Higher Diploma in Science in Web Technologies

Product Design Specification

The Book Boutique

Student Name: Antonijo Galic

Student Number: 17157757

Email address: x17157757@student.ncirl.ie

01. 12. 2018.

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	*Antonijo Galic*	*26/10/2018*			Product Design Specification
2.0	*Antonijo Galic*	*01/12/2018*			Update

UP Template Version: 12/31/07

Table of Contents

1 Introduction

Product design and development involves a number of processes in order to increase the chances of success once in the market. To do this, the process of product design is initiated with the creation of a product design specification or PDS. The PDS documents all of the necessary requirements and constraints the new design must adhere to. (Shah, 2012)

1.1 Purpose of The Product Design Specification Document

The purpose of the product design specification document for The Book Boutique is to describe the architecture and system design for developing a web application to function as an online e-commerce web application.

Product design specification document is drafted and created during early (planning) stage of the project and it will be shared with project supervisor and NCI (National College of Ireland) for purpose of a final project module.

2 General Overview and Design Guidelines/Approach

This section of product design specification (PDS) describes principles and strategies which will be used as a guide for designing and implementing the e-commerce web application The Book Boutique.

2.1 Assumptions, Constraints and Standards

The product design specification document defines that:

- the user has available access to a device which is connected to the internet,
- the e-commerce web application will be mainly developed for latest versions of Google Chrome and Mozilla Firefox web browsers but should be functional and friendly for the user in any browser,

- the e-commerce web application is a web responsive web application and should adapt to all screen resolutions and devices (desktop, laptop, tablet and phone),
- the e-commerce web application will be developed using Node.js,
- the e-commerce web application will communicate with mLab database service that hosts MongoDB database,
- the e-commerce web application will be connected with the Stripe payment platform to provide validation and charging credit cards,
- the e-commerce web application will be hosted using the Heroku platform.

3 Architecture Design

Web application architecture defines the interactions between applications, middleware systems and databases to ensure multiple applications can work together. (Stringfellow, 2017) The essential purpose of a web server architecture is to complete requests made by clients for a website. The clients are typically browsers and mobile apps that make requests using secure HTTPs protocol, either for page resources or a REST API. (Svitla, 2018)

This section outlines the e-commerce web application architecture, how the e-commerce web application interacts with other applications and how is the appropriate data correctly passed between applications.

3.1 Logical View

The logical view of e-commerce web application The Book Boutique addresses the functional requirements of the web application. The main point for a logical view of e-commerce web application is on the navigation bar. All pages for the registered user will be reachable using navigation bar. An unregistered user can only browse books and he needs to be logged in into a web application for additional features. For friendly navigability user can always go backwards in an application where the design standards expect that. The hierarchal diagram is created to show the navigation hierarchy of the e-commerce web application.

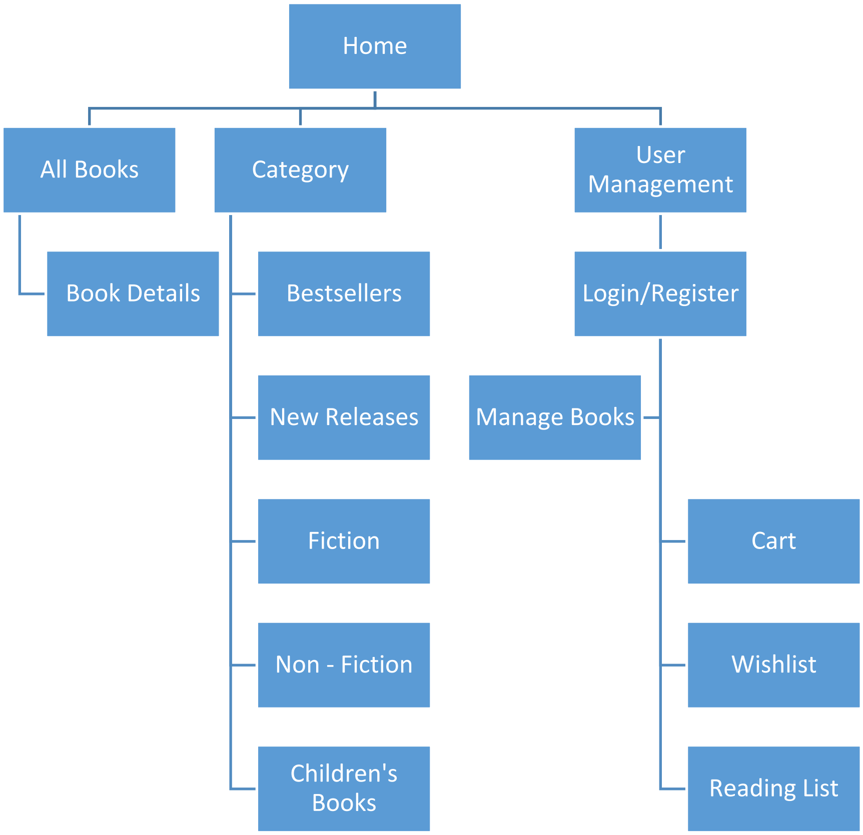


Figure 1. Hierarchal diagram

3.2 Hardware Architecture

The web application normally follows three-tier web architecture consisting of the client, web server, and data source. (Mithun & D'mello, 2017). The image below gives an overview of a hardware architecture for e-commerce web application The Book Boutique.

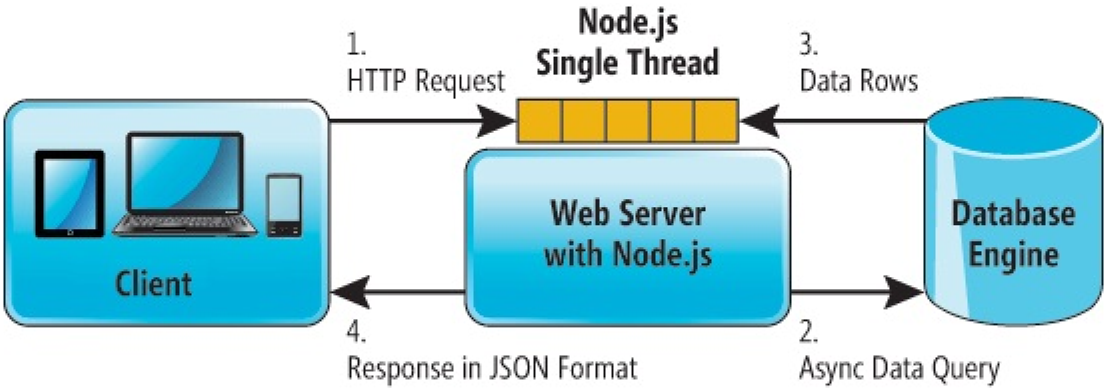


Figure 2. Hardware architecture (Vmoksha, 2017)

3.3 Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. (Eeles, 2006) This section of product design specification document gives details about software architecture for e-commerce web application The Book Boutique.

