

PET PARADISE

Software Design Document

– Tp HCM, May 2021 –

Table of Contents

Table of Contents

II. Overview

Code Packages

Package descriptions & package class naming conventions

Database Schema

II. Code Designs

Log In

a. Class Diagram

- b. Class Specifications
- c. Sequence Diagram
- d. Queries involved

Orders Creation

- a. Class Diagram
- b. Class Specifications
- c. Sequence Diagram
- d. Queries involved

Write Blog

- a. Class Diagram
- b. Class Specifications
- c. Sequence Diagram
- d. Queries involved

Update Users

- a. Class Diagram
- b. Class Specifications
- c. Sequence Diagram
- d. Queries involved

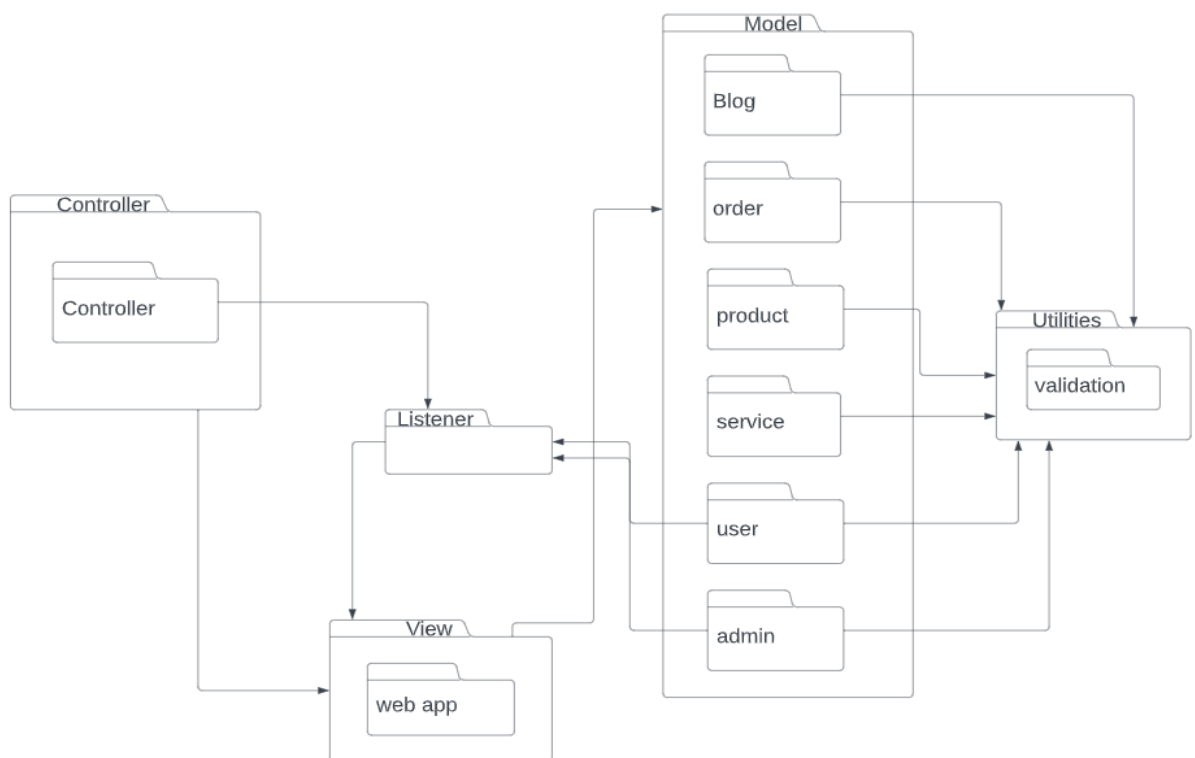
IV. Database Tables

Database Tables

1. Users
2. mentorDetails
3. mentorSkills
4. skills
5. followers
6. comments

II. Overview

Code Packages



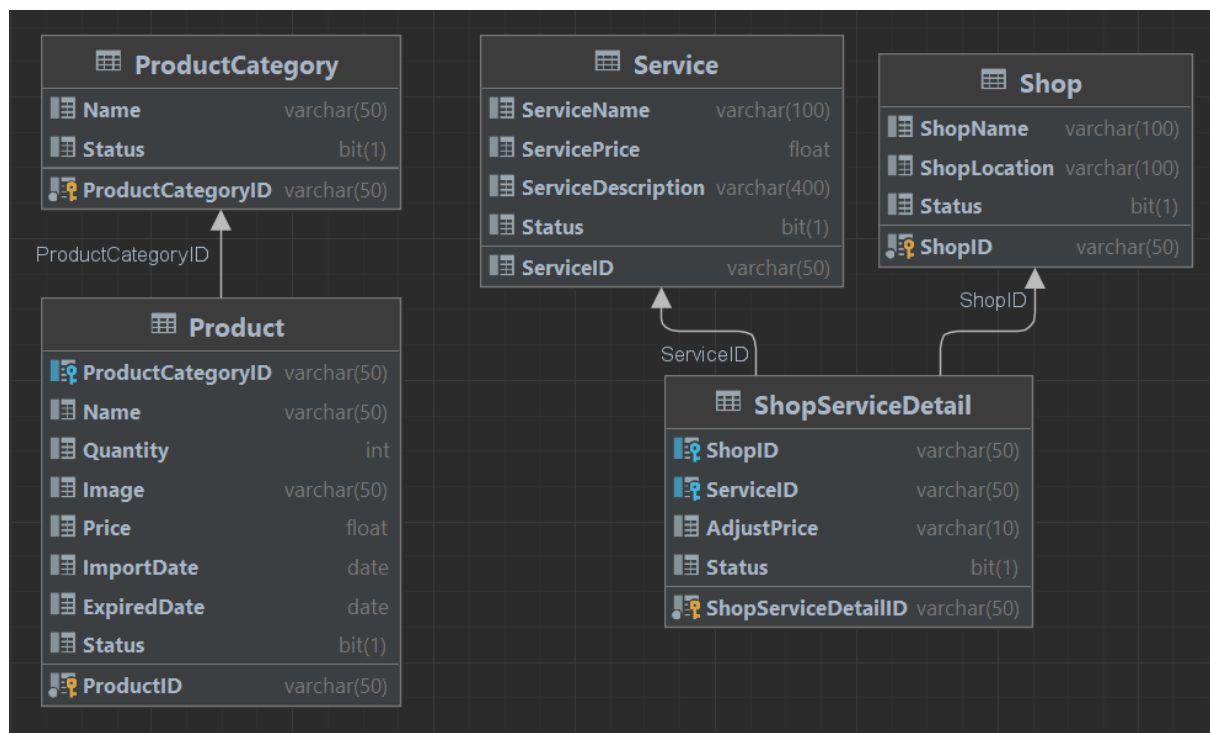
Package descriptions & package class naming conventions

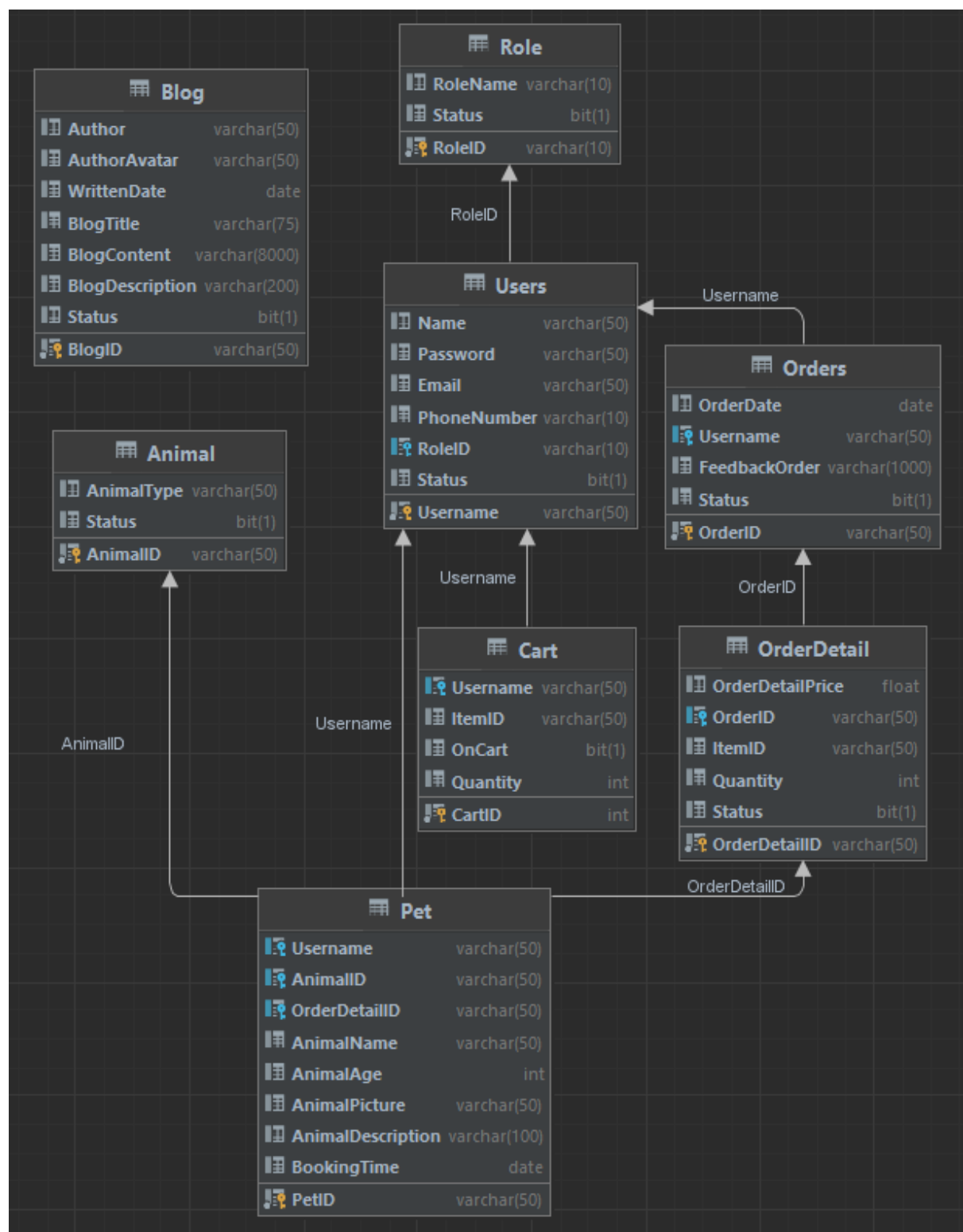
No	Package	Description
----	---------	-------------

