НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни

“Програмування мобільних систем”

Виконав:

студент групи ІВ-83

ЗК ІО-7303

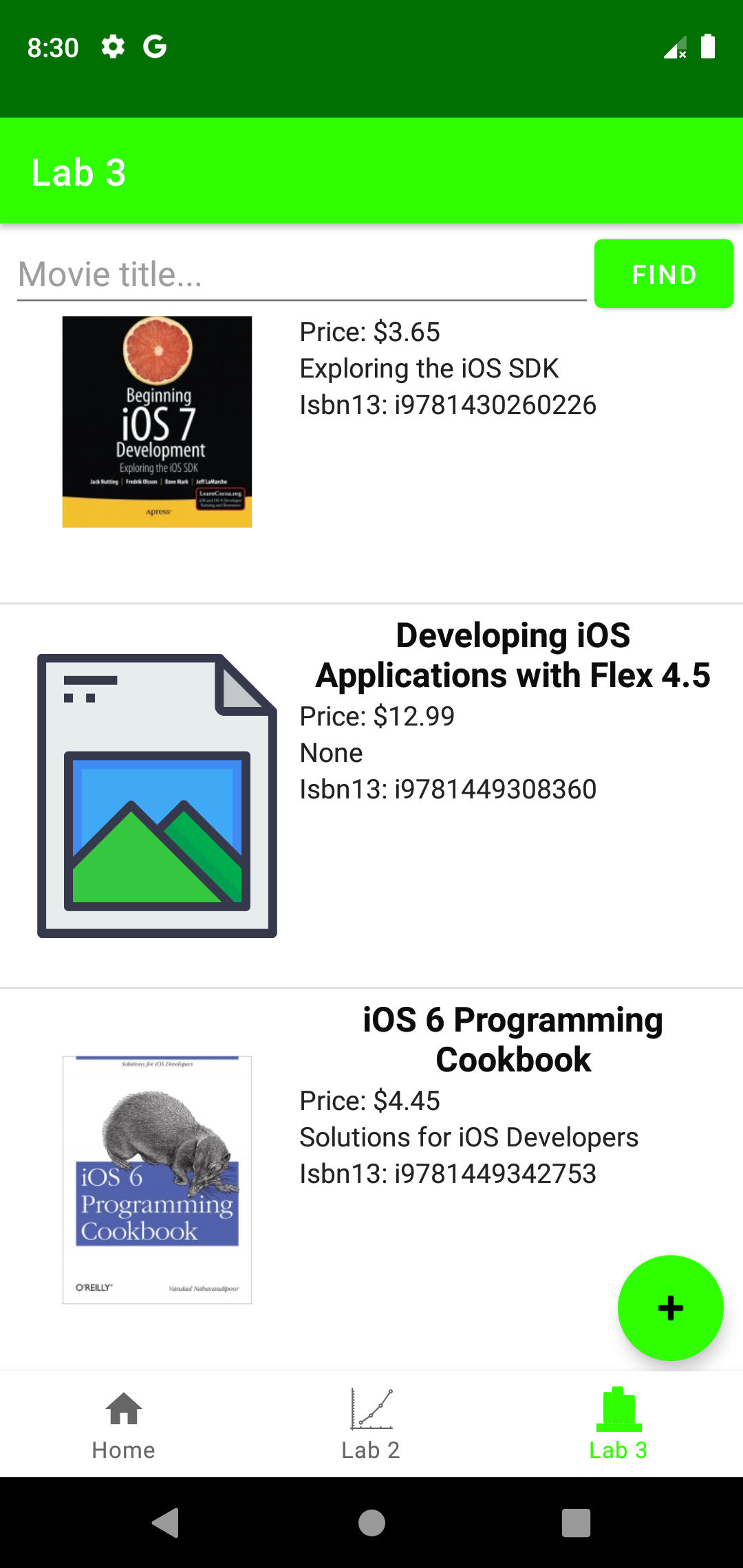
