

ONLINE BOOK STORE

CIS-611 INTRODUCTION TO SOFTWARE ENGINEERING-SEMESTER PROJECT

GRAND VALLEY STATE UNIVERSITY

NAVEENA VARNA

G02014072

Table of Contents

1.INTRODUCTION:	2
2.SCREENSHOTS OF APPLICATION:	3
3.USE CASE DIAGRAM AND DESCRIPTIONS:	13
4.TRACEABILITY MATRIX:	18
4.1 USE CASES TO REQUIREMENTS:	18
4.2 REQUIREMENTS TO TEST PROCEDURES:	19
5.CLASS DIAGRAM:	22
6.ACTIVITY DIAGRAM:	23
7.TEST CASES:	24
7.1 UNIT TEST AND CODE COVERAGE REPORT FOR THE UNIT TEST:	25
7.2 SYSTEM LEVEL TEST AND CODE COVERAGE REPORT FOR SYSTEM LEVEL TEST:	27
8. SOFTWARE PROCESS MODEL:	28
9. CONCLUSION:	28
10. REFERENCES:	29

1.Introduction:

I have created a project (Online Book Shopping) using software engineering. This project provides a web-based solution for a bookstore owner to sell his books online. The project allows the users to register for an account, login, search and browse through various books available and purchase desired books instantly using online payment (Credit/Debit card) or by using Cash on Delivery (Pay Later) option.

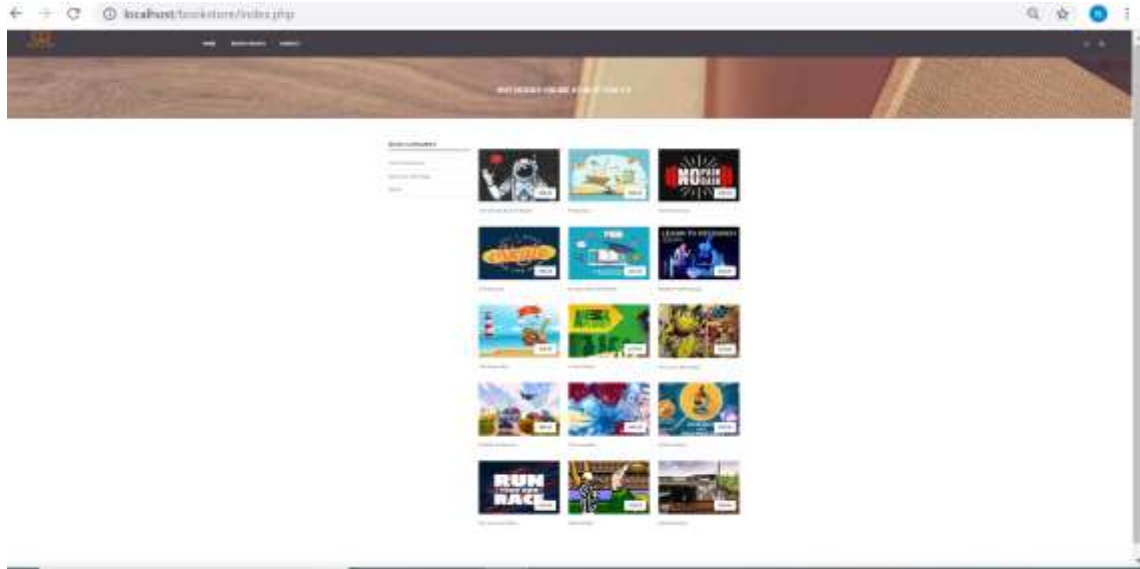
As this is a prototype, I used the test data. For demonstration purposes, I divided the books into 3 categories: Science & Technology, Sports, Action & Adventure. Each category featured 5 books. UML Use-case diagrams are used in system analysis to identify, clarify, and organize system requirements.

I used the unit testing methods to test individual components of the application and System testing methods to verify the functionality against the requirements.

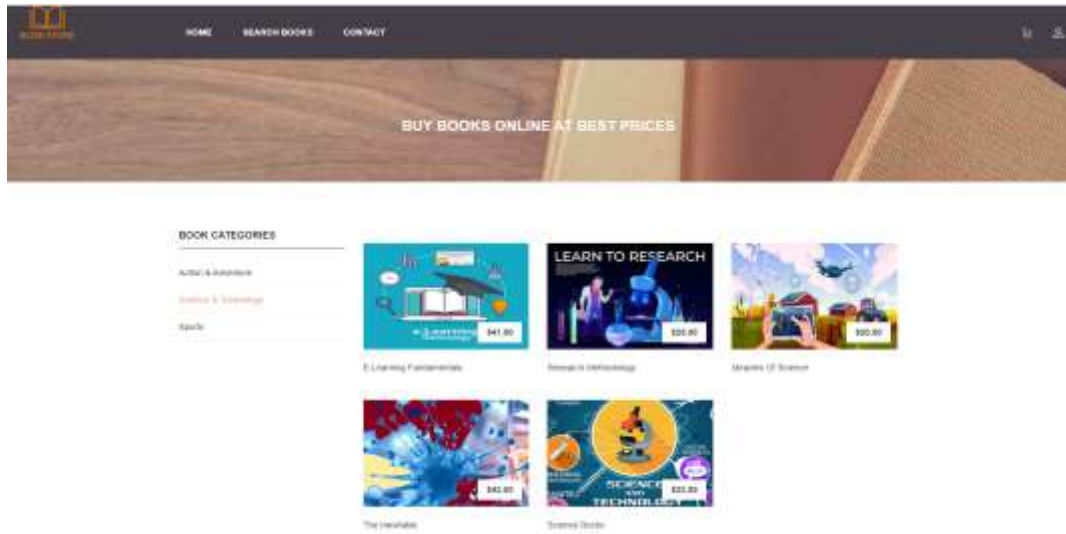
I have developed the project using HTML, CSS, JavaScript, PHP, and MySQL. PHP used to process business logic. HTML, CSS, and JavaScript used to design the user interface and MySQL to store the information.

2.Screenshots of Application:

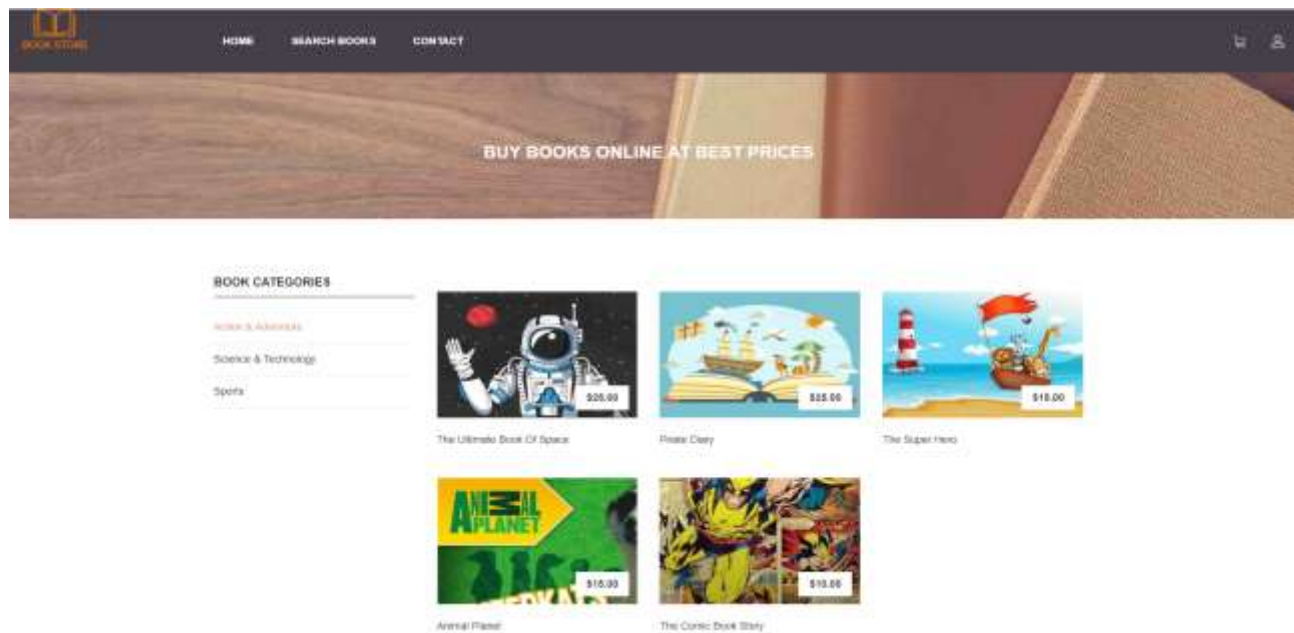
When the user accesses the URL, the following home page will be displayed, where the user can view various menus on the top and book categories on the left side of the screen. The vertical scroll bar on the right side of the page will help the user to scroll through different books vertically.



