UNIVERSITI MALAYSIA TERENGGANU

CSM3123 NATIVE PROGRAMMING K1

LAB TEST

PREPARED FOR:

DR . RABIEI MAMAT

PREPARED BY:

HUSNA ZAHIRA BINTI RUZELI

(S67554)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)
WITH HONOURS

SEMESTER I 2024/2025

Code

MainActivity.kt

```
import android.content.pm.PackageManager
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import android.os.Environment
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import com.example.recyclerviewsqlitedemo.R
import java.io.File
import java.io.FileWriter
import java.io.IOException

class MainActivity : AppCompatActivity() {

    private lateinit var databaseHelper: DatabaseHelper
    private lateinit var titleEditText: EditText
    private lateinit var contentEditText: EditText
    private lateinit var resultTextView: TextView
    private lateinit var searchEditText: EditText
    private lateinit var addButton: Button
    private lateinit var updateButton: Button
    private lateinit var deleteButton: Button
    private lateinit var viewButton: Button
    private lateinit var viewSortedButton: Button
    private lateinit var exportButton: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


databaseHelper = DatabaseHelper(this)
titleEditText = findViewById(R.id.et_title)
contentEditText = findViewById(R.id.et_content)
resultTextView = findViewById(R.id.tv_result)
```

```
    searchEditText = findViewById(R.id.et_search)
    addButton = findViewById(R.id.btn_add)
    updateButton = findViewById(R.id.btn_update)
    deleteButton = findViewById(R.id.btn_delete)
    viewButton = findViewById(R.id.btn_view)
    viewSortedButton = findViewById(R.id.btn_view_sorted)
    exportButton = findViewById(R.id.btn_export)

    addButton.setOnClickListener {
        addNote()
    }
    updateButton.setOnClickListener {
        updateNote()
    }
    deleteButton.setOnClickListener {
        deleteNote()
    }
    viewButton.setOnClickListener {
        viewNotes()
    }
    viewSortedButton.setOnClickListener {
        viewSortedNotes()
    }
    exportButton.setOnClickListener {
        checkPermissionsAndExportNotes()
    }

    searchEditText.addTextChangedListener(object : TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int)
{}
        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
            filterNotes(s.toString())
        }
        override fun afterTextChanged(s: Editable?) {}
    })
}


private fun addNote() {
    val title = titleEditText.text.toString()
    val content = contentEditText.text.toString()
    if (validateTitle(title)) {
        val success = databaseHelper.addNote(title, content)
        Toast.makeText(this, if (success) "Note added successfully!" else "Failed to add
note", Toast.LENGTH_SHORT).show()
    }
}
```

```
private fun updateNote() {
    val title = titleEditText.text.toString()
    val content = contentEditText.text.toString()
    if (validateTitle(title)) {
        val success = databaseHelper.updateNote(title, content)
        Toast.makeText(this, if (success) "Note updated successfully!" else "Failed to update
note", Toast.LENGTH_SHORT).show()
    }
}

private fun deleteNote() {
    val title = titleEditText.text.toString()
    if (validateTitle(title)) {
        val success = databaseHelper.deleteNote(title)
        Toast.makeText(this, if (success) "Note deleted successfully!" else "Failed to delete
note", Toast.LENGTH_SHORT).show()
    }
}

private fun viewNotes() {
    val notes = databaseHelper.getAllNotes()
    resultTextView.text = if (notes.isNotEmpty()) notes.joinToString("\n") else "No notes
found"
}

private fun viewSortedNotes() {
    val notes = databaseHelper.getAllNotes(sorted = true)
    resultTextView.text = if (notes.isNotEmpty()) notes.joinToString("\n") else "No notes
found"
}

private fun validateTitle(title: String): Boolean {
    return if (title.isEmpty()) {
        Toast.makeText(this, "Title cannot be empty", Toast.LENGTH_SHORT).show()
        false
    } else {
        true
    }
}
```

```
    private fun checkPermissionsAndExportNotes() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE), 1)
        } else {
            exportNotesToFile()
        }
    }

    private fun exportNotesToFile() {
        val notes = databaseHelper.getAllNotes()
        if (notes.isEmpty()) {
            Toast.makeText(this, "No notes to export", Toast.LENGTH_SHORT).show()
            return
        }

        val fileName = "notes.txt"
        val file = File(getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS),
fileName)

        try {
            val fileWriter = FileWriter(file)
            notes.forEach {
                fileWriter.write("$it\n")
            }
            fileWriter.close()
            Toast.makeText(this, "Notes exported to ${file.absolutePath}",
Toast.LENGTH_SHORT).show()
        } catch (e: IOException) {
            Toast.makeText(this, "Failed to export notes", Toast.LENGTH_SHORT).show()
            e.printStackTrace()
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out
String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == 1) {
            if ((grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)) {
                exportNotesToFile()
```

```
            } else {
                Toast.makeText(this, "Permission denied to write to external storage",
Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun filterNotes(query: String) {
        val allNotes = databaseHelper.getAllNotes()
        val filteredNotes = allNotes.filter { it.contains(query, ignoreCase = true) }
        resultTextView.text = if (filteredNotes.isNotEmpty())
filteredNotes.joinToString("\n") else "No notes found"
    }
}
```

