ONLYFAN SHOP

Software Design Document

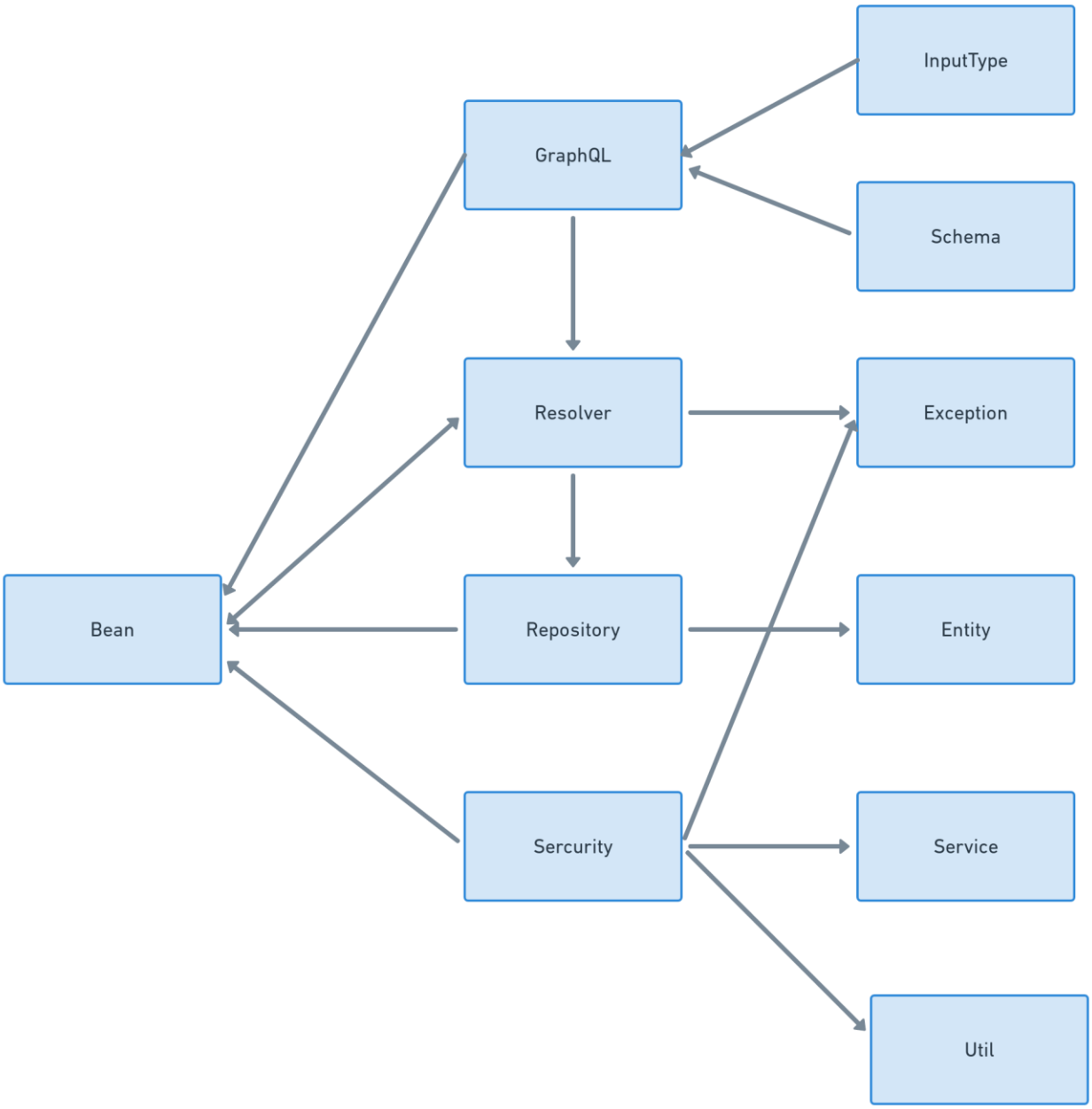
– Danang, November 2021 –

Table of Contents

Table of Contents

I. Overview

1. Code Packages/Namespaces



Package descriptions & package class naming conventions

No	Package	Description
01	*Entity*	*Declare table and relationship in Database* *Class Name in this packege have same name with table in database*
02	*Repository*	*Declare and implements methods interact with database*
03	*Sercurity*	*Authentication and authorization*
04	*Resovler*	*Implements query and mutation of graphql declaration in GraphQL package*
05	*Service*	*Business logic layer of server*
06	*Graphql*	*Declare type, query, mutation for graphql Server*
07	*Util*	*Some method using in orther class to help some function like parse, mapping, ….*
08	*Exception*	*Handle Runtime exception of server*

2. Database Schema

[Provide the tables relationship like example below – following Postgre database naming convention]

