Assignment PRJ301 (Java Web Application Development)

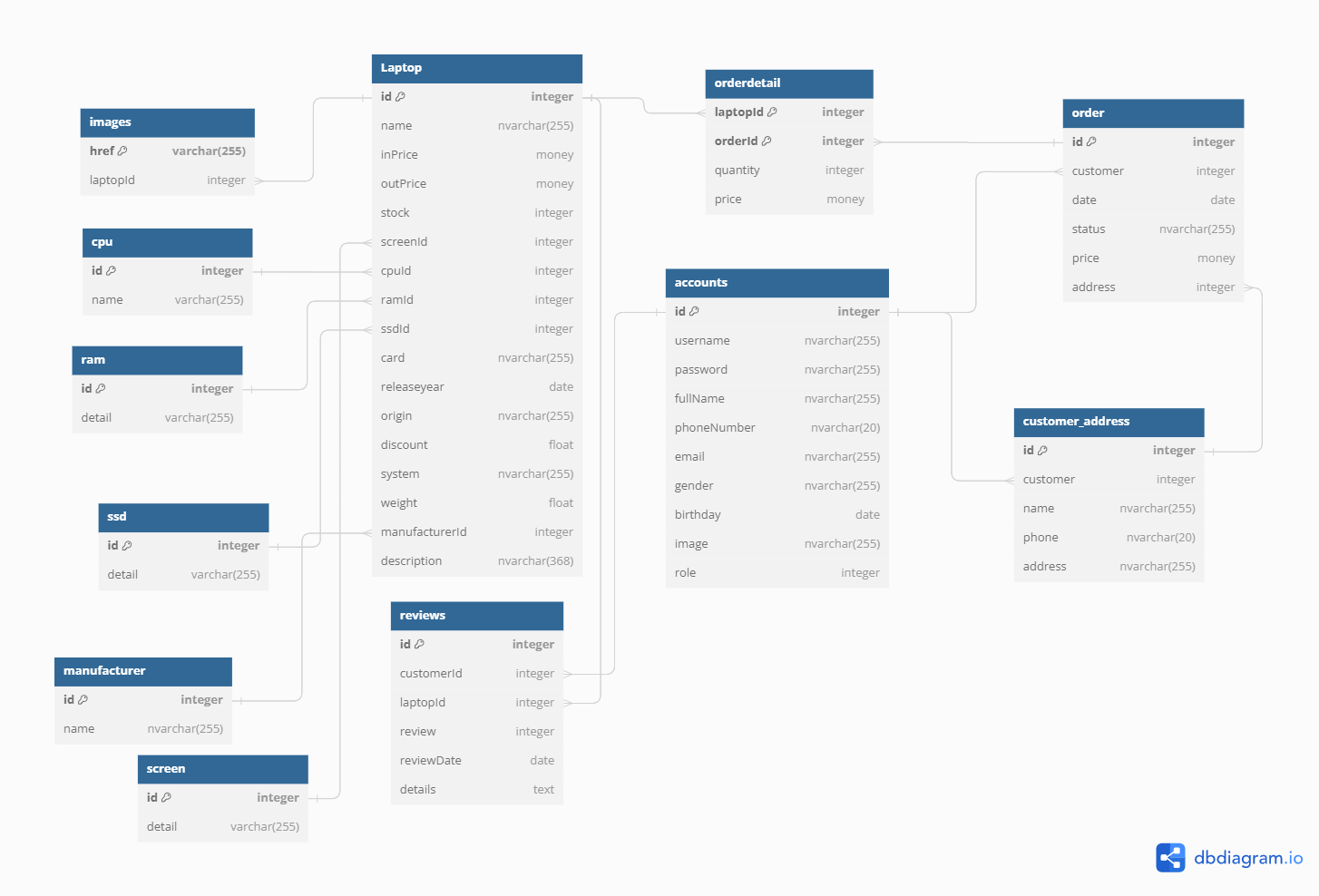
Laptop Shop - E-Commerce Website

1. **Case Study**

The Laptop Shop website is an E-Commerce website tailored specifically for purchasing laptops through internet. Our system incorporates advanced filtering options to allow customers to easily narrow down their search based on specifications such as processor type: ram, cpu, screen, manufacturer, screen, ssd and more. Additionally, customers could also view product detail information, add items to cart, checkout product, give review and comments for the product they bought and customize their profile. About sellers, with some features such as product management, order and accounts tracking, viewing profit.

