BOOKSWAP APPLICATION PROJECT REPORT

Author

Ivan Yaremko

C00239239

Supervisor

Dr Chris Staff

Submission date

25/04/2022

1 CONTENTS

2 INTRODUCTION

The purpose of this document is to discuss my personal experience while developing BookSwap. The document describes the content of the submitted project, illustrates the UI elements, and discusses what elements of the project were achieved and not achieved.

3 DESCRIPTION OF SUBMITTED PROJECT

BookSwap is a web application developed for the second-hand book market. The application facilitates a way for members to trade books with each other. This allows for books to be recycled and reduce the cost of pleasure reading.

Members of the application can:

- Upload books that they intend to trade.
- Search the marketplace for books.
- Request to swap books with other members.
- Update their profiles

The application will be able to:

- Register and authenticate members.
- Allow members to add books by using an external ISBN API [1].
- Allow members to add photos to their profiles by using an external Cloudinary [2] API storage.
- Allow members to search the marketplace by querying the backend database.
- Facilitated swapping books.

3.1 TECHNOLOGIES

3.1.1 .NET with C#

This project uses the .NET framework [3] to develop the backend system of the application. The application utilises .NET's object-oriented programming environment by using C#. BookSwap uses five main libraries for development:

- **.NET Core.**
  This library contains the Inversion of Control container for automatic dependency injection.
- **Web API.**
  This library is used for building HTTP services such as API Controllers which allows clients to ask for information from the application.
- **Entity.**
  Entity is used for Object-Relational Mapping, this allows the use of domain classes without the need of configuring the schema for database tables and columns.
- **Core Identity.**
  This library supports user authentication by managing credentials such as email, passwords, usernames, and tokens.
- **SignalR**.
  SignalR allows the creation of Hubs where clients can connect to and gain the ability to send messages in real time via web sockets.

3.1.2 React with TypeScript

For the client side of BookSwap, the project uses React [4] and TypeScript. React is an open-source library that is used for building user interfaces (UI) for single-page applications (SPA). It is the view layer for web and mobile applications.

React requires additional libraries to be integrated into the development. These libraries are:

- **Axios**.
  Axios [5] is an HTTP client library that allows clients to make requests to a given endpoint. Axios is promise-based, which gives the ability to use JavaScript's `async` and `await` for more readable asynchronous code. Axios can also intercept and cancel requests.
- **MobX.**
  MobX [6] is an open-source state management tool.
- **React Router**
  React Router [7]] is a standard library for routing in React. React Router enables navigation among components in a React Application, it also allows changing the browser URL, and keeps the UI in sync with the URL. React Router enables the display of multiple views in a Single Page Application.
- **React SignalR**
  React SignalR [8] features hooks to connect events to a component. It manages connection to Hubs via web sockets and allows clients to message each other through the hub in real time.

3.1.3 ISBNdb API

ISBNdb [9] gathers data from hundreds of libraries, publishers, and merchants to compile a collection of unique book data searchable by ISBN. ISBNdb is a paid service but they have an academic / non-profit discount for €5 a month. BookSwap uses this API to allow members to retrieve details of the books they are entering into the market.

3.1.4 Cloudinary

Cloudinary [10] is an API service that allows the storage of media such as images and videos. BookSwap uses Cloudinary for members profile images. Instead of storing the files in the application database, BookSwap stores the files in Cloudinary, and in tern Cloudinary provides a public URL to view the image, this URL is stored in the applications database.

3.2 Application Architecture

The goal of using an application architecture is to achieve separation of concerns and achieve a code base that is modular, maintainable, and scalable. BookSwap achieves this by using Clean Architecture and the Command Query Responsibility Segregation pattern (CQRS).

3.2.1 Clean Architecture

Clean architecture was created by Robert C. Martin [11]. The golden rule that makes this architecture work is the dependency rule [11]:

“Source code dependencies must point only inward, toward higher-level policies.”

The core of the system should not be changed if the framework or UI changes. This protects the core of the system and makes the external dependencies completely replaceable.

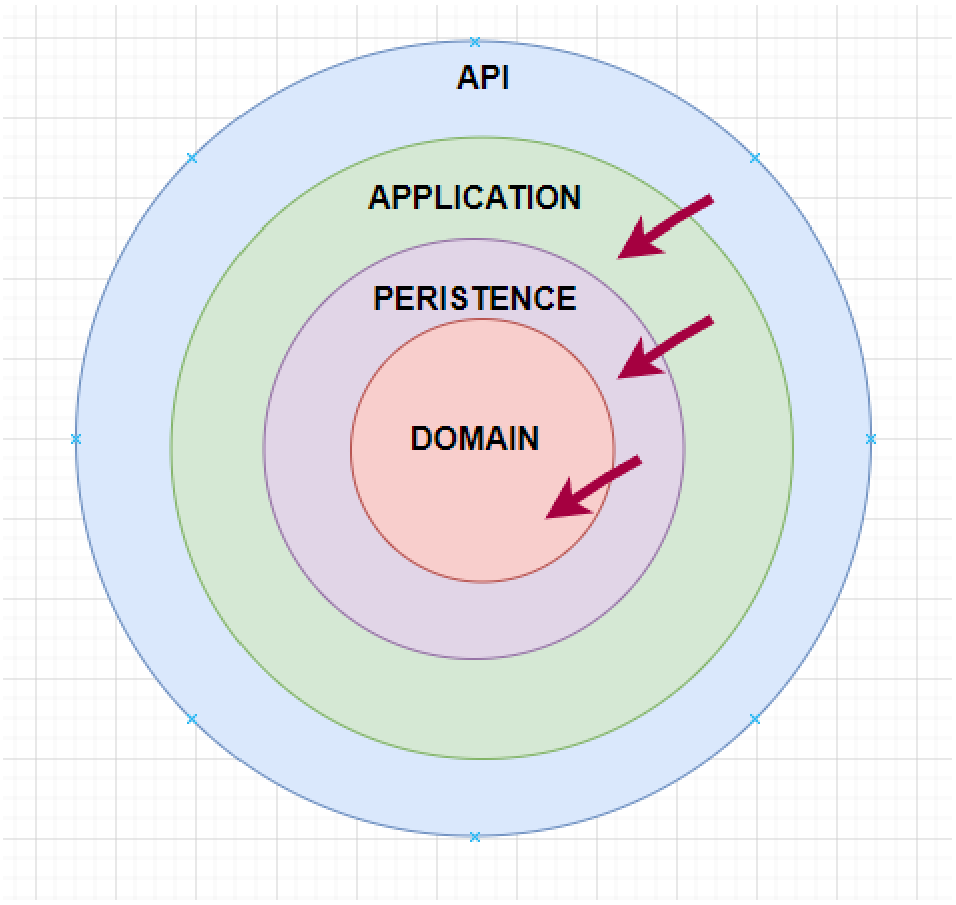


Figure 1 BookSwap implementation of Clean Architecture

The way BookSwap implements this architecture is demonstrated in Figure 1. BookSwap follows the inner dependency rule and isolates different application layers into their own responsibilities.

- **API**
  The API layer is responsible for communicating with HTTP requests. When a HTTP request hits an API endpoints, the API delegates the business logic to the application layer.

- **Application**
  The application layer is responsible for the business logic of the application. This layer uses the necessary services, such as a DataContext in the persistence layer, via constructor injections to write the code for a specific need.
- **Persistence**
  The persistence layers only responsibility is to communicate with the data store.
- **Domain**
  The domain is used to store classes which are used as a reference of database creation, design, and schema.

3.2.2 CQRS

The Command and Query Responsibility Segregation [12] (CQRS) pattern is used to separate the create/update and read operations for a data store. Create, update, and delete operations are known as commands in this pattern and reads are queries. CQRS separates the commands and queries, commands do something with the database and queries read data from the database.

BokSwap implements this pattern by using a library called MediatR [13]. This library allows process managing and supports request/response, command, and queries. To demonstrate this flow of control, when the API receives a HTTP request:

- The API controller will send this request via the mediator send method.
- The request could be either a query (Read) or a command (Write).
- The mediator will send this request to the mediator handler.
- The handler will handle the use case and apply the necessary code and logic.
- The handler will return to the API controller with the necessary information.
- The API controller will then return the information through a HTTP response to the client.