DatabaseHelper.kt

```
package com.example.sqlitedemo

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "NoteDatabase"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NOTES = "Notes"
        private const val COLUMN_ID = "id"
        private const val COLUMN_TITLE = "title"
        private const val COLUMN_CONTENT = "content"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = "CREATE TABLE $TABLE_NOTES ($COLUMN_ID INTEGER
PRIMARY KEY AUTOINCREMENT, $COLUMN_TITLE TEXT,
$COLUMN_CONTENT TEXT)"
        db?.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NOTES")
        onCreate(db)
    }


fun addNote(title: String, content: String): Boolean {
    val db = writableDatabase
    val contentValues = ContentValues().apply {
        put(COLUMN_TITLE, title)
        put(COLUMN_CONTENT, content)
    }
    val result = db.insert(TABLE_NOTES, null, contentValues)
    db.close()
    return result != -1L
}
```

```
    fun updateNote(title: String, content: String): Boolean {
        val db = writableDatabase
        val contentValues = ContentValues().apply {
            put(COLUMN_CONTENT, content)
        }
        val result = db.update(TABLE_NOTES, contentValues, "$COLUMN_TITLE=?",
arrayOf(title))
        db.close()
        return result > 0
    }

    fun deleteNote(title: String): Boolean {
        val db = writableDatabase
        val result = db.delete(TABLE_NOTES, "$COLUMN_TITLE=?", arrayOf(title))
        db.close()
        return result > 0
    }

    fun getAllNotes(sorted: Boolean = false): List<String> {
        val noteList = ArrayList<String>()
        val db = readableDatabase
        val sortOrder = if (sorted) "$COLUMN_TITLE ASC" else null
        val cursor = db.query(TABLE_NOTES, null, null, null, null, null, sortOrder)
        if (cursor.moveToFirst()) {
            do {
                val title =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_TITLE))
                val content =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_CONTENT))
                noteList.add("Title: $title, Content: $content")
            } while (cursor.moveToNext())
        }
        cursor.close()
        db.close()
        return noteList
    }
}
```

Note.kt

```
package com.example.recyclerviewsqlitedemo
data class Note (
    val id: Int,
    var title: String,
    var content: String
)
```

NoteAdapter.kt

```
package com.example.recyclerviewsqlitedemo
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class NoteAdapter(
    private val noteList: List<Note>,
    private val onNoteLongClicked: (Note) -> Unit
) : RecyclerView.Adapter<NoteAdapter.NoteViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
NoteViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.note_item, parent,
false)
        return NoteViewHolder(view)
    }

    override fun onBindViewHolder(holder: NoteViewHolder, position: Int) {
        val note = noteList[position]
        holder.titleTextView.text = note.title
        holder.contentTextView.text = note.content
        holder.itemView.setOnLongClickListener {
            onNoteLongClicked(note)
            true
        }
    }
```

```
    override fun getItemCount(): Int {
        return noteList.size
    }

    class NoteViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val titleTextView: TextView = itemView.findViewById(R.id.tv_title)
        val contentTextView: TextView = itemView.findViewById(R.id.tv_content)
    }
}
```

Activity_main.xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.sqlitedemo.MainActivity"
    android:padding="16dp">

    <EditText
        android:id="@+id/et_search"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Search Notes"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginBottom="16dp"/>


<EditText
    android:id="@+id/et_title"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Title"
    android:maxLength="50"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/et_search"
```

```
    app:layout_constraintBottom_toTopOf="@id/et_content"
    app:layout_constraintVertical_chainStyle="packed"
    android:layout_marginBottom="8dp"/>

<EditText
    android:id="@+id/et_content"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/et_title"
    app:layout_constraintBottom_toTopOf="@id/btn_add"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>

<Button
    android:id="@+id/btn_add"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Add Note"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/et_content"
    app:layout_constraintBottom_toTopOf="@id/btn_update"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>


<Button
    android:id="@+id/btn_update"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Update Note"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_add"
    app:layout_constraintBottom_toTopOf="@id/btn_delete"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>

<Button
    android:id="@+id/btn_delete"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Delete Note"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_update"
    app:layout_constraintBottom_toTopOf="@id/btn_view"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>

<Button
    android:id="@+id/btn_view"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="View All"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_delete"
    app:layout_constraintBottom_toTopOf="@id/btn_view_sorted"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>

<Button
    android:id="@+id/btn_view_sorted"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="View Sorted"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_view"
    app:layout_constraintBottom_toTopOf="@id/btn_export"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>