On the left side of the screen, the user can see different book categories. The user can click on the options and the application filters and the page will show the results based on the user option. If the user clicks on the “Science & Technology” option, then the application shows the following books. As this is a prototype, I confined the results to 5 books. The books are displayed with cover pages for easy readability and a better user interface. The application allows the user to click on the desired book.



Similarly, when the user clicks on the “Action & Adventure” category, the following books are displayed.



Likewise, when the user clicks on the “Sports” category, the following books are displayed.



BOOK CATEGORIES

Action & Adventure

Science & Technology

Sports



No Pain No Gain



A Skating Life



Run Your Own Race

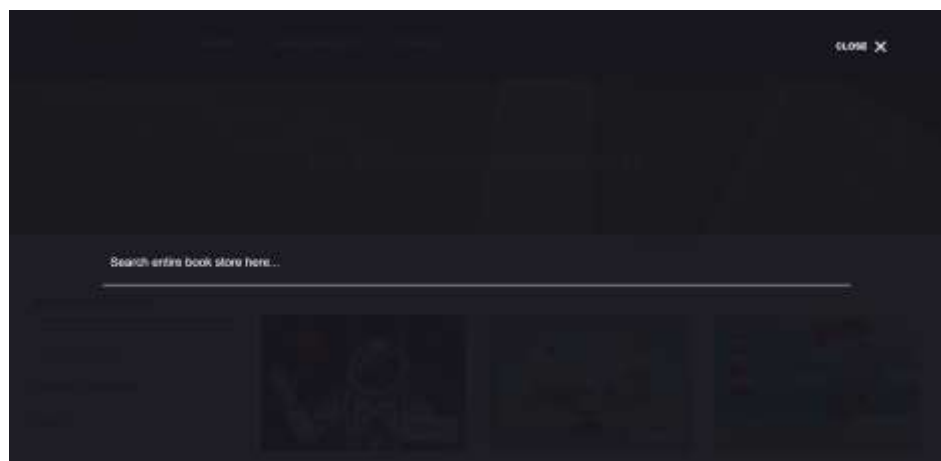


Get in Shape

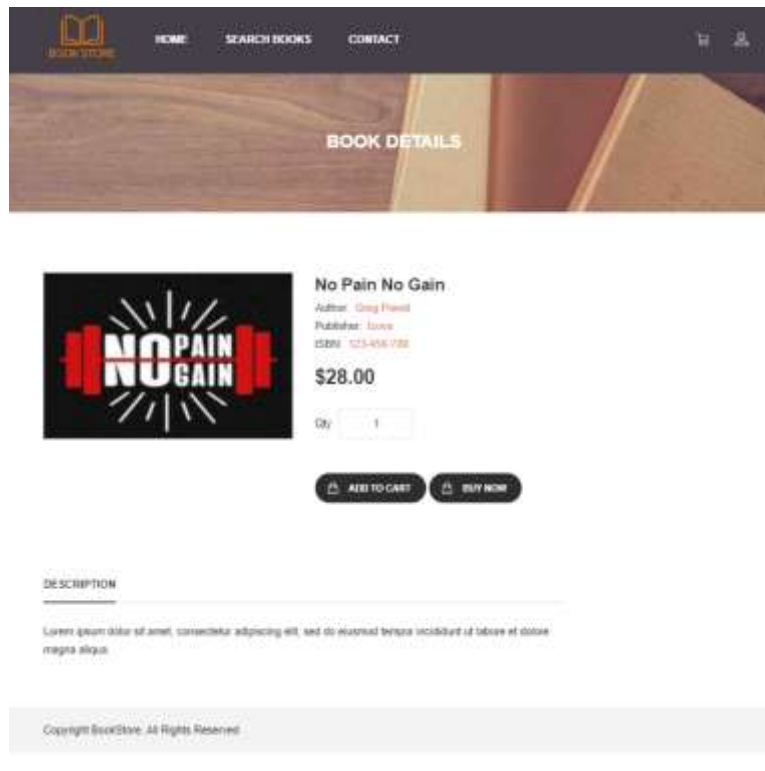


Secret Secrets

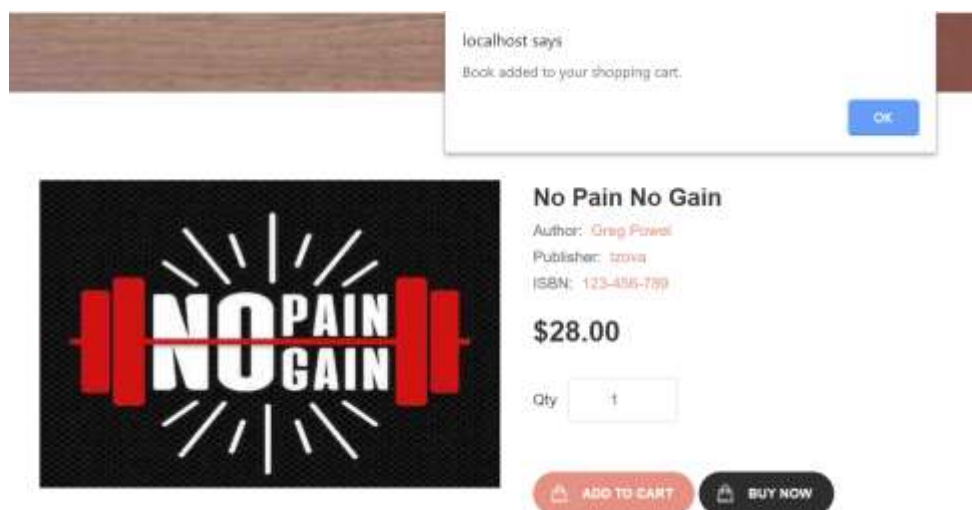
Users can use the search books option to find books using key words.



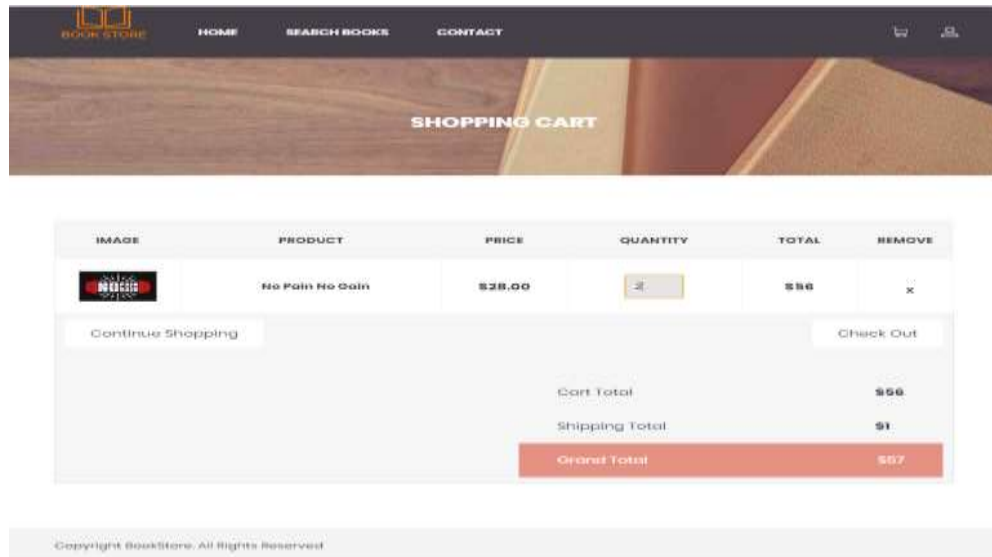
When the user clicks on the desired book, the application goes to a page where more details of the book are displayed, like author name, publisher name, ISBN number, etc. On this page, the application allows the user to add the book to the shopping cart by clicking on the “ADD TO CART” button and allows the user to buy it clicking on the “BUY NOW” button.