3.3 SCREENSHOTS

3.3.1 Home screen

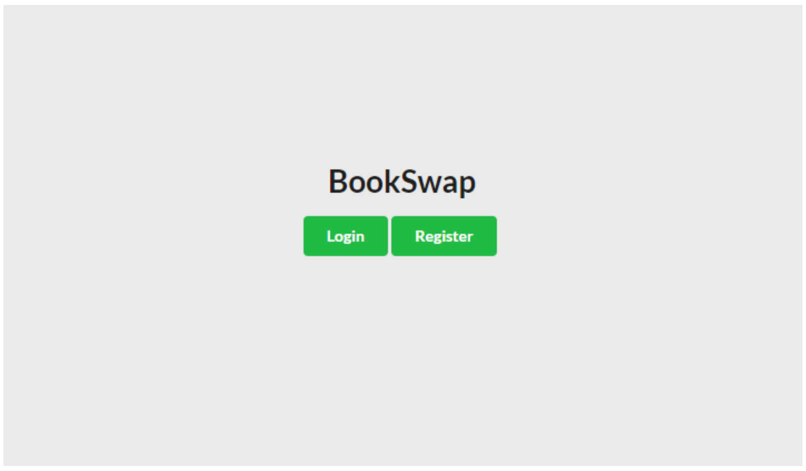


Figure 2 Home screen

In this screen the user is greeted with the home screen of BookSwap. The user then has two options, Logging in or Registering.

3.3.2 Login screen

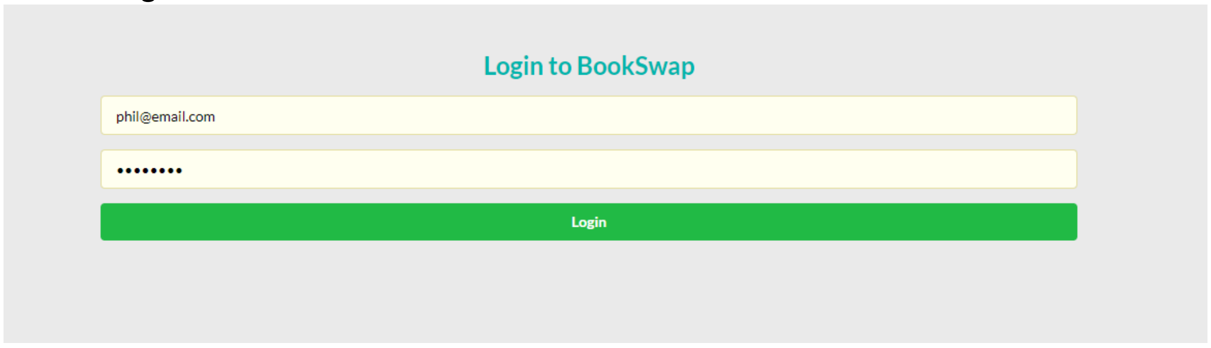
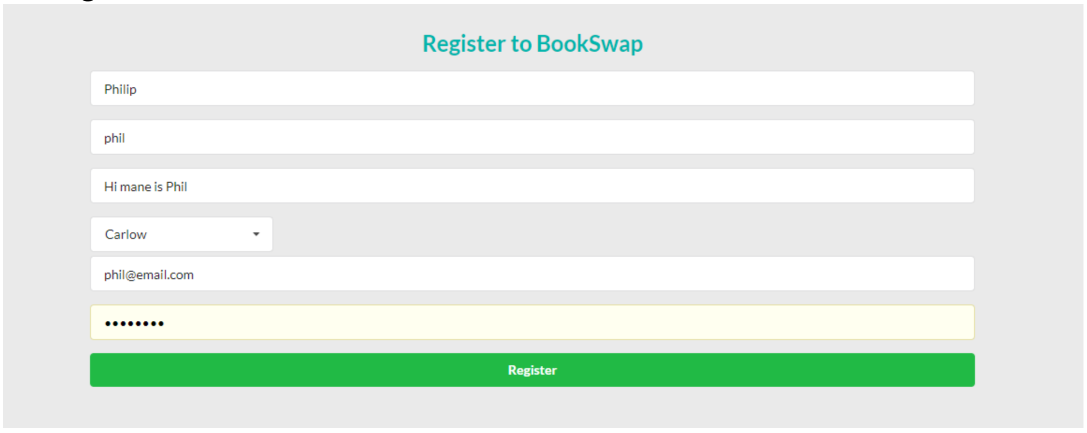


Figure 3 Login screen

In this screen a user can log into the application by providing their email and password.

3.3.3 Register screen

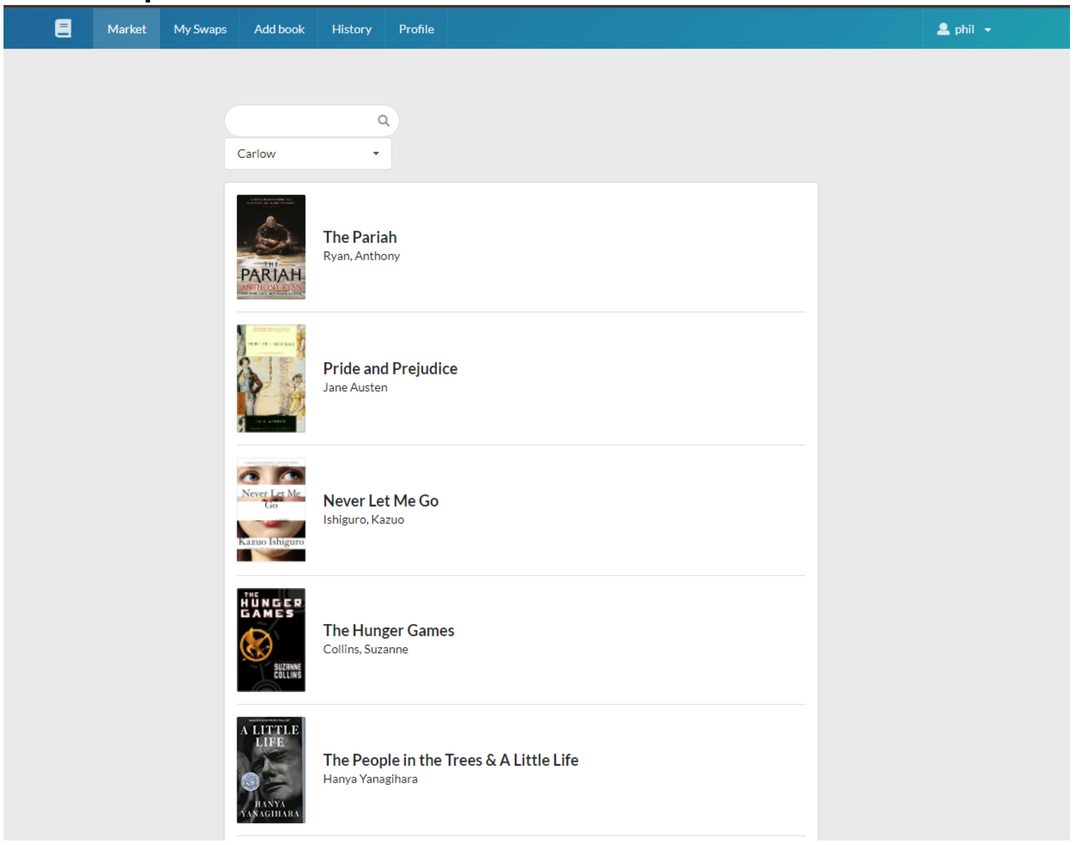

Register to BookSwap

Philip
phil
Hi mane is Phil
Carlow
phil@email.com
••••••••

Figure 4 Register screen

In this screen a user may register with the BookSwap application by providing the necessary details.

3.3.4 Market place dashboard screen



The Pariah	Ryan, Anthony
Pride and Prejudice	Jane Austen
Never Let Me Go	Ishiguro, Kazuo
The Hunger Games	Collins, Suzanne
The People in the Trees & A Little Life	Hanya Yanagihara

Figure 5 Market place dashboard screen

When a user is logged in, they are redirected into this screen. This screen shows the market dashboard.

