|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2018** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | 3D | | face-to-face |

|  |  |
| --- | --- |
| Project name | Books in the Spring |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The old libraries are so last season. Books in the Springprovides the opportunity to choose between various books, rent, read and then return them without having to leave the comfort of your own home. Your opinion matters as you can rate the books you read and we have regular wishlist polls to decide on our newest buys. You can meet many authors, critics and other book lovers at the events that we organize, or by joining one of our many clubs on location. If you have some books that you want to share with the world you can donate them in our special traveling book boxes and get some new and exciting books in return.  The system will be developed using ***Spring 5 Application Development Framework with Spring Boot 2***.The front-end will be a single page web application written in the ***Angular 2*** framework using JavaScript/ TypeScript. The backend will be implemented as a ***REST API using JSON data serialization***. The main user roles (actors in UML) are:  • *Anonymous User* – they can view the information pages and look at the book collections of the library, as well as other user’s profiles and use the traveling book boxes  • *Regular User* (extends *Anonymous User*) – they can borrow a book or take part in the events and clubs. Also the regular users have a personal wishlist that affects the polls, they can vote in the polls as well.  • *Librarian* – they can manage edit and delete user data. They are responsible for ordering the new books that were chosen in the regular polls. The administrators are also responsible for creating and managing the events and clubs. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and a password. | *Anonymous User;* |
| * 1. **Edit user profile** | *Regular Users* can view and edit their personal profile.  *Librarians* can view and edit profiles. | *Regular User, Librarian;* |
| * 1. **Manage Users** | *Librarians* can browse and filter users based on different criteria: first and last name or email. The *Librarian* can choose a user’s profile and view, edit or delete it. | *Librarian;* |
| * 1. **View other user’s profiles** | *Anonymous User* can see a base view of a user’s profile, showing the books that the user has read.  *Regular Users* can see the same as an *Anonymous User* plus the wishlist of that user, all the clubs that the user is in, as well as all events that the user is taking part in.  *Librarians* can see the same as *Regular Users.* | *Anonymous User; Regular User; Librarian;* |
| * 1. **Search books** | The library’s collection can be queries based on the book’s title, the author’s name or the rating given by other users. | *Anonymous User; Regular User; Librarian;* |
| **2.6. Request a book delivery** | The *Regular User* can request a book to be delivered to them at a specific location. | *Regular User;* |
| **2.7. Return a book** | The *Regular User* marks a book as finished and they arrange the returning of the book via the library’s delivery service. | *Regular User;* |
| **2.8. Rate/ comment a book** | The *Regular User* can leave a comment or rating on a book that they have read. | *Regular User;* |
| **2.9. Create an event** | The *Librarian* can create an event by filling a form. | *Librarian;* |
| **2.10. List all events** | The *Regular Users* and *Librarians* can see all events based on their start date, location and special guests. | *Regular User, Librarian;* |
| **2.11. View an event** | The *Regular Users* and *Librarians* can view details about an event. | *Regular User, Librarian;* |
| **2.12. Participate in an event** | The *Regular Users* mark themselves as participants in an event. | *Regular User;* |
| **2.13. Cancel an event** | The *Regular Users* that are participants in that event get an email that the event has been canceled.  The *Librarian* can cancel an event due to unpredicted circumstances. | *Regular User, Librarian;* |
| **2.14. Create a club** | The *Regular Users* can create a club by filling a form. | *Regular User;* |
| **2.15. Search clubs** | The *Regular Users* and *Librarians* can search clubs based on their name and topic. | *Regular User, Librarian;* |
| **2.16. Join a club** | A *Regular User* can join a club, which allows them to see the other members and the schedule of the club’s meetings. | *Regular User;* |
| **2.17. View a club that you have already joined** | The *Regular User* can see the club’s schedule and members. | *Regular User;* |
| **2.18. Manage a club** | The *Regular User* that created the club and the *Librarian* can delete it, update its schedule or remove some of its members. | *Regular User - creator of the club; Librarian;* |
| **2.19. Update your wishlist** | The *Regular User* can add or delete books to their own wishlist. | *Regular User;* |
| **2.20. Create a poll for adding new books** | The *Librarian* can create a poll of the 10 most requested books from the users’ wishlists. The 5 books with the most votes will be added to the library’s collection. | *Librarian;* |
| **2.21. Vote in the ongoing poll** | The *Regular User* can vote for up to 3 books in an ongoing poll. | *Regular User;* |