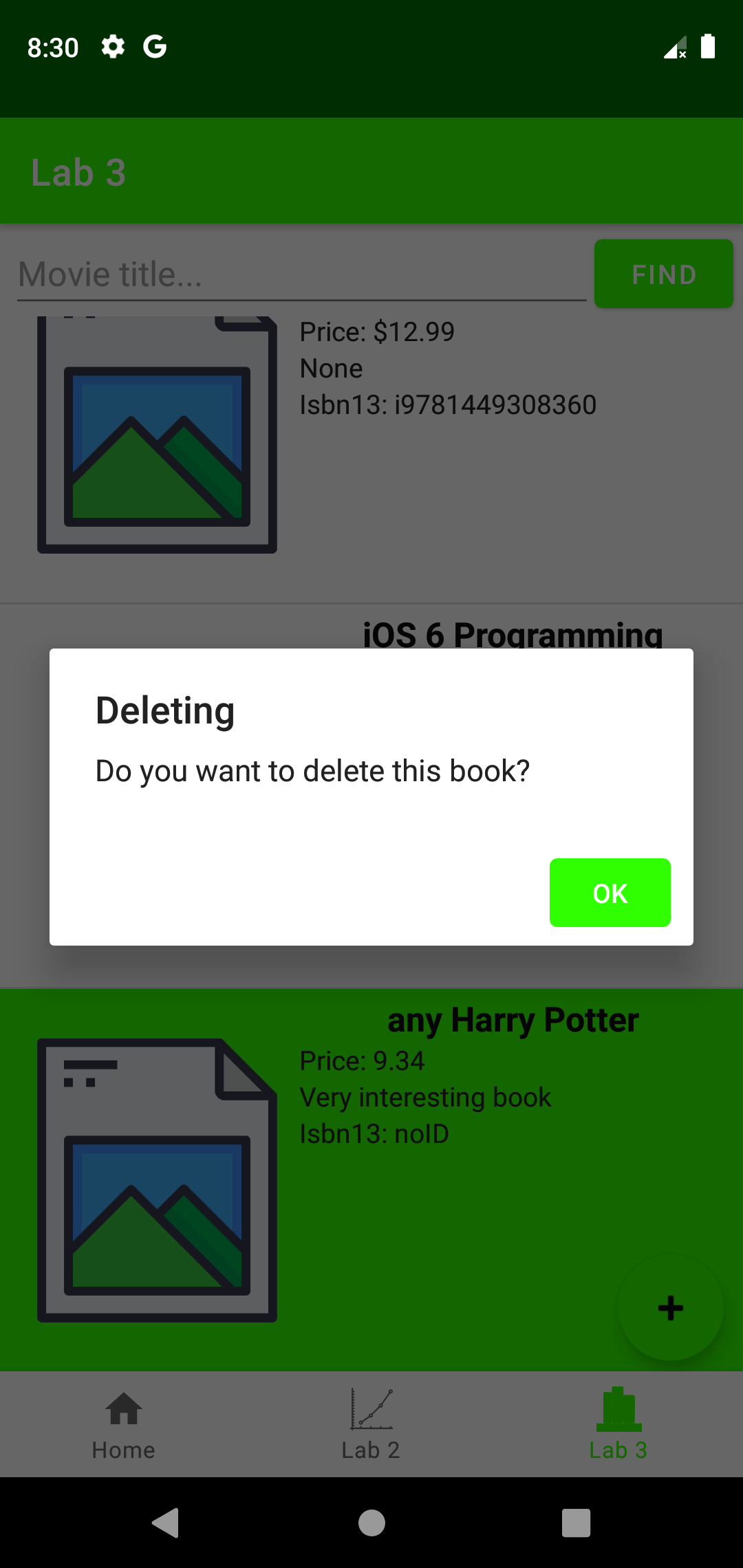
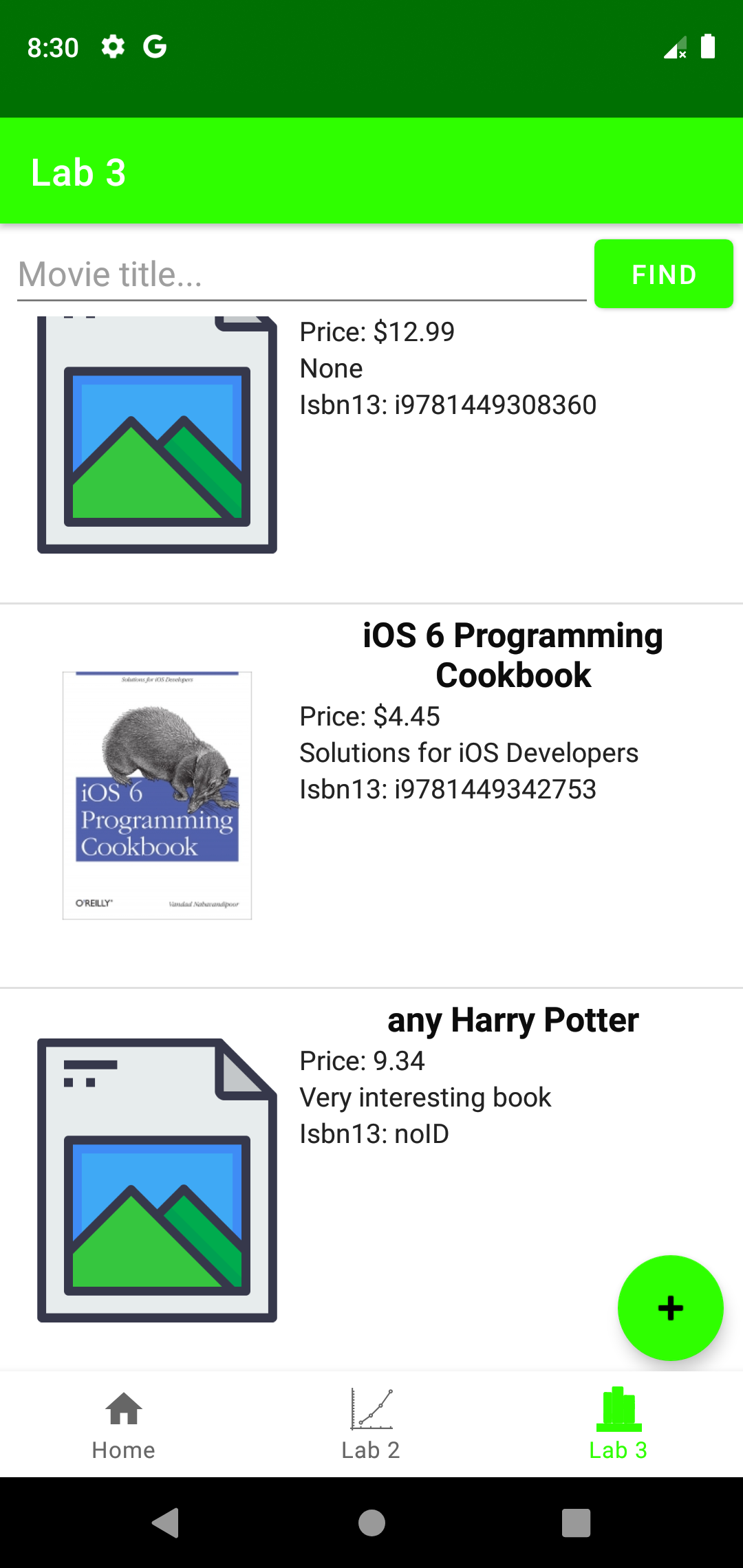
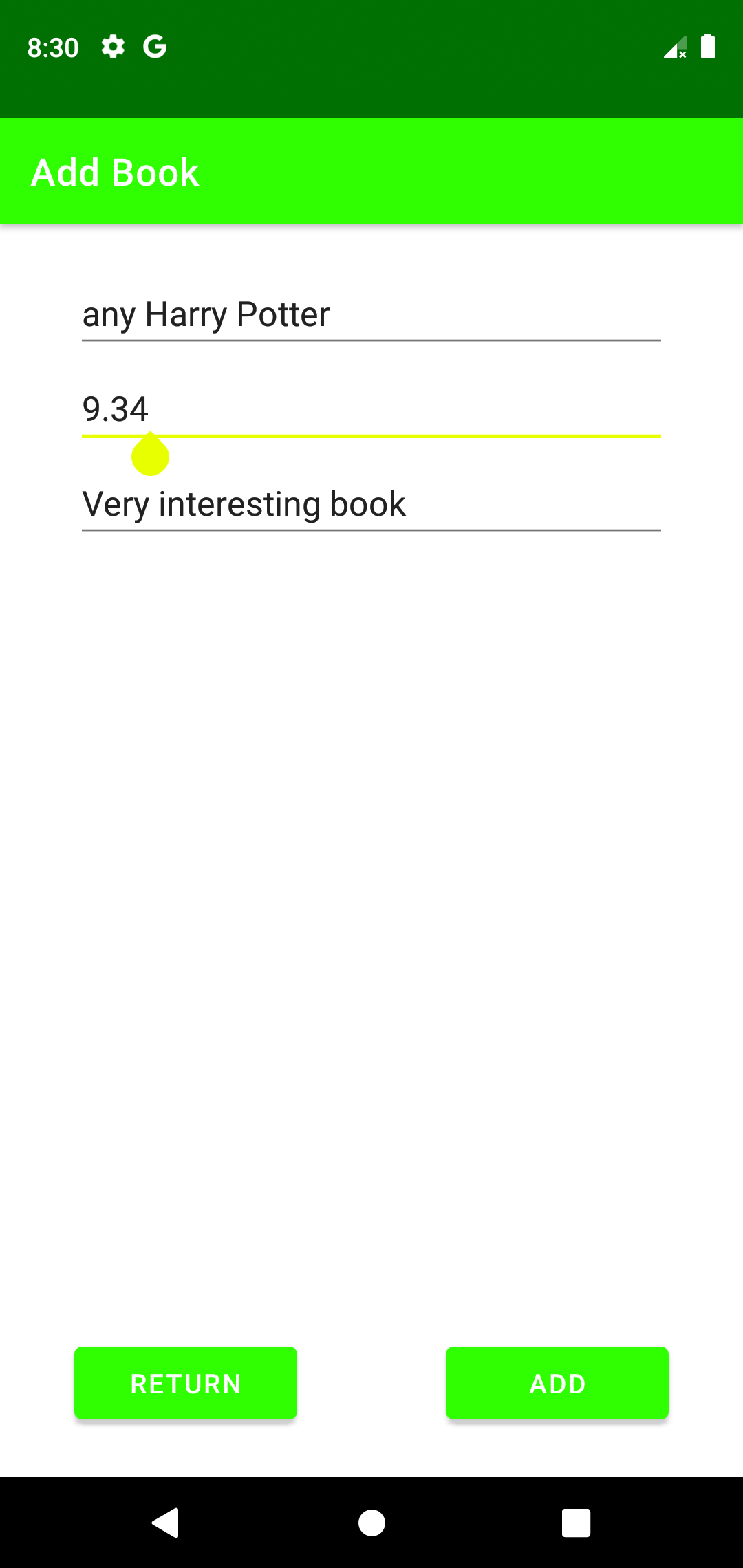
