

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG

ĐỀ TÀI: Xây dựng ứng dụng đọc sách Ergon

Mã Sinh Viên	Họ và Tên	Ngày Sinh	Lớp
2151061207	Nguyễn Lương Nhật Minh	14/02/2003	63CNTT.NB
2051062464	Nguyễn Quang Mạnh	03/10/2002	62TH
2151061182	Vũ Tuấn Hoàng	21/09/2003	63CNTT.VA
2151062806	Trần Tùng Lâm	09/04/2003	63CNTT1

Hà Nội, năm 2024

LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, việc tiếp cận thông tin trở nên dễ dàng hơn bao giờ hết nhờ sự phát triển của các ứng dụng di động. Trong số đó, ứng dụng đọc sách đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của nhiều người, đặc biệt là giới trẻ. Nhận thấy tình hình đó và nhu cầu ngày càng cao của người dùng đối với việc đọc sách trực tuyến, chúng em quyết định thực hiện đề tài “Xây dựng ứng dụng đọc sách Ergon”.

Mục tiêu chính của đề tài này là tạo ra một ứng dụng thân thiện với người dùng, cung cấp kho sách phong phú và đa dạng.

MỤC LỤC

CHƯƠNG 1. MÔ TẢ BÀI TOÁN	6
1.1. Giới thiệu	6
1.2. Chức năng chính	6
1.3. Yêu cầu phi chức năng	6
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG.....	7
2.1. Phân tích yêu cầu:.....	7
2.2. Thiết kế hệ thống:	8
2.3. Triển khai:.....	11
2.4. Vận hành và bảo trì:.....	12
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN.....	13
3.1. Công nghệ đã sử dụng	13
3.2. Tiến độ thực hiện	13
3.3. Hình ảnh sản phẩm	15
KẾT LUẬN	23
Ưu điểm:	23
Nhược điểm:	23
Hướng phát triển:.....	23
DANH MỤC TÀI LIỆU THAM KHẢO	24

MỤC LỤC HÌNH ẢNH

Hình 1. 1 Sơ đồ lớp.....	10
Hình 1. 2 Lưu trữ dữ liệu trên firebase.....	10
Hình 1. 3 Giao diện đăng kí.....	15
Hình 1. 4 Giao diện đăng nhập.....	16
Hình 1. 5 Giao diện trang chủ	17
Hình 1. 6 Giao diện chi tiết sách	18
Hình 1. 7 Giao diện thư viện sách	19
Hình 1. 8 Giao diện đọc sách.....	20
Hình 1. 9 Giao diện tìm kiếm sách	21
Hình 1. 10 Giao diện lịch sử đọc sách.....	22

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

1.1. Giới thiệu

Ứng dụng đọc sách (Ergon) là một ứng dụng đọc sách đơn giản, cho phép người dùng tìm, đọc sách. Ứng dụng sẽ được xây dựng bằng Java, Firebase.

1.2. Chức năng chính

Ứng dụng Ergon cần có các chức năng sau:

- **Đăng nhập/Đăng ký:** tạo và lưu trữ tài khoản của người dùng.
- **Trang chủ:** Liệt kê tất cả các loại sách.
- **Hiển thị sách theo thể loại:** Liệt kê các loại sách theo thể loại.
- **Thông tin sách:** cho phép người dùng xem các thông tin của sách như tóm tắt nội dung, tác giả.
- **Tìm kiếm:** cho phép người dùng tìm kiếm sách theo tên sách hoặc tác giả
- **Tài khoản:** cho phép người dùng biết được những cuốn sách xem được lịch sử đọc.
- **Thư viện:** cho phép người dùng thấy được những quyển sách mình chưa đọc, đang đọc hoặc những quyển sách mình lưu vào yêu thích và những quyển đã đọc xong.
- **Đọc sách:** cho phép người dùng đọc 1 cuốn sách.

1.3. Yêu cầu phi chức năng

- **Dễ sử dụng:** Giao diện cần rõ ràng, dễ hiểu và dễ sử dụng.
- **Hiệu năng:** Ứng dụng cần hoạt động nhanh chóng và hiệu quả, ngay cả khi có nhiều công việc.
- **Độ tin cậy:** Dữ liệu công việc cần được lưu trữ an toàn và không bị mất mát.

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu:

Xác định người dùng:

- Người dùng là bất kỳ ai muốn sử dụng ứng dụng để đọc sách. Họ có thể là học sinh, sinh viên, nhân viên văn phòng, hoặc bất kỳ ai muốn đọc sách để tìm hiểu, giải trí, học tập.

- Người dùng có thể ở mọi lứa tuổi khác nhau, do đó ứng dụng cần dễ sử dụng và thân thiện với người dùng.

Thu thập yêu cầu:

- Dựa trên mô tả bài toán, ta đã xác định được các chức năng chính (Đăng nhập/Đăng ký, Trang chủ, Hiển thị sách theo thể loại, Thông tin sách, Tài khoản, Thư viện, Đọc sách) và yêu cầu phi chức năng (dễ sử dụng, hiệu năng, độ tin cậy) của ứng dụng.

Phân tích yêu cầu:

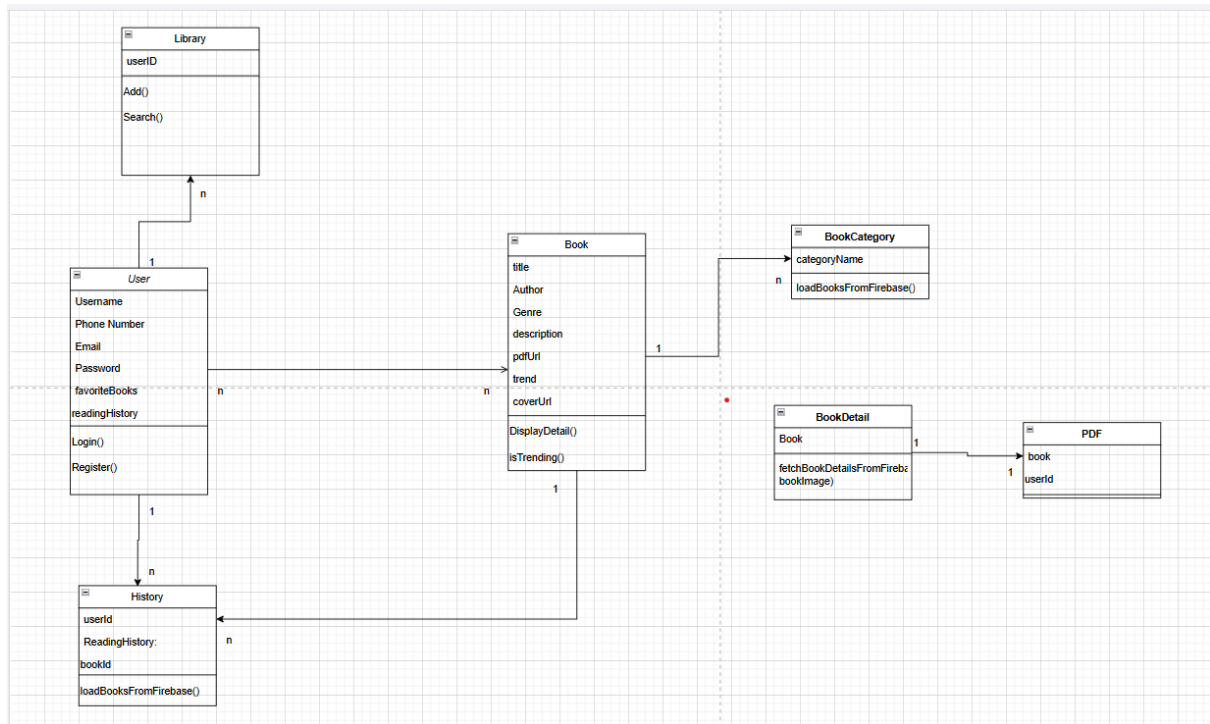
- **Đăng nhập/Đăng ký:** Yêu cầu người dùng đăng ký tài khoản để đọc, sau khi đăng ký thì phải đăng nhập.
- **Trang chủ:** Hiển thị tất cả các loại sách cùng với các thể loại, ví dụ như trending book, văn học.
- **Thông tin sách:** Người dùng xem các thông tin của sách như tóm tắt nội dung, tác giả và các loại sách liên quan
- **Tài khoản:** Người dùng sẽ biết được những cuốn sách đang đọc, xem được lịch sử đọc.
- **Thư viện:** cho phép người dùng thấy được những quyển sách mình chưa đọc, đang đọc hoặc những quyển sách mình lưu vào yêu thích và những quyển đã đọc xong.
- **Đọc sách:** cho phép người dùng đọc 1 cuốn sách.

2.2. Thiết kế hệ thống:

- + User: Quản lý thông tin người dùng, bao gồm các thuộc tính như: Username, Phone Number, Email, Password, favoriteBooks, readingHistory. Các phương thức liên quan là Login() và Register().
- + Book: Quản lý thông tin sách với các thuộc tính như title, author, genre, description, pdfUrl, trend, coverURL.

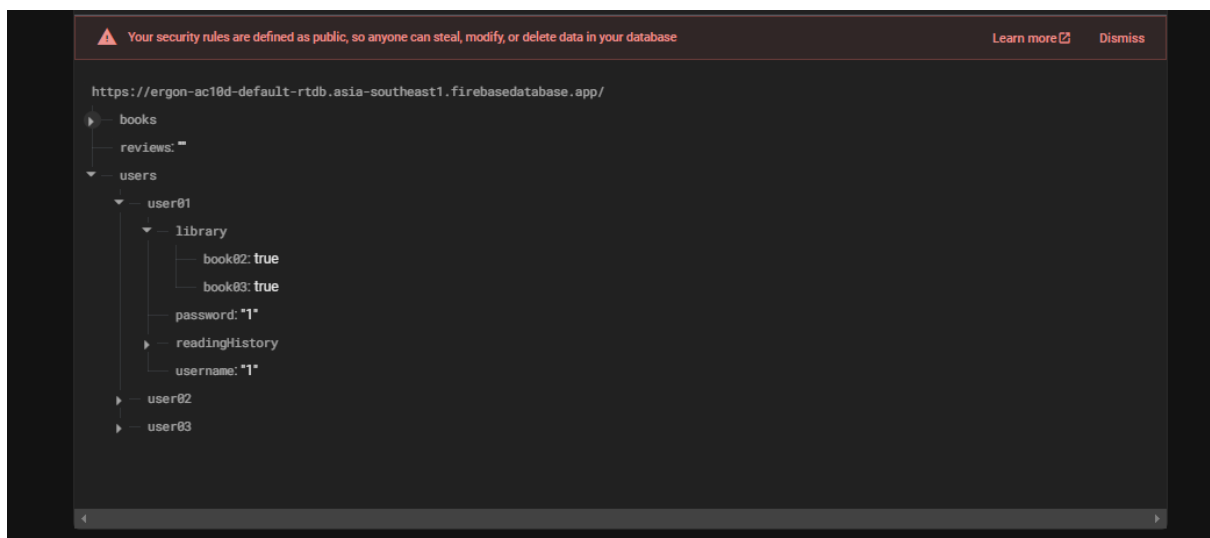
- + History: Quản lí lịch sử đọc của người dùng với `userId`, `ReadingHistory`, `BookId`. Phương thức `loadBooksFromFirebase()` được sử dụng để tải dữ liệu từ Firebase.
- + Library: Đóng vai trò là thư viện sách với `userID`, và các phương thức `Add()` và `Search()` để thêm và tìm kiếm sách.
- + BookCategory: Chứa `categoryName` để phân loại sách và phương thức `loadBooksFromFirebase()` để tải dữ liệu từ Firebase.
- + BookDetail: Hiển thị chi tiết của một cuốn sách với các thuộc tính `Book` và `bookImage`. Phương thức `fetchBookDetailsFromFirebase()` để lấy dữ liệu từ Firebase và hiển thị lên View
- + HomePage: Hiển thị sách ở trang chủ với các thuộc tính `Book` Phương thức `fetchBookDetailsFromFirebase()` để lấy dữ liệu từ Firebase và hiển thị lên View
- + PDF: Quản lý việc hiển thị tài liệu PDF của sách với các thuộc tính `Book` và `userId` để xác định người dùng và cuốn sách đang được đọc.
- + Library: Hiện thị sách đã được thêm vào thư viện với các thuộc tính `Book` Phương thức `fetchBookDetailsFromFirebase()` để lấy dữ liệu từ Firebase và hiển thị lên View

