QA INTERVIEW TASK

CONFIDENTIALITY / DISCLAIMER

This document is confidential. Immedis makes no warranties, express, implied or statutory as to the information in this document.

Complying with all applicable copyright laws is the responsibility of the candidate. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Immedis.

TABLE OF CONTENTS

1 PROJECT DESCRIPTION

1.1 Login

1.1.1. Description

As a user I want to be able to login into the system using correct username and password. If the credentials provided are not correct, the system should return a proper error message. If the credentials are correct, once the user press the “Sign In” button, the user should be redirected to the “Users” screen.

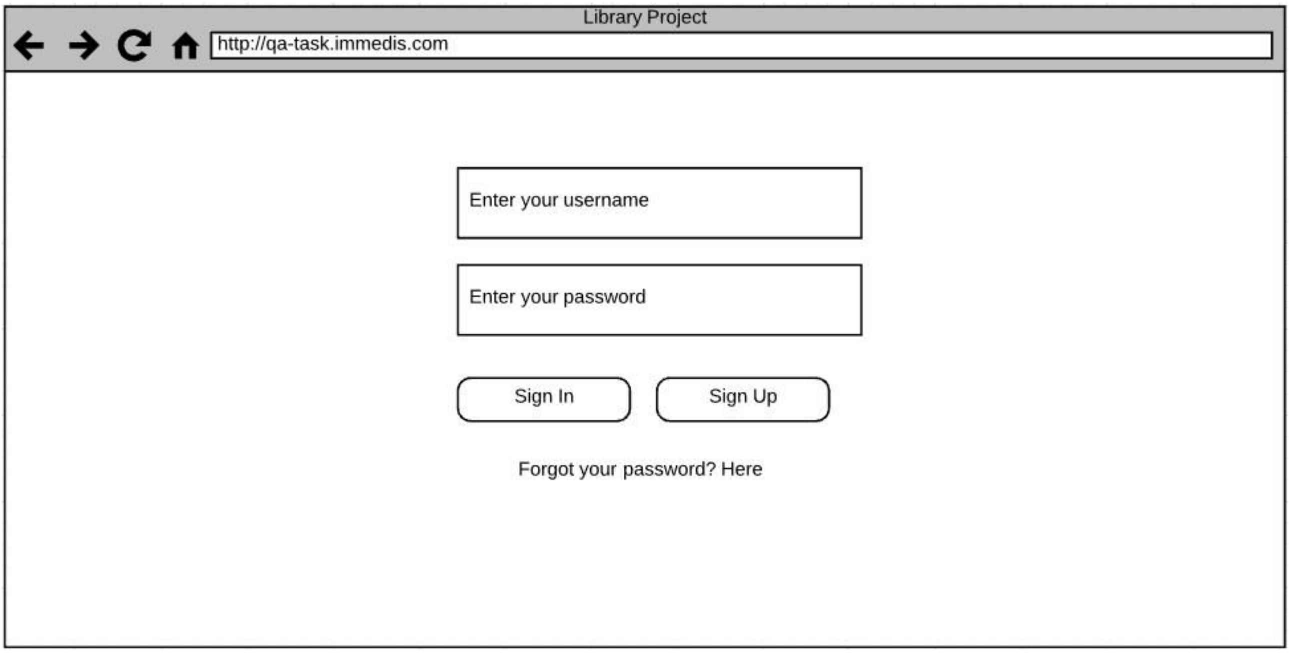
The screen should have the following elements:

- “Username” field;
- “Password” field;
- “Sign In” button;
- “Sign Up” button;
- Hyperlink to the “Forgotten password” screen.

By clicking on the ‘Sign Up’ button, the user should be redirected to the “Sign Up” screen where he can create a new account. (Note: All new accounts are with status ‘Pending approval’. Only library administrators can approve the new accounts)

By clicking in the “Here” word, the user will be redirected to the “Forgotten password” screen. He will be asked to provide his email address. If it exists, the system will send an email to the provided email address. If not, a proper error message will appear.

1.1.2. Mock-ups



1.1.3. Acceptance Criteria

If the provided credentials are correct, the user will be redirected to the “Users” screen.

If the username is correct, but the password not – a proper error message will appear: “The provided credentials are incorrect.”

If the credentials are incorrect, a proper error message should appear: “The provided credentials are incorrect.” By clicking on the “Sign Up” button, the user will be redirected to the “Registration” page.

By clicking on the forgotten password link, the user will be redirected to the “Forgotten password” screen.

