



Book Barn Library

Bahareh Jabarzadnouri &
Kim Alizadeh
Dec. 20th, 2021

High Level View - Library

Security

Data storage
(e.g. e-book
files, photos)

Renting book
online

Search catalog

Late fees

Rental tracking

Access to
quality-assured
resources

User-centric
navigation

Search Catalog – Solution Overview

Book Barn

Search Catalog

Login

Search Catalog




Keyword: FICION

☐ Available

☐ Unavailable

Search

Clear

Cover Photo	Title	Author	Category	Book Type	Availability	ISBN	Book Details
	Business Design	Anthony	FICTION	PAPER	AVAILABLE	1112034444757	View Details
	Hardwork	Anthony	FICTION	EBOOK	UNAVAILABLE	1130366205777	View Details
	Harry Potter	Anthony	FICTION	PAPER	UNAVAILABLE	1112066205707	View Details

Total Items: 8 - Page 1 of 3 [First](#) [Previous](#) [1](#) [2](#) [3](#) [Next](#) [Last](#)

Book Barn

Search Catalog

Login

Search Catalog

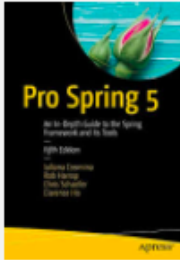

Keyword: PAPER

☐ Available

☐ Unavailable

Search

Clear

Cover Photo	Title	Author	Category	Book Type	Availability	ISBN	Book Details
	Spring Programming	Anthony	SCIENCE	PAPER	AVAILABLE	8012007004405	View Details
	The Dreaming Art	Anthony	ART	PAPER	AVAILABLE	1112066205767	View Details

Total Items: 14 - Page 5 of 5 [First](#) [Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#) [Last](#)

Search Catalog

Keyword: nd nd

keyword cannot contain spaces

☐ Available

☐ Unavailable

Search

Clear

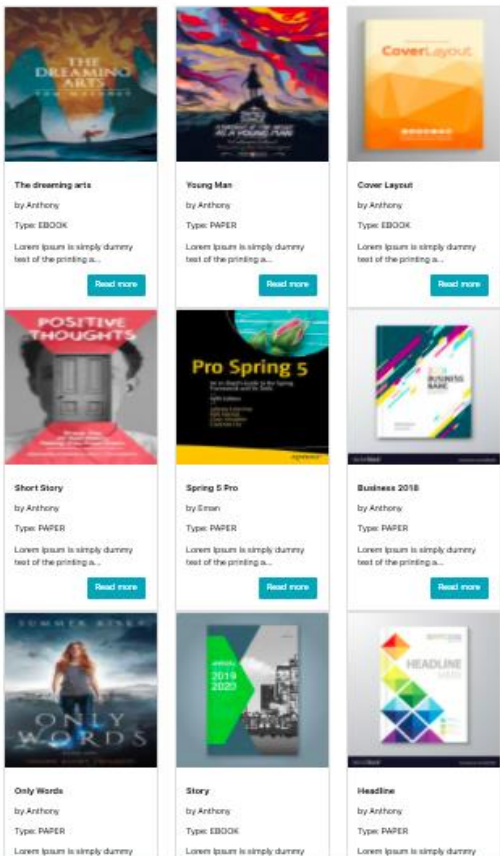
Cover Photo	Title	Author	Category	Book Type	Availability	ISBN	Book Details
-------------	-------	--------	----------	-----------	--------------	------	--------------

Total Items: 0 - Page 1 of 0 [First](#) [Previous](#) [1](#) [0](#) [Next](#) [Last](#)

Member Dashboard – Solution Overview



Recently added



young man



Author:
Anthony

Category:
FICTION

Book Type:
EBOOK

Description:
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

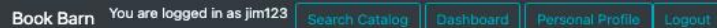
Status:
Available

Publication Date:
2010-11-21

ISBN:
1111111111111

Members who rent books online are entitled to a 1 month rental period.

Rent book!