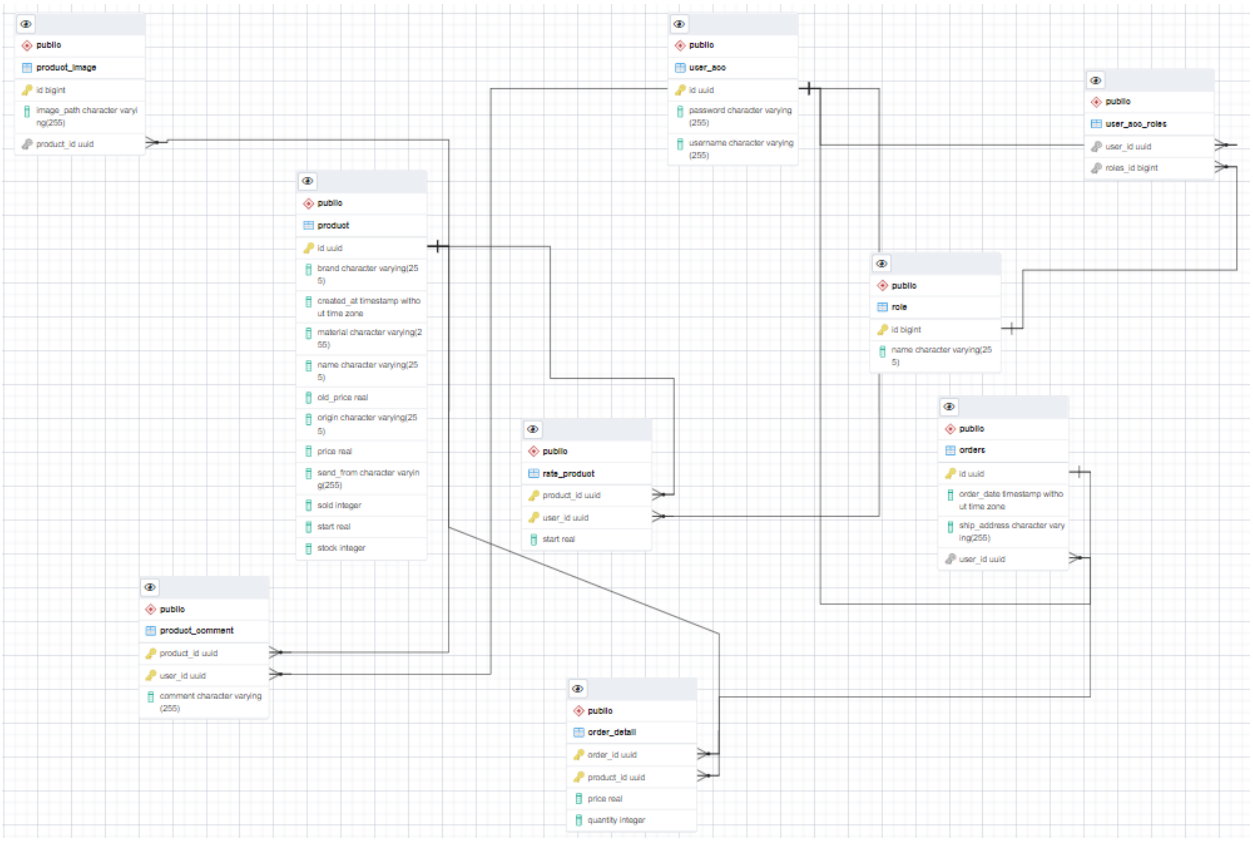


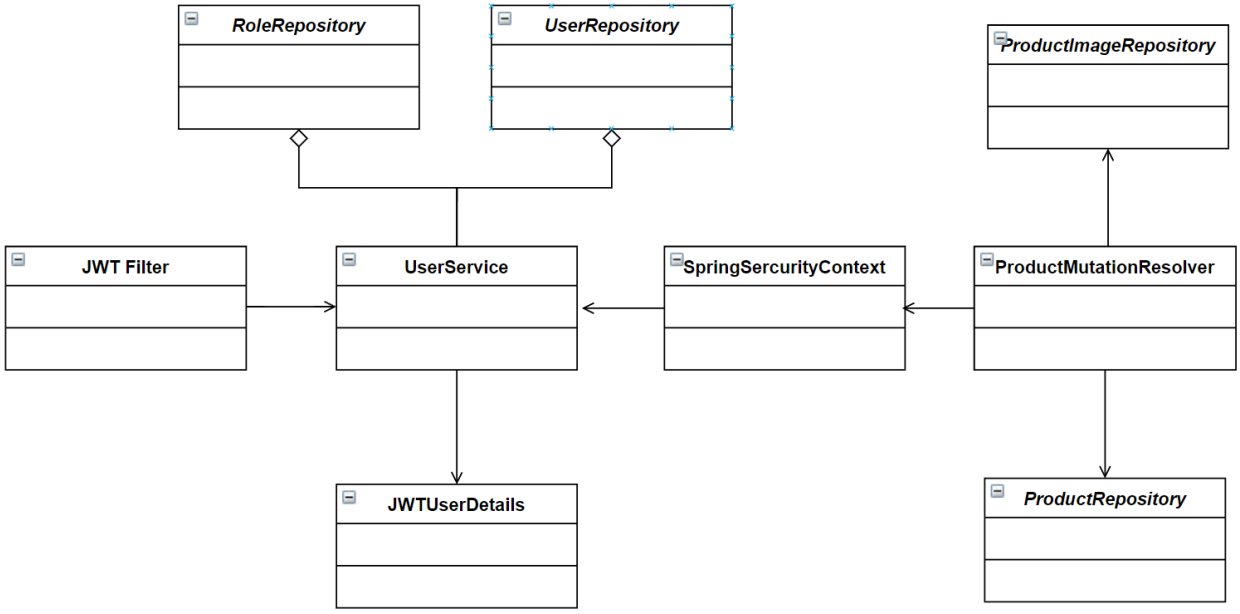
Table descriptions & package class naming conventions are as below

No	Table	Description
01	*product*	*Information about product* *- Primary keys: Id* *- foreign keys: category_Id*
02	*Category*	*Category information* *- PK: Id*
03	*Order*	*Infomation about order of customer, what customer bought, when they boughth.* *PK: Id* *FK: user_Id*
04	*Order details*	*Specification product in products of order.* *FK: Product Id and oredr_Id*
05	*Product Images*	*Image of product* *FK: product_id*
06	*User*	*Information about user in system* *PK: User_Id*
07	*Role*	*Roles having in system* *PK: Role_Id*
08	*User Role*	*Store user_id with their roles id* *FK: User_Id, Role_Id*
09	*Product Comment*	*Comments from user of system for product* *Id: Id* *FK: Product_Id, User_Id*
10	*Rate Product*	*Rate of user for product* *FK: User_Id, product_Id*

II. Code Designs

1. Create product

a. Class Diagram



b. Class Specifications

ProductRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*save*	*Input:* *+ product: Product* *Output: Product* *Save a product object to database and return product has just created*

ProductImageRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*save*	*Input:* *+ imagelink: String* *Output: ProductImage* *Save product images object to database and return product image has just created*

RoleRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findRoleByName*	*Input: + roleName: String*

