

---

# **Software Requirements Specification**

**For**

## **Web Application for managing Library's Activities**

**Written by Claudia Vasile**

Prepared for university project

19.03.2017

## Table of Contents

Table of Contents .....	2
Revision History .....	2
1. Introduction .....	3
1.1 Purpose .....	3
1.2 Product Scope .....	3
1.3 Intended Audience and Reading Suggestions .....	3
1.4 Definitions, Acronyms, and Abbreviations .....	3
1.5 References .....	3
2. Description .....	4
2.1 Product Perspective .....	4
2.2 Product Functions .....	5
2.3 User Classes and Characteristics .....	6
2.4 Operating Environment .....	6
2.5 Design and Implementation Constraints .....	6
2.6 User Documentation .....	6
2.7 Assumptions and Dependencies .....	6
3. External Interface Requirements .....	7
3.1 User Interfaces .....	7
3.2 Hardware Interfaces .....	10
3.3 Software Interfaces .....	10
Client .....	10
Server .....	11
3.4 Communications Interfaces .....	11
4.1 System Feature 1 .....	11
5. Other Nonfunctional Requirements .....	12
5.1 Performance Requirements .....	12
5.3 Security Requirements .....	12
5.4 Software Quality Attributes .....	12
6. Other Requirements .....	13
Appendix A: Glossary .....	13
Appendix B: Analysis Models .....	13

## Revision History

Name	Date	Comments	Version
Claudia Vasile	16.03.2017	First Revision	1

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of a online library application. It will explain the features of the system, the interfaces of the system and what the system will do.

## 1.2 Product Scope

The purpose of this project is to develop a software system that allows the following specific activities:

- The regular user can see the book catalog, choose the one that he wants and borrow it. He can also write a short review about it.
- The admin user which will be the librarian will have the access the entire user database and manage it. He can also add or remove books from the catalog.

More specifically, the scope of this product is to facilitate the process of people trying to borrow books, so, at this point, the product will come to his house after he places his order. When he chooses a book, he can see if the book is available or not: if it is available he can add it to his cart, if it's not, he can only see the description and the time when it will be available. The librarian/admin manages the book catalog: to add or remove a book, or to update a book's description.

## 1.3 Intended Audience and Reading Suggestions

The following document is addressed for both developers and users of the application.

## 1.4 Definitions, Acronyms, and Abbreviations

Acronyms and Abbreviations	Details
USR	Regular user
LB	Librarian/Admin
MVC	Model View Controller

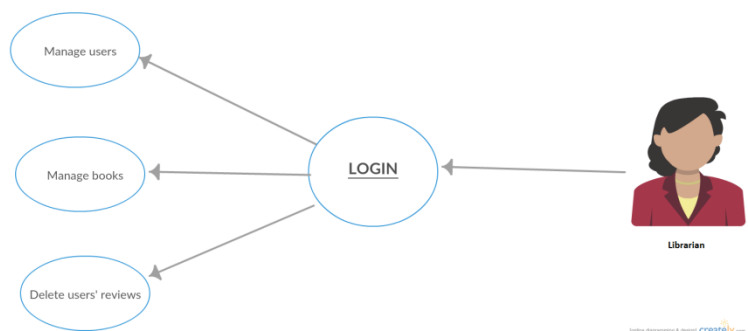
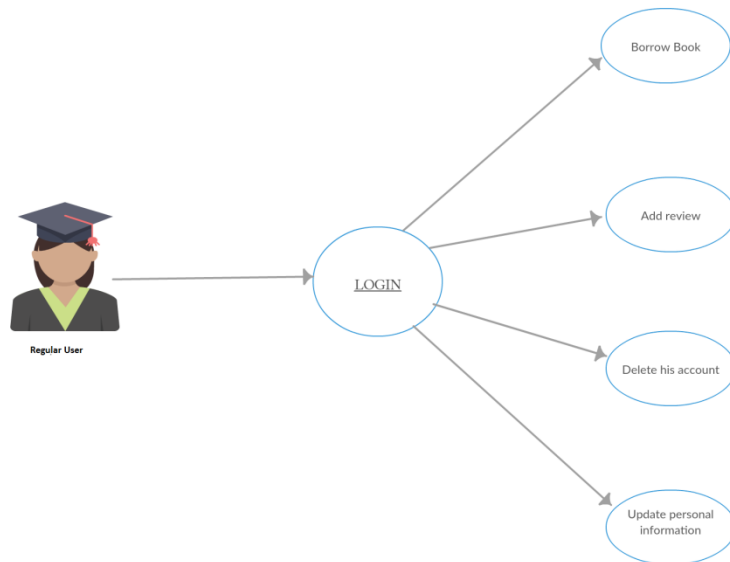
## 1.5 References

