Software Requirements Specification

for

Bookworms Library

**Version 2.1 approved**

**Prepared by Soanca Daniela-Elena**

**Ionele Adelina-Maria**

**Vasile Eduard-Alexandru**

**Faculty of Automatics, Computers and Electronics**

**05.03.2021**

**Table of Contents**

**Table of Contents [ii](#_19c6y18)**

**Revision History ii**

**1. Introduction [1](#_3tbugp1)**

1.1 Purpose [1](#_28h4qwu)

1.2 Document Conventions [1](#_nmf14n)

1.3 Intended Audience and Reading Suggestions [1](#_37m2jsg)

1.4 Product Scope [1](#_1mrcu09)

1.5 References [1](#_46r0co2)

**2. Overall Description [2](#_2lwamvv)**

2.1 Product Perspective [2](#_111kx3o)

2.2 Product Functions [2](#_3l18frh)

2.3 User Classes and Characteristics [2](#_206ipza)

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 4**

3.1 User Interfaces 4

3.2 Hardware Interfaces 8

3.3 Software Interfaces 8

3.4 Communications Interfaces 8

**4. System Features 8**

4.1 Create Account 8

4.2 Log in 9

4.3 Log out 10

4.4 Add book 10

4.5 Remove book 11

4.6 Check books availability 11

4.7 Search for a book 12

4.8 Borrow a book 12

4.9 Profile Manage 12

**5. Other Nonfunctional Requirements 13**

5.1 Performance Requirements 13

5.2 Safety Requirements 13

5.3 Security Requirements 13

5.4 Software Quality Attributes 13

5.5 Business Rules 14

**6. Other Requirements 14**

**Appendix A: Glossary 14**

**Appendix B: Analysis Models 14**

**Appendix C: To Be Determined List 14**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Soanca Daniela | 05.03.2021 | Adding Document Conventions (1.2), Adding Intended Audience and Reading Suggestions(1.3) | 1.1 |
| Vasile Eduard | 05.03.2021 | Adding Purpose (1.1), Adding Product Scope (1.4) | 1.2 |
| Soanca Daniela | 07.03.2021 | Updated Document Conventions(1.2), Adding System Features(4) | 1.3 |
| Soanca Daniela | 08.03.2021 | Adding References(1.5), Adding Product Perspective(2.1), Adding Product Functions(2.2), Adding User Classes and Characteristics(2.3), Updating System Features(4), Adding Operating Environment(2.4) | 1.4 |
| Vasile Eduard | 08.03.2021 | Updating Product Functions(2.2), Updating User Classes And Characteristics(2.3) | 1.5 |
| Ionele Adelina | 09.03.2021 | Updating References(1.5) | 1.6 |
| Vasile Eduard | 09.03.2021 | Updating Operating Environment(2.4), Adding User Interfaces(3.1), Adding User Documentation(2.6), Adding Performance Requirements(5.1), Adding Security Requirements(5.3) | 1.7 |
| Soanca Daniela | 09.03.2021 | Adding Design and Implementation Constraints(2.5), Adding Assumptions and Dependencies(2.7), Adding Software Quality Attributes(5.4) | 1.8 |
| Vasile Eduard | 09.03.2021 | Updating Software Quality Attributes(5.4), Updating Document Conventions (1.2) | 1.9 |
| Soanca Daniela | 10.03.2021 | Updating System Features(4), Adding Software Interfaces(3.3), Updating User Interfaces(3.1) | 2.0 |
| Ionele Adelina | 10.03.2021 | Updating Software Quality Attributes(5.4), Updating Document Conventions(1.2), Updating Design and Implementation Constraints(2.5), Updating User Documentation(2.6), Updating Operating Environment(2.4) | 2.1 |

# Introduction

## Purpose

The purpose of the document is to describe the functionality of a website library.The document will be used by different types of users, including software developers, software architects, business analysts, stakeholders.

