

Higher Diploma in Science in Web Technologies

Requirements Specification

The Book Boutique

Student Name: Antonijo Galic

Student Number: 17157757

Email address: x17157757@student.ncirl.ie

01. 12. 2018.

Requirements Specification (RS)

Document Control

Revision History

Date	Version	Scope of Activity	Prepared	Reviewed	Approved
19/10/2018	1	Create	AB		
01/12/2018	2	Update			

Distribution List

Name	Title	Version
Liam McCabe	Lecturer	

Related Documents

Title	Comments

Table of Contents

1	Introduction	1
1.1	Purpose	1
1.2	Project Scope	1
1.3	Definitions, Acronyms, and Abbreviations	3
2	User Requirements Definition	4
3	Requirements Specification	4
3.1	Functional requirements	4
3.1.1	Use Case Diagram	5
3.1.2	Requirement 1: Browse Books	6
3.1.3	Requirement 2: Log In/Register	8
3.1.4	Requirement 3: Manage Cart	10
3.1.5	Requirement 4: Checkout	12
3.1.6	Requirement 5: Manage Wishlist	14
3.1.7	Requirement 6: Manage Reading List	16
3.1.8	Requirement 7: View Orders	18
3.1.9	Requirement 8: Add Book	20
3.1.10	Requirement 9: Update Book	22
3.1.11	Requirement 10: Delete Book	24
3.2	Non-Functional Requirements	26
3.2.1	Performance/Response time requirement	26
3.2.2	Availability requirement	26
3.2.3	Recover requirement	27
3.2.4	Security requirement	27
3.2.5	Reliability requirement	27
3.2.6	Maintainability requirement	28
3.2.7	Portability requirement	28
3.2.8	Extendibility requirement	28
4	Interface requirements	29
4.1	GUI	29
4.1.1	Home Page	30

Requirements Specification | The Book Boutique

4.1.2	All Books	31
4.1.3	Book Details	32
4.1.4	Register	33
4.1.5	Log In	33
4.1.6	Cart	34
4.1.7	Checkout	34
4.1.8	Wishlist	35
4.1.9	Reading List	35
4.1.10	User Orders	36
4.1.11	Manage Books	36
4.1.12	Add Book	37
4.2	Application Programming Interfaces (API)	37
5	System Architecture	38
6	System Evolution	39
	Bibliography	40

1 Introduction

1.1 Purpose

The purpose of this document is to show the requirements for developing a web application to function as an online e-commerce web application called The Book Boutique.

An online bookstore system is the main function of the web application where the user can check the book information, can register/login to buy the book, can add a book to wishlist or can add a book on the reading list.

The users of a web application are firstly the people who like to read books or the ones who like to buy books like a gift for someone.

1.2 Project Scope

The scope of the project is to develop a fluid and fast e-commerce web application. The completed web application will be user-friendly and it will be designed to be fully responsive so it can automatically adapt to suit all screen resolutions and devices (desktop, laptop, tablet and phone).

Project objectives

The web application has to accomplish the general objective of a web application, objectives on user side and objectives on the admin side. The general objective of the web application is to introduce the site visitor to the site with an overview of the products where application displaying a sample product visually with a view to keep user interest and motivate the user to buy a product(s). Objectives on user side are based on functionality that allows the user to register/login to interact with the application so the user can:

- buy a product – adding the product to cart and complete payment process after entering card details,

Requirements Specification | The Book Boutique

- leave a review – a user can leave a review for a specific book and give a rating to a book,
- add a book to wishlist – after adding book(s) to wishlist user will have a reminder about the book(s) for purchasing in the future,
- add a book on the reading list – by adding a book on the reading list user has the ability to track books which user already read,
- view orders – user can check own history orders.

Admin side objective is to give admin ability to add, edit and delete books which means that admin has full CRUD (Create, Read, Update, Delete) functionality over the application.

Project motivation

First, the motivation for this project is to create a functional e-commerce web application in order to successfully finish final project module. Second, the idea for this project and motivation itself is because my wife likes to read books, likes to buy books and likes to track which books she has already read.

Success conditions

Main success condition of this project is when a user makes a payment for a book(s). Other success conditions of a web application are when user leaves a review or when the user starts to use an application like a tracker for books which user has already read.

Project limitations

This project has some limitations. One of the limitations is that user needs to create an account and be logged in into the web application to have full access. Full access to the web application, in this case, implies adding books(s) to cart, buying book(s), leaving a review, adding a book to wishlist and on the reading list. The other limitation of this project is the time needed to develop a fully functional

application, limited experience of developer and money needed for faster hosting of web application and database hosting.

1.3 Definitions, Acronyms, and Abbreviations

<i>Term</i>	<i>Definition</i>
<i>Admin</i>	One who manages and maintains computer systems and software
<i>Alternate flow</i>	Any path(flow) other than the main path
<i>API</i>	In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software and applications
<i>Bcrypt</i>	Node module which helps to encrypt passwords
<i>Cloud9</i>	Cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser
<i>CRUD</i>	Acronym for Create, Read, Update, Delete
<i>Git</i>	Version-control system for tracking changes in computer files and coordinating work on those files among multiple people
<i>GitHub</i>	Web-based hosting service for version control using Git
<i>Main flow</i>	The most common, guided, "safe," easiest, developer intended path(flow)
<i>Passport.js</i>	Node module which implements authentication
<i>REST</i>	Acronym for REpresentational State Transfer
<i>Stripe</i>	Online payment processing for internet businesses
<i>Web application</i>	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

2 User Requirements Definition

To access and interact with e-commerce web application user must have some essential requirements:

- Device connected to the Internet access – to interact with application user will need a device (desktop, laptop, tablet or mobile phone) connected to the internet access,
- Email account – to register/login into web application user needs to have an email account,
- Debit or credit card – to purchase the book(s) the user must have valid debit or credit card.

