Deakin University

Mobile Application Development

OnTrack Submission

Pass Task 4.1

Submitted By:
Akashdeep Akashdeep
s223040483
2024/04/19 22:12

Tutor:
Shiva Pokhrel

Outcome	**Weight**
Unit Learning Outcome 1	♦♦♦♦◊
Unit Learning Outcome 2	♦♦♦♦◊
Unit Learning Outcome 3	♦♦♦♦◊

Learned about using SQLite database with android application

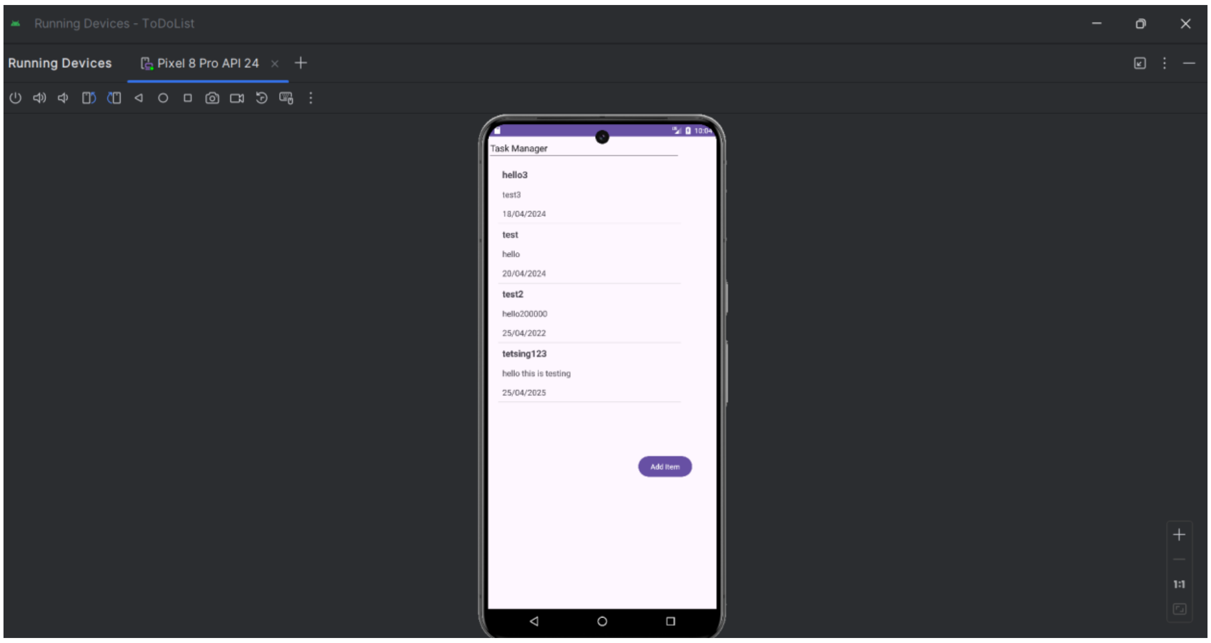
April 19, 2024

SIT708 TASK4.1P

YOUTUBE LINK : https://www.youtube.com/watch?v=5gQdNXqVsCw

Github link https://github.com/akash271291/SIT708TASK4.1P

Screenshot



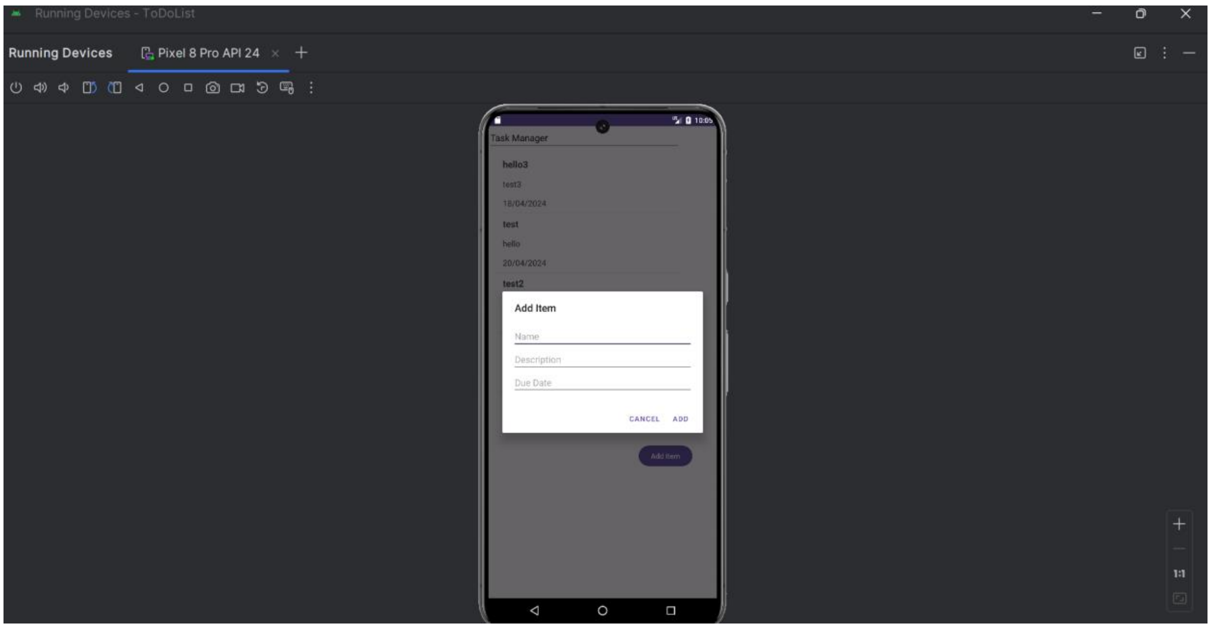
Task Manager

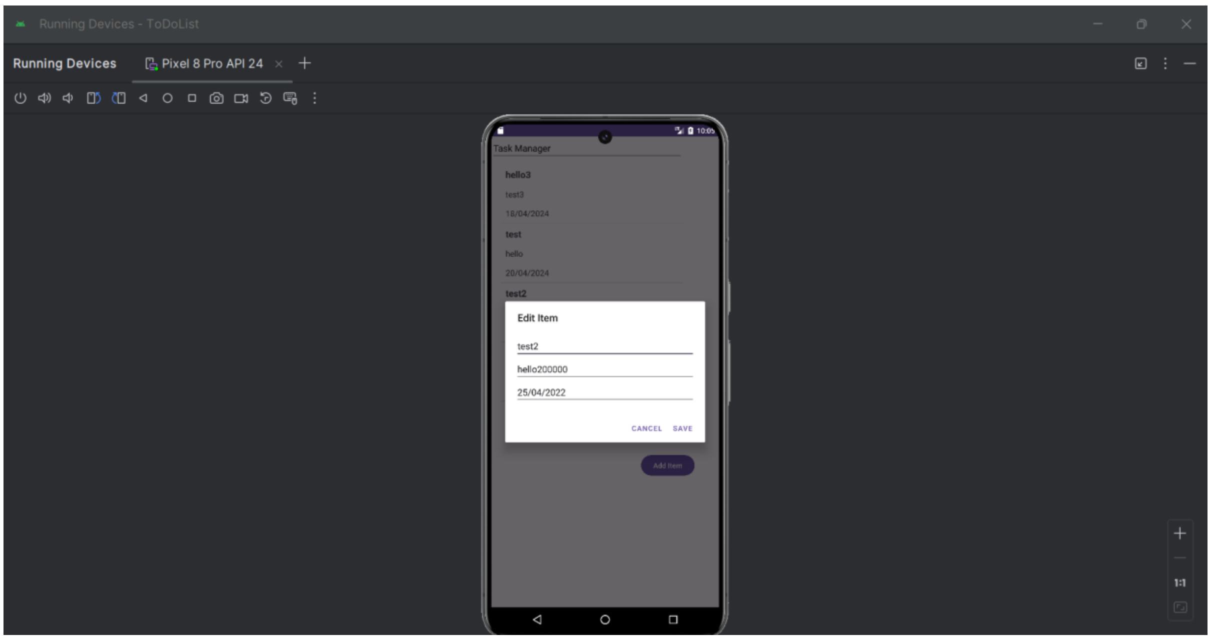
hello3
test3
18/04/2024

test
hello
20/04/2024

test2
hello200000
25/04/2022

tetsing123
hello this is testing
25/04/2025





Task Manager

hello3
test3
18/04/2024

test
hello
20/04/2024

test2

Edit Item

test2
hello200000
25/04/2022

```java
package com.example.todolist;
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 java.util.ArrayList;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "todolist.db";
    private static final String TABLE_NAME = "items";
    private static final String COL_ID = "id";
    private static final String COL_NAME = "name";
    private static final String COL_DESCRIPTION = "description";
    private static final String COL_DUE_DATE = "due_date";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableQuery = "CREATE TABLE " + TABLE_NAME + " (" +
                COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL_NAME + " TEXT, " +
                COL_DESCRIPTION + " TEXT, " +
                COL_DUE_DATE + " TEXT)";
        db.execSQL(createTableQuery);
    }

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

    public boolean insertData(String name, String description, String dueDate) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, name);
        contentValues.put(COL_DESCRIPTION, description);
        contentValues.put(COL_DUE_DATE, dueDate);
        long result = db.insert(TABLE_NAME, null, contentValues);
        return result != -1;
    }

    public ArrayList<Item> getData() {
        ArrayList<Item> itemsList = new ArrayList<>();
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " ORDER BY " +
        ↪   COL_DUE_DATE, null);
        if (cursor.moveToFirst()) {
            do {
                @SuppressLint("Range") int id =
                ↪  cursor.getInt(cursor.getColumnIndex(COL_ID)); // Retrieve ID
                @SuppressLint("Range") String name =
                ↪  cursor.getString(cursor.getColumnIndex(COL_NAME));
                @SuppressLint("Range") String description =
                ↪  cursor.getString(cursor.getColumnIndex(COL_DESCRIPTION));
                @SuppressLint("Range") String dueDate =
                ↪  cursor.getString(cursor.getColumnIndex(COL_DUE_DATE));
                Item item = new Item(id, name, description, dueDate); // Pass ID to
                ↪  the constructor
                itemsList.add(item);
            } while (cursor.moveToNext());
        }
        cursor.close();
        return itemsList;
    }





    public boolean deleteData(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, COL_ID + "=?", new