## Document Conventions

Abbreviations:

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation** | **Definition of abbreviation** |
| 1 | SQL | Structured Query Language |
| 2 | DB | Database |
| 3 | UI | User Interface |
| 4 | SRS | Software Requirement Specification |
| 5 | OS | Operating System |
| 6 | MHz | Megahertz |
| 7 | SP2 | Service Pack 2 |
| 8 | PC | Personal Computer |
| 9 | IP | Internet Protocol |
| 10 | HTTP | Hypertext Transfer Protocol |
| 11 | HTTPS | Hypertext Transfer Protocol Secure |

Fonts:

* Body : Arial , size 11
* Title: Times, size 18
* Subtitle : Times, size 14

## Intended Audience and Reading Suggestions

The project is intended for : users (those who want to borrow books from the library), teachers , library managers, testers (can find errors). We suggest to read it if you have some knowledge in C# and DB.

## Product 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.

## References

|  |  |  |
| --- | --- | --- |
| No. | Name | Link |
| 1 | List of computing and IT abbreviations | [*https://en.wikipedia.org/wiki/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.tutorialspoint.com/csharp/index.htm) |
| 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*](https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio) |
| 4 | C# Tutorials | [*https://www.tutorialsteacher.com/csharp/csharp-tutorials*](https://www.tutorialsteacher.com/csharp/csharp-tutorials) |
| 5 | Create an ASP.NET Core web app in C# - VisualStudio | [*https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-aspnet-core?view=vs-2019*](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-aspnet-core?view=vs-2019) |
| 6 | Web Tutorials | *https://www.w3schools.com/* |
| 7 | Html + CSS + PHP-MySQL Tutorials | *https://marplo.net/* |
| 8 | ASP.NET MVC Tutorials | *https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials* |

# Overall Description

## Product Perspective

This project’s purpose is to facilitate the use of a software system for the automation of the management process of the activity of a library*.* A DB will be used to store the information about the users, books and the library itself.

The application will be used by the inhabitants of the library's city to borrow books much easier, browsing descriptions and deciding which book to choose.

We expect as many users as possible, but not more than the number of inhabitants of the city in which the library is located.

## Product Functions

* 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 change his username
* 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

## User Classes and Characteristics

There are two types of users: the readers (users of the website) and the admin of the website. 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.

## Operating Environment

In order to run the application, an OS and an internet connection are required. The website will run on the following operating systems :

1. Windows 7 or better:

* Chrome (at least version 48.0.2564)
* Firefox (at least version 62.0)
* Opera (at least version 15.0)
* Internet Explorer (at least version 11.0)
* Microsoft Edge

1. MacOS X 10.10 or better

* Chrome (at least version 48.0.2564)
* Firefox (at least version 62.0)
* Opera (at least version 15.0)
* Safari (at least version 8.0)

1. Linux - from 64-bit Ubuntu 14.04+, Debian 8+, Fedora Linux 24+

* Chrome
* Firefox
* Opera

1. For smartphones - Android 5.0 / iOS 8.0 or better (the website will run on desktop mode)

* Chrome
* Firefox
* Opera
* Safari (iOS)

## Design and Implementation Constraints

The employer must provide the following tools: MySQL Community Server 8.0.23, Visual Studio 2019 Community, Microsoft Studio Code, Microsoft Teams. We also need internet connection and ASP.Net CORE MVC 3.1.

We have chosen these work tools because these are the tools that we are most familiar with.

## User Documentation

Customers with account will receive a user manual for the application. On-line support (email, Google Classroom, Microsoft Teams) will be provided in case it’s necessary.

