

System requirement specifications document

Presented to: TA Sally Shaker



May 13, 2022

Ain SHAMS UNIVERSITY

**Presented by:**

Kirollos Rezk Berty 20P4188

Hamza Mazen Sayed Mohamed 20P6216

Amr Bassem Mohamed Dawoud 20P5518

Mohammed Ayman Abbas Zaki 19P9201

Ali Khaled Ahmed Ismail Ragy 20P1342

Contents

1. INTRODUCTION
   1. Purpose
   2. Scope
   3. Abbreviations and Terminologies
2. FEASABILITY ANALYSIS
   1. Hardware Technology
3. SYSTEM REQUIREMENTS
   1. Functional Requirements
   2. Non-Functional Requirements
4. REQUIREMENT VALIDATION
   1. Requirement Traceability Matrix
   2. Source Traceability Matrix
5. CONCLUSION
6. DIAGRAMS
   1. Architecture Models
      1. Description
      2. Diagrams
   2. Use-Case and Swimlane Diagrams
   3. Sequence Diagram
   4. State Diagram
   5. Component Diagram
   6. Class Diagram
7. INTRODUCTION
   1. The purpose of this EBOOK store is to provide ease of access to hundreds of books and different reading material at the press of a button. This system will allow its user to purchase a multitude of books, save them on his device to be accessed anytime, track his favorite literary works and save the more memorable quotes of their favorite books.

The system also includes features that encourage the user to read more often including an achievement system and a token awarding system, tokens can be later used to purchase any of the products provided by the store.

The Scope of this project is limited to the basic operations/functionalities of a library which

includes Searching for a book,

The Scope of this project is limited to the basic operations/functionalities of a library which

includes Searching for a book,

The Scope of this project is limited to the basic operations/functionalities of a library which

includes Searching for a book,

The Scope of this project is limited to the basic operations/functionalities of a library which

includes Searching for a book,

* 1. The Scope of this project is limited to the basic operations/functionalities of an online bookstore which includes Searching for a book, buying a book, checking author and editor information and navigating and highlighting text in a book.
* Users can search for any book by name, id, author name or category.
* Users can view their profiles and modify any personal information.
* Administrator can modify book database, book info and manage user accounts.

Abbreviations and Terminologies:

•LMS -- Library Management System

•CC – Credit Card

•DFD – Data Flow Diagram

•UML – Unified Modelling Language

•CRC – Class Responsibility Collaborators

•IDE – Integrated Development Environment

•OS – Operating System

•GUI – Graphical User Interface

* 1. Abbreviations and Terminologies:
* EBS – Electronic book store
* CC – Credit Card
* DFD – Data Flow Diagram
* UML – Unified Modelling Language
* IDE – Integrated Development Environment
* OS – Operating System
* GUI – Graphical User Interface

1. FEASIBILTY ANALYSIS
   1. Hardware Technology
2. Processor: Intel Core I5
3. Hard-disk Space: 40mb
4. Ram: 4GB
5. SYSTEM REQUIREMENTS
   1. Functional Requirements

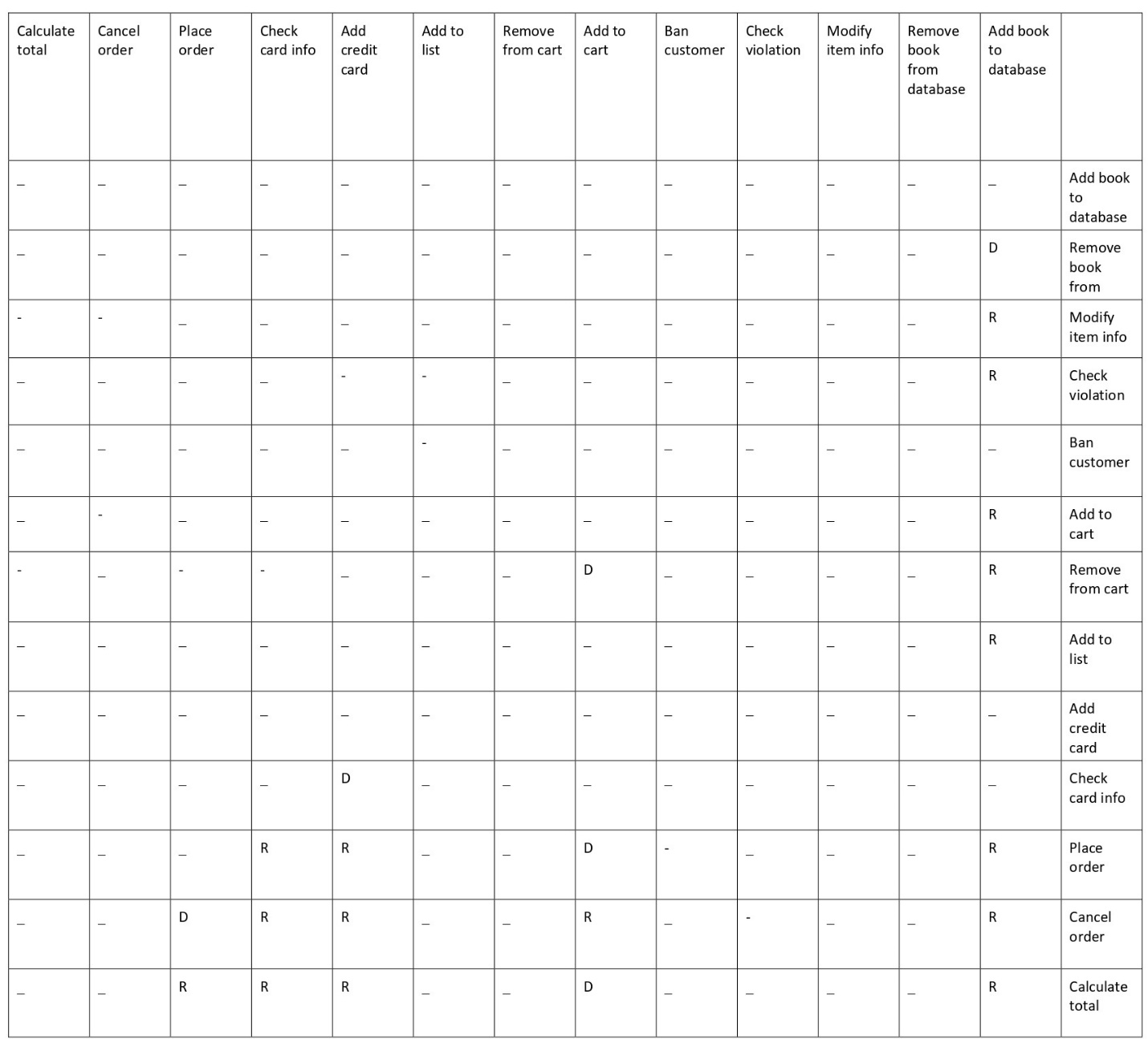
System must have unique usernames during login that distinguish normal user from admin, it will also allow user to search for books and navigate the system’s book database and to purchase and download books using credit card, as well as storing unbought books into a wide array of lists which includes wish lists, cart and favorites list.

