Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 1016­1998](http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf)1 for the full IEEE Recommended Practice for Software Design Descriptions.



1 <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee>­SDD.pdf

# (Team Name)

Bookworms Library

# Software Design Document

Names:

Soanca Daniela-Elena

Vasile Eduard-Alexandru

Ionele Adelina-Maria

Lab Section:

Workstation:

Date: 03/23/2021

**TABLE OF CONTENTS**

1. [**INTRODUCTION 2**](#_gjdgxs)
   1. [Purpose 2](#_30j0zll)
   2. [Scope 2](#_1fob9te)
   3. [Overview 2](#_3znysh7)
   4. [Reference Material 2](#_2et92p0)
   5. [Definitions and Acronyms 2](#_tyjcwt)
2. [**SYSTEM OVERVIEW 2**](#_3dy6vkm)
3. [**SYSTEM ARCHITECTURE 2**](#_1t3h5sf)
   1. [Architectural Design 2](#_4d34og8)
   2. [Decomposition Description 3](#_2s8eyo1)
   3. [Design Rationale 3](#_17dp8vu)
4. [**DATA DESIGN 3**](#_3rdcrjn)
   1. [Data Description 3](#_26in1rg)
   2. [Data Dictionary 3](#_lnxbz9)
5. [**COMPONENT DESIGN 3**](#_35nkun2)
6. [**HUMAN INTERFACE DESIGN 4**](#_1ksv4uv)
   1. [Overview of User Interface 4](#_44sinio)
   2. [Screen Images 4](#_2jxsxqh)
   3. [Screen Objects and Actions 4](#_z337ya)
7. [**REQUIREMENTS MATRIX 4**](#_3j2qqm3)
8. [**APPENDICES 4**](#_1y810tw)

### INTRODUCTION

## Purpose

This software design document describes the architecture and system design of a website library. The document will be used by different types of users, including software developers, software architects, business analysts, stakeholders.

## Scope

The purpose of the application is to manage the books and the borrowings that are

made by the users. We will create a website with a database background. Every user should create an account to be able to search for books in our online library.

## Overview

The first part (Introduction) of this document is a short description of our application, the purpose of SDD document, acronyms, definitions and some references.

The second part (System Overview) of this document is a short description of our system and the use case diagrams are presented.

The third part (System Architecture) of this document describes each level of high level architecture diagram.

The fourth part (Data design) of this document describes the relational model.

The fifth part (Human Interface Design) of this document are some mockups ,a short description for each page and functionalities of our application.

The sixth part (Requirements matrix) of this document Provide a cross reference that traces components and data structures to the requirements in your SRS document.