The e-commerce web application will be developed using online Cloud9 IDE (Integrated Development Environment). In order to develop an e-commerce shopping site that allows fluid and fast environment of the application, the main technologies what will be used are:

- Node.js – as an asynchronous vent driven JavaScript runtime, Node is designed to build scalable network applications,
- Express – is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications,
- MongoDB – stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

Stripe API will be used to provide payment functionality. Stripe is a software that allows individuals and businesses to receive payments over the Internet. Stripe provides the technical, fraud prevention, and banking infrastructure required to operate online payment systems.

Some of the npm (Node Package Manager) modules that will be used for the development of the e-commerce web application are:

- bcrypt-nodejs – helps to encrypt passwords,
- body-parser – allow the application to manipulate data in the application (create, delete, update),
- connect-mongoose – a MongoDB session store backed by mongoose,
- hbs – express.js view engine for handlebars.js,
- mongoose – elegant MongoDB object modelling for node.js,
- passport – express-compatible authentication middleware for Node.js.

3.4 Security Architecture

Security architecture refers to a plan and set of principles that describe the security services that a system is required to provide to meet the needs of its users, the system elements required to provide to implement the services, and also the performance levels required in the elements to deal with the threat environment. (Beal, 2018)

The user needs to be logged in into the application to access specific functionalities of e-commerce web application. Sensible data about the user will be protected in the way that e-commerce web application uses the bcrypt node module for data encryption and Passport.js for user authentication. The user needs to provide login details to verify credentials. If the user is verified, the user will be serialized into the session and logged in to the application. On image, below is explained the Passport.js authentication flow in e-commerce web application.

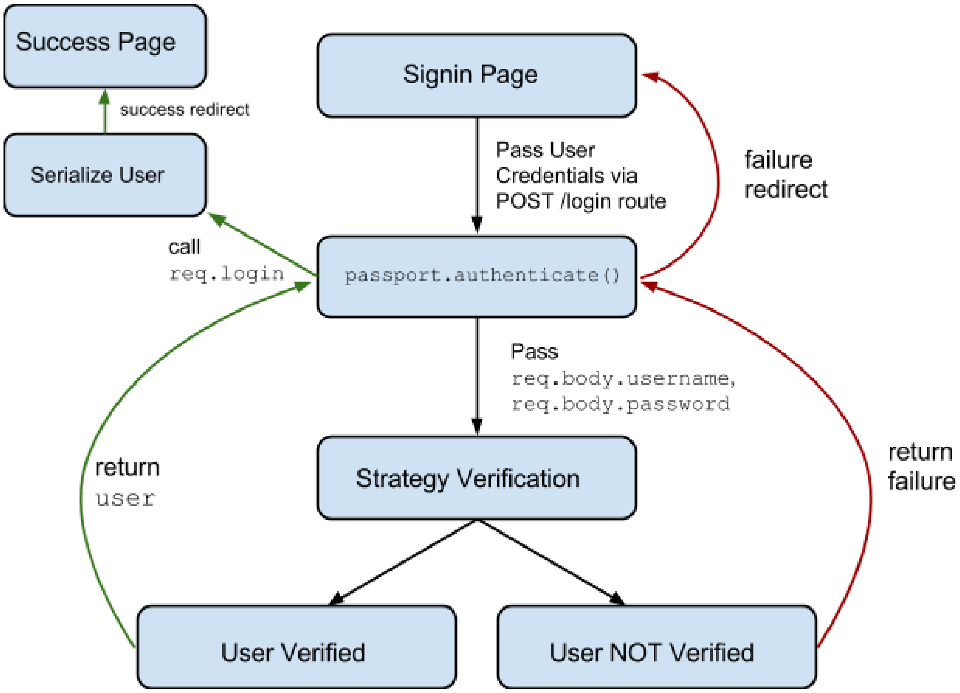


Figure 3. Passport authentication flow (Arneson, 2014)

3.5 Communication Architecture

Web-based communication is critical for web application architecture since the majority of global network traffic, and every single application and device uses that kind of communication. Web-based communication deals with scale, efficiency, robustness and security. Including the server and the client side, two programs running concurrently:

- the code which lives in the browser and responds to user input and
- the code which lives on the server and responds to HTTP requests.

The image gives a general overview of communication architecture in the e-commerce web application. The image doesn’t include communication with Stripe platform.

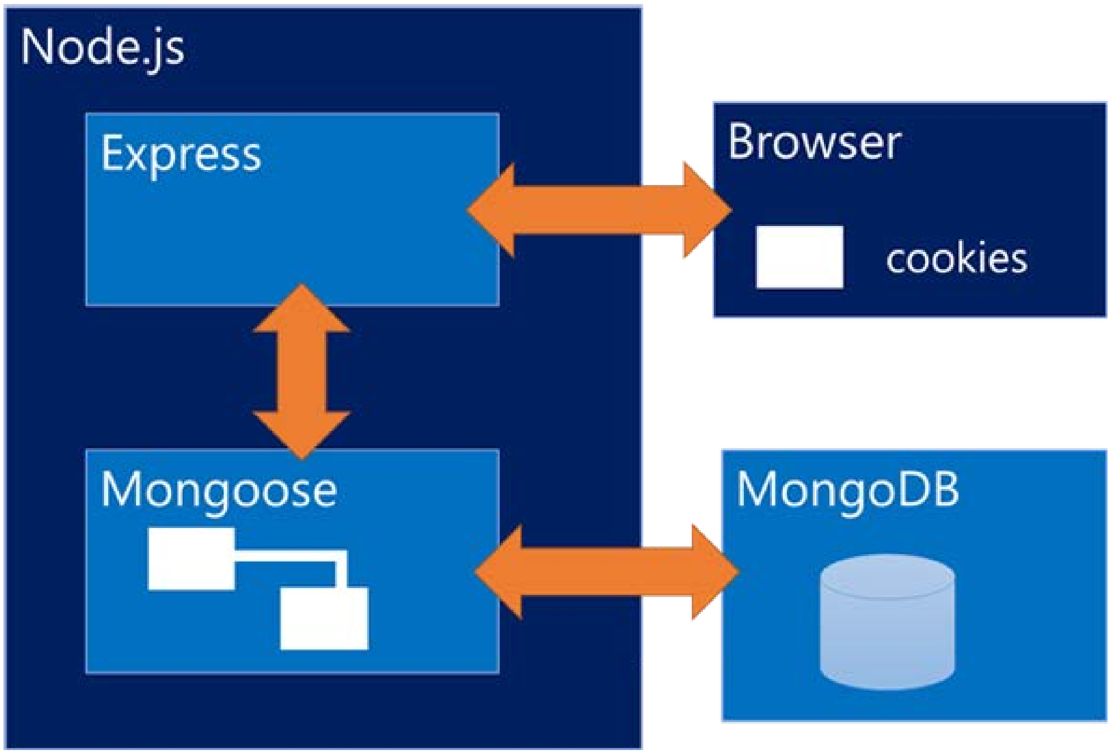


Figure 4. Communication architecture (Terkaly, 2014)

4 System Design

4.1 Use-Cases

Use cases for the e-commerce application The Book Boutique are described in the Requirements Specification document, pages 5 – 25.

