MONASH BOOKS SYSTEM

DOCUMENTATION

Table of Contents

By Di Zhang

1 Design and Architecture

Monash Books System (MBSystem) is a web application that incorporates several Java EE technologies. It exposes both a main interface (for guests and patrons) and an administration interface (for administrators to maintain the system). The business logic for both interfaces is provided by enterprise beans. The enterprise beans use the Java Persistence API to create and store the application's data in the database. Figure 1 below illustrates the architecture of the application.

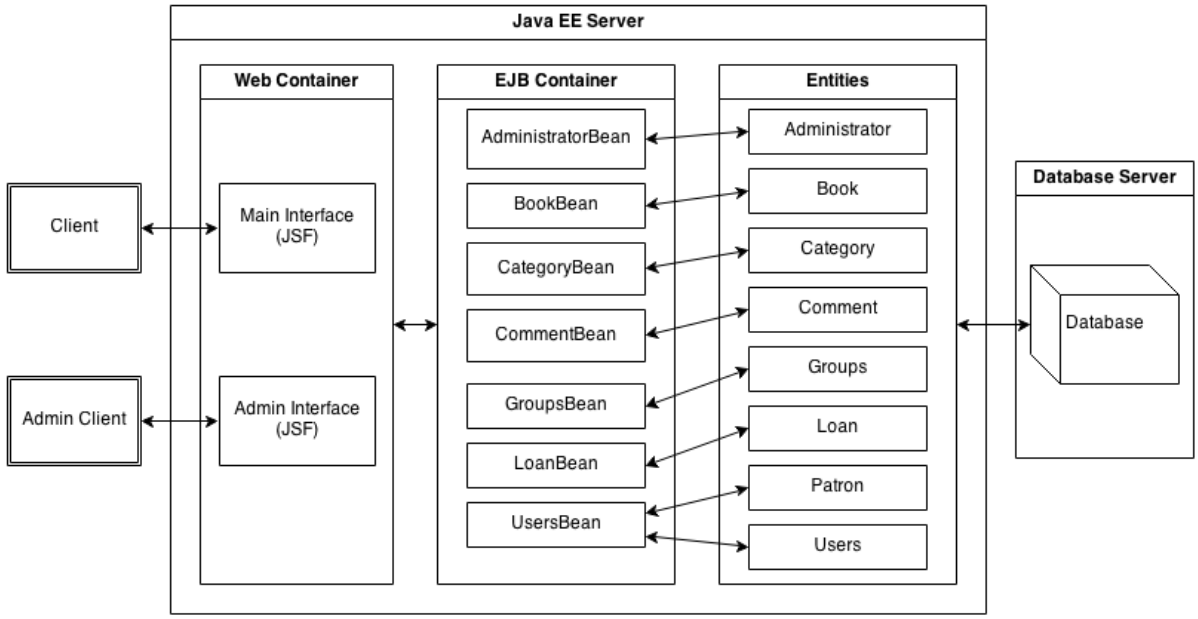

Figure 1: Architecture of the MBSystem Application

MBSystem uses the following Java EE 7 platform features:

- Java Persistence API entities:
  - Bean Validation annotations on the entities for verifying data
  - A standard property in the persistence.xml deployment descriptor to automatically and portably create and delete the tables in the data-source.
- Enterprise beans:
  - Local session beans
  - All enterprise beans packaged within the JAR file
- Contexts and Dependency Injection (CDI):

    - CDI annotations for Java Server Faces components
    - Qualifiers
- JavaServer Faces 2.2 technology, using Facelets for the web front end
    - Templating
    - Composite components
    - Security constraints on the administrative interface

MBSystem has two main user interfaces, both packaged within the WAR file:

- The main interface, for patrons and guests.
- The administrative interface used to perform back office operations.

1.1 Java Persistence API Entities

The entity classes map to the database schema shown in Figure 2. Each entity has validation rules based on business requirements, specified using Bean Validation.