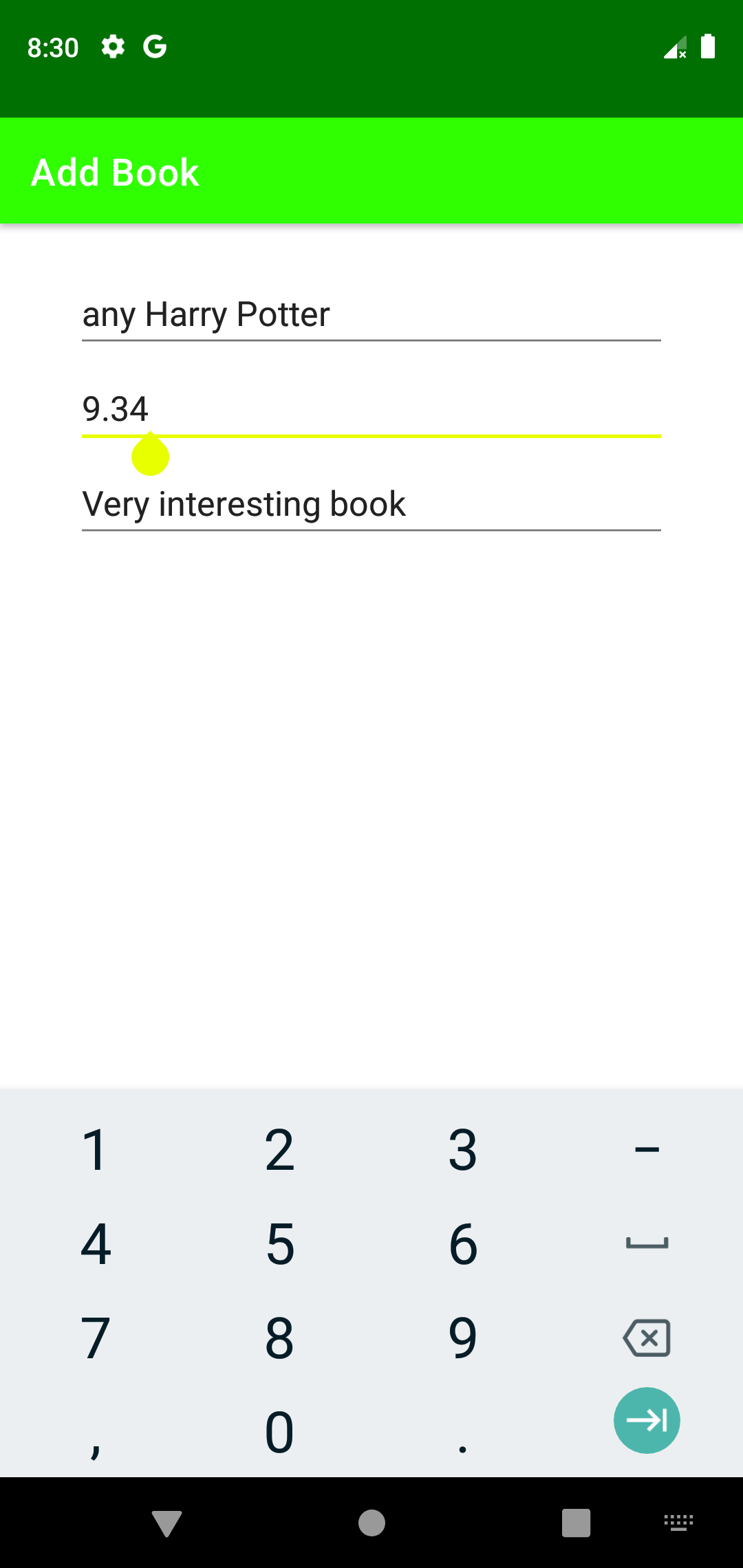
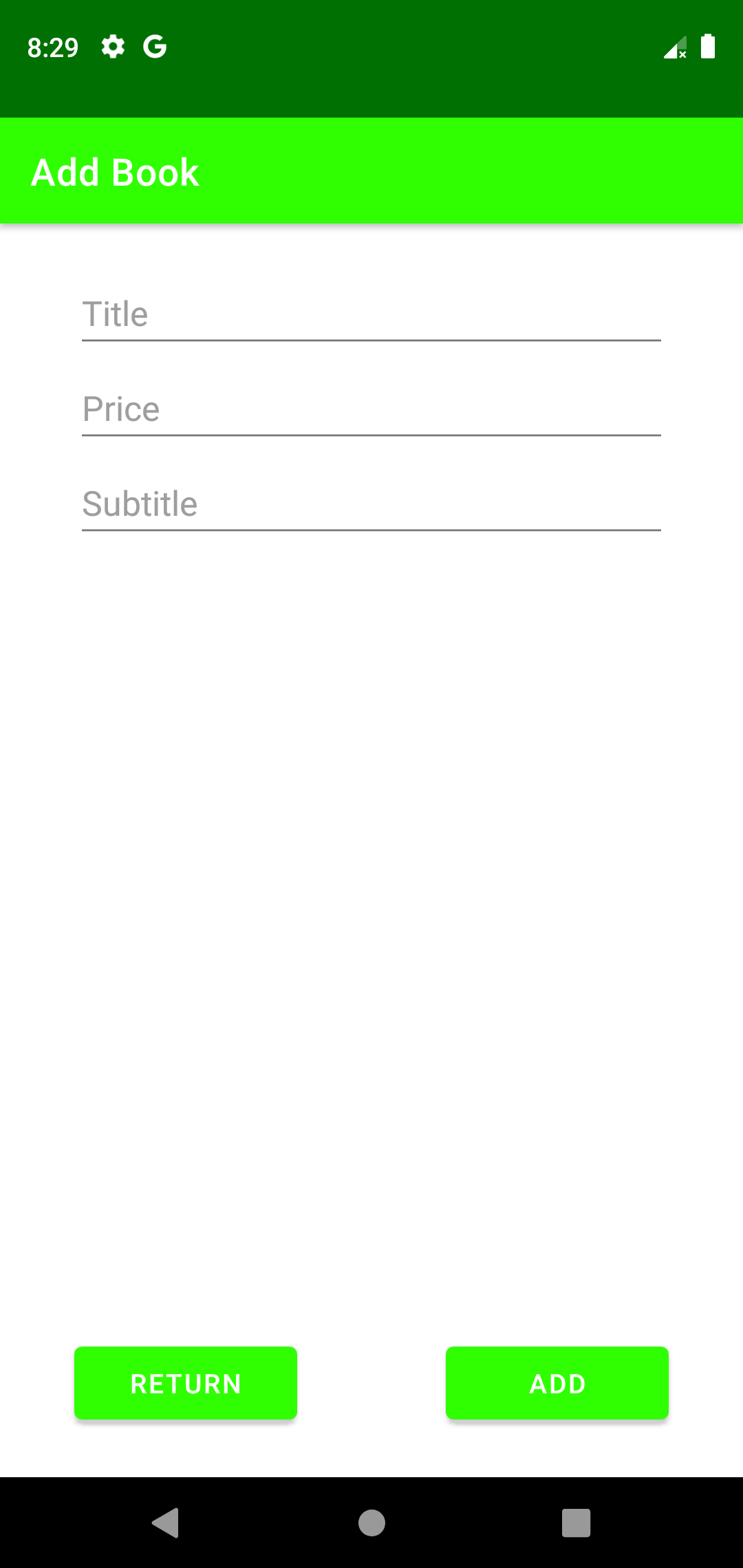