Mối quan hệ giữa các lớp:



Hình 1. 1 Sơ đồ lớp

Thiết kế cơ sở dữ liệu:



Hình 1. 2 Lưu trữ dữ liệu trên firebase

Thiết kế giao diện:

- Hiện thị menu chính với các lựa chọn:
 1. Đăng ký/Đăng nhập
 2. Trang chủ
 3. Thông tin sách
 4. Tài khoản, thẻ loại sách
 5. Thư viện
 6. Đọc sách
- Sử dụng các thông báo rõ ràng để hướng dẫn người dùng nhập liệu và hiển thị kết quả.
- Minh họa:
 - Giao diện đăng ký/ đăng nhập:
 - Giao diện trang chủ:
 - Giao diện thông tin sách:
 - Giao diện tìm kiếm sách:
 - Giao diện lịch sử đọc sách:
 - Giao diện tài khoản:

2.3. Triển khai:

Viết code: Sử dụng Java để cài đặt các adapter, fragment, model, ui.

Xây dựng các layout phù hợp với từng activity.

Sử dụng realtimeDatabase, firestore trên firebase để lưu trữ dữ liệu, xử lý dữ liệu liên quan đến sách.

2.4. Vận hành và bảo trì:

- Cài đặt và triển khai:
- Bảo trì: Sửa lỗi phát sinh, cập nhật chức năng mới (nếu có) và cải thiện hiệu năng của ứng dụng.

CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

3.1. Công nghệ đã sử dụng

- Ngôn ngữ lập trình: Java
- Công cụ: Android Studio
- Thư viện: (tùy chọn) có thể sử dụng các thư viện hỗ trợ đọc/ghi file, xử lý dữ liệu,

3.2. Tiến độ thực hiện

Link github tới dự án: https://github.com/Manh-Ne/CSE441_PROJECT

Hướng dẫn các bước đã thực hiện:

B1. Tạo dự án mới:

- Mở Android Studio.
- Chọn "Create New Project".
- Chọn "Java" làm ngôn ngữ lập trình.
- Chọn JDK phù hợp (ví dụ: JDK 11 hoặc mới hơn).
- Nhập tên dự án (ví dụ: "PROJECT").
- Chọn vị trí lưu trữ dự án.
- Nhấn "Finish" để tạo dự án.

B2. Tạo dự án trên firebase:

Chọn Create a project -> create Realtime Database và Firestore Database.

B3. Tạo các activity, layout, drawable:

- Trong cửa sổ "Project", click chuột phải vào thư mục "src".
- Chọn "New" -> "Package".
- Tạo các adapter, fragment, model, UI sau:

- Model: chứa lớp Book
- Adapter: BookAdapter, ButtonAdapter, CountryCodeAdapter, ImageOnlyAdapter, VanhocAdapter, ViewPagerAdapter.
- Fragment: HomePage, Library, Profile.
- UI: Ebook, History, Menu, PdfViewerActivity, VanHoc

B4. Tạo các lớp:

- Trong mỗi package, click chuột phải và chọn "New" -> "Java Class" để tạo các lớp tương ứng.

B5. Tạo các layout:

- Click chuột phải layout và chọn “New”->”Layout Resource File” để tạo phù hợp.

B5. Viết code:

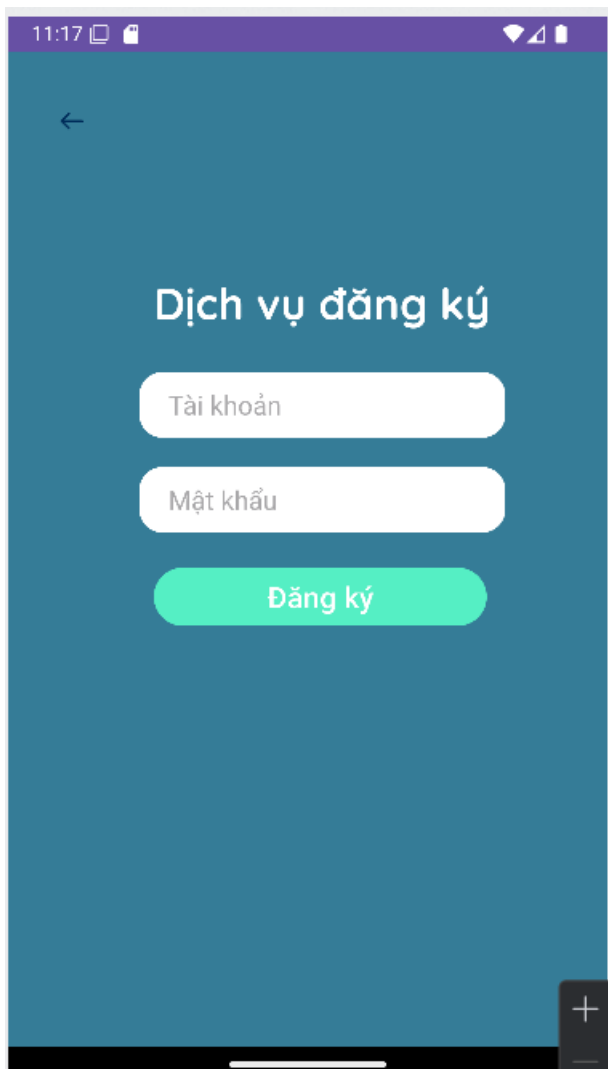
- Bắt đầu viết code cho từng activity, adapter, fragment, model, ui tương ứng với chức năng của ứng dụng.