3.3.5 Search book screen

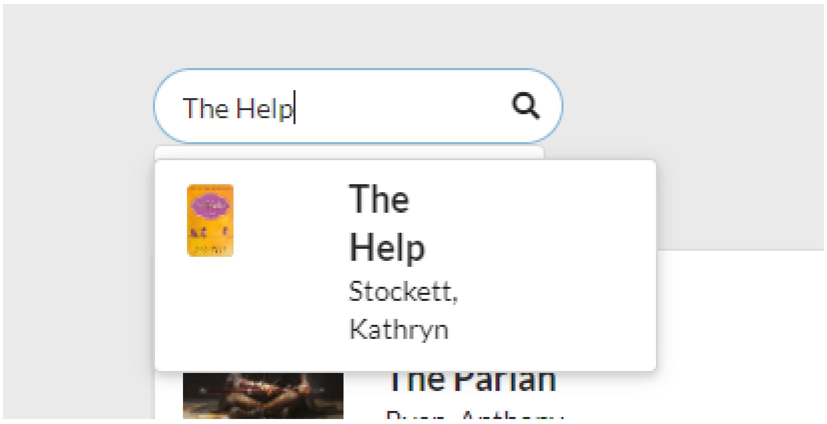
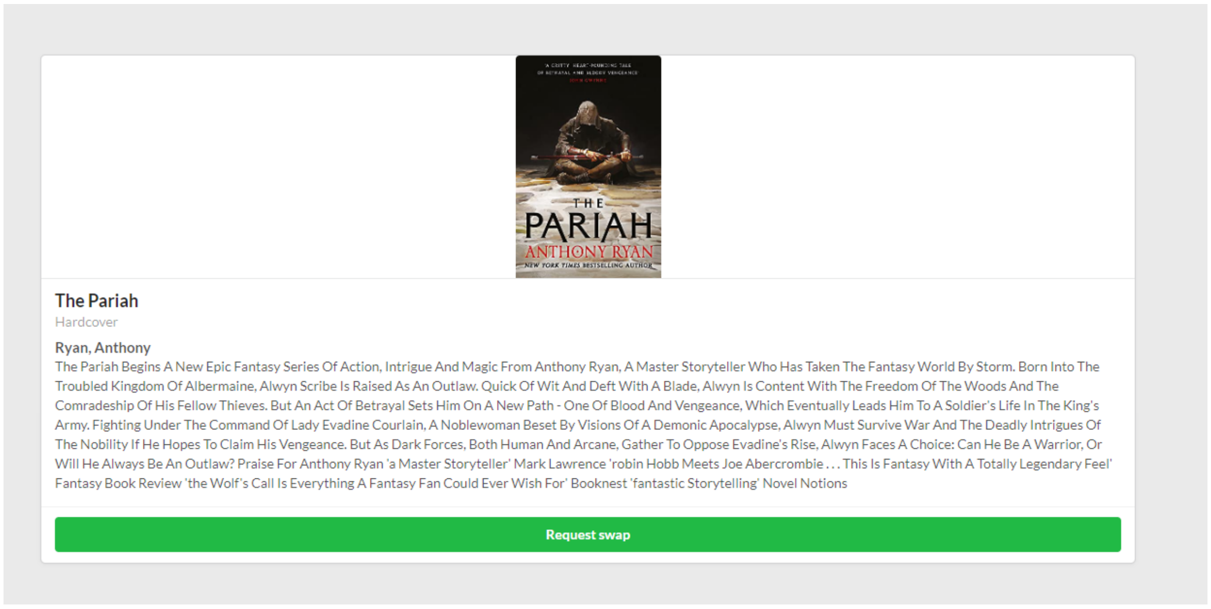


Figure 6 Search book screen

In the market dashboard a user may look for a book in the selected county.

3.3.6 View book details screen


The Pariah

Hardcover

Ryan, Anthony

The Pariah Begins A New Epic Fantasy Series Of Action, Intrigue And Magic From Anthony Ryan, A Master Storyteller Who Has Taken The Fantasy World By Storm. Born Into The Troubled Kingdom Of Albermaine, Alwyn Scribe Is Raised As An Outlaw. Quick Of Wit And Deft With A Blade, Alwyn Is Content With The Freedom Of The Woods And The Comradeship Of His Fellow Thieves. But An Act Of Betrayal Sets Him On A New Path - One Of Blood And Vengeance, Which Eventually Leads Him To A Soldier's Life In The King's Army. Fighting Under The Command Of Lady Evadine Courlain, A Noblewoman Beset By Visions Of A Demonic Apocalypse, Alwyn Must Survive War And The Deadly Intrigues Of The Nobility If He Hopes To Claim His Vengeance. But As Dark Forces, Both Human And Arcane, Gather To Oppose Evadine's Rise, Alwyn Faces A Choice: Can He Be A Warrior, Or Will He Always Be An Outlaw? Praise For Anthony Ryan 'a Master Storyteller' Mark Lawrence 'robin Hobb Meets Joe Abercrombie . . . This Is Fantasy With A Totally Legendary Feel' Fantasy Book Review 'the Wolf's Call Is Everything A Fantasy Fan Could Ever Wish For' Booknest 'fantastic Storytelling' Novel Notions

Figure 7 View book details screen

In this screen a member can see a detailed view of a book they selected from either the marketplace or a user’s profile. They can also request to swap this book with the owner of the book.

3.3.7 My Swaps dashboard screen

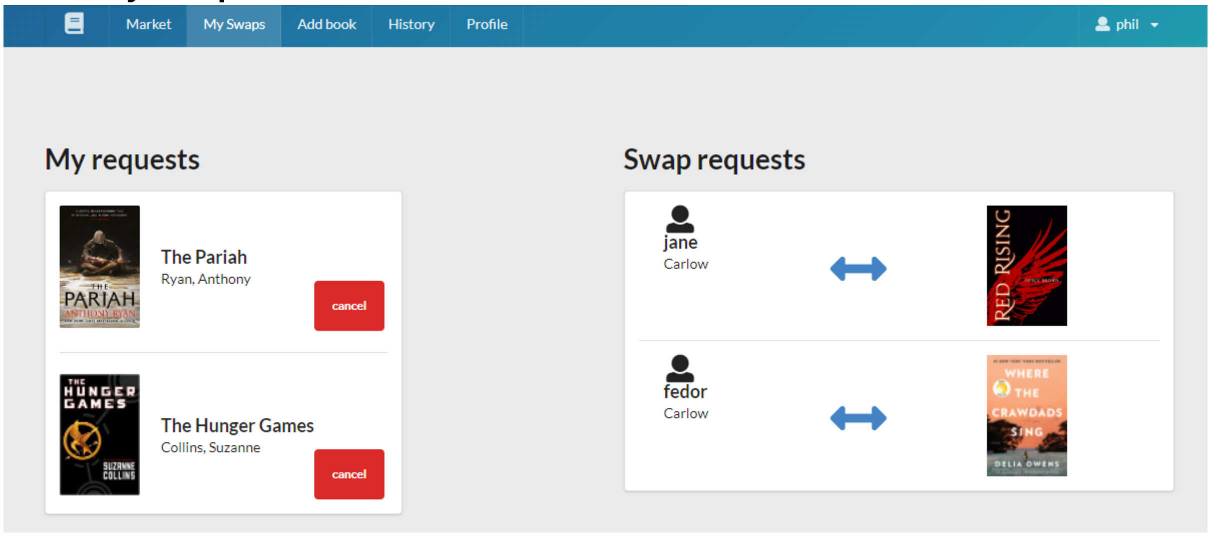


Figure 8 My Swaps dashboard screen

This screen is the dashboard a member sees to review their sways. On the left of the screen the member may review the swaps the requested and on the right swaps being requested from them.