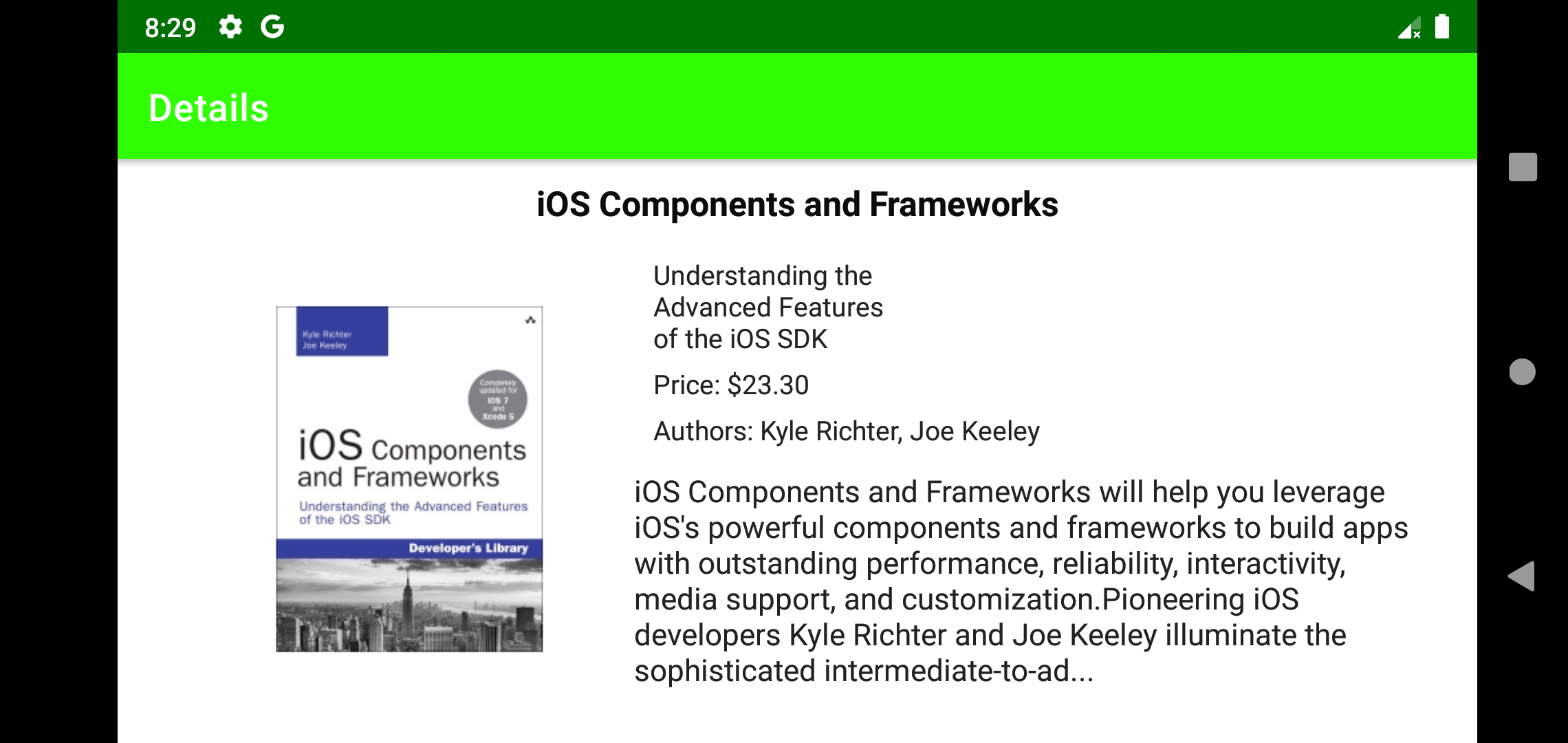
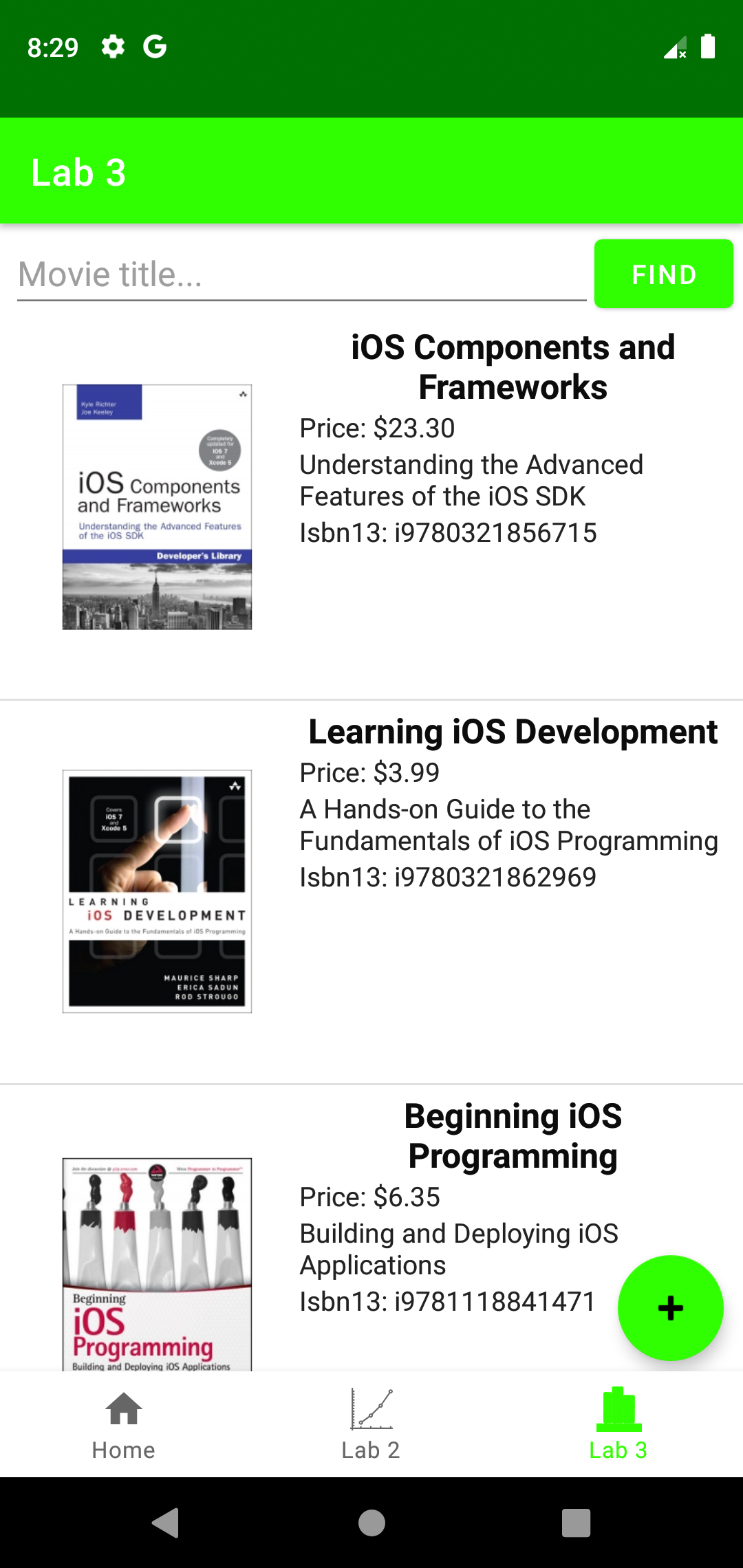
Ву В'єт Тунг

