Software Requirements Specification

for

Automated digital and distributed system for managing the activity of a library

Version 1.2 approved

Prepared by Barbu Andy Valentin,Lupu Ionut, Barbalata George Mihai

04.03.2021

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 3

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

2. Overall Description 3

2.1 Product Perspective 3

2.2 Product Functions 4

2.3 User Classes and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 8

3.3 Software Interfaces 8

3.4 Communications Interfaces 8

4. System Features 8

4.1 System Feature 1 8

4.2 System Feature 2 10

4.3 System Feature 3 10

4.4 System Feature 4 11

4.5 System Feature 5 11

4.6 System Feature 6 11

4.7 System Feature 7 12

4.8 System Feature 8 12

4.9 System Feature 9 12

4.10 System Feature 10 13

4.11 System Feature 11 13

4.12 System Feature 12 14

4.13 System Feature 13 14

5. Other Nonfunctional Requirements 14

5.1 Performance Requirements 14

5.2 Safety Requirements 15

5.3 Security Requirements 15

5.4 Software Quality Attributes 15

5.5 Business Rules 15

6. Other Requirements 16

Appendix A: Glossary 16

Appendix B: Analysis Models 17

Appendix C: To Be Determined List 17

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Barbu Andy | 04.03 | Document created, main part done | 1 |
| Lupu Ionut | 06.03 | Mockups added, REQs added | 1.1 |
| Barbalata george | 10.03 | Document verified, small errors corrected | 1.2 |

# Introduction

## Purpose

The purpose of this project is to help the day to day activities of a library by giving keeping track of the book stock, keeping track of the books that have been borrowed, and by allowing readers to see what books are currently available at what locations and reserving them for later borrowing.

## Document Conventions

This document has the titles of the chapters written with Times font, size 18, bold.  
The subtitles are written with Times font, size 14, bold.

The chapters are written with Arial font, size 11.

## Intended Audience and Reading Suggestions

This document is intended for developers, testers and system administrators that are looking to better understand the inner working of the product. The product itself should have e simple enough appearance and it`s considered to be highly intuitive such that no proper training or reading should be done to be able to fully utilize the product`s functions.

## Product Scope

The main scope of this product is to minimize the manual work done by the library workers by automating the inventory of the library and automating the borrowing process. The readers are also able to search for the desired books and reserving said books without having to physically go to the local library, which should be more appealing to the young readers.

## References

<https://openlibrary.org/> --site example

<https://www.youtube.com/watch?v=w25ea_I89iM&ab_channel=TraversyMedia> – good tutorial for ASP.NET  
[https://dotnet.microsoft.com/apps/aspnet -- ASP.NET](https://dotnet.microsoft.com/apps/aspnet%20--%20ASP.NET) website, all the documents needed

# Overall Description

## Product Perspective

This product is not unique, as others libraries are also using similar products to better manage the book stock and to provide the readers an interface with the library.  
However, this specific product is better suited for the specific needs of the library that’s being done for.

## Product Functions

This product has the following functions:

* A relational database to better keep track of books, readers, borrowings and unwanted clients
* An easy to use query system for the librarians for the usual tasks, such as checking stocks, checking subscribers and so on
* An easy to use query system for the readers, to check the availability of books at certain locations
* An advanced query system for librarians and system administrators, for all and every query needs
* A login based web application, for both librarians and readers, with different functionalities and ways of accessing at the database

## User Classes and Characteristics

This product will be used mostly by librarians and readers, both having different clearance levels.  
The librarians will be able to use the database at its full query power, also having access at reader`s personal information.  
The readers should have access only at books information.

A third category that may use this products are the system administrators. This category is able to also modify the database however they see fit: renaming tables, adding and removing fields, adding and removing books and users etc… These functions should be used carefully, as the system administrators could break the system.

## Operating Environment

As the product is a web application, to be able to use the product, a web browser is necessary with a stable internet connection. The operating system required by the product is specified by the web browser.

To make sure that the application runs as expected, Google Chrome is recommended.