| **2.22. Order new books** | The *Librarian* orders new books and when they arrive adds them to the system. | *Librarian;* |
| **2.23. Add a book to a traveling book box** | The user puts a book in the box, and adds it in the inverotry via the box’s interface. | *Anonymous User; Regular User; Librarian;* |
| **2.24. Take a book from a traveling book box** | The user takes a book from the box and marks it as taken via the box’s interface. | *Anonymous User; Regular User; Librarian;* |
| **2.25 Make notes** | The user can make notes for the book he reads- they can be private or public. If they are public they can be seen from other users. | *Regular User;* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register. Has a search bar for search books. | / |
| * 1. **Profile** | Displays the user’s profile, and the books that the user is currently reading. The user can see other users profiles by searching for them in a search bar. If the user is viewing another user’s profile all editing functions are disabled. | */profile/{userId}* |
| * 1. **Book details** | Shows a book cover, short descriptions, rating and the comments made by users. | */books/{bookId}* |
| * 1. **Events** | Shows a listing of all future events and a search form for filtering events. All events have a button for subscribing to, or a button for canceling the subscription if the user is already subscribed to them. The *Librarian* can add a new event via a button that leads to a form for creating an event. | */events* |
| **3.5 Show event details** | Shows description of the event, time and location, as well as list of users that are subscribed to the event. | */events/{eventId}* |
| **3.6. User Registration** | Presents a view allowing the *Anonymous Users* to register. | */register* |
| **3.7. Login** | Presents a view allowing users to login. | */login* |
| **3.8. Clubs** | Lists all clubs with a filtering form. The clubs that the user has joined have a view button and all others have a join button. | */clubs* |
| **3.9. Club details** | Presents the club details: A short description, list of members and a schedule for the club’s meetings. The page also contains a button for canceling the user’s participation in that club. | */clubs/{clubId}* |
| **3.10. Wishlist** | Shows the wishlist of some user, if this is the user’s personal wishlist they can edit it by adding or removing books from it. | */profile/{userId}/wishlist* |
| **3.11. Users** | Shows all users. The librarian can delete the user’s profile with the delete button and edit their profile with the edit button. | */users* |
| **3.12. About** | Presents information about the *Books in the Spring* library. | */about* |
| **3.13. Polls** | If opened by a *Regular User* allows for voting on the current poll, while presenting live results for it.  If opened by a *Librarian* they can see the results of the last/ current poll. They can also add a new poll. | */poll* |
| **3.14. Traveling books box interface** | Lists all books that are in the box. Allows for adding and taking books from the box via buttons. | */box* |
| **3.15. Adding books** | Opens a form for adding a new book that the library has bought, or when adding a new book in the traveling books box. | */books/add* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Spring 5 with Spring Boot 2 Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET - List all users based on filters (only for *Librarians*) | */users* |
| * 1. **User** | GET - Get profile  PUT - Edit profile  DELETE - Delete the user’s profile (only for *Librarians*) | */profile/{userId}* |
| **4.3. Register** | POST - Registers a new user | */register* |
| **4.4. Login** | POST - User Credentials (e-mail address and password) and receive a valid Security Token to use in subsequent API requests. | */login* |
| **4.5. Logout** | DELETE - Logout request for ending the active session. | */logout* |
| **4.6. Events** | GET - List all events based on filters  GET /{eventId} - Get event details  DELETE /{eventId} - Cancels an event (only for *Librarians*)  POST /{eventId}/subscribe - Subscribe to an event  POST /{eventId}/unsubscribe - Unsubscribe from an event | */events* |
| **4.7. Clubs** | GET - Lists all clubs  GET /my-clubs/{userId} - Lists all clubs that the user is part of  PUT /{clubId} - Updates information about the clubs’ schedule/ members (only for *Librarians* and the *Club’s creator*)  DELETE /{clubId} - Deletes the club (only for *Librarians* and the *Club’s creator*)  POST - Creates a club  POST /{clubId}/join - Adds the user to the members of the club  POST /{clubId}/leave - Removes the user from the clubs member list | */clubs* |
| **4.8. Wishlist** | GET - Show the wishlist of the user  PUT - Send the modified list when editing | */profile/{userId}/wishlist* |
| **4.9. Books** | GET - List all books based on params  GET /{bookId} - View book details  POST - Add a new book (only for *Librarians*)  POST /{bookId}/rate - Leave a rating for a book  POST /{bookId}/comment - Leave a comment for a book | */books* |
| **4.10. Traveling books box** | GET /{boxId}/books - List all books in the current box  GET /{boxId}/books/{bookId} - View book in the current box  DELETE /{boxId}/books/{bookId} - Takes a book from the current box  POST /{boxId}/books - Add a new book to the box | */box/* |
| **4.11. Polls** | GET - Show poll results  POST /{bookId} - Vote for a given book | */poll/* |