B6. Chạy:

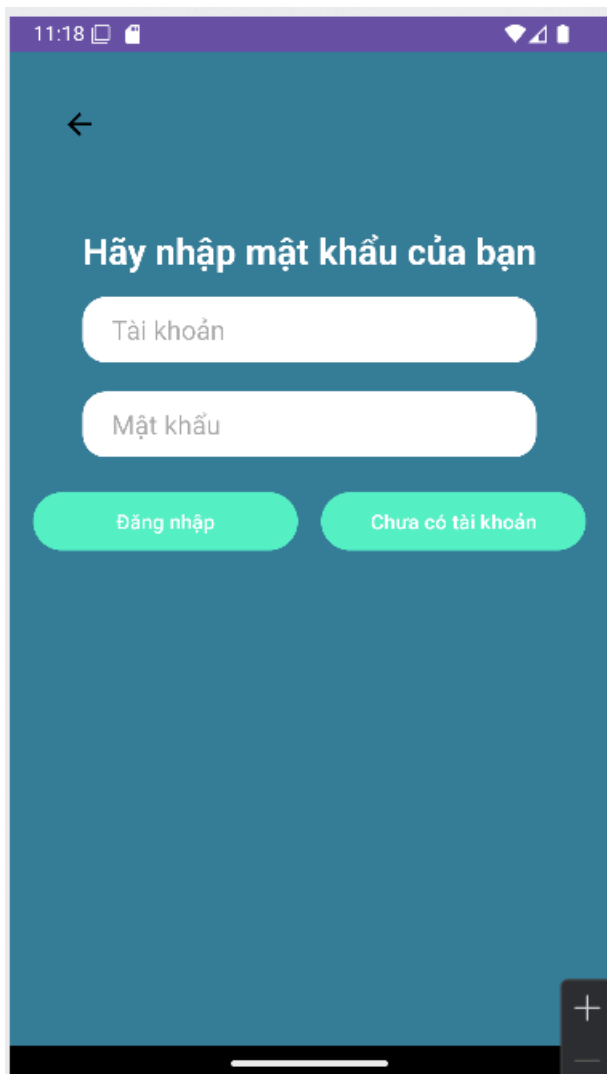
Chạy ứng dụng từ Android Studio bằng cách click vào dấu tam giác nằm ngang bên phải (Run)

Kiểm tra các chức năng của ứng dụng, sửa lỗi và hoàn thiện code.