        ↪  String[]{String.valueOf(id)}) > 0;
    }


    public boolean updateItem(Item item) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_NAME, item.getName());
        contentValues.put(COL_DESCRIPTION, item.getDescription());
        contentValues.put(COL_DUE_DATE, item.getDueDate());
        int result = db.update(TABLE_NAME, contentValues, COL_ID + "=?", new
        ↪  String[]{String.valueOf(item.getId())});
        return result != -1;
    }

}


-------------------------------------------------

package com.example.todolist;

public class Item {
    private int id; // Added ID field
    private String name;
    private String description;
    private String dueDate;

    public Item(int id, String name, String description, String dueDate) {
        this.id = id;
        this.name = name;
        this.description = description;
        this.dueDate = dueDate;
    }

    // Getter and setter methods for ID
    public int getId() {
        return id;
    }

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

    // Getter and setter methods for other fields
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getDueDate() {
        return dueDate;
    }

    public void setDueDate(String dueDate) {
        this.dueDate = dueDate;
    }
}


-------------------------

package com.example.todolist;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import java.util.ArrayList;

public class ItemAdapter extends ArrayAdapter<Item> {

    public ItemAdapter(Context context, ArrayList<Item> items) {
        super(context, 0, items);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // Get the data item for this position
        Item item = getItem(position);

        // Check if an existing view is being reused, otherwise inflate the view
        if (convertView == null) {
            convertView =
             ↪ LayoutInflater.from(getContext()).inflate(R.layout.item_layout,
             ↪ parent, false);
        }

        // Lookup view for data population
        TextView tvName = convertView.findViewById(R.id.tvName);
        TextView tvDescription = convertView.findViewById(R.id.tvDescription);
        TextView tvDueDate = convertView.findViewById(R.id.tvDueDate);

        // Populate the data into the template view using the data object
        tvName.setText(item.getName());
        tvDescription.setText(item.getDescription());
        tvDueDate.setText(item.getDueDate());

        // Return the completed view to render on screen
        return convertView;
    }
}


---------------------
package com.example.todolist;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;

import androidx.appcompat.app.AppCompatActivity;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
```

```java
import java.io.IOException;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.io.BufferedReader;
import java.util.Collections;
import java.util.Comparator;
import java.util.Date;

public class MainActivity extends AppCompatActivity {

    private ArrayList<Item> items; // Modified to hold Item objects
    private ItemAdapter itemsAdapter; // Custom adapter for Item objects