		Output: Role *Find a role in database by name and return Role*

UserRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findByUsername*	*Input:* *+ username : String* *Output: User* *Find a role in database by name and return User*

JWTFilter

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*dofilterInternal*	*Input: HttpRequest, HttpResponse, FilterChain* *If request header contain token then create a SpringSercurity authentication context for the userDetails which was parsed By UserService class*
02	*getToken*	*Take token from the request by split "Authorization" header of request*

UserService

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*getUserByToken*	*Input:* *+ token: String* *Output: JWTUserDetail* *Get token from parameter then get username by decode token if token invalid throw error else find User in database and map it to JWTUserDetail Object*
02	*getUserDetails*	*Input: User object, token* *Output: JWTUserDetail* *Create a UserDetails object from user object and token*

JWTUserDetail

Class Methods

[Provide the detailed description for the class methods]

No		Description
01		*Extend UserDetail Class of SpringSercurity framework*

ProductMutationResolver

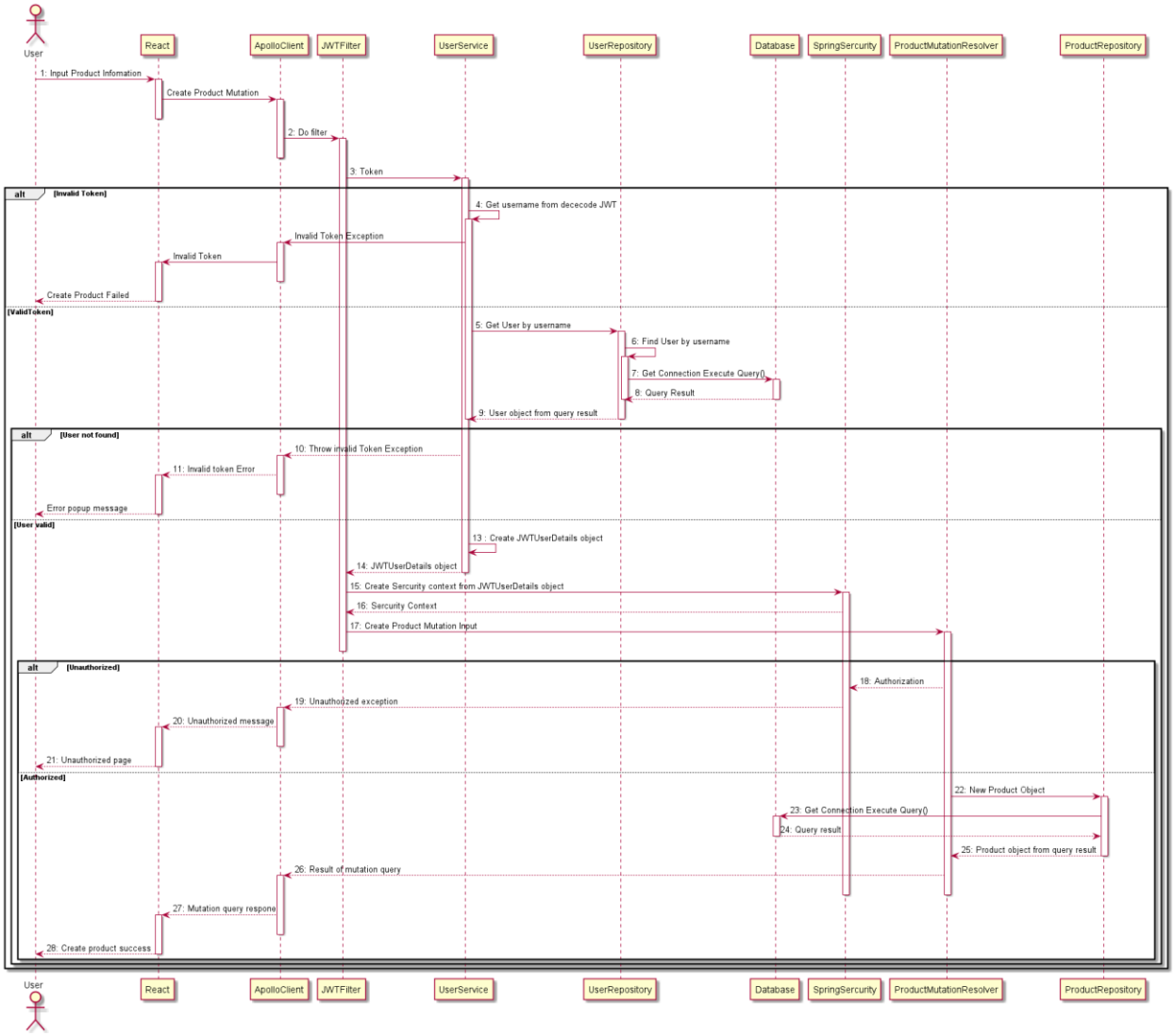
Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*createProduct*	*Input:* *+ createProductInput: CreateProductInput* *Output: Product* *Implement Product mutation query requested by client*