01	controller	<p>Contains all controlling components of the web app (servlets).</p> <p>Naming convention: [Function]Controller</p> <p>For example: LoginController</p>
02	listener	<p>Contains all listener of the web app</p> <p>Naming convention: [Type]Listener</p> <p>For Example: RequestListener</p>
03	web app	<p>Contains all components related to displaying the results (html, jsp)</p> <p>Naming convention: [functionName].html or [functionName].jsp</p> <p>For Example: viewUserProfile.jsp</p>
04	requests	<p>Contains all components related to Mentee's requests data</p> <p>Naming convention: [Request][DAO/DTO]</p>
05	admin	<p>Contains all components related to Admin's data</p> <p>Naming convention: [Admin][DAO/DTO]</p>
06	users	<p>Contains all components related to users' data</p> <p>Naming convention: [Users][DAO/DTO]</p>

07	utilities	<p>Contains components to support the application in performing underlying interactions (connect database)</p> <p>Naming: [Interaction]utils</p> <p>For Example: DButils</p>
----	-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Database Schema





No	Table	Description
01	Users	<ul style="list-style-type: none"> - All information about a registered user is stored here, including Mentees' and Mentors' information. - Primary keys: UserID. - Foreign keys: none.
02	OrderDetails	<ul style="list-style-type: none"> - Some of the information about order. - Primary keys: OrderdetailID. - Foreign keys: OrderID.
03	Order	<ul style="list-style-type: none"> - When a customer completely checks out, the system creates an order . - Primary keys: OrderID. - Foreign keys: Username.
04	Product	<ul style="list-style-type: none"> - The items are being sold in stores. - Primary keys: ProductID. - Foreign keys: ProductCategoryID.
05	Product Category	<ul style="list-style-type: none"> - Description about product. - Primary keys: ProductCategoryID. - Foreign keys: none.