## Design and Implementation Constraints

For this specific product, no constraints have been put in place, other than the deadline. The developers are allowed to use all and every technology that they consider fit for the developing process.

## User Documentation

No user documentation will be delivered as the product is considered to be easy enough to use and highly intuitive.

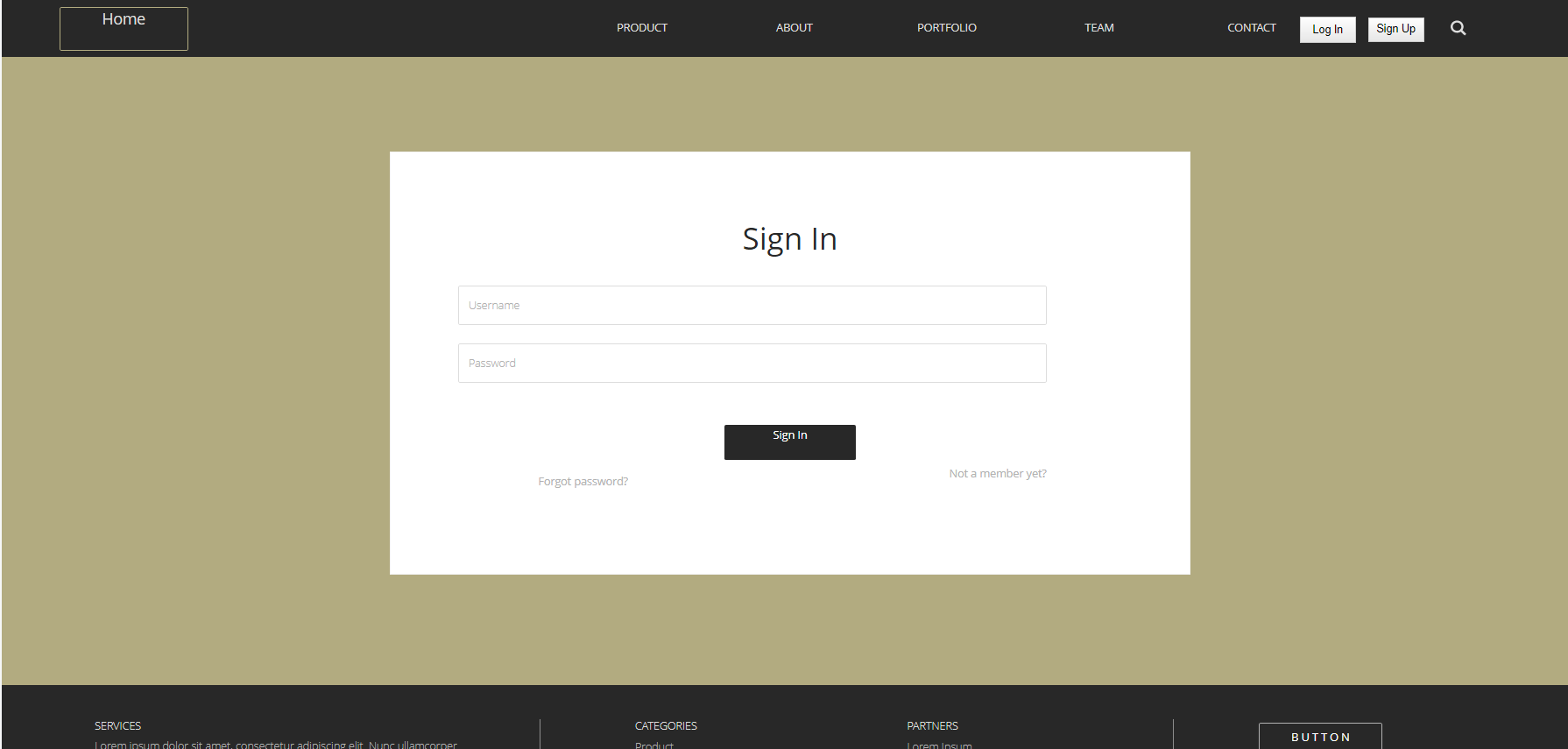
## Assumptions and Dependencies

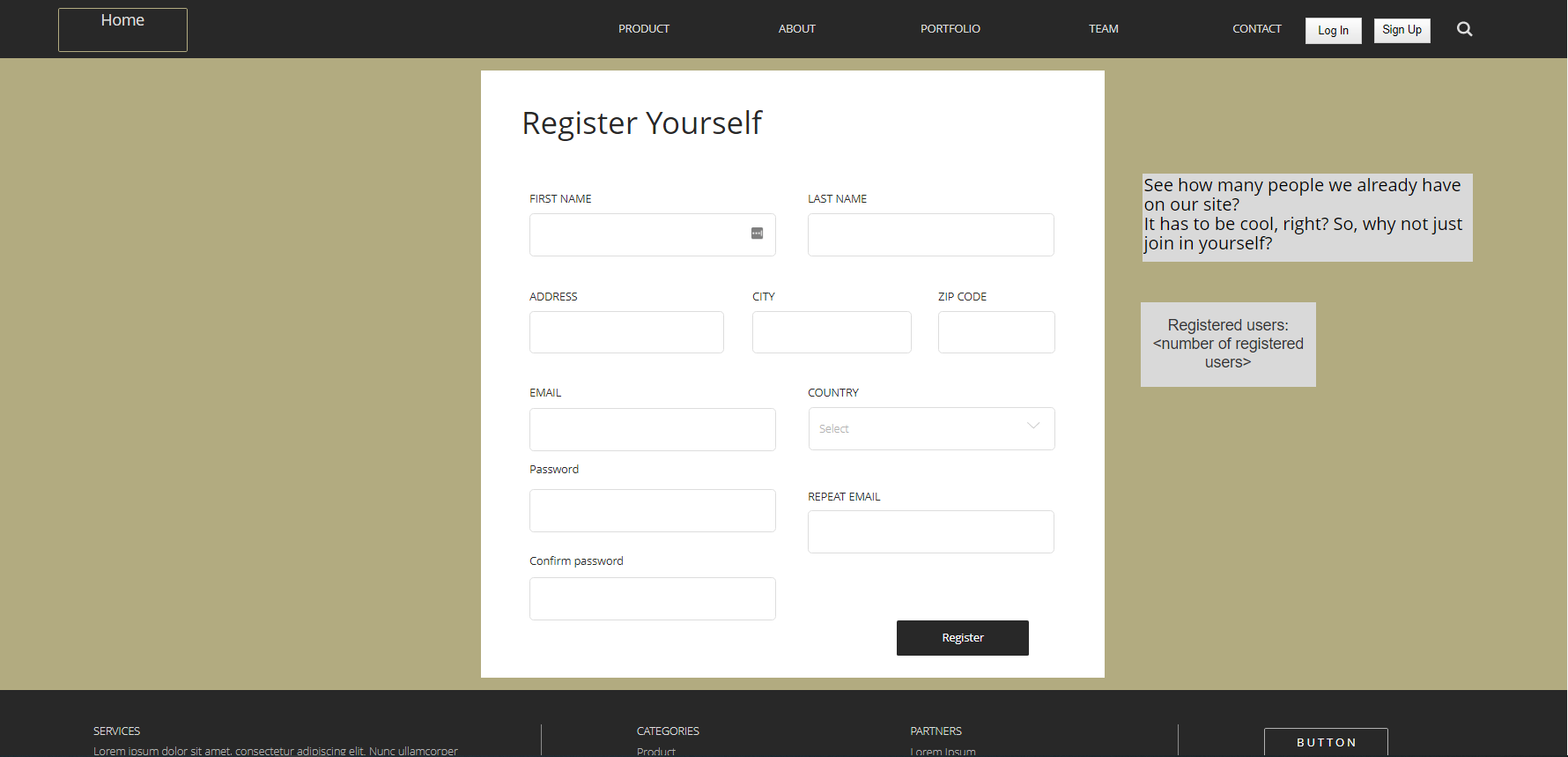
This product should not be dependent on any other product and should work as expected, as long as the internet browser on the accessing machine is up to date.

