UETLib

Service Oriented Architecture



Author: Group 8 – SOA Course

Instructor: Dr. Vo Đinh Hieu

September 29, 2015

# Course project

## Instructor

Dr. Vo Dinh Hieu

University of Engineering and Technology – VNU

## Team members

* Truong Quoc Tuan (Leader) - K57CA Student ID: 12020416
* Nguyen Tuan Phong - K57CA Student ID: 12020288
* Le Van Giap - K57CA Student ID: 12020493

Table of Contents

[Requirement 4](#_Toc431429186)

[Overview 5](#_Toc431429187)

[Implementation 5](#_Toc431429188)

[Server (include website for students) 6](#_Toc431429189)

[Desktop application 11](#_Toc431429190)

[Mobile application 14](#_Toc431429191)

# Requirement

We are required to build an online library system for students.

Students can use web browser or mobile application to view books’ information that the library has.

Librarians have a desktop application to manage books in the library database with many functions such as view book lists, add book to database, remove book from database, and modify book information.

# Overview

UEBlib is an online library system for UET students.

UETlib consists three parts including web application, desktop application and mobile application. The web and mobile application are for students, and the desktop application, with more functions, is for librarians.

This report describes the system requirements, overview and our work on implementation.

Our system will function like the picture below:



We have deployed our system on our private server: <http://128.199.89.183:3000/>

If you have not had account yet, signup new one to login.

We also public our project on github: <https://github.com/tuantq57/soa-assignment1>

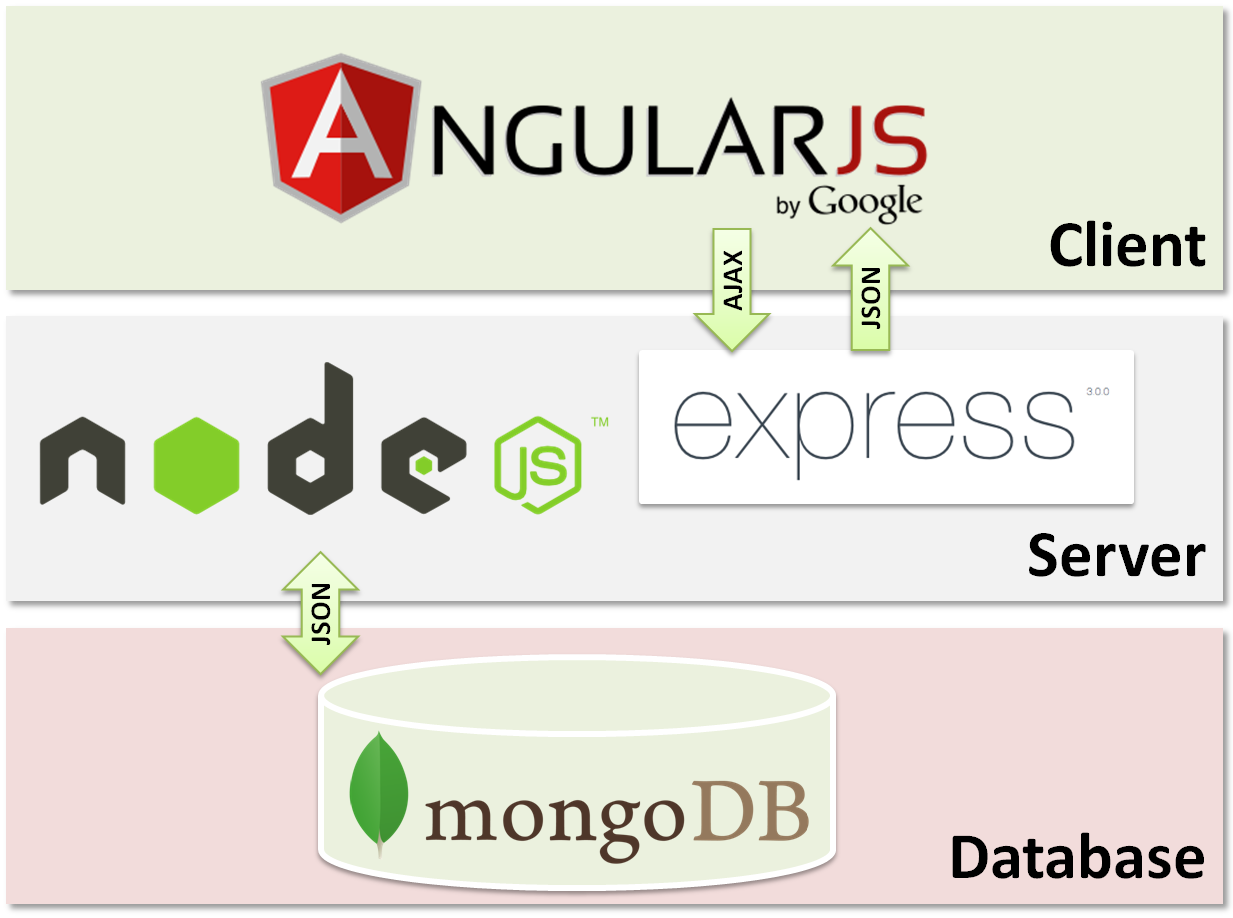
For the mobile application, everyone can download the UETLib.apk file (virus-free) and install in Android smartphone via the following URL if interested:

<https://github.com/tuantq57/soa-assignment1/releases>

# Implementation

## Server (include website for students)

For the server, we use MEAN Framework ([MongoDB](https://www.mongodb.org/), [ExpressJS](http://expressjs.com/), [AngularJS](https://angularjs.org/) and [Node.js](https://nodejs.org/)) to build our system including back-end and front-end (website).   
MEAN is new technology for full-stack web with many powerful tools and plugins and wide supported community. It also organizes system as MVC model, which is very familiar with every web developer.

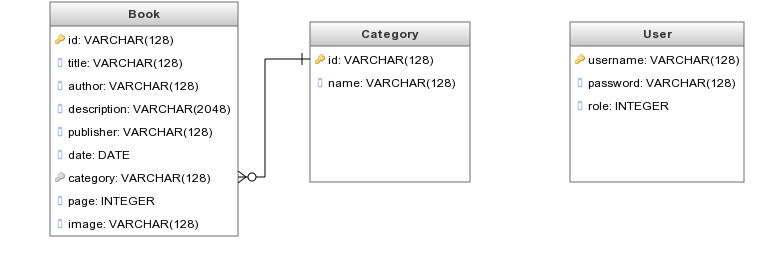


We will explain why we choose MEAN Framework for some reasons below:

* Node.js allows us to use JavaScript on the back-end as well as the front-end which can save us from having to learn a separate language.
* The NoSQL nature of MongoDB allows us to quickly and easily manage the data, we can flexibility to create data structures, which is a very valuable attribute when we are trying to build a product without clear specifications.
* Because of using JavaScript to implement front-end and back-end, we use JSON object to communicate all the time between out modules. It is the big plus point of MEAN Framework.
* Finally, these technologies have a lot of community support behind them so finding answers to questions is going to be much easier.

We use [MongoLab](https://mongolab.com/) which is a database web service to store our data. They provide account with free storage of 500MB and built-in MongoDB server-side tools.

