Software Specification Document

Books-R-Us

Chanice Campbell
Justin Chiu
Eric Holbrook
Chris Rowe
Michael Tostenson

Table of Contents

1. Introduction

1.1 Purpose of the system

Due to the popularity of online shopping, many bookstores can be found online only. The purpose of Books-R-Us is to develop an online bookstore system for a respected client that will allow registered users of that system to easily purchase books. This system is not specific to one bookstore, it is intended to be used by a variety of bookstores.

1.2 Scope of the system

Books-R-Us is a website that allows customers to create an account to be able to purchase books from vendors found on the website. This website supports at least a hundred users.

1.3 Objectives and success criteria of the project

The success of the application depends upon meeting the following core set of objectives:

- The implementation of the user interface allows customers to purchase books.
- The website does not crash at any point during transaction processing.
- The design of the system follows the models found in the system models section.

2. Similar Existing Systems

Systems similar to this proposed system are amazon.com and barnesandnoble.com. Amazon allows unregistered users to browse the catalog but not purchase items, just as this system is designed. Only registered users can purchase items, a functionality

shared between these systems. All three systems allow registered users to update profile information, such as address and card information. Amazon does not allow users to pick up in-store, though, which is something our system and Barnes&Noble allow. Prices are able to be updated by valid users, which is a universal functionality through the two examples and this proposed system.

3. Proposed System

3.1 Overview

A online book store which manages buying of books by registered users from registered vendors. It also allows unregistered users to browse the catalog, and administrators to update information and get sale reports. Additionally, books are able to be added and updated by vendors and administrators.

3.2 Functional Requirements

Unregistered Users

1. Unregistered users receive an email confirmation within three minutes of registering a new customer.
2. Unregistered users can register by giving their name, email address, address and birthdate. Each person must provide a unique username and password.

Users

3. A registered user must be able to login and logout from the system.
4. Users must have the ability to choose a book to purchase.
5. Users can subscribe to and unsubscribe from promotions and latest news offered by the bookstore administrators.
6. Users must be able to use their promotion codes in order to take advantage of current discount promotions.
7. The user must be able to modify his/her user profile, or even remove his or her registration.
8. Users must be able to subscribe to and unsubscribe from promotions and latest news offered by the bookstore administrators.
9. Users must be able to add/remove books to the shopping cart.
10. Users can view and update their shopping cart at any time.
11. Registered users unwilling to provide online payment information can reserve books up to 5 days to be purchased and picked up at the bookstore.

Admins/Vendors

12. Admins/vendors must enable business owners to pull predefined reports; namely End of Day sales reports, low-inventory notices, book sales report and publisher sales report.

Admins

13. Administrators can add, delete or update member information, book information (title, category, publisher, etc.) and order information.

Vendors

14. Vendors can update only his/her published book information by adding/removing books to/from the inventory and update prices.

System Requirements

15. The system completes a user's registration by sending a confirmation email.
16. System must utilize authentication by checking username and password.

17. The system shall provide a secure checkout facility where users can provide their payment information.
18. The system must send order confirmation to the customer's email. Order confirmation should include Customer name, Confirmation number, Order ID, Order Date, Shipping address (if applicable), Payment method, ordered items and total amount in dollars.
19. The system must allow different payment methods using cash (in store only), credit card and checks.

3.3 Non-functional Requirements

3.3.1 User interface and human factors

- The system must have an easy-to-use user interface (UI) with screens designed for each part of the system's functionality and suitable for different types of users (Administrators, vendors, users and registered users).
- Slideshow on the homepage should never freeze
- The system provides a user friendly catalog that shows a list of available books sorted by title. This list includes book title, author, and price.

3.3.2 Documentation

- The database can hold at least 1000 records of books.
- The database supports at least 3 vendors.

3.3.3 Hardware considerations

- There must be enough RAM to store information related to the system.

3.3.4 Performance characteristics

- The website should be available twenty four hours a day as long as it is not down for maintenance.
- The website loads the homepage within 5 seconds.
- Switching to a new webpage will not exceed 2 seconds.
- The system must provide multi-user access of at least 5 users, assuring correct concurrent behavior.

3.3.5 Error handling and extreme conditions

- The website should be able to handle any errors regarding to logging in.
- The system should lose less than .5% of data in the case of a power outage.

3.3.6 System interfacing

- The system uses a server to send and receive information to the user/client end.

3.3.7 Quality issues

- The system implementation follows as specified in the 3.5 System Models section.

3.3.8 System modifications

- The system is modifiable and allows new functionalities to be implemented in the future.

3.3.9 Physical environment

- The user interface must be accessible from any common Web browser.
- The database information is locally stored in the developing computers.

3.3.10 Security issues

- The system should encrypt all payment information.
- The system should offer password protection when the user is logging in or registering for an account.

3.3.11 Resources and management issues

- The system should maintain suitable authorization information and authenticate access.
- System should have adequate space allocated in the database for user/book information to be stored.

3.4 Pseudo Requirements (Constraints)

- The system is programmed primarily through the use of the J2EE framework alongside Java servlets for handling our web application components.
- The client-side scripting will be done in the Javascript web-based language.
- AJAX/JSON-based calls will be used to populate client-facing data seamlessly without page reloads/redirects
- The system uses MySQL as a database to store persistent data.
- The system must be Web-based.

3.5 System Models

3.5.1 Scenarios

Scenario for new Customer Andrew making a purchase

- Andrew browses the online book catalog. Andrew selects book of interest to purchase. Andrew creates an account. Andrew submits payment and shipment information. Andrew purchases books. Andrew receives order confirmation.

Scenario for existing Customer Alex making a purchase

- Alex logs in the system. Alex browses the online book catalog. Alex selects book of interest to purchase. Alex submits payment and shipment information. Alex purchases books. Alex receives order confirmation.

Scenario for admin/vendor Adam adding a book

- Adam logs in the system. Adam A clicks on "Books." Adam clicks on "Add a book." Adam fills in book information. Adam confirms and submits the new book.

Scenario for admin/vendor Arwin editing a book

- Arwin logs in the system. Arwin browses the catalog of books. Arwin clicks on "edit" next to the book Arwin wants to edit. Arwin clicks on "Edit a book." Arwin fills in the new book information. Arwin confirms and submits the new edited book information.

Scenario for admin/vendor Aaron removing a book

- Aaron logs in the system. Aaron browses the catalog of books. Aaron click on "remove" next to the book Aaron wants to remove.

Scenario for admin/vendor Art pulling a report of any type

- Art logs into the system. Art clicks on "Reports." Art clicks on the report type he wishes to pull. Art is shown the specified report on the web page.