The user can click on the “ADD TO CART” button to add a book to the shopping cart, the application shows a pop up indicating that the book has been added successfully to the shopping cart.



The application shows a shopping cart icon on the top right corner of the page. When the user clicks on the icon then the application takes them to a page where it shows all the books that have been added to the shopping cart with the total price in the bottom. The application also shows shipping cost and allows the user to continue shopping or check out right away, by selecting the “Continue Shopping” button or the “Check Out” button.



The screenshot shows the 'SHOPPING CART' page. At the top, there is a navigation bar with 'BOOK STORE', 'HOME', 'SEARCH BOOKS', and 'CONTACT'. A shopping cart icon is in the top right. The main content area has a header 'SHOPPING CART' over a book background. Below is a table with columns: IMAGE, PRODUCT, PRICE, QUANTITY, TOTAL, and REMOVE. One item is listed: 'No Pain No Gain' with a price of \$28.00 and a quantity of 2, totaling \$56. Below the table are 'Continue Shopping' and 'Check Out' buttons. A summary section shows 'Cart Total: \$56', 'Shipping Total: \$1', and 'Grand Total: \$57'. The footer contains 'Copyright BookStore. All Rights Reserved'.

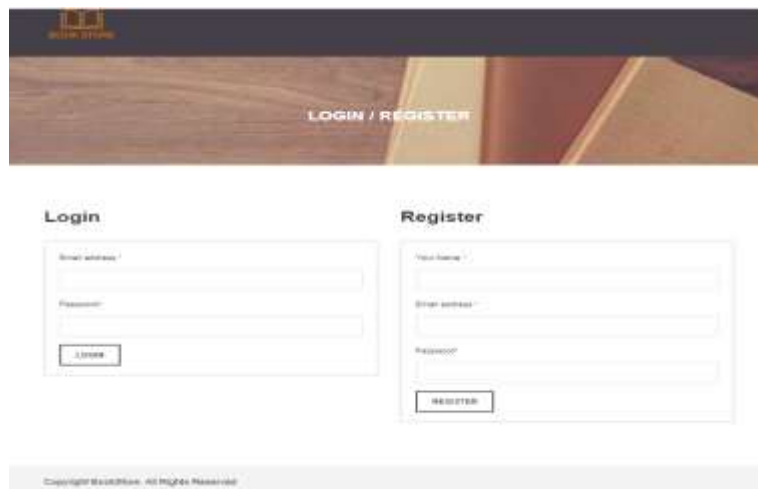
IMAGE	PRODUCT	PRICE	QUANTITY	TOTAL	REMOVE
	No Pain No Gain	\$28.00	2	\$56	X

[Continue Shopping](#) [Check Out](#)

Cart Total: \$56
Shipping Total: \$1
Grand Total: \$57

Copyright BookStore. All Rights Reserved

If the user selects the “Continue Shopping” button the application takes the user to the home page. If the user selects the “Check Out” button the application takes the user to the following page. The application allows the user to login with registered credentials or register as a new user.



The screenshot shows the 'LOGIN / REGISTER' page. It has a header with 'BOOK STORE' and a navigation bar with 'HOME', 'SEARCH BOOKS', and 'CONTACT'. The main content area has a header 'LOGIN / REGISTER' over a book background. Below are two forms: 'Login' and 'Register'. The 'Login' form has fields for 'Email address' and 'Password', and a 'LOGIN' button. The 'Register' form has fields for 'Your Name', 'Email address', and 'Password', and a 'REGISTER' button. The footer contains 'Copyright BookStore. All Rights Reserved'.

Login

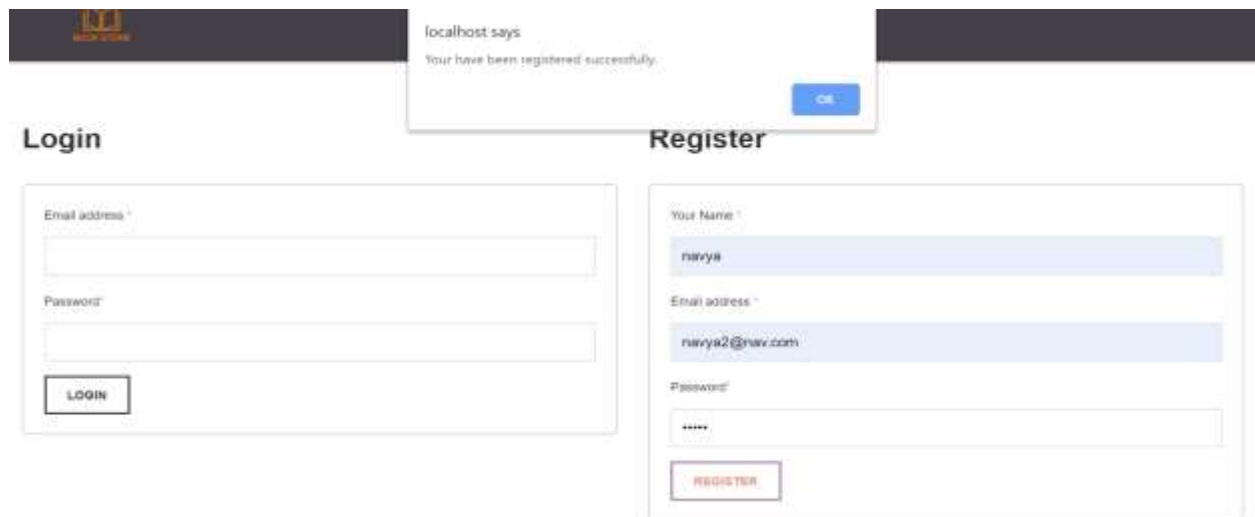
Email address:
Password:

Register

Your Name:
Email address:
Password:

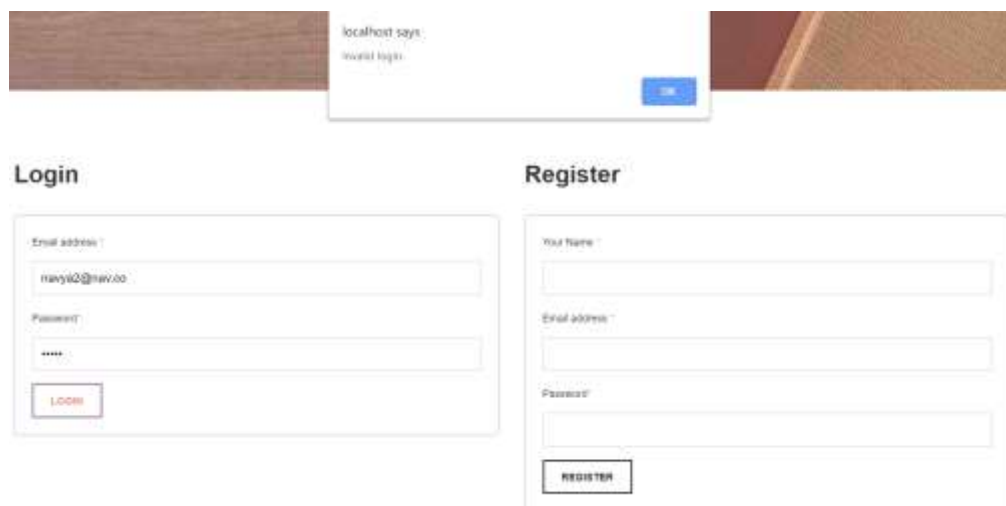
Copyright BookStore. All Rights Reserved

If the user opts to register, then the user can enter the required details and click the “Register” button. Upon successful registration, the application shows a pop up saying that the user has been registered successfully.




The screenshot displays a web application interface. At the top, a dark header bar contains a logo on the left. A white pop-up box is centered, displaying the text "localhost says" and "You have been registered successfully." with a blue "OK" button. Below the pop-up, there are two main sections: "Login" on the left and "Register" on the right. The "Login" form includes fields for "Email address" and "Password", with a "LOGIN" button at the bottom. The "Register" form includes fields for "Your Name", "Email address", and "Password", with a "REGISTER" button at the bottom. The "Email address" field in the Register form is pre-filled with "navya2@nav.com".

If the user is already registered the user can login in with the registered credentials. If the user enters the wrong credentials, then the application shows a pop up saying that the user credentials are invalid.



