

Personal Portfolio 2

Andrew Cooper - n8911762

Overview

The following are descriptions and justifications for each of the artefacts included in my submission for the first personal portfolio. The five artefacts included are:

1. Book Information Suggestions using Google Books API
2. Search Suggestion Autocomplete & Animation for Home View
3. Process Design Document for Borrowing, Returning & Rating
4. Bug Report: Extensive Styling Bugs
5. Bug Report: Create Books Bugs

Artefact 1: Book Information Suggestions using Google Books API

Implementing book suggestions using the Google Books API was a great learning opportunity, and has contributed business value to the Texchange product through improved user experience for submitting books. Two source files are included for this artefact; `BooksController.cs` and `CreateBooksAutocomplete.js`.

There are three methods within the Books Controller that represent the artefact. These relate to querying the Google Books API, extracting the required data from the response, and packaging it into a JSON object that can be returned to the View for displaying. In order to complete this I needed to learning how to use Google Books APIs, and specifically the v1 Books API, and to set up a consumer Google Developer account.

The included JavaScript file includes jQuery required to use the jQuery UI interface library, for displaying the results of the query via customised autocomplete behaviour for forms. Completing this development task required learning jQuery and also researching asynchronous HTTP POST methods.

Artefact 2: Search Suggestion Autocomplete & Animation for Home page

Animation and autocomplete has also been added for the search field on the home page view. This provides business value in two ways. Firstly, it adds visual interest to the home page through animation. Secondly, it improves search functionality since suggested titles are actually available to be borrowed.

The animation was achieved using JavaScript timeout handlers, and is included in the included `AnimateSearchSuggestions.js` script. The placeholder attribute of the textbox is updated repeatedly, simulating typing in a title.

Implementation of autocomplete for the text area has capitalised on functionality provided in the jQuery UI interface library, and a controller method to return available titles. An important consideration for this functionality was whether the page should load synchronously with the titles list, or whether they could be loaded separately. Since they're both small tasks it probably doesn't make a significant difference, but loading asynchronously was chosen as an interesting challenge. In order to achieve this, some common book titles are hard coded to allow the page to load and the animation to commence, before the database-accurate list of book titles have returned. Once these are returned, the list of titles used in the animation is updated. Some careful consideration of timing was required to implement this in a safe manner.

See the included JavaScript files, `AnimateSearchSuggestions.js`, `AutocompleteSearchSuggestions.js` as well as the C# controller file `BooksController.cs` files for the relevant code.

Artefact 3: Process Design Document for Borrowing, Returning and Rating

The "Borrowing, Returning and Rating" process contributed to four important user stories. These are "Book Transaction Process", "Book Delivery", "Binary User Rating System" and "Token Management". Considering the extent of this process, it was not able to be solved in a short amount of time during a meeting. I documented a valid process before presenting it to the team as a solution to the many problems involved, including both UX flow (but not interface design) and software concerns.

The document included is my own work (no collaboration). The solution was pitched to Liza and Eric during a development team meeting, and we all agreed on some minor changes which included removal of the suggested "Notifications" column in the Users table (since this count

can be calculated from other database information). We agreed to use this slightly modified plan going forward.

Please see the included Process Design Document as evidence of my contribution to this part of the project.

Artefact 4: Bug Report: Extensive Styling Bugs

Although I fixed many bugs throughout Sprints 3 & 4 (see the [bug tracker](#) set up during Release 1), the styling bugs related to “Piggy Bank Glyph Not Showing” were particularly interesting and demonstrate more depth than some of the simpler bugs. For this reason a Bug Report has been compiled and included as Artefact 4.

Artefact 5: Bug Report: Create Books Bugs

In addition to the styling bugs mentioned in Artefact 4, there were some interesting considerations made for the fix to “Hidden Book Entry Fields” and “Pressing Enter on Book Submit Form” (see details in [bug tracker](#)), involving multiple languages and event handlers.

Please see the included Bug Report for Artefact 5.