<Button
    android:id="@+id/btn_export"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Export Notes"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_view_sorted"
    app:layout_constraintBottom_toTopOf="@id/tv_result"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"/>
```

```
    <TextView
        android:id="@+id/tv_result"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text=""
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btn_export"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginTop="8dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

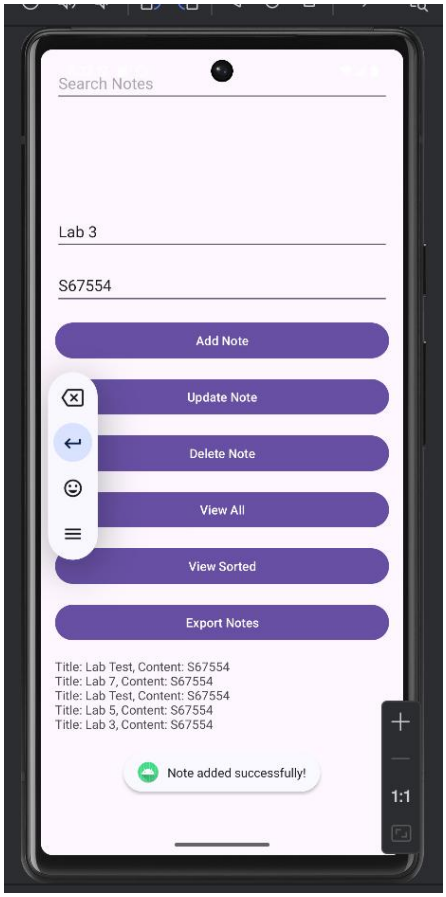
Note_item.xml

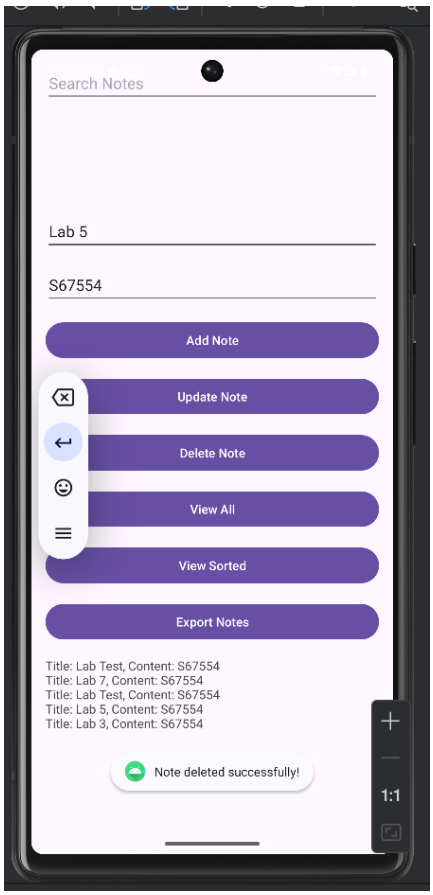
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="8dp">
    <TextView
        android:id="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="Title"/>
    <TextView
        android:id="@+id/tv_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:text="Content"/>
</LinearLayout>
```

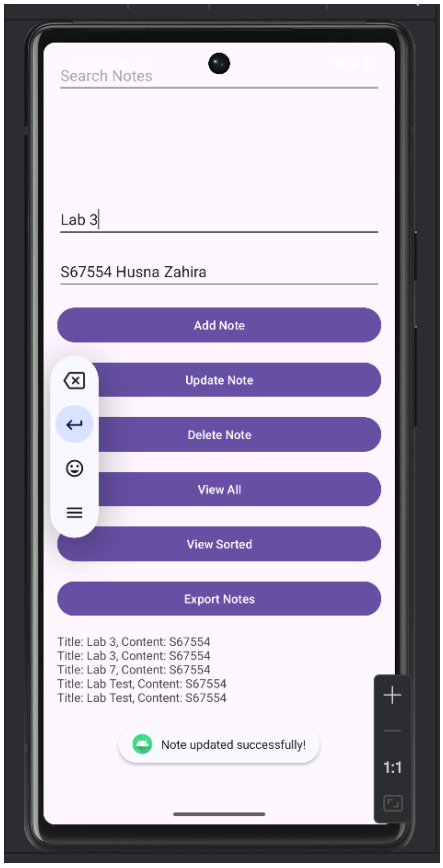
Dialog_add_note.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <EditText
        android:id="@+id/et_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Title"/>
    <EditText
        android:id="@+id/et_content"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Content"/>
</LinearLayout>
```

Output






Search Notes
Lab 3
S67554 Husna Zahira
Add Note
Update Note
Delete Note
View All
View Sorted
Export Notes
Title: Lab 3, Content: S67554
Title: Lab 3, Content: S67554
Title: Lab 7, Content: S67554
Title: Lab Test, Content: S67554
Title: Lab Test, Content: S67554
Note updated successfully!
1:1