Київ 2021

**Варіант № 7303%2+1=2**

|  |
| --- |
| **Варіант 2** |
| Предметна область – книги  [Архів для завантаження](https://drive.google.com/file/d/1c18F6XvCBc69Lq-GaqUmrvta4N6w2Wt8/view?usp=sharing) |

**Скріншоти роботи додатка**



**Лістинг коду**

**AddBookActivity.java**

package ua.kpi.comsys.IO7303.ui.library;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.io.ByteArrayOutputStream;  
import java.util.List;  
  
import ua.kpi.comsys.IO7303.R;  
  
public class AddBookActivity extends AppCompatActivity {  
 int resBookList;  
 private List<Book> books;  
 ByteArrayOutputStream bos;  
 String booksFile = "books\_list\_custom.txt";  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*add\_book\_layout*);  
 bos = new ByteArrayOutputStream();  
 Bundle arguments = getIntent().getExtras();  
 resBookList = (int)arguments.get("booklistId");  
 }  
  
 public void addBtn(View view) { // ADD  
 String bookTitle = ((EditText)findViewById(R.id.*addTitle*)).getText().toString();  
 String bookSubtitle = ((EditText)findViewById(R.id.*addSubbtitle*)).getText().toString();  
 String bookPrice = ((EditText)findViewById(R.id.*addPrice*)).getText().toString();  
  
 String imageName;  
  
 if(bookTitle.length()<1){  
 Toast.*makeText*(view.getContext(), "Uncorrected title", Toast.*LENGTH\_LONG*).show();  
 }  
 else {  
 JsonHandler jsonHandler = new JsonHandler(resBookList);  
 jsonHandler.*setFileUserName*(booksFile);  
 books = jsonHandler.*importBooksFromJSON*(view.getContext());  
  
 imageName = "";  
  
 books.add(new Book(bookTitle, bookSubtitle, bookPrice, "noID", imageName));  
  
 jsonHandler.*exportToJSON*(view.getContext(), books);  
  
 finish();  
 Toast.*makeText*(view.getContext(), "Added successfully", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 public void returnBtn(View view) { // RETURN  
 finish();  
 }  
}

**Book.java**

package ua.kpi.comsys.IO7303.ui.library;  
  
public class Book {  
 private String title, subtitle, price, isbn13, image, authors, publisher, pages, year, rating, desc;  
  
 public Book(String title, String subtitle, String price, String isbn13, String image) {  
 this.title = title;  
 this.subtitle = subtitle;  
 this.price = price;  
 this.isbn13 = isbn13;  
 this.image = image;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public String getSubtitle() {  
 return subtitle;  
 }  
  
 public void setSubtitle(String subtitle) {  
 this.subtitle = subtitle;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
  
 public String getIsbn13() {  
 return isbn13;  
 }  
  
 public void setIsbn13(String isbn13) {  
 this.isbn13 = isbn13;  
 }  
  
 public String getImage() {  
 return image;  
 }  
  
 public void setImage(String image) {  
 this.image = image;  
 }  
  
 public String getAuthors() {  
 return authors;  
 }  
  
 public void setAuthors(String authors) {  
 this.authors = authors;  
 }  
  
 public String getPublisher() {  
 return publisher;  
 }  
  
 public void setPublisher(String publisher) {  
 this.publisher = publisher;  
 }  
  
 public String getPages() {  
 return pages;  
 }  
  
 public void setPages(String pages) {  
 this.pages = pages;  
 }  
  
 public String getYear() {  
 return year;  
 }  
  
 public void setYear(String year) {  
 this.year = year;  
 }  
  
 public String getRating() {  
 return rating;  
 }  
  
 public void setRating(String rating) {  
 this.rating = rating;  
 }  
  
 public String getDesc() {  
 return desc;  
 }  
  
 public void setDesc(String desc) {  
 this.desc = desc;  
 }  
  
 @Override  
 public String toString() {  
 return title + '\n' +  
 subtitle + '\n' +  
 "Price: " + price;  
 }  
}

**BookDetail.java**

package ua.kpi.comsys.IO7303.ui.library;  
  
import android.os.Bundle;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import ua.kpi.comsys.IO7303.R;  
  
public class BookDetail extends AppCompatActivity {  
 Book book;  
 String imbdId;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*book\_info*);  
  
 Bundle arguments = getIntent().getExtras();  
 imbdId = arguments.get("Isbn13").toString();  
  
 int res = this.getResources().getIdentifier(imbdId, "raw", this.getPackageName()); // поиск ИД по имени  
  
 JsonHandler jsonHandler = new JsonHandler(res);  
 jsonHandler.*setUserFileEnable*(false);  
  
  
 if(res!=0) {  
 book = jsonHandler.*importBookFromJSON*(this); // если есть такой ИД  
 TextView title = findViewById(R.id.*bookDetailTitle*);  
 TextView subtitle = findViewById(R.id.*subtitleDetail*);  
 TextView price = findViewById(R.id.*priceDetail*);  
 TextView rating = findViewById(R.id.*ratingDetail*);  
 TextView publisher = findViewById(R.id.*publisherDetail*);  
 TextView authors = findViewById(R.id.*authorsDetail*);  
 TextView year = findViewById(R.id.*yearDetail*);  
 TextView pages = findViewById(R.id.*pagesDetail*);  
 TextView desc = findViewById(R.id.*desc*);  
  
