Online Book Store

Ahmad Allahham

ahmall-0@student.ltu.se

Group 1

2023-03-27

Rapport for Sprint 3

Executive Summary:

The purpose of this report is to provide an overview of the development of an online book store. This platform aims to provide users with an intuitive and seamless shopping experience by offering a vast array of books across different genres. The online book store will be developed in two stages: database and front-end, back-end.

In the first stage, the database will be developed to store all the necessary information about books and the available inventory. This database will be the foundation for the platform and will be instrumental in ensuring data integrity, consistency, and security.

The second stage will involve the development of the front-end and back-end of the platform. The front-end will be designed to provide users with an easy-to-use interface to search, browse, and purchase books. The back-end, on the other hand, will handle the business logic of the platform and will ensure that all transactions are processed efficiently and securely.

In conclusion, the development of the online book store is expected to revolutionize the way books are sold and purchased online. With the increasing popularity of e-commerce, the platform has the potential to become a major player in the online book market and will provide users with an unparalleled shopping experience.

User Stories:

1. As a customer, I want to be able to search for books by title, author, or genre, so that I can find what I am looking for quickly and easily.
2. As a customer, I want to be able to view detailed information about a book, including description, so that I can make an informed decision about whether to purchase it.
3. As a customer, I want to be able to add books to my cart.
4. As a customer, I want to be able to view my order history and track the delivery of my purchases, so that I can keep track of my purchases and know when to expect them.
5. As Manager, I want to be able to manage the books in the store, including adding new books, updating existing books, and removing books that are out of stock, so that I can keep the store's inventory up-to-date.
6. As Manager, I want to be able to view and analyze sales data, so that I can make informed decisions about which books to stock and how to price them.

7. As Manager, I want to be able to manage customer orders, including shipping orders, so that I can provide a smooth and efficient ordering process for customers.

Use Cases:

Use case1 (Customer's process)

- sign up or sign in as Customer

- Go to Library or search a book

- Order a book

- Check the Customer's orders

- Check out (payment method)

Use case2 (Manager's process)

- Sign in as Manager

- Create book (The Manager can edit and delete book details)

- Create product (The Manager can edit and delete product details)

- Manager can add another Manager

- Manager can delete and delete order details

- Manager can delete customer details

Requirements:

1. User-friendly interface: The system must have a user-friendly interface that is easy to navigate and understand. We use HTML, CSS and JavaScript.
2. Fast and reliable search: The system must have a fast and reliable search function that returns relevant results quickly.
3. Order tracking: The system must have an order tracking system that allows customers to view their order history and track the delivery of their purchases.
4. Back-end management: The system must have a back-end management system that allows administrators to manage the books in the store and view sales data.

We use the fastAPI framework from Python and initialize code and push it then on Github.

Assumptions:

1. Customers have access to the internet and can use a web browser to access the online book store.
2. The online book store's database is regularly backed up to prevent data loss in the event of a technical failure.
3. The online book store is in compliance with all relevant privacy and data protection regulations.

System Architecture Description and Implementation Overview:

The system architecture of the online book store is a complex network of interrelated components that work together to facilitate the storage, retrieval, and management of data. The architecture consists of two main components: the front end and the back end.

The front end is responsible for presenting the user interface and providing a way for users to interact with the data stored in the back end. It typically involves the use of web technologies such as HTML, CSS, and JavaScript to design and implement the user interface. The user interface is designed to provide an intuitive and user-friendly experience for customers, allowing them to browse and search for books, view detailed information about books, and place orders for books.

The back end is responsible for storing and managing the data associated with the online book store. It typically involves the use of a database management system (DBMS) to store and manage the data, as well as server-side programming languages such as Python to implement the business logic and processing of data. The back end provides a secure and reliable means of storing and managing the data associated with the online book store, and it enables the front end to retrieve and manipulate data in a manner that meets the requirements of the business.

The implementation is still in progress. Both Login and Register are implemented now. In conclusion, the system architecture of the online book store represents a complex and sophisticated system that requires careful planning, design, and implementation to ensure that it meets the requirements of the business and provides an optimal user experience for customers.

Test cases:

1. **Check connection to Swagger UI**
    - Start swagger page test
    - See all endpoint we created

2. **Registration and login test**
    - Go to the registration endpoint in the Swagger page.
    - Entre information in the forms.
    - Login with the same information to confirm the registration was successful.