4.2 Database Design

The server will store data in the MongoDB database and will be hosted at mlab.com. Mongoose is the ODM (Object Document Mapper) which will be used to communicate with our MongoDB database. MongoDB database will have next database collections (tables).

Books	Orders	Sessions	Users
bookId	orderId	sessionId	userId
cover	userId	session	email
title	cart { items { item { All from books collection }, qty, price }, totalQty, totalPrice }	expires	password
author			confirmpassword
publisher			address
description			secondaddress
category			city
year	address		state
price	name		zip
reviews	paymentId		

4.3 Data Conversions

The e-commerce web application will use body-parser node module to read the incoming request and parse it into a format from which it can easily extract relevant information in case that needs it.

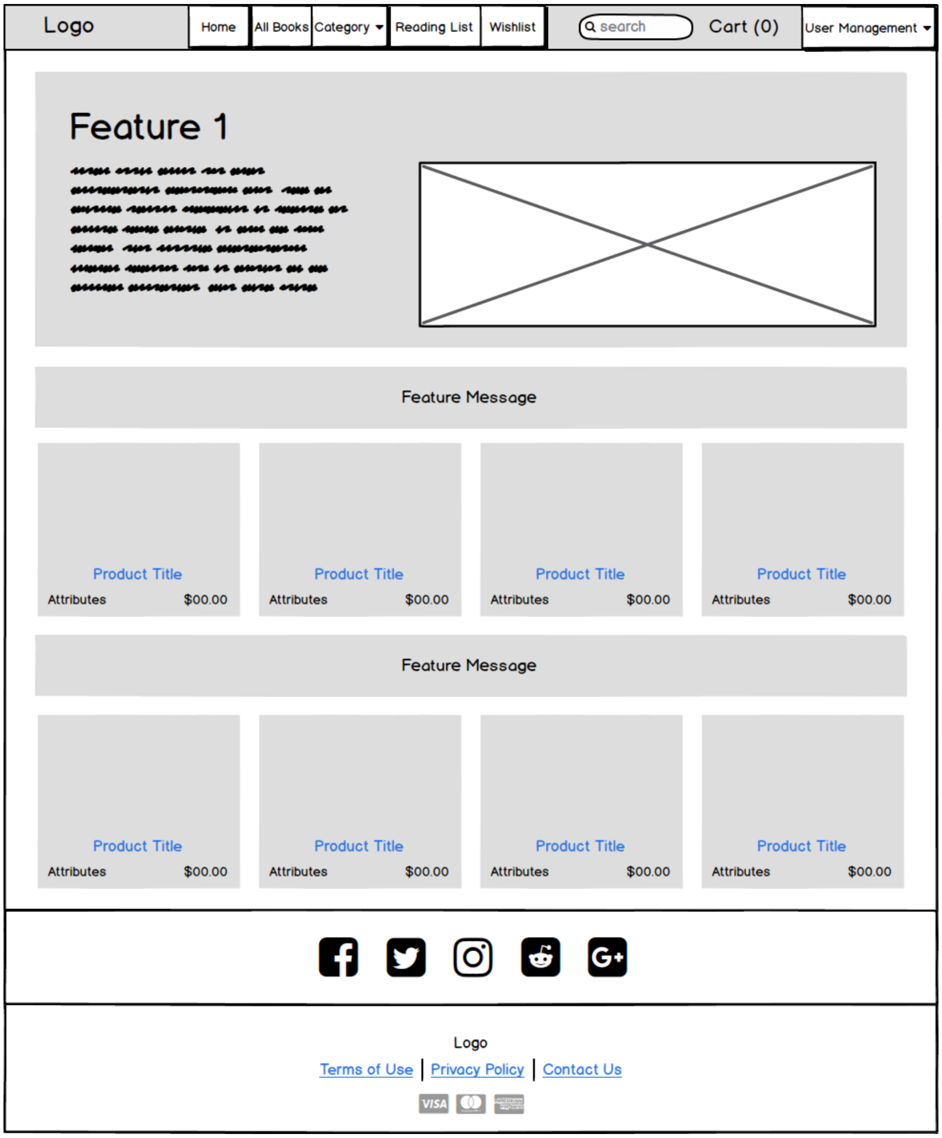
To communicate with the MongoDB database e-commerce web application will use mongoose. Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in typecasting, validation, query building, business logic hooks and more, out of the box.

4.4 Application Program Interfaces

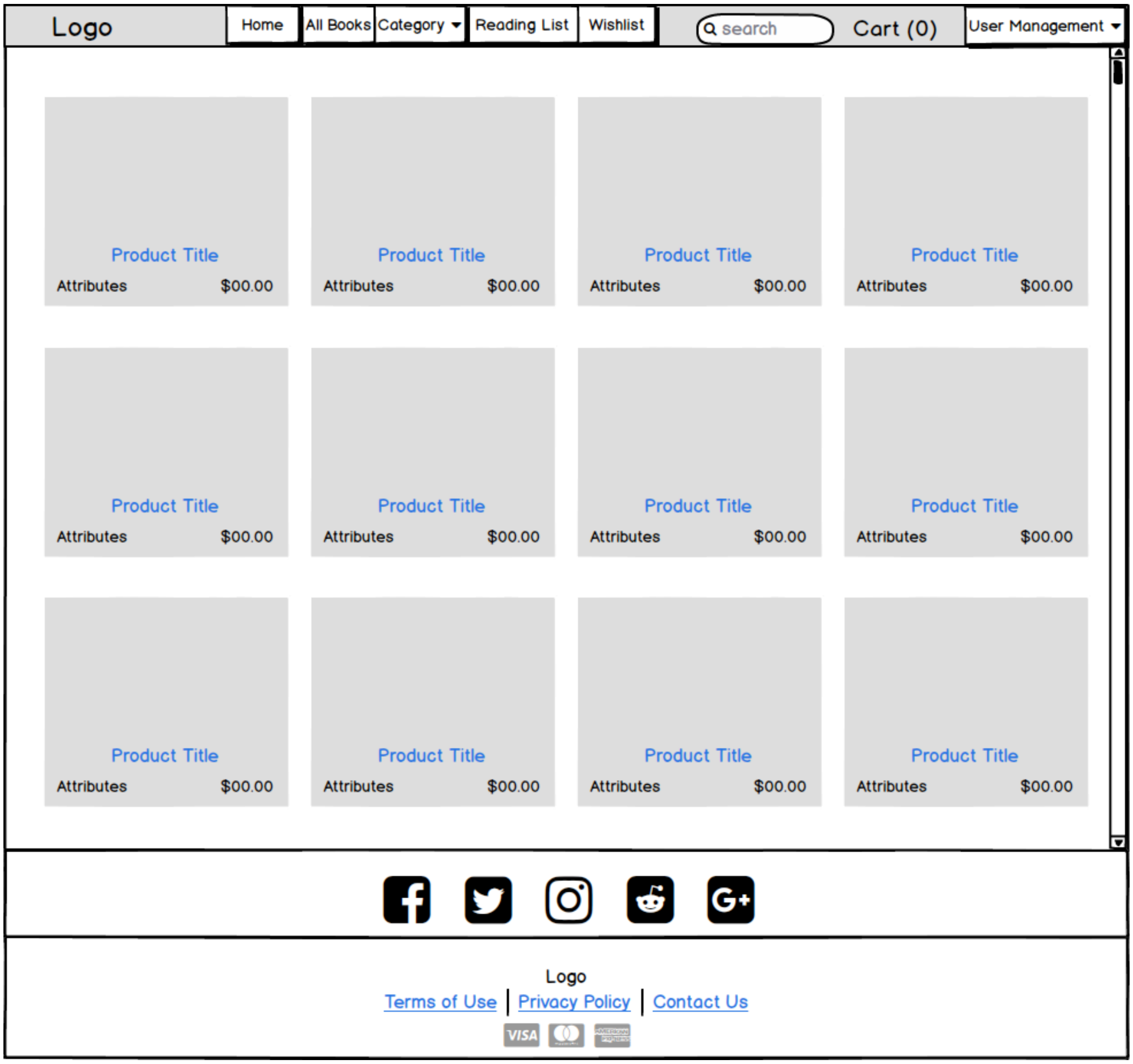
Use cases for the e-commerce application The Book Boutique are described in the Requirements Specification document, page 37.