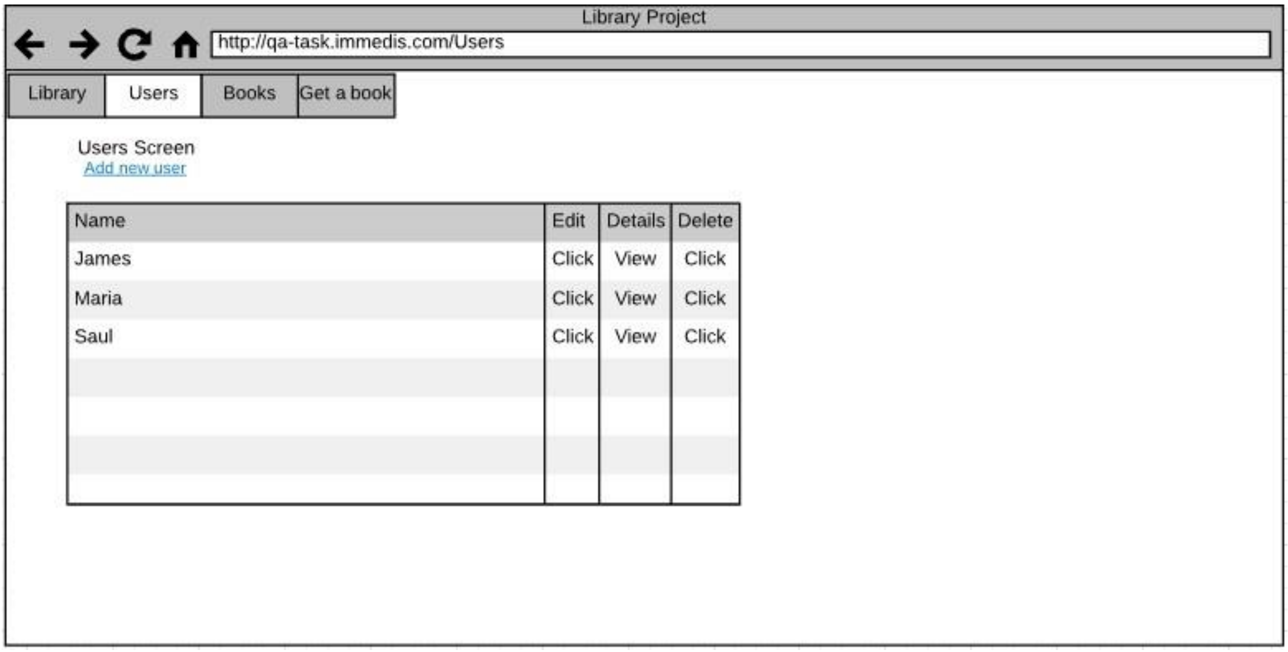
1.2. Users Module

1.2.1. Description

As a system user with proper rights, I want to be able to see the “Users” module. This module should contain the following functionalities:

- Listing all existing users – it should be part of the main screen. The list should contain all existing library users (not to be mistaken with the system users) in the system. It should be placed in a table. The first column should be with the names of the users. The following columns should give the ability to edit, view or delete user;
- Creating new users – A button should be placed on the main screen above the users table. By clicking on that button, the system user should be redirected to a new screen where he can add new user.
  - Validations:
    - The name of the user should up to 100 symbols. An appropriate error message should appear if the entered name exceeds that limitation;
    - An appropriate error message should appear if the “Name” field is empty.
- Editing existing users – the system user will have the ability to edit users by clicking on the “Edit” button from the list or redirecting to “/Users/Edit/<user_id>”. The same validation for the “Name” field from the “Create new user” screen is applicable here;
- Deleting existing users – the system user will have the ability to delete users from the users list. By clicking on the button, the user will be asked if he wants to delete that record. If he confirms that action, the record will be deleted and will no longer exist;
- Ability to view the user’s details – the system user will have the ability to view the user’s details by clicking on the “Details” button. He will be redirected to a new screen with the details of the selected user.