3.5.2 Use case list and models

Primary Actor	Use Cases
Unregistered user	1. New Login 2. Register 3. Transparent Browse

Registered user	4. Login 5. Logout 6. Browse 7. Add to Cart 8. Remove from Cart 9. View Cart 10. Checkout 11. Reserve Books 12. Modify Profile information 13. Delete Account 14. Subscribe to Promotions/News 15. Unsubscribe from Promotions/News 16. View Order Status 17. Add promotion code
Book Seller / Vendor	18. Browse 19. Login 20. Logout 21. Modify Profile information 22. Delete Account 23. Update Book Info 24. Add New Book 25. Remove Book 26. View End of Day Sales Report 27. Receive Low-Inventory Notice 28. View Book Sales Report 29. View Publisher Sales Report 30. View inventory 31. View order information
Admin	30. Browse 31. Logout 32. Modify Profile information 33. Delete Account 34. Update Book Info 35. Add New Book 36. Remove Book 37. Add Book Info 38. Delete Book Info 39. View End of Day Sales Report 40. Receive Low-Inventory Notice 41. View Book Sales Report

	42. View Publisher Sales Report 43. Add Member Info 44. Update Member Info 45. Delete Member Info 46. View Inventory 47. View Order Information 48. Update Order Information 49. Delete Order Information 50. View Promotions 51. Start Promotion 52. End Promotion

Use Case ID:	1		
Use Case Name:	New Login		
Created By:	Michael Tostenson	Last Updated By:	
Date Created:	6/12/17	Date Last Updated:	

Actors:	Unregistered user
Description:	An unregistered user accesses the website through their chosen browser. In order to get full functionality they go to login. Since they have no username, password, or other information stored they are redirected to the registration page.
Trigger:	
Preconditions:	1. User goes to online store 2. User not registered
Postconditions:	1. User is redirected to registration page.
Normal Flow:	**1.0 New Login** 1. User arrives on homepage 2. Login button is pressed 3. User is brought to login screen 4. User is prompted for a username and password

	5. Login attempt fails 6. User is asked if they would like to register 7. User then redirected to registration page to submit information and pick a username
Alternative Flows:	**1.1 User Doesn't Register** (branch after 6) 1. User declines registration 2. User is returned to home page
Exceptions:	None?
Includes:	None
Priority:	Medium
Frequency of Use:	Every full User will go through this once per account
Business Rules:	
Special Requirements:	1. User can return to homepage at any time
Assumptions:	None?
Notes and Issues:	1. Maximum number of login attempts is 3

Use Case ID:	2		
Use Case Name:	Register		
Created By:	Eric Holbrook	Last Updated By:	
Date Created:	6/13/17	Date Last Updated:	

Actors:	Unregistered User
Description:	An unregistered user is registering for an account.
Trigger:	
Preconditions:	1. User not registered

Postconditions:	1. User information is stored in database
Normal Flow:	**2.0 Register** 1. User arrives on registration page 2. User fills in personal information, including username, password, date of birth, gender, address, and email address. 3. User clicks “Register.” 4. The system will then verify that the username is unique. 5. If the username is unique, the system will send a verification code to their email address. 6. The user will enter the verification code. 7. User is directed back to the home page.
Alternative Flows:	**2.1 Cancel registration** 1. A cancel button is present that will take the user back to the home page.
Exceptions:	**2.0.E.1 Wrong username** (after step 4) 1. If the username is not unique, the registration page will prompt the user for a new username.
Includes:	None
Priority:	High
Frequency of Use:	Once per new user
Business Rules:	None
Special Requirements:	1. Password must be at least 8 characters long
Assumptions:	None
Notes and Issues:	None

Use Case ID:	3
Use Case Name:	Transparent Browse

Created By:	Michael Tostenson	Last Updated By:	Michael Tostenson
Date Created:	6/13/17	Date Last Updated:	6/13/17

Actors:	Unregistered User
Description:	Unregistered User looks through the catalogue without the option to purchase or add books to a shopping cart.
Trigger:	
Preconditions:	1. Unregistered User
Postconditions:	None
Normal Flow:	3.0 Transparent Browse 1. Unregistered User lands on homepage 2. User clicks browse 3. Display book catalogue
Alternative Flows:	**3.1 Search Criteria** (branch after 2) 1. User enters search options 2. Return to step 3
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Often but not most frequent scenario
Business Rules:	None
Special Requirements:	1. Unregistered Users should not have any facility to purchase but it will still display the same.
Assumptions:	None
Notes and Issues:	None

Use Case ID:	4		
Use Case Name:	Login		
Created By:	Michael Tostenson	Last Updated By:	Michael Tostenson
Date Created:	6/13/17	Date Last Updated:	6/13/17

Actors:	Registered User, Vendor, Admin
Description:	Registered User, Vendor, or Admin (hence called User) logs in to use full website functions pertaining to their role.
Trigger:	
Preconditions:	1. User is registered
Postconditions:	1. User has access to website functions
Normal Flow:	1. User lands on homepage 2. User decides to login 3. Redirect to login page 4. User enters username and password 5. Verify username and password with database 6. Login user 7. Redirect to homepage
Alternative Flows:	**4.1 User Forgets Login** (branch after 3) 1. User notifies system that they have forgotten their password 2. Send verification code with username to confirm user 3. User enters code 4. Prompt user for new password and confirm 5. Redirect to login page
Exceptions:	**4.0.E.1 Login Fail** (at step 5) 1. System informs user that username or password is incorrect 2. User is then prompted to enter their information again

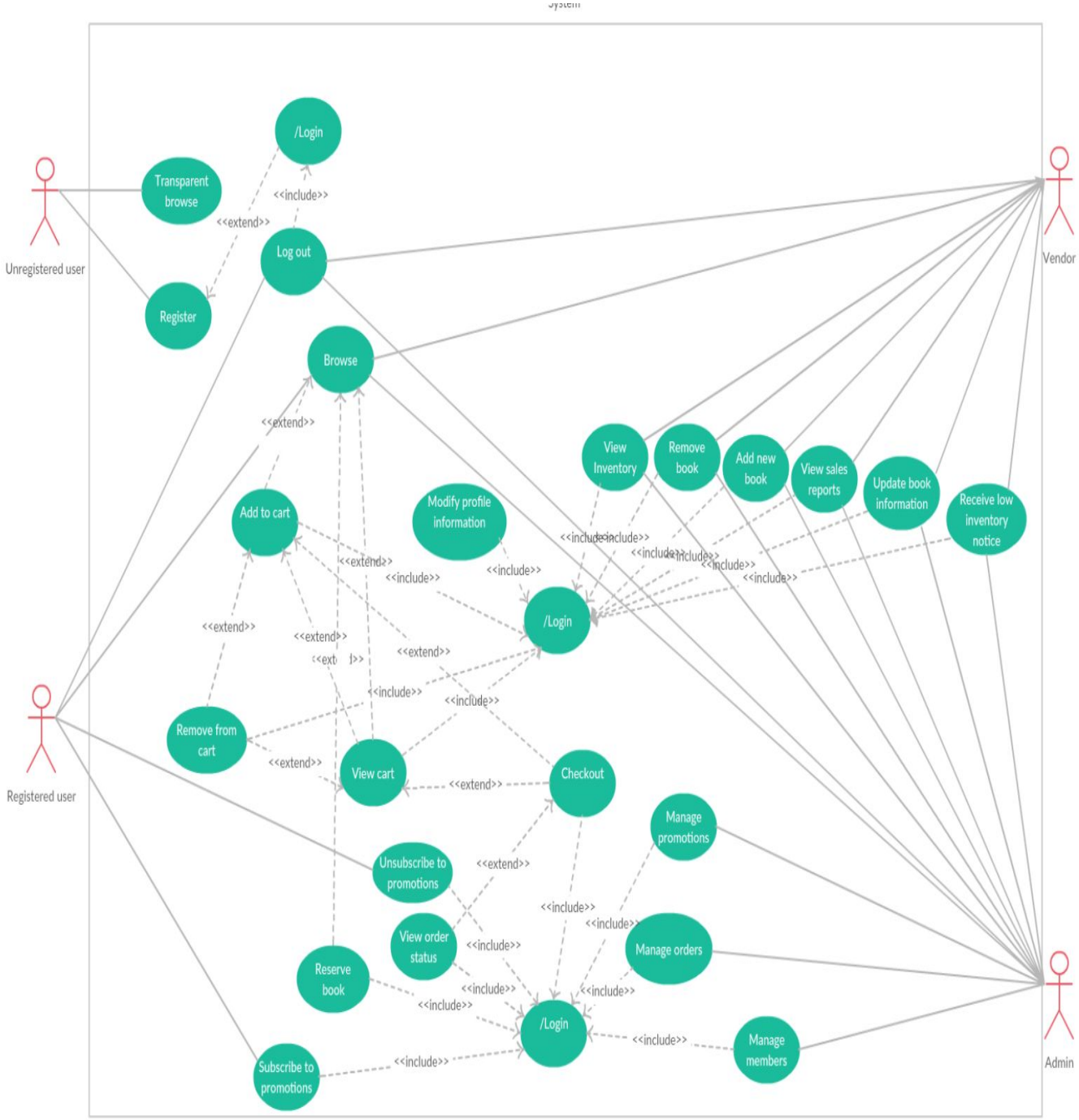
	3. If user fails three times they are locked
Includes:	None
Priority:	Medium
Frequency of Use:	Often
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	5		
Use Case Name:	Logout		
Created By:	Chanice Campbell	Last Updated By:	
Date Created:	6/13/17	Date Last Updated:	