Here is some schemas used in our project:

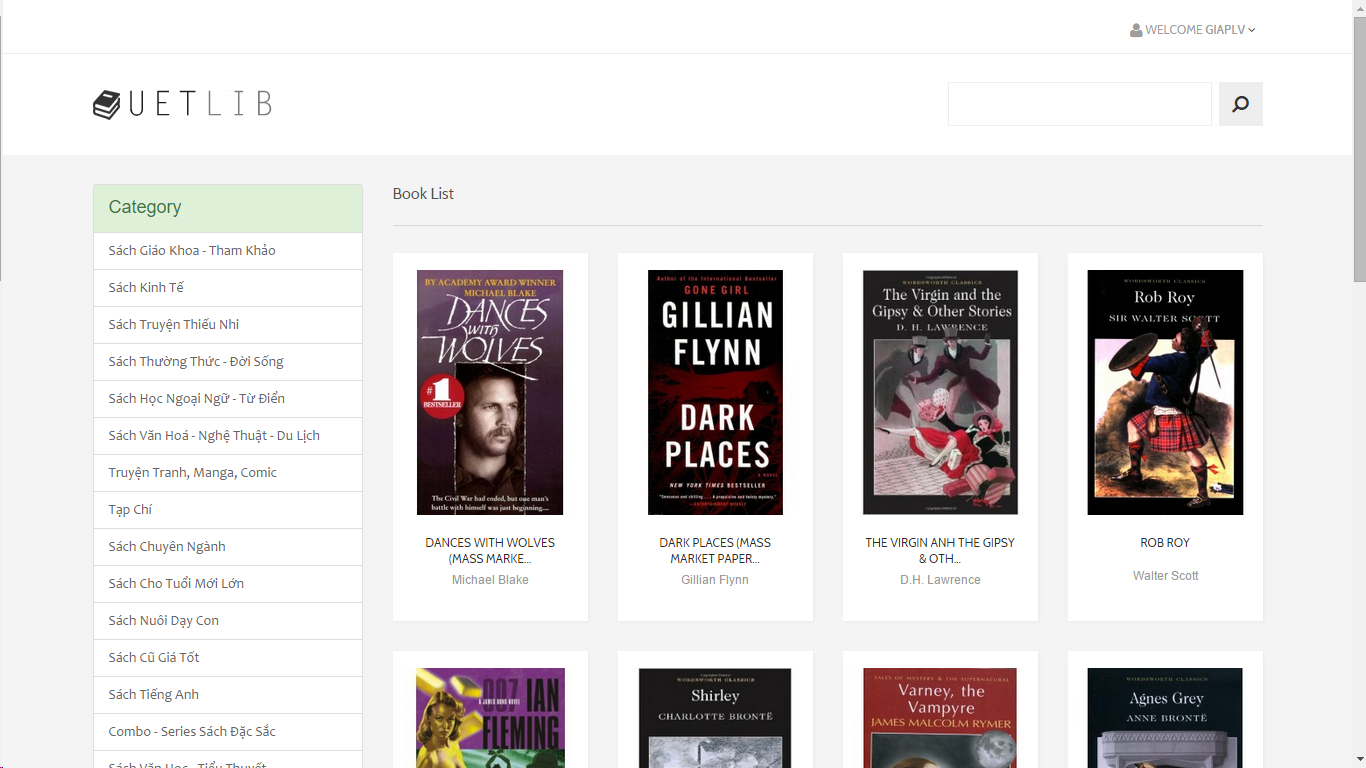


There are some screenshots we took from our website:

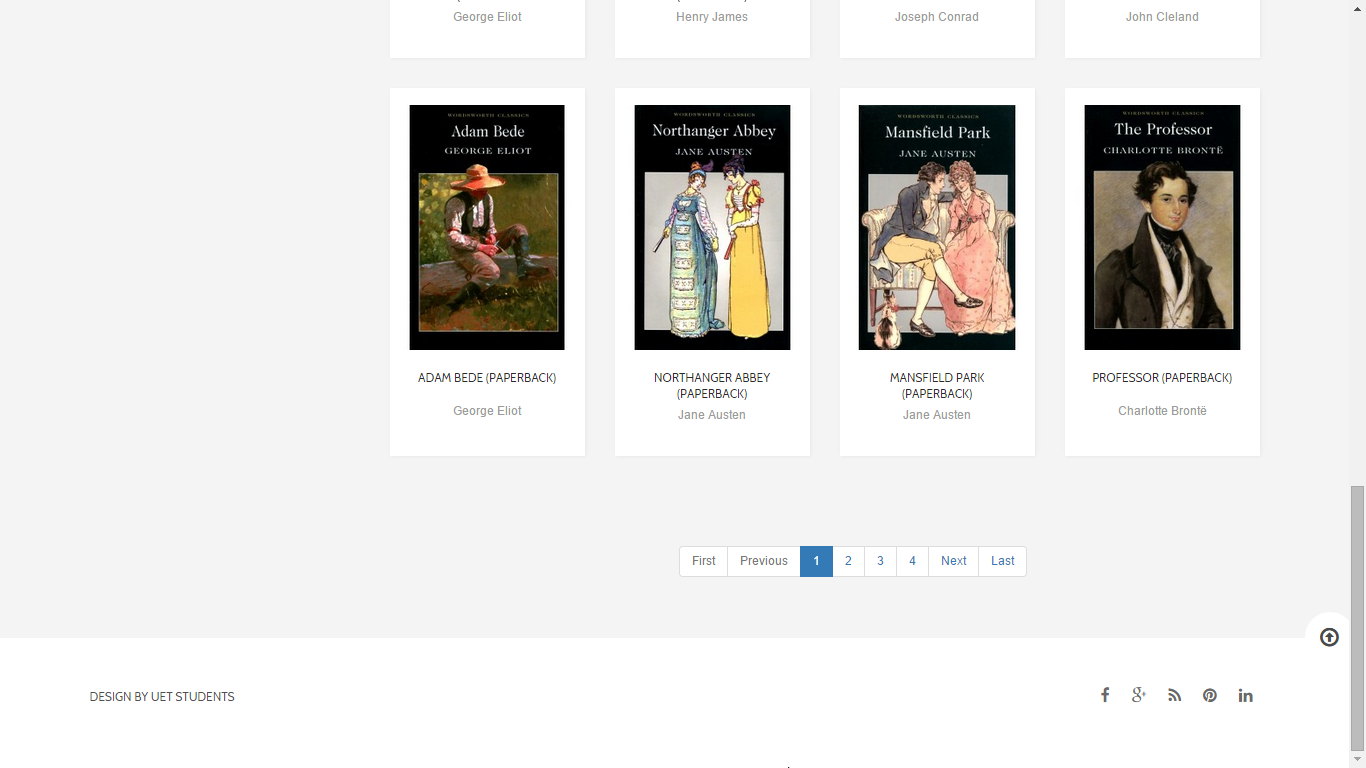
* Login form



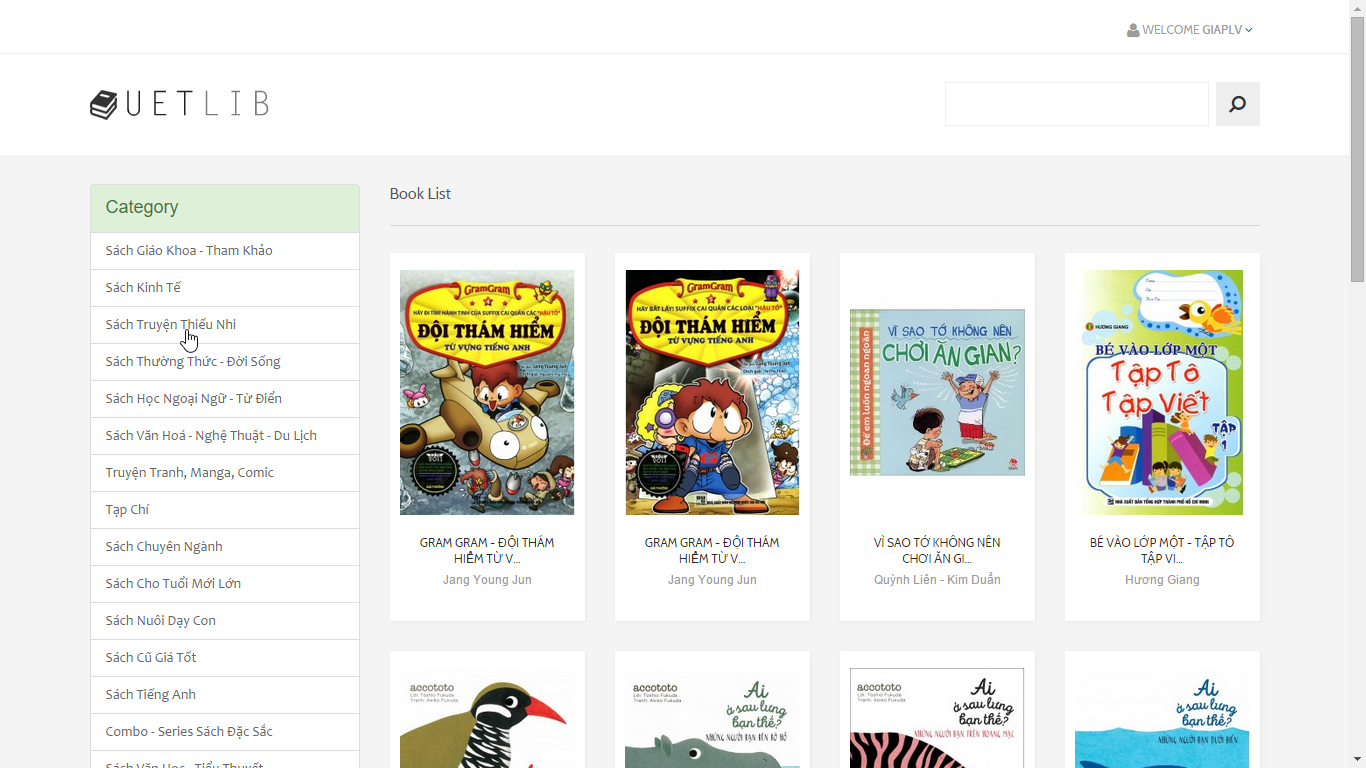
* After login, students can access the main page