## Assumptions and Dependencies

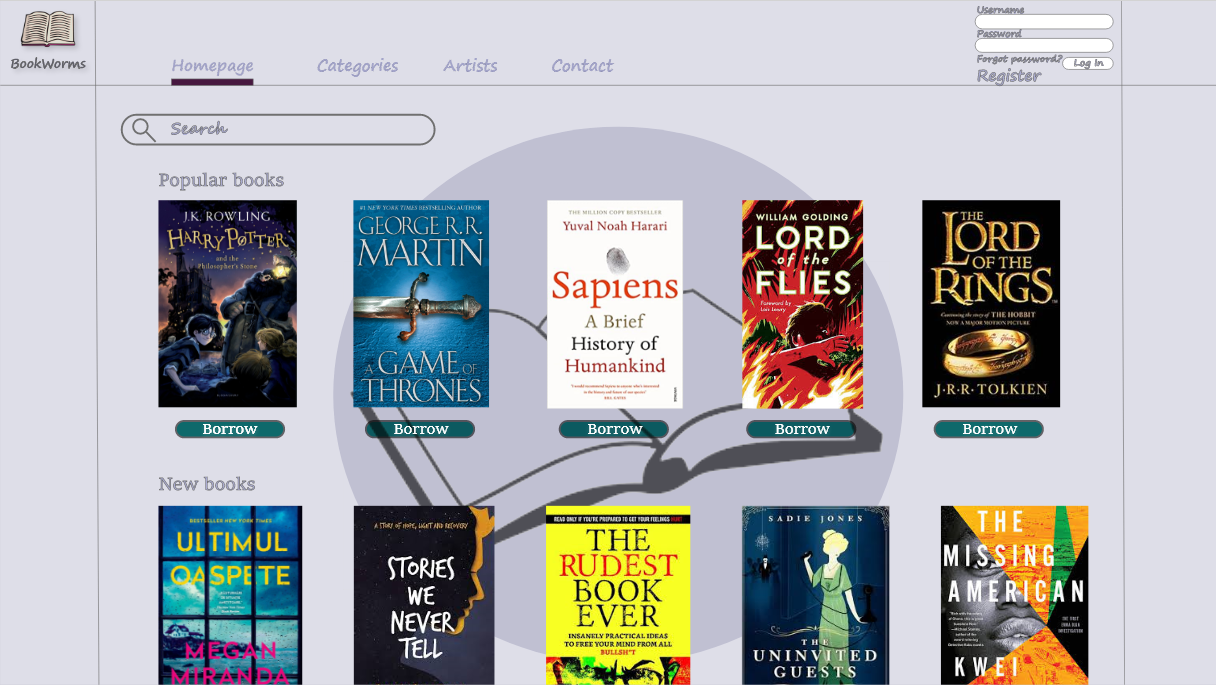
The requirements needed to run the application :

* An internet connection with a speed of minimum 1MB/s
* A device (phone, PC etc) with a browser like Chrome, Mozilla Firefox, Opera, Internet Explorer
* Peripherals (keyboard, mouse)