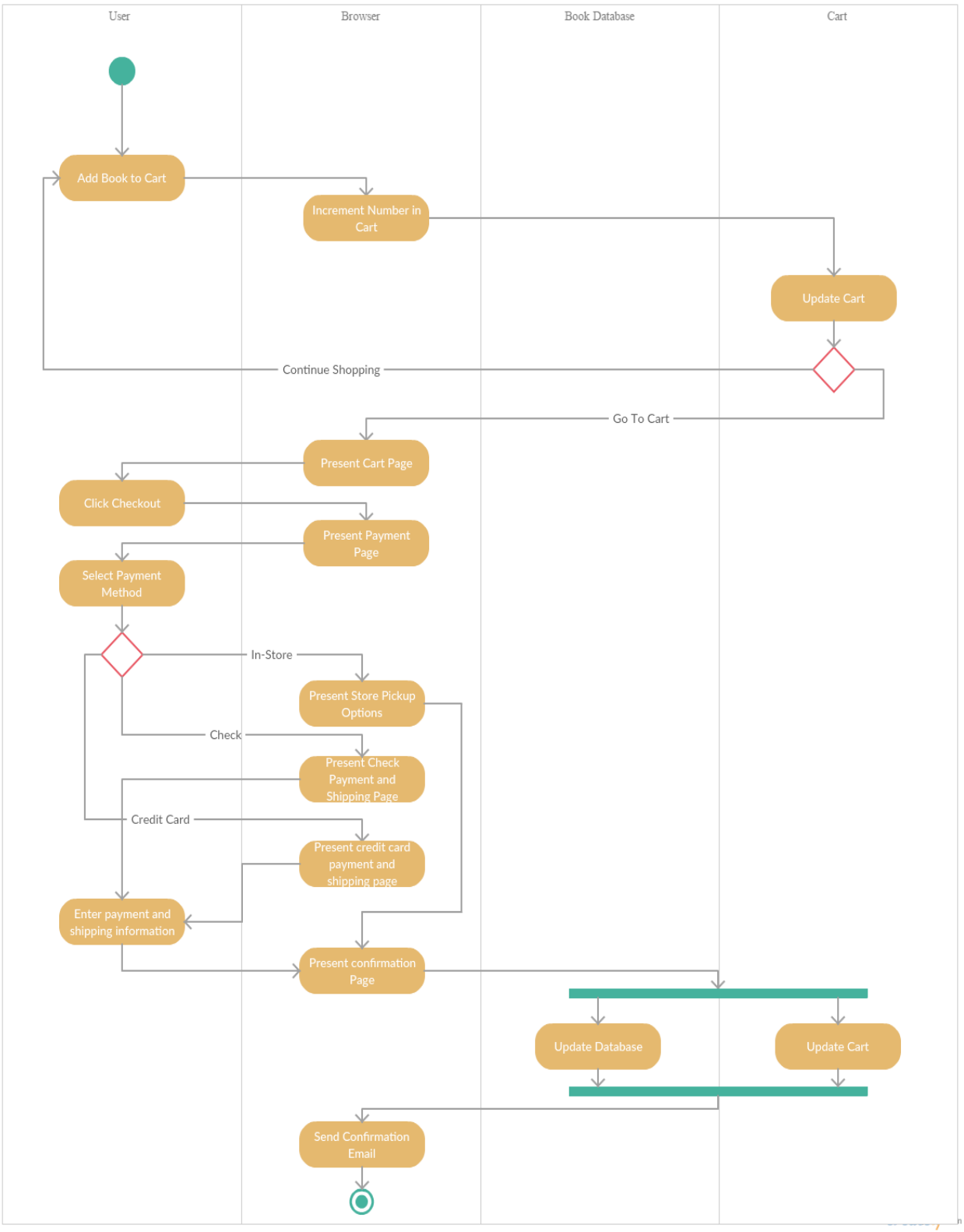
Actors:	Registered User, Admin, and Vendor
Description:	Registered Users, admin, and vendors(henceforth called user) have the capability to logout from the site at any point in their browsing experience.
Trigger:	
Preconditions:	1. Users are logged in
Postconditions:	1. Session tracking will end 2. Redirect webpage to homepage
Normal Flow:	1. User is on any page in the webite 2. User decides to end their session on the site 3. User locates the logout button on the site

	4. After logging out, user is redirected to homepage
Alternative Flows:	
Exceptions:	
Includes:	None
Priority:	Medium
Frequency of Use:	Every time users are ready to leave the website
Business Rules:	None
Special Requirements:	
Assumptions:	1. There will be session tracking that will enable the user to still be logged in for another session if they choose not to log out.
Notes and Issues:	None

3.5.2.1 Use Case Diagram


System
/Login
Transparent browse
<<include>>
<<extend>>
Log out
Register
Unregistered user
Vendor
Browse
<<extend>>
View Inventory
Remove book
Add new book
View sales reports
Update book information
Receive low inventory notice
Add to cart
Modify profile information
<<include>>
<<extend>>
/Login
Registered user
Remove from cart
View cart
Checkout
Manage promotions
Unsubscribe to promotions
View order status
Manage orders
Reserve book
Subscribe to promotions
Manage members
Admin