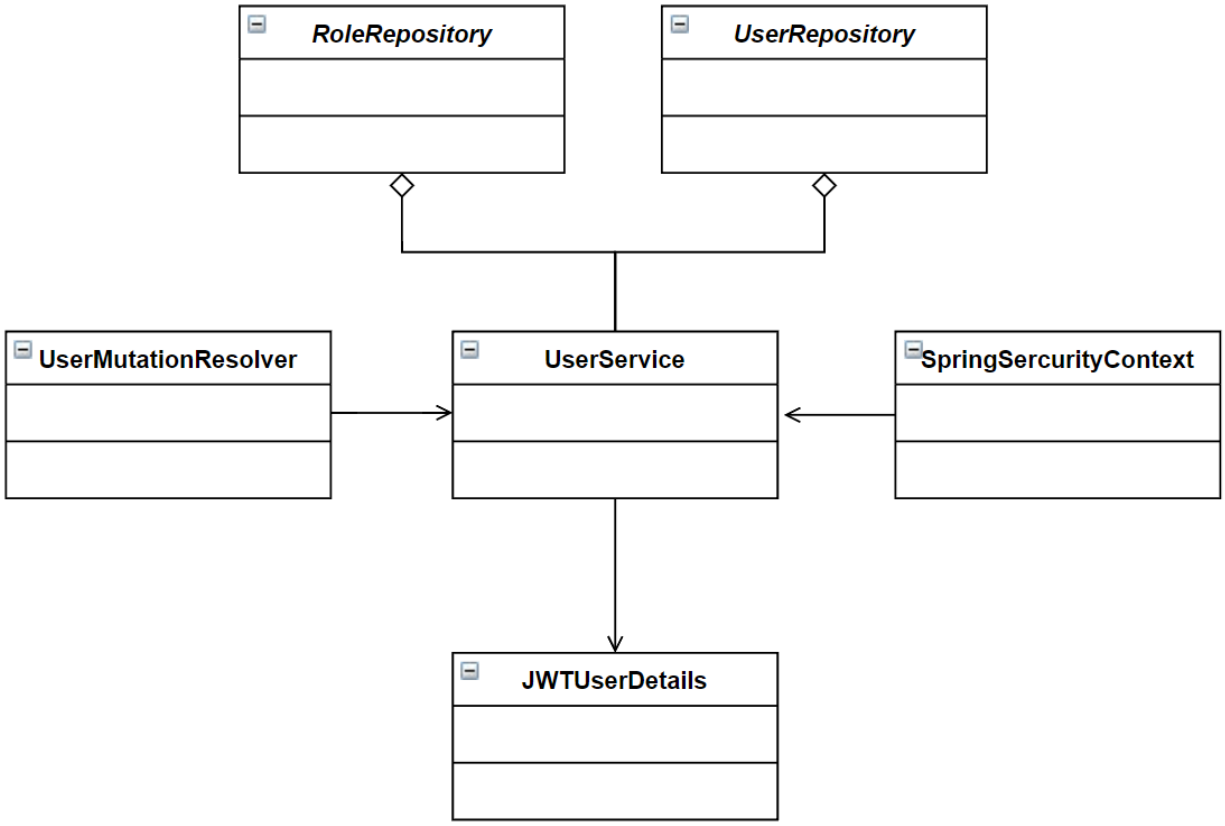
c. Sequence Diagram(s)

[Provide the sequence diagram(s) for the feature, see the sample below]



2. Login

a. Class Diagram



b. Class Specifications

RoleRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findRoleByName*	*Input: + roleName: String* *Output: Role* *Find a role in database by name and return Role*

UserRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findByUsername*	*Input:* *+ username : String* *Output: User* *Find a role in database by name and return User*

UserMutationResolver

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*login*	*Input:* *+ username: String* *+ password: String* *Check username and password is valid, if valid build SpringSecurity context, call UserService handle business login about JWT token then return current login user with it token*

UserService

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*getCurrentUser*	*Input:* *Output: User* *Map current login user in SpringSecurity context from UserDetail type to User type*

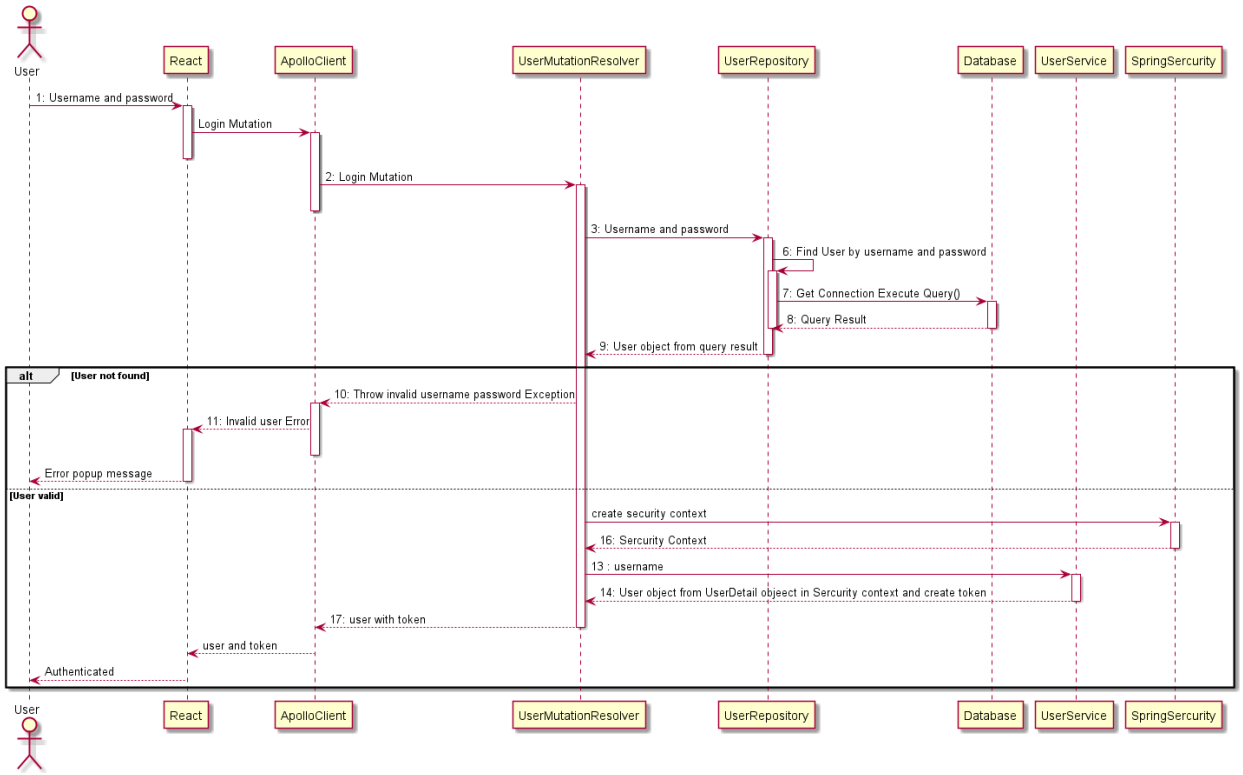
JWTUserDetail

Class Methods

[Provide the detailed description for the class methods]