3.3.8 View swap request screen

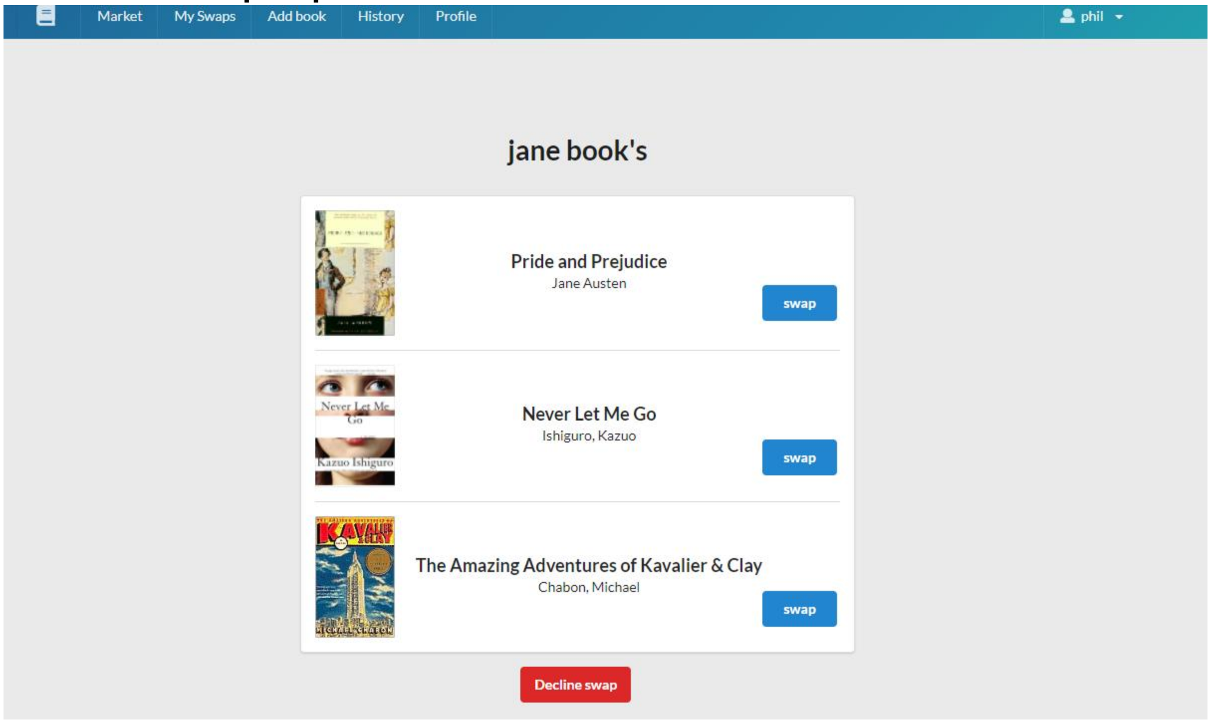


Figure 9 View swap request screen

This screen is accessed by selecting a swap that is being requested of the logged in user. The member may select a book to swap from the list provided or cancel the swap.

3.3.9 Swap history dashboard

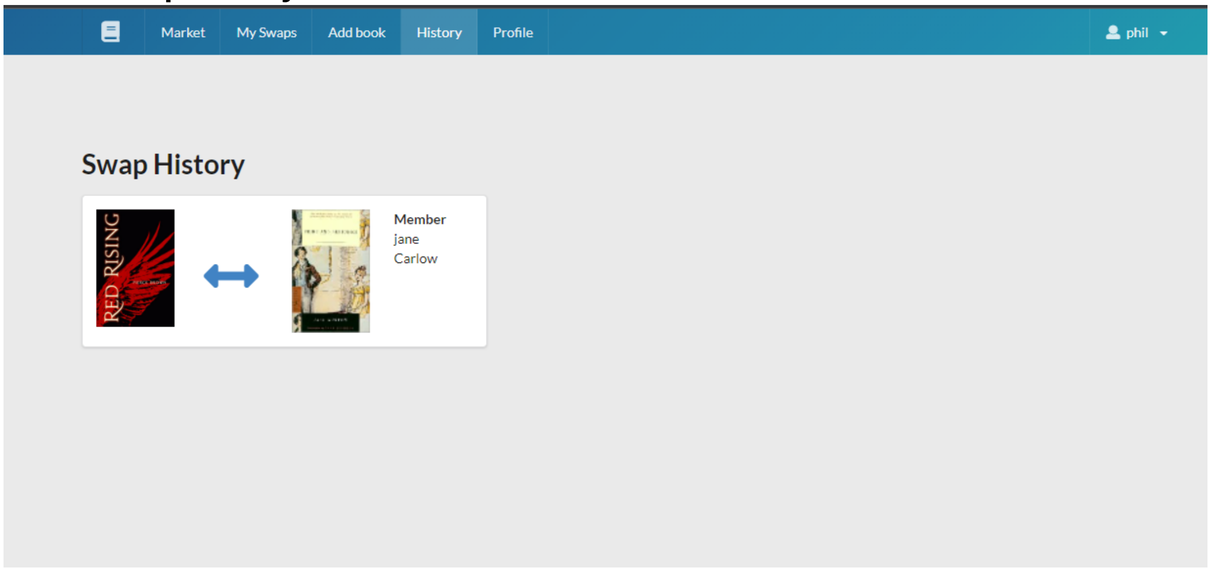


Figure 10 Swap history dashboard screen

In this screen a member may see their swap history.

3.3.10 Chat with member screen

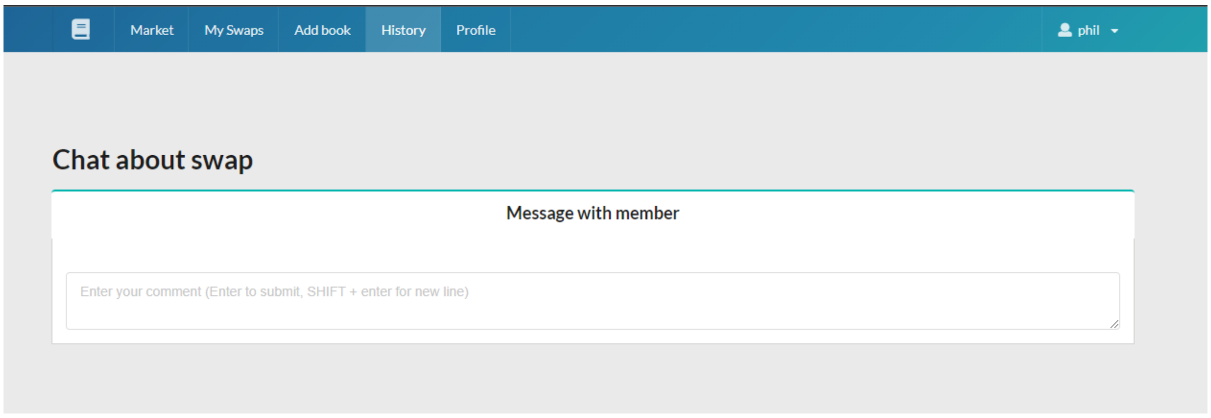


Figure 11 Chat with member screen

This screen is accessed by clicking on one of the swaps from the history dashboard. In this screen both members of the swap may message each other.