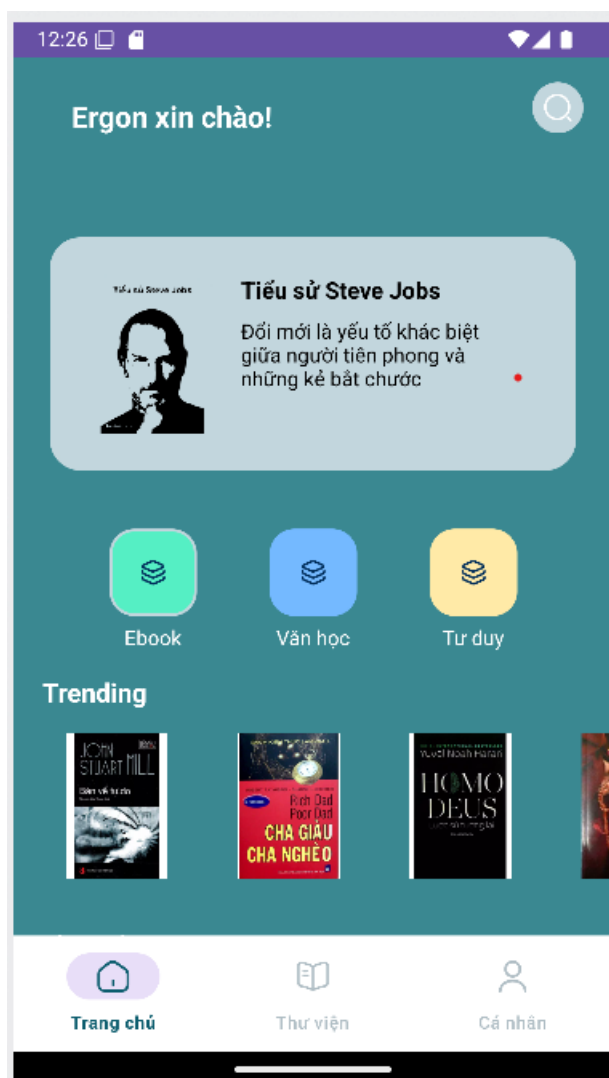
3.3. Hình ảnh sản phẩm



Hình 1. 3 Giao diện đăng kí



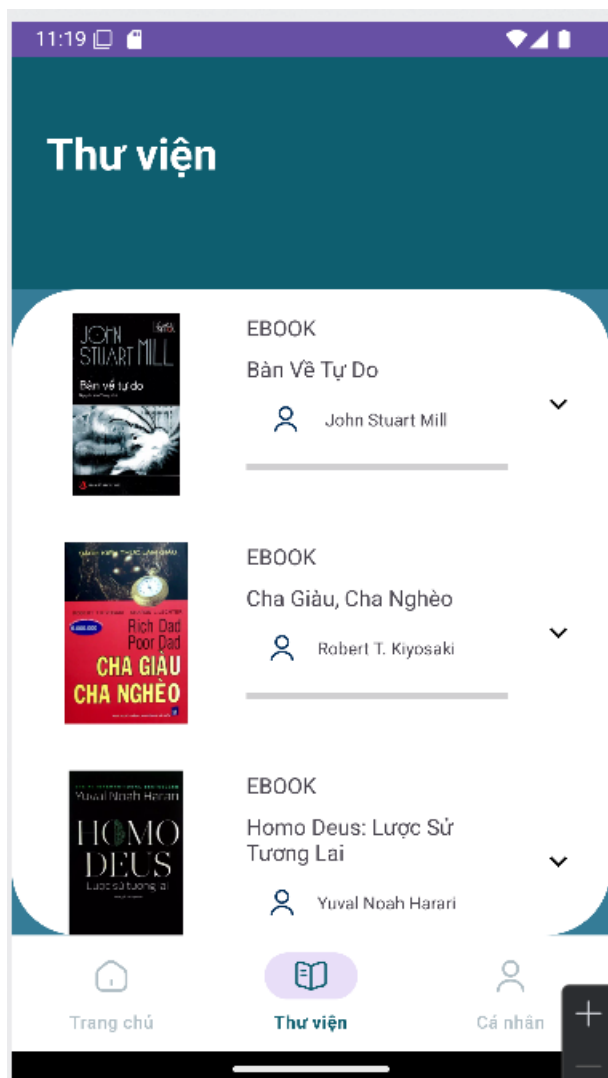
Hình 1. 4 Giao diện đăng nhập



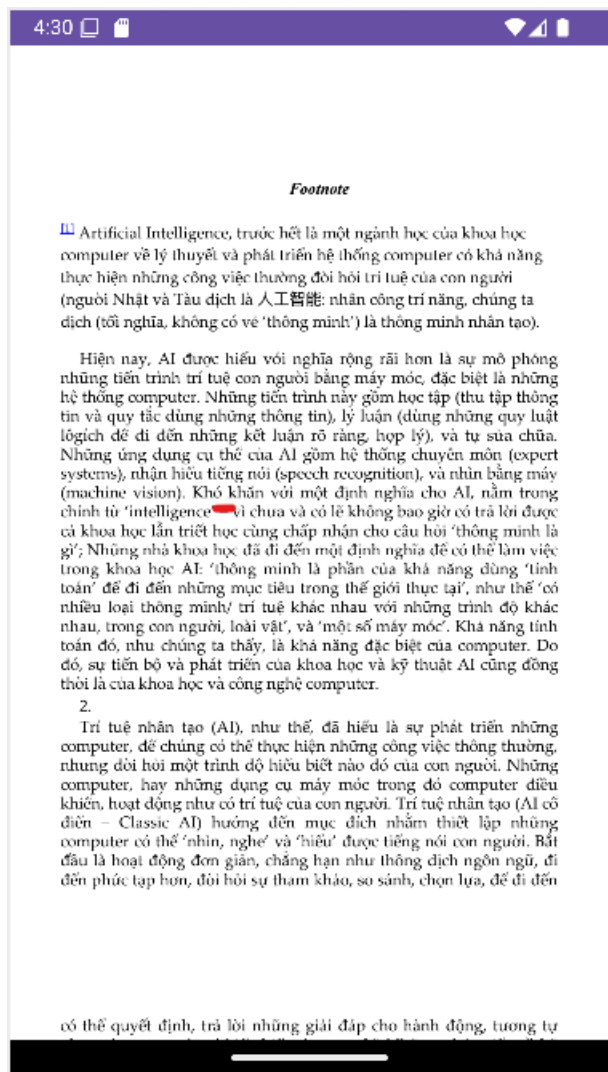
Hình 1. 5 Giao diện trang chủ



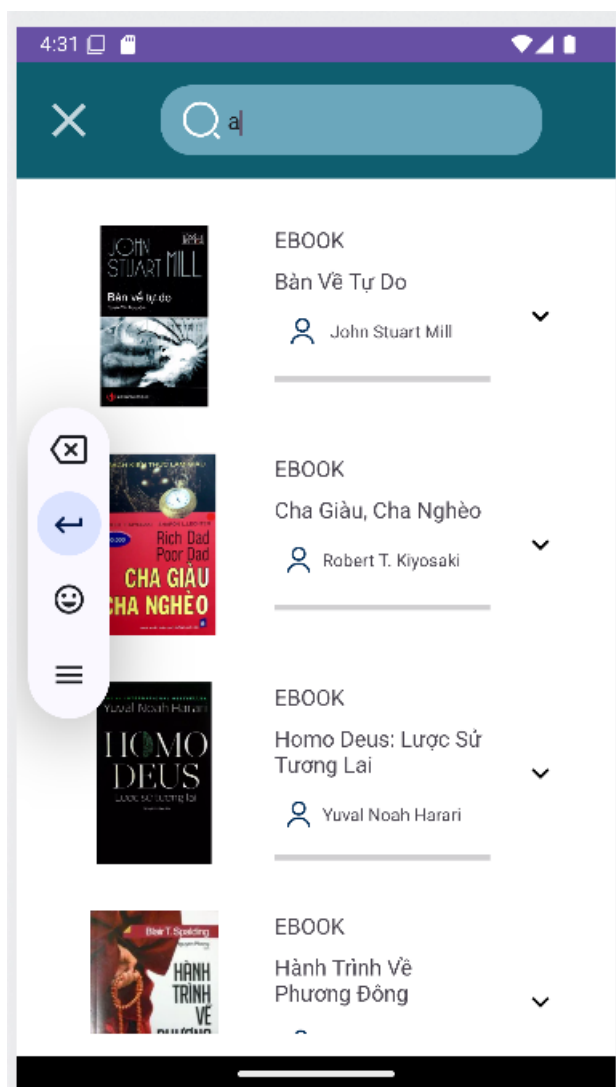
Hình 1. 6 Giao diện chi tiết sách



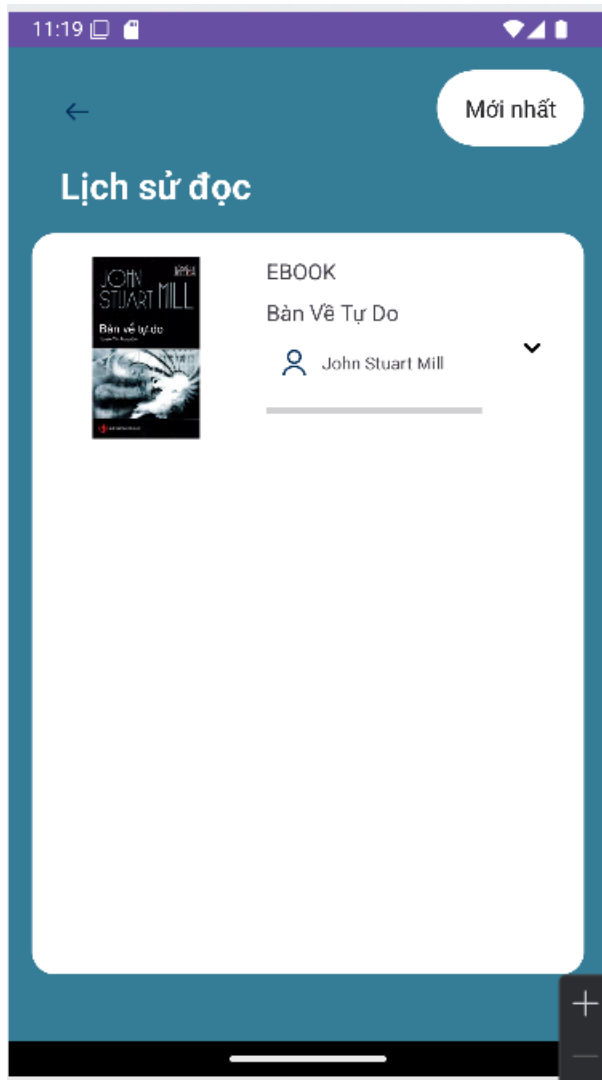
Hình 1. 7 Giao diện thư viện sách



Hình 1. 8 Giao diện đọc sách



Hình 1. 9 Giao diện tìm kiếm sách



Hình 1. 10 Giao diện lịch sử đọc sách

KẾT LUẬN

Ưu điểm:

- + Giúp chúng em hiểu được quy trình làm 1 dự án app android
- + Các chức năng như Đăng nhập/Đăng kí và Thư viện giúp quản lí dữ liệu người dùng, lịch sử đọc tốt hơn.
- + Trang chủ, Hiển thị sách theo thể loại và tìm kiếm giúp người dùng nhanh chóng tiếp cận sách
- + Chức năng đọc sách cung cấp trải nghiệm đọc trực tiếp trên ứng dụng, tiết kiệm thời gian

Nhược điểm:

- + Giao diện bố cục chưa tối ưu hóa, trải nghiệm người dùng chưa tốt.
- + Dữ liệu quản lí còn ít
- + Một số chức năng vẫn chưa được hoàn thiện tốt
- + Vẫn chưa tạo được phần cho admin để có thể thêm sửa xóa sách

Hướng phát triển:

- + Cải thiện giao diện tăng cường trải nghiệm người dùng
- + Bổ sung thêm phần xếp hạng sách để giúp người dùng dễ tìm kiếm theo nhu cầu.
- + Bổ sung thêm phần ghi chú, highlight giúp người đọc quản lí thông tin tiện lợi hơn.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Android Studio Tutorial: <https://www.geeksforgeeks.org/android-studio-tutorial/>
- [2]. Coding Android Apps của Margaret Kozak Polk
file:///C:/Users/ADMIN/Documents/Zalo%20Received%20Files/Margaret%20Kozak%20Polk%20-%20Coding%20Android%20Apps%20(2024,%20Chapman%20and%20Hall_CRC)%20[10.1201_9781003286325]%20-%20libgen.li.pdf

Phụ lục

Code minh họa đầy đủ

- Book.java

```
package com.example.project.model;

import java.io.Serializable;

public class Book implements Serializable {
    private String title;
    private String author;
    private String contentDescription;
```



```
private String coverUrl;
private String genre;
private String pdfUrl;
private Boolean trend;
private String content;

public Book() {
}

public Book(String title, String author, String
contentDescription, String coverUrl, String genre, String pdfUrl,
Boolean trend, String content) {
    this.title = title;
    this.author = author;
    this.contentDescription = contentDescription;
    this.coverUrl = coverUrl;
    this.genre = genre;
    this.pdfUrl = pdfUrl;
    this.trend = trend;
    this.content = content;
}

public String getContent() {
    return content;