4.5 User Interface Design

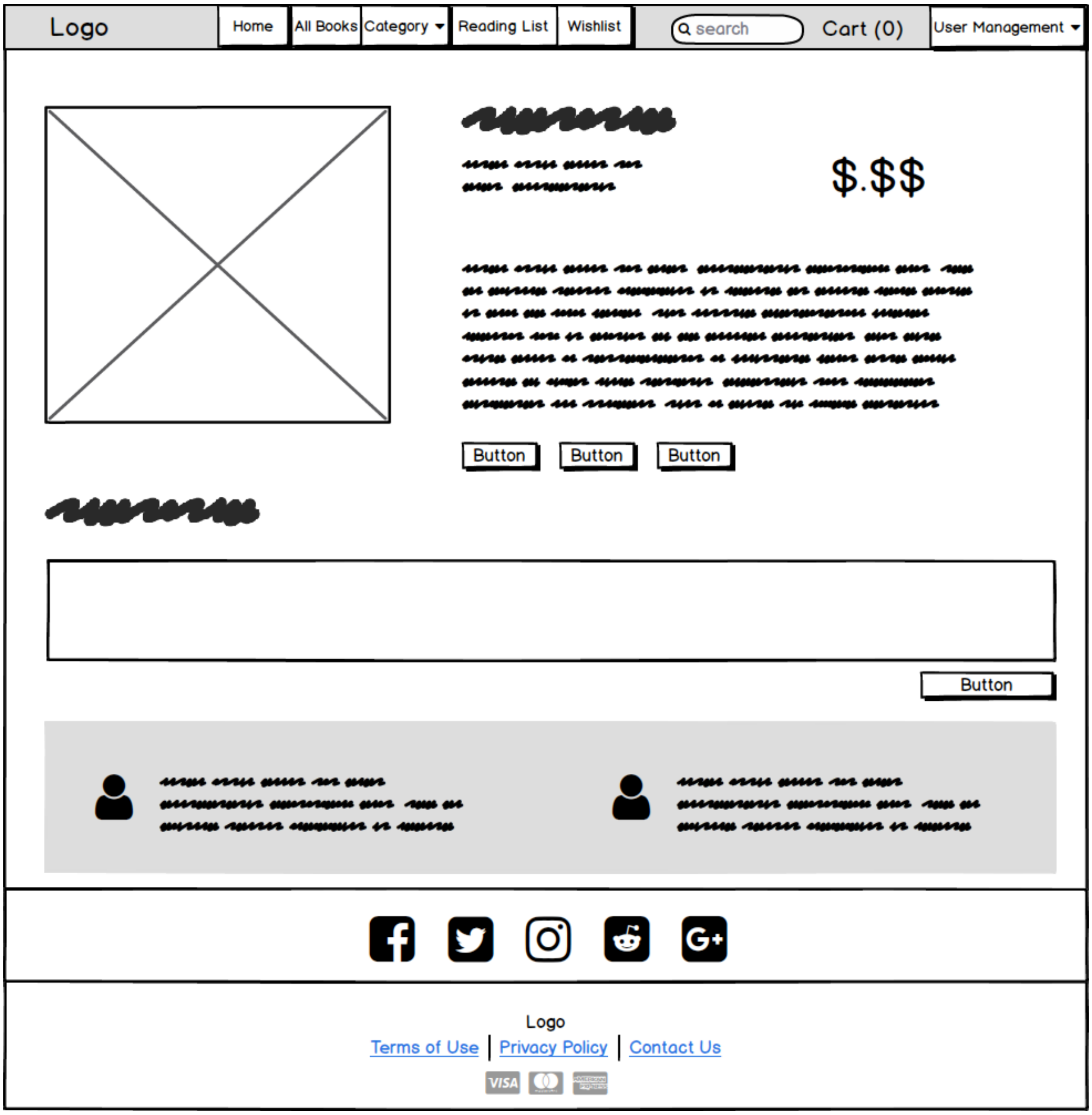
4.5.1 Home Page



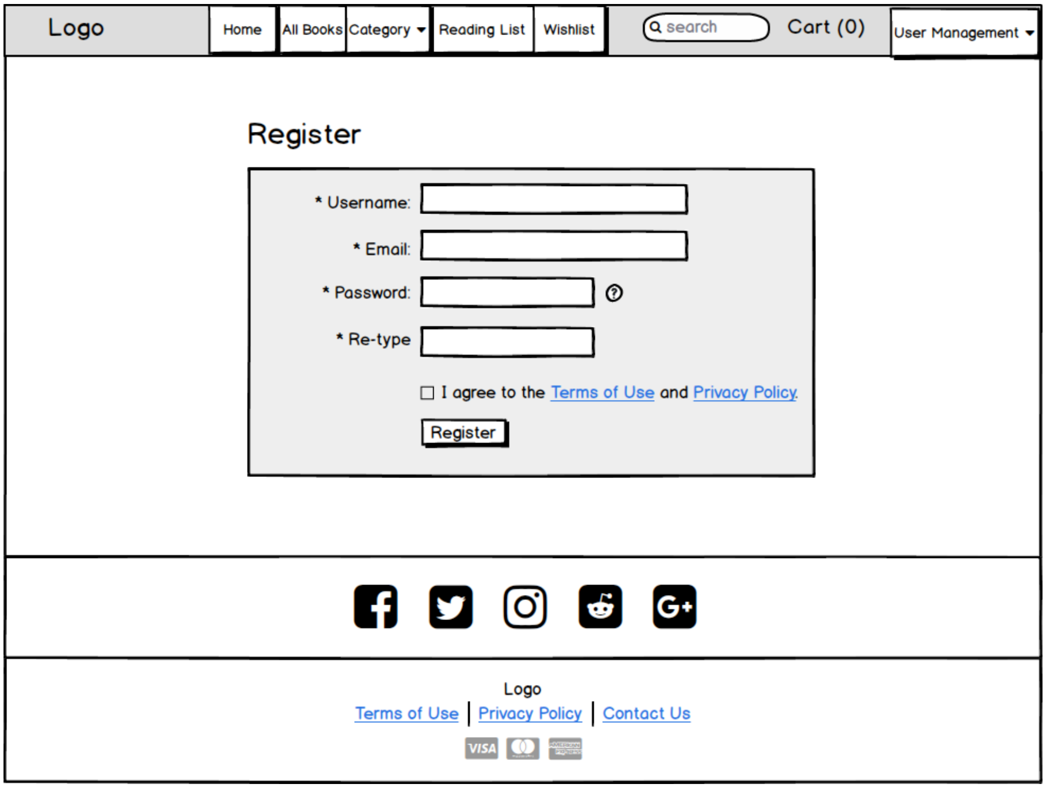
4.5.2 All Books



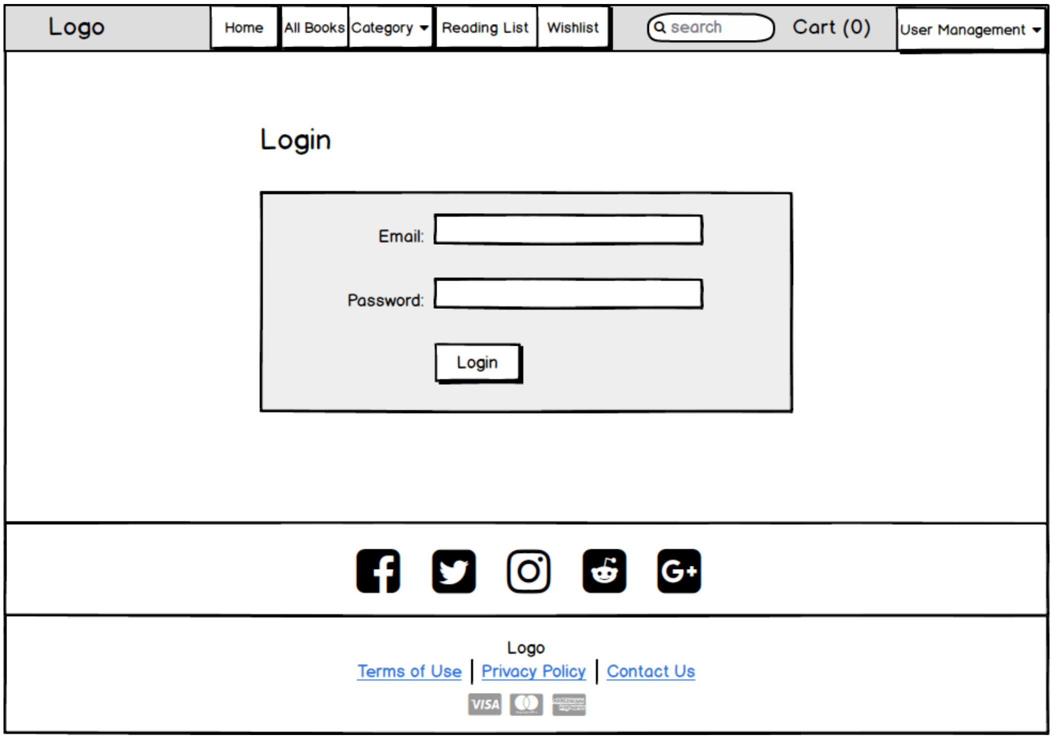
4.5.3 Book Details



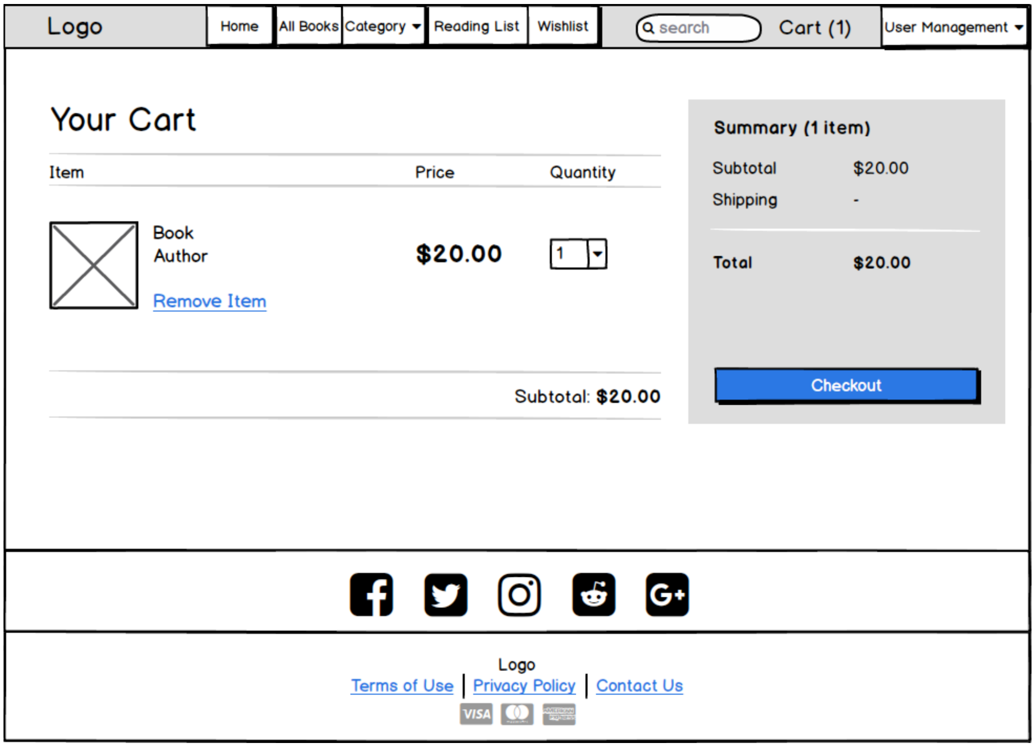