The screenshot displays a web application interface. At the top, a dark header bar contains a logo on the left. A white pop-up box is centered, displaying the text "localhost says" and "Invalid login." with a blue "OK" button. Below the pop-up, there are two main sections: "Login" on the left and "Register" on the right. The "Login" form includes fields for "Email address" and "Password", with a "LOGIN" button at the bottom. The "Email address" field is pre-filled with "navya2@nav.co". The "Register" form includes fields for "Your Name", "Email address", and "Password", with a "REGISTER" button at the bottom.

Both the fields in the login are mandatory and it is indicated by a red color “*” on top right side of the fields.



Login

Email address *


navya2@nav.com

Password *

.....

LOGIN

Once the user is authenticated then the application takes the user to the “Checkout” page. If the user clicks on the checkout button, the application goes to the “Checkout” page where the user can review the order and enter shipping details.



HOME SEARCH BOOKS

CHECKOUT

Shipping Information

First name * Last name *

navya

Phone * Address *

State* Country*

MI USA

YOUR ORDER

Book Name	Total
No Pain No Gain (2)	\$56
Cart Subtotal	\$56
Shipping	\$1
ORDER TOTAL	\$57

Payment

Cash on Delivery

Card Payment Debit/Credit

Copyright BookStore All Rights Reserved

The user will not be able to proceed to the payment until all the shipping details are entered. If the user chooses a payment method, then the application shows a pop up saying that shipping details must be entered.

The screenshot displays a web application interface. On the left, a 'Shipping Information' form is visible with fields for First name (navya), Last name, Phone, Address, State (MI), and Country (USA). A dark overlay pop-up is centered on the screen, containing the text 'localhost says' and 'Please fill in shipping information', with an 'OK' button. To the right, a 'OUR ORDER' summary table is shown:

Book Name	Total
No Pain No Gain x 2	\$56
Cart Subtotal	\$56
Shipping	\$1
ORDER TOTAL	\$57

Below the order summary, a 'Payment' section offers two options: 'Cash on Delivery' and 'Card Payment Debit/Credit'.

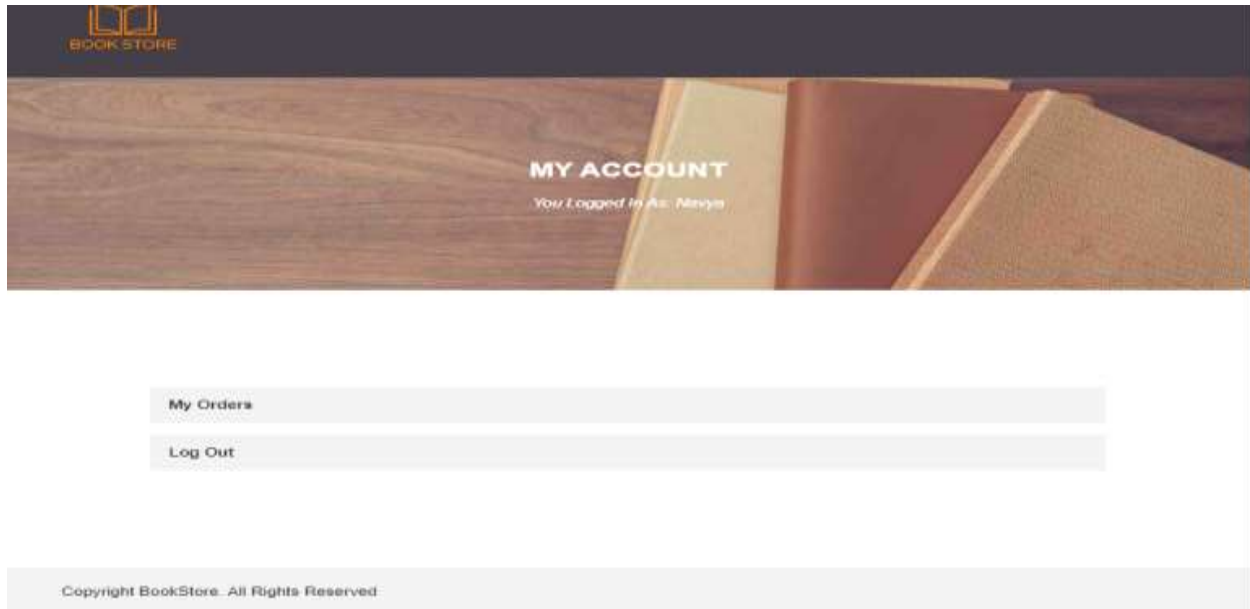
Once the shipping information is entered the user can choose “Cash on Delivery” or “Card Payment Debit/ Credit” option. Once the option is clicked then the application shows a pop up indicating successful placement of order.

This screenshot shows the application after successful order placement. The 'Shipping Information' form now contains the values: First name (navya), Last name (var), Phone (1234567890), Address (315 Iowa), State (MI), and Country (USA). A dark overlay pop-up displays 'localhost says' and 'Your order has been placed successfully.', with an 'OK' button. The 'OUR ORDER' summary table remains the same as in the previous screenshot:

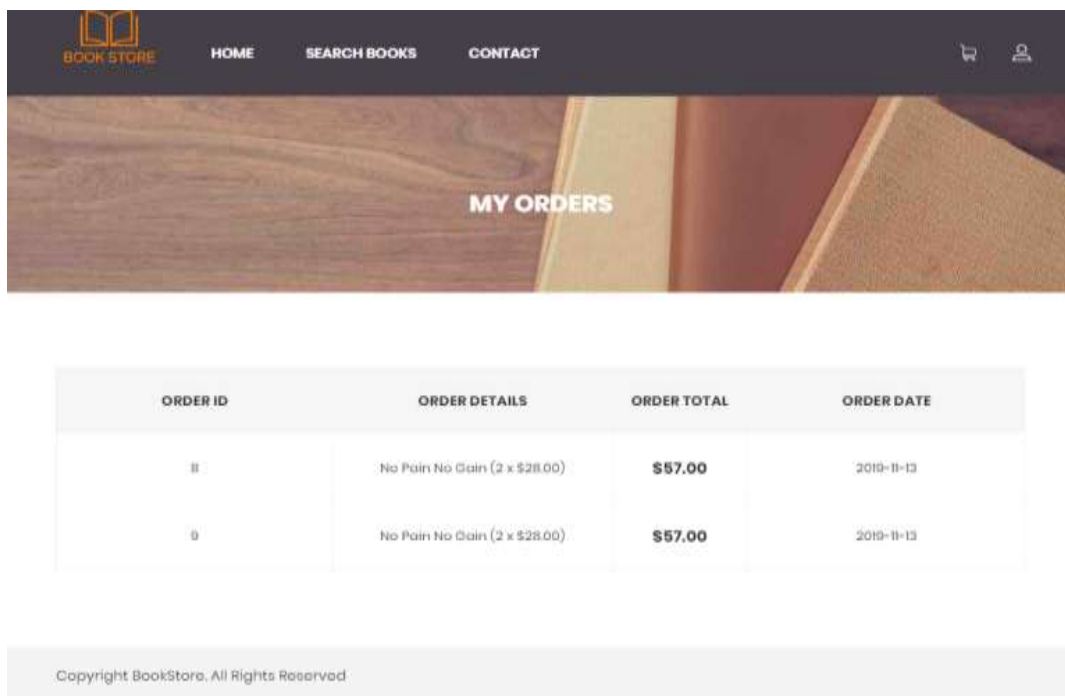
Book Name	Total
No Pain No Gain x 2	\$56
Cart Subtotal	\$56
Shipping	\$1
ORDER TOTAL	\$57

The 'Payment' section with 'Cash on Delivery' and 'Card Payment Debit/Credit' options is still visible at the bottom.