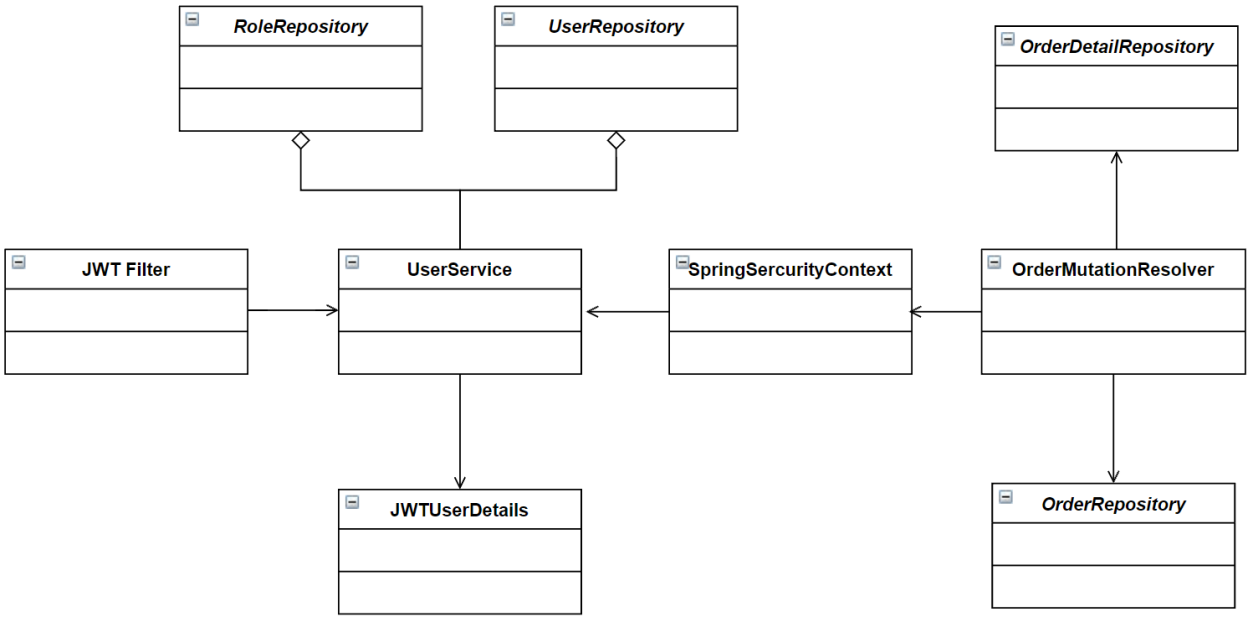
No		Description
01		*Extend UserDetail Class of SpringSercurity framework*

c. Sequence Diagram(s)



3. Order

a. Class Diagram



b. Class Specifications

OrderRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*save*	*Input:* *+ order: Order* *Output: Order* *Save a order object to database and return product has just created*

OrderDetailRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*save*	*Input:* *+ orderDetail: OrderDetail* *Output: OrderDetail* *Save product images object to database and return orderDetail has just created*

RoleRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description

01	*findRoleByName*	*Input: + roleName: String* *Output: Role* *Find a role in database by name and return Role*

UserRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findByUsername*	*Input:* *+ username : String* *Output: User* *Find a role in database by name and return User*

JWTFilter

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*dofilterInternal*	*Input: HttpRequest, HttpResponse, FilterChain* *If request header contain token then create a SpringSercurity authentication context for the userDetails which was parsed By UserService class*
02	*getToken*	*Take token from the request by split "Authorization" header of request*

UserService

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*getUserByToken*	*Input:* *+ token: String* *Output: JWTUserDetail* *Get token from parameter then get username by decode token if token invalid throw error else find User in database and map it to JWTUserDetail Object*
02	*getUserDetails*	*Input: User object, token* *Output: JWTUserDetail* *Create a UserDetails object from user object and token*

JWTUserDetail

Class Methods

[Provide the detailed description for the class methods]