<http://es6-features.org/>  
<https://www.w3.org/>  
<http://getbootstrap.com/>

## 2. Description

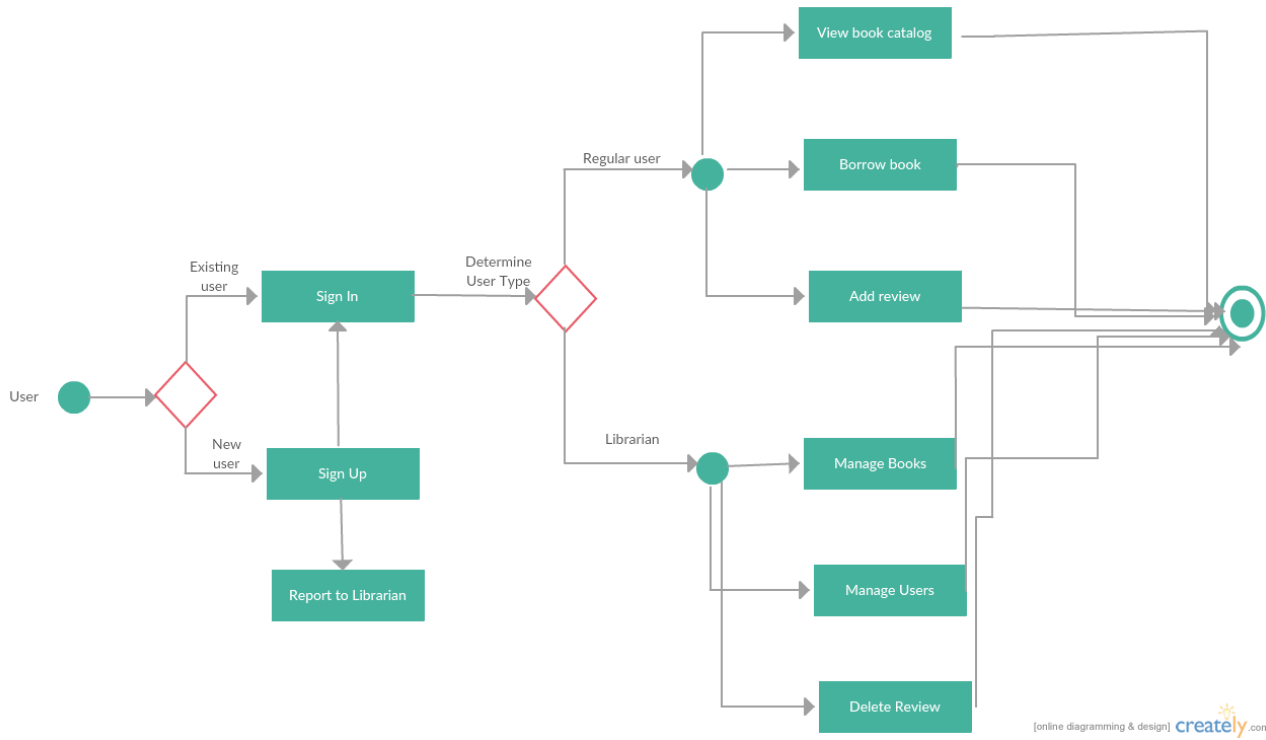
### 2.1 Product Perspective

#### USE CASE DIAGRAM:



This product consists of a web application designed to rent books and read a short description about them.

