Rainforest Books

Database Design and Web Application Development

Kevin Lanigan 10186146

Simon Melouah 10204367

**Introduction**

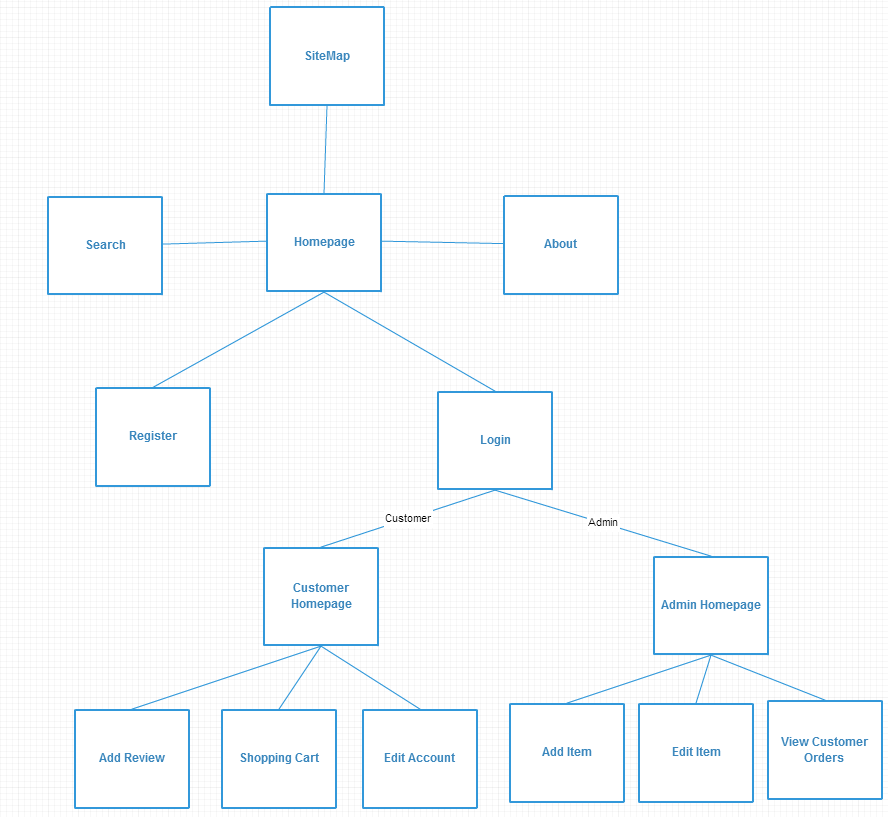
Rainforest Books is a website set up to meet the demands of eager readers around the world. The objectives were to provide a service that would both present an easy to use user interface as well as a robust solution to the online book market. This was achieved through a well-designed front end combined with simplistic, versatile functionalities which include:

* Customer register page
* User login page
* Site layout specific to the customer (after login)
* Persistent Shopping cart (customers)
* Edit credentials page (customers)
* Search for items
* Add review to a product (customers)
* Site layout specific to admin
* Add item page (admin)
* Edit item page (admin)
* View a list of customer orders (admin)

**Target Audience**

The service was aimed towards any individual who has a passion for reading but isn’t keen on spending an excessive amount on books. The emphasis on the reasonable sale prices also means that the website can be used for purchasing gifts as well. The website also sells book accessories (lights, bookmarks, covers, magnifying glasses etc.) so that the user can fully enhance their reading experiences.