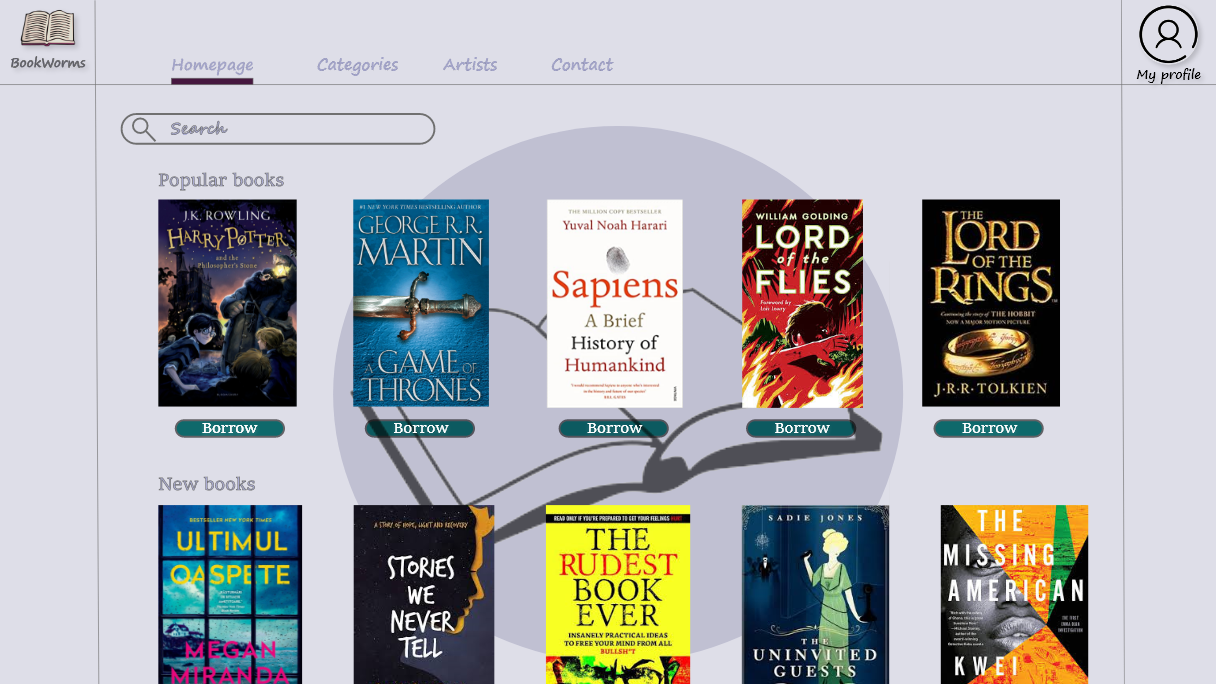
# External Interface Requirements

## User Interfaces

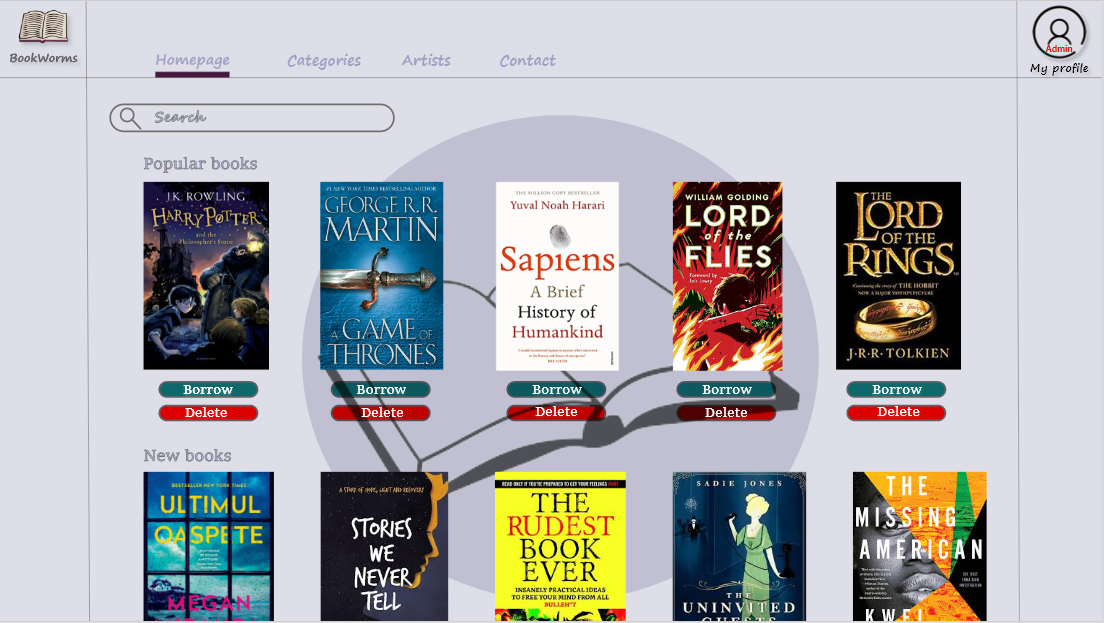
When opening the website, the homepage will be the first thing that appears. This page consists of some of the books that our website will provide, the log-in and register buttons, a search bar and some functionalities like “Homepage”, “Categories” etc.