    private ListView lvItems;
    private DatabaseHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvItems = findViewById(R.id.lvItems);
        dbHelper = new DatabaseHelper(this); // Initialize DatabaseHelper
        items = new ArrayList<>();
        itemsAdapter = new ItemAdapter(this, items); // Use custom adapter
        lvItems.setAdapter(itemsAdapter);
        readItems();
        setupListViewListener();
        setupItemClickListener();

        // Sort tasks by due date
        Collections.sort(items, new Comparator<Item>() {
            DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

            @Override
            public int compare(Item item1, Item item2) {
                try {
                    Date dueDate1 = dateFormat.parse(item1.getDueDate());
                    Date dueDate2 = dateFormat.parse(item2.getDueDate());
                    return dueDate1.compareTo(dueDate2);
                } catch (ParseException e) {
                    e.printStackTrace();
                    return 0;
                }
            }
        });

        itemsAdapter.notifyDataSetChanged(); // Notify adapter of data change
    }


    private void setupListViewListener() {
        lvItems.setOnItemLongClickListener(
                new AdapterView.OnItemLongClickListener() {
                    @Override
                    public boolean onItemLongClick(AdapterView<?> adapter,
                                                   View item, int pos, long id) {
                        removeItem(pos);
                        return true;
                    }
                });
    }

    private void setupItemClickListener() {
        lvItems.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, final int
            ↪  position, long id) {
                // Inflate the layout for the dialog
                LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
                View dialogView = inflater.inflate(R.layout.dialog_edit_item, null);

                final EditText etName = dialogView.findViewById(R.id.etName);
                final EditText etDescription =
                ↪  dialogView.findViewById(R.id.etDescription);
                final EditText etDueDate = dialogView.findViewById(R.id.etDueDate);

                // Set current values
                etName.setText(items.get(position).getName());
                etDescription.setText(items.get(position).getDescription());
                etDueDate.setText(items.get(position).getDueDate());

                // Build the dialog
                AlertDialog.Builder builder = new
                ↪  AlertDialog.Builder(MainActivity.this);
                builder.setView(dialogView);
                builder.setTitle("Edit Item");
                builder.setPositiveButton("Save", new
                ↪  DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {
                        // Get the updated values
                        String name = etName.getText().toString();
                        String description = etDescription.getText().toString();
                        String dueDate = etDueDate.getText().toString();