## Reference Material

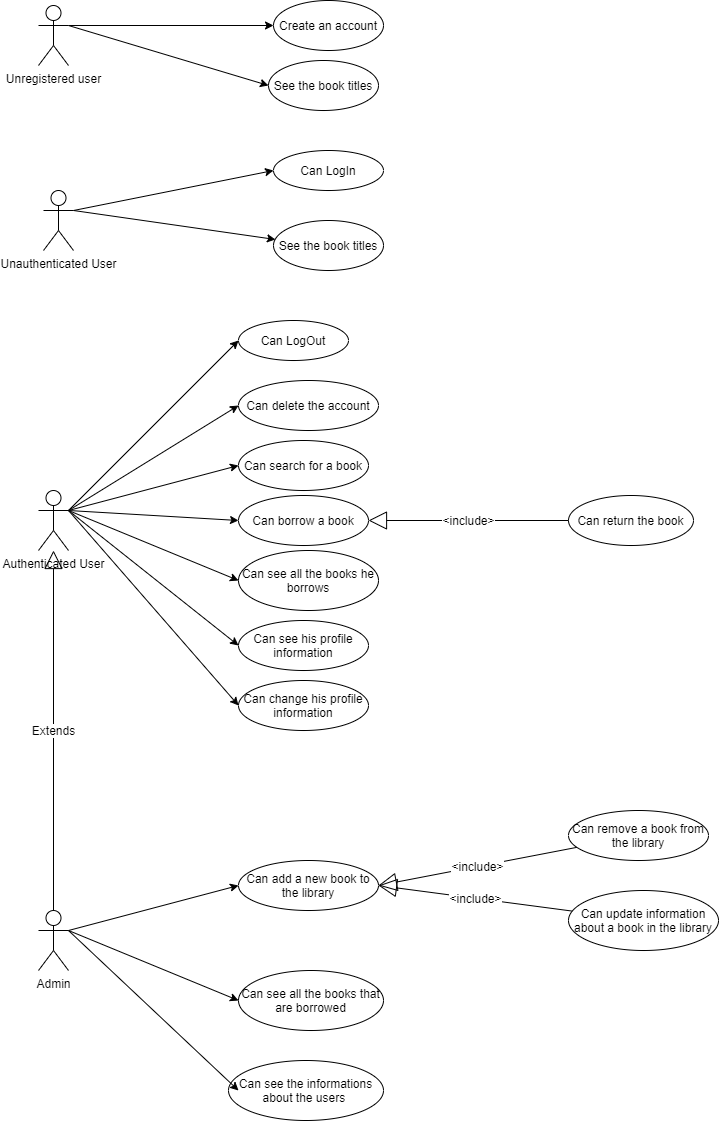
|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Link** |
| 1 | List of computing and IT abbreviations | <https://en.wikipedia.org/wiki/List_of_computing_and_IT_abbreviations> |
| 2 | C# Tutorials | <https://www.tutorialspoint.com/csharp/index.htm>  <https://www.tutorialsteacher.com/csharp/csharp-tutorials> |
| 3 | Get started with ASP.NET Core MVC | <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio> |
| 4 | Create an ASP.NET Core web app in C# - VisualStudio | <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-aspnet-core?view=vs-2019> |
| 5 | Web Tutorials | <https://www.w3schools.com/> |
| 6 | Html + CSS + PHP-MySQL Tutorials | <https://marplo.net/> |
| 7 | ASP.NET MVC Tutorials | <https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials> |

## Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation** | **Definition of abbreviation** |
| 1 | SQL | Structured Query Language |
| 2 | User | Any person who has interaction with the system who is not a developer |
| 3 | DB | Database |
| 4 | UI | User Interface |
| 5 | SRS | Software Requirements Specification |
| 6 | SDD | Software design document |
| 7 | System | A composite of equipment, skills and techniques capable of performing or supporting an operational role or both. A complete system includes all equipment, related facilities, material, software, services and personnel required for its operation. |
| 8 | Stakeholder | Any person who has interaction with the system who is not a developer |

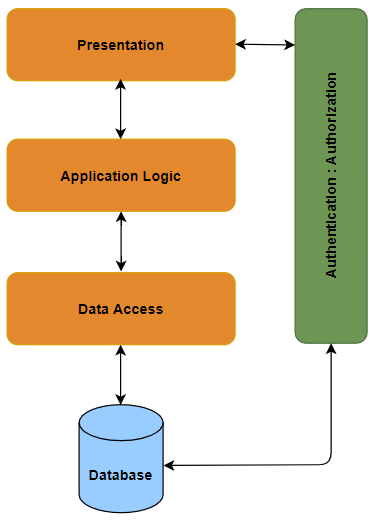
### SYSTEM OVERVIEW

This application is made for users that want to borrow a book, just with the help of the internet. After creating the online library application , readers will be able to connect with a username and a password. The users can create an account and borrow a book, can search for a book, see all the books they borrow and also can return the book.



### SYSTEM ARCHITECTURE

## Architectural Design



The most common architecture is the layered architecture pattern, otherwise known as the n-tier architecture pattern.

**Presentation layer** would be responsible for handling all user interface and browser communication logic, whereas a business layer would be responsible for executing specific business rules associated with the request. Also the presentation layer doesn’t need to know or worry about how to get customer data, as it only needs to display that information on a screen in particular format.

The **application layer** doesn’t need to be concerned about how to format customer data for display on a screen or even where the customer data is coming from, as it only needs to get the data from the data access layer, perform application logic against the data (e.g.: calculate values or aggregate data) and pass that information up to the presentation layer.

**Data access layer** in computer software is a layer of a computer program which provides simplified access to data stored in persistent storage of some kind, such as an entity-relational database.