}

public Boolean getTrend() {
    return trend;
}

public String getTitle() {
    return title;
}

public String getAuthor() {
    return author;
}

public String getContentDescription() {
    return contentDescription;
}

public String getCoverUrl() {
```

```

        return coverUrl;
    }

    public String getGenre() {
        return genre;
    }

    public String getPdfUrl() {
        return pdfUrl;
    }
}

```

- **BookAdapter**

```

package com.example.project.adapter;

import android.content.Context;
import android.content.Intent;
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 com.bumptech.glide.Glide;
import com.example.project.BookDetail;
import com.example.project.R;
import com.example.project.model.Book;

import java.util.List;

public class BookAdapter extends
RecyclerView.Adapter<BookAdapter.BookViewHolder> {
    private List<Book> booksList;
    private Context context;
}

```

```

public BookAdapter(Context context, List<Book> booksList)
{
    this.context = context;
    this.booksList = booksList;
}

@NonNull
@Override
public BookViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.ite
m_history, parent, false);
    return new BookViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull BookViewHolder
holder, int position) {
    Book book = booksList.get(position);
    holder.title.setText(book.getTitle());
    holder.author.setText(book.getAuthor());
    Glide.with(holder.itemView.getContext())
        .load(book.getCoverUrl())
        .into(holder.imageView);
    holder.itemView.setOnClickListener(v -> {
        Intent intent = new Intent(context,
BookDetail.class);
        intent.putExtra("book_image",
book.getCoverUrl());
        context.startActivity(intent);
    });
}

@Override
public int getItemCount() {
    return booksList.size();
}

static class BookViewHolder extends
RecyclerView.ViewHolder {

```

```

        TextView title, author;
        ImageView imageView;

        public BookViewHolder(@NonNull View itemView) {
            super(itemView);
            imageView =
itemView.findViewById(R.id.imageView);
            title = itemView.findViewById(R.id.tv2);
            author = itemView.findViewById(R.id.tv3);
        }
    }
}

```

- **HomePage**

```

package com.example.project.fragment;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.project.BookDetail;
import com.example.project.SearchActivity;
import com.example.project.adapter.ImageOnlyAdapter;
import com.example.project.R;
import com.example.project.SpaceItemDecoration;
import com.example.project.ui.Ebook;