3. **Add book to library**
    - Go to the login page to login to an account.
    - Press the *Add book* button to add a book.
    - Enter information (book's description, author,...)in the form.
    - Click *Add*

4. **Add product to the bookstore**
    - Press the *Add product* button to add a product for customer.
    - Enter information (price and quantity) in the form.
    - Click *Add*

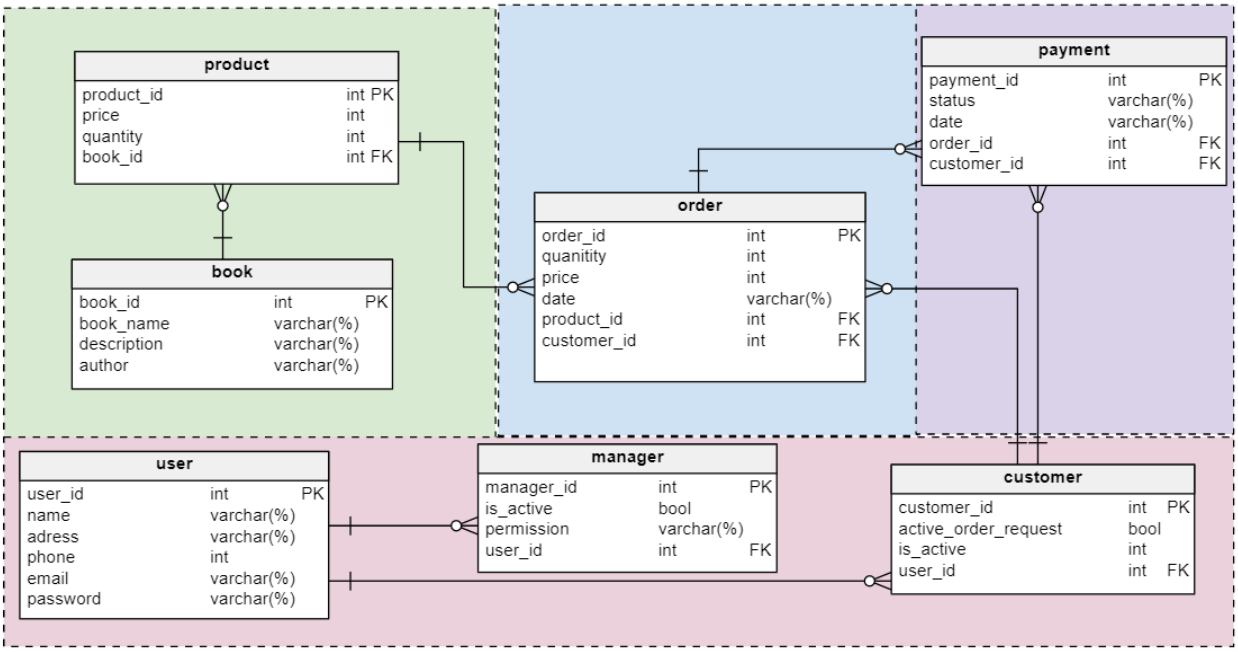
5. **Add order to the my order**
    - Press the *Add order* button to add an order for customer.
    - Select product and quantity in the form.
    - Click *Order*

6. **Delete book/product/order**
    - Go to the login page to login to an account.
    - Press the *Delete book/product/order* button to delete.
    - Enter the book/produkt/order ID.
    - Click Delete.
7. **In process**

Current backlog:

Group	URL	Status	Priority	Developer
	/login	implemented	critical	
	/register	implemented	critical	
	/password/update	implemented	low	
	/password/reset	implemented	low	
/auth	/validate-email	implemented	ignore	
	/home/	implemented	critical	
	/customer/update	implemented	critical	-
	/customer/get/	implemented	critical	-
	/customer/details	implemented	critical	-
	/customer/delete	implemented	critical	-
	/customer/add	implemented	critical	
/customer			~~ignore~~	
	/product/get/	implemented	critical	
	/product/details	implemented	critical	
	/product/delete	implemented	critical	
	/product/add	implemented	critical	
/admin	/payment/status	implemented	normal	
	/book/update	implemented	critical	
	/books/get/	implemented	critical	
	/book/details	implemented	critical	-
	/book/delete	implemented	critical	-
/books	/book/add	implemented	critical	-
	/order/update	implemented	critical	
	/orders/get/	implemented	critical	
	/order/details	implemented	critical	-
/order	/order/delete	implemented	critical	-
	/order/add	implemented	critical	-
	/Manager/update	implemented	critical	
	/Manager/get/	implemented	critical	
	/Manager/details	implemented	critical	
	/Manager/delete	implemented	critical	-
/manager	/Manager/home	implemented	critical	-

A database schema (E-R diagram):



Links to code:

https://github.com/1Ahm1/D0018e_project