МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт

з лабораторної роботи № 5 з дисципліни

Програмування під Android

Виконав: Перевірив:

ст. гр. ПЗПІ-23-4 ст. викладач кафедри ПІ

Юзков Олександр Сокорчук Ігор Петрович

Харків 2024

ЗАВДАННЯ

Доповнити завдання №4 наступними функціями:

реалізувати різне відображення для портретної та ландшафтної орієнтації екрану за допомогою тих самих фрагментів;

створити дві редаговані теми (тло, гарнітура шрифту, колір тексту тощо) і застосовувати їх в залежності від налаштувань програми.

додати можливість задавати в налаштуваннях програми розмір шрифту.

Переробити завдання №4 так, щоб дані нотаток зберігалися у базі даних SQLite

ХІД РОБОТИ

Продовжуємо працювати над лабораторною роботою №4. Почнемо з першого завдання: створення окремих макетів для портретної та ландшафтної орієнтацій екрана з використанням однакових фрагментів.

Для цього потрібно відкрити файли розмітки та створити нові фрагменти, які будуть використовуватися для відображення списку, а також сторінок додавання та редагування нотаток. Основна різниця між портретною та ландшафтною орієнтаціями полягатиме в розташуванні елементів інтерфейсу.

Наприклад, на головній сторінці в ландшафтному режимі кнопка додавання буде розташована у правій колонці, тоді як у портретному — внизу екрана. На сторінках додавання та редагування блок для роботи із зображенням переміститься до правої колонки.

Повний код розмітки наведено в [ДОДАТОК А].

Перейдемо до другого пункту роботи: редаговані теми. Для цього створимо темну тему.

<resources xmlns:tools="http://schemas.android.com/tools">

<!-- Base application theme. -->

<style name="Theme.Androidpzpi234YuzkovOleksandrlabtask5" parent="Theme.MaterialComponents.DayNight.DarkActionBar">

<!-- Primary brand color. -->

<item name="colorPrimary">@color/blue\_light</item>

<item name="colorPrimaryVariant">@color/blue\_dark</item>

<item name="colorOnPrimary">@color/black</item>

<!-- Secondary brand color. -->

<item name="colorSecondary">@color/green\_dark</item>

<item name="colorSecondaryVariant">@color/green\_dark</item>

<item name="colorOnSecondary">@color/black</item>

<!-- Status bar color. -->

<item name="android:statusBarColor">?attr/colorPrimaryVariant</item>

<!-- Customize your theme here. -->

</style>

</resources>

Далі зазначимо кольори:

<resources>

<color name="blue\_light">#FFBB86FC</color>

<color name="blue">#FF6200EE</color>

<color name="blue\_dark">#FF3700B3</color>

<color name="green\_light">#FF03DAC5</color>

<color name="green\_dark">#FF018786</color>

<color name="black">#FF000000</color>

<color name="white">#FFFFFFFF</color>

<color name="primary\_light">#6200EE</color>

<color name="primary\_dark\_light">#3700B3</color>

<color name="accent\_light">#03DAC5</color>

<color name="background\_light">#FFFFFF</color>

<color name="text\_light">#000000</color>

<color name="primary\_dark">#BB86FC</color>

<color name="primary\_dark\_dark">#3700B3</color>

<color name="accent\_dark">#03DAC5</color>

<color name="background\_dark">#121212</color>

<color name="text\_dark">#FFFFFF</color>

</resources>

Тепер третій пункт, що треба було додати: можливість задавати в налаштуваннях програми розмір шрифту.

Для зміни шрифту використується метод applyFontSize(), який налаштовує розмір шрифтів для елементів у вашому RecyclerView. В залежності від значення isFontLarge, шрифт буде більший або стандартний:

private void applyFontSize() {

float fontSize = isFontLarge ? 24f : 16f;

RecyclerView recyclerView = findViewById(R.id.recyclerViewNotes);

for (int i = 0; i < recyclerView.getChildCount(); i++) {

RecyclerView.ViewHolder holder = recyclerView.findViewHolderForAdapterPosition(i);

if (holder != null) {

TextView title = holder.itemView.findViewById(R.id.title);

TextView description = holder.itemView.findViewById(R.id.description);

if (title != null) title.setTextSize(fontSize);

if (description != null) description.setTextSize(fontSize);

}

}

}

Тепер перейдемо до зберігання даних в базі даних.

Для цього скористаємося класом DatabaseHelper, що розширює SQLiteOpenHelper. В ньюму прописуємо створення бази та логіку керування даними [ДОДАТОК А]

Таким чином, всі завдання було виконано.

ВИСНОВКИ

Під час виконання лабораторної роботи я здобув досвід роботи з базою даних SQLite в рамках повноцінного Android проєкту. Я закріпив свої знання з тем і стилів, зокрема у зміні кольорів, шрифтів та інших елементів інтерфейсу по всьому застосунку.

Посилання на відео-звіт завантажений на YouTube: https://youtu.be/YCgwqZNlX1U

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Документація Andoid developers SQLiteDatabase. URL: <https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase> (дата звернення: 10.01.2025).
2. Документація Andoid developers Save data using SQLite. URL: <https://developer.android.com/training/data-storage/sqlite> (дата звернення: 05.12.2024).
3. Документація JavaRush з роботи з базами даних. URL: <https://javarush.com/ua/quests/lectures/ua.questhibernate.level07.lecture00> (дата звернення: 10.01.2025).
4. Документація ExamClouds з локалізації в Java. URL: <https://www.examclouds.com/java/java-core-russian/internationalisation> (дата звернення: 10.01.2025).

ДОДАТОК А

Код програми

Увесь код програми можно знайти за посиланням у GitHub:

https://github.com/NureIuzkovOleksandr/android-pzpi-23-4-iuzkov-oleksandr