User should be able to modify their personal info and check all acquired or hidden achievements as well as the number of tokens (currency rewarded upon completing achievements) and the rewards they unlock.

* 1. Non-Functional Requirements

System must process a database of all books offered in the store in pdf format,

It must respect international copyright protection rules when adding books to the database, it must also be compatible with Windows 10 and 11, and operate without internet connection allowing user to access already downloaded books.

1. REQUIREMENT VALIDATION
   1. Requirement Traceability Matrix
   2. Source Traceability Matrix

|  |  |  |
| --- | --- | --- |
|  | Customer | Adminstrator |
| Login | ✓ | ✓ |
| Add book to database |  | ✓ |
| Remove book from database |  | ✓ |
| Modify books info |  | ✓ |
| Check violation |  | ✓ |
| Ban customer |  | ✓ |
| add to cart | ✓ |  |
| Add to wishlist | ✓ |  |
| Add to favourites | ✓ |  |
| Purchase book | ✓ |  |
| View cart | ✓ |  |
| Review book | ✓ |  |
| View account | ✓ |  |
| Modify account | ✓ |  |
| Check achievements | ✓ |  |
| Check tokens | ✓ |  |
| Add bookmark | ✓ |  |
| Remove bookmark | ✓ |  |

1. CONCLUSION

In conclusion, the following document specified information regarding the nature of this software, its many functionalities as an E-book store including providing access to many books online and the ability to navigate and filter them with ease, it also specified how the software has affordable system requirements and is meant to run on almost all basic computers and laptops as well as requiring no internet.

The non-functional and functional requirements have been also specified and all the requirement relationships with each other have been included in the requirement traceability matrix.

1. ANALYSIS AND DESIGN
   1. Architecture Models:
      1. Description

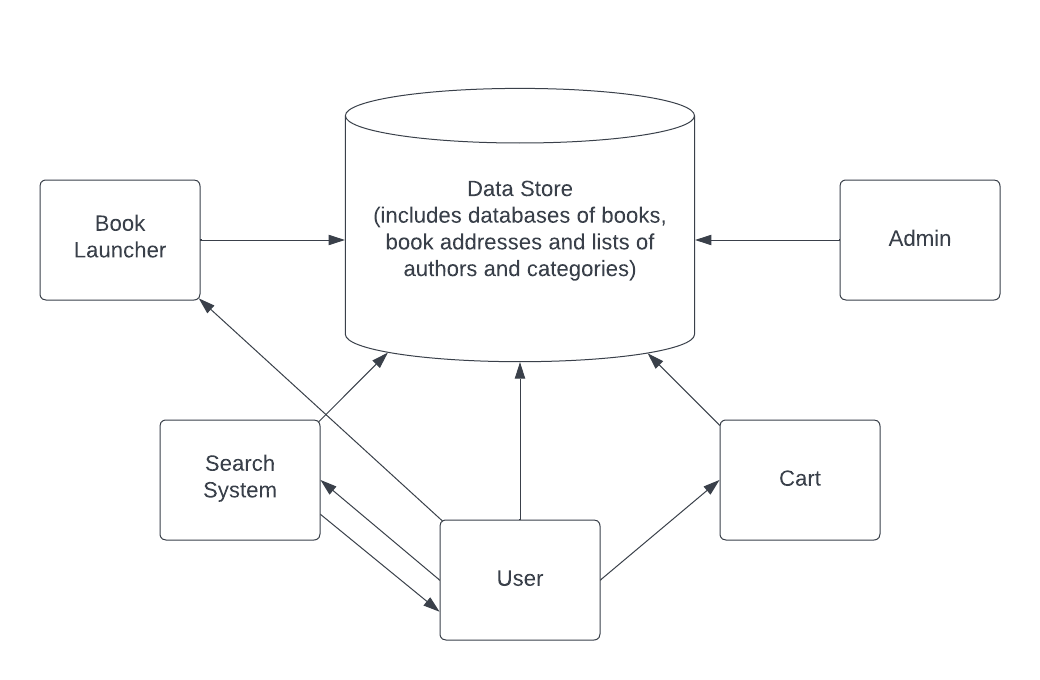
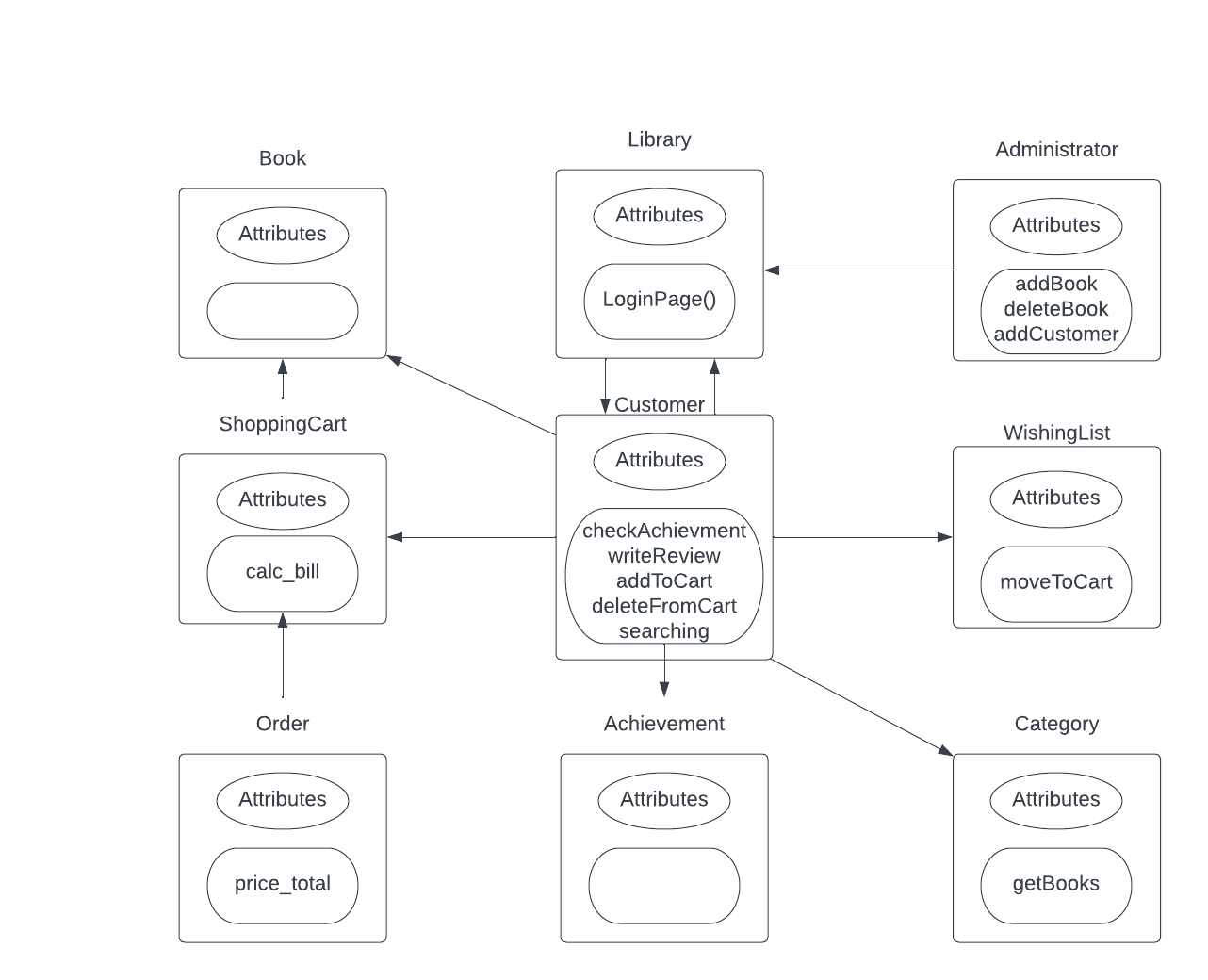
For the system architecture we chose to implement two models: the data centric model and the object oriented-model.

