

## Application Layer (Use Cases) 2

اینجا ساختارهای سیستم نوشته می‌شوند.

هر عملیات مهم = یک Use Case

مثال:



## (API چیز) Presentation Layer 1

اینجا فقط:

- request
- use case
- response



## Domain Layer 3

اینجا قوانین بیزنسی هستند.

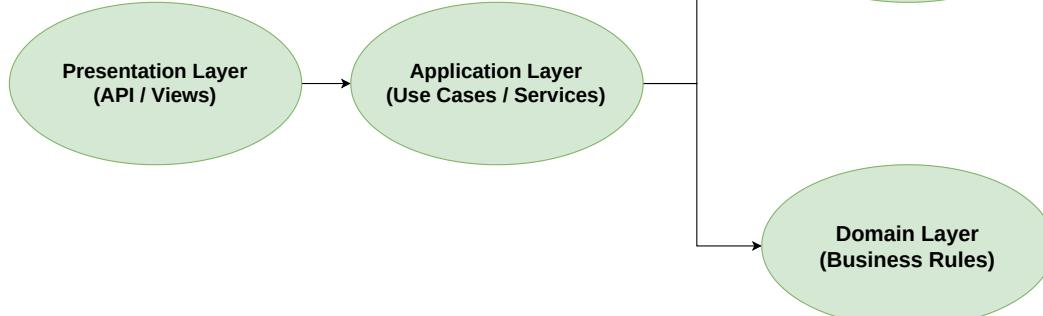
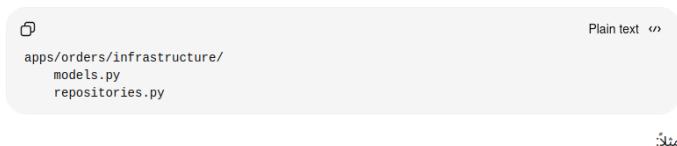


### Clean Architecture

- 1 Presentation Layer (API / Views)
- 2 Application Layer (Use Cases / Services)
- 3 Domain Layer (Business Rules)
- 4 Infrastructure Layer (Django ORM / Database / External Services)

## Infrastructure Layer 4

اینجا اتصال به دیتابیس، ORM و فایل سیستم است.



```

accounts/
    |
    +-- models.py
        |
        +-- User
        |
        +-- Profile
    |
    +-- serializers.py
    |
    +-- views.py
    |
    +-- permissions.py

```

#### UserType

- 1- regular
- 2- admin
- 3- superuser

User				
	<u>Column</u>	<u>Type</u>	<u>Validation</u>	<u>Description</u>
PK	id	BigInt	auto increment	
	phone	phone field	as a user name	
	password			
	type	IntegerField		
	is_active	BooleanField		
	is_staff	BooleanField		
	created_date	DatetimeField		
	updated_date	DatetimeField		

#### Profile

	<u>Column</u>	<u>Type</u>	<u>Validation</u>	<u>Description</u>
PK	id	BigInt	auto increment	
FK	user	OneToOne		
	first_name	CharField	250 char	
	last_name	CharField	250 char	
	email	email field	regex for iranian number	
	image	ImageField		
	created_date	DatetimeField		
	updated_date	DatetimeField		

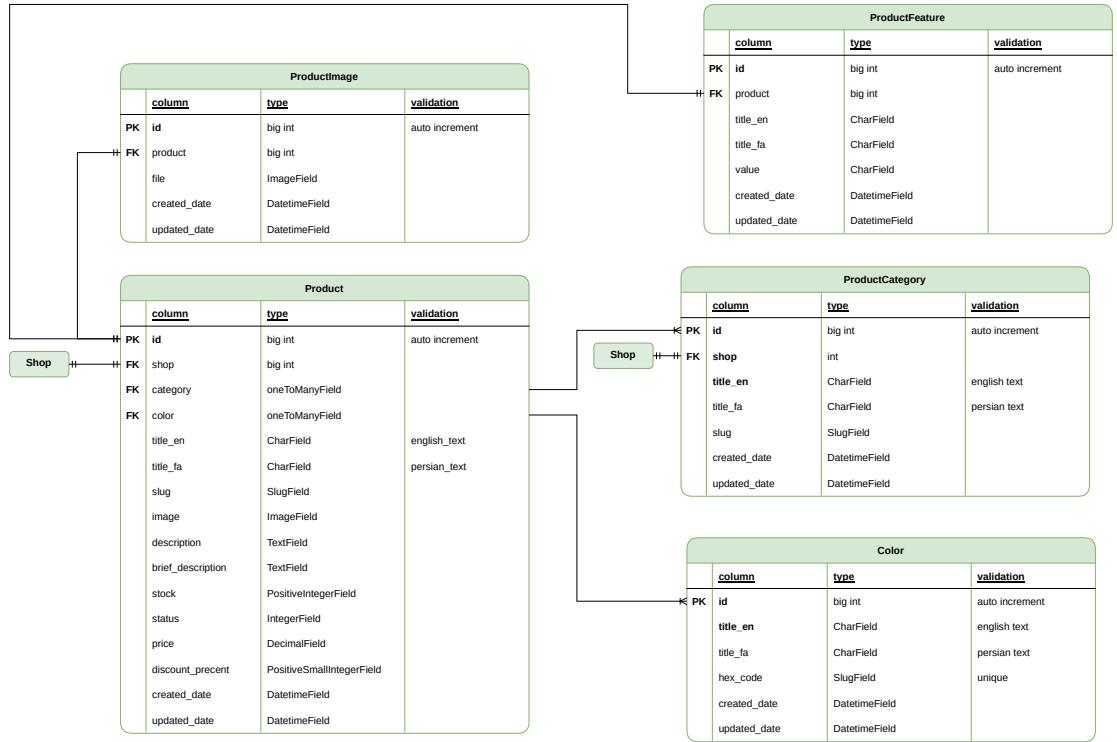
```
shops/  
  └── models.py  
      └── Shop
```

```
services.py # منطق مربوط به اشتراک
```

