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-23-2 ст. викладач кафедри ПІ

Горбатюк Єгор Андрійович 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

Так як лабораторна робота дуже обширна і велика, у ході роботи буде перелічено основний функціонал і призначення методів і полів классів. Повний код проекта можна подивитися або у додатках, або на GitHub.

MainActivity.java

Головна активність додатку, яка відповідає за відображення списку нотаток та основну навігацію.

Основні елементи:

* recyclerView — компонент для відображення списку нотаток.
* noteAdapter — адаптер для управління виглядом нотаток у списку.
* noteList — список нотаток, що відображаються.
* allNotes — повний перелік всіх нотаток.
* buttonSearch і buttonAddNote — кнопки для пошуку та додавання нотаток.
* searchView і searchDialog — інструменти для пошуку.
* currentSearchQuery — поточний текст пошукового запиту.

Основні методи:

1. onCreate()
   * Ініціалізує елементи інтерфейсу.
   * Налаштовує RecyclerView та адаптер.
   * Створює початковий список нотаток.
   * Встановлює обробники подій для кнопок.
2. showSearchDialog()
   * Відображає діалогове вікно для пошуку.
   * Налаштовує роботу SearchView.
   * Обробляє введені пошукові запити.
3. onNoteClick()
   * Реакція на вибір нотатки у списку.
   * Відображає контекстне меню з опціями.
4. editNote() і deleteNote()
   * Відповідають за редагування та видалення нотаток відповідно.
5. filterNotes() і restoreAllNotes()
   * Фільтрують нотатки за пошуковим запитом.
   * Повертають повний список нотаток після завершення пошуку.

EditNoteActivity.java

Активність для створення та редагування окремих нотаток.

Основні елементи:

* titleEditText — поле для введення заголовка.
* descriptionEditText — поле для опису нотатки.
* prioritySpinner — випадаючий список для вибору пріоритету.
* noteImageView — компонент для відображення обраного зображення.
* chooseImageButton — кнопка вибору зображення.
* imagePath — шлях до вибраного зображення.

Основні методи:

1. onCreate()
   * Ініціалізує інтерфейсні компоненти.
   * Налаштовує випадаючий список для пріоритету.
   * Заповнює поля при редагуванні існуючої нотатки.
   * Встановлює обробники подій для кнопок.
2. openGallery()
   * Відкриває галерею для вибору зображення.
3. onActivityResult()
   * Обробляє вибір зображення та оновлює його відображення.
4. saveNote()
   * Збирає дані з усіх полів.
   * Формує об’єкт нотатки з датою/часом.
   * Повертає результат в головну активність.

Note.java

Клас-модель, що описує структуру нотатки.

Поля:

* title — заголовок.
* description — опис.
* priority — пріоритет (1-3).
* dateTime — дата і час створення.
* imagePath — шлях до зображення.

Методи:

* Конструктор для створення нотаток.
* Геттери і сеттери для доступу до полів.
* equals() — для порівняння нотаток.
* hashCode() — для генерації хеш-коду.

NoteAdapter.java

Адаптер, що відповідає за відображення нотаток у RecyclerView.

Основні функції:

1. onCreateViewHolder()

* Створює нові об’єкти ViewHolder для окремих нотаток.

1. onBindViewHolder()
   * Наповнює елементи інтерфейсу даними нотатки.
   * Відображає пріоритет та зображення.
2. NoteViewHolder
   * Зберігає посилання на компоненти інтерфейсу.
   * Обробляє натискання на нотатку.

XML-файли:

1. activity\_main.xml
   * Головний інтерфейс з RecyclerView.
   * Панель управління з кнопками для пошуку та додавання нотаток.
2. activity\_edit\_note.xml
   * Поля для введення заголовка, опису та вибору пріоритету.
   * Елемент для відображення та вибору зображення.
   * Кнопки збереження та скасування.
3. item\_note.xml
   * Шаблон для відображення окремої нотатки у списку.
   * Включає заголовок, опис, пріоритет та зображення.
4. note\_options\_menu.xml
   * Контекстне меню з опціями редагування та видалення нотатки.

ВИСНОВКИ

Під час виконання лабораторної роботи було створено додаток для роботи з нотатками. Він підтримує базовий набір функцій: створення, редагування, видалення, пошук нотаток, а також встановлення пріоритетів.