3.3.11 Create book – get book details screen

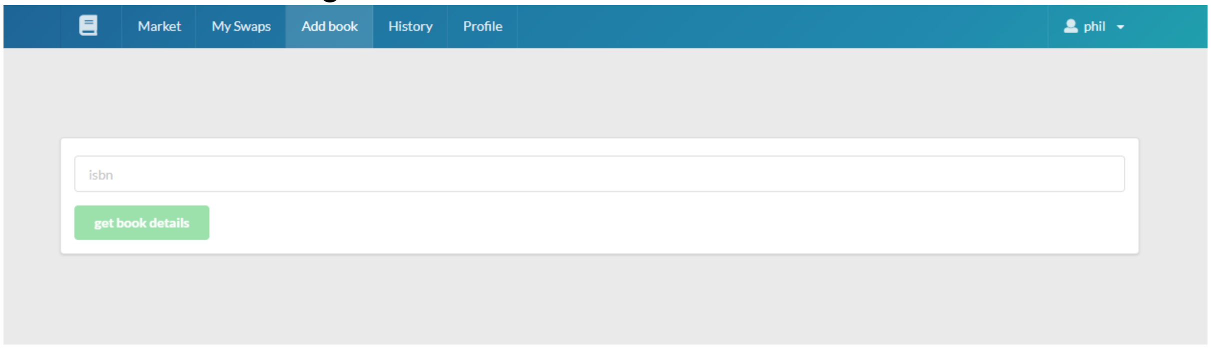
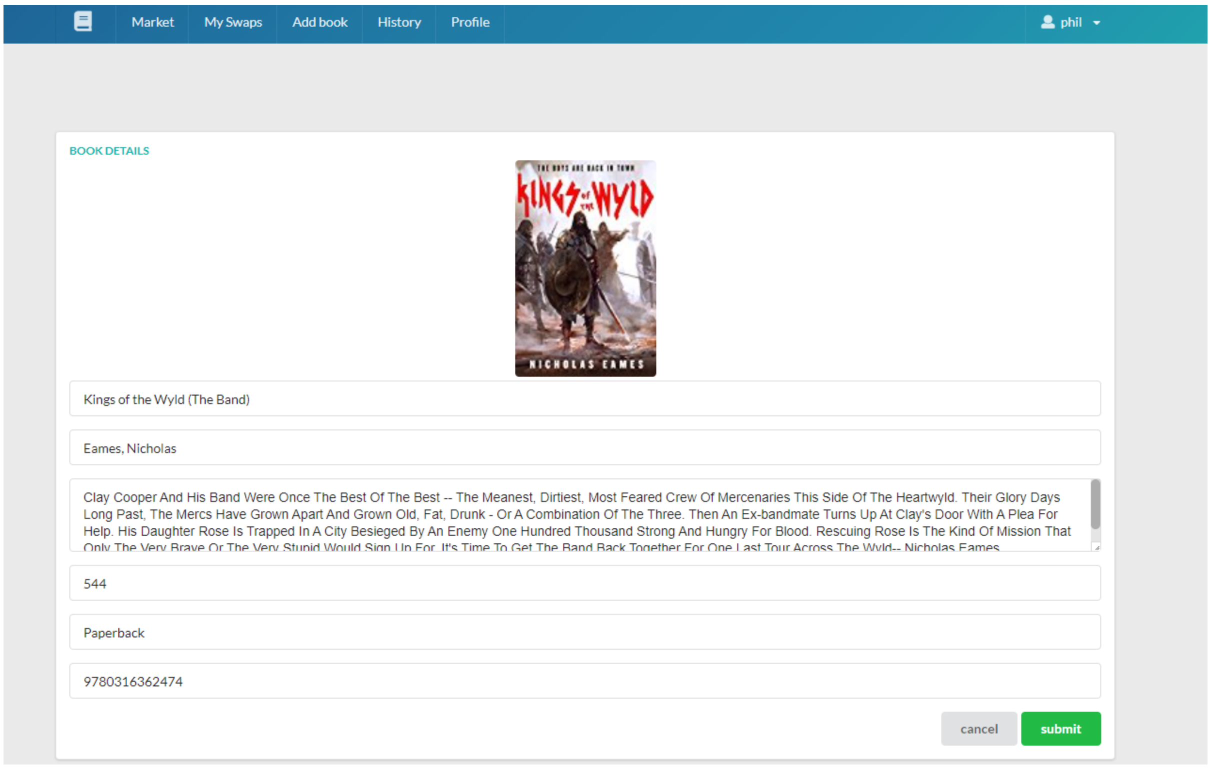


Figure 12 Get book details screen

This screen is used when a member wishes to add a book to the marketplace.

3.3.12 Create book – view details of new book to create



Kings of the Wyld (The Band)

Eames, Nicholas

Clay Cooper And His Band Were Once The Best Of The Best -- The Meanest, Dirtiest, Most Feared Crew Of Mercenaries This Side Of The Heartwyld. Their Glory Days Long Past, The Mercs Have Grown Apart And Grown Old, Fat, Drunk - Or A Combination Of The Three. Then An Ex-bandmate Turns Up At Clay's Door With A Plea For Help. His Daughter Rose Is Trapped In A City Besieged By An Enemy One Hundred Thousand Strong And Hungry For Blood. Rescuing Rose Is The Kind Of Mission That Only The Very Brave Or The Very Stupid Would Sign Up For. It's Time To Get The Band Back Together For One Last Tour Across The Wyld-- Nicholas Eames

544

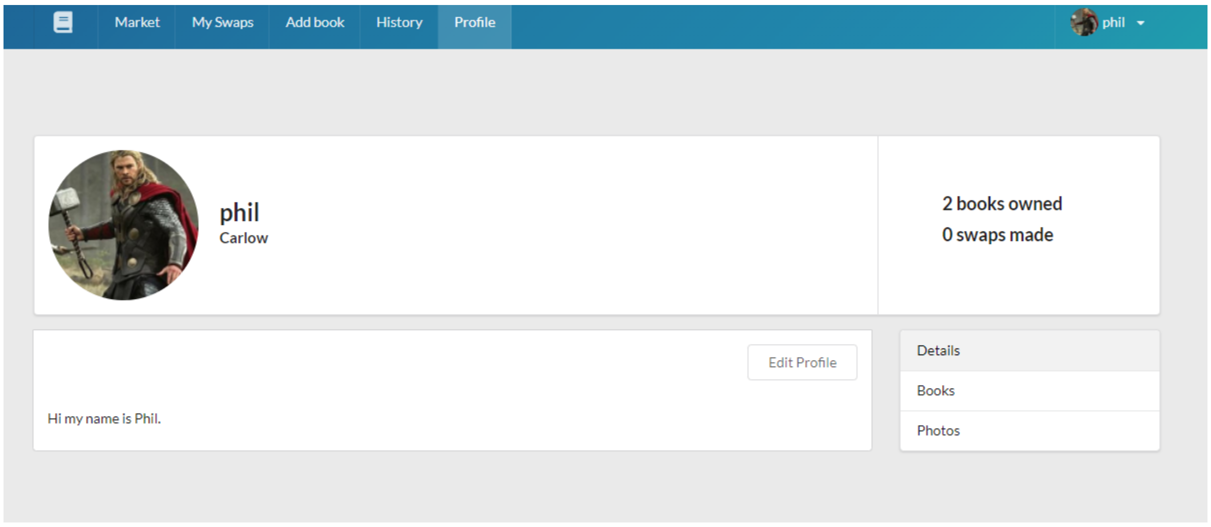
Paperback

9780316362474

Figure 13 View details of book to create

This screen is accessed once the member has entered a valid ISBN, the details of the book is displayed in this form.

3.3.13 Profile dashboard



phil
Carlow

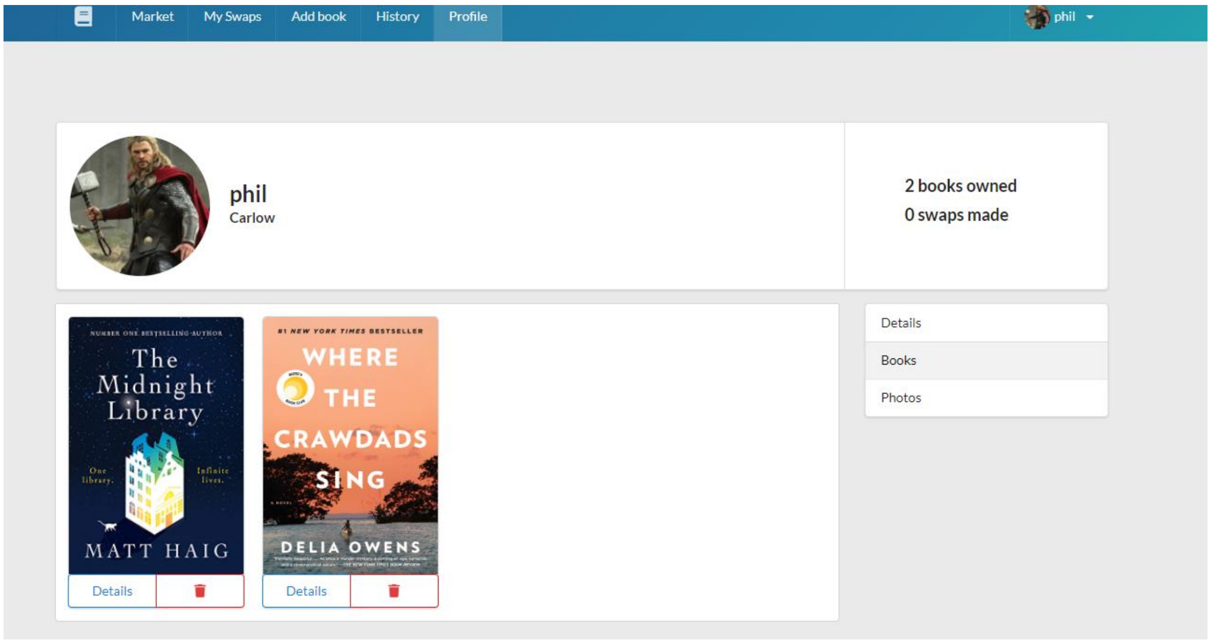
2 books owned
0 swaps made

Hi my name is Phil.