3 Requirements Specification

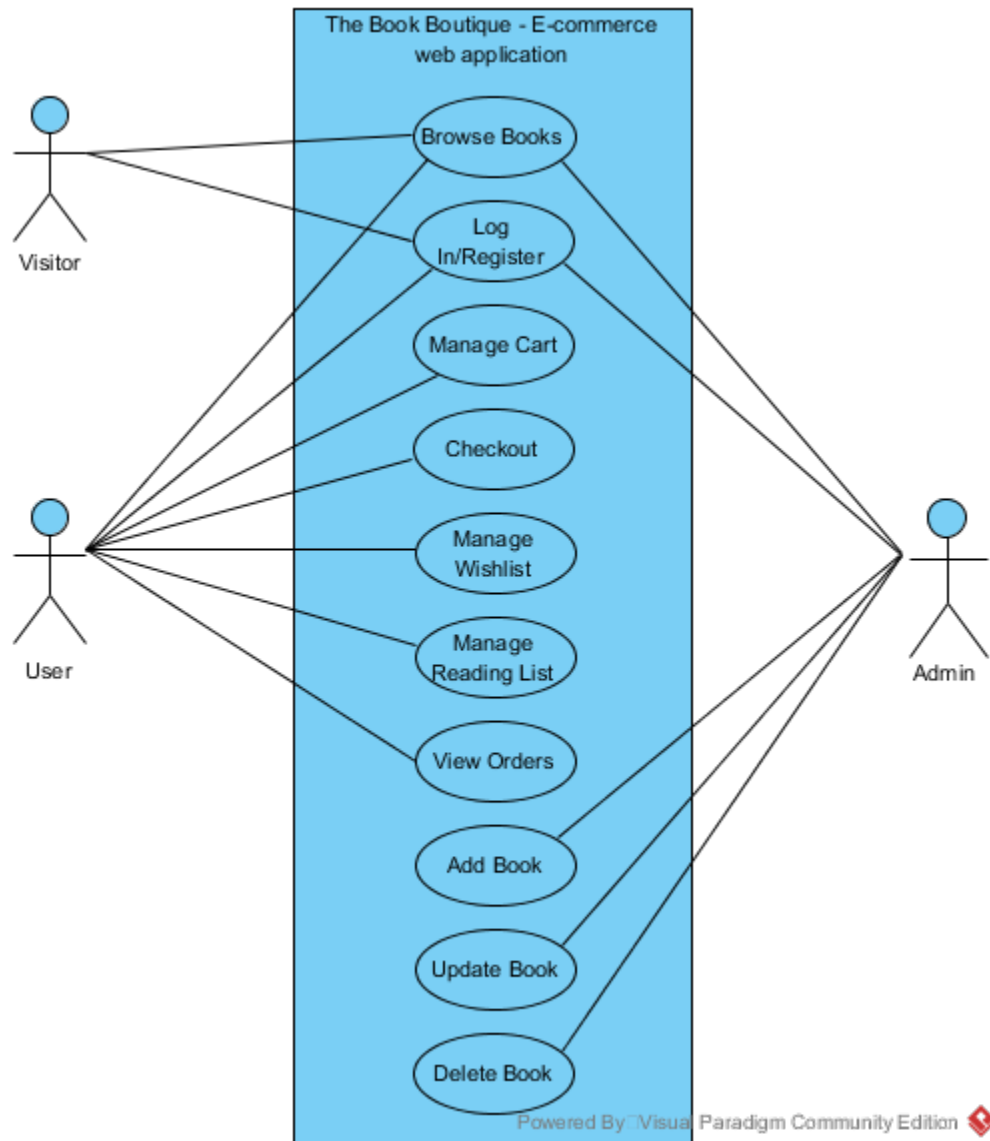
This section outlines all functional and non-functional requirements of e-commerce web application.

System requirement specification or SRS frameworks software development, it documents every operation and dictates how software should behave, it can be as detailed as what a button should do and should be as complete and correct as possible. The purpose of a specification document is to describe the behaviour as well as the different functionalities of an application or software in a specific environment. (Osetskyi, 2018)

3.1 Functional requirements

This section usually consists of a hierarchical organization of requirements, with the business/functional requirements at the highest level and the detailed system requirements listed as their child items. (Inflectra, 2018) This section also contains a use case diagram that illustrates how the actors of web application interact with the system and describe all use cases necessary for functional work of e-commerce web application.

3.1.1 Use Case Diagram

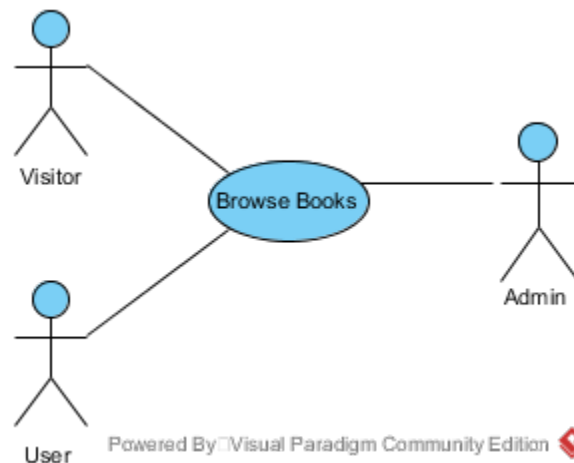


3.1.2 Requirement 1: Browse Books

3.1.2.1 Description & Priority

The user navigates to The Book Boutique e-commerce web application to browse books. Browsing books is the first step in using e-commerce web application.

3.1.2.2 Use Case



Scope

The scope of this use case is for a user to browse books inside e-commerce web application.

Description

This use case describes how user is browsing books.

Flow Description

Precondition

The user has internet access and wants to browse (potentially buy) books.

Activation

This use case starts when the user opens The Book Boutique e-commerce web application.

Main flow

1. The user can browse books using the navigation bar. [A1 Search books]
2. The e-commerce web application displayed books based on user selection.
3. The user selects a book to view details.
4. The e-commerce web application presents the details about the book with book price and options to add a book to cart, add a book to wishlist or add a book on the reading list.
5. The user selects one of the options. [A2 No interest in the specific book]
6. The user goes back to browse more books.

Alternate flow

(A1 Search books)

1. The user is using the search bar to browse books.
2. The e-commerce web application display search results depend on search keyword.
3. The use case continues at position 3 of the main flow.

(A2 No interest in the specific book)

1. The user doesn't have an interest in a specific book.
2. The use case continues at position 6 of the main flow.

Termination

The user proceeds to the shopping cart or left the web application.

Postcondition

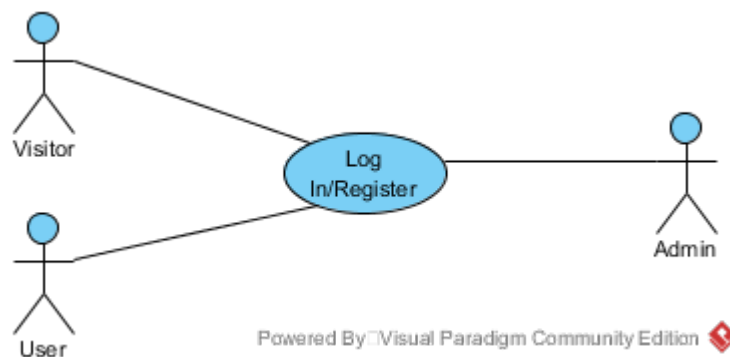
The e-commerce web application goes to wait state.

3.1.3 Requirement 2: Log In/Register

3.1.3.1 Description & Priority

The use case describes how user is entering into own account on the web application in order to be verified to interact with the application. This requirement is necessary in order to buy a book, add a book to wishlist, leave a review or add a book on the reading list.

3.1.3.2 Use Case



Scope

The scope of this use case is to enable the user to enter into own account on the web application.

Description

This use case describes steps to log in into a web application.

Flow Description

Precondition

The e-commerce web application is ready to be activated.

Activation

The user navigates to the web application.

Main flow

1. The user navigates to the navigation bar and clicks Log In button.
[A1 Register account]
2. The e-commerce web application returns Log In page with log in form.
3. The user enters the log in details.
4. The user clicks on the Submit button.
5. The e-commerce web application verifies the log in details. [E1 Incorrect log in details]
6. The e-commerce web application allows access to the Profile page of the web application.