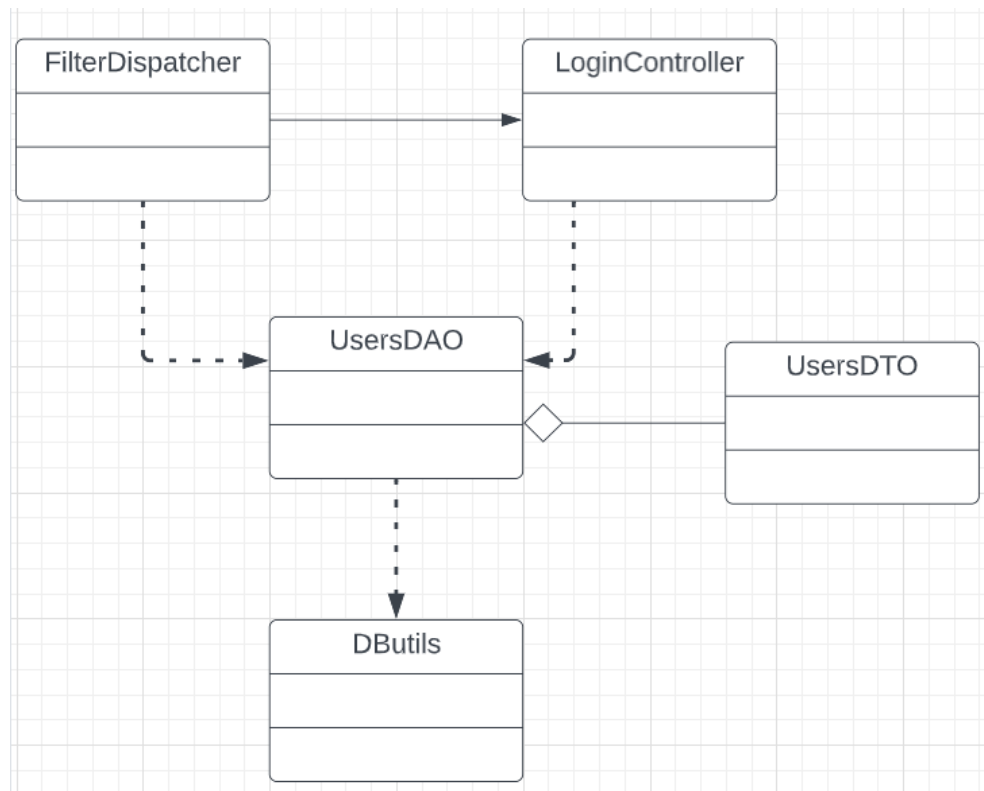
06	Role	<ul style="list-style-type: none"> - Roles include admin and user. - Primary keys: RoleID. - Foreign keys: none.
07	Shop	<ul style="list-style-type: none"> - Information about each store. - Primary keys: ShopID. - Foreign keys: none.
08	ShopServiceDetail	<ul style="list-style-type: none"> - Details about the branch of Pet's Paradise. - Primary keys: ShopServiceDetailID. - Foreign keys: ShopID, ServiceID.
09	Service	<ul style="list-style-type: none"> - Describe the service that the store can serve. - Primary keys: ServiceID. - Foreign keys: none.
10	Cart	<ul style="list-style-type: none"> - Cart contains products which are chosen by the customer. - Primary keys: CartID. - Foreign keys: Username.
11	Pet	<ul style="list-style-type: none"> - Information of customer's pet. - Primary keys: PetID. - Foreign keys: Username, AnimalID, OrderDetailID.

12	Animal	<ul style="list-style-type: none"> - Type of pet. - Primary keys: AnimalID. - Foreign keys: none.
12	Blog	<ul style="list-style-type: none"> - Review and advertise the store to customer. - Primary keys: BlogID. - Foreign keys: none.

II. Code Designs

Log In

a. Class Diagram



b. Class Specifications

FilterDispatcher Class

No	Method	Description
01	FilterDispatcher	- Default constructor for this class
02	doFilter	<p>Input: ServletRequest object, ServletResponse object , FilterChain object</p> <p>Output: The resource to which the user will be directed.</p> <p>- Perform filtering operations between classes and resources (i.e navigation, data compression)</p>

03	Init	<p>Input: filterConfig object</p> <p>Output: none</p> <p>- Perform actions when the filter is accessed for the first time</p>
04	toString	<p>- Return a string representation of the FilterDispatcher object</p>

LoginController Class

No	Method	Description
01	ProcessRequests	<p>Input: HttpServletRequest object, HttpServletResponse object</p> <p>- Process the Request coming from the client</p>