 ImageView poster = (ImageView) findViewById(R.id.*imageDetail*);  
  
 int img;  
 try {  
 String posterName = book.getImage().replaceAll(".png","").toLowerCase();  
 img = getResources().getIdentifier(posterName, "drawable", getPackageName()); // поиск ИД по имени  
 } catch (Exception e){img = 0;};  
  
  
 if(img!=0) poster.setImageResource(img); // если есть такой ИД  
 else poster.setImageResource(R.drawable.*no\_image*); // стандартная картинка  
  
 title.setText(book.getTitle());  
 subtitle.setText(book.getSubtitle());  
 price.setText("Price: "+ book.getPrice());  
 rating.setText("Rating: "+ book.getRating());  
 publisher.setText("Publisher: "+ book.getPublisher());  
 authors.setText("Authors: "+ book.getAuthors());  
 year.setText("Year: "+ book.getYear());  
 pages.setText("Pages: "+ book.getPages());  
 desc.setText(book.getDesc());  
  
 }  
 else {Toast.*makeText*(this, "Information not found", Toast.*LENGTH\_LONG*).show(); finish();}  
 }  
  
 public void setBook(Book book) {  
 this.book = book;  
 }  
}

**JsonHandler.java**

package ua.kpi.comsys.IO7303.ui.library;  
  
import android.content.Context;  
import android.content.res.Resources;  
  
import com.google.gson.Gson;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.util.List;  
  
public class JsonHandler {  
 private static int *RESOURCE\_NAME*;  
 private static String *FILE\_USER\_NAME*;  
 private static Boolean *userFileEnable* = false;  
 private static File *f*;  
  
 public JsonHandler(int resourceName){  
 *RESOURCE\_NAME* = resourceName;  
 }  
  
 public static void setFileUserName(String fileUserName) {  
 *FILE\_USER\_NAME* = fileUserName;  
 }  
  
 public static void setUserFileEnable(Boolean userFileEnable) {  
 JsonHandler.*userFileEnable* = userFileEnable;  
 }  
  
 public static boolean exportToJSON(Context context, List<Book> dataList) {  
  
 Gson gson = new Gson();  
 DataItems dataItems = new DataItems();  
 dataItems.setBooks(dataList);  
 String jsonString = gson.toJson(dataItems);  
  
 FileOutputStream fileOutputStream = null;  
  
 try {  
 fileOutputStream = context.openFileOutput(*FILE\_USER\_NAME*, Context.*MODE\_PRIVATE*);  
 fileOutputStream.write(jsonString.getBytes());  
 *userFileEnable* = true;  
 return true;  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 if (fileOutputStream != null) {  
 try {  
 fileOutputStream.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return false;  
 }  
  
  
 public static List<Book> importBooksFromJSON(Context context) {  
 InputStreamReader streamReader = null;  
 FileInputStream fileInputStream = null;  
  
 try{  
 Gson gson = new Gson();  
  
 *f* = new File(context.getFilesDir() + "/"+*FILE\_USER\_NAME*);  
 if(*f*.exists()){  
 *userFileEnable* = true;  
 }  
 else{  
 try(FileWriter writer = new FileWriter(*f*)){  
 writer.write(*getStringFromRawFile*(context));  
 writer.flush();  
 *userFileEnable* = true;  
 }  
 catch(IOException ex){  
 ex.printStackTrace();  
 }  
 }  
  
 DataItems dataItems = gson.fromJson(*getStringFromRawFile*(context), DataItems.class);  
  
 return dataItems.getBooks();  
 }  
 catch (Exception ex){  
 ex.printStackTrace();  
 }  
 finally {  
 if (streamReader != null) {  
 try {  
 streamReader.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if (fileInputStream != null) {  
 try {  
 fileInputStream.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
  
 return null;  
 }  
  
 public static Book importBookFromJSON(Context context) {  
 InputStreamReader streamReader = null;  
 FileInputStream fileInputStream = null;  
  
 try{  
 Gson gson = new Gson();  
 Book book = gson.fromJson(*getStringFromRawFile*(context), Book.class);  
 return book;  
 }  
 catch (Exception ex){  
 ex.printStackTrace();  
 }  
 finally {  
 if (streamReader != null) {  
 try {  
 streamReader.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if (fileInputStream != null) {  
 try {  
 fileInputStream.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 return null;  
 }  
  
 private static class DataItems {  
 private List<Book> books;  
  
 List<Book> getBooks() {  
 return books;  
 }  
 void setBooks(List<Book> books) {  
 this.books = books;  
 }  
 }  
  
 public static String getStringFromRawFile(Context context) {  
 InputStream is = null;  
 if(!*userFileEnable*) {  
 Resources r = context.getResources();  
 is = r.openRawResource(*RESOURCE\_NAME*);  
 }  
 else {  
 try {  
 is = new FileInputStream(*f*);  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
  
 String myText = null;  
 try {  
 myText = *convertStreamToString*(is);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 try {  
 is.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return myText;  
 }  
  
 static String convertStreamToString(InputStream is) throws IOException {  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 int i = is.read();  
 while( i != -1)  
 {  
 baos.write(i);  
 i = is.read();  
 }  
 return baos.toString();  
 }  
}

**LibraryFragment.java**

package ua.kpi.comsys.IO7303.ui.library;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
  
import ua.kpi.comsys.IO7303.R;  
  
public class LibraryFragment extends Fragment {  
 private List<Book> books;  
 private List<Book> foundBooks = new ArrayList<>();  
 private List<Book> booksToShow = new ArrayList<>();  
 private BookAdapter adapter;  
 ListView listView;  
 String customBooksList = "books\_list\_custom.txt";  
 Boolean addStatus = false;  
 Boolean searchMode = false;  
 Book removedBook = null;  
 Boolean elemAddOnStop = false;  
  
  
 @Override  
 public void onResume() {  
 super.onResume();  
 if(addStatus){  
 requireActivity().recreate();  
 addStatus = false;  
 }  
 }  
  
 public View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 View root = inflater.inflate(R.layout.*fragment\_third\_tab*, container, false);  
  
 JsonHandler jsonHandler = new JsonHandler(R.raw.*bookslist*);  
 jsonHandler.*setFileUserName*(customBooksList);  
  
 books = jsonHandler.*importBooksFromJSON*(getContext());  
 listView = root.findViewById(R.id.*booksList*);  
  