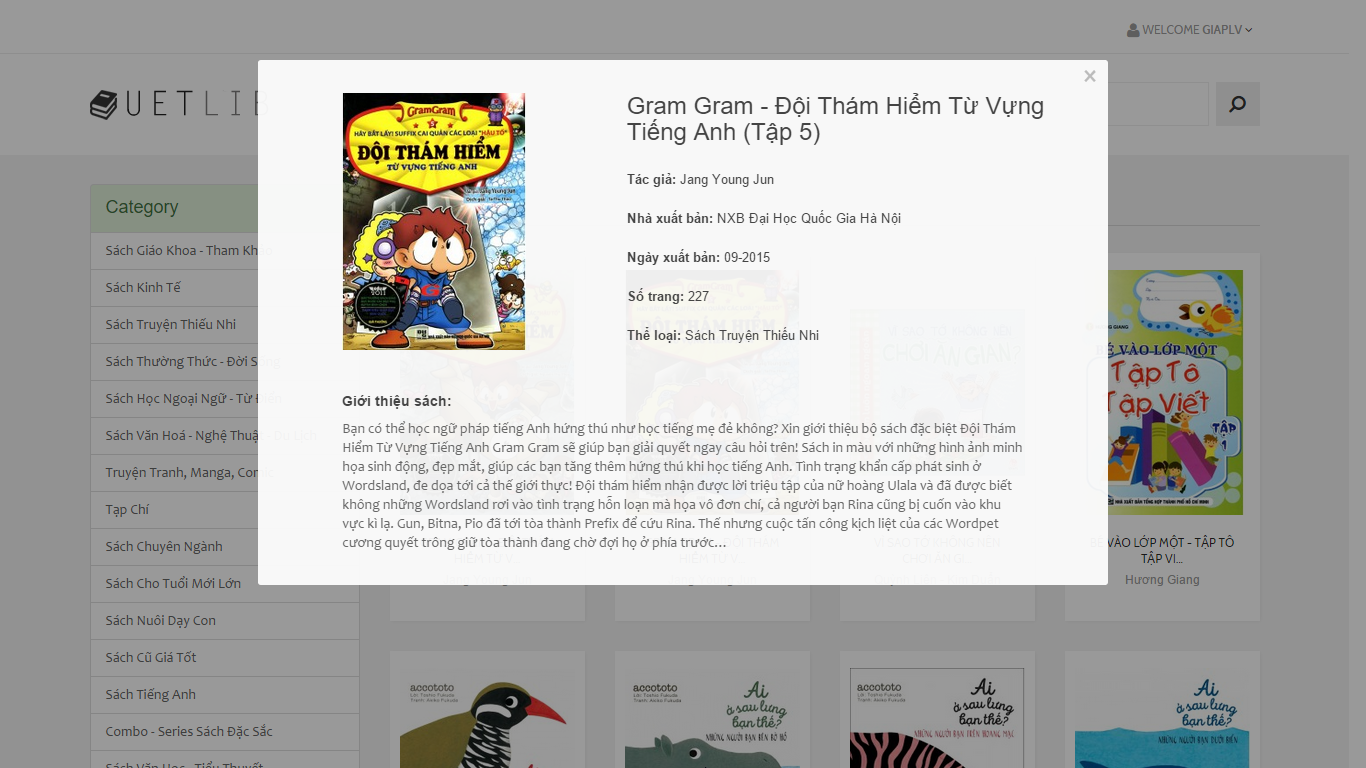
* Modern Pagination system



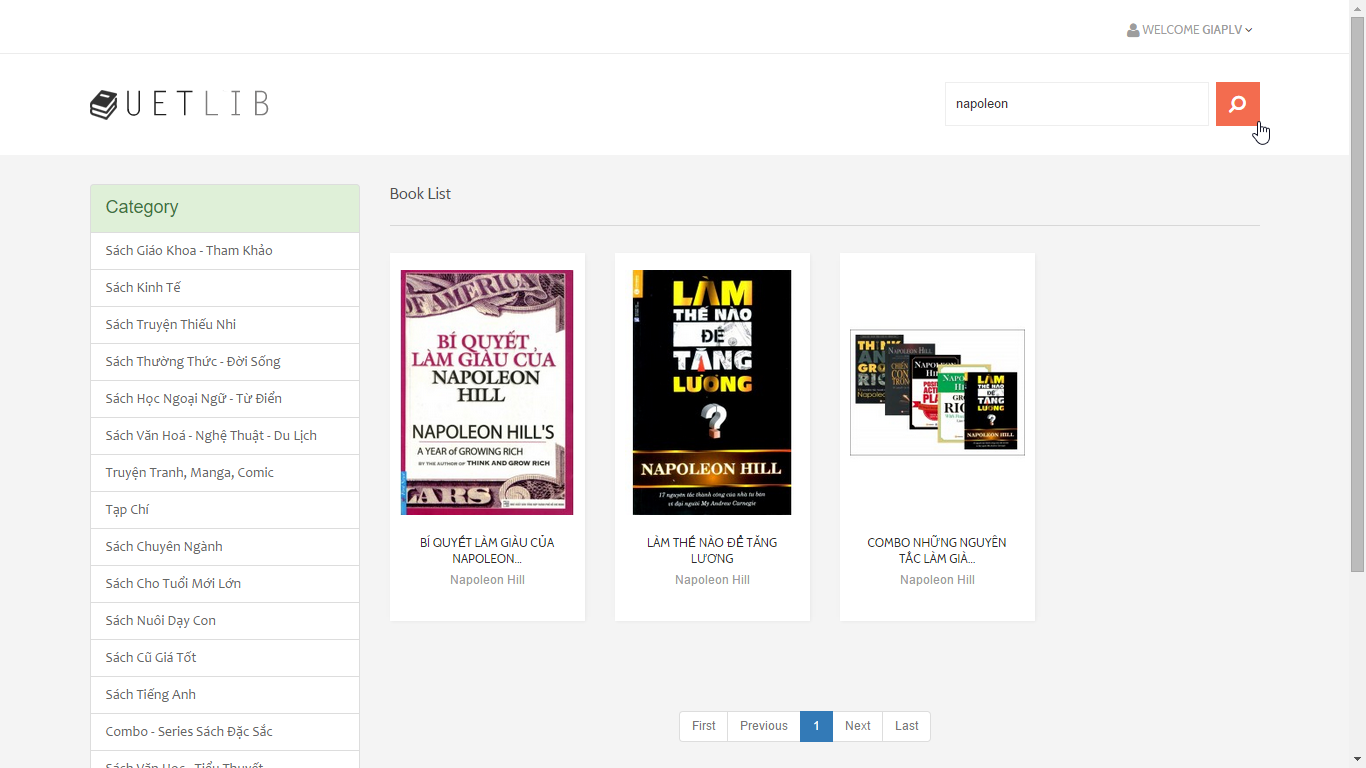
* View book by category



* View book detail



* Students can also search title and author of the books



## Desktop application

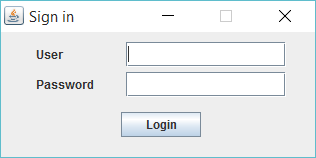
We built our desktop application, which allows the library staffs to manage the books, on Java platform. This app has these main functions: view all books of the library, edit the details of books, add a new book and delete books.

We use the standard HTTP requests to make communication between the app and the server.

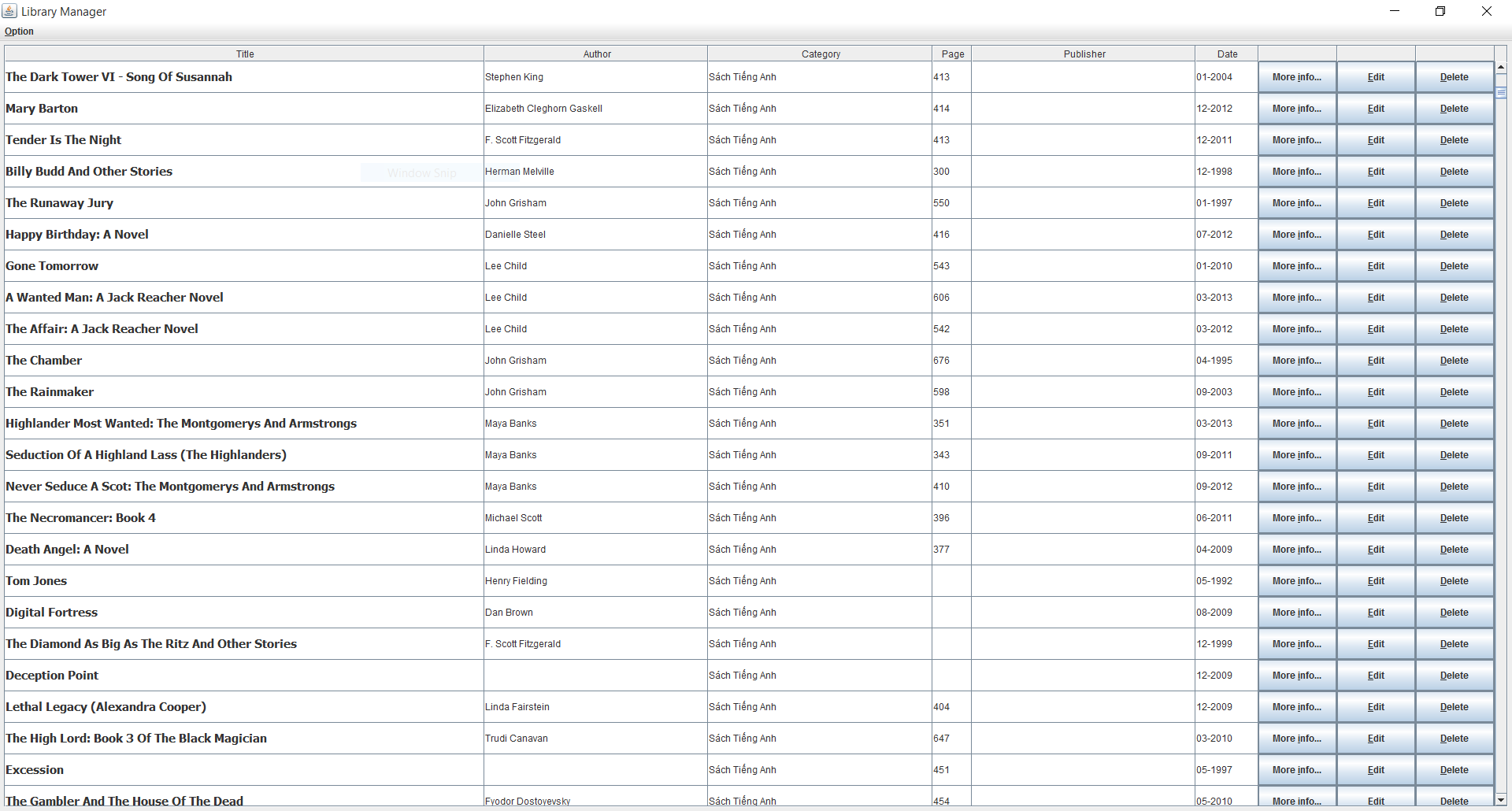
The user interface of the app is built by the library Java Swing.