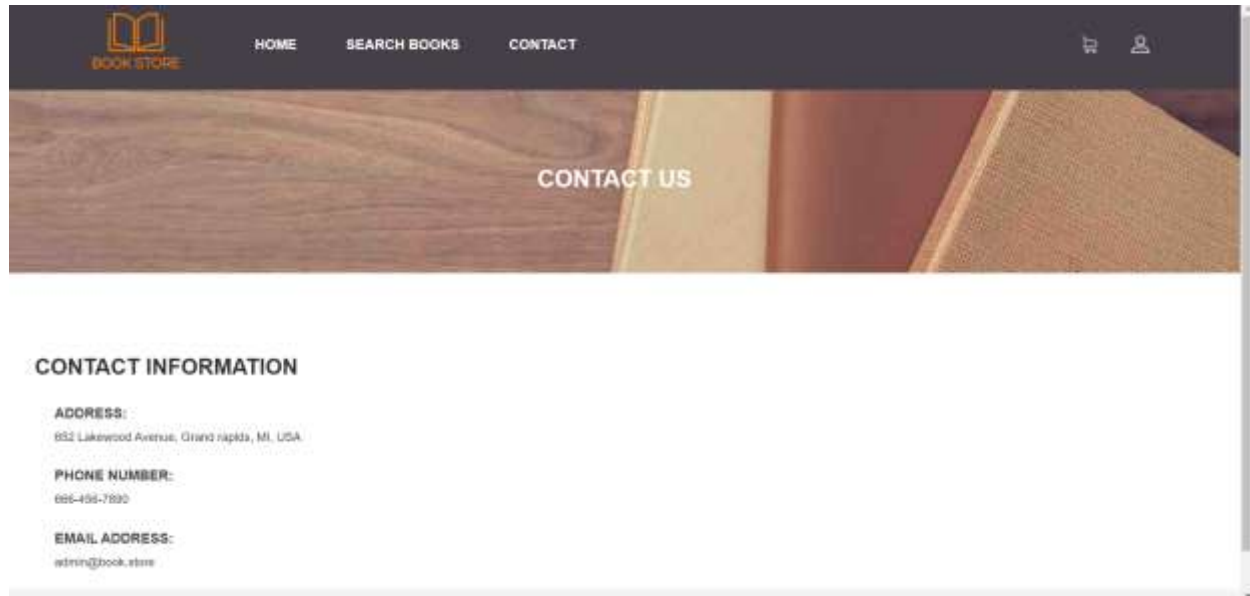
When the user clicks on the human icon on the top right corner of the page, the application takes the user to the account page where the user can see options to view previous orders and logout.



If the user clicks on the “My Orders” option, then the application shows the previous orders.



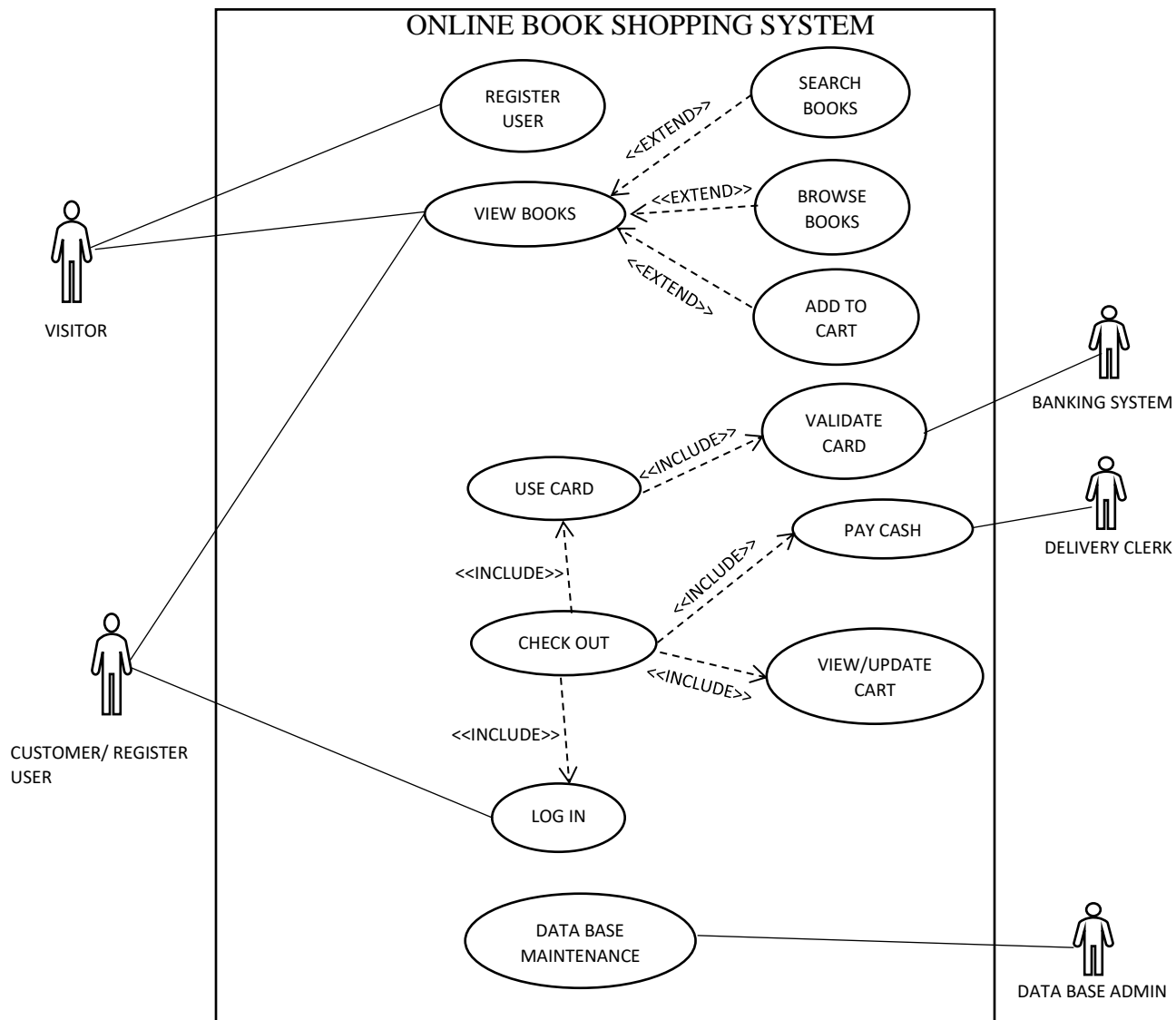
If the user has any questions, the User can use the “Contact Us” option to reach out to the bookstore owner.



3. Use Case Diagram and Descriptions:

Use case Diagrams are used to describe the System boundary, Elements in the System and Elements in the System context. I have used the Use case diagram to find out the requirements of the systems. I have followed the Lamport suggestion as below:

As per Lamport, A major reason today software is so full of bugs is that programmers jump straight in to writing code. He suggests that “write even a rough sketch of what their programs will do before they start coding (Reading Response-The Coming Software Apocalypse).



Use-Case descriptions:

USE CASE : Register User

ACTORS : Visitor (Primary), Customer, Banking system, Delivery clerk

DESCRIPTION

1. The visitor/customer browse the online book shopping system using shopping portal URL.
2. The Visitor/Customer can see all the books available in the shopping portal.
3. The only register user can purchase the books through the shopping portal.
4. The visitor can use the “Register User” use-case to create an account and purchase the books.
5. The visitor asked to enter the personal details on the registration page and chose the username and password to create an account.

USE-CASES : None

USE CASE : View Books

ACTORS : Visitor (Primary), Customer (primary), Banking System, Delivery clerk

DESCRIPTION

1. The visitor and Customer can browse the online book shopping system shopping portal to find the books.
2. The Visitor and Customers can see all the books available in the shopping portal.

USE-CASES : The visitor and customer may use the “Search Books” use-case to search the books by entering the keywords in the search button. The visitor and customer may use the “Browse books by category” use -case to browse the books by category (Science, Arts). The visitor and customer use the “added To Cart” use-case to add the wish books to the shopping cart.

USE CASE : Log In

ACTORS : Customer (Primary), Visitor, Admin (Primary)

DESCRIPTION

1. The customer browses the Log in Page.
2. The customer enters the username and password in the Log In page.
3. The system validates the username and password, if its validated display the customer account information and able to purchase the books, if not display the error information.

USE-CASES : Must have completed the “Register User” use-case if the customer is new (Visitor).

USE CASE : Check out

ACTORS : Customer (Primary), Banking system, Delivery clerk

DESCRIPTION

1. The customer browses the shopping system portal to find the books.
2. The customer added the required books to the shopping cart.
3. The customer uses the checkout use case to purchase the books.
4. During the check out, the customer has an option to purchase the books using a Credit/Debit card or by paying cash on delivery also the customer has an option view or update the shopping cart.