Regarding the data centric model, to perform a user command, the command will normally involve displaying or modifying a book or a group of books (in the form of a shopping cart or a Wishlist) which will involve in turn navigating through databases, to be precise each subsystem will navigate through a database to fulfil its purpose, each subsystem will also navigate through a database to interact with another subsystem since the interaction entails an exchange information, the user will add a book to his cart by fetching the book from the book database or modify his personal information by also accessing a database .

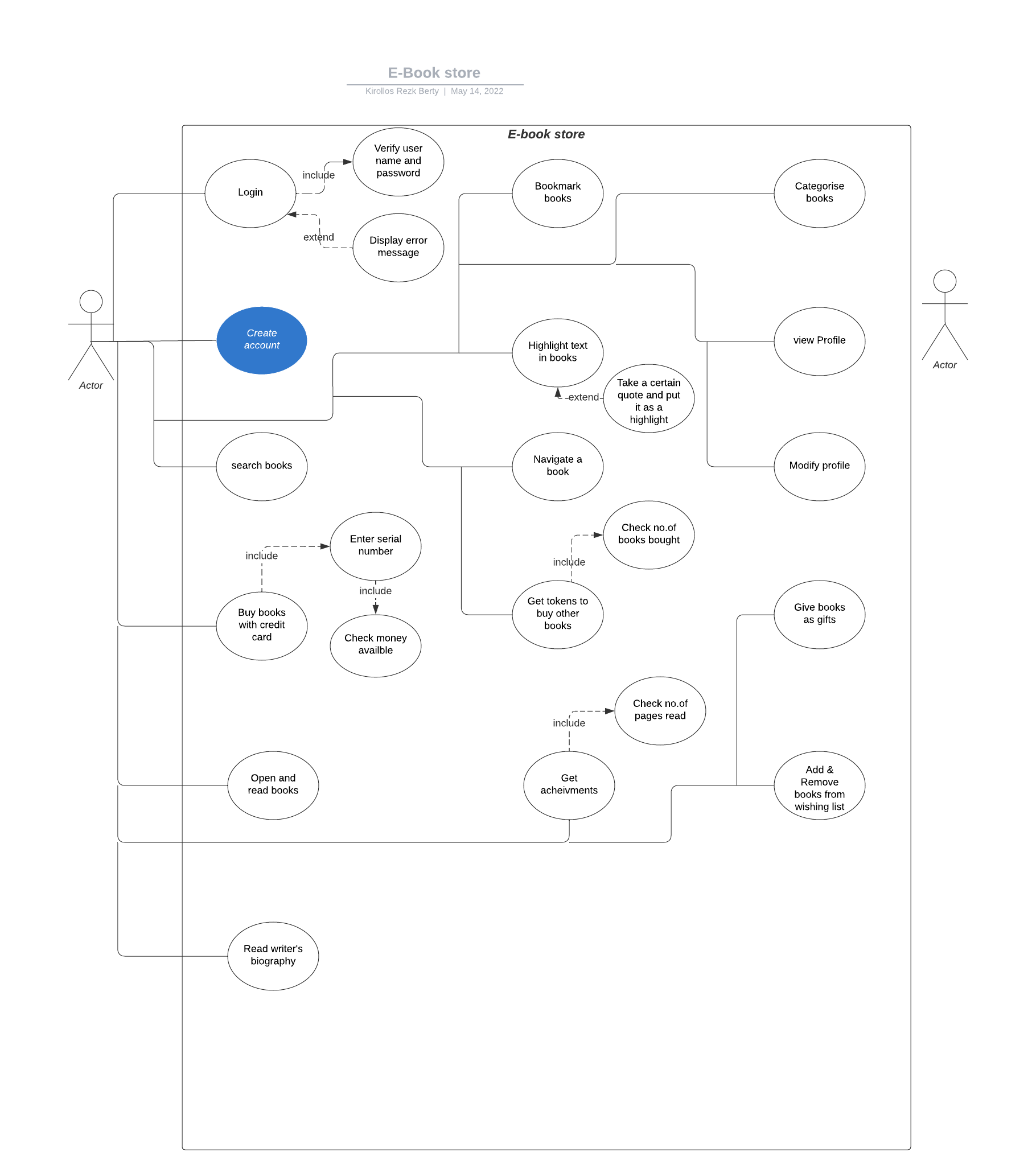
As for the object-oriented model, the logic behind this choice is the fact that we will use object-oriented programming to create the system.