Alternate flow

(A1 Register account)

1. The new user navigates to the navigation bar and clicks Register button.
2. The e-commerce web application returns Register page with the registration form.
3. The user enters personal details to the registration form.
4. The user clicks on the Submit button.
5. The e-commerce web application stores the information.
6. The use case continues at position 6 of the main flow.

Exceptional flow

(E1 Incorrect log in details)

1. The e-commerce web application returns message with incorrect log in details.
2. The e-commerce web application asks user to re-enter log in detail(s).
3. The user re-enters log in detail(s).
4. The use case continues at position 4 of the main flow.

Termination

This use case terminates when web application redirects the user to the web application like authenticate user.

Postcondition

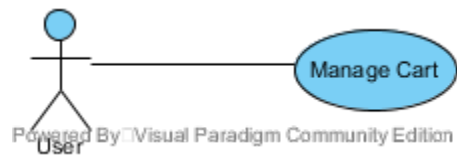
The e-commerce web application goes to wait state.

3.1.4 Requirement 3: Manage Cart

3.1.4.1 Description & Priority

The use case describes how a user can manage books in the shopping cart. The user can choose a quantity for a specific book, can remove the book from the cart and can see the total price per book and the total amount of cart.

3.1.4.2 Use Case



Scope

The scope of this use case is to enable the user to manage own shopping cart.

Description

This use case describes steps to manage books in the shopping cart.

Flow Description

Precondition

The user is logged in into e-commerce web application and the user has added book(s) to the shopping cart.

Activation

The user navigates to Cart button on navigation bar.

Main flow

1. The user navigates to the navigation bar and clicks Cart button.
2. Cart page is displayed to the user.
3. The user checks shopping cart one more time before checkout.
4. The user is satisfied with the book(s) in the shopping cart. [A1 Remove book] [A2 Decrease quantity]
5. The user clicks on the Checkout button.

Alternate flow

(A1 Remove book)

1. The user is not satisfied with the book in the shopping cart.
2. The user clicks Remove button on a specific book in the shopping cart.
3. The book is not anymore displayed in shopping cart.
4. The use case continues at position 3 of the main flow.

(A2 Decrease quantity)

1. The user is not satisfied with the quantity for a specific book.
2. The user decrease quantity for a specific book.
3. The quantity of book is decreased in the shopping cart list.
4. The use case continues at position 3 of the main flow.

Termination

This use case terminates when the user clicks the Checkout button.

Postcondition

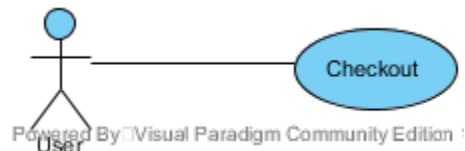
The e-commerce web application goes to wait state.

3.1.5 Requirement 4: Checkout

3.1.5.1 Description & Priority

The use case describes how user is making payment in e-commerce web application. The user needs to fill up the card detail form with valid information in order to finish the transaction.

3.1.5.2 Use Case



Scope

The scope of this use case is to enable the user to make a transaction.

Description

This use case describes steps from Checkout button to the finished transaction.

Flow Description

Precondition

The user is logged in and the cart is populated with the book(s) on user satisfaction.

Activation

The user clicks on the Checkout button in the shopping cart.

Main flow

1. The user clicks on the Checkout button in the shopping cart.
2. The e-commerce web application displays the Checkout page with a form. [A1 Back to shopping cart]
3. The user enters personal details including card details.

4. The user clicks Confirm Payment button.
5. The e-commerce web application together with Stripe API verifies inputs entered from the user. [E1 Incorrect card details]
6. The e-commerce web application redirects the user to the Home page.
7. The message “You have successfully bought the product!” is displayed to the user.

Alternate flow

(A1 Back to the shopping cart)

1. The user decides to change the shopping cart detail(s).
2. The user clicks on the link Return to Cart.
3. The e-commerce web application displays Cart page.
4. The user changes the shopping cart detail(s).
5. The use case continues at position 1 of the main flow.

Exceptional flow

(E1 Incorrect card details)

1. The web application returns a message with incorrect card details.
2. The web application asks from user to re-enter card detail(s).
3. The user re-enters card detail(s).
4. The use case continues at position 4 of the main flow.

Termination

This use case terminates when the user receives a message “You have successfully bought the product!”.

Postcondition

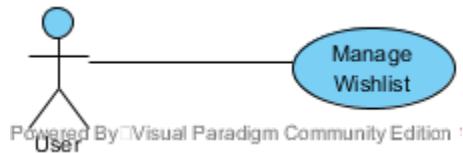
The e-commerce web application goes to wait state.

3.1.6 Requirement 5: Manage Wishlist

3.1.6.1 Description & Priority

The use case describes how a user can manage a book(s) added to the wishlist. The user can add a book to the cart or remove book(s) from the wishlist.

3.1.6.2 Use Case



Scope

The scope of this use case is to enable the user to manage own wishlist.

Description

This use case describes steps to manage books in the wishlist.

Flow Description

Precondition

The user is logged in into e-commerce web application and the user is adding book(s) to wishlist.

Activation

The user navigates to Wishlist button on navigation bar.

Main flow

1. The user navigates to the navigation bar and clicks Wishlist button.
2. Wishlist page is displayed to the user.
3. The user checks books in the wishlist. [A1 Add book to cart]
4. The user is satisfied with the book(s) in the wishlist. [A2 Remove book]
5. The user leaves the Wishlist page.

Alternate flow

(A1 Add book to cart)

1. The user decides to buy a book from wishlist.
2. The user clicks Add to cart button for a specific book.
3. The user bought a specific book.
4. The use case continues at position 3 of the main flow.

(A2 Remove book)

5. The user is not satisfied with the book in the wishlist.
6. The user clicks Remove button on a specific book in the wishlist.
7. The book is not anymore displayed in the wishlist.
8. The use case continues at position 3 of the main flow.

Termination

This use case terminates when the user left wishlist page.

Postcondition

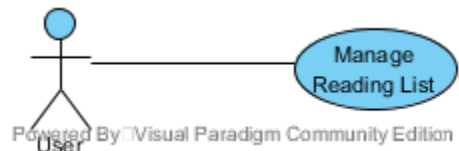
The e-commerce web application goes to wait state.

3.1.7 Requirement 6: Manage Reading List

3.1.7.1 Description & Priority

The use case describes how a user can manage a book(s) added to the reading list. Reading list represents a list of books which user has already read. The user can remove book(s) from reading list or sort books per name or author.

3.1.7.2 Use Case



Scope

The scope of this use case is to enable the user to manage own reading list.

Description

This use case describes steps to manage books in the reading list.

Flow Description

Precondition

The user is logged in into e-commerce web application and the user is adding book(s) to the reading list.

Activation

The user navigates to Reading list button on navigation bar.

Main flow

1. The user navigates to the navigation bar and clicks Reading list button.
2. Reading list page is displayed to the user.
3. The user checks books on the reading list. [A1 Remove book]

Requirements Specification | The Book Boutique

4. The user sorts books per author. [A2 Sort books per name]
5. The e-commerce web application displays books sorted per author.
6. The user checks one more time books in the reading list.
7. The user leaves Reading list page.