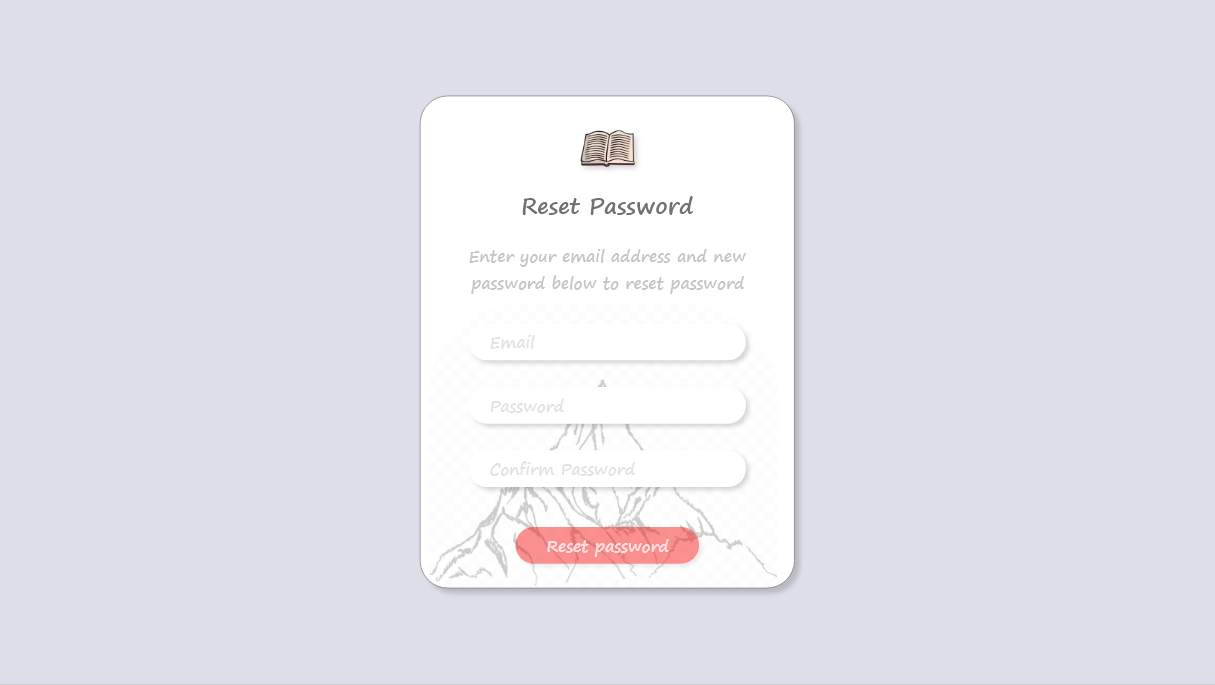
Once the user is logged in, the Log In interface won’t show up anymore. Instead, a My Profile button will appear.



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



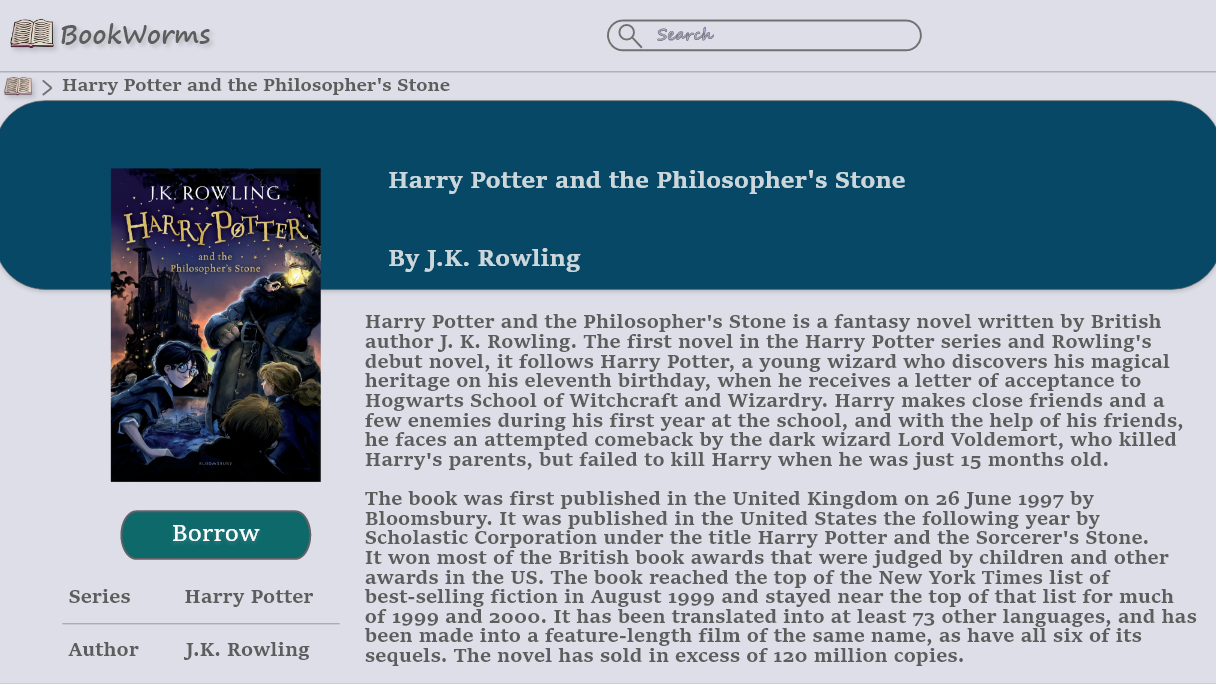
If “Forgot password?” is clicked, the following page will pop up.