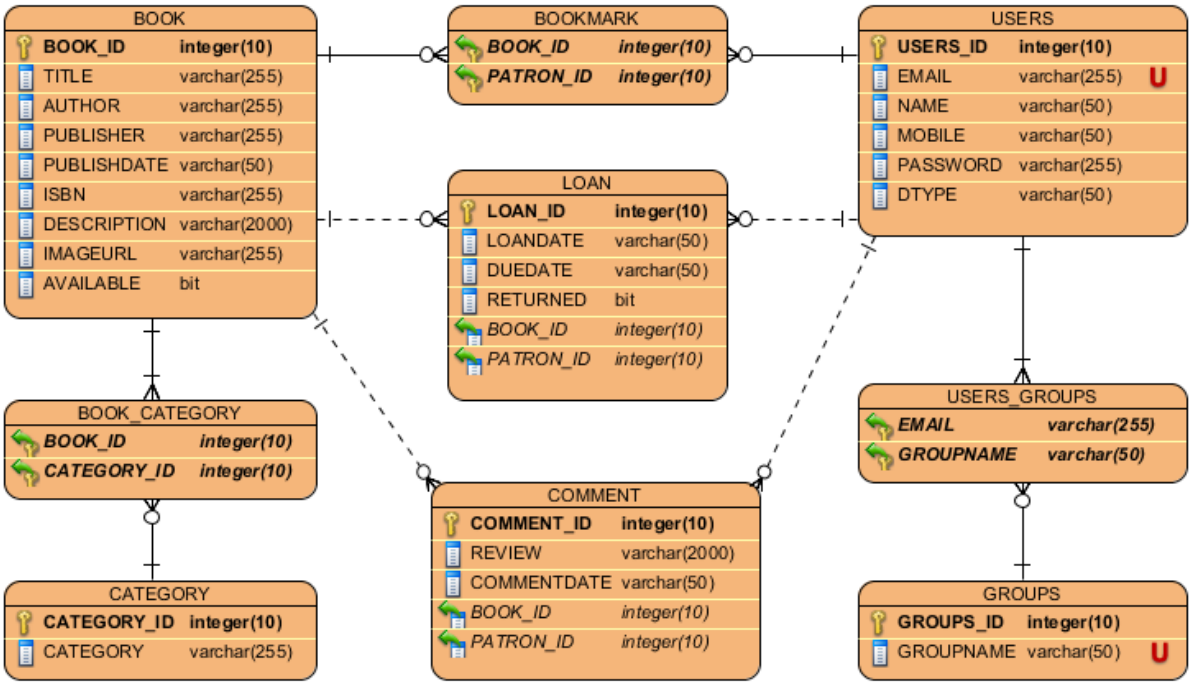

Figure 2: Entity Relationship Diagram

The database schema contains 9 tables:

- BOOK, which has a one-to-many relationship with BOOK_CATEGORY
- CATEGORY, which has a one-to-many relationship with BOOK_CATEGORY
- BOOK_CATEGORY, which has a many-to-one relationship with BOOK and CATEGORY (it is the join table between those two tables)
- USERS, which has a one-to-many relationship with USERS_GROUPS
- GROUPS, which has a one-to-many relationship with USERS_GROUPS
- USERS_GROUPS, which has a many-to-one relationship with USERS and GROUPS (it is the join table between those two tables)
- BOOKMARK, which has a many-to-one relationship with BOOK and USERS (it is the join table between those two tables)
- LOAN, which has a many-to-one relationship with BOOK and USERS
- COMMENT, which has a many-to-one relationship with BOOK and USERS

The entity classes that correspond to these tables are as follows (located in the mbsystem.entity package):

- Book, which defines attributes for books.
- Category, which defines attributes for book categories.
- Users, which defines attributes common to administrators and patrons. These attributes are the user's email, name, mobile number and password. The generated table for the Users entity also has a DTYPE field that represents the discriminator column. Its value identifies the subclass (Administrator or Patron) to which the user belongs.
- Administrator, a specialization of Users.
- Patron, a specialization of Users with fields for Book, Loan and Comment objects.
- Groups, which represents the group (USERS or ADMINS) to which the user belongs.
- Loan, which defines attributes for loans placed by patrons.
- Comment, which defines attributes for comments posted by patrons.

1.2 Enterprise Beans

MBSytem uses stateless session beans as façades for interactions with the JPA entities described above. The enterprise beans provide the business logic for the application and are located in the mbsystem.ejb package.

AbstractFacade is not an enterprise bean but an abstract class that receives a Type<T> and implements the common operations (CRUD) for this type, where <T> is a JPA entity.