Alternate flow

(A1 Remove book)

1. The user has accidentally added a book to the reading list.
2. The user clicks Remove button on a specific book on the reading list.
3. The book is not anymore displayed on the reading list.
4. The use case continues at position 6 of the main flow.

(A2 Sort books per name)

1. The user sort books per the name of the book.
2. The e-commerce web application displays books sorted per the name of the book.
3. The use case continues at position 6 of the main flow.

Termination

This use case terminates when the user left Reading list page.

Postcondition

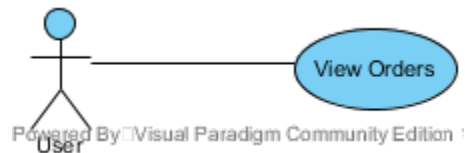
The e-commerce web application goes to wait state.

3.1.8 Requirement 7: View Orders

3.1.8.1 Description & Priority

The use case describes how a user can view orders on the own orders page. On the order list, the user can see details for a specific order. Order details are the name of the book, author of the book, quantity, price per book and total price.

3.1.8.2 Use Case



Scope

The scope of this use case is to enable the user to view order list on own orders page.

Description

This use case describes steps for the user to check own orders.

Flow Description

Precondition

The user already made the transaction(s).

Activation

The user activates e-commerce web application.

Main flow

1. The user navigates to the navigation bar and clicks the button to navigate to the My orders page. [A1 Log In requested]
2. The e-commerce web application allows access to the My orders page of the web application.
3. The e-commerce web application displays orders made by the user.

4. The user returns to the Homepage of e-commerce web application.

Alternate flow

(A1 Log In requested)

1. The button to navigate to the user profile is not displayed because the user is not logged in into e-commerce web application.
2. The user navigates to the navigation bar and clicks Log In button.
3. The e-commerce web application returns to Log In page with log in form.
4. The user enters the log in details.
5. The user clicks on the Submit button.
6. The e-commerce web application verifies the log in details. [E1 Incorrect log in details]
7. The use case continues at position 2 of the main flow.

Exceptional flow

(E1 Incorrect log in details)

1. The e-commerce web application returns message with incorrect log in details.
2. The e-commerce web application asks from user to re-enter log in detail(s).
3. The user re-enters log in detail(s).
4. The use case continues at position 5 of the A1 alternate flow.

Termination

This use case terminates when the user left My orders page.

Postcondition

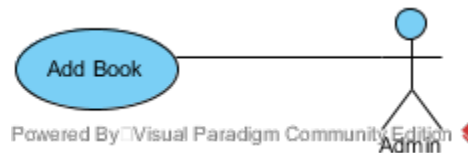
The e-commerce web application goes to wait state.

3.1.9 Requirement 8: Add Book

3.1.9.1 Description & Priority

The use case describes how a user(admin) can add a book(s) to the e-commerce web application. This requirement is necessary in order to populate the database and to display book(s) in the e-commerce web application.

3.1.9.2 Use Case



Scope

The scope of this use case is to enable a user(admin) to create a book(s) which will be displayed in the e-commerce web application.

Description

This use case describes steps for user(admin) to add/create a book(s) to display in the e-commerce web application.

Flow Description

Precondition

The user is logged in into e-commerce web application like admin.

Activation

The user(admin) clicks on the Manage button.

Main flow

1. The user(admin) clicks on Manage Books button to navigate to Manage Books page.
2. The e-commerce web application displays Manage Books page.
3. The user(admin) clicks on Add book button.

Requirements Specification | The Book Boutique

4. The e-commerce web application returns Add book page with a form for adding/creating a book to the e-commerce web application.
5. The user(admin) enters the book details into the form.
6. The user clicks on the Submit button.
7. The e-commerce web application verifies the book details. [E1 Incorrect book details]
8. The e-commerce web application redirects user(admin) to the All books page.

Exceptional flow

(E1 Incorrect book details)

1. The e-commerce web application returns a message with incorrect book detail(s).
2. The e-commerce web application asks user(admin) to re-enter book detail(s).
3. The user(admin) re-enters book detail(s).
4. The use case continues at position 6 of the main flow.

Termination

This use case terminates when the user(admin) returns to the All books page of e-commerce web application.

Postcondition

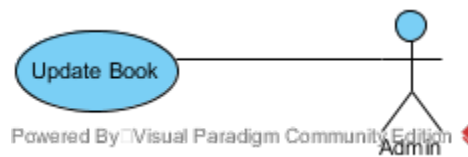
The e-commerce web application goes to wait state.

3.1.10 Requirement 9: Update Book

3.1.10.1 Description & Priority

The use case describes how a user(admin) can update(edit) book(s) in the e-commerce web application. This requirement is necessary in order to update pre-populated book detail(s) in order to display book(s) in the e-commerce web application.

3.1.10.2 Use Case



Scope

The scope of this use case is to enable a user(admin) to update pre-populated book detail(s).

Description

This use case describes steps for user(admin) to update book detail(s).

Flow Description

Precondition

The user is logged in into e-commerce web application like admin.

Activation

The user(admin) clicks on the Manage Books button.

Main flow

1. The user(admin) clicks on Manage Books button to navigate to Manage Books page.
2. The e-commerce web application displays Manage Books page.

3. The user(admin) clicks on the Update Book button for a specific book.
4. The e-commerce web application returns Edit page with a pre-populated form for editing book details.
5. The user(admin) changes the book detail(s).
6. The user clicks on the Submit button.
7. The e-commerce web application verifies the book details. [E1 Incorrect book details]
8. The e-commerce web application redirects user(admin) to the Manage Books page.

Exceptional flow

(E1 Incorrect book details)

1. The e-commerce web application returns a message with incorrect book detail(s).
2. The e-commerce web application asks from the user(admin) to re-enter book detail(s).
3. The user(admin) re-enters book detail(s).
4. The use case continues at position 6 of the main flow.

Termination

This use case terminates when the user(admin) returns to the Manage Books page of e-commerce web application.

Postcondition

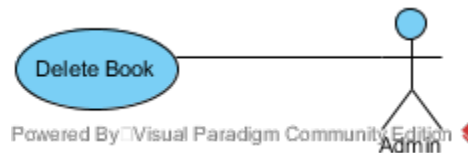
The e-commerce web application goes to wait state.

3.1.11 Requirement 10: Delete Book

3.1.11.1 Description & Priority

The use case describes how a user(admin) can delete the book(s) in the e-commerce web application. This requirement is necessary in order to delete the book(s) from the database and from the e-commerce web application.

3.1.11.2 Use Case



Scope

The scope of this use case is to enable a user(admin) to delete existing book(s).

Description

This use case describes steps for user(admin) to delete the book(s).

Flow Description

Precondition

The user is logged in into e-commerce web application like admin.

Activation

The user(admin) clicks on the Manage Books button.

Main flow

1. The user(admin) clicks on Manage Books button to navigate to Manage Books page.
2. The e-commerce web application displays Manage Books page.
3. The user(admin) clicks on the Delete Book button for a specific book.
4. The e-commerce web application refreshes the Manage Books page.