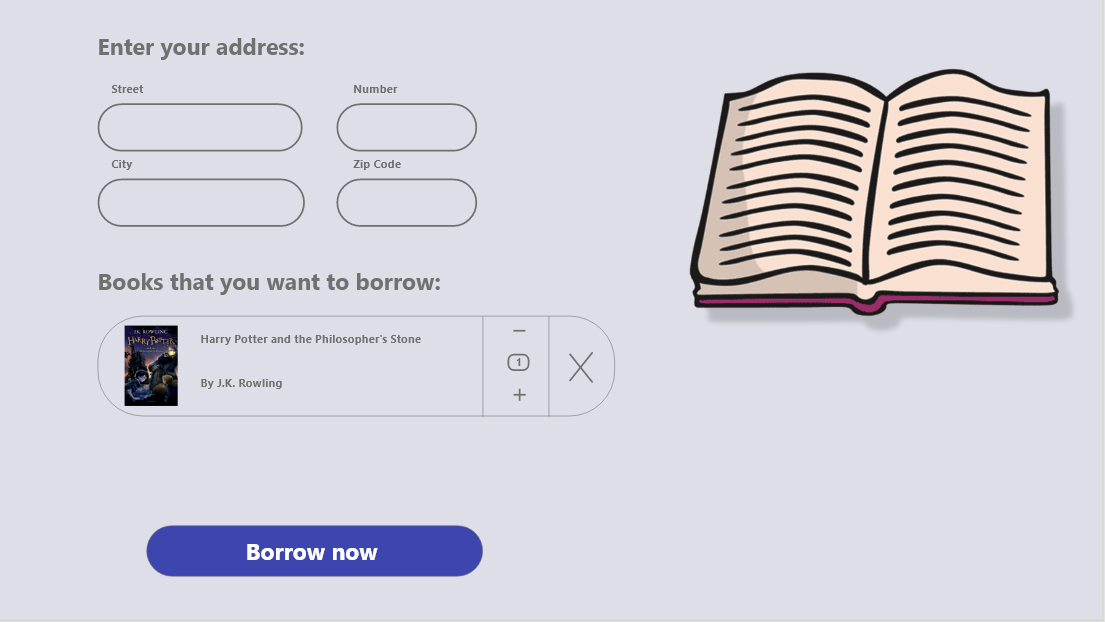
By pressing the “Register” button, the user will be redirected to the registration page.



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.



## Hardware Interfaces

Not applicable.

## Software Interfaces

Information is read from the DB.

The software application is directly linked to the DB in order to get information about the users, books and the library itself. The software application cannot make changes to the DB.

The web interface sends information directly to the DB, so it can make changes to it.

Some example of the functions that will be implemented are:

register (firstName, lastName, username, password, email, phone number)

login (username, password)

searchBook (bookTitle)

addBook (bookName, bookAuthor, releaseDate)

deleteBook ()

borrowBook (bookName)

## Communications Interfaces

Having that it’s a web application, the communication between the client and server is made through HTTP/HTTPS. The web address is set by the IP address and the default port (80 for HTTP, 443 for HTTPS).

# System Features

## Create User Account

**4.1.1 Description and Priority**

The user will be able to register by completing some text boxes with information.

High priority rated on a relative scale with a 9 (rated on a relative scale from a low of 1 to a high of 9).

**4.1.2 Stimulus/Response Sequences**

The input (to create an account):

- first name

- last name

- password

- username

- e-mail address

- phone number

The output will be :

- access permitted

- access rejected and a message like “account already exists” , “the username already exists, please choose another one”.