**2.** **A database design**

# Entity Relationship Diagram (ERD):



**Logical Design:**

1. **Laptop table:**

Store information of products (laptop)

• id: Unique identifier for the laptop (Primary Key).

• name: Name of the laptop (nvarchar(255)).

• inPrice: Purchase price of the laptop (money).

• outPrice: Selling price of the laptop (money).

• stock: Quantity of the laptop available in stock (integer).

• screenId: Foreign key referencing the Screen table to specify screen details.

• cpuId: Foreign key referencing the CPU table to specify CPU details.

• ramId: Foreign key referencing the RAM table to specify RAM details.

• ssdId: Foreign key referencing the SSD table to specify SSD details.

• card: Graphics card information (nvarchar(255)).

• releaseyear: Release year of the laptop (date).

• origin: Origin country or region of the laptop (nvarchar(255)).

• discount: Discount percentage applied to the laptop (float).

• system: Operating system of the laptop (nvarchar(255)).

• weight: Weight of the laptop (float).

• manufacturerId: Foreign key referencing the Manufacturer table to specify the manufacturer details.

• description: Description of the laptop (nvarchar(368)).

1. **Screen table:**

Store types of screen

• id: Unique identifier for the screen (Primary Key).

• detail: Details about the screen (varchar(255)).

3. **Cpu table:**

Store types of cpu

• id: Unique identifier for the cpu(Primary Key).

• name: Details about the screen (varchar(255)).

**4. Manufacturer table:**

Store types of manufacturer

• id: Unique identifier for the manufacturer (Primary Key).

• name: Details about the screen (varchar(255)).

1. **Ssd table:**

Store types of Ssd

• id: Unique identifier for the ssd(Primary Key).

• detail: Details about the screen (varchar(255)).

**6**. **Ram table:**

Store types of ram

• id: Unique identifier for the ram(Primary Key).

• detail: Details about the screen (varchar(255)).

**7. Images table:**

Store image URLs or paths for each laptop in the "Laptop" table

* + href: URL or path to the image file (Primary Key).
  + laptopId: Foreign key referencing the "Laptop" table to specify which laptop the image belongs to.

**8. Order table:**

Store information about customer orders

* id: Unique identifier for the order (Primary Key).
* customer: Foreign key referencing the "Accounts" table to specify the customer who placed the order.
* date: Date when the order was placed.
* status: Status of the order (nvarchar(255)).
* price: Total price of the order (money).
* address: Foreign key referencing the "Customer\_Address" table to specify the delivery address for the order.

**9. Customer\_Address table:**

Store customer address information

* id: Unique identifier for the customer address (Primary Key).
* customer: Foreign key referencing the "Customer" table to specify the customer associated with the address.
* name: Name associated with the address (nvarchar(255)).
* phone: Phone number associated with the address (nvarchar(20)).
* address: Address details (nvarchar(255)).

**10. Accounts table:**

Store account information

* id: Unique identifier for the account (Primary Key).
* username: Username for the account (nvarchar(255)).
* password: Password for the account (nvarchar(255)).
* fullName: Full name of the account holder (nvarchar(255)).
* phoneNumber: Phone number of the account holder (nvarchar(20)).
* email: Email address of the account holder (nvarchar(255)).
* gender: Gender of the account holder (nvarchar(255)).
* birthday: Birthday of the account holder (date).
* image: URL or path to the image file for the account holder (nvarchar(255)).
* role: Role of the account holder, include customer and admin(integer).