5. The e-commerce web application is not displaying anymore deleted book.

Termination

This use case terminates when the user(admin) returns to the Manage Books page of e-commerce web application and web application is not displaying anymore deleted book.

Postcondition

The e-commerce web application goes to wait state.

3.2 Non-Functional Requirements

Non-functional requirements essentially define the quality of platform. A functional requirement will define its behaviour, but the non-functional requirements will define how well it will exhibit that behaviour. A well-defined set of non-functional requirements will have the following benefits: (Moffat, 2018)

- Better code quality,
- Better security,
- Better user experience,
- Less bugs,
- Better speed and performance,
- Better uptime.

3.2.1 Performance/Response time requirement

Accurately measuring the performance characteristics of web applications is an important aspect of making web applications faster. (Mann & Wang, 2013) However, performance and response time of this e-commerce web application should depend on the technology stack used for building a web application and quality of connection to the internet.

3.2.2 Availability requirement

High availability is the ability of the cluster to effectively avoid downtime. High availability is usually defined in terms of uptime, such as 99.999%, 24/7/365 (always), or business hours during business days. To design for high availability, all system components must be made redundant or recoverable enough so that no single component can fail and bring the entire environment to a stop in a way that violates the high availability requirements. (Jaspersoft, 2018) Based on this information e-commerce web application The Book Boutique should have high availability of uptime. Unless the web application is non-operational, the web application should present a notification to the user that the web application is temporarily unavailable.

The payment system should have high availability which is controlled by Stripe payment platform.

Availability of this web application depends also on user(s) availability to connect on the internet via personal computer or mobile devices.

3.2.3 Recover requirement

The e-commerce web application will have a daily backup system so web application can continue to function and recover in the case of a system failure.

Code of the web application will be pushed to GitHub in case that something goes wrong with Cloud9 service.

3.2.4 Security requirement

This section covers the way that web application is safeguarded against faults from internal and external sources. Security requirement of this web application including that:

- sensible data will be protected in the way that e-commerce web application uses the bcrypt module for encryption and passport.js for authentication,
- to add book(s) to cart, buy the book(s) or leave a review, the user needs to register/login to the web application,
- passwords will be hidden at the point of entry,
- validation and charging credit cards are controlled by Stripe payment platform.

3.2.5 Reliability requirement

Reliability requirement of this e-commerce web application depends on user connection to the internet and hosting of the web application. In a normal working environment, it is expected that failures of any kind will be kept to an absolute minimum.

3.2.6 Maintainability requirement

This section defines that web application will be updated or fixed firstly out of existing e-commerce web application. If web application is supposed to be updated that will be done at an off-peak time of the week. In case that e-commerce web application requires to be fixed because of some kind of failure or bugs appearance it will be done as soon as possible.

3.2.7 Portability requirement

E-commerce web application will be designed and developed to be fully responsive and it will automatically adapt to suit all screen resolutions and devices (desktop, laptop, tablet and phone).

3.2.8 Extendibility requirement

This project defines creating a web application to function as an online e-commerce web application for selling books. There are possibilities that web application can be extended on e-commerce application similar to Amazon or existing web application can keep the focus on books and sell movies based on books.

4 Interface requirements

With the explosion of so much software that people use in their daily personal and professional lives, the user interface (UI) is absolutely critical to the successful use of that software. (Koelsch, 2016) This section outlines internal requirements (GUI – graphical user interface) and external requirements (API – application programming interface).

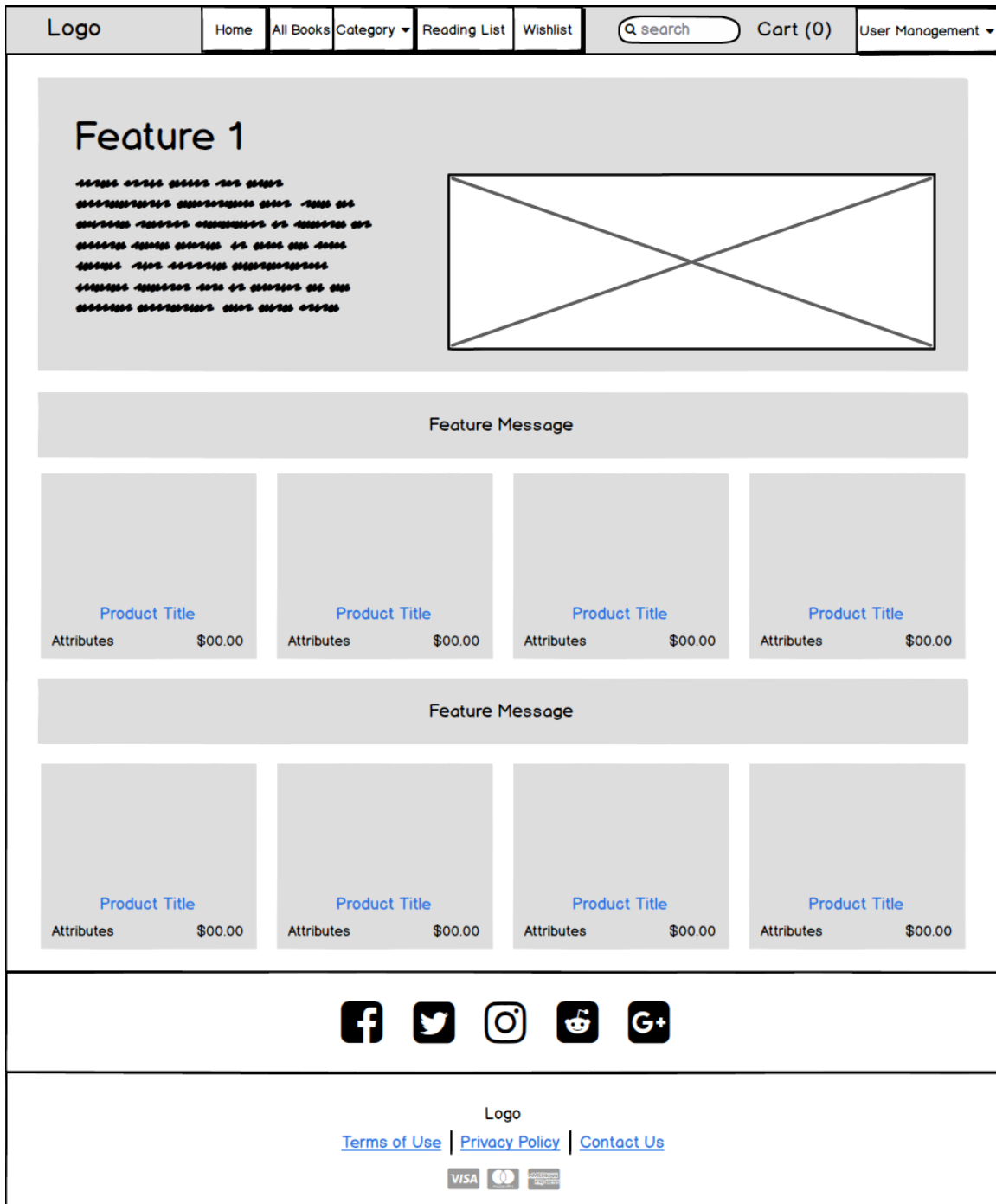
4.1 GUI

A graphical user interface (GUI) is an interface through which a user interacts with electronic devices such as computers, hand-held devices and other appliances. This interface uses icons, menus and other visual indicator (graphics) representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text. GUI representations are manipulated by a pointing device such as a mouse, trackball, stylus, or a finger on a touch screen. (Techopedia, 2018)