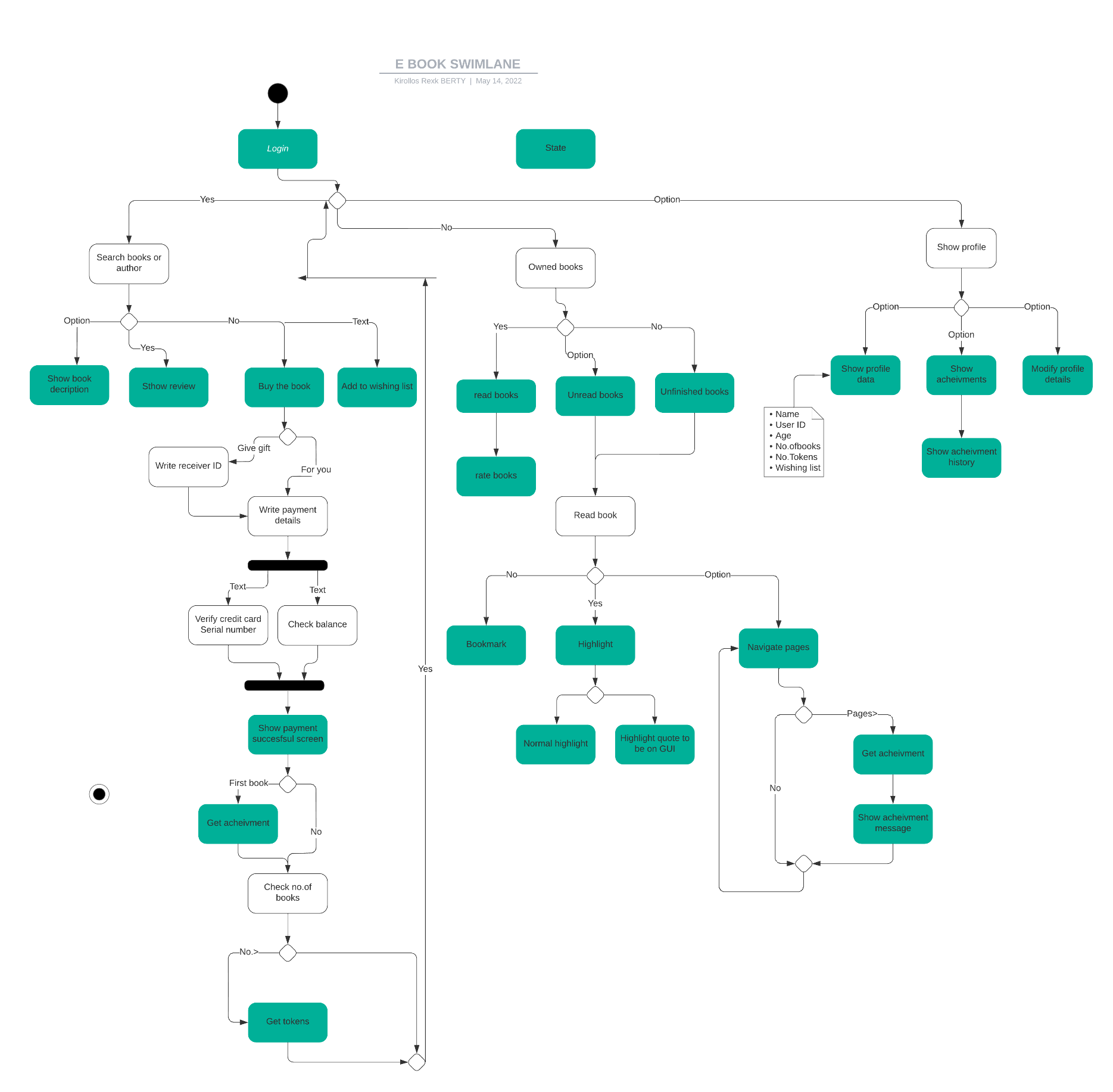
Our subsystems represent real-life objects (includes specific books and users) and classes such as books, customers, carts, etc... these objects will interact with each other through function call, for the customer to access a book he will have to use the book class’s respective getter or search using customer’s search method that will access the author or category classes which should both have list of books belonging to them corresponding to each author or category.