USE-CASES : The customer must have to complete the “Log In” use-case to purchase the books. The customer must have to use the “Use Card” use-case to purchase books using a Credit/Debit card. Similarly, the customer must have to use the “Pay Cash” use-case to purchase book using cash. The customer must have to complete the “View/Update cart” use-Case to view or update the shopping cart.

USE CASE : View/Update Cart

ACTORS : Customer (Primary), Banking System, Delivery clerk,

DESCRIPTION

1. The customer added the required books to the shopping cart.
2. During the check out, the customer can able to view the shopping cart and if the customer wants, customers can increase or decrease the quantity of the books also customers can remove the books from the shopping cart.

USE-CASES : The customer must have completed the “Log-In” use-case.

USE CASE : Use Card

ACTORS : Customer (Primary), the Banking system

DESCRIPTION

1. The customer chose to pay with a credit/debit card to purchase the books.
2. The customer is asked to enter the card details and billing information.
3. The customer enters the credit/debit card and billing information on the billing page.
4. The customer is asked to enter the shipping information.

USE-CASES : The customer must have to use the “Validate Card” use-case to validate the Credit/Debit card. The customer must have completed the “Log-In” use-case.

USE CASE : Validate Card

ACTORS : Banking system (Primary), Customer

DESCRIPTION

1. The customer is asked to enter the credit/debit card details and billing information. Once the customer enters the details, a request is sent to the banking system for validation.
2. If the details are valid, the banking system allows to process the payment and the customer is authorized to proceed with the next steps.
3. If the details are invalid, the banking system declines the process.

USE-CASES : The customer must have completed the “Log-In” use-case.

USE CASE : Pay Cash

ACTORS : Customer (Primary), Delivery clerk

DESCRIPTION

1. The customer chose to purchase books using cash.
2. The customer is asked to enter the delivery address.
3. The customer pays cash to the delivery clerk during the delivery.
4. The delivery clerk processes the payment and delivered the books.

USE-CASES : The customer must have completed the “Log-In” use-case.

USE CASE : Database Maintenance

ACTORS : Data base Admin (Primary)

DESCRIPTION

1. The Admin can add/modify the books in the data base.
2. The admin can add/modify the users.
3. The admin can perform the scheduled data base/portal maintenance.

USE-CASES : None

4.Traceability Matrix:

I have mapped the use cases to requirements and requirements to test procedures by using forward traceability. It shows which requirements are linked to which use cases and test procedures. Using the Traceability matrix if any requirements changes in the application, we can identify the impact on the application. The traceability matrix helps me to write the test cases for the application. I have written the requirements using the template format.

4.1 Use cases to Requirements:

Table 4.1.1

Use cases to Requirements

UC-ID	Use cases	Req-ID	Requirements
UC-1	Registration	Req-1	The application shall allow the users to register in the application by providing the essential details in order to purchase the books.
UC-2	Login	Req-2	The user shall login in to the application with a valid username and password in order to purchase the book.
UC-3	View books	Req-3	The user shall be able to view the list of the available books on the application home page by viewing the book name, book image, book price.
UC-4	Search books	Req-4	The user shall be able to search the books entering the keywords in the search list by book name, author, publisher or book description.
UC-5	Add to cart	Req-5	The user shall be able to add the desired book into the cart by clicking the “add to cart” option.
		Req-6	The user shall be able to view and modify the cart by clicking the “view/modify” cart option.
UC-6	Check out	Req-7	The application should list out all the books in the cart by clicking the “check out” option.

		Req-8	The user shall provide a shipping address in order to complete the payment process.
UC-7	Logout	Req-9	The user shall be able to can log out of the system by clicking on the “Log Out” option on the “My Account” page.

REQ-ID UC-ID	REQ-1	REQ-2	REQ-3	REQ-4	REQ-5	REQ-6	REQ-7	REQ-8	REQ-9
UC-1	×								
UC-2		×							
UC-3			×						
UC-4				×					
UC-5					×	×			
UC-6							×	×	
UC-7									×

4.2 Requirements to Test Procedures:

Table 4.2.1

Requirements to Test Procedures

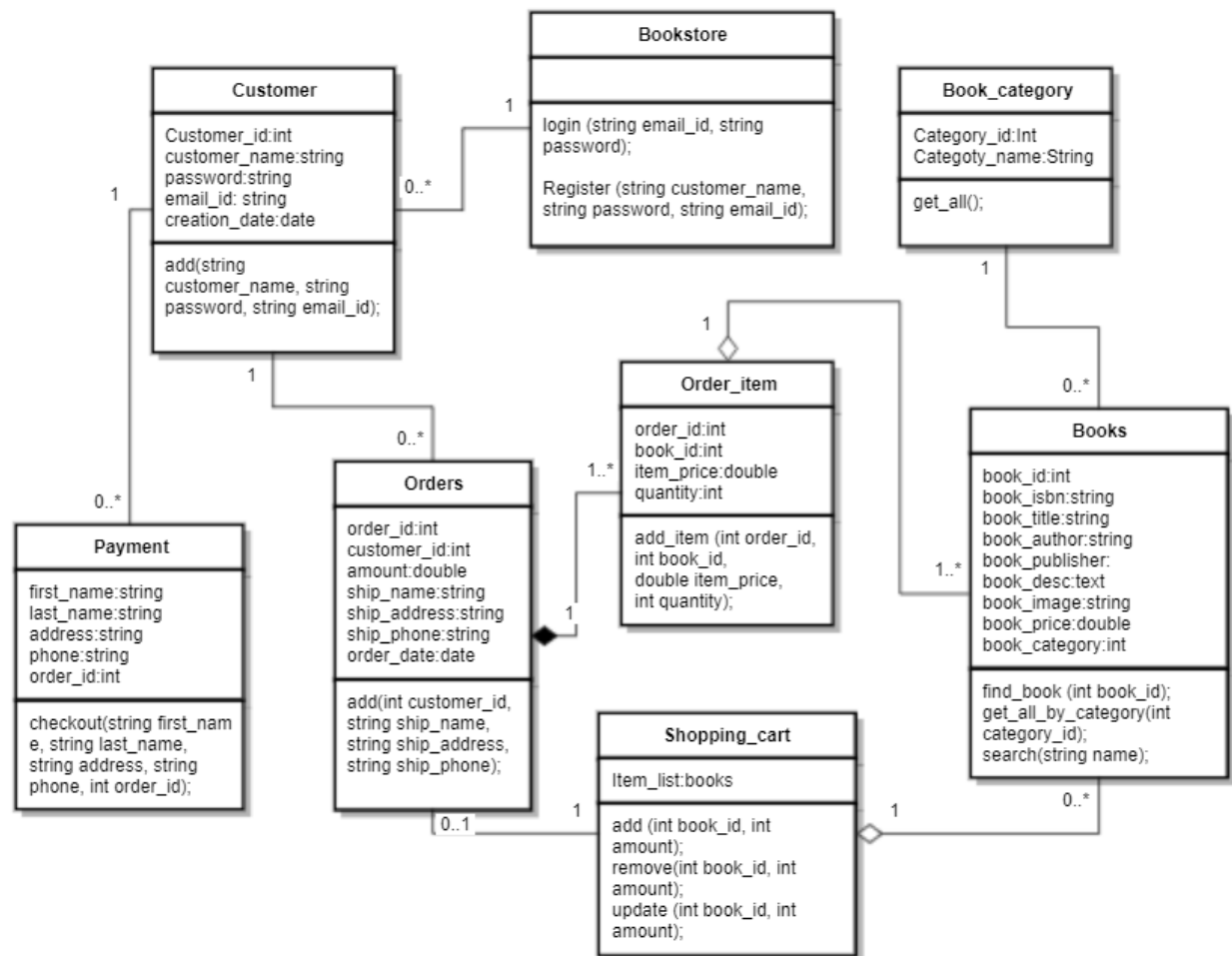
Requirements-ID	TP-ID	Test Procedures
Req-1	TP-1	Without entering the details of the mandatory fields and clicking the Register button, the page should display an error message to the user and not move forward to other pages.
	TP-2	If information is entered in all the fields correctly, the page will display the success information. After “OK” clicked, it will forward it to the main page.