UserDAO Class

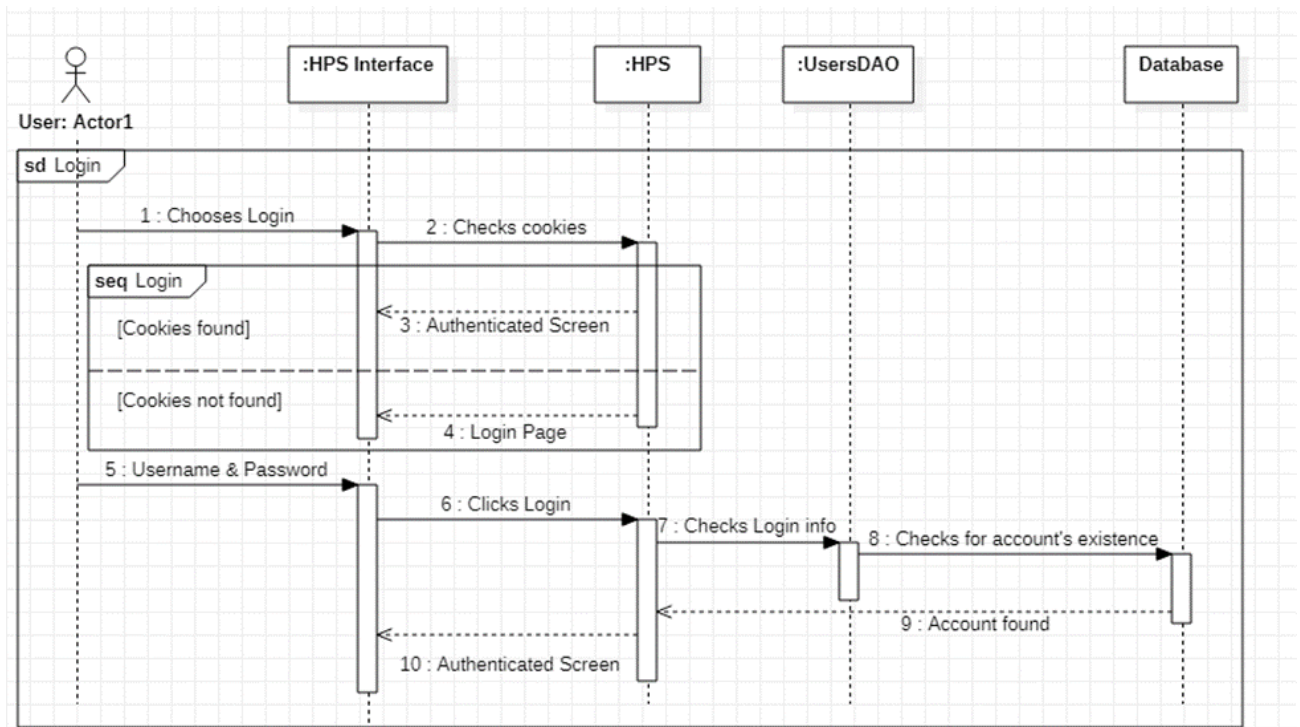
No	Method	Description
01	CheckLogin	<p>Input: String username, String password</p> <p>Output: UserDTO object</p> <p>- Check for the existence of the account in the Database and return the corresponding User object</p>

DButils Class

No	Method	Description
----	--------	-------------

01	makeConnection	Input: None - Establish the Database connection based on predetermined parameters.
----	----------------	-------------------------------------------------------------------------------------------

c. Sequence Diagram

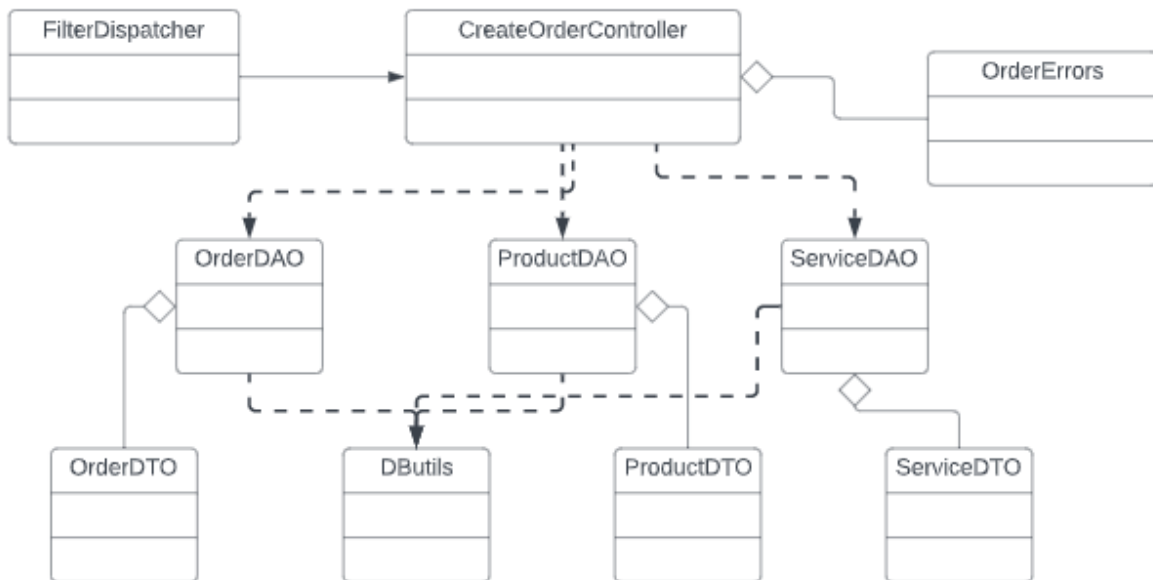


d. Queries involved

SELECT * FROM users WHERE username = @username AND password = @password

Orders Creation

a. Class Diagram



b. Class Specifications

FilterDispatcher Class

No	Method	Description
01	FilterDispatcher	- Default constructor for this class
02	doFilter	<p>Input: ServletRequest object, ServletResponse object , FilterChain object</p> <p>Output: The resource to which the user will be directed.</p> <p>- Perform filtering operations between classes and resources (i.e navigation, data compression)</p>