3.5.2.2 Activity Diagram



3.5.3 Domain object model

3.5.3.1 Data Dictionary

Class		
Users	Users are people who have registered with the website. Users are the general customers of this system and are the ones that purchase books off the website.	
	Public: Yes	
	Relationships	Association: Cart, Order, Admin, UserController Aggregations: None Generalization: None
	Variables - id:int, email:varchar, username:varchar, password:varchar, seclevel:int, shipaddress:varchar, cardsaved:int, fname:varchar, lname:varchar	
	Functions - login(), logout(), reserveBook(), ModifyProfileInfo(), deleteAccount(), subscribe(), unsubscribe(), viewOrderStatus(), addPromotion()	

Class		
Admins	Admins have access to every functionality of the system. They have the highest security level. Admins are the only ones with user management clearance. Admins can pull reports, view inventory, and start/end promotions.	
	Public: No	
	Relationships	Association: User, AdminController, PublisherSales, BookSales, DayReport Aggregations: None Generalization: None
	Variables - id:int, email:varchar, username:varchar, password:varchar, seclevel:int, shipaddress:varchar, cardsaved:int, fname:varchar, lname:varchar	
	Functions - updateBookInfo(), addNewBook(), removeBook(), getEndOfDayReport(), loInventoryNotice(), bookSalesReport(), manageMembers(), viewInventory(), manageOrderInfo(), managePromotions()	

Class		
Vendors/ Publisher	Vendors are the ones selling books on the system. Vendors have partial security access. Vendors can do everything an Admin can do except manage users. Vendors are also called Publishers in our system.	
	Public: No	
	Relationships	Association: PublisherController, User, BookSales, DayReport, PublisherSales Aggregations: None Generalization: None
	Variables - id:int, email:varchar, username:varchar, password:varchar, seclevel:int, shipaddress:varchar, cardsaved:int, fname:varchar, lname:varchar	
	Functions - updateBookInfo(), addNewBook(), removeBook(), getEndOfDayReport(), lowInventoryNotice(), bookSalesReport(), publisherSalesReport(), viewInventory(), manageOrderInfo(), managePromotions()	

Class		
Book	This is a single item in which there are many existing in the system. They can also be a part of shopping carts.	
	Public: No	
	Relationships	Association: Genre Aggregations: Cart Generalization: None
	Variables - ISBN: int, title: string, author: string, publisher: string, genres: <List> Genre, year: int, price: double, stock: int, sold: int, hold: int	
	Functions - none	

Class		
Genre	This is a single Genre object that is linked to a book.	
	Public: No	
	Relationships	Composition: Book Aggregations: None Generalization: None
	Variables - book_id: int, genreType: string	
	Functions - none	

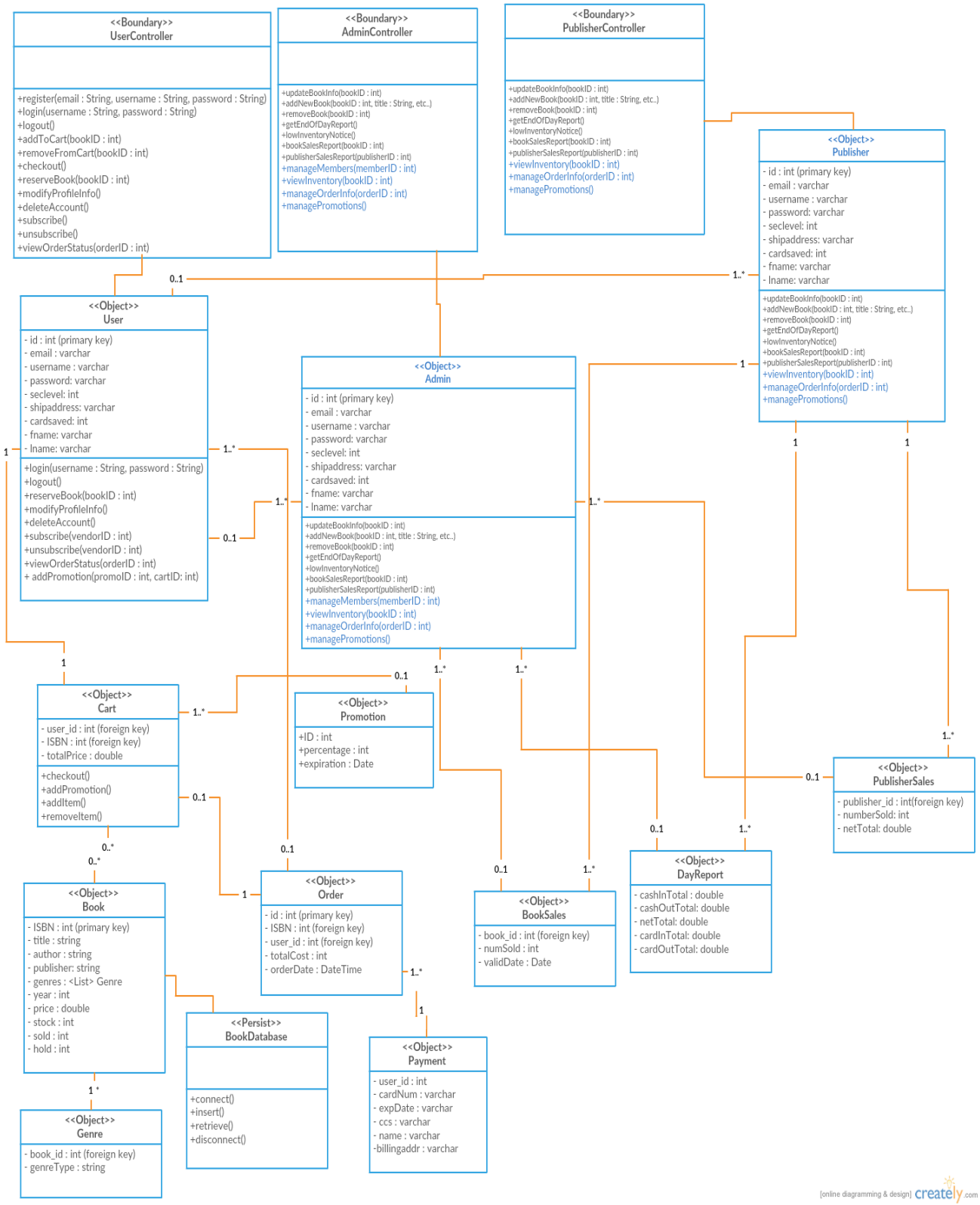
Class		
Cart	This is an object that each user will have one of at any given time. Each cart will contain any number of books. Orders will all contain one cart (as the basis for what was purchased)	
	Public: No	
	Relationships	Association: Book Aggregations: Order Generalization: None
	Variables - user_id: int, ISBN: int, totalPrice: double	
	Functions - checkout(), addPromotion(), addItem(), removeItem()	

Class		
Order	This is a single object that contains a customer's order information.	
	Public: No	
	Relationships	
	Variables - id: int, ISBN: int, user_id: int, totalCost: int, orderDate: DateTime	
	Functions - None	

Class		
Payment	This is a single object that is linked to an order. An order can have a single payment object linked to it. This object holds the customer's payment information.	
	Public: No	
	Relationships	Association: Order Aggregations: None Generalization: None
	Variables - user_id: int, cardNum: varchar, expDate: carchar, css: varchar, name: varchar, billingaddr: varchar	
	Functions - None	