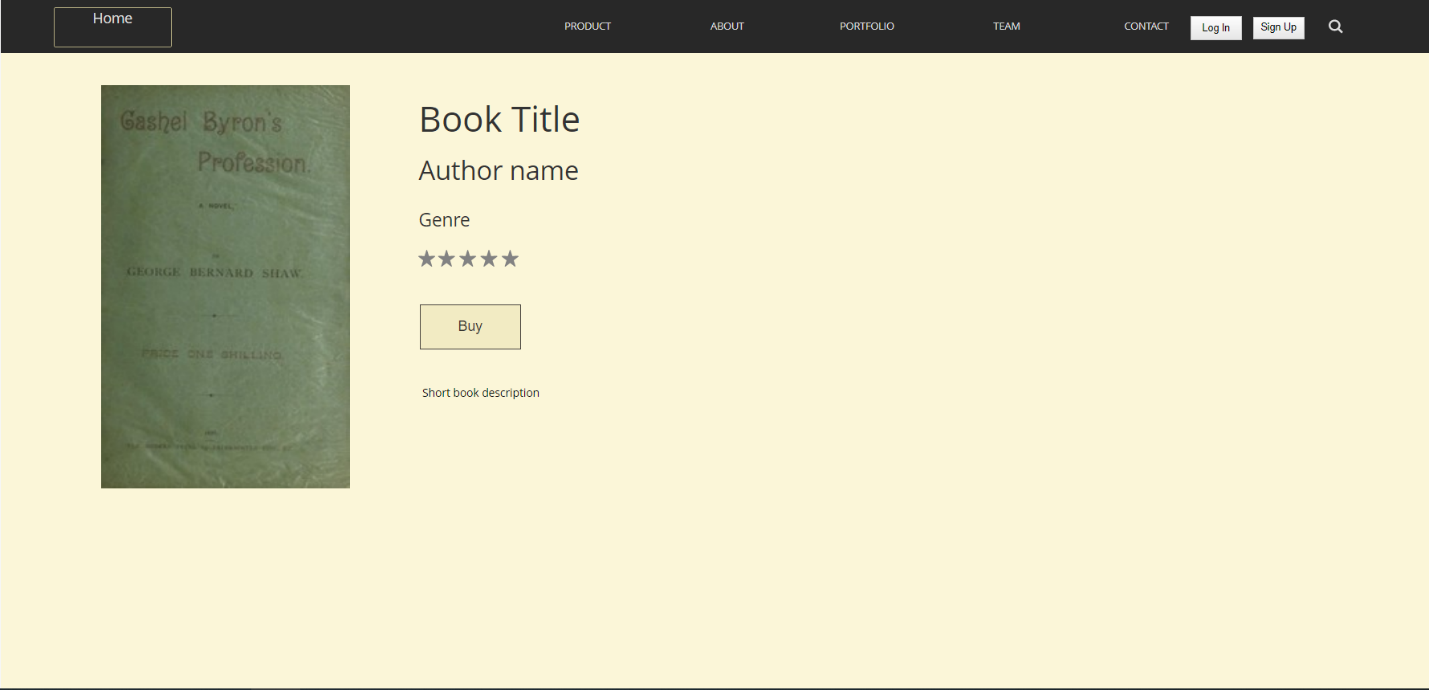
# External Interface Requirements

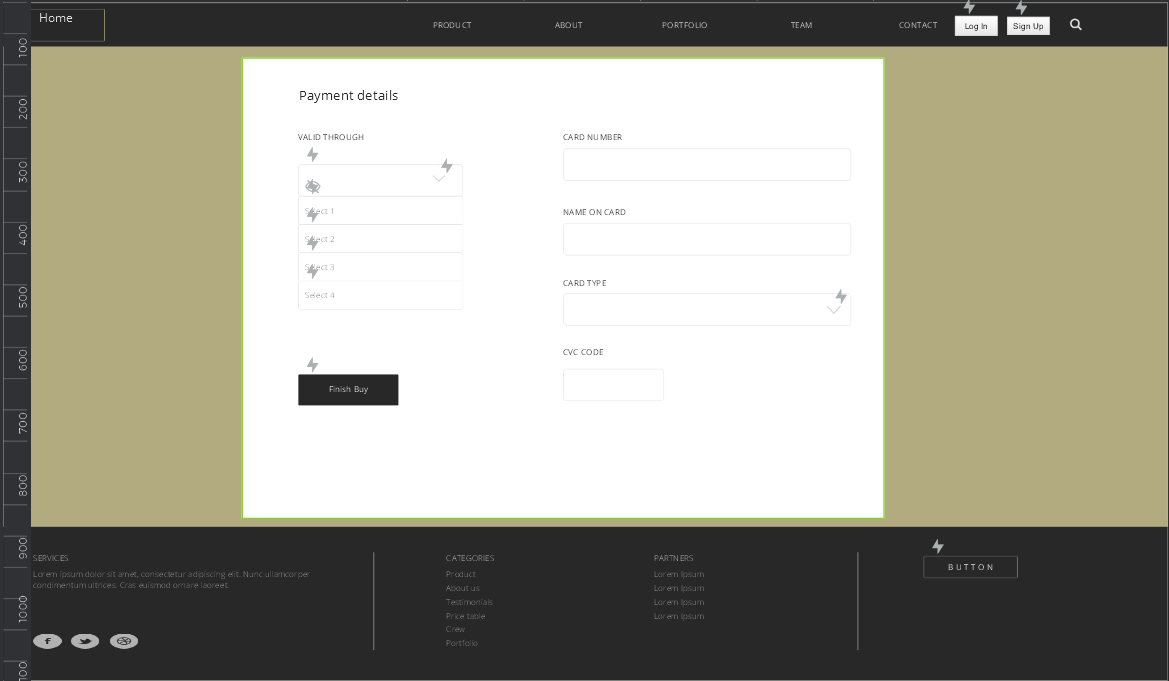
## User Interfaces

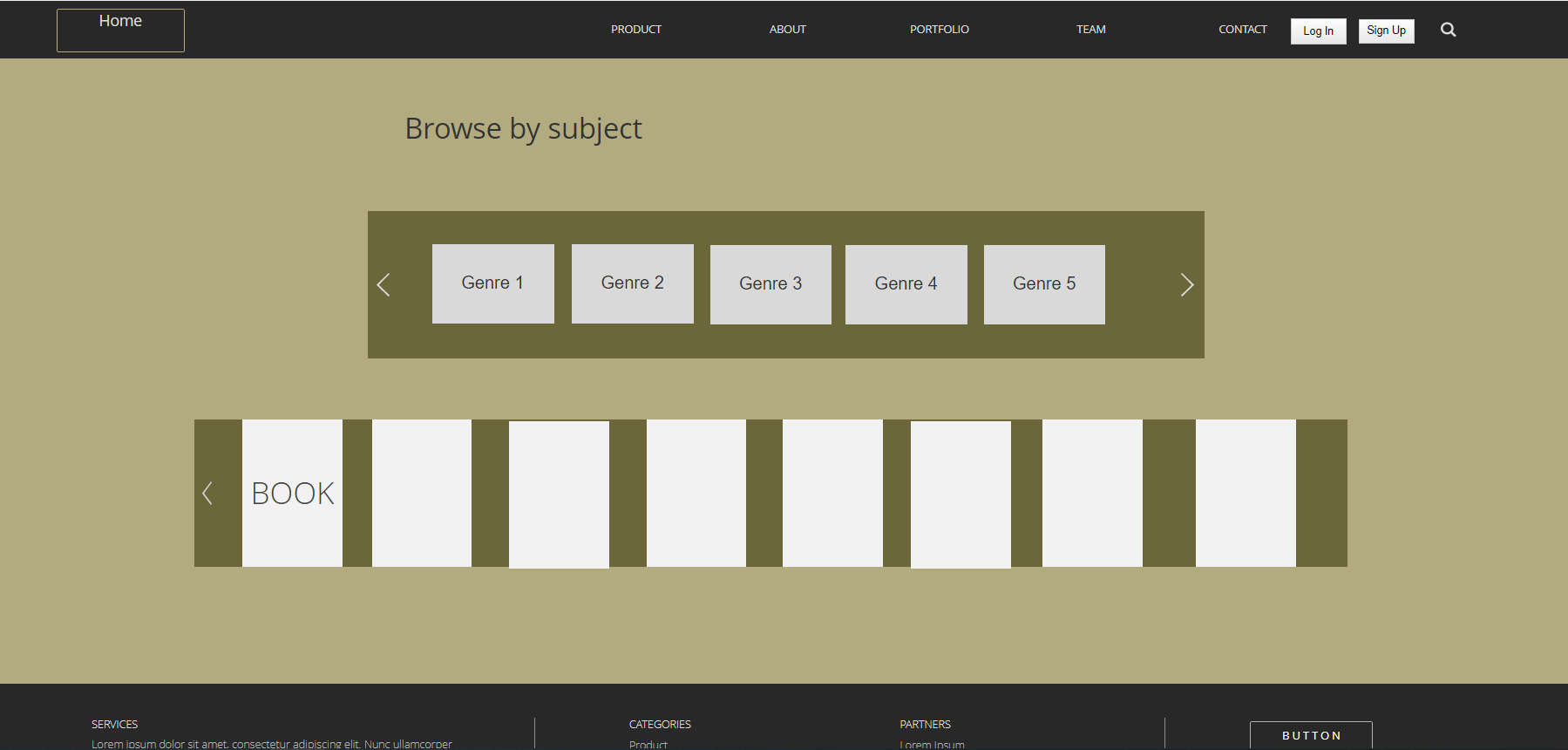
The user interfaces should be (using should be, as not a single interface has been done yet at the writing of this document) easy to use and highly intuitive. The style which we are looking towards is that of the https://openlibrary.org/ website. Any user with some knowledge of internet and websites should be able to use this product.











## Hardware Interfaces

Hardware interfaces are not of interest in the developing of this project, as the high level languages and the IDE`s used are capable of handling everything hardware related. This product should be able to be run on all and every up to date web browser, both PC or mobile.

## Software Interfaces

The software interfaces used will be able to communicate to each other in various ways

The SQL server will communicate to the backend application by using the built-in functions of the ASP.NET Core 3.1.