## 2.2 Product Functions



### ACTIVITY DIAGRAM:

The application will implement the following main functionalities:

- The user interface is available for usage through web browsers (Google Chrome, Mozilla Firefox, Internet Explorer), secure, with different access level for users, depending of their role.
- The data from the Online Library will be easily accessible to all the users of the application.
  - The application will serve as an online platform for borrowing books.
  - The librarian will be able to add or delete books, add description, delete reviews. Also, after the user reads the book, he will send it back to the library and the librarian needs to give his accept about the arriving of the book
  - The usual users borrow books, read description, leave a review. Also, after the books arrives at this house and he reads it, he will send it back. He only has to pay for the transport fee.

## **2.3 User Classes and Characteristics**

There will be two types of users

- Admin user type which will be assigned to the Librarian.
- Regular user type

## **2.4 Operating Environment**

The application will support integration for Google Chrome, Mozilla Firefox and Internet Explorer.

## **2.5 Design and Implementation Constraints**

The issues that will limit the options available to the developers are:

- Large number of entries into the database, that could be hard to manage and query.
- Security of the accounts
- Integration on different web browser.

## **2.6 User Documentation**

Not in this scope.

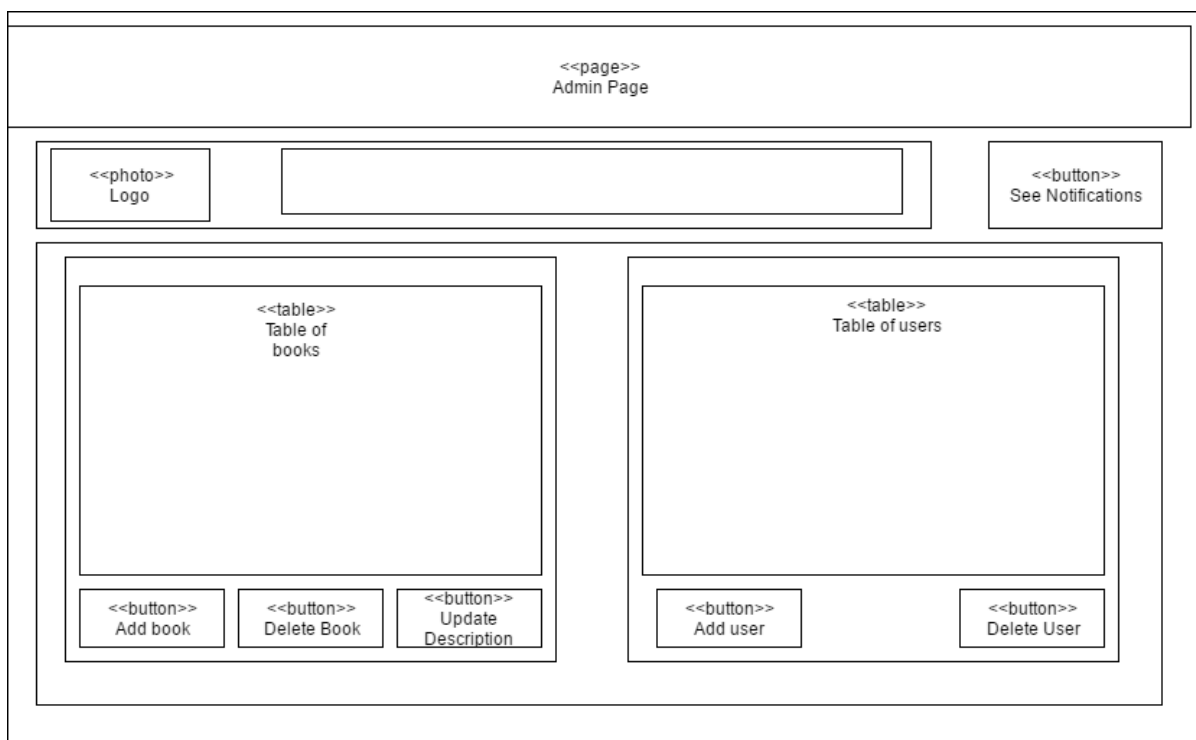
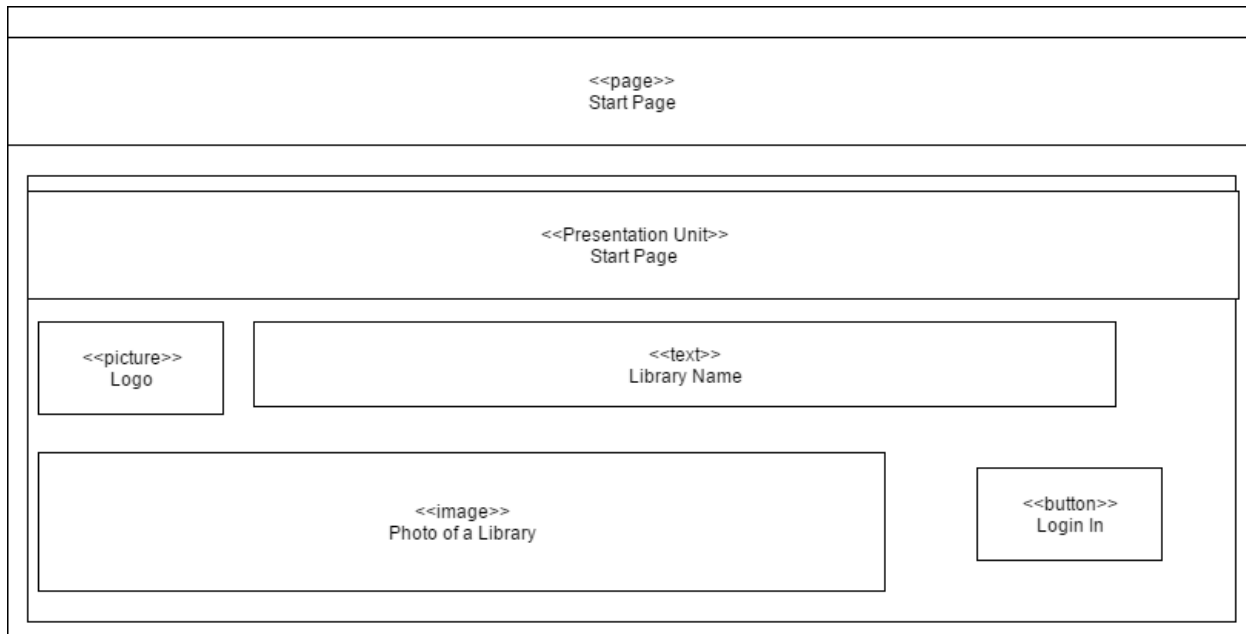
## **2.7 Assumptions and Dependencies**