Here are some screen shots of our desktop application:

* Log-in form



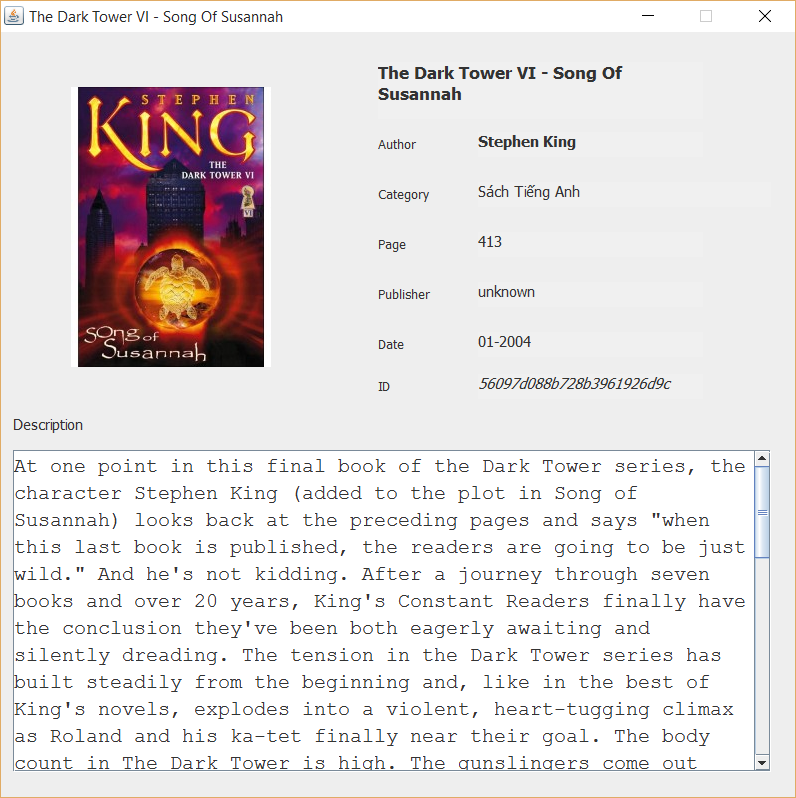
* After logging in, the staff will be able to see all the books of the library



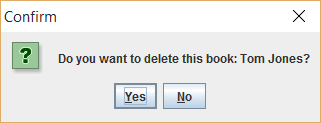
* And use the functions of the app



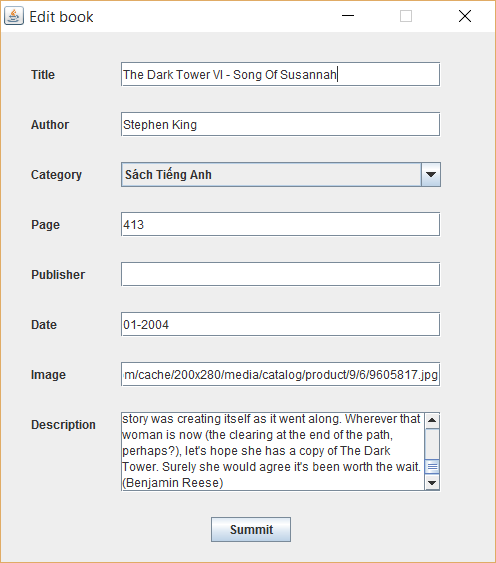
* Function – view all the details of a book



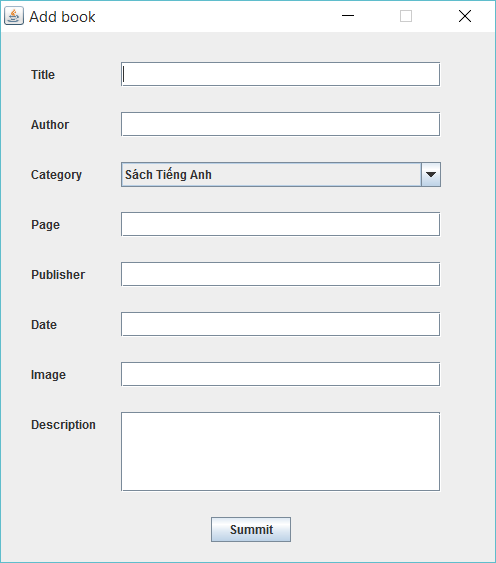
* Function – delete a book – the confirm dialog