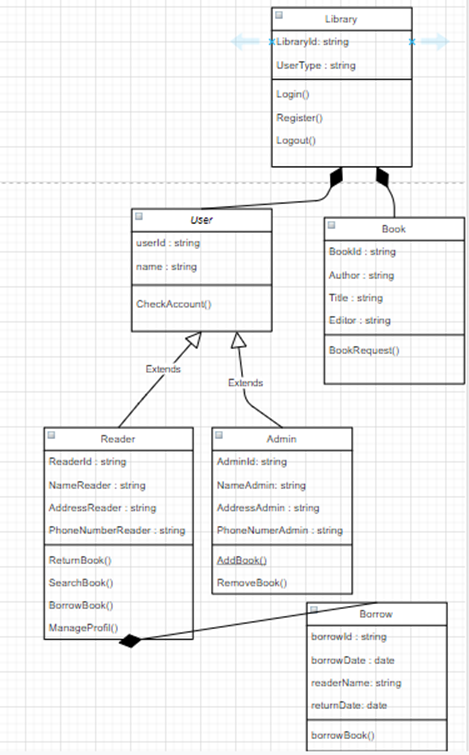
The **database layer** takes care of data access. An object from the database layer can write itself to one or more tables. In the database layer, you’ll find things like database, connection, table, SQL and result set.

**Authentication** is the process of verifying the identity of an individual. A unique identifier is associated with a user which is the username. A combination of username and password are used in order to authenticate a user. A user can interact with a web application using multiple actions. Access to certain actions or pages can be restricted. **Authorization** is the process of controlling user access through assigned roles and privileges.

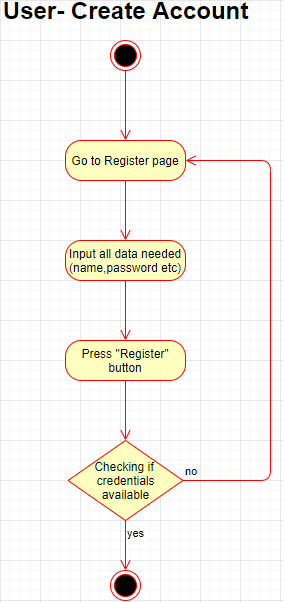
The layered architecture pattern is a solid general-purpose pattern, making it a good starting point for most applications, particularly when you are not sure what architecture pattern is best suited for your application.