**11. OrderDetail table:**

Store details of individual laptop items within an order.

* laptopId: Foreign key referencing the "Laptop" table to specify the laptop included in the order (Primary Key).
* orderId: Foreign key referencing the "Order" table to specify the order to which the detail belongs (Primary Key).
* quantity: Quantity of the specified laptop in the order.
* price: Price per unit of the specified laptop (money).

**12. Reviews table:**

Store customer reviews for laptops

* id: Unique identifier for the review (Primary Key).
* customer\_id: Foreign key referencing the "Accounts" table to specify the customer who submitted the review.
* laptop\_id: Foreign key referencing the "Laptop" table to specify the laptop being reviewed.
* review: Rating given by the customer for the laptop (integer).
* reviewDate: Date when the review was submitted.
* details: Textual details or comments provided by the customer in the review.

**Physical Design:**

**https://drive.google.com/file/d/1-8hgFyi0TR\_qHVyY\_aydLzMQh1bajFOv/view?usp=sharing**

**3.** **System Design**

The project used MVC Architecture ( Model-View-Controller ):

**Model layer:**

Its most important purpose is to connect the database, process data and prepare data for transfer to other components.

Business Object :

1. Laptop

* Represents laptop products
* Has attributes such as id, name, inPrice,... of laptop products
* Connects to table [Laptop] in database

1. Accounts

* Represents customers and admins in shop
* Has attributes such as id, fullname, password, username, email, phoneNumber, gender, birthday, image, role
* Connects to table [Accounts] in database

1. Manufacturer, Ram, Cpu, Ssd, Screen

* Used to filter different laptop based on Manufacturer, Ram, Cpu, Ssd, Screen
* Has two attributes id and name (or detail)
* Respectively connects to table [Manufacturer], [Ram], [Cpu], [Ssd], [Screen]

1. Order

* Represents order s’information of customer
* Has attributes such as id, date, customerId, status, totalMoney, address
* Connects to table [order] in database

1. OrderDetail

* Represents in more detail of order, it can accommodate more products
* Has attributes such as orderId, laptopId, quantity, price
* Connects to table [orderdetail] in database

1. Address

* Represents address of order from customer
* Has attributes such as id, Account, name, address
* Connects to table [customer\_address] in database

1. Review

* Represents customers feedbacks on the product they bought
* Has attributes such as id, customerId, laptopId, review, reviewDate, details
* Connects to table [Review] in database

1. Cart and Items

* Storage product when customer add to cart
* Items has attributes such as Laptop, quantity, price
* Cart is a list of items

DAO classes (Data access object):

* + Like a special manager for handling data behind the scenes.
  + Its job is to handle all the details of talking to the database, like saving, retrieving, updating, and deleting data.
  + Some example of it may includes : AccountDAO, LaptopDAO, CpuDAO, RamDAO, AddressDAO, OrderDAO, OrderDetailDAO, ScreenDAO, SsdDAO, ManufacturerDAO using methods to peform CRUD (Create, Read, Update, Delete) operation on date.

**Controller layer:**

Servlet :

* It receives requests from the user, interacts with the Model to process data, and then forwards the processed data to the View for presentation to the user.
* It includes class such as: AccountServlet, BuyServlet, CheckoutServlet, ReviewServlet, SignUpServlet, AdminProductServlet,... Each classes has their own function that manages the flow of information between the user, the data, and the interface.

Filter :

* A component that intercepts and processes requests and responses between the client and the server before they reach the servlet.
* It includes LoginFilter, LoginAdminFilter to prevent user from accessing to area that are unauthorized.

**View layer :**

JSP File :

* + - JSP files contain HTML markup mixed with Java code (scriptlets), JSP tags, and expressions.
    - Used to dynamically generate the user interface based on data received from the Model.
    - JSP files are typically placed within the Web Pages folder. For example, home.jsp, cart.jsp, shop.jsp, profile.jsp, checkout.jsp, admin\_order.jsp,... use to display to the user.