1.2.2. Mock-Ups

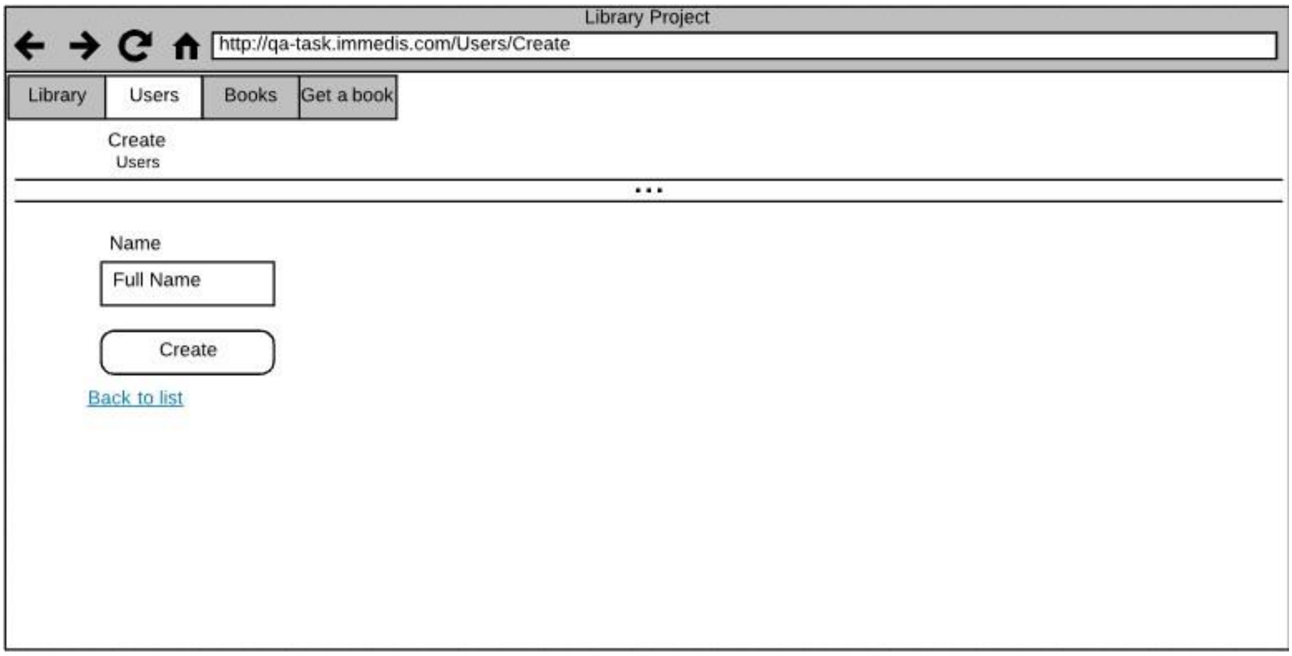


Users Screen

Add new user

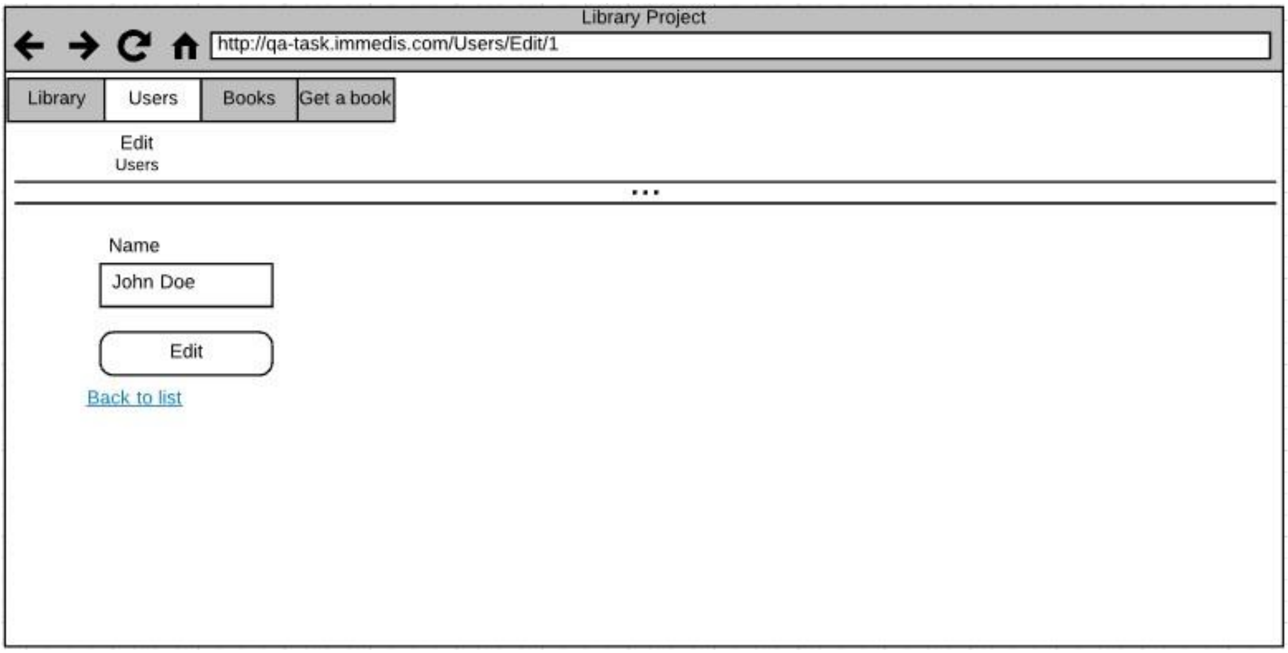
Name	Edit	Details	Delete
James	Click	View	Click
Maria	Click	View	Click
Saul	Click	View	Click

Project Description



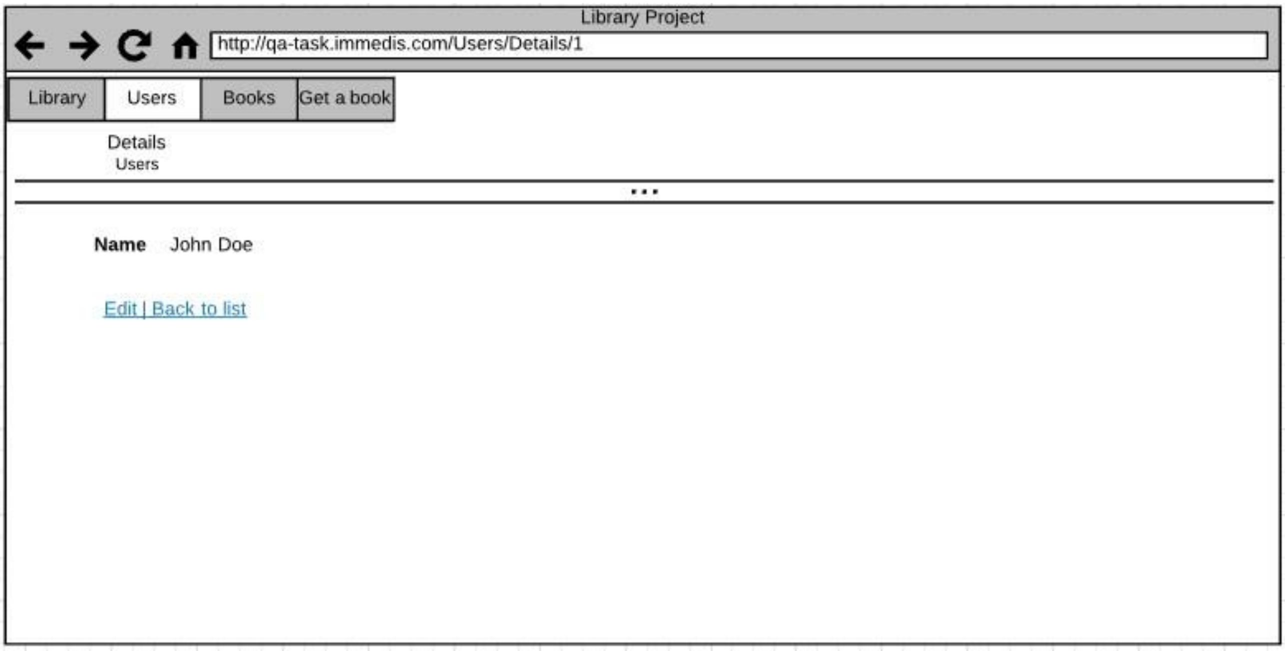
http://qa-task.immedis.com/Users/Create

Full Name



http://qa-task.immedis.com/Users/Edit/1

John Doe



Details
Users

Name John Doe

Edit | Back to list

1.2.3. Acceptance Criteria

By clicking on the "Users" button from the main menu, the system user should be redirected to the users list.

