SW2 – Project Evaluation Form

- **Each team must submit the following Documentation that contains:**
  - **Project Description in detail.**
  - **Class Diagram. And Database Schema.**
- **Each team must submit the project via GitHub: - Source Code.**
  - **Video Demo for running (2 – 5 Minutes). - Documentation and Evaluation Form.**
- **The Evaluation will start with giving all teams 30 marks then check the following criteria:**

Violation Level	Full	Medium	Sm
Documentation	**-5**	**-2**	**-**
Not Apply MVC (it does not Separate Business logic from GUI). Example of violation: write the implantation for a method such as an inset item into the database inside the Button Action method)	**-6**	**- 3**	**-**
Violate clean code – Variables	**-2**	**-1**	**-.**
Violate clean code – Functions	**-2**	**-1**	**-.**
Violate Single-responsibility Principle	**-2**	**-1**	**-.**
Violate Open-closed Principle	**-2**	**-1**	**-.**

Violate the Liskov Substitution Principle	**-2**	**-1**	**-.0**
Violate Interface Segregation Principle	**-2**	**-1**	**-.0**
Violate Dependency Inversion Principle	**-2**	**-1**	**-.0**
Not Upload code to GitHub	**-1**		
Only One Branch Without Merge (GitHub)	**-2**		
Only One Contribution (GitHub)	**-2**		
Total Minus from Grade			

Design Pattern Bounce	**+4**
Bounce on Overall Work	**+2**
Total Team Grade / 30	

Full violated –They violate the principle in all cases in the Code.

Medium Violated – They apply the principle in some cases and violate it in some cases (in total the student applies the principle on average).

Small violated – They Apply the principle in most cases except for a very few cases, no more than one or two.

Name (Arabic)	ID	Individual Bounce +2	Grade
حبيبة خالد صالح حسن	201900258		
جهاد إبراهيم عبد العزيز الملاح	201900243		
تقى سلامة احمد	201900237		
سارة محمد جواتشى	201900334		
ريم محمد حسين	201900317		

Online bookstore:

A) Project Description:

-The main objective of the project is to create an online bookstore that allows users to search and purchase a book based on title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. The Administrator will have additional functionalities when compared to the common user, admin can update, delete and approve a transaction for buyers.
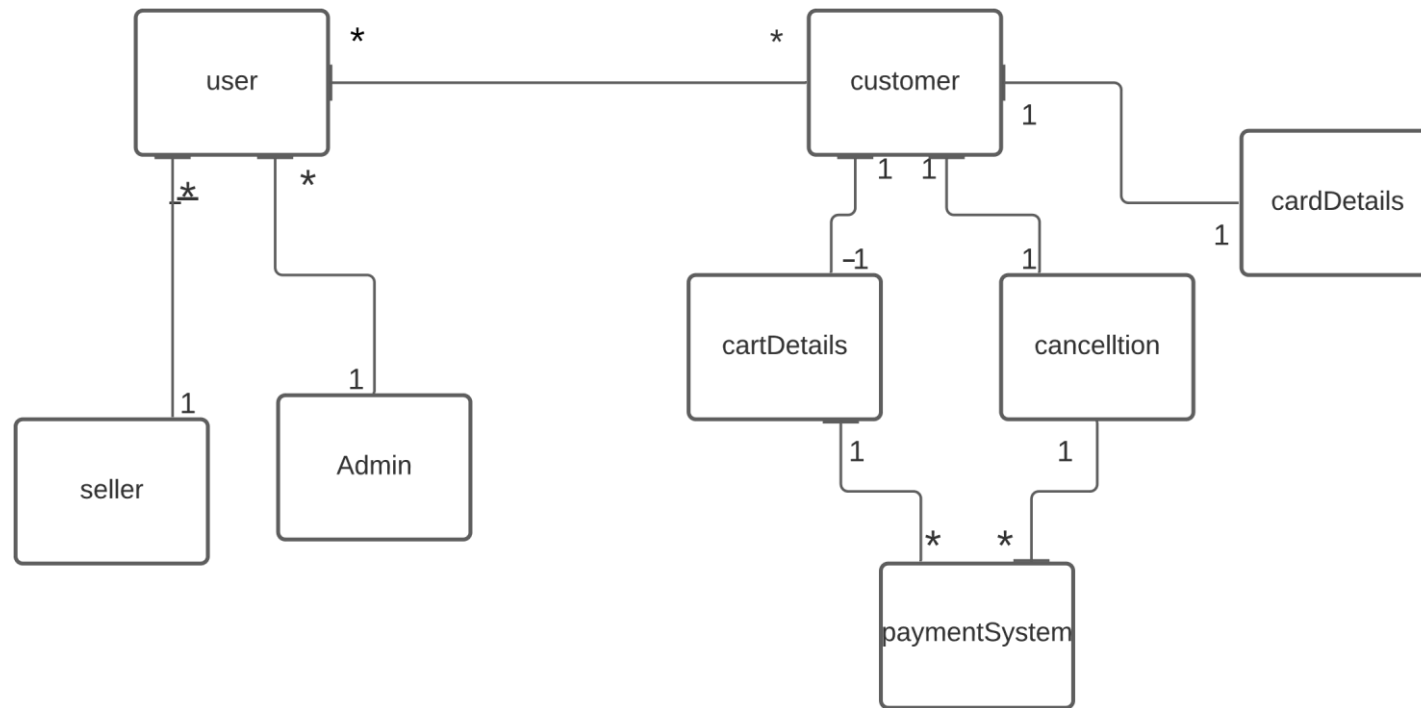
-There are many online bookstores like Powell's, Amazon which were designed using Html. I want to develop a similar website using .NET, SQL Server.