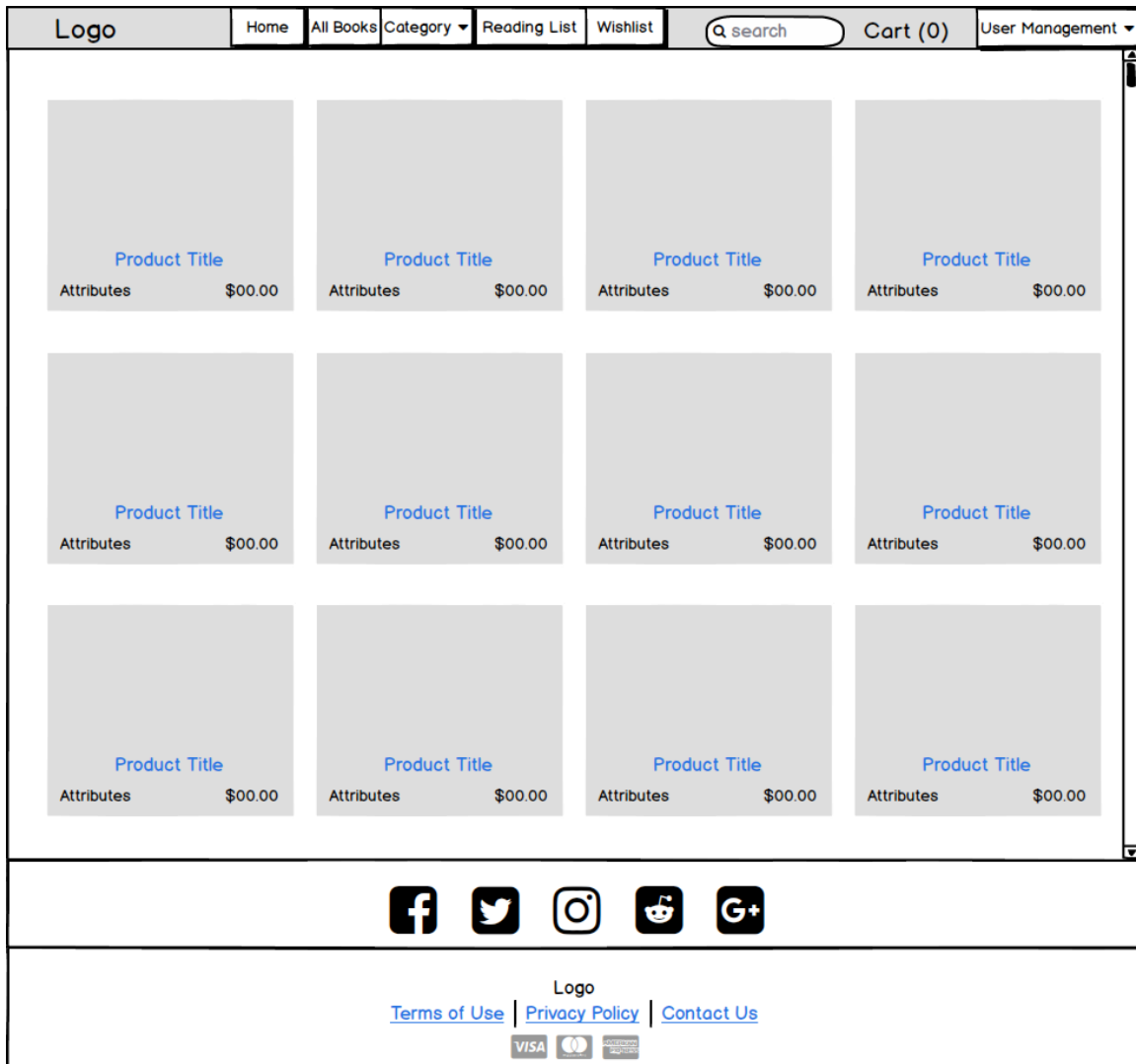
The e-commerce web application The Book Boutique in short has next GUI. On the top of the web application a menu with buttons necessary for user interaction with the web application will be displayed. Homepage will be displaying books on offer and giving a short overview of books in general. Book detail page gives a description of book, book reviews and displaying buttons for interaction with the user. Register and login page have input form and will required text input from the user. Cart page displays books added to cart with quantity, price and total price. On cart page is a button which will take user to Checkout page where the user needs to fill up form with card details. Wishlist and Reading List pages have list of books added from the user. Order history from the user is displayed on the user profile page.

The following wireframes illustrate how a graphical user interface for this e-commerce web application can be implemented.