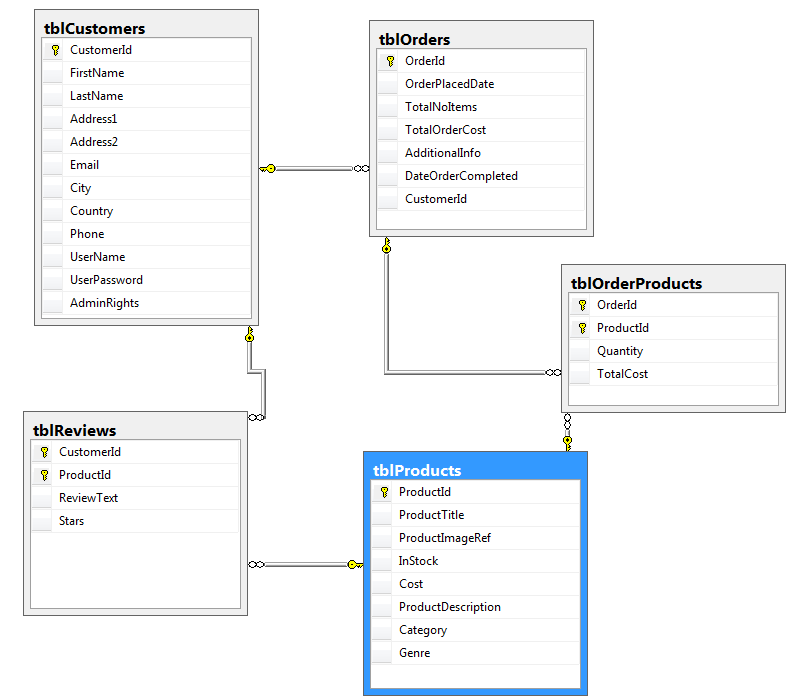
**Hierarchical Structure**



**Visual Design**

The site was conceived as an imaginary competitor to Amazon.com. It is the opinion of the creators that the Amazon site contains too much information and can be difficult to navigate. The design of this site has been inspired by the perceived flaws of the Amazon site. Rainforest Books employs a simple header and footer. The header contains a search bar, links to other parts of the site and a drop down menu. A minimalist sans-serif font (WeblySleek UI or Segoe UI) is employed for legibility and aesthetic appeal. The site has been designed with the intention of presenting a minimalist feel scrolling layout. The choice of background is used to elicit a tranquil pleasant shopping experience for users. Panels on the site employ a HTML 5 transparency feature which gives enough contrast to allow easy reading of the text they contain while allowing the user to see the background image at the same time. As bootstrap was employed to create this site, the container class which holds the main content of the site will never exceed a width 1170px and adjusts dynamically to ensure the site always presents well on any monitor or device.

**Database Design**



The database was designed to allow the site to store information about products, users, orders and reviews. To achieve that, 5 tables were created. Customers, Products can be considered the primary tables in the design as they both can exist independent of each other. The Order and Review tables are implemented to allow the Site to store and retrieve information about customer orders and customer ratings for products. In this design, a customer can only write one review per book. However as an order can contain many products and many products can be a part of many orders, it was necessary to create the table OrderProducts to account for this many-to-many relationship. The website interacts with this database through the use of LINQ queries.

## **Technologies:**

**HTML/aspx:** The website was created using Visual Studio 2013 community. It was set up as web forms project using ASP.NET. As such it uses code that runs on the server to generate web pages dynamically and is compatible with any browser or client device. It also allows for functions such as session states and cookies for model view control.

**CSS/ BootStrap:** The site was predominantly created using bootstrap components, however these components were further styled using a CSS style-sheet.

**Javascript:** To implement alerts on the site, C# calls clientside script (JavaScript) for occasions such as adding an item to a cart.

**JQuery:** Jquery is using in conjunction with bootstrap components for features such as the bootstrap carousel and dropdown menus.

**AJAX JSON:** This technology is implemented in conjunction with a LINQ query joining 3 tables to generate a list of orders for admin users.

**XML:** For persistent storage of the Shopping Cart, a list of cart items is serialised into XML format and then this is stored in a table in the database along with the userID and a reference number. The reference number is then stored as a cookie on the user’s computer. When the user logs in again, the reference number and the userId is used to retrieved the xml object of the list. This list is then de-serialised and is displayed in the shopping cart page for the user.

**AJAX Toolkit:** An attempt at implementing an autocomplete extender for the search box was made however this was ultimately unsuccessful and was removed from the project.

**SQL:** The database tables were initially created using SQL statements and these can be found in the project folder.

**Maintenance**

The site has been designed with the aim of being easily maintainable by an administrator with no web development knowledge. Administrator accounts are able to add and edit products on the website. It is the authors’ belief that this aim has not been fully achieved. A remove feature for products, which would be desirable, was not implemented, as it would cause issues with data integrity in the database. In a full version of the site, this function would have to be included to account for products that go out of production, or products that are no longer stocked. Also to give an account admin privileges, this has to be set in the database. This could be difficult for an admin without any knowledge of SQL Server Management Studio

For users, they are able to autonomously maintain their account details to ensure they have their most current addresses, and change their passwords if they choose.