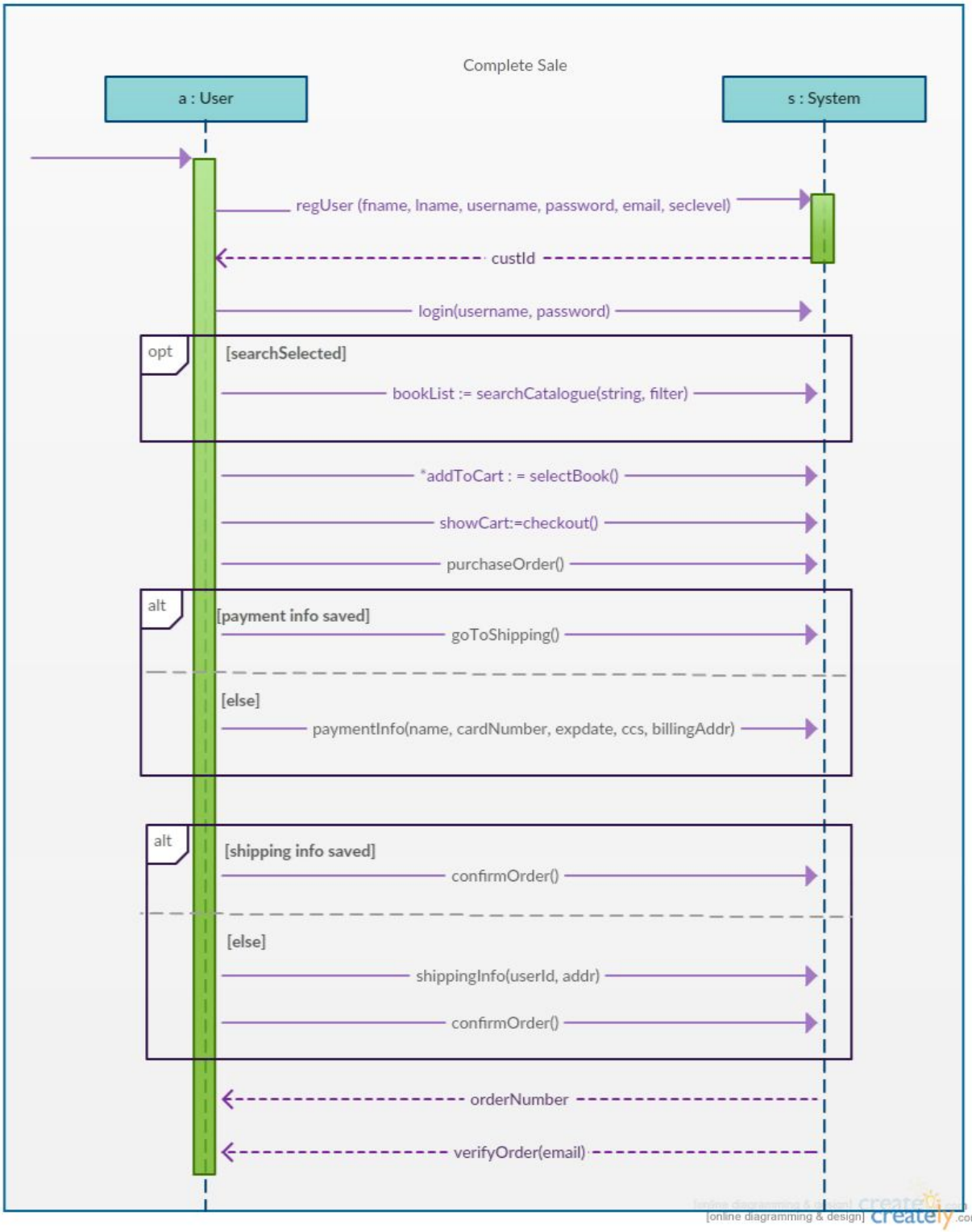
Class		
Promotion	This is a single object that is linked to a cart.	
	Public: No	
	Relationships	Aggregations: None Generalization: None Composition: Cart
	Variables - promo_id: int, cart_id: int,	
	Functions - none	