The list should be populated with all users available in the system.

The system user should have the ability to create, edit, view or delete users.

When creating or editing users, the system user should provide a correct name in the "Name" field.

If the "Name" field is left empty, appropriate error message should appear.

When deleting a user, the system should ask for confirmation. If the system user confirms the action, the library user should be deleted.

1.3. Books Module

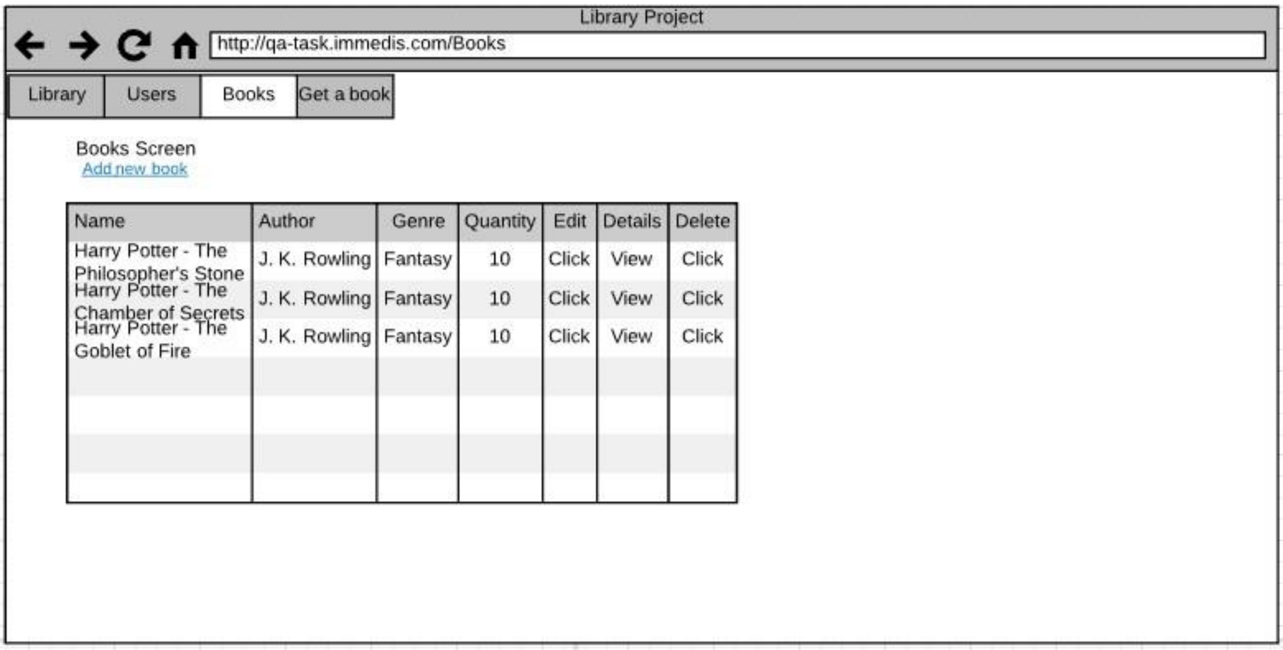
1.3.1. Description

As a system user with proper rights, I want to be able to see the "Books" module. This module should contain the following functionalities:

- Listing all existing books – it should be part of the main screen. The list should contain all existing books in the system. It should be placed in a table. The first column should be with the names of the books, following by "Author", "Genre" and "Quantity" (used to show how many available books are left). The other columns should give the ability to edit, view or delete user;
- Creating new books – a button should be placed on the main screen above the table. By clicking on that button, the system user should be redirected to a new screen where he can add new books.
  - Validations:
    - The name of the book should up to 250 symbols. An appropriate error message should appear if the entered name exceeds that limitation;
    - The name of the author should be up to 100 symbols. An appropriate error message should appear if the entered name exceeds that limitation;
    - The genre should be up to 50 symbols. Appropriate error message should appear if the entered value exceeds the field's limitation;
    - An appropriate error message should appear if one of the fields is empty.

- Editing existing books – the system user will have the ability to edit books by clicking on the "Edit" button from the list or redirecting to "/Books/Edit/<book_id>". The same validation for the fields from the "Add new book" screen is applicable here;
- Deleting existing books – the system user will have the ability to delete books from the books list. By clicking on the button, the user will be asked if he wants to delete that record. If he confirms that action, the record will be deleted and will no longer exist;
- Ability to view the books' details – the system user will have the ability to view the details of a book by clicking on the "Details" button. He will be redirected to a new screen with the details of the selected book.

