BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC FPT**

------🙢⯎🙠*------*



**PRJ301 ASSIGNMENT REPORT**

**Online BookShop KAT**

**TEACHER STUDENT:**

**Phan Dang Cau Nguyen Thien Phuc RollNo: HE186083**

Major: Software Engineering

**Mar 2024**

table of content

[Table Of Contents i](#_Toc8022291)

[case study 1](#_Toc8022296)

[design 2](#_Toc8022300)

[2.1. Database Design 2](#_Toc8022302)

[2.1.1. Logical Design 2](#_Toc8022302)

[2.1.1. Physical Design 3](#_Toc8022302)

[2.2. System Design 6](#_Toc8022302)

[screenshot 8](#_Toc8022301)

[conclusion 3](#_Toc8022304)

CASE STUDY

The Bookshop Website: KAT

**KAT** is an online platform dedicated to promoting the love of literature by offering a wide array of books from various genres and authors. Our goal is to connect readers with captivating stories and thought-provoking literature while providing a seamless and convenient online shopping experience.

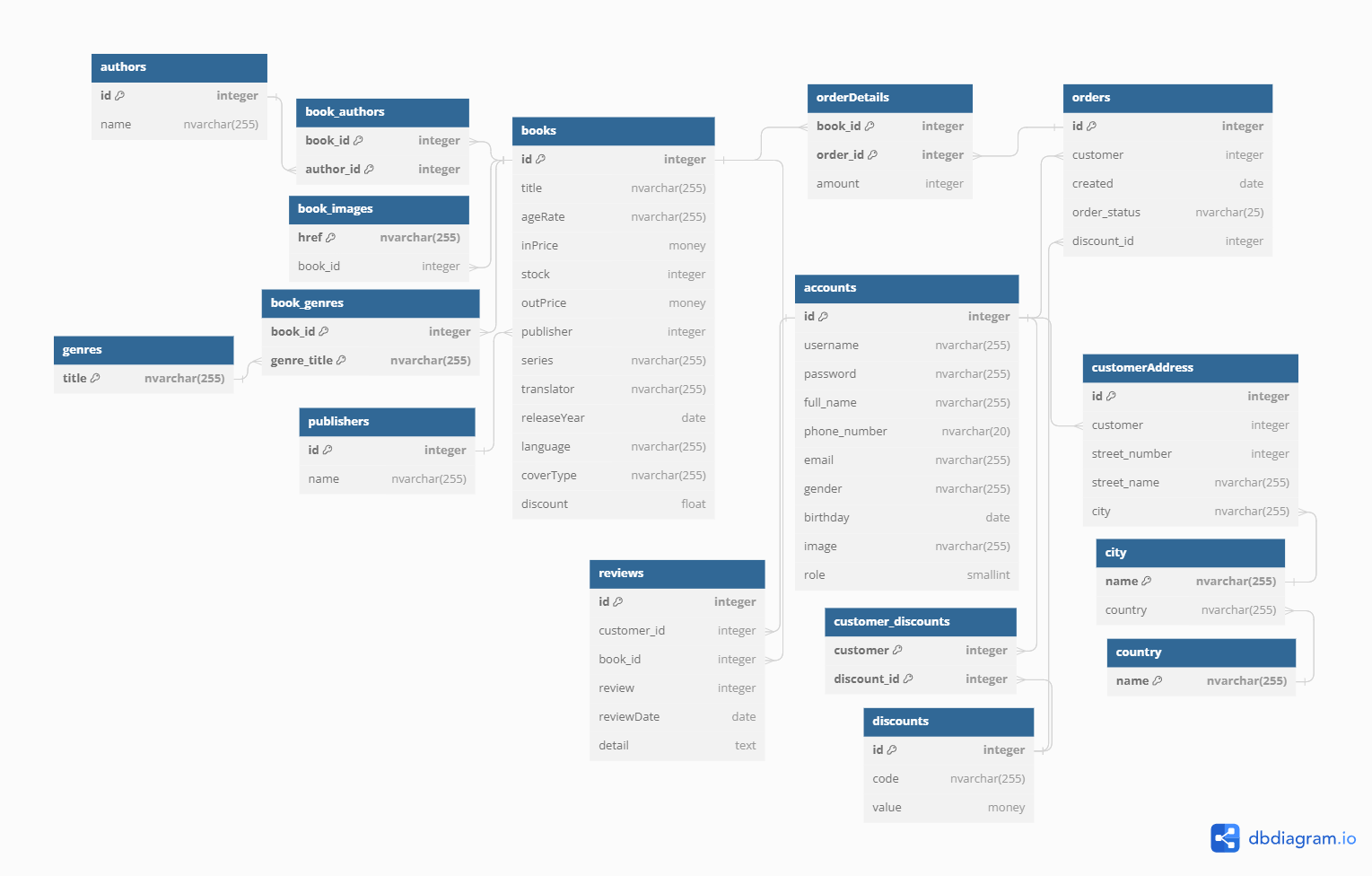
For customers, our platform offers a plethora of features to enhance their browsing and purchasing journey. They can easily explore different book collections,add their favorite titles to the cart, and seamlessly place orders with just a few clicks. Additionally, a contact form is readily available for any inquiries, ensuring that customers receive prompt assistance whenever needed.