* Function – edit a book



* Function – add a new book



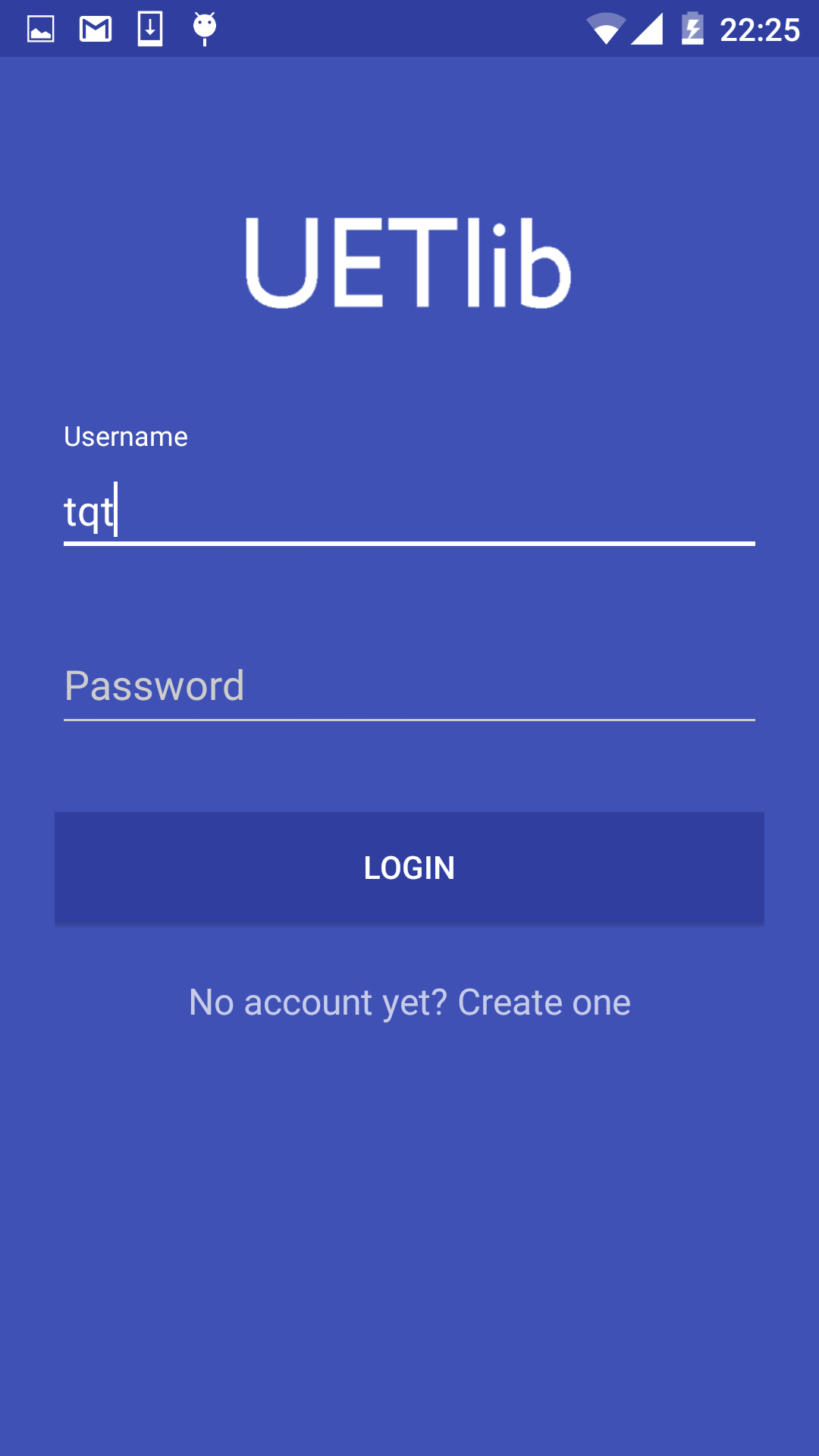
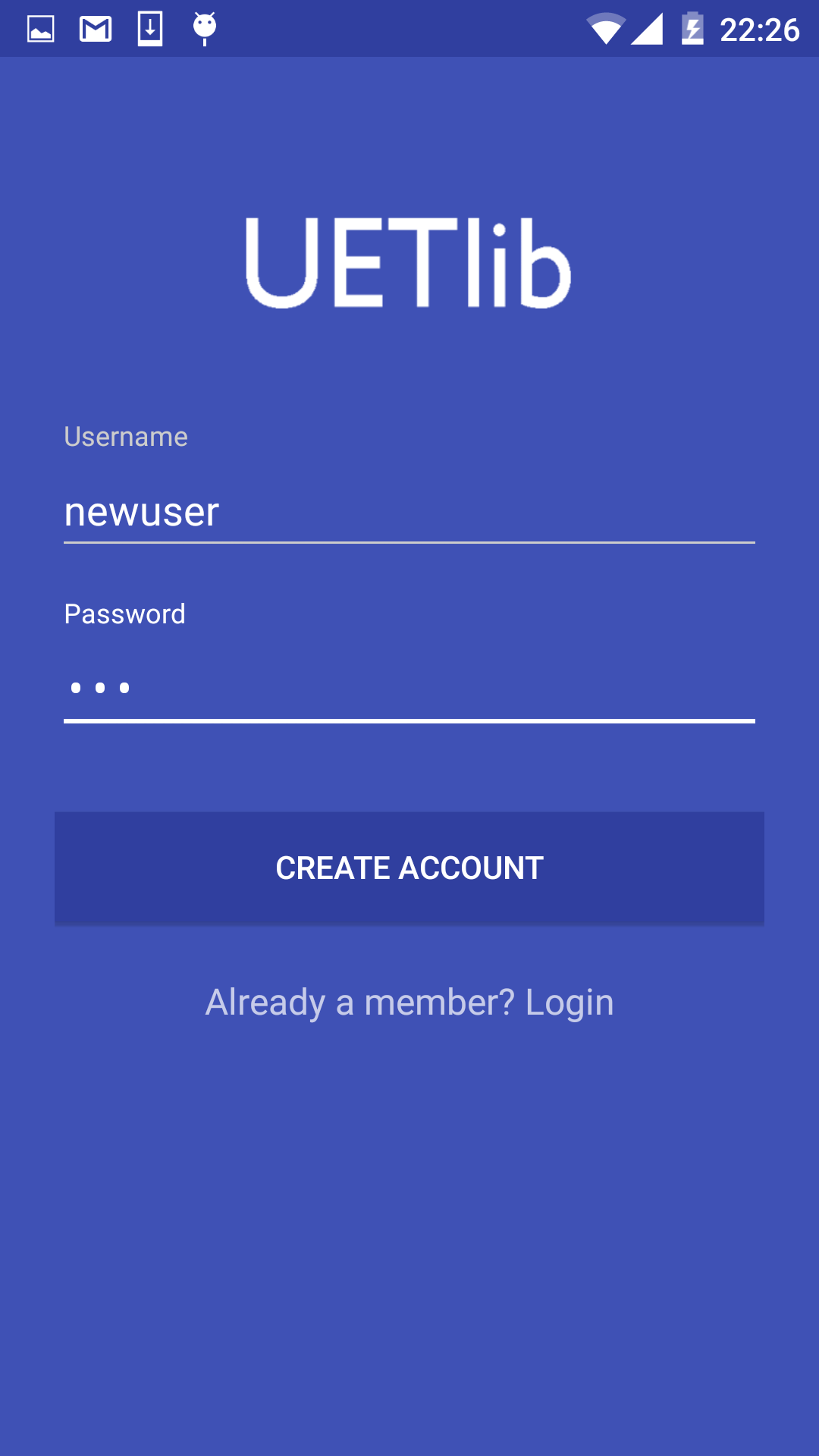
## Mobile application

We have website for students and desktop application for library staffs to access our system. Besides, we also build an Android mobile application for students to easily search and view books’ information on their smartphone.