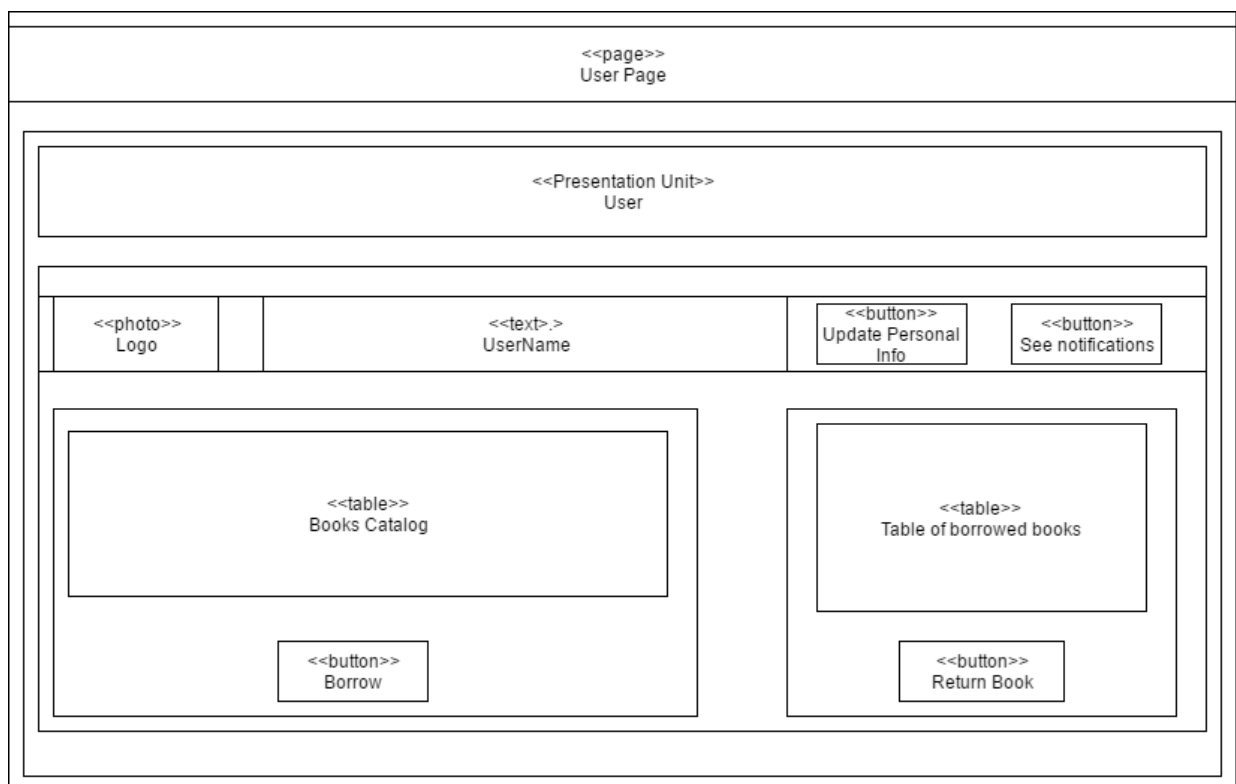
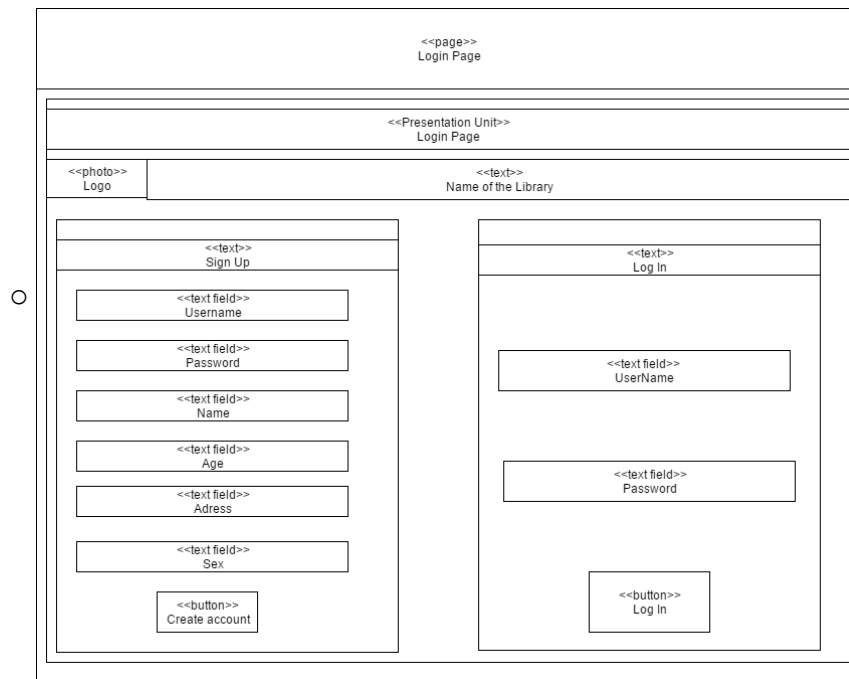
- Java
- CSS
- HTML
- Bootstrap
- JavaScript

