

Portfolio 2

Eric Brieese – n8742197 – Group 85

Artefact 1 – Token Functions

The code extract in the file 'token functions.cs' was done by me to gives us functions for handling the user's tokens which are, in turn, used to control how many books a given user may borrow.

A user is given one token on registration. Once a user successfully borrows a book one token is subtracted. A user who successfully lends out a book is awarded one token. Tokens have a one-time use and the system is intended to encourage users to upload books to propagate the site and keep the wheels turning. It also gives us an opportunity to introduce revenue raising by selling additional tokens to user however this hasn't been investigated or researched.

The function *getTokens(string id)* allow us to get how many tokens a user has given their id so that we can determine if they can or cannot borrow a book.

The function *setTokens(string id, int quantity)* allows us to set a user's token count to any integer value we choose. It was going to be used to allow the admin to easily change a user's token count but that feature didn't make it into release 2.

The function *incrementTokens(string id)* and *decrementTokens(string id)* raise and lower the user's token count respectively. These functions are used when a user successfully borrows a book or lends a book.

Code can be found in `Controllers/AccountController.cs`

Artefact 2

This artefact is a script/storyboard for the team's release 2 presentation. This is an important document since it is a requirement of our presentation to demonstrate all features/stories that we had developed and so having a script with directions and chronological orders that would demonstrate a typical user's interaction with the site as well as touching on all of our stories makes sure that we have a planned and polished presentation.

Artefact 3

This artefact is a terms of service document that I wrote up. While I haven't explicitly studied this in law yet I did a little bit of reading and with inspiration for other websites and stuff, wrote up some of the more important parts that I could think of such as account suspension, copyright stuff, virtual currency (tokens), and use of user data and information. I also made a page for the terms of service on the site, added a link to the terms in the footer to show on every page and added a signification of acceptance of the terms to the registration page.

The terms can be seen in the file 'terms.cshtml'. This file can also be accessed from `Views/Home/terms.cshtml`

Artefact 4

During sprint 4 some troubles were found regarding error checking for data entered in fields during book creation and editing. The error checking was split between the model and the controller and was fragmenting which errors users were seeing depending on which errors were being caught. We

decided that to give us more flexibility over input control error checking we should move all of the checks to the controller.

The code extract in 'error checking.cs' shows the code I wrote to check the various inputs. The checks are fairly straightforward and involve things like making sure the year of a book can only be numbers and that certain fields can't be left empty.

This error checking function is used for both book editing and book creation and can be found (along with implementations) in Controllers/BooksController.cs

Artefact 5

This is similar to Article 4 in that it also involves error checking. As with book editing/creation we decided to move the errors for account registration into the AccountController. This code checks that the user hasn't entered empty fields, ridiculously long phone numbers (or numbers containing digits) or emails not from QUT students. The extract in file 'registration error checking.cs' shows this and the code can be found in some more context in Controllers/AccountController.cs