

Bookstore Project rEPORT

COMP 3005 Project (Fall 2021)



December 18, 2021

Group Members:

Moiz Abbasi

Mohamed AbouZiab

Mutaz Aldisi

Table of Contents

[**Conceptual Design** 1](#_Toc90375048)

[**Reduction to Relation Schemas** 1](#_Toc90375049)

[**1.** **User** 1](#_Toc90375050)

[**2.** **Basket** 1](#_Toc90375051)

[**3.** **Author** 2](#_Toc90375052)

[**4.** **Publisher** 2](#_Toc90375053)

[**5.** **Book** 2](#_Toc90375054)

[**6.** **BasketBook** 2](#_Toc90375055)

[**Normalization of Relation Schema** 2](#_Toc90375056)

[**1.** **User** 2](#_Toc90375057)

[**2.** **Basket** 2](#_Toc90375058)

[**3.** **Author** 3](#_Toc90375059)

[**4.** **Publisher** 3](#_Toc90375060)

[**5.** **Book** 3](#_Toc90375061)

[**6.** **BasketBook** 3](#_Toc90375062)

[**Database Schema Diagram** 3](#_Toc90375063)

[**Implementation** 4](#_Toc90375064)

[**Tools and Technologies** 4](#_Toc90375065)

[**Functional Requirements** 5](#_Toc90375066)

[**Customer Application** 5](#_Toc90375067)

[**1.** **Register** 5](#_Toc90375068)

[**2.** **Login** 5](#_Toc90375069)

[**3.** **Dashboard** 6](#_Toc90375070)

[**4.** **Track Order** 6](#_Toc90375071)

[**Admin Application** 7](#_Toc90375072)

[**1.** **Manage Books** 7](#_Toc90375073)

[**2.** **Manage Publishers** 7](#_Toc90375074)

[**3.** **Manage Publishers** 8](#_Toc90375075)

[**4.** **Manage Orders** 8](#_Toc90375076)

# **Conceptual Design**

Here is the conceptual design ER diagram.



# **Reduction to Relation Schemas**

## **User**

* username PK
* password
* address
* gender

## **Basket**

* id PK
* username FK
* ship\_address
* status

## **Author**

* name PK
* address

## **Publisher**

* name PK
* address PK
* phone

## **Book**

* isbn PK
* title
* year
* pages
* price
* genre
* count
* auth\_name FK
* pub\_name FK
* pub\_address FK

## **BasketBook**

* basket\_id PK
* book\_isbn PK, FK
* count

# **Normalization of Relation Schema**

According to the design given above it is clear that database is clearly in its normalized form. Many to many relations are divided into one to many relations. There are several functional dependencies for each table.

## **User**

All the information in this table depends on username.

## **Basket**

Information in this table can be accessed using id of the order. Information can also be accessed using username that is foreign key in this table from User table.

## **Author**

Author information depends on author name.

## **Publisher**

Publisher’s information depends on both name and address as composite primary key.

## **Book**

Book information can be accessed using its ISBN number as primary key which returns only one book. But books can also be accessed using author or publisher name that are foreign keys.

## **BasketBook**

Basket information can be accessed using basket id and also username.

# **Database Schema Diagram**

Final database schema diagram is given below.

Diagram

Description automatically generated

# **Implementation**

This application is desktop based and written using C#. It is developed using Visual Studio IDE to provide the most out of it. This project comprises of two applications, one for administration and other one for users.

## **Tools and Technologies**

Following tools and technologies are used in this project.

* .NET Framework
* PostgreSQL

## **Functional Requirements**

Following are the requirements of the system which state that what each module in the applications can do or should do.

* Add, remove, or update a book
* Add authors
* Add publishers
* Update orders
* Register a user
* Login a user
* Add books to basket
* Delete basket
* Checkout a basket
* Track use orders

Above mentioned functional requirements are all in working condition now.

## **Customer Application**

### **Register**

In this form, a new user ca signs up to get most out of the application and order his loved books.

**Graphical user interface, application

Description automatically generated**

### **Login**

To login, a user needs to be already registered. His credentials are matched in the database and if exists, he is redirected to the Dashboard where he can order new books and track his orders.

Graphical user interface, application

Description automatically generated

### **Dashboard**

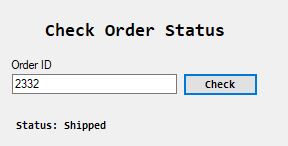
In dashboard, customer can add books to basket and checkout. He is required to input the shipping address before checkout. Here, he can find an option to track his all orders using tracking id.

Graphical user interface, application

Description automatically generated

### **Track Order**

In this section, the customer can track his all previous order by providing order tracking id.



## **Admin Application**

### **Manage Books**

In this section, an admin can access all the books. He can delete any one of them and can add new book also. He is provided with an option to update the stock for a specific book.

Graphical user interface, table

Description automatically generated

### **Manage Publishers**

In this section, new publisher can be added, and the admin is presented with a list of all registered publishers in the system.

Graphical user interface, application, email

Description automatically generated

### **Manage Publishers**

In this section, new author can be added, and the admin is presented with a list of all registered authors in the system.

Graphical user interface, text, application

Description automatically generated

### **Manage Orders**

This is the part, which interacts with the customer application module. All customers’ orders are listed here, and the admin can update the status of any selected order and those changes are reflected in the customer side also.

Graphical user interface, text, application

Description automatically generated