No		Description
01		*Extend UserDetail Class of SpringSercurity framework*

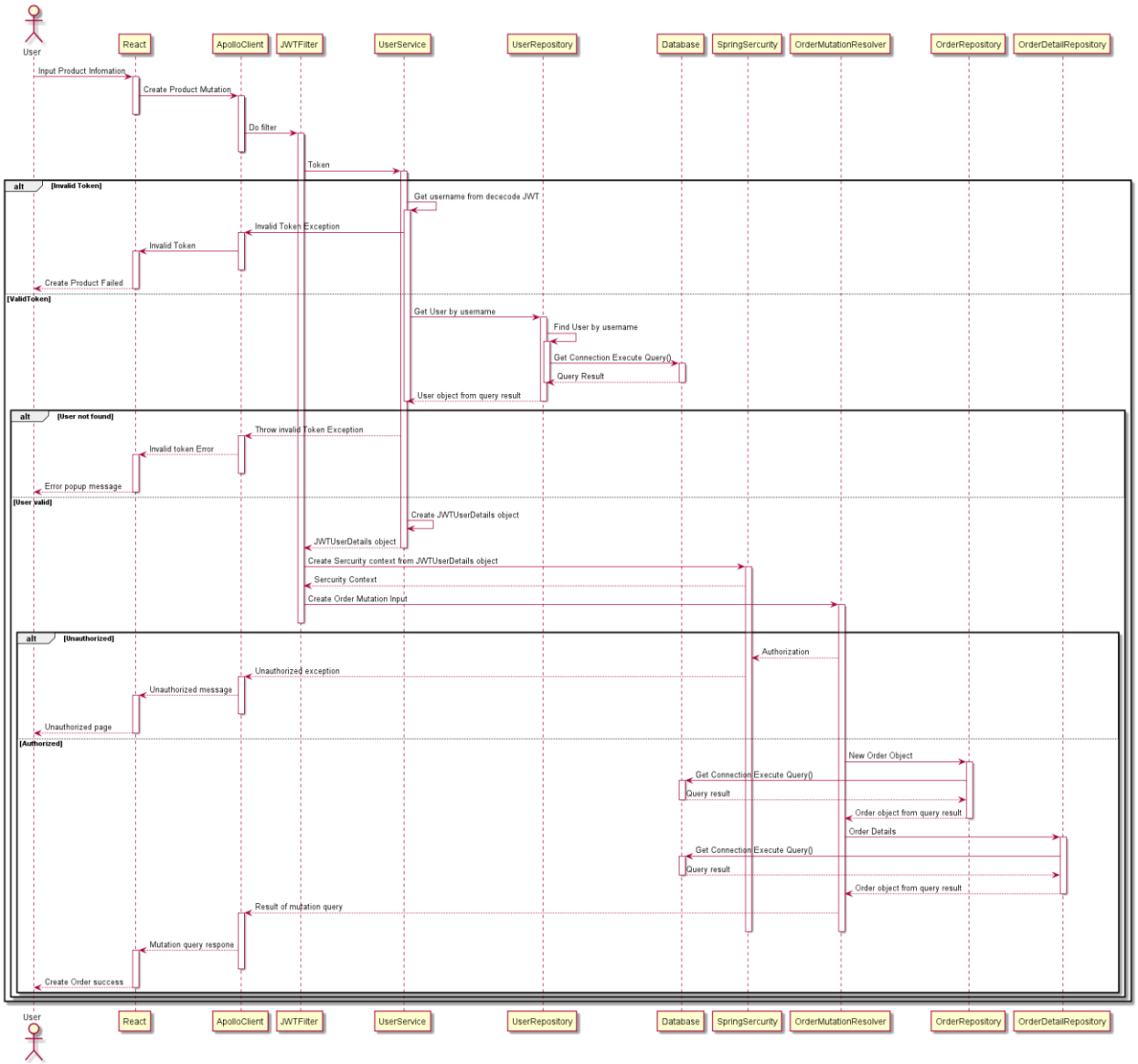
ProductMutationResolver

Class Methods

[Provide the detailed description for the class methods]

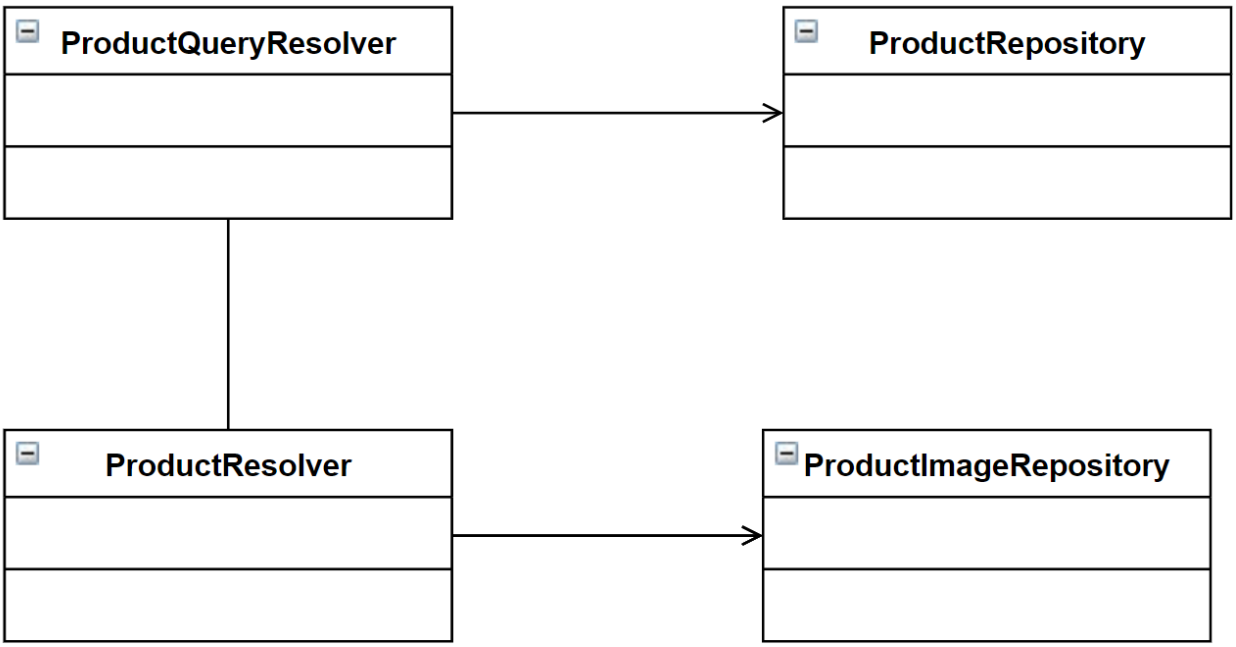
No	Method	Description
01	*createProduct*	*Input:* *+ createProductInput: CreateProductInput* *Output: Product* *Implement Product mutation query requested by client*

c. Sequence Diagram(s)



4. Pagination Product

a. Class Diagram



b. Class Specifications

ProductRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findAll*	*Input:* *+ pageable: Pageable* *Output: Paginated Products* *Find All pagination products*

ProductImageRepository

Class Methods

[Provide the detailed description for the class methods]

No	Method	Description
01	*findAllByProductId*	*Input:* *+ ProductId: String* *Output: List of ProductImage* *Find All Image of a Product*