4.5.4 Register



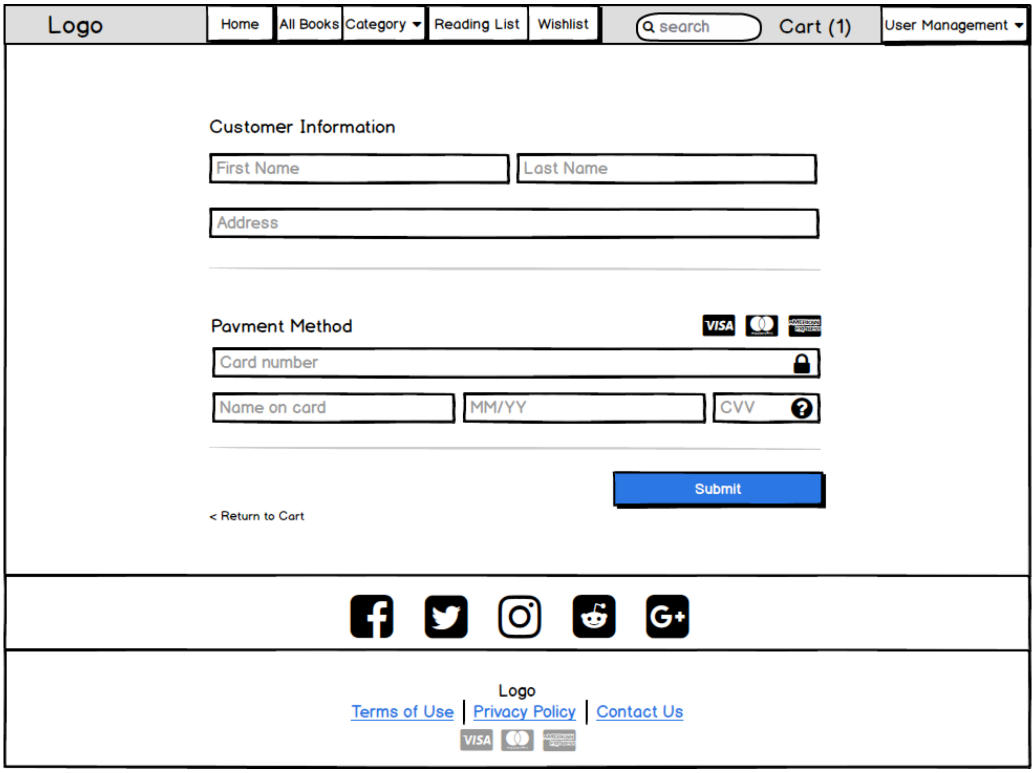
4.5.5 Log In



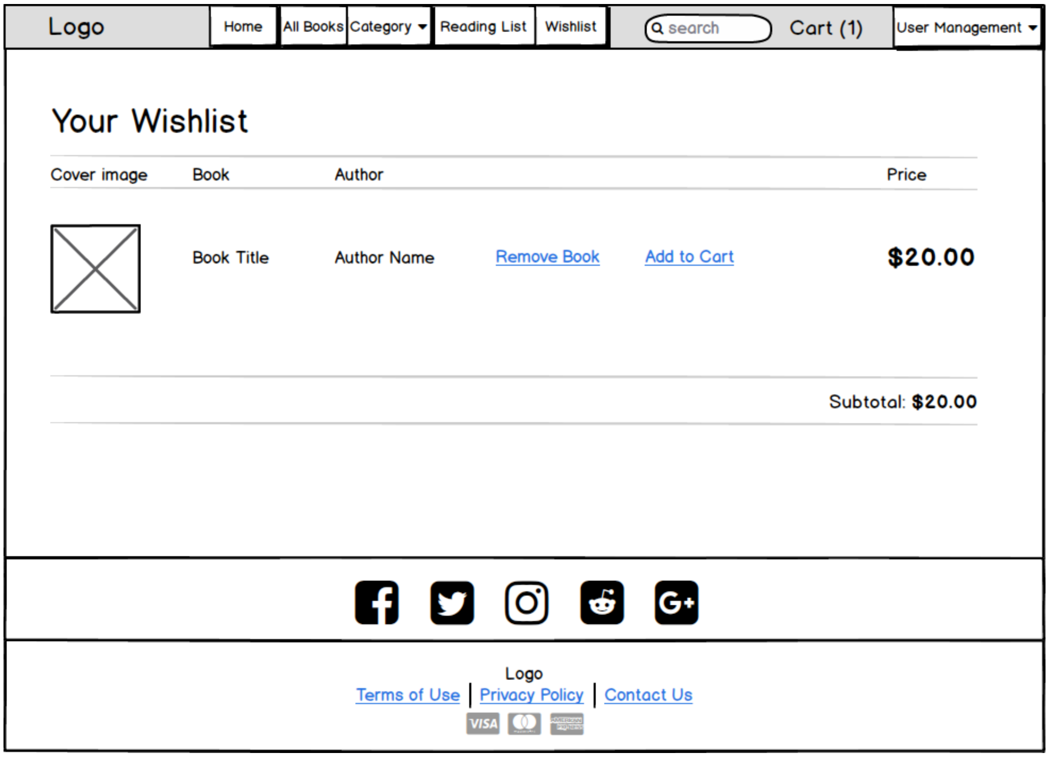
4.5.6 Cart



4.5.7 Checkout



4.5.8 Wishlist

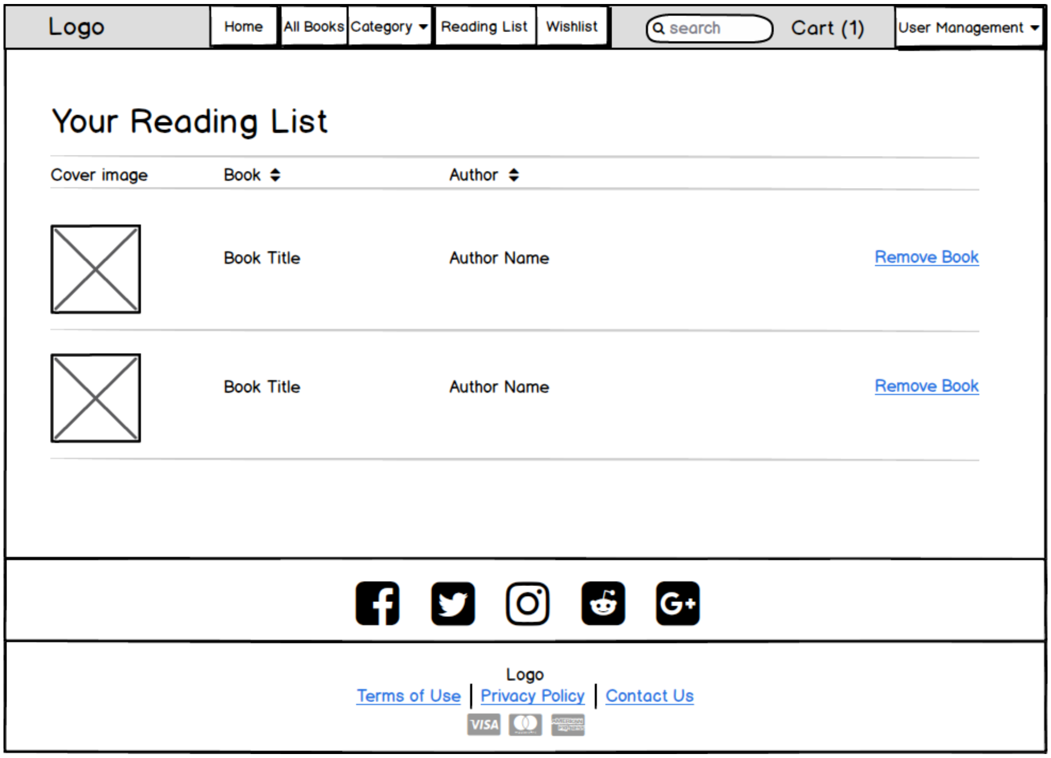


Your Wishlist