Посилання на відеозвіт: <https://youtu.be/VhsBeEQLRGw>

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Курс «Програмування під андроід». Dl nure:
2. Офіційний сайт Android developer: <https://developer.android.com/>

ДОДАТОК А

package com.example.labtask4;  
  
import android.app.AlertDialog;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.ImageButton;  
import android.widget.SearchView;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.PopupMenu;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
import java.util.List;  
  
  
  
// 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.  
  
public class MainActivity extends AppCompatActivity implements NoteAdapter.OnNoteListener {  
  
 private RecyclerView recyclerView;  
 private NoteAdapter noteAdapter;  
 private List<Note> noteList;  
 private List<Note> allNotes;  
 private ImageButton buttonSearch;  
 private ImageButton buttonAddNote;  
 private SearchView searchView;  
 private AlertDialog searchDialog;  
 private String currentSearchQuery = "";  
 private static final int *REQUEST\_CODE\_ADD\_NOTE* = 1;  
 private static final int *REQUEST\_CODE\_EDIT\_NOTE* = 2;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 recyclerView = findViewById(R.id.*recycler\_view*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 noteList = new ArrayList<>();  
 allNotes = new ArrayList<>();  
 noteAdapter = new NoteAdapter(noteList, this);  
 recyclerView.setAdapter(noteAdapter);  
  
 buttonSearch = findViewById(R.id.*button\_search*);  
 buttonAddNote = findViewById(R.id.*button\_add\_note*);  
  
 buttonSearch.setOnClickListener(v -> {  
 showSearchDialog();  
 });  
  
 buttonAddNote.setOnClickListener(v -> {  
 Intent intent = new Intent(MainActivity.this, EditNoteActivity.class);  
 startActivityForResult(intent, *REQUEST\_CODE\_ADD\_NOTE*);  
 });  
 }  
  
 @Override  
 public void onNoteClick(int position) {  
 showNoteOptions(position);  
 }  
  
 private void showSearchDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 searchView = new SearchView(this);  
 searchView.setIconified(false);  
 searchView.setQuery(currentSearchQuery, false);  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 currentSearchQuery = query;  
 filterNotes(query);  
 return true;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
 currentSearchQuery = newText;  
 filterNotes(newText);  
 return true;  
 }  
 });  
 builder.setView(searchView);  
 searchDialog = builder.create();  
 searchDialog.setOnDismissListener(dialog -> {  
 if (currentSearchQuery.isEmpty()) {  
 restoreAllNotes();  
 }  
 });  
 searchDialog.show();  
 }  
  
 private void showNoteOptions(int position) {  
 View view = recyclerView.findViewHolderForAdapterPosition(position).itemView;  
 PopupMenu popup = new PopupMenu(this, view);  
 popup.inflate(R.menu.*note\_options\_menu*);  
  
 popup.setOnMenuItemClickListener(item -> {  
 if (item.getItemId() == R.id.*action\_edit*) {  
 editNote(position);  
 return true;  
 } else if (item.getItemId() == R.id.*action\_delete*) {  
 deleteNote(position);  
 return true;  
 }  
 return false;  
 });  
  
 popup.show();  
 }  
  
 private void editNote(int position) {  
 Note note = noteList.get(position);  
 Intent intent = new Intent(this, EditNoteActivity.class);  
 intent.putExtra("title", note.getTitle());  
 intent.putExtra("description", note.getDescription());  
 intent.putExtra("priority", note.getPriority());  
 intent.putExtra("imagePath", note.getImagePath());  
  
 intent.putExtra(EditNoteActivity.*EXTRA\_NOTE\_POSITION*, position);  
  
 startActivityForResult(intent, *REQUEST\_CODE\_EDIT\_NOTE*);  
 }  
  
 private void deleteNote(int position) {  
 if (position >= 0 && position < noteList.size()) {  
 Note noteToRemove = noteList.get(position);  
 noteList.remove(position);  
 allNotes.remove(noteToRemove);  
 noteAdapter.notifyItemRemoved(position);  
 } else {  
 }  
 }  
  
 private void filterNotes(String query) {  
 currentSearchQuery = query;  
 List<Note> filteredList = new ArrayList<>();  
 for (Note note : allNotes) {  
 if (note.getTitle().toLowerCase().contains(query.toLowerCase()) ||  
 note.getDescription().toLowerCase().contains(query.toLowerCase())) {  
 filteredList.add(note);  
 }  
 }  
 noteList.clear();  
 noteList.addAll(filteredList);  
 noteAdapter.notifyDataSetChanged();  
 }  
  
 private void restoreAllNotes() {  
 currentSearchQuery = "";  
 noteList.clear();  
 noteList.addAll(allNotes);  
 noteAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if (resultCode == *RESULT\_OK* && data != null) {  
 String title = data.getStringExtra("title");  
 String description = data.getStringExtra("description");  
 int priority = data.getIntExtra("priority", 1);  
 String dateTime = data.getStringExtra("dateTime");  
 String imagePath = data.getStringExtra("imagePath");  
  