ProductQueryResolver

Class Methods

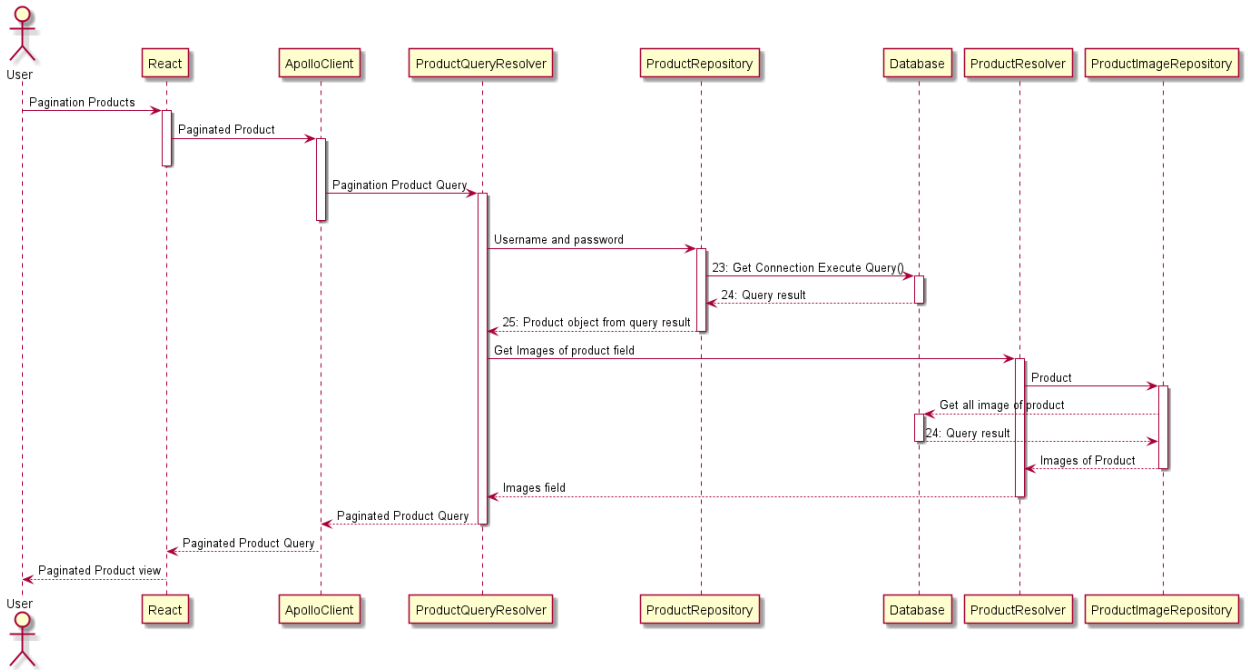
No	Method	Description
01	*pagingProduct*	*Input:* *+ limit: int* *+ page: int* *+ filed: string* *+ order: string* *Output: List of Product* *Implement paging product query from client*

ProductResolver

Class Methods

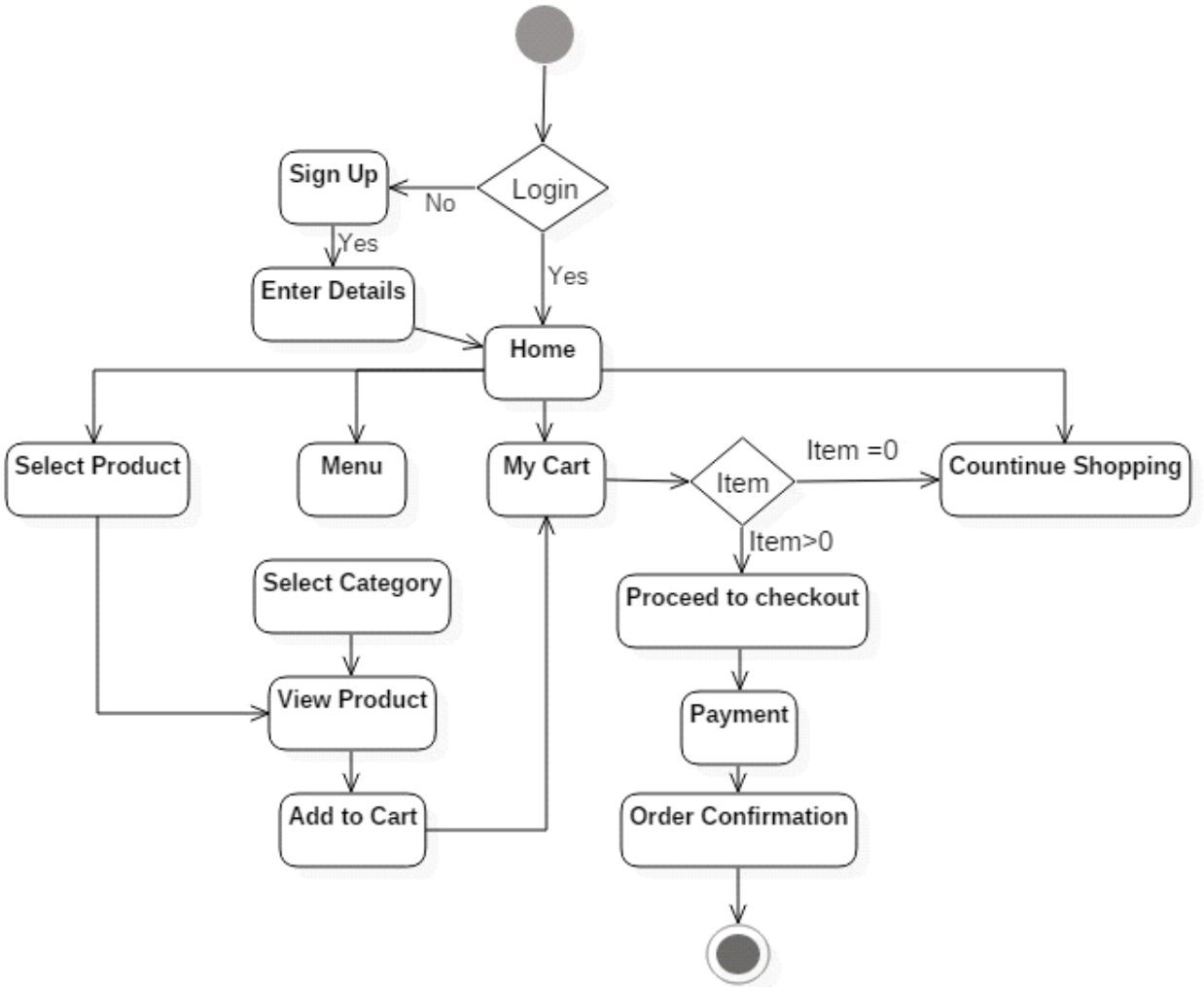
No	Method	Description
01	*images*	*Input:* *+product: Product* *Output: List of ProductImage* *Set result for images filed in paging Prouct query*

c. Sequence Diagram(s)