Cover image	Book	Author			Price
	Book Title	Author Name	Remove Book	Add to Cart	$20.00

Subtotal: **$20.00**

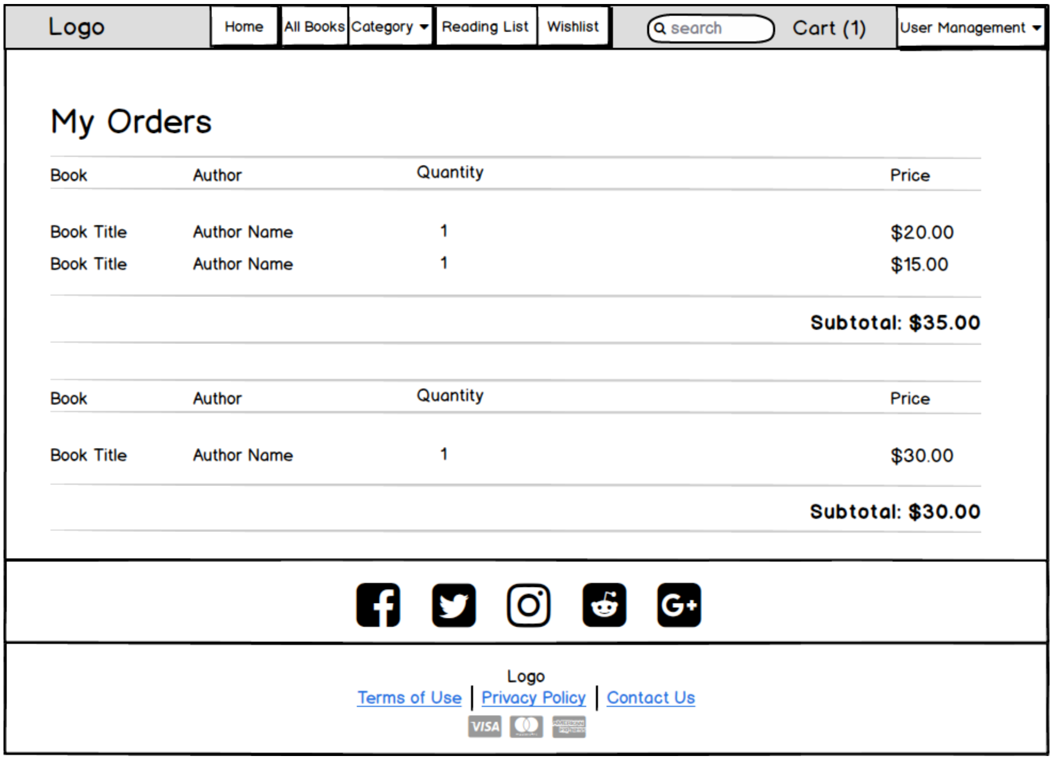
4.5.9 Reading List



Your Reading List

Cover image	Book ⇕	Author ⇕	