03	Init	<p>Input: filterConfig object</p> <p>Output: none</p> <p>- Perform actions when the filter is accessed for the first time</p>
04	toString	<p>- Return a string representation of the FilterDispatcher object</p>

CreateOrderController Class

No	Method	Description
01	processRequest	<p>Input: HttpServletRequest object, HttpServletResponse object</p> <p>- Process the Request coming from the client</p>

OrderDAO Class

No	Method	Description
01	createOrder	<p>Input: String OrderID , String OrderDetailID, String ProductID, String ServiceID,String UserID, String title, String status.</p> <p>Output: Object result</p> <p>- Create a new Request with the specified parameters</p>

OrderDAO Class

No	Method	Description
----	--------	-------------

01	Product / Service	<p>Input: String[] productName / String[] serviceName</p> <p>Output: String</p> <p>- Return a comma-separated string containing the IDs of all selected products/services during request creation.</p>
----	-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

OrderError Class

No	Method	Description
01	setContentLengthError	<p>Input: String ContentErrorMessage</p> <p>Output: None</p> <p>- Set the value for the ContentLengthError field</p>
02	setTitleLengthError	<p>Input: String TitleErrorMessage</p> <p>Output: None</p> <p>- Set the value for the TitleLengthError field</p>

DButils Class

No	Method	Description
01	makeConnection	<p>Input: None</p> <p>- Establish the Database connection based on pre-determined parameters.</p>

c. Sequence Diagram

d. Queries involved

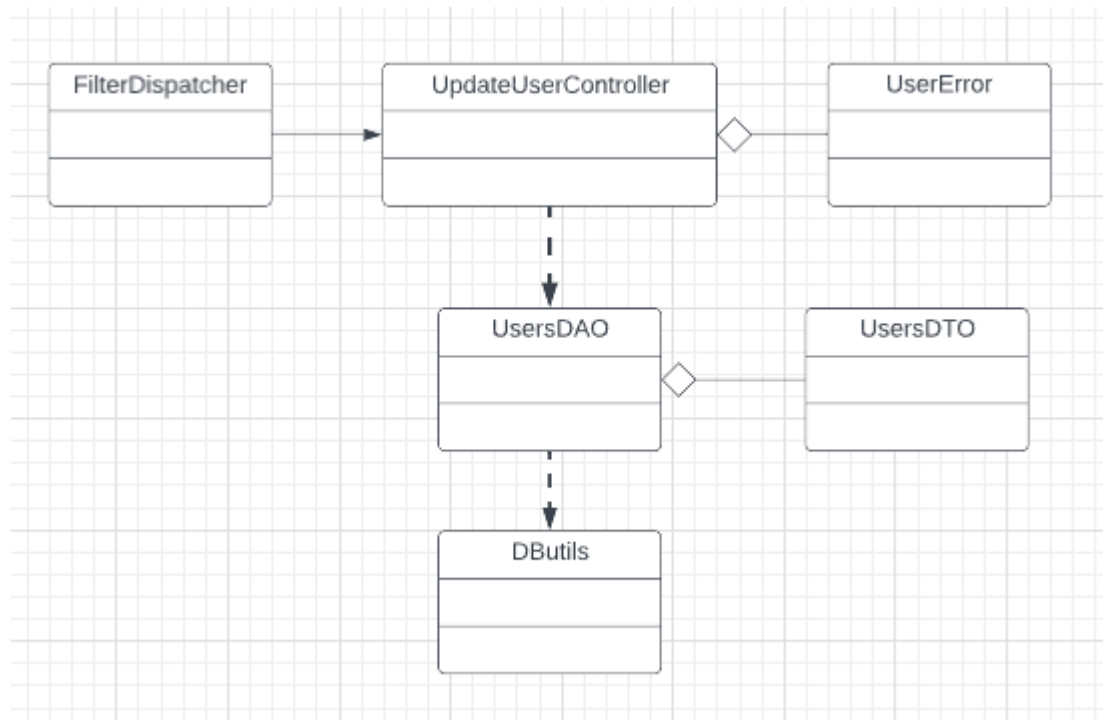
SELECT MAX(OrderID) as OrderID from requests

SELECT productID FROM product WHERE productName like
@productName

SELECT serviceID FROM service WHERE serviceName like @seviceName

Write blog

a. Class Diagram



b. Class Specifications

FilterDispatcher Class

No	Method	Description
01	FilterDispatcher	- Default constructor for this class

02	doFilter	<p>Input: ServletRequest object, ServletResponse object , FilterChain object</p> <p>Output: The resource to which the user will be directed.</p> <p>- Perform filtering operations between classes and resources (i.e navigation, data compression)</p>
03	Init	<p>Input: filterConfig object</p> <p>Output: none</p> <p>- Perform actions when the filter is accessed for the first time</p>
04	toString	<p>- Return a string representation of the FilterDispatcher object</p>