Req-2	TP-3	Without entering the valid username, password and clicking the login button, the page should display an error message “Invalid login” to the user and will not forward it to other pages.
	TP-4	If all the details (email id and password) are entered correctly, it will forward it to the main page.
Req-3	TP-5	This page will show all the available books in the store and it should allow browsing the books by category.
Req-4	TP-6	Click the search menu and it goes to the search page. Enter the book keywords and click the search button, then this page will show the results that contain the keywords.
Req-5	TP-7	Click the add button in the book details page and the system will show a success message. Click “OK”, the system will direct to the shopping cart with the book inside the shopping cart.
Req-6	TP-8	Click the decrease button in the cart for one specific book. The book’s amount should be decreased by one.
	TP-9	Click the decrease quantity button in the cart for one specific book, the book’s amount should be decreased by one.
	TP-10	Click the remove button for one specific book. The book should be removed from the cart.
Req-7	TP-11	Logging in is mandatory to check out the cart. Click the checkout button on the cart page, then it will go to the checkout page, which will list out all the added books in the cart.
Req-8	TP-12	Enter required fields in the shipping information and “Click cash on delivery” or “Card payment debit/credit”.
	TP-13	The page will show the confirmation message, click “OK” then the order will be placed. The shopping cart should be empty.
Req-9	TP-14	After login, at any page click the icon on the right side of the top menu header. This will go to the “My Account” page then click logout. Once logout is done then it will redirect to the login/register page.


TP- ID REQ-ID	TP -1	TP -2	TP -3	TP -4	TP- 5	TP -6	TP- 7	TP- 8	TP- 9	TP- 10	TP- 11	TP- 12	TP- 13	TP- 14
REQ-1	×	×												
REQ-2			×	×										
REQ-3					×									
REQ-4						×								
REQ-5							×							
REQ-6								×	×	×				
REQ-7											×			
REQ-8												×	×	
REQ-9														×

5. Class Diagram:

Class diagrams in the UML are structure diagrams show the static structure of the system. In my own words “We need to make a blueprint before we make any product”. Class diagram Provides an overview of an application. Proper documentation required for any project development. (UML Class diagrams can be used for documenting process during project development).



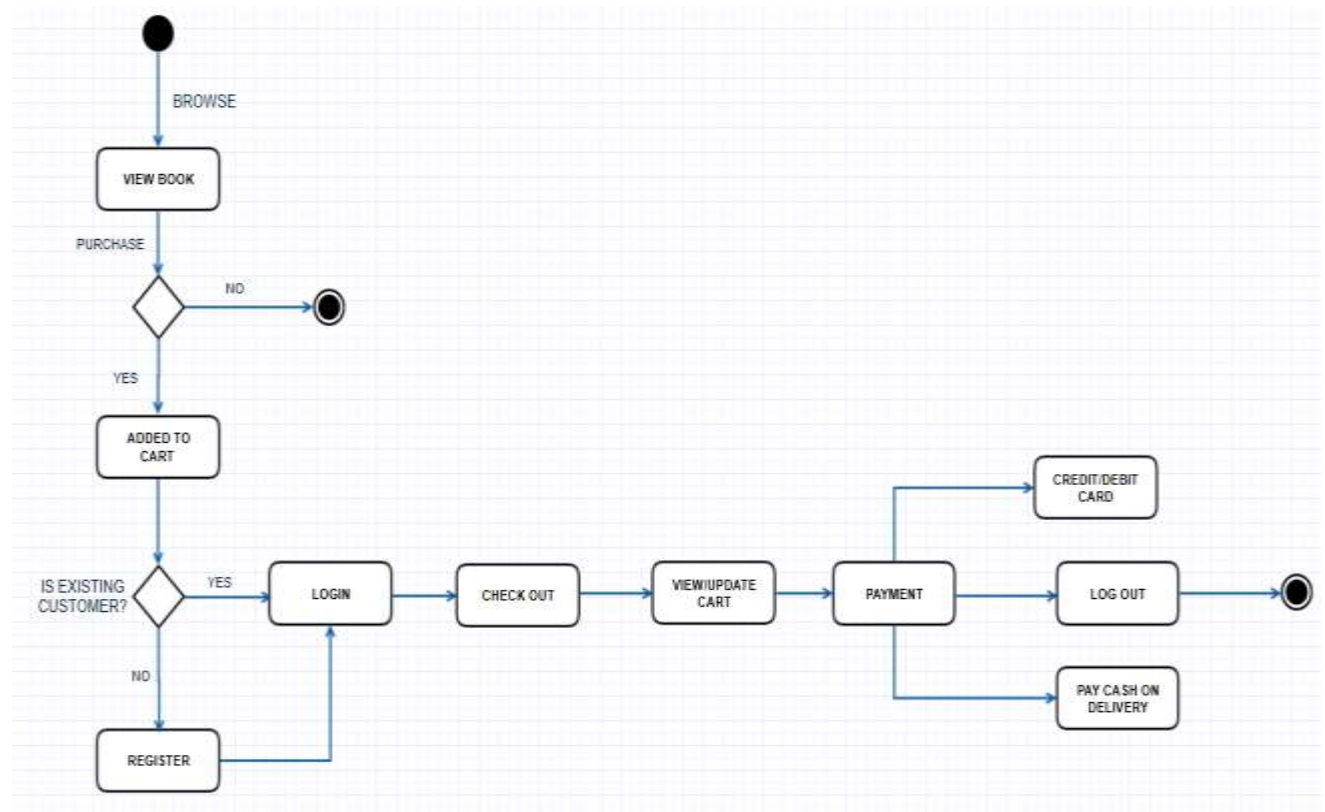
Composition: Between Orders and Order_items 

Aggregation: Order_item and Books, Shopping_cart and Books 

Association: 

6.Activity Diagram:

An activity diagram is a visual representation of flow for an application. An Activity diagram is a behavioral diagram and depicts the dynamic behavior of an application. We can grasp the flow of the program very easily through an Activity diagram.



7.Test Cases:

Any application must be tested to find out the performance of the application and find out the whether application meets the requirements or not. The testing process helped to find out the errors in the application and improve the code quality.

I have completed the Unit test and System test. I have analyzed the Cyclometric complexity during initial testing (Without Changes) and final Testing (With changes).

Registration:

Table 7.1

Registration

Test Case	Description	Expected Result	Actual Result	Result
Registration	User forgot to enter fields	The message should display to fill required fields	Error message showing in the popup.	Passed
Registration	User enter all required fields Correctly	The message should display registration is successful to click “OK”	Registration successful message is showing.	Passed

Login:

Table 7.2

Login

Test Case	Description	Excepted result	Actual result	Result
Login	The user enters wrong email id	Popup Should be showing invalid login	Error message showing in the popup	Passed
Login	The user enters the wrong password	Popup showing invalid login	Error message showing in the popup	Passed
Login	User enter valid email id and password	Popup showing login is successful	Login is successful	Passed

Cart:

Table 7.3

Cart

Test Case	Description	Excepted result	Actual result	Result
Cart	The user can select a book and clicks add to cart button	The book should be added to cart	The book added to cart	Passed
Cart	User can remove the book from the cart	Books and the total amount of cart should be updated	Books and the total amount of cart is updated	Passed
Cart	Users can click the continue shopping button and select the books to cart.	Books and the total amount of carts should be updated.	Books and the total amount of carts is updated	Passed











7.1 Unit test and Code coverage report for the unit test:

Unit testing is testing bits of code in isolation with test code and verify its correctness. In procedural programming, a unit may be a separate function or procedure. I used Punit Code Coverage analysis along with Xdebug to generate reports.

Note: DBUnit extension is not available in the PHPUnit software packages. I have installed the dbunit extension using the composer and it is showing the error message “package phpunit/dbunit is abandoned and no replacement packages also not available (suggested)”. So, I have taken all the functions of input data from the database to PHPUnit Testing.










