МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт

з лабораторної роботи № 4 з дисципліни

Програмування під Android

Виконав: Перевірив:

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

Прядко Максим Сокорчук Ігор Петрович

Харків 2024

ЗАВДАННЯ

Розробити застосунок для ведення нотаток.

Застосунок повинен дозволяти переглядати список нотаток, створювати нову нотатку, редагувати нотатку та видаляти нотатку. Кожна нотатка повинна мати назву, опис, важливість (три класи), час і дату призначення і картинку (вибирається з галереї).

У списку нотатки відображаються з іконкою з галереї і часом створення (праворуч зверху більш дрібним шрифтом), а також графічним відображенням класу важливості (іконка з множини).

Потрібно реалізувати функції додавання, видалення, редагування (без збереження між перезапуск додатку) нотаток.

Видалення/редагування викликаються з контекстного меню при довгому натисненні, додавання - з основного меню в ActionBar. У ActionBar реалізувати пошук / фільтрація заміток за змістом тексту, а також фільтрацію за важливістю. Текстові елементи користувацького інтерфейсу повинні бути локалізовані українською та англійською мовами.

ХІД РОБОТИ

Створюємо новий проєкт та обираємо Empty Views Activity.

Використаємо ActionBar для розміщення елементів інтерфейсу для взаємодії.

Використаємо елемент інтерфейсу SearchView для пошуку заміток.

Використаємо Spinner для обрання рівня важливості замітки.

Використаємо RecyclerView для відображення списку заміток.

Відображення заміток у RecyclerView.  
Код розмітки можна буде переглянути у ДОДАТКУ А

Створимо клас Note, який буде містити всю інформацію про замітку ДОДАТОК Б.

Створимо розмітку note\_item, яка відображає один запис у списку ДОДАТОК В.

Створимо NoteAdapter, який буде наслідувати клас RecyclerView.Adapter<NoteAdapter.NoteViewHolder>.

Перевизначимо метод onCreateViewHolder для отримання потрібної розмітки через LayoutInflater.

Перевизначимо метод onBindViewHolder для прив'язки даних до елементів розмітки.

Для holder.itemView перевизначимо події setOnClickListener і setOnCreateContextMenuListener для відображення запису і створення контекстного меню для редагування і видалення.

Фрагмент коду для реалізації методу onBindViewHolder для NoteAdapter:

1. @Override  
2. public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) 3. {  
4. Note note = notes.get(position);  
5. holder.titleTextView.setText(note.getTitle());  
6 holder.descriptionTextView.setText(note.getDescription());  
7 byte[] imageBytes = *StandartIconCheck*(note.getImageData(), context);  
8 9.holder.iconImageView.setImageBitmap(BitmapFactory.*decodeByteArray*(imageBy10. tes, 0, imageBytes.length));  
11 int importance = note.getImportance();  
12 int imageSRC;  
13 if (importance == 0){  
14 imageSRC = R.drawable.*ic\_low\_importance*;  
15 } else if (importance == 1){  
16 imageSRC = R.drawable.*ic\_medium\_importance*;  
17 } else {  
18 imageSRC = R.drawable.*ic\_high\_importance*;  
19 }  
20 holder.importanceImageView.setImageResource(imageSRC);  
21 holder.creationDateTextView.setText(note.getCreationDate());  
22  
23 holder.itemView.setOnCreateContextMenuListener(new 24.View.OnCreateContextMenuListener() {  
25 @Override  
26 public void onCreateContextMenu(ContextMenu menu, View v, 27.ContextMenu.ContextMenuInfo menuInfo) {  
28. menu.add(0, R.id.*edit*, 0, R.string.*edit*);  
29. menu.add(1, R.id.*delete*, 0, R.string.*delete*);  
30. currentIndex = holder.getAdapterPosition();  
31. }  
32. });

Створення, редагування і перегляд розмітки

Реалізуємо розмітку activity\_create\_and\_edit\_note для створення, перегляду і редагування розмітки.

Створимо діалогові вікна для обрання дати і часу.

Створимо перевірку, щоб при створенні або редагуванні запис його назва не була пустою.