UpdateUserController Class

No	Method	Description
01	processRequest	<p>Input: HttpServletRequest object, HttpServletResponse object</p> <p>- Process the Request coming from the client</p>

UserDAO Class

No	Method	Description
----	--------	-------------

01	UpdateUser	<p>Input: String[] userID</p> <p>Output: String</p> <p>- Automatically update an incremental UserID based on the ID of the last request</p>
----	------------	---------------------------------------------------------------------------------------------------------------------------------------------

UserDAO Class

No	Method	Description
01	User	<p>Input: String[] userName</p> <p>Output: String</p> <p>- Return a comma-separated string containing the IDs of all selected user during request creation.</p>

UserError Class

No	Method	Description
01	setContentLengthError	<p>Input: String ContentErrorMessage</p> <p>Output: None</p> <p>- Set the value for the ContentLengthError field</p>
02	setTitleLengthError	<p>Input: String TitleErrorMessage</p> <p>Output: None</p> <p>- Set the value for the TitleLengthError field</p>

DButilsClass

No	Method	Description
01	makeConnection	Input: None - Establish the Database connection based on pre-determined parameters.

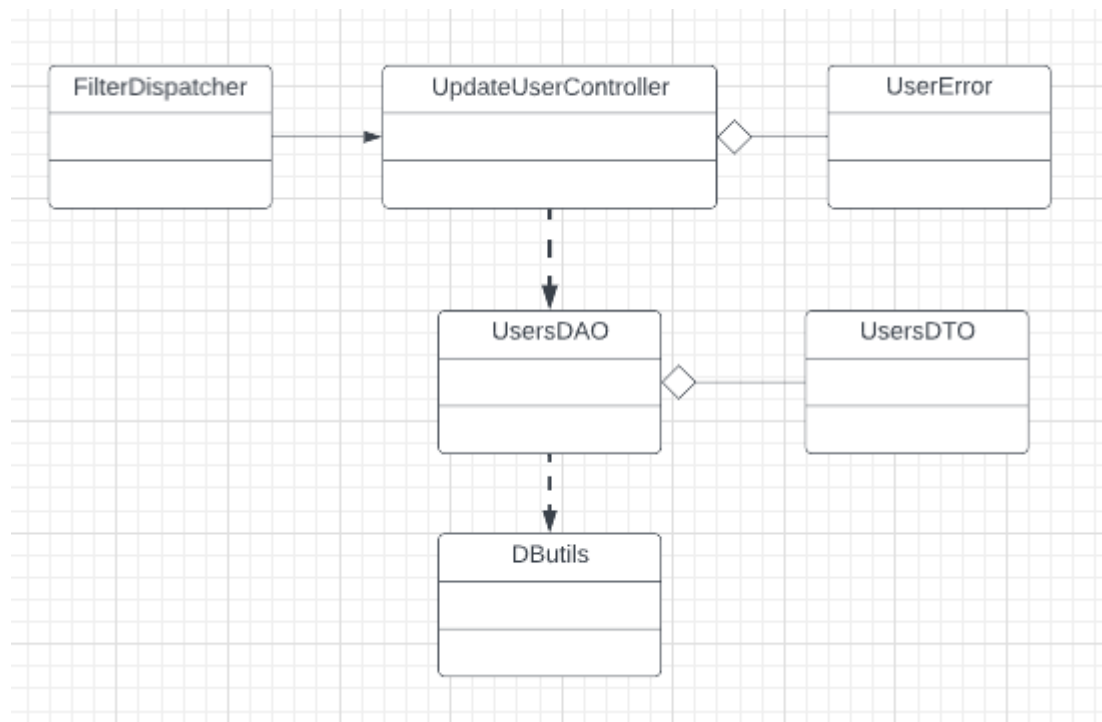
c. Sequence Diagram

d. Queries involved

UPDATE Users SET Name, Password, Email, PhoneNumber, WHERE UserName

Update users

a. Class Diagram



b. Class Specifications

FilterDispatcher Class

No	Method	Description
01	FilterDispatcher	- Default constructor for this class
02	doFilter	Input: ServletRequest object, ServletResponse object , FilterChain object Output: The resource to which the user will be directed. - Perform filtering operations between classes and resources (i.e navigation, data compression)
03	Init	Input: filterConfig object Output: none - Perform actions when the filter is accessed for the first time
04	toString	- Return a string representation of the FilterDispatcher object

UpdateUserController Class

No	Method	Description
01	processRequest	Input: HttpServletRequest object, HttpServletResponse object - Process the Request coming from the client

UserDAO Class

No	Method	Description
01	UpdateUser	<p>Input: String[] userID</p> <p>Output: String</p> <p>- Automatically update an incremental UserID based on the ID of the last request</p>