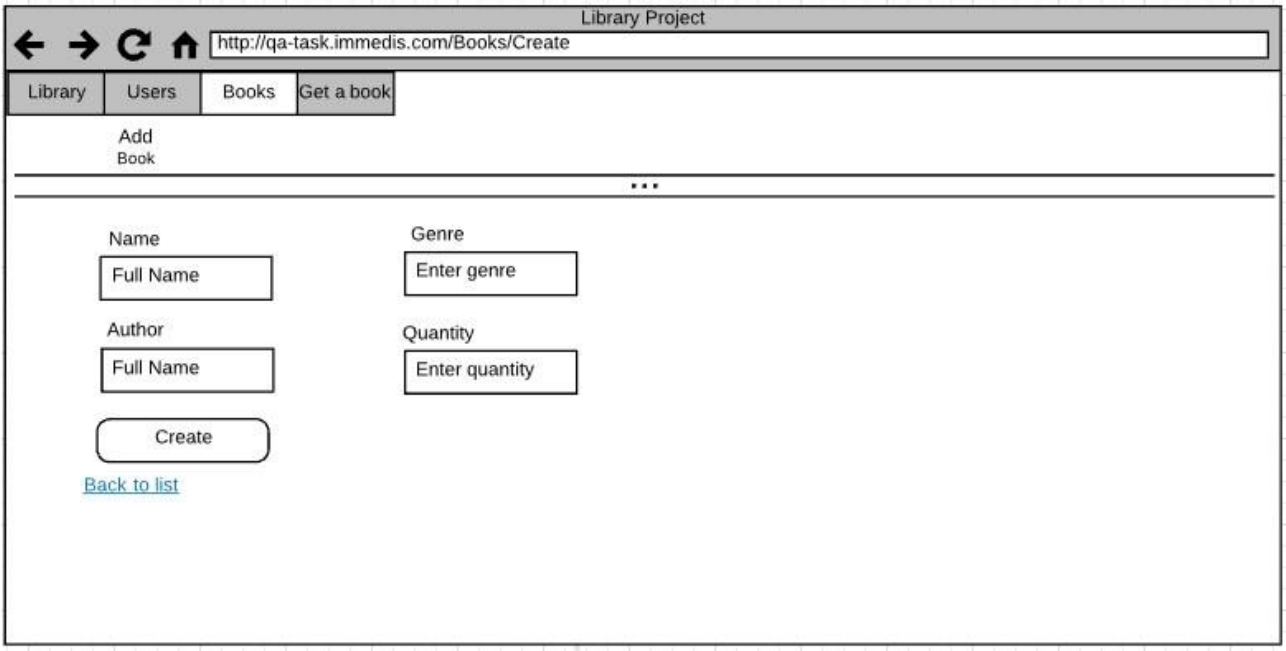
1.3.2. Mock-Ups

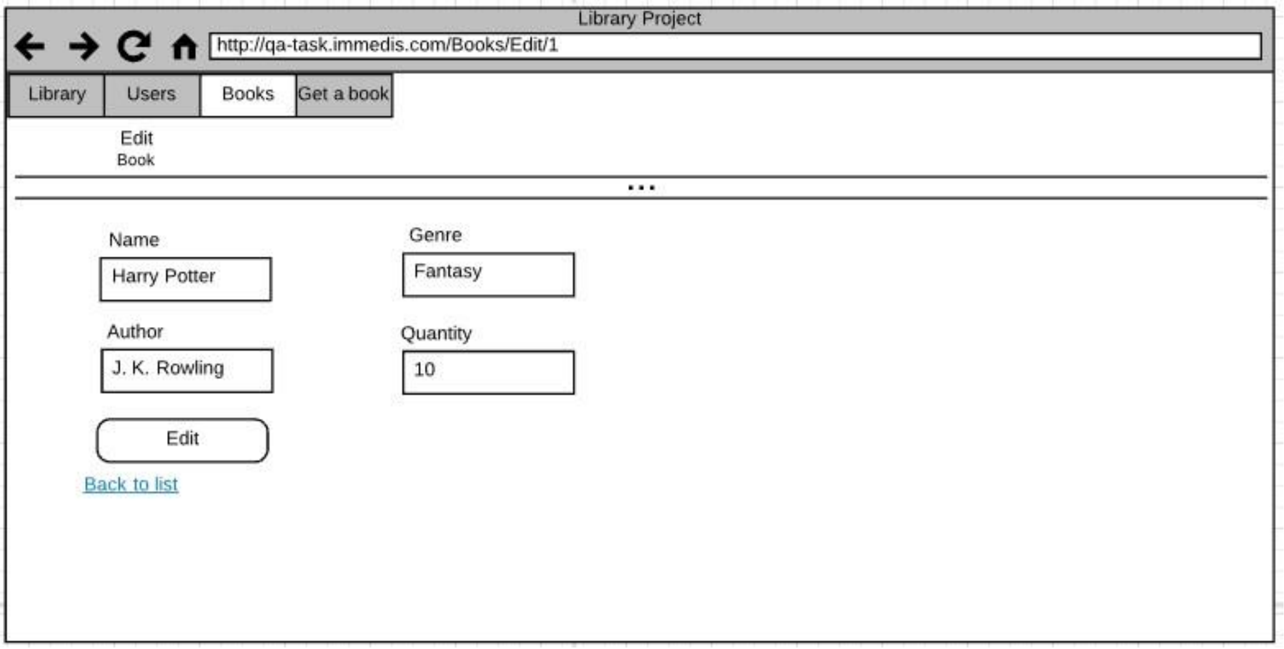


Books Screen

Add new book

Name	Author	Genre	Quantity	Edit	Details	Delete
Harry Potter - The Philosopher's Stone	J. K. Rowling	Fantasy	10	Click	View	Click
Harry Potter - The Chamber of Secrets	J. K. Rowling	Fantasy	10	Click	View	Click
Harry Potter - The Goblet of Fire	J. K. Rowling	Fantasy	10	Click	View	Click