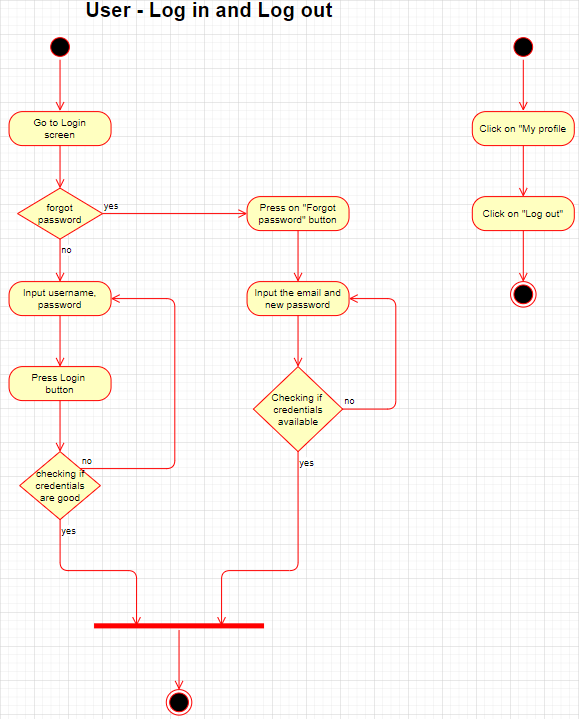
## Decomposition Description

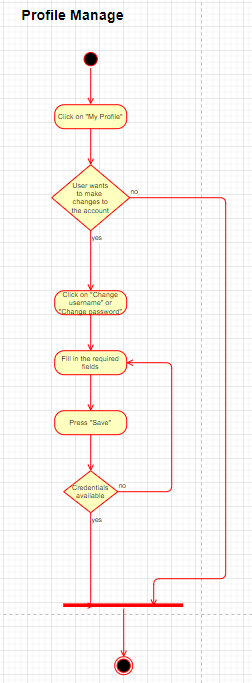
**CLASS DIAGRAM**

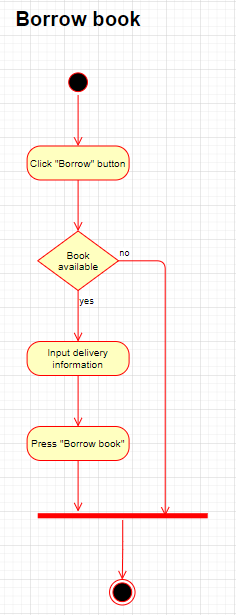


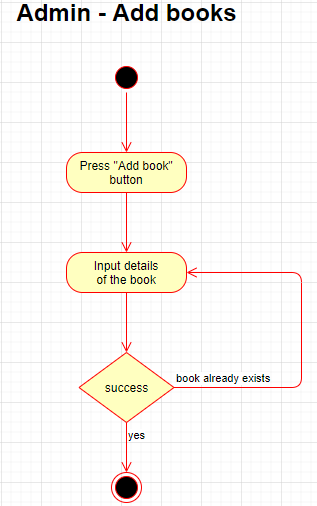
**ACTIVITY DIAGRAM**

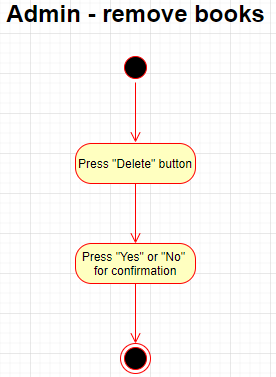


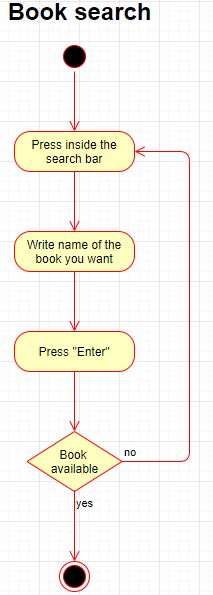


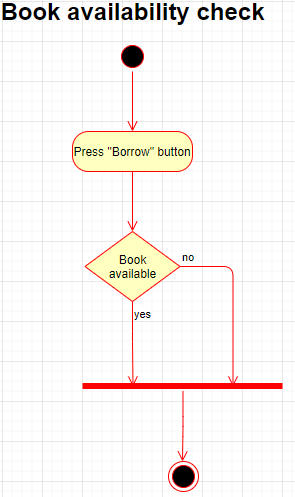










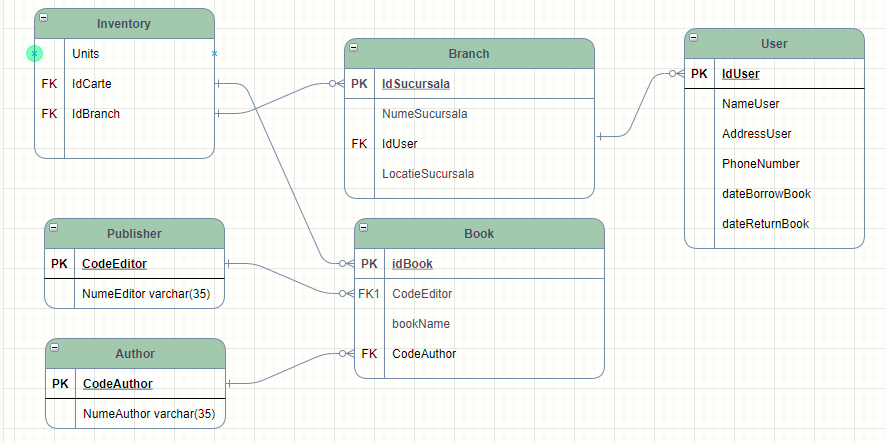


## Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

### DATA DESIGN

## Data Description



## Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

### COMPONENT DESIGN

Not applicable

### HUMAN INTERFACE DESIGN

## Overview of User Interface

Users can be either logged in or not. The logged in users can access the library, see the

description of the books and will be able to borrow a book, while the users that are not logged in

cannot see anything besides the book titles. Also, there are admins that are able to see all the books that are borrowed, to delete or add certain books, see the information about the users and

update descriptions about a book.