We do not need to check the age because the library is for everyone.

**4.1.3 Functional Requirements**

REQ-1: One textbox must be available in order to input first name

REQ-2: One textbox must be available in order to input the last name

REQ-3: One textbox must be available in order to input a password

REQ-4: One textbox must be available in order to input an email address

REQ-5: One textbox must be available in order to input an username

REQ-6: One textbox must be available in order to input a phone number

REQ-7: When the “Create account” button will be pressed, the information is read from the interface

REQ-8: It should be checked if the email address already exists in the DB

REQ-9: If it exists in the DB, an account cannot be created

REQ-10: If it doesn’t exist in the DB, then the email address will be stored in the DB and an account can be created

REQ-11: It should be checked if the username already exists in the DB

REQ-12: If it exists in the DB, an account cannot be created

REQ-13: If it doesn’t exist in the DB, then username will be stored in the DB and an account can be created

REQ-14: It should be checked if the phone number already exists in the DB

REQ-15: If it exists in the DB, an account cannot be created

REQ-16: If it doesn’t exist in the DB, then the phone number will be stored in the DB and an account can be created

## User Log in

* + 1. **Description and Priority**

The user will be able to log in after completing some text boxes with information.

High priority rated on a relative scale with a 9 (rated on a relative scale from a low of 1 to a high of 9).

**4.1.2 Stimulus/Response Sequences**

The input (to log in):

-username

-password

The output will be :

- access permitted

- access rejected

**4.1.3 Functional Requirements**

REQ-17: One textbox must be available in order to input the username

REQ-18: One textbox must be available in order to input the password

REQ-19: The username will be read from DB to check if it coincides with the input introduced

REQ-20: The password will be read from the DB to check if it coincides with the one that has been introduced

REQ-21: If they match then the access will be set to true

REQ-22: If they are different then the access will be set to false

REQ-23: It should be checked if the textboxes are filled

REQ-24: If they are, then access will be granted

REQ-25: If they aren’t, then the access will be set to false (access rejected)

REQ-26: If the password or username were forgotten, the user can click on “Forgot Password?” button, then the user will be redirected to another page

REQ-27: Two textboxes must be available in order to input the new password (“new password”and “confirm password”)

REQ-28: One textbox must be available in order to input the email address

REQ-29: It should be checked if the email address already exists in the DB

REQ-30: If it exists in the DB, then the new password will be stored in the DB and a confirmation mail should be received

REQ-31: If it doesn’t exist in the DB, an exception is thrown.

## User Log out

* + 1. **Description and Priority**

The users will be able to log out after pess “Log out” button from “My profile” section

Low priority.

**4.1.2 Stimulus/Response Sequences**

.The user can press the “Log out” button and then he will be redirected to Homepage.

**4.1.3 Functional Requirements**

REQ-32: After pressing the “Log out” button the user will be redirected to the Homepage

## Add books by Admin

* + 1. **Description and Priority**

The admins will be able to add books to the library.

High priority.

**4.1.2 Stimulus/Response Sequences**

.The input for add a book:

* name of the book
* author of the book
* release date

The output will be:

* added book
* invalid book (if the book already exists in the database)

**4.1.3 Functional Requirements**

REQ-33: A textbox must be available in order to input the name of the book

REQ-34: A textbox must be available in order to input the author of the book

REQ-35: A textbox must be available in order to input the release date of the book

REQ-36: If one of the textboxes is empty, an error will be shown

REQ-37: If the book already exists in the DB, an error will be displayed

REQ-38: If it doesn’t exist in the DB, then the new book will be added to the DB

## Remove books by Admin

* + 1. **Description and Priority**

The admins will be able to remove books from the library.

Medium priority.

**4.1.2 Stimulus/Response Sequences**

If the admins find a book that does not exist in the physical library , they will be able to remove that book from the website by pressing the “Delete” button.

**4.1.3 Functional Requirements**

REQ-39: The admin should be able to press the “Delete” button

REQ-40: An image will pop up, asking for confirmation

REQ-41: If “Yes” is pressed, the book will be deleted from the DB and the website

REQ-42: If “No” is pressed, the DB and website will remain unchanged and nothing is going to happen

## Books availability check

* + 1. **Description and Priority**

The users will be able to check if a book is available in the library.