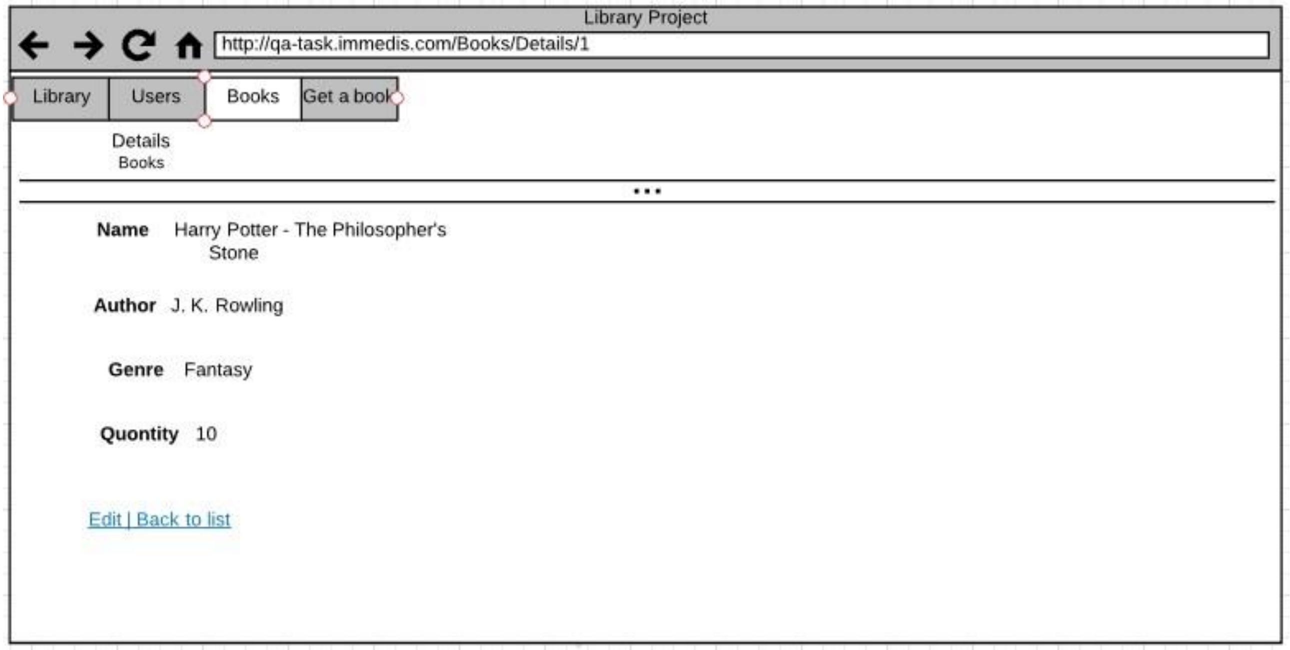


Name: Harry Potter

Genre: Fantasy

Author: J. K. Rowling

Quantity: 10



Name Harry Potter - The Philosopher's Stone

Author J. K. Rowling

Genre Fantasy

Quontity 10

1.3.3. Acceptance Criteria

By clicking on the “Books” button from the main menu, the system user should be redirected to the books list.

The list should be populated with all books available in the system.

The system user should have the ability to create, edit, view or delete books.

When creating or editing books, the system user should provide a correct name in the “Name” field, correct name of the author, genre and quantity.

If the one or more of the fields are left empty, appropriate error message should appear.

When deleting a book, the system should ask for confirmation. If the system user confirms the action, the book should be deleted.

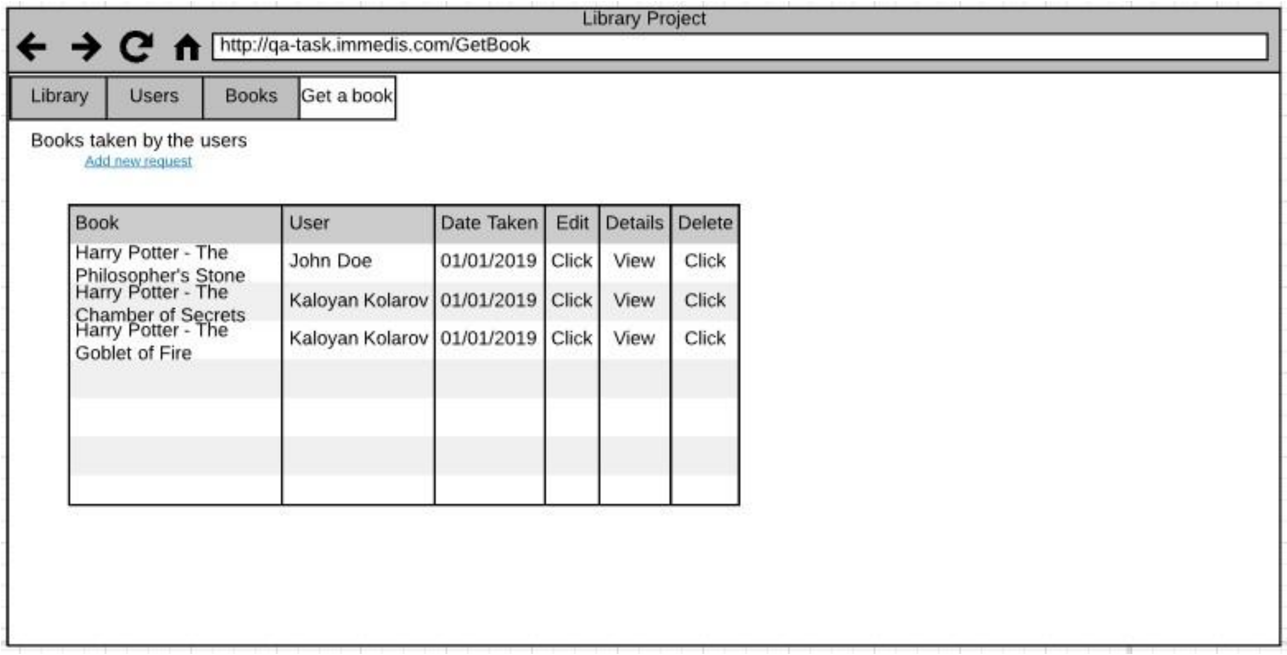
1.4. Get a Book Module

1.4.1. Description

As a system user with proper rights, I want to be able to see the "Get a Book" module. This module should contain the following functionalities:

- Listing all books taken – it should be part of the main screen. The list should contain all books taken by the existing users. The table contains the following columns: ○ "Book" – the name of the book; ○ "User" – the user that reserved that book; ○ "Date Taken" – the date when the book was taken; ○ "Actions" – for editing, viewing and deleting a record;
- Add new request – a button should be placed on the main screen above the table. By clicking on that button, the system user should be redirected to a new screen where he can add a new request. The system user should be able to select a book from a drop-down which will contain all existing books. The same applies for the user. Once the request is saved, the book's quantity should be decreased;
- Editing existing request – the system user will have the ability to edit requests by clicking on the "Edit" button from the list or redirecting to "/GetBook/Edit/<id>". Here the user will be able to change the book or the user's name;
- Deleting existing record – the system user will have the ability to delete requests from the list. By clicking on the button, the user will be asked if he wants to delete that record. If he confirms that action, the record will be deleted and will no longer exist. The quantity of the book should be increased by 1;
- Ability to view the requests details – the system user will have the ability to view the details of the request by clicking on the "Details" button. He will be redirected to a new screen with the details of the selected record.