Profile

Shop				
	<u>Column</u>	<u>Type</u>	<u>Validation</u>	<u>Description</u>
PK	id	BigInt	auto increment	
	name_en	CharField	english text	
	name_fa	CharField	persian text	
FK	owner	OneToOne		
	instagram_id	CharField	unique	
	telegram_channel_id	CharField	unique	
	is_active	BooleanField		
	created_date	DatetimeField		
	updated_date	DatetimeField		

products/  
 └── models.py  
 └── Product  
 └── Color  
 └── ProductCategory  
 └── ProductImage  
 └── ProductFeature  
 └── services.py # مدیریت موجودی



customers/  
└── models.py  
    └── Customer  
        └── Address

Note:  
هدف این است که هر فروشنگ‌گاهی برای هر مشتری یک بروگایل داشته باشد. حال ممکن است این مشتری با همین شماره در فروشگاه دیگری هم بروگایل داشته باشد و در عین حال در آب ما هم آکانت داشته باشد. بعضی سه بروگایل متفاوت داشته باشد

Customer				
	Column	Type	Validation	Description
PK	id	BigInt	auto increment	
	phone		unique for every shops	
FK	shop	OneToMany		
Profile	profile	OneToOne	null and blank True	
	first_name	CharField	250 char	
	last_name	CharField	250 char	
	email	email field	optional	
	created_date	DatetimeField		
	updated_date	DatetimeField		

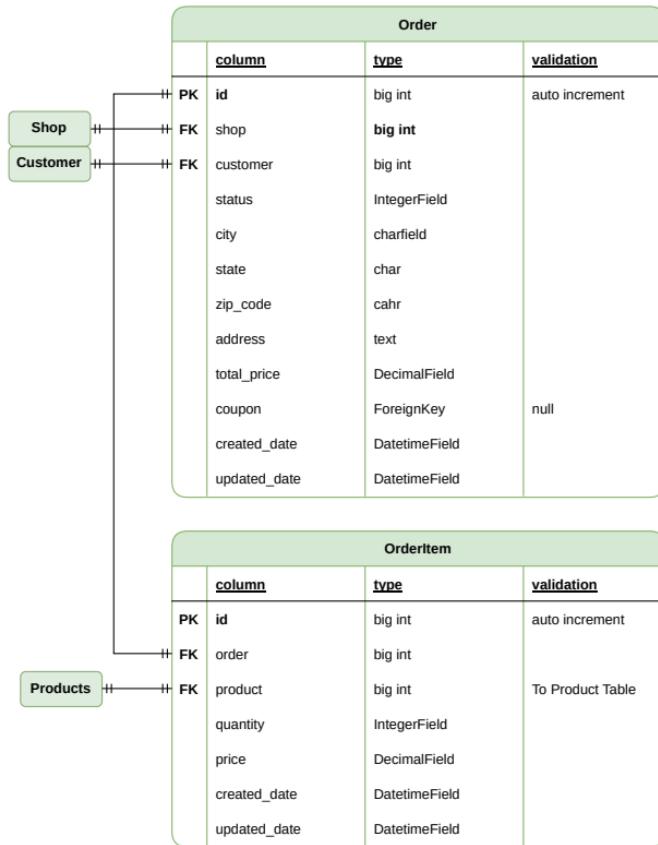
Address				
	Column	Type	Validation	Description
PK	id	BigInt	auto increment	
FK	customer	OneToOne		
	state	CharField	250 char	
	city	CharField	250 char	
	address	CharField	regex for iranian number	
	zip_code	CharField	regex for iranian zip code	
	receiver.phone			
	created_date	DatetimeField		
	updated_date	DatetimeField		

```
sellers/
└── models.py
    └── Seller
```

Seller					
	<u>Column</u>	<u>Type</u>	<u>Validation</u>	<u>Description</u>	
Shop	PK id	BigInt	auto increment		
Profile	FK shop	OneToMany			
	FK profile	OneToOne	null and blank True		

orders/

- models.py
  - Order
  - OrderItem
- services.py # منطق ثبت سفارش
- selectors.py # کوئری های پیچیده



```

payments/
└── models.py
    └── Payment
        └── OnlinePayment
        └── CashPayment
└── services.py # تایید برد پرداخت

```