 if (requestCode == *REQUEST\_CODE\_ADD\_NOTE*) {  
 Note newNote = new Note(title, description, priority, dateTime, imagePath);  
 allNotes.add(0, newNote);  
 noteList.add(0, newNote);  
 noteAdapter.notifyItemInserted(0);  
 recyclerView.smoothScrollToPosition(0);  
 }  
 else if (requestCode == *REQUEST\_CODE\_EDIT\_NOTE*) {  
 int position = data.getIntExtra(EditNoteActivity.*EXTRA\_NOTE\_POSITION*, -1);  
 if (position != -1) {  
 Note updatedNote = new Note(title, description, priority, dateTime, imagePath);  
  
 noteList.set(position, updatedNote);  
  
 if (position < allNotes.size()) {  
 allNotes.set(position, updatedNote);  
 }  
 noteAdapter.notifyDataSetChanged();  
 }  
 }  
 }  
 }  
}

ДОДАТОК Б

package com.example.labtask4;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Spinner;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Calendar;  
import java.util.Locale;  
  
public class EditNoteActivity extends AppCompatActivity {  
 private EditText titleEditText;  
 private EditText descriptionEditText;  
 private Spinner prioritySpinner;  
 private ImageView noteImageView;  
 private Button chooseImageButton;  
 private String imagePath = "";  
 public static final String *EXTRA\_NOTE\_POSITION* = "note\_position";  
 private static final int *REQUEST\_CODE\_PICK\_IMAGE* = 3;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_edit\_note*);  
  
 titleEditText = findViewById(R.id.*edit\_note\_title*);  
 descriptionEditText = findViewById(R.id.*edit\_note\_description*);  
 prioritySpinner = findViewById(R.id.*edit\_note\_priority*);  
 noteImageView = findViewById(R.id.*note\_image\_view*);  
 chooseImageButton = findViewById(R.id.*button\_choose\_image*);  
 Button saveButton = findViewById(R.id.*button\_save*);  
  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*priorities*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 prioritySpinner.setAdapter(adapter);  
  
 Intent intent = getIntent();  
 if (intent.hasExtra("title")) {  
 titleEditText.setText(intent.getStringExtra("title"));  
 descriptionEditText.setText(intent.getStringExtra("description"));  
 prioritySpinner.setSelection(intent.getIntExtra("priority", 0) - 1);  
  
 String receivedImagePath = intent.getStringExtra("imagePath");  
 if (receivedImagePath != null && !receivedImagePath.isEmpty()) {  
 imagePath = receivedImagePath;  
 try {  
  
 Uri imageUri = Uri.*parse*(imagePath);  
 noteImageView.setImageURI(imageUri);  
 } catch (Exception e) {  
 noteImageView.setImageResource(android.R.drawable.*ic\_menu\_gallery*);  
 imagePath = "";  
 }  
 }  
 }  
  
 chooseImageButton.setOnClickListener(v -> openGallery());  
 saveButton.setOnClickListener(v -> saveNote());  
 }  
  
 private void openGallery() {  
 Intent intent = new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 startActivityForResult(intent, *REQUEST\_CODE\_PICK\_IMAGE*);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == *RESULT\_OK* && requestCode == *REQUEST\_CODE\_PICK\_IMAGE* && data != null) {  
 Uri selectedImageUri = data.getData();  
 if (selectedImageUri != null) {  
 imagePath = selectedImageUri.toString();  
 noteImageView.setImageURI(null);  
 noteImageView.setImageURI(selectedImageUri);  
 }  
 }  
 }  
  
 private void saveNote() {  
 String title = titleEditText.getText().toString();  
 String description = descriptionEditText.getText().toString();  
 int priority = prioritySpinner.getSelectedItemPosition() + 1;  
  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.*getDefault*());  
  
 Calendar calendar = Calendar.*getInstance*();  
  
 calendar.add(Calendar.*HOUR\_OF\_DAY*, 2);  
  
 String currentDateTime = sdf.format(calendar.getTime());  
  
  
 int position = getIntent().getIntExtra(*EXTRA\_NOTE\_POSITION*, -1);  
  
 Intent resultIntent = new Intent();  
 resultIntent.putExtra("title", title);  
 resultIntent.putExtra("description", description);  
 resultIntent.putExtra("priority", priority);  
 resultIntent.putExtra("dateTime", currentDateTime);  
 resultIntent.putExtra("imagePath", imagePath);  
 resultIntent.putExtra(*EXTRA\_NOTE\_POSITION*, position);  
  
 setResult(*RESULT\_OK*, resultIntent);  
 finish();  
 }  
}

ДОДАТОК В

package com.example.labtask4;  
  
public class Note {  
 private String title;  
 private String description;  
 private int priority;  
 private String dateTime;  
 private String imagePath;  
  
 public Note(String title, String description, int priority, String dateTime, String imagePath) {  
 this.title = title;  
 this.description = description;  
 this.priority = priority;  
 this.dateTime = dateTime;  
 this.imagePath = imagePath;  
 }  
  