```

```

import com.example.project.ui.VanHoc;
import com.example.project.model.Book;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class HomePage extends Fragment {
    private RecyclerView recyclerTrending, recyclerDexuat;
    private ImageOnlyAdapter trendingAdapter, dexuatAdapter;
    // Chinh sửa
    private List<String> trendingBookImages,
dexuatBookImages; // Chinh sửa
    private DatabaseReference databaseReference;
    private ImageButton btnVanHoc, btnEbook;
    private ImageView btnSearch;
    @SuppressWarnings("MissingInflatedId")
    @Override
    public View onCreateView(@NonNull LayoutInflater
inflater, @Nullable ViewGroup container, @Nullable Bundle
savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment_home_page, container,
false);

        recyclerTrending =
view.findViewById(R.id.recycler_trending);
        LinearLayoutManager trendingLayoutManager = new
LinearLayoutManager(getContext(),
LinearLayoutManager.HORIZONTAL, false);

recyclerTrending.setLayoutManager(trendingLayoutManager);

        trendingBookImages = new ArrayList<>();
        trendingAdapter = new
ImageOnlyAdapter(trendingBookImages, this::openBookDetail);
        recyclerTrending.setAdapter(trendingAdapter);

```

```

        recyclerDexuat =
view.findViewById(R.id.recycler_dexuat);
        LinearLayoutManager dexuatLayoutManager = new
LinearLayoutManager(getContext(),
LinearLayoutManager.HORIZONTAL, false);
        recyclerDexuat.setLayoutManager(dexuatLayoutManager);

        dexuatBookImages = new ArrayList<>();
        dexuatAdapter = new
ImageOnlyAdapter(dexuatBookImages, this::openBookDetail);
        recyclerDexuat.setAdapter(dexuatAdapter);

        int spaceInPixels =
getResources().getDimensionPixelSize(R.dimen.item_space);
        recyclerTrending.addItemDecoration(new
SpaceItemDecoration(spaceInPixels));
        recyclerDexuat.addItemDecoration(new
SpaceItemDecoration(spaceInPixels));

        databaseReference =
FirebaseDatabase.getInstance().getReference("books");
        fetchBooksFromFirebase();

        btnSearch = view.findViewById(R.id.ic_search);
        btnVanHoc = view.findViewById(R.id.ic_van_hoc);
        btnEbook = view.findViewById(R.id.ic_ebook);

        btnSearch.setOnClickListener(v -> {
            Intent intent = new Intent(getActivity(),
SearchActivity.class);
            startActivity(intent);
        });
        btnVanHoc.setOnClickListener(v -> {
            Intent intent = new Intent(getActivity(),
VanHoc.class);
            startActivity(intent);
        });
        btnEbook.setOnClickListener(v -> {
            Intent intent = new Intent(getActivity(),
Ebook.class);

```

```

        startActivity(intent);
    });

    return view;
}

private void fetchBooksFromFirebase() {
    databaseReference.addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
snapshot) {

            trendingBookImages.clear(); // Chỉnh sửa
            dexuatBookImages.clear(); // Chỉnh sửa
            int trendingCount = 0;
            int dexuatCount = 0;
            for (DataSnapshot dataSnapshot :
snapshot.getChildren()) {
                Book book =
dataSnapshot.getValue(Book.class);
                if (book != null) {
                    if (trendingCount < 10) {

trendingBookImages.add(book.getCoverUrl()); // Chỉnh sửa
                        trendingCount++;
                    } else if (dexuatCount < 10) {

dexuatBookImages.add(book.getCoverUrl()); // Chỉnh sửa
                        dexuatCount++;
                    }
                }
            }
            trendingAdapter.notifyDataSetChanged(); //
Chỉnh sửa
            dexuatAdapter.notifyDataSetChanged(); //
Chỉnh sửa
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
error) {

```

```

        Toast.makeText(getApplicationContext(), "Failed to load
books.", Toast.LENGTH_SHORT).show();
    }

});

}

private void openBookDetail(String bookImage) {
    Intent intent = new Intent(getActivity(),
BookDetail.class);
    intent.putExtra("book_image", bookImage);
    startActivity(intent);
}
}

```

History

```

package com.example.project.ui;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.project.BookDetail;
import com.example.project.R;
import com.example.project.adapter.BookAdapter;
import com.example.project.model.Book;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

```



```

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class History extends AppCompatActivity {

    private RecyclerView recyclerView;
    private BookAdapter bookAdapter;
    private List<Book> bookList;
    private Spinner spinner;
    private ImageButton btnBack;
    private DatabaseReference booksRef;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        initializeViews();
        setUpSpinner();
        setUpRecyclerView();
        loadBooksFromFirebase();
        setListeners();
    }

    private void initializeViews() {
        btnBack = findViewById(R.id.button_back);
        recyclerView = findViewById(R.id.recyclerView);
        spinner = findViewById(R.id.spinner);
        booksRef =
FirebaseDatabase.getInstance().getReference("books");
    }

    private void setUpSpinner() {
        String[] items =
getResources().getStringArray(R.array.spinner_items);

```

```

        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, items);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_drop
down_item);

        spinner.setAdapter(adapter);
    }

    private void setUpRecyclerView() {
        bookList = new ArrayList<>();
        bookAdapter = new BookAdapter(this, bookList);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        recyclerView.setAdapter(bookAdapter);
    }

    private void loadBooksFromFirebase() {
        // Lấy userId từ SharedPreferences
        SharedPreferences sharedPreferences =
getSharedPreferences("UserPrefs", Context.MODE_PRIVATE);
        String userId = sharedPreferences.getString("user_id",
null);

        if (userId != null) {
            DatabaseReference historyRef =
FirebaseDatabase.getInstance().getReference("users").child(userId).c
hild("readingHistory");

            historyRef.addValueEventListener(new
ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
                    if (dataSnapshot.exists()) {
                        bookList.clear();
                        for (DataSnapshot bookSnapshot :
dataSnapshot.getChildren()) {
                            String bookId = bookSnapshot.getKey();
                            loadBookDetails(bookId); // Gọi phương
thức để lấy chi tiết sách
                        }
                    }
                }
            });
        }
    }

```

```

        } else {
            Toast.makeText(History.this, "Không có lịch
sử đọc nào", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
        Toast.makeText(History.this, "Failed to load
reading history: " + databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
} else {
    Toast.makeText(this, "Người dùng không hợp lệ. Vui lòng
đăng nhập lại.", Toast.LENGTH_SHORT).show();
    finish(); // Đóng activity nếu không có người dùng
}
}

private void loadBookDetails(String bookId) {
    DatabaseReference bookRef = booksRef.child(bookId);
    bookRef.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            Book book = dataSnapshot.getValue(Book.class);
            if (book != null) {
                book.setId(bookId);
                bookList.add(book);
                bookAdapter.notifyDataSetChanged();
            }
        }
    })

    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
        Toast.makeText(History.this, "Failed to load book

