In-depth Design

Abstract; implemented;
Property

Inventory

Self._items: Dict {item_id: quantity}
Self._catalog: Dict{item_id:StoreItem}

Set_catalog(catalog)
Get_catalog()
Add_item(item_id, quantity)
Remove_item(item_id)
Update_quantity(item_id, quantity)
Get_quantity(item_id)
Search_items(name, category, min_price, max_price)

@property
Items()

StoreItem(ABC)

Self._item_id: int
Self._title: str
Self._price: float
Self._height: int
Self._width: int
Self._weight: float
Self._description: str

get_description()
apply_discount(discount)
__repr__()

@property
item_id()
title()
price()
description()

Table(StoreItem)

get_description()

Bed(StoreItem)

self._pillow_count: int

get_description()

Closet(StoreItem)

self._with_mirror: bool

get_description()

Chair(StoreItem)

self._material: str

get_description()

Sofa(StoreItem)

self._seating_capacity: int

get_description()

ShoppingCart

Self._inventory: Inventory
Self._cart_items: Dict{item_id : quantity}
Self._total_price: float

Add_furniture(item_id, quantity)
Remove_furniture(item_id, quantity)
Show_total_price()
Apply_discount(discount_percentage)
Get_item_by_id(item_id, catalog)

User

Self._username: str
Self._full_name: str
Self._email: str
Self._password: str
Self._address: str
Self._phone_number: str
Self._is_logged: bool
Self._order_hist: List[Order]

_hash_password(password)
verify_password(password)
sign_up()
login(email, password)
manage_profile(full_name, address, phone_number)
view_order_history()
add_order(order)
__repr__()

@property
username()
email()
full_name()

UserOrderDictionary (Observer Pattern)

Self._user_orders: Dict[str, List[Order]]

update(order: Order)
get_orders_for_user(user: User)
clear_user_orders(user: User)

Order

self._user: User
Self._itms: List(StoreItem)
Self._total_price: float
Self._status: str

@property
user()
items()
total_price()

update_status(new_status)
__repr__()

MainStore

Main()

1. Initialize inventory, shopping cart and users
2. authenticate user (login/signup)
3. then execute actions using API – shoppingCart interaction
4. Checkout and order processing

API

Get_items()
Get_item()
Post_register_user()
Post_login_user()
Put_update_user_profile()
Get_user_profile()
Get_all_orders()
Post_create_order()
Post_add_item_to_cart()
Delete_remove_item_from_cart()
Put_update_inventory_item()
Delete_remove_inventory_item()
Post_apply_cart_discount()
Post_checkout()