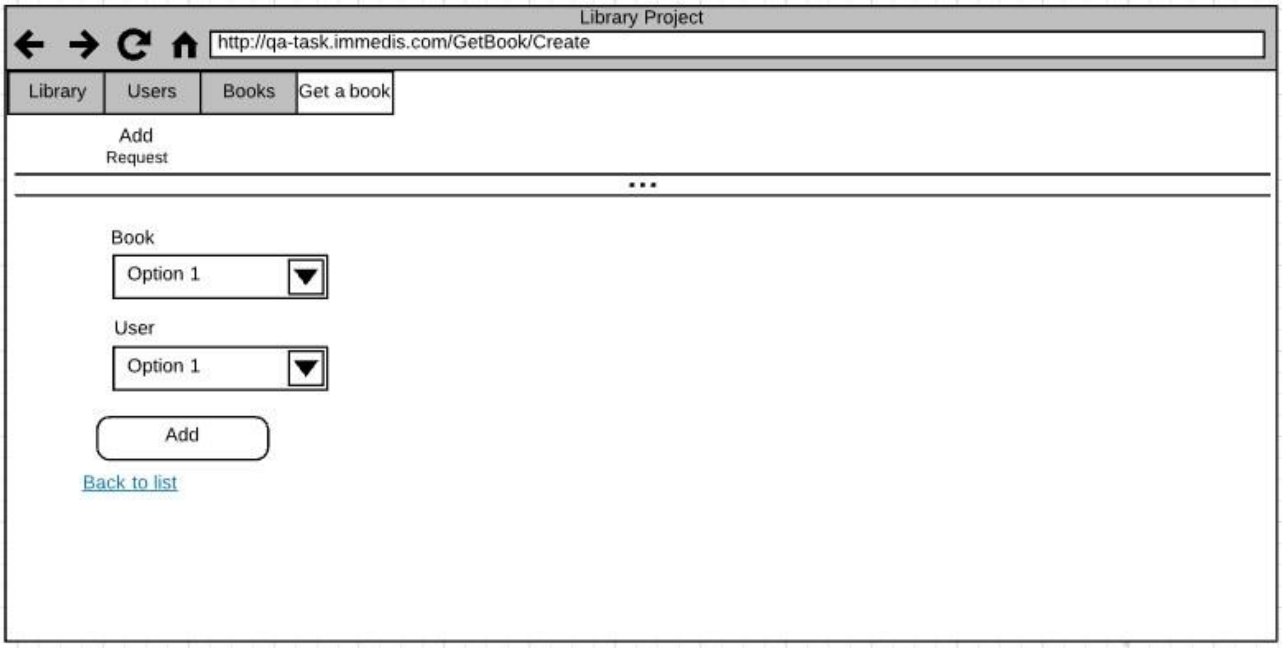
1.4.2. Mock-Ups



Books taken by the users

Book	User	Date Taken	Edit	Details	Delete
Harry Potter - The Philosopher's Stone	John Doe	01/01/2019	Click	View	Click
Harry Potter - The Chamber of Secrets	Kaloyan Kolarov	01/01/2019	Click	View	Click
Harry Potter - The Goblet of Fire	Kaloyan Kolarov	01/01/2019	Click	View	Click

