item name="android:statusBarColor">?attr/colorPrimary</item>
    <item name="colorAccent">@color/light_grey</item>
    <item name="colorPrimaryDark">@color/dark_grey</item>
</style>

<style name="Screen">
    <item name="android:background">@color/white</item>
</style>

<style name="ToolBar">
    <item name="android:background">@color/dark_green</item>
    <item name="android:textColor">@color/white</item>
</style>

<style name="AnyMainText">
    <item name="android:textColor">@color/dark_grey</item>
</style>

<style name="HintText">
    <item name="android:textColor">@color/dark_grey</item>
    <item name="android:textColorHint">@color/light_grey</item>
</style>

<style name="Button">
    <item name="android:textColor">@color/white</item>
    <item name="android:backgroundTint">@color/light_green</item>
</style>

<style name="ImgButton">
    <item name="android:background">@color/dark_green</item>
</style>

<style name="ImgButtonCat">
    <item name="android:background">@color/turquoise</item>
</style>

<style name="ImgWhiteLinkButton">
    <item name="android:background">@color/dark_turquoise</item>
</style>

<style name="ConstraintLayoutBookTitle">
    <item name="android:background">@color/dark_turquoise</item>
</style>

<style name="ImgPlusButton">
    <item name="android:background">@color/white</item>
</style>

<style name="QuantityTxtView">
    <item name="android:textColor">@color/white</item>
    <item name="android:background">@color/light_green</item>

```

```

</style>

<style name="BookNameTxtView">
    <item name="android:textColor">@color/dark_grey</item>
    <item name="android:background">@color/dark_turquoise</item>
</style>

<style name="RecyclerView">
    <item name="android:textColor">@color/dark_grey</item>
    <item name="android:background">@color/turquoise</item>
</style>

<style name="RecViewTitle">
    <item name="android:textColor">@color/light_grey</item>
</style>

<style name="SpinnerBackground">
    <item name="android:popupBackground">@color/white</item>
</style>

<style name="mySpinnerItemStyle">
    <item name="android:textColor">@color/dark_grey</item>
    <item name="android:background">@color/white</item>
</style>

<style name="Custom.OutlinedBox"
parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox"
tools:ignore="PrivateResource">
    <item name="android:background">@color/white</item>
    <item name="android:textColorHint">@color/light_grey</item>
    <item name="android:textColor">@color/dark_grey</item>
</style>

<style name="RatingBar" parent="Theme.AppCompat">
    <item name="colorControlNormal">@color/light_grey</item>
    <item name="colorControlActivated">@color/stars</item>
</style>
</resources>

<resources xmlns:tools="http://schemas.android.com/tools">
    <style name="Theme.LisBookt" parent="ThemeOverlay.MaterialComponents.Dark">
        <item name="colorPrimary">@color/dark_grey</item>
        <item name="colorPrimaryVariant">@color/light_grey</item>
        <item name="colorOnPrimary">@color/black</item>
        <item name="colorSecondary">@color/dark_grey_night</item>
        <item name="colorSecondaryVariant">@color/light_grey_night</item>
        <item name="colorOnSecondary">@color/deep_blue</item>
        <item name="colorAccent">@color/white</item>
    </style>

    <style name="Screen">

```



```

    <item name="android:background">@color/deep_blue</item>
</style>

<style name="ToolBar">
    <item name="android:background">@color/dark_grey</item>
    <item name="android:textColor">@color/white</item>
</style>

<style name="AnyMainText">
    <item name="android:textColor">@color/white</item>
</style>

<style name="HintText">
    <item name="android:textColor">@color/white</item>
    <item name="android:textColorHint">@color/white</item>
</style>

<style name="Button">
    <item name="android:textColor">@color/white</item>
    <item name="android:backgroundTint">@color/dark_grey</item>
</style>

<style name="ImgButton">
    <item name="android:background">@color/dark_grey</item>
</style>

<style name="ImgButtonCat">
    <item name="android:background">@color/dark_grey</item>
</style>

<style name="ImgPlusButton">
    <item name="android:background">@color/deep_blue</item>
</style>

<style name="ImgWhiteLinkButton">
    <item name="android:background">@color/dark_grey</item>
</style>

<style name="ConstraintLayoutBookTitle">
    <item name="android:background">@color/dark_grey</item>
</style>

<style name="QuantityTxtView">
    <item name="android:textColor">@color/white</item>
    <item name="android:background">@color/dark_grey</item>
</style>

<style name="BookNameTxtView">
    <item name="android:textColor">@color/white</item>
    <item name="android:background">@color/dark_grey_night</item>
</style>

```

```

<style name="RecyclerView">
    <item name="android:textColor">@color/white</item>
    <item name="android:background">@color/dark_grey_night</item>
</style>

<style name="RecViewTitle">
    <item name="android:textColor">@color/light_grey_night</item>
</style>

<style name="SpinnerBackground">
    <item name="android:popupBackground">@color/dark_grey_night</item>
</style>

<style name="mySpinnerItemStyle">
    <item name="android:textColor">@color/white</item>
    <item name="android:background">@color/deep_blue</item>
</style>

<style name="Custom.OutlinedBox"
parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox"
tools:ignore="PrivateResource">
    <item name="android:background">@color/deep_blue</item>
    <item name="android:textColorHint">@color/white</item>
    <item name="android:textColor">@color/white</item>
</style>

<style name="RatingBar" parent="Theme.AppCompat">
    <item name="colorControlNormal">@color/light_grey</item>
    <item name="colorControlActivated">@color/stars</item>
</style>
</resources>
plugins {
    id 'com.android.application'
}

android {
    compileSdk 33

    defaultConfig {
        applicationId "com.liakh.lisbook"
        minSdk 23
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

```

```
    }  
  }  
  compileOptions {  
    sourceCompatibility JavaVersion.VERSION_1_8  
    targetCompatibility JavaVersion.VERSION_1_8  
  }  
}  
  
dependencies {  
  
  implementation 'androidx.appcompat:appcompat:1.6.1'  
  implementation 'com.google.android.material:material:1.8.0'  
  implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
  testImplementation 'junit:junit:4.13.2'  
  androidTestImplementation 'androidx.test.ext:junit:1.1.5'  
  androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'  
  
  implementation 'com.google.android.gms:play-services-ads:19.3.0'  
}
```