Other beans extend AbstractFacade, inject the PersistenceContext, and implement any needed custom methods:

- AdministratorBean
- BookBean
- CategoryBean
- CommentBean
- GroupsBean
- LoanBean
- UsersBean

1.3 Interfaces

The main interface allows all users to perform the following tasks:

- Browsing the books
- Registering as a patron

It allows registered patrons to perform the following tasks:

- Logging in as a patron
- Adding and removing his/her bookmark for a book
- Requesting a book for loan and returning it
- Posting and deleting his/her comments for a book

- Viewing his/her personal profile

It also allows administrators to log in as an administrator and go to the administration interface.

The administration interface allows administrators to perform the following tasks:

- Book maintenance (create, view, update, delete)
- Category maintenance (create, view, update, delete)
- Patron maintenance (create, view, update, delete)
- Loan maintenance (view, update, delete)
- Comment maintenance (view, delete)

MBSystem uses CDI managed beans as controllers for interactions with Facelets pages. It follows the MVC (Model-View-Controller) pattern and applies the same pattern to all entities and pages.

1.3.1 Facelets Files Used in the Main Interface

The following Facelets files are used in the main interface:

template.xhtml

Template file for the main interface.

topbar.xhtml

Page for the login area in the header, at the top of the screen.

menubar.xhtml

Page for the menu area in the header, with navigation links and a search bar.

index.xhtml

Landing page for the main interface.

booklist.xhtml

Page displaying a list of book (this could be all books, books in a certain category, or book searching results).

bookdetails.xhtml

Page displaying the details of a book. For a logged in patron, it also shows components that allows the patron to add bookmark, request a book for loan and post comments.

login.xhtml

Login page specified in web.xml. The main login interface is provided in topbar.xhtml, but this page appears if there is a login error.

register.xhtml

Registeration page for new patrons.

profile.xhtml

Page displaying the profile of a logged in patron. It also shows components that allow the patron to remove existing bookmarks, return a book and delete existing comments.

error.xhtml

Error file if something goes wrong.

1.3.2 Facelets Files Used in the Administration Interface

The Facelets pages for the administration interface are located in the web/admin directory.

admin/template.xhtml

Template file for the administration interface.

admin/topbar.xhtml

Page for the link to the main interface at the top of the screen in the header.

admin/menubar.xhtml

Page for the navigation links in the header.

admin/index.xhtml

Landing page for the administration interface.

admin/book directory

Pages related to book management
(Create.xhtml, Edit.xhtml, List.xhtml, View.xhtml)

admin/category directory

Pages related to category management
(Create.xhtml, Edit.xhtml, List.xhtml, View.xhtml)

admin/patron directory

Pages related to patron management
(Create.xhtml, Edit.xhtml, List.xhtml, View.xhtml)

admin/loan directory

Pages related to loan management (Edit.xhtml, List.xhtml, View.xhtml)

admin/comment directory

Pages related to comment management (List.xhtml, View.xhtml)

1.3.3 Managed Beans

MBSystem uses the following CDI managed beans that interact with the necessary enterprise beans and Facelets pages to control the way the data will be displayed and manipulated. The beans are in the mbsystem.controller package.

- IndexController
    - Creating the default administrator, patron, categories and books when the application is initially started.
- BookController
- BookListController
- BookDetailsController
- CategoryController
- CommentController
- LoanController
- PatronController
- ProfileController
- UserController
    - Controlling the logging in and logging out of a patron or an administrator.

1.3.4 Helper Classes

The following helper classes, found in the mbsystem.util package, are usded in the main interface, the administration interface, or both of them:

CategoryConverter

A JavaServer Faces converter for the Category entity class. This class contains methods to convert Category instances to strings and back again, so they can be used in the user-interface components of the application.

DigestUtil

Class used by PatronController to generate an encrypted password for a patron.

JsfUtil

Class used for JavaServer Faces operations, such as queuing messages on a FacesContext instance.

PaginationHelper

An abstract class with methods used by the managed beans on the administration interface.

1.3.5 Qualifiers

MBSystem defines the following qualifier in the mbsystem.qualifier package:

@LoggedIn

Qualifies a user as having logged in.