## Communications Interfaces

To be able to use this product, first of all, a stable internet connection is needed. Second, an up to date internet browser is necessary. As most internet web application, the communication standard used it HTTPS.   
The users can access (communicate to) the database through an electronic form.

# System Features

## Account creation

4.1.1 Description and Priority

To use the query function of the product, a user should have an account. This account is created through the web application. Readers, librarians and system administrators use the same login system, but have different clearance levels at using the database.

This feature has a high priority.

4.1.2 Stimulus/Response Sequences

There should be a register button, which will prompt the user to a page where they can create an account, given some basic information (email, name etc).  
There will also be a log in button, which should prompt the user to a login page.

There should be a different query page for each category of accounts.

4.1.3 Functional Requirements

REQ-1: The register submit button should register the new user in the data base

REQ-2: The login button should login the user at the correct clearance level for the account

REQ-3: When erroneous/wrong data is written in the text cassette, an error should alert the user and not grant the login

REQ-4: Password recovery option

REQ-5: No duplicate username/email allowed

REQ-6: Repeating password feature, to make sure that the user does not mistype the password

## Query function

4.2.1 Description and Priority

The query function(s) should be able to be accessed by all users, but at different levels. Readers should be able to easily search for a book.  
A librarian should be able to search for books, readers, stock information and user information.

This feature has a high priority.

4.2.2 Stimulus/Response Sequences

There should be several preset queries for easy access at usual items.