Реалізуємо обрання зображення

Фрагмент коду для створення або редагування запису:

1. createOrEditButton.setOnClickListener(new View.OnClickListener() {

2. @Override

3. public void onClick(View v) {

4. if (titleEditTextButton.getText().toString().isEmpty()){

5. AlertDialog.Builder builder = new AlertDialog.Builder(CreateAndEditNoteActivity.this);

6. builder.setTitle(R.string.incorrectInput);

7. builder.setMessage(R.string.noTitle);

8. builder.setPositiveButton("Ок", new DialogInterface.OnClickListener() {

9. @Override

10. public void onClick(DialogInterface dialog, int which) {

11.

12. }

13. });

14. builder.create().show();

15. return;

16. }

17.

18. if (getIntent().getBooleanExtra("edit", false)){

19. Note note = notes.get(getIntent().getIntExtra("index", 0));

20. Notes.EditNote( note.getNumber(),

21. titleEditTextButton.getText().toString(),

22. descriptionEditTextButton.getText().toString(),

23. importanceSpinner.getSelectedItemPosition(),

24. editTextDate.getText().toString(),

25. editTextTime.getText().toString(),

26. note.getCreationDate(),

27. imageBytes);

28. } else {

29. Notes.AddNote(titleEditTextButton.getText().toString(),

30. descriptionEditTextButton.getText().toString(),

31. importanceSpinner.getSelectedItemPosition(),

32. editTextDate.getText().toString(),

33. editTextTime.getText().toString(),

33. dayOfMonth + "." + month + "." + year,

34. imageBytes);

35. }

36. setResult(RESULT\_OK);

37. finish();

38. }

39. });

Створимо клас DBHelper

Перевизначимо метод onCreate для створення нової бази даних.

Створимо методи для додавання, редагування і вилучення запису з бази даних.

Створимо метод для фільтрації записів. ДОДАТОК Г

Створимо клас ImportanceAdapter ДОДАТОК Д

Створимо клас ImportantItem, який містить в собі назву і зображення важливості.

Для відображення важливості запису створимо розмітку spinner\_item.

ВИСНОВКИ

Під час виконання лабораторної роботи №4, я навчився реалізовувати повноцінний застосунок для введення нотаток. Я навчився використовувати різні елементи інтерфейсу, такі як SearchView, Spinner і RecyclerView, налаштував адаптери для відображення даних в інтерфейсі. Також, навчився створювати різні діалогові вікна і передавати значення через Intent.

Відеозвіт на YouTube. URL: <https://youtu.be/T2q_0qPpDiU>

ДОДАТОК А

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/white"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="2"  
 android:text="@string/notes"  
 android:textColor="@color/black"  
 android:textSize="20dp" />  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/noteSearchView"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="4"  
 app:queryHint="@string/search" />  
  
 <Spinner  
 android:id="@+id/importanceFilter"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="3"  
 android:minHeight="48dp" />  
  
 <Button  
 android:id="@+id/newNoteButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="2"  
 android:backgroundTint="@color/purple"  
 android:text="+" />  
 </LinearLayout>  
  
 </androidx.appcompat.widget.Toolbar>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/notesRecycleView"/>  
  
  
</LinearLayout>

ДОДАТОК Б

package nure.priadko.maksym;  
  
import android.net.Uri;  
  
public class Note {  
  
 private int number;  
  
 private String title;  
  
 private String description;  
  
 private int importance;  
  
 private String eventDate;  
  
 private String eventTime;  
  
 private String creationDate;  
  
 private byte[] imageData;  
  
 public Note(int number, String title, String description, int importance, String eventDate, String eventTime, String creationDate, byte[] imageData){  
 this.number = number;  
 this.title = title;  
 this.description = description;  
 this.importance = importance;  
 this.eventDate = eventDate;  
 this.eventTime = eventTime;  
 this.creationDate = creationDate;  
 this.imageData = imageData;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public int getImportance() {  
 return importance;  
 }  
  
 public String getEventDate() {  
 return eventDate;  
 }  
  
 public String getEventTime() {  
 return eventTime;  
 }  
  
 public String getCreationDate() {  
 return creationDate;  
 }  
  
 public byte[] getImageData() {  
 return imageData;  
 }  
  
 public int getNumber() {  
 return number;  
 }  
}

ДОДАТОК В

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/iconImageView"  
 android:layout\_width="0dp"  
 android:layout\_height="100dp"  
 android:layout\_weight="1"  
 android:src="@drawable/ic\_launcher\_background"  
 android:layout\_margin="8dp"  
 />  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="100dp"  
 android:orientation="vertical"  
 android:layout\_weight="3">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1">  
  