Project Description



Library Project

http://qa-task.immedis.com/GetBook/Create

Add

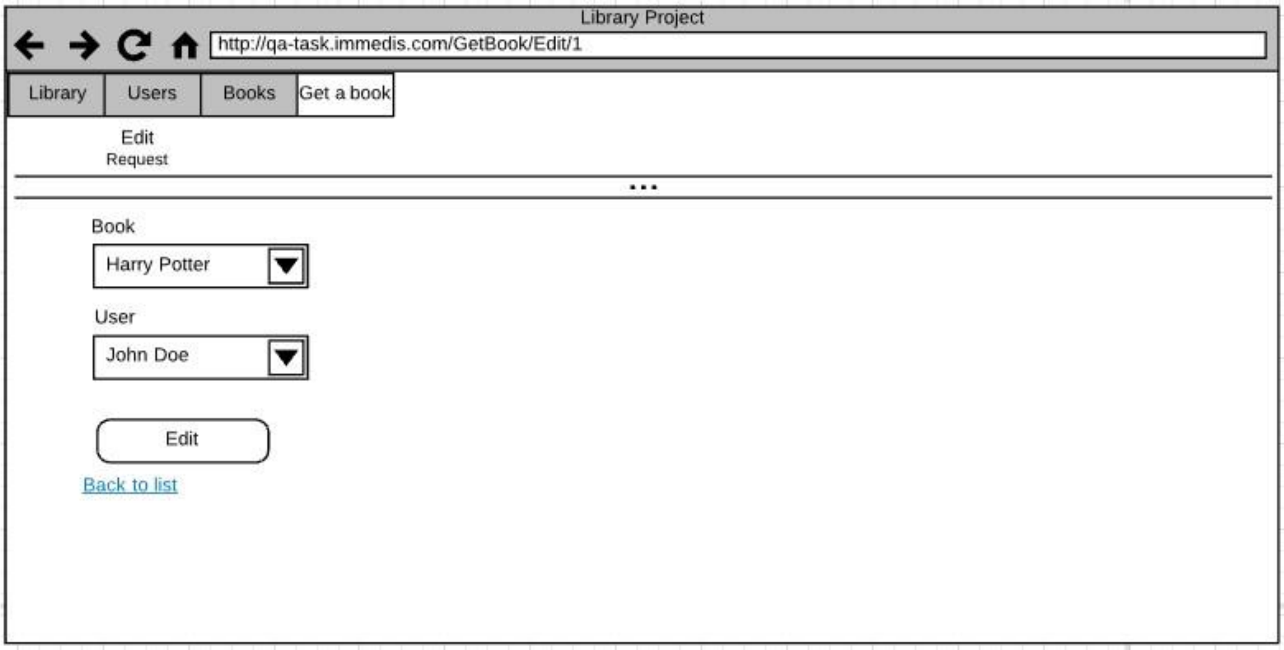
Request

Book

Option 1

User

Option 1



Library Project

http://qa-task.immedis.com/GetBook/Edit/1

Edit

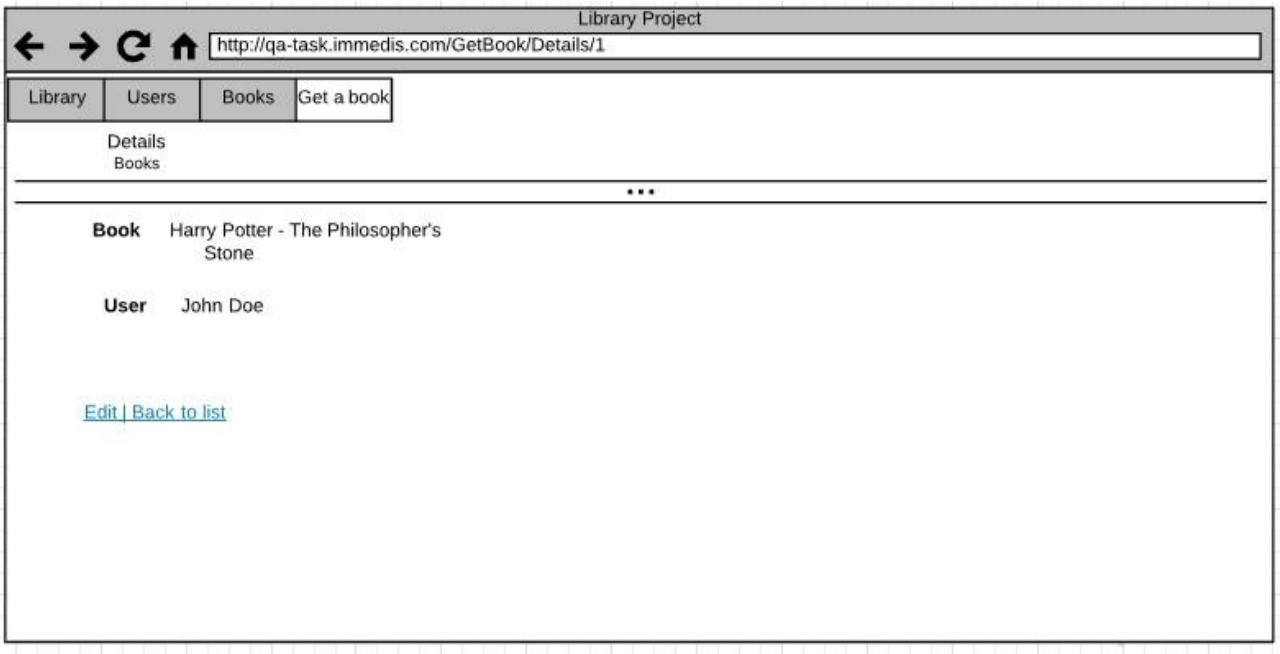
Request

Book

Harry Potter

User

John Doe



Book Harry Potter - The Philosopher's Stone

User John Doe

Edit | Back to list

1.4.3. Acceptance Criteria

By clicking on the “Get a Book” button from the main menu, the user should be redirected to the module and presented with a list with all requests.

The list should contain all existing requests.

When adding a new request, the system user should select a book and a user. Once added, the quantity of that book should be decreased.

When editing a request, the system user should have the same options. The same logic is applied for that.

When deleting a request, the system should ask for confirmation. If the system user confirms the action, the record should be deleted. Once deleted, the book’s quantity should be increased.

2. MANUAL RELATED TASKS

2.1. Test Plan Preparation

Prepare a test plan and test cases in Excel to cover the functionalities from above.

2.2. Test Case Execution and Bug Tracking

Review the site, execute the test cases and give your feedback about the design, validations and other issues which you have found. Please, describe at least three of the issues like you are reporting them to a developer in a bug tracking system.

3. TECHNICAL RELATED TASKS

3.1. UI Testing

Select the test cases which you think are good for regression testing from the tasks above and create a test scripts using Selenium WebDriver. Please, specify which programming language, framework, WebDriver version and browser you will use for this type of testing. Please, attach the project/scripts in your reply. If you find something, please describe it like you are reporting an issue in a bug tracking system.

3.2. API Testing

The system has an API underneath for communication with the database. Review the existing API and prepare test cases (requests) using Postman or similar tool for API testing. If you find something, please describe it like you are reporting an issue in a bug tracking system.

4. ENVIRONMENTS AND CREDENTIALS

- URL: https://qa-task.immedis.com/
- Username: admin
- Password: 123456
- API Documentation: https://documenter.getpostman.com/view/8102633/SWTHZukf