For sellers, The Bookshelf Website offers a robust set of tools to manage their inventory, customer accounts, and orders efficiently. Sellers can easily add new books, update existing listings, and monitor stock levels to ensure a seamless shopping experience for customers. They also have access to valuable insights such as revenue and profit metrics, empowering them to make informed decisions to optimize their business operations.

In essence, The Bookshelf Website serves as a digital haven for book lovers, bridging the gap between readers and captivating literature while fostering a thriving online community centered around the joy of reading.

DESIGN

DATABASE DESIGN



Entity-Relationship Diagram (ERD)

2.1.1 Logical Design:

1. **Accounts:**

* **id:** A unique identifier for each account (Primary Key).
* **username:** The username associated with the account.
* **password:** The password for the account.
* **role:** Indicates the role of the account, such as customer or administrator.
* **email:** The email address linked to the account.
* **fullname:** The full name of the account holder.

2. **Authors:**

* **id:** Unique identifier for each author (Primary Key).
* **name:** The name of the author.

3. **Categories:**

* **id:** Unique identifier for each category (Primary Key).
* **title:** The title or name of the category.

4. **Publishers:**

* **id:** Unique identifier for each publisher (Primary Key).
* **name:** The name of the publisher.

5. **Books:**

* **id:** Unique identifier for each book (Primary Key).
* **title:** The title of the book.
* **price:** The price of the book.
* **stock:** The quantity of the book available in the inventory.
* **category\_id:** Foreign key referencing the category to which the book belongs.
* **publisher\_id:** Foreign key referencing the publisher of the book.
* **author\_id:** Foreign key referencing the author of the book.

6. **Book Images:**

* **href:** Unique identifier for each book image (Primary Key).
* **book\_id:** Foreign key referencing the book to which the image belongs.

7. **Cart Items:**

* **account\_id:** Foreign key referencing the account that added the item to the cart.
* **book\_id:** Foreign key referencing the book added to the cart.
* **amount:** The quantity of the book added to the cart.

**Relationships:**

* Each account can have multiple cart items and orders.
* Each author can have written multiple books.
* Each category can have multiple books.
* Each publisher can have published multiple books.
* Each book can have multiple images, be in multiple cart items, and have multiple order details.

This detailed description provides a comprehensive understanding of the entities, attributes, and relationships within the Book Shop database, facilitating effective management and utilization of the data.

2.1.2 Physical Design

1. Table: accounts

* **id:** INT (Primary Key, Identity)
* **username:** VARCHAR(255) NOT NULL
* **password:** VARCHAR(255) NOT NULL
* **role:** INT NOT NULL
* **email:** NVARCHAR(255)
* **fullname:** NVARCHAR(255)

2. Table: authors

* **id:** INT (Primary Key, Identity)
* **name:** NVARCHAR(255)

3. Table: categories

* **id:** INT (Primary Key, Identity)
* **title:** NVARCHAR(255)

4. Table: publishers

* **id:** INT (Primary Key, Identity)
* **name:** NVARCHAR(255)

5. Table: books

* **id:** INT (Primary Key, Identity)
* **title:** NVARCHAR(255)
* **price:** MONEY
* **stock:** INT
* **category\_id:** INT (Foreign Key referencing categories.id)
* **publisher\_id:** INT (Foreign Key referencing publishers.id)
* **author\_id:** INT (Foreign Key referencing authors.id)

6. Table: book\_images

* **href:** NVARCHAR(255) (Primary Key)
* **book\_id:** INT (Foreign Key referencing books.id)

7. Table: cart\_items

* **account\_id:** INT (Foreign Key referencing accounts.id)
* **book\_id:** INT (Foreign Key referencing books.id)
* **amount:** INT
* **Primary Key:** (account\_id, book\_id)

Indexes:

* Primary Keys: Automatically create clustered indexes.
* Foreign Keys: Automatically create non-clustered indexes.
* Additional indexes may be considered based on query patterns and performance optimization needs.