## 3. External Interface Requirements

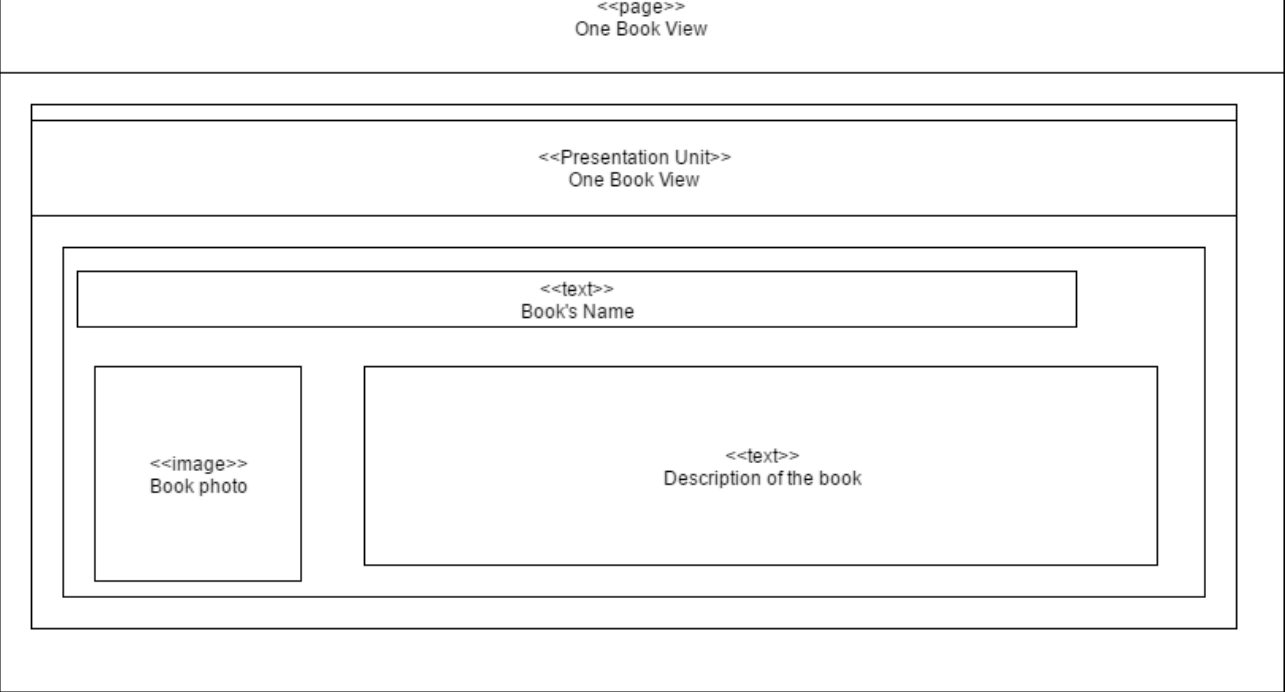
### 3.1 User Interfaces

PRESENTATION DIAGRAMS:

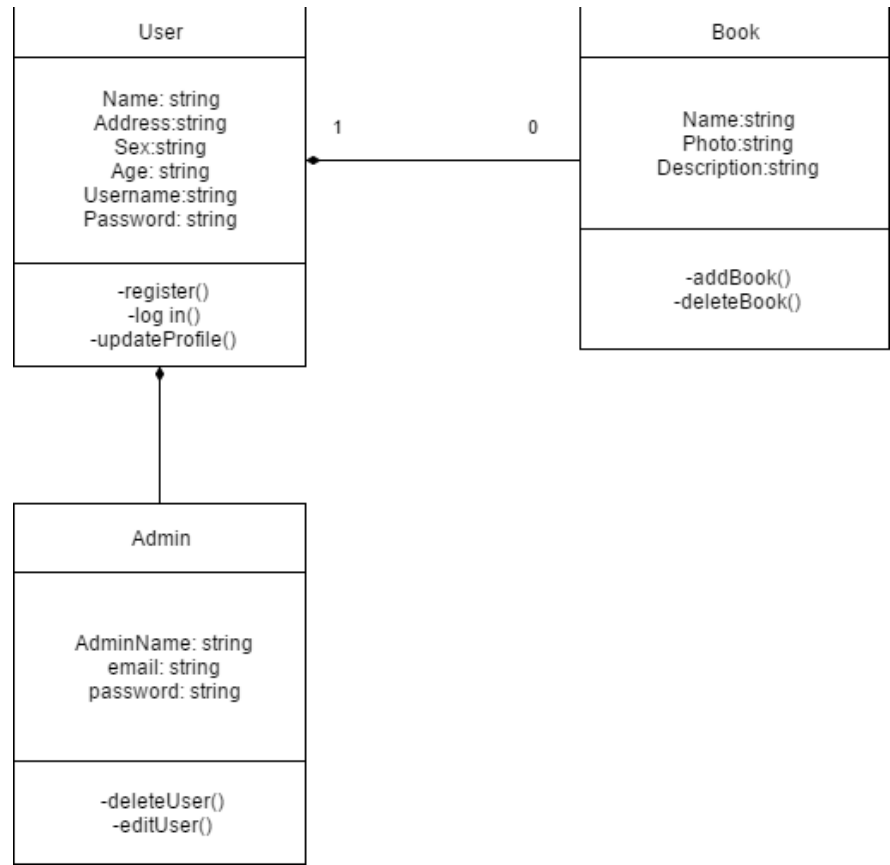








**CLASS DIAGRAM:**



Main window – Introductory page – a presentation of the Library

- User view – in this view a list of books will be displayed with the possibility to add one to cart
- Admin view – in this view it will be 2 types of lists: one for the books and one for the users.
- Login view
- One book View – see a complete description of a Book: with photo and description

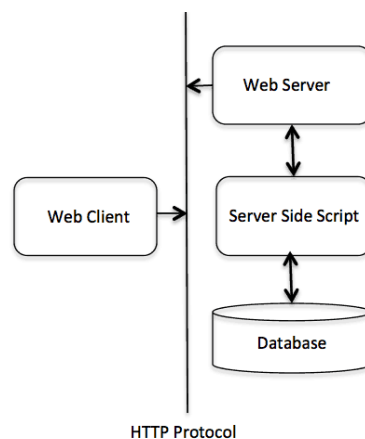
## 3.2 Hardware Interfaces

Not in this scope.

## 3.3 Software Interfaces

The application uses HTTP Client -Server communication.

The client and server are the end systems also known as Hosts. The Client initiates contact with the server to request for a service. Such as for Web, the client is implemented in the web browser and for e-mail, it is mail reader. And in similar fashion, the Server provides the requested service to client by providing with the web page requested and mail server delivers e-mail.



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

### Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

## Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

### 3.4 Communications Interfaces

- Transfer protocol used: HTTP

## 4. System Features

ID	SPECIFICATION
1	For information storage and management, a database served needs to be configured.
2	Librarians have the access to entire users database and books database, being able to add/edit/delete books or users. They are the administrators of the system and they have the full access rights.
3	Regular users will be able to see the books catalog, add them to cart and leave a review.
4	Authentication

### 4.1 System Feature 1

#### 4.1.1 Description and Priority

- **Authentication/Log In Function**

One of the main features of the application will be the Login. Each new person who enters the website will have the possibility to create a new account. After he creates it and tries to log in, the authentication will be checked on the server side, the data will be sent through a POST request, in order to determine if the user exists. If he doesn't exist, he will be asked to try again.

#### 4.1.2 Stimulus/Response Sequences

The Main Window of the website will show a short presentation of the Library, and the possibility for the users to log in.

The Login View will give the user the possibility to enter his username and password. After a server-side validation, the user will be redirected to the current page, according to their role(regular user/admin).

After login, the View for admin staff will present a table of existing users with personal details and the books that they borrowed and a second table with the books that exists in the system. Buttons will provide the admin the possibility to perform different operations.

For students, the View will consist of a table of books with a photo and a short description of them. Also, it will exist a button in order for them to change his personal information.

#### **4.1.3 Functional Requirements**

REQ-1: After receiving the logged in flag, the app will redirect user to the main view, this being rendered based on the 'usertype' property which will be present in the received JSON from server.

REQ-2 After establishing the view, the user would be able to do a series of CRUD operations. For example: → For admin-type users: on clicking 'add a new new student', a POST request will be made and if this will be successful, a new record will appear in the tables. → For students: on clicking 'view personal report', a GET request will be made, and a list of personal grades will appear.:

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

Google chrome Version 56.0.2924.87

Mozilla 52.0

Internet Explorer 11

Not in this scope

### **5.2 Safety Requirements**

Not in this scope

### **5.3 Security Requirements**

#### **5.4 Software Quality Attributes**

##### **5.4.1. Quality Assurance**

The most important quality assurance aspects that will be taken into consideration:

- Protection of confidential data
- Easy maintenance by the administrative staff of the Library
- The system will be available anytime.

## **6. Other Requirements**

Not in this scope.

## **Appendix A: Glossary**

Not in this scope.

## **Appendix B: Analysis Models**

For this application we will be using a specific combination between Waterfall and Agile