Online Book store is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. -after a successful login. It will display all the book categories and will have a search keyword option to search for the required book.
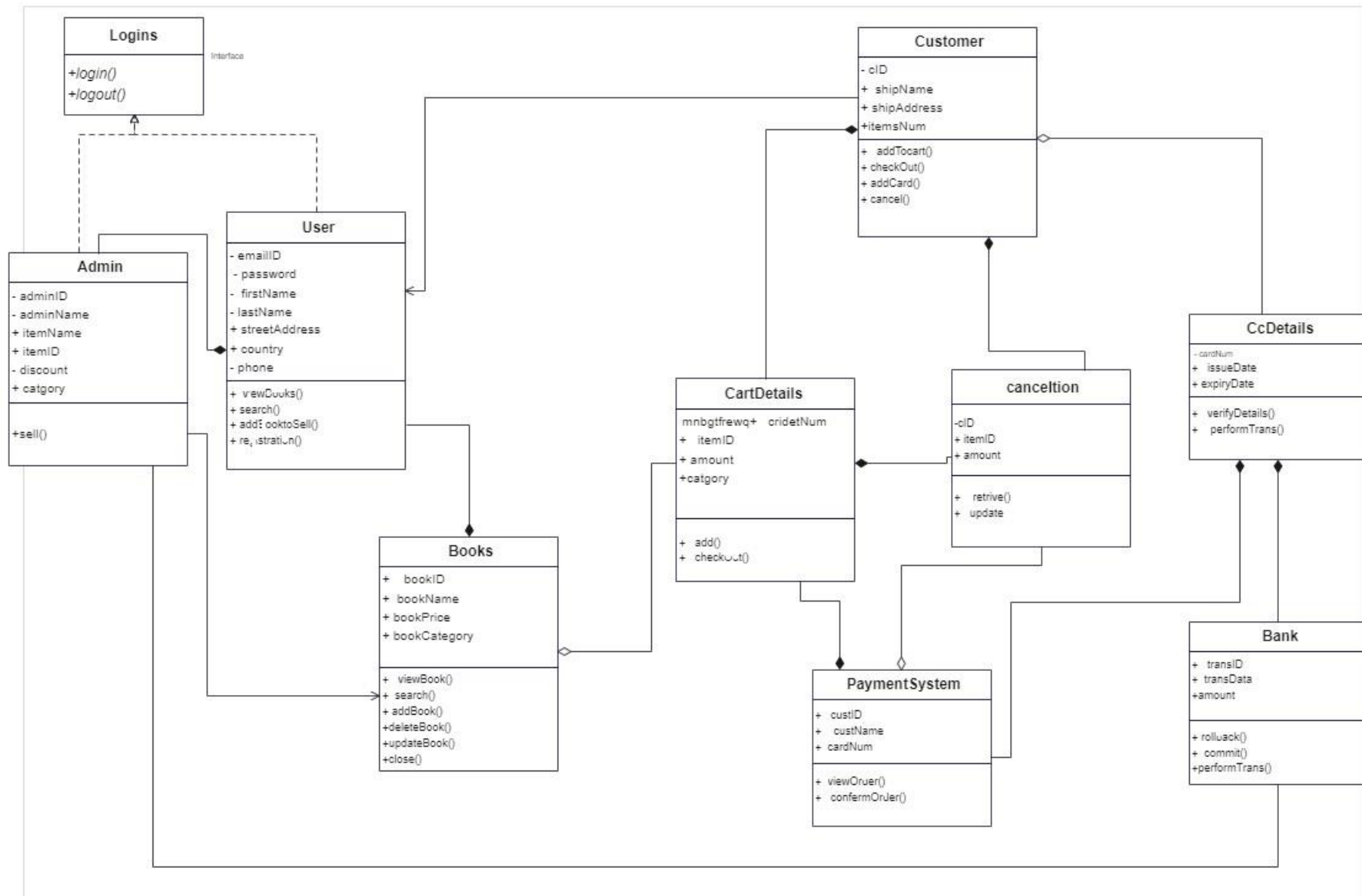
- after customers view the books and choose what they