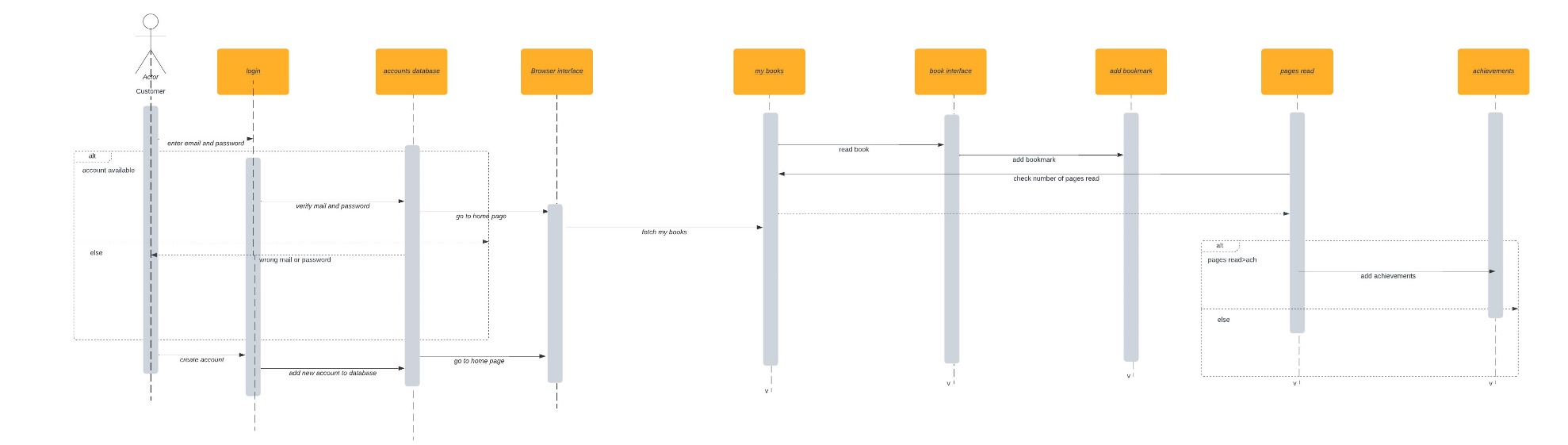
* + 1. Diagrams

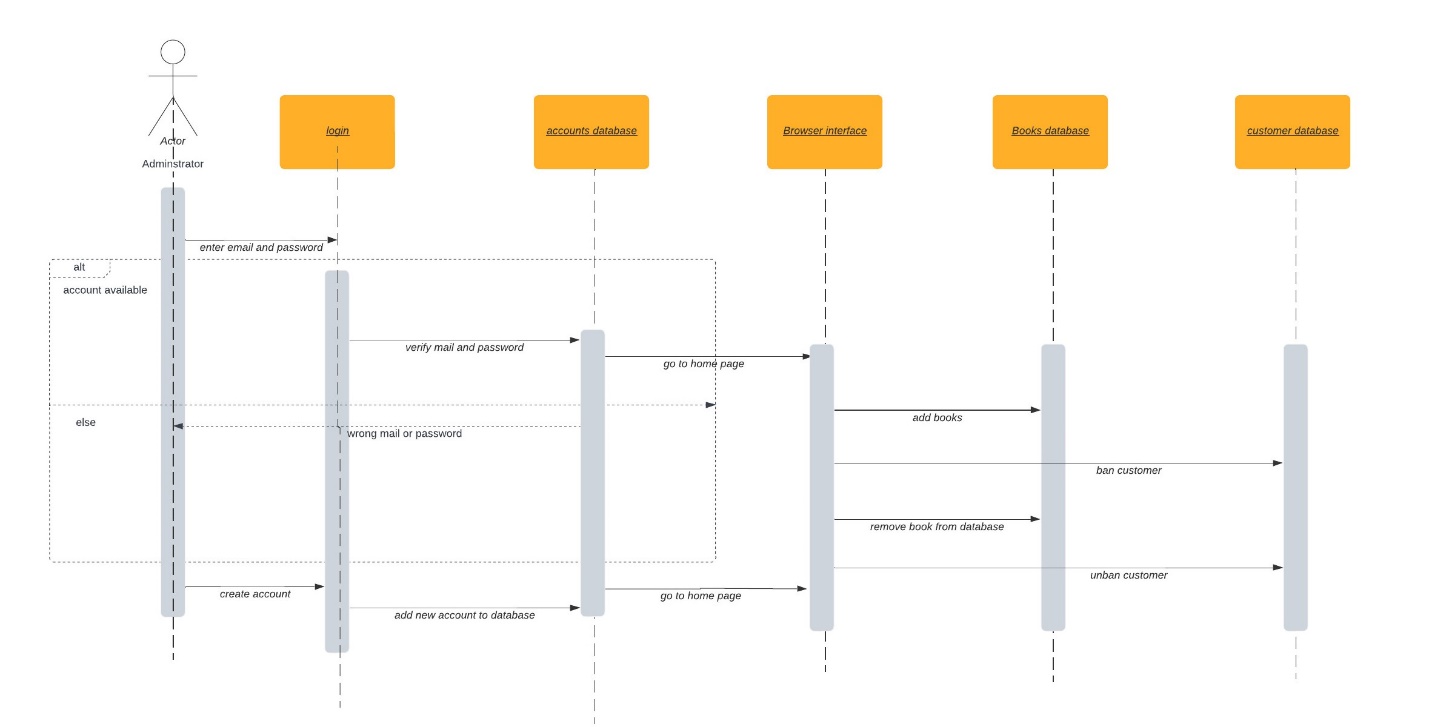


* 1. Use Case and Swimlane Diagrams:

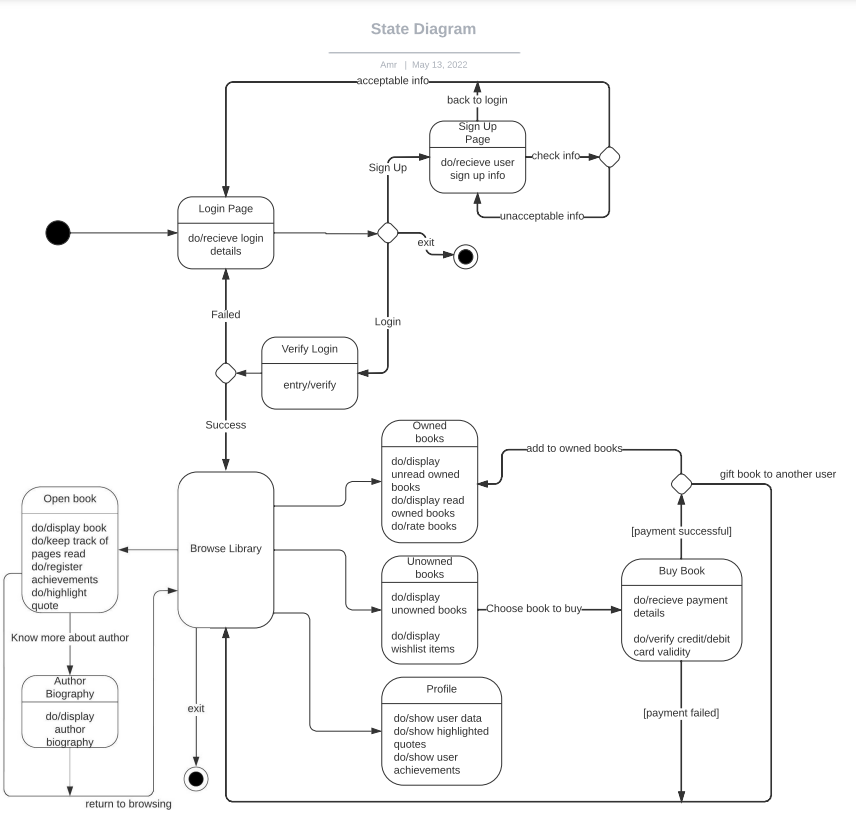


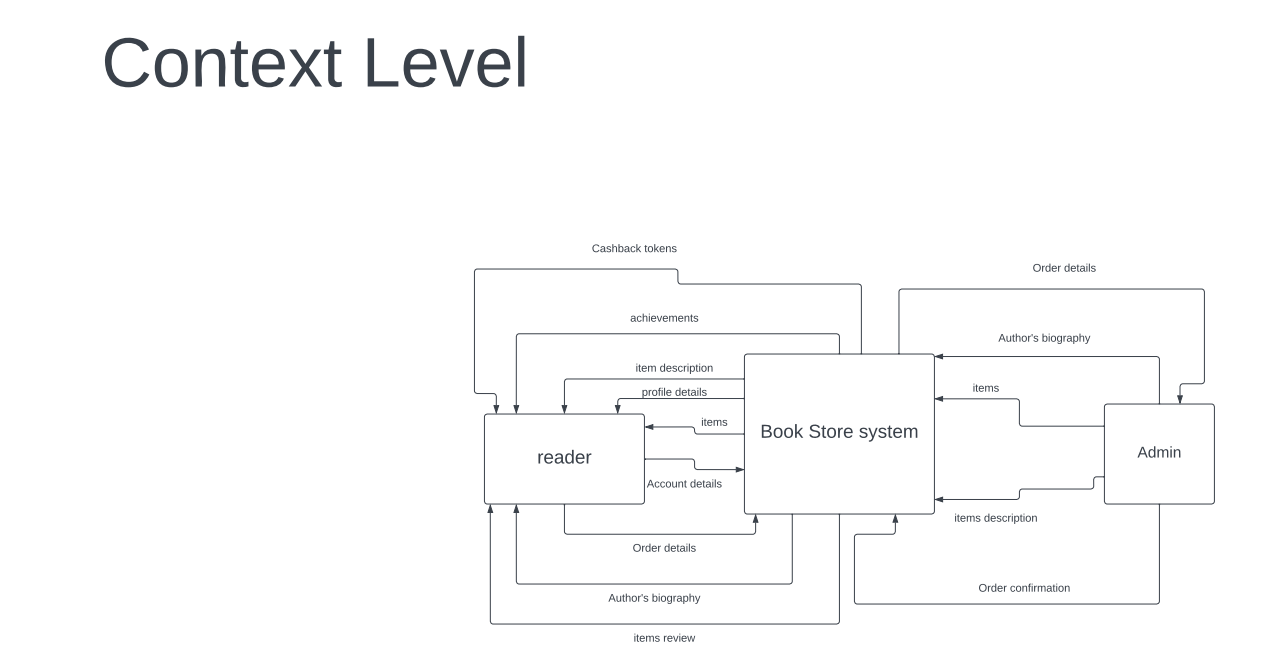
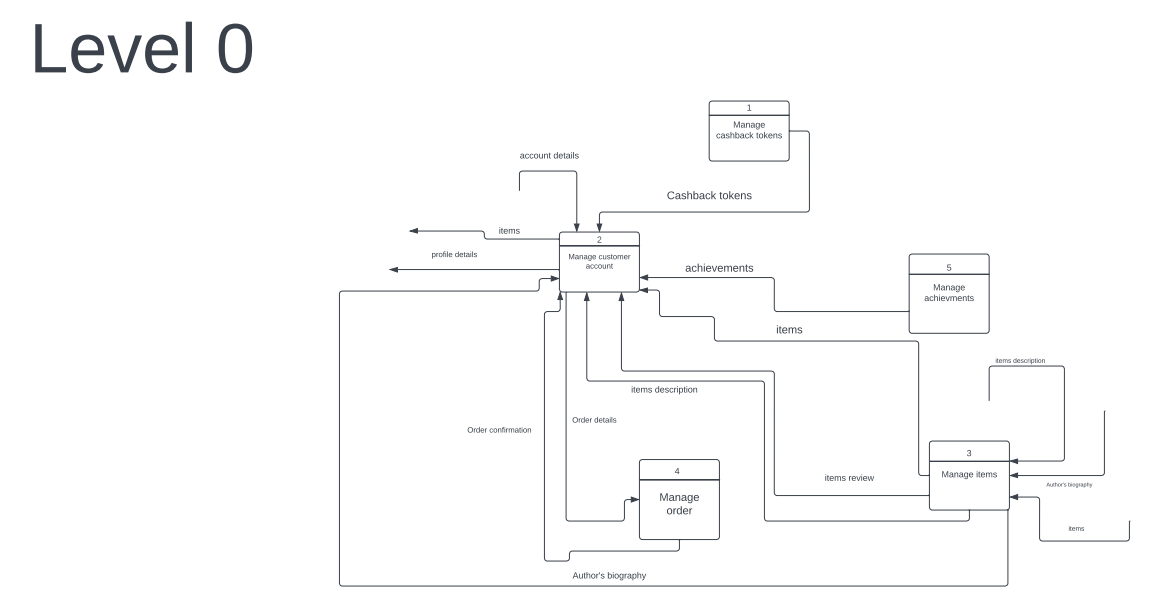