	Book Title	Author Name	Remove Book
	Book Title	Author Name	Remove Book

4.5.10 User Orders



My Orders

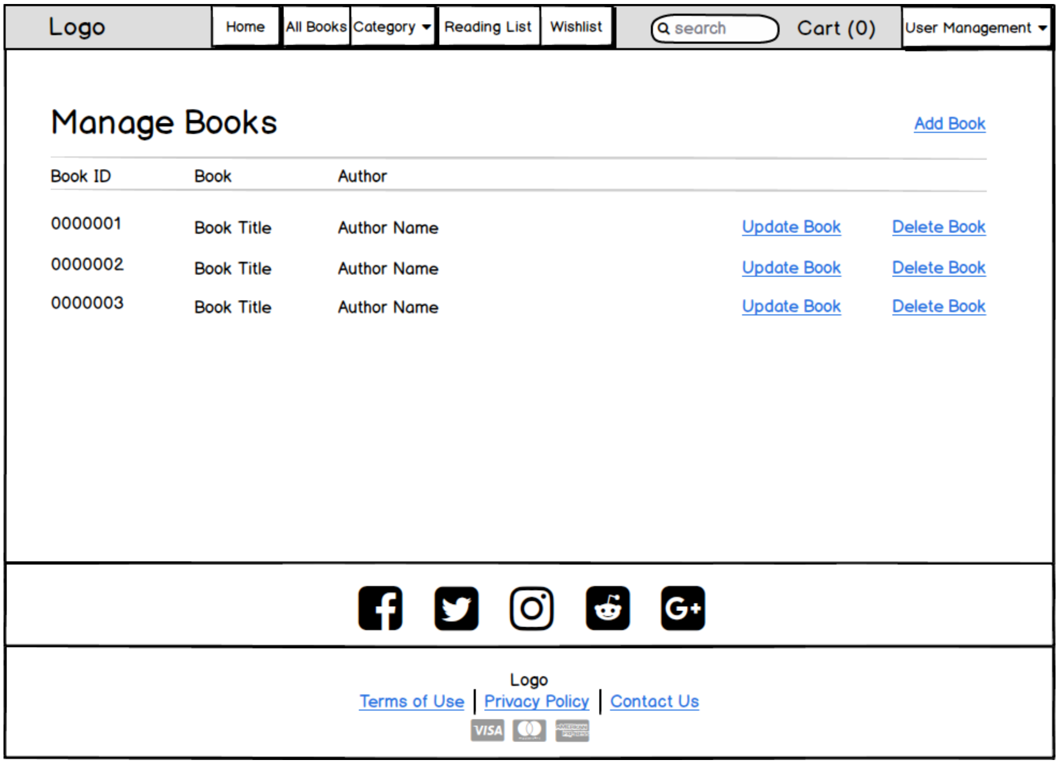
Book	Author	Quantity	Price
Book Title	Author Name	1	$20.00
Book Title	Author Name	1	$15.00