2 Server Configuration - security realm

Configuration Name: server-config

Realm Name: JDBCAuthentication
Class Name: com.sun.enterprise.security.ee.auth.realm.jdbc.JDBCRealm

Properties specific to this Class

JAAS Context: *	jdbcRealm Identifier for the login module to use for this realm
JNDI: *	jdbc/sample JNDI name of the JDBC resource used by this realm
User Table: *	USERS Name of the database table that contains the list of authorized users for this realm
User Name Column: *	EMAIL Name of the column in the user table that contains the list of user names
Password Column: *	PASSWORD Name of the column in the user table that contains the user passwords
Group Table: *	GROUPS Name of the database table that contains the list of groups for this realm
Group Table User Name Column:	GROUPNAME Name of the column in the user group table that contains the list of groups for this realm
Group Name Column: *	GROUPNAME Name of the column in the group table that contains the list of group names
Password Encryption Algorithm: *	SHA-256 This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.
Assign Groups:	USERS,ADMINS

3 Running the Application

Running the application involves the following scenarios:

3.1 To browse the books

1. Build and Run the application in NetBeans IDE. The Monash Books | Home page opens and displays the latest books.
2. **Search** for books with title or author.
3. Click the **Book Collection** link in the menu to view all books. On the page that appears, click one of the categories to filter the books.
4. Clicking on the **image or title** of a book to view its details.

3.2 To register as a patron

1. Click **Register** button at the top-right corner of the page.
2. Fill in the form fields then click **Submit**. (All fields are required. The **Email** value must not be registered before, the **Mobile Number** value must be a valid Australian mobile number, and the **Password** value must be 6 to 20 characters in length.)

3.3 To bookmark, borrow and comment a book

1. To log in as the patron you created, or as the patron already in the database, enter the **email** and **password** and click **Log In**. The pre-existing user has the email di.zhang.com@gmail.com and password patron1.
2. Navigate to a book details page.
3. On the page that appears, click **Borrow** button to request the book for loan (the button only appears when the book is available); click **Bookmark** button to add bookmark (you can keep clicking the button but a bookmark for the book will only be added once); enter a review and click **Submit**.

3.4 To view own patron profile

1. After logging in, click your **Email** (green button) at the top-right of the page.

2. On the page that appears, you can view your personal information, your loans, your bookmarks and your comments. You can also return books, remove your bookmarks or delete your comments.
3. After you finish, you can click **Log out** button to log out your account.

3.5 To manage data as an administrator

1. Login in the same way that you did as a patron, only with email dzha73@student.monash.edu and password admin1. The application can tell whether you are an administrator or a patron.
2. If you logged in as an administrator, the **Administration Site** button appears at the top-right of the page. Clicking it will open the Administration | Home page.
3. On the administration landing page, your will see the links that lead you to pages that manage data for books, categories, patrons, loans or comments.
4. Click **Home** in the header to return to the administration landing page, or click **Main Site** at the top of the page to return to the main site. You can log out on the main site.

3.5.1 Loan maintenance

- Loans are only allowed to be created by a patron on the main interface. Administrators are only able to view, update and delete loans.
- When editing a loan, only the **Due Date** field and **Returned** field can be changed. If the due date is before now and the book is not returned, it shows “Overdue true”. The administrator can extend a loan by changing its due date. For example, if a loan is overdue today and you change the due date to tomorrow, it will show “Overdue false”. Whether the loan is overdue is also reflected on the patron profile on the main interface. To test this, try updating the due date to a time before now. It will show “Overdue” warning on the patron’s profile next to the loan information.

4 External Libraries and Resources used

Bootstrap v3.0.0

Copyright 2013 Twitter, Inc

Licensed under the Apache License v2.0

http://www.apache.org/licenses/LICENSE-2.0

jQuery JavaScript Library v1.10.2

http://jquery.com/

Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors

Released under the MIT license http://jquery.org/license

Date: 2013-07-03T13:48Z

Joda-Time 2.3 API

Licensed under the Apache License v2.0

http://www.apache.org/licenses/LICENSE-2.0

Released on 2013-08-16

5 Reference

Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito and Chinmayee Srivathsa (2013). The Java EE 7 Tutorial. Retreived from http://docs.oracle.com/javaee/7/tutorial/doc/home.htm