```

```

        details: " + databaseError.getMessage(), Toast.LENGTH_SHORT).show();
    }
    });
}

private void setListeners() {
    btnBack.setOnClickListener(v -> finish());
}

private void openBookDetail(String bookId) {
    Intent intent = new Intent(History.this, BookDetail.class);
    intent.putExtra("book_id", bookId);
    startActivity(intent);
}
}

```

- **LibraryFragment**

```

• package com.example.project.fragment;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.project.ui.BookDetailActivity;
import com.example.project.R;
import com.example.project.adapter.BookAdapter;
import com.example.project.model.Book;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

```

```

public class LibraryFragment extends Fragment {

    private BookAdapter bookAdapter;
    private List<Book> bookList;
    private DatabaseReference databaseReference;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater
inflater, ViewGroup container, Bundle savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment_library, container,
false);

        RecyclerView rvBookList =
view.findViewById(R.id.recyclerView);
        rvBookList.setLayoutManager(new
LinearLayoutManager(getContext()));
        bookList = new ArrayList<>();
        bookAdapter = new BookAdapter(getContext(),
bookList);
        rvBookList.setAdapter(bookAdapter);

        // Firebase reference to the library
        databaseReference =
FirebaseDatabase.getInstance().getReference("users").child("l
ibrary");

        // Fetch data from Firebase
        loadBooksFromFirebase();

        return view;
    }

    private void loadBooksFromFirebase() {
        // Lấy userId từ SharedPreferences
        SharedPreferences sharedPreferences =
getContext().getSharedPreferences("UserPrefs",
Context.MODE_PRIVATE);
        String userId =
sharedPreferences.getString("user_id", null);

        if (userId != null) {
            DatabaseReference libraryRef =
FirebaseDatabase.getInstance().getReference("users").child(us
erId).child("library");

            libraryRef.addValueEventListener(new
ValueEventListener() {
                @Override
                public void onDataChange(@NonNull
DataSnapshot dataSnapshot) {
                    if (dataSnapshot.exists()) {
                        bookList.clear();
                        for (DataSnapshot bookSnapshot :
dataSnapshot.getChildren()) {
                            String bookId =
bookSnapshot.getKey();
                            loadBookDetails(bookId); // Gọi
phương thức để lấy chi tiết sách

```

```

    }
    } else {
        Toast.makeText(getContext(), "Không
có sách trong thư viện!", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onCancelled(@NonNull
DatabaseError databaseError) {
    Toast.makeText(getContext(), "Failed to
load reading history: " + databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
}
});
} else {
    Toast.makeText(getContext(), "Người dùng không
hợp lệ. Vui lòng đăng nhập lại.", Toast.LENGTH_SHORT).show();
    getActivity().finish();
}
}

private void loadBookDetails(String bookId) {
    DatabaseReference bookRef =
FirebaseDatabase.getInstance().getReference("books").child(bookId); // Tham chiếu đến sách
    bookRef.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            Book book =
dataSnapshot.getValue(Book.class);
            if (book != null) {
                book.setId(bookId);
                bookList.add(book);
                bookAdapter.notifyDataSetChanged();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

            Toast.makeText(LibraryFragment.this.getContext(), "Failed to
load book details: " + databaseError.getMessage(),
Toast.LENGTH_SHORT).show(); // Sửa thành Library.this
        }
    });
}

private void openBookDetail(String bookId) {
    Intent intent = new Intent(getContext(),
BookDetailActivity.class);
    intent.putExtra("book_id", bookId);
    startActivity(intent);
}
}

```

- ProfileFragment

```
• package com.example.project.fragment;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.example.project.ui.MainActivity;
import com.example.project.ui.HistoryActivity;
import com.example.project.R;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class ProfileFragment extends Fragment {
    private TextView tvLib, tvUser;
    private ImageButton btnLogout;

    @SuppressLint("MissingInflatedId")
    @Override
    public View onCreateView(LayoutInflater inflater,
        ViewGroup container, Bundle savedInstanceState) {
        View view =
inflater.inflate(R.layout.fragment_profile, container,
false);

        tvLib = view.findViewById(R.id.tv_Lib);
        tvUser = view.findViewById(R.id.tv_name);

        btnLogout = view.findViewById(R.id.logout);
        SharedPreferences sharedPreferences =
getActivity().getSharedPreferences("UserPrefs",
Context.MODE_PRIVATE);
        String userId =
sharedPreferences.getString("user_id", null);

        if (userId != null) {
            CountBook(userId, tvLib); // Gọi hàm CountBook để
đếm sách
            Nameuser(userId, tvUser);
        }

        LinearLayout linearHistory =
view.findViewById(R.id.linear_History);
        btnLogout.setOnClickListener(new
```

```

View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder dialog = new
AlertDialog.Builder(getActivity())
            .setTitle("Xác nhận đăng xuất")
            .setMessage("Bạn chắc chắn muốn đăng
xuất không?")
            .setNegativeButton("Không",
(dialogInterface, i) -> dialogInterface.dismiss())
            .setPositiveButton("Có",
(dialogInterface, i) -> {

                Intent intent = new
Intent(getActivity(), MainActivity.class);

                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
                getActivity().finish();
            });

        dialog.show();
    }
});

linearHistory.setOnClickListener(v -> {
    Intent intent = new Intent(getActivity(),
HistoryActivity.class);
    startActivity(intent);
});

return view;
}

public void CountBook(String userId, TextView tvLib) {
    DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReference("users").child(us
erId).child("library");
    databaseReference.addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
snapshot) {
            int count = (int)
snapshot.getChildrenCount();
            tvLib.setText(count + " cuốn sách");
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
error) {
            // Xử lý khi có lỗi xảy ra
        }
    });
}

public void Nameuser(String userId, TextView tvUser) {
    DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReference("users").child(us
erId).child("username");

```



```
        databaseReference.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
snapshot) {
                String name =
snapshot.getValue(String.class);
                tvUser.setText(name);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
error) {

            }
        });
    }
}
```