Subtotal: $35.00

Book	Author	Quantity	Price
Book Title	Author Name	1	$30.00

Subtotal: $30.00

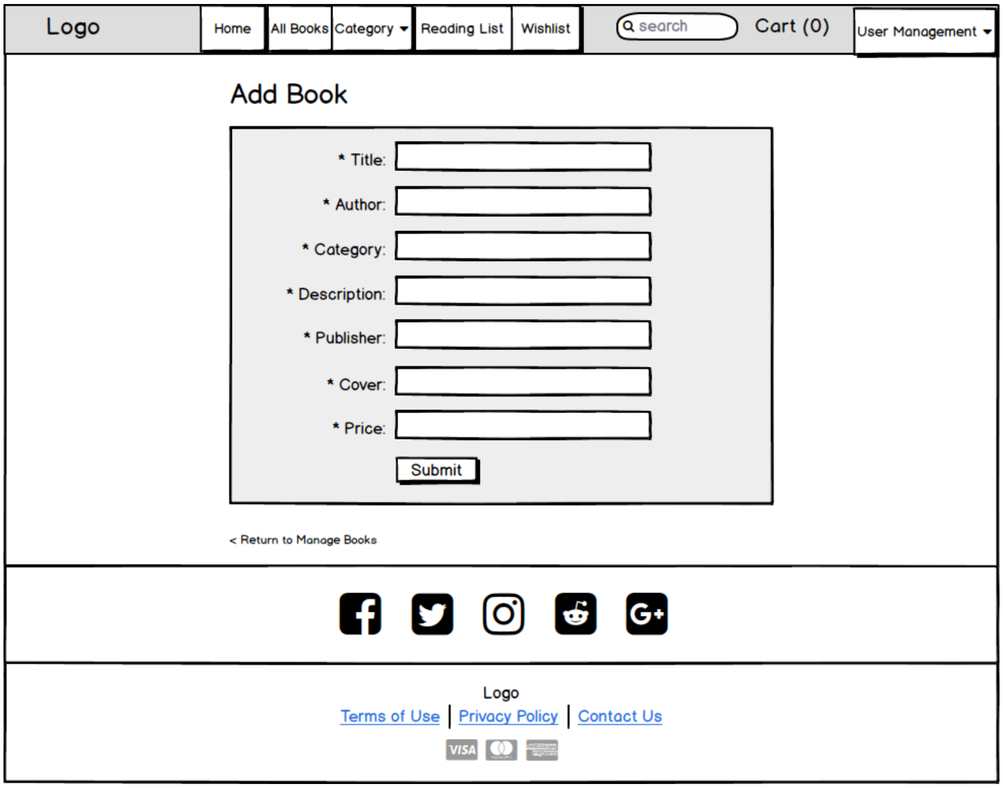
4.5.11 Manage Books



Manage Books

Book ID	Book	Author		
0000001	Book Title	Author Name	Update Book	Delete Book
0000002	Book Title	Author Name	Update Book	Delete Book
0000003	Book Title	Author Name	Update Book	Delete Book

4.5.12 Add Book



4.6 Performance

There are many benefits of using Node.js. While most of its users mention the advantages related to the development process, such as increased developer productivity, improved developer satisfaction, and reduced development cost. Half of the respondents of Node.js 2017 User Survey noticed improved application performance in comparison to other solutions. (Ciszewski, 2018) Some of the reasons why Node.js is technology to develop this e-commerce web application are:

- Node.js is good at multitasking. It can process multiple tasks concurrently in one thread, instead of queueing them,
- Node.js works on the fastest V8 Javascript engine that is used in Google Chrome. Node.js is developed by Google which increases the chances that it will be well supported in the future,
- the technology stack enables the application to handle much more requests at the same time than other another solution.

Bibliography

Arneson, E., 2014. *CenturyLink.* [Online]
Available at: https://www.ctl.io/developers/blog/post/build-user-authentication-with-node-js-express-passport-and-mongodb
[Accessed 23 October 2018].

Beal, V., 2018. *Webopedia.* [Online]
Available at: https://www.webopedia.com/TERM/S/security_architecture.html
[Accessed 23 October 2018].

Ciszewski, B., 2018. *netguru.* [Online]
Available at: https://www.netguru.co/blog/nodejs-performance-web-application-benefit
[Accessed 23 October 2018].

Eeles, P., 2006. *IBM.* [Online]
Available at: https://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html#notes
[Accessed 22 October 2018].

Mithun, S. & D'mello, B. J., 2017. *Web Development with MongoDB and Node.* Third Edition ed. s.l.: Packt Publishing.

Shah, S., 2012. *Jensen Consulting.* [Online]
Available at: https://www.jensen-consulting.co.uk/2012/05/16/writing-a-product-design-specification/
[Accessed 21 October 2018].

Stringfellow, A., 2017. *Stackify.* [Online]
Available at: https://stackify.com/web-application-architecture/
[Accessed 21 October 2018].

Svitla, 2018. *Svitla.* [Online]
Available at: https://svitla.com/blog/web-application-architecture
[Accessed 21 October 2018].

Terkaly, B., 2014. *Microsoft | Developer.* [Online]
Available at: https://blogs.msdn.microsoft.com/brunoterkaly/2014/02/02/installing-express-and-how-to-use-node-js-packages-with-visual-studio/
[Accessed 22 October 2018].

Vmoksha, 2017. *Vmoksha.* [Online]
Available at: https://vmokshagroup.com/blog/building-restful-apis-using-node-js-express-js-and-ms-sql-server/
[Accessed 22 October 2018].

Appendix A: References

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
Requirements Specification Document v.2	Outlines all requirements of e-commerce web application.	

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
API	In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software and applications.
body-parser	Allow the application to manipulate data in the application (create, delete, update).
Cloud9	Cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser.
Heroku	Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.
HTTP	Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML.
IDE	An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
JSON	JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

mLab	mLab is a fully managed cloud database service that hosts MongoDB databases.
MongoDB	MongoDB is a free and open-source cross-platform document-oriented database program.
mongoose	Elegant MongoDB object modelling for Node.js. Mongoose provides a straight-forward, schema-based solution to model your application data.
Node.js	Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.
npm	npm is the package manager for JavaScript and the world's largest software registry.
Passport.js	Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped into any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.
REST	Acronym for REpresentational State Transfer. REST is an architectural style that defines a set of constraints to be used for creating web services.
Stripe	Online payment processing for internet businesses.
Web application	A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.

Appendix C: Figures