The functionalities:

* User can create an account
* User/Admin can log in
* User/Admin can log out
* User/Admin account can be deleted
* User/Admin can borrow a book
* User/Admin can see all the books he borrows
* User/Admin can return the book
* User/Admin can search for a book
* User/Admin can see his profile information
* User/Admin can change his profile information
* The admin can see all the books that are borrowed
* The admin can add a new book to the library
* The admin can remove a book from the library
* The admin can update information about a book in the library
* The admin can see the informations about the users

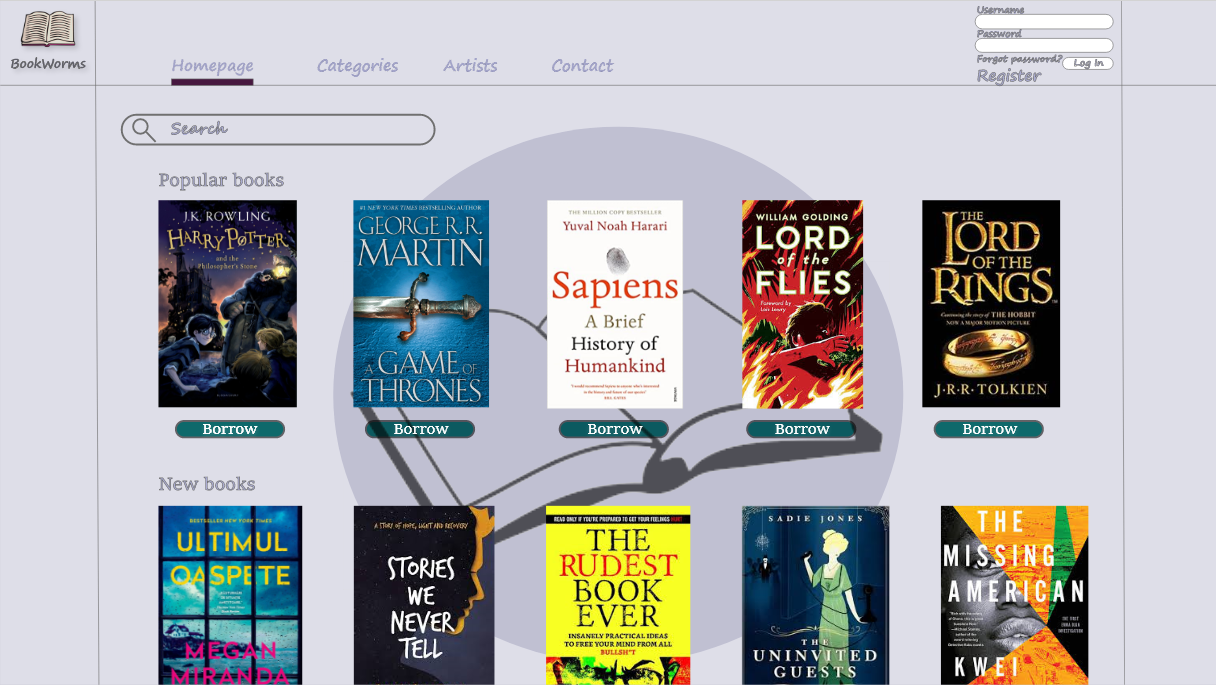
When opening the Bookworks website, there will be the home page. The user can log in if he’s already registered , otherwase he can register by introducing the first name, last name, username, password, email and phone number. If logged in as a user, instead of the login/register button should be the “My profile”. In the first page is also presented a lot of books and search bar.

Also if the user forgot his password, he can press ”Forgot password” and a new tab will appear and should write email, password and new password.