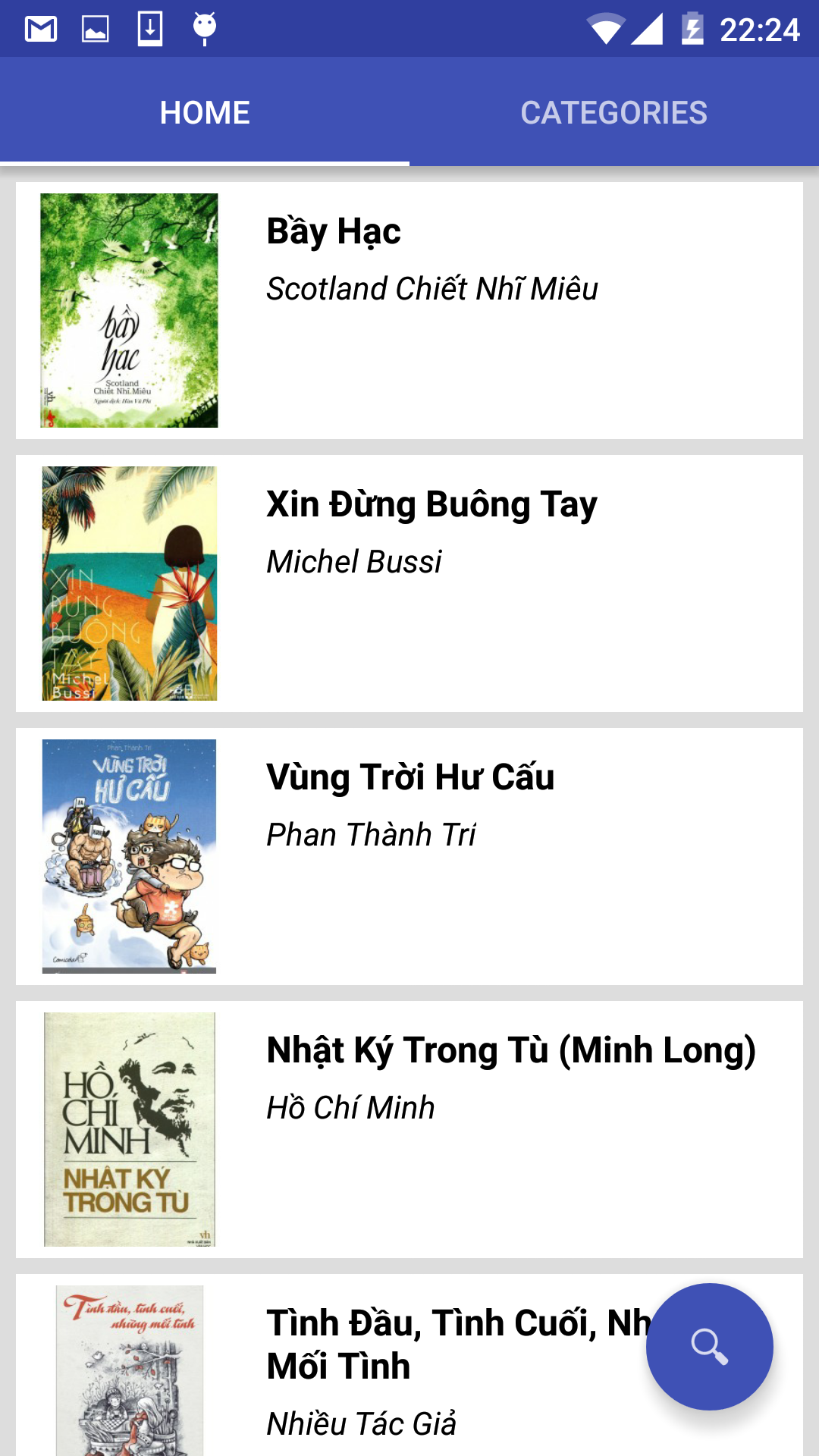
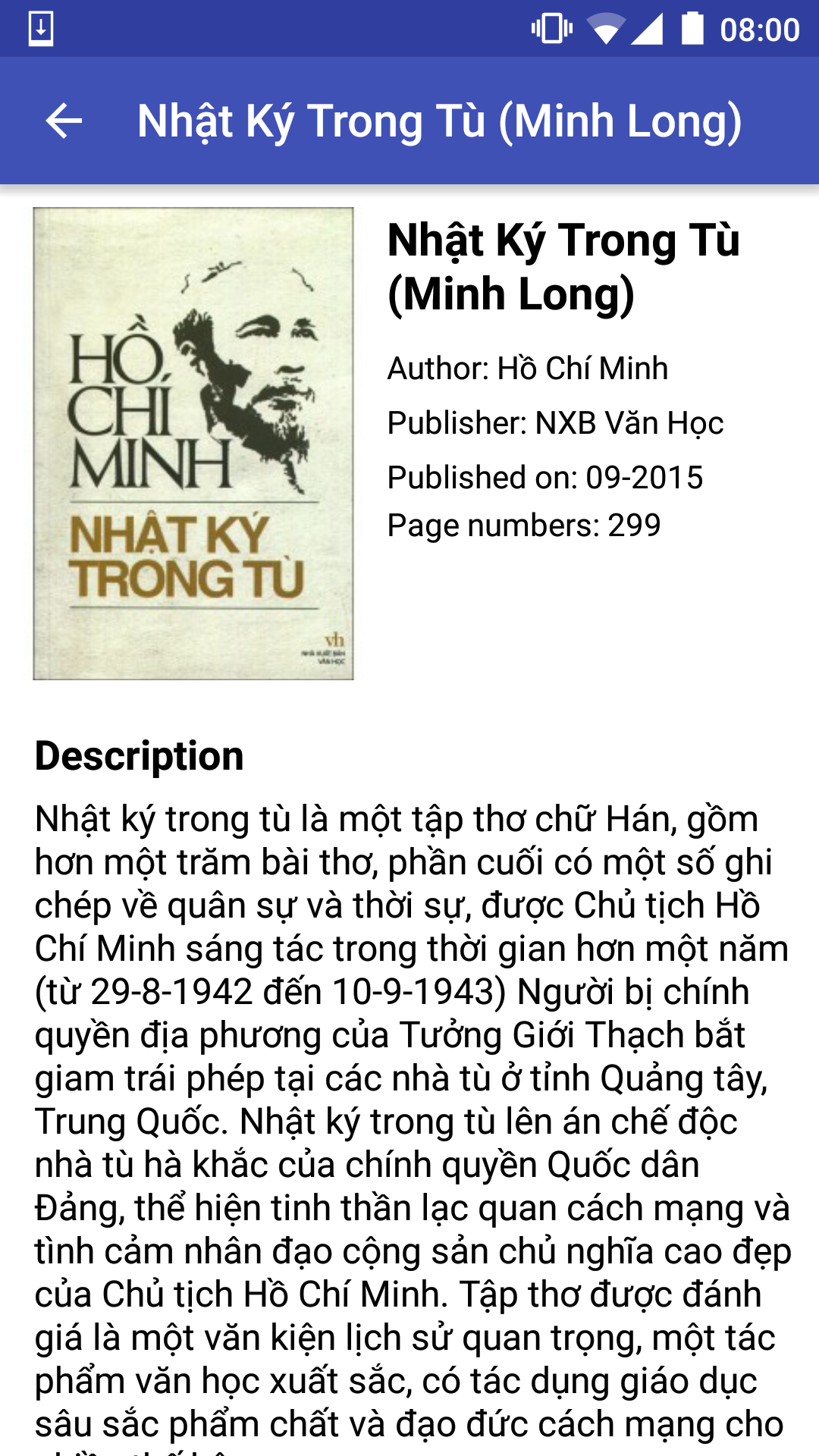
This application has all functions that we implemented on the website. We also use standard HTTP requests to communicate between mobile app and server like on the desktop application and the website.