 // Геттери та сеттери  
 public String getTitle() { return title; }  
 public void setTitle(String title) { this.title = title; }  
 public String getDescription() { return description; }  
 public void setDescription(String description) { this.description = description; }  
 public int getPriority() { return priority; }  
 public void setPriority(int priority) { this.priority = priority; }  
 public String getDateTime() { return dateTime; }  
 public void setDateTime(String dateTime) { this.dateTime = dateTime; }  
 public String getImagePath() { return imagePath; }  
 public void setImagePath(String imagePath) { this.imagePath = imagePath; }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Note note = (Note) o;  
 return title.equals(note.title) &&  
 description.equals(note.description) &&  
 priority == note.priority &&  
 dateTime.equals(note.dateTime);  
 }  
  
 @Override  
 public int hashCode() {  
 int result = title.hashCode();  
 result = 31 \* result + description.hashCode();  
 result = 31 \* result + priority;  
 result = 31 \* result + dateTime.hashCode();  
 return result;  
 }  
}

ДОДАТОК Г

package com.example.labtask4;  
  
import android.net.Uri;  
import android.util.Log;  
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;  
import java.util.List;  
  
public class NoteAdapter extends RecyclerView.Adapter<NoteAdapter.NoteViewHolder> {  
  
 private List<Note> noteList;  
 private OnNoteListener onNoteListener;  
  
 public NoteAdapter(List<Note> noteList, OnNoteListener onNoteListener) {  
 this.noteList = noteList;  
 this.onNoteListener = onNoteListener;  
 }  
  
 @NonNull  
 @Override  
 public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*item\_note*, parent, false);  
 return new NoteViewHolder(view, onNoteListener);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) {  
 Note note = noteList.get(position);  
  
 holder.noteTitle.setText(note.getTitle());  
 holder.noteDateTime.setText(note.getDateTime());  
 holder.noteDescription.setText(note.getDescription());  
  
 if (note.getImagePath() != null && !note.getImagePath().isEmpty()) {  
 try {  
 holder.noteIcon.setImageURI(Uri.*parse*(note.getImagePath()));  
 } catch (Exception e) {  
 holder.noteIcon.setImageResource(android.R.drawable.*ic\_menu\_gallery*);  
 }  
 } else {  
 holder.noteIcon.setImageResource(android.R.drawable.*ic\_menu\_gallery*);  
 }  
  
 switch (note.getPriority()) {  
 case 1:  
 holder.notePriority.setImageResource(android.R.drawable.*star\_off*);  
 break;  
 case 2:  
 holder.notePriority.setImageResource(android.R.drawable.*star\_on*);  
 break;  
 case 3:  
 holder.notePriority.setImageResource(android.R.drawable.*btn\_star\_big\_on*);  
 break;  
 default:  
 holder.notePriority.setImageResource(android.R.drawable.*ic\_menu\_help*);  
 break;  
 }  
 }  
  
 @Override  
 public int getItemCount() {  
 return noteList.size();  
 }  
  
 public class NoteViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 TextView noteTitle, noteDateTime, noteDescription;  
 ImageView noteIcon, notePriority;  
 OnNoteListener onNoteListener;  
  
 public NoteViewHolder(@NonNull View itemView, OnNoteListener onNoteListener) {  
 super(itemView);  
 noteTitle = itemView.findViewById(R.id.*note\_title*);  
 noteDateTime = itemView.findViewById(R.id.*note\_date\_time*);  
 noteDescription = itemView.findViewById(R.id.*note\_description*);  
 noteIcon = itemView.findViewById(R.id.*note\_icon*);  
 notePriority = itemView.findViewById(R.id.*note\_priority*);  
 this.onNoteListener = onNoteListener;  
  
 itemView.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 if (onNoteListener != null) {  
 int position = getAdapterPosition();  
 if (position != RecyclerView.*NO\_POSITION*) {  
 onNoteListener.onNoteClick(position);  
 }  
 }  
 }  
 }  
  
 public interface OnNoteListener {  
 void onNoteClick(int position);  
 }  
  
 public void updateNotes(List<Note> newNotes) {  
 this.noteList = newNotes;  
 notifyDataSetChanged();  
 }  
}