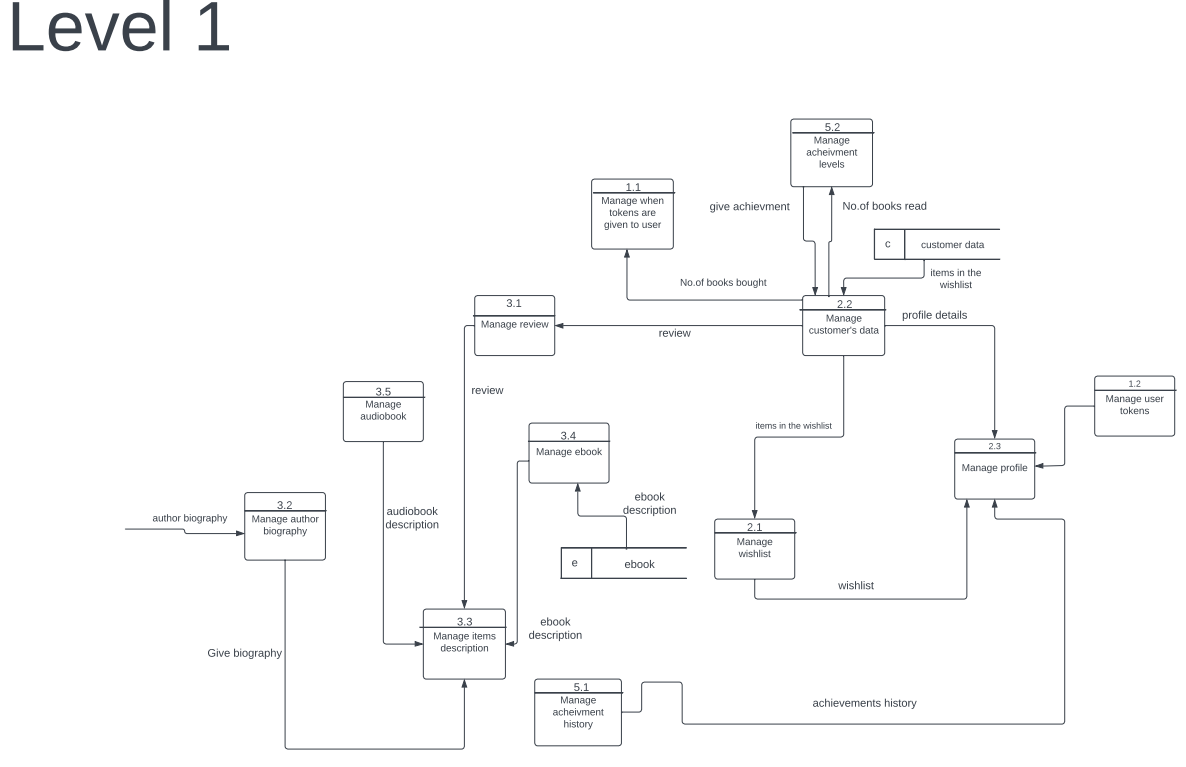
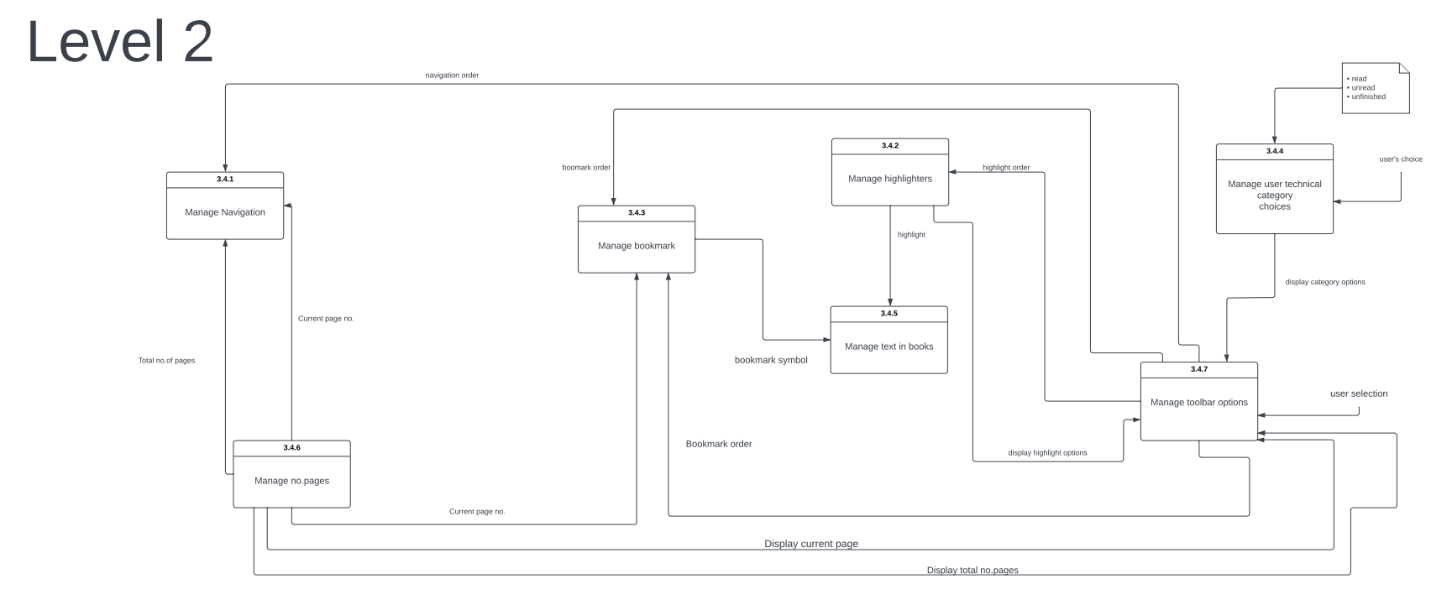
* 1. Sequence Diagram