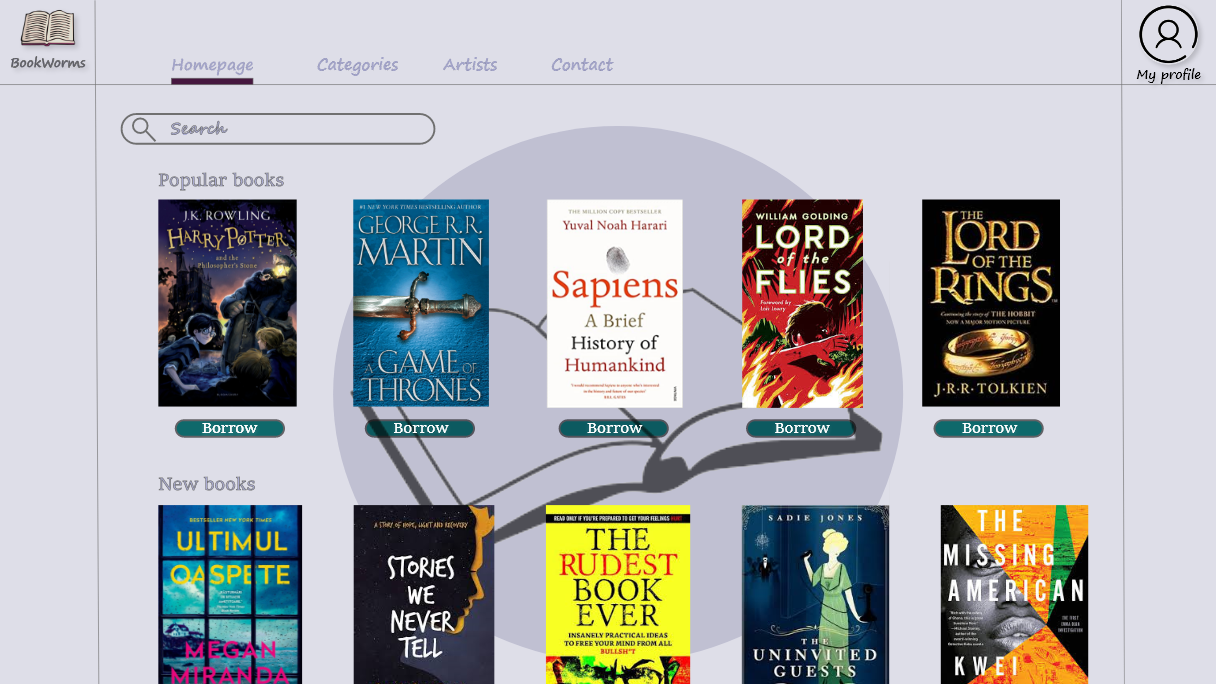
By clicking on a book from home page the user will be redirected to another page and here will be displayed some information about the book. If the user want to borrow a book , he can click on “Borrow” botton and will be asked to fill in some information.

## Screen Images

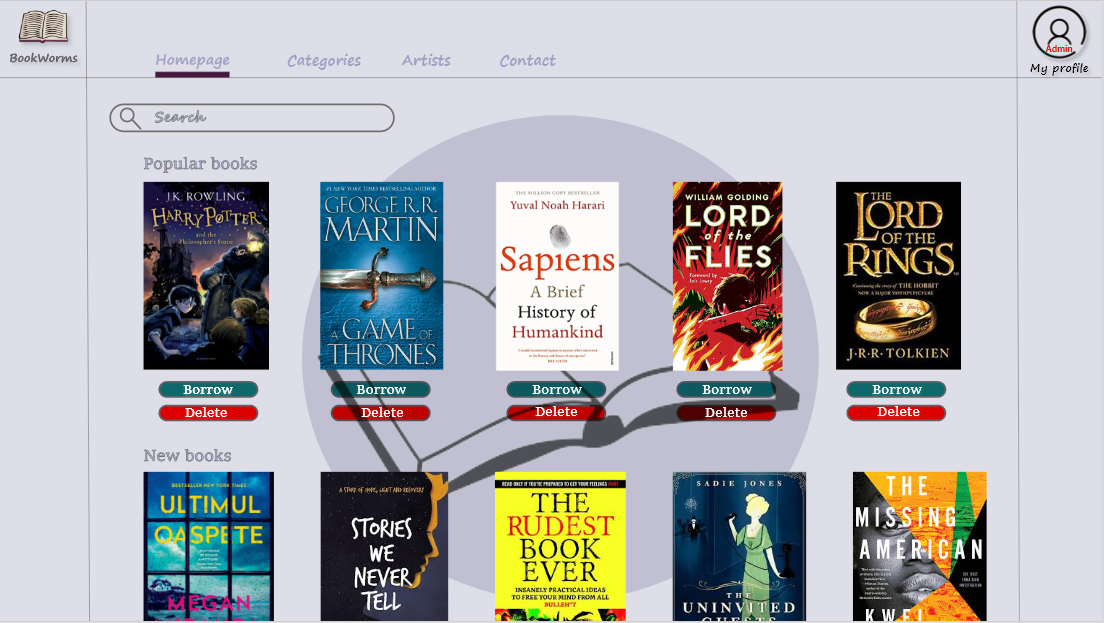
When opening the website, the homepage will be the first thing that appears. This page consists of some of the books that our website will provide, the log-in and register buttons, a search bar and some functionalities like “Homepage”, “Categories” etc.



