­­

**Modelling an Online Library System using Event-B**COMP1216: Software Modelling and Design

University of Southampton

May 2021

**Group 7**

Victoria Dimitrova (vd2n20)

Archie French (af9g19)

Jordan Harle (jsh1g20)

Yaqin Hasan (ykh1e19)

**Summary:**

As part of the COMP1216 module and as a continuation of the first group coursework where a design specification for an online library system was created using UML, the library system was then formally modelled using Event-B as the second group coursework. The primary requirements of the system are summarized as follows:

* Users can login with their password (multiple device logins are ignored) and have different permission levels which determine what they’re able to do within the system
* Books, websites and articles exist within the system as resources which have properties such as title, author and url.
* Reading lists can be created by lecturers within the system, which contain multiple resources.
* Books can be borrowed using a token system which limits the instances of that book being borrowed to the number of licenses it has, following which, users can request to be added to a first come first serve queue for that book.

**Solution:**

**REQ1**: The system manages a set of registered users

The variable users is a subset of the container set USER, it contains all registered users.

**REQ2**: The system should only allow a user to log in to the system using the correct password

This was achieved by mapping every user to a password during registration and checking this mapping whenever the user wants to log in.

**REQ3**: A logged in user can log out of the system

The users set contains a loggedIn and a loggedOut subset. The user will by default be placed in the loggedOut subset when they are registered and they will be moved to the loggedIn subset if they successfully login. They will be moved back to the loggedOut subset if they log out.

**REQ4 & 5**: Administrators and lecturers are special users of the system & A user cannot be an administrator and a lecturer at the same time

Every user has a discrete permission level between 1 and 4 inclusive. With 1 being student, 2 being lecturer, 3 being administrator and 4 being root administrator.

**REQ6**: The root user is a special administrator

It was assumed that the root user is created during initialization and would be used to create any additional users from then on.

**REQ7**: An administrator can register a new user with a password

It was henceforth assumed that only a logged in administrator can register a new user or perform any other administrative actions.

This also extended to all other users having to be logged in to perform any actions. However, they do not need to be logged in to have an action performed on them. For example, an administrator must be logged in to make USER1 a lecturer, but USER1 doesn’t have to be logged in.

**REQ8**: A logged in user can change their password

The mapping of user to their password will change accordingly

**REQ9**: An administrator can set the role of another registered user to administrator or lecturer

It was henceforth assumed that the root administrator had all the permissions of a normal administrator. This requirement was fulfilled using the SetAdmin and SetLecturer events.

**REQ10**: Resources are either books, websites or articles

All books, websites and articles are contained within the set *resources*, they all have *type* mappings to their *TYPE*, which can be either *BOOK*, *WEBSITE* or *ARTICLE*. This was done using a constant to make sure all resources can only have one type.

**REQ11 & 12 & 13**: Resources have their appropriate fields

This was achieved through mapping each resource to a set of field values.

**REQ14 & 15**: Only administrators can add or remove books & only lecturers can add or remove websites and articles

This was achieved through guard conditions in the events AddBook, RemoveBook, AddWebsite, RemoveWebsite, AddArticle and RemoveArticle.

**REQ16 & 17**: The system has utility functions for searching for resources by title & for searching for books by ISBN

This was achieved by defining the events GetResourceByTitle and GetBookByISBN which use the previously defined mappings of resources to their title and books to their ISBN

**REQ18 % 19**: Only lecturers can create reading lists & only a reading list’s creator can add or remove existing resources from it

This was achieved through the CreateReadingList event where each reading list is mapped to its creator. It was also assumed that a RemoveReadingList event would exist and only the reading list creator would be able to remove the reading list.

For the AddToReadingList and RemoveFromReadingList events, a guard condition specifying that the user requesting those actions must be the creator was also added.

**Borrowing Books:**

Whenever a new book is created using AddBook, the number of licenses it has is also specified and mapped to from the book. A token is created whenever a user requests to borrow a certain book. If the book is available, the number of licenses available will be decremented accordingly.

If the number of tokens created for a certain book is greater than or equal to the number of licenses the book has when a user requests to borrow it, the user is added to a first come first serve queue. When the book becomes available, it will automatically be lent to the first user in the queue.