need, they can manage a shopping cart which will include all the books they selected. The user can edit, delete and update his shopping cart. A final shopping cart summary is displayed which includes all the items the user selected and the final total cost.
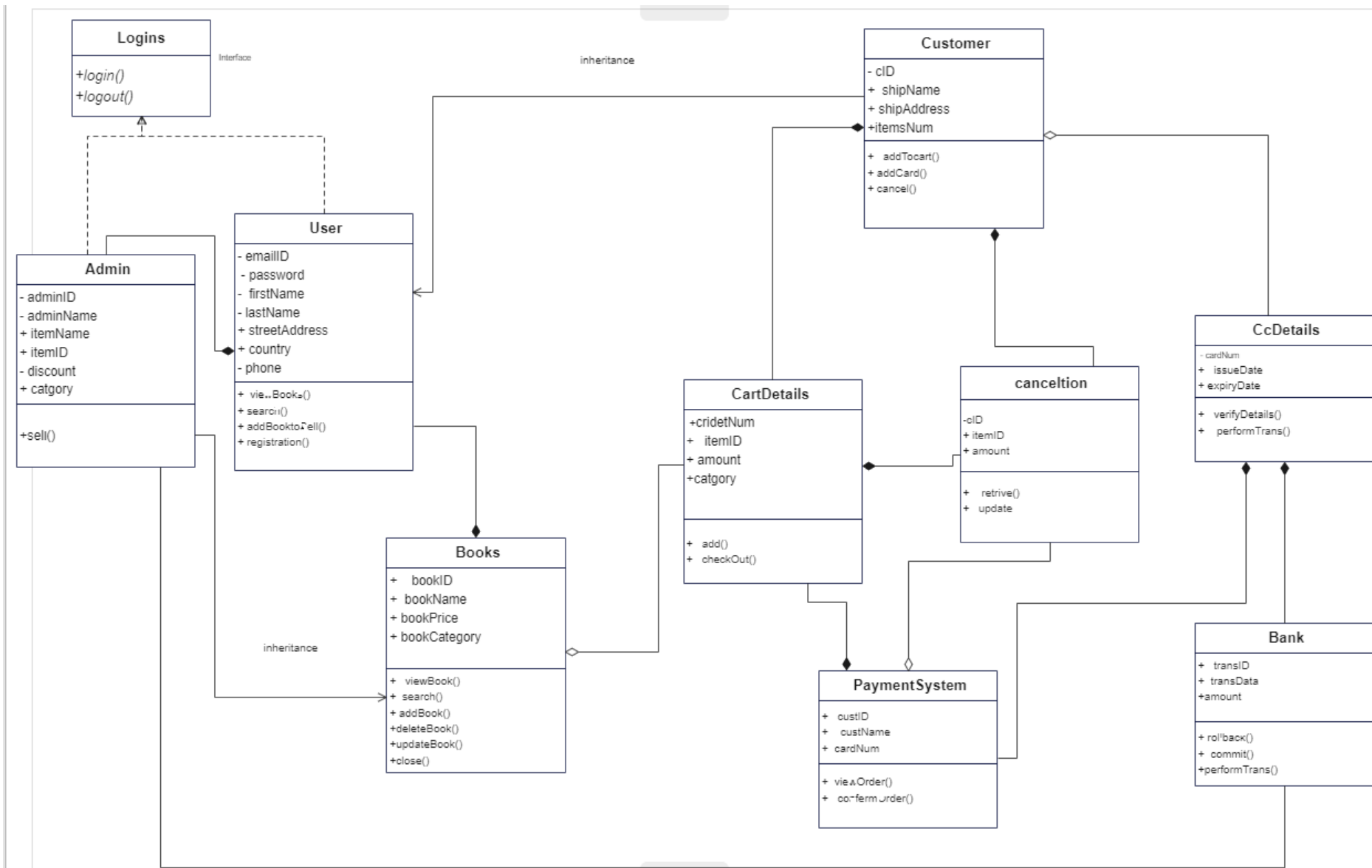
B) Class Diagram:

1)stage one:

2) stage two:

3) stage three:

Logins «Interface»
+*login()*
+*logout()*

Customer
- cID
+ shipName
+ shipAddress
+itemsNum
+ addTocart()
+ addCard()
+ cancel()

User
- emailID
- password
- firstName
- lastName
+ streetAddress
+ country
- phone
+ viewBooks()
+ search()
+ addBooktoSell()
+ registration()

Admin
- adminID
- adminName
+ itemName
+ itemID
- discount
+ catgory
+sell()

CcDetails
- cardNum
+ issueDate
+ expiryDate
+ verifyDetails()
+ performTrans()

CartDetails
+cridetNum
+ itemID
+ amount
+catgory
+ add()
+ checkOut()

canceltion
-cID
+ itemID
+ amount
+ retrive()
+ update

Books
+ bookID
+ bookName
+ bookPrice
+ bookCategory
+ viewBook()
+ search()
+ addBook()
+deleteBook()
+updateBook()
+close()

PaymentSystem
+ custID
+ custName
+ cardNum
+ viewOrder()
+ confermOrder()

Bank
+ transID
+ transData
+amount
+ rollback()
+ commit()
+performTrans()

inheritance

C) Database Schema:

Bank

pk,fk	card_number
pk,fk	cvv_no
	expire_date

order

pk	order_no	int
fk	customer_id	int
	customer_name	string
	customer_address	string
	shipment_date	date
fk	book_id	int
fk	card_number	Type

customer

pk	customer_id	int
	customer_name	string
	customer_address	string
	customer_phone	int
	customer_zipcode	int
pk	card_number	int
pk	cvv_no	int

shipment

fk	book_id
fk	order_no
	shipment_date
	chargeCardTime
	packing_time
	customer_adress