Figure 14 Profile dashboard screen

This screen is the profile dashboard screen.

3.3.14 View owned books screen



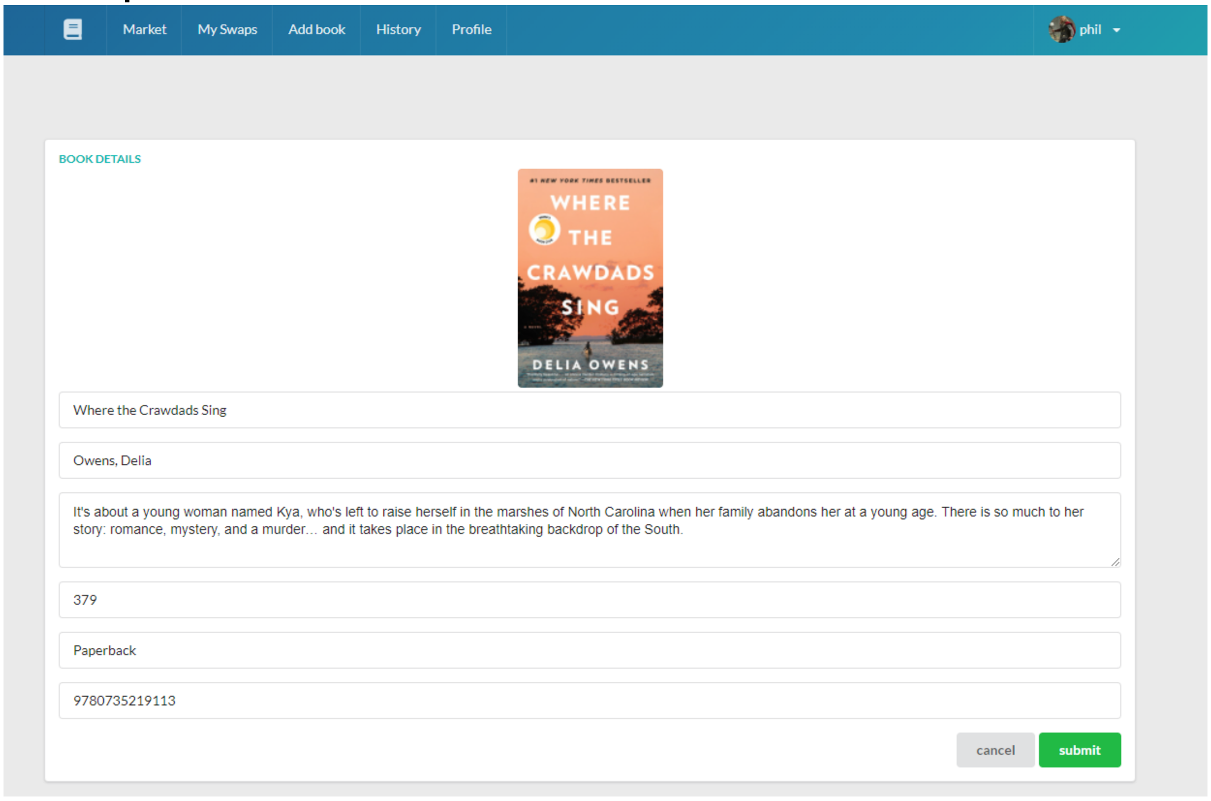
phil
Carlow

2 books owned
0 swaps made

Figure 15 View books owned screen

A member may look at their books they have on offer in the marketplace.

3.3.15 Update owned book details screen



Where the Crawdads Sing

Owens, Delia

It's about a young woman named Kya, who's left to raise herself in the marshes of North Carolina when her family abandons her at a young age. There is so much to her story: romance, mystery, and a murder… and it takes place in the breathtaking backdrop of the South.

379

Paperback

9780735219113

Figure 16 Update book screen

A member may update the details of a book they own. This screen is accessed via the member's profile dashboard.

3.3.16 Profile photos screen

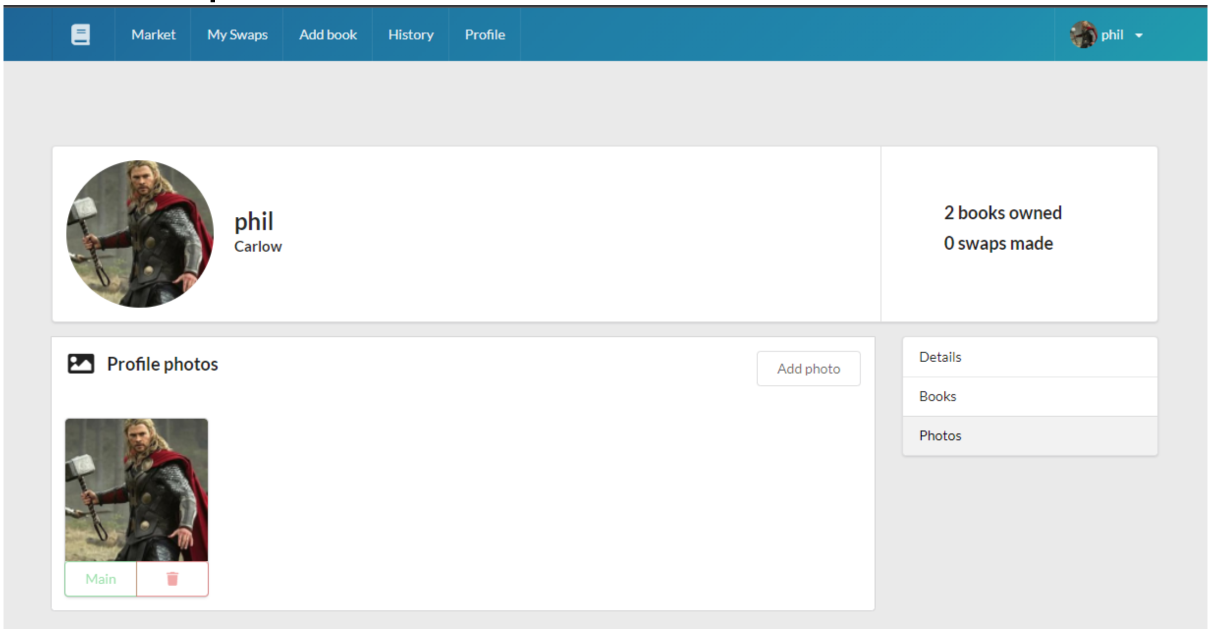


Figure 17 Profile photos screen

In this screen a member may delete a photo or set a photo as their main photo.