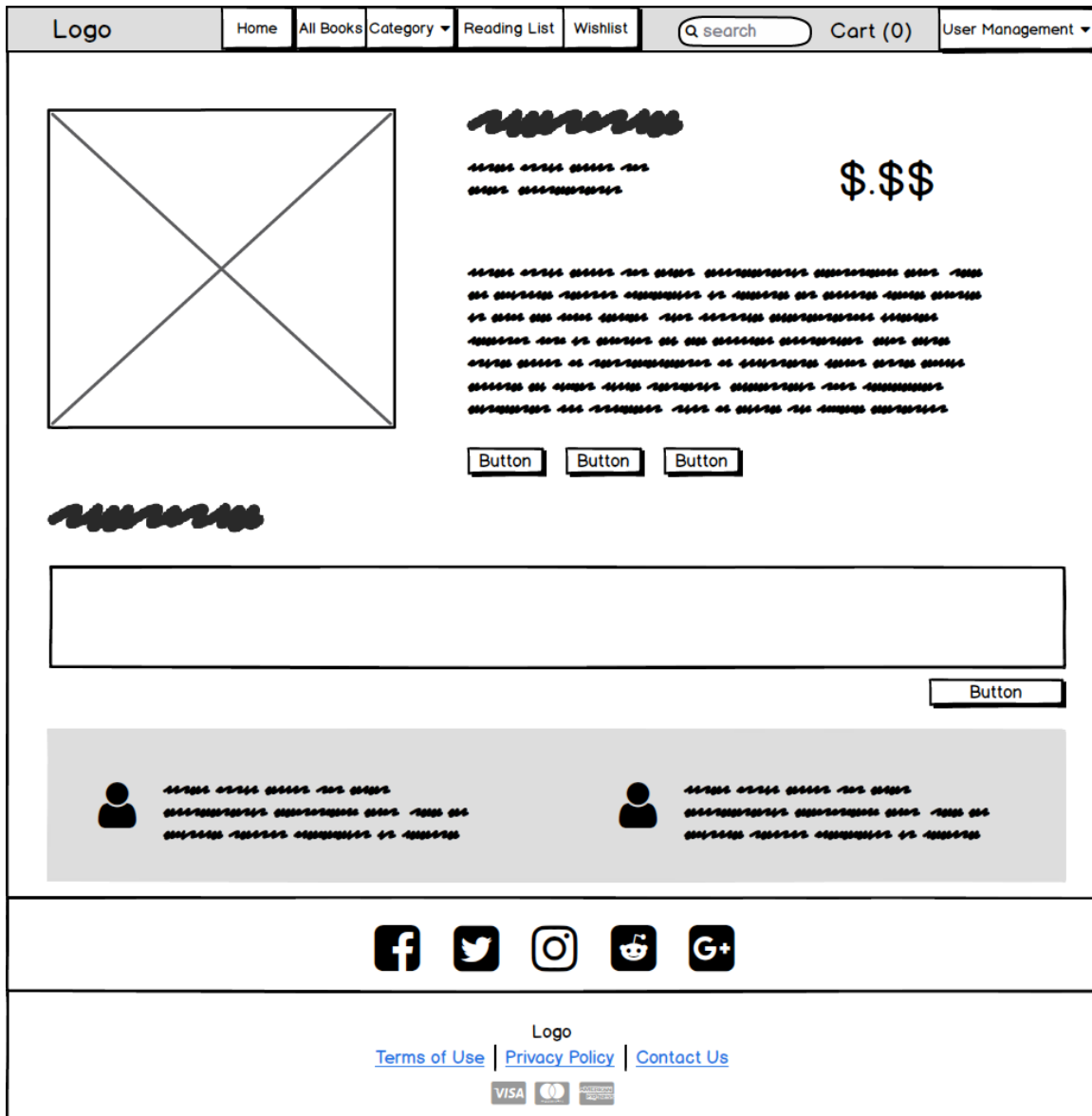
4.1.1 Home Page



4.1.2 All Books



4.1.3 Book Details



4.1.4 Register

Logo	Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (0)	User Management ▾
------	------	-----------	------------	--------------	----------	---------------------------------------	----------	-------------------

Register






* Username:

* Email:

* Password: ?




* Re-type

☐ I agree to the [Terms of Use](#) and [Privacy Policy](#).



Logo

[Terms of Use](#) | [Privacy Policy](#) | [Contact Us](#)






4.1.5 Log In

Logo	Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (0)	User Management ▾
------	------	-----------	------------	--------------	----------	---------------------------------------	----------	-------------------

Login




Email:

Password:














Logo




[Terms of Use](#) | [Privacy Policy](#) | [Contact Us](#)

4.1.6 Cart

Logo		Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (1)	User Management ▾												
<h3>Your Cart</h3> <table border="1"> <thead> <tr> <th>Item</th> <th>Price</th> <th>Quantity</th> </tr> </thead> <tbody> <tr> <td>  <div> Book Author Remove Item </div> </td> <td>\$20.00</td> <td>1 ▾</td> </tr> </tbody> </table>							Item	Price	Quantity	 <div> Book Author Remove Item </div>	\$20.00	1 ▾	<h4>Summary (1 item)</h4> <table border="1"> <tbody> <tr> <td>Subtotal</td> <td>\$20.00</td> </tr> <tr> <td>Shipping</td> <td>-</td> </tr> <tr> <td>Total</td> <td>\$20.00</td> </tr> </tbody> </table>			Subtotal	\$20.00	Shipping	-	Total	\$20.00
Item	Price	Quantity																			
 <div> Book Author Remove Item </div>	\$20.00	1 ▾																			
Subtotal	\$20.00																				
Shipping	-																				
Total	\$20.00																				
<div>Subtotal: \$20.00</div>							<div>Checkout</div>														
<div>      </div>																					
<div> Logo Terms of Use Privacy Policy Contact Us    </div>																					


4.1.7 Checkout






Logo		Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (1)	User Management ▾
<div>Customer Information</div> <div> <input type="text" value="First Name"/> <input type="text" value="Last Name"/> </div> <div> <input type="text" value="Address"/> </div>									
<div>Payment Method</div> <div>    </div> <div> <input type="text" value="Card number"/> </div> <div> <input type="text" value="Name on card"/> <input type="text" value="MM/YY"/> <input type="text" value="CVV"/> </div>									
<div>< Return to Cart</div>							<div>Submit</div>		
<div>      </div>									
<div> Logo Terms of Use Privacy Policy Contact Us    </div>									

4.1.8 Wishlist

Logo	Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (1)	User Management ▾
------	------	-----------	------------	--------------	----------	---------------------------------------	----------	-------------------




Your Wishlist

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

Logo



[Terms of Use](#) |
 [Privacy Policy](#) |
 [Contact Us](#)










4.1.9 Reading List

Logo	Home	All Books	Category ▾	Reading List	Wishlist	<input type="text" value="Q search"/>	Cart (1)	User Management ▾
------	------	-----------	------------	--------------	----------	---------------------------------------	----------	-------------------




Your Reading List

Cover image	Book ▴ ▾	Author ▴ ▾	
	Book Title	Author Name	Remove Book
	Book Title	Author Name	Remove Book















Logo









[Terms of Use](#) |
 [Privacy Policy](#) |
 [Contact Us](#)

4.1.10 User Orders

Logo	Home	All Books	Category ▾	Reading List	Wishlist	Q search	Cart (1)	User Management ▾																												
<h2>My Orders</h2> <table border="1"> <thead> <tr> <th>Book</th> <th>Author</th> <th>Quantity</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>Book Title</td> <td>Author Name</td> <td>1</td> <td>\$20.00</td> </tr> <tr> <td>Book Title</td> <td>Author Name</td> <td>1</td> <td>\$15.00</td> </tr> <tr> <td colspan="3"></td> <td>Subtotal: \$35.00</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Book</th> <th>Author</th> <th>Quantity</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>Book Title</td> <td>Author Name</td> <td>1</td> <td>\$30.00</td> </tr> <tr> <td colspan="3"></td> <td>Subtotal: \$30.00</td> </tr> </tbody> </table>									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
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																																	
<div>      </div>																																				
<div>    </div>																																				

4.1.11 Manage Books

Logo	Home	All Books	Category ▾	Reading List	Wishlist	Q search	Cart (0)	User Management ▾
<h2>Manage Books</h2> Add Book								
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				
<div>      </div>								
<div>    </div>								

4.1.12 Add Book

The screenshot shows the 'Add Book' form within a web application interface. The top navigation bar includes a 'Logo' button, links for 'Home', 'All Books', 'Category' (with a dropdown arrow), 'Reading List', and 'Wishlist'. It also features a search bar with a magnifying glass icon and the text 'Q search', a 'Cart (0)' indicator, and a 'User Management' dropdown menu. The main content area is titled 'Add Book' and contains a form with the following fields, each preceded by an asterisk (*): 'Title', 'Author', 'Category', 'Description', 'Publisher', 'Cover', and 'Price'. Each field is represented by a text input box. Below these fields is a 'Submit' button. A link '< Return to Manage Books' is located below the form. The footer section contains social media icons for Facebook, Twitter, Instagram, Reddit, and Google+. Below the icons are links for 'Terms of Use', 'Privacy Policy', and 'Contact Us', preceded by a 'Logo' text. At the very bottom, there are logos for 'VISA', 'MasterCard', and 'American Express'.