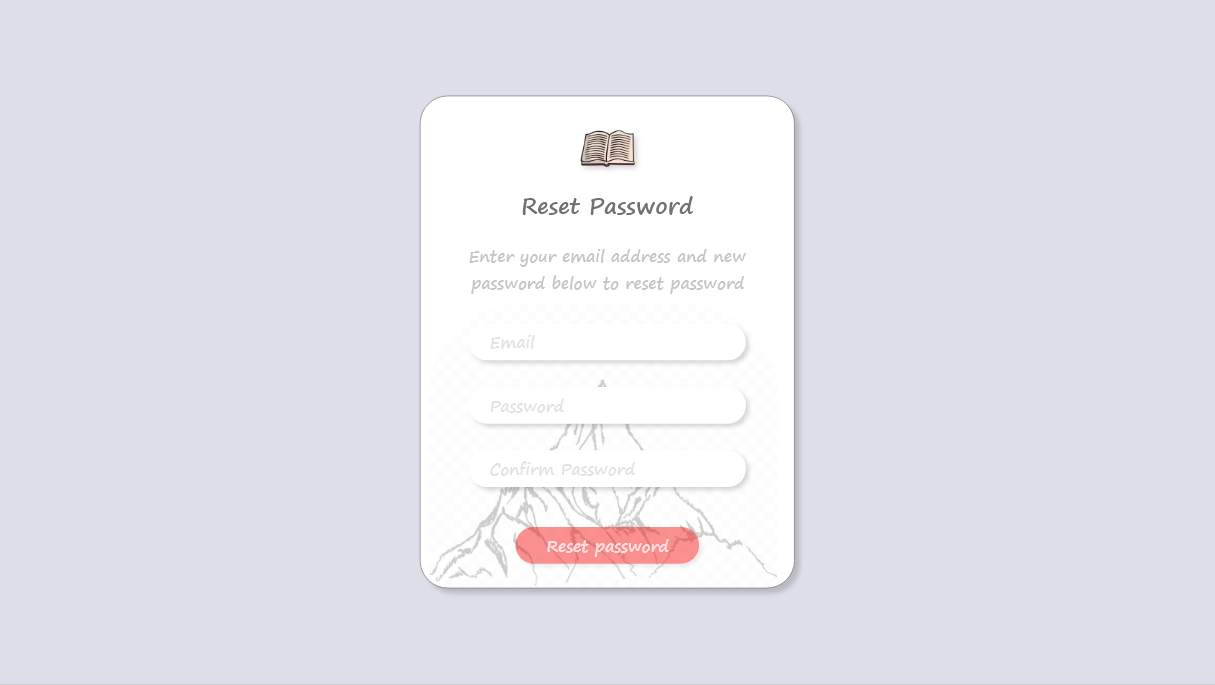
Once the user is logged in, the Log In interface won’t show up anymore. Instead, a My Profile button will appear.



The Homepage from an admin point of view (an admin can delete books).