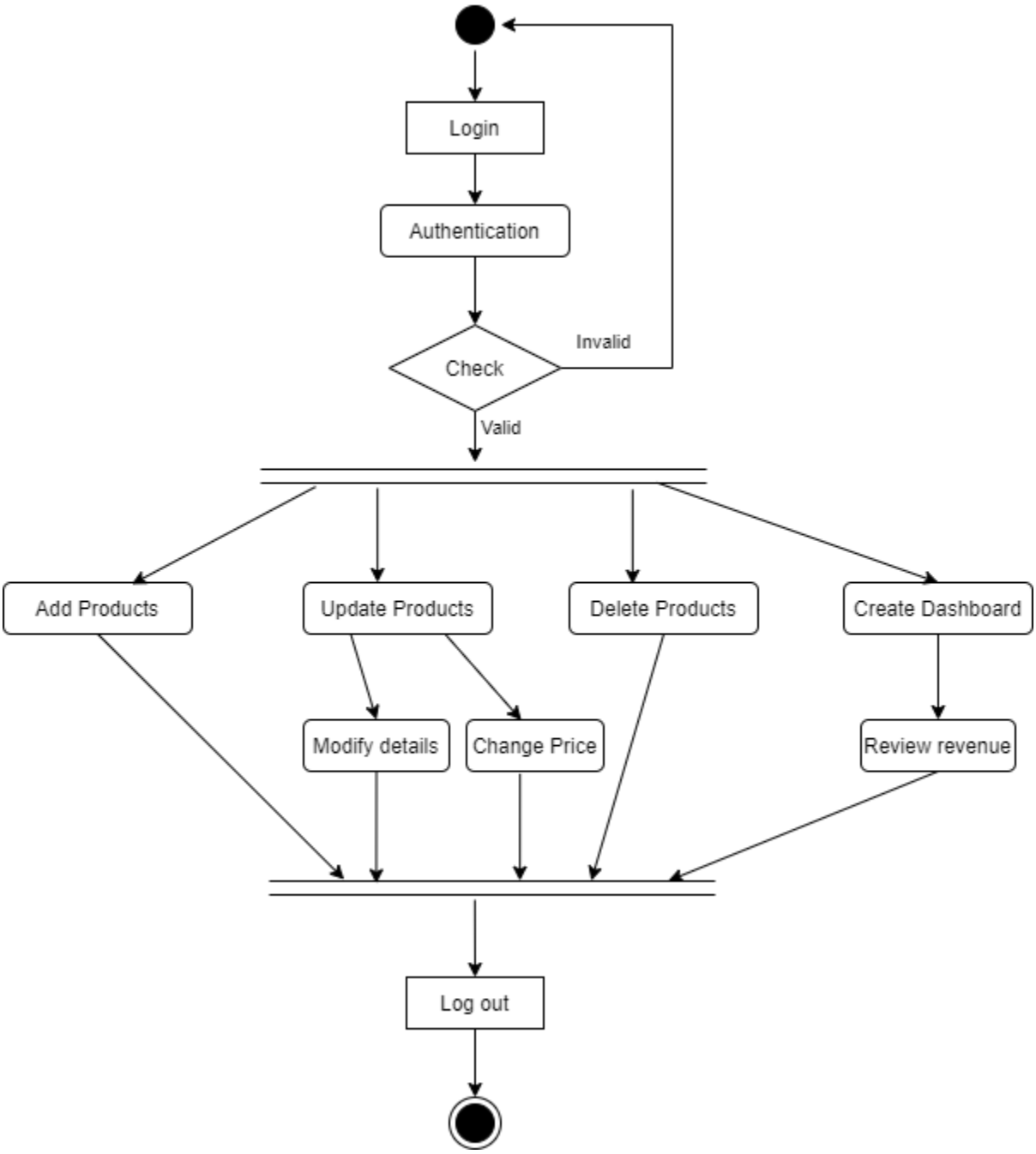


III. Activity Diagrams

1. Customer



2. Administrator



IV. Database Tables

1. Product

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ID	UUID	X	X	X	
2	brand	Text				
3	CreatedAt	Timestamp				
4	Material	Text				

5	Name	Text				
6	OldPrice	Real				
7	Origin	Text				
8	Price	Real				
9	SentFrom	Text				
10	Sold	Int				
11	Star	Real				
12	Stock	Int				

2. ProductImage

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ID	BigInt	X	X	X	
2	ImagePath	Text				
3	ProductId	UUID		X	X	

3. ProductComment

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ProductID	UUID		X	X	Foreign key from product table
2	UseID	UUID		X	X	Foreign key from user table
3	Commnet	TEXT				

4. RateProduct

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ProductID	UUID		X	X	Foreign key from product table
2	UseID	UUID		X	X	Foreign key from user table
3	Star	Real				

5. User

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ID	UUID	X	X	X	
2	Username	Text	X	X		
3	Password	Text				

6. Role

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ID	BigInt	X	X	X	
2	name	Text	X	X		

7. UserRole

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	UserId	UUID		X	X	Foreign Key from User table
2	RoleId	BigInt		X	X	Foreign Key from Role table

8. Order

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ID	UUID	X	X	X	
2	OrderDate	TimeStamp		X		
3	UserID	UUID			X	Foreign key from User table
4	ShipAddress	Text				

9. OrderDetail

#	Field name	Type	Unique	Not Null	PK/FK	Notes
1	ProductId	UUID		X	X	Foreign key from Product table
2	OrderId	UUID		X	X	Foreign key from User table
3	Price	Real				
4	Quantity	Int				