3.3.17 Add photo to profile screen

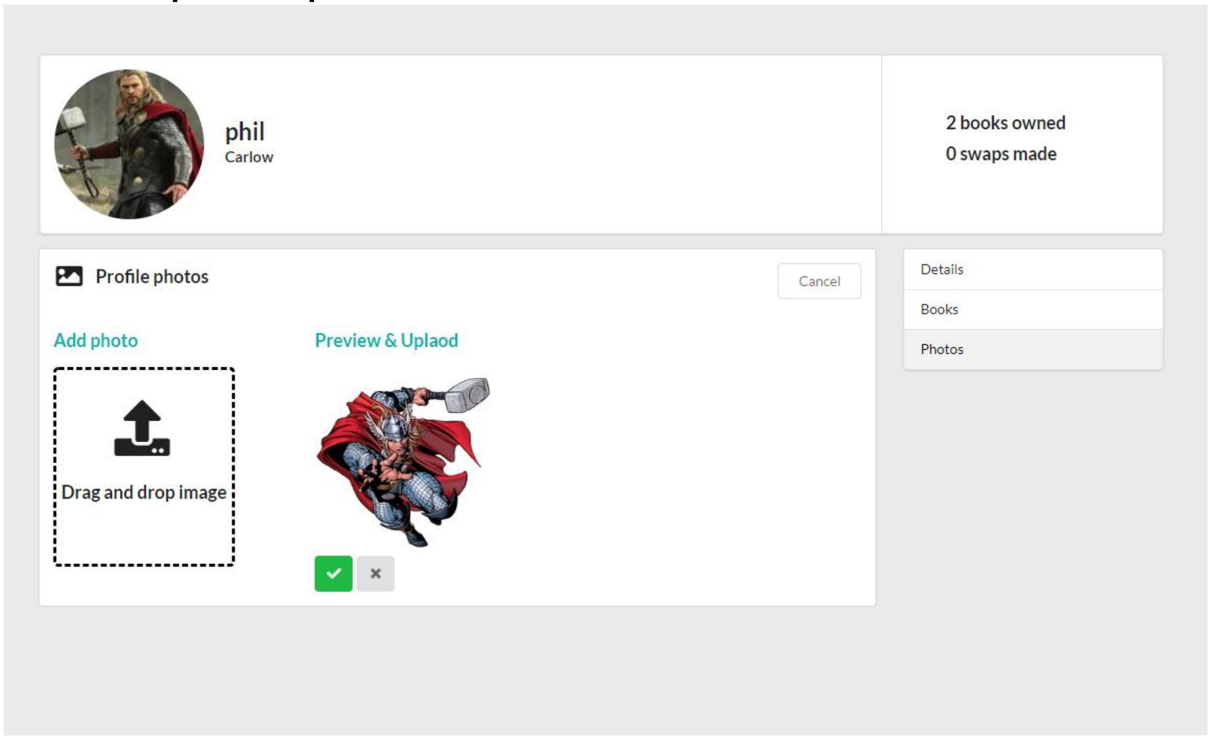


Figure 18 Add photo screen

In this screen a member may add a photo to their profile.

3.4 Entity Relationship Diagram

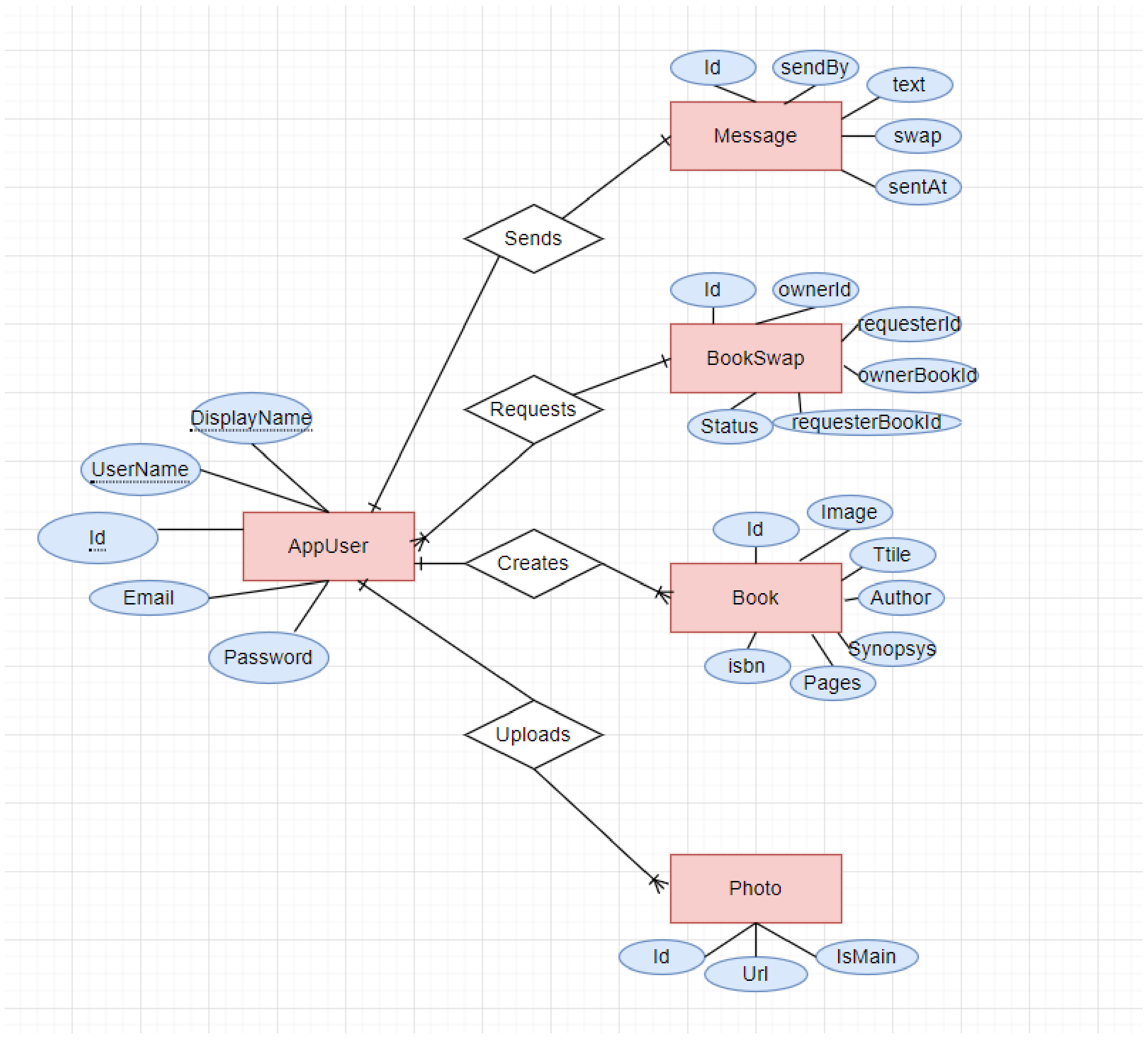
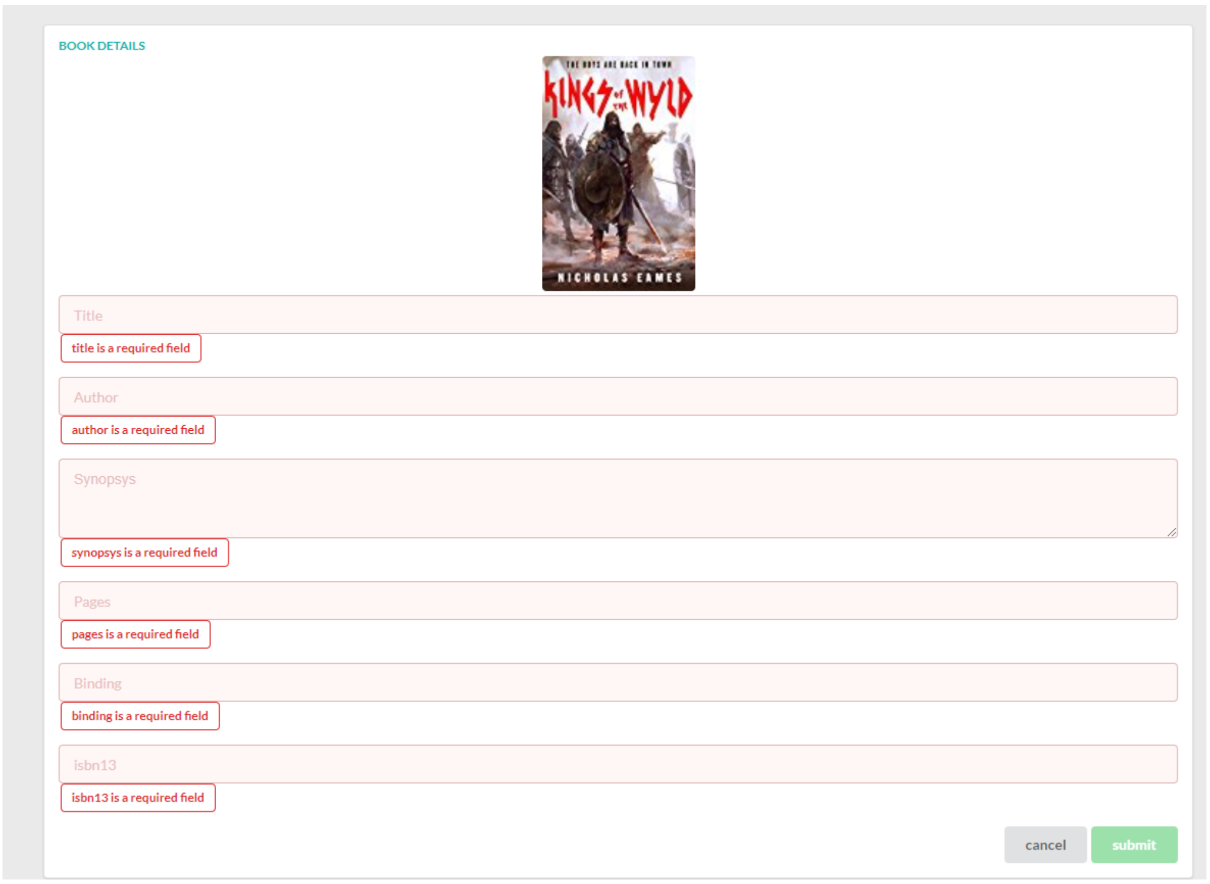