 <TextView  
 android:id="@+id/titleTextView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="3"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:text="ііііssss"  
 android:textSize="14sp" />  
  
 <ImageView  
 android:id="@+id/importanceImageView"  
 android:layout\_width="0dp"  
 android:layout\_height="40dp"  
 android:layout\_weight="1"  
 android:src="@drawable/ic\_launcher\_background"  
 />  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/descriptionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Ts"  
 android:textSize="20dp"  
 android:ellipsize="end"  
 android:maxLines="2"/>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/creationDateTextView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="TextView" />  
</LinearLayout>

ДОДАТОК Г

DBhelper  
  
package nure.priadko.maksym;  
  
import android.annotation.SuppressLint;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import androidx.annotation.Nullable;  
  
public class DBHelper extends SQLiteOpenHelper {  
  
  
 public DBHelper(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {  
 super(context, name, factory, version);  
 }  
  
 @SuppressLint("SQLiteString")  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("CREATE TABLE notes(number INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT, description TEXT," +  
 " importance INTEGER, event\_date TEXT, event\_time TEXT, creation\_date TEXT, image\_data BLOB)");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 }  
  
 public void Drop(){  
 SQLiteDatabase db = getWritableDatabase();  
 db.execSQL("DROP TABLE notes");  
 onCreate(db);  
 }  
  
 public void onInsert(String title, String description, int importance, String eventDate, String eventTime, String creationDate, byte[] imageData){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = CreateNote(title, description, importance, eventDate, eventTime, creationDate, imageData);  
 db.insert("notes", null, contentValues);  
 }  
  
 public void onUpdate(int number, String title, String description, int importance, String eventDate, String eventTime, String creationDate, byte[] imageData){  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues contentValues = CreateNote(title, description, importance, eventDate, eventTime, creationDate, imageData);  
 db.update("notes", contentValues, "number = ?", new String[]{String.*valueOf*(number)});  
 }  
  
 public ContentValues CreateNote(String title, String description, int importance, String eventDate, String eventTime, String creationDate, byte[] imageData){  
 ContentValues contentValues = new ContentValues();  
 contentValues.put("title", title);  
 contentValues.put("description", description);  
 contentValues.put("importance", importance);  
 contentValues.put("event\_date", eventDate);  
 contentValues.put("event\_time", eventTime);  
 contentValues.put("creation\_date", creationDate);  
 contentValues.put("image\_data", imageData);  
 return contentValues;  
 }  
  
 public void onDelete(int number){  
 SQLiteDatabase db = getWritableDatabase();  
 db.delete("notes", "number = ?", new String[]{String.*valueOf*(number)});  
 }  
  
 public Cursor onSelect(){  
 SQLiteDatabase db = this.getReadableDatabase();  
 return db.query("notes", null, null, null, null, null, null);  
 }  
  
 public Cursor Filter(String title, String description, boolean importance, int importanceLayer){  
 SQLiteDatabase db = this.getReadableDatabase();  
 String query = "SELECT \* FROM notes WHERE (title LIKE ? OR description LIKE ?)";  
 String[] selectionArgs = new String[]{"%" + title + "%", "%" + description + "%"};  
 if (importance){  
 query += "AND importance = ?";  
 selectionArgs = new String[]{"%" + title + "%", "%" + description + "%", String.*valueOf*(importanceLayer)};  
 }  
 return db.rawQuery(query, selectionArgs);  
 }  
}

ДОДАТОК Д

ImportanceAdapter  
  
package nure.priadko.maksym;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
  
import java.util.List;  
  
public class ImportanceAdapter extends ArrayAdapter<ImportanceItem> {  
  
 public ImportanceAdapter(@NonNull Context context, List<ImportanceItem> resource) {  
 super(context, 0, resource);  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent){  
 ImportanceItem item = getItem(position);  
  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.*spinner\_item*, parent, false);  
  
 ImageView imageView = convertView.findViewById(R.id.*importanceIcon*);  
 TextView textView = convertView.findViewById(R.id.*importanceTextView*);  
  
 if (item.GetImageResId() != null){  
 imageView.setImageResource(item.GetImageResId());  
 }  
  
 textView.setText(item.GetTitle());  
  
 return convertView;  
 }  
  
 @Override  
 public View getDropDownView(int position, View convertView, ViewGroup parent){  
 return getView(position, convertView, parent);  
 }  
}