Low priority.

**4.1.2 Stimulus/Response Sequences**

The user should be able to check if a book is available by pressing the “Borrow” button. Then a message will be displayed : if the book is available - “In stock” , otherwise - “Out of stock”.

**4.1.3 Functional Requirements**

REQ-43: If the “Borrow” button is pressed then the number of that specific book available in the DB should be compared to 0

REQ-44: If it is 0, the book is out of stock

REQ-45: If it is bigger than 0, then the book is in stock

## Book search

* + 1. **Description and Priority**

The users will be able to search for a book in the library.

Low priority.

**4.1.2 Stimulus/Response Sequences**

The user should be able to search a book by name in the Search bar. Once he hits enter, the search will begin.

**4.1.3 Functional Requirements**

REQ-46: The book is being searched by name which is introduced in the textbox

REQ-47: The enter button is pressed and the search begins

REQ-48: It’s checked to see if the book is available in the DB

REQ-49: If the book is available, it’s going to show up on the screen

REQ-50: If the book is not available, an error message will appear

## Book borrowing

* + 1. **Description and Priority**

The users will be able to borrow a book from the library.

High priority.

**4.1.2 Stimulus/Response Sequences**

The user should click on the “Borrow” button , if the book is available, then it can be borrowed

**4.1.3 Functional Requirements**

REQ-51: If the “Borrow” button is clicked and the book is in stock , the user will be able to complete some textboxes

REQ-52: Four textboxes will appear, asking for the user’s City, Street, Number, Zip Code, the date and hour on which the book shall be picked up from the library

REQ-53: After the information are completed, the user will be scheduled at the given time and date

REQ-54: The DB will be modified accordingly when the book will be picked up

## Profile Manage

* + 1. **Description and Priority**

The users can access the section “My profile” where it will display information about them and they will be able to change some of the details.

Medium priority.

**4.1.2 Stimulus/Response Sequences**

After logging in, the users can change some details about them ,like : their username, password.

**4.1.3 Functional Requirements**

REQ-55: A textbox for each field will show up to input the information that the user wants to change

REQ-56: If some changes are made, the button “Save” must be pressed

REQ-57: After the changes, the new fields will be updated in the DB

# Other Nonfunctional Requirements

## Performance Requirements

The application should be able to load in two seconds.

Minimum system requirements in order to access the website through a browser (Google Chrome, Mozilla Firefox, Opera, Internet Explorer) :

* Intel Pentium 4 / AMD Athlon 64 processor or later that’s SSE2 capable
* 100 MB of hard drive space (Mozilla Firefox needs 200 MB)
* 128 MB of RAM (Mozilla Firefox needs 512 MB)
* at least Android 5.0 Lollipop or iOS 8.0 for mobile devices

## Safety Requirements

Not applicable.

## Security Requirements

User registration is used in order to help better security and privacy. Also, user authentication is available, meaning that every user has his own username and password. If the “user” and “password” parameters which the user entered are the same with the ones saved in our database, access is granted.

## Software Quality Attributes

Quality code rules:

* Classes names are written by using Upper Camel Case
* Method names and variable names are written using lower Camel Case
* Method name are verbs
* A method should have a single purpose
* A method should use all provided parameters
* Variables are nouns
* Braces are used with if, else, for, do and while statements even when the body contains only one statement
* Place comment on a separate line, not at the end of the line
* Begin comment text with an uppercase letter
* Write only one statement per line
* Write only one declaration per line
* Use meaningful names for query variables
* A method should either change the state of the object or return a value, not both
* Add a blank line between if, else, for, do and while statements and their body code
* Add the first brace on the same line as the if, else, for, do and while statement is

Functional quality rules:

* The project shouldn’t have warnings. Treat warnings as errors
* Maintainability - the code should be written in such a way that it favors implementation of new functions

## Business Rules

Not applicable.

# Other Requirements

Not applicable

**Appendix A: Glossary**

Not applicable

**Appendix B: Analysis Models**

Not Applicable

**Appendix C: To Be Determined List**

Not applicable