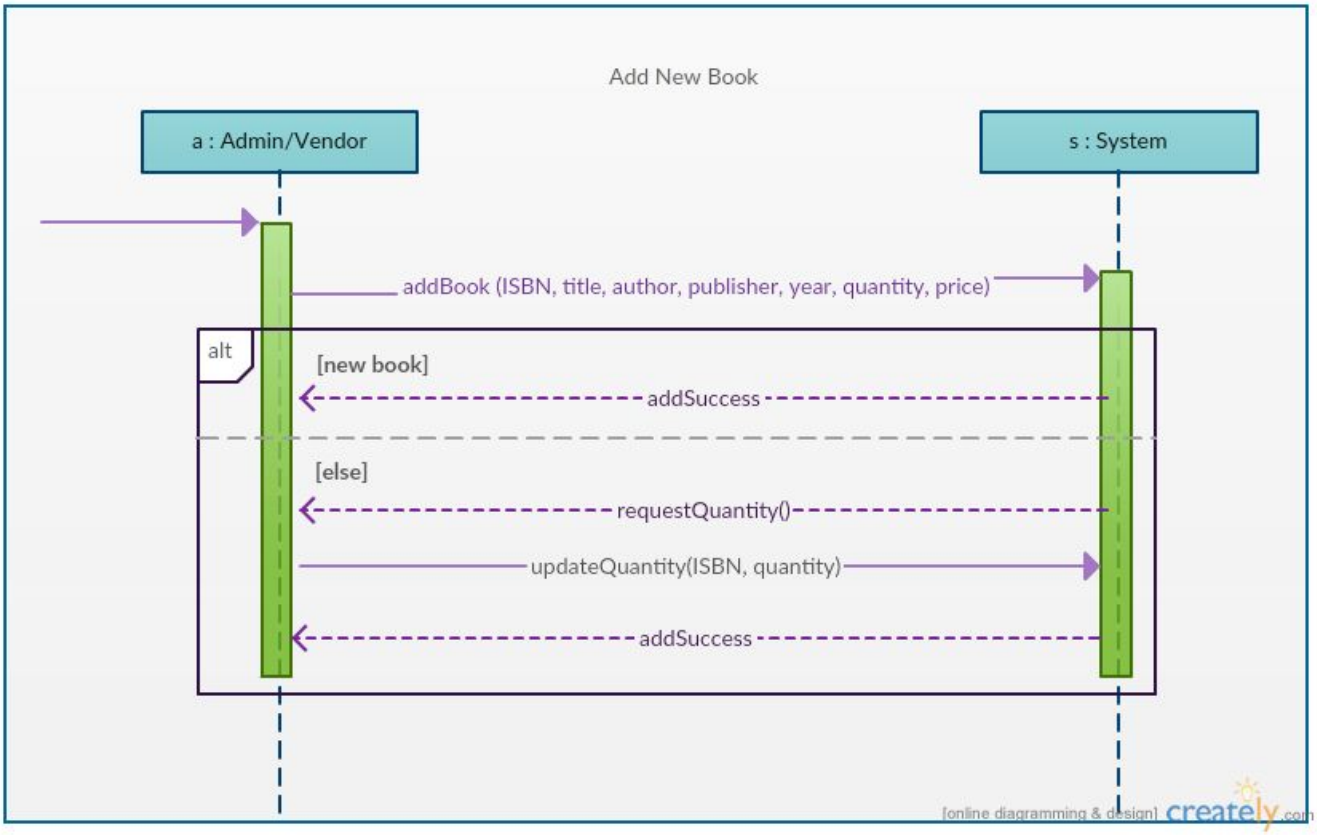
3.5.3.2 Class Diagram

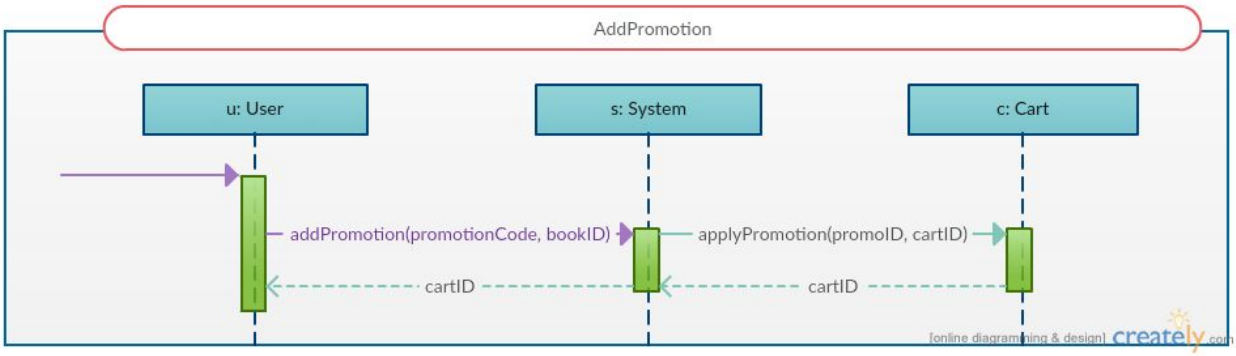


3.5.4 Dynamic Models

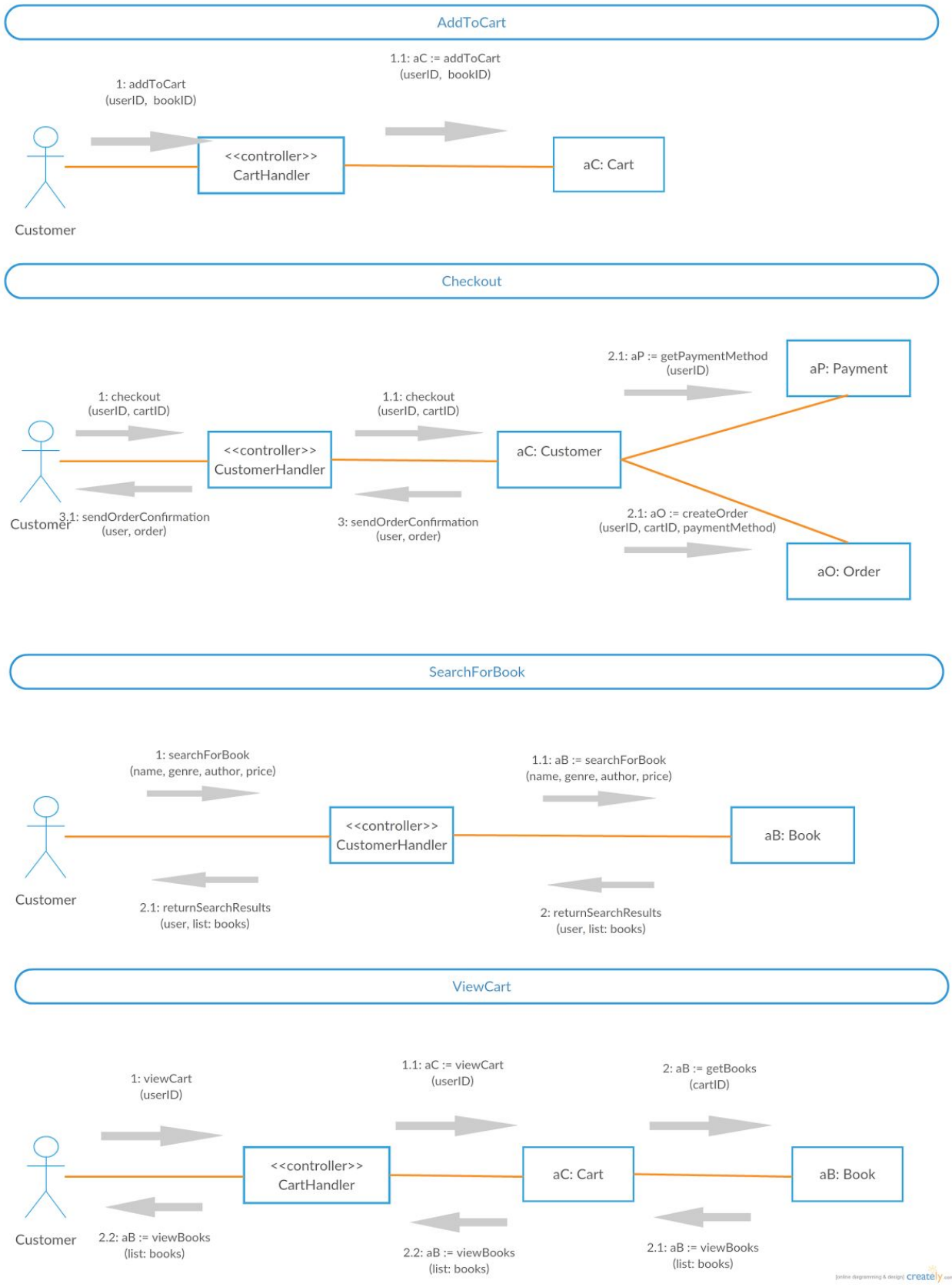
3.5.4.1 System Sequence Model




Add New Book
a : Admin/Vendor
s : System
addBook (ISBN, title, author, publisher, year, quantity, price)
alt
[new book]
addSuccess
[else]
requestQuantity()
updateQuantity(ISBN, quantity)
addSuccess
[online diagramming & design] creately.com


AddPromotion
u: User
s: System
c: Cart
addPromotion(promotionCode, bookID)
applyPromotion(promoID, cartID)
cartID
cartID
[online diagramming & design] creately.com