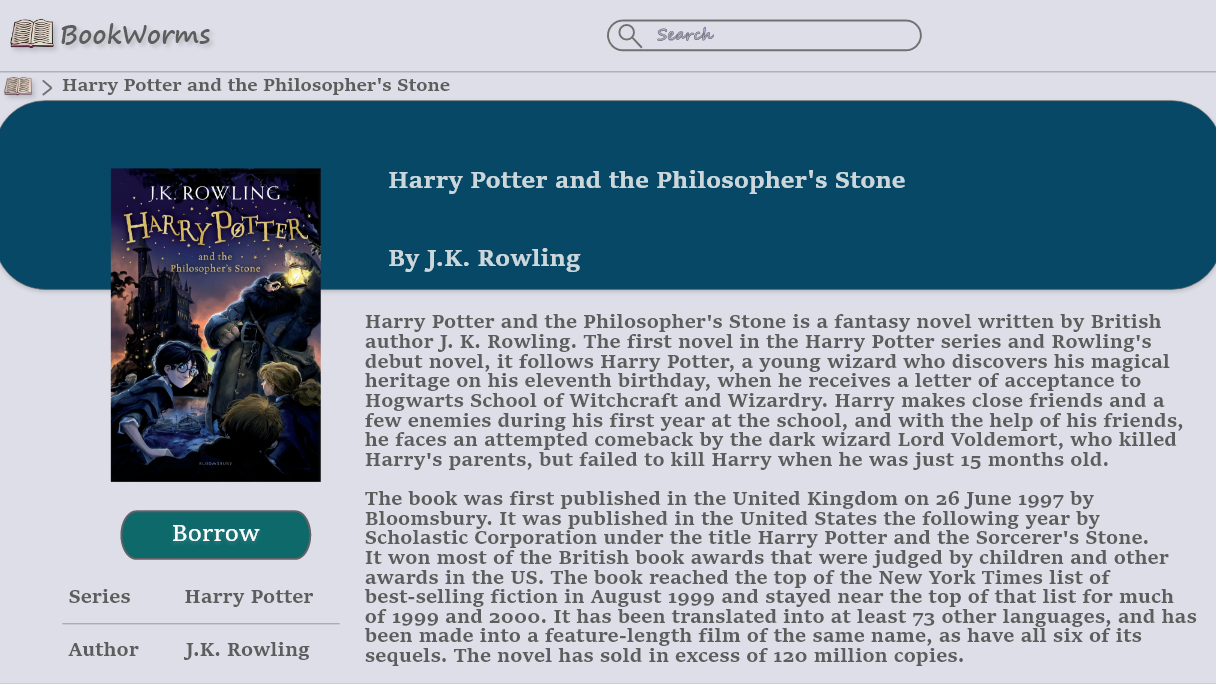
If “Forgot password?” is clicked, the following page will pop up.



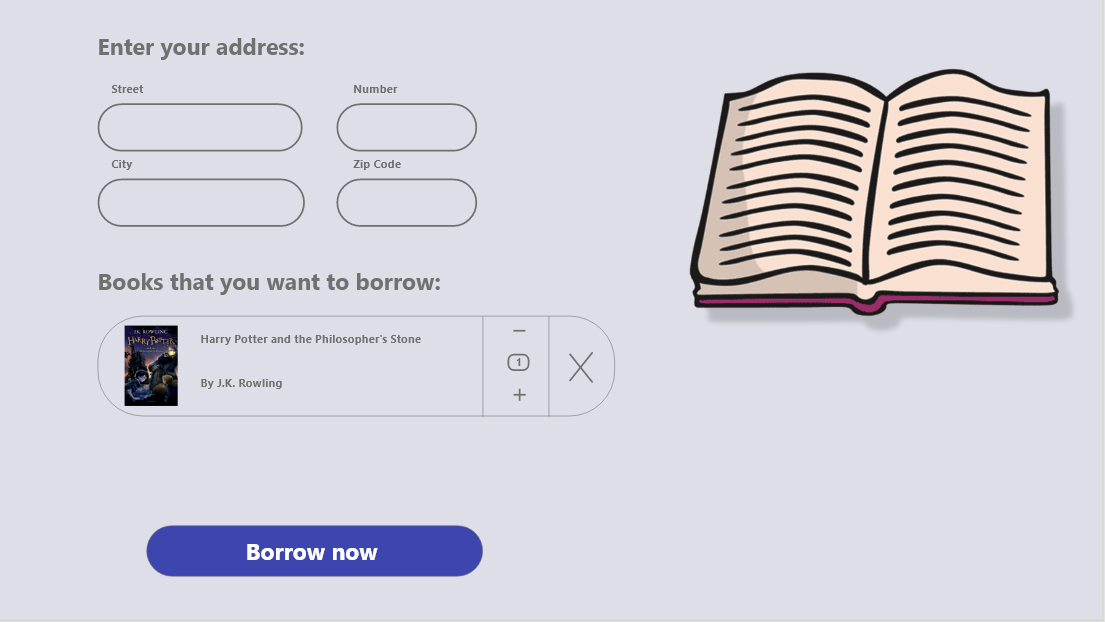
By pressing the “Register” button, the user will be redirected to the registration page.



By clicking on a book from the homepage, the user will be redirected to a different page that will consist of some brief description of that book.



Whenever an user wants to borrow a certain book, he will be redirected to the “Borrow page” and will be asked to fill in some information.



### REQUIREMENTS MATRIX

Provide a cross reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional

requirements from the SRS. Refer to the functional requirements by the numbers/codes that you

gave them in the SRS.

### APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.