books

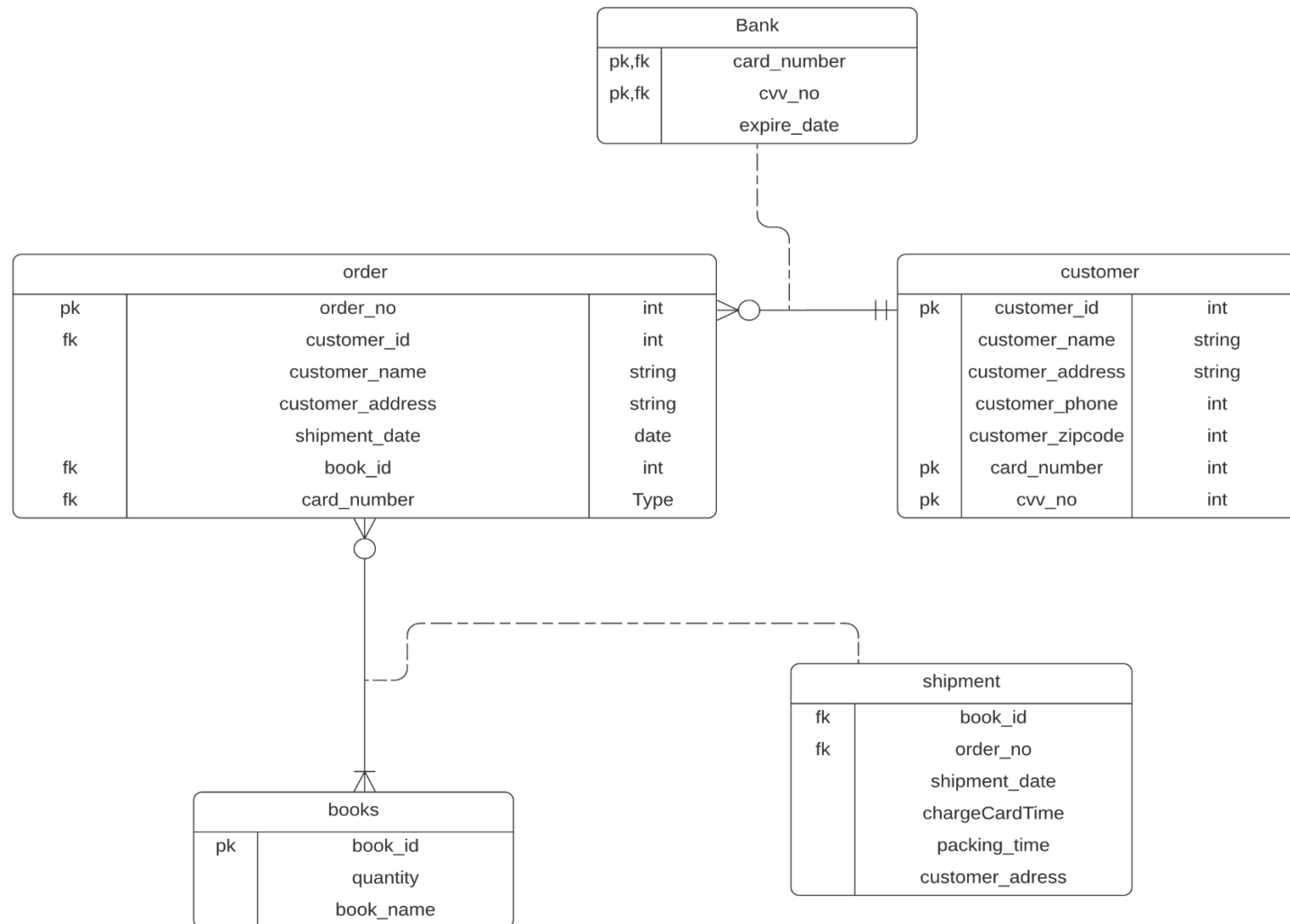|	
pk	book_id
	quantity
	book_name

D) Design Pattern:

Name: Singleton design pattern

Description: Using when we need to ensure that only one object of a particular class need to be created, this is useful when exactly one object is needed to coordinate actions across the system, to ensure that class has one object.

Context: It is very common to find classes for which only one instance should exist.

Problem:

1) ensure that it is never possible to create more than one instance of the admin class.

2) provide a global point of access to that instance.

Forces:

The use of a public constructor cannot guarantee that no more than one instance will be crated.

The singleton instance must also be accessible to all classes that require it, therefore it must often be public.

Solution:

1) Make the default constructor private, to prevent other objects from using the new operator with the Singleton class.

2)Create a static creation method that acts as a constructor, this method calls the private constructor to create an object and saves it in a static field. All following calls to this method return the cached object.

Delegation pattern:

Description: the delegation pattern is an object- oriented design

pattern that allows object composition to achieve the same code

reuse as inheritance. This used when you want to make method that

another class has a method which provides the required service

Problem: how use method already exists in another class without

inheritance?

Affects in system design:

it will minimize development cost by using update method in two classes so we will avoid duplication.

Immutable pattern:

Description: This pattern uses to saving unique information by making the instance cannot be change.

Problem:

How do you create a class whose instance are immutable? protected the id user and admin from any changing so the system will saving unique id to all of them without duplication and any problem that will occur as: two members had same id.