**Option #2: Contact Book**

Geoffery Tyler Van Antwerp

Colorado State University Global

Mr. Farhad Bari

CSC475

9/29/2024

**Analysis**

The purpose of this project was to create a simple contact book application that allows users to add, edit, and delete contacts on an Android device. For this, I used SharedPreferences as the storage method, which enabled saving contacts as key-value pairs (with the contact’s name as the key and the phone number as the value). The goal was to keep things simple while ensuring the application provided the basic functionality of managing contacts.

During development, the biggest challenge I encountered was making the interaction with the contact list smoother. Initially, users could only manage contacts by manually entering their name and phone number into text fields. This wasn’t very user-friendly, especially when there were multiple contacts. To improve the experience, I added functionality that allows users to select a contact by tapping on it from the list. Once a contact is selected, its details are automatically filled in the input fields, making it much easier to edit or delete without re-entering information. Implementing this required working with the ListView and handling user selections via an OnItemClickListener. Another small challenge was ensuring that the list of contacts updated instantly whenever a contact was added, edited, or deleted. This was resolved by refreshing the contact list after each action.

Through this project, I got hands-on experience with SharedPreferences, which is a lightweight storage solution for small amounts of key-value data in Android. I also improved my skills in designing user interfaces with EditText, Buttons, and ListView, as well as handling user input and making the UI more dynamic with immediate feedback. Learning how to handle Android UI events, especially for user selections in ListView, was particularly valuable. Another obstacle that I am continuing to improve on throughout the weeks is working with the Android Studio. Getting the project formatted correctly has been a challenge but it is getting easier with each passing week. Overall, this project was a great way to practice not only the core concepts of data storage and retrieval but also how to make simple Android applications more user-friendly.

**Screenshot**

A computer screen shot of a program

Description automatically generated

**Source Code**

package com.example.contactbook;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ListView;

import android.widget.ArrayAdapter;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Map;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

EditText editTextName, editTextPhone;

Button buttonAdd, buttonEdit, buttonDelete;

SharedPreferences sharedPreferences;

ListView listViewContacts;

ArrayAdapter<String> adapter;

ArrayList<String> contactsList;

String selectedContactName = null;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

editTextName = findViewById(R.id.editTextName);

editTextPhone = findViewById(R.id.editTextPhone);

buttonAdd = findViewById(R.id.buttonAdd);

buttonEdit = findViewById(R.id.buttonEdit);

buttonDelete = findViewById(R.id.buttonDelete);

listViewContacts = findViewById(R.id.listViewContacts);

sharedPreferences = getSharedPreferences("contacts", MODE\_PRIVATE);

contactsList = new ArrayList<>();

adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, contactsList);

listViewContacts.setAdapter(adapter);

loadContacts();

buttonAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String name = editTextName.getText().toString();

String phone = editTextPhone.getText().toString();

addContact(name, phone);

}

});

buttonEdit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (selectedContactName != null) {

String phone = editTextPhone.getText().toString();

editContact(selectedContactName, phone);

}

}

});

buttonDelete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (selectedContactName != null) {

deleteContact(selectedContactName);

}

}

});

listViewContacts.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

String contact = contactsList.get(position);

String[] contactDetails = contact.split(":");

selectedContactName = contactDetails[0].trim(); // Get the contact name

String phone = contactDetails[1].trim(); // Get the phone number

editTextName.setText(selectedContactName);

editTextPhone.setText(phone);

}

});

}

private void addContact(String name, String phone) {

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString(name, phone);

editor.apply();

loadContacts();

clearFields();

}

private void editContact(String name, String phone) {

if (sharedPreferences.contains(name)) {

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString(name, phone);

editor.apply();

loadContacts();

clearFields();

}

}

private void deleteContact(String name) {

if (sharedPreferences.contains(name)) {

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.remove(name);

editor.apply();

loadContacts();

clearFields();

}

}

private void loadContacts() {

contactsList.clear();

Map<String, ?> allContacts = sharedPreferences.getAll();

for (Map.Entry<String, ?> entry : allContacts.entrySet()) {

contactsList.add(entry.getKey() + ": " + entry.getValue().toString());

}

adapter.notifyDataSetChanged();

}

private void clearFields() {

editTextName.setText("");

editTextPhone.setText("");

selectedContactName = null;

}

}