4.2.3 Functional Requirements

REQ-1: Several preset queries for easy access, available for readers and librarians

REQ-2: “Custom made” query for advanced, not preset, queries, available for librarians and system administrators

REQ-3: When erroneous/wrong data is written in the text cassette, an error should alert the user about the wrong query input. Eventually, the error message given by the SQL server should be displayed.

REQ-4: Search allows partial names matching / word fragment matching

## Reserve books

4.3.1 Description and Priority

To make the acquisition of a book easier and more pleasant to the younger generation, the application should allow a user to reserve a book at a specific library, for later pick-up.

This feature has a low to medium priority.

4.3.2 Stimulus/Response Sequences

After querying for a book, a reader can reserve that book at the desired library.

4.3.3 Functional Requirements

REQ-1: Easy to access reserve button for a specific book

REQ-2: The book should be reserved for 2 days after the button press (current and next day)

REQ-3: A user can only reserve one book

REQ-4: When a book is reserved, as long as it wasn`t picked up, it can be unreserved

REQ-5: User has to be logged in. If they’re not, the register page with be popped up.

## Backup

4.4.1 Description and Priority

Very high priority feature. Should back-up the database at regular intervals or when decided so by the system administrators.

4.4.2 Stimulus/Response Sequences

After some time before the last back-up, an automatic backup should be done, or after given the specific command.

4.4.3 Functional Requirements

REQ-1: Automatic backup

REQ-2: Manual backup

REQ-3: Previous version restore point

REQ-4: No changes to the database can be made while in backup mode

## Book recommendation

4.5.1 Description and Priority

Low priority. When logged in, the main page will recommend books similar to what the reader has already read, based on author and genre.

4.5.2 Stimulus/Response Sequences

No stimulus needed, the application will do this feature automatically.

4.5.3 Functional Requirements

REQ-1: The app will recommend books based on genre and authors already read.

REQ-2: If the user is new, random books will be recommended.

REQ-3: Books appear in a slider.

REG-4: There will be a fixed number at a time in the slider, and clicking an arrow will lead to the next set of books, with the same number of books.

## Favorite books

4.6.1 Description and Priority

Low priority. Each user can “favorite” a book, for easier access.

4.6.2 Stimulus/Response Sequences

When the ‘favorite” button is pressed, the book is added to their favorite books list.

4.6.3 Functional Requirements

REQ-1: The user has a favorite books list, which starts as empty.

REQ-2: When pressed the “favorite” button, a pop-up will appear, announcing that the book has been added.

REQ-3: Heart icon button near the book image that can be pressed.

REQ-4: User has to be logged in. If they’re not, the register page with be popped up.

## Log out account

4.7.1 Description and Priority

High priority. Drops the current logged in instance.

4.7.2 Stimulus/Response Sequences

A button to log out will replace the log in/register buttons, if an user is logged in.

4.7.3 Functional Requirements

REQ-1: A log in instance has to be active in order for the functionality to work.

REQ-2: Log out button.

REG-3: After pressing the button, the user will be prompted a message if they really want to log out.

## Add a book

4.8.1 Description and Priority

Low priority. User can request adding a book to the library.

4.8.2 Stimulus/Response Sequences

The user has to enter the “More” menu and to select the “Add a book” element.

4.8.3 Functional Requirements

REQ-1: User has to be logged in. If they’re not, the register page with be popped up.

REG-2: A form has to be completed with the required information about the book, like author etc.

REQ-3: The form has to be completed properly, otherwise there will be an error.

## Random book

4.9.1 Description and Priority

Low priority. Browse option that brings the user to a randomly generated book page from the site.

4.9.2 Stimulus/Response Sequences

The user has to enter the browse menu and click the Random Page element.

4.9.3 Functional Requirements

REQ-1: The library has to not be empty.

REQ-2: A subject can be chosen for the random book.

## Book Subjects List

4.10.1 Description and Priority

High. A slider list with all book subjects available in the library.