Welcome to your dashboard

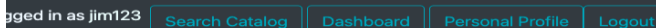
You have rented a book! The Due Date is: 2022-01-19

Your Rental History

Keyword:

Search Clear

Cover Photo	Title	Author	ISBN	Book Type	Start Date	Due Date	Return Date	E-Book Link	Return E-Book
	Programming	Anthony	8012033305555	EBOOK	2021-12-19	2022-01-19	2021-12-22	N/A	
	Hardwork	Anthony	1130366205777	EBOOK	2021-12-19	2022-01-19		View E-Book	Return



Personal Profile

Username	Firstname	Lastname	Email	Address	Age	Sex	Actions
jim123	jim	smith	jim@jim.com	111 street	38	Male	Update your personal info

Admin dashboard - Solution Overview



Manage Books

Keyword:

Search

Clear

Add Book

Cover Photo	Title	ISBN	Author	Category	Book Type	Publication Date	Description	Availability	Actions
	Hardwork	1130366205777	Anthony	FICTION	EBOOK	2014-10-11	Lorem ipsum is simply dummy text of the printing a...	Available	<div>Update</div> <div>Delete</div>
	Harry Potter	1112066205707	Anthony	FICTION	PAPER	2020-11-11	Lorem ipsum is simply dummy text of the printing a...	Available	<div>Update</div> <div>Delete</div>

Welcome Admin to your personal dashboard

Username	Firstname	Lastname	Email	Address	Age	Sex	Role	Actions
admin123	admin	boss	admin@admin.com	123 street	50	Female	Admin	<div>Update your personal info</div>

In order to view all users, adding, modifying or delete a user: [click here](#)

In order to view all users, adding, modifying or delete a user: [click here](#)

Manage Users

Keyword:

Search

Clear

Add User

Username	Firstname	Lastname	Email	Address	Age	Sex	Role	Actions
admin123	admin	boss	admin@admin.com	123 street	50	Female	Admin	<div>Update User</div> <div>Delete User</div>
flor123	flor	flora	flor@flor.com	120 down street	19	Female	Member	<div>Update User</div> <div>Delete User</div>

Admin - Solution Overview



Book Barn You are logged in as admin123

[Logout](#)[Search Catalog](#)[Manage Rentals](#)[Manage Users](#)[Manage Books](#)[Dashboard](#)

Rentals

A book was rented succesfully for a user!

Keyword:

Cover Photo	Title	Author	ISBN	Book Type	Rented By	Start Date	Due Date	Return Date	Availability	
	Business Design	Anthony	1112034444757	PAPER	jim123	2021-12-14	2021-12-29	2021-12-30	Available	<input type="button" value="Update Rental"/> <input type="button" value="Delete Rental"/>
	Harry Potter	Anthony	1112066205707	PAPER	jim123	2021-12-08	2021-12-31		Unavailable	<input type="button" value="Update Rental"/> <input type="button" value="Delete Rental"/>

Total Items: 3 - Page 1 of 2 [First](#) [Previous](#) 1 [2](#) [Next](#) [Last](#)

Book Barn You are logged in as admin123

[Logout](#)[Search Catalog](#)[Manage Rentals](#)[Manage Users](#)[Manage Books](#)[Dashboard](#)

Business Design



Author:

Anthony

Category:

FICTION

Book Type:

PAPER

Description:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Status:

Available

Publication Date:

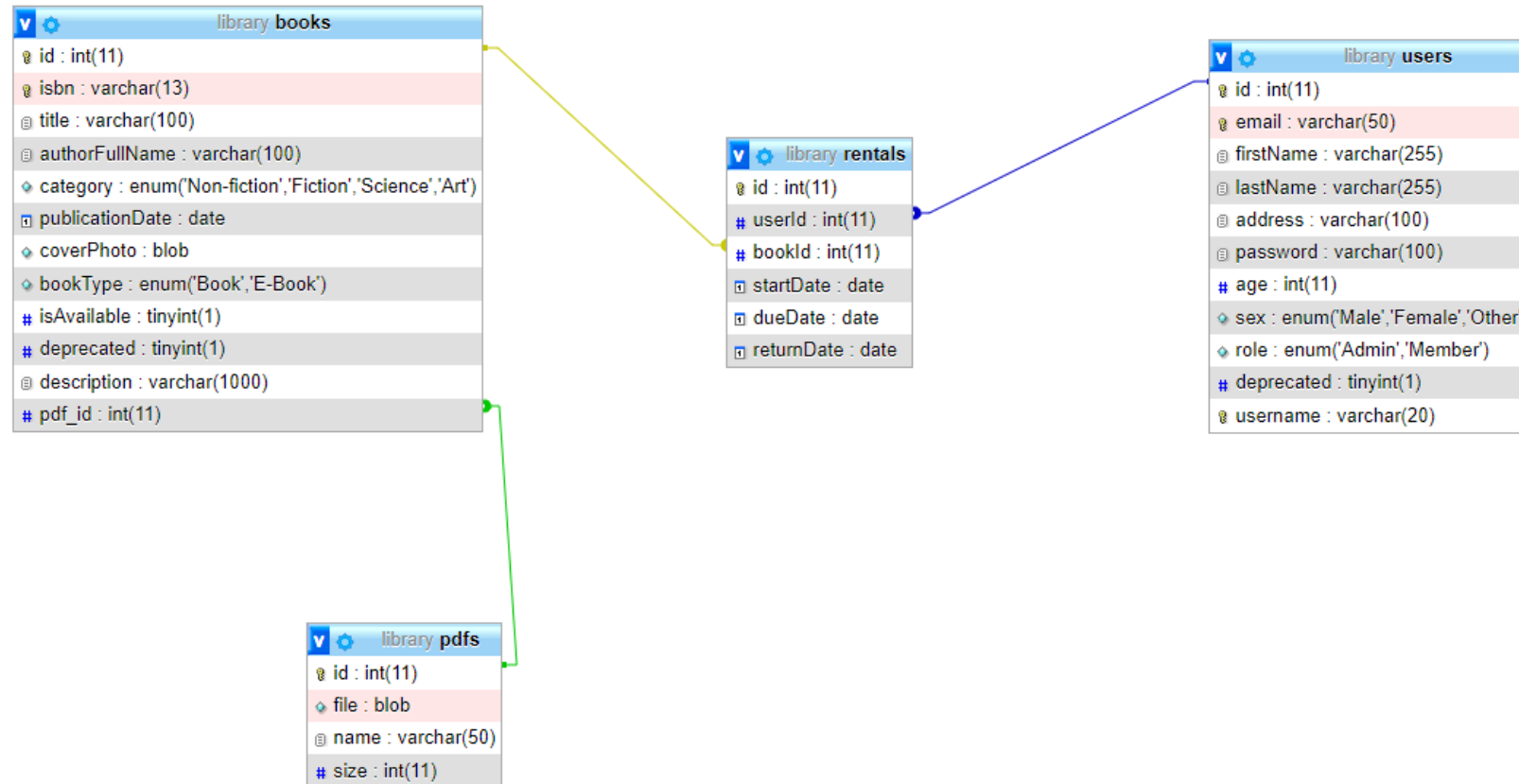
1910-11-11

ISBN:

1112034444757

Members who rent books online are entitled to a 1 month rental period.

Database



Challenges and Solutions

- Deprecate Books/Users while we have them in database

```
@GetMapping("/admin/deleteUser")
public String deleteUser (@RequestParam Long userId, RedirectAttributes redirAttrs) {
    User user = userRepository.findById(userId).get();
    List<Rental> rental = rentalRepository.findByUserUsernameAndReturnDateIsNull(user.getUsername());
    if (!rental.isEmpty()) {
        redirAttrs.addFlashAttribute( attributeName: "userNotDeleted", attributeValue: "WARNING: Cannot delete! This user has not returned a rented book!");
    } else {
        user.setDeprecated(true);
        userRepository.save(user);
        redirAttrs.addFlashAttribute( attributeName: "userDeleted", attributeValue: "User successfully deleted");
    }
    return "redirect:/admin/manageUsers";
}
```

```
public Page<User> findAllByDeprecatedFalse(String keyword, Pageable pageable);

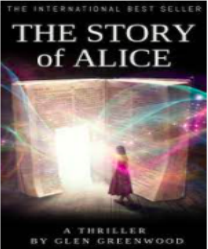
public Page<User> findAllByDeprecatedFalse(Pageable pageable);
```