3.5.4.2 Communication diagram



3.5.5 User Interface

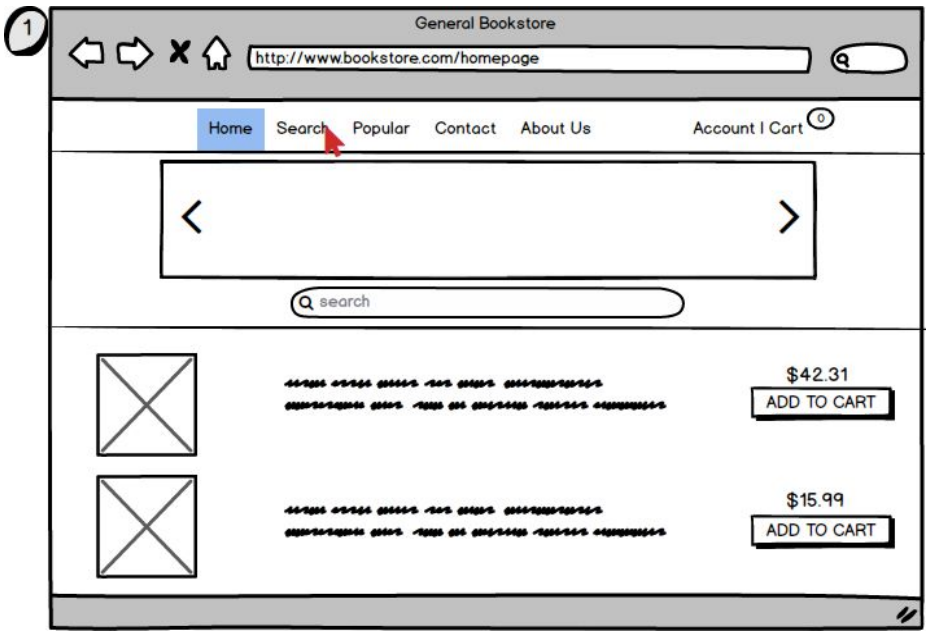


Figure 3.5.5.1 Home screen

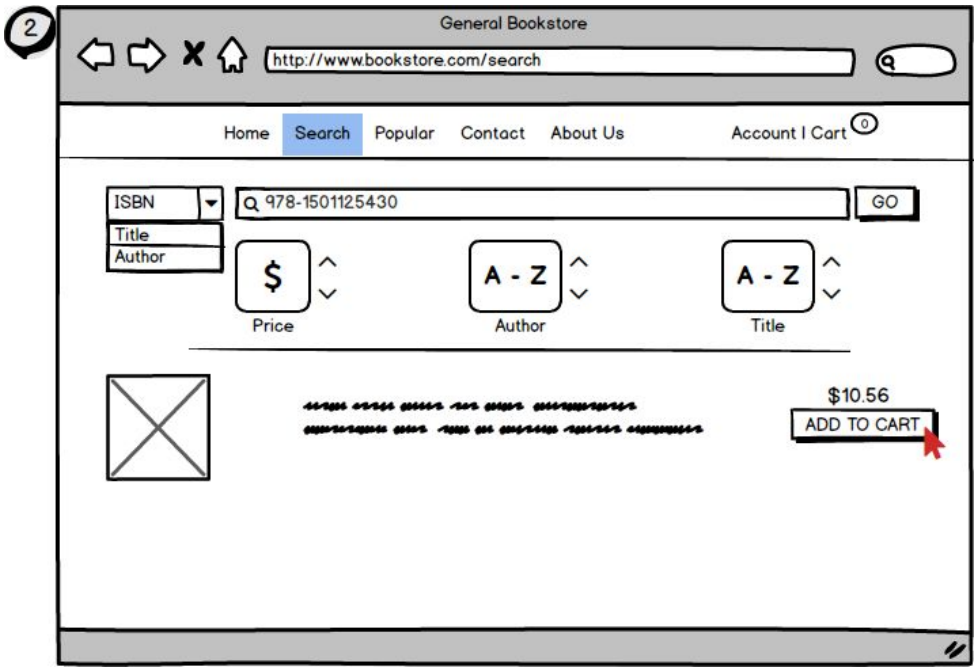


Figure 3.5.5.2 Search screen and add to cart

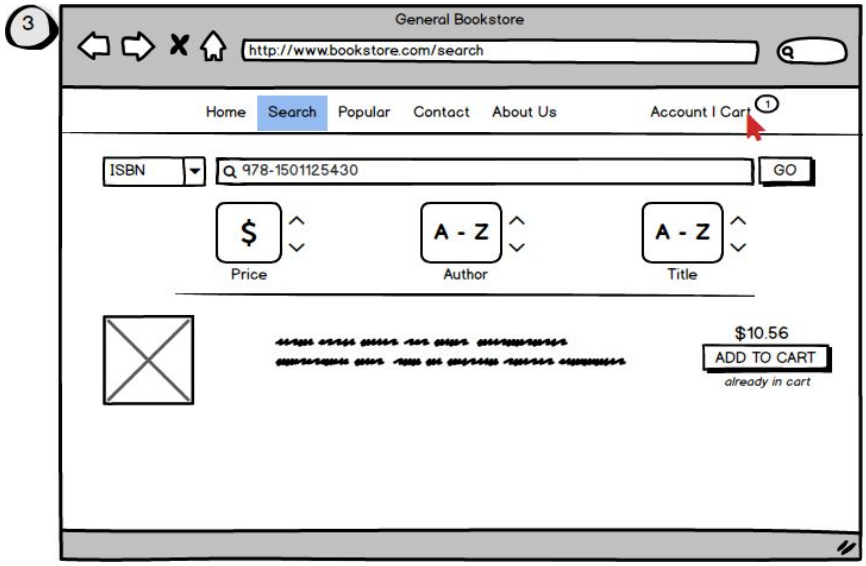


Figure 3.5.5.3 Selecting Cart from user point of view

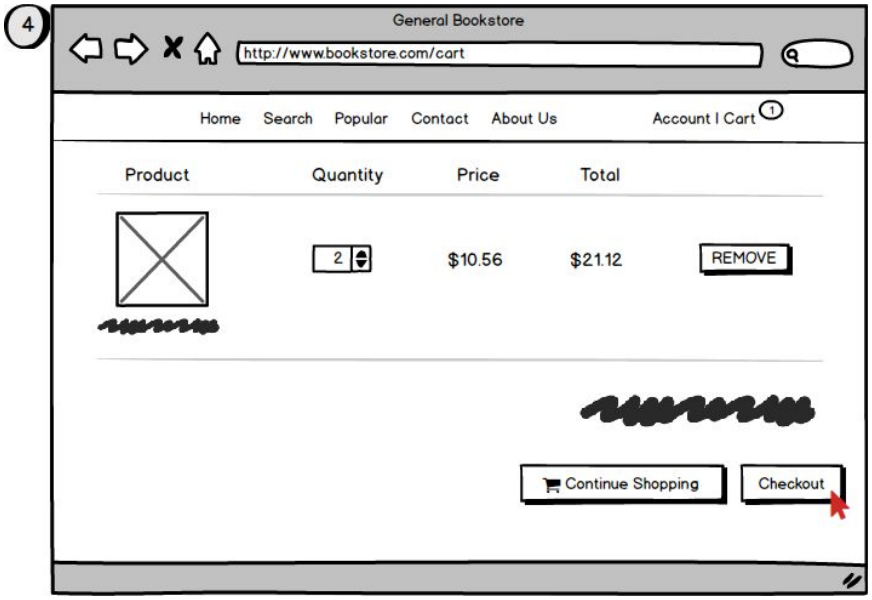


Figure 3.5.5.4 Review cart screen

Figure 3.5.5.5 Select payment method screen

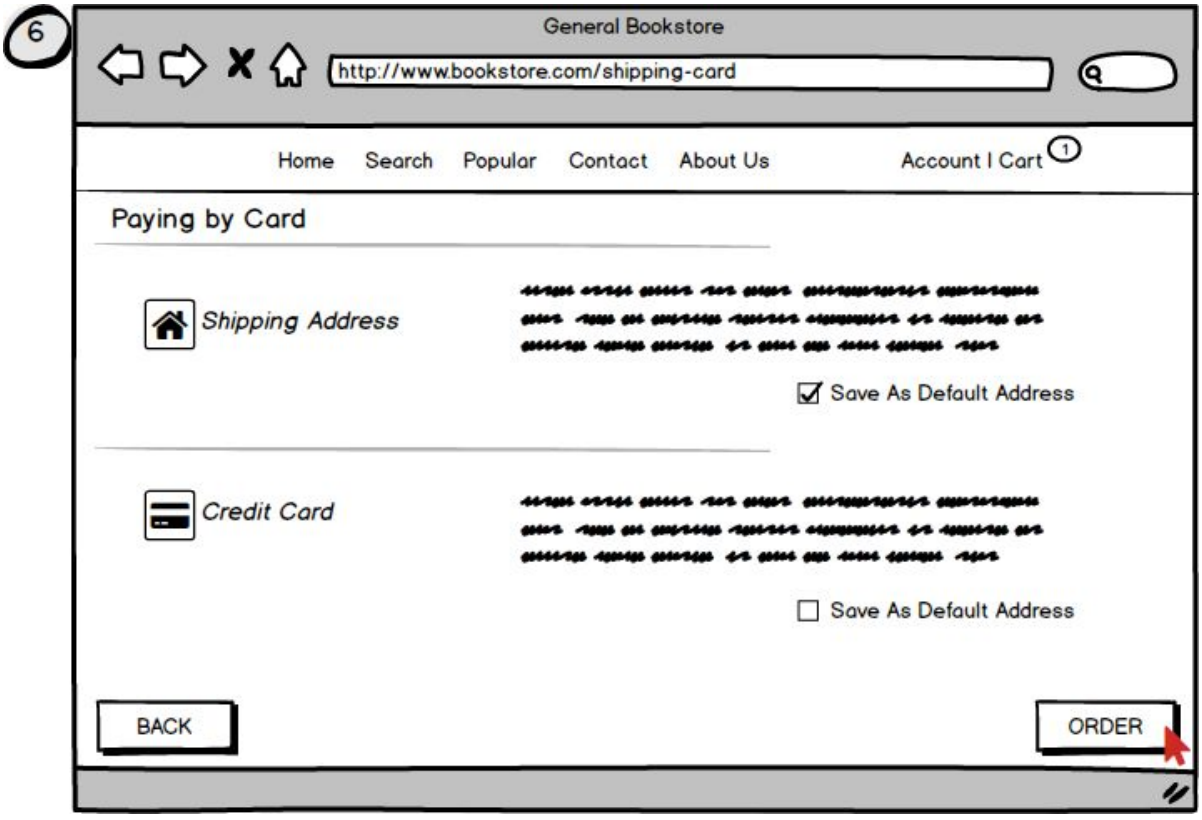


Figure 3.5.5.6 Review order screen