4.10.2 Stimulus/Response Sequences

User can click one of the subjects icons, which will lead him to a page with the list of books from the specific subject.

4.10.3 Functional Requirements

REQ-1: Library has to not be empty, or otherwise the list will be empty.

REQ-2: It’s implemented in a slider that can be navigated by clicking on 2 arrows – left and right.

REQ-3: Subjects have to be unique.

REQ-4: The number of books present in the library from the respective subject is displayed under the subject icon.

REQ-5: The number of books present in the library for only one book

## Menu Header

4.11.1 Description and Priority

High priority. The header div that contains all the main menu elements like: browse, more, search, log in, register.

4.11.2 Stimulus/Response Sequences

The user can interact with the menu elements, which have their own features. Clicking the site icon should bring the user to the home page.

4.11.3 Functional Requirements

REQ-1: The header should contain all the required components, in an organized and intuitive manner.

REQ-2: The should contain a link towards the home page.

## Book Slider

4.12.1 Description and Priority

High priority. A slider that contains a fixed number of books at a time.

4.12.2 Stimulus/Response Sequences

The user can navigate the slider by clicking the left or right arrow. If a book picture is clicked, then the user is sent to the page of the respective book. If the user is not registered, clicking the borrow button will lead to the registration page.

4.12.3 Functional Requirements

REQ-1: Borrow button under the book image.

REQ-2: Around 6 books are displayed at a time

REQ-3: The list gets shifted by the number of books displayed, so a completely new list of books appears.

## Area with number of already registered users on the register page

4.13.1 Description and Priority

Low Priority.Text area that shows the number of already registered members

4.13.2 Stimulus/Response Sequences

It can’t be interacted with by the user.

4.13.3 Functional Requirements

REQ-1: A link to the database needs to be properly formed, in order to querry the correct number of registered users.

# Other Nonfunctional Requirements

## Performance Requirements

The system doesn’t have a high complexity, so it is expected to be able to run smoothly on most machines that are still in use.

## Safety Requirements

Not applicable.

## Security Requirements

The basic users should not be able to modify the database in any way, except to register a new account and reserve books. This safety regulation should be implemented in two ways:  
Different categories of users should use a different authentication method to the SQL server to prevent the possibility of wrongfully modifying the database, and the query should be checked BEFORE submitting it to the SQL server.

The only category of users that should be able to modify the database and, potentially break the system, are the system administrators. They are not subject to any restrictions.

The project should come with a backup functionality.

For the register/login function, and to make sure that the database remains secured against unjust modifications done by unqualified users (readers), we’re using ASP.NET Core specific functions for authentication/login.

## Software Quality Attributes

The main qualities of this product will be ease of use and the query power, which will be implemented as a “custom made” query, which the (authorized) users will be able to write.

The software will be implemented in such a way that allows modularization and will make sure that new functions can be easily added.

The code will be written with ease of maintenance in mind.

Code quality rules:

1. Strict DOCTYPE for HTML

2. Proper identation. New line after starting a tag, and ending the tag on a new line.

3. External CSS and Js files.

4. Properly nested tags. Each new nested tag is marked by a tab in addition to the parent tag, relative to the start of the row.  
5. Eliminating unnecessary divs or spans, or other such wrappers.

6. Proper name conventions, like class names, id names, variable names etc.

7. Leaving all styling to the CSS file, so style tags should be mostly absent from the HTML file.

8. Proper ordering of contents.

9. Proper object incapsulation for Javascript.

10. Spaces around operators.

## Business Rules

To assure that the database is not corrupted by unauthorized users, only the system administrators can alter the database in any way they see fit.

Librarians/users that work at the library will have access to modify the content of the database with some restrictions. Usually, it should revolve around adding/removing books, users, and borrowing status.

Readers are only able to view the books available and reserve them.

# Other Requirements

SQL server will be used as the database. It will be maintained by one of the system administrators.

Appendix A: Glossary

Not applicable

Appendix B: Analysis Models

Mockups: should be attached together with this file

Appendix C: To Be Determined List