Challenges and Solutions

Flash Messages

```
@GetMapping("/admin/deleteBook")
public String deleteBook(@RequestParam Long bookId, RedirectAttributes redirAttrs){
    Book book = bookRepository.findById(bookId).get();
    List<Rental> rental = rentalRepository.findByBook_IdAndAndReturnDateIsNull(bookId);
    if(!rental.isEmpty()){
        redirAttrs.addFlashAttribute(attributeName: "bookNotDeleted", attributeValue: "WARNING: Cannot delete! This book is rented");
    }
    else {
        book.setDeprecated(true);
        bookRepository.save(book);
        redirAttrs.addFlashAttribute(attributeName: "bookDeleted", attributeValue: "Book successfully deleted");
        //bookRepository.deleteById(bookId);
    }
    return "redirect:/admin/manageBooks";
}
```

WARNING: Cannot delete! This book is rented

Cover Photo	Title	ISBN	Author	Category	Book Type	Publication Date	Description	Availability	Actions
	Alice Story	8012007005555	Anthony	SCIENCE	PAPER	1900-11-11	Lorem Ipsum is simply dummy text of the printing a...	Unavailable	<button>Update</button> <button>Delete</button>

Manage Books

Book successfully deleted

Keyword:

Search

Clear

Challenges and Solutions

- Spring Security redirect based on Roles
- Solution:

```
@Configuration
public class CustomAuthenticationSuccessHandler implements AuthenticationSuccessHandler {

    @Override
    public void onAuthenticationSuccess(HttpServletRequest httpServletRequest, HttpServletResponse httpServletResponse,
        Authentication authentication) throws IOException, ServletException {

        Set<String> roles = AuthorityUtils.authorityListToSet(authentication.getAuthorities());
        if (roles.contains("ROLE_Admin")) {
            httpServletResponse.sendRedirect(location: "/admin");
        } else {
            httpServletResponse.sendRedirect(location: "/member/dashboard");
        }
    }
}
```

Challenges and Solutions

- Uploading files
 - @Lob, [] byte
 - MultipartFile

```
@PostMapping("/admin/addBook")
public String addBook(@Valid Book book, BindingResult bindingResult,
    RedirectAttributes redirAttrs,
    @RequestParam("coverPhoto") MultipartFile multipartFile1,
    @RequestParam("ebookFile") MultipartFile multipartFile2)
```

```
byte[] file1 = multipartFile1.getBytes();
byte[] file2 = multipartFile2.getBytes();
```

```
spring.servlet.multipart.enabled=true
```

Challenges and Solutions

"Failed to convert property value of type
org.springframework.web.multipart.support.StandardMultipartHttpServletRequest\$StandardMultipartFile to required type byte[]"

- Solution:

```
@InitBinder
protected void initBinder(HttpServletRequest request, ServletRequestDataBinder binder) throws ServletException {

    // Convert multipart object to byte[]
    binder.registerCustomEditor(byte[].class, new ByteArrayMultipartFileEditor());
}
```

Challenges and Solutions

- Managing file sizes
- Solution:
 - Storing E-book PDFs in a separate table
 - Setting file size limits

```
Long maxImageSize = 1024*1024L;
```

```
if(multipartFile1.getSize() > maxImageSize){  
    FieldError imageSize = new FieldError( objectName: "book", field: "coverPhoto", defaultMessage: "File size must not exceed 1MB.");  
    bindingResult.addError(imageSize);  
}
```

Challenges and Solutions

```
spring.servlet.multipart.max-file-size=2MB  
spring.servlet.multipart.max-request-size=2MB
```

- @ControllerAdvice

```
@ExceptionHandler(MaxUploadSizeExceededException.class)  
public String handleFileUploadError(RedirectAttributes redirAttrs) {  
    redirAttrs.addFlashAttribute(attributeName: "error", attributeValue: "File cannot exceed 2MB");  
    return "redirect:/admin/manageBooks";  
}
```

File cannot exceed 2MB

```
<div class="text-danger">[[${error}]]</div>
```

What We Learned

- Sending e-mail confirmations when a book is rented
- Gmail SMTP Server with Java Mail Sender

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-mail</artifactId>  
</dependency>
```

```
spring.mail.host=smtp.gmail.com  
spring.mail.port=587  
spring.mail.username=bookbarn.fsd01@gmail.com  
spring.mail.password=[REDACTED]  
spring.mail.properties.mail.smtp.auth=true  
spring.mail.properties.mail.smtp.starttls.enable=true
```

Signing in to Google



Password

Last changed Dec 16

2-Step Verification

✓ On

App passwords

1 password

What We Learned