Constraints:

* Foreign Key Constraints: Ensure referential integrity between related tables.
* Primary Key Constraints: Ensure uniqueness and identify each record uniquely.
* NOT NULL Constraints: Ensure mandatory fields are not left empty where required.

Relationships:

* One-to-Many Relationships:
  + One author can write multiple books.
  + One category can have multiple books.
  + One publisher can publish multiple books.
  + One book can have multiple images.
  + One account can have multiple cart items.
* Many-to-One Relationships:
  + Many books belong to one category.
  + Many books are published by one publisher.
  + Many books are written by one author.
  + Many images belong to one book.
  + Many cart items belong to one account.

SYSTEM DESIGN

The BookShop KAT websites adheres to the MVC architecture

Models:

* **DAO (Data Access Object) Classes:**
  + Each business object typically has a corresponding DAO class responsible for interacting with the database to perform CRUD (Create, Read, Update, Delete) operations.
  + For example, this have ProductDAO, SizeDAO, CategoryDAO, OrderDAO, CustomerDAO, and ReviewDAO classes, each containing methods to retrieve, insert, update, and delete records in their respective database tables.
* Models:
  + Accounts
  + Author
  + Book
  + Cart
  + Category
  + Publisher
  + Item

Controllers:

* **Servlets (Controllers):**
* Servlets act as controllers in this model, handling incoming requests from clients, processing data, and invoking appropriate DAO methods to interact with the database.
* For example, it contains ProductServlet, CollectionsServlet , LoginServlet, …, each responsible for handling requests related to their corresponding business objects.
* **WebManager (Singleton):**
  + To lower of all the creation whenever we call DAO,

Views:

* **JSP Files (Views):**
* JSP files serve as views in this model, responsible for presenting data to users in a user-friendly format.
* These files lies in web folder. For example, this have index.jsp, cart.jsp, accountDetail.jsp, or checkOut.jsp files, each containing HTML markup mixed with Java code (JSP tags) to dynamically display data retrieved from the servlets/controllers.

Overall, this model follows the MVC (Model-View-Controller) architecture, where business objects represent the model layer, DAO classes handle data access operations, servlets act as controllers to handle business logic and requests, and JSP files serve as views to render the user interface. The database design aligns with this model by providing a structured schema to store and manage data related to shoe products, orders, customers, reviews, and categories.

Sequence Diagram:

Coding Convention:

**Naming convention:**

(Follow <https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html>)

* Class: UpperCamelCase, noun (Dog, Cat)
* Variable: lowerCamelCase, noun (numberOfLegs, SpeciesGenus)
* Method: lowerCamelCase, verb (run, growl, climbTree)
* Interface: IUpperCamelCase, adjective. If used to share a behaviour/verb, able (I4Legs, IMovable)

SCREENSHOT

Shop Page

A screenshot of a computer

Description automatically generated

Login Page

A screenshot of a computer

Description automatically generatedRegister Page

A screenshot of a computer

Description automatically generated

Account Details Page

A screenshot of a computer

Description automatically generated

Category Page

A screenshot of a book

Description automatically generatedCart Page

A screenshot of a computer

Description automatically generated

Checkout Page

A screenshot of a computer

Description automatically generated

Admin Page

A screenshot of a computer

Description automatically generated

BOOK MANAGER

A group of icons on a white background

Description automatically generated

Add Book Page

A screenshot of a computer

Description automatically generated

Account Manager

A screenshot of a computer

Description automatically generated

Edit Account

A screenshot of a computer

Description automatically generated

CONCLUSION

**Pros**: The Book Shop website serves as a digital platform to selling different books, providing customers with a seamless shopping experience with a decent number of features that a book selling website would provide.

**Cons**: The design of the website is not 100% clean since this is the first time, I have engaged in building a website from scratch. The application has not been optimized yet and at some case take a little time to load.

**Learning Experience:** Developing the Book Shop KAT website offers me insights into designing and building a complete website, giving me more experience, both coding technical and report documenting.

**Future Improvements:** Future enhancements could focus on enhancing customer engagement through interactive features such as chat bot or direct messaging through Facebook Messenger, implementing features like Favorite or Wishlist to help seller get more data from customer to develop furthermore their brand.