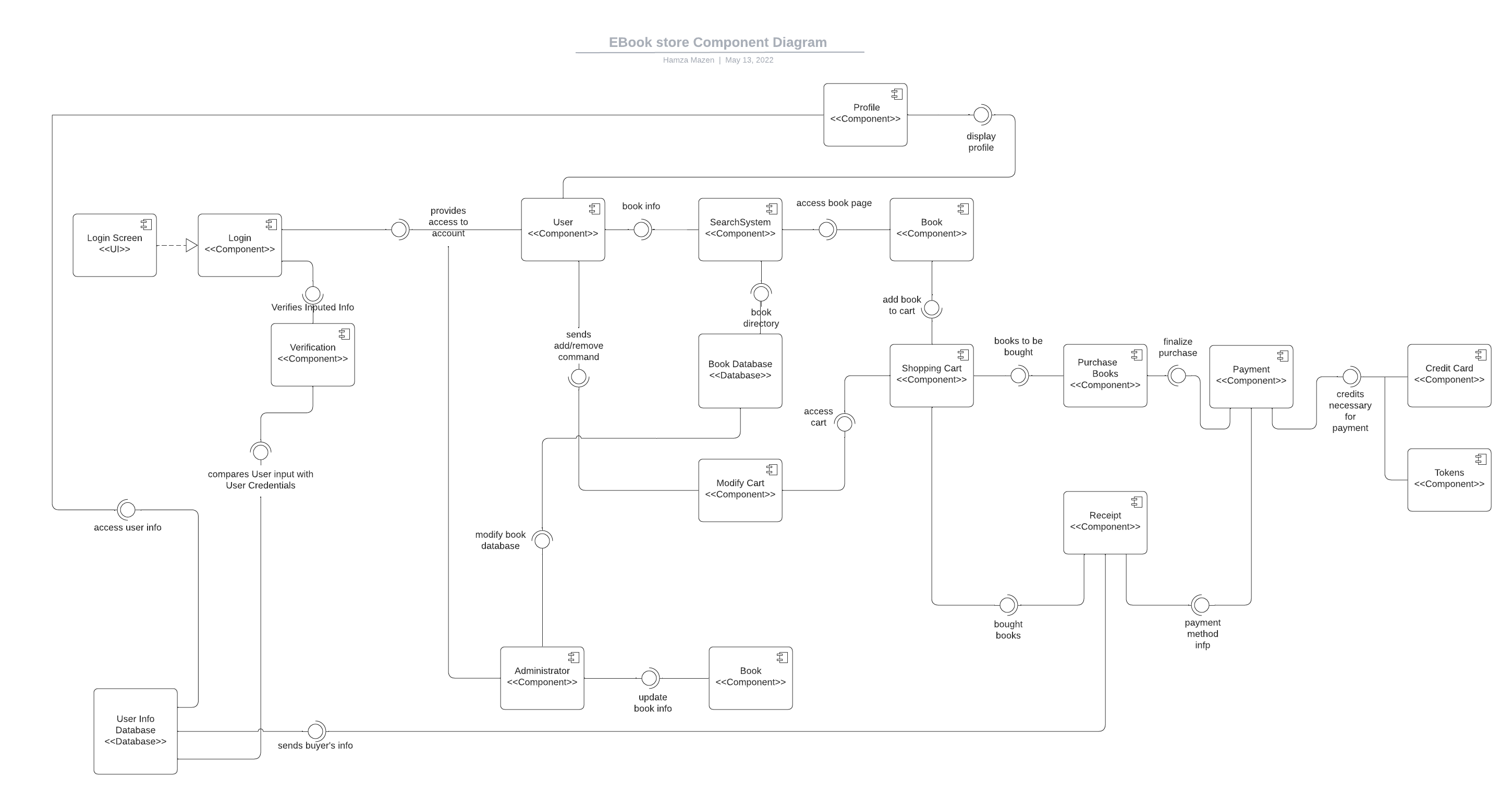
* 1. State Diagram

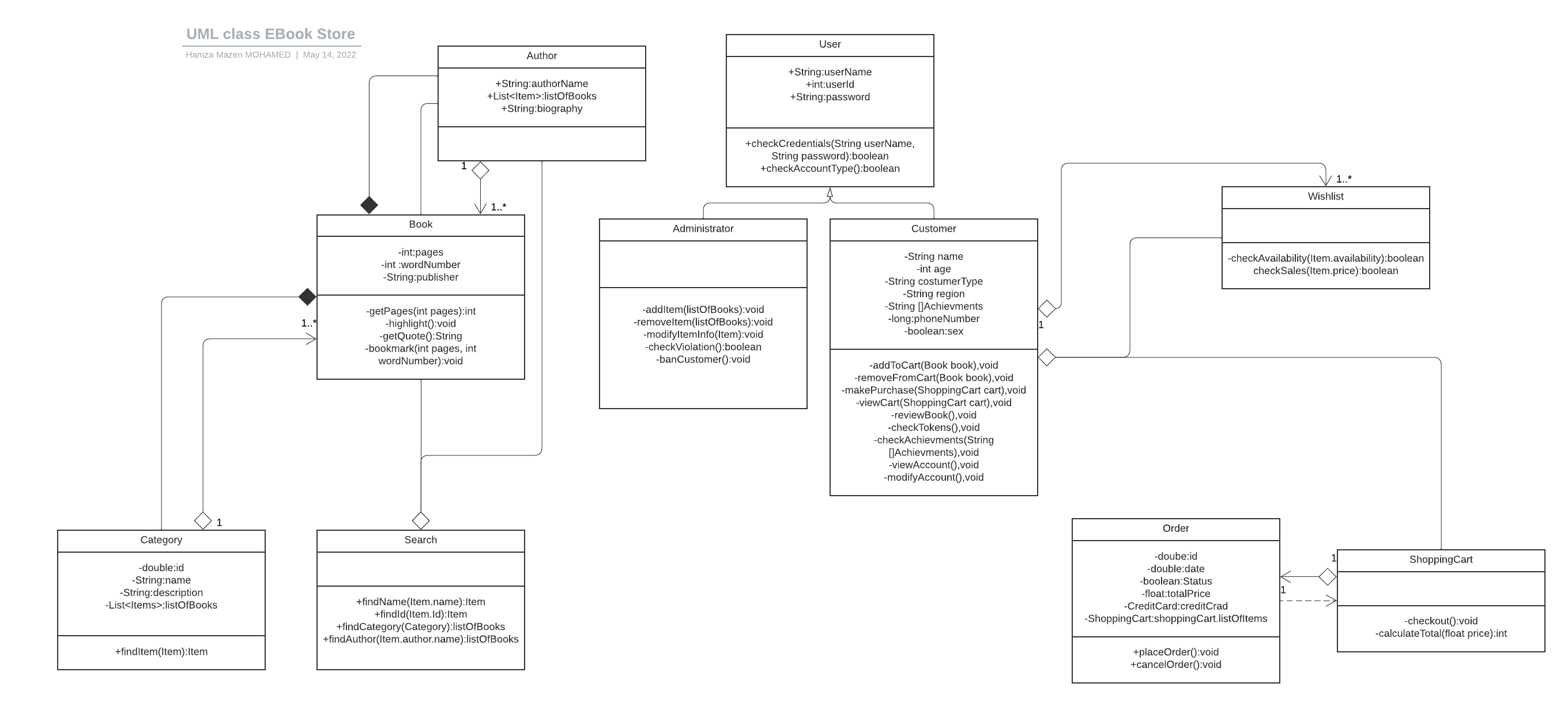


* 1. Data Flow Diagrams



* 1. Component Diagram





6.7 Class Diagram