                        // Update the item in the list
                        Item updatedItem = items.get(position);
                        updatedItem.setName(name);
                        updatedItem.setDescription(description);
                        updatedItem.setDueDate(dueDate);

                        // Update the item in the database
                        dbHelper.updateItem(updatedItem);

                        // Refresh the list view
                        items.clear();
                        items.addAll(dbHelper.getData()); // Now getData() returns
                        ↪  ArrayList<Item>
                        itemsAdapter.notifyDataSetChanged();
                    }
                });
                builder.setNegativeButton("Cancel", null);
                builder.show();
            }
        });
    }

    public void onAddItem(View v) {
        // Inflate the layout for the dialog
        LayoutInflater inflater = LayoutInflater.from(this);
        View dialogView = inflater.inflate(R.layout.dialog_add_item, null);

        final EditText etName = dialogView.findViewById(R.id.etName);
        final EditText etDescription = dialogView.findViewById(R.id.etDescription);
        final EditText etDueDate = dialogView.findViewById(R.id.etDueDate);

        // Build the dialog
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setView(dialogView);
        builder.setTitle("Add Item");
        builder.setPositiveButton("Add", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                // Get the entered values
                String name = etName.getText().toString();
                String description = etDescription.getText().toString();
                String dueDate = etDueDate.getText().toString();

                // Add the item to the database
                dbHelper.insertData(name, description, dueDate);

                // Refresh the list view
                items.clear();
                items.addAll(dbHelper.getData()); // Now getData() returns
                ↪  ArrayList<Item>
                itemsAdapter.notifyDataSetChanged();
            }
        });
        builder.setNegativeButton("Cancel", null);
        builder.show();
    }

    private void removeItem(int position) {
        Item itemToRemove = items.get(position);
        dbHelper.deleteData(itemToRemove.getId());
        items.remove(position);
        itemsAdapter.notifyDataSetChanged();
    }
```

```java


    private void readItems() {
        items.clear(); // Clear the existing items list
        items.addAll(dbHelper.getData()); // Retrieve items from the database
    }


    private void writeItems() {
        File filesDir = getFilesDir();
        File todoFile = new File(filesDir, "todo.txt");
        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(todoFile));
            for (Item item : items) {
                String itemString = item.getName() + " - " + item.getDescription() +
                ↪  " - " + item.getDueDate();
                writer.write(itemString);
                writer.newLine();
            }
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```