```
public void sendSimpleEmail(String toEmail, String username, String title, String author, String dueDate) {  
  
    SimpleMailMessage message = new SimpleMailMessage();  
    String body = String.format("Congradulations %s, a book just rented the book: %s by %s! You must return it by: %s ",  
                                username, title, author, dueDate);  
  
    String subject = "Book Rented";  
    message.setFrom("bookbarn.fsd01@gmail.com");  
    message.setTo(toEmail);  
    message.setText(body);  
    message.setSubject(subject);  
    try{  
        mailSender.send(message);  
    }  
    catch(MailException ex) {  
        System.err.println(ex.getMessage());  
    }  
}
```

Book Rented Inbox x

bookbarn.fsd01@gmail.com

to me ▾

Congradulations jim123, you have just rented the book: Book2 by John Bark! You must return it by: 2022-01-17

```
service.sendSimpleEmail( toEmail: "bookbarn.fsd01@gmail.com", principal.getName(), book.getTitle(),  
                          book.getAuthorFullName(), rental.getDueDate().toString());|
```

What we learned

- Retrieving and displaying relevant information in small chunks for the search catalog
- Solution:
 - Query using keyword and availability
 - Pagination using Pageable interface
 - Sorting using Sort class

Search Catalog

Keyword:

☐ Available

☐ Unavailable

What we learned

Keyword queries

```
@Query("SELECT b FROM Book b WHERE b.deprecated = false AND " +  
      "CONCAT(b.title, ' ', b.authorFullName, ' ', b.category, ' ', b.bookType, ' ', b.isbn) " +  
      " LIKE %?1%")  
public Page<Book> findAllByDeprecatedFalse(String keyword, Pageable pageable);
```

```
@Query("SELECT b FROM Book b WHERE b.deprecated = false AND b.isAvailable=true AND " +  
      "CONCAT(b.title, ' ', b.authorFullName, ' ', b.category, ' ', b.bookType, ' ', b.isbn) " +  
      " LIKE %?1%")  
public Page<Book> findAllByDeprecatedFalseAndAvailable(String keyword, Pageable pageable);
```

```
@Query("SELECT b FROM Book b WHERE b.deprecated = false AND b.isAvailable=false AND " +  
      "CONCAT(b.title, ' ', b.authorFullName, ' ', b.category, ' ', b.bookType, ' ', b.isbn) " +  
      " LIKE %?1%")  
public Page<Book> findAllByDeprecatedFalseAndAvailableIsFalse(String keyword, Pageable pageable);
```

- Caveat: only works for 1 keyword, not multiple keywords

Keyword:

keyword cannot contain spaces

What we learned

Sorting and Pagination

```
public interface BookRepoPagination extends PagingAndSortingRepository <Book, Long> {
```

```
    public Page<Book> findAllByDeprecatedFalse(Pageable pageable);
```

```
    public Page<Book> listAll(int pageNumber, String keyword, String sortField, String sortDir) {
```

```
        // Create pageable object
        Pageable pageable = PageRequest.of( page: pageNumber - 1, size: 2, sort);
```

```
        return bookRepoPagination.findAllByDeprecatedFalse(pageable);
```

```
Page<Book> page = bookService.listAllAvailable(currentPage, keyword, sortField, sortDir, available);
```

```
Long totalItems = page.getTotalElements();
```

```
int totalPages = page.getTotalPages();
```

```
List<Book> bookList = page.getContent();
```

What we learned

Sorting and Pagination

```
public Page<Book> listAll(int pageNumber, String keyword, String sortField, String sortDir) {  
    Sort sort = Sort.by(sortField);  
    sort = sortDir.equals("asc") ? sort.ascending() : sort.descending();
```

```
model.addAttribute(attributeName: "sortField", sortField);  
model.addAttribute(attributeName: "sortDir", sortDir);
```



Future Work

- Late fees tracking & payment
- Dynamic queries
- File compression
- Email reminders for due date
- Password recovery
- More front-end development



Summary

- Accomplished many of the goals in our proposal
- Successfully combined different technologies
- Created a functional library system