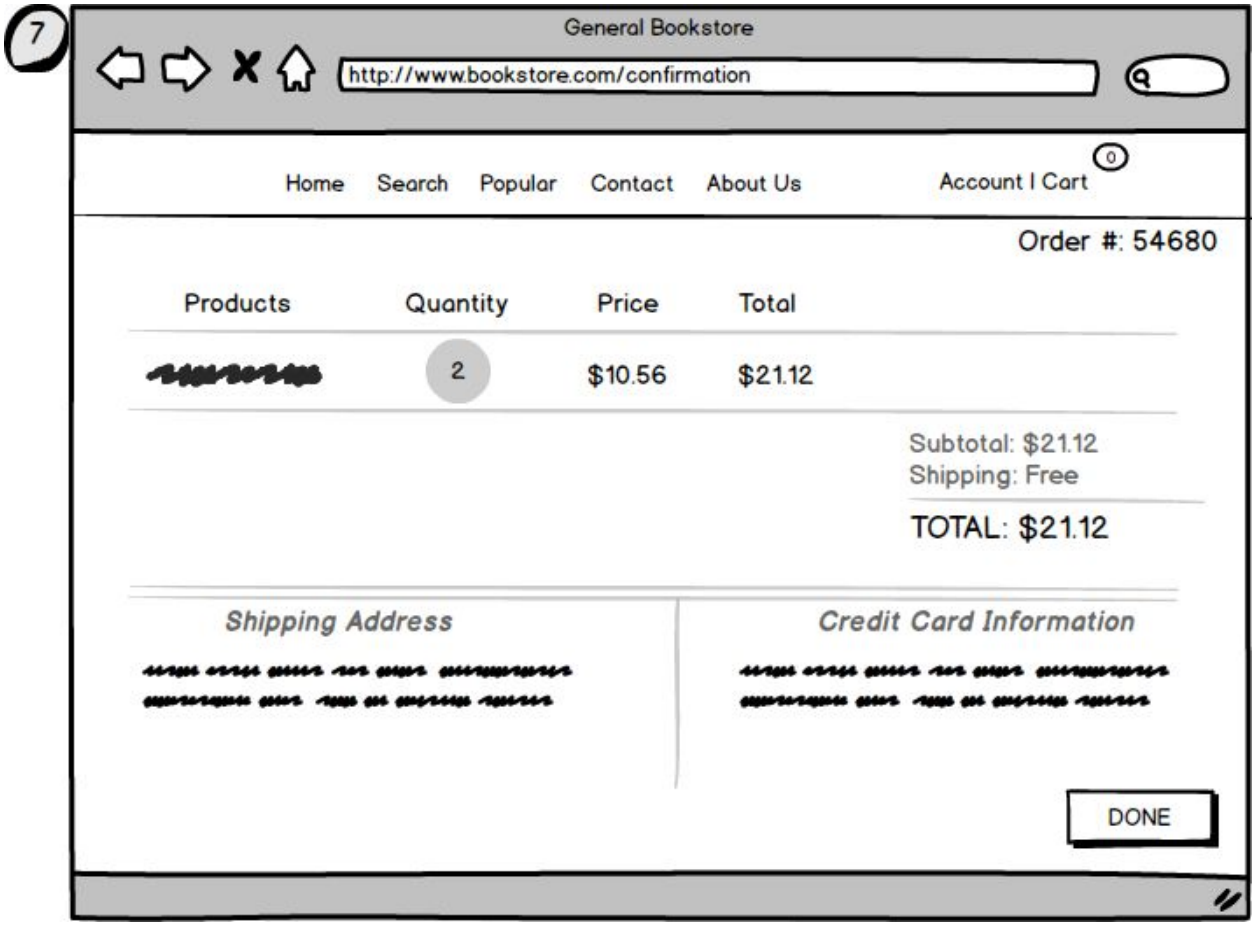


Figure 3.5.5.7 Confirm order screen

4. Glossary

- Publisher- in our system, a publisher another name for a vendor.
- Customer- a customer is both registered and unregistered users. Anyone external to our system is considered a potential customer.
- Genre- we made genre its own class because we believe that books can contain more than one genre. Therefore, for example, if a user searches for a drama then there are dramas of different types such as thrillers, comical, or adventurous.