 EditText searchRequest = root.findViewById(R.id.*searchField*);  
 Button searchBtn = root.findViewById(R.id.*buttonSearch*);  
 FloatingActionButton addBookButton = root.findViewById(R.id.*buttonAddBook*);  
  
 if(books != null){  
 adapter = new BookAdapter(getActivity(), R.layout.*book\_list*, books);  
  
 listView.setAdapter(adapter);  
 Toast.*makeText*(getContext(), "Loaded", Toast.*LENGTH\_LONG*).show();  
 }  
 else{  
 Toast.*makeText*(getContext(), "Load failed...", Toast.*LENGTH\_LONG*).show();  
 }  
  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() { //DETAIL  
 @Override  
 public void onItemClick(AdapterView<?> parent, View itemClicked, int position,  
 long id) {  
 Toast.*makeText*(getContext(), booksToShow.get((int)id).getTitle(),  
 Toast.*LENGTH\_SHORT*).show();  
  
 startActivity(new Intent(getContext(), BookDetail.class).putExtra("Isbn13", booksToShow.get((int)id).getIsbn13()));  
 }  
 });  
  
 addBookButton.setOnClickListener(new View.OnClickListener() { // ADD  
 public void onClick(View view) {  
 addStatus = true;  
 startActivity(new Intent(getContext(), AddBookActivity.class).putExtra("booklistId", R.raw.*bookslist*));  
 }  
 });  
  
 listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() { // DELETE  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View itemClicked, int position,  
 long id) {  
 if(!searchMode) {  
 itemClicked.setBackgroundResource(R.color.*green*);  
 try {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
 builder.setTitle("Deleting");  
 builder.setMessage("Do you want to delete this book?");  
 builder.setCancelable(true);  
 builder.setOnCancelListener(new DialogInterface.OnCancelListener() {  
 @Override  
 public void onCancel(DialogInterface dialog) {  
 itemClicked.setBackgroundResource(R.color.*white*);  
 }  
 });  
  
 builder.setPositiveButton(android.R.string.*yes*, new DialogInterface.OnClickListener() { // YES  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 removedBook = books.remove((int) id);  
 adapter.notifyDataSetChanged();  
 jsonHandler.*exportToJSON*(getContext(), books);  
 elemAddOnStop = true;  
 dialog.dismiss();  
 }  
  
 });  
 AlertDialog dialog = builder.create();  
 dialog.show();  
  
 } catch (Exception e) {  
 Toast.*makeText*(getContext(), "Deleting error", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 else Toast.*makeText*(getContext(), "To delete. you must leave the search mode", Toast.*LENGTH\_LONG*).show();  
 return true;  
 }  
 });  
  
 searchBtn.setOnClickListener(new View.OnClickListener() { // FIND  
 public void onClick(View view) {  
 String fieldText = searchRequest.getText().toString().toLowerCase();  
 BookAdapter adapter2;  
 foundBooks.clear();  
  
 if (fieldText.equals("!reset")){  
 File userFile = new File(view.getContext().getFilesDir() + "/" + customBooksList);  
  
 try(FileWriter writer = new FileWriter(userFile)){  
 jsonHandler.*setUserFileEnable*(false);  
 writer.write(jsonHandler.*getStringFromRawFile*(getContext()));  
 writer.flush();  
 }  
 catch(IOException ex){  
 ex.printStackTrace();  
 }  
 getActivity().recreate();  
 Toast.*makeText*(getContext(), "User list has been reset", Toast.*LENGTH\_LONG*).show();  
 adapter2 = new BookAdapter(getActivity(), R.layout.*book\_list*, books);  
 }  
  
 else if(!fieldText.equals("")){  
 searchMode = true;  
 for (int i = 0; i < books.size(); i++) {  
 if(books.get(i).getTitle().toLowerCase().contains(fieldText)){  
 foundBooks.add(books.get(i));  
 }  
 }  
  
 if(foundBooks.isEmpty()){  
 Toast.*makeText*(getContext(), "Nothing found", Toast.*LENGTH\_LONG*).show();  
 }  
 else Toast.*makeText*(getContext(), "OK", Toast.*LENGTH\_LONG*).show();  
 adapter2 = new BookAdapter(getActivity(), R.layout.*book\_list*, new ArrayList<>(foundBooks));  
 }  
 else {  
 searchMode = false;  
 adapter2 = new BookAdapter(getActivity(), R.layout.*book\_list*, books);  
 };  
 listView.setAdapter(adapter2);  
 }  
 });  
  
 return root;  
 }  
  
 private class BookAdapter extends ArrayAdapter<Book>{  
 BookAdapter(Context context, int textViewResourceId, List<Book> objects) {  
 super(context, textViewResourceId, objects);  
 booksToShow = objects;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater inflater = getLayoutInflater();  
 View row = inflater.inflate(R.layout.*book\_list*, parent, false);  
 TextView title = row.findViewById(R.id.*bookTitle*);  
 TextView subtitle = row.findViewById(R.id.*bookSubtitle*);  
 TextView price = row.findViewById(R.id.*bookPrice*);  
 TextView isbn13 = row.findViewById(R.id.*bookIsbn13*);  
  
 title.setText(handle(booksToShow.get(position).getTitle()));  
 subtitle.setText(handle(booksToShow.get(position).getSubtitle()));  
 price.setText("Price: " + handle(booksToShow.get(position).getPrice()));  
 isbn13.setText("Isbn13: " + handle(booksToShow.get(position).getIsbn13()));  
  
 ImageView currImg = row.findViewById(R.id.*image*);  
  
 String imageName = booksToShow.get(position).getImage();  
 int res = getContext().getResources().getIdentifier(imageName.replaceAll(".png",  
 ""), "drawable", getContext().getPackageName());  
  
 if(res!=0) currImg.setImageResource(res);  
 else currImg.setImageResource(R.drawable.*no\_image*);  
 return row;  
 }  
  
 public String handle(String str){  
 if(str.equals("")) return "None";  
 else return str;  
  
 }  
 }  
}