4.2 Application Programming Interfaces (API)

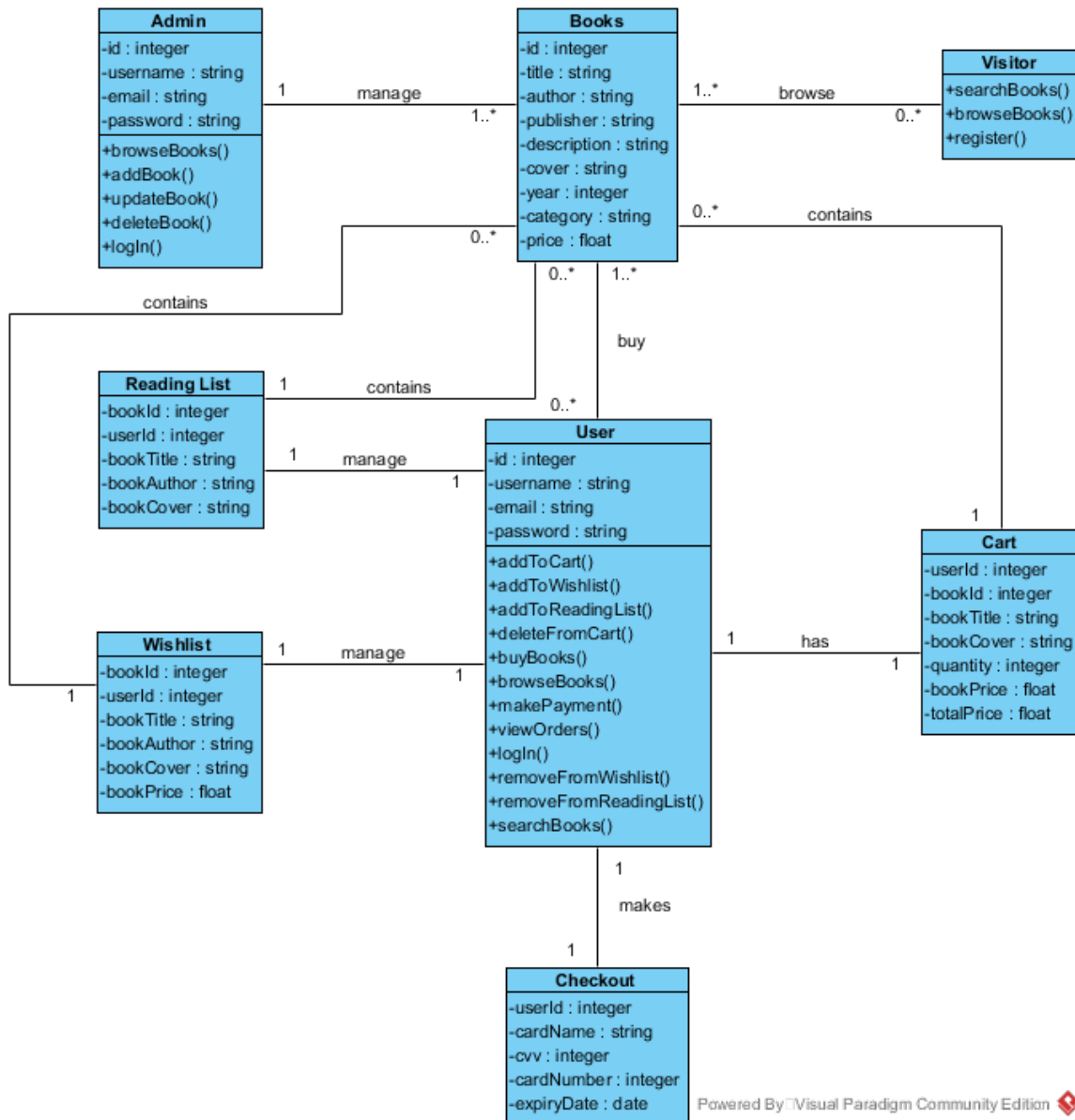
An application program interface (API) is code that allows two software programs to communicate with each other. The API defines the correct way for a developer to write a program that requests services from an operating system (OS) or other application. APIs are implemented by function calls composed of verbs and nouns. The required syntax is described in the documentation of the application being called. (Rouse, 2017)

This e-commerce web application will use the next type of APIs:

- Node.js RESTful API for creating, updating, retrieving or deleting users and books.
- Stripe API in payment process of e-commerce web application ensuring validation and charging credit cards. Credit card information is sent directly to Stripe, ensuring sensitive data never hits the application server.

5 System Architecture

The class diagram has been created (on the image below) to define system architecture of e-commerce web application. A class diagram describes the structure of a system on the way that shows the system's classes, their attributes, operations (or methods) and the relationships among objects.



6 System Evolution

The e-commerce web application The Book Boutique function like a web application for selling books. There are possibilities that web application could be evolved on selling other products connected with books like DVDs of movies based on books and similar. This e-commerce web application could be also evolved onto other products than books but that will require the name change of e-commerce web application.

It is expected that e-commerce web application is fully responsive on all screen resolutions and devices. In case that application is requesting an update to another screen resolutions that will be updated as soon as possible.

At the first, e-commerce web application will use the limited subscription for database hosting plan and with growing base of products and users web application will switch subscription to a bigger data plan or even unlimited data plan.

In the future, The Book Boutique e-commerce web application could be specially developed like mobile application for Android and iOS.

Bibliography

Inflectra, 2018. *Inflectra*. [Online]

Available at: <https://www.inflectra.com/ideas/topic/requirements-definition.aspx>

[Accessed 12 October 2018].

Jaspersoft, 2018. *Jaspersoft*. [Online]

Available at: <https://community.jaspersoft.com/documentation/jasperreports-server-ultimate-guide/v55/availability-requirements>

[Accessed 13 October 2018].

Koelsch, G., 2016. *Requirements Writing for System Engineering*, s.l.: Apress.

Mann, J. & Wang, Z., 2013. *W3C Recommendation*. [Online]

Available at: <https://www.w3.org/TR/2013/REC-performance-timeline-20131212/>

[Accessed 13 October 2018].

Moffat, B., 2018. *The future of customer engagement and commerce*. [Online]

Available at: <http://www.the-future-of-commerce.com/2018/05/02/how-non-functional-requirements-impact-ux/>

[Accessed 13 October 2018].

Osetskyi, V., 2018. *DZone*. [Online]

Available at: <https://dzone.com/articles/how-to-write-the-system-requirements-specification>

[Accessed 12 October 2018].

Rouse, M., 2017. *Techtarget*. [Online]

Available at: <https://searchmicroservices.techtarget.com/definition/application-program-interface-API>

[Accessed 16 October 2018].

Techopedia, 2018. *Techopedia*. [Online]

Available at: <https://www.techopedia.com/definition/5435/graphical-user-interface-gui>

[Accessed 15 October 2018].