Figure 19 BookSwap ER Diagram

4 Adherence to the Functional Specification

4.1 Adding books usability


BOOK DETAILS

Title

title is a required field

Author

author is a required field

Synopsys

synopsys is a required field

Pages

pages is a required field

Binding

binding is a required field

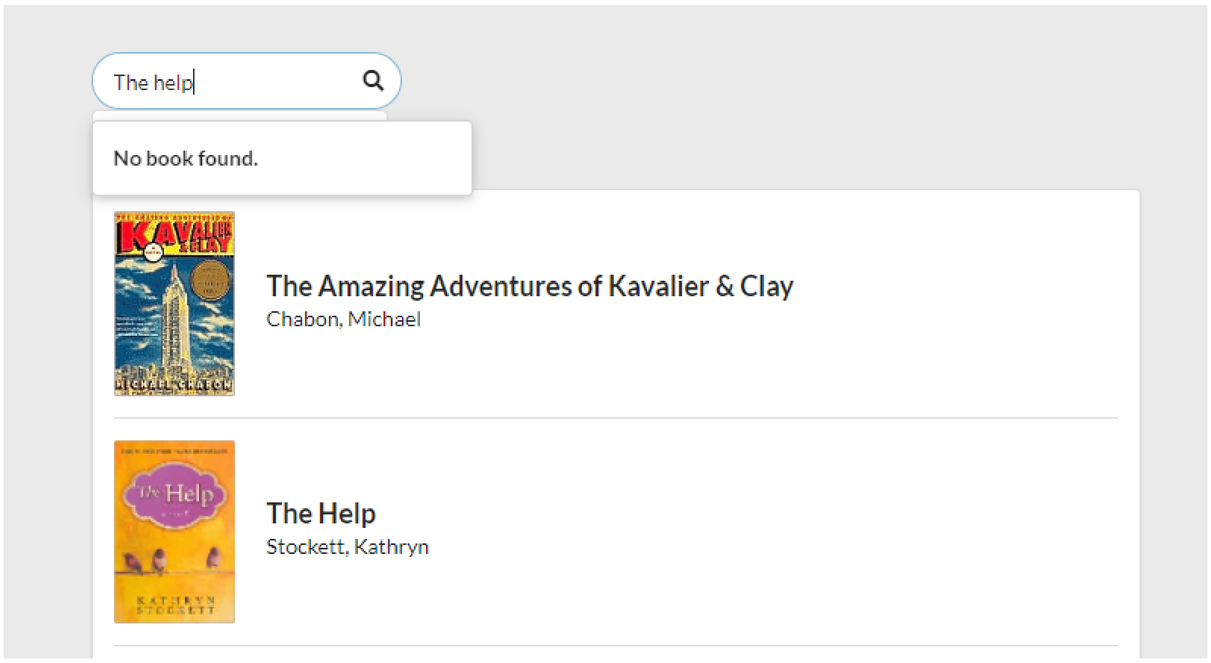
isbn13

isbn13 is a required field

Figure 20 Error display

Figure 20 illustrates the errors shown when a member is trying to add a book to the marketplace. The application displays the required fields required to be filled in.

4.2 SEARCH BOOK USABILITY


The help

No book found.

The Amazing Adventures of Kavalier & Clay
Chabon, Michael

The Help
Stockett, Kathryn

Figure 21 Search book

Figure 21 shows how sensitive the search feature is. The search text requires to be exact; it does not account for lower case letters or auto completeness.

5 LEARNING OUTCOMES

5.1 TECHNICAL LEARNING

I have improved with my programming knowledge by following an architecture for my code base and implementing a pattern. Working on this final year project has improved my full-stack development experience. I have a much broader context now on the different systems it takes to develop to release a product.

5.2 ISSUES

The biggest issue I had with this project is time and project management. Overall, I did not implement my full original vision for this project. Due to personal issues, I could not put in the hours required in developing this project during term 2.
It was hard to keep a consistent schedule week to week. Especially when I had time and mental capacity to develop, I had to first spend time going back through all my code to bring back my memory and what I was doing.

If I was to redo this project from the start, I would focus on developing the main core functionality of the application. I would particularly focus on failing first trying to develop and figure out the technical requirements for the main functionality. It would have been easier to build additional functionality around the core functionality.

6 REVIEW OF PROJECT

6.1 ACHIEVED

This project has achieved nearly all the core functionality features:

- **Authenticate**
  The ability to register and login with the application, as well as using a secure JSON Web Token so that authenticate users may communicate with the backend system.
- **Search**
  Members may search for books in the marketplace, either through the search bar or through using the provided list.

- **CRUD Books**
  Members have the ability to create, read, update, and delete books.

- **CRUD Profile**
  Members have the ability to update their profiles and be able to set images alongside their profile.

- **Swap books**
  The application provides the functionality for members to request a book through the marketplace, have the owner of the requested book review the request. The owner can decide whether to swap their book with one of the books from the requestor or cancel the swap request.

- **Chat**
  Currently the chat feature is not working as intended, but the UI elements exists along with the backend logic. The configuration needs to be correctly implemented for the members to be able to message each other.

6.2 NOT ACHIEVED

The following non-core functionalities were not achieved while developing this project:

- **Wish list.** Ability for members to add books they wish to swap
- **Novel recommendation.** A machine learning algorithm that recommends books that member would like.
- **Google map.** A map integration into the chat system.
- **Member rating.** A member scoring system to show case the trust level of a member.

7 REFERENCES

[1] ISBNdb API Documentation v2 | ISBNdb. (2021, August 18). Isbndb.Com. Retrieved April 25, 2022, from https://isbndb.com/apidocs/v2

[2] Cloudinary.com. 2022. Cloudinary Image & Video Management - Documentation Home | Cloudinary. [online] Available at: <https://cloudinary.com/documentation> [Accessed 25 April 2022].

[3] G. (2021, September 15). *Overview of .NET Framework - .NET Framework*. Microsoft Docs. https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview

[4] Components and Props –. (2021, June 2). React. https://reactjs.org/docs/components-and-props.html

[5] *npm: axios*. (2021, October 25). Npm. https://www.npmjs.com/package/axios

[6] *MobX: Ten minute introduction to MobX and React*. (2021, October 5). Mobx.Js.Org. Retrieved Accessed 25 April 2022, from https://mobx.js.org/getting-started

[7] *React Router: Declarative Routing for React*. (2021, July 21). ReactRouterWebsite. Retrieved Accessed 25 April 2022, from https://reactrouter.com/

[8] npm. 2022. *react-signalr*. [online] Available at: <https://www.npmjs.com/package/react-signalr> [Accessed 25 April 2022].

[9] *ISBNdb API Documentation v2 | ISBNdb*. (2021, August 18). Isbndb.Com. Retrieved 25 April 2022, from https://isbndb.com/apidocs/v2

[10] Cloudinary.com. 2022. Cloudinary Image & Video Management - Documentation Home | Cloudinary. [online] Available at: <https://cloudinary.com/documentation> [Accessed 25 April 2022].

[11] Martin, R. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series)* (1st ed.). Pearson.

[12] Narumoto, M. N. (2021, June 14). *What is the CQRS pattern? - Azure Architecture Center*. Microsoft Docs. https://docs.microsoft.com/en-us/azure/architecture/patterns/cqrs

[13] Bogard, J., 2021. *GitHub - jbogard/MediatR: Simple, unambitious mediator implementation in .NET*. [online] GitHub. Available at: <https://github.com/jbogard/MediatR> [Accessed 25 April 2022].

8 PLAGIARISM DECLARATION

INSTITUTE *of* TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name (Printed) : Ivan Yaremko

Student Number : C00239239

Signature:

Recoverable Signature

X [illegible]

Ivan Yaremko

Signed by: 6c456de4-a264-4906-bbdd-cd8e10592da5