**Search Engine Optimisation**

As the majority of web traffic is directed by the main search engines such as Google, Bing etc. (Although more is being directed by social media now). It is imperative that sites are optimized for this reality. Rainforest Books employs several of the following strategies to optimise its SEO rating.

* The content on Rainforest books is well written and well structured. This benefits both the website viewers and the search engines.
* Rainforest Books writes amusing book reviews which can result in people posting these reviews on social media. This will help the SEO rating and draw more users to the site. Similarly to this, the more links to the website from trusted websites will also improve its rating.
* In a fully produced site, it would also by the aim to post a weekly witty blog, book review, or write a post about an author or other website. This will encourage more links to the website
* For images on your site, be sure to complete all their attributes so the search engines can process them.
* As the internet famously has no attention span, each page of the site is not too long and is easy to scan for information.
* Each page contains relevant title and meta tags (description and keywords).
* H1 tags are used appropriately to highlight keywords.
* Each page of the site contains links to every other part of the site.
* The site has a html site map, it does not contain an XML site map as it hasn’t been hosted yet, however in a full version this would be generated for the site.

**Strengths and Weaknesses**

*Strengths-*

* The website is visually appealing and easy to navigate.
* Users can search for items or display a list of items relative to a given genre or title.
* Customers can register accounts and login.
* Customers can add items to a shopping cart and remove them accordingly.
* Customers can store their shopping cart for another session if they so choose.
* Customers can edit their account details.
* Customers can add reviews to an item along with a star rating out of five.
* Admins can add or edit items including uploading an image for an item, as well as adding a description. The pages for these functions are only available to admins.
* Validation is used to display warning messages to user attempting to add in duplicate details, invalid entries or leave details empty in forms.

*Limitations-*

Although the solution was visually appealing and relatively robust there were factors that would need to be addressed for a live webpage version.

* Firstly was the issue of the search bar not rendering properly on chrome and as a result of this had to be changed completely to a much wider search bar within its own region, allowing it appear the same in both browsers.
* Secondly was that although the user could register and edit their details on the site, they couldn’t delete their account as this would compromise data integrity.
* The layout for the star system when the customer was adding a review wasn’t visually appealing but worked functionally well. For the live web version each radio button would be represented by an image of star.
* For the shopping cart the customer has to click remove item then update shopping cart to see changes there is also no checkout available after they go to purchase. For a live version it would be made easier for the customer to add or remove items from the shopping cart, the payment would be done either through Paypal or another secure card payments scheme.
* When a customer registers, their username must be unique, although this only applies to customers and not customers and admins. Due to this, if a customer registers with a user name and password the same as that of an admin the webpage will not log them in. This could be resolved by having a separate login for the customer and admin or by having an admin table and not allowing the customer to have the same username as either an admin or another customer.
* The lack of an author credential and allowing the user to search by author is a small factor that would be addressed for a live version
* For the Edit Item page, the admin user has to search for the item/book they want to edit. In a live web version the “Add to Cart” link would be replaced with an Edit Item link for the admin view allowing them to use the default search bar rather than having a separate one for the Edit Item page.

**Individual Contribution:**

|  |  |
| --- | --- |
| Kevin | Relational database designed, SQL statements to create this design, wrote Insert SQL statements for test data |
| Kevin | Made the general master page, outlined the style format of the site using bootstrap |
| Simon | Created Add product page, implemented add product functionality |
| Simon | Created Edit product page, implement edit Product page functionality |
| Simon | Set up Register user page, implement add customer to database functionality |
| Simon | Form Validation on data entries |
| Kevin | Site Map |
| Kevin | Created and designed UserSession class, Shopping cart class, Admin Session Class implementing session state to keep data effectively. |
| Simon | Created Models for data obects in C# |
| Simon | Set up Context Class to allow db communication |
| Kevin | Ajax JSON call to allow admin to view a table of orders |
| Simon | Upload images for product upload |
| Kevin | Store Shopping cart persistently using XML object, database and cookies |
| Simon | Login implementation using database, password encryption |
| Kevin | Add reviews, delete reviews |
| Kevin | Product display, Review Display |
| Simon | Created hierarchical diagram |
| Simon | Search Engine Optimisation in website |
| Kevin | Attempted to implement Autocomlete extender using AJAX controlkit, but ultimately wasn’t successful |
| Kevin | Displaying random 6 products on homepage |