UserDAO Class

No	Method	Description
01	User	<p>Input: String[] userName</p> <p>Output: String</p> <p>- Return a comma-separated string containing the IDs of all selected user during request creation.</p>

UserError Class

No	Method	Description
01	setContentLengthError	<p>Input: String ContentErrorMessage</p> <p>Output: None</p> <p>- Set the value for the ContentLengthError field</p>
02	setTitleLengthError	<p>Input: String TitleErrorMessage</p> <p>Output: None</p> <p>- Set the value for the TitleLengthError field</p>

DButilsClass

No	Method	Description
01	makeConnection	Input: None - Establish the Database connection based on pre-determined parameters.

c. Sequence Diagram

d. Queries involved

UPDATE Users SET Name, Password, Email, PhoneNumber, WHERE
UserName

IV. Database Tables

Database Tables

1. Users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	UserID	varchar	8	Yes	Yes	PK	Start with AD (Admin) , US (User)
2	Username	varchar	50	Yes	Yes		
3	Email	varchar	50	Yes	Yes		
4	Password	varchar	50		Yes		

5	Status	bit					
6	PhoneNumber	varchar	10				
7	RoleID	varchar	10				

2. Shop

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ShopID	varchar	50	Yes	Yes		
2	ShopName	varchar	100				
3	ShopLocation	varchar	50				
4	Status	bit					0 is inactive, 1 (default) is active.

3. ShopServiceDetail

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ShopServiceDetailID	varchar	50	Yes	Yes	PK	
2	ShopID	varchar	50		Yes	FK	
3	ServiceID	varchar	50		Yes	FK	
4	AdjustPrice	varchar					
5	Status	bit					0 is inactive, 1 (default) is active.

4. Service

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ServiceID	varchar	8	Yes	Yes	PK	
2	ServiceName	varchar	50				
3	ServicePrice	varchar	50				

4	ServiceDescription	varchar	50				
5	Status	bit			Yes		0 is inactive, 1 (default) is active.

5. followers

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	followID	varchar	10	Yes	Yes	PK	
2	menteeID	varchar	8		Yes	FK	FK to users table.
3	mentorID	varchar	8		Yes	FK	FK to users table.
4	timestamp	datetime			Yes		

6. Role

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
---	------------	------	------	--------	----------	-------	-------

1	RoleID	varchar	10	Yes	Yes	PK	
2	RoleName	varchar	10				
3	Status	bit					0 is inactive, 1 (default) is active.

7. ProductCategory

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ProductCategoryID	varchar	50	Yes	Yes	PK	
2	Name	varchar	50				
3	Status	bit					

8. Product

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
---	------------	------	------	--------	----------	-------	-------

1	ProductID	varchar	50	Yes	Yes	PK	
2	ProductCategoryID	varchar	50				
3	Name	varchar	50				
4	Quantity	int					
5	Image	varchar	50				
6	Price	float					
7	ImportDate	date					
8	ExpiredDate	date					
9	Status	bit					0 is inactive, 1 (default) is active.

9. Pet

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	PetID	varchar	50	Yes	Yes	PK	
2	Username	varchar	50			FK	

3	AnimalID	varchar	50			FK	
4	OrderDetailID	varchar	50			FK	
5	AnimalName	varchar	50				
6	AnimalAge	int					
7	AnimalPicture	varchar	50				
8	AnimalDescription	varchar	50				
9	BookingTime	date					

10. Orders

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	OrdersID	varchar	50	Yes	Yes	PK	
2	OrdersDate	date					
3	Username	varchar	50			FK	
4	FeedbackOrder	varchar	1000				
5	Status	bit					

6	ShopID	varchar	50				
---	--------	---------	----	--	--	--	--

11. OrdersDetail

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	OrderDetailID	varchar	50	Yes	Yes	PK	
2	OrderDetailPrice	float					
3	OrderID	varchar	50			FK	
4	ItemID	varchar	60				
5	Quantity	int					
6	Status	bit					0 is inactive, 1 (default) is active.

12. Cart

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
---	------------	------	------	--------	----------	-------	-------

1	CartID	Int (auto increment)		Yes	Yes	PK	
2	UserName	float				FK	
3	ItemID	varchar	50				
4	OnCart	bit					
5	Quantity	int					

13. Blog

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	BlogID	Int (auto increment)	50	Yes	Yes	PK	
2	Author	varchar	50				
3	AuthorAvatar	varchar	50				
4	WrittenDate	date					
5	BlogTitle	varchar	75				

5	BlogContent	varchar	8000				
5	BlogDescription	varchar	200				
5	Status	bit					0 is inactive, 1 (default) is active.