Registration:

C:\xampp\htdocs\UnitTests / RegistrationTest.php

	Code Coverage									
	Classes and Traits			Functions and Methods				Lines		
Total		100.00%	1 / 1		100.00%	2 / 2	CRAE		100.00%	10 / 10
RegistrationTest		100.00%	1 / 1		100.00%	2 / 2	6		100.00%	10 / 10
testUserRegistration					100.00%	1 / 1	1		100.00%	3 / 3
verifyRegistration					100.00%	1 / 1	5		100.00%	7 / 7





















Login:

C:\xampp\htdocs\UnitTests / LoginTest.php

	Code Coverage									
	Classes and Traits			Functions and Methods				Lines		
Total		100.00%	1 / 1		100.00%	2 / 2	CRAE		100.00%	11 / 11
LoginTest		100.00%	1 / 1		100.00%	2 / 2	6		100.00%	11 / 11
testUserLogin					100.00%	1 / 1	1		100.00%	3 / 3
verifyLogin					100.00%	1 / 1	5		100.00%	8 / 8

Cart test:

C:\xampp\htdocs\UnitTests / CartTest.php

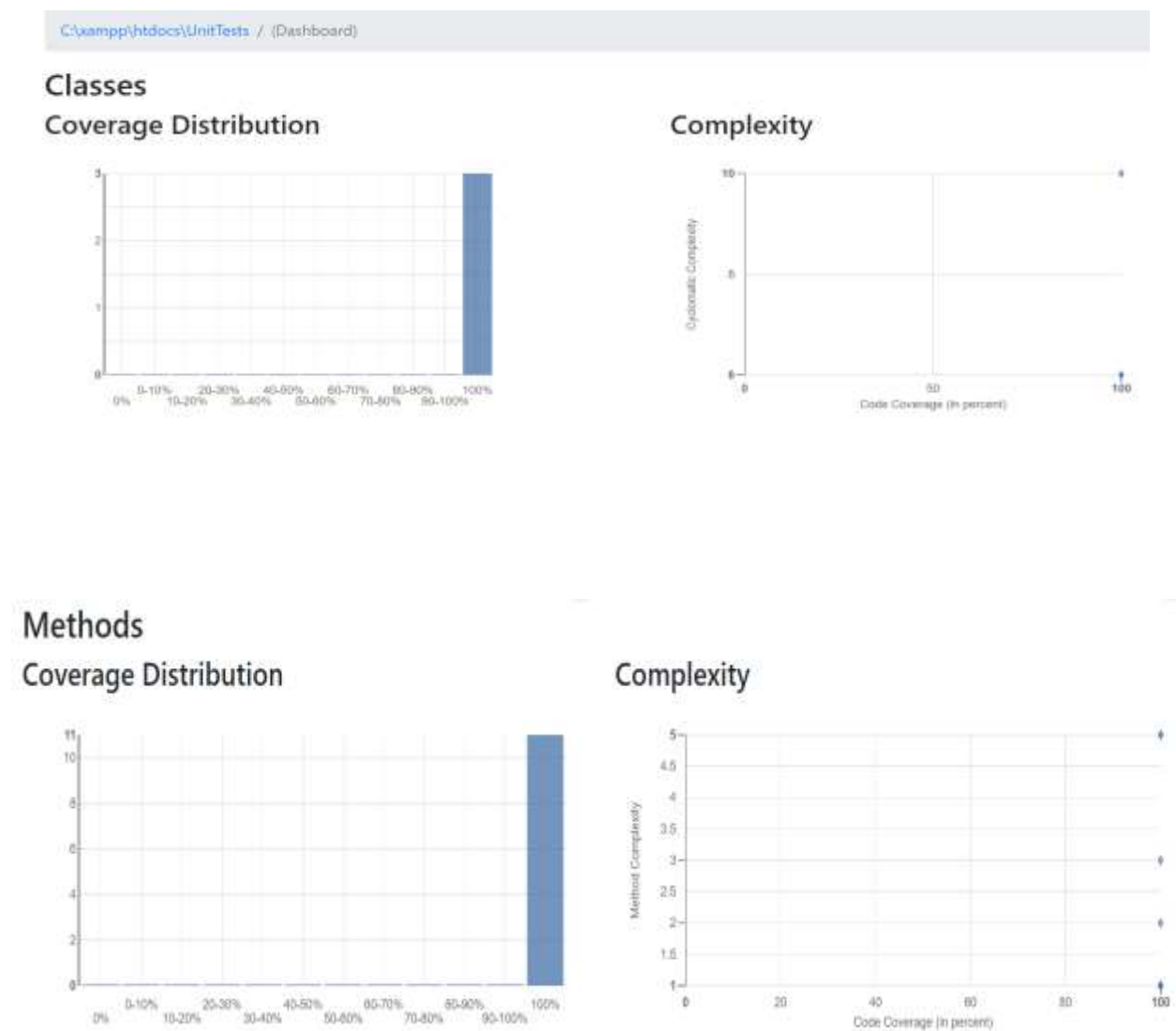
	Code Coverage									
	Classes and Traits			Functions and Methods				Lines		
Total		100.00%	1 / 1		100.00%	7 / 7	CRAE		100.00%	23 / 23
CartTest		100.00%	1 / 1		100.00%	7 / 7	10		100.00%	23 / 23
testItemTotal					100.00%	1 / 1	1		100.00%	3 / 3
testCartTotal					100.00%	1 / 1	1		100.00%	5 / 5
testItemRemovedFromCart					100.00%	1 / 1	1		100.00%	2 / 2
removeItem					100.00%	1 / 1	3		100.00%	6 / 6
getItemTotal					100.00%	1 / 1	1		100.00%	3 / 3
getCartTotal					100.00%	1 / 1	2		100.00%	6 / 6
savedCart					100.00%	1 / 1	1		100.00%	2 / 2

7.2 System level test and Code coverage report for system level test:

System testing:

System Testing is performed on the fully integrated and complete application. I used PHPUnit code coverage analysis along with the Xdebug tool to generate reports.

Code-coverage report:



8. Software process model:

I have used the waterfall software process model. I have understood all my requirements at the early stage of the project. I have completed the testing after code development. The online bookstore project is a mid-size project.

I have followed the following path to complete my project.

- Requirements
- Design
- Implementation
- Testing

9. Conclusion:

The online bookstore project helped me to understand the in-depth practical knowledge about Software Engineering course. I have applied all the principles learned in the class throughout this semester (Use case Diagram, Class diagram, Test cases, Cyclometric complexity, Traceability Matrices, Software process model). For example, during the Code coverage report analysis I understand the importance of cyclometric complexity along with the CRAP index, Project Risks, Code coverage and lines executed in real time (practically) also I learned how to develop a software engineering project.

NOTE: I try to install the PHPUnit on my computer, but I am unable to install, So I consulted the Knowledge market to help the installation process. The people at the Knowledge market helped me to install the PHPUnit also they explain to me how to install the PHPUnit and they explained where I made the mistake during the installation process.

Also, I went over with the people in the writing center (Steel case library) to find any mistakes (Proof-read, Grammar, spelling, sentence formation) in the project report.

10. References:

Freepik-graphics resources for everyone. (n.d.). Retrieved October 19th, 2019 from:

<https://www.freepik.com/>

W3Schools.com-Php Tutorial Retrieved October 15th, 2019 from:

<https://www.w3schools.com>

Tutorial Point Simply easy learning-Php and MySQL retrieved October 15th, 2019 from:

<https://www.tutorialspoint.com/php/index.htm>

What is a Class Diagram? (n.d.). Retrieved November 5th, 2019 from:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

UML Association Vs aggregation Vs Composition (n.d.). Retrieved from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>

PHPUnit-PHPUnit Manual retrieved November 11th, from <https://phpunit.readthedocs.io/en/8.4/>

[Course code]. (January 24th, 2017). Unit testing with PHPUnit: Introduction Retrieved from https://www.youtube.com/watch?v=k9ak_rv9X0Y&list=PLfdiltiRHWGXSggf05W-pJbD47-_d8bJ

Xdebug-installation Retrieved November 14th, 2019 from <https://xdebug.org/>

jQuery-Downloading jQuery Retrieved November 10th, 2019 from <https://jquery.com/download/>

Learn Git and GitHub without any code Retrieved October 24th, 2019 from <https://github.com/twbs/bootstrap/find/master>

Boighor – Free Books Library Ecommerce Store Retrieved October 12th, 2019 from:

<https://themehunt.com/item/1527507-boighor-free-books-library-ecommerce-store>

Gliffy UML Diagram Retrieved on November 5th, 2019 from <https://giffyglyph.com/>

Code coverage report analysis: Sebastian Bergmann-php[tek]

<https://www.youtube.com/watch?v=rPWdD18KQtw>

In Class Notes and Reading Articles