There are some screenshots that we took one the real mobile running out application:

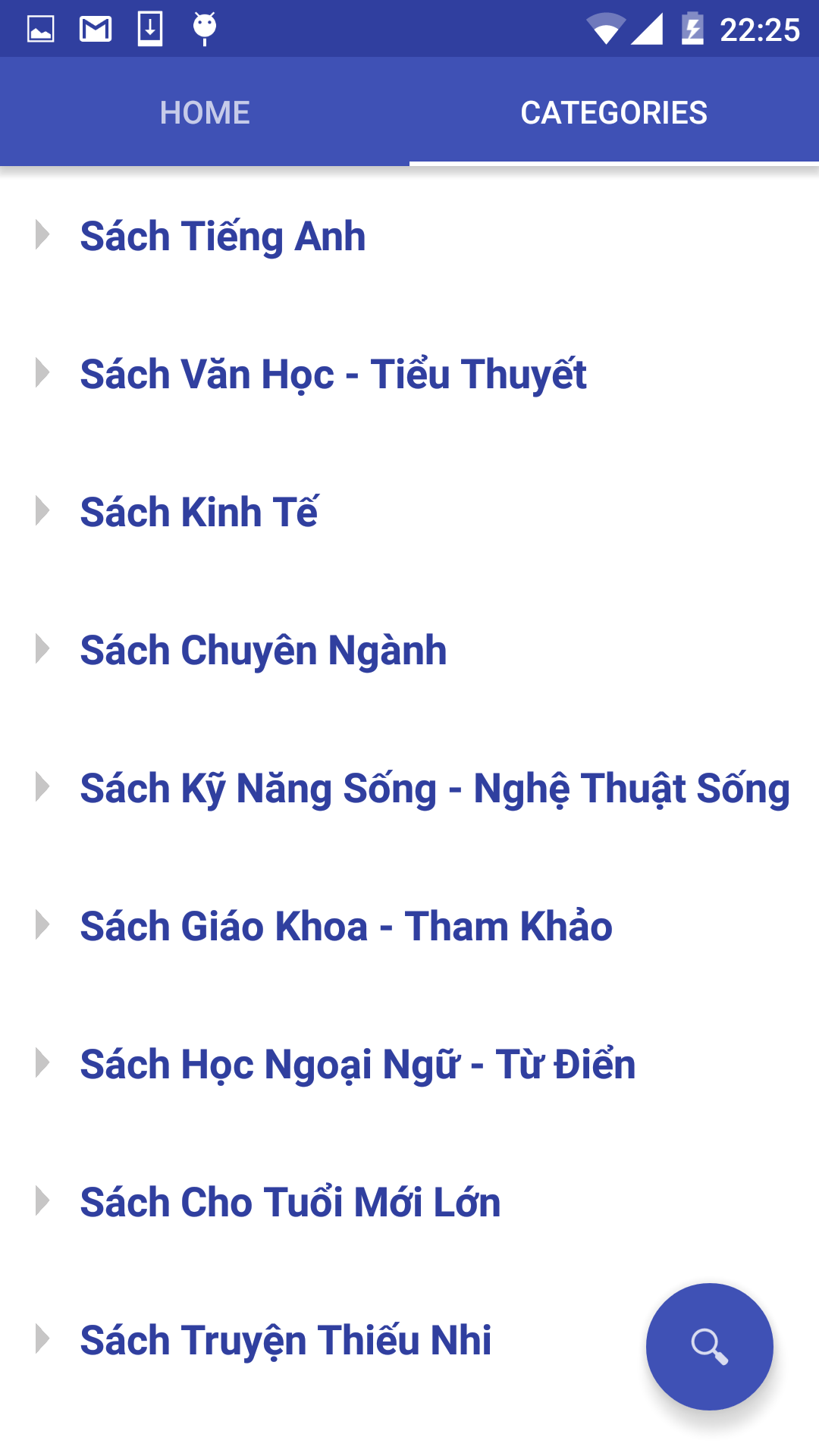
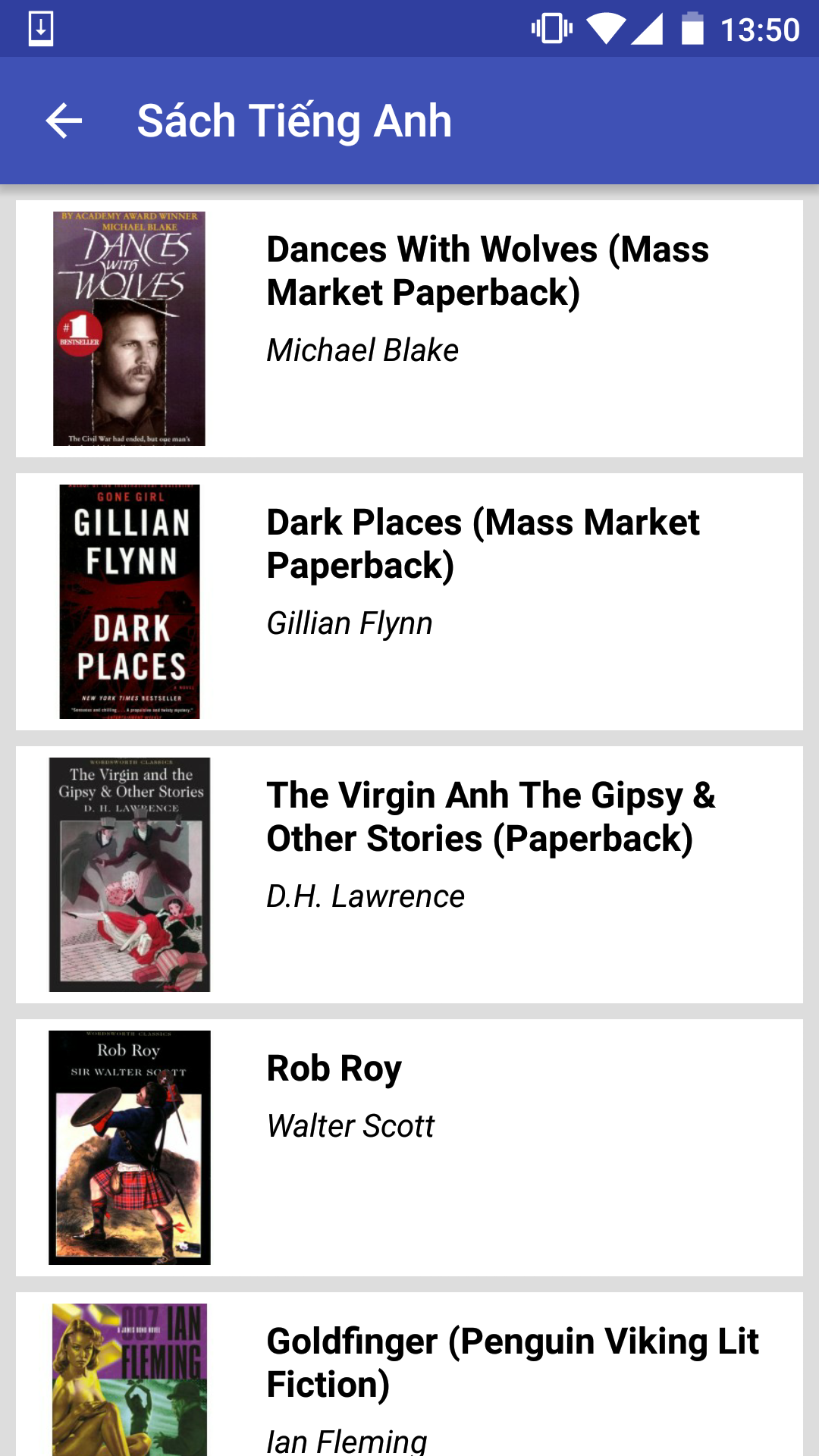
* Login and signup

* Home and book’s information

* We also can view books by category

* The most useful function is searching, students can search title and author of the books.